



Norwegian University of Life Sciences
Faculty of Science and Technology

Philosophiae Doctor (PhD)
Thesis 2023:34

Data- and Expert-driven Feature Selection for Predictive Models in Healthcare - Towards Increased Interpretability in Underdetermined Machine Learning Problems

Data- og ekspertdrevet variabelseleksjon
for prediktive modeller i helsevesenet
- Mot økt tolkbarhet i underbestemte
maskinlæringsproblemer

Anna Selina Jenul

Data- and Expert-driven Feature Selection for Predictive Models in Healthcare

Towards Increased Interpretability in Underdetermined
Machine Learning Problems

Data- og ekspertdrevet variabelseleksjon for prediktive
modeller i helsevesenet

Mot økt tolkbarhet i underbestemte maskinlæringsproblemer

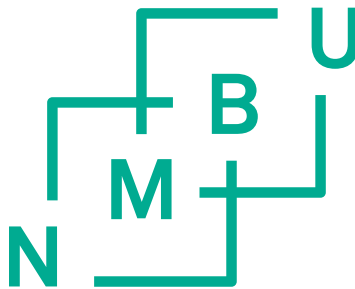
Philosophiae Doctor (PhD) Thesis

Anna Jenul

Norwegian University of Life Sciences

Faculty of Science and Technology

Ås, 2023



Thesis number: 2023:34
ISSN: 1894-6402
ISBN: 978-82-575-2062-5

Abstract

Modern data acquisition techniques in healthcare generate large collections of data from multiple sources, such as novel diagnosis and treatment methodologies. Some concrete examples are electronic healthcare record systems, genomics, and medical images. This leads to situations with often unstructured, high-dimensional heterogeneous patient cohort data where classical statistical methods may not be sufficient for optimal utilization of the data and informed decision-making. Instead, investigating such data structures with modern machine learning techniques promises to improve the understanding of patient health issues and may provide a better platform for informed decision-making by clinicians. Key requirements for this purpose include (a) sufficiently accurate predictions and (b) model interpretability. Achieving both aspects in parallel is difficult, particularly for datasets with few patients, which are common in the healthcare domain. In such cases, machine learning models encounter mathematically underdetermined systems and may overfit easily on the training data. An important approach to overcome this issue is feature selection, i.e., determining a subset of informative features from the original set of features with respect to the target variable. While potentially raising the predictive performance, feature selection fosters model interpretability by identifying a low number of relevant model parameters to better understand the underlying biological processes that lead to health issues.

Interpretability requires that feature selection is stable, i.e., small changes in the dataset do not lead to changes in the selected feature set. A concept to address instability is ensemble feature selection, i.e. the process of repeating the feature selection multiple times on subsets of samples of the original dataset and aggregating results in a meta-model. This thesis presents two approaches for ensemble feature selection, which are tailored towards high-dimensional data in healthcare: the Repeated Elastic Net Technique for feature selection (RENT) and the User-Guided Bayesian Framework for feature selection (UBayFS). While RENT is purely data-driven and builds upon elastic net regularized models, UBayFS is a general framework for ensembles with the capabilities to include expert knowledge in the feature selection process via prior weights and side constraints. A case study modeling the overall survival of cancer patients compares these novel feature selectors and demonstrates their potential in clinical practice.

Beyond the selection of single features, UBayFS also allows for selecting whole feature groups (feature blocks) that were acquired from multiple data sources, as those mentioned above. Importance quantification of such feature blocks plays a key role in tracing information about the target variable back to the acquisition modalities. Such information on feature block importance may lead to positive

effects on the use of human, technical, and financial resources if systematically integrated into the planning of patient treatment by excluding the acquisition of non-informative features. Since a generalization of feature importance measures to block importance is not trivial, this thesis also investigates and compares approaches for feature block importance rankings.

This thesis demonstrates that high-dimensional datasets from multiple data sources in the medical domain can be successfully tackled by the presented approaches for feature selection. Experimental evaluations demonstrate favorable properties of both predictive performance, stability, as well as interpretability of results, which carries a high potential for better data-driven decision support in clinical practice.

Sammendrag

Moderne datainnsamlingsteknikker i helsevesenet genererer store datamengder fra flere kilder, som for eksempel nye diagnose- og behandlingsmetoder. Noen konkrete eksempler er elektroniske helsejournalssystemer, genomikk og medisinske bilder. Slike pasientkohortdata er ofte ustrukturerte, høydimensjonale og heterogene og hvor klassiske statistiske metoder ikke er tilstrekkelige for optimal utnyttelse av dataene og god informasjonsbasert beslutningstaking. Derfor kan det være lovende å analysere slike datastrukturer ved bruk av moderne maskinlæringsteknikker for å øke forståelsen av pasientenes helseproblemer og for å gi klinikerne en bedre plattform for informasjonsbasert beslutningstaking. Sentrale krav til dette formålet inkluderer (a) tilstrekkelig nøyaktige prediksjoner og (b) modelltolkbarhet. Å oppnå begge aspektene samtidig er vanskelig, spesielt for datasett med få pasienter, noe som er vanlig for data i helsevesenet. I slike tilfeller må maskinlæringsmodeller håndtere matematisk underbestemte systemer og dette kan lett føre til at modellene overtilpasses treningsdataene. Variabelseleksjon er en viktig tilnærming for å håndtere dette ved å identifisere en undergruppe av informative variabler med hensyn til responsvariabelen. Samtidig som variabelseleksjonsmetoder kan lede til økt prediktiv ytelse, fremmes modelltolkbarhet ved å identifisere et lavt antall relevante modellparametere. Dette kan gi bedre forståelse av de underliggende biologiske prosessene som fører til helseproblemer.

Tolkbarhet krever at variabelseleksjonen er stabil, dvs. at små endringer i datasettet ikke fører til endringer i hvilke variabler som velges. Et konsept for å adressere ustabilitet er ensemblevariableseleksjon, dvs. prosessen med å gjenta variabelseleksjon flere ganger på en delmengde av prøvene i det originale datasett og aggregere resultater i en metamodel. Denne avhandlingen presenterer to tilnærminger for ensemblevariableseleksjon, som er skreddersydd for høydimensjonale data i helsevesenet: "Repeated Elastic Net Technique for feature selection" (RENT) og "User-Guided Bayesian Framework for feature selection" (UBayFS). Mens RENT er datadrevet og bygger på elastic net-regulariserte modeller, er UBayFS et generelt rammeverk for ensembler som muliggjør inkludering av ekspertkunnskap i variabelseleksjonsprosessen gjennom forhåndsbestemte vektorer og sidebegrensninger. En case-studie som modellerer overlevelsen av kreftpasienter sammenligner disse nye variabelseleksjonsmetodene og demonstrerer deres potensiale i klinisk praksis.

Utover valg av enkelte variabler gjør UBayFS det også mulig å velge blokker eller grupper av variabler som representerer de ulike datakildene som ble nevnt over. Kvantifisering av viktigheten av variabelgrupper spiller en nøkkelrolle for forståelsen av hvorvidt datakildene er viktige for responsvariabelen. Tilgang til slik informasjon kan føre til at bruken av menneskelige, tekniske og økonomiske

ressurser kan forbedres dersom informasjonen integreres systematisk i planleggingen av pasientbehandlingen. Slik kan man redusere innsamling av ikke-informative variabler. Siden generaliseringen av viktighet av variabelgrupper ikke er triviell, undersøkes og sammenlignes også tilnærminger for rangering av viktigheten til disse variabelgruppene.

Denne avhandlingen viser at høydimensjonale datasett fra flere datakilder fra det medisinske domenet effektivt kan håndteres ved bruk av variabelseleksjonsmetodene som er presentert i avhandlingen. Eksperimentene viser at disse kan ha positiv en effekt på både prediktiv ytelse, stabilitet og tolkbarhet av resultatene. Bruken av disse variabelseleksjonsmetodene bærer et stort potensiale for bedre datadrevet beslutningsstøtte i klinisk praksis.

Acknowledgement

During my time as PhD student at the Norwegian University of Life Sciences, I was supported by many people to whom I would like to express my gratitude.

First and foremost, I want to thank my supervisor Assoc. Prof. Oliver Tomic. Thank you very much for the excellent supervision of my thesis, support in technical and administrative matters, numerous fruitful discussions and inputs, and for always being there when I needed assistance of any kind. My research would not have been possible without you.

Furthermore, I am grateful to my co-supervisors Prof. Cecilia Marie Futsæther, Prof. Kristian Hovde Liland, and Prof. Ulf Geir Indahl, who contributed with interesting discussions and inputs to the various topics covered during my research. Special thanks also to my co-supervisor Prof. Jürgen Pilz from the University of Klagenfurt, who has been a valuable mentor already through my study times in Klagenfurt and contributed to my PhD research with his profound technical know-how and his excellent research network.

Moreover, I want to express gratitude to Henning Langen Stokmo, Mona-Elisabeth Revheim, and other collaborators from various research departments at Oslo University Hospital. Their valuable support as reliable partners and data providers, as well as their contributions with medical expertise on healthcare datasets was crucial for the success of my thesis.

In spring 2022, I conducted a research stay at the University of British Columbia in Vancouver, Canada. I am very grateful for this unique opportunity to collect valuable experience from both a professional and personal perspective. Thanks a lot to NMBU for providing the travel grant and to Prof. James Zidek, Prof. Will Welch, and Prof. Nancy Heckman from UBC for making the stay possible. It was an unforgettable experience.

Last but not least, I want to thank my family and friends for their never-ending support. My greatest thanks go to my partner Stefan who always supported me from a technical and mental side. Thank you for your immense support during the last three years!

Anna Jenul
Ås, 21.02.2023

Contents

| | |
|----------------------------------------------------------------|-------------|
| Abstract | i |
| Sammendrag | iii |
| Acknowledgement | v |
| Contents | viii |
| List of Papers | ix |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Challenges | 3 |
| 1.3 Research questions | 6 |
| 1.4 Structure of the thesis | 7 |
| 2 Background | 9 |
| 2.1 Machine learning and statistics in healthcare | 9 |
| 2.2 Feature selection in high-dimensional datasets | 11 |
| 2.2.1 State-of-the-art feature selectors | 12 |
| 2.2.2 Stability in feature selection | 15 |
| 2.3 Multi-source data analysis | 16 |
| 3 Methods | 19 |
| 3.1 Statistical fundamentals | 19 |
| 3.1.1 Linear regression | 20 |
| 3.1.2 Logistic regression | 21 |
| 3.1.3 Regularization | 21 |
| 3.1.4 Overview Bayesian statistics | 24 |
| 3.2 Data preprocessing | 26 |
| 3.2.1 Data scaling and transformations | 27 |
| 3.2.2 Outlier detection and handling of missing data | 29 |
| 3.2.3 Data encoding | 30 |

| | | |
|----------|------------------------------------------------|------------|
| 3.3 | Outcome prediction models | 30 |
| 3.3.1 | k NN regression and classification | 31 |
| 3.3.2 | Decision trees | 31 |
| 3.3.3 | Artificial Neural Networks | 32 |
| 3.4 | Performance metrics | 34 |
| 3.4.1 | Classification | 35 |
| 3.4.2 | Regression | 36 |
| 4 | Paper Summaries | 39 |
| 4.1 | Paper I | 39 |
| 4.2 | Paper II | 41 |
| 4.3 | Paper III | 42 |
| 4.4 | Paper IV | 45 |
| 5 | Discussion & Conclusion | 47 |
| 5.1 | Research Question I | 47 |
| 5.2 | Research Question II | 49 |
| 5.3 | Research Question III | 50 |
| 5.4 | Outlook | 51 |
| | List of Abbreviations | 53 |
| | List of Figures | 55 |
| | Bibliography | 57 |
| A | Papers I & Ia | 67 |
| B | Papers II & IIa | 85 |
| C | Paper III | 119 |
| D | Paper IV | 133 |

List of Papers

Paper I

- [46] Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsaether, C.M., Tomic, O.: RENT—repeated elastic net technique for feature selection. *IEEE Access* **9**, 152333–152346 (2021)

Paper II

- [48] Jenul, A., Schrunner, S., Pilz, J., Tomic, O.: A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). *Machine Learning* **111**(10), 3897–3923 (2022)

Paper III

- [44] Jenul, A., Schrunner, S., Huynh, B.N., Helin, R., Futsaether, C.M., Liland, K.H., Tomic, O.: Ranking feature-block importance in artificial multiblock neural networks. In: *International Conference on Artificial Neural Networks*. pp. 163–175. Springer (2022)

Paper IV

- [50] Jenul, A., Stokmo, H.L., Schrunner, S., Revheim, M.E., Hjortland, G.O., Tomic, O.: Towards understanding the survival of patients with high-grade gastroenteropancreatic neuroendocrine neoplasms: An investigation of ensemble feature selection in the prediction of overall survival. *arXiv preprint arXiv:2302.10106* (2023)

Additional Papers

- [42] Jenul, A., Bhattarai, B., Liland, K.H., Jiao, L., Schrunner, S., Futsaether, C., Granmo, O.C., Tomic, O.: Component based pre-filtering of noisy data for improved Tsetlin machine modelling. In: 2022 International Symposium on the Tsetlin Machine (ISTM). pp. 57–64. IEEE (2022)
- [57] Kuras, A., Jenul, A., Brell, M., Burud, I.: Comparison of 2D and 3D semantic segmentation in urban areas using fused hyperspectral and lidar data. *Journal of Spectral Imaging* **11** (2022)
- [88] Schrunner, S., Scheiber, M., Jenul, A., Zernig, A., Kästner, A., Kern, R.: Machine learning based indicators to enhance process monitoring by pattern recognition. arXiv preprint arXiv:2103.13058 (2021)

Software Papers (Ia, IIa)

- [45] Jenul, A., Schrunner, S., Huynh, B.N., Tomic, O.: RENT: A Python package for repeated elastic net feature selection. *Journal of Open Source Software* **6**(63), 3323 (2021)
- [43] Jenul, A., Schrunner, S.: UBayFS: An R package for user guided feature selection. *Journal of Open Source Software* **8**(81), 4848 (2023)

Oral Presentations

- * Jenul, A.: Data science for treatment outcome prediction: towards interpretable models combining healthcare data from multiple sources. Women in Data Science Conference Villach, Austria (2021), technical vision talk
- * Jenul, A., Schrunner, S., Huynh, B.N., Helin, R., Futsaether, C.M., Liland, K.H., Tomic, O.: Ranking feature-block importance in artificial multiblock neural networks. In: International Conference on Artificial Neural Networks. pp. 163–175. Springer (2022)
- * Jenul, A., Schrunner, S., Pilz, J., Tomic, O.: A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). European Conference on Machine Learning (2022), journal track

Poster Presentations

- * Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsaether, C.M., Tomic, O.: RENT - repeated elastic net technique for feature selection. Geilo Winter School (2021)

Chapter 1

Introduction

1.1 Motivation

Data science is a rapidly growing field that has spread into various application domains within the last decades, including the medical and healthcare sector [26, 55, 65, 99]. Due to digitalization and the availability of high-end equipment, it is possible to collect data from multiple sources and use those to make informed decisions regarding patient diagnosis, and treatment [16, 59]. Such data sources can be the acquisition of medical images for tumor segmentation to plan radiation therapy, gene expression data, or basic blood values.

The application of machine learning techniques for outcome prediction and interpretation of patient data carries a high potential to enhance state-of-the-art procedures in the clinic [87]. Benefits include more precise diagnoses and decisions via automatized analysis of large datasets and personalized disease treatment [26]. On the one hand, treatment strategies can be transferred from one patient group to another patient group with similar clinical attributes. On the other hand, unsuccessful treatment strategies could be terminated and replaced with more efficient alternatives. Moreover, data science can help assess assumptions made by experts in the healthcare field regarding their statistical evidence, as well as provide new insights from detecting latent/hidden information in the data [28]. Although there are promising examples of successful implementation of data science in the field of healthcare, caution is required. Interpretation and use of poor models may have a big negative impact on a patient's treatment. A negative example is IBM's Watson supercomputer that recommended incorrect cancer treatments¹. For training the underlying model, the software relied only on synthetic cancer patients rather than real patients. Furthermore, only a few clinicians per cancer type consult the

¹<https://www.statnews.com/wp-content/uploads/2018/09/IBMs-Watson-recommended-unsafe-and-incorrect-cancer-treatments-STAT.pdf>, 14.01.2023

software development by giving their treatment recommendations. Overall, the supercomputer could not represent the real world, so its recommendations were unreliable.

The transition from acquisition to tabular data often results in high-dimensional data, especially when parameters from images or genes are extracted. In general, measurements are referred to as features. Features are variables that characterize and describe a patient. Imagine a linear regression model where features define the explanatory variables, such as a person's height or age. Thereby, the columns of a data matrix are features.

Features can be of various types. While many features represent dates such as the "date of diagnosis" or "start of treatment", features resulting from questionnaires are strings or categorical. Furthermore, the feature space of an object can be high-dimensional due to modern collecting techniques. Not all measurements are commonly conducted on each patient, leading to missing values in datasets. Imagine two patients, where one requires surgery and the other not – all features describing surgery or its effects are given only for one of the two patients.

Compared to the number of features, the number of patients in a study is often small. In this thesis, we refer to such datasets as short-wide datasets (fewer samples than features). From a mathematical perspective, datasets with a small number of samples (patients) and a high number of features can lead to underdetermined problems when training a machine learning model, i.e., model parameters are not unique. Therefore, the interpretation of such a model is less reliable. As a consequence, the application of machine learning methodology is not trivial and needs thorough validation. Furthermore, models may suffer from the curse of dimensionality, i.e., the higher the dimension, the harder the problem [107]. High-dimensionality in this thesis refers to underdetermined problems where the number of features is higher than the number of samples and cannot be compared to high-dimensionality in natural language processing or streaming data.

Many different component-based approaches, such as principal component analysis, exist to cope with the high-dimensionality of healthcare datasets. Even though the dimensionality decreases, those approaches transform the original features into another subspace, making feature interpretation difficult. Hence, it would be advantageous to remain in the space of the original features, provided that it is possible to remove redundant and non-informative features. If possible, feature selection, i.e., selecting a subset of original, explanatory features or variables containing relevant information from the dataset, is the better alternative to reduce dimensionality than transformation-based methods. Smaller models are more interpretable and less affected by the curse of dimensionality. With the side effect of noise removal and variance reduction, feature selection can improve machine learning outcome predictors in terms of performance and speed [32].

While many different feature selection approaches are available, from supervised to semi-supervised and unsupervised, from filter to wrapper and embedded methods,

one important criterion is the stability of the feature selection. In this context, stability means that when slightly changing the training data, the selected feature set should not vary much [78]. When adding new objects to a short-wide dataset, the selected feature set may change. Hence, two independent sample sets from the same data distribution, where only the number of samples varies, can lead to different results. One possible reason is collinearity between features, i.e., correlations between predictor features in the dataset. Therefore we take advantage of this by selecting features using an ensemble of feature selectors. In this way, we can gain insight into the stability of which features are selected and which are not. This strategy delivers a feature statistic, allowing for better stability of the feature set. We address this issue in papers I and II of this thesis by inventing the stable ensemble algorithms RENT [46], and UBayFS [48].

In addition, medical experts can provide prior knowledge of feature importance. It may be advantageous to include this knowledge in addition to the data-driven acquisition of feature selections as done in UBayFS.

Feature selection or importance ranking on multi-source data can be applied source-wise, as well. Paper III [44] introduces different strategies to quantify the importances of different sources in artificial neural networks.

1.2 Challenges

Training statistical machine learning models on a short-wide dataset causes mathematical issues when trying to invert the data matrix. The system is underdetermined, and optimization may lead to an infinite number of optimal solutions. Therefore, the system is unstable and neither reliable nor interpretable. Algebraically this can be explained by the example of ordinary least squares estimators of a linear model [34] (without intercept)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\mathbf{y} \in \mathbb{R}^m$ represents the target, $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a data matrix, $\boldsymbol{\beta} \in \mathbb{R}^n$ the data coefficients, and $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ is an error term. More information about linear models is provided in Section 3.

Definition 1.2.1 (Rank) *Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a data matrix. The row and column ranks of \mathbf{X} are the number of linearly independent rows and columns, respectively. The rank of \mathbf{X} is defined as the number of linearly independent columns, i.e., $\text{rank}(\mathbf{X}) \leq \min(m, n)$.*

Row- and column ranks are always equal. As a result of Definition 1.2.1, $\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{X}^T) = \text{rank}(\mathbf{X}^T \mathbf{X})$. Furthermore, \mathbf{X} has full rank when $\text{rank}(\mathbf{X}) = \min(m, n)$.

The ordinary least squares estimator for a linear model is given as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y},$$

where $\dim(\mathbf{X}^T \mathbf{X}) = n \times n$. In case that $m < n$, $\mathbf{X}^T \mathbf{X}$ does not have full rank, since $\text{rank}(\mathbf{X}^T \mathbf{X}) = \text{rank}(\mathbf{X}) \leq m < n$. A short-wide data matrix \mathbf{X} resulting from real-world data with lots of numerical measurements cannot have $n - \text{rank}(\mathbf{X})$ linearly independent columns, which results in the underdetermined system. Hence, there is no unique solution for $\hat{\beta}$. The instability hinders reliable interpretation of the model and the contributing features. Technically, two ways to resolve this issue exist:

1. To increase the sample size; this is not easy as the generation of reliable training samples is not easy. Especially in healthcare, the collection of new patients is limited to the number of people suffering from a disease, data protection regulations, and privacy issues.
2. To decrease the number of features with dimensionality reduction tools.

Especially in healthcare, the information to predict treatment outcomes accurately can be distributed over multiple inhomogeneous data sources, see Figure 1.1. One of the most common sources is clinical data, including measurements that can be easily taken from patients, such as blood values and disease stages. Frequently, patient histology is known, which reveals information about previous diseases and treatments of patients.

Additionally, a huge number of features can be extracted from different medical imaging modalities, i.e., MRI, CT, or PET images. By means of image processing software, texture features can be extracted and represented as tabular data. Quantitative features extracted from regions of interest in medical images are usually referred to as radiomics data [67, 104].

It is challenging to model inhomogeneous data sources jointly — with machine learning techniques, we aim to fill this gap.

Features can be of different types, making data sources inhomogeneous. While gene expression data are usually numerical, different disease stages or patients' answers to patient questionnaire evaluations can be in text format, ordinal categories, or nominal categories. When collecting dates from treatments, we face a lot of features describing dates - from which usable features must be calculated. Hence, different encoding strategies are necessary to make collected data suitable for machine learning algorithms.

Another relevant issue is missing data. Not all patients receive identical treatment or are measured with the same parameters. A person from whom no medical images are taken has much fewer features present than a person with images. In a perfect world with thousands of samples, we would omit the patients where any feature is missing, but considering the fact that we have only a limited number of patients available, this approach is not a viable option. Data imputation is a relevant approach when considering healthcare data but must be handled with great care, as imputation algorithms often reduce feature variances and distort the analysis [85].

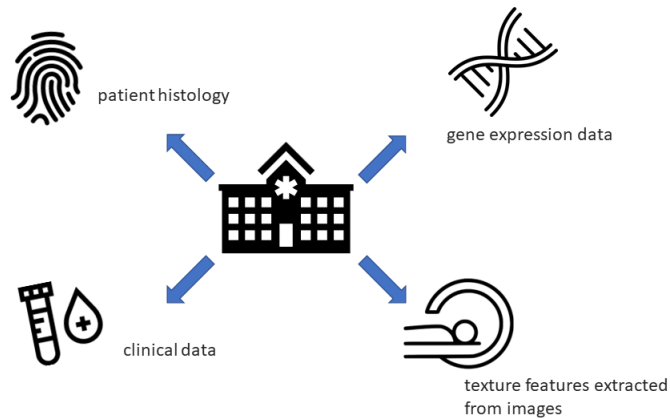


Figure 1.1: Examples of different data sources.

When predicting a patient’s overall survival, the target function is typically censored [83]. The follow-up status of each patient is measured for a certain period up to the maximum follow-up time. If the person dies within this period, the target has detailed information about how long the patient lived, and the patient’s lifetime is not censored. However, if a patient is still alive after the maximum follow-up time, the lifetime is right-censored. Nevertheless, patients can die within the follow-up period due to other causes, such as car accidents. The lifetime of such patients is right-censored, too, because we cannot follow up on the event of interest (death caused by the disease) anymore [60]. Hence, premature death due to other causes is not properly reflected in the dataset. Right censoring due to other events is called loss-to-follow-up censoring. It is essential to consider patient censoring to adequately preprocess a dataset for ML models, which are not targeted toward survival analysis [97].

Once the preprocessed dataset is available, it is challenging to choose an optimal strategy for analysis. Specifically, the data interpretation is only meaningful when we can be sure that the feature set is stable and delivers an accurate prediction of the target function when used for an ML model [51]. Unstable feature selection can result in the selection of irrelevant features and potentially erroneous interpretations and conclusions. Clinical experts usually know the dataset’s most important key parameters from previous or similar analyses. Hence, combining their expertise with data-driven methodology may lead to accurate and interpretable feature selection [69].

In addition, the selection of an appropriate machine learning model is crucial.

Healthcare data can be nonlinear [37], and often linear models are not sufficient to cover the full complexity of the problems. Deep learning models can model non-linear data efficiently but easily overfit when only few samples are available for training. Furthermore, the question of data-source importance arises — do we need all data sources, or are some redundant? This is an important issue as healthcare systems could reduce costs by skipping unimportant measurements for new patients knowing those sources provide non-informative data that may not contribute positively with regard to model performance. Furthermore, expert knowledge about feature importance is often available from previous analyses of similar datasets and scientific literature. This information should be incorporated into the analysis but many algorithms do not provide this option.

1.3 Research questions

This thesis aims to develop feature selection algorithms to detect and interpret the most important key parameters in high-dimensional datasets with potential block structures. We set a focus on healthcare datasets where the number of features often exceeds the number of samples. Not only do we consider single feature importances but also the importance of whole feature-blocks acquired from various sources. To ensure reliable results, the predictive performance of machine learning models, which relies on the selected features, has to be considered, as well. With these challenges at hand, the thesis aims to make contributions to the following research questions (RQs):

- RQ1 How can we perform feature selection with a low number of samples in high-dimensional datasets to support a good tradeoff between predictive performance and model interpretation?
- RQ2 How can we exploit background knowledge from experts in order to extend the capabilities of data-driven methods to make feature selection more interpretable and accurate?
- RQ3 How can the concept of block methods, i.e., considering each data source as a separate data block and considering the block structure when modeling, deliver additional information to improve data interpretation and outcome prediction?

Paper I [46] targets the first research question (RQ1) using a data-driven ensemble feature selection approach. Performing feature selection on distinct subsets of the training data delivers a statistic about selected features and feature importance. Compared to a single feature selection model, where a feature may be selected based on spurious correlations, ensemble models provide more stable and reliable features, as we can build statistics such as the count of how often a feature is selected.

The second paper [48] extends the data-driven approach by including prior knowl-

edge directly in the feature selection process, contributing to RQ2. With the help of Bayesian methodology, we build a model where we combine data-driven feature selection with domain expertise.

Paper III [44] introduces different strategies for quantifying block importances in a neural network architecture. This approach enables the importance ranking of different data sources contributing to the model.

Paper IV [50] shows a real-world application of the invented ensemble feature selectors on a cancer dataset. Feature selection with and without prior knowledge is compared, and the corresponding feature sets are analyzed from a data-scientific and clinical perspective.

1.4 Structure of the thesis

Chapter 2 describes different machine learning applications in healthcare. The special focus lies on feature selection methodology and stability of feature selection. Chapter 3 introduces the machine learning methodology that is used throughout the thesis. The topics include regularization, data preprocessing, outcome prediction models, and outcome prediction metrics. In Chapter 4, we present outlines of the research papers of this thesis. Finally, Chapter 5 summarizes and discusses methods and main results presented in this thesis. Furthermore, an outlook on possible future work is provided.

Chapter 2

Background

2.1 Machine learning and statistics in healthcare

The applications of machine learning algorithms in healthcare are vast, including, amongst others, medical image analysis, decision support, and personalized treatment.

Widely used for clinical trials, statistical hypothesis testing is the most traditional way of using statistics in medical applications. Hypothesis testing determines whether certain parameters are influential or whether a test group differs from a control group based on a certain probability [82]. The Null-hypothesis, e.g., medication has the same effect on the test, and the control group is tested versus an alternative hypothesis, stating that the influence is not equal. The decision depends on a significance level and the corresponding p -value. Typical tests include versions of Student's t -test, or analysis of variance (ANOVA). Classical statistics and hypothesis tests cannot be necessarily applied to all types of data that are produced today. More advanced tasks like handling a high number of input features or the delineation of tumorous tissue require more advanced methods from the field of machine learning.

With medical image analysis, tumor delineation of cancerous images from different imaging techniques such as MRI, CT, or PET is done with the help of machine learning models, including shallow learning methods, deep learning techniques, or different thresholding strategies [12, 31, 79, 80]. Convolutional neural networks can detect patterns in medical images, and with enough training data, algorithms find tumors on new images with high accuracy [64, 86, 108]. Prominent architectures for image segmentation tasks are Ronneberger's U-net [24], a 2D segmentation technique, and different 3D extensions [20, 73], which can be beneficial as medical images consist of multiple slices and are hence considered 3-dimensional. The challenges of medical image segmentation are limited annotated data, model over-

fitting, and excessive training times [36]. For the first issue, data augmentation and transfer learning are common approaches. It is necessary to adjust the network architecture to prevent overfitting and to limit the computational complexity. Furthermore, the target organ, i.e., the region that needs to be identified and delineated, varies in size, position, and shape from patient to patient, making accurate delineations even more challenging.

Another prominent field within medical image analysis is the extraction of radiomic features from medical images to support informed decision-making [93, 101]. Radiomics are different parameters, such as image textures, assuming that features can quantify tumor characteristics as tumorous tissue differs from surrounding tissue. Usually, radiomic features are high-dimensional, including, amongst others, first-order statistics and shape-based 2D and 3D parameters [104]. As the number of extracted radiomic features is user-defined and ranges from a few to thousands, feature selection is essential since they are used for outcome prediction or treatment selection [13, 105].

Outcome prediction on healthcare data is another important topic. Due to the increasing number of measurements, information from data can be extracted, and accurate models can be trained. For example, assume we aim to predict the survival time of a patient (output) based on their clinical measurements (input). Such models may provide valuable insight into what input features can lead to longer survival. Underlying algorithms are linear/non-linear, adjustable for regression and classification tasks, and supervised/semi-supervised or even unsupervised [5, 27, 91]. Treatment outcome prediction has to be performed with low uncertainty as wrong outputs can be fraught with consequences for patients [90].

Machine learning for personalized treatment selection is another important pillar of healthcare data science, as it is difficult to draw general conclusions for a patient cohort. It is an evolving field with lots of potential but also many challenges [26]. Since medication in personalized treatment is adapted to the patient, a better effect can be reached for individuals, and the negative side effects of wrong treatments can be reduced. Furthermore, offering more patient-tailored medication might lead to cost savings and a better experimental design for new medication. On the other hand, healthcare data are complex, non-linear and noisy, making modeling challenging. Furthermore, there might be information leaking in the data, which directly affects the predictive performance. Small patient cohorts and latent variables, such as social and environmental influences, do not make personalized treatment decisions and the interpretation of machine learning easier. A relevant future perspective of personalized treatment is causal inference modeling, for example, causal Bayesian networks [8].

Targets such as overall survival or time to disease recurrence are (right-)censored, meaning patients are not further tracked after a maximal follow-up time is reached. Hence, the target informs us about events before the maximal follow-up time rather than about the time after. Imagine two patients where one dies a few days after

the maximal follow-up time, and the other is still alive after several years. From a pure data perspective, they are handled equally even though their treatment outcomes differ significantly. Another issue originates from patients that leave the study such that the practitioners do not know how the disease proceeded. Commonly such events are modeled with statistical lifetime/survival models such as Cox-Regression [116] or Kaplan-Meier models [21].

Recently, the use of machine learning methodology for survival analysis has gained more interest [98, 109]. The issue with high-dimensional data is present, which is why combining survival models with feature selection is an interesting research topic [96]. Even though survival analysis is not the focus of this thesis, research on feature selection of high-dimensional data for survival modeling would be a natural future research area.

2.2 Feature selection in high-dimensional datasets

Dimensionality reduction is the process of reducing the feature space of a dataset by different algorithms or transformation approaches. Especially component-based methods are standard linear approaches to reduce the dimensionality of a dataset [66, 94]. Component-based methods find a lot of applications in the field of chemometrics [56]. The most basic method is Principal Component Analysis (PCA) [2], where the dataset is represented in a vector space that is spanned by its eigenvectors. The new features, known as components, are linear combinations of the original features. PCA relies on a variance decomposition where the amount of represented variance in the data decreases with the increasing number of components. By containing the main systematic variation in the first few components, PCA can be considered a noise-filtering method, as the remaining components contain mainly noise. In this way, only the first few components need to be considered for interpretation (instead of many original features) whilst the noise removal makes interpretation easier [14]. Being completely unsupervised, PCA is not ideal for performing outcome prediction. In other terms, the data dimensionality can be reduced, but a separate prediction model must be trained on top. Principal component regression (PCR) is a well-established method that takes this approach. First, we extract principal components with PCA and then train a linear regression model with those components as input and the given target as dependent variable [74]. Another supervised approach is called Partial Least Squares Regression (PLSR) [1], where the target and the data are decomposed together, adjusting the model to both input and output. Unsupervised and supervised extensions of component-based approaches for multiple data sources exist, such as Multiple Factor Analysis (MFA) [3], Sequential and Orthogonalized Partial Least Squares Regression (SO-PLS regression) [75], or the Response Oriented Sequential Alternation (ROSA) [63]. Those methods are capable of reducing dimensionality while accounting for block structures. Furthermore, the output can be multivariate as well. Still, the components are combinations of the original features. what hampers the interpretation of original features if a high number of

variables is present in the data.

A non-linear unsupervised alternative to PCA for dimensionality reduction is UMAP - Uniform Manifold Approximation and Projection [70], which models the manifold with a fuzzy topological structure.

Feature selection has become a prominent research field with the aim of developing algorithms where the number of variables can be reduced to make data interpretation easier, remove noise, and make the training of machine learning algorithms faster and more resource efficient. Different approaches can be used either to clean the data on a low level, i.e., remove features that represent only noise and keep the rest, or on a broad level where only the most relevant features shall be kept. Various feature selectors exist for supervised, unsupervised, or semi-supervised tasks [11, 72, 92, 95]. Figure 2.1 illustrates how feature selection works intuitively.

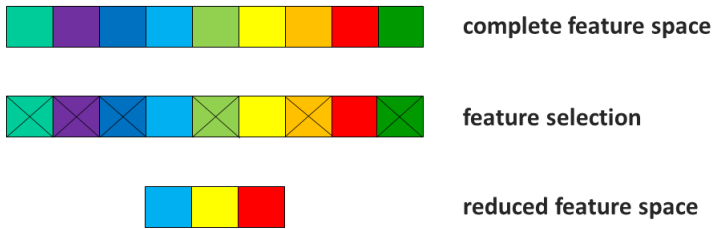


Figure 2.1: In feature selection, the complete feature space is reduced to a smaller set of relevant features. Redundant features are removed.

2.2.1 State-of-the-art feature selectors

Basically, feature selection techniques have been divided into three groups, *filter*, *wrapper*, and *embedded* methods [32]. Filter approaches are independent of a machine learning algorithm. They rely on different statistical measurement criteria, such as correlation coefficients between features, correlation coefficients between features and targets, or mutual information metrics. Usually, the top- k features are evaluated with the statistical criterion.

Three established filter approaches are the *Laplacian score*, *Fisher score*, and the *minimum Redundancy Maximal Relevance criterion* (mRMR).

The Laplacian score [35] relies on Laplacian eigenmaps. The main concept and algorithm is described here - for more detailed information, see [7]. Basically, we compute a k nearest neighbor (k NN) graph of the samples. Let \mathbf{s}_i , and \mathbf{s}_j be two samples (rows of the data matrix \mathbf{X}), $i, j \in \{1, \dots, m\}$, and $\mathbf{S} \in \mathbb{R}^{m \times m}$ be a weight matrix describing the local structure of the data space. If \mathbf{s}_i and \mathbf{s}_j are neighbors, then $S_{i,j} = \exp(-\frac{\|\mathbf{s}_i - \mathbf{s}_j\|^2}{\gamma})$, where γ is a constant; otherwise $S_{i,j} = 0$. An unweighted, binarized version of \mathbf{S} would be equivalent to the adjacency matrix. Furthermore,

$$\begin{aligned} \mathbf{1}_m &= \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{m \times 1} \\ \mathbf{D} &= \text{diag}(\mathbf{S} \cdot \mathbf{1}_m) \\ \mathbf{L} &= \mathbf{D} - \mathbf{S}, \end{aligned}$$

where \mathbf{D} is used to center \mathbf{S} , resulting in the graph Laplacian matrix \mathbf{L} . We define for feature $r \in \{1, \dots, n\}$, let

$$\tilde{\mathbf{x}}_r = \mathbf{x}_r - \frac{\mathbf{x}_r^T \mathbf{D} \mathbf{1}_m}{\mathbf{1}_m^T \mathbf{D} \mathbf{1}_m} \mathbf{1}_m$$

Then the Laplacian score \mathcal{L} is defined as

$$\mathcal{L}_r = \frac{\tilde{\mathbf{x}}_r^T \mathbf{L} \tilde{\mathbf{x}}_r}{\tilde{\mathbf{x}}_r^T \mathbf{D} \tilde{\mathbf{x}}_r}$$

The Laplacian score can be used for unsupervised and supervised feature selection. The label information can be included in the graph structure for supervised use cases.

The Fisher score [35] is a supervised filter method for classification problems. Let $y_i \in \{1, \dots, c\}$ denote a categorical target variable of sample $i \in \{1, \dots, m\}$, where c defines the number of distinct classes. Further, let $\boldsymbol{\mu}^{(r)} \in \mathbb{R}^c$, $\boldsymbol{\sigma}^{(r)} \in \mathbb{R}^c$, and $\boldsymbol{\nu}^{(r)} \in \mathbb{R}^c$ denote the mean, standard deviation, and number of samples in each class with respect to feature r , respectively, and let $\mu^{(r)}$ denote the mean of feature r across all samples of the dataset. The Fisher score for feature r is then defined as

$$\mathcal{F}_r = \frac{(\boldsymbol{\nu}^{(r)})^T (\boldsymbol{\mu}^{(r)} - \mu^{(r)} \mathbf{1}_c)^2}{(\boldsymbol{\nu}^{(r)})^T (\boldsymbol{\sigma}^{(r)})^2},$$

where exponents are applied element-wise to the vectors.

The minimum Redundancy Maximal Relevance criterion (mRMR) [81] can be used for regression and classification tasks. It relies on the concept of mutual information I , which is defined for two random variables X_1 and X_2 as follows:

$$I(X_1, X_2) = \sum_{x_1 \in \mathcal{S}(X_1)} \sum_{x_2 \in \mathcal{S}(X_2)} p_{X_1, X_2}(x_1, x_2) \cdot \log \frac{p_{X_1, X_2}(x_1, x_2)}{p_{X_1}(x_1) \cdot p_{X_2}(x_2)},$$

where p_{X_1, X_2} denotes the joint density function of X_1 and X_2 , p_{X_1}, p_{X_2} denote the marginal density functions of X_1 , and X_2 , respectively, and $\mathcal{S}(\cdot)$ denotes the support of a random variable. Mutual information quantifies the amount of information obtained about X_2 through observing X_1 , and vice versa. If X_1 and X_2 are independent, the mutual information is 0.

As the name suggests, mRMR combines the maximal relevance and the minimal redundancy criterion. Assuming that $\delta \subseteq \{1, \dots, n\}$ is a feature set, the maximal relevance criterion aims to maximize the following expression

$$\max_{\delta} I(\{\mathbf{x}_i, i = 1, \dots, n\}, \mathbf{y})$$

Hence, the maximal relevance criterion tries to find a feature set that jointly has the largest dependence on the target \mathbf{y} . Unfortunately, the criterion is prone to selecting many redundant variables. Therefore, the minimum redundancy criterion comes into play. The minimum redundancy criterion minimizes the expression

$$\min_{\delta} \frac{1}{|\delta|^2} \sum_{i, j \in \delta} I(\mathbf{x}_i, \mathbf{x}_j)$$

The mRMR criterion maximizes the difference between the maximal relevance and the minimal redundancy criteria, i.e.,

$$\max_{\delta} I(\{\mathbf{x}_i, i = 1, \dots, n\}, \mathbf{y}) - \frac{1}{|\delta|^2} \sum_{i, j \in \delta} I(\mathbf{x}_i, \mathbf{x}_j)$$

Filter methods are fast compared to the wrapper and embedded approaches but neglect performance, as no ML model is trained [32]. Wrapper techniques are often search-based optimization algorithms like forward selection, backward elimination, or genetic algorithms. Different feature sets are selected in each iteration and evaluated via a "fitness" function. The feature set is increased/decreased by some criterion in the next iteration and re-validated.

For example, consider a forward selection based on a linear regression model. In the first step, we build a linear regression model with each of the n features separately, i.e., n models with only one covariate. The fitting of a linear model happens under the Null-hypothesis that feature coefficients are zero. Hence, the first and

”best” feature that is selected is the one with the smallest p -values. The evaluation of the p -value represents the fitness function. In the second step, we fix the first selected feature and fit $n - 1$ models where we combine the first feature with every other feature. Again, the feature combination with the lowest p -value is chosen. This procedure continues until we reach the significance level.

Backward elimination with linear regression models is similar. We start with the full feature set and recursively remove features. The genetic algorithm is a well-established discrete optimization procedure with binary encoding, i.e., 1 if a feature is selected and 0 if not. Hence, it can easily be used for feature selection. A description of the genetic algorithm is provided in Paper II.

Unfortunately, wrapper approaches are hard to apply for high-dimensional datasets as they easily overfit and are time-consuming.

A good compromise is to use embedded feature selectors, i.e., machine learning models that integrate the feature selection directly, such as Lasso regression, elastic net regression or decision trees, see Section 3. Extensions and combinations of the different techniques exist. Especially filter and wrapper methods are often applied in a combined framework. Those types are called *hybrid* feature selectors [39].

A term related to feature selection is *feature importance ranking* [38, 110]. While feature selection reduces the total number of features, feature importance ranking quantifies how much each feature contributes to a learning algorithm.

2.2.2 Stability in feature selection

A selected feature set is stable if minor changes in the dataset or the initialization of the feature selection algorithm do not lead to a huge variation of the selected features. Otherwise, we speak of an instable feature selector [51].

To compensate for the lack of stability, ensemble approaches have become of interest [9, 10]. Based on the principle of ensemble learning, where we assume that computing multiple models is better than one, the concept is transferred to feature selection. Hence, we apply the same feature selector to either the same data with different model initialization or to different subsets of the data. Assuming that we have K different models, we receive a feature set $\delta_k, k = 1, \dots, K$ for each model. The use of distinct algorithms is possible but not recommended for high-dimensional datasets [89]. With multiple feature sets, we can apply a function to $\delta_1, \dots, \delta_K$ to investigate the common information of the feature selectors. An easy example of ensemble feature selection is counting how often each feature is selected across all feature sets δ_k , i.e., the final feature set is determined as

$$\delta^* = \left\{ i \in \{1, \dots, n\} : \frac{1}{K} |\{k : i \in \delta_k\}| \geq t \right\},$$

where t defines the threshold indicating a minimum frequency of feature i in the

ensemble. With this counting strategy, the final feature set δ^* is less prone to contain unimportant features and more stable than each individual feature selector δ_k [71].

Similarity-based measures are the conventional way of measuring the stability of feature selectors. Assuming that a dataset has n features, we perform K distinct feature selection runs. Let $\mathcal{Z} = \{\delta_1, \dots, \delta_K\}$ be the different feature sets. Furthermore, Φ is the stability function. The similarity-based stability measure is defined as

$$\Phi_{\text{sim}}(\mathcal{Z}) = \frac{1}{K(K-1)} \sum_{i=1}^K \sum_{\substack{j=1 \\ j \neq i}}^K \phi(\delta_i, \delta_j), \quad (2.1)$$

where ϕ is a similarity measure, such as Hamming distance or the Jaccard index. The higher $\Phi_{\text{sim}}(\mathcal{Z})$, the more stable is the feature selector.

Nogueira et al. [78] propose five desirable properties of a stability measure, where the similarity-based approach does not fulfill all of them. Therefore, the authors propose a new measure. For this purpose, we denote $\bar{k} = \frac{1}{K} \sum_{i=1}^K |\delta_k|$ as the average number of selected features, and $f_j = \frac{1}{K} |\{k \in \{1, \dots, K\} : j \in \delta_k\}|$ as the average number of features sets containing feature j . Then, the proposed stability measure $\Phi_{\text{Nog}}(\mathcal{Z})$ is defined as

$$\Phi_{\text{Nog}}(\mathcal{Z}) = 1 - \frac{\frac{1}{n} \sum_{j=1}^n \frac{K}{K-1} f_j (1 - f_j)}{\frac{\bar{k}}{n} \left(1 - \frac{\bar{k}}{n}\right)}. \quad (2.2)$$

Φ_{Nog} fulfills all desirable properties presented in [78]. A perfectly stable feature selector has $\Phi_{\text{Nog}} = 1$. The lower bound of Φ_{Nog} is $-\frac{1}{K-1}$.

2.3 Multi-source data analysis

In this thesis, multi-source data describes the same cohort of samples with multiple distinct feature sets, e.g., image features, genetic features, or clinical features. Multi-source data can be homogeneous or heterogeneous, where the feature sets provide complementary information to characterize samples [62]. Due to the distinct perspectives from which we observe the sample cohort, we can sustain a good understanding of the data and push machine learning models in performance.

Nevertheless, multi-source data are not trivial to model and need good preparation, data arrangement, and approaches from data scientists. In [77], the authors

propose to split the modeling of multi-source data into three types: early, intermediate, and late integration methods. While in early integration, the distinct data sources get concatenated right before modeling, late integration trains a separate model for each source first. It combines the outputs to make a final prediction. Intermediate models combine information from data sources at one point inside the learning algorithm. Even though early integration by concatenating features is the most intuitive approach and can improve outcome prediction, information may get lost due to the different data source characteristics and scales. The variance in each data source may be different, which may affect algorithms. Multi-source data appear in various formats, including categorical features, text features, and image features. Combining and scaling data can remove internal structure information, e.g., local information on image data. Support vector machines and Lasso regularized models are the most used methodologies for modeling concatenated features. Especially as ensemble learners, regularized models can deliver stable results, also in terms of feature selection. Group lasso is a regularized linear regression on data with underlying group structure (i.e., sources) [25]. It incorporates a group-wise regularization penalty in the loss function. Depending on the regularization parameters, an entire group of coefficients may become zero during optimization. Hence, it can be a good choice for multi-source data [18].

In [62], the authors propose the efficiency of Bayesian networks and tree-based models for multi-source data. Bayesian networks naturally model different data structures and simultaneously include prior knowledge. Decision trees can be used as early or late integration. They do not require data scaling or transformation as they follow a rule-based approach. Hence, decision trees preserve the local structure of all data sources. Furthermore, kernel-based models and artificial neural networks where data sources are modeled separately and merged in a later step are introduced. Regularized, Bayesian and tree-based models serve as embedded feature selectors, as well, making them even more valuable for multi-source data [112].

Chapter 3

Methods

This section addresses a general overview of the methodology used in this thesis, including model regularization, data preprocessing, outcome prediction models, and performance metrics. We focus on short-wide datasets.

3.1 Statistical fundamentals

As one main pillar of data science, statistics offers many useful methods that can be applied to various use cases. We also use Bayesian statistics in this thesis, which is why the fundamentals are described in this section as well. The two main concepts, a) linear regression, which is the basis of many statistical modeling approaches, and b) the foundations of Bayesian statistics, will be introduced.

Two essential terms in machine learning affecting the performance of outcome prediction models are overfitting, and underfitting [85]. Overfitting occurs when the model learns patterns/details in the training data that may not be relevant or present in the testing data. When fitting the training data exactly, the model starts incorporating noise, is poorly generalizable, and hence, becomes unreliable. Especially when the number of covariates largely exceeds the number of samples, the model might give a perfect fit on the training data. Still, due to almost zero variation, the prediction of the testing data will be inaccurate. Models that overfit have a low bias but high variance. Approaches to prevent overfitting are regularization, feature selection, or collecting more training data.

On the other hand, a model underfits if it cannot capture the trend or the data structure well enough. Such models have a low variance but high bias. For example, underfitting occurs if we try to fit non-linear data with a linear model. Underfitting can be resolved by collecting more features via feature extraction or by using an ML algorithm that is capable of capturing the non-linear nature of

the dataset.

To determine an appropriate machine learning model, we aim to optimize the tradeoff between bias and variance — both should be low. An illustration of overfitting, ideal fitting, and underfitting is shown in Figure 3.1.

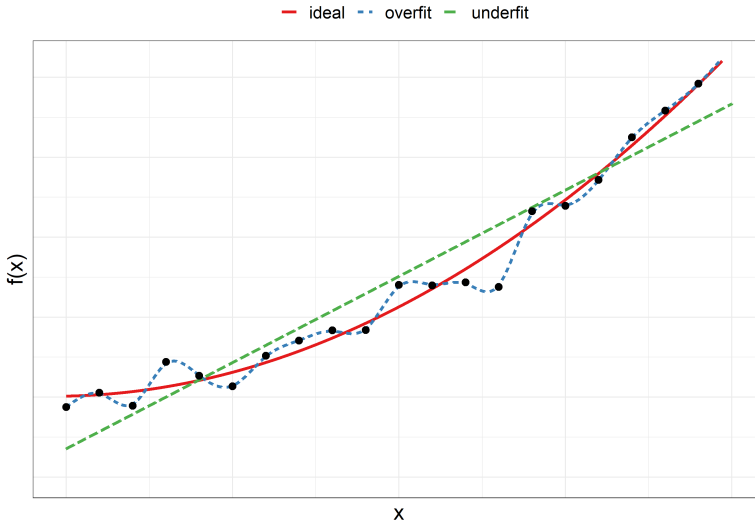


Figure 3.1: Comparison of an overfitting, an underfitting, and an ideal model.

3.1.1 Linear regression

Let $\mathbf{X} \in \mathbb{R}^{m \times n} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ be a data matrix consisting of n features or covariates, and let \mathbf{y} be the target. The linear regression model [34] without intercept is given as

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \dots + \beta_n \mathbf{x}_n + \boldsymbol{\varepsilon} = \mathbf{X}^T \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (3.1)$$

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)$ are the model coefficients (feature weights in machine learning terminology), and $\boldsymbol{\varepsilon}$ is the error term. In a linear regression model with intercept, replace \mathbf{X} by the design matrix $\tilde{\mathbf{X}} = (\mathbf{1}_m, \mathbf{X})$, and $\boldsymbol{\beta}$ by $\tilde{\boldsymbol{\beta}} = (\beta_0, \boldsymbol{\beta})$. Each linear regression model with intercept can be transferred to a linear regression model without intercept by standardizing the data a-priori.

If \mathbf{X} has full rank, the ordinary least squares estimate $\hat{\boldsymbol{\beta}}$ is unique and can be computed explicitly as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

However, short-wide datasets where \mathbf{X} does not have full rank lead to the problem with underdetermined systems, see Section 1.3.

A linear regression model requires some assumptions that should be checked

- **Linearity:** As its name says, a linear regression can only capture the linear relationship between features and the target.
- **Normal distributed error:** ε must be i.i.d. normally distributed, $\varepsilon \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$.
- **Variance homogeneity:** σ^2 is independent of the covariates.

Linear Regression builds the foundation for different generalized linear models, such as logistic regression, an outcome prediction model for classification problems.

3.1.2 Logistic regression

Other than in linear regression setups, logistic regression [34] aims to model and predict classes rather than numerical values.

