Norwegian University
of Life Sciences

**Master's Thesis 2023 30 ECTS**
Faculty of Science and Technology

# Treatment Outcome Prediction in Locally Advanced Cervical Cancer: A Machine Learning Approach using Feature Selection on Multi-Source Data

Sina Rokhideh
Data Science

# Acknowledgements

I would like to express my sincere appreciation to the individuals who have assisted me during my academic journey and the completion of my master's degree:

First and foremost, I am deeply grateful to my thesis supervisor, Stefan Schrunner, for his unwavering guidance, encouragement, and support throughout my research. His invaluable insights and constructive feedback during our regular meetings have significantly influenced and shaped my research. I am truly thankful for his mentorship.

I would also like to extend my profound gratitude to my co-supervisors, Oliver Tomic and Cecilia Marie Futsæther, for sharing their knowledge and experiences with me. Their valuable perspectives and constructive criticism have helped refine my research approach and methodology.

I want to acknowledge the REALTEK faculty and staff at the Norwegian University of Life Sciences (NMBU) for providing a supportive academic environment and abundant knowledge and resources that have greatly assisted me in my studies.

Lastly, I would like to express my heartfelt thanks to my wife, Helia, for her unwavering love, support, and encouragement throughout my master's program. Without her constant understanding, patience, and selfless support, I would not have been able to complete this academic journey. I also want to convey my deep gratitude to my family, especially my mother, who has been a pillar of support from a distance. Her teachings have played a crucial role in shaping my mindset and inspiring me to remain motivated and diligent in pursuing my goals.

Sina Rokhideh
Ås, Norway - July 17, 2023

# Abstract

Cancer is a significant global health issue, and cervical cancer, one of the most common types among women, has far-reaching impacts worldwide. Researchers are studying cervical cancer from various perspectives, conducting thorough investigations, and utilizing novel technologies to gain a deeper understanding of the disease and its risk factors. Machine learning has increasingly found applications in cancer research due to its ability to analyze complex data relationships, recognize patterns, adapt to new information, and integrate with other technologies. By harnessing predictive machine learning models to anticipate treatment outcomes before commencing any therapies, healthcare providers might be able to make more informed decisions, allocate resources effectively, and provide personalized care.

Despite significant efforts in the scientific community, the development of accurate machine learning models for cervical cancer treatment outcome prediction faces several open challenges and unresolved questions. A major challenge in developing accurate prediction models is the limited availability and quality of data. The quantity and quality of data differ across various datasets, which can significantly affect the performance and applicability of machine learning models. Additionally, it is crucial to identify the most informative and relevant features from diverse data sources, including clinical, imaging, and molecular data, to ensure accurate outcome prediction. Moreover, cancer datasets often suffer from class imbalance. Addressing this issue is another essential step to prevent biased predictions and enhance the overall performance of the models.

This study aims to improve the prediction of treatment outcomes in patients with locally advanced cervical cancer by utilizing a multi-source dataset and developing different machine-learning models. The dataset includes various data sources, such as medical images, gene scores, and clinical data. A preprocessing pipeline is developed to optimize the data for training machine-learning models. The Repeated Elastic Net Technique (RENT) is also employed as a feature selection method to reduce dataset dimensionality, improve model training time, and identify the most influential features for classifying patients' treatment results. Furthermore, the Synthetic Minority Oversampling Technique (SMOTE) is used to address data imbalance in the dataset, and its impact on model performance is assessed.

The study's findings indicate that the available data exhibit promising capabilities in early predicting patients' treatment outcomes, suggesting that the developed models have the potential to serve as valuable auxiliary tools for medical professionals. Although the performance of the models remained relatively unchanged after implementing the RENT method, the models' average training time was reduced by over 8-fold in the worst case. Moreover, when imposing stricter feature selection criteria, clinical features were shown to have a more prominent role in predicting treatment results than other data sources. Ultimately, the study revealed that by balancing the dataset using the SMOTE technique, the average performance of specific models could be enhanced by up to 44 times.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Cancer is a complex and devastating disease that affects millions of people worldwide. It is an umbrella term referring to any disease where abnormal cells proliferate without limitation, exceeding normal limits and infiltrating nearby areas or disseminating to distant organs. This process plays a pivotal role in cancer-related deaths. Each type of cancer possesses unique features, and any tissue in the body can contribute to cancer development [5]. The scientific community focuses heavily on cancer research for several reasons, including [6]:

- Impact: Cancer is a leading cause of death globally and significantly impacts individuals, families, and societies.
- Complexity: Although the fundamental processes contributing to cancer development are similar, the specific regulations that govern it vary depending on the cancer type. Over the years, significant strides have been made in identifying the molecular structures responsible for triggering cancer development. However, understanding the distinct characteristics of each type of cancer requires in-depth research.
- Diagnostic and medicine advancements: Improved diagnostic tools and techniques enable early cancer detection and the development of better therapies and medicine that are more targeted, effective, and have fewer side effects. The developments in this regard are expected to enhance the quality of life for those affected by the disease.

## 1.1 Motivation and Background

Among women, cervical cancer is the fourth most common type of cancer and the fourth leading cause of cancer-related deaths, with approximately 604,000 fresh cases and 342,000 deaths documented globally in 2020. Cervical cancer is diagnosed more often than any other type in 23 countries (Figure 1.1.A) and is responsible for the majority of cancer fatalities in 36 countries (Figure 1.1.B), mainly located in sub-Saharan Africa, Melanesia, South America, and Southeastern Asia [1].

Various risk factors, such as tobacco use, alcohol consumption, an unhealthy diet, physical inactivity, and air pollution, can influence any form of cancer. In 2018, approximately 13% of cancer cases worldwide were linked to infections that could potentially cause cancer, including Helicobacter pylori, human papillomavirus (HPV), hepatitis B virus, hepatitis C virus, and Epstein-Barr virus. While specific types of HPV heighten the chances of developing cervical cancer, infection with HIV raises the risk of this cancer by six times. [7].

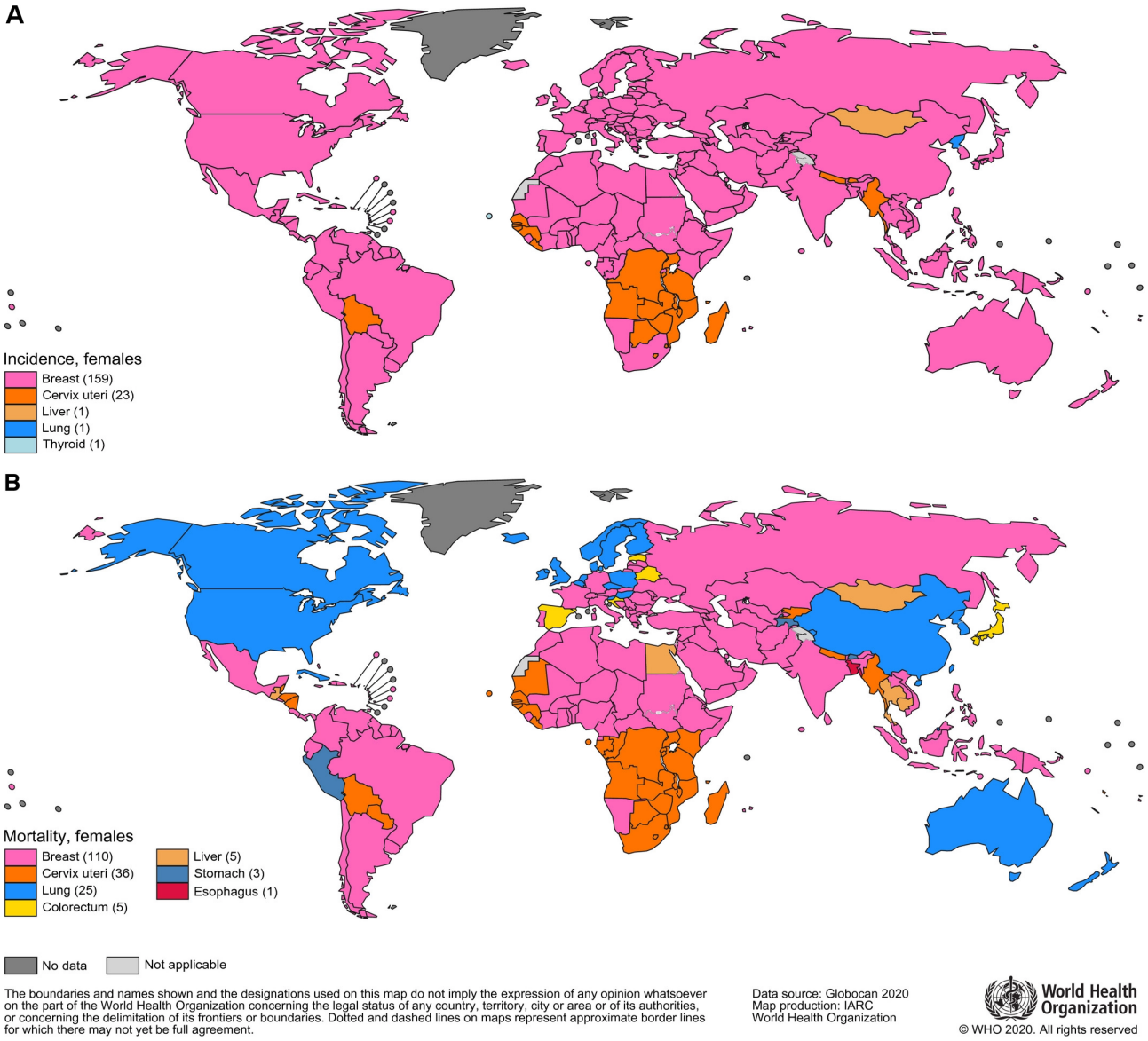



Figure 1.1: Most Common Type of (A) Cancer Incidence and (B) Cancer Mortality in 2020 in Each Country Among Women. The numbers of countries represented in each ranking group are included in the legend [1].

The considerable impact of cervical cancer on the economy and well-being emphasizes the necessity for interventions that can prevent and treat this condition. According to research conducted on cervical cancer patients in the United States from 2006 to 2015, these patients' average yearly medical expenses were significantly greater compared to healthy people, with cervical cancer patients spending an average of $10,031 per year while healthy individuals only spent $4,913. Cervical cancer patients experienced a substantial decline in their quality of life compared to those without cancer. Those affected by cervical cancer were more prone to experiencing limitations in various areas, including physical activities, social interactions, mental well-being, and overall health. Nearly all measures of quality of life demonstrated a greater level of impairment among cervical cancer patients compared to healthy individuals [8].

In a report titled "Roadmap to accelerate the elimination of cervical cancer as a public health problem in the WHO European Region 2022–2030," the World Health Organization (WHO) suggests that nations can expedite the process of eradicating cervical cancer as a public health

concern by embracing fundamental principles, adopting innovative methods, and allocating resources to crucial shifts [9]. Besides taking preventive steps like HPV vaccination or screening and treating pre-cancerous lesions, other key priorities include:

- Ensuring clear information is given to patients about their condition and possible treatment side-effects and involving them in decision-making, which may include discussions on preserving fertility and reproductive health.
- Providing equal access to top-notch diagnostic services and high-quality, appropriate treatments (such as surgery, chemotherapy, and radiotherapy – both external beam and brachytherapy) for all stages of the disease. [9].

The successful treatment of locally advanced cervical cancer often necessitates a combination of external beam radiotherapy, brachytherapy, and cisplatin-based chemotherapy [10]. Identifying patients at high risk of treatment failure is crucial for customizing treatment to match each individual's risk profile. Researchers are actively working on several fronts to achieve this goal, encompassing:

- Biomarkers and Precision Medicine: Biomarkers are quantifiable biological markers that aid in diagnosing cancer, forecasting treatment response, and tracking disease advancement. Researchers are actively identifying and creating novel biomarkers to improve the early detection and diagnosis of cancer while also guiding personalized treatment approaches. Precision medicine seeks to customize therapies according to an individual's distinctive genetic profile, tumor attributes, and other relevant factors [11].
- Imaging Technologies: Advancements in imaging technologies are crucial in cancer diagnosis, staging, and treatment monitoring. Researchers are improving existing imaging techniques such as magnetic resonance imaging (MRI), computed tomography (CT), and positron emission tomography (PET), as well as developing novel imaging modalities like molecular imaging for more accurate and detailed cancer imaging [12].
- Artificial Intelligence (AI): AI and machine learning (ML) algorithms are being employed to analyze vast amounts of data, including genomic profiles, medical imaging, and patient records. This assists in improving cancer diagnosis, predicting treatment response, and identifying patterns that aid in developing personalized treatment strategies [13].

## 1.2 Objective

The primary question this research aims to answer is how accurately the overall treatment outcome of patients with locally advanced cervical cancer can be predicted using a multi-source dataset and machine-learning classification models. To address this question, the following steps will be taken:

A) A data preprocessing pipeline will be created to optimize the dataset for training the machine learning models.

B) Three machine-learning classification models will be developed to identify the optimal hyperparameters and obtain the models' scores.

C) The results will be visualized and analyzed using an appropriate evaluation metric.

This research also includes two secondary objectives:

1. Employing the RENT feature selection technique with the aim of a) reducing the dataset dimensions and assessing its impact on model performance and training time, and b) identifying the most influential features in the dataset to obtain the best models' performance in predicting patients' treatment outcomes.
2. Balancing the sample distribution in the dataset using the SMOTE class balancing method and assessing its impact on the models' performance.

## 1.3 Structure of the Thesis

The thesis introduction provides an overview of the subject being studied, including its background, the motivation for conducting the research, and the overarching goals of the study. The theory section delves into the concepts related to various data sources (including medical images, clinical examinations, and gene expressions), machine learning, classification models, and appropriate evaluation methods. The third chapter discusses the data sources utilized in this research, data preparation for machine learning models, and the methodologies employed for conducting the experiments and analyzing the outcomes. In chapter four, the study findings are revealed and thoroughly examined. The discussion chapter delves deeper into the problem and explores potential avenues for further research. Finally, the sixth chapter draws a conclusion from the study.

# Chapter 2

# Theory

## 2.1 Medical Imaging

Today, various imaging techniques, such as magnetic resonance imaging (MRI), computed tomography (CT), or positron emission tomography (PET), are employed globally in evaluating cervical cancer. CT is commonly used for evaluating and staging cervical cancer due to its widespread availability, but it has limitations in accurately detecting the spread of cancer within the cervix. Recent studies using advanced CT techniques have shown slightly improved results. However, the use of CT for staging is currently restricted to patients with advanced disease or those who cannot undergo MRI. Hybrid imaging methods like PET-CT or PET-MRI are more effective than traditional approaches in identifying metastatic lymph nodes, offering high diagnostic accuracy. However, their role in the initial evaluation of cervical cancer is still uncertain. PET-CT demonstrates moderate accuracy in local staging, while PET-MRI shows promise in evaluating primary tumors but requires further investigation with larger patient populations [14].

Nowadays, MRI is the preferred imaging technique for assessing the extent of cervical cancer due to its superior ability to differentiate between cancerous and normal tissues, thanks to its high contrast resolution [14]. Traditional contrast-enhanced MRI provides a single image of tumor enhancement following contrast injection, offering valuable anatomical data but lacking functional information. Dynamic contrast-enhanced MRI (DCE-MRI) involves rapid MRI sequences captured before, during, and after the swift intravenous injection of a gadolinium-based contrast agent, creating a movie-like data structure. DCE-MRI allows for the visualization of both physiological and morphological changes. In DCE-MRI, tumors typically exhibit swift, intense enhancement and a relatively quick washout compared to healthy tissues [15]. Figure 2.1 displays instances of cross-sectional time series of the heart, kidney, and prostate utilizing DCE-MRI.

The effectiveness of DCE-MRI in forecasting tumor responses has been extensively studied. Tumors exhibiting low perfusion traits are linked to tumor hypoxia, an unfavorable prognostic indicator in cervical cancer [16]. On the other hand, tumors with higher oxygen levels may be more receptive to radiation and chemotherapy, resulting in a more favorable prognosis [17]. DCE-MRI can be employed to anticipate cervical cancer treatment outcomes [16] [18] [19] and demonstrate progressive alterations in tumor perfusion throughout therapy [20].

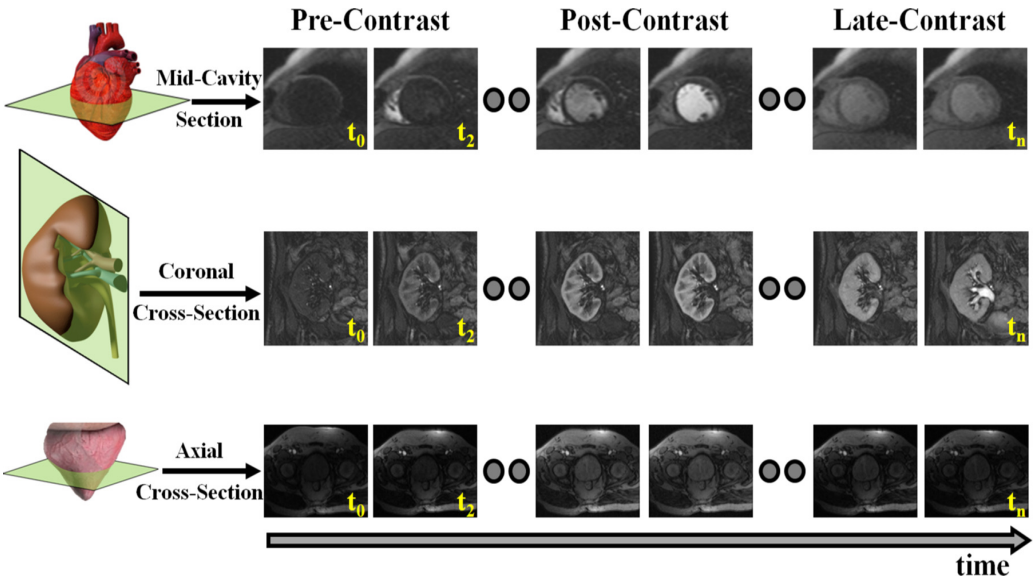



Figure 2.1: Cross section DCE-MRI images of the heart, kidney, and prostate at different time instants before and after administering the contrast agent into the bloodstream. Source: [2]

## 2.2 Pharmacokinetic Analysis

Once the contrast agent has been administered to the targeted tissue, the signal intensity changes on DCE-MR images, showing a progressive decrease in signal intensity over time, can be evaluated through either semi-quantitative or quantitative means. Semi-quantitative assessment involves calculating parameters from the time-intensity curve, which is straightforward and enables simple calculation of signal intensity changes. However, these parameters are not easily applicable to different MR scanners or pulse sequences because baseline signal levels vary between systems, and the measurements do not accurately represent the concentration of contrast in the region of interest. Additionally, semi-quantitative methods lack inherent physiological significance [17].

In contrast, the quantitative evaluation, which involves converting signal intensity to concentration and estimating parameters that describe physiology, relies on contrast agent concentration curves over time and utilizes pharmacokinetic models to calculate permeability constants [21]. Typically, alterations in signal intensity, i.e., relative signal intensity (RSI), can be measured over time [22]:

$$RSI(t) = \frac{SI(t) - SI(0)}{SI(0)} \tag{2.1}$$

where $SI(t)$ represents the signal intensity at a given time point $t$, and $SI(0)$ indicates the signal intensity prior to the injection of the contrast agent. Pharmacokinetic models can be applied to the RSI equation using Levenberg-Marquardt's least squares minimization [23]. The Brix pharmacokinetic model [24] describes RSI as:

$$RSI(t) = A_{Brix} \frac{K_{ep}}{K_{el} - K_{ep}} (e^{-K_{ep}t} - e^{-K_{el}t}) \tag{2.2}$$

where $A_{Brix}$ is the amplitude of contrast uptake, $K_{ep}$ is the transfer rate of the contrast agent from the tumor tissue into the plasma, and $K_{el}$ is the tracer washout rate from the bloodstream. The fitting procedure permits unrestricted variations of all three parameters, except for the constraints that $A_{Brix}$, $K_{ep}$, and $K_{el}$ must be greater than or equal to 0. [22]. RSI can also be

described with the Tofts pharmacokinetic model [25]:

$$RSI(t) = K^{trans} AIF(t) \otimes e^{-\frac{K^{trans}}{V_e}t} \tag{2.3}$$

where $K^{trans}$ denotes the transfer rate of the contrast agent from blood to the extravascular extracellular space (EES), $AIF$ represents the contrast agent concentration at a given time $t$, and $V_e$ defines the fraction of EES volume. With the exception of the constraints that $0 <= V_e <= 1$ and $K^{trans} >= 0$, both parameters can vary without restrictions during the fitting process [26].

Studies suggest that the pharmacokinetic parameters obtained from DCE-MRI investigations have the potential for predictive applications in managing cervical cancer. In their study, Andersen et al. [27] demonstrate that by employing two pharmacokinetic parameters, $K^{trans}$ and $V_e$, it is possible to categorize intratumoral regions with similar vascularization into three groups. Additionally, they found that the volume fraction of one of these groups was significantly linked to primary tumor control, as evidenced by a log-rank survival test. Specifically, patients with a high volume fraction of voxels from DCE-MR images were observed to have a reduced risk of treatment failure.

Semple et al. [28] show that integrating radiologic assessment with pharmacokinetic modeling, particularly the $K^{trans}$ parameter, and applying it to DCE-MRI data prior to chemoradiation treatment made it feasible to anticipate over 88% of the variation in therapy response. This approach could improve therapy response prediction, leading to the development of personalized and more effective therapy plans for these specific patients with locally advanced cervical cancer.

## 2.3 Clinical Examination and Gene Expression

The FIGO classification, a globally accepted staging method for cervical cancer, exclusively depends on clinical examination for determining tumor stage. The present FIGO classification acknowledges imaging methods as a supplementary tool for staging cervical cancer [29]. Different studies have demonstrated that imaging outperforms clinical examination alone in accurately assessing cervical carcinoma, especially in the initial cancer stages [30] [31] [32] [33] [34].

However, employing imaging techniques without clinical evaluation in advanced stages of cancer, where the tumor extends beyond the uterine cervix, necessitates additional research. Notably, tumors that have spread into the surrounding supporting tissues, known as parametria (stage 2B from the Figo classification), are significant, as they are typically considered a contraindication for surgical intervention [35]. Multiple studies indicate that, on average, the specificity for identifying parametrial invasion is higher in clinical examination than in MRI [36] [37] [38] [39]. Sodeikat et al. [35] provide evidence that when a gynecologic oncologist performs a clinical assessment of the parametrium during general anesthesia, accompanied by MR images displayed in the operating room, it yields higher accuracy in detecting parametrial tumor involvement in cervical cancer compared to relying solely on MR imaging.

On the other hand, Solid tumors often exhibit hypoxia, frequently linked to unfavorable outcomes across various cancer types and treatment methods such as radiotherapy, chemotherapy, surgery, and potentially immunotherapy. Hypoxia is not evenly distributed within tumors but varies among different tumors and within the same tumor. Moreover, there are significant variations in the prevalence and, importantly, the hypoxia level among patients. Therefore, the

hypoxia level could play a crucial role in determining how effective cancer treatments such as radiation, chemotherapy, and targeted molecular drugs are, and it can significantly impact the therapeutic outcomes [40]. Several research investigations have been carried out to explore the impact of varying hypoxia levels on the progression of cancer and treatment results [41] [42] [43].

The hypoxia level in a tumor can be achieved indirectly by assessing gene expression signatures, which reflect the cellular response to low oxygen levels. This approach holds the potential for developing classifiers that can assist in medical decision-making. Gene expression assays are more straightforward to standardize compared to standardizing images across different MR machines [44]. Gene signatures provide insights into the underlying resistance mechanisms in individual tumors, aiding in selecting personalized, combined radiotherapy regimes and optimizing targeted therapies [45].

Additionally, multigene signatures have proven valuable in guiding treatment choices for various cancer types, including breast and prostate cancer [46] [47]. Halle et al. [48] suggest that DCE-MRI can detect patients with hypoxia-related chemoresistance by associating hypoxia-related gene sets with a previously established prognostic DCE-MRI parameter ($A_{Brix}$). This could potentially prompt a shift in treatment strategy, advancing the move towards a more tailored approach to therapy.

## 2.4 Machine Learning

Machine learning (ML) is a form of artificial intelligence (AI) that enables systems to learn from data and recognize patterns without much human interaction. Machine learning's impact on daily life has been substantial, and it has the potential to improve healthcare accuracy, forecasting, and quality of care to a significant extent. Current developments in ML and AI support physicians and analysts in identifying healthcare trends, developing models for predicting illnesses, and performing their roles more effectively [49].

### 2.4.1 Learning Techniques and Algorithms

Most AI and ML algorithms rely on three learning techniques, Supervised, Unsupervised, and Reinforcement learning. Supervised learning is employed in training classification and prediction algorithms by utilizing past examples or outputs. A critical aspect of this approach is that the training set involves both features and corresponding predictions or outcomes. Supervised learning entails deriving knowledge from the features in the training set to construct a model that can accurately predict outcomes and subsequently employ this model to predict results using new features in the testing data set. Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), and Artificial Neural Networks (ANN) are some machine learning algorithms implementing supervised learning techniques [50].

Unsupervised learning is another ML-based method commonly used for data analysis and clustering. Unlike supervised learning, unsupervised learning is not typically focused on prediction applications but on data analysis, stratification, and reduction. Unsupervised clustering methods primarily aim to group unclassified or uncategorized data into independent clusters using algorithms. Although data preprocessing and feature extraction are typically performed before

the input in most machine learning types, unsupervised learning involves identifying underlying relationships or features in the data and grouping them according to their similarities. Some examples of unsupervised learning approaches include K-Means Clustering, Principal Component Analysis (PCA), and Hierarchical Clustering [51].

Reinforcement learning is a distinct learning method that differs from both supervised and unsupervised learning. It operates based on rewards and generates a strategy for addressing a specific problem. Reinforcement learning techniques can impact their surroundings, seek to optimize the error criterion, and are considered the most comparable form of learning to that observed in humans and animals. The Recurrent Neural Network (RNN) is an example of a neural network commonly used in reinforcement learning [52]. As mentioned, many machine learning algorithms have been explored in different healthcare research. However, the implementation of reinforcement learning in healthcare applications is currently limited due to requirements for structure, heterogeneous data, definition, and implementation of rewards, as well as extensive computational resources. Nonetheless, reinforcement learning still holds immense potential to make significant advancements in healthcare.

### 2.4.2 ML and Medical Imaging

Medical imaging has undergone significant advancements with the help of machine learning due to the digital nature of data and the availability of structured data formats. Machine learning-based approaches have been applied to various imaging modalities such as MRI, CT, and X-Ray. Multiple models based on machine learning have been developed for identifying tumors [53], lesions [54], tears [55], and fractures [56].

Shao et al. [57] conducted a study wherein they utilized pharmacokinetic parameters obtained from DCE-MR images. They developed models for predicting cervical cancer by employing machine learning techniques like SVM and deep learning, including a hybrid model called APITL built with convolutional neural networks (CNNs). These models could predict cervical cancer with an accuracy of over 94%. Torheim et al. [58] devised a machine-learning approach to automatically segment and outline cervical cancer tumors. They merged data from multiparametric MRI and demonstrated that their technique achieved excellent sensitivity and specificity. Remarkably, their method required no input or adjustments from users, making it a valuable tool for radiologists.

### 2.4.3 Overfitting

There is a persistent problem in machine learning known as "overfitting." This occurs when a model cannot generalize well from observed data to new, unseen data. As a result, the model may perform well on the data it was trained on but poorly on new data. The reason for this is that an overfitted model has difficulty handling information that is different from what it was trained on. Instead of learning the underlying patterns in the data, overfitted models tend to memorize all the data, including any noise or irrelevant information [59].

The Bias-Variance dilemma refers to the conflicting requirement for models to predict both training and unseen samples well. Bias evaluates the overall deviation of predictions from the

correct values when rebuilding the model on different training datasets. A model with high bias oversimplifies the underlying relationships in the data and consistently makes systematic errors. On the other hand, Variance quantifies the stability or variability of the model's predictions when trained multiple times on various subsets of the training data. A model with high variance demonstrates high flexibility and can fit the training data accurately. However, it may struggle to generalize to new, unseen data. The bias-variance dilemma presents a challenge because optimizing for one often means sacrificing the other [60].

Regularization is a method employed in machine learning to avoid overfitting and enhance the overall performance of a model by mitigating its tendency to become too specialized to the training data. Taking the linear regression model as an example, where there are $n$ predictors $x_1$, $x_2$, ..., $x_n$, the anticipated outcome $\hat{y}$ is determined by:

$$\hat{y} = w_0 + w_1 x_1 + ... + w_n x_n \tag{2.4}$$

Where $w_0$ to $w_n$ are the weights or coefficients of the predictors. The model fitting process generates a coefficient vector denoted as $w = (w_0, w_1, ..., w_n)$. With that being mentioned, regularization involves incorporating a penalty component into the model's cost function. The cost function is a mathematical function that quantifies the discrepancy between the predicted values of a model and the actual values in the training data. The introduced penalty term encourages the model to find a balance between fitting the training data well and avoiding excessive complexity [60]. Various regularization techniques are frequently employed in machine learning, including:

1. L1 regularization, also known as Lasso regularization, incorporates the total absolute values of the coefficients into the cost function during model training. This regularization technique promotes solutions with fewer non-zero coefficients, effectively conducting feature selection by driving some coefficients to precisely zero. The L1 regularization can be written as follows [60]:

   $$L1 : \|w\|_1 = \lambda \sum_{j=1}^{n} |w_j| \tag{2.5}$$

   Here, the hyperparameter $\lambda$ represents the degree of regularization. This implies that as the value of $\lambda$ is raised, the regularization effect becomes stronger, causing the model's weights to decrease in magnitude. The acceptable range for $\lambda$ can differ based on the implementation or framework employed. However, a typical range for lambda is $[0, +\infty)$.

2. L2 regularization, also called Ridge regularization, introduces the sum of squared coefficients into the objective function. This regularization method encourages coefficients to have smaller and smoother magnitudes by reducing their overall values. The equation for L2 regularization can be represented as follows [60]:

   $$L2 : \|w\|_2^2 = \lambda \sum_{j=1}^{n} w_j^2 \tag{2.6}$$

   Likewise, the hyperparameter $\lambda$ controls the regularization strength and ranges typically from 0 to $+\infty$.

3. Elastic Net integrates L1 and L2 regularization, achieving a trade-off between selecting relevant features and shrinking coefficients. It is particularly beneficial when dealing with

predictors that are correlated. The following notation gives the elastic net regularization [3]:

$$ElasticNet : \gamma(\alpha L1 + (1 - \alpha)L2) \tag{2.7}$$

Where $\gamma$ is the regularization parameter controlling the overall amount of regularization applied, and $\alpha$ is the mixing parameter determining the balance between L1 and L2 regularization. $\gamma$ and $\alpha$ are typically set within the range of [0, 1]. Elevating the $\gamma$'s value from 0 to 1 leads to a more robust regularization, while for the $\alpha$, 0 represents pure L2 regularization, and 1 signifies pure L1 regularization.