Based on Equation 3.1, we need a transformation of y_i , resulting in $z_i = g(y_i)$ to get $P(y_i = 1)$ for some sample $i \in \{1, \dots, m\}$, i.e., the probability of class 1, when classes 0 and 1 are possible in a binary setup. In logistic regression, the probabilities are obtained by using the sigmoid function, s.t. $g(y_i) = \frac{1}{1 + \exp(-y_i)}$. Hence, if

$$\begin{aligned}\lim_{y_i \rightarrow \infty} g(y_i) &= 1 \\ \lim_{y_i \rightarrow -\infty} g(y_i) &= 0 \\ \lim_{y_i \rightarrow 0} g(y_i) &= 0.5\end{aligned}$$

The closer the predicted value is to 1, the more likely the predicted sample belongs to class 1. On the other hand, the closer the prediction is to 0, the more likely the sample belongs to class 0. Maximum likelihood estimation is used to determine the model coefficients $\hat{\beta}$.

3.1.3 Regularization

One solution to prevent overfitting in machine learning models is regularization. Prominent approaches include adding Ridge, Lasso, and elastic net penalty terms to the optimization of (generalized) linear models [34]. Regularized models can simultaneously improve the outcome prediction and remove redundant features.

Ridge regression - L2 Penalty

The Ridge regularization adds an L2 term to the optimization of β , shrinking the coefficients towards zero. The optimal $\hat{\beta}$ is determined as follows

$$\hat{\beta}_{\text{Ridge}} = \underset{\beta}{\operatorname{argmin}} \{ \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2 \}.$$

The parameter $\lambda \in \mathbb{R}^+$ controls the strength of shrinkage - the higher λ , the more the coefficients are forced towards zero. In matrix terms, the optimal beta is given as

$$\hat{\beta}_{\text{Ridge}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_n)^{-1} \mathbf{X}^T \mathbf{y},$$

where \mathbf{I}_n is the $(n \times n)$ identity matrix, which makes the term $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_n)$ invertible, even if $\mathbf{X}^T \mathbf{X}$ does not have full rank. By introducing a small bias in the coefficient estimates, their standard errors reduce; hence, the estimates are more accurate in the case of multicollinearity than ordinary least squares [34].

Figure 3.2 illustrates how Ridge regularization affects the optimization of an unregularized regression model for the 2-dimensional case. Considering the term $\|\mathbf{y} - \mathbf{X}\beta\|_2^2$, the optimal $\hat{\beta}$ is represented by a grey dot. For the optimal solution, the model overfits maximally. Non-optimal solutions are illustrated as ellipses around the grey dot, where the target value is constant for each contour. On the other hand, the optimal $\hat{\beta}$ for the penalty term $\lambda \|\beta\|_2^2$ is in the center of the coordinate system (blue dot), i.e., when both β_1 and β_2 are zero. In this case, the model underfits maximally as there is practically no model since all coefficients are zero. Again, non-optimal solutions are represented by circle-shaped contour lines around the blue dot. The optimal $\hat{\beta}$ is the red dot, where the ellipse and the circle intersect. Hence, we determine the compromise between the two parts. Hence, we reach a balance between over- and underfitting of the model.

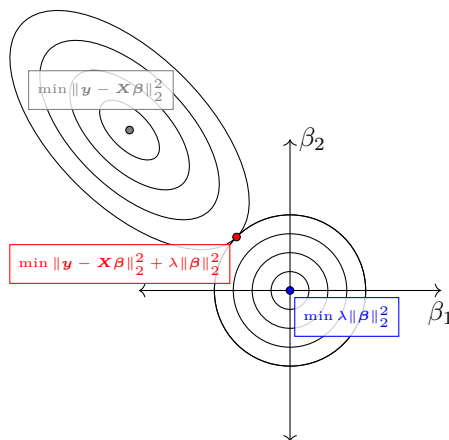


Figure 3.2: Optimizing $\hat{\beta}$ for the 2-dimensional case in Ridge regression setups based on [34]. All minimizations are referred to β .

Even though the $\hat{\beta}$ values are pushed against zero, they are most likely not exactly zero. This scenario is tackled with Lasso regression.

Lasso regression - L1 penalty

Similar to Ridge regression, a parameter λ together with an L1 penalty term shrinks the coefficients towards zero. The Lasso estimate is given as follows [34]:

$$\hat{\beta}_{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \{ \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 \},$$

where $\|\beta\|_1 = \sum_{j=1}^n |\beta_j|$ denotes the L1-norm.

Due to the L1 regularization, the circle in Figure 3.2 is exchanged with a rectangle, see Figure 3.3 (a). Hence, it is more likely that some estimated $\hat{\beta}$ values are exactly zero, offering the additional advantage of integrated feature selection directly during the regression.

The higher λ is, the stronger the regularization affects the optimization, and the more coefficients are set to zero. The estimation of $\hat{\beta}_{\text{Lasso}}$ is solved using quadratic optimization algorithms, see [34].

Comparison between Lasso and Ridge regularization

Ridge regression reduces the complexity of the model but not the number of features. Lasso does shrinkage simultaneously with feature selection, but the feature selection can be unstable, especially for high-dimensional datasets [113]. Imagine two highly correlated features on which we perform feature selection. In two different feature selection runs, either feature can be selected randomly and without any reason why that particular feature was preferred over the other. Hence, the interpretation of selected features can be difficult.

In general, more options to choose the penalty are available through $L^p, p \geq 0$ norms. These regularizers are called Bayes estimates and are beyond the scope of this thesis.

Independent of the regularize, feature scaling is important to ensure that different features are not affected differently by the penalty based on their units.

Elastic net regularization

Elastic net regression is a combination of Lasso and Ridge, that uses the advantages of both, see Figure 3.3 (b). On the one hand, it is capable of removing unimportant features like Lasso regression. On the other hand, it can handle multicollinearities by shrinking all coefficients towards zero like Ridge regression. The estimate is computed as follows:

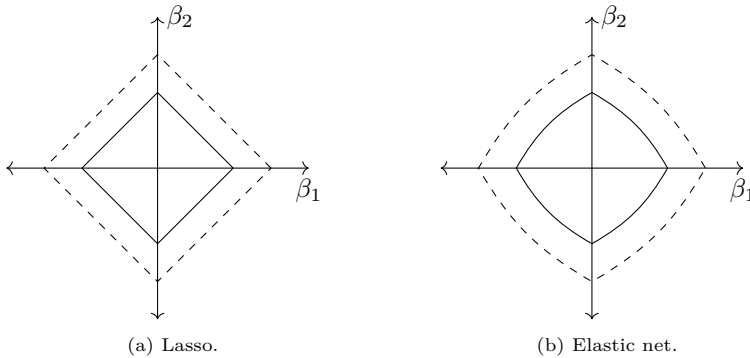


Figure 3.3: Shapes of the penalty terms in the optimization formula for Lasso regularization and elastic net regularization in the 2-dimensional case based on [34].

$$\hat{\beta}^{\text{elastic net}} = \underset{\beta}{\operatorname{argmin}} \left\{ \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \left(\alpha \cdot \|\beta\|_2^2 + (1 - \alpha) \cdot \|\beta\|_1 \right) \right\},$$

where $\alpha \in [0, 1]$ is the parameter balancing the Ridge and Lasso terms. If $\alpha = 1$, the formula reduces to Ridge regression; for $\alpha = 0$, we get the Lasso penalty. A good tradeoff must be found for each setup individually.

3.1.4 Overview Bayesian statistics

The two pillars of statistics are a) frequentist statistics and b) Bayesian statistics. The main difference is that from a frequentists perspective, an unknown parameter θ is assumed to have a fixed value which is usually estimated with methods such as maximum likelihood, or method of moments, while from a Bayesian point of view, the unknown parameter follows a probability distribution [29]. Hence, θ can be characterized by a full probability density function. The basics of Bayesian statistics will be explained for a one-dimensional variable θ but is easily adapted for a parameter vector $\boldsymbol{\theta}$.

Bayes' theorem [29] plays a major role in Bayesian inference. To define the theorem, we need to introduce the concept of *conditional probability* first.

Definition 3.1.1 (Conditional Probability) *Let A and B be two random variables or events. The conditional probability*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

is the probability of A given that B is true. If A and B are independent, $P(A \cap B) = P(A) \cdot P(B)$ and hence, $P(A|B) = P(A)$.

As an example, imagine event A represents *heart attack* and B , *smoker and high cholesterol*. Given B , the probability of A will be higher than for people who do

not smoke and have a low cholesterol level. Otherwise, if B represents the event *blond hair*, the events A and B are independent and the probability of a heart attack will not change if the person has blond hair or not.

For Bayes' theorem we assume that an unknown parameter θ and some observed data \mathbf{y} are given. The theorem states that

$$P(\theta|\mathbf{y}) = \frac{P(\mathbf{y}, \theta)}{P(\mathbf{y})} = \frac{P(\mathbf{y}|\theta) \cdot P(\theta)}{P(\mathbf{y})} \propto P(\mathbf{y}|\theta) \cdot P(\theta).$$

While $P(\mathbf{y}|\theta)$ represents information from data (collected based on the true underlying parameter θ), $P(\theta)$ describes the prior knowledge about θ . While in frequentist statistics, no prior knowledge is considered, a Bayesian framework assumes that we have some prior information about θ given.

Evaluating the normalization constant is necessary to explicitly calculate probabilities from Bayes' theorem. As $p(\mathbf{y})$ is a high-dimensional integral

$$P(\mathbf{y}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} P(y_1, y_2, \dots, y_n) dy_1, dy_2, \dots, dy_n,$$

the computation requires computational statistics such as Monte Carlo Markov Chain algorithms to be solved. Dependent on the problem, it can be enough to consider that $P(\theta|\mathbf{y}) \propto P(\mathbf{y}|\theta) \cdot P(\theta)$, as $P(\mathbf{y})$ is independent of θ and hence, only a scaling factor.

Likelihood

To find the optimal parameter θ , frequentist statistics focuses on the optimization of the likelihood $p(\mathbf{y}|\theta)$, a distribution describing the data. With maximum likelihood strategies, the optimal θ is determined by swapping the two variables and considering $p(\theta|\mathbf{y})$. This strategy corresponds to a uniform (and therefore, in most cases, non-informative) prior, in accordance with Bayes' theorem. Here, θ is a single numeric value when optimized. In Bayesian statistics, we combine the likelihood distribution with a prior distribution to infer the posterior distribution of θ , as shown in the third picture of Figure 3.4, offering more flexibility than a single value. In this example, $p(\theta|\mathbf{y})$ is Beta distributed.

Prior, posterior and parameter estimation

As its name indicates, the prior distribution $p(\theta)$ models a-priori knowledge about the underlying parameter θ . Different types of prior knowledge are available, also for cases where nothing is known. Imagine a random person whom we want to estimate the probability of a heart attack but no information about the person or their lifestyle is known. To still use a Bayesian framework, non-informative priors can be used to model the probability.

Bayesian models often rely on the use of conjugate priors, which massively reduce the computational burden associated with inference tasks by allowing for analytical solutions. The defining characteristic of conjugate priors is that the posterior distribution remains within the same distribution family as the prior [54]. In this case, only parameter updates must be computed, but the numeric computation of $p(\mathbf{y})$ is obsolete. For example, when the likelihood is binomial, and the prior is Beta distributed, the posterior follows a Beta distribution as well [54], see Figure 3.4.

The probability mass function of the Binomial distribution is given as follows:

$$P(y|\theta) = \binom{N}{y} \theta^y (1 - \theta)^{N-y},$$

where N denotes the number of draws, $\theta \in \Theta = [0, 1]$ indicates the success probability, and y is the number of successes. As a prior distribution, we assume a Beta distribution with parameters α and β , which has the density function

$$P(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1},$$

where $B(\cdot, \cdot)$ is the Beta function.

Then,

$$\begin{aligned} P(\theta|y) &= \frac{P(y|\theta)P(\theta)}{\int_{\Theta} P(y|\eta)P(\eta)d\eta} \\ &= \frac{\binom{N}{y} \theta^y (1 - \theta)^{N-y} \cdot \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1}}{\int_0^1 \binom{N}{y} \eta^y (1 - \eta)^{N-y} \cdot \frac{1}{B(\alpha, \beta)} \eta^{\alpha-1} (1 - \eta)^{\beta-1} d\eta} \\ &= \frac{\binom{N}{y} \frac{1}{B(\alpha, \beta)} \cdot \theta^{y+\alpha-1} (1 - \theta)^{N-y+\beta-1}}{\binom{N}{y} \frac{1}{B(\alpha, \beta)} \cdot \underbrace{\int_0^1 \eta^{y+\alpha-1} (1 - \eta)^{N-y+\beta-1} d\eta}_{B(y+\alpha, N-y+\beta)}} \\ &= \text{Beta}(y + \alpha, N - y + \beta) \end{aligned}$$

Once the posterior distribution is estimated, we can compute different statistics from the distribution of θ . A common estimate is the mean value. Also confidence intervals are possible, as we consider a distribution and not a single value.

3.2 Data preprocessing

Data preprocessing is the most important and time-consuming work for data scientists. In a perfect world, datasets have no missing data, only numeric columns

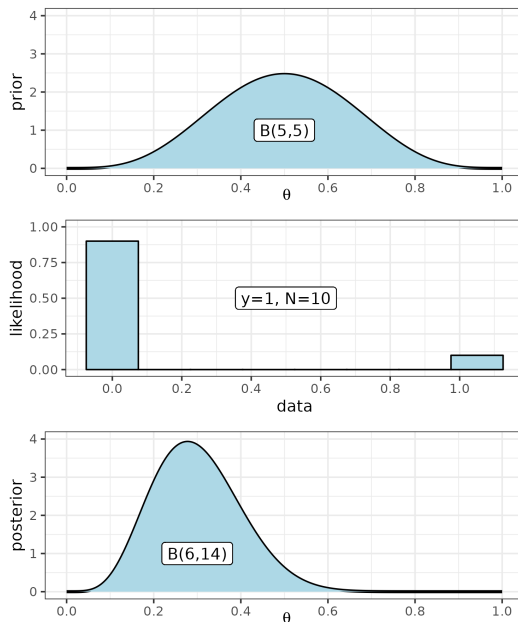


Figure 3.4: Illustration how to combine prior and likelihood to posterior distribution (Bayesian inference) based on [54]. In the likelihood, y defines the number of heads when throwing a (biased) coin N times.

on the same scale and enough samples to train a machine learning model. Reality is different, though - datasets come in various formats, including text features, dates, or numeric and integer features, have missing data, can have duplicates in rows and columns, outliers, or are on completely different scales. To train an adequate model, data scientists have to prepare a dataset carefully in advance.

3.2.1 Data scaling and transformations

Data features are usually on different scales. Imagine two features \mathbf{x}_1 , and \mathbf{x}_2 , where $\mathbf{x}_1 \in [0, 1]^m$, and $\mathbf{x}_2 \in [10^3, 10^4]^m$. Furthermore, assume that we know \mathbf{x}_1 is informative, while \mathbf{x}_2 is non-informative. Without data scaling, feature \mathbf{x}_2 might indicate a higher influence on the machine learning model due to its absolute scale, even though in reality the feature has no information at all. In use cases where data is collected from different sources, preprocessing is extremely important as different sources may have an even higher chance of being on different scales. When bringing all features to the same scale, they have equal conditions when training a model [85]. Another advantage of feature scaling is faster convergence for specific algorithms such as gradient descent, where the feature directly influences the weight update, i.e., without scaling, weights of features with higher values would update faster. Furthermore, many statistical models require scaled features, such

as k NN-clustering, where a distance metric between features is computed, or PCA, where feature variance plays a major role. Features with larger ranges also have higher variances and would be prioritized in the algorithm without preprocessing. Nevertheless, scale-invariant algorithms exist, such as tree-based methods that rely on decision rules.

The most common scaling strategies are normalization and standardization. We also consider Yeo-Johnson transformation in this work.

Normalization

Normalization, also known as Min-Max scaling, transforms the data to a common range $[0, 1]$. A feature \mathbf{x} is normalized with the following formula [85]:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \quad (3.2)$$

Especially in image analysis with RGB images (pixels values between 0 and 255), normalization is a frequently used preprocessing strategy.

Standardization

With standardization we center \mathbf{x} at mean 0 and with standard deviation 1:

$$\mathbf{x}_{\text{st}} = \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})},$$

where $\mu(\mathbf{x})$ is the mean, and $\sigma(\mathbf{x})$ is the standard deviation of \mathbf{x} . With this transformation, mean and standard deviation are equal to those of a standard normal distribution, what can be advantageous when training a machine learning model in terms of convergence and speed. Furthermore, a model with standardized features is less prone to outliers than a model with normalized features [85].

Yeo-Johnson transformation

The Yeo-Johnson transformation [114] is an extension of the Box-Cox transformation, with the aim of stabilizing and making the data more normally distributed. The basic Box-Cox transform for a feature \mathbf{x} and sample $i \in \{1, \dots, m\}$ is given as

$$x_i^{bx} = \begin{cases} \frac{x_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(x_i) & \text{if } \lambda = 0, \end{cases}$$

where $\lambda \in \mathbb{R}$.

Box-Cox is only defined for positive \mathbf{x} , i.e., features must not contain negative values. As this is usually not the case in data analysis, a more powerful transformation is needed: Yeo-Johnson offers a solution by allowing negative and zero values. Yeo-Johnson transformation is defined by

$$x_i^{yj} = \begin{cases} \frac{((x_i + 1)^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0, x_i \geq 0 \\ \log(x_i + 1) & \text{if } \lambda = 0, x_i \geq 0 \\ -\frac{((-x_i + 1)^{2-\lambda} - 1)}{2 - \lambda} & \text{if } \lambda \neq 2, x_i < 0 \\ -\log(-x_i + 1) & \text{if } \lambda = 2, x_i < 0. \end{cases}$$

For positive x_i , Yeo-Johnson transformation is a Box-Cox transformation with input $x_i + 1$. Specifically, Yeo-Johnson transformation has different powers for positive and negative values and is more flexible than Box-Cox transformation.

3.2.2 Outlier detection and handling of missing data

Outliers in datasets appear frequently [100]. Reasons may be measurement errors in technical equipment, human-made errors when collecting data, or simply samples that behave differently from the majority. Dependent on the use case, it must be discussed whether outliers should be removed or if a model should take them into account. Outliers can disturb the training of machine learning models by introducing a bias. On the other hand, they may capture information from extreme cases and help models to learn the corresponding information. We will focus briefly on outlier removal techniques in this chapter. For one-dimensional variables, different approaches exist. The most intuitive approach is to check the estimated quantiles of the dataset's distribution — a boxplot can be a useful tool to get a quick graphical overview [100]. One-dimensional approaches determine outliers in each feature separately. Hence, if a sample is an outlier in one of the features, it is removed from the dataset. For high-dimensional datasets, isolation forests and the local outlier factor algorithm are established tools [17].

Missing data imputation is another big issue related to the fact that the sample set should not get too small by removing too much data [85]. When a feature is not measured for a sample (e.g., no MRI measurements for a patient), the corresponding dataset entry is denoted as missing. In datasets with thousands of samples, those with missing values can be easily removed, and still enough data for training remains. Otherwise, for datasets with few samples, data imputation makes more sense. While common univariate feature imputation methods rely on summary statistics like mean, median, and quantiles, multivariate imputing is more challenging and has to consider more than one feature simultaneously.

In paper 4, we use the Nearest Neighbor imputation strategy [102] to compensate for missing values. Assume that a feature is missing for a specific sample. The

imputed value is defined as a (weighted) average of the values across its k -nearest neighbors, with no missing value in that specific feature. The k -nearest neighbors are determined with the Euclidean distance that can handle missing data [23]. Specifically, the metric computes the distance between all complete features, i.e., features where no entry is missing.

3.2.3 Data encoding

Machine learning models cannot interpret categorical features [85]. Therefore, categorical features must be encoded, i.e., converted to a numerical scale, beforehand. The encoding strategy depends on whether the categories are ordinal or not. If no order is given, such as the feature describing the hospital where a person is treated, one-hot encoding is a good choice. We generate a separate column for each unique hospital and insert a 1 at position i of hospital j if patient i is treated in hospital j . For the other hospitals, we insert 0 at position i .

Otherwise, if a feature contains an order, we aim to retain this information when training a model. With one-hot-encoding, any order would get lost. Ordinal encoding is a good way to encode a categorical feature and simultaneously preserve the order of the feature. Imagine the feature describing a disease stage. Intuitively, the higher the stage, the more progressed the disease and the more dangerous it is for a patient. In this example, ordinal encoding assigns an integer to each unique stage. The higher the stage, the higher the integer. Usually, the "lowest" category receives 0 or 1. The numbers are increasing monotonically. Even though ordinal encoding preserves an order, we cannot assume that the step width between two consecutive stages is equal. Hence, we propose a slight adaptation of ordinal encoding, where a feature is encoded in multiple columns, similar to one-hot encoding. The columns represent the categories in an increasing order. Sample i gets a 1 in column j if its category is below j or equal to j . An example is given in Table 3.1.

Table 3.1: Examples of different feature encoding strategies. Ordinal strategies assume that the levels are ordered as follows: $A < B < C < D$.

| feature level | one-hot encoding | ordinal encoding | proposed encoding |
|---------------|------------------|------------------|-------------------|
| A | 1 0 0 0 | 0 | 1 0 0 0 |
| B | 0 1 0 0 | 1 | 1 1 0 0 |
| C | 0 0 1 0 | 2 | 1 1 1 0 |
| D | 0 0 0 1 | 3 | 1 1 1 1 |

3.3 Outcome prediction models

Real-world datasets can be complex, and thus, linear approaches are often insufficient to achieve high predictive performance. In this section, we will briefly

summarize the outcome prediction models used in the papers of this thesis, beyond (generalized) linear models. In the following, we cover k NN regressors/classifiers, decision trees, and artificial neural networks.

3.3.1 k NN regression and classification

The k NN method for regression or classification [34] is a non-linear approach pursuing an intuitive and interpretable concept. Given a training and a testing set, we determine the k -nearest neighbor samples $\mathcal{N}_k(s_i)$ in the training set for each sample s_i in the testing set based on the Euclidean distance on the feature space. We predict the output y_i either with the average output value (regression) or with the modal value (classification) of its k -nearest neighbors from the training set:

$$\hat{y}_i = \begin{cases} \frac{1}{k} \sum_{l \in \mathcal{N}_k(s_i)} y_l & \text{for regression} \\ \text{mode}\{y_l : l \in \mathcal{N}_k(s_i)\} & \text{for classification,} \end{cases}$$

where mode denotes the modal value of a given set of discrete values.

Although k NN is an efficient non-linear approach, its limitation is that it weights all features equally when computing the neighborhood based on a Euclidean distance in the feature space. Thus, no preference is given to more informative features, and scaling has a strong impact on the neighborhoods.

3.3.2 Decision trees

Decision trees are supervised algorithms pursuing the idea of binary decision rules. Starting from a root node, a new sample is propagated through a tree until it reaches a leaf node [85]. Each internal node represents a binary decision based on thresholding a single feature, which determines whether the left or right child node is used. Leaf nodes describe the outcome (the class for classification setups or the numeric estimate in the case of regression). The size of the tree depends on the number of features and some user setups.

An example of a decision tree is given in Figure 3.5, where we classify three different animals (bird, elephant, and cat) based on key features such as the ability to fly or weight. A new sample enters the root node, where either the left or the right child node is selected, depending on the ability of the new animal to fly. If the right child node is selected, another split is done based on the weight of the animal.

The advantages of tree-based methods are good interpretability and the ability to handle categorical as well as numerical variables. Decision trees are recursive methods, meaning that a splitting rule is applied in each internal node. Once the tree is built, the prediction of new samples is easy by just starting at the root and propagating downwards. A challenge when working with decision trees

is determining which feature to use in each node, e.g., the root node. The most common approach for the training process is to use a purity measure given by the Gini index [84], which is related to the concept of entropy.

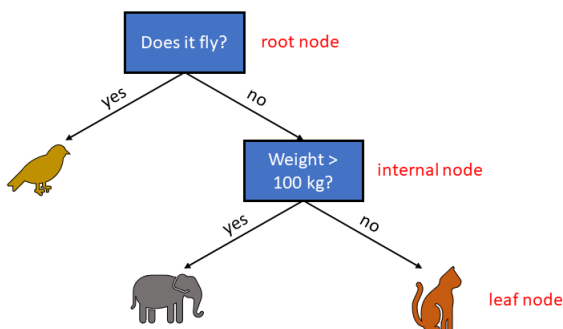


Figure 3.5: Exemplary decision tree example to predict classes *bird*, *elephant*, and *cat*.

For regression trees, other criteria based on error terms, such as the mean squared error (MSE), are used [85].

When a tree exceeds a certain depth, i.e., a high number of internal nodes between the root node and leaf nodes, overfitting is possible. This issue occurs frequently if a leaf node is classified based on a low number of samples — a possible counter-measure is pruning, i.e., restricting the depth of a tree.

Decision trees are known to be unstable as the choice of the feature for a given node is dependent on the input samples [61]. A single-tree model can be extended to a random forest with multiple trees, increasing the stability of the outcome prediction and preventing overfitting. For random forests, multiple trees are trained on different subsamples of the training data. The subsamples are not mutually exclusive, i.e., a sample can appear in more than one tree. In the end, a majority voting decides which class an object is assigned to, i.e., the class which is selected most times among the trees wins.

3.3.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) were first introduced in the 1940s [68] with the intention of understanding the human brain. During the last decades, they have gained attention, especially since modern hardware has sufficient computational power to perform large-scale computations. Due to their flexibility and broad applicability, ANNs are used in different setups, including image analysis, modeling sequential and time series data, or basic outcome prediction for (multi-class)

classification and regression tasks [85]. In this thesis, we consider only the basic structure of ANNs; extensions can be found in [4, 85].

ANNs have a complex model structure and comprise a large number of parameters. The basic concept behind them is the adaptive linear neuron (Adaline) for binary classification [85], shown in Figure 3.6. The first part is similar to a linear regression model, where features are multiplied with weights. A bias term of 1 represents the intercept. This multiplication leads to the layer output z . In mathematical terms

$$z = \mathbf{w}^T \mathbf{x}.$$

On top of z , we apply an activation function φ , where we denote the activated layer output as $a = \varphi(z)$. The activation is linear in the case of Adaline, i.e., $a(z) = z$, but can be non-linear in more advanced network architectures. To obtain the binarized output y , Adaline uses a threshold function $h(\cdot)$ to convert the continuous value z , i.e.,

$$y = h(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

In Adaline, the weights \mathbf{w} are estimated via gradient descent. Among other settings, the user must define a learning rate and a loss function [40] to train the model. Computing the error with the continuous $a(z)$ instead of the binarized y is more efficient and leads to faster convergence [85]. An Adaline model can be converted to a logistic regression model by exchanging the linear activation with a sigmoid activation and by adjusting the loss function.

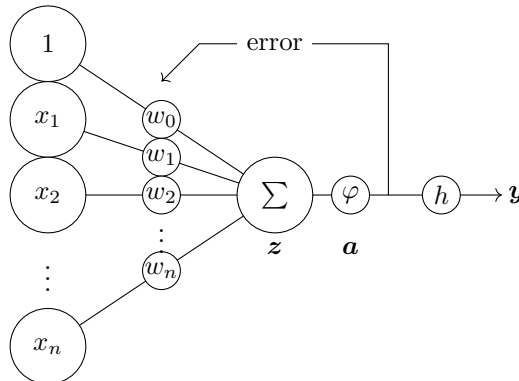


Figure 3.6: Adaline classifier based on [85].

For more advanced architectures, non-linear activation functions such as *tanh*, *relu*, or *sigmoid* are used to a large extent.

A multilayer ANN is the concatenation and stacking of multiple layers of neurons, see Figure 3.7. If the number of layers is high, we arrive at so-called deep artificial neural network. The features \mathbf{x} of a sample enter the network through the input layer. The sample propagates through a network of different linear operations and (non-)linear activations until it reaches the output layer y . Each layer contains a bias term, denoted as $b^{(0)}, b^{(1)}, b^{(2)}$ in Figure 3.7. The backpropagation algorithm is the most common approach to training ANN parameters. It relies on a composition of the chain rule; for more details, see [85]. In general, ANNs offer a large range of layer architectures to the user, making the model flexible.

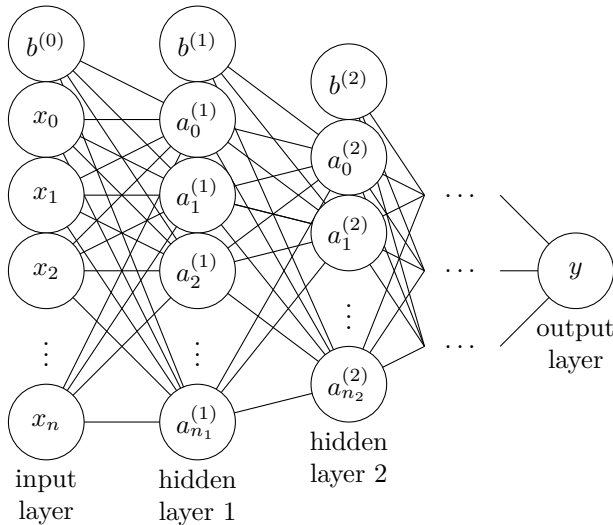


Figure 3.7: Multilayer perceptron model based on [85].

An efficient network architecture is beneficial for training. Even though deep networks can incorporate non-linear relations in the data, they are prone to overfitting due to the high number of weight parameters that must be estimated while training. A considerably large sample size is necessary to train a neural network reliably. In addition, neural networks are often called "black-box" models, meaning that it is difficult to interpret them and understand the inner process of the architecture, especially for deep ANNs.

3.4 Performance metrics

Performance metrics are tools for measuring the performance of machine learning models. They are closely connected to loss functions, which are used during model training [85]. Measuring the performance of a machine learning model quantifies how well it predicts data. Furthermore, performance metrics can indicate if a

model overfits when used together with various validation techniques.

3.4.1 Classification

Assuming a binary classification problem with a negative and a positive class, performance measures are usually combinations of four expressions:

1. True Positive (TP): number of samples correctly predicted as positive;
2. False Positive (FP): number of samples incorrectly predicted as positive;
3. True Negative (TN): number of samples correctly predicted as negative;
4. False Negative (FN): number of samples incorrectly predicted as negative.

The most basic performance measure for classification setups is *accuracy* (ACC), where we compute the percentage of correctly classified samples among all samples

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

Even though accuracy is an intuitive metric, it is not a good choice for unbalanced datasets. Assume a dataset where 90% of the test samples belong to the negative and only 10% to the positive class. Even if all samples are assigned to the negative class, the accuracy would be exactly 90% and hence, suggests that the model works well. On the contrary, we are not able to predict the positive class at all, in this case. Accuracy is appropriate if we do not focus on incorrect predictions but rather on how many data samples were predicted correctly. When either class positive or class negative is of particular interest, F1 score is the better choice. The F1 score metric is the harmonic mean between precision and recall. While precision measures the amount of TP over all positively predicted samples, recall determines the number of TP over all truly positive samples.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.3}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.4}$$

$$\text{F1} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{3.5}$$

The outcome prediction performance is considered as good if the F1 score in a binary setup is high, no matter which of the two classes is considered the "positive" class. All metrics (accuracy, precision, recall, F1 score) are bounded between 0 and 1, where 0 means that all samples are incorrect and 1 represents perfect performance.

F1 score shows two weaknesses. First, the metric is different for both classes making interpretation not trivial, and second, it does not consider the TN samples at

all. A solution to these issues is the usage of the Matthews Correlation Coefficient (MCC) as performance metric [19]. It is bounded between -1 and 1, where -1 indicates total disagreement, 0 means random class assignment, and 1, perfect prediction. The metric is a combination of all four – TP, FP, TN, and FN:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})}}$$

All metrics are applicable to multiclass problems by using macro-, or micro-averaging [115].

Definition 3.4.1 *Let B be an arbitrary classification metric, and c the number of distinct classes. Then,*

$$B_{macro} = \frac{1}{c} \sum_{j=1}^c B(\text{TP}_j, \text{FP}_j, \text{TN}_j, \text{FN}_j)$$
$$B_{micro} = B \left(\sum_{j=1}^c \text{TP}_j, \sum_{j=1}^c \text{FP}_j, \sum_{j=1}^c \text{TN}_j, \sum_{j=1}^c \text{FN}_j \right),$$

where TP_j is the number of true positive samples for class j . The same applies to TN , FN , and FP , correspondingly.

3.4.2 Regression

The output of a regression model is a numeric value. Hence, performance metrics are based on distances between the output and the ground truth. The most common metric is the mean squared error (MSE) [85]. Assuming that $\hat{\mathbf{y}}$ represents the predictions,

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2.$$

Considering the formula, the squaring has the effect of penalizing large errors more than small errors. Intuitively, MSE shall be minimized. The smaller, the better the model. The interpretation of MSE is difficult due to a) squared effects, and b) it has a lower bound of 0 (perfect prediction) but no upper bound. A good strategy to estimate model performance is to compare the MSE to the MSE of a poor baseline model, defining an upper bound by that.

Different extensions of MSE exist with the aim of normalizing the error in some way. One method is called root mean squared error (RMSE). As the name already indicates, it computes the root of MSE:

$$\text{RMSE} = \sqrt{\text{MSE}}.$$

The advantage of RMSE over MSE is that the error is on the same scale as \mathbf{y} . A useful adaption of RMSE is the root mean squared error inter-quantile range (RMSEIQR) which is less sensitive to extreme values because RMSE is divided by its inter-quantile range (IQR), which is the difference between the 75th and 25th percentiles of the data:

$$\text{IQR} = Q(0.75) - Q(0.25),$$

where Q represents the empirical quantile function of the data.

Easier to interpret, the coefficient of determination (R^2 score) [22] is defined as

$$R^2 = 1 - \frac{\overbrace{\sum_{i=1}^m (y_i - \hat{y}_i)^2}^{\text{SSR}}}{\underbrace{\sum_{i=1}^m (y_i - \bar{y})^2}_{\text{SST}}},$$

where \bar{y} is the mean of \mathbf{y} . The metric has an upper bound of 1, which means perfect prediction. R^2 score computes the ratio between the residual sum of squares (SSR) and the total sum of squares (SST).

Chapter 4

Paper Summaries

This thesis contains four main publications. While papers I-III provide feature selection methodology, paper IV presents an application in the healthcare domain.

In paper I, we present a data-driven ensemble feature selection approach (RENT) that delivers stable feature sets on high-dimensional datasets.

While RENT performed well in multiple data-driven scenarios, additional user knowledge is often available for many datasets and could be utilized in the modeling process. Thus, paper II introduces a user-guided Bayesian framework that combines data-driven feature selection with prior information.

Paper III relates to papers I and II in terms of ranking features in block-wise data, targeting a similar problem to feature selection. The article incorporates an artificial neural network with block-wise input data. Using different concepts of ranking block-feature importances, paper III additionally allows us to understand black-box models better.

Finally, paper IV demonstrates how RENT and UBayFS can be applied to determine the most important features of a dataset modeling the overall survival of cancer patients.

4.1 Paper I

The first paper describes the repeated elastic net technique for feature selection (RENT). The method relies on an ensemble of generalized linear models with elastic net regularization trained on distinct subsets of the data.

Rather than evaluating a single feature selection run, the ensemble of models evaluated by RENT explores a summary statistic over the frequency of each feature being selected throughout the ensembles. The empirical distribution of feature

coefficients across elementary models gives us important feedback about its importance and stability, see Figure 4.1.

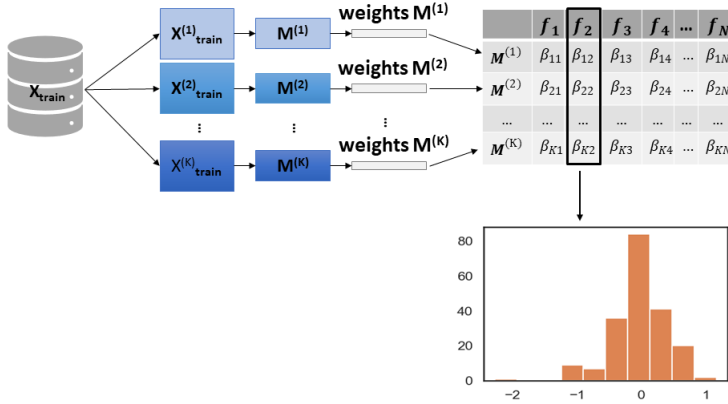


Figure 4.1: RENT workflow based on [46]. First, we split the training data into distinct subsets. We run a feature selection algorithm on each subset and consider the feature coefficients from which we build an empirical distribution for each feature. Different statistics are computed from this distribution.

Hence, we introduce three criteria that must be met by a feature in order to be selected. The first criterion τ_1 requires that a certain ratio of elementary models must select a feature, i.e., the frequency throughout the ensemble is sufficiently high. Furthermore, the stability of the sign of the feature coefficients is evaluated, as alternating signs indicate instability (second criterion τ_2). The third criterion τ_3 introduces a Student’s t -test to assess whether the feature coefficients differ significantly from zero. It may be the case that a feature fulfills the first two criteria but has small coefficients among the elementary models. The contribution is still minimal, and the feature can be discarded.

Finally, we select a feature i if it exceeds pre-defined cutoff values t_1, t_2, t_3 for all three criteria, i.e., $\tau_1(i) > t_1$, $\tau_2(i) > t_2$, and $\tau_3(i) > t_3$. The paper suggests to select the cutoff criteria either via the Bayesian information criterion [76], or through any other hyperparameter selection technique, which is up to the user. While high cutoff values deliver a small feature set with important features, low cutoff values can serve as a noise-removal and general cleaning of the dataset by removing features with little or no contribution. The use of the presented criteria makes the method stable and applicable to short-wide as well as long-thin datasets.

Another benefit of the ensemble approach is that the user obtains a predictive performance value from each elementary model on the left-out validation set. With this information, we can detect an outlier in the data sample efficiently by controlling the rate of incorrect classifications throughout the ensemble. If this number is high, the sample might be an outlier and either removed from the dataset for subsequent analysis steps or investigated further.

In the paper, a comparison with the stability selection framework [71] and random forests shows that RENT delivers a) a high F1-score for the predictive performance of a model trained with the selected features and b) high stability of the selected feature set, already for a low number of ensembles. In comparison with RENT, the stability selection framework has high stability but a weaker performance, while random forest delivers more accurate predictions for new samples but is unstable.

In summary, the paper illustrates that RENT is a valid and well-performing extension of the body of research on feature selection. RENT outperforms state-of-the-art ensemble approaches regarding the tradeoff between performance and stability.

4.2 Paper II

Paper II introduces UBayFS, a (U)ser guided (Bay)esian Framework for (F)eature (S)election, an ensemble feature selector embedded in a Bayesian framework. UBayFS relies on two sources of information: data and domain knowledge. Information from data is modeled via a multinomial likelihood function. Specifically, we build an ensemble of an arbitrary feature selector and determine the selection frequency for each feature, similar to the approach in Paper I. A multinomial distribution describes the probability distribution of the feature selection frequency counts of all features. Due to its generic framework, UBayFS allows any arbitrary feature selector to be used as elementary model.

In many situations, prior knowledge about feature importance is available from experts or prior analyses of similar data. By including relevant prior information, UBayFS incorporates prior feature weights via a Dirichlet prior. Features with higher prior importance get a higher weight than the remaining features. As the Dirichlet distribution is the conjugate prior with respect to the multinomial likelihood, the posterior distribution is of Dirichlet type, as well.

With this information, we aim to infer the underlying feature importances. In addition, UBayFS allows for additional side constraints, such as a maximal number of features, must-link constraints, indicating which features must be selected together, or cannot-link constraints, where at most one out of a set of features can be selected. As an example, a cannot-link constraint is useful for highly correlated features.

The Bayesian model and the side constraints are optimized jointly via a genetic

algorithm [30], which returns the final feature set. Constraints may be defined between whole data sources. Furthermore, the penalty strength of violating side constraints is steered by a separate parameter, giving the user even more flexibility. An illustration of UBayFS is provided in Figure 4.2.

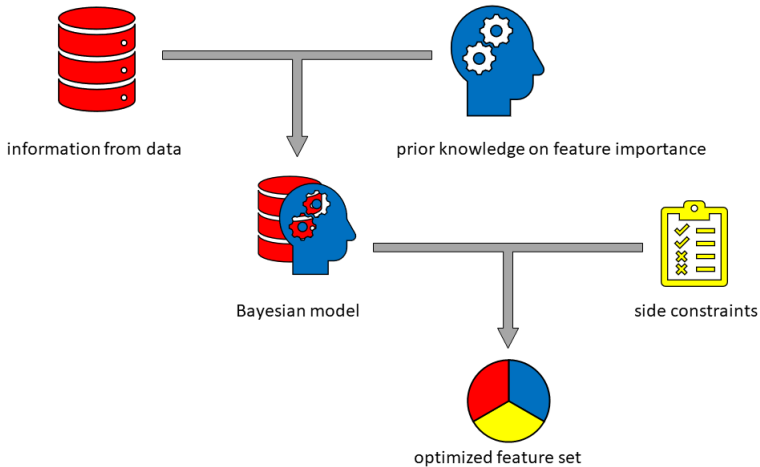


Figure 4.2: UBayFS framework [43]. The framework combines information from data with prior feature importance knowledge and additional side constraints.

The presented feature selection framework is flexible and can include different types of user knowledge in the feature selection procedure. We show that incorporating prior information in the UBayFS setup can improve the performance of an outcome prediction model and the stability of the feature set. In addition, the use of ensemble feature selection as likelihood ensures the stability of the selected feature set.

UBayFS can also be used for feature-block selection and thereby represents an alternative to group Lasso [25].

4.3 Paper III

While papers I and II propose methods to select a subset of features from the training dataset, the approach in paper III quantifies block-feature importance as a post-hoc step in ANNs. Therefore, paper III relates to papers I and II in the following aspect: first, paper III particularly targets ANNs, while paper I and II leave the choice of the predictive model open - thereby, paper III contributes to

the understanding of feature importance in black-box models. Further, the article explicitly ranks the feature blocks by their importance to the outcome prediction model. Even though ANNs are state-of-the-art models in machine learning, they are often described as "black-box" models because of their difficult interpretation. Once enough data for training a network is available, those architectures are promising and can easily outperform classical approaches.

Hence the presented approach is a post-processing technique where we assume that all data are used to train the network. The article's primary goal is to quantify block importance, i.e., which data source contributes most, second most, or least. Hence, we get some information from the ANN. We use a multiblock-ANN strategy for the block-wise data, where each data block enters the network through a separate branch. The branches are merged in a concatenation layer where the information propagates forward until it reaches the output layer, see Figure 4.3.

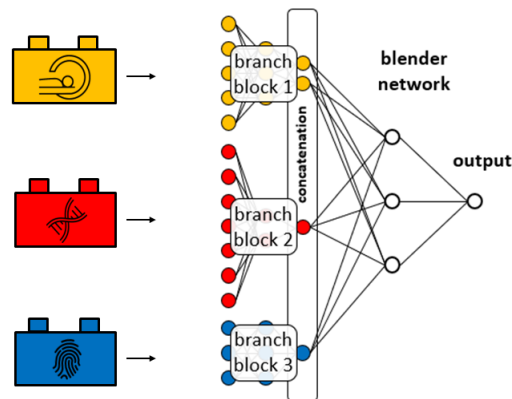


Figure 4.3: Illustration of a Multiblock Artificial Neural Network [44]. Each data source enters the network through a separate branch. The information is merged in a concatenation layer.

We propose three distinct strategies to quantify block-feature importance. The first strategy is the easiest and relies on summary statistics of single features. Strategy two and three are the main contributions and are illustrated in Figure 4.4.

In the first strategy, we claim that a feature block is important if it consists of features with high individual importance, which is measured based on variational gradients. The block with the highest average feature importance is ranked most important.

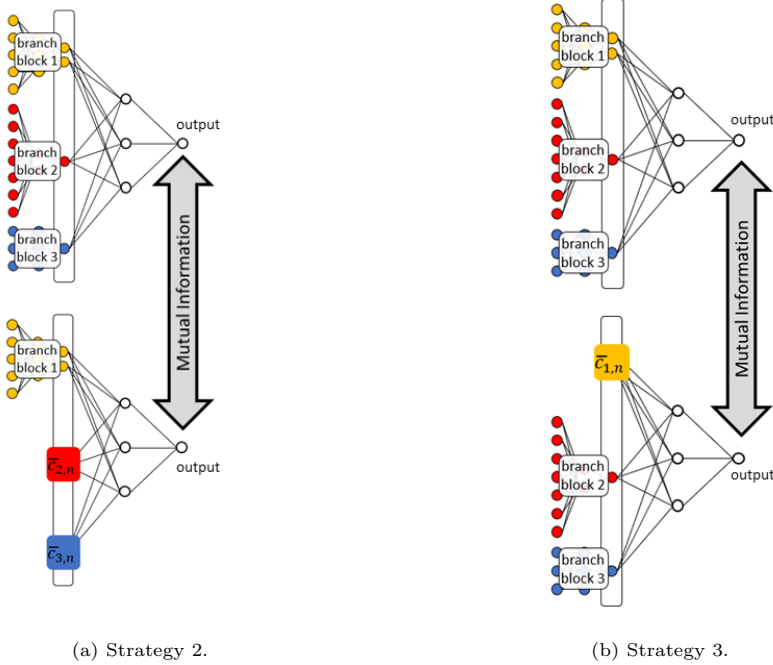


Figure 4.4: Strategies to quantify feature-block importance [44].

The second strategy considers how a block is important if it can explain a large part of the model output on its own, which is computed based on a mutual information criterion.

Strategy three is similar to strategy two, but in this scenario, we quantify a block as important if its removal significantly changes the model output.

The presented architectures can be used for single-feature importance quantification, as well, by considering each feature as a separate block. Furthermore, beyond the paper’s content, the approach can also be used for feature selection and is especially useful for categorical features. Once a categorical feature is encoded, we can define all encoded variables as a block and carry out the presented procedure.

Different simulation studies show that all three criteria are well-defined and rank the features correctly.

4.4 Paper IV

The last article puts the approaches developed in papers I and II into practice [50]. On a real-world cancer dataset, the goal is to compare how purely data-driven feature selection behaves compared to feature selection with integrated expert knowledge. We show that RENT and UBayFS are able to reveal relevant information for novel clinical insights and interpretation and hence, have the potential to impact research in the field of healthcare.

The investigated dataset contains clinical features from patients with gastroenteropancreatic neuroendocrine neoplasms, a heterogeneous type of malignancy. A neoplasm is an abnormal and excessive growth of tissue, and neuroendocrine tumors are such neoplasms that affect cells in the endocrine and nervous systems. Patients have a low survival prognosis - mostly below two years. Features from five heterogeneous data sources are available. Those comprise patient characteristics, baseline blood values, histology, imaging, and treatment. After preprocessing, we obtain a total of 113 features. As this type of cancer is rare, the cohort consists of only 63 patients, which induces an underdetermined system when training machine learning models. In the presented paper, we determine a set of most influencing features on the overall survival of patients. As an additional level of complexity, the dataset consists of both, censored and uncensored samples. We use a target transformation to avoid conflicts resulting from censored data.

Two different experiments show how RENT and UBayFS select features on this dataset. Due to the small sample size and the desired stability analysis of the two ensemble feature selectors, we split the data into five folds, where four folds are used for feature selection and subsequent outcome prediction modeling, and the remaining fold is used for testing. To measure the predictive performance of the selected features we use a k NN, and a linear model, which are evaluated in terms of root mean squared error. Multiple studies have shown that several features, such as age, tumor differentiation, or the primary tumor site, impact the overall survival of patients. The second experiment deals with the integration of those features into UBayFS.

In experiment 1, we assume no prior knowledge about feature importances - we compare the selected features of RENT and UBayFS by determining whether the feature sets delivered by each method are similar, and how many of the a-priori important features get selected. The results indicate that overall, the two methods deliver a large overlap in their selected feature sets, out of which a high number is also identified as important by experts. The average correlation between selected features is below 25% for both methods, indicating that they do not select redundant features. The stability of the selected feature sets among five folds is slightly higher for UBayFS (≈ 0.60) compared to RENT (≈ 0.50). We show that the predictive performance of selected features is dependent on the data fold — as there are only 12-13 patients in each test fold, a single outlier can have a huge impact on the performance.

Experiment 2 demonstrates how UBayFS integrates prior knowledge into the feature selection process. Hence, we increase the prior weights of the established features and start the feature selection. While the predictive performance is on a similar level (slightly better) as in experiment 1, the feature set stability increases with the height of the prior weight. Still, the average feature correlation is low.

Overall, the paper demonstrated the practical use of both feature selectors in clinical practice and shows promising insights, which have the potential to improve the understanding of cancer.

Chapter 5

Discussion & Conclusion

This thesis makes a contribution to the development of stable and interpretable feature selection algorithms for high-dimensional datasets with possible block structures. Especially the incorporation of expert knowledge into ensemble feature selection models, together with block-structured datasets, goes beyond the capabilities of state-of-the-art methodology in the area.

In this chapter, we discuss the results of this thesis in the context of the research questions defined in Chapter 1 and present an outlook for future research.

5.1 Research Question I

The first research question deals with the problem of selecting stable and interpretable features in short-wide datasets. With the results of this thesis, including experimental results associated with RENT and UBayFS, as well as a real-world use case on cancer data (paper IV), we can show that ensemble feature selection is a valid and efficient approach to overcoming the challenges associated with RQ I. We show that RENT outperforms state-of-the-art feature selectors in terms of performance and stability in multiple scenarios. The method removes irrelevant features and demonstrated to provide good and stable results on short-wide as well as long-thin datasets in the experiments conducted in paper I. Furthermore, due to its ensemble basis with underlying regularized linear models, RENT offers a framework to analyze samples and features simultaneously, which delivers deep insight into the datasets and helps to interpret results. Beyond feature selection, RENT may be used to detect sample outliers in the dataset by evaluating the average classification accuracy for each sample across elementary models (if the probability of a sample being assigned class 1 is 0.55, it is above the threshold of 0.5 and assigned to class 1, but the decision was associated with a high level of uncertainty). The different cutoff criteria and the possible interpretations make

RENT a powerful extension of the established stability selection framework [71].

Some capabilities of RENT are extended by UBayFS in paper II, which can incorporate prior weights and constraints to improve model interpretation even more. For instance, a dataset with a categorical feature requires encoding into multiple binary features during preprocessing. While ordinary ensemble feature selectors randomly select a subset of these encoded features, UBayFS can be steered via a constraint to select all or none of the corresponding binary features. In case a dataset contains an intrinsic block structure, e.g., from different data sources, UBayFS can include this type of information in the model, as well.

In summary, UBayFS does not beat the predictive performance of RENT and group Lasso in each scenario, but its increased flexibility to incorporate user information (see RQ 2) and information about the data structure in the model facilitates interpretation and allows for a larger scope of applications compared to state-of-the-art feature selection frameworks. Using the proposed methods, we provide a tool for a large range of scenarios of high-dimensional feature selection.

A main limitation of the presented feature selection frameworks is the sampling strategy of ensemble feature selectors for small sample sizes: common data-splitting strategies are cross-validation and random sampling. Both are not ideal as the sample size of a single ensemble model becomes even smaller, which increases the errors of the model [106]. The problem of modeling small sample sizes has been addressed in literature such as [52, 103]. Consequences of small datasets are a lack of generalization (target distributions are not well represented in the data) and difficulties in optimization (overfitting, even more, underdetermined systems). Anyway, using a single train/test split on small datasets would lead to non-representative results, which makes ensembles inevitable [6]. Hence, further research is required on integrating more accurate sampling strategies into RENT and UBayFS. For instance, an extension of RENT [111] suggests to use a bagging and boosting strategy. Instead of computing an elastic net regularized model in each step of the ensemble, they propose to use a boosted elastic net version instead. Furthermore, a bootstrapping strategy determines subsamples to train the ensemble models on. The authors claim that the presented extension of RENT further improves the stability of a feature set.

Another known downside is that feature selection does not always have a unique optimum. In some cases, pairs of features are redundant and, consequently, exchangeable in a final feature set. This issue has only a minor effect on predictive performance as long as at least one optimal feature set can be found. However, the stability and interpretability of selected feature sets by human experts are likely to decline. In order to raise awareness of this issue, a close investigation of the correlation structure of datasets as part of an exploratory data analysis is recommended.