## 2.5 Classification Models

As explained in Section 2.4.1, classification models in supervised learning are algorithms employed to classify or categorize data into predefined classes. These models are trained on labeled datasets, where each instance is assigned a known class label. A classification model aims to establish a connection between input features and their respective class labels, enabling the prediction of class labels for new, unseen instances. This section delves into three prominent examples of classification models extensively employed in machine learning.

### 2.5.1 Logistic Regression

Logistic regression, a linear model used for binary classification, is suitable when dealing with linearly separable classes and is simple to implement. It stands as one of the most widely employed algorithms in the industry for classification tasks, even capable of handling multi-class classification scenarios. It is vital to emphasize that logistic regression, despite its name, functions as a classification model rather than a regression model [60].

To explain the idea behind logistic regression as a probabilistic model, let us first introduce the odds ratio: the odds in favor of a particular event. The odds ratio, denoted as the ratio of the event occurring to not occurring ($\frac{p}{(1-p)}$), captures the likelihood of a positive event, which does not necessarily imply a good outcome but refers to the event we want to predict. For instance, it could be the probability that a patient has a certain disease, where the positive event represents the class label $y = 1$. The logit function can be defined as the natural logarithm of the odds ratio or the log-odds [60]

$$\text{logit}(p) = \log \frac{p}{(1 - p)} \tag{2.8}$$

By applying the logit function, we transform input values ranging from 0 to 1 into the entire real-number range. This transformation facilitates the expression of a linear relationship between feature values and the log-odds

$$\text{logit}\Big(p(y = 1|x)\Big) = w_0x_0 + w_1x_1 + \ldots + w_mx_m = \sum_{i=0}^{m} w_ix_i = w^Tx \tag{2.9}$$

where $x_0$ to $x_m$ represent the feature values, $w_0$ to $w_m$ indicate the weights assigned to those features, and the term $p(y = 1|x)$ denotes the conditional probability that a given sample is assigned to class 1 given its features [60].

Our primary focus lies in predicting the probability that a specific sample belongs to a particular class, which entails the inverse of the logit function. This inverse is called the logistic sigmoid function, or the sigmoid function ($\phi(z)$), due to its distinct S-shape (as illustrated in Figure 2.2).

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{2.10}$$

The net input, denoted as $z$, represents the linear combination of weights and sample features, where $z = w^T x = w_0 x_0 + w_1 x_1 + \ldots + w_m x_m$. It is worth noting that $w_0$ denotes the bias unit, an additional input value set equal to 1. A threshold function can then be employed to convert the predicted probability into a binary outcome ($\hat{y}$) [60].

$$\hat{y} = \begin{cases} 1 & \text{if } \phi(z) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

The excellent capability of logistic regression to predict the probability of a patient having a specific disease based on certain symptoms contributes to its widespread popularity within medicine [60].

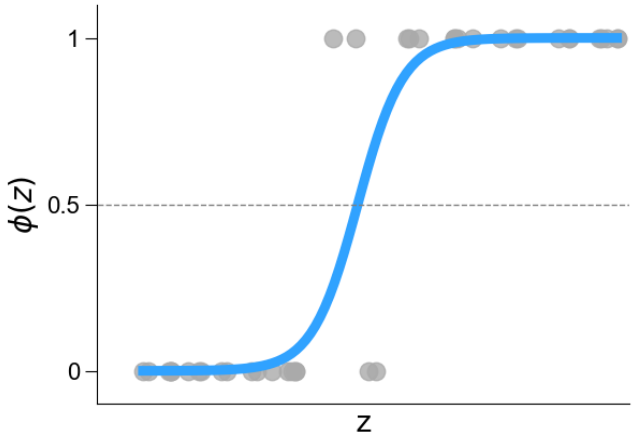



Figure 2.2: The Logistic Sigmoid Function Curve maps real-number inputs onto a bounded range of [0, 1], making it a powerful mathematical tool to predict binary outcomes. The sigmoid function (equation 2.10) exhibits an S-shaped curve, characterized by its property of asymptotically approaching 1 as the net input (z) tends towards positive infinity and approaching 0 as the net input tends towards negative infinity.

#### 2.5.1.1 Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a popular optimization algorithm employed in machine learning to train models such as logistic regression. It is a modified version of Gradient Descent designed to determine the best model parameters by minimizing a cost function. In the traditional Gradient Descent, the algorithm calculates the gradients of the cost function, which are the partial derivatives of the function concerning its parameters. It takes into account all training samples in the dataset and adjusts the model parameters based on the average of these gradients. However, this approach can be computationally expensive, especially with large datasets.

In contrast, SGD takes a different approach by randomly selecting a single training example or a small subset of examples (known as a mini-batch) for gradient computation at each iteration. The model's parameters are then updated based on this estimated gradient. This process is repeated for a fixed number of iterations or until convergence. In the context of SGD, an epoch refers to a complete pass through the entire training dataset during the training process. In other words, an epoch is completed when the algorithm has processed each training example or mini-batch once. The maximum number of epochs is a hyperparameter determining how often the SGD algorithm will iterate over the entire training dataset.

### 2.5.2 Random Forest

If interpretability is important, decision tree classifiers serve as appealing models. As the term "decision tree" implies, this model operates by sequentially dividing our data using a binary split based on feature values. The decision tree begins with the complete dataset and chooses the optimal feature and threshold to divide the data, forming child nodes. This process continues until a stopping criterion is met. The leaf nodes represent the final predictions based on the majority class or mean value. When making a prediction, the tree follows the splitting rules from the root to the leaf node associated with the sample's features [60]. The concept of the decision tree is visualized in Figure 2.3.

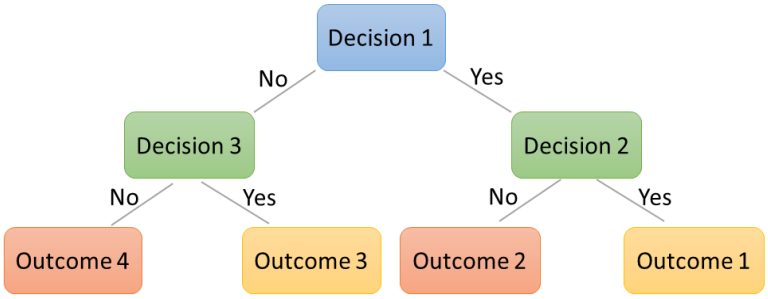



Figure 2.3: The decision tree concept. This model acquires a sequence of questions to deduce the class labels of the samples by analyzing the features present in our training dataset.

A random forest (Figure 2.4) can be seen as a collection of decision trees aiming to mitigate the high variance of individual trees by averaging their results. This approach creates a more robust model with improved generalization and reduced vulnerability to overfitting. In fact, Random Forest generates multiple subsets of the original training data using a technique called bagging, where each subset, known as a bootstrap sample, is used to train an independent decision tree. This approach introduces variability by training trees on different data subsets, which diminishes the risk of overfitting to specific patterns or noise in the data. Random Forest also employs random feature selection, whereby a subset of features is randomly chosen for each tree. This process promotes diversity among the trees, reducing their reliance on particular features and enabling them to capture different data aspects, resulting in a more robust model. Random forests have become widely popular in machine-learning applications over the past decade due to their strong classification performance and scalability [60].

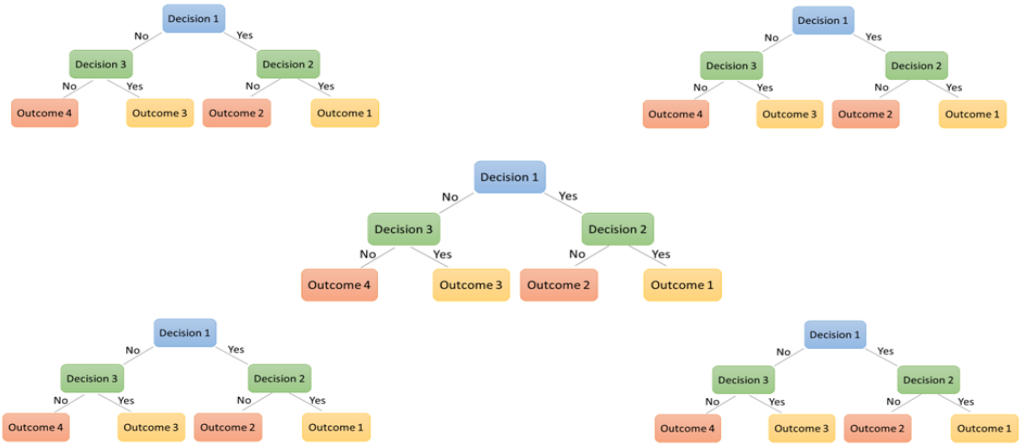



Figure 2.4: The random forest concept. This model is an ensemble learning method that combines multiple decision trees. Each decision tree in the random forest is built independently using a random subset of the training data. This process introduces randomness into the model and helps reduce overfitting.

The random forest algorithm can be summarized in four straightforward steps:

1. Randomly select a subset of $n$ samples (with replacement) from the training set.
2. Create a decision tree using the chosen subset. At each node:
   (a) Randomly pick $d$ features without repetition.
   (b) Split the node using the feature that yields the best split based on the objective function, such as maximizing information gain.
3. Repeat steps 1 and 2 $k$ times.
4. Combine the predictions of each tree and assign the class label based on the majority vote, which can also be used to obtain probability distribution over the classes.

Although not as interpretable as decision trees, random forests have a distinct advantage when selecting hyperparameters. Unlike decision trees, we do not need to worry extensively about choosing optimal hyperparameter values for random forests. The ensemble nature of random forests makes them robust to noise introduced by individual decision trees, reducing the need for pruning. In practice, the main parameter we need to focus on is the number of trees ($k$) chosen for the random forest (step 3). Generally, increasing the number of trees enhances the performance of the random forest classifier, albeit at the cost of increased computational resources [60].

Decreasing the bootstrap sample size ($n$ in step 1) can increase the diversity among the individual trees. This is because the probability of including a specific training sample in the bootstrap sample diminishes. Consequently, reducing the size of the bootstrap samples enhances the randomness of the random forest and aids in mitigating overfitting. However, smaller bootstrap samples tend to result in a lower overall performance of the random forest, with a reduced gap between training and test performance, indicating subpar test performance overall. On the other hand, increasing the bootstrap sample size can exacerbate overfitting. The bootstrap

samples and individual decision trees become more similar as they learn to fit the original training dataset closely.

Random Forest incorporates a criterium (in step 2.b) that assesses the splitting quality during the construction of decision trees in the forest. This is crucial in guiding the algorithm's evaluation and selecting the optimal feature for dividing the data at each tree node. Random forests commonly employ two well-known criteria for this purpose [60]:

- Gini impurity: The Gini impurity measures the probability of misclassifying a randomly chosen element in the dataset. A low Gini impurity indicates that the elements within a node predominantly belong to a single class. The criterion minimizes the Gini impurity by selecting the feature that produces the purest splits. In a binary classification scenario, the Gini impurity of a node is calculated using the following formula:

$$\text{Gini(node)} = 1 - p(A)^2 - p(B)^2 \tag{2.12}$$

  In this equation, $p(A)$ represents the proportion of elements of class A within the node, while $p(B)$ represents the proportion of elements of class B.

  The Gini impurity value spans from 0 to 0.5, where 0 denotes a node with complete purity (all elements belonging to the same class), and 0.5 signifies a completely impure node (an equal distribution of elements from both classes). When determining the optimal feature to split the data at a specific node, the algorithm calculates the Gini impurity for each potential split. It then computes the information gain associated with that split which is the difference between the current Gini of the node and the weighted average of the Ginies of the child nodes resulting from the split. By selecting the feature with the highest information gain, the algorithm identifies the one that would lead to the most substantial reduction in Gini impurity when generating the splits.

- Entropy: Entropy measures the level of impurity or disorder in a set of elements. In the context of decision trees, it quantifies the uncertainty in the target variable given the values of a specific feature. The criterion based on entropy aims to minimize the information gain, representing the reduction in entropy achieved by splitting a particular feature. Considering a binary classification problem, the entropy of a node is calculated using the following formula:

$$\text{Entropy(node)} = -p(A) \times log_2\, p(A) - p(B) \times log_2\, p(B) \tag{2.13}$$

  Where $p(A)$ is the proportion of elements belonging to class A in the node, and $p(B)$ is the proportion of elements belonging to class B.

  The entropy value ranges from 0 to 1, where 0 represents a pure node (all elements belong to the same class) and 1 represents a completely impure node (an equal distribution of elements from both classes). When deciding on the best feature to split the data at a particular node, the algorithm calculates the entropy for each possible split and then computes the information gain associated with that split. When creating the splits, the algorithm selects the feature with the highest information gain, indicating the feature that would lead to the most significant reduction in entropy or impurity.

### 2.5.3 Support Vector Machines

The Support Vector Machine (SVM) is a widely used and robust learning algorithm. Its main goal is to find the best hyperplane that separates data points into different classes, maximizing the margin, which is the distance between the decision boundary and the closest training samples of each class. The explained max-margin concept does not allow misclassification, and by doing so, SVM aims to improve generalization, avoid overfitting and handle unseen data more effectively. Support vectors are crucial in SVM and refer to the data points nearest to the decision boundary or within the margin. These points are vital in defining the hyperplane, making SVM memory-efficient and suitable for high-dimensional data [60]. Figure 2.5 depicts the concept of a support vector machine.

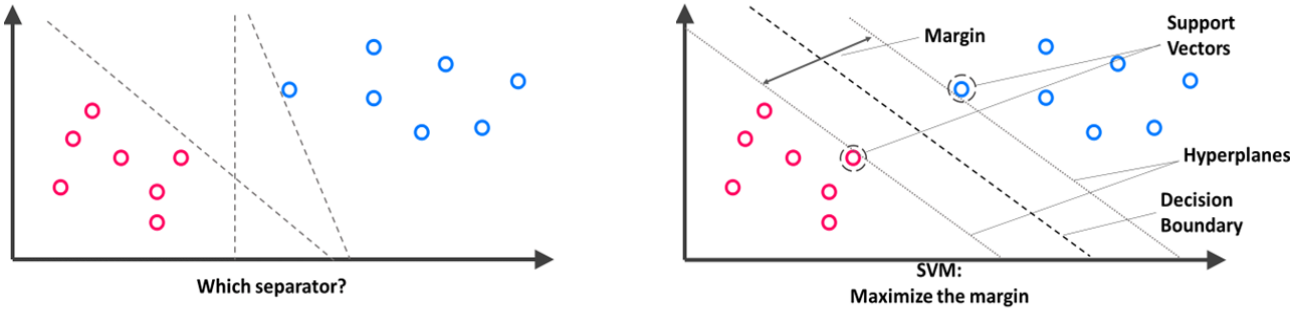



Figure 2.5: Visualization of the SVM concept showcasing the fundamental elements of the SVM algorithm, including decision boundary, hyperplanes, margin, and support vectors.

Furthermore, SVM incorporates a regularization parameter, denoted as $C$, which controls the balance between maximizing the margin and minimizing classification errors. As shown in Figure 2.6, A smaller $C$ value widens the margin but may misclassify some training samples, meaning it may struggle to capture the underlying patterns and exhibit high bias (underfitting). In comparison, a larger $C$ value aims to classify all training samples accurately, potentially resulting in a narrower margin and causing the model to become overly sensitive to the training data (overfitting) [60].

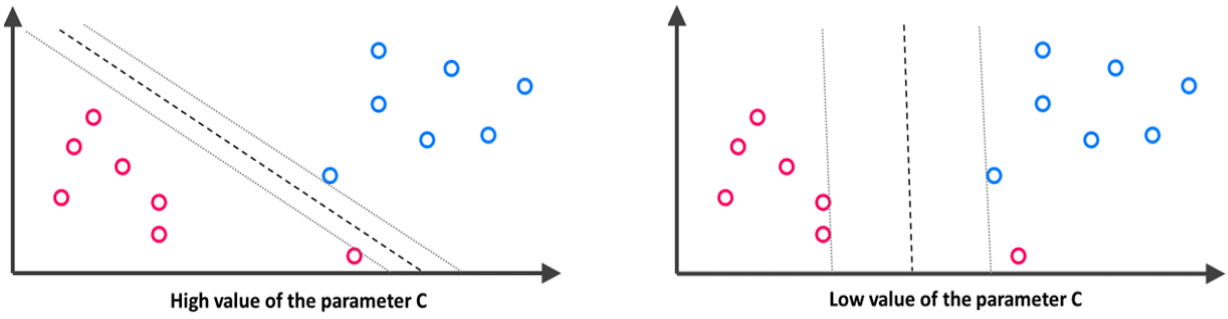



Figure 2.6: The effect of the regularization parameter, $C$, on the decision boundary in the SVM algorithm. A smaller value of $C$ leads to a wider margin between the classes. However, a larger value of $C$ aims to classify all training examples correctly.

For handling non-linearly separable data, SVM utilizes the kernel trick. The main idea behind the kernel trick is to implicitly map the original input data into a higher-dimensional feature space where the data becomes linearly separable, even if it was not linearly separable in the

original input space. This technique uses a kernel function that calculates the similarity or inner product between data points in the transformed feature space. It measures how similar or dissimilar two data points are in the new space. Using the kernel function, the SVM algorithm can implicitly learn a decision boundary that maximally separates the data points belonging to different classes. Commonly used kernel functions include [61]:

- Linear Kernel: The linear kernel represents the original feature space and is equivalent to no kernel transformation.
- Radial Basis Function (RBF) Kernel: The RBF kernel is popular as it can effectively represent complex non-linear relationships. It converts the data into a space with infinite dimensions, employing a Gaussian distribution centered around each support vector. Mathematically, the RBF kernel function can be defined as: [60]:

$$K(x^{(i)}, x^{(j)}) = e^{-\gamma \|x^{(i)} - x^{(j)}\|^2} \tag{2.14}$$

  Where $x^{(i)}$ and $x^{(j)}$ represent two data points in the original input space, $\|x^{(i)} - x^{(j)}\|^2$ denotes the Euclidean distance or squared norm between $x^{(i)}$ and $x^{(j)}$, and $\gamma$ is a parameter that controls the width of the Gaussian distribution. A smaller $\gamma$ value results in a broader distribution and smoother decision boundaries, potentially leading to underfitting. Conversely, a larger $\gamma$ value narrows the distribution and can result in more complex decision boundaries, potentially leading to overfitting. Tuning the $\gamma$ parameter is important to achieve the right balance and avoid overfitting or underfitting.
- Sigmoid Kernel: The sigmoid kernel maps the data into a higher-dimensional space using a sigmoid function. It is often used in binary classification problems.
- Polynomial Kernel: The polynomial kernel maps the data into a higher-dimensional space using polynomial functions.

## 2.6 Evaluation Metrics

Evaluation metrics in machine learning are metrics used to evaluate and measure the effectiveness of a machine learning model. These metrics help quantify the model's performance on a given task, such as classification. Using evaluation metrics, researchers and practitioners can compare different models, fine-tune their algorithms, and decide which models to deploy for their specific objectives. The following section will investigate multiple evaluation metrics and their calculation methods.

Before exploring different scoring metrics in detail, it is helpful to analyze a confusion matrix, which illustrates the performance of a machine learning classification model. Regarding Figure 2.7, in binary classification, the confusion matrix is a square matrix that displays the number of True positive (TP), True negative (TN), False positive (FP), and False negative (FN) predictions generated by a classifier. [62].

To clarify, TP denotes instances that the model correctly predicts as positive, while FP refers to instances that are inaccurately predicted as positive when they are actually negative. Similarly, TN represents instances that the model correctly predicts as negative, while FN indicates

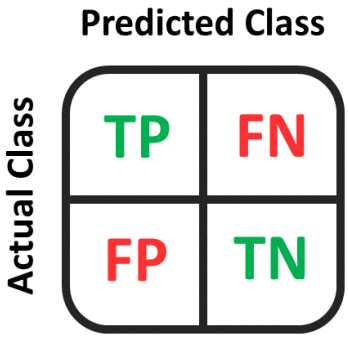



Figure 2.7: Confusion Matrix in machine learning - True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are four parts of the matrix enabling the evaluation of a model's classification performance.

instances that are inaccurately predicted as negative when they are actually positive. The confusion matrix can also be utilized for multi-class classification problems. In this scenario, the rows of the confusion matrix correspond to the true classes, while the columns correspond to the predicted classes, similar to the binary case. It should also be noted that the labels assigned to the classes as "positive" or "negative" can vary depending on the application, and the interpretation of the labels is determined by the user or the specific context of the classification problem. By providing the predicted classes against the actual ones, the confusion matrix is a foundation for calculating other evaluation metrics.

### 2.6.1 Accuracy

Accuracy (ACC) is a commonly used evaluation metric in machine learning that measures the overall correctness of a model's predictions. It quantifies the proportion of correctly classified instances (positive and negative) out of the total instances [62].

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} \tag{2.15}$$

This metric typically ranges from 0 to 1, where 1 represents perfect accuracy, indicating that all predictions the model makes match the actual labels. A value of 0 indicates no accuracy, meaning that none of the predictions align with the actual labels. It is important to note that accuracy alone might not be sufficient to assess model performance, especially when the dataset is imbalanced, or the costs of false positives and negatives are unequal [63]. Other evaluation metrics, such as Recall or F1-score, consider the specific requirements and costs associated with false positives and negatives.

### 2.6.2 Precision, Recall and F1-score

Precision (PRE), Recall (REC), and F1-Score are evaluation metrics commonly used in machine learning for classification tasks. These metrics provide insights into a model's performance, particularly when identifying positive instances is crucial. In particular, Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. In contrast, Recall measures the proportion of correctly predicted positive instances out of all

actual positive instances. [64].

$$PRE = \frac{TP}{TP + FP} \tag{2.16}$$

$$REC = \frac{TP}{TP + FN} \tag{2.17}$$

Both $PRE$ and $REC$ have a range of 0 to 1, where values closer to 1 indicate better performance, while values closer to 0 indicate poorer performance in their respective aspects of evaluation. The F1-Score is a composite measurement that finds a balance between Precision and Recall by calculating their harmonic mean. It is particularly useful when there is an uneven class distribution or when false positives and negatives need to be minimized [64].

$$\textit{F1-Score} = 2 \times \frac{PRE \times REC}{PRE + REC} \tag{2.18}$$

The F1-Score ranges from 0 to 1, with 1 representing ideal Precision and Recall and 0 indicating the worst performance. Considering limitations, Precision disregards false negatives and may yield misleading outcomes if false negatives carry significant implications. Recall overlooks false positives, potentially providing incomplete insights into the overall model performance. Similarly, the F1-Score fails to consider the costs related to false positives and negatives, making it less appropriate in certain situations.

### 2.6.3 Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) is another evaluation metric used to assess the performance of classification models. It takes into account all four components of the confusion matrix to provide a balanced measure of a model's performance, especially in scenarios with imbalanced datasets [64].

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{2.19}$$

The MCC ranges between -1 and +1, where +1 indicates a perfect prediction, 0 represents a random prediction, and -1 indicates a completely opposite prediction. The MCC is particularly beneficial when dealing with imbalanced datasets where one class is dominant, as it considers the performance across all classes and balances the impact of false positives and false negatives [64].

# Chapter 3

# Materials and Methods

## 3.1 Data

This study utilized a multi-source dataset compiled from various research studies conducted by different researchers. The dataset's composition and origins are briefly outlined below.

Anderson et al. [26] conducted a study to assess the predictive value of pharmacokinetic parameters derived from DCE-MRI before chemoradiotherapy in cervical cancer patients. They performed MRI scans on the patient cohort using a 1.5 T Signa Horizon LX tomograph. The resulting images were analyzed to calculate the RSI for each tumor voxel and time point. Two pharmacokinetic models, Brix and Tofts, were then applied to the RSI data in each voxel. The obtained pharmacokinetic parameters ($A_{Brix}, K_{ep}, K_{el}, K^{trans}, V_e$) were later used to create histograms representing the distribution of values for each parameter and tumor. The histogram values were eventually analyzed using a percentile screening method to identify the most clinically relevant portion of the intratumoral parameter distribution in predicting progression-free survival and locoregional control. To evaluate the prognostic significance of pharmacokinetic parameters, researchers also conducted univariate and multivariate Cox regression analyses incorporating representative pharmacokinetic parameters, tumor volume, FIGO stage, and lymph node status.

In a study by Hompland et al. [65], prostate cancer patients were examined using diffusion-weighted MR images to derive the apparent diffusion coefficient and fractional blood volume. They introduced the CSH (consumption and supply-based hypoxia) tool, which integrates oxygen consumption and supply data into a single image to visualize hypoxia in tumors. Afterward, they employed the images generated by CSH to predict the hypoxia condition of each pixel within a tumor and picture the predicted value within an image. In a separate study, Hillestad et al. [66] utilized the CSH tool using the $V_e$ and $K^{trans}$ parameters extracted from DCE-MR images of cervical cancer tumors. Since these parameters indicate tumor oxygen consumption and oxygen supply, respectively, the CSH tool was employed to create a pixel-wise plot of $K^{trans}$ versus $V_e$ for each tumor. This plot represented decreasing oxygen consumption along the horizontal $V_e$-axis and increasing oxygen supply along the vertical $K^{trans}$-axis. Using the generated images, the researchers then developed an algorithm to estimate surrogate measures of hypoxia levels in cervical cancer tumors. Furthermore, they also utilized gene set enrichment analysis (GSEA) with a collection of 50 hallmark gene sets from the Molecular Signature Database [67]

to identify the association between MRI-derived hypoxia levels and gene expression profiles. As a result, they determined distinct hypoxia levels linked to each hallmark.

In a separate study, Fjeldbo et al. [44] developed a classifier using specific signature genes and a predetermined threshold to categorize cervical cancer patients into two groups: a more hypoxic group and a less hypoxic group. These groups exhibited distinct responses to chemoradiotherapy, and the classifier could provide an early indication of the risk of hypoxia-related failure in treatment. Briefly, the classifier utilizes gene expression signatures of tumors as an indirect measure of hypoxia. It consists of three parameters for each gene, determined using a pre-established algorithm and recorded for future tumor classification. When classifying a new tumor, the expression of each classifier gene in that tumor is compared to the recorded parameters to calculate two expression distances, Dmore and Dless. By considering the difference between $D_{less}$ and $D_{more}$ ($D_{less}$ - $D_{more}$), tumors with a negative difference are categorized as less hypoxic. In contrast, tumors with a positive difference are categorized as more hypoxic. Researchers then examined the $A_{Brix}$ pharmacokinetic parameter obtained from DCE-MRI images of tumors as a hypoxia indicator and discovered a strong correlation between the $D_{less}$ - $D_{more}$ values and $A_{Brix}$.

Yoshihara et al. [68] introduced ESTIMATE (Estimation of STromal and Immune cells in MAlignant Tumour tissues using Expression data) in their study. This new algorithm utilizes the unique properties of cancer sample transcriptional profiles. The algorithm focuses on stromal and immune cells, the primary non-tumor constituents of tumor samples. By identifying specific signatures associated with the infiltration of stromal and immune cells in tumor tissues, the researchers performed single-sample gene set-enrichment analysis (ssGSEA) to calculate stromal and immune scores. These scores served as the foundation for the ESTIMATE score, enabling the estimation of tumor purity by inferring tumor cellularity and the presence of different infiltrating normal cells in tumor tissue.

### 3.1.1 Description of the Data Blocks

There were a total of six different data blocks available for analysis, each corresponding to one data source.

- Clinical Data: comprised clinical data from 291 patients with six specific features.
- Gene Scores: contained information on 54 features, including 50 Hallmark gene scores, 3 ESTIMATE scores, and the $D_{less}$ - $D_{more}$ values of the same 291 patients.
- The remaining four data blocks provide information about pharmacokinetic parameters obtained from DCE-MR images:
  - ABrix Data: included information on 118 patients and nine features extracted using the Brix pharmacokinetic model.
  - Ve Data: encompassed data from 67 patients with nine features derived using the Tofts pharmacokinetic model.
  - Ktrans Data: contained information on 67 patients with nine features derived using the Tofts pharmacokinetic model.

– CSH Data: combining the Ve and Ktrans parameters encompassed data from 67 patients with nine features.

A summary explanation of each available pharmacokinetic parameter can be found in Table 3.1, while Table 3.2 provides explanations for each feature in the six data blocks.

Table 3.1: Summary of Pharmacokinetic Parameters Assessed in the Study [4]

| Parameter | Description |
|---|---|
| ABrix | Is the amplitude of contrast uptake and in this data has been associated with hypoxia. Low ABrix values are more hypoxic than high ABrix values. |
| Ve | Is a measure of the extracellular extravascular volume fraction (Space between cells) and can be negatively correlated with tumor cell density. |
| Ktrans | Denotes the transfer rate of the contrast agent from blood to the extravascular extracellular space. |
| CSH | A combination of Ve and Ktrans (proxies of oxygen consumption and supply respectively) images to new images of hypoxia. High values of CSH are more hypoxic than lower values. |

Table 3.2: Overview of Dataset Features [4]

| Dataset | Variable | Description |
|---|---|---|
| All datasets | PasientID | Patient number |
| Clinical data | FIGO_stage | Staging of the tumor. FIGO, Federation International de Gynecologie et d'Obstetrique |
| | LN_status | Lymph node invovlement. 0 = no, 1 = yes |
| | n.voxels | Tumor volume estimated by the number of voxels in images of tumor |
| | Tumor_volum_mm3 | Tumor volume in cubic millimeters |
| | FIGO_stage_2groups | Dividing Figo stage into two groups. 0 = 2B and below, 1 = 3A and above |
| Gene scores | Dless_MINUS_Dmore | 6-gene hypoxia classifier from Fjeldbo et al, Clin Cancer Res 2016. Negative values indicate less hypoxic tumors while positive values indicate more hypoxic tumors. The classification threshold is zero. |
| | ESTIMATEScore | ESTIMATE: Estimate of Stromal and Immune Cells in Malignant Tumor Tissues from Expression Data. Yoshihara et al 2013, Nature Comm. Predicts the presence of stromal and immune cells in tumor tissue. The method is based on single sample gene set enrichment analysis (ssGSEA) algorithm. ESTIMATEScorenumeric scalar specifying tumor cellularity |
| | ESTIMATE_ImmuneScore | StromalScorenumeric scalar specifying the presence of stromal cells in tumor tissue |