Apart from real-world redundancies between features, spurious correlations in the feature space of a short-wide dataset are likely to occur and need to be treated

with care. In [15], the authors show that high-dimensional datasets may contain an arbitrary number of spurious feature correlations. The computation of correlation matrices during exploratory data analysis helps to detect high correlations between features. Nevertheless, randomly removing features because of existing correlations with other features can be risky because context may be lost. Since a distinction between spurious and real-world correlations is mostly infeasible in short-wide datasets, expert information is often the only solution to this issue. Data providers can help to determine whether correlations are spurious or not. More research on how to efficiently handle the risk of spurious correlations based on expert knowledge is yet required.

5.2 Research Question II

One possibility to distinguish between spurious and proper correlations in a dataset is to use expert knowledge. Domain experts are commonly able to provide valuable insights into the processes underlying the collected datasets and have a good understanding of which correlations appear on purpose and which do not. RQ II raises the question of how to incorporate prior expert knowledge about features into a feature selection model.

UBayFS delivers, to the best of our knowledge, unique properties in the field of feature selection. Incorporating prior information has not yet been extensively considered in feature selection publications. The capability to combine data-driven methods with prior information and additional side constraints of the features increases the scope of potential applications and allows the user to make use of all information at hand.

By assigning prior importance weights, we steer the model towards selecting features that optimize the utility function together with the important features. Nevertheless, a separate parameter can regularize to which degree the model shall take prior information into account when selecting features. Hence, the user can adjust the strength of the provided prior knowledge.

The possibility of including side constraints makes UBayFS even more flexible. We can define block structures, the maximal number of blocks or features to be selected, or incorporate feature correlations.

In the article, we could show that the UBayFS framework delivers interpretable and accurate results and is more flexible than other methods presented in the literature.

Although experimentation with defining hypothetical expert knowledge is possible with UBayFS, care must be taken if such unverified expert knowledge is taken into account to reach scientific conclusions about treatment outcomes or feature importance. Otherwise, the risk of self-reinforcement of decisions is high. Statistically, prior weights act as a bias which may be intended if the prior weights

are well grounded in the existing literature. However, using speculations as prior knowledge would lead to unreliable results and may be misinterpreted by users.

Finally, a limitation of UBayFS is that its model structure is more complex compared to data-driven ensemble feature selectors. Therefore, understanding how different inputs interact when determining a final feature set requires a good model understanding and statistical expertise.

5.3 Research Question III

Component-based models such as SO-PLS [75] and the group Lasso [25] are state-of-the-art to model feature blocks. Even though both consider block structure in the data, both approaches are based on linear models and cannot capture more complex dependencies in the data.

Our proposed methods, UBayFS and the block-quantification approach presented in paper III, give insight into block importance.

UBayFS offers block-wise constraints, e.g., the model must not select two or more blocks simultaneously. In paper II, we use max-size constraints to limit the number of blocks from which features are selected. This, again, demonstrates the flexibility of UBayFS being applicable to a broad range of application scenarios. However, UBayFS cannot be applied a posteriori, i.e., it is not able to shed light on the dependency structure of black-box models, such as Artificial Neural Networks. For this purpose, the methods proposed in paper III can be used.

Paper III relies on multiblock artificial neural networks, where we assume that a network already delivers a good performance and can model complex data. Generally, ANNs can capture non-linearities and are more powerful than traditional approaches. As every block enters the network through a separate branch, we defined three strategies to quantify the network's feature-block importance and open the "black box" of neural networks. Furthermore, the method can be used to quantify single-feature importance, as well, by considering each feature as a separate block.

The downside of ANNs is their need for large sample sizes. The deeper the network is, the more parameters need to be estimated. Hence, for datasets with few samples, the network overfits easily and is unreliable. Since the proposed block ranking is applied post-hoc, however, tuning ANNs to gain a good performance is out of the scope of this work.

When using the methods proposed in paper III, we assume that the extracted features in each block ideally represent the information contained in the block. Still, we do not know whether the feature extraction pipeline is optimal or leaves options for improving the data representation, which is beyond the scope of this thesis.

5.4 Outlook

The broad field of feature selection in healthcare opens many future research directions. In paper IV, we presented one possible solution for censored data - anyway, the approach might not be applicable to other datasets. Developing feature selection methods for censored data is an important topic that requires even more attention. Lifetime models are state-of-the-art in statistics but do not always satisfy the requirements of predictive machine learning models. For instance, the Cox proportional hazard model models individual risks of patients but is not suitable for making lifetime predictions directly. Thus, results from this model cannot be directly integrated into the proposed frameworks discussed in this thesis. An interesting future goal would be to integrate censoring information directly into established machine learning methods.

Regarding the methods developed within this thesis, we might consider other baseline models for RENT instead of the generalized linear model with Lasso regularization. Using algorithms that handle censored data instead might have a huge impact, especially combined with non-linearity. Hence, research in that direction is needed. UBayFS, on the other hand, is a general framework where adaptations with censored feature selection algorithms could be made easily. For the feature-block importance ranking in paper III, we aim to evaluate the method on a larger dataset with neural networks capable of integrating censored data. Much research is targeted to the field of ANNs in combination with lifetime models [58, 117].

Another future goal is to extend this work for setups with more complex target variables, such as multi-class, ordered categorical, or even multivariate responses. Methodologically RENT and UBayFS are capable of modeling multi-class targets in their existing frameworks, yet implementations must be adapted for such scenarios. On the other hand, multivariate target variables require fundamental changes to the feature selection frameworks, for instance, by using multi-target regression models [53]. The same holds for ordinal categorical targets, which may be described by ordinal logistic regression models [33].

As paper IV showed interesting results based on the proposed feature selectors, we aim to extend our research to different cancer types or even other diseases. This gives us new insights from a clinical perspective but also to understand the behavior of RENT and UBayFS better.

From a long-term perspective, one could focus on decision support systems, where the developed methodology is integrated into personalized treatment planning. A sophisticated understanding of biological processes with help of data science is likely to lead to better prognoses for patients and may facilitate treatment in the clinic. Ultimately, the conclusions from this thesis may contribute to the improvement of treatment plans for serious diseases like cancers and thereby benefit healthcare professionals and patients.

List of Abbreviations

| Mathematical notations | |
|-------------------------------------------------------------------|---------------------------------------------------------|
| $\mathbf{X} \in \mathbb{R}^{m \times n}$ | data matrix |
| $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times (n+1)}$ | data matrix with intercept term |
| $n \in \mathbb{N}$ | number of features/variables (columns of \mathbf{X}) |
| $m \in \mathbb{N}$ | number of samples/observations (rows of \mathbf{X}) |
| $\mathbf{y} \in \mathbb{R}^m$ | target feature/variable |
| $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ or \mathbb{R}^n | regression parameters (with or without intercept) |
| $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ | model error term |
| $\mathbf{1}_k \in \mathbb{R}^k$ | k -dimensional vector of ones |
| $\boldsymbol{\delta} \in \{0, 1\}^n$ | feature set |
| $c \in \mathbb{N}$ | number of classes |
| $K \in \mathbb{N}$ | number of models in an ensemble |
| $\mathcal{S} \subseteq \mathbb{R}^k$ | support of a k -dimensional probability distribution |
| $p : \mathcal{S} \rightarrow \mathbb{R}^+$ | probability density function |
| $t \in \mathbb{R}$ | threshold |
| $\lambda \in \mathbb{R}^+, \alpha \in [0, 1]$ | regularization parameters |
| $\theta \in \mathbb{R}$ or $\boldsymbol{\theta} \in \mathbb{R}^k$ | general model parameter(s) |

| Abbreviations | |
|----------------------|--------------------------------------|
| MRI | magnetic resonance imaging |
| CT | computer tomography imaging |
| PET | positron emission tomography imaging |
| ML | machine learning |
| RQ | research question |
| ANOVA | analysis of variance |
| PCA | Principal Component Analysis |
| PCR | Principal Component Regression |

| | |
|--------|------------------------------------------------------|
| PLSR | Partial Least Squares Regression |
| SO-PLS | Sequential and Orthogonalized Partial Least Squares |
| ROSA | Response Oriented Sequential Alternation |
| MFA | Multiple Factor Analysis |
| UMAP | Uniform Manifold Approximation and Projection |
| mRMR | minimum Redundancy Maximal Relevance criterion |
| RENT | Repeated Elastic Net Technique for Feature Selection |
| UBayFS | User-Guided Bayesian Framework for Feature Selection |
| ANN | Artificial Neural Network |
| k NN | k -nearest neighbors |
| ACC | accuracy |
| (R)MSE | (root) mean squared error |
| IQR | inter-quartile range |
| R^2 | coefficient of determination |
| MCC | Matthews Correlation Coefficient |

List of Figures

| | | |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | Examples of different data sources. | 5 |
| 2.1 | In feature selection, the complete feature space is reduced to a smaller set of relevant features. Redundant features are removed. . | 12 |
| 3.1 | Comparison of an overfitting, an underfitting, and an ideal model. | 20 |
| 3.2 | Optimizing $\hat{\beta}$ for the 2-dimensional case in Ridge regression setups based on [34]. All minimizations are referred to β | 22 |
| 3.3 | Shapes of the penalty terms in the optimization formula for Lasso regularization and elastic net regularization in the 2-dimensional case based on [34]. | 24 |
| 3.4 | Illustration how to combine prior and likelihood to posterior distribution (Bayesian inference) based on [54]. In the likelihood, y defines the number of heads when throwing a (biased) coin N times. | 27 |
| 3.5 | Exemplary decision tree example to predict classes <i>bird</i> , <i>elephant</i> , and <i>cat</i> | 32 |
| 3.6 | Adaline classifier based on [85]. | 33 |
| 3.7 | Multilayer perceptron model based on [85]. | 34 |
| 4.1 | RENT workflow based on [46]. First, we split the training data into distinct subsets. We run a feature selection algorithm on each subset and consider the feature coefficients from which we build an empirical distribution for each feature. Different statistics are computed from this distribution. | 40 |
| 4.2 | UBayFS framework [43]. The framework combines information from data with prior feature importance knowledge and additional side constraints. | 42 |
| 4.3 | Illustration of a Multiblock Artificial Neural Network [44]. Each data source enters the network through a separate branch. The information is merged in a concatenation layer. | 43 |
| 4.4 | Strategies to quantify feature-block importance [44]. | 44 |

Bibliography

- [1] Abdi, H.: Partial least square regression (PLS regression). *Encyclopedia for research methods for the social sciences* **6**(4), 792–795 (2003)
- [2] Abdi, H., Williams, L.J.: Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* **2**(4), 433–459 (2010)
- [3] Abdi, H., Williams, L.J., Valentin, D.: Multiple factor analysis: principal component analysis for multitable and multiblock data sets. *Wiley Interdisciplinary reviews: computational statistics* **5**(2), 149–179 (2013)
- [4] Aggarwal, C.C., et al.: *Neural networks and deep learning*. Springer **10**, 978–3 (2018)
- [5] Alanazi, H., Abdullah, A., Qureshi, K., Ismail, A.: Accurate and dynamic predictive model for better prediction in medicine and healthcare. *Irish Journal of Medical Science (1971-)* **187**(2), 501–513 (2018)
- [6] An, C., Park, Y.W., Ahn, S.S., Han, K., Kim, H., Lee, S.K.: Radiomics machine learning study with a small sample size: Single random training-test set split may lead to unreliable results. *PloS one* **16**(8), e0256152 (2021)
- [7] Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) *Advances in Neural Information Processing Systems*. vol. 14. MIT Press (2001)
- [8] Bica, I., Alaa, A.M., Lambert, C., Van Der Schaar, M.: From real-world patient data to individualized treatment effects using machine learning: current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics* **109**(1), 87–100 (2021)
- [9] Bolón-Canedo, V., Alonso-Betanzos, A.: Recent advances in ensembles for feature selection, vol. 147. Springer (2018)
- [10] Bolón-Canedo, V., Alonso-Betanzos, A.: Ensembles for feature selection: A review and future trends. *Information Fusion* **52**, 1–12 (2019)

-
- [11] Bolón-Canedo, V., Sánchez-Marroño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. *Knowledge and information systems* **34**(3), 483–519 (2013)
- [12] Bonte, S., Goethals, I., Van Holen, R.: Machine learning based brain tumour segmentation on limited data using local texture and abnormality. *Computers in biology and medicine* **98**, 39–47 (2018)
- [13] Bortolotto, C., Lancia, A., Stelitano, C., Montesano, M., Merizzoli, E., Agustoni, F., Stella, G., Preda, L., Filippi, A.R.: Radiomics features as predictive and prognostic biomarkers in NSCLC. *Expert Review of Anticancer Therapy* **21**(3), 257–266 (2021)
- [14] Bro, R., Smilde, A.K.: Principal component analysis. *Analytical methods* **6**(9), 2812–2831 (2014)
- [15] Calude, C.S., Longo, G.: The deluge of spurious correlations in big data. *Foundations of science* **22**(3), 595–612 (2017)
- [16] Cao, B., He, L., Kong, X., Philip, S.Y., Hao, Z., Ragin, A.B.: Tensor-based multi-view feature selection with applications to brain diseases. In: 2014 IEEE International Conference on Data Mining. pp. 40–49. IEEE (2014)
- [17] Cheng, Z., Zou, C., Dong, J.: Outlier detection using isolation forest and local outlier factor. In: Proceedings of the conference on research in adaptive and convergent systems. pp. 161–168 (2019)
- [18] Cherrington, M., Lu, J., Airehrour, D., Thabtah, F., Xu, Q., Madanian, S.: Feature selection: Multi-source and multi-view data limitations, capabilities and potentials. In: 2019 29th International Telecommunication Networks and Applications Conference (ITNAC). pp. 1–6. IEEE (2019)
- [19] Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics* **21**(1), 1–13 (2020)
- [20] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016)
- [21] D’Arrigo, G., Leonardis, D., Abd ElHafeez, S., Fusaro, M., Tripepi, G., Roumeliotis, S.: Methods to analyse time-to-event data: the Kaplan-Meier survival curve. *Oxidative medicine and cellular longevity* **2021** (2021)
- [22] Di Bucchianico, A.: Coefficient of determination (R²). *Encyclopedia of statistics in quality and reliability* (2008)
- [23] Dixon, J.K.: Pattern recognition with partly missing data. *IEEE Transactions on Systems, Man, and Cybernetics* **9**(10), 617–621 (1979)

-
- [24] Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al.: U-Net: deep learning for cell counting, detection, and morphometry. *Nature methods* **16**(1), 67–70 (2019)
- [25] Friedman, J., Hastie, T., Tibshirani, R.: A note on the group lasso and a sparse group lasso (2010)
- [26] Fröhlich, H., Balling, R., Beerenwinkel, N., Kohlbacher, O., Kumar, S., Lengauer, T., Maathuis, M.H., Moreau, Y., Murphy, S.A., Przytycka, T.M., et al.: From hype to reality: data science enabling personalized medicine. *BMC medicine* **16**(1), 1–15 (2018)
- [27] Gao, S., Calhoun, V.D., Sui, J.: Machine learning in major depression: From classification to treatment outcome prediction. *CNS neuroscience & therapeutics* **24**(11), 1037–1052 (2018)
- [28] Garcia-Vidal, C., Sanjuan, G., Puerta-Alcalde, P., Moreno-García, E., Soriano, A.: Artificial intelligence to support clinical decision-making processes. *EBioMedicine* **46**, 27–29 (2019)
- [29] Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: *Bayesian Data Analysis*. Chapman and Hall/CRC (Nov 2013)
- [30] Givens, G.H., Hoeting, J.A.: *Computational statistics*, vol. 703. John Wiley & Sons (2012)
- [31] Groendahl, A.R., Knudtsen, I.S., Huynh, B.N., Mulstad, M., Moe, Y.M., Knuth, F., Tomic, O., Indahl, U.G., Torheim, T., Dale, E., et al.: A comparison of methods for fully automatic segmentation of tumors and involved nodes in PET/CT of head and neck cancers. *Physics in Medicine & Biology* **66**(6), 065012 (2021)
- [32] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of machine learning research* **3**(Mar), 1157–1182 (2003)
- [33] Harrell, Jr, F.E., Harrell, F.E.: Ordinal logistic regression. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis* pp. 311–325 (2015)
- [34] Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer (2009)
- [35] He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. *Advances in neural information processing systems* **18** (2005)
- [36] Hesamian, M.H., Jia, W., He, X., Kennedy, P.: Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging* **32**(4), 582–596 (2019)
-

-
- [37] Higgins, J.P.: Nonlinear systems in medicine. *The Yale journal of biology and medicine* **75**(5-6), 247 (2002)
- [38] Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems* **32** (2019)
- [39] Hsu, H.H., Hsieh, C.W., Lu, M.D.: Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications* **38**(7), 8144–8150 (2011)
- [40] Janocha, K., Czarnecki, W.M.: On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659* (2017)
- [41] Jenul, A.: Data science for treatment outcome prediction: towards interpretable models combining healthcare data from multiple sources. *Women in Data Science Conference Villach, Austria* (2021), technical vision talk
- [42] Jenul, A., Bhattarai, B., Liland, K.H., Jiao, L., Schrunner, S., Futsaether, C., Granmo, O.C., Tomic, O.: Component based pre-filtering of noisy data for improved Tsetlin machine modelling. In: *2022 International Symposium on the Tsetlin Machine (ISTM)*. pp. 57–64. IEEE (2022)
- [43] Jenul, A., Schrunner, S.: UBayFS: An R package for user guided feature selection. *Journal of Open Source Software* **8**(81), 4848 (2023)
- [44] Jenul, A., Schrunner, S., Huynh, B.N., Helin, R., Futsaether, C.M., Liland, K.H., Tomic, O.: Ranking feature-block importance in artificial multiblock neural networks. In: *International Conference on Artificial Neural Networks*. pp. 163–175. Springer (2022)
- [45] Jenul, A., Schrunner, S., Huynh, B.N., Tomic, O.: RENT: A Python package for repeated elastic net feature selection. *Journal of Open Source Software* **6**(63), 3323 (2021)
- [46] Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsaether, C.M., Tomic, O.: RENT—repeated elastic net technique for feature selection. *IEEE Access* **9**, 152333–152346 (2021)
- [47] Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsaether, C.M., Tomic, O.: RENT - repeated elastic net technique for feature selection. *Geilo Winter School* (2021)
- [48] Jenul, A., Schrunner, S., Pilz, J., Tomic, O.: A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). *Machine Learning* **111**(10), 3897–3923 (2022)
- [49] Jenul, A., Schrunner, S., Pilz, J., Tomic, O.: A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). *European Conference on Machine Learning* (2022), journal track

-
- [50] Jenul, A., Stokmo, H.L., Schrunner, S., Revheim, M.E., Hjortland, G.O., Tomic, O.: Towards understanding the survival of patients with high-grade gastroenteropancreatic neuroendocrine neoplasms: An investigation of ensemble feature selection in the prediction of overall survival. arXiv preprint arXiv:2302.10106 (2023)
- [51] Khaire, U.M., Dhanalakshmi, R.: Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences* **34**(4), 1060–1073 (2022)
- [52] Kokol, P., Kokol, M., Zagoranski, S.: Machine learning on small size samples: A synthetic knowledge synthesis. *Science Progress* **105**(1) (2022)
- [53] Korneva, E., Blockeel, H.: Towards better evaluation of multi-target regression models. In: *Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020)*, Ghent, Belgium, September 14–18, 2020, Proceedings. pp. 353–362. Springer (2020)
- [54] Kruschke, J.: *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press (2014)
- [55] Kubben, P., Dumontier, M., Dekker, A.: *Fundamentals of clinical data science*. Springer Nature (2019)
- [56] Kumar, K.: Principal component analysis: Most favourite tool in chemometrics. *Resonance* **22**(8), 747–759 (2017)
- [57] Kuras, A., Jenul, A., Brell, M., Burud, I.: Comparison of 2D and 3D semantic segmentation in urban areas using fused hyperspectral and lidar data. *Journal of Spectral Imaging* **11** (2022)
- [58] Kvamme, H., Borgan, Ø.: Continuous and discrete-time survival prediction with neural networks. *Lifetime data analysis* **27**, 710–736 (2021)
- [59] Labroski, A.: *Multi-view versus single-view machine learning for disease diagnosis in primary healthcare*. Ph.D. thesis, ETSI-Informatica (2018)
- [60] Leung, K.M., Elashoff, R.M., Affi, A.A.: Censoring issues in survival analysis. *Annual review of public health* **18**(1), 83–104 (1997)
- [61] Li, R.H., Belford, G.G.: Instability of decision tree classification algorithms. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 570–575 (2002)
- [62] Li, Y., Wu, F.X., Ngom, A.: A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics* **19**(2), 325–340 (2018)
- [63] Liland, K.H., Næs, T., Indahl, U.G.: ROSA—a fast extension of partial least squares regression for multiblock data analysis. *Journal of Chemometrics* **30**(11), 651–662 (2016)
-

-
- [64] Liu, Y., Stojadinovic, S., Hrycushko, B., Wardak, Z., Lau, S., Lu, W., Yan, Y., Jiang, S.B., Zhen, X., Timmerman, R., et al.: A deep convolutional neural network-based automatic delineation strategy for multiple brain metastases stereotactic radiosurgery. *PloS one* **12**(10), e0185844 (2017)
- [65] Maier-Hein, L., Vedula, S.S., Speidel, S., Navab, N., Kikinis, R., Park, A., Eisenmann, M., Feussner, H., Forestier, G., Giannarou, S., et al.: Surgical data science for next-generation interventions. *Nature Biomedical Engineering* **1**(9), 691–696 (2017)
- [66] Martens, H., Naes, T.: *Multivariate calibration*. John Wiley & Sons (1992)
- [67] Mayerhoefer, M.E., Materka, A., Langs, G., Häggström, I., Szczypiński, P., Gibbs, P., Cook, G.: Introduction to radiomics. *Journal of Nuclear Medicine* **61**(4), 488–495 (2020)
- [68] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**, 115–133 (1943)
- [69] McDermott, J.E., Wang, J., Mitchell, H., Webb-Robertson, B.J., Hafen, R., Ramey, J., Rodland, K.D.: Challenges in biomarker discovery: combining expert insights with statistical analysis of complex omics data. *Expert opinion on medical diagnostics* **7**(1), 37–51 (2013)
- [70] McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018)
- [71] Meinshausen, N., Bühlmann, P.: Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(4), 417–473 (2010)
- [72] Miao, J., Niu, L.: A survey on feature selection. *Procedia Computer Science* **91**, 919–926 (2016)
- [73] Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
- [74] Næs, T., Martens, H.: Principal component regression in NIR analysis: viewpoints, background details and selection of components. *Journal of chemometrics* **2**(2), 155–167 (1988)
- [75] Næs, T., Romano, R., Tomic, O., Måge, I., Smilde, A., Liland, K.H.: Sequential and orthogonalized PLS (SO-PLS) regression for path analysis: Order of blocks and relations between effects. *Journal of Chemometrics* **35**(10), e3243 (2021)
- [76] Neath, A.A., Cavanaugh, J.E.: *The Bayesian information criterion: background, derivation, and applications*. Wiley Interdisciplinary Reviews: Computational Statistics **4**(2), 199–203 (2012)

-
- [77] Noble, W.S., et al.: Support vector machine applications in computational biology. *Kernel methods in computational biology* **71**, 92 (2004)
- [78] Nogueira, S., Sechidis, K., Brown, G.: On the stability of feature selection algorithms. *J. Mach. Learn. Res.* **18**(1), 6345–6398 (2017)
- [79] Oreiller, V., Andrearczyk, V., Jreige, M., Boughdad, S., Elhalawani, H., Castelli, J., Vallières, M., Zhu, S., Xie, J., Peng, Y., et al.: Head and neck tumor segmentation in PET/CT: the HECKTOR challenge. *Medical image analysis* **77**, 102336 (2022)
- [80] Pacilio, M., Basile, C., Shcherbinin, S., Caselli, F., Ventroni, G., Aragno, D., Mango, L., Santini, E.: An innovative iterative thresholding algorithm for tumour segmentation and volumetric quantification on SPECT images: Monte Carlo-based methodology and validation. *Medical physics* **38**(6Part1), 3050–3061 (2011)
- [81] Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* **27**(8), 1226–1238 (2005)
- [82] Petrie, A., Sabin, C.: *Medical statistics at a glance*. John Wiley & Sons (2019)
- [83] Prinja, S., Gupta, N., Verma, R.: Censoring in clinical trials: review of survival analysis techniques. *Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine* **35**(2), 217 (2010)
- [84] Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence* **41**, 77–93 (2004)
- [85] Raschka, S., Mirjalili, V.: *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd (2019)
- [86] Rehman, A., Khan, M.A., Saba, T., Mehmood, Z., Tariq, U., Ayesha, N.: Microscopic brain tumor detection and classification using 3D CNN and feature selection architecture. *Microscopy Research and Technique* **84**(1), 133–149 (2021)
- [87] Sanchez-Martinez, S., Camara, O., Piella, G., Cikes, M., González-Ballester, M.Á., Miron, M., Vellido, A., Gómez, E., Fraser, A.G., Bijmens, B.: Machine learning for clinical decision-making: challenges and opportunities in cardiovascular imaging. *Frontiers in Cardiovascular Medicine* **8**, 2020 (2022)

-
- [88] Schrunner, S., Scheiber, M., Jenul, A., Zernig, A., Kästner, A., Kern, R.: Machine learning based indicators to enhance process monitoring by pattern recognition. arXiv preprint arXiv:2103.13058 (2021)
- [89] Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., Alonso-Betanzos, A.: Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems* **118**, 124–139 (2017)
- [90] Senders, J.T., Staples, P.C., Karhade, A.V., Zaki, M.M., Gormley, W.B., Broekman, M.L., Smith, T.R., Arnaout, O.: Machine learning and neurosurgical outcome prediction: a systematic review. *World neurosurgery* **109**, 476–486 (2018)
- [91] Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., Khovanova, N.: Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. *Biomedical Signal Processing and Control* **52**, 456–462 (2019)
- [92] Sheikhpour, R., Sarram, M.A., Gharaghani, S., Chahooki, M.A.Z.: A survey on semi-supervised feature selection methods. *Pattern Recognition* **64**, 141–158 (2017)
- [93] Shur, J.D., Doran, S.J., Kumar, S., Ap Dafydd, D., Downey, K., O’Connor, J.P., Papanikolaou, N., Messiou, C., Koh, D.M., Orton, M.R.: Radiomics in oncology: a practical guide. *Radiographics* **41**(6), 1717–1732 (2021)
- [94] Smilde, A.K., Næs, T., Liland, K.H.: *Multiblock Data Fusion in Statistics and Machine Learning: Applications in the Natural and Life Sciences*. John Wiley & Sons (2022)
- [95] Solorio-Fernández, S., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: A review of unsupervised feature selection methods. *Artificial Intelligence Review* **53**(2), 907–948 (2020)
- [96] Spooner, A., Chen, E., Sowmya, A., Sachdev, P., Kochan, N.A., Trollor, J., Brodaty, H.: A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Scientific reports* **10**(1), 1–10 (2020)
- [97] Srujana, B., Verma, D., Naqvi, S.: Machine learning vs. survival analysis models: a study on right censored heart failure data. *Communications in Statistics-Simulation and Computation* pp. 1–18 (2022)
- [98] Srujana, B., Verma, D., Naqvi, S.: Machine Learning vs. survival analysis models: a study on right censored heart failure data. *Communications in Statistics-Simulation and Computation* pp. 1–18 (2022)
- [99] Subrahmanya, S.V.G., Shetty, D.K., Patil, V., Hameed, B.Z., Paul, R., Smriti, K., Naik, N., Somani, B.K.: The role of data science in healthcare

-
- advancements: applications, benefits, and future prospects. *Irish Journal of Medical Science* (1971-) **191**(4), 1473–1483 (2022)
- [100] Suri, N.N.R.R., M, N.M., Athithan, G.: *Outlier Detection: Techniques and Applications*. Springer International Publishing (2019)
- [101] Tomaszewski, M.R., Gillies, R.J.: The biological meaning of radiomic features. *Radiology* **298**(3), 505–516 (2021)
- [102] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**(6), 520–525 (2001)
- [103] Vabalas, A., Gowen, E., Poliakoff, E., Casson, A.J.: Machine learning algorithm validation with a limited sample size. *PloS one* **14**(11), e0224365 (2019)
- [104] Van Griethuysen, J.J., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R.G., Fillion-Robin, J.C., Pieper, S., Aerts, H.J.: Computational radiomics system to decode the radiographic phenotype. *Cancer research* **77**(21), e104–e107 (2017)
- [105] Van Timmeren, J.E., Cester, D., Tanadini-Lang, S., Alkadhi, H., Baessler, B.: Radiomics in medical imaging—“how-to” guide and critical reflection. *Insights into imaging* **11**(1), 1–16 (2020)
- [106] Varoquaux, G.: Cross-validation failure: Small sample sizes lead to large error bars. *Neuroimage* **180**, 68–77 (2018)
- [107] Verleysen, M., François, D.: The curse of dimensionality in data mining and time series prediction. In: *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings* 8. pp. 758–770. Springer (2005)
- [108] Vivanti, R., Joskowicz, L., Lev-Cohain, N., Ephrat, A., Sosna, J.: Patient-specific and global convolutional neural networks for robust automatic liver tumor delineation in follow-up CT studies. *Medical & biological engineering & computing* **56**(9), 1699–1713 (2018)
- [109] Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)* **51**(6), 1–36 (2019)
- [110] Wojtas, M., Chen, K.: Feature importance ranking for deep learning. *Advances in Neural Information Processing Systems* **33**, 5105–5114 (2020)
- [111] Wong, L.M., Ai, Q.Y.H., Zhang, R., Mo, F., King, A.D.: Radiomics for discrimination between early-stage nasopharyngeal carcinoma and benign hyperplasia with stable feature selection on MRI. *Cancers* **14**(14), 3433 (2022)
-

- [112] Wu, C., Zhou, F., Ren, J., Li, X., Jiang, Y., Ma, S.: A selective review of multi-level omics data integration using variable selection. *High-throughput* **8**(1), 4 (2019)
- [113] Xu, H., Caramanis, C., Mannor, S.: Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE transactions on pattern analysis and machine intelligence* **34**(1), 187–193 (2011)
- [114] Yeo, I.K., Johnson, R.A.: A new family of power transformations to improve normality or symmetry. *Biometrika* **87**(4), 954–959 (2000)
- [115] Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* **26**(8), 1819–1837 (2013)
- [116] Zhang, Z., Reinikainen, J., Adeleke, K.A., Pieterse, M.E., Groothuis-Oudshoorn, C.G.: Time-varying covariates and coefficients in Cox regression models. *Annals of translational medicine* **6**(7) (2018)
- [117] Zhao, L., Feng, D.: Deep neural networks for survival analysis using pseudo values. *IEEE journal of biomedical and health informatics* **24**(11), 3308–3314 (2020)

Appendix A

Papers I & Ia

| | |
|---------------------|------------------------------------------------------------------------------------------------------------------------------|
| title: | RENT—Repeated Elastic Net Technique for Feature Selection |
| authors: | Anna Jenul , Stefan Schrunner, Kristian Hovde Liland, Ulf Geir Indahl, Cecilia Marie Futsæther, Oliver Tomic |
| date: | 11/2021 |
| publication: | IEEE Access |
| doi: | https://doi.org/10.1109/ACCESS.2021.3126429 |

| | |
|---------------------|---------------------------------------------------------------------------------------|
| title: | RENT: A Python Package for Repeated Elastic Net Feature Selection |
| authors: | Anna Jenul , Stefan Schrunner, Bao Ngoc Huynh, Oliver Tomic |
| date: | 07/2021 |
| publication: | Journal of Open Source Software |
| doi: | https://doi.org/10.21105/joss.03323 |

Received October 17, 2021, accepted October 29, 2021, date of publication November 8, 2021, date of current version November 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126429

RENT—Repeated Elastic Net Technique for Feature Selection

ANNA JENUL ¹, (Graduate Student Member, IEEE), STEFAN SCHRUNNER, (Member, IEEE), KRISTIAN HOVDE LILAND ¹, ULF GEIR INDAHL, CECILIA MARIE FUTSÆTHER, AND OLIVER TOMIC ²

Faculty of Science and Technology, Norwegian University of Life Sciences, 1430 Ås, Norway

Corresponding author: Anna Jenul (anna.jenul@nmbu.no)

This work was supported by the Norwegian Cancer Society under Grant 182672-2016.

ABSTRACT Feature selection is an essential step in data science pipelines to reduce the complexity associated with large datasets. While much research on this topic focuses on optimizing predictive performance, few studies investigate stability in the context of the feature selection process. In this study, we present the Repeated Elastic Net Technique (RENT) for Feature Selection. RENT uses an ensemble of generalized linear models with elastic net regularization, each trained on distinct subsets of the training data. The feature selection is based on three criteria evaluating the weight distributions of features across all elementary models. This fact leads to the selection of features with high stability that improve the robustness of the final model. Furthermore, unlike established feature selectors, RENT provides valuable information for model interpretation concerning the identification of objects in the data that are difficult to predict during training. In our experiments, we benchmark RENT against six established feature selectors on eight multivariate datasets for binary classification and regression. In the experimental comparison, RENT shows a well-balanced trade-off between predictive performance and stability. Finally, we underline the additional interpretational value of RENT with an exploratory post-hoc analysis of a healthcare dataset.

INDEX TERMS Elastic net regularization, exploratory analysis, ensemble feature selection, generalized linear models, selection stability.

I. INTRODUCTION

A predictive task involves a dataset consisting of N -dimensional row vectors $X = (x_1^T, \dots, x_I^T) \in \mathbb{R}^{I \times N}$ and an associated vector of target values $\mathbf{y} = (y_1, \dots, y_I) \in \mathbb{T}^I$, where the target space \mathbb{T} may represent a set of classes (classification task) or a subset of the real numbers (regression task). In this study, our focus lies on generalized linear models (GLMs), which model the target as a linear combination of the inputs with weights $\beta \in \mathbb{R}^N$, followed by a transformation. The columns of the data matrix describe object characteristics, denoted as features. Since data acquisition techniques evolve steadily, situations where the number of features N exceeds the number of objects I often occur. In such setups, mathematical obstacles, like spurious correlations and multicollinearity issues causing model overfitting, trigger the necessity to reduce the number of features by using

some feature selection approach [1]. These issues are characteristic of various domains, including healthcare [2], [3], biomedicine [4], text mining [5] and botany [6]. A successful feature selection approach will decrease the model complexity, improve the model stability and provide more useful model interpretations.

A feature selector θ_F decomposes the data space into a direct sum of selected features (V_1) and non-selected features (V_2) according to the given feature set $F \subset \{1, \dots, N\}$,

$$\mathbb{R}^N = V_1 \oplus V_2, \text{ s.t. } V_1 \cong \mathbb{R}^{|F|} \text{ and } V_2 \cong \mathbb{R}^{N-|F|},$$

and projects all objects from \mathbb{R}^N to the subspace V_1 , i.e.

$$\theta_F : \mathbb{R}^N \rightarrow V_1, \theta_F(\mathbf{x}) = \text{proj}_{V_1}(\mathbf{x}).$$

The goal of good feature selection is to determine the feature set F^* , which enables a predictive model to obtain the most accurate prediction. Predictive quality is measured using a metric $q(\hat{\mathbf{y}}, \mathbf{y})$, such as F1 score, where $\hat{\mathbf{y}}, \mathbf{y} \in \mathbb{T}^{I_{\text{test}}}$ denote the vectors containing predicted target values $\hat{\mathbf{y}}$ after feature

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang ¹.

selection and ground truth target values \mathbf{y} , both referring to a set of test data X_{test} of size $|X_{\text{test}}| = I_{\text{test}}$. An optimal feature set F^* is characterized by

$$F^* = \arg \max_{F \subset \{1, \dots, N\}} q(\hat{\mathbf{y}}^F, \mathbf{y}).$$

A taxonomy of feature selection techniques distinguishes between filter, wrapper, and embedded approaches. Filter approaches rank features by an importance criterion, such as mutual information or correlation coefficients between features and target variables. Baseline filters include the Fisher score [7] and the Laplacian score [8] as well as algorithms from the relief family [9]. Approaches like mRMR [10] or the stratified feature weight method [11] aim to resolve the issue that correlated and redundant features are not well handled by classical filters [12]. A combination of different filter approaches is suggested in [13]. Wrapper approaches select features concerning their prediction performance. By training supervised models on different subsets of the entire feature set, the subset delivering the most accurate predictions on a test set is chosen. This strategy often causes overfitting issues and high computational costs [14]. Prominent wrapper approaches are forward/backward selection, such as recursive feature selection [15], and heuristic searches like simulated annealing or genetic algorithms [16].

The third category of feature selection methods, embedded feature selection, integrates the selection step directly into the learning algorithm. A class of embedded methods, which is particularly important in this work, comprises regularization for GLMs: During parameter estimation, regularization terms are added as penalties to the target function. While the well-established LASSO [17] uses an L1 term $\lambda_1(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$ for this purpose and delivers a sparse parameter vector, L2-regularization $\lambda_2(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2$ handles multicollinearities by pulling the L2-norm of the parameter vector $\boldsymbol{\beta}$ towards zero. The effects of both regularization terms are combined in the elastic net $\lambda_{\text{enet}}(\boldsymbol{\beta})$ [18], defined as

$$\lambda_{\text{enet}}(\boldsymbol{\beta}) = \gamma[\alpha\lambda_1(\boldsymbol{\beta}) + (1 - \alpha)\lambda_2(\boldsymbol{\beta})], \quad (1)$$

with parameters $\alpha \in [0, 1]$ and γ to weight the regularization terms and to define the regularization strength, respectively. Other representatives of embedded feature selection models are tree-based models, such as decision trees or regression trees. Ensembles of tree-based architectures are called random forests [19]. Graph-based approaches together with elastic net regularization further play a key role in recent works [20]–[22], where the authors demonstrate a graph-based structurally interacting elastic net method incorporating pairwise relationships between objects via a feature graph, or [23], proposing a solution for $\ell_{2,0}$ -norm regularized feature selection via linear discriminant analysis.

Most feature selection approaches suffer from the phenomenon that minor changes in the random initialization or train-test-split of the model lead to major variations in the selected feature set—this issue is referred to as lack

of stability and is investigated in [24] and [25]. In agreement with [26] and [27], the authors argue that L1 regularisation on GLMs is generally unstable. They claim that the issue can be resolved by investigating ensemble feature selection, where θ_F is derived from a set of independently trained (elementary) feature selectors $\theta_{F_1}, \dots, \theta_{F_K}$, such that $\theta_F = \phi(\theta_{F_1}, \dots, \theta_{F_K})$. The operator ϕ acts as a meta-model based on information from the elementary models θ_{F_k} , $k = 1, \dots, K$. A basic approach is to build such a meta-model by counting the frequency of selection for each feature across all feature sets F_k , expressed by

$$F^* = \left\{ i \in \{1, \dots, N\} : \tau_1(i) = \frac{1}{K} \left[\sum_{k=1}^K \mathbb{1}_{\{i \in F_k\}} \right] \geq t_1 \right\},$$

where $t_1 \in [0, 1]$ is a scalar representing a minimum selection frequency threshold. This approach assumes that each elementary feature set F_k consists of a subset of important features (with correspondingly higher probabilities for being selected), and a small, random subset of unimportant features. The final, selected feature set F^* is less likely to contain unimportant features than each of the elementary feature sets F_k . Hence, model stability is increased as shown by Meinshausen and Bühlmann [28], who propose such a feature selector named stability selection. Even though the stability selection framework is intuitive and reasonable, the corresponding feature weights may be small—not significantly different from zero—or have alternating signs across the elementary models. Thus, features might be selected although resulting in ambiguous or contradictory information and hence, deteriorating interpretability and predictive performance. This means that further insights into the predictive power of features have to be gained from the distribution of weights, which is not considered by Meinshausen and Bühlmann [28].

The present work suggests the novel repeated elastic net technique (RENT) for feature selection. RENT is based on the idea of model ensembles discussed in [28]. Besides merely calculating the frequency of each feature, we also focus on the empirical distribution of the feature weights resulting from elastic net regularized models. Thereby, we extend the model ensemble framework to combine three rigid selection criteria: 1) how often is a feature selected?; 2) to which degree do the feature weights alternate between positive and negative values?; 3) are feature weights significantly different from 0? The final feature selection of RENT consists of the features that satisfy all three selection criteria. When required, the RENT framework can be extended with additional custom criteria to refine the feature selection process according to the user's a priori insights and requirements. By taking elastic net regularization into account, RENT aims at optimizing predictive performance and model stability simultaneously. In contrast, the concept of stability selection focuses on model stability as the primary target. We suggest a hyperparameter selection procedure based on the Bayesian information criterion (BIC) to balance the number of features and the predictive performance. In the experiments section, we explore

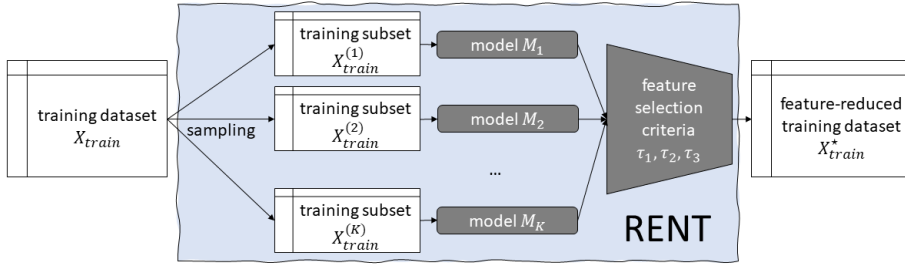


FIGURE 1. RENT feature selection pipeline. When using other feature selection methods, the blue frame is replaced by the other feature selectors, listed in Table 3.

and evaluate RENT extensively using real-world datasets for both classification and regression problems. In addition, we use the information provided by the ensemble models for an exploratory post-hoc analysis with statistical tools, including principal component analysis. Our implementation (in Python code) is publicly available and published in the Journal of Open Source Software [29].

II. REPEATED ELASTIC NET TECHNIQUE FOR FEATURE SELECTION

In this section, we present the methodological concept of RENT which relies on regularized logistic regression for binary classification problems and regularized linear regression for regression problems. We introduce the idea of elastic net regularization combined with repeated training of machine learning models on unique subsets of the training data to investigate feature selection stability. Finally, we define three quality metrics that influence the feature selection.

A. ENSEMBLE TRAINING AND SELECTION CRITERIA

Given a set of training data $X_{train} = \{x_i : i = 1, \dots, I_{train}\}$ where x_i denotes an object from the N -dimensional feature space, our concept builds on sampling K unique i.i.d. (independent and identically distributed) subsets $X_{train}^{(k)} \subset X_{train}$ of size $I_{train}^{(k)}$. As shown in Fig. 1, a regularized GLM M_k is trained on $X_{train}^{(k)}$ for each $k = 1, \dots, K$.

The evaluation of each model M_k is performed on the validation set $X_{val}^{(k)} = X_{train} \setminus X_{train}^{(k)}$ (here, \setminus denotes the set difference operator). To further improve robustness, we include the option to introduce more variation across the K models, by randomly varying the number of objects drawn from $X_{train}^{(k)}$ between the models within user-specified limits. For each feature n in X_{train} , $n = 1, \dots, N$, we observe the trained weights $\beta_{k,n}$ throughout models M_k , $k = 1, \dots, K$. For the purpose of feature selection, we acquire relevant information about the importance of feature n across all models from $\beta_n = (\beta_{1,n}, \dots, \beta_{K,n})$. All such vectors β_n , $n = 1, \dots, N$, are aggregated in a matrix B of dimension $(K \times N)$. Since all models comprise L1 regularization terms, the vectors of

feature weights β_n are typically sparse. However, entries are not constant due to 1) variations in the training subsets and 2) numerical deviations in the parameter optimization. Hence, a straightforward measure of feature relevance is the relative frequency $c(\beta_n)$, counting how often a feature was selected on average across the K models or, in other words, calculating the relative frequency as an estimate of the probability for the parameter of the n -th feature to be non-zero:

$$c(\beta_n) = \frac{1}{K} \sum_{k=1}^K \mathbb{1}_{[\beta_{k,n} \neq 0]}. \quad (2)$$

Furthermore, we observe two other empirical summary statistics of the feature parameter estimate distributions in the rows of B : the feature-specific mean $\mu(\beta_n)$ and variance $\sigma^2(\beta_n)$ of the feature weights

$$\mu(\beta_n) = \frac{1}{K} \sum_{k=1}^K \beta_{k,n}, \quad (3)$$

$$\sigma^2(\beta_n) = \frac{1}{K-1} \sum_{k=1}^K (\beta_{k,n} - \mu(\beta_n))^2. \quad (4)$$

In general, we consider the n -th feature to be a candidate for selection in RENT if

- 1) $c(\beta_n)$ is large, i.e. the feature is selected in many of the K elastic net models;
- 2) the estimates in β_n resulting from the K models do not alternate much between positive and negative signs (stability);
- 3) the mean of distribution resulting from the K parameter estimates in β_n is significantly non-zero.

These three simple and transparent requirements may be formulated in corresponding mathematical expressions, to form three quality metrics for assessing a feature n :

$$\tau_1(\beta_n) = c(\beta_n); \quad (5)$$

$$\tau_2(\beta_n) = \frac{1}{K} \left| \sum_{k=1}^K \text{sign}(\beta_{k,n}) \right|; \quad (6)$$

$$\tau_3(\beta_n) = t_{K-1} \left(\frac{|\mu(\beta_n)|}{\sqrt{\frac{\sigma^2(\beta_n)}{K}}} \right), \quad (7)$$

where $t_{K-1}(\cdot)$ denotes the cumulative distribution function of Student's t -distribution with $K - 1$ degrees of freedom.

Considering the second quality metric $\tau_2(\beta_n)$, the ideal case for feature n would be that all weights have the same sign—either all positive or all negative. In case of constant signs among all weights, $\tau_2(\beta_n)$ equals $\tau_1(\beta_n)$. Though, for a considerably large K , we should expect that at least slight sign variations for some features may occur. $\tau_2(\beta_n)$ simply allows the user to define a required minimum proportion of the parameter estimates to have the same sign. The third quality metric $\tau_3(\beta_n)$ —identifying consistently high model parameter estimates—is chosen such that it corresponds to the well-known statistical Student's t -test with rejection of the null hypothesis

$$H_0 : \mu(\beta_n) = 0.$$

In case that the null hypothesis holds, the test statistic

$$T = \frac{\mu(\beta_n)}{\sqrt{\frac{\sigma^2(\beta_n)}{K}}}$$

will follow a Student's t -distribution with $K - 1$ degrees of freedom. The deployed term evaluates the probability of the test statistic under the H_0 -distribution and thus, provides a thresholding at the chosen level of significance.

In order to define feature selection criteria from quality metrics $\tau_1(\beta_n)$, $\tau_2(\beta_n)$ and $\tau_3(\beta_n)$, we introduce corresponding cutoff values $t_1, t_2, t_3 \in [0, 1]$. Specifically, a feature $n \in F$ is added to the selected feature set F^* , if it satisfies all three criteria: $\tau_i \geq t_i, \forall i \in \{1, 2, 3\}$. Further criteria can be included by the user if necessary. In the provided setup, these quality metrics may be considered as hyper-parameters of the RENT method, allowing the user to regulate the feature selector, by tuning the thresholds t_1, t_2 and t_3 . The cardinality of the selected features F^* will increase, if any of these thresholds are decreased and vice versa. All three metrics, τ_1, τ_2 and τ_3 , are bounded by the interval $[0, 1]$, which facilitates the specification of appropriate thresholds. Since τ_3 can be associated with a Student's t -test, the threshold t_3 for a 5% or 1% significance level, corresponds to the thresholds $t_3 = 0.95$ and $t_3 = 0.99$, respectively.

B. HYPERPARAMETER SELECTION

RENT involves hyperparameters at different stages of the method: before training the elementary models, regularization parameters γ and α control the restrictiveness of the feature selection in the ensemble, followed by the parameters t_1, t_2, t_3 determining the final feature set. Thereby, the latter cutoff parameters are (a) dependent on the choice of the regularization parameters, and (b) have mutual dependencies.

Hyperparameter selection is commonly performed using an additional validation dataset or cross-validation—both options are not optimal for RENT, since a validation subset

would reduce the number of objects in a high-dimensional dataset even further, and cross-validation would add a substantial computational burden to the procedure. Thus, we deploy an alternative approach from statistical model selection: the Bayesian information criterion (BIC) delivers a trade-off between the information content (quantified as the likelihood) of the model and the model complexity in terms of the number of estimated parameters [30]. BIC is defined as

$$\text{BIC} = -2 \log \hat{\mathcal{L}} + I_{\text{train}} \log \rho, \quad (8)$$

where $\hat{\mathcal{L}}$ denotes the estimated likelihood of the predictive model, and ρ denotes the number of estimated model parameters. In contrast to similar information measures like the Akaike information criterion (AIC), BIC is known for stronger penalization of model complexity leading to a lower number of selected features, which is favorable in the case of RENT. By minimizing BIC, models with high information content and low complexity are favored. In ordinary linear regression models and other standard GLMs, the number of estimated model parameters equals the number of variables, i.e., features, plus one parameter for the offset β_0 ; thus, we set $\rho = |F| + 1$. The likelihood $\hat{\mathcal{L}}$ can be determined from the distribution assumptions of the GLM model, such as the normal distribution of errors in the ordinary least squares regression model, resulting in the sum of squared errors (SSE) as negative log-likelihood function.

RENT uses a two-step hyperparameter estimation procedure: a grid search for regularization parameters α and λ with BIC as target function is performed on the full training dataset first (step 1). Then, the RENT ensemble is trained given the best regularization parameter combination. Finally, in step 2 another grid search for cutoff parameters t_1, t_2, t_3 is performed using the same concept as in step 1.

C. TRAINING RUNTIME COMPLEXITY OF RENT

Since RENT is an ensemble method built on GLMs as elementary models, the runtime complexity of RENT is expressed as a multiple of the runtime complexity of GLMs, denoted by \mathcal{O}_{GLM} . In essence, \mathcal{O}_{GLM} depends on the applied type of GLM, the parameter optimization algorithm, and the implementation. For instance, a runtime complexity of $\mathcal{O}_{GLM} = \mathcal{O}(N^3 + I_{\text{train}} \cdot N^2)$ is reported for Lasso by reducing the computation to solving a least squares regression problem [31]. Variants using iterative algorithms are rather judged by the overall experimental runtime and the runtime complexity per update cycle, while the number of iterations is hard to determine a priori—such information is provided for GLMs with elastic net regularization in [32].

Given the first variant, RENT runs an ensemble comprising K independent GLMs, each trained on a number of N features, which delivers a complexity of

$$\mathcal{O} \left(KN^2 \cdot (N + I_{\text{train}}^{(K)}) \right),$$

TABLE 1. Classification (class.) and regression (reg.) datasets used for evaluation of the feature selection methods.

| task | acronym | dataset | source | # feat. | size # obj. (train/test) | class balance | |
|--------|----------|----------------------------|-----------|----------|-----------------------------|------------------------|------------------------|
| | | | | | | % class 0 (train/test) | % class 1 (train/test) |
| class. | c0 | synthetic dataset | simulated | 1000 | 175/75 | 0.50/0.47 | 0.50/0.53 |
| | c1 | MNIST _{0,1} | [34] | 784 | 12 665/2115 | 0.47/0.46 | 0.53/0.54 |
| | c2 | MNIST _{4,9} | | 784 | 11 791/1991 | 0.50/0.49 | 0.50/0.51 |
| | c3 | Breast cancer Wisconsin | [35] | 30 | 399/170 | 0.62/0.65 | 0.38/0.35 |
| | c4 | Dexter text classification | [36] | 20 000 | 300/300 | 0.5/0.5 | 0.5/0.5 |
| c5 | OVA Lung | [37] | 10 935 | 1083/462 | 0.92/0.92 | 0.08/0.08 | |
| reg. | r0 | synthetic dataset | simulated | 1000 | 175/75 | - | - |
| | r1 | Milk protein dataset | [38] | 6179 | 45/45 | - | - |

where $J_{train}^{(K)} < I_{train}$ denotes the sample size of each subset during RENT training. In addition, hyper-parameter tuning requires training c GLMs, where c is a constant given by the number of level combinations for regularization and cutoff parameters, resulting in

$$\mathcal{O}\left(cN^2 \cdot (N + I_{train})\right).$$

In total, an upper bound to the full runtime complexity of RENT is given by

$$\mathcal{O}\left((K + c) \cdot N^2 \cdot (N + I_{train})\right). \quad (9)$$

III. EXPERIMENTS

We demonstrate the potential of RENT as a feature selection method through experiments on multiple datasets. First, we verify the overall concept in a validation study in Section III-D. Second, we evaluate the performance of RENT in comparison with seven feature selection methods and a baseline elastic net regularized model, in Section III-E. Based on one dataset, we illustrate how the stability of RENT behaves compared to the stability of established ensemble methods based on the number of unique elementary models $K \in \mathbb{N}$.

A. EXPERIMENTAL SETUP AND DATASETS

Experiments are conducted on multivariate datasets from various domains, including real-world data and synthetic data for both binary classification (class.) and regression tasks (reg.); datasets are listed in Table 1. The size of each dataset is denoted via the number of features (#feat) and the number of objects (#obj) divided into train and test sets (train/test). Train-test-splits are performed by stratified random sampling. Further, the *class balance* indicates the percentage of class representation for each classification dataset (train/test).

The broad selection of use cases, including high-dimensional datasets, demonstrates the flexibility and applicability of RENT. Simulated datasets (c0) and (r0) were produced using scikit-learn [33] functions *make_classification* and *make_regression*, respectively. For the MNIST dataset, two binary classification problems are defined by restricting the classes: MNIST _{c_1, c_2} indicates that

only instances from classes c_1 and c_2 , where $c_1, c_2 \in \{0, \dots, 9\}$, were used, ignoring objects from other classes.

A feature selector is trained on X_{train} , then the training data X_{train} is projected into the subspace spanned by the selected features. This column-reduced training dataset is denoted by X_{train}^* . In our experiments we train an unregularized linear/logistic regression model M^* on X_{train}^* . Evaluation is based on the predictive performance obtained from M^* on the previously unseen test data X_{test} . It is important to note, however, that it may be necessary to use regularization for the model M^* to avoid overfitting, especially if the reduced X_{train}^* has more features than objects.

B. EVALUATION METRICS

We use two different measures for quantitative evaluation of the prediction performance in classification settings: F1 score (F1) and Matthews correlation coefficient (MCC) [39]. The F1 score represents the harmonic mean of precision (PR) and recall (RC). Denoting the entries of the confusion matrix by TP (true positive), FP (false positive), FN (false negative), and TN (true negative), the performance measures are defined as follows:

$$PR = \frac{TP}{TP + FP}; \quad (10)$$

$$RC = \frac{TP}{TP + FN}; \quad (11)$$

$$F1 = 2 \cdot \frac{PR \cdot RC}{PR + RC}; \quad (12)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{P \cdot (TP + FN) \cdot (TN + FP) \cdot N}}, \quad (13)$$

where $P = TP + FP$ and $N = TN + FN$ are the sums of the predicted positives and negatives, respectively. Note that F1 scores can be calculated for both class labels, depending on which class is considered as “positive”. F1 is more appropriate than accuracy for imbalanced class distributions because the larger class dominates the latter. A disadvantage of F1 is that it does not take into account TN. Therefore, MCC provides more representative results if both classes are equally relevant in the prediction problem and the number of TN objects is high. F1 score, precision, and recall are bounded

between $[0, 1]$, where 0 represents a complete disagreement between predicted and actual class, and 1 denotes a perfect match. MCC is bounded between $[-1, 1]$, where -1 denotes that all objects are classified incorrectly, 0 indicates complete randomness, and 1 denotes correct classification of each object, respectively.

For regression problems, we evaluate the root mean squared error of prediction (RMSEP) on the test dataset X_{test} with cardinality I_{test} , defined as

$$RMSEP = \sqrt{\frac{1}{I_{test}} \sum_{i=1}^{I_{test}} (y_i - \hat{y}_i)^2}, \quad (14)$$

and the coefficient of determination (R^2) [40]

$$R^2 = 1 - \frac{\sum_{i=1}^{I_{test}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{I_{test}} (y_i - \bar{y})^2}, \quad (15)$$

where y_i represents the true output of object x_i , \hat{y}_i represents the prediction of y_i and \bar{y} represents the mean of the outputs y_i , $i \in \{1, \dots, I_{test}\}$. While $RMSEP$ is always non-negative we seek its minimization. R^2 on the other hand may take negative values but has an upper bound of 1 (associated with perfect predictions) and we therefore seek its maximization.

Besides predictive performance, selection stability is assessed using a measure suggested in [24] evaluating the different outcomes of multiple feature selection runs in a combinatorial way. Specifically, the suggested measure computes a ratio between the sample variance of observed feature frequencies and the theoretical variance, given that the feature selector is stable (null hypothesis). The authors clarify that their measure fulfills five consistency criteria and is asymptotically bounded by the interval $[0, 1]$, where 1 denotes optimal stability. Their concept of measuring feature selection stability by aggregating multiple independently trained models underlines the relevance of our ensemble approach and supports the idea to achieve stability by combining K independent feature selection model runs.

C. RENT HYPERPARAMETER SELECTION

In Section II-B we introduce hyperparameter selection for both the elastic net modeling and the three cutoff parameters based on the BIC. More precisely, we evaluate the elastic net hyperparameter combinations of $\gamma \in \{1e^{-2}, 1e^{-1}, 1\}$ and $\alpha \in \{0, 0.1, 0.25, 0.5, 0.75, 0.9, 1\}$. To find the best combination concerning BIC, we train a single logistic/linear regression model with each pairwise combination of hyperparameters γ and α on the training dataset. After determining optimal elastic net parameters γ and α for a given dataset in Table 1, all ensemble models M_1, \dots, M_K in RENT are trained with these parameters. Once all K models are fitted on their respective training subsets $X_{train}^{(k)}$, we select the cutoff hyperparameters t_1, t_2 and t_3 with BIC, in the same way as for the elastic net hyperparameter search. For this purpose

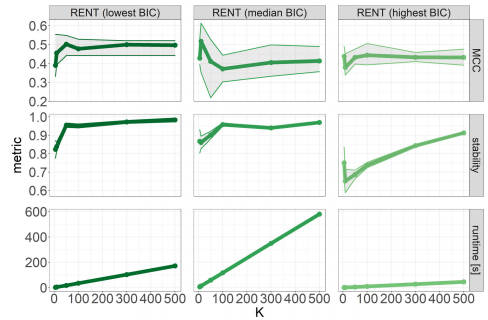


FIGURE 2. Comparison of RENT with different hyperparameter setups (elastic net regularization and cutoff) for dataset c0 to varying numbers of ensemble models K . Each setup is evaluated in 30 independent runs. The plot shows mean values (bold line) and empirical 2.5% (lower line) and 97.5% (upper line) quantiles.

we perform a grid search on $t_1 \in [0.2, 1]$ with stepsize 0.05, t_2 within the same range and $t_3 \in \{0.9, 0.95, 0.975, 0.99\}$, representing different significance levels in the t -test. A comparison of three different hyperparameter settings based on the lowest, median, and highest BIC values is shown for dataset c0 in Fig. 2 for a varying number of elementary models $K \in \{5, 10, 50, 100, 300, 500\}$. For each hyperparameter setting (t_1, t_2, t_3) leading to the lowest, median and highest BIC, respectively, 30 independent runs of RENT are carried out. Each run is conducted on the same training dataset but with a distinct (random) model weight initialization. Performance is measured via the MCC; runtimes are given in seconds and refer to one single run for each method. Across all 30 independent runs, the mean is calculated together with the empirical 2.5% and 97.5% quantiles (corresponding to a two-sided 5% confidence interval) for stability, performance, and runtime, respectively.

We can observe that, as expected, the setup of RENT with optimal hyperparameters (lowest BIC) outperforms those settings with median and highest BIC and achieves the highest stability. Especially RENT based on hyperparameter settings (t_1, t_2, t_3) with the highest BIC is unstable, even though the performance remains in an acceptable range. Regarding runtime, it takes about 600 seconds for RENT with median BIC to run a single model for $K = 500$, which is much longer than for the other two settings. A reason for this might be a harder optimization task for specific hyperparameter combinations where it takes more steps for the logistic regression model to converge.

In summary, we observe the following behavior of RENT with lowest BIC at an increasing number of models K :

- on average, good MCC and stability are achieved simultaneously, even with low K ;
- as expected, stability increases significantly from 0.75 for $K = 5$, saturating at a value close to 1;
- average MCC shows little change from $K = 100$ to $K = 500$;

TABLE 2. Prediction results per dataset of the validation study (MCC for c0-c5, R^2 for r0 and r1) showing the total number of features, the number of features selected with RENT (Δ), and the performance metrics. The column RENT gives the MCC/ R^2 of a predictive model trained after feature selection. ^a ≥ 0.99 .

| data-set | # features total | # features Δ | MCC / R^2 | | |
|----------|------------------|---------------------|-------------------|-------|-------|
| | | | RENT | (VS1) | (VS2) |
| c0 | 1 000 | 12 | 0.50 | 0.07 | -0.02 |
| c1 | 784 | 37 | 0.99 ^a | 0.98 | 0.00 |
| c2 | 784 | 124 | 0.95 | 0.81 | 0.00 |
| c3 | 30 | 4 | 0.96 | 0.81 | -0.01 |
| c4 | 20 000 | 5 | 0.33 | 0.00 | 0.00 |
| c5 | 10 935 | 16 | 0.91 | 0.23 | 0.00 |
| r0 | 1 000 | 28 | 0.99 | -0.19 | -0.90 |
| r1 | 6 179 | 8 | 0.71 | - | - |

- runtime increases linearly with the number of trained models.

Hence, our results support the analysis in [4] that repeated use of regularized elastic net models is useful to achieve stable and reproducible results, while keeping the predictive performance at a high level. Our observations further indicate, that no major benefit can be achieved by increasing the number of models to more than approximately 100 with respect to the observed metrics on the given dataset. Therefore, $K = 100$ seems to be a valid default regarding the trade-off between stability and time for the datasets used in this study. If computation time is critical, the user may set K to a lower number but needs to consider that the distribution of the weights may be insufficiently covered and that this may have an impact on the stability of feature selection.

Alternatively, instead of using BIC, the user may set hyperparameters γ and α manually or use cross-validation to obtain a customized trade-off between predictive performance, stability, and the number of selected features. Note that this approach may be more subjective and that the computational cost can be higher than using BIC, especially if cross-validation is used.

D. VALIDATION STUDY OF FEATURES SELECTED WITH RENT

To demonstrate the validity of features selected with RENT, we apply two validation study setups (VS1) and (VS2). In (VS1) we draw random features, while in (VS2) we randomly permute labels of the test dataset. In both cases, we build logistic regression models, predict on an unseen test dataset and compare MCC scores to predictions based on features selected by RENT. The comparisons are performed via one-sided Student's t -tests where the null hypotheses claim that the MCC of RENT is lower or equal to the average MCCs obtained from (VS1) or (VS2), respectively. For regression datasets, the analog procedure is applied using R^2 as a quality metric. Both tests are conducted at a significance level of 0.05.

(VS1) Compare a number of $\ell \in \mathbb{N}$ randomly selected feature sets, representing inefficient feature selections,

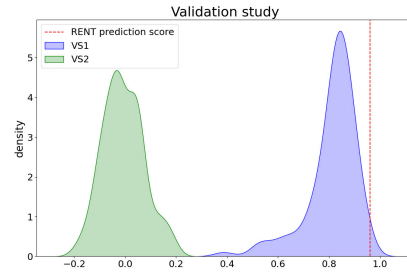


FIGURE 3. Empirical distributions of MCC scores in studies (VS1) and (VS2) on dataset c3 represent the validation study's results. The red line indicates the MCC based on RENT features.

to the features selected by RENT. The steps of the procedure are:

- sample ℓ independent, random feature subsets from X_{train} , containing Δ features each, where Δ corresponds to the number of features selected by the RENT approach
- train a new model for each of the ℓ feature sets by restricting X_{train} to those features
- predict the labels of X_{test} with each of the ℓ models and compute MCCs
- perform a Student's t -test, assuming as null hypothesis that the MCC value obtained from RENT is drawn from the same distribution

(VS2) Compare the predictive performance of a model based on features selected with RENT on the real X_{test} labels, to the predictive performance of ℓ randomly permuted labels of X_{test} . The steps of the procedure are:

- train a model on X_{train} with the features selected with RENT
- randomly permute y_{test} ℓ -times and compute the average MCC over the ℓ permutations
- perform a Student's t -test, assuming as null hypothesis that the MCC value obtained from RENT is drawn from the same distribution

Performance results from (VS1) and (VS2) provide a reliable indicator of whether models based on features selected by RENT perform better than models based on randomness. Table 2 shows the average MCC of (VS1) and (VS2) in the columns MCC/ R^2 . All corresponding p -values from the Student's t -tests are significantly lower than 0.05, mostly below $1e^{-15}$, where ℓ equals 100. Since the standard deviation of the mean decreases with the sample size, a higher explanatory power of the Student's t -tests can be achieved by setting ℓ to a larger value. However, the runtime increases linearly in ℓ . The estimated densities of (VS1) and (VS2) for the Breast cancer Wisconsin dataset (c3) are plotted in Fig. 3.

In general, these two validation studies are not limited to RENT and may be applied to other feature selection methods

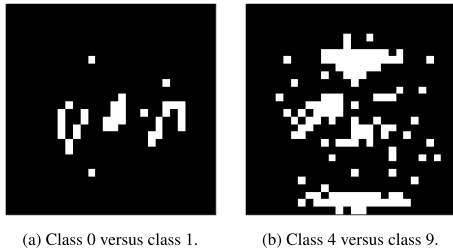


FIGURE 4. Visualization of features selected by RENT for the MNIST datasets c1 (class 0 versus class 1) and c2 (class 4 versus class 9) from the 28×28 images of the numbers. Selected features are colored in white.

and metrics, as well. Overall, the null hypotheses in both VS1 and VS2 were rejected for all datasets, indicating that RENT performs significantly better than models based on randomness as described in the validation approaches. The experimental results in Table 2 show that (VS2) is close to zero across all datasets, as one would expect from the experimental setup. (VS1) performance is similar to the performance of RENT models for datasets c1. This fact may be explained by the individual information content of each feature: especially two-class subsets extracted from MNIST contain many mutually or highly correlated features. Therefore, many different feature combinations lead to good predictions.

RENT results for MNIST (datasets c1 and c2) are visualized in Fig. 4. We observe that 1) different features are relevant for distinguishing the class pairs 0-1 and 4-9 and 2) features relevant for 0-1 are typically located in the center of the image, whereas those relevant for 4-9 are more distributed across the image. Overall, distinguishing between 4 and 9 is more complex. Therefore, the number of selected features is much higher in this case than when classifying the numbers 0 and 1.

E. COMPARISON OF RENT WITH ESTABLISHED FEATURE SELECTORS

The validation study in Section III-D showed that RENT is a valid feature selection approach for all datasets used in this study. Hence, we compare RENT to the methods listed in Table 3 as follows: 1) seven established feature selectors applied to classification datasets; 2) five feature selectors applied to regression datasets; 3) a baseline logistic/linear regression model M° with elastic net regularization [17]. For each feature selector, software implementations are publicly available. To compare RENT to traditional filter methods, we consider the Laplacian score (L-score) [8], Fisher score (F-score) [7], mRMR [10], and a representative of the relief family, reliefF [41]. Specifically, we select the top features according to the scores provided by each filter method. Further, we study the behavior of recursive feature elimination (RFE) [15] representing a wrapper based approach. Finally, our comparison also involves two prototypes of

TABLE 3. Established feature selection techniques representing benchmarks for the experimental evaluation of RENT.

| method | implementation |
|---------------------------|----------------|
| elastic net (M°) | Python [33] |
| StabSel | R [43] |
| RFC/RFR | Python [33] |
| L-score | R [44] |
| F-score | R [44] |
| mRMR | R [45] |
| ReliefF | Python [46] |
| RFE | Python [33] |

state-of-the-art ensemble feature selectors: stability selection (StabSel) [28] and the random forest [42], which can be used for both classification (RFC) and regression (RFR) problems.

In contrast to other methods, RENT and M° share the advantage that the user does not have to specify the size of the selected feature set as input. Instead, the number of selected features is indirectly controlled via the elastic net regularization parameters γ and α . Similarly, for StabSel the exact number may be specified optionally. Otherwise, it is determined indirectly by a cutoff and an upper bound per-family error rate (PFER) [43]. For fair performance comparison of the remaining investigated methods, the size of the selected feature set is set to the number of features returned by RENT, denoted as Δ .

For StabSel, we perform a 5-fold cross-validated grid search to estimate adequate parameter settings. The elementary feature selection method is the logistic regression model with L1 regularization; the number of models equals K . Furthermore, we perform a grid search for stability selection on the interval [0.6, 0.9] for the cutoff value and [0.05, 0.95] for the PFER value, with a 0.05 step size each. In our study, the random forest serves as a filter, delivering a ranking of the features. The features with the highest ranks are selected and used as input for M^* where the number of the selected features corresponds to the number of features Δ selected by RENT. To fit the random forest model, we set the number of unique trees to K . Other parameters are set to the defaults. Computations are performed on standardized train datasets. All model parameters used for the established methods, such as the neighborhood graph construction in L-score or the step size in RFE, are set to the default values, except for c4 and c5, where the step size is increased to 100 in order to obtain results in a moderate runtime. The regularization parameters γ and α for M° are set to those used for RENT. The results for all datasets and methods are provided in Table 4 for binary classification datasets and in Table 5 for regression datasets.¹

For classification problems, the results achieved with RENT feature selection are competitive with the best results of the other methods, yielding better or equally high F1 scores for predicting class 0 in five out of six datasets. For c0, the performance is only 0.01% below the top value of 0.75.

¹The GitHub repository https://github.com/annajenul/RENT_article_results stores example code to reproduce the results.

TABLE 4. F1 scores and MCC results for classification datasets. ^a ≥ 0.99 , ^b returned error.

| | M° | RENT | StabSel | RFC | L-score | F-score | mRMR | relieFF | RFE |
|-------------------|-------------------------|-------------------------|-------------------------|-------------------------|---------|-------------------------|---------------|-------------|-------------------------|
| F1 class 0 | | | | | | | | | |
| c0 | 0.65 | 0.74 | 0.69 | 0.75 | 0.67 | 0.75 | 0.74 | 0.56 | 0.66 |
| c1 | 0.99^a | 0.99^a | 0.99^a | 0.99^a | 0.43 | 0.99^a | ^{-b} | 0.01 | 0.99^a |
| c2 | 0.97 | 0.97 | 0.93 | 0.96 | 0.85 | 0.94 | 0.64 | 0.85 | 0.97 |
| c3 | 0.97 | 0.97 | ^{-b} | 0.93 | 0.89 | 0.93 | 0.95 | 0.87 | 0.90 |
| c4 | 0.72 | 0.72 | 0.71 | 0.72 | 0.01 | 0.72 | ^{-b} | 0.27 | 0.25 |
| c5 | 0.99 | 0.99 | 0.98 | 0.99 | 0.96 | 0.99 | 0.99 | 0.99 | 0.98 |
| F1 class 1 | | | | | | | | | |
| c0 | 0.63 | 0.75 | 0.72 | 0.75 | 0.72 | 0.75 | 0.75 | 0.53 | 0.7 |
| c1 | 0.99^a | 0.99^a | 0.99^a | 0.99^a | 0.76 | 0.99^a | ^{-b} | 0.70 | 0.99^a |
| c2 | 0.97 | 0.97 | 0.93 | 0.96 | 0.87 | 0.95 | 0.79 | 0.85 | 0.97 |
| c3 | 0.98 | 0.99 | ^{-b} | 0.96 | 0.94 | 0.96 | 0.97 | 0.93 | 0.95 |
| c4 | 0.50 | 0.47 | 0.51 | 0.47 | 0.67 | 0.47 | ^{-b} | 0.69 | 0.69 |
| c5 | 0.94 | 0.92 | 0.71 | 0.90 | 0.26 | 0.85 | 0.93 | 0.86 | 0.74 |
| MCC | | | | | | | | | |
| c0 | 0.29 | 0.50 | 0.41 | 0.50 | 0.38 | 0.50 | 0.50 | 0.10 | 0.36 |
| c1 | 0.99^a | 0.99^a | 0.99^a | 0.99^a | 0.39 | 0.99^a | ^{-b} | 0.04 | 0.99^a |
| c2 | 0.94 | 0.95 | 0.86 | 0.92 | 0.72 | 0.89 | 0.54 | 0.70 | 0.94 |
| c3 | 0.95 | 0.96 | ^{-b} | 0.89 | 0.83 | 0.89 | 0.92 | 0.80 | 0.86 |
| c4 | 0.33 | 0.33 | 0.33 | 0.33 | 0.00 | 0.33 | ^{-b} | 0.23 | 0.23 |
| c5 | 0.93 | 0.91 | 0.70 | 0.89 | 0.28 | 0.84 | 0.92 | 0.86 | 0.71 |

TABLE 5. RMSEP and R^2 results for regression datasets.

| | M° | RENT | StabSel | RFR | L-score | mRMR | RFE |
|-------------------------|--------------|-------------|---------|--------|---------|-------|--------|
| RMSEP | | | | | | | |
| r0 | 21.73 | 24.01 | 79.21 | 150.98 | 237.37 | 93.11 | 123.22 |
| r1 | 0.12 | 0.15 | 0.16 | 0.16 | 0.32 | 0.17 | 0.23 |
| R^2 | | | | | | | |
| r0 | 0.99 | 0.99 | 0.86 | 0.47 | -0.30 | 0.80 | 0.65 |
| r1 | 0.82 | 0.71 | 0.68 | 0.66 | -0.35 | 0.62 | 0.34 |

Also, for class 1, RENT achieves the highest performance for four out of six datasets. For dataset c5 the performance is close to the best F1 score. MCC, which is more robust than the F1 score for unbalanced class settings, is highest for five out of six datasets with RENT feature selection. For c5, M° has a higher MCC, but with a much higher number of features (593 features) than RENT (16 features). With the regression datasets, RENT achieves a performance superior to StabSel, RFR, L-score, mRMR, and RFE and competitive performance to M° for both measures, RMSEP and R^2 .

In many cases, M° is not able to restrict the number of features as efficiently as RENT. Using the same regularization parameters as for RENT, M° selects the following number of features: 290 (c1) vs. 37 for RENT and 593 (c5) vs. 16 for RENT, respectively (see Table 2).

Overall, StabSel achieves good results for all datasets underlining the merits of ensemble feature selection concepts. Note that no results could be obtained for dataset c3 since no feature reached a sufficient selection frequency across all models. The random forest yields competitive results for most datasets in classification (RFC) setups but performs noticeably worse in regression (RFR) setups. L-score and mRMR appear to provide weak performance scores compared to their competing feature selectors. For L-score, the low scores can be explained by its unsupervised setup, which makes it harder

to relate the model to any target variable. With mRMR, especially the performances for c1, c2, and c4 are weak. For c1 and c4 this weakness can partly be attributed to the available implementation, which produced an error for these datasets (denoted with superscript ^b in Table 4). On the other hand, F-score performs well, especially for predicting class 0. The relieFF method achieves good results for c5 but is among the poorest feature selectors for c0, c1, c2 and c3. Neither F-score nor relieFF are applicable to regression problems using the available implementations. Dataset c4 is of particular interest since opposite behavior can be observed among the feature selectors. RENT, StabSel, RFC, and F-score perform well when predicting class 0, whereas the other methods achieve higher scores for class 1. Hence, we assume that the features selected from c4 introduce a bias towards class 0 or class 1, respectively. In terms of MCC, which accounts for both classes in parallel, the best results are achieved by RENT, StabSel, RFC, and F-score.

Fig. 5 depicts the experimental results for comparing the ensemble feature selectors with varying K , given the same setup as Fig. 2. While RFC achieves a similar performance as RENT, it is the most unstable ensemble approach for lower number of trees K . Even for high K , RFC never achieves the same stability as RENT and StabSel. Furthermore, RFC has the highest variance in MCC. On the other hand, StabSel

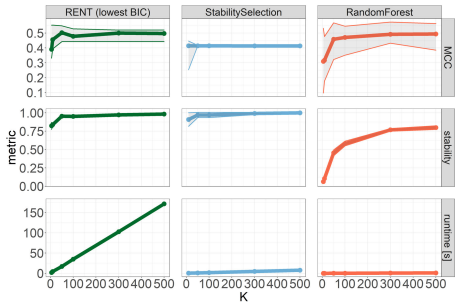


FIGURE 5. Comparison of stability, performance, and runtime of three ensemble based feature selectors, where K is the number of elementary models in RENT.

reaches higher stability but lower MCC scores than its competitors between $K = 50$ and $K = 100$. The provided stability analysis underlines the strong properties of RENT, compared to random forests, which are known to be unstable in multiple scenarios [47].

Regarding the computational costs² of the ensemble feature selectors in our study, Fig. 5 demonstrates that the runtime increases linearly in K for all methods. RENT takes longer to compute which might be caused by the fact that the implementation does not yet exploit the full potential for runtime optimization and different implementations and programming languages were used for elementary operations.

IV. EXPLORATORY POST-HOC ANALYSIS

As an ensemble model approach, RENT offers additional information that can be integrated into exploratory post-hoc analyses. The two post-hoc analyses presented in this section give the user tools to 1) further investigate objects in the dataset and identify which of those are difficult to predict and which not; 2) exploit this information in a principal component analysis model trained on the selected features to understand why some objects are difficult to predict.

A. ANALYSIS OF TRAINING OBJECTS

Based on the ensemble of elementary models in RENT, it is possible to compute summary statistics on a single-object level. Such information may contribute to improved interpretability of the model in general and single objects in the data in particular. For this purpose, we analyze the predictions of individual objects across all models M_k , $k = 1, \dots, K$. For binary classification problems, we observe the distribution of correct and incorrect classifications of single objects in X_{val}^k , and thereby gain insights into the consistency of assigning an object to its true class. From a statistical perspective, this means that we can identify objects with deviating properties belonging to the same class based on the information whether the label of an object is difficult to predict or not. For regression problems, we similarly use the

²All results were acquired by running R 4.1.1 and Python 3.8.10 on a Windows 10 machine with a 4-core Intel i5 CPU 1.8 GHz and 512 GB RAM.

TABLE 6. In-depth analysis of predictions for four patients from the Breast cancer Wisconsin dataset (dataset c3), see Fig. 6. # val set denotes how often the object was part of a validation set (between 1 and $K = 100$), true class is the true class, # incorrect describes how often the object was incorrectly predicted and % incorrect is the corresponding percentage.

| object | # val set | true class | # incorrect | % incorrect |
|--------|-----------|------------|-------------|-------------|
| 3 | 23 | 0 | 0 | 0 |
| 6 | 26 | 0 | 26 | 100 |
| 78 | 28 | 1 | 2 | 7.1 |
| 102 | 24 | 0 | 13 | 54.2 |

mean absolute errors. Below, we will exemplify the proposed post-hoc analysis for dataset c3.

Given an object $x_i \in X_{val}^k$, the logistic regression model M_k outputs a class probability \hat{y}_i of x_i being assigned class 1 (ProbC1). Among the K models built within RENT, we obtain a ProbC1 value each time an object x_i appears in X_{val}^k , $k = 1, \dots, K$. Aggregating this information by object, we can derive statistics and describe the distribution of the ProbC1s for each object $x_i \in X_{val}^k$ by a histogram, as shown in Fig. 6. These results are generated from dataset c3 (Breast cancer Wisconsin), where we denote a single object in the dataset as a cancer patient. Incorrect predictions provide evidence for patients that are hard to classify or show different characteristics compared to patients from the same class that are easy to classify. We observe that patient 3 belongs to class 0 and that the predicted probabilities of patient 3 are consistently below 0.5, which is the standard decision boundary for logistic regression models. In other words, patient 3 is predicted correctly every time she is part of X_{val}^k . Patient 6 belongs to class 0; however, the predicted probabilities are consistently above 0.5, meaning that her class label is always mispredicted. For patients 78 and 102 we observe probabilities both above and below 0.5, indicating that the class predictions of these two patients are rather uncertain, however, to a different degree. Fig. 6 reflects the detailed information provided in Table 6.

With a % incorrect of 54.2%, the class predictions for patient 102 are extremely unstable among the 24 models, where this patient was part of the validation set. This type of information on prediction stability may provide a good starting point for detailed studies on how a hard-to-classify object differs from objects that are consistently assigned to the correct class. Thus, it could be of high relevance, inter alia for medical experts, who may identify patients with deviating data characteristics. These difficulties may arise from dominating phenomena in the measured features or measurement errors.

In this way, ensemble based approaches such as RENT allow in-depth analysis of the distribution of class probabilities rather than restricting to single class predictions.

B. PRINCIPAL COMPONENT ANALYSIS (PCA) ON SELECTED FEATURES

By a PCA [48] of X_{train}^* , we can obtain a better understanding of the properties of objects and their relation to the features selected by RENT. Note, that unlike in machine learning,

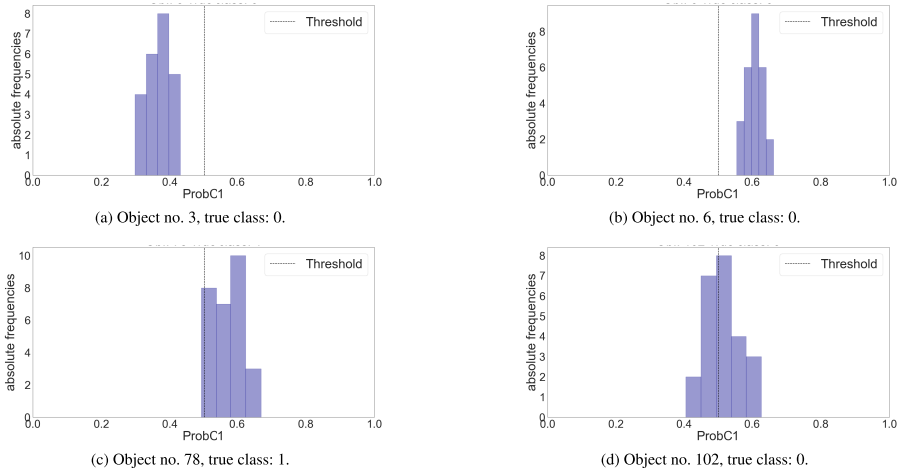
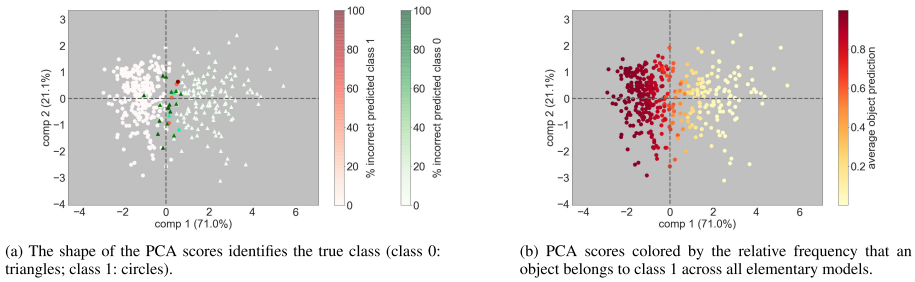


FIGURE 6. Distribution of the class probability \hat{y}_j of x_j being assigned class 1 (ProbC1s) for objects (patients) 3, 6, 78, and 102 estimated by K different models of RENT. The first axis shows the ProbC1s, the second axis shows the absolute frequencies.



(a) The shape of the PCA scores identifies the true class (class 0: triangles; class 1: circles).

(b) PCA scores colored by the relative frequency that an object belongs to class 1 across all elementary models.

(c) Correlation loadings plot.

FIGURE 7. PCA scores and correlation loadings of the Breast cancer Wisconsin dataset after RENT feature selection. The scores in Fig. 7a and 7b provide an overview of how the objects are distributed in the subspace spanned by components 1 and 2. The correlation loadings in Fig. 7c indicate how the selected features contribute to the variance explained by components 1 and 2.

where PCA is often only used for feature extraction, visualization (plotting) of PCA scores, PCA loadings, and PCA correlation loadings [49] may be efficient for the purpose of model interpretation. Fig. 7a and Fig. 7b show the scores³

³The calculations rely on the PCA implementation provided in the Python package “hogorm” package [50].

of the first two principal components (comp 1 and comp 2) applied to the Breast cancer Wisconsin dataset, but with hues based on different information acquired from the ensemble. Every data point in the scores plot represents one object in the data or specifically to this dataset, one patient. The first two principal components explain 92.1% of the total variance in the data that are contained in the selected features.

The remaining 7.9% are explained by the remaining principal components. In particular, Fig. 7a shows how the objects are distributed in the sub-space spanned by components 1 and 2. The two classes are well separated, with the objects of class 1 (circles) on the left side of the plot and class 0 (triangles) on the right side. Using results from RENT, the information in this plot may be further enhanced by coloring each object by its true class (class 0 - green triangles; class 1 - red circles) graded according to % incorrect in Table 6. Higher color saturation refers to a higher percentage of incorrect label predictions—suggesting that the object shows an anomalous behavior, which the model cannot sufficiently cover. In this example, objects with ambiguous classes accumulate in the middle area, most of them are close to the intuitive decision boundary. Fig. 7b shows the same scores as seen in Fig. 7a. However, the objects are colored by their average ProbC1 (the average probability of an object belonging to class 1). Again, objects with either a very high or a low value cluster on the right and the left-hand side, respectively. Objects—which we know are difficult to classify—with scores for comp 1 ranging between -0.5 and 1 are located in the center of the image. PCA can also be performed on each class separately, to investigate within-class variations, as shown in extensive Jupyter notebook examples in the RENT GitHub repository.

In addition to the PCA scores, the correlation loadings plot is shown in Fig. 7c, where every point represents one feature in the plane spanned by comp 1 and comp 2. The correlation loadings plot encodes 1) the level of contribution of the selected features to each of the components 1 and 2, and 2) how much of the variance in each feature is explained by the two components. The further away a correlation loading is located from the origin, the higher the amount of explained variance for the feature it represents. The inner and outer circles represent 50% and 100% of explained variances, respectively. Among the four selected features, feature 8, 21 and 28 contribute most to the first component that separates the two classes. It is also evident that these three features are highly correlated, as they are located so close to each other in Fig. 7c. Moreover, they are close to the outer ring, meaning that comp 1 explains nearly 100% of the variance in those features. Feature 22 contributes to both components 1 and 2, but is the feature that contributes most to component 2. By superimposing the scores onto the correlation loadings plots, we can gather information on how the scores and features are interrelated. Features 8, 21 and 28 and objects of class 0 are in the same regions (right side)—indicating that objects of class 0 have high values for these features, while objects of class 1 on the opposite side (left side) have low values for those features.

The above examples of post-hoc analysis illustrate how combining ensemble information with exploratory analysis by PCA, can provide deeper insight into the data.

V. DISCUSSION

In summary, RENT performs well on all experimental datasets presented in this study when compared to the other

feature selection methods. In particular, a good trade-off between predictive performance and selection stability is achieved. We observe that 1) RENT is consistently among the best performing methods, 2) if outperformed by others, the difference in performance is mostly negligible, and 3) the often lower number of features selected by RENT is a clear benefit. RENT does not fail for any of the presented datasets, whereas other methods show weaknesses on at least one dataset, with regard to either a very large number of selected features or poor predictive quality. In particular, RENT consistently performs well, whether the data are long-thin—many objects compared to the number of features—or short-wide—relatively few objects compared to the number of features. The initial intention of RENT was to target short-wide datasets, which are particularly challenging when it comes to feature selection. In the presented evaluations, datasets c4 (Dexter text classification), c5 (OVA Lung), and r1 (Milk proteins) clearly fall into this category.

In addition to competitive performance, the number of features selected by RENT for the studied datasets is comparably low, which is a strength of RENT in terms of interpretability of results. Furthermore, the object-wise visualization demonstrated in Section IV-A can provide previously unseen insights into the properties of the dataset, which may be particularly relevant for medical applications, but also for many other applications in general.

Robustness with regard to noisy data is another strength of RENT, which can be achieved by the extensive use of drawing subsamples from the training set. Particularly the baseline model M^0 , which is used as a benchmark in the experiments and achieves high performance on multiple datasets, is susceptible to poor initializations and hence, potentially less reliable for the selection of features. Although computationally more intensive than the comparing methods, RENT is less susceptible to poor initializations or convergence issues of optimization routines compared to other approaches.

In total, RENT has five model parameters to adjust by the user: two account for regularization intensity (γ and α) and three cutoffs control the strictness of feature selection (t_1 , t_2 and t_3). Both sets of hyperparameters are related, since a softer regularization allows a larger number of features, requiring higher cutoffs (and vice versa). Based on the presented parameter selection procedure using BIC, feature selectors which deliver a low number of features are favored in both stages.

In the current formulation, RENT is applicable for binary classification and regression problems. As introduced in [51], multiclass feature selection is not trivial and will be part of further research. However, a multiclass classification problem can be split into several binary problems, using schemes such as one-vs-one (OVO), one-vs-all (OVA), or error-correcting output coding (ECOC), as described in [52].

VI. CONCLUSION

In this work, we presented a feature selection technique for binary classification and regression problems. The algorithm

builds on the idea of training multiple elastic net regularized models on unique training data subsets. In particular, we define feature importance criteria based on the empirical distribution of feature-wise model weights. Features are selected if their associated weights are regularly assigned high non-zero values with stable signs across the individual models of the ensemble.

We provided experiments on datasets from different disciplines, demonstrating that RENT is effective with respect to quantitative performance measures and interpretability and robustness. For the presented setups, the stability is very high even with a moderate number of ensemble models and in five out of six binary classification datasets, RENT achieves the highest MCC scores compared to the established feature selectors used in this study. For the regression datasets, RENT performed better or almost equal to the competing approaches. Further, we showed how to utilize information from the ensemble of models in a post-hoc analysis, advancing single-object interpretability.

ACKNOWLEDGMENT

In special, the authors thank Tormod Næs (Nofima) and Yngve Mardal Moe (University of Oslo) for their constructive discussions and valuable input to this work. Further, they thank the reviewers for their feedback which helped them to improve the article.

REFERENCES

- [1] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining," in *Proc. Feature Selection Data Mining*, 2010, pp. 4–13.
- [2] A. S. Ashour, M. K. A. Nour, K. Polat, Y. Guo, W. Alsaggaf, and A. El-Attar, "A novel framework of two successive feature selection levels using weight-based procedure for voice-loss detection in Parkinson's disease," *IEEE Access*, vol. 8, pp. 76193–76203, 2020.
- [3] G. Li, T. Yuan, C. Li, J. Zhuo, Z. Jiang, J. Wu, D. Ji, and H. Zhang, "Effective breast cancer recognition based on fine-grained feature selection," *IEEE Access*, vol. 8, pp. 227538–227555, 2020.
- [4] Y. Saeyns, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2008, pp. 313–325.
- [5] X. Bai, X. Gao, and B. Xue, "Particle swarm optimization based two-stage feature selection in text mining," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2018, pp. 1–8.
- [6] A. Moghimi, C. Yang, and P. M. Marchetto, "Ensemble feature selection for plant phenotyping: A journey from hyperspectral to multispectral imaging," *IEEE Access*, vol. 6, pp. 56870–56884, 2018.
- [7] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
- [8] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA: MIT Press, 2005, pp. 507–514.
- [9] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. Nat. Conf. Artif. Intell.*, vol. 2, 1992, pp. 129–134.
- [10] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
- [11] R. Chen, N. Sun, X. Chen, M. Yang, and Q. Wu, "Supervised feature selection with a stratified feature weighting method," *IEEE Access*, vol. 6, p. 15087–15098, 2018.
- [12] M. Cherrington, F. Tabtah, J. Lu, and Q. Xu, "Feature selection: Filter methods performance challenges," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCCIS)*, 2019, pp. 1–4.
- [13] A. U. Haq, D. Zhang, H. Peng, and S. U. Rahman, "Combining multiple feature-ranking techniques and clustering of variables for feature selection," *IEEE Access*, vol. 7, pp. 151482–151492, 2019.
- [14] J. Loughrey and P. Cunningham, "Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets," in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.* London, U.K.: Springer, 2004, pp. 33–43.
- [15] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
- [16] J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*. Jason Brownlee, 2011. [Online]. Available: <https://github.com/clever-algorithms/CleverAlgorithms>
- [17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
- [18] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning With Sparsity: The Lasso and Generalizations*. Boca Raton, FL, USA: CRC Press, 2015.
- [19] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, "Feature selection with ensembles, artificial variables, and redundancy elimination," *J. Mach. Learn. Res.*, vol. 10, pp. 1341–1366, Jul. 2009.
- [20] L. Cui, L. Bai, Z. Zhang, Y. Wang, and E. R. Hancock, "Identifying the most informative features using a structurally interacting elastic net," *Neurocomputing*, vol. 336, pp. 13–26, Apr. 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231218312736>
- [21] L. Cui, L. Bai, Y. Wang, X. Jin, and E. R. Hancock, "Internet financing credit risk evaluation using multiple structural interacting elastic net feature selection," *Pattern Recognit.*, vol. 114, Jun. 2021, Art. no. 107835.
- [22] L. Cui, L. Bai, Y. Wang, P. S. Yu, and E. R. Hancock, "Fused lasso for feature selection using structural information," *Pattern Recognit.*, vol. 119, Nov. 2021, Art. no. 108058.
- [23] T. Pang, F. Nie, J. Han, and X. Li, "Efficient feature selection via $\ell_{2,0}$ -norm constrained sparse regression," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 880–893, Jun. 2019.
- [24] S. Nogueira, K. Sechidis, and G. Brown, "On the stability of feature selection algorithms," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6345–6398, 2017.
- [25] A. Bommert, J. Rahnenführer, and M. Lang, "A multicriteria approach to find predictive and sparse models with stable feature selection for high-dimensional data," *Comput. Math. Methods Med.*, vol. 2017, Aug. 2017, Art. no. 7907163.
- [26] H. M. Bovelstad, S. Nygard, H. L. Storvold, M. Aldrin, O. Borgan, A. Friggis, and O. C. Lingjaerde, "Predicting survival from microarray data a comparative study," *Bioinformatics*, vol. 23, no. 16, pp. 2080–2087, Aug. 2007.
- [27] H. Xu, C. Caramanis, and S. Mannor, "Sparse algorithms are not stable: A no-free-lunch theorem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 187–193, Jan. 2012.
- [28] U. Meinshausen and P. Bühlmann, "Stability selection," *J. Roy. Stat. Soc. B. Stat. Methodol.*, vol. 72, no. 4, pp. 417–473, 2010.
- [29] A. Jenul, S. Schrunner, B. Huynh, and O. Tomic, "RENT: A Python package for repeated elastic net feature selection," *J. Open Source Softw.*, vol. 6, no. 63, p. 3323, Jul. 2021, doi: [10.21105/joss.03323](https://doi.org/10.21105/joss.03323).
- [30] K. P. Burnham and D. R. Anderson, "Multimodel inference: Understanding AIC and BIC in model selection," *Sociol. Methods Res.*, vol. 33, no. 2, pp. 261–304, Nov. 2004.
- [31] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–451, 2004. [Online]. Available: <http://www.jstor.org/stable/3448465>
- [32] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010. [Online]. Available: <https://www.jstatsoft.org/index.php/jss/article/view/v033i01>
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [34] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [35] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proc. Nat. Acad. Sci. USA*, vol. 87, no. 23, pp. 9193–9196, 1990.

- [36] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA, USA: MIT Press, 2005, pp. 545–552. [Online]. Available: <http://papers.nips.cc/paper/2728-result-analysis-of-the-nips-2003-feature-selection-challenge.pdf>
- [37] G. Stiglic and P. Kokol, "Stability of ranked gene lists in large microarray analysis studies," *J. Biomed. Biotechnol.*, vol. 2010, pp. 1–9, Jan. 2010.
- [38] K. H. Liland, B.-H. Mevik, E.-O. Rukke, T. Almøy, and T. Isaksson, "Quantitative whole spectrum analysis with MALDI-TOF MS, Part II: Determining the concentration of milk in mixtures," *Chemometric Intell. Lab. Syst.*, vol. 99, no. 1, pp. 39–48, Nov. 2009.
- [39] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta-Protein Struct.*, vol. 405, no. 2, pp. 442–451, Oct. 1975.
- [40] N. R. Draper and H. Smith, *Applied Regression Analysis*, vol. 326. Hoboken, NJ, USA: Wiley, 1998.
- [41] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Appl. Intell.*, vol. 7, no. 1, pp. 39–55, Jan. 1997.
- [42] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [43] B. Hofner and T. Hothorn. (2021). *STABS: Stability Selection with Error Control*. [Online]. Available: <https://CRAN.R-project.org/package=stabs>
- [44] K. You. (2020). *Rdimtools: Dimension Reduction Estimation Methods*. [Online]. Available: <https://CRAN.R-project.org/package=Rdimtools>
- [45] J. De, S. Papillon-Cavanagh, C. Olsen, N. El-Hachem, G. Bontempi, and B. Haibe-Kains, "MRMR: An R package for parallelized mRMR ensemble feature selection," *Bioinformatics*, vol. 29, no. 18, pp. 2365–2368, Sep. 2013.
- [46] R. Olson. (Mar. 2016). *ReliefF: First Release*. [Online]. Available: <https://doi.org/10.5281/zenodo.47803>
- [47] M. L. Calle and V. Urrea, "Letter to the editor: Stability of random forest importance measures," *Briefings Bioinf.*, vol. 12, no. 1, pp. 86–89, Jan. 2011.
- [48] K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis*, vol. 15. London, U.K.: Academic, 1979, p. 518.
- [49] H. Martens and M. Martens, "Modified jack-knife estimation of parameter uncertainty in bilinear modelling by partial least squares regression (PLSR)," *Food Quality Preference*, vol. 11, nos. 1–2, pp. 5–16, Jan. 2000.
- [50] O. Tomic, T. Graff, K. Liland, and T. Nås, "Hoggorm: A Python library for explorative multivariate statistics," *J. Open Source Softw.*, vol. 4, no. 39, p. 980, Jul. 2019. [Online]. Available: <http://joss.theoj.org/papers/10.21105/joss.00980>
- [51] W. Gao, J. Hu, Y. Li, and P. Zhang, "Feature redundancy based on interaction information for multi-label feature selection," *IEEE Access*, vol. 8, pp. 146050–146064, 2020.
- [52] S. Schrunner, O. Bluder, A. Zernig, A. Kaestner, and R. Kern, "A comparison of supervised approaches for process pattern recognition in analog semiconductor wafer test data," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 820–823.



ANNA JENUL (Graduate Student Member, IEEE) received the M.Sc. degree in mathematics from the University of Klagenfurt, Austria. She is currently pursuing the Ph.D. degree in the field of data science with the Norwegian University of Life Sciences, working on the application and development of statistical methods in the healthcare sector. Her research interests include machine learning, data engineering, and applied statistics.



STEFAN SCHRUNNER (Member, IEEE) received the M.Sc. degree in technical mathematics from the University of Klagenfurt, Austria, in 2016, and the Ph.D. degree in computer science from the Graz University of Technology, Austria, in 2019. He is currently a Postdoctoral Fellow in data science with the Norwegian University of Life Sciences. His research interests include machine learning and applied statistics, including image processing, pattern recognition, spatial statistics, and Bayesian models.



KRISTIAN HOVDE LILAND received the Ph.D. degree in applied statistics in 2010. He is currently an Associate Professor in data science with the Faculty of Science and Technology, Norwegian University of Life Sciences. His research interests include multivariate statistics and machine learning with a large spectrum of projects ranging from theoretic to applied and often with an emphasis on scientific programming.



ULF GEIR INDAHL is currently a Professor in statistics with the Faculty of Science and Technology, Norwegian University of Life Sciences. He is interested in the development of new methodology as well as a broad range of applications ranging from high-dimensional multivariate data analysis to machine learning and artificial neural nets.



CECILIA MARIE FUTS/ETHER received the M.Sc. degree in applied mathematics from the University of Oslo and the Ph.D. degree in physics from the Norwegian University of Science and Technology. She is currently a Professor of physics with the Faculty of Science and Technology, Norwegian University of Life Sciences. Her research interests include medical physics and applications of machine learning and deep learning in medical diagnostics.



OLIVER TOMIC received the Dr. scient. degree in gas-sensor array technology and applied multivariate statistics from the Norwegian University of Life Sciences, in 2004. He is currently an Associate Professor of data science with the Faculty of Science and Technology, Norwegian University of Life Sciences. His research interests include machine learning and applied multivariate statistics with a specific focus on multiblock methods. He has a special interest in the analysis of healthcare data and food-related data but also participates in research in several different domains that involve machine learning and multivariate statistics.

...

RENT: A Python Package for Repeated Elastic Net Feature Selection

Anna Jenul¹, Stefan Schrunner¹, Bao Ngoc Huynh², and Oliver Tomic¹

¹ Department of Data Science, Norwegian University of Life Sciences ² Department of Physics, Norwegian University of Life Sciences

DOI: [10.21105/joss.03323](https://doi.org/10.21105/joss.03323)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Mikkel Meyer Andersen](#) ↗

Reviewers:

- [@maximtrp](#)
- [@arunmano121](#)

Submitted: 29 April 2021

Published: 28 July 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Due to modern data acquisition techniques, the number of generated features in measurement data keeps increasing. This increase can make the analysis with standard machine learning methods difficult because of underdetermined systems where the dimensionality of the feature space (number of features) exceeds the dimensionality of the object space (number of observations). A concrete example of such a situation is data acquisition in the healthcare domain, where the number of patients (observations) suffering from a specific condition may be relatively low, but a lot of measurements (number of features) are generated for each patient to acquire a good understanding of the patient's health. A very common challenge is that not all features in a high dimensional space are equally important for predictive tasks — many might even be redundant. Feature selection deals with finding the most relevant features of a dataset. With help of appropriate methodology, feature selection can reduce (a) the complexity of and (b) noise in the dataset. More importantly, data interpretation of the model becomes easier with fewer features, which is of great importance within domains such as healthcare. Even though feature selection is a well-established research topic, relatively few approaches are focusing on the stability of the selection. The important question at hand is: can we trust that the selected features are really valid or is their selection very dependent on which observations are included in the data? Providing information on the stability of feature selection is vital, especially in wide data sets where the number of features can be many times higher than the number of observations. Here, the inclusion or exclusion of a few observations can have a high impact on which features may be selected.

Statement of Need

To get an understanding of which features are important and how stable the selection of each feature in the dataset is, a user-friendly software package is needed for this purpose. The RENT package, implementing the feature selection method of the same name ([Jenul et al., 2021](#)), provides this information through an easy-to-use interface. The package includes functionalities for binary classification and regression problems. RENT is based on an ensemble of elastic net regularized models, which are trained on randomly, iid subsets of the rows of the full training data. Along with selecting informative features, the method provides information on model performance, selection stability, as well as interpretability. Compared to established feature selection packages available in R and Python, such as `Rdimtools` ([You, 2020](#)) implementing Laplacian and Fisher scores or the scikit-learn feature selection module ([Pedregosa et al., 2011](#)) implementing recursive feature elimination and sequential feature selection, RENT creates a deeper understanding of the data by utilizing information acquired through the ensemble. This aspect is realized through tools for post hoc data analysis, visualization, and feature selection validation provided with the package, along with an efficient and user-friendly implementation of the main methodology.

Concept and Structure of RENT

At its core, RENT trains K independent elastic net regularized models on distinct subsets of the training dataset. Each subset is generated using the scikit-learn function `train_test_split()` which delivers an iid sample from the full training dataset. The sampling processes of different subsets are mutually independent, with the condition that a single data point can appear at most once in each subset. A data point, however, can appear in multiple subsets. The framework is demonstrated in Figure 1.

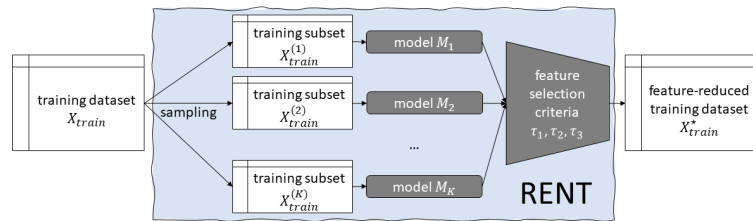


Figure 1: Summary of RENT method (Jenul et al., 2021).