Continued on next page

**Table 3.2 – continued from previous page**

| Dataset | Variable | Description |
|---|---|---|
| | ESTIMATE_StromalScore | ImmuneScorenumeric scalar specifying the level of infiltrating immune cells in tumor tissue |
| Gene scores | Variables starting with Score_HALLMARK | Hallmark gene sets from the MSigDB. Scores calculated as mean of median-centered log2-transformed gene expression levels |
| ABrix | ABrix interval 1 | The fraction of voxels with ABrix values between (-0.24, 0.56] |
| | ABrix interval 2 | The fraction of voxels with ABrix values between (0.56, 1.06] |
| | ABrix interval 3 | The fraction of voxels with ABrix values between (1.06, 1.56] |
| | ABrix interval 4 | The fraction of voxels with ABrix values between (1.56, 2.06] |
| | ABrix interval 5 | The fraction of voxels with ABrix values between (2.06, 2.56] |
| | ABrix interval 6 | The fraction of voxels with ABrix values between (2.56, 3.06] |
| | ABrix interval 7 | The fraction of voxels with ABrix values between (3.06, 4.06] |
| | ABrix interval 8 | The fraction of voxels with ABrix values between (4.06, 10] |
| | ABrix below1.56 | The fraction of voxels with ABrix values below 1.56. This parameter is used to reflect hypoxia. |
| Ve | Ve interval 1 | The fraction of voxels with Ve values between (0, 0.1] |
| | Ve interval 2 | The fraction of voxels with Ve values between (0.1, 0.2] |
| | Ve interval 3 | The fraction of voxels with Ve values between (0.2, 0.3] |
| | Ve interval 4 | The fraction of voxels with Ve values between (0.3, 0.4] |
| | Ve interval 5 | The fraction of voxels with Ve values between (0.4, 0.5] |
| | Ve interval 6 | The fraction of voxels with Ve values between (0.5, 0.6] |
| | Ve interval 7 | The fraction of voxels with Ve values between (0.6, 0.7] |
| | Ve interval 8 | The fraction of voxels with Ve values between (0.7, 10] |
| Ktrans | Ktrans interval 1 | The fraction of voxels with Ktrans values between (0, 0.05] |
| | Ktrans interval 2 | The fraction of voxels with Ktrans values between (0.05, 0.1] |
| | Ktrans interval 3 | The fraction of voxels with Ktrans values between (0.1, 0.15] |

Continued on next page

**Table 3.2 – continued from previous page**

| Dataset | Variable | Description |
|---|---|---|
| Ktrans | Ktrans interval 4 | The fraction of voxels with Ktrans values between (0.15, 0.2] |
| | Ktrans interval 5 | The fraction of voxels with Ktrans values between (0.2, 0.25] |
| | Ktrans interval 6 | The fraction of voxels with Ktrans values between (0.25, 0.3] |
| | Ktrans interval 7 | The fraction of voxels with Ktrans values between (0.3, 0.4] |
| | Ktrans interval 8 | The fraction of voxels with Ktrans values between (0.4, 1] |
| CSH | CSH interval 1 | The fraction of voxels with CSH values between (0.05, 0.1] |
| | CSH interval 2 | The fraction of voxels with CSH values between (0, 0.05] |
| | CSH interval 3 | The fraction of voxels with CSH values between (-0.05, 0] |
| | CSH interval 4 | The fraction of voxels with CSH values between (-0.1, -0.05] |
| | CSH interval 5 | The fraction of voxels with CSH values between (-0.15, -0.1] |
| | CSH interval 6 | The fraction of voxels with CSH values between (-0.2, -0.15] |
| | CSH interval 7 | The fraction of voxels with CSH values between (-0.3, -0.2] |
| | CSH interval 8 | The fraction of voxels with CSH values between (-∞, -0.3] |

In addition, the Y-Block dataset included information regarding the treatment outcomes of 67 patients. This was represented by a variable that indicated the overall recurrence status of the tumor within 60 months, irrespective of whether it was a local or distant recurrence. A detailed description of this data block can be found in Table 3.3.

Table 3.3: Summary of Features in Y-Block

| Variable | Description |
|---|---|
| PasientID | Patient number |
| Status_res_dodcancer_60mnd | Recurrence status: 0 = no recurrence, 1 = recurrence |

Regarding the description of the numeric properties of the data blocks, Table 3.4 provides details on the discrete and numerical features in each block, along with their dimensions (where the first value denotes the sample count and the second value indicates the number of features.) Additionally, Figure 3.1 depicts the distribution of classes within the binary variable "Status_res_dodcancer_60mnd" in the Y-Block dataset, including 42 patients who have completely recovered after the treatment period (referred to as Cured) and 25 who have encountered

a cervical cancer tumor relapse (referred to as Relapsed). This variable will serve as the target variable in this study.

Table 3.4: Information about the size and data type of features in each data block

| Blok | Dimention | Feature Name | Data Type |
|---|---|---|---|
| Clinical data | (291, 6) | FIGO_stage<br>The rest of the features | Categorical<br>Numerical |
| Gene scores | (291, 55) | All features | Numerical |
| ABrix | (118, 10) | All features | Numerical |
| Ve | (67, 9) | All features | Numerical |
| Ktrans | (67, 9) | All features | Numerical |
| CSH | (67, 9) | All features | Numerical |
| Y-Block | (67, 2) | All features | Numerical |

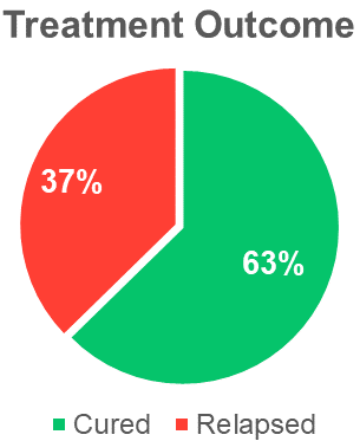



Figure 3.1: The distribution of binary target classes present in the dataset.

## 3.2 Programming and Software

### 3.2.1 Scikit-Learn

The Python-based `scikit-learn` project [69] offers an open-source machine-learning library. It aims to provide efficient and reliable machine-learning tools that are easily accessible to individuals without expertise in machine learning and can be reused in various scientific domains. Rather than being a new domain-specific language, the project functions as a library that offers machine learning idioms in a high-level, general-purpose programming language. `Scikit-learn` is a collection of functions and classes imported into Python programs as a library. Hence, having basic knowledge of Python programming is preferable. The library encompasses traditional learning algorithms, model selection and evaluation tools, and preprocessing procedures. The 1.2.2 version of `Scikit-learn` was employed in this study.

### 3.2.2 Pandas

`Pandas` [70] is a Python library that offers statistical tools and data structures. Statistical data sets frequently come in a tabular format, which comprises a two-dimensional list of observations and field names for each observation. Typically, an observation can be identified uniquely by one or more labels or values. While structured or record arrays may be useful in certain scenarios, they are not as versatile or user-friendly as other statistical environments. Instead, `Pandas` offers the DataFrame class, which presents various beneficial functionalities for data structures. A DataFrame object is adjustable in size, can be transformed into different forms, can store mixed-type data (numerical/categorical), can be appropriately aligned with data sets of differing sizes, and can be utilized to identify, eliminate, or substitute missing data, among other capabilities. `Pandas` version 1.5.2 was utilized in this research.

### 3.2.3 Matplotlib and Seaborn

Data visualization plays an essential role in the scientific process, and creating effective visualizations enables individuals to comprehend their data and effectively communicate their findings to others. `Matplotlib` [71] is a fundamental plotting library for the Python programming language and is an open-source plotting toolkit. It is the most frequently employed among Python visualization packages and can export visualizations to popular image formats such as PDF, SVG, JPG, PNG, BMP, and GIF. It has the ability to create a variety of visualization styles, such as line graphs, scatter plots, histograms, bar charts, error charts, pie charts, box plots, and more.

`Seaborn` [72] is another Python library that produces statistical graphics. It streamlines the process of creating graphics by generating complete graphics with minimal arguments in a single function call, thus enabling quick prototyping and exploratory data analysis. Additionally, `Seaborn` provides numerous options for customization and exposes the underlying `Matplotlib` objects, making it possible to develop polished, publication-quality figures. `Matplotlib` version 3.7.0 and `Seaborn` version 0.12.2 were employed in this study.

### 3.2.4 Imbalanced-Learn

`Imbalanced-learn` is a Python toolbox available as open-source software. Its goal is to offer a diverse collection of methods for managing imbalanced datasets, which are frequently encountered in pattern recognition and machine learning. The toolbox employs the latest methods, such as SMOTE, that can be sorted as under-sampling, over-sampling, a combination of over and under-sampling, and ensemble learning methods. `Imbalanced-learn` is fully compatible with scikit-learn and is a component of the scikit-learn-contrib supported project [73]. This research took advantage of `Imbalanced-learn` version 0.10.1.

In order to replicate and further investigate the experiments conducted in this master's thesis, the code utilized is openly available in a dedicated GitHub repository. Interested readers can access the code repository at the provided link[1].

[1] https://github.com/SinaRokhideh/Cervical_Cancer_Outcome_Prediction

## 3.3 Data Preprocessing

The initial stage in developing a machine learning model is data preprocessing, which involves cleaning and preparing raw data to make it appropriate for machine learning models. Actual data frequently contains noise or missing values and may be in a format that cannot be used directly for machine learning. Therefore, cleaning and formatting the data before performing any operation is necessary. Data preprocessing techniques are used to achieve these objectives [74]. The following tasks are commonly involved in data preprocessing:

- Cleaning data and managing missing values
- Converting qualitative and/or quantitative data
- Normalizing or standardizing data
- Extracting features
- Reducing dimensions or selecting features

This section initially presents an overview of various preprocessing methods and then provides a detailed description of the data preprocessing pipeline employed in this study.

### 3.3.1 Missing Values

Typically, the occurrence of missing values is ascribed to human error during data processing, equipment malfunction resulting in machine error, respondents who decline to answer certain questions, drop-out in studies, and the merging of unrelated data [75]. Missing values issue is widespread across all domains that handle data and can give rise to various problems such as a decline in performance, difficulties in data analysis, and biased outcomes caused by the differences between complete and incomplete data [76]. Furthermore, the severity of missing values is partly determined by the quantity of missing data, the missing data pattern, and the underlying mechanism responsible for the missing values [77].

How missing data is observed and recorded in a dataset is referred to as missing data patterns. Although no universal catalog of missing data patterns is available in the literature, it has identified three primary missing data patterns: univariate, monotone, and non-monotone.

- **Univariate**: A univariate missing data pattern arises when only a single variable has missing data [78], an infrequent occurrence in most fields and usually seen in experimental studies.
- **Monotone**: In this missing data pattern, the data missingness is linked to the values of another variable within the dataset. Specifically, if a particular observation has a missing value for a variable, it indicates that all subsequent variables for that observation are also missing. Since missing value patterns are readily observable, the monotone missing data pattern is easier to manage than the following pattern [79].

- **Non-monotone**: In this pattern, the data missingness is unrelated to any specific pattern or order in the dataset. The missing values are randomly distributed across the variables and observations without any clear relationship to other variables or observations [80].

Various methods exist to manage missing data. One approach to handling incomplete datasets is eliminating rows and/or columns with missing values. However, this strategy results in losing information that may be useful (despite being incomplete), especially in situations where the number of samples available for the study is highly restricted (as is the case in this particular study). A more effective approach is to estimate the missing values using the available data, a technique called imputation [81]. In practice, there are two types of imputation algorithms [82]:

- **Univariate**: Fills in missing values for one feature or variable in the dataset using only the non-missing values within that feature, for instance, when missing values are substituted with the mean or median value of the existing data for the corresponding variable.
- **Multivariate**: Estimates missing values by considering all available feature dimensions. For example, when the missing values are imputed iteratively using conditional models for each variable, taking into account the relationships with other variables in the dataset.

### 3.3.2 Categorical Features

The efficiency of a machine learning model relies not only on the model itself and the hyperparameters used but also on how we handle and input different variable types into the model. As many machine learning models can only handle numerical variables, it is crucial to preprocess categorical variables beforehand. This involves converting categorical variables into numerical equivalents to enable the model to comprehend and extract useful information.

Categorical variables are a form of data that we can measure using nominal or ordinal scales and divide into groups, such as sex, race, educational level, and age group. A nominal variable is a variable with values that cannot be ordered, like gender, where it does not make sense to say that "Male" comes before "Female." On the other hand, Ordinal variables can be ranked but are not necessarily associated with numerical values. For example, dress size is an ordinal variable with "Medium" or "Large" levels. "categorical data" and "qualitative data" are often interchangeable [83].

**One-hot encoding** is a frequently used method to convert categorical attributes into a suitable format for machine learning models. It involves creating a sparse vector where only one element is set to 1, and the rest are 0, which is helpful for representing finite sets of strings. Although high cardinality can result in high dimensional feature vectors, one-hot encoding remains popular due to its simplicity. A one-hot vector is a $1 \times N$ matrix with 0 in all its cells except for one that is set to 1 to uniquely identify one string value [84]. For example, if our dataset contains the Sex attribute of different individuals, it can be transformed into numerical values using the one-hot method, as depicted in Figure 3.2.

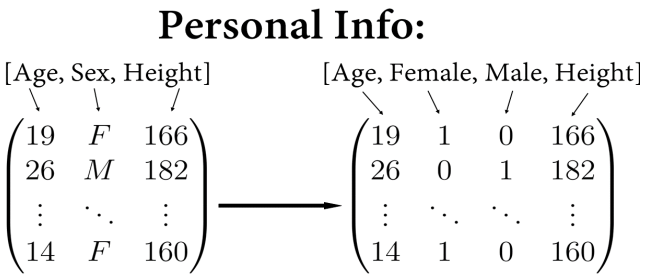



Figure 3.2: Nominal Feature Encoding. In this example, a nominal categorical feature in the dataset called Sex has been transformed into numerical values using the One-Hot encoding method.

### 3.3.3 Data Scaling

Feature scaling is a vital step in data preprocessing that ensures all variables or features in a dataset are in a comparable range. This process is crucial for numerous machine learning algorithms, such as LR or SVM, which rely heavily on the numerical characteristics of features to make accurate predictions [85]. Scaling the features prevent the domination of those with larger magnitudes or ranges, which could otherwise overshadow the influence of other features during the learning process. Additionally, feature scaling aids optimization algorithms like gradient descent in converging faster by preventing oscillation and slow convergence caused by features with disparate scales. Consequently, it also helps reduce the time required to learn the predictive model [86].

Two commonly used methods for making features comparable are normalization and standardization. Normalization typically involves rescaling features to a range between 0 and 1, a specific instance of min-max scaling. To normalize the data, one can easily apply min-max scaling to each feature column. In this process, the new value, $x^{(i)}_{norm}$, for a given sample, $x^{(i)}$, can be calculated using the following formula [86]:

$$x^{(i)}_{norm} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

Here, $x^{(i)}$ represents a specific sample, $x_{min}$ is the smallest value in the feature column, and $x_{max}$ is the largest value. While min-max scaling and normalization are widely employed techniques for bringing values within a specific range, standardization is often more advantageous for machine learning algorithms, particularly optimization algorithms like gradient descent. This preference arises because specific linear models such as LR and SVM initialize weights to 0 or values close to 0. Using standardization, feature columns are centered around a mean of 0 and have a standard deviation of 1, resulting in a normal distribution shape. This facilitates weight learning, making it easier for the algorithm to converge. Furthermore, standardization preserves valuable information about outliers and reduces the algorithm's sensitivity toward them. In contrast, min-max scaling restricts the data to a limited range, potentially discarding useful outlier information [60].

The formula for standardization is given by

$$x^{(i)}_{std} = \frac{x^{(i)} - \mu_x}{\sigma_x} \tag{3.2}$$

In this equation, for each sample $x^{(i)}$, $\mu_x$ represents the sample mean of a specific feature column, while $\sigma_x$ corresponds to the respective standard deviation [86].

### 3.3.4 Imbalanced Data

Machine learning has consistently been affected by the problem of imbalanced datasets. It is challenging to extract knowledge accurately from datasets with a skewed distribution, where one of the target classes (majority class) has substantially more instances than the other class (minority class). In simple terms, the primary issue with predicting imbalanced datasets is the accuracy of predicting both majority and minority classes [87].