Based on three statistical cutoff criteria τ_1 , τ_2 and τ_3 , relevant features are selected. While τ_1 counts how often each feature was selected over K models, τ_2 quantifies the stability of the feature weights — a feature where the K weight signs alternate between positive and negative is less stable than a feature where all weights are of a constant sign. The third criterion τ_3 deploys a Student's t -test to judge whether feature weights are significantly different from zero. The presented implementation builds on an abstract class `RENT_Base` with a general skeleton for feature selection and post hoc analysis. Two inherited classes, `RENT_Classification` and `RENT_Regression`, offer target-specific methods. The constructor of `RENT_Base` initializes the different user-specific parameters such as the dataset, elastic net regularization parameters, or the number of models K . After training, feature selection is conducted by use of the cutoff criteria. Deeper insights are provided by a matrix containing the cutoff criteria values of each feature, as well as a matrix comprising raw model weights of each feature throughout the K elementary model. For initial analysis of the results, the package delivers multiple plotting functions, such as a barplot of τ_1 . Additionally, two validation studies are implemented: first, a model based on random feature selection is trained, while second, a model based on randomly permuted labels of the test dataset is obtained. Results of both validation models are compared to a model built with RENT features using Student's t -tests as well as empirical densities.

In addition to feature selection, RENT offers a detailed summary of prediction accuracies for the training objects. For each training object, this information can be visualized as histograms of class probabilities for classification problems or histograms of mean absolute errors for regression problems, respectively. For extended analysis, principal component analysis reveals properties of training objects and their relation to features selected by RENT. For computation and visualization of principal components, RENT uses functionality from the `hoggorm` and `hoggormplot` packages (Tomic et al., 2019).

Ongoing Research and Dissemination

The manuscript RENT - Repeated Elastic Net Technique for Feature Selection is currently under review. Further, the method and the package are used in different master thesis projects at the Norwegian University of Life Sciences, mainly in the field of healthcare data analysis.

Acknowledgements

We thank Runar Helin for proofreading the documentation.

References

- Jenul, A., Schrunner, S., Liland, K. H., Indahl, U. G., Futsaether, C. M., & Tomic, O. (2021). *RENT – repeated elastic net technique for feature selection*. <http://arxiv.org/abs/2009.12780>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Tomic, O., Graff, T., Liland, K. H., & Næs, T. (2019). Hoggorm: A python library for explorative multivariate statistics. *The Journal of Open Source Software*, 4(39). <https://doi.org/10.21105/joss.00980>
- You, K. (2020). *Rdimtools: Dimension reduction and estimation methods*. <https://CRAN.R-project.org/package=Rdimtools>

Appendix B

Papers II & IIa

| | |
|---------------------|-------------------------------------------------------------------------------------------------------|
| title: | A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS) |
| authors: | Anna Jenul , Stefan Schrunner, Jürgen Pilz, Oliver Tomic |
| date: | 08/2022 |
| publication: | Machine Learning |
| doi: | https://doi.org/10.1007/s10994-022-06221-9 |

| | |
|---------------------|---------------------------------------------------------------------------------------|
| title: | UBayFS: An R Package for User Guided Feature Selection |
| authors: | Anna Jenul , Stefan Schrunner |
| date: | 01/2023 |
| publication: | Journal of Open Source Software |
| doi: | https://doi.org/10.21105/joss.04848 |



A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS)

Anna Jenul¹ · Stefan Schrunner¹  · Jürgen Pilz² · Oliver Tomic¹

Received: 15 December 2021 / Revised: 24 May 2022 / Accepted: 2 July 2022 /
Published online: 22 August 2022
© The Author(s) 2022

Abstract

Feature selection reduces the complexity of high-dimensional datasets and helps to gain insights into systematic variation in the data. These aspects are essential in domains that rely on model interpretability, such as life sciences. We propose a (U)ser-Guided (Bay)esian Framework for (F)eature (S)election, UBayFS, an ensemble feature selection technique embedded in a Bayesian statistical framework. Our generic approach considers two sources of information: data and domain knowledge. From data, we build an ensemble of feature selectors, described by a multinomial likelihood model. Using domain knowledge, the user guides UBayFS by weighting features and penalizing feature blocks or combinations, implemented via a Dirichlet-type prior distribution. Hence, the framework combines three main aspects: ensemble feature selection, expert knowledge, and side constraints. Our experiments demonstrate that UBayFS (a) allows for a balanced trade-off between user knowledge and data observations and (b) achieves accurate and robust results.

Keywords Ensemble feature selection · Bayesian model · Dirichlet-multinomial · User constraints

Editors: Krzysztof Dembczynski and Emilie Devijver.

Anna Jenul and Stefan Schrunner have contributed equally to this work.

✉ Stefan Schrunner
stefan.schrunner@nmbu.no

Anna Jenul
anna.jenul@nmbu.no

Jürgen Pilz
juergen.pilz@aau.at

Oliver Tomic
oliver.tomic@nmbu.no

¹ Department of Data Science, Norwegian University of Life Sciences, Ås, Norway

² Department of Statistics, University of Klagenfurt, Klagenfurt, Austria

1 Introduction

Feature selection pursues two major goals: to improve the performance of predictive algorithms like classification, regression, or clustering models as well as to improve data understanding and interpretability. Both aspects are of significant interest in the field of life science, such as healthcare, where major decisions may be based on data analysis. Here, two sources of information are often available: large-scale collections of data from multiple sources and profound knowledge from domain experts. Previous works tend to handle these sources as opposites, see Cheng et al. (2006), or neglect expert knowledge completely, see Pozzoli (2020). However, a combination of both can be valuable to compensate for underdetermined problem setups from high-dimensional datasets, which are prevalent in healthcare data analysis. Moreover, meta-information on the feature set may leverage interpretability. Works such as Liu and Zhang (2015) consider constraints between samples but neglect constraints between features. The extension of L1 regularization to the so-called *Group Lasso* (Yuan & Lin, 2006) and its variants (Ida et al., 2019) account for block structure but cannot handle more complex constraint types. Elementary approaches to integrating user knowledge and feature selection include Guan Guan et al. (2009), who suggest manually adding user-defined features to the feature selection output of algorithms. A more advanced model by Brahim and Limam (2014) embeds prior knowledge into three particular feature selection algorithms. Though, their work neither allows a direct generalization to other feature selectors nor the integration of more general types of prior knowledge, such as side constraints. Hence, there is a lack of general and sophisticated frameworks for feature selection that combine data-driven methods with user knowledge and deliver transparent results.

Apart from measuring predictive model performance, properties like stability and reproducibility of the feature selector are essential for transparency. A model-independent approach for improving feature selection stability is to deploy ensembles of elementary feature selectors. Recent research by Bose (2021), and Jenul (2021) pursued this idea by utilizing sub-sampling strategies to generate model ensembles as such provide feature stability measures aside from good predictive performance. Seijo-Pardo et al. (2017) conclude that meta-models composed of elementary feature selectors improve the performance and robustness of the selected feature set in many cases. However, to the best of our knowledge, probabilistic approaches that exploit both — a sound statistical framework and individual model benefits of using an ensemble elementary feature selectors — are not yet available.

A prominent framework with the capability to combine data and expert knowledge is Bayesian statistics, which has been applied for feature selection in linear models, see O'Hara and Sillanpää (2009). Intentions behind the usage of Bayesian methodology vary significantly between authors and do not necessarily involve expert knowledge. Examples include Dalton (2013), who investigates sparsity priors, and Goldstein et al. (2020), who suggest a Bayesian framework to quantify the level of uncertainty in the underlying feature selection model. Other Bayesian approaches for feature selection include Saon and Padmanabhan (2001), and Lyle et al. (2020), but these works do not investigate the usage of expert knowledge as prior. Although the availability of expert knowledge plays a role in life sciences, none of these approaches strongly emphasizes domain knowledge about features, nor do they involve specific prior constraints defined by the user.

In this work, we propose a novel Bayesian approach to feature selection that incorporates expert knowledge and maintains considerable model generality. We aim to fill the gap between data-driven feature selection on one side and purely expert-focused feature

selection on the other side. Our presented probabilistic approach, UBayFS, combines a generic ensemble feature selection framework with the exploitation of domain knowledge. Hence, it supports interpretability and improves the stability of the results. For this purpose, feature importance votes from independent elementary feature selectors are merged with constraints and feature weights specified by the expert. Constraints may be of a general type, such as selecting a maximum number of features or blocks of features. Both inputs, likelihood and prior, are aggregated in a sound statistical framework, producing a posterior probability distribution over all possible feature sets. We use a Genetic Algorithm for discrete optimization to efficiently optimize the posterior feature set in high-dimensional datasets. In an extensive experimental evaluation, we analyze UBayFS in a variety of model setups involving prior knowledge and constraints. Results on open-source datasets are benchmarked against state-of-the-art feature selectors in terms of predictive performance and stability, underlining the potential of UBayFS.

Notations We will denote vectors by bold, uncapitalized, and matrices by bold, capitalized letters. Non-bold, uncapitalized letters indicate scalars or functions, and non-bold, capitalized letters indicate sets or constants. $\|\cdot\|_1$ denotes the $L1$ -norm. $[N]$ is an abbreviation of the set of indices $1, \dots, N$. The N -dimensional vector of ones will be written as $\mathbf{1}_N$. Furthermore, we refer to sets of features by their feature indices, such as $S \subseteq [N]$, or by a binary membership vector $\delta^S \in \{0, 1\}^N$ with components $(\delta^S)_n = \begin{cases} 1 & \text{if } n \in S, \\ 0 & \text{otherwise.} \end{cases}$

2 User-guided ensemble feature selector

Given a finite set of N features, the goal of UBayFS is to find an optimal subset of feature indices $S^* \subset [N]$, or, equivalently, $\delta^* = \delta^{S^*} \in \{0, 1\}^N$. We assume that information is available from

1. Training data to collect evidence by conventional data-driven feature selectors—we denote this as information from data \mathbf{y} ,
2. The user's domain knowledge encoded as subjective beliefs $\alpha \in \mathbb{R}^N$ about the importance of features, where $\alpha_n > 0$ for all $n \in [N]$, and
3. Side constraints, given as inequality system $A\delta \leq \mathbf{b}$, to ensure that the obtained feature set conforms with practical requirements and restrictions.

UBayFS assumes a feature importance vector $\theta \in [0, 1]^N$, $\|\theta\|_1 = 1$, which is probabilistic and not directly observable, such that evidence about θ is collected from data \mathbf{y} and prior weights α . Our model aims to maximize the accumulated importances $\delta^T \theta$ of the selected features subject to side constraints $A\delta \leq \mathbf{b}$. More specifically, we maximize the utility function

$$U(\delta, \theta) = \delta^T \theta - \lambda \kappa(\delta), \quad \lambda > 0, \quad (1)$$

where $\kappa(\delta)$ is a non-negative scalar function which penalizes the degree of violation of the constraints. The precise form of $\kappa(\cdot)$ will be given later. Clearly, we require that $\kappa(\delta) = 0$, if $A\delta \leq \mathbf{b}$ is satisfied. In Eq. 1, $\lambda > 0$ plays the role of a Lagrange parameter, $\lambda \kappa(\delta)$ increases the amount of penalization imposed on a feature set violating the constraints. In terms of statistical decision theory, a Bayes decision should maximize the posterior expected utility

$$\mathbb{E}_{\theta|y}[U(\delta, \theta(y))] = \delta^T \mathbb{E}_{\theta|y}[\theta(y)] - \lambda \kappa(\delta) \longrightarrow \max_{\delta \in \{0,1\}^N}. \quad (2)$$

We denote the optimal feature set according to Eq. 2 by δ^* . The importance parameter θ is inferred from data from elementary feature selectors trained on subsets of the dataset, summarized as y , as well as prior feature importance scores α . Thus, the posterior probability distribution of θ given observations y , $p(\theta|y)$, is decomposed using Bayes' theorem into

$$p(\theta|y) \propto p(y|\theta) \cdot p(\theta), \quad (3)$$

where $p(y|\theta)$ describes the model likelihood (evidence from elementary feature selector model) and $p(\theta)$ describes the density of a prior distribution (user domain knowledge).

The remainder of this Section focuses on determining the missing model components to define the problem stated in Eq. (2), comprising (a) the feature importances θ , discussed in Sect. 2.1 and 2.2, and (b) the function κ , discussed in Sect. 2.3. Finally, Sect. 2.4 suggests the discrete optimization procedure to solve Eq. (2).

2.1 Ensemble feature selection as likelihood

To collect information about feature importances from the given dataset, we train an ensemble of M elementary feature selectors of the same model type on distinct training subsets. The selection of a feature index set $\delta^{(m)}$ comprising a constant number of $l = \|\delta^{(m)}\|_1$ features in each elementary model m out of a total of M models can be interpreted as a result of drawing l balls from an urn, where each ball has a distinct color representing one feature $n \in [N]$. Over all elementary models, y collects the counts of each feature being selected, resulting in a count vector in

$$y = \sum_{m=1}^M \delta^{(m)} \in \{0, \dots, M\}^N. \quad (4)$$

Each elementary feature selector delivers a proposal for an optimal feature set. Thus, we let the frequency of drawing a feature throughout $\delta^{(1)}, \dots, \delta^{(M)}$ represent its *importance* by defining the latent importance parameter vector $\theta \in [0, 1]^N$, $\|\theta\|_1 = 1$, as the success probabilities of sampling each feature in an individual urn draw. In a statistical sense, we interpret the result from each elementary feature selector as realization from a multinomial distribution with parameters θ and l .¹ This multinomial setup delivers the likelihood $p(y|\theta)$ as joint probability density

$$p(y|\theta) = \prod_{m=1}^M f_{\text{mult}}(\delta^{(m)}; \theta, l), \quad (5)$$

where $f_{\text{mult}}(\delta^{(m)}; \theta, l)$ denotes the density of a multinomial distribution with success probabilities θ and a number of l urn draws. Relevant notations are summarized in Table 1.

¹ The exact way to describe this procedure is a multivariate hypergeometric distribution, since each feature occurs at most once in a set, but an approximation using the multinomial distribution facilitates computation.

Table 1 Notations for likelihood parameters

| Input and elementary models | |
|--------------------------------------|---------------------|
| $n \in [N]$ | Feature indices |
| $m \in [M]$ | Elementary models |
| $\delta \in \{0, 1\}^N$ | Feature index set |
| $\theta \in \Theta \subset [0, 1]^N$ | Feature importances |
| $y \in \{0, \dots, M\}^N$ | Feature counts |

2.2 Expert knowledge as prior weights

To constitute the prior distribution, UBayFS uses expert knowledge as a-priori weights of features. Since the domain of the distribution of feature importances θ is defined to be a simplex $\theta \in \Theta \subset [0, 1]^N, \|\theta\|_1 = 1$, the Dirichlet distribution is a natural choice as prior distribution, which is widely used in data science problems, such as Nakajima et al. (2014). Thus, we initially assume that a-priori

$$p(\theta) = f_{\text{Dir}}(\theta; \alpha), \tag{6}$$

where $f_{\text{Dir}}(\theta; \alpha)$ denotes the density of the Dirichlet distribution with positive $\alpha = (\alpha_1, \dots, \alpha_N)$. Since the Dirichlet distribution is a conjugate prior of the multinomial distribution, the posterior distribution results in a Dirichlet type, again, see DeGroot (2005). Thus, it holds for the posterior density that

$$p(\theta|y) \propto f_{\text{Dir}}(\theta; \alpha^\circ), \tag{7}$$

where the parameter update is obtained in closed form by

$$\alpha^\circ = \alpha + y. \tag{8}$$

In case of integer-valued prior weights α , they may be interpreted as pseudo-counts in the context of modelling success probabilities in an urn model—comparable to the information gained if the corresponding counts were observed in a multinomial data sample. In UBayFS, we obtain α as feature weights provided by the user. If no user knowledge is available, the least informative choice is to specify uniform counts with a small positive value, such as $\alpha_{\text{unif}} = 0.01 \cdot \mathbf{1}_N$.

2.2.1 Generalized Dirichlet model

Even though the presented Dirichlet-multinomial model is a popular choice due to its favorable statistical properties, it implicitly assumes that classes (in our case, features) are mutually independent. However, high-dimensional datasets frequently involve complex correlation structures between the features. To account for this aspect, we generalize the setup by replacing the Dirichlet prior distribution with some generalized Dirichlet distribution. The highest level of generalization is achieved by Hankin (2010), who introduced the hyperdirichlet distribution, which may take arbitrary covariance structures into account. The hyperdirichlet distribution maintains the conjugate prior property with respect to the multinomial likelihood, and thus, inference is tractable; however, the analytical expression of the expected value involves the intractable normalization constant and, as a result,

requires numerical means such as Monte-Carlo Markov Chain (MCMC) methods, which may face computational challenges due to the high dimensionality of the problem.

A compromise between the complexity of the problem and the flexibility of the covariance structure is given by an earlier version of the generalized Dirichlet distribution by Wong (1998), which is a special case of the hyperdirichlet setup, but more general than the standard Dirichlet distribution. In addition to the properties of the hyperdirichlet distribution, the expected value of the generalized Dirichlet distribution can be directly evaluated from the distribution parameters. Section 3 provides an experimental evaluation of the proposed variants to account for covariance structures in the UBayFS model.²

2.3 Side constraints as regularization

Practical setups may require that a selected feature set fulfills certain consistency requirements. These may involve a maximum number of selected features, a low mutual correlation between features, or a block-wise selection of features. UBayFS enables the feature selection model to account for such requirements via a function κ , which incorporates a system of K inequalities restricting the feature set δ , $\mathbf{A}\delta - \mathbf{b} \leq 0$, where $\mathbf{A} \in \mathbb{R}^{K \times N}$ and $\mathbf{b} \in \mathbb{R}^K$. Each single constraint $k \in [K]$ can be evaluated via an inadmissibility function $\kappa_k(\cdot)$, such that

$$\kappa_k(\delta) = \begin{cases} 0 & \text{if } (\mathbf{a}^{(k)})^T \delta - b^{(k)} \leq 0 \\ 1 & \text{otherwise,} \end{cases} \quad (9)$$

where $\mathbf{a}^{(k)}$ is the k -th row vector of \mathbf{A} and $b^{(k)}$ the k -th element of \mathbf{b} . UBayFS generalizes the setup by relaxing the constraints: in case that a feature set δ violates a constraint, it shall be assigned a higher penalty rather than being excluded completely. This effect is achieved by replacing $\kappa_k(\cdot)$ with a relaxed inadmissibility function $\kappa_{k,\rho}(\cdot)$ based on a logistic function with relaxation parameter $\rho \in \mathbb{R}^+ \cup \{\infty\}$:

$$\kappa_{k,\rho}(\delta) = \begin{cases} 0 & \text{if } (\mathbf{a}^{(k)})^T \delta \leq b^{(k)} \\ 1 & \text{if } (\mathbf{a}^{(k)})^T \delta > b^{(k)} \wedge \rho = \infty \\ \frac{1 - \xi_{k,\rho}}{1 + \xi_{k,\rho}} & \text{otherwise,} \end{cases} \quad (10)$$

with $\xi_{k,\rho} = \exp(-\rho((\mathbf{a}^{(k)})^T \delta - b^{(k)}))$. Fig. 1 illustrates that a large parameter $\rho \rightarrow \infty$ lets the inadmissibility converge pointwise towards the associated hard constraint. A low ρ changes the shape of the penalization to an almost constant function in a local neighborhood around the decision boundary, such that only a minor difference is made between feature sets that fulfill and those that violate a constraint.³

Finally, the joint inadmissibility function $\kappa(\cdot)$ aggregates information from all constraints

$$\kappa(\delta) = 1 - \prod_{k=1}^K (1 - \kappa_{k,\rho}(\delta)), \quad (11)$$

² Details on the generalized prior distributions are provided in Appendix A.

³ for a proof see Appendix A

Fig. 1 The effect of ρ on $\kappa_{k,\rho}$ for soft constraints

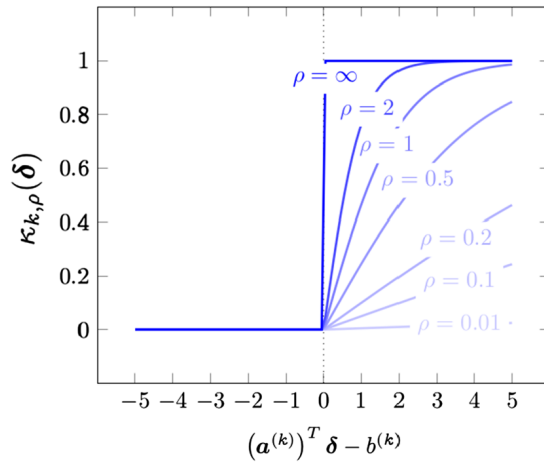


Table 2 Notations used for prior parameters

| Prior parameters | |
|-----------------------------------------------------|-------------------------|
| $\alpha, \alpha^\circ \in \mathbb{R}^N$ | Prior/posterior weights |
| $k \in [K]$ | Constraint index |
| $A \in \mathbb{R}^{K \times N}, b \in \mathbb{R}^K$ | Inequality system |
| $\rho \in \mathbb{R}^K$ | Relaxation parameters |
| $\kappa(\cdot) : \{0, 1\}^N \rightarrow [0, 1]$ | Joint inadmissibility |

which originates from the idea that $\kappa = 1$ (maximum penalization) if at least one $\kappa_{k,\rho} = 1$, while $\kappa = 0$ (no penalization) if all $\kappa_{k,\rho} = 0$.

Note that different relaxation parameters may be used to prioritize the constraints among each other, hence κ involves a parameter vector $\rho = (\rho_1, \dots, \rho_K)$. Notations related to prior parameters and constraints are summarized in Table 2.

2.3.1 Feature decorrelation constraints

Commonly, feature sets with low mutual correlations are preferred since they tend to contain less redundant information. A special case of prior constraints can be defined to enforce that such feature sets are selected. We will refer to such constraints as decorrelation constraints. Decorrelation constraints are pairwise cannot-link constraints between highly correlated features, i.e., features i and j with a correlation coefficient $\tau_{i,j}$ exceeding a predefined absolute threshold $|\tau_{i,j}| > \tau$. For each such pair $i, j \in [N], i \neq j$, a constraint is added to the constraint system as follows: the vector \mathbf{a} with elements

$$a_n = \begin{cases} 1 & \text{if } n \in \{i, j\} \\ 0 & \text{else,} \end{cases} \tag{12}$$

and an element $b = 1$ are appended to \mathbf{A} and \mathbf{b} , respectively. We set the shape parameter ρ to the odds ratio of the absolute correlation coefficient $\tau_{i,j}$, given as

$$\rho = \frac{|\tau_{ij}|}{1 - |\tau_{ij}|}. \quad (13)$$

Hence, features with higher absolute correlations are assigned higher penalties and vice versa. As a result, the selected feature set contains features with lower mutual correlations.⁴

2.3.2 Feature block priors

User knowledge may as well be available for *feature blocks* rather than for single features. Feature blocks are contextual groups of features, such as those extracted from the same source in a multi-source dataset. It can be desirable to select features from a few distinct blocks so that the model does not depend on all sources at once. While prior weights can be trivially assigned on block level, we transfer the concept of side constraints to feature blocks.

Feature blocks are specified via a block matrix $\mathbf{B} \in \{0, 1\}^{W \times N}$, where 1 indicates that the feature $n \in [N]$ is part of block $w \in [W]$ and 0, else. Even though a full partition of the feature set is common, feature blocks are neither required to be mutually exclusive, nor exhaustive. Along with the block matrix \mathbf{B} , an inequality system between blocks consists of a matrix $\mathbf{A}^{\text{block}} \in \mathbb{R}^{K \times W}$ and a vector $\mathbf{b}^{\text{block}} \in \mathbb{R}^K$. To evaluate whether a block is selected by a feature set δ , we define the block selection vector $\delta^{\text{block}} \in \{0, 1\}^W$, given by

$$\delta^{\text{block}} = (\mathbf{B}\delta \geq \mathbf{1}_W), \quad (14)$$

where \geq refers to an element-wise comparison of vectors, delivering 1 for a component, if the condition is fulfilled, and 0, otherwise. In other words, a feature block is selected, if at least one feature of the corresponding block is selected. Although block constraints introduce non-linearity into the system of side constraints, they can be used in the same way as linear constraints between features and integrated into the joint inadmissibility function κ .

2.4 Optimization

Exploiting the conjugate prior property, the posterior density of θ can be expressed as a Dirichlet, generalized Dirichlet or hyperdirichlet distribution, respectively. The expected value $\mathbb{E}_\theta[\theta]$ can be computed either in a closed-form expression (Dirichlet or generalized Dirichlet) Wong (1998), or simulated via a sampling procedure (hyperdirichlet) Hankin (2010). It remains to solve the discrete optimization problem in Eq. (2) as a final step.

⁴ We suggest to use Spearman's rho as correlation coefficient, since it is robust (in contrast to Pearson's correlation coefficient) and faster to compute than Kendall's tau.

Algorithm 1 Probabilistic sampling algorithm to initialize GA.**Require:** α° , \mathbf{A} , \mathbf{b} , ρ , sample size Q

```

1:  $G \leftarrow \{\}$ 
2: for  $q \in [Q]$  do
3:    $\delta \leftarrow (0, 0, \dots, 0)$ 
4:   generate a permutation  $\pi$  on  $[N]$  by sampling  $N$  times without
      replacement with probabilities proportional to  $\alpha^\circ$ 
5:   for  $i = \pi(1), \dots, \pi(N)$  do
6:     define  $\delta^\dagger$  as  $\delta_n^\dagger \leftarrow \begin{cases} \delta_n & n \neq i \\ 1 & n = i \end{cases}$  for each  $n \in [N]$ 
7:     sample  $u \sim \text{Unif}_{[0,1]}$ 
8:     if  $u \leq r_{\delta^\dagger, \delta}$  then
9:       update  $\delta \leftarrow \delta^\dagger$ 
10:    end if
11:  end for
12:   $G \leftarrow G \cup \{\delta\}$ 
13: end for
14: return  $G$ 

```

Since an analytical minimization of the resulting knapsack problem is not feasible, we determine a numerical optimum δ^* by using discrete optimization: we deploy the Genetic Algorithm (GA) described by Givens and Hoeting (2012). To guarantee a fast convergence towards an acceptable solution, it is beneficial to provide initial samples, which are good candidates for the final solution. For this purpose we propose a probabilistic sampling algorithm, Alg. 1: In essence, the algorithm creates a random permutation of all features, $\pi : [N] \rightarrow [N]$, by weighted and ordered sampling without replacement. The weights represent the posterior parameter vector α° . Then, the algorithm iteratively accepts or rejects feature $\pi(n)$ with a success probability

$$r_{\delta^\dagger, \delta} = \begin{cases} \frac{1-\kappa(\delta^\dagger)}{1-\kappa(\delta)} & \text{if } \kappa(\delta) < 1 \\ 0 & \text{else,} \end{cases} \quad (15)$$

denoting the admissibility ratios of feature sets with and without feature $\pi(n)$. The generated sample accounts for high feature weights by low ranks, resulting in a higher probability to be accepted in the acceptance/rejection step.

The Genetic Algorithm (GA) for discrete optimization is initialized using Algorithm 1. Starting with an initial set of feature membership vectors $\{\delta^0 \in \{0, 1\}^N\}$, GA creates new vectors $\delta^t \in \{0, 1\}^N$ as pairwise combinations of two preceding vectors δ^{t-1} and $\tilde{\delta}^{t-1}$ in each iteration $t \in [T]$. A combination refers to sampling component δ_n^t from either δ_n^{t-1} or $\tilde{\delta}_n^{t-1}$ in a uniform way and adding minor random mutations to single components. The posterior density serves as fitness when deciding which vectors δ^{t-1} and $\tilde{\delta}^{t-1}$ from iteration $t-1$ should be combined to δ^t —the fitter, the more likely to be part of a combination.

The runtime of GA depends linearly on the population size, and the number of iterations. A good trade-off between runtime and convergence properties is important—a small population size, for example, might lead to faster convergence but might get trapped

towards a local minimum. Further, the runtime is dependent on the complexity to compute the fitness function, which in turn depends on the dimensionality of the problem.

3 Experiments and results

Our numerical experiments evaluate the performance, flexibility, and applicability of UBayFS in two parts: first, a study conducted on synthetic datasets demonstrates the properties of the various model parameters, including

- a. The number of elementary models M (1a),
- b. The prior weights α in a block-wise setup (1b),
- c. The constraint types and their shapes ρ in a block-wise setup (1c), as well as
- d. The type of prior distribution to account for feature dependencies (1d).

The second part of our experiment is conducted on real-world classification datasets from the life science domain. In a comparison with state-of-the-art ensemble feature selectors, we demonstrate that UBayFS delivers similar model performances. Our setups include ordinary and block feature selection without prior knowledge to ensure a fair comparison. Finally, we conduct a case study with expert knowledge available from biological investigations, and demonstrate how informative priors increase model performance in practice.

3.1 Default parameters

Six types of feature selectors are evaluated as elementary models for UBayFS:

- Minimum Redundancy Maximum Relevance (mRMR) Ding and Peng (2005),
- Fisher score Bishop (1995),
- Decision tree for classification Breiman et al. (1984),
- Recursive feature elimination (RFE) Guyon et al. (2002),
- Hilbert-Schmidt Independence Criterion Lasso (HSIC) Yamada et al. (2014),
- Lasso Tibshirani (1996).

However, the main focus of the present work is to evaluate the generic concept of UBayFS rather than to provide an in-depth analysis of these elementary feature selectors.

Our implementation of UBayFS in R (R Core Team, 2020)⁵ uses the Genetic Algorithm package authored by Scrucca (2013) with $T = 100$ and $Q = 100$; in most cases, convergence is achieved after around ten iterations. By default, each UBayFS setup comprises an uninformative prior with $\alpha_n = 0.01$ for all $n \in [N]$, and a max-size constraint instructing to select b_{MS} features, which is determined individually for each dataset. Thus, by default, the constraint system is given as:

$$A = (1 \ 1 \ \dots \ 1), \mathbf{b} = b_{MS}, \rho = 1.$$

⁵ For implementation and experimental setups, see <https://github.com/annajenu/UBayFS> and https://github.com/annajenu/UBayFS_experiments; for details, see Appendix B.

No further user knowledge or side constraints are introduced unless stated explicitly in the particular setups. Each setup is executed in $I = 10$ independent runs $i \in [I]$, representing distinct random splits of the dataset \mathcal{D} into train data $T_{\text{train}}^{(i)}$ and test data $T_{\text{test}}^{(i)} = \mathcal{D} \setminus T_{\text{train}}^{(i)}$ (stratified 75%/25% split).

3.2 Evaluation metrics

For the synthetic datasets, performance is measured by the F1 score of correctly / incorrectly selected features since the ground truth about the relevance of features is known from the simulation procedure. For real-world data, F1 scores refer to the predictive results obtained by training a classification model after feature selection, and judge the feature selection quality indirectly. Furthermore, all experiments evaluate the *stability* measure by Nogueira et al. (2018) across I independent feature selection runs. Stability ranges asymptotically in $[0, 1]$, where 1 indicates that the same features are selected in every run (perfectly stable). *Runtime*⁶ refers to the time the model requires to perform feature selection, including elementary model training and optimization, but excluding any predictive model trained on top of the feature selection results. Since prior parameters have a minor influence on the runtime, times will not be provided for experiments investigating these aspects.

3.3 Experiment 1: simulation study

To investigate major properties of UBayFS, we simulate four different datasets:

- i. An additive model (experiment 1a) similar to *Data1* in Yamada et al. (2014), composed of a $(x_1, \dots, x_{1000}) \sim 1000 \times 1000$ data matrix simulated from a Gaussian distribution $N(\mathbf{0}_{1000}, \mathbf{I}_{1000})$, and a binary target variable

$$f(\mathbf{x}, \varepsilon) = g(-2 \sin(2x_1) + x_2^2 + x_3 + \exp(-x_4) + \varepsilon),$$

where x_1, \dots, x_4 denote the features 1 to 4 and $\varepsilon \sim N(0, 1)$. The function g transforms z into a class variable by

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{otherwise;} \end{cases}$$

- ii. A non-additive model (experiment 1a) similar to *Data2* in Yamada et al. (2014), equivalent to the setup of i., except for a multiplicative target variable

$$f(\mathbf{x}, \varepsilon) = g(x_1 \cdot \exp(2x_2) + x_3^2 + \varepsilon);$$

- iii. A simulated dataset (experiment 1b, 1c) with group structure among the features, produced via *make_classification* (Pedregosa, 2011), delivering a 512×256 dataset with 8 feature blocks à 32 features—4 of these blocks contain relevant features (4 important features per block), 2 blocks contain redundant features representing arbitrary linear combinations of the relevant features (3 redundant features per block);

⁶ CentOS Linux 7.9.2009, Intel Xeon(R) CPU E5-2650 @ 2.60GHz, 3 GB RAM, R v3.6.0.

- iv. Another dataset simulated via *make_classification*, comprising 32 features in total (16 important, 16 redundant) without block structure. This smaller dataset (64×32) has a complicated correlation structure due to the high number of redundant features and is used to evaluate UBayFS variants that take feature dependence into account (experiment 1d).

The maximum number of selected features b_{MS} is set to the ground truth number of relevant features, i.e. $b_{MS} = 4$ (dataset i.), $b_{MS} = 3$ (dataset ii.), and $b_{MS} = 16$ (datasets iii. and iv.), respectively. The default constraint shape parameters for MS is set to $\rho_{MS} = 1$. Unless otherwise stated, the prior weights are set to a constant, uninformative value of $\alpha = 0.01$ for all features.

In addition to the constraint shape ρ associated with a single constraint, λ balances the overall impact of side constraints with the Dirichlet-multinomial model. A small parameter $\lambda < 1$ is not recommended since a lack of influential constraints (including the MS constraint) results in selecting all features due to an unregularized utility function U . On the other hand, a high λ has a similar effect as setting all shape parameters uniformly to $\rho = \infty$; thus, all constraints are required to be fulfilled. In this study, λ has only a minor impact on the resulting model metrics and, therefore, is set to $\lambda = 1$.

3.3.1 Experiment 1a—likelihood parameters

Figure 2 demonstrates the effect of an increasing number of elementary models M to build the feature selector. M represents the parameter to steer the likelihood. Due to their excessive runtimes, HSIC and RFE are computed only for $M \leq 10$, while all other elementary feature selectors are evaluated for up to $M = 200$.

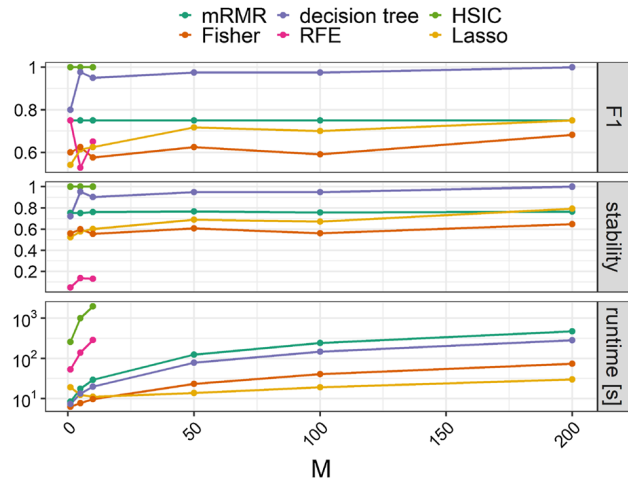
As expected, a higher M contributes largely to the runtime of the model, which increases linearly. In contrast, both F1 scores and stability values begin to saturate at around $M = 50$ to $M = 100$ models. Even though large ensembles are intractable with HSIC and RFE, small ensembles with $M = 5$ allow HSIC to retrieve almost all features, whereas simpler elementary feature selectors struggle to achieve high performances and stabilities even at higher levels of M . We conclude that large M does not necessarily improve the results but significantly impacts the runtime. Thus $M \approx 100$ appears to be a reasonable choice in the subsequent settings, except for HSIC and RFE, where $M = 5$ will be set as a default.

3.3.2 Experiment 1b—“correct” and “incorrect” prior weights

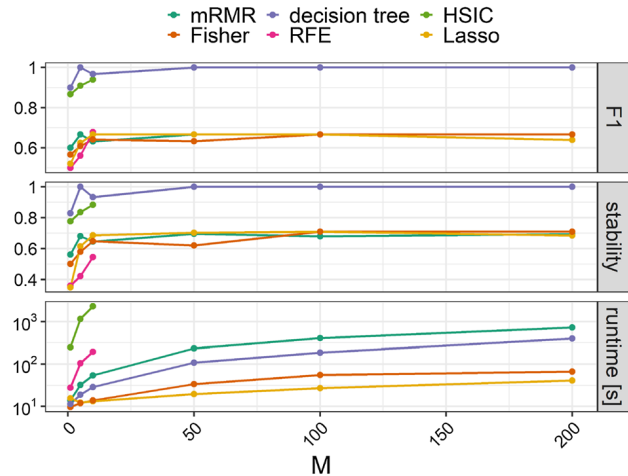
To investigate the effect of prior weights α , we alter the prior weights in dataset iii. by feature block. A constant prior weight α_R is assigned to all features from relevant blocks, i.e., blocks 1-4 containing informative and non-informative features. In contrast, features from blocks 5-8 (containing only non-informative features) are assigned a constant prior weight α_{-R} —thereby, we simulate that the expert has approximate, yet not exact beliefs about feature relevance. By assigning higher prior weights $\alpha_R > \alpha_{-R}$, the experiment simulates an agreement between the expert belief and the ground truth (“correct prior”), while a lower $\alpha_R < \alpha_{-R}$ represents “wrong” prior information (“incorrect prior”). To simulate correct and incorrect prior knowledge at different levels, we increase α_R while setting α_{-R} to the default value 0.01, and vice versa.

Figure 3 illustrates that, as expected, feature selection performance in terms of F1 scores (evaluated with respect to the ground truth features) increases for higher α_R and decreases

Fig. 2 Different numbers of elementary models M



(a) additive classification dataset



(b) non-additive classification dataset

for higher α_{-R} . Thus, across all elementary feature selectors, an improvement of the uninformative case $\alpha_R = \alpha_{-R} = 0.01$ can be achieved by an informative prior, if the prior represents a reasonable overlap with reality—this holds even though the relevant blocks also contain uninformative features, which are incremented by α_R as well. On the other hand, erroneous prior knowledge can impact the feature selection results negatively. In contrast to the feature-wise F1 scores, stability remains mostly unaffected from strong prior knowledge on relevant or irrelevant blocks—incorrect prior knowledge merely tends to decrease stability to a minor degree.

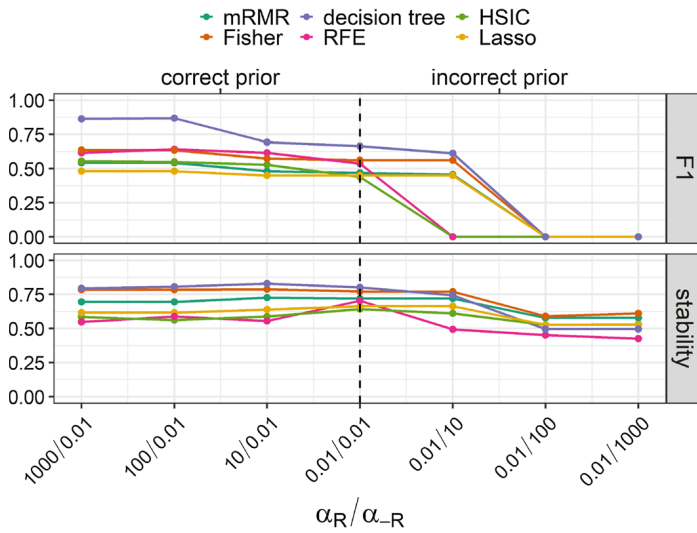


Fig. 3 Different prior weights assigned to relevant blocks, α_R , and to non-relevant blocks, α_{-R}

3.3.3 Experiment 1c—side constraints

We investigate the following opposite constraint types:

- *Block-max-size* (BMS): features are selected from at most b_{BMS} distinct blocks, and
- *Max-per-block* (MPB): at most b_{MPB} features are selected from each block.

BMS is designed to enforce a clustering behavior, where all selected features originate from a maximum number of $b_{BMS} = 4$ blocks. On the other hand, MPB aims to disperse the selection, indicating that a maximum number of $b_{MPB} = 2$ features per block is favorable. The strength of these constraints is steered via the corresponding shape parameters ρ_{BMS} and ρ_{MPB} , respectively, while $\rho = 0$ indicates that a constraint is omitted. From a default case of $\rho_{BMS} = \rho_{MPB} = 0$ (no block constraints), we investigate the behavior of UBayFS under one of the two constraints at a time at an increasing level of ρ_{BMS} or ρ_{MPB} .

Fig. 4 illustrates how the opposite side constraints BMS and MPB affect the model at different levels of relaxation parameters. Both constraint types have a slightly negative impact on the outcome in terms of F1 and stability. This is caused by the fact that the “best” feature set has to be determined under a side constraint, which is not compatible with the ground truth—the ground truth defines 16 features out of four distinct blocks to be relevant, which cannot be covered by any of the constraints. Therefore, we can observe that UBayFS can handle such scenarios and still deliver appropriate and near-optimal solutions.

3.3.4 Experiment 1d—between-feature correlations

In Sect. 2, multiple variants were discussed to account for datasets with a given correlation structure. On the one hand, the UBayFS framework permits to account for between-feature correlations via a generalization of the prior distribution; on the other hand, we may enforce that the highly correlated features should not be selected jointly via a decorrelation constraint. Both variants are different insofar as generalized priors aim to deliver a more

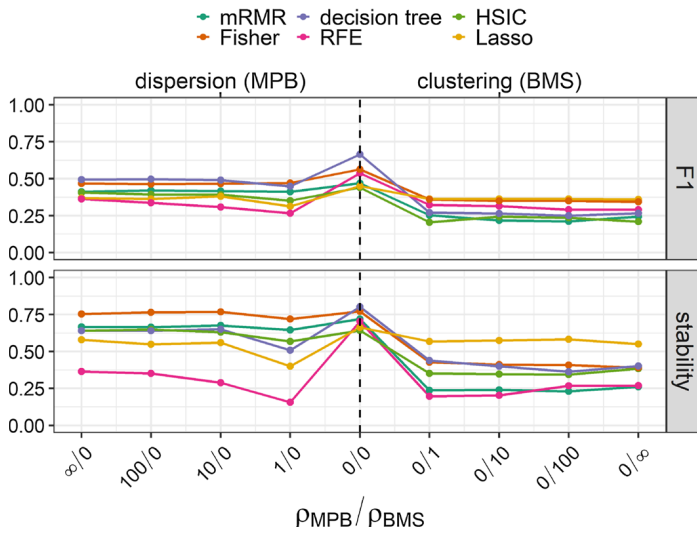


Fig. 4 Different prior constraints assigned to blocks: MPB (maximum one feature per block) and BMS (block max-size) constraint types at distinct levels of ρ . The special case $\rho = 0$ indicates that the corresponding constraint is omitted

appropriate estimation of the expected feature importances by correcting for dependencies in the observed feature sets, while decorrelation constraints directly affect the optimization procedure for δ .

In this experiment, we investigate both possibilities to account for between-feature correlations, along with combinations of both: we set a decorrelation constraint between all features with a mutual Spearman correlation $\tau > 0.4$ as described in Sect. 2.3, such that joint selection of highly correlated features is penalized. Further, we apply the following prior setups:

- Dirichlet prior distribution (default),
- Generalized Dirichlet distribution Wong (1998),
- Hyperdirichlet distribution Hankin (2010).

Our experiment involves all combinations of prior setups with and without decorrelation constraint, executed on dataset iv. To measure the effect of decorrelation, we further evaluate the redundancy rate (RED) Zhao et al. (2010), defined as the average absolute Pearson correlation among selected features. A small RED is commonly preferred in practical setups.

The results in Fig. 5 show that neither feature-wise F1 scores nor stabilities change significantly between the prior models. Thus, the default Dirichlet model seems sufficient in practice. However, introducing decorrelation constraints has a slightly negative impact on stability, while yielding a small improvement in F1 scores and RED. Nonetheless, the most significant change between the variants can be observed with respect to runtime, which reflects the high computational burden associated with the hyperdirichlet prior model—even on a small dataset, the runtimes show a significant increase on a logarithmic scale. Thus, higher-dimensional datasets can only be tackled at an enormous computational cost with the hyperdirichlet setup.

Fig. 5 Different setups to account for dependence structures between features

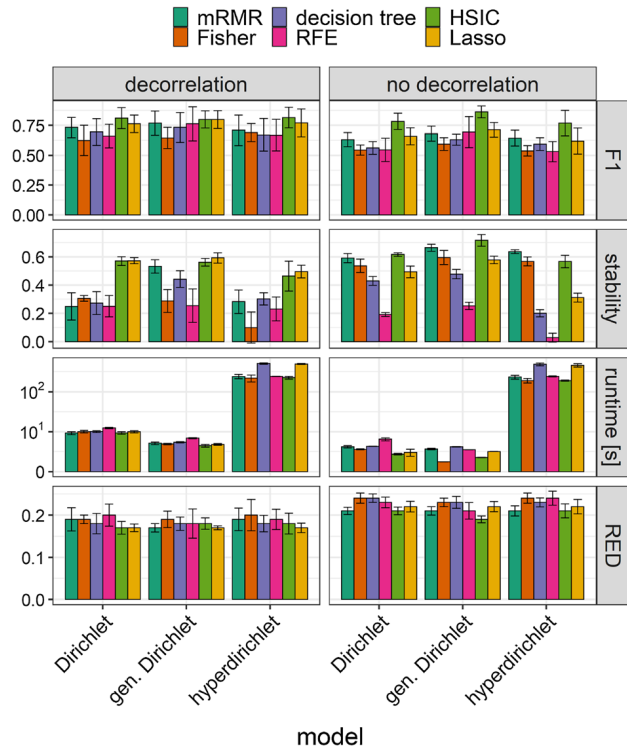


Table 3 Real-world binary classification datasets from the life science domain used for experimental evaluation. For p53, a stratified subset out of > 16000 rows was used from the original dataset for this experiment

| Dataset / source | # Features | # Rows | b_{MS} | # Blocks | b_{BMS} |
|-----------------------------------------------------------------|------------|--------|----------|----------|-----------|
| Breast cancer wisconsin (BCW) Wolberg and mangasarian (1990) | 30 | 569 | 5 | 3 | 1 |
| Heart disease (HD) Detrano (1989) | 46 | 101 | 5 | – | – |
| Mice protein expression (MPE) Higuera et al. (2015) | 77 | 552 | 5 | – | – |
| Colon gene expression (COL) Yang and Zou (2015) | 100 | 62 | 5 | 20 | 2 |
| LSVT voice rehabilitation Tsanas (2013) | 310 | 126 | 10 | 14 | 2 |
| p53 Danziger (2006) | 5409 | 351 | 20 | 2 | 1 |
| Prostate (PRO) Singh (2002) | 6033 | 102 | 20 | – | – |
| Leukaemia (LEU) Golub (1999) | 7129 | 72 | 20 | – | – |
| Lung cancer (LUNG) Gordon (2002) | 12533 | 181 | 100 | – | – |

3.4 Experiment 2: real-world datasets

Numerical studies are conducted on eight open-source datasets presenting binary classification problems from the life science domain, see Table 3. For simplicity and due to extensive runtimes, we restrict the choice of the elementary feature selector for UBayFS to mRMR, Fisher, and decision tree with an uninformative prior, an MS constraint, and $M = 100$. The number of selected features is specified according to the size of the dataset ($b_{MS} = 5 / 10 / 20 / 100$ for datasets with fewer than 100 / between 100 and 1000 / between 1000 and 10000 / more than 10000 features, respectively).

In addition to conventional feature selection (scenario 1) with max-size constraint b_{MS} , specified in Table 3, we evaluate a block feature selection (scenario 2) for datasets with block-wise feature structure. For block feature selection, up to b_{MS} features should be selected from at most b_{BMS} distinct blocks.⁷ Random forests (RF) Breiman (2001), and RENT Jenul (2021) (representing ensemble feature selectors that extend the concepts of decision trees and elastic net regularized models, respectively) are used as state-of-the-art benchmarks for standard feature selection, while Sparse Group Lasso (GL) Ida et al. (2019) is used as the benchmark for block feature selection. To conform with UBayFS, RENT and RF are adjusted to $M = 100$ elementary models, and all models are tuned to select approximately the same number of features, b_{MS} . Since RENT and GL cannot be instructed to select b_{MS} features directly, regularization parameters are determined via bisection, such that the number of selected features is approximately equal to b_{MS} .

The selected features cannot be evaluated directly in real-world datasets due to unknown ground truth on the feature relevance. Therefore, we train predictive models on $T_{train}^{(i)}$ after feature selection and evaluate the selected features indirectly via the predictive performance on the test instances. To reduce the influence of the predictive model type, we train two distinct classifiers on $T_{train}^{(i)}$ after feature selection, and report F1 scores for predictions on $T_{test}^{(i)}$ for both. The choice of baseline classifiers to obtain the prediction comprises:

- generalized linear model: logistic regression (GLM),
- support vector machine (SVM).

3.4.1 Results

Tables 4 and 5 present the results of the experiments on real-world data. Thereby, UBayFS achieves good predictive F1 scores throughout the different datasets, even though no expert knowledge is introduced to ensure a fair comparison. In the block feature selection setups, UBayFS benefits from block constraints and shows more flexibility than Sparse Group Lasso. Altogether, UBayFS can keep up with its competitors in terms of predictive performance in a diverse range of scenarios (low-dimensional and high-dimensional data, as well as unconstrained and constrained setups) while providing higher flexibility to introduce additional information or constraints. Overall, the results reflect that a particular strength of UBayFS lies in delivering a good trade-off between stabilities and predictive performance, compared to competitors such as RF, which deliver high F1 scores, but very low stabilities.

⁷ Details on the block structure of the datasets are provided in Appendix B.

Table 4 UBayFS with three distinct elementary feature selectors (M: mRMR, F: Fisher, T: decision tree) is compared to ensemble feature selectors RF and RENT in a standard feature selection scenario. UBayFS with additional (BMS) constraint is compared to Sparse Group Lasso (GL) for block-feature selection on datasets with block structure. Average F1 scores are given for different predictive models (GLM, SVM). The best scores for each dataset and evaluation metric are marked in bold—standard feature selection and block feature selection are assessed separately

| Dataset | Standard feature selection | | | | | Block feature selection | | | |
|------------------------------------------------|----------------------------|-------------|-------------|-------------|-------------|-------------------------|-------------|-------------|-------------|
| | RF | RENT | UBayFS | | | GL | UBayFS | | |
| | | | M | F | T | | M | F | T |
| (a) Average F1 score per run (predictor: GLM). | | | | | | | | | |
| BCW | 0.95 | 0.97 | 0.96 | 0.97 | 0.95 | 0.96 | 0.96 | 0.96 | 0.96 |
| HD | 0.92 | 0.88 | 0.91 | 0.90 | 0.93 | – | – | – | – |
| MPE | 0.86 | 0.95 | 0.87 | 0.83 | 0.83 | – | – | – | – |
| COL | 0.85 | 0.83 | 0.83 | 0.78 | 0.88 | 0.82 | 0.74 | 0.77 | 0.89 |
| LSVT | 0.70 | 0.75 | 0.80 | 0.84 | 0.68 | 0.77 | 0.67 | 0.79 | 0.59 |
| p53 | 0.71 | 0.66 | 0.80 | 0.78 | 0.80 | 0.63 | 0.76 | 0.79 | 0.79 |
| PRO | 0.88 | 0.89 | 0.78 | 0.85 | 0.84 | – | – | – | – |
| LEU | 0.88 | 0.93 | 0.88 | 0.91 | 0.95 | – | – | – | – |
| LUNG | 0.93 | 0.97 | 0.91 | 0.90 | 0.92 | – | – | – | – |
| Dataset | Standard feature selection | | | | | Block feature selection | | | |
| | RF | RENT | UBayFS | | | GL | UBayFS | | |
| | | | M | F | T | | M | F | T |
| (b) Average F1 score per run (predictor: SVM). | | | | | | | | | |
| BCW | 0.95 | 0.97 | 0.96 | 0.96 | 0.94 | 0.97 | 0.96 | 0.96 | 0.95 |
| HD | 0.92 | 0.88 | 0.91 | 0.91 | 0.95 | – | – | – | – |
| MPE | 0.87 | 0.95 | 0.89 | 0.84 | 0.84 | – | – | – | – |
| COL | 0.86 | 0.85 | 0.87 | 0.83 | 0.88 | 0.81 | 0.82 | 0.79 | 0.89 |
| LSVT | 0.75 | 0.75 | 0.80 | 0.84 | 0.71 | 0.80 | 0.79 | 0.79 | 0.57 |
| p53 | 0.81 | 0.82 | 0.81 | 0.80 | 0.82 | 0.84 | 0.77 | 0.82 | 0.80 |
| PRO | 0.91 | 0.90 | 0.87 | 0.88 | 0.85 | – | – | – | – |
| LEU | 0.96 | 0.94 | 0.88 | 0.95 | 0.96 | – | – | – | – |
| LUNG | 0.98 | 0.97 | 0.98 | 0.96 | 0.94 | – | – | – | – |

Figures 6 and 7 give additional insights into the performances of the UBayFS variants in the standard feature selection and block feature selection scenario, respectively. Differences between the F1 scores obtained by the different elementary feature selectors underline that UBayFS inherits benefits and drawbacks from its underlying elementary model type—in particular, the decision tree and HSIC achieved top results. Nevertheless, the building of ensembles allows to compensate in parts for mediocre stabilities.

3.4.2 Case study with prior knowledge

Our evaluations underlined the applicability of UBayFS in real-world scenarios. However, due to the absence of prior knowledge, these scenarios covered only parts of the capabilities of

Table 5 Mean stabilities of UBayFS with three distinct elementary feature selectors (M: mRMR, F: Fisher, T: decision tree), compared to ensemble feature selectors RF and RENT in standard feature selection, as well as to GL in block feature selection scenarios. The best scores in each row are marked in bold for each scenario

| Dataset | Standard feature selection | | | Block feature selection | | | | | |
|---------|----------------------------|-------------|-------------|-------------------------|-------------|-------------|-------------|-------------|------|
| | RF | RENT | UBayFS | | | GL | UBayFS | | |
| | | | M | F | T | | M | F | T |
| BCW | 0.73 | 0.87 | 0.87 | 1.00 | 0.61 | 0.90 | 0.80 | 0.80 | 0.80 |
| HD | 0.45 | 0.87 | 0.88 | 0.65 | 0.59 | – | – | – | – |
| MPE | 0.72 | 0.87 | 0.92 | 0.85 | 0.77 | – | – | – | – |
| COL | 0.39 | 0.67 | 0.80 | 0.72 | 0.81 | 0.56 | 0.84 | 0.72 | 0.82 |
| LSVT | 0.31 | 0.59 | 0.72 | 0.79 | 0.55 | 0.73 | 0.66 | 0.88 | 0.31 |
| p53 | 0.11 | 0.56 | 0.34 | 0.34 | 0.36 | 0.68 | 0.19 | 0.25 | 0.31 |
| PRO | 0.17 | 0.53 | 0.56 | 0.61 | 0.42 | – | – | – | – |
| LEU | 0.07 | 0.64 | 0.46 | 0.76 | 0.53 | – | – | – | – |
| LUNG | 0.18 | 0.78 | 0.80 | 0.79 | 0.40 | – | – | – | – |

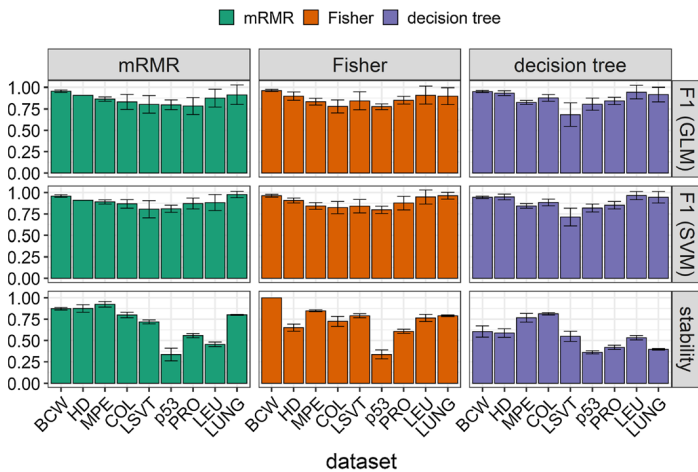


Fig. 6 Performance results of UBayFS feature selection on real-world datasets (MS constraint). F1 scores are determined after training and predicting a classifier (GLM or SVM) after feature selection. Results show mean values over $I = 10$ runs along with standard deviations

the method. To exploit prior knowledge in practice, we revisit the lung cancer genome dataset (LUNG): in the dataset, eight gene expression features were identified as relevant in biological studies by Guan Guan et al. (2009). Thus, we assign higher prior weights α_R to a-priori relevant features, while all other features get assigned the default prior weight $\alpha_{-R} = 0.01$. Our setups include one with “weak” prior ($\alpha_R = 20$), and one with “strong” prior ($\alpha_R = 100$), in addition to the setup without prior, shown in Table 4. The max-size constraint is set to $b_{MS} = 100$.

As summarized in Table 6, incorporating prior knowledge leads to an improvement of UBayFS results in most cases. Thus, the absolute performance lies in a similar top range as those reported in previous work by Brahim and Limam (2014), who evaluated averaged accuracies in a comparable setup on the same dataset (> 0.99 avg. accuracy). However, the comparability of accuracies is limited due to the unbalanced nature of the dataset. Between the UBayFS setups, results with weak prior are similar to those from no-prior results in the case of stable elementary feature selectors (mRMR and Fisher). In contrast, weak prior results resemble the strong prior in the case of a

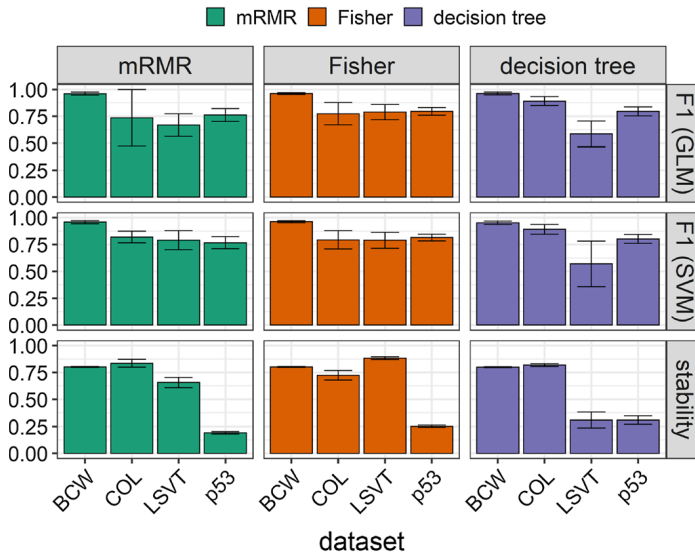


Fig. 7 Performance results of UBayFS block feature selection on real-world datasets (MS and BMS constraints). F1 scores are determined after training and predicting a classifier (GLM or SVM) after feature selection. Results show mean values over $I = 10$ runs along with standard deviations

Table 6 Average performance scores delivered by UBayFS on the LUNG dataset with and without prior knowledge

| Setup | GLM | | | SVM | | | Stability | | |
|-------------------------------------|------|------|------|------|------|------|-----------|------|------|
| | M | F | T | M | F | T | M | F | T |
| Without prior | 0.91 | 0.90 | 0.92 | 0.98 | 0.96 | 0.94 | 0.80 | 0.79 | 0.40 |
| With prior ($\alpha_{imp} = 20$) | 0.91 | 0.90 | 0.91 | 0.98 | 0.96 | 0.96 | 0.80 | 0.79 | 0.45 |
| With prior ($\alpha_{imp} = 100$) | 0.91 | 0.94 | 0.91 | 0.98 | 0.96 | 0.96 | 0.82 | 0.81 | 0.45 |

non-stable elementary feature selector (decision tree). Thus, a weak prior has a higher impact on the final results if the elementary models are more diverse.

3.4.3 Runtime

Runtimes of all methods and datasets are provided in Table 7. Given a fixed set of model parameters, it becomes obvious that the major factor influencing the runtime of UBayFS is the number of features (columns) rather than the number of samples (rows). UBayFS runtimes refer to the MS setup—however, experiments showed only minor differences to the runtimes in the block feature selection setup. While RF and GL are more tractable in high-dimensional datasets, RENT seems to suffer from data dimensionality to a more considerable extent.

Across larger datasets, the main influencing factor on the runtime is the number and type of elementary models. For example, on the LUNG dataset (> 12000 features), the training procedure of 100 mRMR models as elementary models comprised 40 minutes

Table 7 Average runtime per run [s]

| Dataset | RF | RENT | GL | UBayFS | | |
|---------|-------|--------|-------|--------|-------|--------|
| | | | | M | F | T |
| BCW | 6.7 | 3.4 | 10.9 | 6.2 | 2.2 | 4.3 |
| HD | 6.3 | 3.2 | – | 1.8 | 1.6 | 2.1 |
| MPE | 9.4 | 24.3 | – | 12.3 | 5.3 | 9.6 |
| COL | 6.1 | 3.8 | 4.6 | 3.7 | 2.9 | 3.6 |
| LSVT | 10.0 | 77.9 | 9.0 | 6.4 | 6.7 | 9.6 |
| p53 | 80.2 | 2712.3 | 112.7 | 366.8 | 125.6 | 440.3 |
| PRO | 29.8 | 1217.2 | – | 370.9 | 232.6 | 708.0 |
| LEU | 41.5 | 980.9 | – | 263.0 | 160.8 | 549.5 |
| LUNG | 116.8 | 2834.1 | – | 1930.3 | 535.1 | 1885.0 |

(88% of UBayFS runtime), while optimization using the Genetic Algorithm comprised 5 minutes (11% of UBayFS runtime).⁸

4 Discussion and conclusion

The presented Bayesian feature selector UBayFS has its strength in combining information from a data-driven ensemble model with expert prior knowledge targeted at life science applications. The generic framework is flexible in the choice of the elementary feature selector type, allowing a broad scope of applications scenarios by deploying adequate elementary feature selectors, such as those suggested by Sechidis and Brown (2018) for semi-supervised or Elghazel and Aussem (2015) for unsupervised problems. An extension of the presented experiments to multiple classes or multi-label classification problems (one object is not uniquely assigned to one class) is straightforward as well if the elementary feature selector is capable of tackling such datasets, such as Petković et al. (2020).

In general, the choice of the elementary feature selector is a central step when deploying the concept in practice—in particular, the size and structure of a dataset need to be taken into account. This work presented a broad range of elementary models to provide user guidance in practical setups. The option to build ensembles combining different model types, as discussed by Seijo-Pardo et al. (2017), turned out to deteriorate the stability of ensemble feature selectors and hence, is not considered in this study.

UBayFS presents two ways to account for feature dependencies: a generalized prior model as well as a decorrelation constraint. The latter effectively restricts the results, such that a simultaneous selection of highly correlated features is penalized. The generalizations of the prior model correct the estimated feature importances by the dependencies—in a low-dimensional scenario, the hyperdirichlet variant is the most accurate choice. However, this variant becomes intractable, if the dimensionality exceeds a few hundred features and requires simulation to determine the expected value in almost any case, preventing from analytically exact solutions. Since our experiments depicted that feature importances obtained from each of the three prior setup types are numerically similar, a conventional Dirichlet setup seems to deliver a sufficiently accurate approximation

⁸ Runtime information refers to the current version of the implementation and is subject to further code optimization.

for high-dimensional datasets. This observation is also supported by the fact that many elementary feature selectors, such as mRMR or HSIC, can account for between-feature correlations, thus reducing the need to consider correlations in the meta-model.

Prior information from experts is introduced via prior feature weights and linking constraints describing between-feature dependencies, represented in a system of side constraints. Via a relaxation parameter, the inadmissibility is transferred into a soft constraint, favoring solutions that fulfill the constraints and penalizing violations. Introducing user knowledge directly into the feature selection process opens new opportunities for data analysis in life science applications. Still, such methodology bears the potential of intentional or unintentional misuse: as demonstrated in the experiment, the integration of unreliable or incorrect user knowledge may distort predictive results. Users have to be aware that UBayFS may contain subjective inputs and thus, take precautions to ensure that prior information is sufficiently verified, e.g., by published research in the field.

Based on the results from extensive experimental evaluations on multiple open-source datasets, a clear benefit of the proposed feature selector lies in the balance between predictive performance and stability. Particularly in life sciences, where few instances are available in high-dimensional datasets, user-guided feature selection is an opportunity to guide models to achieve otherwise intractable results. UBayFS delivers more flexibility to integrate domain knowledge than established state-of-the-art approaches. A practical limitation of UBayFS is that the runtime is arguably slower than simpler feature selectors, which becomes an obstacle in very high-dimensional datasets. The use of highly optimized algorithms like the Genetic Algorithm, along with an initialization using the suggested Alg. 1 mitigates this issue. However, it cannot compensate for the computational burden of training multiple elementary models.

Appendix A theory

A.1 Convergence of inadmissibility function

The point-wise convergence $\kappa_{k,\rho} \xrightarrow{\rho \rightarrow \infty} \kappa_k$ holds for arbitrary $\mathbf{A} \in \mathbb{R}^{K \times N}$ and $\mathbf{b} \in \mathbb{R}^K$ on the domain $\mathcal{D} = \{0, 1\}^N$.

Proof From the definition of $\kappa_{k,\rho}(\boldsymbol{\delta})$, the claim is trivially fulfilled for

$$\boldsymbol{\delta} \in \left\{ \boldsymbol{\delta}' \in \{0, 1\}^N : (\mathbf{a}^{(k)})^T \boldsymbol{\delta}' - b^{(k)} \leq 0 \right\}.$$

In the opposite case, we define λ_k as $\lambda_k = (\mathbf{a}^{(k)})^T \boldsymbol{\delta} - b^{(k)} > 0$. It holds that

$$\begin{aligned} \kappa_{k,\rho}(\boldsymbol{\delta}) &= \frac{1 - \xi_{k,\rho}}{1 + \xi_{k,\rho}} \\ &= \frac{1 - \exp(-\rho\lambda_k)}{1 + \exp(-\rho\lambda_k)}. \end{aligned}$$

Since $\lambda_k > 0$, we obtain $-\rho\lambda_k \xrightarrow{\rho \rightarrow \infty} -\infty$, and thus $\xi_{k,\rho} = \exp(-\rho\lambda_k) \xrightarrow{\rho \rightarrow \infty} 0$. It follows that $\kappa_{k,\rho}(\boldsymbol{\delta}) \xrightarrow{\rho \rightarrow \infty} 1$. Hence, we have shown a point-wise convergence of

$$\kappa_{k,\rho}(\boldsymbol{\delta}) \xrightarrow{\rho \rightarrow \infty} \begin{cases} 1 & \text{if } \lambda_k \leq 0 \\ 0 & \text{if } \lambda_k > 0, \end{cases}$$

which equals to κ_k on the domain \mathcal{D} .

A.2 Generalizations of the Dirichlet distribution

In Sect. 2.2, we discuss the possibility to replace the Dirichlet distribution with one out of two generalized variants:

- the generalized Dirichlet distribution, and
- the hyperdirichlet distribution.

Both variants preserve the conjugate prior property with respect to the multinomial likelihood, as explained by the corresponding authors who had introduced these generalizations. In this part, we provide a short overview on the probability density functions, parameters and (posterior) expected values of these distributions, as these quantities are relevant for the UBayFS setup.

The standard Dirichlet distribution, see e.g. DeGroot (2005), is commonly defined by the probability density function

$$f_{\text{Dir}}(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{n=1}^N \theta_n^{\alpha_n - 1}, \quad (16)$$

where $B(\boldsymbol{\alpha}) = \frac{\prod_{n=1}^N \Gamma(\alpha_n)}{\Gamma(\sum_{n=1}^N \alpha_n)}$ denotes the multivariate beta function. Due to the simple parameter update in the inference step, we obtain the posterior expected value

$$\mathbb{E}_{\boldsymbol{\theta}|\mathbf{y}}[\boldsymbol{\theta}] = \frac{1}{\|\boldsymbol{\alpha}^\circ\|_1} \boldsymbol{\alpha}^\circ,$$

where $\boldsymbol{\alpha}^\circ = \boldsymbol{\alpha} + \mathbf{y}$.

In essence, the generalized Dirichlet distribution by Wong (1998) adds an additional parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{N-1}$ to the parameter vector $\boldsymbol{\alpha}$ from the Dirichlet distribution and is defined via the probability density

$$f_{\text{gDir}}(\boldsymbol{\theta}') = \prod_{n=1}^{N-1} \frac{1}{B(\alpha_n, \beta_n)} (\theta'_n)^{\alpha_n - 1} \left(1 - \sum_{i=1}^n \theta'_i \right)^{\gamma_n}, \quad (17)$$

where $B(\alpha_n, \beta_n) = \frac{\Gamma(\alpha_n)\Gamma(\beta_n)}{\Gamma(\alpha_n + \beta_n)}$, $\gamma_n = \beta_n - \alpha_{n+1} - \beta_{n+1}$ for $n \in [N-2]$, and $\gamma_{N-1} = \beta_{N-1} - 1$. In contrast to the standard Dirichlet setting, the distribution is defined on the $N-1$ -dimensional space, relaxing the side constraint $\|\boldsymbol{\theta}\|_1 = 1$ to $\|\boldsymbol{\theta}'\|_1 \leq 1$, $\boldsymbol{\theta}' \in \mathbb{R}^{N-1}$ — both are

equivalent, if $\theta_n = \theta'_n$ for $n \in [N - 1]$, and $\theta_N = 1 - \sum_{n=1}^{N-1} \theta'_n$. The posterior expected value for the generalized Dirichlet distribution is given in closed-form by

$$(\mathbb{E}_{\theta}[\theta])_n = \begin{cases} \frac{\alpha_n + y_n}{\alpha_n + \beta_n + v_n} & n = 1 \\ \frac{\alpha_n + y_n}{\alpha_n + \beta_n + v_n} \prod_{i=1}^{n-1} \frac{\beta_i + n_{i+1}}{\alpha_i + \beta_i + n_i} & n = 2, \dots, N - 1 \\ \prod_{i=1}^{N-1} \frac{\beta_i + n_{i+1}}{\alpha_i + \beta_i + v_i} & n = N, \end{cases}$$

where $v_n = \sum_{i=n}^N y_i$, see Wong (1998).

An even more general version is the hyperdirichlet distribution by Hankin (2010), who characterizes the distribution by the probability density function

$$f_{\text{hDir}}(\theta) \propto \left(\prod_{n=1}^N \theta_n \right)^{-1} \prod_{G \in \mathcal{P}([N])} \left(\sum_{i \in G} \theta_i \right)^{\mathcal{F}(G)}, \quad (18)$$

where $\mathcal{P}(\cdot)$ denotes the power set and $\mathcal{F}(G)$ denotes the parameter for each possible subset of $[N]$. Since the closed-form expression of the expected value involves the normalization constant, which is intractable in practical high-dimensional setups, we deploy the Metropolis-Hastings (MH) algorithm implemented in Hankin (2017) to sample from the hyperdirichlet distribution and determine the expected value empirically from the sample mean.

Table 8 Dataset sources

| Name | Link |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HD | https://archive.ics.uci.edu/ml/datasets/heart+disease |
| BCW | https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic) |
| MPE | https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression |
| COL | https://github.com/cran/gglasso |
| LVST | https://archive.ics.uci.edu/ml/datasets/LSVT+Voice+Rehabilitation |
| p53 | https://archive.ics.uci.edu/ml/datasets/p53+Mutants |
| LEU | see R package <i>spls</i> Chung et al. (2019) |
| PRO | see R package <i>propOverlap</i> Mahmoud (2014) |
| LUNG | https://leo.ugr.es/elvira/DBCRepository/LungCancer/LungCancer-Harvard2.html |

Table 9 Block indices for datasets with block structure. Feature names indicate the column name patterns, which is used for defining blocks

| Dataset | Block no | Indices | Feature names |
|---------|----------|------------------------------------|---------------|
| BCW | 1 | 1–10 | Mean |
| | 2 | 11–20 | Error |
| | 3 | 21–30 | Worst |
| COL | 1 | 1–5 | |
| | 2 | 6–10 | |
| | ⋮ | ⋮ | |
| | 20 | 96–100 | |
| LSVT | 1 | 97–124 | Delta |
| | 2 | 160–179, 200–219, 251–270, 291–310 | Det |
| | 3 | 129–139, 220–230 | E |
| | 4 | 140–159, 180–199, 231–250, 271–290 | Entropy |
| | 5 | 62–67 | GNE |
| | 6 | 52–53 | HNR |
| | 7 | 77–82 | IMF |
| | 8 | 1–30 | Jitter |
| | 9 | 84–96 | MFCC |
| | 10 | 54–55 | NHR |
| | 11 | 56–58 | OQ |
| | 12 | 31–51 | Shimmer |
| | 13 | 68–76 | VFER |
| | 14 | 59–61, 83, 125–128 | Other |
| p53 | 1 | 14826 | |
| | 2 | 4827–5408 | |

Appendix B Experimental datasets

All real-world datasets are publicly available (status: 12/2021), see Table 8. For datasets with block structure (BCW, COL, LSVT and p53), block indices are given in Table 9.

Acknowledgements In special we thank Kristian Hovde Liland (NMBU), Cecilia Marie Futsaether (NMBU) and Eirik Malinen (University of Oslo) for their constructive discussions and valuable input for this work, as well as Michael P. Alley (Penn State University) for proof-reading the paper.

Author contributions AJ, SS and JP developed the theory part of this work. AJ, SS and OT planned and conducted the associated experiments. AJ and SS wrote the manuscript. All authors contributed to the proof-reading and editing of the paper.

Funding Open access funding provided by Norwegian University of Life Sciences. This work was partly funded by the Norwegian Cancer Society (Grant no. 182672-2016).

Availability of data and materials All real-world datasets are publicly available, see Appendix B.

Code availability Code is made publicly available on GitHub, see <https://github.com/annajenu/UBayFS>.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Consent to participate All authors consented to the submission of the manuscript.

Consent for publication All real-world datasets are obtained from publicly available platforms under open licenses. All figures in this manuscript are created by the authors.

Ethics approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.
- Bose, S., Das, C., Banerjee, A., Ghosh, K., Chattopadhyay, M., Chattopadhyay, S., & Barik, A. (2021). An ensemble machine learning model based on multiple filtering and supervised attribute clustering algorithm for classifying cancer samples. *Peer J Computer Science*, 7, e671.
- Brahim, A. B., & Limam, M. (2014). New prior knowledge based extensions for stable feature selection. In *2014 6th international conference of soft computing and pattern recognition (SoCPaR)* (pp. 306–311).
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.
- Cheng, T.-H., Wei, C.-P. & Tseng, V.S. (2006). Feature selection for medical data mining: Comparisons of expert judgment and automatic approaches. In *19th IEEE symposium on computer-based medical systems (CBMS'06)* (p. 165-170).
- Chung, D., Chun, H. & Keles, S. (2019). spls: sparse partial least squares (SPLS) regression and classification [Computer software manual]. R package version 2.2-3.
- Dalton, L. A. (2013). Optimal Bayesian feature selection. In *2013 IEEE global conference on signal and information processing* (p. 65-68).
- Danziger, S., Swamidass, S., Zeng, J., Dearth, L., Lu, Q., Chen, J., et al. (2006). Functional census of mutation sequence spaces: The example of p53 cancer rescue mutants. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(2), 114–124.
- DeGroot, M. H. (2005). *Optimal statistical decisions*. Wiley.
- Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.-J., Sandhu, S., et al. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64(5), 304–310.
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(02), 185–205.
- Elghazel, H., & Aussem, A. (2015). Unsupervised feature selection with ensemble learning. *Machine Learning*, 98(1), 157–180.
- Givens, G. H., & Hoeting, J. A. (2012). *Computational statistics* (Vol. 703). John Wiley & Sons.
- Goldstein, O., Kachuee, M., Karkkainen, K., & Sarrafzadeh, M. (2020). Target-focused feature selection using uncertainty measurements in healthcare data. *ACM Transactions on Computing for Healthcare*, 1(3), 1–17.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 531–537.
- Gordon, G. J., Jensen, R. V., Hsiao, L.-L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., et al. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62(17), 4963–4967.
- Guan, P., Huang, D., He, M., & Zhou, B. (2009). Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of Experimental & Clinical Cancer Research*, 28(1), 1–7.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1), 389–422.

- Hankin, R. K. S. (2010). A generalization of the Dirichlet distribution. *Journal of Statistical Software*, 33(11), 1–18.
- Hankin, R.K.S. (2017). Partial rank data with the hyper2 package: Likelihood functions for generalized Bradley-Terry models. *The R Journal*, 9.
- Higuera, C., Gardiner, K. J., & Cios, K. J. (2015). Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLoS one*, 10(6), e0129126.
- Ida, Y., Fujiwara, Y. & Kashima, H. (2019). Fast sparse group lasso. *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.
- Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsaether, C.M. & Tomic, O. (2021). RENT—repeated elastic net technique for feature selection. *IEEE Access*, 9, 152333–152346.
- Liu, M., & Zhang, D. (2015). Pairwise constraint-guided sparse learning for feature selection. *IEEE Transactions on Cybernetics*, 46(1), 298–310.
- Lyle, C., Schut, L., Ru, R., Gal, Y., & van der Wilk, M. (2020). A Bayesian perspective on training speed and model selection. *Advances in neural information processing systems*, 33, 10396–10408.
- Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z. & Lausen, B. (2014). propOverlap: feature (gene) selection based on the proportional overlapping scores [Computer software manual]. R package version 1.0
- Nakajima, S., Sato, I., Sugiyama, M., Watanabe, K. & Kobayashi, H. (2014). Analysis of variational Bayesian latent Dirichlet allocation: Weaker sparsity than MAP. *Advances in neural information processing systems* (Vol. 27). Curran Associates, Inc.
- Nogueira, S., Sechidis, K., & Brown, G. (2018). On the stability of feature selection algorithms. *Journal of Machine Learning Research*, 18(174), 1–54.
- O’Hara, R. B., & Sillanpää, M. J. (2009). A review of Bayesian variable selection methods: What, how and which. *Bayesian Analysis*, 4(1), 85–117.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Petković, M., Džeroski, S., & Kocev, D. (2020). Multi-label feature ranking with ensemble methods. *Machine Learning*, 109(11), 2141–2159.
- Pozzoli, S., Soliman, A., Bahri, L., Branca, R. M., Girdzijauskas, S., & Brambilla, M. (2020). Domain expertise-agnostic feature selection for the analysis of breast cancer data. *Artificial Intelligence in Medicine*, 108, 101928.
- R Core Team. (2020). *R: A language and environment for statistical computing* [Computer software manual]. Austria.
- Saon, G., & Padmanabhan, M. (2001). Minimum Bayes error feature selection for continuous speech recognition. *Advances in Neural Information Processing Systems*, 13, 800–806.
- Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53(4), 1–37.
- Sechidis, K., & Brown, G. (2018). Simple strategies for semi-supervised feature selection. *Machine Learning*, 107(2), 357–395.
- Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118, 124–139.
- Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2), 203–209.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 73(3), 273–282.
- Tsanas, A., Little, M. A., Fox, C., & Ramig, L. O. (2013). Objective automatic assessment of rehabilitative speech treatment in Parkinson’s disease. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(1), 181–190.
- Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23), 9193–9196.
- Wong, T.-T. (1998). Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation*, 97(2), 165–181.
- Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., & Sugiyama, M. (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, 26(1), 185–207.
- Yang, Y., & Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6), 1129–1141.
- Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1), 49–67.
- Zhao, Z., Wang, L., Liu, H. (2010). Efficient spectral feature selection with minimum redundancy. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 24, pp. 673–678).




UBayFS: An R Package for User Guided Feature Selection

Anna Jenul ^{1*} and Stefan Schrunner ^{1*}

1 Norwegian University of Life Sciences, Ås, Norway * These authors contributed equally.

DOI: [10.21105/joss.04848](https://doi.org/10.21105/joss.04848)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: Øystein Sørensen 

Reviewers:

- [@dhvalden](#)
- [@aaronpeikert](#)
- [@EugeneHao](#)

Submitted: 01 October 2022

Published: 27 January 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Statement of Need

Feature selection, also known as variable selection in statistics, is the process of selecting important variables (features) from a list of variables in a dataset. When training predictive models, the intention behind removing the least informative features from a dataset beforehand is (a) to reduce the computational burden and mathematical limitations associated with the curse of dimensionality, and (b) to increase interpretability of the model by allowing the user to obtain insights into the relevant input variables. In particular, feature selection speeds up the training process of machine learning models, especially when the dataset is high-dimensional.

The R ([R Core Team, 2022](#)) package UBayFS implements the user-guided framework for feature selection proposed in Jenul et al. ([2022](#)), which incorporates information from the data and prior knowledge from domain experts. [Figure 1](#) demonstrates the framework. Different approaches for integrating prior knowledge in feature selection exist, though there is a lack of general and sophisticated frameworks that deliver stable and reproducible feature selection along with implementations. With its generic setup and the possibilities to specify prior weights as well as side constraints, UBayFS shows the flexibility to be applied in a broad range of application scenarios, which exceed the capabilities of conventional feature selectors while preserving large model generality. Besides side constraints, such as the option to specify a maximum number of features, the user can add must-link constraints (features must be selected together) or cannot-link constraints (features must not be selected together). In addition, constraints can be defined on feature-block level, as well. Thus, UBayFS is also capable of solving more general problems such as block feature selection. A parameter ρ regulates the shape of a penalty term accounting for side constraints, where feature sets that violate constraints lead to a lower target value. State-of-the-art methods do not cover such scenarios.

The presented R package UBayFS provides an implementation along with an interactive Shiny dashboard, which makes feature selection available to R-users with different levels of expertise. The implementation allows the user to define their own feature selectors via a function interface or to use one out of three state-of-the-art feature selectors for building the generic ensemble of feature selectors covering the data-driven component of UBayFS. State-of-the-art choices include:

- Laplacian score
- Fisher score
- mRMR

R offers multiple packages implementing feature selection methodology. To name a few, [caret](#) ([Kuhn, 2022](#)) is an essential machine learning repository, containing models with built-in feature selection such as tree based methods (for instance `rpart2`), regularized approaches like `lasso`, and non-integrated feature selectors such as recursive feature elimination `rfe`. Other examples are the Boruta ([Kursa & Rudnicki, 2010](#)) package implementing the Boruta feature selector or the `GSelection` ([Majumdar et al., 2019](#)) package containing `hsic` `lasso`

feature selection. All feature selectors available in R can be used as underlying ensemble feature selectors in UBayFS. Prior weights can be specified for single features or whole blocks as weight vectors. Linear side constraints are implemented via a matrix A and a right side b or with a customized function for specific constraint types. Hence, the sophisticated statistical model is summarized in a user-friendly and easy-to-use package.

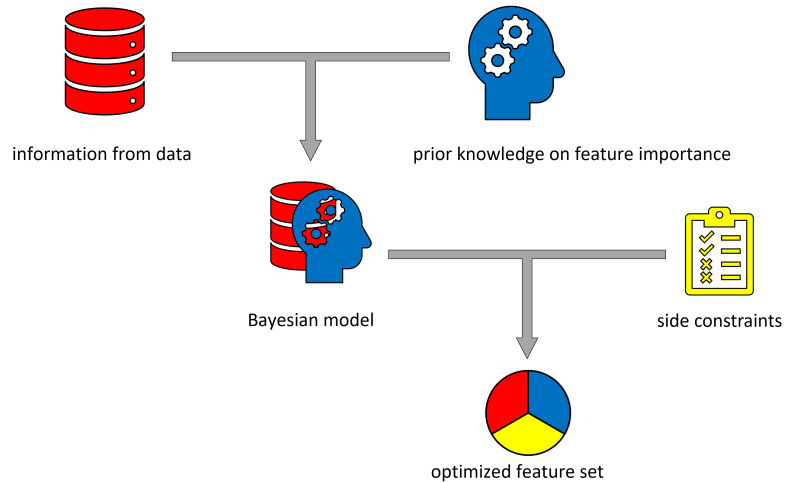


Figure 1: At first, UBayFS elaborates information directly from data via ensemble feature selection. This information is merged with prior expert knowledge (a-priori feature weights) in a Bayesian model framework. Additionally, the user can include further side constraints such as a maximum number of features or cannot-link constraints between features. The final step comprises the optimization with respect to the model’s utility function, including the side constraints.

Concept of UBayFS

As described in Jenul et al. (2022), UBayFS is a Bayesian ensemble feature selection framework. The methodology is based on quantifying a random variable θ , representing feature importances, given evidence collected from the data, denoted as y . In particular, y counts the number of elementary models in the generic ensemble of feature selectors, which select a particular feature. Statistically, we interpret the result from each elementary feature selector as a realization from a multinomial distribution with parameters θ , where $\theta \in [0, 1]^N$ defines the success probabilities of sampling each feature in an individual feature selection and thus the success probability in the ensemble. Both sources of information are combined using Bayes’ Theorem:

$$p(\theta|y) \propto p(y|\theta) \cdot p(\theta).$$

In the framework of UBayFS, $p(y|\theta)$ represents the data-driven component (implemented via a multinomial likelihood), while $p(\theta)$ describes the user knowledge part modeled with a Dirichlet distribution. Due to the conjugate prior property of the Dirichlet distribution, the posterior parameter update has a tractable form and can be computed analytically. Side constraints are represented by a system of linear inequalities $A \cdot \delta - b \leq 0$, where $A \in \mathbb{R}^{K \times N}$, $b \in \mathbb{R}^K$, and $0 \in \mathbb{R}^K$ is the K -dimensional vector of zeros. K is defined as the total number of constraints. The comparison is performed elementwise.

In UBayFS, a relaxed inadmissibility function $\kappa_{k,\rho}(\delta)$ is used as a penalization for the violation

of a given side constraint $k = 1, \dots, K$. The joint inadmissibility function κ pursues the idea that $\kappa = 1$ (maximum penalization) if at least one $\kappa_{k,\rho} = 1$, while $\kappa = 0$ (no penalization) if all $\kappa_{k,\rho} = 0$. A more detailed description is provided in the original paper (Jenul et al., 2022).

To obtain an optimal feature set δ^* , we use a target function $U(\delta, \theta)$ which represents a posterior expected utility of feature sets δ given the posterior feature importance parameter θ , regularized by the inadmissibility function $\kappa(\cdot)$

$$\delta^* = \max_{\delta \in \{0,1\}^N} (\mathbb{E}_{\theta|y}[U(\delta, \theta(y))]) = \max_{\delta \in \{0,1\}^N} (\delta^T \mathbb{E}_{\theta|y}[\theta(y)] - \lambda \kappa(\delta)).$$

The optimization is implemented via a genetic algorithm along with a greedy algorithm for initialization, suggested by Jenul et al. (2022) to find a proper start vector for the optimization.

Package Summary

The function `build.UBaymodel()` initializes an S3 class object `UBaymodel` and computes the ensemble of elementary feature selectors. In the current version, linear feature selectors such as Fisher score, Laplacian score (You & Shung, 2022), and mRMR (Jay et al., 2013) are supported as integrated options. Any arbitrary feature selector can be defined manually and used as input. In addition, the number of elementary models M is specified. The user can directly set prior weights inside the build function. Constraints are either provided as a matrix A and a right side b , or built using the `buildConstraints()` function, which supports `max-size`, `must-link`, and `cannot-link` constraints on both feature and block level. `UBayFS` requires at least one constraint limiting the total number of features to be selected (“max-size”). The level of constraint-relaxation is steered with an input parameter ρ . In addition, the weights for single features or feature blocks are set with `setWeights()`.

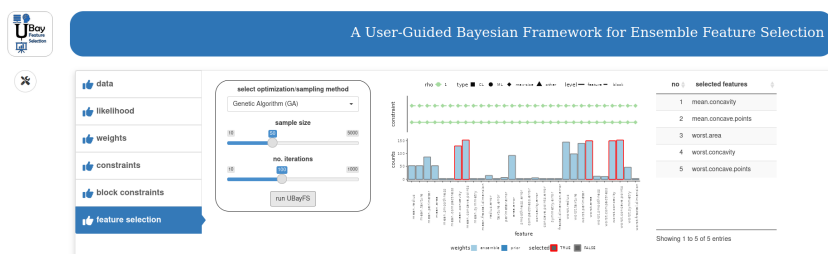
The function `admissibility()` allows the user to evaluate the penalty term for a given feature set under a set of constraints. After initializing the model and computing the ensembles, the `train.UBaymodel()` function optimizes the feature set via a genetic algorithm (Scrucca, 2013) with greedy initialization. According to empirical evaluations, the greedy initialization decreases the runtime and leads to faster convergence towards an optimal feature set. Finally, the package implements the generic functions `print.UBaymodel()`, `plot.UBaymodel()`, and `summary.UBaymodel()` as well as an evaluation function `evaluateFS()` to report and visualize results. Two vignettes guide the user through the package and demonstrate how the method can be deployed in common application scenarios, including how user knowledge is specified and how feature- and block-wise constraints are set.

Interactive Shiny Dashboard

The function `runInteractive()` opens an interactive Shiny dashboard allowing the user to load and analyze data interactively. However, due to computational limitations, it is not recommended to use the HTML interface for larger datasets (> 100 features or > 1000 samples). Instead, functions should be called from the R console in such cases. Figure 2 shows the dashboard with the different tabs:

- **data:** Load the dataset and specify whether row names, column names, or a block structure is present. A demo dataset is ready to be loaded and used for a first touch on the package.
- **likelihood:** Select elementary feature selectors for ensemble feature selection, the number of models M , the number of features in each model, and the ratio of the train-test split. Further, the dashboard allows the user to mix different elementary feature selectors, although this option is not recommended due to limited stability (Seijo-Pardo et al., 2017).

- **weights:** The prior feature weights are set by the user. For block feature selection, it is possible to set weights for blocks; otherwise, for a single feature.
- **constraints / block constraints:** In this task, the user sets different constraints (at least a max-size constraint). The penalty ρ can be varied here as well.
- **feature selection:** In the dashboard's last step, an optimization procedure determines the final feature set. A plot of the final result is produced - also, the model can be saved as an Rdata file and loaded to the dashboard again.



Adina Jenul < adina.jenul@ntnu.no >
Stefan Schrunner < stefan.schranner@ntnu.no >



Figure 2: Illustration of the Shiny HTML dashboard.

Ongoing research

Based on the present UBayFS package, ongoing work focuses on the implementation of even more types of expert constraints and elementary feature selection models. Moreover, a Python package with similar functionality is planned for the future.

Acknowledgements

We would like to thank Prof. Jürgen Pilz (University of Klagenfurt) and Prof. Oliver Tomic (Norwegian University of Life Sciences), who contributed to the development of the methodology and supported us with ideas and fruitful discussions, as well as Kristian Hovde Liland (Norwegian University of Life Sciences) for testing the R package.

References

- Jay, N. D., Papillon-Cavanagh, S., Olsen, C., El-Hachem, N., Bontempi, G., & Haibe-Kains, B. (2013). mRMRe: An R package for parallelized mRMR ensemble feature selection. *Bioinformatics*, 29(18), 2365–2368. <https://doi.org/10.1093/bioinformatics/btt383>
- Jenul, A., Schrunner, S., Pilz, J., & Tomic, O. (2022). A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). *Machine Learning*, 111(10), 3897–3923. <https://doi.org/10.1007/s10994-022-06221-9>
- Kuhn, M. (2022). *Caret: Classification and regression training*. <https://CRAN.R-project.org/package=caret>

- Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *Journal of Statistical Software*, 36(11). <https://doi.org/10.18637/jss.v036.i11>
- Majumdar, S. G., Rai, A., & Mishra, D. C. (2019). *GSelection: Genomic selection*. <https://CRAN.R-project.org/package=GSelection>
- R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53(4). <https://doi.org/10.18637/jss.v053.i04>
- Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118, 124–139. <https://doi.org/10.1016/j.knosys.2016.11.017>
- You, K., & Shung, D. (2022). Rdimtools: An R package for dimension reduction and intrinsic dimension estimation. *Software Impacts*, 14, 100414. <https://doi.org/10.1016/j.simpa.2022.100414>



Appendix C

Paper III

| | |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| title: | Ranking Feature-Block Importance in Artificial Multiblock Neural Networks |
| authors: | <u>Anna Jenul</u> , Stefan Schrunner, Bao Ngoc Huynh, Runar Helin, Cecilia Marie Futsæther, Kristian Hovde Liland, Oliver Tomic |
| date: | 09/2022 |
| publication: | International Conference on Artificial Neural Networks |
| doi: | https://doi.org/10.1007/978-3-031-15937-4_14 |



Ranking Feature-Block Importance in Artificial Multiblock Neural Networks

Anna Jenul^(), Stefan Schrunner^(), Bao Ngoc Huynh^(), Runar Helin^(),
Cecilia Marie Futsaether^(), Kristian Hovde Liland^(), and Oliver Tomic^()

Norwegian University of Life Sciences, Universitetstunet 3, 1432 Ås, Norway
{anna.jenul,stefan.schranner,ngoc.huynh.bao,runar.helin,
cecilia.futsaether,kristian.liland,oliver.tomic}@nmbu.no

Abstract. In artificial neural networks, understanding the contributions of input features on the prediction fosters model explainability and delivers relevant information about the dataset. While typical setups for feature importance ranking assess input features individually, in this study, we go one step further and rank the importance of groups of features, denoted as feature-blocks. A feature-block can contain features of a specific type or features derived from a particular source, which are presented to the neural network in separate input branches (multiblock ANNs). This work presents three methods pursuing distinct strategies to rank feature-blocks in multiblock ANNs by their importance: (1) a composite strategy building on individual feature importance rankings, (2) a knock-in, and (3) a knock-out strategy. While the composite strategy builds on state-of-the-art feature importance rankings, knock-in and knock-out strategies evaluate the block as a whole via a mutual information criterion. Our experiments consist of a simulation study validating all three approaches, followed by a case study on two distinct real-world datasets to compare the strategies. We conclude that each strategy has its merits for specific application scenarios.

Keywords: Feature-block importance · Importance ranking · Multiblock neural network · Explainability · Mutual information

1 Introduction

In machine learning, datasets with an intrinsic block-wise input structure are common; blocks may represent distinct data sources or features of different types and are frequently present in datasets from industry [7], biology [3], or healthcare [5]. For example, in healthcare, heterogeneous data blocks like patient histology, genetics, clinical data, and image data are combined in outcome prediction models. However, good prediction models do not necessarily depend equally on each block. Instead, some blocks may be redundant or non-informative. Identifying the key data sources in multi-source treatment outcome models promises to deliver new insights into the behavior of black-box models like ANNs. In particular, potential benefits include improving model explainability, reducing costly

data acquisitions that do not contribute to the model prediction, and allowing domain experts to explore latent relations in the data. Thus, there is a need to measure the importances of feature-blocks, denoted as feature-block importance ranking (BIR).

In order to exploit the internal structure of the block-wise data in neural networks, a multiblock ANN (M-ANN) architecture is used. As depicted in Fig. 1, the M-ANN consists of a separate input branch for each block, a concatenation layer to merge information from all branches, and a blender network to map the information to the model output. The architecture allows for any type of network layer, depth, activation, or other network parameters, including the special case where the concatenation layer equals the input layer (block branches of depth 0).

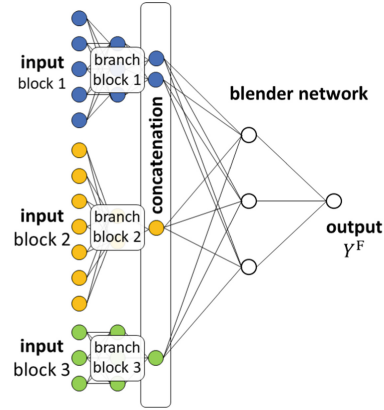


Fig. 1. M-ANN architecture.

Ranking individual features by their importances (feature importance ranking, FIR) has been studied for different types of ANNs [8, 14, 15]. An extensive evaluation [9] showed that versions of the variational gradient method (VarGrad) [1, 2] outperformed competitors such as guided backprop and integrated gradients. For BIR, however, a combination of features in one block may accumulate a larger amount of information than each feature separately due to informative non-linear relations between features. Hence, using FIR might oversimplify the problem of measuring block importance since interactions between features of the same block are disregarded. Nevertheless, the strategy of reducing BIR to a simple summary metric (sum, mean, max) over FIR scores is considered in our evaluation.

A related problem to FIR is feature selection, where the input dimensionality is reduced to the most influential features as part of the preprocessing. Feature selection is widely studied in ANNs. Furthermore, specialized feature selectors can account for block structures like UBayFS [11] or groupLasso [16]. Conceptually, these feature selectors aim to improve model performance and classify an entire block as important/unimportant in a binary way before model training. In contrast, our BIR problem is considered a post-processing procedure, focusing on analyzing the model after training without influence on the model performance.¹

This study presents and discusses three distinct approaches to quantify the importance of feature-blocks in M-ANNs. While exploiting the flexibility of ANNs and their capacities to learn complex underlying patterns in the data, the

¹ BIR may be used for block feature selection if deployed as filter method—however, this aspect is beyond the scope of the present work.

discussed methods aim to deliver insights into the trained network’s dependence structure on the distinct input blocks and thereby foster model explainability. We propose three paradigms for BIR: a block is considered as important if

1. it consists of features with high FIR scores (composition strategy), or
2. it explains a large part of the network output (knock-in strategy), or
3. its removal significantly changes the network output (knock-out strategy).

We evaluate and discuss the proposed paradigms in a simulation study and present two case studies on real-world datasets, where the behaviors of the proposed ranking strategies become apparent.

In the following, bold letters denote vectors and matrices; non-bold letters denote scalars, functions or sets. Square brackets denote index sets $[n] = \{1, \dots, n\}$.

2 Block Importance Ranking Methods

We assume data input \mathbf{x} from some feature space $D \subset \mathbb{R}^N$, $N \in \mathbb{N}$, following a probability distribution $\mathbf{X} \sim P_X$, and a univariate target variable $y \in T \subset \mathbb{R}$ following a probability distribution $Y \sim P_Y$. Given training data $(\mathbf{x}, y) \in D_{\text{train}} \times T_{\text{train}} \subset D \times T$, model parameters $\mathbf{w} \in W \subset \mathbb{R}^M$, $M \in \mathbb{N}$, are trained with respect to some loss term $e : D \times T \rightarrow \mathbb{R}^+$,

$$\mathbf{w}^* = \min_{\mathbf{w} \in W} e(f_{\mathbf{w}}(\mathbf{x}), y),$$

where the ANN is a function $f_{\mathbf{w}} : D \rightarrow T$ given weights \mathbf{w} .

In an M-ANN architecture, see Fig. 1, the block structure of the model input is represented by a direct sum of subspaces $D = \bigoplus_{b=1}^B D_b$, each corresponding to one block $b \in [B]$ with dimension $N_b = \dim(D_b)$, $N = \sum_{b=1}^B N_b$. Each block enters a distinct branch of the network that processes the block input. Afterwards, the outputs of all branches are merged in a concatenation layer, which consists of n_b nodes associated with each block b , respectively. A so-called blender network $f_{\mathbf{w}}^{\text{blender}}$ connects the concatenation layer to the network output. Network training is performed using backpropagation, where all block branches and the blender network are trained simultaneously in an end-to-end manner.

2.1 Composite Strategy

Our first paradigm composes block importance measures from FIR in a direct way. As a prototype of state-of-the-art FIR methods, we use VarGrad [2]. VarGrad builds on the idea that variations of an important feature provoke measurable variations in the output. Under the assumption that features are on a common scale, we estimate the gradient of the function $f_{\mathbf{w}}$ with respect to each feature by adding small random perturbations in the input layer. A large

variance in the gradient indicates that the network output depends strongly on a feature, i.e., the feature is important. We denote the importance of feature $n \in [N]$ as quantified by VarGrad, by $\alpha_n \in \mathbb{R}^+$.

To translate the feature-wise importance measure to feature-blocks in M-ANNs, we deploy a summary metric φ over all single-feature importances in a block $b \in [B]$. Thus, block importances $\gamma_\varphi^{(b)}$ are defined as

$$\gamma_\varphi^{(b)} = \varphi(\alpha_1^{(b)}, \dots, \alpha_{N_b}^{(b)}), \quad (1)$$

where $\alpha_n^{(b)}$ denotes the n^{th} feature associated with the b^{th} block. Intuitive choices for φ are either the sum, mean, or maximum operator, denoted as φ_{sum} , φ_{mean} , or φ_{max} , respectively. Rankings based on mean and sum are equal, if all blocks contain the same number of features. Operators φ_{sum} and φ_{mean} accumulate the individual feature importances: a block with multiple features of high average importances is preferred over blocks with few top features and numerous unimportant features. In contrast, φ_{max} compares the top-performing features out of each block, while neglecting all other's contributions. Statistical properties of block importance quantifiers implementing the composite strategy are transmitted from (i) the feature importance ranking method and (ii) the summary metric. Since this approach cannot capture between-feature relations, potentially impacting the importance of a block, we suggest two other paradigms.

2.2 Knock-In Strategy

The knock-in strategy is inspired by work on the information bottleneck [4], demonstrating that node activations can be exploited for model interpretation in ANNs. In the concatenation layer of the M-ANN (Fig. 1), where information from the blocks enters the blender network, activations are of particular importance since they represent an encoding of the block information. When passing model input \mathbf{x} through the network, we denote the activation of the n^{th} node associated with block $b \in [B]$ in the concatenation layer by $c_{b,n}(\mathbf{x})$, $n \in [n_b]$. The average activation of the n^{th} node in block $b \in [B]$ across all training data $\mathbf{x} \in D_{\text{train}}$ is denoted by $\bar{c}_{b,n}$.

For BIR, we compute a pseudo-output by passing data of only one block b through the network. For this purpose, we introduce a pseudo-input $\mathbf{v}^{(b)}(\mathbf{x})$ as

$$\mathbf{v}_{b',n}^{(b)}(\mathbf{x}) = \begin{cases} c_{b',n}(\mathbf{x}) & \text{if } b' = b \\ \bar{c}_{b',n} & \text{otherwise,} \end{cases} \quad (2)$$

where $b' \in [B]$, and $n \in [n_b]$. By propagating pseudo-input $\mathbf{v}^{(b)}(\mathbf{x})$ through the blender network, we obtain the pseudo-output $f_w^{\text{blender}}(\mathbf{v}^{(b)}(\mathbf{x}))$. The main assumption behind the knock-in strategy is that high agreement between output $f_w(\mathbf{x})$ and pseudo-output $f_w^{\text{blender}}(\mathbf{v}^{(b)}(\mathbf{x}))$ indicates a high importance of block b , since information from b is sufficient to recover most of the model output. In contrast, a large discrepancy between the two quantities indicates low explanatory power of the block b , and thus, a lower block importance. The concept to generate knock-in pseudo-outputs is illustrated in Fig. 2.

We implement the knock-in concept via the mutual information (MI) [6], an information-theoretic measure to quantify the level of joint information between two discrete random variables Z and Z' , defined as

$$\text{MI}(Z, Z') = \sum_z \sum_{z'} p_{Z, Z'}(z, z') \log_2 \left(\frac{p_{Z, Z'}(z, z')}{p_Z(z)p_{Z'}(z')} \right).$$

If Z and Z' are independent, $\text{MI}(Z, Z')$ is 0. Otherwise, $\text{MI}(Z, Z')$ is positive, where a high value indicates a large overlap in information. To quantify the joint and marginal distributions of continuous variables Z and Z' , two-dimensional and one-dimensional histograms can be used as non-parametric estimators for $p_{Z, Z'}$, p_Z , and $p_{Z'}$, respectively. We denote the number of equidistant histogram bins along each axis by $\ell \in \mathbb{N}$. It follows from the properties of entropy [6] that an upper bound to $\text{MI}(Z, Z')$ is given by $\log_2(\ell)$.