For example, in a disease diagnosis scenario, we have a dataset where only 2 out of 100 patients are diagnosed with a disease. This means that the majority class is 98% of patients without the disease, while the minority class is only 2% with the disease. If our model predicts that all 100 patients do not have the disease, it is biased towards the majority class due to the significant difference in the number of records. Confusion matrices are usually used to evaluate how well the model classifies the target classes. In this scenario, the accuracy would be 98/(98+2)=0.98 or 98%. This means that although the model fails to identify the minority class, the accuracy score would still be 98%.

Therefore, it is crucial to identify the minority classes correctly, and our conventional model accuracy calculation methods are ineffective for imbalanced datasets. Essentially, the model must not only focus on detecting the majority class but also assign equal significance or weight to the minority class. While there is no definitive solution to this issue, various techniques are available for solving the class imbalance problem. These may include using appropriate evaluation metrics, random resampling (either through undersampling or oversampling), the synthetic minority oversampling technique, and more [87].

Section 2.6 explained choosing appropriate evaluation metrics for datasets exhibiting imbalanced or balanced distribution. Resampling is another method that can be used to address an imbalanced dataset. In this method, the minority class can be increased by randomly adding records of the same class to the dataset, and this process is known as oversampling. Conversely, the majority class can be reduced by randomly deleting rows to match the size of the minority class, and this method is known as undersampling. By resampling the data in this way, we can achieve a balanced dataset for both classes, enabling the classifier to treat both classes equally [88].

However, merely replicating instances of the minority class may not necessarily provide new information or insights into the model. To address this, an alternative technique called Synthetic Minority Oversampling Technique (SMOTE) can be used for oversampling. SMOTE generates new instances for the minority class by synthesizing data from existing instances. This technique works by examining instances of the minority class and selecting a random nearest neighbor from the $k$ nearest neighbors, and then creating a new synthetic instance at a random location within the feature space [89].

The main steps involved in the SMOTE algorithm are as follows:

1. Identify the minority class, which is the one in the dataset that has fewer instances.

2. Randomly pick an instance from the minority class.
3. Calculate the $k$ nearest neighbors of the chosen instance from the minority class. The user typically specifies $k$.
4. For each selected instance, randomly choose one of its $k$ nearest neighbors. Create a synthetic instance by combining attributes from the selected neighbor and the original instance. The synthetic instance is created along the line connecting the two instances.
5. Repeat steps 2 to 4 until the desired oversampling level is achieved or the minority class is appropriately balanced with the majority class [89].

Given all the explanations provided in this section, the data preprocessing pipeline utilized in this study is illustrated in Figure 3.3.

Due to the unavailability of registered information for specific patients across all datasets, only the data belonging to patients whose PatientID appeared in all datasets (comprising 67 patients) were preserved as the first step. Following that, considering Figure, the missing data within each dataset were initially substituted using the `scikit-learn`'s SimpleImputer class and the most_frequent strategy. This involved replacing the missing values with the most commonly occurring value for the same feature. It is worth noting that during this step, only the Clinical data block contained a single missing value that was addressed through replacement. Categorical features were processed using the OneHotEncoder class, while the numeric features were prepared using the StandardScaler class from `scikit-learn`, ensuring they were centered around a mean of 0 and had a standard deviation of 1. Ultimately, the dataset used for the subsequent stages of the study consisted of data from 67 patients and 96 different features, along with the binary target variable extracted from the Y-Block dataset.

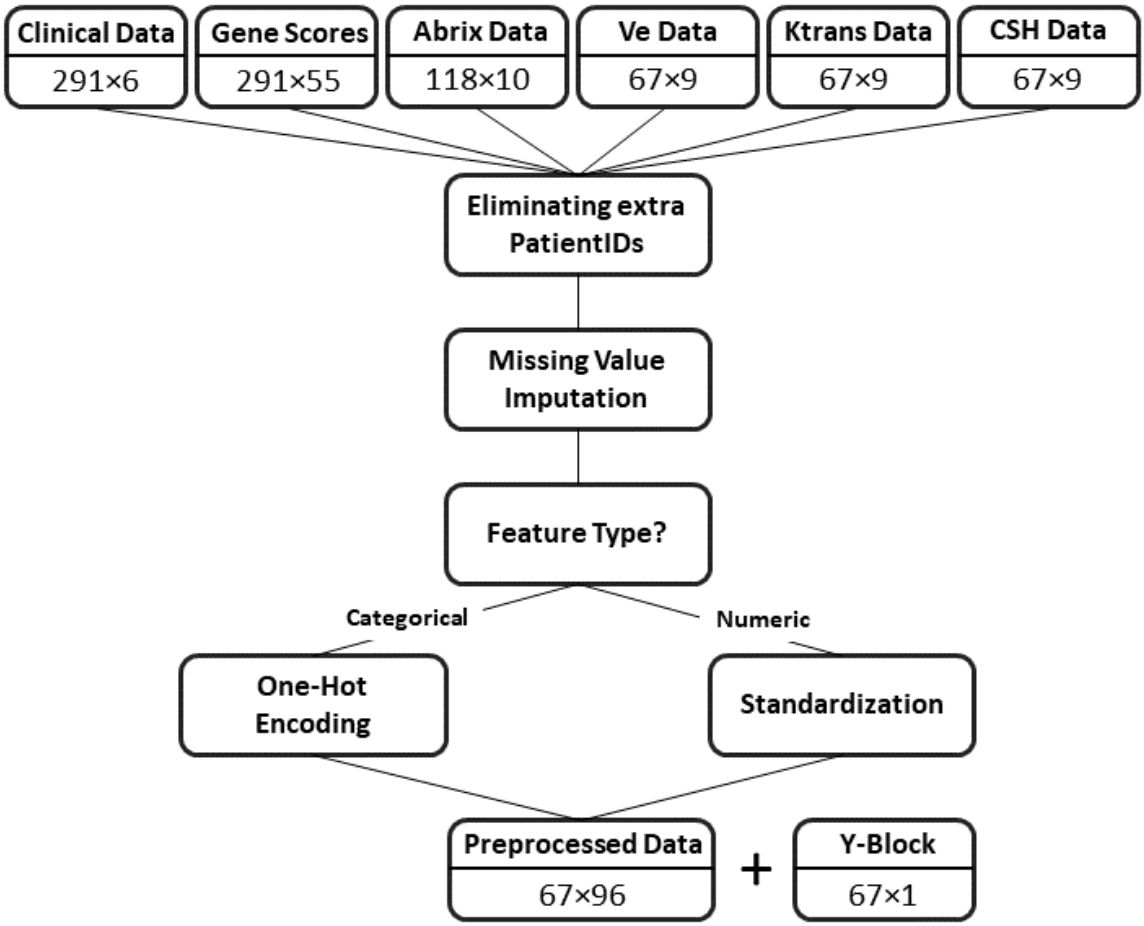



Figure 3.3: Preprocessing pipeline for the data blocks

## 3.4 Feature Selection

The swift advancement of contemporary technologies is causing an exceptional speed of data generation, including images, videos, texts, and voices gathered from social interactions, cloud computing, or medical equipment. Analyzing health and medical data is essential for enhancing the accuracy of diagnoses, treatments, and prevention. However, machine learning and data mining researchers often face challenges in analyzing health data as they are usually high-dimensional, i.e., they have many features and may include duplicated, noisy, or irrelevant information. One way to address this issue is through feature selection techniques [90].

Feature selection involves selecting a subset of features from the original set based on specific criteria that identify the most relevant features within the dataset. This technique aids in compressing the data processing scale by removing redundant or irrelevant features. Feature selection techniques generally pursue two primary objectives. Firstly, they aim to alleviate the curse of dimensionality, i.e., when there is a substantial imbalance between the number of features and the number of samples in a dataset, by reducing complexity and enhancing the mathematical properties of the machine learning model. Secondly, they strive to enhance result interpretability by identifying the most crucial influential features. Alvarez et al. [91] have utilized feature selection techniques in machine learning to enhance the diagnosis of Primary Progressive Aphasia as a group of neurodegenerative disorders using high-dimensional data

derived from FDG-PET images. Their approach obtained similar or even better classification and clustering outcomes with only half the features.

Feature selection methods can be classified into wrapper, filter, and embedded approaches. Wrapper methods choose features based on their predictive performance. They train supervised models on various subsets of the feature set and select the subset that yields the most accurate predictions on a test set. Filter methods rank features using criteria like mutual information or correlation coefficients between features and target variables. On the other hand, embedded feature selection methods integrate the selection process directly into the learning algorithm. One type of embedded method involves regularization in generalized linear models, where regularization terms are incorporated as penalties into the target function during parameter estimation. [3].

### 3.4.1 Repeated Elastic Net Technique (RENT)

The majority of feature selection methods face a common challenge known as instability, where even slight modifications in the random initialization or the division of data into training and testing sets can cause substantial variations in the chosen feature set [92]. Additionally, some feature selection methods encounter another issue: the selected features may have small weights or exhibit alternating signs across different elementary models. This can lead to the selection of unclear or conflicting information, which in turn can adversely affect both the comprehensibility and the ability to make accurate predictions.

The Repeated Elastic Net Technique (RENT) is an ensemble-driven approach for feature selection that falls under the category of embedded feature selection models. It aims to identify robust features for binary classification tasks by employing a logistic regression (LR) model with elastic net regularization, which is trained on multiple subsets of the data [3]. The RENT pipeline is illustrated in Figure 3.4.

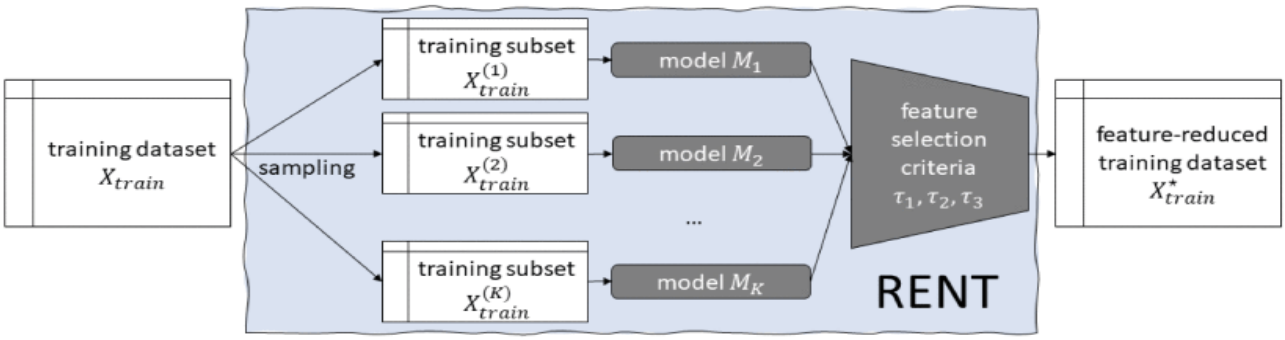



Figure 3.4: The RENT workflow. It involves dividing the input dataset into K submodels for training. Subsequently, it selects features based on three criteria that measure the feature selection percentage, stability, and weight. The outcome is a collection of features that have been chosen [3].

By randomly selecting the primary training data and replacing some samples, unique subsets are created for each model in the ensemble. This process allows for a more accurate assessment of the features' relevance by determining the frequency of feature selection across multiple models. Elastic Net is responsible for determining the inclusion of specific features in each

model. Features that are not selected are assigned a zero weight, while the chosen features have non-zero weights [3].

After training, each model possesses a feature weight vector represented by n, which is subsequently incorporated into a weight matrix denoted as B. In a feature space with N dimensions, the weight matrix B will have dimensions of (K × N), where K represents the number of models. The user can control the frequency of feature selection across all K models using a user-provided threshold ($\tau_1$).

$$\tau_1(\beta_n) = c(\beta_n) = \frac{1}{K}\sum_{K=1}^{K} 1_{[\beta_{K,n}\neq 0]}; \tag{3.3}$$

The significance of a feature, as determined by $c(\beta_n)$, is calculated based on its average occurrence frequency across the $K$ models.

The stability of a feature ($\tau_2$) is determined by the occurrence of only a few weight signals switching between positive and negative values. Ideally, a feature should have weights that are uniformly signed, either all positive or all negative. When all non-zero weights share the same polarity, the significance of $\tau_2$ reaches its maximum potential, matching or surpassing the value of $\tau_1$. The user can specify the desired proportions of feature weights with the same sign [3].

$$\tau_2(\beta_n) = \frac{1}{K}\left|\sum_{K=1}^{K} \text{sign}(\beta_{K,n})\right|; \tag{3.4}$$

In an ideal scenario, a feature consistently demonstrates substantially non-zero weights across all K submodels with minimal variance ($\tau_3$). The $\tau_3$ criterion is defined as

$$\tau_3(\beta_n) = t_{K-1}\left(\frac{|\mu(\beta_n)|}{\sqrt{\frac{\sigma^2(\beta_n)}{K}}}\right); \tag{3.5}$$

The formula involves the feature-specific mean ($\mu$), variance ($\sigma$), and the cumulative density function of the Student's t-distribution with $K-1$ degrees of freedom ($t_{K-1}$). The user is empowered to establish a threshold value ($\tau_3$) between 0 and 1 for the analysis. For instance, a $\tau_3$ value of 0.975 corresponds to a 5% significance level. These selection criteria enable the user to define the level of strictness in the feature selection process. All RENT criteria, including $\tau_1$, $\tau_2$, and $\tau_3$, fall within the range of 0 to 1 ($[\tau_1, \tau_2, \tau_3] \in [0, 1]$) [3].

Several Python functions were developed to implement the RENT feature selection technique in this study. Since the data was initially divided into seven subsets, a function was designed to create a RENT ensemble for each subset. This function utilizes the $RENT\_Classification$ function with various hyperparameters from the author's official Python package [93]. Ultimately, the custom function returns a Python dictionary comprising 7 RENT ensembles, each using training and test data specific to one subset to start the feature selection procedure. Table 3.5 briefly explains some parameters within the $RENT\_Classification$ function and the corresponding values tested for each parameter.

Subsequently, another function was created to train all RENT ensemble models simultaneously. As discussed in section 3.4.1, the third criterion in the RENT method, $\tau_3$, can be associated with the well-known statistical Student's t-test. By setting this criterion to 0.975, a significance level of 5% was established for all models. However, determining the appropriate values for

Table 3.5: The hyperparameters tested in the RENT_Classification function along with the corresponding values that were examined.

| Parameter | Description | Value(s) |
|---|---|---|
| C | Inverse values of $\gamma$ (the elastic net regularization strength) | $10^i \ \forall \ i \in$ [-3,-2,...,2] |
| l1_ratios | The elastic net mixing parameter ($\alpha$), with 0 <= l1_ratios <= 1. l1_ratios=0 corresponds to L2 penalty, l1_ratios=1 to L1 | [0, 0.1, 0.25, 0.5, 0.75, 0.9, 1] |
| classifier | Classification algorithm used for model training (Logistic Regression ('logreg') by default) | 'logreg' |
| K | The number of models within each ensemble | 100 |

the other two criteria, $\tau_1$ and $\tau_2$, required experimentation. Therefore, a new Python function was developed in this study to test various values for these criteria across all seven folds. The function simultaneously considers values ranging from 0.1 to 0.9, with an increment of 0.1, aiming to identify the selected features using each value for both criteria. Specifically, the function returns a Python dictionary containing a list of selected feature names when the pair ($\tau_1$, $\tau_2$) matches any of the abovementioned values across all seven subsets of data (N.B. $\tau_1$ and $\tau_2$ have the same value in all pairs).

With the collection of selected features for each fold and different values of the pair ($\tau_1$, $\tau_2$), it became feasible to construct the ultimate function for training machine learning models solely using the chosen features. It is important to highlight that the hyperparameter values employed for the machine learning models in this phase were set to match the values obtained through GridSearchCV when the models were trained using all available features since the aim was to observe the impact on the models' performances solely by altering the number of introduced features, rather than modifying other parameters.

## 3.5 Baseline Models

A baseline model in machine learning serves as a simple and often naive benchmark against which the performance of more complex models can be compared. The main goal of a baseline model is to establish a minimum level of performance that any model should surpass to be considered useful. Baseline models are typically simple and easy to implement. They may involve basic algorithms or heuristic approaches that rely on basic assumptions about the data. In a classification problem, a typical baseline model could adopt different approaches. For instance, it might randomly assign class labels to samples based on the observed class distribution in the training data. Alternatively, it may assign labels without considering the class distribution at all. Another random classifier might assign all labels to the samples based on the majority class observed in the dataset.

When used as a baseline model, a random classifier does not consider any patterns or features in the data and serves as a baseline for random guessing. That being stated, this kind of baseline typically has an accuracy score corresponding to the baseline accuracy determined by the class distribution in the data. Mathematically, in a binary classification scenario, the probability of

the true classes ($c(x_i)$) of samples $x_1, \ldots, x_n$ belonging to class 0 could be denoted as $p$, while the probability of them belonging to class 1 would be $(1-p)$. In other words, $P(c(x_i) = 0) = p$ and $P(c(x_i) = 1) = 1 - p$. A random classifier that assigns a label ($\hat{c}(x_i)$) according to the observed class distribution would have the same probability of assigning labels to sample $x_i$, i.e., $P(\hat{c}(x_i) = 0)$ is $q$, where $q = p$, and $P(\hat{c}(x_i) = 1)$ is $(1 - q)$. However, since it is random, $\hat{c}(x_i)$ is independent of $c(x_i)$, and thus, the probability that both are 0 is:

$$P(c(x_i) = 0 \text{ and } \hat{c}(x_i) = 0) = P(c(x_i) = 0) \times P(\hat{c}(x_i) = 0) = p \times q \tag{3.6}$$

Similar calculations can be performed to determine the probabilities of three other combinations:

$$P(c(x_i) = 0 \text{ and } \hat{c}(x_i) = 1) = P(c(x_i) = 0) \times P(\hat{c}(x_i) = 1) = p \times (1 - q) \tag{3.7}$$

$$P(c(x_i) = 1 \text{ and } \hat{c}(x_i) = 0) = P(c(x_i) = 1) \times P(\hat{c}(x_i) = 0) = (1 - p) \times q \tag{3.8}$$

$$P(c(x_i) = 1 \text{ and } \hat{c}(x_i) = 1) = P(c(x_i) = 1) \times P(\hat{c}(x_i) = 1) = (1 - p) \times (1 - q) \tag{3.9}$$

These equations correspond to the four components of the confusion matrix, namely true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP), respectively. Since all samples $x_1, \ldots, x_n$ are independent of each other and have an equal probability of being classified as TP, TN, FP, or FN, the expected value for each component is multiplied by $n$. Once all the components are available, various performance metrics, as discussed in Section 2.6, can be directly calculated.

It is worth noting that If the random classifier disregards the class distribution, regardless of the $p$ values, both $q$ and $(1 - q)$ will equal 0.5 or 50%. Similarly, if the classifier assigns all samples to the majority class, the value of either $q$ or $(1 - q)$ will be 1, while the other will be 0, depending on the label of the majority class. In addition, when a random classifier assigns all samples to the majority class, its MCC score will be undefined. This occurs because one column of the confusion matrix (either TP and FP or both TN and FN) will have a zero value, resulting in a division by 0 in the MCC equation. However, research conducted by Chicco et al. [64] demonstrates that the MCC score tends to approach zero even in this scenario.

This study employs all three abovementioned random classifiers as baseline models. On the other hand, it also examines the impact of employing RENT feature selection and SMOTE balancing methods on the performance of machine learning models. This allows for comparing the scores obtained by these models before and after implementing the mentioned methods. In other words, the scores obtained from models trained using all the features in the dataset without sample balancing can serve as a second benchmark for evaluating the performance of models trained exclusively with features selected by the RENT technique or those utilizing balanced datasets through the SMOTE method.

## 3.6 Workflow

Before delving into the workflow, an explanation will be provided regarding data partitioning for training machine learning models. This step is crucial and serves as this study's initial stage following data preprocessing.

### 3.6.1 Data Splitting

Using the complete dataset to fit the model would result in overfitting and may cause inaccurate forecasts in upcoming situations. Thus, reserving a fraction of the dataset for testing and verifying the model's performance before deployment can help prevent unanticipated problems resulting from overfitting. When building statistical and machine learning models, it is typical to divide the dataset into two subsets: training and testing. The training subset is utilized to fit the model and estimate the unknown parameters. Subsequently, the accuracy of the model is assessed using the testing subset [94].

The most straightforward approach to divide the dataset is randomly splitting it into two sections for training and testing, such as using the initial 70% of the data for training the model and the remaining 30% for evaluation (as shown in Figure 3.5.) Despite its simplicity, this method has some limitations. On the one hand, when the dataset is small, this method can lead to high variance because different test sets can produce vastly different results due to random partitioning. Some partitions may contain easy-to-classify samples, while others may contain difficult ones. On the other hand, in typical machine learning scenarios, we also want to tune and compare different parameter settings to enhance the model's prediction performance on unseen data. However, reusing the same test dataset for this purpose can result in overfitting as it becomes part of the training data [60].

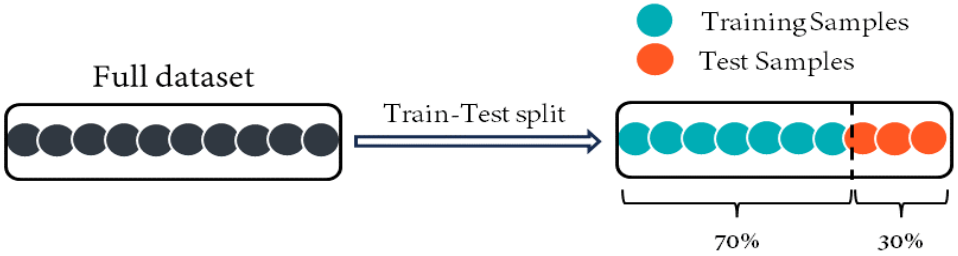



Figure 3.5: The Train-Test split method. In this example, the dataset has been partitioned into training and test sets using a 70/30 ratio.

Cross-validation is a method to address the abovementioned challenges. In cross-validation, the dataset is divided into smaller groups multiple times, and the model's performance is assessed and averaged for each group. This helps minimize the effect of partition randomness on the results. Cross-validation is a popular method to strike a balance between low Bias and low Variance in a model. It can also be employed when comparing multiple models or searching for the best model hyperparameters during evaluation. [95]. Numerous cross-validation methods exist that specify various approaches to splitting a dataset. This research utilizes the two most commonly employed methods: k-fold and leave-one-out.

As shown in Figure 3.6, the K-fold cross-validation method involves dividing the dataset into $k$ equally-sized subsets without replacement. The train-test procedure is then repeated $k$ times, where each time, one of the $k$ subsets is used as the test set, and the remaining $k-1$ subsets are used for training. The model's performance estimate is obtained by averaging the scores over the $k$ trials. This technique is advantageous because it is a resampling method without replacement, ensuring that each sample point is used once for validation and $k$-1 times for training, thereby resulting in a lower variance estimate of model performance than the simplest train-test split method [60].

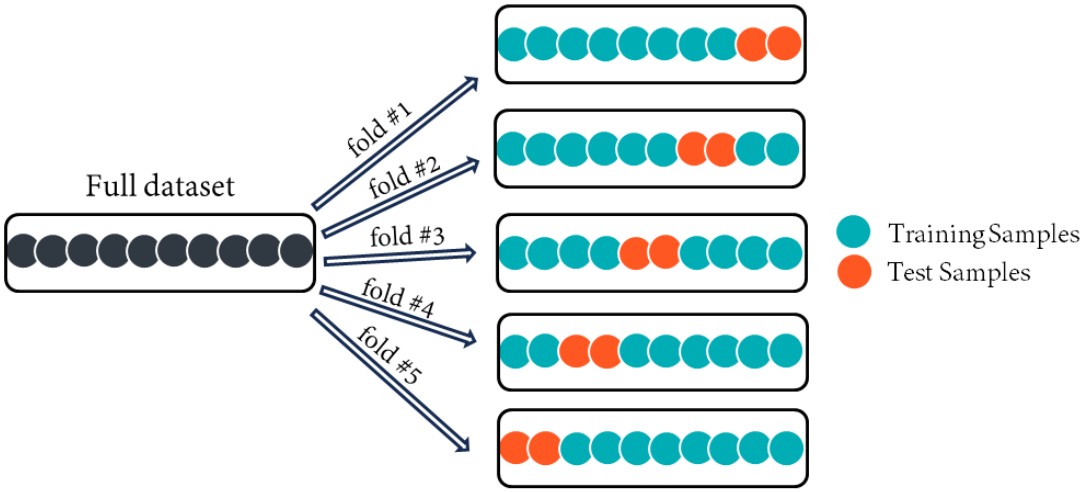



Figure 3.6: The K-fold cross-validation technique. In this example, the dataset has been partitioned into five training and test sets using a 5-fold cross-validation.

However, when dealing with limited training data, increasing the number of folds can be helpful. Raising the value of $k$ allows more training data to be used in each iteration, resulting in a lower bias when estimating generalization performance by averaging individual model estimates. Referring to figure 3.7, Leave-One-Out (LOO) cross-validation involves training a machine-learning model $n$ times, where $n$ is the size of the dataset. In each iteration, only one sample is utilized as the test set, while the rest are used for training the model. In essence, LOO is an extreme form of k-fold cross-validation. Nevertheless, higher $k$ values will lead to longer runtime for cross-validation algorithms and estimates with greater variance due to the more similar training folds. [96].

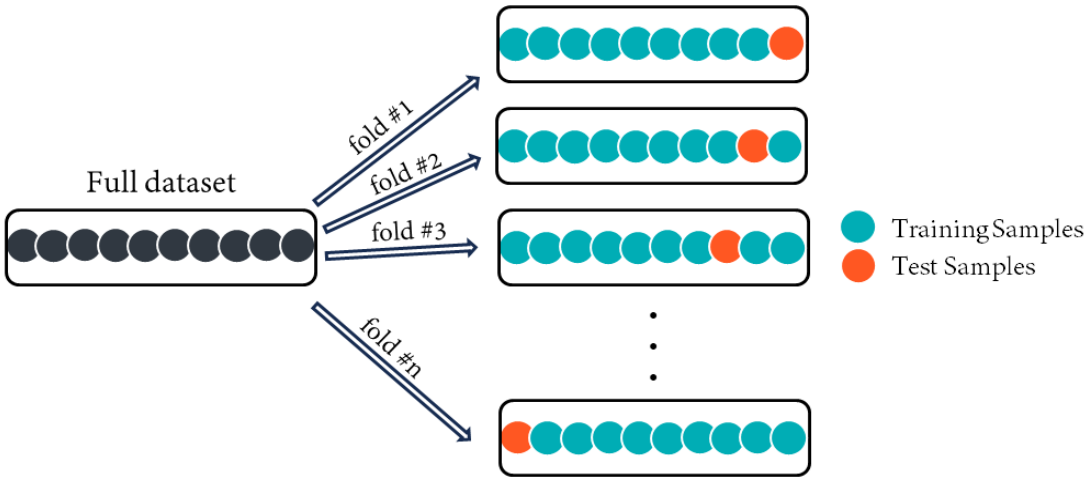



Figure 3.7: This example employs the Leave-One-Out cross-validation approach and divides the dataset into $n$ training and test sets, where $n$ represents the total number of samples in the dataset.

As previously mentioned, cross-validation methods are also useful when looking for the best model hyperparameters during evaluation. One approach is to divide the dataset into three parts: a training set, a validation set, and a test set. The training set is used to fit different models, while selecting optimal values of tuning parameters relies on the performance of the validation set. After achieving satisfactory hyperparameter tuning, we estimate the models'

generalization performance on the test dataset. This concept is illustrated in Figure 3.8, where a validation set is used to repeatedly assess the model's performance after training using different parameter values [60].

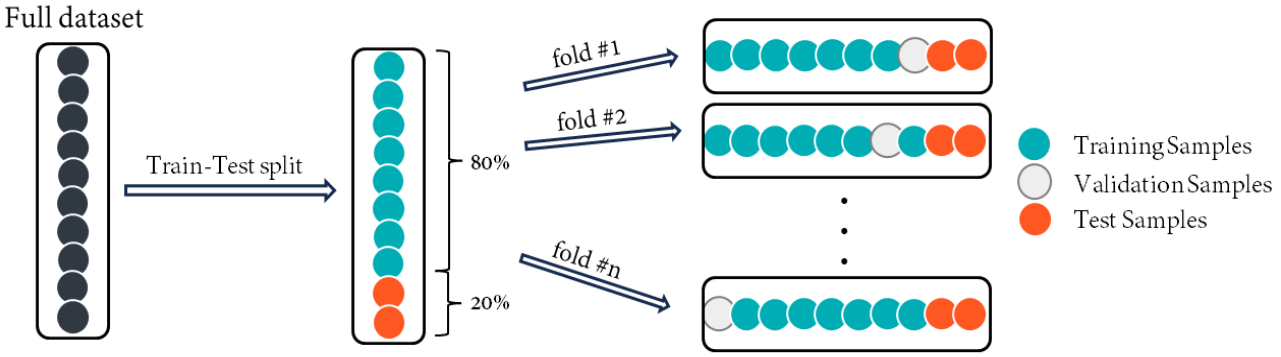



Figure 3.8: In this example, the train-test split method was applied to partition the dataset into training and testing subsets. In order to adjust the model's hyperparameters, the leave-one-out cross-validation technique was then employed, which divides the training set into n segments, where n corresponds to the total number of data samples in the training subset.

The study follows the workflow depicted in Figure 3.9. In the initial step, the preprocessed data is divided into seven subsets using a personalized Python function. Within this function, the `scikit-learn`'s StratifiedKFold class is used to partition the initial dataset while maintaining the proportion of samples for each class, resulting in a Python dictionary that contains a user-defined number of distinct training and test subsets. By generating multiple folds, each representing a different train-test-split, the estimation of the model's performance becomes more reliable and robust. This approach also facilitates optimal hyperparameter tuning and maximizes the utilization of the available data compared to using a single train-test-split. Additionally, considering the total number of samples available for the study, the dataset was divided into seven distinct subsets, each with nearly equal sizes, to establish an approximate training-to-test ratio of 85:15.

The subsequent procedures were designed based on the overarching objectives of the study. Firstly, to evaluate the effectiveness of the available data in predicting the treatment outcomes of locally advanced cervical cancer using machine learning models. Secondly, to explore the influence of the RENT feature selection technique and the SMOTE balancing method on the classification performance of the models.

For implementing the ML classifiers, the support vector machine algorithm utilized the SVC class, the random forest algorithm employed the RandomForestClassifier class (RFC), and the logistic regression algorithm used the SGDClassifier class (SGC) from `scikit-learn`. It is worth mentioning that SGC, which incorporates stochastic gradient learning (SGD) for regularized linear models, was chosen over the LogisticRegression class due to its greater flexibility in adjusting hyperparameters. SGC functions like a logistic regression classifier by selecting the *log_loss* value as the *loss* hyperparameter. Furthermore, this research took advantage of the `scikit-learn`'s GridSearchCV class to explore and identify the optimal set of hyperparameters for each classifier using the LOO method. Table 3.6 displays the hyperparameters examined in each classifier and the values tested for each parameter.