As shown in Fig. 2, the random variable of (full) model output, $Y^{\text{F}} = f_w(\mathbf{X})$, and the random variable of the pseudo-output with respect to block b , $Y^{(b)} = f_w^{\text{blender}}(\mathbf{v}^{(b)}(\mathbf{X}))$, where \mathbf{X} follows the input distribution $P_{\mathbf{X}}$, are used to measure knock-in (KI) block importance as

$$\gamma_{\text{KI}}^{(b)} = \frac{\text{MI}(Y^{\text{F}}, Y^{(b)})}{\log_2(\ell)}. \quad (3)$$

2.3 Knock-Out Strategy

The knock-out paradigm is an ablation procedure where one block at a time is removed from the model in order to measure the impact of the remaining blocks. We pursue a similar approach as in the knock-in paradigm and specify knock-out pseudo-inputs $\mathbf{v}^{(-b)}(\mathbf{x})$ as

$$\mathbf{v}_{b',n}^{(-b)}(\mathbf{x}) = \begin{cases} \bar{c}_{b',n} & \text{if } b' = b \\ c_{b',n}(\mathbf{x}) & \text{otherwise,} \end{cases} \quad (4)$$

for an arbitrary block $b \in [B]$. Thus, the definition in Eq. 4 represents an opposite behavior of Eq. 2 in the knock-in case. In analogy to the knock-in notation, we denote the random variable of pseudo-outputs with respect to $\mathbf{v}^{(-b)}$ as $Y^{(-b)} = f_w^{\text{blender}}(\mathbf{v}^{(-b)}(\mathbf{X}))$. The knock-out concept is illustrated in Fig. 3. In contrast to knock-in, we assume that leaving out block b having a relevant impact on the final output delivers a more dissimilar pseudo-output to the full output since relevant information is lost. Removing an unimportant block preserves the relevant information and delivers a pseudo-output similar to the full output.

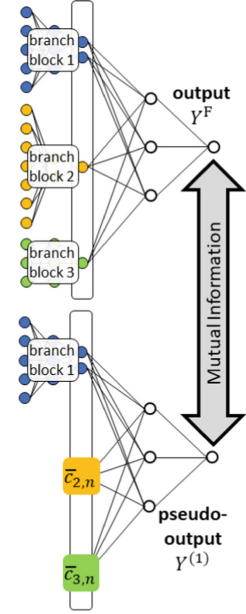


Fig. 2. Knock-in strategy: Pseudo-outputs for feature-block $b = 1$ are generated by activating block b , while imputing averaged activations for all other blocks.

Finally, we define the importance of block $b \in [B]$ with respect to the knock-out strategy (KO) as

$$\gamma_{\text{KO}}^{(b)} = \frac{\log_2(\ell) - \text{MI}(Y^F, Y^{(-b)})}{\log_2(\ell)}. \quad (5)$$

For both, KI and KO, importance scores $\gamma_{\text{KI}}^{(b)}$ and $\gamma_{\text{KO}}^{(b)}$ are bounded between 0 (unimportant block) and 1 (important block).

3 Experiments

As a proof of concept, we conduct two experiments to assess BIR in M-ANNs. The first experiment involves six simulated, non-linear regression problems, where our simulation setup delivers information on the ground truth block importances. This experiment verifies that our suggested measures can identify the ground truth block rankings, defined by their corresponding paradigms. Real-world datasets are evaluated in two case studies in experiment 2, where no exact ground truth block ranking is available. Instead, we compare BIR strategies to each other.

3.1 Simulation Experiment

We simulate a synthetic datasets along with six distinct target functions, denoted as setups S1a–S1c and S2a–S2c. The dataset consists of $N = 256$ features, divided randomly into $B = 8$ blocks (B1–B8) à $N_b = 32$ features. The sample size is set to $|D_{\text{train}}| = 10\,000$ and $|D_{\text{test}}| = 10\,000$. All features are simulated from a multivariate normal distribution with mean vector $\boldsymbol{\mu} = \mathbf{0}$ and a randomized covariance matrix $\boldsymbol{\Sigma}$; hence a non-trivial correlation structure is imposed.²

Setups S1a–S1c and S2a–S2c differ in the parameters used to compute the non-linear target variable y , which is simulated via a noisy linear combination of the squared features with coefficient matrix $\boldsymbol{\beta}^{(b)} \in \mathbb{R}^{N_b \times N_b}$, given as

$$y = \underbrace{\sum_{b=1}^8 \mathbf{x}^T \boldsymbol{\beta}^{(b)} \mathbf{x}}_{g(\mathbf{x})} + \varepsilon_{\text{noise}}, \text{ where } \boldsymbol{\beta}^{(b)} = \begin{pmatrix} \beta_{\text{imp}} & 0 & \dots & 0 & | & 0 & \dots & 0 \\ \beta_{\text{int}} & \beta_{\text{imp}} & \dots & 0 & | & 0 & \dots & 0 \\ \dots & \dots & \ddots & \dots & | & \dots & \dots & \dots \\ \beta_{\text{int}} & \beta_{\text{int}} & \dots & \beta_{\text{imp}} & | & 0 & \dots & 0 \\ \hline 0 & 0 & \dots & 0 & | & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & | & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & | & 0 & \dots & 0 \end{pmatrix}. \quad (6)$$

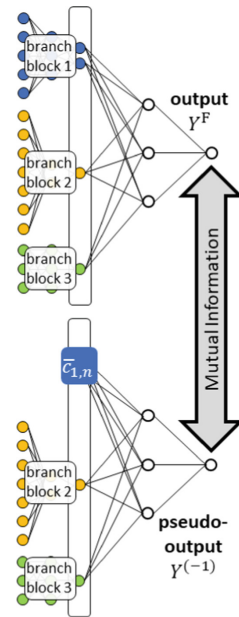


Fig. 3. Knock-out strategy: Pseudo-outputs are generated by activating all but one blocks.

² Code and details on simulation and network architecture are available at https://github.com/annajenul/Block_Importance_Quantification.

Table 1. Specifications for matrix $\beta^{(b)}$: block importance is steered via count N_{imp} , coefficient β_{imp} , and interaction β_{int} of the important features.

| Setup | Block | | | | | | | | | | | | |
|-------|------------------|----------------------|----------------------|------------------|----------------------|----------------------|-----|------------------|----------------------|----------------------|------------------|----------------------|----------------------|
| | B1 | | | B2 | | | ... | B7 | | | B8 | | |
| | N_{imp} | β_{imp} | β_{int} | N_{imp} | β_{imp} | β_{int} | ... | N_{imp} | β_{imp} | β_{int} | N_{imp} | β_{imp} | β_{int} |
| S1a | 2 | 7 | 0 | 2 | 6 | 0 | ... | 2 | 1 | 0 | 0 | 0 | 0 |
| S1b | 7 | 2 | 0 | 6 | 2 | 0 | ... | 1 | 2 | 0 | 0 | 0 | 0 |
| S1c | 1 | 7 | 0 | 2 | 6 | 0 | ... | 7 | 1 | 0 | 0 | 0 | 0 |
| S2a | 2 | 7 | 1 | 2 | 6 | 1 | ... | 2 | 1 | 1 | 0 | 0 | 1 |
| S2b | 7 | 2 | 1 | 6 | 2 | 1 | ... | 1 | 2 | 1 | 0 | 0 | 1 |
| S2c | 1 | 7 | 1 | 2 | 6 | 1 | ... | 7 | 1 | 1 | 0 | 0 | 1 |

The matrix $\beta^{(b)} \in \mathbb{R}^{N_b \times N_b}$ contains an $N_{\text{imp}} \times N_{\text{imp}}$ quadratic sub-matrix consisting of coefficients β_{imp} of important features, i.e. features with relevant contribution to the target, and interactions β_{int} . The noise parameter σ_{noise} is set to 10% of the standard deviation of the linear combination $g(\mathbf{x})$ across the generated samples \mathbf{x} . As shown in Table 1, block importances are varied between the setups and as follows

- S1a: varying coefficients of important features, but constant counts;
- S1b: varying counts of important features, but constant coefficients;
- S1c: varying counts and coefficients of important features;
- S2a–S2c: same as S1a–S1c, but with interaction terms between features.

Due to the randomized correlation matrix of the feature generation, unimportant features may be correlated with important features, as well as with the target y .

For each setup, we trained the described M-ANN model in 30 independent runs with distinct weight initializations after data standardization. Since BIR methods are deployed post-hoc and assume a model with appropriate performance, runs with poor performances ($R2 < 0.8$) were excluded from the analysis after outlier removal. Hence, the number of model runs in the analysis was 20 (S1a, S1b, S2a, S2b), 18 (S1c), and 19 (S2c), respectively. The remaining models achieved an average performance of ≥ 0.9 (R2 score) and ≤ 0.2 (RMSEIQR: root mean squared error scaled by inter-quartile range) on the test set.

For evaluation, importance scores across all model runs were tested for significant differences using a pairwise Wilcoxon-test with Bonferroni correction. If the p-value in a comparison between two blocks was above a significance level of 0.01, both were counted as tie. Figure 4 illustrates the distributions of BIR scores after min-max-normalization by setup and method, along with rankings (colors) based on significant group differences. All methods discovered the intrinsic ranking in dataset S1a. In dataset S1b, knock-in, knock-out, and VarGrad-mean identified the ranking by underlying important feature counts N_{imp} , while VarGrad-max failed to deliver a significant distinction between blocks with higher counts of

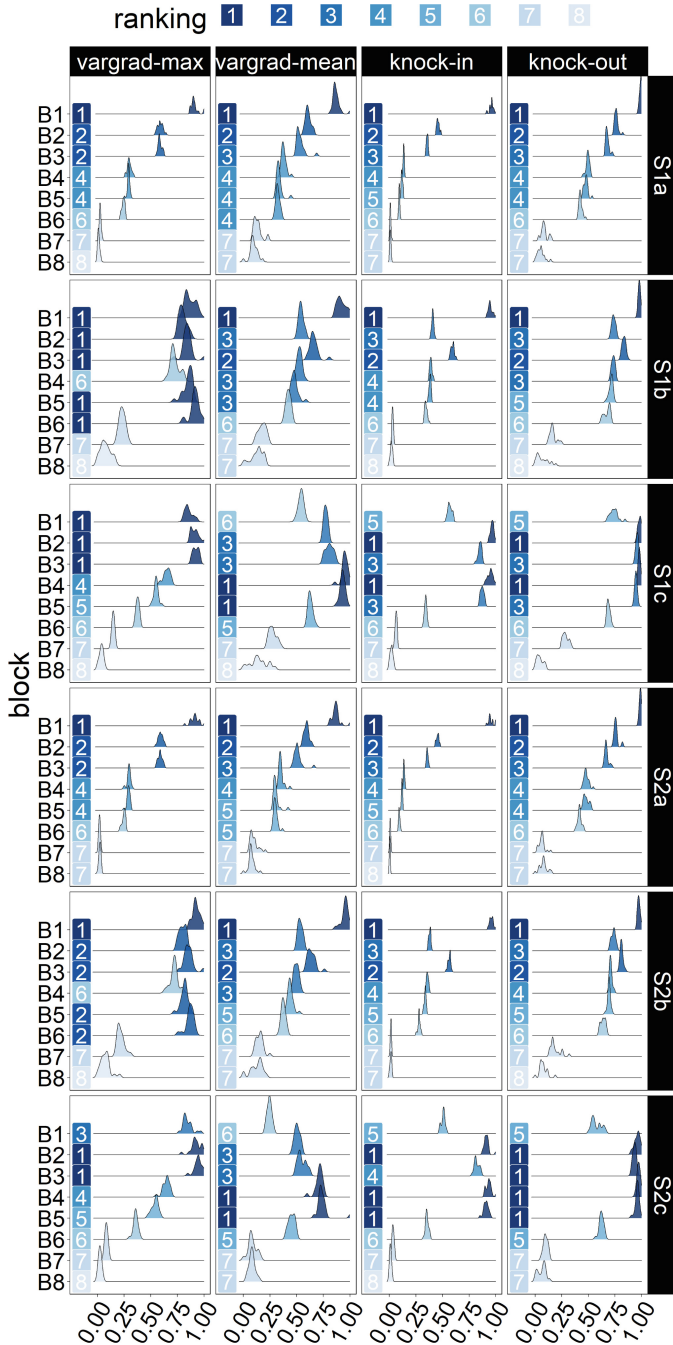


Fig. 4. Distributions of the normalized BIR scores across model runs. Rankings are indicated by colors and refer to significant group differences based on a pairwise Wilcoxon-test (significance level 0.01).

Table 2. Averaged Spearman’s rank correlation coefficients comparing each ranking to the ground truth BIR for each paradigm across model runs. Standard deviations were ≤ 0.03 for S1a, S1b, S2a and S2b, and ≤ 0.06 for S1c and S2c.

| Paradigm | Dataset | | | | | |
|--------------------------|---------|------|------|------|------|-------|
| | S1a | S1b | S1c | S2a | S2b | S2c |
| Composite (VarGrad-max) | 0.97 | 0.58 | 0.93 | 0.98 | 0.58 | 0.91 |
| Composite (VarGrad-mean) | 0.98 | 0.95 | 0.96 | 0.97 | 0.95 | -0.40 |
| Knock-in | 0.99 | 0.98 | 0.85 | 0.97 | 0.99 | 0.89 |
| Knock-out | 0.99 | 0.98 | 0.81 | 0.99 | 0.99 | 0.89 |

important features. For dataset S1c, VarGrad-max mostly ranked by underlying β_{imp} and ignored N_{imp} , while knock-in, knock-out and VarGrad-mean delivered trade-offs between counts N_{imp} and coefficients β_{imp} of important features. In setups S2a, S2b, and S2c with between-feature interactions, the same rankings as in S1a, S1b, and S1c could be obtained by all methods with negligible deviations. Hence, we conclude that all metrics remain stable in more complex scenarios.

We further validated the paradigms by comparing the results to their corresponding ground truth block importances, determined by the real coefficients in the simulation setup. For the composite max and mean paradigms, the corresponding maxima and means over $\beta^{(b)}$, were used as references. Ground truth importances for knock-in (KI), and knock-out (KO) were based on the explained variances of the single block b in the underlying linear combination, given as

$$KI_b = \mathbb{E} \left(y - \left(\mathbf{x}^{(b)} \right)^T \boldsymbol{\beta}^{(b)} \mathbf{x}^{(b)} \right), \text{ and } KO_b = \mathbb{E} \left(y - \sum_{\substack{b'=1 \\ b' \neq b}}^8 \left(\mathbf{x}^{(b')} \right)^T \boldsymbol{\beta}^{(b')} \mathbf{x}^{(b')} \right),$$

where $\mathbf{x}^{(b)}$ denotes projection of input \mathbf{x} on the subspace of block b , D_b . The comparison between the rankings based on (average) predicted importance scores and ground truth rankings was made using Spearman’s correlation coefficient, see Table 2. With two exceptions, all correlation values were at a high level, indicating that our methods accurately predicted the ground truth. Spearman’s correlation coefficient is not representative in S1b, and S2b with respect to the maximum metric since the ground truth ranking is equal for blocks B1–B7. In S2c VarGrad-mean is distracted by decreasing β_{imp} and an increasing number of interaction terms, although underlying block importances are in increasing order with respect to the mean metric.

3.2 Real-World Experiment

Since verification on simulated data showed that the presented approaches match the ground truth according to their paradigms, we deployed the methods on two

real-world datasets, where underlying block importance is unknown. Prior to analysis, both datasets were standardized on the trained data. Again, we trained 30 independent model runs.

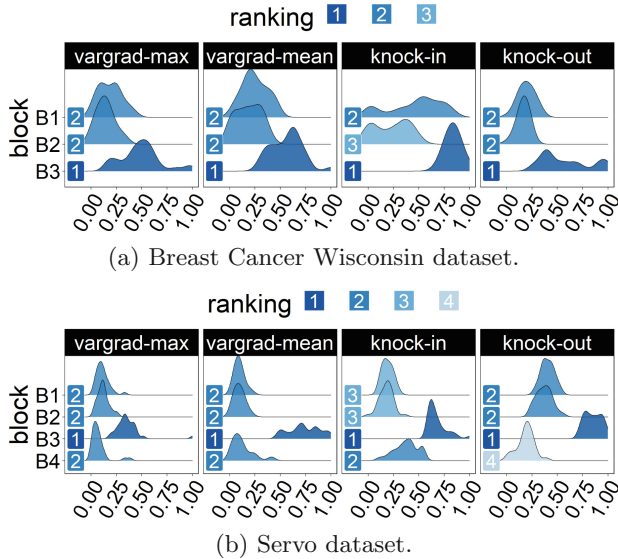


Fig. 5. Distributions of normalized BIR scores in experiment 2 across model runs.

The Breast Cancer Wisconsin dataset (BCW) [13] describes a binary classification problem (malignant or benign tumor) and consists of 569 samples (398 train, 171 test) and three blocks with ten features each, representing groups of distinct feature characteristics (mean values, standard deviations, and extreme values of measured parameters). The average performance was 0.95 (accuracy) and 0.96 (F1 score) without outliers. The average scores and rankings delivered by BIR methods are shown in Fig. 5a. All four paradigms discovered that block 3 is dominant, which agrees with previous research on the dataset [10]. However, knock-in was the only method that distinguished between the importances of B1 and B2. According to [10], block B1 contains overlapping information with B3, while B2 is rather non-informative. Thus, the experiment underlines a difference between knock-in and knock-out rankings in the presence of redundancies.

Servo [12] is a dataset containing 167 samples (120 train, 47 test), a univariate, numeric target variable, and four features, two of which are categorical variables with four levels each, and two are numerical variables. Each feature was assigned its own block. One-hot encoding was performed for the two blocks containing categorical features, leading to two blocks (B1 and B2) of four binary features, each. Blocks corresponding to numerical features (B3 and B4) contain one feature each. In the 30 M-ANN model runs, an average performance of 0.21 (RMSEIQR) and 0.87 (R2) was obtained without outliers. Figure 5b shows that

for all four paradigms, block B3 was most important. While VarGrad methods delivered a binary ranking, knock-in and knock-out suggested a ranking with 3 and 4 distinct importance levels, respectively—thus, the level of detail was higher in the MI-based rankings compared to VarGrad methods.

4 Discussion

Our experiments demonstrated several differences between the proposed strategies. While the composite strategy evaluates features individually and depends on two user-selected parameters (the feature-wise ranking scheme and the summary metric), the knock-in and knock-out strategies consider each block a closed unit. They require no selection of a summary statistic. MI-based rankings deliver a score in $[0, 1]$, while VarGrad has no upper bound. However, the discretization associated with the mutual information calculation may influence the importance scores and, thus, the rankings by knock-in and knock-out. All strategies are applicable for multivariate target variables, as well. However, an MI-based comparison between outputs and pseudo-outputs is prone to suffer from the curse of dimensionality since higher-dimensional probability distributions are compared to each other. On the contrary, the vanishing gradient problem can influence VarGrad in deep ANN architectures. All approaches delivered accurate experimental results, but only knock-in and knock-out provided a consistent ranking of blocks with minor importance in dominant blocks, such as for the servo dataset.

Even though knock-in and knock-out rely on the same concept of assessing pseudo-outputs related to each block, their properties and interpretations differ. The knock-in strategy determines whether a block can deliver a reasonable target description independently from the remaining blocks. This interpretation of block importance evaluates the performance achieved if we reduce the model to solely one block at a time. In contrast, knock-out quantifies whether the contribution of a block can be compensated by any other block. If two blocks contain redundant information about the target, knock-in delivers high values for both blocks since each block individually has high explanatory power. In contrast, knock-out penalizes redundant blocks since each of them can be removed without loss of information. This property became evident in the BCW experiment, where B3 was dominant but shared overlapping information with B1: knock-in was the only approach that discovered the higher information content in B1 compared to the uninformative B2.

5 Conclusion

We have demonstrated three strategies to rank the importance of feature-blocks as post-processing in ANNs with block-wise input structure. The composite strategy, which is a direct generalization of feature-wise importance rankings, provided promising results in most cases, but selecting the correct summary statistic was crucial. Knock-in and knock-out strategies, implemented using an information-theoretic measure on the model outputs, delivered a trade-off

between the extremes of maximum and mean feature importance in the composite case. All methods uncovered the true block importance with high accuracy and delivered new insights into the ANN's behavior. Still, computing multiple proposed metrics is advantageous for making informative block ranking decisions.

Acknowledgment. The authors gratefully acknowledge the financial support from internal funding scheme at Norwegian University of Life Sciences (project number 1211130114), which financed the international stay at the University of British Columbia, and thereby fostered the completion of this work.

References

1. Adebayo, J., Gilmer, J., Goodfellow, I., Kim, B.: Local explanation methods for deep neural networks lack sensitivity to parameter values. arXiv (2018)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
3. Alnemer, L.M., et al.: Multiple sources classification of gene position on chromosomes using statistical significance of individual classification results. In: International Conference on Machine Learning and Applications and Workshops, vol. 1, pp. 7–12 (2011). <https://doi.org/10.1109/ICMLA.2011.101>
4. Amjad, R.A., Geiger, B.C.: Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(9), 2225–2239 (2019). <https://doi.org/10.1109/TPAMI.2019.2909031>
5. Cao, B., He, L., Kong, X., Philip, S.Y., Hao, Z., Ragin, A.B.: Tensor-based multi-view feature selection with applications to brain diseases. In: IEEE International Conference on Data Mining, pp. 40–49 (2014)
6. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, Hoboken (2012)
7. Dagnely, P., Tourwé, T., Tsiorkova, E.: Annotating the performance of industrial assets via relevancy estimation of event logs. In: IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1261–1268 (2018). <https://doi.org/10.1109/ICMLA.2018.00205>
8. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: AAAI Conference on Artificial Intelligence, vol. 33, pp. 3681–3688 (2019)
9. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
10. Jenul, A., Schrunner, S., Liland, K.H., Indahl, U.G., Futsæther, C.M., Tomic, O.: Rent-repeated elastic net technique for feature selection. *IEEE Access* **9**, 152333–152346 (2021)
11. Jenul, A., Schrunner, S., Pilz, J., Tomic, O.: A User-Guided Bayesian Framework for Ensemble Feature Selection in Life Science Applications (UBayFS). arXiv (2021)
12. Quinlan, J.R.: Combining instance-based and model-based learning. In: International Conference on Machine Learning, pp. 236–243 (1993)

13. Street, W.N., Wolberg, W.H., Mangasarian, O.L.: Nuclear feature extraction for breast tumor diagnosis. In: Acharya, R.S., Goldgof, D.B. (eds.) *Biomedical Image Processing and Biomedical Visualization*, vol. 1905, pp. 861–870. SPIE (1993). <https://doi.org/10.1117/12.148698>
14. Wojtas, M., Chen, K.: Feature importance ranking for deep learning. *Adv. Neural. Inf. Process. Syst.* **33**, 5105–5114 (2020)
15. Yu, R., et al.: NISP: pruning networks borisusing neuron importance score propagation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9194–9203 (2018)
16. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **68**(1), 49–67 (2006)

Appendix D

Paper IV

title: Towards Understanding the Survival of Patients with High-Grade Gastroenteropancreatic Neuroendocrine Neoplasms: an Investigation of Ensemble Feature Selection in the Prediction of Overall Survival

authors: Anna Jenul, Henning Langen Stokmo, Stefan Schrunner, Mona-Elisabeth Revheim, Geir Olav Hjortland, Oliver Tomic

date: 02/2023

publication: arXiv.org (preprint)

doi: <https://doi.org/10.48550/arXiv.2302.10106>

TOWARDS UNDERSTANDING THE SURVIVAL OF PATIENTS WITH HIGH-GRADE GASTROENTEROPANCREATIC NEUROENDOCRINE NEOPLASMS: AN INVESTIGATION OF ENSEMBLE FEATURE SELECTION IN THE PREDICTION OF OVERALL SURVIVAL

A PREPRINT

• **Anna Jenul**

Department of Data Science
Norwegian University of Life Sciences
Ås, Norway
anna.jenul@nmbu.no

• **Henning Langen Stokmo***

Division of Radiology and Nuclear Medicine
Oslo University Hospital
Oslo, Norway
h.l.stokmo@studmed.uio.no

• **Stefan Schrunner**

Department of Data Science
Norwegian University of Life Sciences
Ås, Norway
stefan.schranner@nmbu.no

• **Mona-Elisabeth Revheim†**

Division of Radiology and Nuclear Medicine
Oslo University Hospital
Oslo, Norway
monar@ous-hf.no

• **Geir Olav Hjortland**

Department of Oncology
Oslo University Hospital
Oslo, Norway
goo@ous-hf.no

• **Oliver Tomic**

Department of Data Science
Norwegian University of Life Sciences
Ås, Norway
oliver.tomic@nmbu.no

February 21, 2023

ABSTRACT

Determining the most informative features for predicting the overall survival of patients diagnosed with high-grade gastroenteropancreatic neuroendocrine neoplasms is crucial to improve individual treatment plans for patients, as well as the biological understanding of the disease. Recently developed ensemble feature selectors like the Repeated Elastic Net Technique for Feature Selection (RENT) and the User-Guided Bayesian Framework for Feature Selection (UBayFS) allow the user to identify such features in datasets with low sample sizes. While RENT is purely data-driven, UBayFS is capable of integrating expert knowledge a priori in the feature selection process. In this work we compare both feature selectors on a dataset comprising of 63 patients and 134 features from multiple sources, including basic patient characteristics, baseline blood values, tumor histology, imaging, and treatment information. Our experiments involve data-driven and expert-driven setups, as well as combinations of both. We use findings from clinical literature as a source of expert knowledge. Our results demonstrate that both feature selectors allow accurate predictions, and that expert knowledge has a stabilizing effect on the feature set, while the impact on predictive performance is limited. The features *WHO Performance Status*, *Albumin*, *Platelets*, *Ki-67*, *Tumor Morphology*, *Total MTV*, *Total TLG*, and SUV_{\max} are the most stable and predictive features in our study.

*Henning Langen Stokmo is also affiliated with the Institute of Clinical Medicine, University of Oslo, Oslo, Norway

†Mona-Elisabeth Revheim is also affiliated with The Intervention Centre, Oslo University Hospital, Oslo, Norway, and the Institute of Clinical Medicine, University of Oslo, Oslo, Norway

1 Introduction

Gastroenteropancreatic (GEP) neuroendocrine neoplasms (NEN) are heterogeneous types of malignancies increasingly common over the last three decades [1, 2]. High-grade GEP NEN encompasses both neuroendocrine tumors grade 3 (NET G3) and neuroendocrine carcinomas (NEC), where NEC is further subdivided into small cell (SC) and large cell carcinomas (LC). According to the WHO 2019 Classification of Tumors: Digestive System Tumors, NET G3 are well differentiated (WD), whilst NEC are poorly differentiated (PD), both with a Ki-67 proliferation index (Ki-67) $> 20\%$ [3]. Although both NET G3 and NEC share features of immunohistochemical staining with chromogranin A and synaptophysin, they are considered morphologically different [4].

The prognosis for patients with advanced GEP NEC is poor, with a median survival of less than 12 months [5,6], whilst the prognosis for locoregional GEP NEC is higher; 20.7 months [7]. Numerous recently published studies [5, 8–14] have shown the prognostic importance of several parameters on overall survival (OS) such as age, performance status (PS), primary tumor site, tumor differentiation, TNM-stage, serum lactate dehydrogenase (LDH), serum platelet levels, proliferation marker Ki-67, maximum standardized uptake value (SUV_{max}), total metabolic tumor volume (tMTV) and total total lesion glycolysis (tTLG). Establishing more robust prognostic parameters and validating established parameters is essential to provide optimal care for this patient group.

Forecasting the OS of cancer patients as a major indicator of treatment success by machine learning models is of high relevance to offering optimal individual treatments for patients. In particular, accurate outcome prediction models pave the way for decision support in clinical practice. Since GEP NEN are rare, however, the data basis for training purely data-driven models is limited, leading to problems like overfitting, spurious correlations, and, consequently, to inaccurate predictions [15–17]. Two major approaches are at hand to overcome these issues: (a) increasing the number of samples (either by collecting more data or by artificial data augmentation) or (b) reducing the dimensionality of the feature space. In this work, we elaborate on approach (b), where our method of choice is feature selection. While general dimensionality reduction methods like Principal Component Analysis [18] transform the data to a new domain and thereby make identification of influencing factors difficult, feature selection reduces the dimension by subsetting the dataset by columns. As a result, a subset of the original features is retained, and the interpretability of the data columns is preserved.

Beyond the obvious benefit that predictive models become tractable, feature selection has the potential to improve the understanding of biological processes by clinical experts [19]. In particular, feature selectors point to input data parameters, which are related to explaining the target variable by a data-driven model. This information may either support or contradict existing hypotheses about the underlying biological processes or disclose previously unknown relations. The evaluation and interpretation of the findings require close collaboration between clinical experts and data scientists. However, such an application of feature selectors is still less common in machine learning, where the focus typically lies exclusively on optimizing performance metrics.

State-of-the-art research in feature selection with applications in healthcare, such as L1 regularization [20], decision trees [21], Laplace scores [22], or the minimum redundancy-maximum relevance (mRMR) criterion [23], are mainly data-driven and may suffer from well-known limitations. Among these limitations is the problem that minor changes, such as the inclusion of new or removal of old samples, may have significant effects on the set of selected features — the property of feature sets to remain invariant under such changes to the dataset is referred to as feature selection stability and investigated in [24]. The usage of ensemble feature selectors, which train multiple feature selectors on subsets of the samples in a dataset, has recently been investigated extensively [25] and achieves a higher feature selection stability compared to a single feature selection run, while retaining a similar predictive performance, as used e.g., in random forest methods [26]. More recently, this fact has been exploited to introduce more stable feature selection methods tailored for healthcare applications, which offer a large potential with respect to the aspects discussed above [19,27].

This paper aims to improve the understanding and insights into the OS in patients with high-grade GEP NEN by applying recently developed ensemble feature selection techniques. Specifically, we evaluate the Repeated Elastic Net Technique for Feature Selection (RENT) [27], as well as the User-Guided Bayesian Framework for Feature Selection (UBayFS) [19] on a dataset containing 63 patients diagnosed with high-grade GEP NEN. Our experiments compare both ensemble feature selectors in setups with and without the use of expert information. Our main goals are: (I) to determine the most informative set of features with respect to the outcome prediction task; (II) to interpret those selected features clinically — to evaluate the first goal, we measure the quality of the selected feature set in terms of predictive performance and selection stability. Another aspect of interest is: (III) to determine the effect of integrating prior expert knowledge into the feature selection process, compared to a purely data-driven pipeline. To this end, we discuss the feature selection results with respect to their clinical relevance and potential to improve our understanding of what influences OS of GEP NEN patients.

Notations In the following, we denote the input data matrix by $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m denotes the number of patients, and n denotes the number of features. Further, the target variable is denoted by $\mathbf{y} \in \mathbb{R}^m$. A feature set S is characterized by the indices, $S \subseteq \{1, \dots, n\}$. Vectors and matrices are indicated by bold letters.

2 Materials and Methods

2.1 GEP NEN dataset

Statements of Ethics This study was done in concordance with the Declaration of Helsinki. Approval from the regional committee for medical and health research ethics (2012/490, 2012/940, 2018/1940) and the local data protection officer was obtained. Informed consent was obtained from all patients at the time of inclusion but was waived for the patients in terminal phase and deceased.

Patient cohort Patients were identified from a single institutional cohort at Oslo University Hospital, also included in two multi-institutional Nordic NEC registries organized by the Nordic Neuroendocrine Tumor Group, previously described by [8]. In short, this cohort consisted of 192 patients included between January 2000 and July 2018, with GEP NEC classified according to the WHO 2010-classification [28]. In addition, all patients who had performed a fluorine-18 labeled 2-deoxy-2-fluoroglucose ([18F]FDG) positron emission tomography/computed tomography (PET/CT) within 90 days of their histological evaluation were eligible for inclusion. A hundred and seven patients did not have PET/CT performed, and two patients had no metabolic active lesions available for evaluation. Seventeen patients had more than 90 days between their biopsy and PET/CT, leaving 66 patients available for inclusion in this study.

Histological re-evaluation As described previously in [8], the histological re-evaluation was performed on both core biopsies and surgical specimens from GEP NEC primary tumors and metastases. These were re-classified according to the most recent WHO 2019-classification [3] with regards to synaptophysin, chromogranin A, and the proliferation marker Ki-67. In this study, only the re-evaluated histology features were used, while the original histology block was discarded.

PET/CT acquisition All PET/CT scans were done according to the European Association of Nuclear Medicine (EANM) guidelines [4, 5] as part of the clinical routine. The three PET scanners used were a 40-slice Siemens Biograph mCT hybrid PET/CT system (Siemens Healthineers, Erlangen, Germany), a Siemens Biograph 64, and a 64-slice General Electric (GE) Discovery 690 (GE Healthcare, Waukesha, WI, USA). Both Biograph PET/CTs were both EANM Research Ltd. (EARL)-accredited, whilst the Discovery 690 followed similar routine quality controls harmonizing with the two Biographs for cross-calibration. All acquisitions were from the vertex or skull base to mid-thighs. Before the PET acquisition, a low dose CT was acquired for anatomical information and attenuation correction. Parameters from PET were extracted using the ROI Visualisation, Evaluation, and Image Registration (ROVER) software v3.0.5 (ABX GmbH).³

Treatment All patients received treatment in the form of surgery, chemotherapy, or a combination of both. In total, 54 patients received the standard treatment of platinum-based chemotherapy. Patients could have surgery prior to or after [18F]FDG PET/CT. Evaluation of response to chemotherapy treatment was done with CT using the Response Evaluation Criteria in Solid Tumors (RECIST) [29].

Outcome variable Our outcome variable, or outcome target, was overall survival (OS) in months. This can be defined as the time a patient remains alive from the time of diagnosis to death of any cause; hence, it is not disease-specific. It is a reliable and easily available survival measure [30]. We can analyze such survival data, i.e., the time from diagnosis to the time of death, using the Kaplan-Meier estimator. For those patients who did not experience the event during the time of the study (or during follow-up) (i.e., death), they are said to be 'censored' [31]. Being 'censored' means that we do not know when this event will occur, only that it has not happened at the end of the study (or during follow-up). Across the full dataset, the empirical distribution of the outcome variable is illustrated as a histogram in Fig. 1.

³The detailed imaging- and extraction protocol is described in [8].

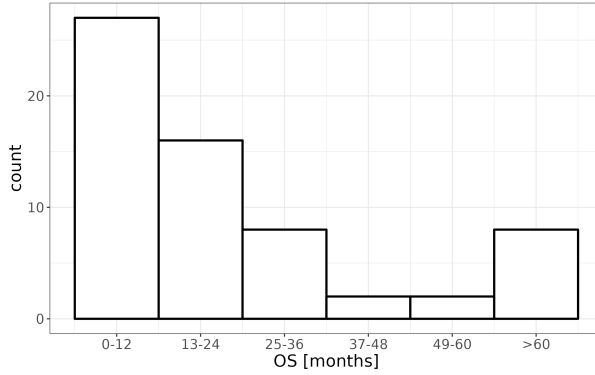


Figure 1: Distribution of the overall survival in months.

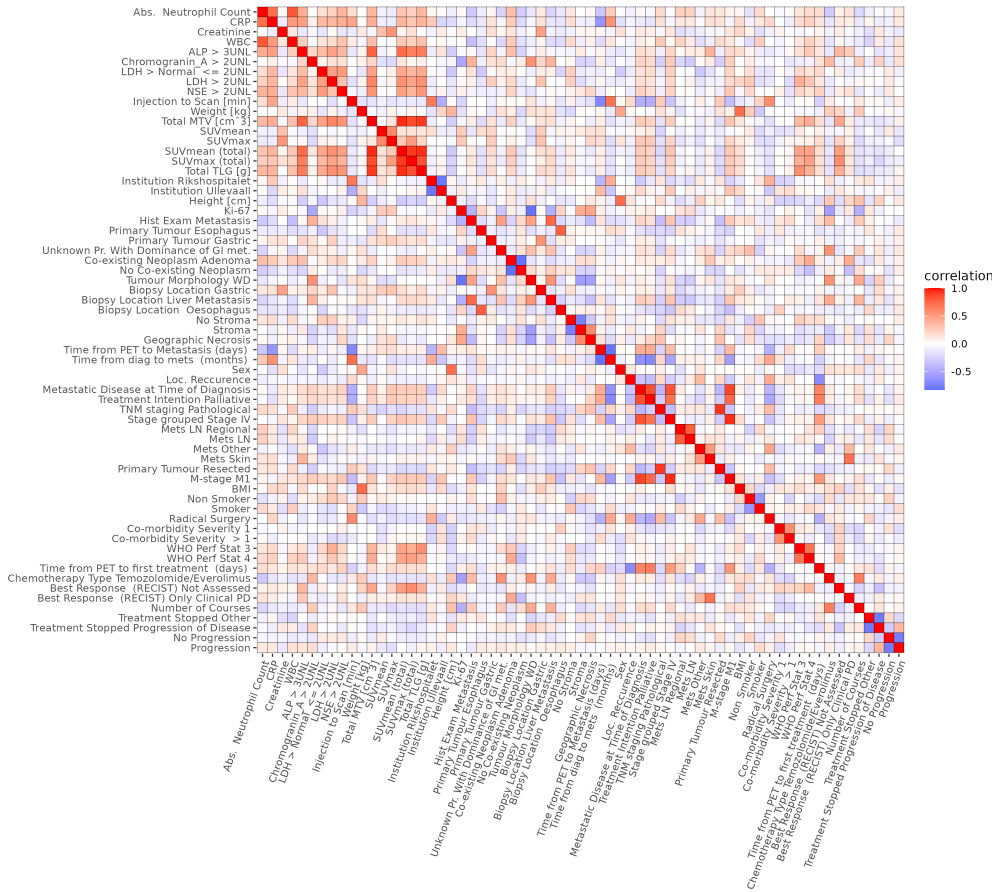


Figure 2: Correlations between input features (features with absolute correlations ≤ 0.5 were removed).

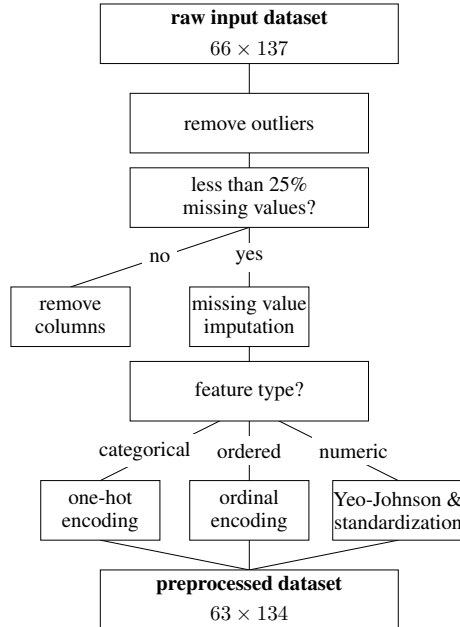


Figure 3: Preprocessing pipeline for the dataset.

Data blocks The data were grouped into five different blocks

- (p) patient characteristics
- (b) baseline blood values
- (h) re-evaluated histology
- (i) PET/CT imaging
- (t) treatment

The data contained mainly categorical and ordinal features with very few continuous variables. An overview of the pairwise correlations between the $n = 134$ features is provided in Fig. 2.

2.2 Data preprocessing

The data preprocessing consists of several chronological steps prior to applying the ensemble feature selectors, see Fig. 3.

Data cleaning The first step in data preprocessing is to discard features known to be unimportant, such as features with only one unique value for all patients or duplicated features. Furthermore, we remove all data columns containing more than 25% missing values across all patients. By this criterion, we remove 16 features from block (p), one feature from block (b), six features from block (h), 14 features from block (i), and eight features from block (t).

Further, three patients are excluded from the experiments due to a high number of missing values in at least one block. All subsequent preprocessing steps are conducted on the remaining 63 patients and are applied by block to retain the homogeneous block structure.

Missing values Some values were missing because the clinicians did not fill out the case registration forms (CRF) properly or completely. Amongst other reasons, this may be because the information was missing in the patient journal, a blood sample was not done, a parameter was forgotten registered in the patient journal, or because the patients are

Table 1: One-hot versus ordinal encoding of a 4-level variable (levels A, B, C, D). Ordinal encoding assumes an order of the levels (here: $A < B < C < D$).

| level | one-hot encoding | ordinal encoding |
|-------|------------------|------------------|
| A | (0,0,0) | (0,0,0) |
| B | (0,0,1) | (0,0,1) |
| C | (0,1,0) | (0,1,1) |
| D | (1,0,0) | (1,1,1) |

referred from other hospitals. Such features, which are unavailable for a large percentage of patients, cannot be assessed properly in a data-driven manner and were therefore excluded — an imputation of those features would be unreliable due to the small sample size and may introduce incorrect or misleading information into the model.

As a second step, we impute the features with less than 25% missing values via an adaptation of the k -nearest neighbors (k NN) imputation algorithm [32]. The number of features and the number of patients that have at least one missing value for each block are: (p) (7:25), (b) (5:16), (h) (7:6), (i) (2:2), and (t) (3:3) where the first number represents the number of features and the second number represents the number of patients.

In particular, we restrict the feature space to non-missing columns and compute a matrix of pair-wise distances between all patients. We denote the set comprising the k -nearest neighbors of patient i by $N_k(i) \subseteq \{1, \dots, m\}$. Assuming that feature j is missing for patient i , we impute $x_{i,j}$ by $x_{i,j}^{\text{imp}}$, representing the median (instead of the mean, as suggested by [32]) of feature j across the patient’s k nearest neighbors where the feature value is known, i.e.

$$x_{i,j}^{\text{imp}} \leftarrow \text{median} \{x_{l,j} : l \in N_k(i)\}. \quad (1)$$

Ordered categorical features are transformed to an integer scale before interpolation. The usage of an odd value of k (by default, we use $k = 5$) guarantees that the median returns an integer, which is a clear benefit over the mean when using the technique for ordered features.

Categorical feature encoding Categorical features require encoding in order to be processed alongside numeric variables in predictive models. In particular, we distinguish between ordinal and nominal categorical variables: Nominal variables (i.e., variables without an internal order of the feature levels), such as *clinical institution*, are one-hot encoded [33]. Given a feature j with c_j feature levels, the one-hot encoding produces a set of $c_j - 1$ binary features $\{e_2, \dots, e_{c_j}\}$, given as follows:

$$(e_l)_i = \begin{cases} 1 & \text{if } x_{i,j} = l, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

for $l \in \{2, \dots, c_j\}$ indicating the feature level. The number of one-hot/ordinal categorical features is: 21/5 for block (p), 0/3 for block (b), 10/2 for block (h), 1/0 for block (i), and 5/0 for block (t). To avoid linear dependencies between features, the first feature level is not represented by a binary vector in the encoded space, but rather contributes to the model intercept, see Tab. 1.

Features with an internal order among their levels (ordinal variables), such as the *WHO performance status* with levels 0, 1, 2, 3, and 4, require an ordinal encoding to retain the relevant information about the order. Under the assumption that the influence of a feature increases from lower to higher levels (i.e., higher levels comprise the lower levels and an additive effect), the following encoding is used:

$$(e_l)_i = \begin{cases} 1 & \text{if } x_{i,j} \leq l, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

for feature level $l \in \{2, \dots, c_j\}$. Again, the first feature level, which would be assigned a value of 1 across all samples in the encoded space, is not assigned a binary vector in the encoded space. A comparison between one-hot and ordinal encoding is provided in Tab. 1. In contrast to transforming to an integer scale, this binary ordinal encoding preserves the order among the categories but does not pretend equal distances between the categories on a numerical scale.

Feature transformation and normalization During our experiments, we split the dataset into train and test sets. To normalize the distribution of each numeric feature, we use the Yeo-Johnson power transformation along with standardization [34]. The Yeo-Johnson power transformation is an extension of the well-established Box-Cox transformation with the benefit that it enables the transformation of negative and zero values. The intention is to bring the data closer to a normal distribution by simultaneously stabilizing data variance. For a given feature j , Yeo-Johnson’s power transform is defined as

Table 2: Encoding of the target variable "overall survival" (OS) [months].

| level | encoding |
|-------------------|----------|
| $OS \leq 12$ | 1 |
| $12 < OS \leq 24$ | 2 |
| $24 < OS \leq 36$ | 3 |
| $36 < OS \leq 48$ | 4 |
| $48 < OS \leq 60$ | 5 |
| $60 < OS$ | 6 |

$$x_{i,j}^{YJ} \leftarrow \begin{cases} \frac{((x_{i,j} + 1)^{\lambda_j} - 1)}{\lambda_j} & \text{if } \lambda_j \neq 0, x_{i,j} \geq 0 \\ \log(x_{i,j} + 1) & \text{if } \lambda_j = 0, x_{i,j} \geq 0 \\ -\frac{((-x_{i,j} + 1)^{2-\lambda_j} - 1)}{2 - \lambda_j} & \text{if } \lambda_j \neq 2, x_{i,j} < 0 \\ -\log(-x_{i,j} + 1) & \text{if } \lambda_j = 2, x_{i,j} < 0. \end{cases} \quad (4)$$

Commonly, the transformation parameter λ_j is estimated from the data using a maximum likelihood approach. After the Yeo-Johnson transformation, we scale the data to zero mean and variance of 1. To prevent biased train and test data, the transformation parameter λ_j and the mean and variance for the standardization are estimated on the training data in each split separately.

Encoding of the target variable Even though machine learning models for censored data are evolving, most present predictive models cannot handle censored data [35]. To avoid the problem presented by censored data, we encode the OS in months into an integer value (1-6). Using 60 months median follow-up time as a reference, there are no censored patients with OS below 60 months. Considering survival on a yearly basis we use the representation of the target variable in our experiments as in Tab. 2. Since each level in the encoded space equals one year, predictive errors used in the remainder of this paper refer to a yearly scale.

2.3 Feature Selection Methods

In this work, we investigate two ensemble feature selection methods, which have been tailored to fit the requirements of datasets in the life science domain: the Repeated Elastic Net Technique for Feature Selection (RENT) [27] and the User-Guided Bayesian Framework for Feature Selection (UBayFS) [19]. Both methods build on the principle of (a) randomly subsampling the input dataset and (b) training an elementary feature selection model on each sample. The final feature set is determined by applying a meta-model on the feature sets selected by the elementary models, see Fig. 4. In the case of RENT, the elementary feature selector type is restricted to elastic net regularization [36] using logistic regression models for binary classification problems or ordinary least squares linear regression models for regression problems, while UBayFS operates on an arbitrary elementary model type.

RENT The rules to obtain a final feature set further demonstrate the distinct scopes of the methods: RENT defines three criteria τ_1 , τ_2 and τ_3 for the selection of features based on the distribution of their weights across the elementary models; (I) the number of times the feature weights are non-zero (τ_1) is above a level specified by the user; (II) the alternation of the sign of the feature weights does not surpass a user-defined level (τ_2); (III) the size of the feature weights deviate significantly from 0 (τ_3). The hyperparameters for RENT comprise of a number M of elementary models, an internal data split ratio, two parameters associated with the elastic net regularization in the elementary models (C and ℓ_1), as well as one cut-off parameter for each of the three criteria τ_1 , τ_2 , τ_3 .

UBayFS In contrast, UBayFS combines the selection frequency of each feature across the elementary models with prior information from domain experts, along with side constraints. In particular, the prior weighting of features is possible, along with the definition of linear side constraints between features (and feature blocks). In practice, weights can represent knowledge about the importance of features, which is verified from previous publications. Side constraints enable the user to restrict the feature set's maximum size \max_s and account for the intrinsic block structure during feature selection (e.g., in multi-source datasets). Hence, RENT implements a purely data-driven approach based on Elastic Net, while UBayFS is a general meta-model with capabilities to integrate contextual information about the

data generation process. In its most basic setup, UBayFS requires as hyperparameters a number of elementary models M and an internal data split ratio, a maximum number of features \max_{gs} , and a model type to use as the elementary feature selector.

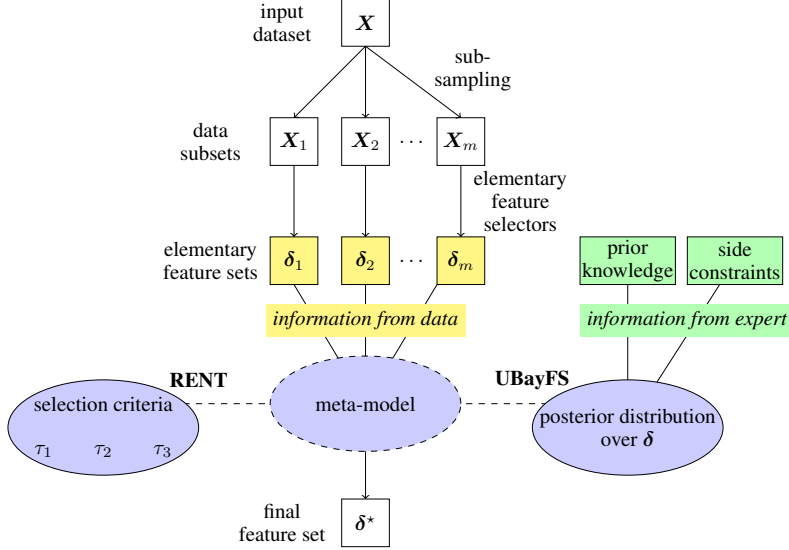


Figure 4: Overall structure of both ensemble feature selection methods, RENT and UBayFS. After training elementary feature selectors, information is combined in a meta-model. While RENT uses information from data only, UBayFS additionally includes expert information.

2.4 Outcome prediction

Linear regression Given a set of selected features S , we make use of linear regression models [37] to model the target variable \mathbf{y} . In its simplest form, the linear regression model (with intercept) is given as

$$\mathbf{y} = \tilde{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad (5)$$

where $\boldsymbol{\beta} \in \mathbb{R}^{n+1}$ is the model parameter vector, $\tilde{\mathbf{X}}$ denotes the matrix containing one column of ones, followed by the sub-matrix of \mathbf{X} restricted to the columns contained in S . Further, $\boldsymbol{\varepsilon} \sim N(0, \sigma^2)$ denotes the model error with constant error variance $\sigma > 0$. By default, parameters of linear regression models are obtained via ordinary least squares (OLS), i.e. by minimizing the least squares error

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2. \quad (6)$$

Once the parameter vector $\boldsymbol{\beta}$ is estimated by optimizing Eq. 6 analytically, predictions are obtained by evaluating $\hat{\mathbf{y}} = \tilde{\mathbf{X}}\boldsymbol{\beta}$.

k -nearest neighbor (k NN) regression As an alternative to the linear regression model, a k -nearest neighbor (k NN) regression model [37] is used to compute predictive results. In contrast to the linear regression model, the k NN model does not assume a linear relationship between the predictors and the target variable. Similar to the k NN method used for missing value imputation in Section 2.2, a neighborhood $N_k(i)$ of sample i containing the k nearest training data points with respect to a Euclidean metric on the feature space is computed for any data point x_i . The prediction for the target value y_i corresponding to sample i is given by the mean of the neighbor's target values

$$\hat{y}_i = \frac{1}{k} \sum_{l \in N_k(i)} y_l. \quad (7)$$

Note that, the neighborhood $N_k(i)$ is a subset of the training samples only, while \hat{y}_i may represent both, training or test samples.

Both predictive models, linear regression as well as the k NN regression model, are known to suffer from the curse of dimensionality — hence, we can assume that selecting a high number of features deteriorates each model. The opposite extreme for both methods, i.e., selecting no features at all, leads to predicting the output with the mean over the training data regardless of the input. Thus, we expect a well-performing feature selector to deliver a proper subset S of the feature set $\{1, \dots, n\}$, which allows both predictive models to perform better than the baselines given by (a) the overall mean of the target variable, and (b) a model including all features.

2.5 Implementation

Parts of our analyses are conducted in the programming languages R [38]; other parts are conducted in Python [39]. We use the open-source implementations for RENT [40] and UBayFS [41]. For data preparation and preprocessing, we deploy the R package *caret* [42], and the Python package *scikit-learn* [43]. Fold indices are shared between R and Python. Predictive models are trained and evaluated in R using the *caret* package for all model setups. All plots are created using package *ggplot2* [44].

All results are produced on an Intel Core i7 CPU @1.8 GHz, 32GB RAM under a Windows 11 Pro operating system.

3 Experiments

Our experimental results are structured into a pre-study, where we determine optimal hyperparameters for the feature selection algorithms, followed by two main experiments. Experiment 1 focuses on the comparison of the two models, RENT and UBayFS, on the dataset without accounting for additional expert knowledge. Experiment 2 is operated on UBayFS only, as prior information and additional side constraints are included in the feature selection.

Our main focus in the experiments lies on the selected feature sets, along with the impact of the feature selection on predictive performance. We provide feature counts from both of the investigated feature selectors, RENT and UBayFS, across five different train-test splits of the dataset. Unless specified otherwise, all experiments are conducted using the hyperparameters determined during the pre-study.

3.1 Experimental setup

Model parameters Both algorithms, RENT and UBayFS, are trained on $M = 100$ ensemble models and internal 0.75/0.25-splits for sub-sampling the dataset. The underlying elementary feature selector for RENT is, by definition, an elastic net regularized linear regression model. Thus, RENT requires five hyperparameters to be determined during the pre-study (2 elastic net regularization parameters, ℓ_1 and C , as well as three thresholds τ_1, τ_2 , and τ_3 for the selection criteria). In order to make results comparable with UBayFS, we further deploy a side condition to restrict the search space to settings, which deliver a maximum number of features \max_s during validation. Thus, the number of features selected by RENT is approximately equal to the pre-defined parameter \max_s .

UBayFS uses minimum redundancy max relevance (mRMR) [23] as an elementary feature selector. The internal number of features in each elementary model is set to \max_s , i.e., each elementary model selects exactly \max_s features. For the meta-model, the same parameter \max_s is used to restrict the maximum number of selected features via a max-size side constraint (hard constraint) — while different levels of \max_s are evaluated in experiment 1, the parameter is set to the default $\max_s = 20$ in experiment 2. Further, unless otherwise stated, prior feature weights in UBayFS are set uniformly to 0.1 across all features, which results in a non-informative prior.

Train-test splits As the ratio between the number of patients and features is unbalanced, with 63 patients and 134 encoded features, the reliability of the feature ranking results must be validated to reduce the risk of spurious correlations and overfitting. Hence, we perform a 5-fold split of the dataset. For all possible permutations, we use four folds for training UBayFS or RENT, as well as the predictive models and the remaining fold for testing. Hyperparameters are determined on each split separately by internally subsetting the 4-fold training set (nested split). The 5-fold splits and hyperparameters determined in the pre-study remain the same across all experiments.

For each feature selection method, we provide the selection frequencies of each feature across the 5 folds, i.e., a feature obtains an importance score between 0 and 5 according to the number of folds it was selected for. For predictive performance scores, a linear regression model and a k NN regression model are trained on the same training folds, using the features from the preceding feature selection, and evaluate the prediction error on the test set (averaged across all folds).

Table 3: Selected hyperparameters for each train/test split.

| parameter | | fold | | | | |
|-----------|----------|-------|-------|-------|-------|-------|
| | | 1 | 2 | 3 | 4 | 5 |
| RENT | ℓ_1 | 0.3 | 0 | 0.3 | 0.3 | 0.3 |
| | C | 1 | 1 | 1 | 1 | 1 |
| | τ_1 | 0.3 | 0.5 | 0.3 | 0.35 | 0.35 |
| | τ_2 | 0.3 | 0.5 | 0.4 | 0.35 | 0.35 |
| | τ_3 | 0.975 | 0.975 | 0.975 | 0.975 | 0.975 |
| UBayFS | \max_s | 20 | 20 | 20 | 20 | 20 |

Performance metrics To assess whether a feature set contains relevant information for training predictive models, we analyze the predictive performance in a regression setup following the feature selection step. The performance is quantified using the root mean squared error (RMSE), which has a lower bound of 0 and shall be minimized.

Using the stability criterion introduced by [24], we further evaluate the feature selection stability across the five folds for RENT and UBayFS. The computed score is bounded in the interval $[0, 1]$; a value of 1 indicates perfect stability, i.e., the same feature set is selected in each model, while 0 indicates that selected feature sets show no overlap.

Furthermore, the redundancy rate (RED) returns an intrinsic feature set quality measure by computing the average absolute Pearson correlation among the selected features. Small correlations are desirable as highly correlated features represent redundant information. Equally to the absolute Pearson correlation coefficient, RED is bounded in $[0, 1]$.

In experiment 2, we additionally assign prior weights to a subset of features — therefore, we also evaluate the percentage of prior-elevated features (PERC) in the selected feature sets as well. If PERC is high, features extracted via data-driven feature selectors match the domain experts’ knowledge. However, a low PERC does not necessarily contradict expert knowledge since the features may be highly correlated, and therefore, similar information may be encoded in multiple distinct sets of features.

3.2 Pre-Study

The pre-study aims to determine the optimal hyperparameters for RENT. Given a 0.75/0.25 outer train-test split as specified above, only train data are used for hyperparameter selection. For this purpose, 4-fold cross-validation is performed on each train dataset (using the same 4 folds as in the outer train-test split). Across the resulting 4 models, hyperparameters are selected by maximizing predictive performance in a grid search over the parameter space $C \in \{1, 10, 100, 1000\}$, $\ell_1 \in \{0, 0.1, 0.2, \dots, 1\}$, $\tau_1, \tau_2 \in \{0, 0.05, 0.1, \dots, 1\}$, and $\tau_3 = 0.975$ (fixed).

The runtime for the full computation associated with the pre-study (parameter selection and final feature selection) for RENT comprised approx. 350 sec (16 cores, 24 threads in parallel). Since UBayFS does not require parameter selection, the runtime to evaluate the feature selection model for different levels of \max_s (see Experiment 1) is shorter (approx. 65 sec without parallelization).

Tab. 3 shows the hyperparameters identified for RENT and UBayFS in each train-test split (given by the numbers of the test folds 1-5). Due to the restriction of the maximum number of features, the stated parameters may not represent global maxima for the performance of RENT; however, comparability between the methods is preserved. Furthermore, since the number of features is restricted, the selected hyperparameters are in a similar range between the folds.

3.3 Experiment 1: feature selection without prior knowledge

Having determined hyperparameters for each fold in the pre-study, RENT, and UBayFS are applied in each data split to the training dataset to select an optimized feature set for a given \max_s on a purely data-driven basis.

Selected features For each feature, selection frequencies across the five test folds are further provided in Tab. 4 (columns RENT and UBayFS, $w = 0.1$). In addition to the selection frequency, the table indicates whether a feature shows a positive or negative impact on the target variable according to the coefficients in the linear model, if selected. Thereby, ++ and -- indicate that a feature always shows the same sign across all predictive models. In contrast, + and - indicate a majority of positive or negative coefficients across the predictive models, respectively.

Table 4: Feature selection frequencies across five folds by RENT and UBayFS (with different prior weight levels w for selected features). Features with increased prior weights in the UBayFS setup reported in the last column are highlighted with asterisks. In parentheses, + and - indicate that the majority of linear regression models containing the feature assigned it a positive or negative coefficient, respectively; ++ or -- indicate that all models containing the feature had equal signs of their coefficients, while no parentheses indicate that the distribution of signs was even.

| block | feature | RENT | UBayFS | block | feature | RENT | UBayFS |
|-------------------------|-----------------------------------------|-----------|----------|----------------------------------------------|----------------------------------------------|----------|-----------|
| | | $w = 0.1$ | $w = 50$ | | | $w = 50$ | $w = 110$ |
| (a) | * Age at Diagnosis | 2 | 1(-) | 5(-) | * Kt67 | 5(-) | 5(-) |
| | Time from PET to Metastasis (days) | 1(+) | 1(+) | 5(-) | * Hist Exam Metastasis | 0 | 0 |
| | Time from PET to Diagnosis (days) | 1(+) | 1(+) | 5(-) | * Primary Tumour Site | 0 | 0 |
| | Time from diag to mets (months) | 0 | 0 | 0 | * Primary Tumour Cellularity | 0 | 0 |
| | Time from diag to mets (months) | 0 | 0 | 0 | * Primary Tumour Cellularity/det | 0 | 4(+) |
| | Loc. Adv. Resectable Disease | 0 | 0 | 0 | * Primary Tumour Gastric | 0 | 1(+) |
| | Loc. Adv. Resectable Disease | 0 | 0 | 0 | * Primary Tumour Other abdominal | 0 | 5(-) |
| | Recurrence | 0 | 0 | 0 | * Primary Tumour Rectum | 1(+) | 1(+) |
| | Metastatic Disease at Time of Diagnosis | 3(+) | 1(+) | 0 | * Unknown Pr. With Dominance of GI met. | 0 | 0 |
| | Treatment Intention Palliative | 4(-) | 3(-) | 0 | * Co-existing Neoplasm Adenoma | 0 | 0 |
| | Living Alone | 2(+) | 1(-) | 0 | * No Co-existing Neoplasm | 0 | 0 |
| | * TNM staging Pathological | 0 | 0 | 0 | * Tumour Morphology WD | 0 | 0 |
| | Stage grouped Stage IV | 0 | 0 | 5(-) | * Chromogranin A Staining | 4(+) | 3(-) |
| | Mets Bone | 5(-) | 5(-) | 0 | * Architecture Solid | 1(+) | 0 |
| | Mets LN Regional | 0 | 0 | 0 | * Architecture Organoid | 1(+) | 1(+) |
| | Mets LN Regional | 0 | 0 | 0 | * Vessel Pattern | 1(+) | 1(+) |
| | Mets LN Retro | 0 | 0 | 0 | * Vessel Pattern Disant | 2(-) | 1(-) |
| | Mets Liver | 0 | 0 | 0 | * Biopsy Location Gastric | 0 | 0 |
| | Mets Lung | 0 | 0 | 0 | * Biopsy Location Liver Metastasis | 0 | 0 |
| | Mets Other | 0 | 0 | 0 | * Biopsy Location Lung Metastasis | 0 | 0 |
| | Mets Shin | 0 | 0 | 0 | * Biopsy Location Oesophagus | 0 | 0 |
| | Primary Tumour Resected | 0 | 0 | 0 | * Biopsy Location Pancreas | 0 | 0 |
| | M-stage M1 | 1(-) | 0 | 0 | * Nerve Invasion | 2 | 1(-) |
| | Non Smoker | 3(+) | 1(+) | 0 | * Stromal Reaction | 4(+) | 3(+) |
| | Smoker | 0 | 0 | 0 | * Geographical Necrosis | 0 | 0 |
| | Radical Surgery | 0 | 0 | 0 | * Stromal Reaction in Staining 2+ | 0 | 0 |
| Co-morbidity Severity 1 | 0 | 0 | 0 | * Stromal Reaction in Staining 3+ | 0 | 0 | |
| T-stage T2 | 0 | 0 | 0 | * Infiltration to Scan (mm) | 2 | 2(+) | |
| T-stage T3 | 0 | 0 | 0 | * Weight [kg] | 2(-) | 0 | |
| T-stage T4 | 2(-) | 2(-) | 0 | * Total MTV (cm ³) | 3(+) | 1(-) | |
| N-stage N1 | 0 | 0 | 0 | * SUVmax | 0 | 4(+) | |
| * WHO Perf Stat 1 | 0 | 0 | 0 | * SUVmean | 2 | 4 | |
| * WHO Perf Stat 2 | 0 | 0 | 0 | * SUVmax (total) | 1(+) | 0 | |
| * WHO Perf Stat 3 | 4(-) | 5(-) | 3(-) | * SUVmean (total) | 5(-) | 5(-) | |
| * WHO Perf Stat 4 | 0 | 0 | 0 | * Institutional Risk/bospital | 4(+) | 4(+) | |
| | | | | * Height (cm) | 0 | 0 | |
| | | | | * Height (cm)BMI | 2(-) | 0 | |
| | | | | * Time from PET to first treatment (days) | 0 | 0 | |
| | | | | * Chemotherapy Type Cisplatin/Etoposide | 4(+) | 3(+) | |
| | | | | * Chemotherapy Type Other | 0 | 0 | |
| | | | | * Chemotherapy Type Temozolomide/Capitabine | 1(+) | 0 | |
| | | | | * Best Response (RECIST) Not Assessed | 5(+) | 4(+) | |
| | | | | * Best Response (RECIST) Stable Disease | 0 | 0 | |
| | | | | * Best Response (RECIST) Only Clinical PD | 0 | 0 | |
| | | | | * Best Response (RECIST) Progressive Disease | 2(+) | 0 | |
| | | | | * Best Response (RECIST) Progressive Disease | 0 | 0 | |
| | | | | * Best Response (RECIST) Stable Disease | 0 | 0 | |
| | | | | * Reintroduction with Cisplatin/Etoposide | 4(-) | 4(-) | |
| | | | | * Treatment Stopped | 1(+) | 3(+) | |
| | | | | * Treatment Stopped Progression of Disease | 0 | 0 | |
| | | | | * Treatment Stopped Toxicity | 5(-) | 4(+) | |
| | | | | * Progressive | 3 | 3(+) | |
| | | | | | | 1(-) | |
| (b) | * Abs. Neutrophil Count | 0 | 0 | 0 | * Time from PET to first treatment (days) | 0 | 0 |
| | Albumin | 2 | 4(-) | 5(+) | * Chemotherapy Type Cisplatin/Etoposide | 4(+) | 3(+) |
| | Creatinine | 5(-) | 5(-) | 5(+) | * Chemotherapy Type Other | 0 | 0 |
| | Hemoglobin | 0 | 0 | 0 | * Chemotherapy Type Temozolomide/Capitabine | 1(+) | 0 |
| | WBC | 1(-) | 1(-) | 0 | * Best Response (RECIST) Not Assessed | 5(+) | 4(+) |
| | ALP > Normal <= 3UNL | 4(-) | 5(-) | 0 | * Best Response (RECIST) Stable Disease | 0 | 0 |
| | ALP > Normal <= 3UNL | 1(+) | 1(-) | 0 | * Best Response (RECIST) Only Clinical PD | 0 | 0 |
| | ALP > Normal <= 3UNL | 0 | 0 | 0 | * Best Response (RECIST) Progressive Disease | 2(+) | 0 |
| | Chromogranin-A > Normal <= 2UNL | 0 | 0 | 0 | * Best Response (RECIST) Progressive Disease | 0 | 0 |
| | Chromogranin-A > Normal <= 2UNL | 0 | 0 | 0 | * Best Response (RECIST) Stable Disease | 0 | 0 |
| | LDH > Normal <= 2UNL | 0 | 0 | 4(+) | * Reintroduction with Cisplatin/Etoposide | 4(-) | 4(-) |
| | LDH > 2UNL | 0 | 0 | 1(+) | * Treatment Stopped | 1(+) | 3(+) |
| | LDH > 2UNL | 0 | 0 | 5(+) | * Treatment Stopped Progression of Disease | 0 | 0 |
| | NSE > 2UNL | 0 | 0 | 0 | * Treatment Stopped Toxicity | 5(-) | 4(+) |
| | NSE > 2UNL | 0 | 0 | 0 | * Progressive | 3 | 3(+) |
| | * Platelets | 2(-) | 4(-) | 5(-) | | | 1(-) |

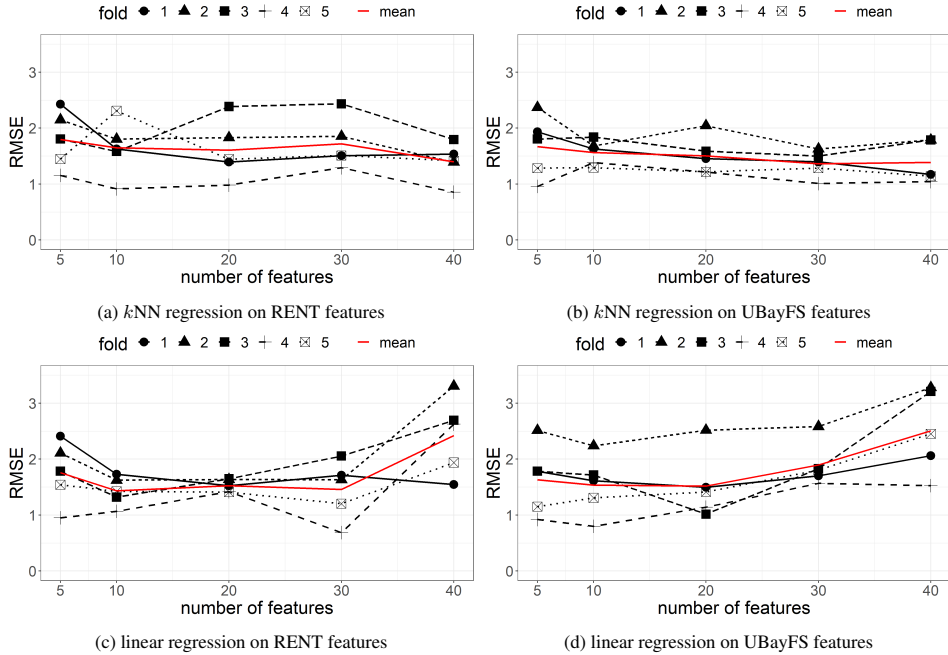


Figure 5: Predictive performances (on test set) of models trained after feature selection for different numbers of features.

Predictive performance Further, Fig. 5 illustrates the predictive performances of k NN and linear regression models trained after UBayFS and RENT feature selection. The plot shows the RMSE for each fold given a predefined number of selected features \max_s .

Notably, RENT performs better using the linear regression model as the predictor, while UBayFS shows a better performance in combination with k NN. The stronger performance of RENT with linear regression may be a result of the fact that the underlying feature selection in RENT is based on a regularized linear regression model. UBayFS, however, is based on mRMR, which does not build upon a linear predictive model.

While linear regression results deteriorate at a higher number of features ($\max_s > 30$), the k NN model retains a similar performance level, which suggests that the curse of dimensionality does not yet have a strong effect on the Euclidean distance for the given feature space dimensionalities. For the linear model, overfitting is triggered by a large ratio between the number of features and the number of patients.

Among all compared methods, differences between the folds are obvious: for instance, fold 4 is predicted with the least RMSE across all combinations of feature selector, predictive model, and \max_s . On the other hand, fold 2 is associated with a large RMSE in the models based on UBayFS, while fold 3 shows similar behavior for the k NN model based on RENT features. Potentially, differences between folds may be caused by two factors (or combinations of both):

- the cohort of patients in the *training set* does not represent the global distribution of the data well — e.g., the training data do not contain a sufficient number of samples with particularly high or low target values (bad prediction due to a bad model);
- the cohort of patients in the *test set* is particularly hard to estimate, e.g., due to outliers (bad prediction in spite of an appropriate model);

Due to the low number of only 12-13 patients in each fold, even a low number of hard-to-predict outliers may deteriorate RMSE results significantly.

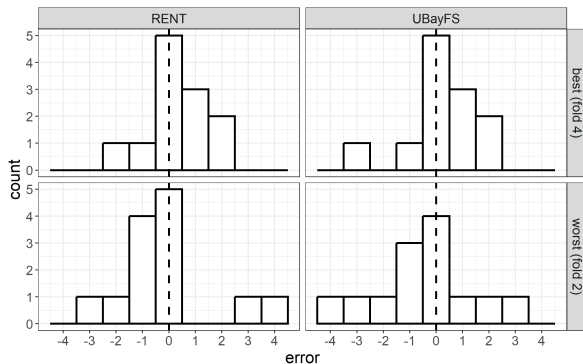


Figure 6: Histograms of errors on the test set (predicted value by k NN - ground truth) of the folds performing best (fold 4) and worst (fold 2) at $\max_s = 20$ features.

Residuals In order to shed light on the dynamics leading to the differences in performance between the data folds, histogram plots of the residuals for fold 2 (worst fold in UBayFS) and fold 4 (best fold across most setups) at $\max_s = 20$ are provided in Fig. 6. Residuals are defined as the difference between the true value and the prediction; thus a positive or negative residual value indicates an underestimation or overestimation of the lifetime, respectively.

In contrast to fold 4, the residuals from fold 2 are more dispersed. All histograms are symmetric and centered around 0, which indicates that all methods are able to estimate the intercept correctly. In both folds, the prediction model was able to predict the correct lifetime category for almost half of the patients in the test set. However, the histogram indicates that predictive models based on both feature selectors overestimate the lifetime in test fold 4 (positive errors), while lifetimes in test fold 2 are rather slightly overestimated (negative errors). The main difference in performance between fold 2 and fold 4 is driven by dispersion, i.e. by a minority of patients, which show a high error — due to the small sample size, even a small number of such outliers can impact the total RMSE significantly.

When considering patients with absolute residual values > 2.5 as outliers, RENT, and UBayFS show 3 outliers in fold 2, each (RENT: 2 positive, 1 negative; UBayFS: 1 positive, 2 negative). Both methods commonly misclassify one patient with true target value 6 and predictions 2.6 (UBayFS) and 2 (RENT), which substantiates the highest positive outlier in both histograms. The remaining two outliers of each method refer to different patients.

Stability In addition to the performance evaluation, we further investigate qualitative aspects of the selected feature sets, as shown in Fig. 7. The demonstrated stabilities and redundancy rates (RED) of the feature sets selected by RENT and UBayFS across the five folds tend to increase with \max_s . While RENT has a slightly lower and more fluctuating stability (around 0.5), UBayFS shows a clear convergence at around 0.6. The RED is below 0.25 for all possible numbers of features, indicating that both RENT and UBayFS select features with small correlations.

3.4 Experiment 2: feature selection with prior knowledge

Previous research on GEP NEN shows that some features impact the survival of patients; those are *Age at diagnosis*, *WHO performance status*, *Primary tumor location*, *Tumor morphology*, *Tumor differentiation*, *Lactate dehydrogenase (LDH)*, *Platelets*, *Albumin*, *Ki-67*, *SUV_{max}*, and *TNM-staging* [5, 8–14]. Tumor differentiation is highly correlated to tumor morphology, so we do not include the feature in this work. Furthermore, findings by [8] indicate a high relevance of the features *Total MTV [cm³]* and *Total TLG [g]*, which shall be investigated.

In this experiment, we focus on these features (a total number of 22 features in the encoded space) within our feature selection and prediction pipeline. In particular, during experiment 1, the aforementioned features comprise 30% of the final feature sets (on average across the five folds and given $\max_s = 20$ features, each). We refer to this score as PERC (percentage of selected features supported by literature). In the following, we deploy prior weights on these features to investigate how UBayFS as a hybrid feature selector combining information from experts and data, performs in comparison to the pure data-driven methods presented in experiment 1. Since RENT cannot incorporate prior feature importances, this evaluation is restricted to UBayFS.

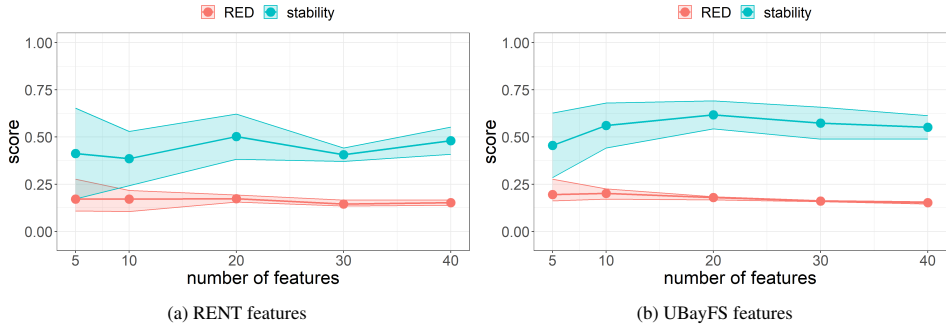


Figure 7: Stabilities and redundancy rates (RED) of feature sets selected by RENT and UBayFS ($\max_s = 20$ features, each).

Specifically, we increase the prior weight of the 22 features supported by literature (referred to as prior-elevated features) to the following levels: $w \in \{0.1, 10, 20, \dots, 100, 110\}$ — after evaluating all levels with respect to predictive performance, we restrict to special cases $w = 0.1$ (non-informative prior weighting), $w = 50$ (mediocre prior weighting), and $w = 110$ (strong prior weighting). After applying UBayFS with the given levels of prior information, we examine how the feature set and the predictive performance develop. The case of 0.1 is equivalent to the uniform case without prior knowledge (default setup for UBayFS in experiment 1). In contrast, prior weight 110 indicates that each prior-elevated feature already is assigned a higher score than the maximum score that can be achieved throughout the elementary models ($M = 100$) — as a result, the selected features are exclusively restricted to those with prior information and elementary feature selectors in UBayFS are only used to select a feature set of $\max_s = 20$ features among the 22 prior-elevated features.

Predictive performance Fig. 8 shows the average performances along with the standard deviations across the 5 test folds. In general, lower levels of prior weights do not significantly impact the performance, although a minor improvement can be observed in folds 4 (k NN) and fold 3 (linear model) up to $w = 40$. By increasing the prior weight to a higher level, performance levels lead to stronger variability and an increase of RMSE in the better-performing folds, such as fold 4. Finally, if the prior weight is set to the maximum level of 110, all folds converge to a similar level since the data-driven feature selection hardly contributes to these setups. Thus, a potential conclusion is that moderate levels of prior knowledge can slightly increase models’ capabilities. In contrast, strong prior knowledge leads to a convergence towards the global mean performance across all folds — such prior setup acts as a strong restriction of the search space exploited by the feature selector.

Stability In contrast to the minor effects of prior knowledge on predictive performance, stability increases significantly, as shown in Fig. 8. Finally, at a maximum level of $w = 110$, stability converges towards an almost perfectly stable solution. This is due to the restriction of the search space to the prior-elevated features, which results in a selection of 20 out of only 22 features in total. As expected, the percentage of selected features supported by literature (PERC) also increases linearly with the level of prior weights provided. The redundancy rate between the selected features shows a slight decrease, indicating that the prior-elevated features contain only small correlations.

4 Discussion

Experiment 1 In our first experiment, we left out prior expert knowledge and let the feature selection be purely data-driven. We know that certain features were prognostic for survival in earlier studies, as mentioned in experiment 2 below. We wanted to study whether the same prognostic features would still be selected and if there were any currently unknown prognostic features that could be further researched. Comparing the two first columns in Tab. 4 we can see which features are selected repeatedly in different folds with RENT and UBayFS. We must keep in mind that we cannot directly compare the importance of the features in terms of a coefficient (e.g similar to Cox regression), just that they are repeatedly selected in each fold. Further, the correlation between features must also be considered when comparing the importance of features which we can find in Fig. 2. Two or more features with a moderate/high correlation contain the same information with respect to the model, and one fold may choose one over the other, whilst

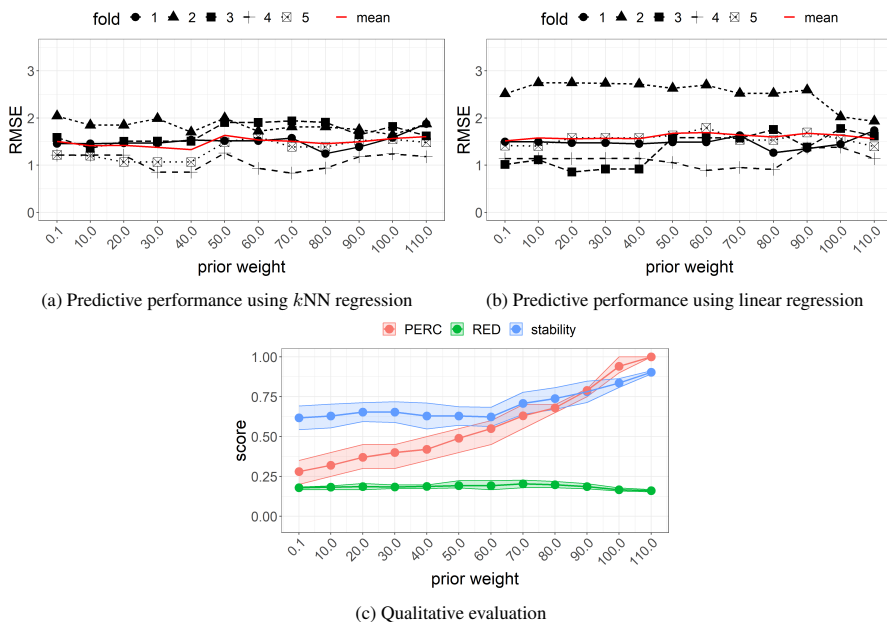


Figure 8: Experiment 2: predictive performances on fold 1-5, and qualitative metrics of features sets produced by UBayFS at different levels of prior knowledge on features with evidence from literature ($\max_s = 20$).

another fold may choose a highly correlated one instead. This results in a lower number for both features, not reflecting the importance of both when comparing them with a feature with a high number.

In block (p) (baseline patient characteristics) we have a few features that one would expect to be prognostic for OS. One obvious one would be *TNM-stage IV* disease which does not seem to be chosen at all by RENT and UBayFS. But looking at the correlation heatmap in Fig. 2 we see that this feature is highly correlated to several other features, among those *Metastatic Disease at Time of Diagnosis* and *Treatment Intention Palliative*. We see that this last one gets chosen four out of five times with RENT and five out of five times with UBayFS which probably explains why *TNM-stage IV* does not seem to be important. Having a palliative treatment intention usually means you have stage IV disease. This is also a well-known prognostic indicator from the literature [6]. Bone metastasis is usually a poor prognostic indicator in several types of cancers [45] and it is not surprising that this is chosen all the time. We also know that *WHO PS* is a prognostic indicator in these patients. This is also reflected in the number of folds it is chosen by RENT and UBayFS, but it is only WHO level 2 that seems to be important. That said, Fig. 2 shows that WHO levels 3 and 4 are highly correlated to some of the SUV-parameters which might contribute to those never being selected. *Radical Surgery* is quite often chosen by both RENT and UBayFS and is also a predictable prognostic indicator. Having radical surgery means that all viable tumors are removed, and that is only possible if you have a low tumor burden. This underlines the importance of surgery in the curative intended treatment of this type of cancer.

Next, in block (b) (baseline blood values), we see that both *CRP* and *ALP > Normal <= 3UNL* get selected equally many times by both RENT and UBayFS, and both have a high number indicating importance over the other features in this block. A high CRP at baseline has previously been shown to be a poor prognostic feature in some studies [5,46,47], whilst others have not replicated this [48]. This is probably not surprising as this has been shown to be a poor prognostic indicator in advanced cancer patients in a palliative setting, and especially in GEP NEN [49–52]. *ALP* has also been shown in studies to be prognostic for a shorter OS [5,53,54]. For *Albumin* and *Platelets*, RENT chose these only half as many times as UBayFS. Both have been shown to be prognostic indicators of OS [5,6]. Interestingly, *Haemoglobin*, *WBC*, *LDH*, and *Chromogranin A* are barely chosen or are not chosen neither by RENT nor UBayFS. All these features have previously been shown to be prognostic for OS [5].

Moving on to block (h) (re-evaluated histology) we have quite a few features that are well-known prognostic indicators. The strongest one from the literature is probably *Ki-67* which is used in the classification system of NEN. The second strongest is probably *Tumor Morphology* which has been shown in several studies to be prognostic for OS [5, 6]. We see that *Ki-67* is chosen every time from all five folds both for RENT and UBayFS supporting this feature as a strong prognostic indicator for OS. Further, *Tumor Morphology* gets chosen four out of five times with RENT and three out of five times with UBayFS. This is also to be expected since we know that patients with NET G3 have a better OS than those patients with NEC [55]. What is surprising is that most tumor sites, especially those patients with unknown primary and esophagus NEN, are not chosen by RENT or UBayFS. *Primary Tumor Site* has been shown to be prognostic in several studies [5, 6]. Several of the features like *Stroma*, *Architecture*, *Vessel Pattern*, *Co-existing neoplasm*, and *Geographic Necrosis* are considered typical for either NET G3 or NEC [56], and one might assume these are highly correlated with *Tumor Morphology*. Although this is not reflected in Fig. 2. Almost none of these features are chosen with RENT or UBayFS except for *Stroma*. NET G3 typically have hyalinized stroma and NEC have desmoplastic stroma [56].

Further, in block (i) (PET/CT imaging) the interesting features are *Total MTV*, *Total TLG*, and the SUV-parameters. From Fig. 2 and previous literature [8] we know that these features are often (if not always) highly correlated. Hence, the selection of SUV_{\max} (*total*) instead of the other features is probably related to this. Moreover, we know from previous studies [8, 11–13] that global measures such as *Total MTV* and *Total TLG* are poor prognostic indicators for OS in these tumors, but we lack stronger evidence in form of larger studies. Here we see that SUV_{\max} (*total*) is chosen in all five folds both for RENT and UBayFS supporting the previous findings that PET-parameters are good prognostic features of OS.

Finally, in block (t) (treatment) we can see a few features are selected often. *Chemotherapy treatment with cisplatin/etoposide* is not surprisingly a predictor for OS, and most of the patients did indeed receive this combination. No chemotherapy is obviously detrimental. We also see that the *Chemotherapy treatment with temozolomide/everolimus* gets chosen often both by RENT and UBayFS. This is probably because this chemotherapy regimen is more often chosen for those patients with a low *Ki-67* and these are more likely to be NET G3 which already have a better OS. Further, both *Number of Courses* and *Progression* are two features that are selected often by RENT and UBayFS. *Progression* and *No Progression* are obviously poor prognostic indicators, and one could assume that the higher *Number of Courses* a patient receives the longer before they have progression and hence they live longer. This is of course only an assumption and interpretation of the data at hand. It is a bit surprising that the response evaluation results did not get chosen. One would assume that patients with the best response - stable disease would fare better than those with progressive disease. Looking at Fig. 2 the features from this block have low correlation coefficients.

Experiment 2 Here we added prior expert knowledge and assigned two different weights. A weight $w = 50$ means approximately 50% expert-driven and 50% data-driven. A weight $w = 110$ means almost purely an expert-driven approach where we effectively force the selection of features only from the subset of those from prior expert knowledge. We concentrated on features that are well documented in several previous studies, although there exist more features in the literature suggesting prognostic values than these. The features selected from prior expert knowledge are listed in the first paragraph in Section 3.4 and marked by an asterisk in Tab. 4.

If we concentrate on the second, third, and fourth columns, which shows the difference between roughly 0%, 50% and 100% expert-driven, we see that none of the marked features drops in importance as we increase the value of expert knowledge. Some features that were never chosen with a pure data-driven model are still not chosen. One could argue that these are probably not strong features to begin with, or that other features contain the same and/or stronger information. A few features only get chosen when almost completely removing the data-driven part and make a huge leap from not being chosen to being chosen five times. We argue that one should be careful to put too much importance on these features as we expect these are more or less forced to be chosen.

A few features stand out by being stable across all values of w ; *WHO Performance Status*, *Albumin*, *Platelets*, *Ki-67*, *Tumor Morphology*, *Total MTV*, *Total TLG*, and SUV_{\max} . It would be bold to assume that these features are the most important and stable predictors of OS from the subset of expert knowledge markers, but that would probably be too premature. Further, it is also interesting to notice that even though several parameters from PET are highly correlated, several are still chosen very often by the model. This is in line with the results of our previous study ([8]). Moreover, it is a bit surprising that *Primary Tumor Site*, especially *Unknown Primary* and *Esophagus*, is not chosen more often as these are well-known negative predictors of OS [5, 6].

We also notice that some of the other non-marked features drop in importance as we increase w , and this is probably related to the fact that the features overlap in the information they add to the model. A few of these features are also moderately or highly correlated. E.g. *CRP* is correlated with quite a few of the other blood markers, and this could explain why it falls in importance when increasing w . *Mets Bone* (bone metastases) is not listed in the correlation

heatmap and thus has no moderate or high correlations with other features, but still completely falls out. Bone metastases usually occur late in several cancers and is a poor prognostic feature. Hence, one should assume that this feature and similar ones like *CRP*, *ALP*, which performs well with low values of w falls off in the pure knowledge-driven model because the model is "forced" to select only marked features. We must remember that the $w = 110$ is an extreme expert-driven model which is probably not clinically relevant but was added to explore and evaluate what the model did in this extreme situation. This is a small, novel study with few patients and really the first of its kind for exploring and evaluating RENT and UBayFS on clinical data. Using these ensemble feature selectors may be used for validating already established features, or to find new features not previously known. Evaluation into which w is optimal should be explored further in future studies.

5 Conclusion

In conclusion, although we cannot ascertain how important different features are compared to each other and if they contribute to poorer or better survival, we do find similar results as several previous studies. The most stable and predictive features in our study are *WHO Performance Status*, *Albumin*, *Platelets*, *Ki-67*, *Tumor Morphology*, *Total MTV*, *Total TLG*, and *SUV_{max}*.

From a data science perspective, we demonstrated the capabilities of the ensemble feature selection techniques RENT and UBayFS for healthcare problems — in particular, the inclusion and comparison of expert- and data-driven setups, as well as combinations of both, allow the user to gain relevant information for clinical use.

References

- [1] Benjamin E. White, Brian Rous, Kandiah Chandrakumar, Kwok Wong, Catherine Bouvier, Mieke Van Hemelrijck, Gincy George, Beth Russell, Rajaventhhan Srirajaskanthan, and John K. Ramage. Incidence and survival of neuroendocrine neoplasia in England 1995–2018: A retrospective, population-based study. *The Lancet Regional Health - Europe*, 23:100510, December 2022.
- [2] Raziye Boyar Cetinkaya, Bjarte Aagnes, Espen Thiis-Evensen, Steinar Tretli, Deidi S. Bergstuen, and Svein Hansen. Trends in incidence of neuroendocrine neoplasms in Norway: A report of 16, 075 cases from 1993 through 2010. *Neuroendocrinology*, 104(1):1–10, November 2015.
- [3] International Agency for Research on Cancer. *WHO Classification of Tumours. Digestive System Tumours*. World Health Organization Classification of Tumours. IARC, 1 edition, July 2019.
- [4] Guido Rindi, David S. Klimstra, Behnoush Abedi-Ardekani, Sylvia L. Asa, Frederik T. Bosman, Elisabeth Brambilla, Klaus J. Busam, Ronald R. de Krijger, Manfred Dietel, Adel K. El-Naggar, Lynnette Fernandez-Cuesta, Günter Klöppel, W.Glenn McCluggage, Holger Moch, Hiroko Ohgaki, Emad A. Rakha, Nicholas S. Reed, Brian A. Rous, Hironobu Sasano, Aldo Scarpa, Jean-Yves Scoazec, William D. Travis, Giovanni Tallini, Jacqueline Trouillas, J.Han van Krieken, and Ian A. Cree. A common classification framework for neuroendocrine neoplasms: an International Agency for Research on Cancer (IARC) and World Health Organization (WHO) expert consensus proposal. *Modern Pathology*, 31(12):1770–1786, December 2018.
- [5] H. Sorbye, S. Welin, S.W. Langer, L.W. Vestermark, N. Holt, P. Osterlund, S. Dueland, E. Hofslis, M.G. Guren, K. Ohrling, E. Birkemeyer, E. Thiis-Evensen, M. Biagini, H. Gronbaek, L.M. Soveri, I.H. Olsen, B. Federspiel, J. Assmus, E.T. Janson, and U. Knigge. Predictive and prognostic factors for treatment and survival in 305 patients with advanced gastrointestinal neuroendocrine carcinoma (WHO G3): The NORDIC NEC study. *Annals of Oncology*, 24(1):152–160, January 2013.
- [6] Arvind Dasari, Kathan Mehta, Lauren A. Byers, Halfdan Sorbye, and James C. Yao. Comparative study of lung and extrapulmonary poorly differentiated neuroendocrine carcinomas: A SEER database analysis of 162, 983 cases. *Cancer*, 124(4):807–815, December 2017.
- [7] Arvind Dasari, Chan Shen, Anjali Devabhaktuni, Ruda Nighot, and Halfdan Sorbye. Survival according to primary tumor location, stage, and treatment patterns in locoregional gastroenteropancreatic high-grade neuroendocrine carcinomas. *The Oncologist*, 27(4):299–306, February 2022.
- [8] Henning Langen Stokmo, Mahmoud Aly, Inger Marie Bowitz Lothe, Austin J. Borja, Siavash Mehdizadeh Seraj, Rina Ghorpade, Xuan Miao, Geir Olav Hjortland, Eirik Malinen, Halfdan Sorbye, Thomas J. Werner, Abass Alavi, and Mona-Elisabeth Revheim. Volumetric parameters from [18F]FDG PET/CT predicts survival in patients with high-grade gastroenteropancreatic neuroendocrine neoplasms. *Journal of Neuroendocrinology*, 34(7):e13170, 2022.

- [9] M Heetfeld, C N Chougnnet, I H Olsen, A Rinke, I Borbath, G Crespo, J Barriuso, M Pavel, D O'Toole, T Walter, and other Knowledge Network members. Characteristics and treatment of patients with G3 gastroenteropancreatic neuroendocrine neoplasms. *Endocrine-Related Cancer*, 22(4):657–664, June 2015.
- [10] Sangwon Han, Hyo Sang Lee, Sungmin Woo, Tae-Hyung Kim, Changhoon Yoo, Baek-Yeol Ryoo, and Jin-Sook Ryu. Prognostic value of 18F-FDG PET in neuroendocrine neoplasm. *Clinical Nuclear Medicine*, Publish Ahead of Print, May 2021.
- [11] David L. Chan, Elizabeth J. Bernard, Geoffrey Schembri, Paul J. Roach, Meaghan Johnson, Nick Pavlakakis, Stephen Clarke, and Dale L. Bailey. High metabolic tumour volume on 18-fluorodeoxyglucose positron emission tomography predicts poor survival from neuroendocrine neoplasms. *Neuroendocrinology*, 110(11-12):950–958, November 2019.
- [12] Ho Seong Kim, Joon Young Choi, Dong Wook Choi, Ho Yeong Lim, Joo Hee Lee, Sun Pyo Hong, Young Seok Cho, Kyung-Han Lee, and Byung-Tae Kim. Prognostic value of volume-based metabolic parameters measured by 18F-FDG PET/CT of pancreatic neuroendocrine tumors. *Nuclear Medicine and Molecular Imaging*, 48(3):180–186, February 2014.
- [13] Sun Min Lim, Hyunki Kim, Beodeul Kang, Hyo Song Kim, Sun Young Rha, Sung Hoon Noh, Woo Jin Hyung, Jae-Ho Cheong, Hyoung-II Kim, Hyun Cheol Chung, Mijin Yun, Arthur Cho, and Minkyu Jung. Prognostic value of 18F-fluorodeoxyglucose positron emission tomography in patients with gastric neuroendocrine carcinoma and mixed adenoneuroendocrine carcinoma. *Annals of Nuclear Medicine*, 30(4):279–286, February 2016.
- [14] Giovanni Centonze, Patrick Maisonneuve, Natalie Prinzi, Sara Pusceddu, Luca Albarello, Eleonora Pisa, Massimo Barberis, Alessandro Vanoli, Paola Spaggiari, Paola Bossi, Laura Cattaneo, Giovanna Sabella, Enrico Solcia, Stefano La Rosa, Federica Grillo, Giovanna Tagliabue, Aldo Scarpa, Mauro Papotti, Marco Volante, Alessandro Mangogna, Alessandro Del Gobbo, Stefano Ferrero, Luigi Rolli, Elisa Roca, Luisa Bercich, Mauro Benvenuti, Luca Messerini, Frediano Inzani, Giancarlo Pruneri, Adele Busico, Federica Perrone, Elena Tamborini, Alessio Pellegrinelli, Ketevani Kankava, Alfredo Berruti, Ugo Pastorino, Nicola Fazio, Fausto Sessa, Carlo Capella, Guido Rindi, and Massimo Milione. Prognostic factors across poorly differentiated neuroendocrine neoplasms: a pooled analysis. *Neuroendocrinology*, November 2022.
- [15] P. Kubben, M. Dumontier, and A. Dekker. *Fundamentals of Clinical Data Science*. Springer International Publishing, 2019.
- [16] Mattea L. Welch, Chris McIntosh, Benjamin Haibe-Kains, Michael F. Milosevic, Leonard Wee, Andre Dekker, Shao Hui Huang, Thomas G. Purdie, Brian O'Sullivan, Hugo J.W.L. Aerts, and David A. Jaffray. Vulnerabilities of radiomic signature development: The need for safeguards. *Radiotherapy and Oncology*, 130:2–9, January 2019.
- [17] David Wallis and Irène Buvat. Clever Hans effect found in a widely used brain tumour MRI dataset. *Medical Image Analysis*, 77:102368, April 2022.
- [18] Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, April 2016.
- [19] Anna Jenul, Stefan Schrunner, Jürgen Pilz, and Oliver Tomic. A user-guided bayesian framework for ensemble feature selection in life science applications (UBayFS). *Machine Learning*, 111(10):3897–3923, 2022.
- [20] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [21] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984.
- [22] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS'05*, page 507–514, Cambridge, MA, USA, 2005. MIT Press.
- [23] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(02):185–205, 2005.
- [24] Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown. On the stability of feature selection algorithms. *Journal of Machine Learning Research*, 18(174):1–54, 2018.
- [25] Verónica Bolón-Canedo and Amparo Alonso-Betanzos. *Recent advances in ensembles for feature selection*. Intelligent Systems Reference Library. Springer International Publishing, Basel, Switzerland, 1 edition, May 2018.

- [26] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
- [27] Anna Jenul, Stefan Schrunner, Kristian Hovde Liland, Ulf Geir Indahl, Cecilia Marie Futsæther, and Oliver Tomic. RENT—repeated elastic net technique for feature selection. *IEEE Access*, 9:152333–152346, 2021.
- [28] International Agency for Research on Cancer. *WHO classification of tumours of the digestive system*. World Health Organization Classification of Tumours. IARC, 4 edition, November 2010.
- [29] E.A. Eisenhauer, P. Therasse, J. Bogaerts, L.H. Schwartz, D. Sargent, R. Ford, J. Dancey, S. Arbuck, S. Gwyther, M. Mooney, L. Rubinstein, L. Shankar, L. Dodd, R. Kaplan, D. Lacombe, and J. Verweij. New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1). *European Journal of Cancer*, 45(2):228–247, January 2009.
- [30] A. B. Mariotto, A.-M. Noone, N. Howlader, H. Cho, G. E. Keel, J. Garshell, S. Woloshin, and L. M. Schwartz. Cancer survival: An overview of measures, uses, and interpretation. *JNCI Monographs*, 2014(49):145–186, November 2014.
- [31] J. M. Bland and D. G. Altman. Statistics notes: Survival probabilities (the Kaplan-Meier method). *BMJ*, 317(7172):1572–1580, December 1998.
- [32] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. Springer, New York, NY, 1 edition, May 2013.
- [33] Alice Zheng. *Feature Engineering for Machine Learning*. O’Reilly Media, Sebastopol, CA, April 2018.
- [34] I.-K. Yeo. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, December 2000.
- [35] B Srujana, Dhananjay Verma, and Sameen Naqvi. Machine learning vs. survival analysis models: a study on right censored heart failure data. *Communications in Statistics-Simulation and Computation*, pages 1–18, 2022.
- [36] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, April 2005.
- [37] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009.
- [38] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022.
- [39] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.
- [40] Anna Jenul, Stefan Schrunner, Bao Ngoc Huynh, and Oliver Tomic. RENT: A Python package for repeated elastic net feature selection. *Journal of Open Source Software*, 6(63):3323, 2021.
- [41] Anna Jenul and Stefan Schrunner. UBayFS: An R package for user guided feature selection. *Journal of Open Source Software*, 8(81):4848, 2023.
- [42] Max Kuhn. *caret: Classification and Regression Training*, 2022. R package version 6.0-93.
- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [44] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
- [45] Genlian Chen, Qiang Xu, Shengjun Qian, Zhan Wang, and Shicheng Wang. Survival analysis in gastrointestinal neuroendocrine carcinoma with bone metastasis at diagnosis. *Frontiers in Surgery*, 9, January 2022.
- [46] Ömer Komaç, Göksel Bengi, Özgül Sağol, and Mesut Akarsu. C-reactive protein may be a prognostic factor for the whole gastroenteropancreatic neuroendocrine tumor group. *World Journal of Gastrointestinal Oncology*, 11(2):139–152, February 2019.
- [47] Anna Nießen, Simon Schimmack, Marta Sandini, Dominik Fliegner, Ulf Hinz, Magdalena Lewosinska, Thilo Hackert, Markus W. Büchler, and Oliver Strobel. C-reactive protein independently predicts survival in pancreatic neuroendocrine neoplasms. *Scientific Reports*, 11(1), December 2021.
- [48] Patricia Freis, Emmanuelle Graillot, Pascal Rousset, Valérie Hervieu, Laurence Chardon, Catherine Lombard-Bohas, and Thomas Walter. Prognostic factors in neuroendocrine carcinoma: biological markers are more useful than histomorphological markers. *Scientific Reports*, 7(1), January 2017.
- [49] Niklas Gebauer, Maria Ziehm, Judith Gebauer, Armin Riecke, Sebastian Meyhöfer, Birte Kulemann, Nikolas von Bubnoff, Konrad Steinestel, Arthur Bauer, and Hanno M. Witte. The glasgow prognostic score predicts survival outcomes in neuroendocrine neoplasms of the gastro-entero-pancreatic (GEP-NEN) system. *Cancers*, 14(21):5465, November 2022.

- [50] Koji Amano, Isseki Maeda, Tatsuya Morita, Tomofumi Miura, Satoshi Inoue, Masayuki Ikenaga, Yoshihisa Matsumoto, Mika Baba, Ryuichi Sekine, Takashi Yamaguchi, Takeshi Hirohashi, Tsukasa Tajima, Ryohei Tatara, Hiroaki Watanabe, Hiroyuki Otani, Chizuko Takigawa, Yoshinobu Matsuda, Hiroka Nagaoka, Masanori Mori, and Hiroya Kinoshita. Clinical implications of c-reactive protein as a prognostic marker in advanced cancer patients in palliative care settings. *Journal of Pain and Symptom Management*, 51(5):860–867, May 2016.
- [51] Peter C. Hart, Ibraheem M. Rajab, May Alebraheem, and Lawrence A. Potempa. C-reactive protein and cancer—diagnostic and therapeutic insights. *Frontiers in Immunology*, 11, November 2020.
- [52] Shiva Shrotriya, Declan Walsh, Amy S. Nowacki, Cliona Lorton, Aynur Aktas, Barbara Hullihen, Nabila Benanni-Baiti, Katherine Hauser, Serkan Ayvaz, and Bassam Estfan. Serum c-reactive protein is an important and powerful prognostic biomarker in most adult solid tumors. *PLOS ONE*, 13(8):e0202555, August 2018.
- [53] Thomas E. Clancy, Tanya P. Sengupta, Jessica Paulus, Fawzia Ahmed, Mei-Sheng Duh, and Matthew H. Kulke. Alkaline phosphatase predicts survival in patients with metastatic neuroendocrine tumors. *Digestive Diseases and Sciences*, 51(5):877–884, May 2006.
- [54] Monica Ter-Minassian, Jennifer A Chan, Susanne M Hooshmand, Lauren K Brais, Anastassia Daskalova, Rachel Heafield, Laurie Buchanan, Zhi Rong Qian, Charles S Fuchs, Xihong Lin, David C Christiani, and Matthew H Kulke. Clinical presentation, recurrence, and survival in patients with neuroendocrine tumors: results from a prospective institutional database. *Endocrine-Related Cancer*, 20(2):187–196, January 2013.
- [55] Halfdan Sorbye, Eric Baudin, and Aurel Perren. The problem of high-grade gastroenteropancreatic neuroendocrine neoplasms. *Endocrinology and Metabolism Clinics of North America*, 47(3):683–698, September 2018.
- [56] Hege Elvebakken, Aurel Perren, Jean-Yves Scoazec, Laura H. Tang, Birgitte Federspiel, David S. Klimstra, Lene W. Vestermark, Abir S. Ali, Inti Zlobec, Tor Å. Myklebust, Geir O. Hjortland, Seppo W. Langer, Henning Gronbaek, Ulrich Knigge, Eva Tiensuu Janson, and Halfdan Sorbye. A consensus-developed morphological re-evaluation of 196 high-grade gastroenteropancreatic neuroendocrine neoplasms and its clinical correlations. *Neuroendocrinology*, 111(9):883–894, October 2020.

ISBN: 978-82-575-2062-5

ISSN: 1894-6402



Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no