On top of that, the SMOTE and make_pipeline classes from the `Imbalanced-Learn` library were employed to address data imbalance by applying the SMOTE method with the minority

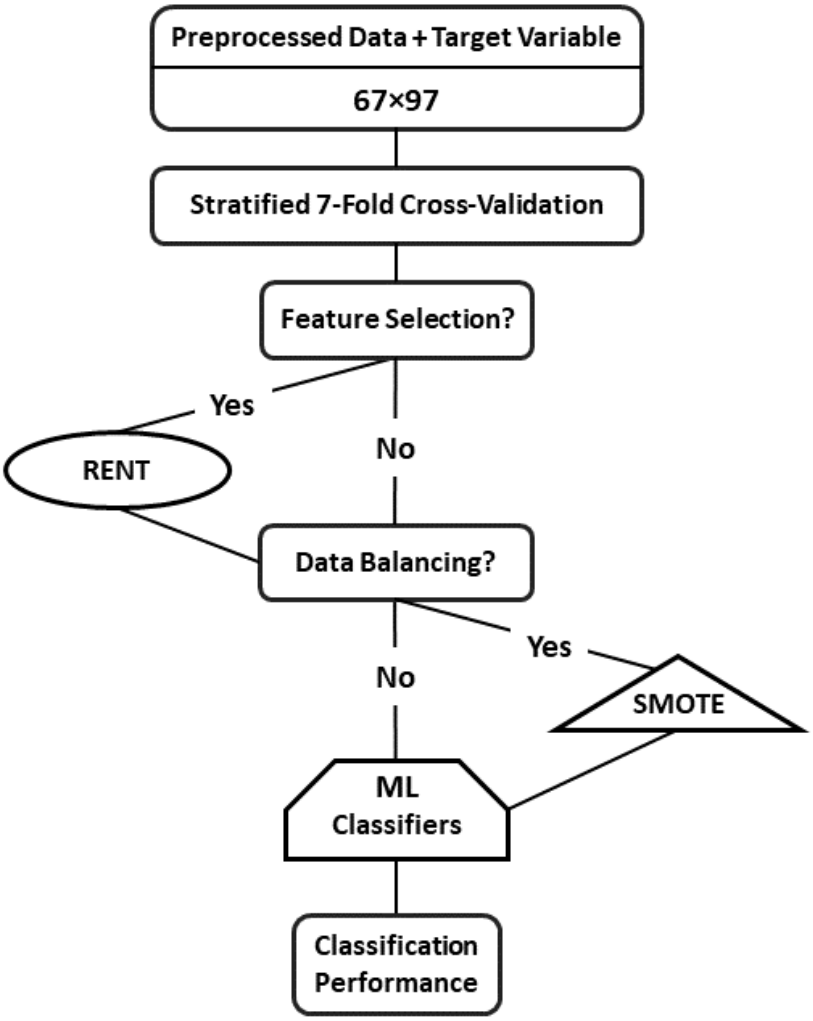



Figure 3.9: The workflow used in this research

sampling strategy and constructing appropriate pipelines for each classifier, respectively. This study also utilized the `scikit-learn`'s classification_report and matthews_corrcoef classes to obtain comprehensive performance reports for the classifiers. These reports include metrics such as Precision, Recall, F1-Score, and Accuracy for the classification_report class and the MCC score for the matthews_corrcoef class.

Table 3.6: The hyperparameters examined in each machine learning classifier used in this study and their corresponding values.

| **Classifier** | **Parameter** | **Description** | **Value(s)** |
|---|---|---|---|
| SGC | loss | The loss function to be used | 'log_loss' |
| | penalty | The regularization term to be used | ['elasticnet', 'None'] |
| | alpha | Constant that multiplies the regularization term. The higher the value, the stronger the regularization | [0.0, 0.1, 1.0, 10.0] |
| | l1_ratio | The elastic net mixing parameter ($\alpha$), with 0 <= l1_ratio <= 1. l1_ratio=0 corresponds to L2 penalty, l1_ratio=1 to L1. | [0.0, 0.5, 1.0] |
| | max_iter | The maximum number of epochs | [100, 1000, 8000] |
| RFC | criterion | The function to measure the quality of a split | ['entropy', 'gini'] |
| | n_estimators | The number of trees in the forest | [500, 1000, 1500] |
| SVC | C | Regularization parameter. The strength of the regularization is inversely proportional to C | $10^i \; \forall \; i \in$ [-4,-2,...,2] |
| | kernel | Specifies the kernel type to be used in the algorithm | ['linear', 'rbf'] |
| | gamma | Kernel coefficient for the $rbf$ kernel | $10^i \; \forall \; i \in$ [-4,-2,...,2] |

# Chapter 4

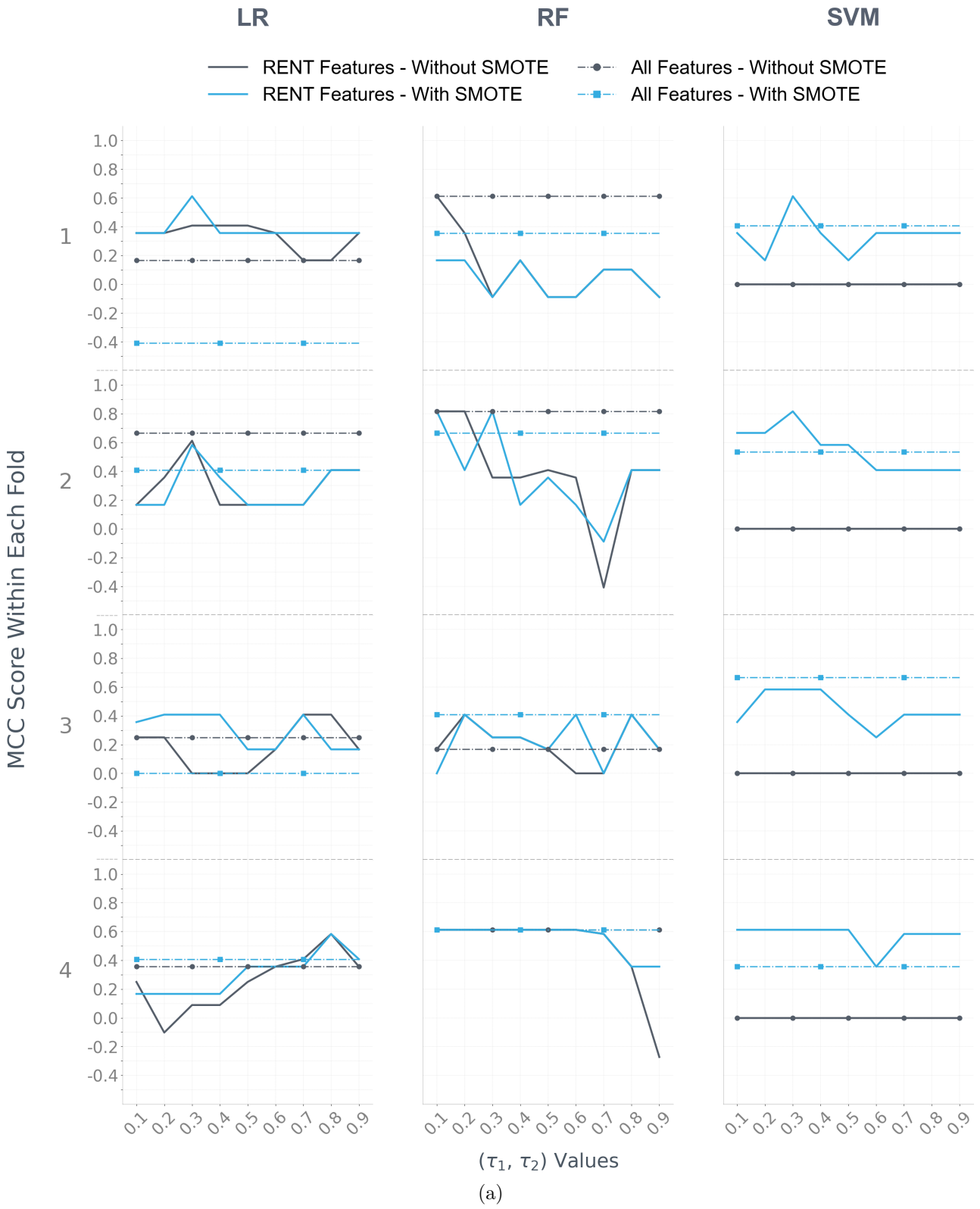
# Experiments and Results

This chapter aims to objectively present the findings, highlighting the key observations and statistical analyses performed. In this chapter, the results are organized in the following manner: Initially, a comprehensive evaluation is conducted on the results derived from machine learning models trained with features selected through the RENT feature selection technique. This evaluation aims to determine the optimal values for the ($\tau_1$, $\tau_2$) pairs associated with each model. In the subsequent section, the study will compare the baseline models and those utilizing the RENT and SMOTE methods. This comparison seeks to evaluate the models' effectiveness in achieving the primary research objective of exploring the potential for early detection of treatment outcomes in locally advanced cervical cancer using DCE-MRI data. Finally, to address the second primary goal, the most informative features within the available dataset in achieving the highest prediction scores will be identified and introduced using the RENT feature selection technique.

## 4.1 RENT Hyperparameter Selection

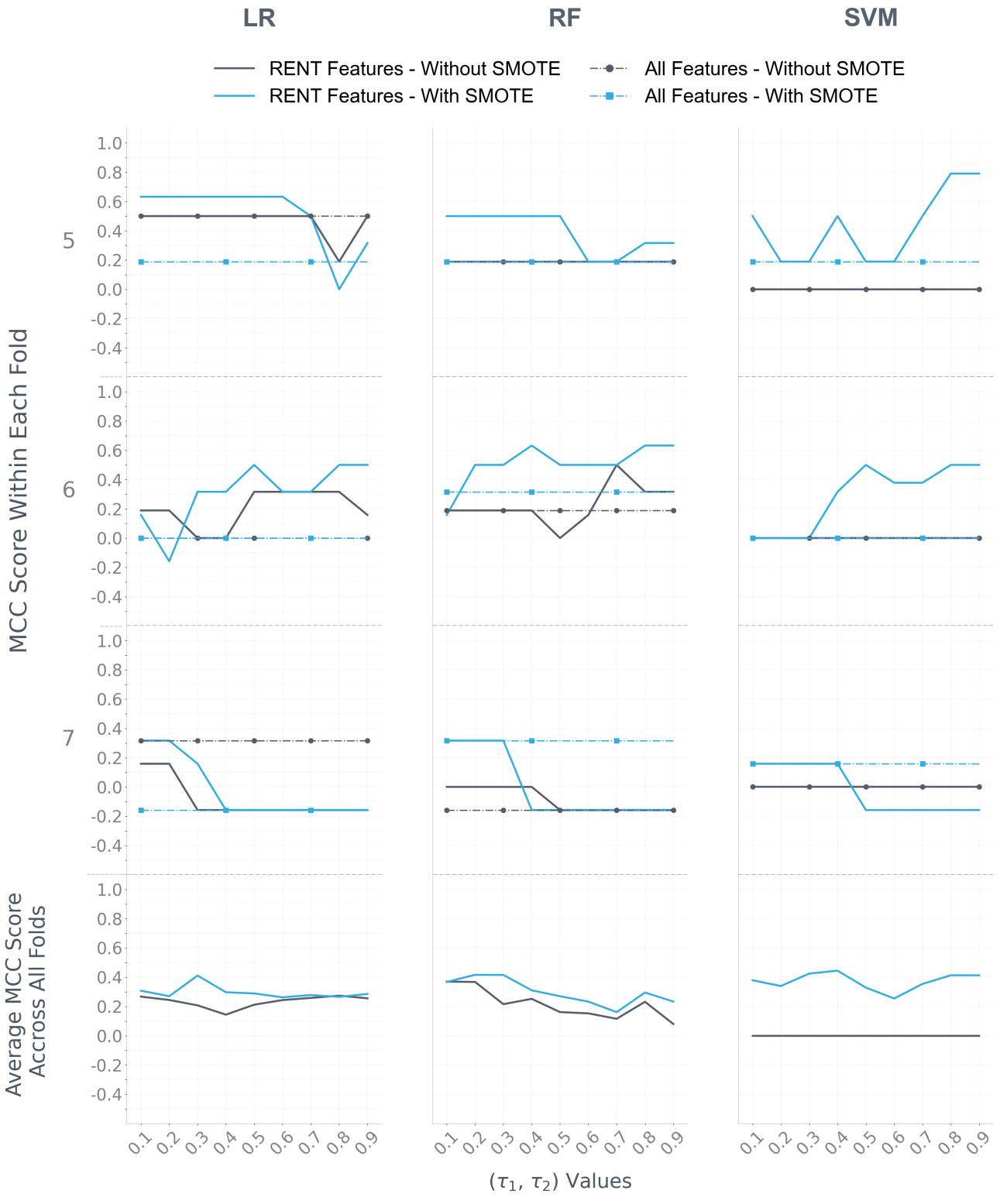
Once the RENT ensemble was created for every seven subsets of data using the values presented in Table 3.5, the optimal combination of C and l1-ratios hyperparameters was determined for each ensemble. The chosen values for all data subsets were 0.1 and 0.25 for the C and l1-ratios hyperparameters, respectively. This means that every ensemble employs both L1 and L2 regularization techniques, with a ratio of 25 to 75 and an overall strength of 0.1, to carry out the feature selection process.

In the next step, machine learning models were trained using the chosen features for all seven folds after collecting the selected features for different values of the ($\tau_1$, $\tau_2$) pairs, where $\tau_1 = \tau_2$ in all pairs. Figures 4.1.a and 4.1.b depict the MCC scores acquired from logistic regression (LR), random forest (RF), and support vector machine (SVM) for every combination of ($\tau_1$, $\tau_2$) values in each fold. The final row of Figure 4.1.b displays the average MCC score across all seven folds.




(a)

Figure 4.1: Classification MCC scores obtained from Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) models with different values for the ($\tau_1$, $\tau_2$) pair in the initial four data folds. Solid lines represent the scores of the models trained with the RENT features, while dashed lines with markers depict the scores of the models trained with all available features. The markers on the dashed lines are purely for visual convenience and do not indicate specific data points.




(b)

Figure 4.1: Classification MCC scores obtained from Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) models with different values for the ($\tau_1$, $\tau_2$) pair in the last three data folds. The average scores across all folds are presented in the final row of the figure. Solid lines represent the scores of the models trained with the RENT features, while dashed lines with markers depict the scores of the models trained with all available features. The markers on the dashed lines are purely for visual convenience and do not indicate specific data points.

Within these figures, the scores obtained from models that utilized the RENT-selected features for training are depicted by solid lines. On the other hand, the scores obtained by models that utilized all available features are represented by dashed lines with markers. Furthermore, the lines are color-coded as blue or gray to indicate whether the models employed the SMOTE balancing method. It is worth noting that the dashed lines with markers exhibit a consistent value within each fold because they are not affected by changes in $\tau$ values, and the markers are purely for visual convenience and do not indicate specific data points. In addition, the average scores of the dashed lines could not be presented in these graphs as they possess distinct values across different folds, regardless of the $\tau$ values. Therefore, those scores will be examined in the subsequent section.

According to Figures 4.1.a and 4.1.b, while the models utilizing the balanced datasets through the SMOTE method generally exhibit higher scores compared to those without SMOTE, the scores achieved by all three models vary for different combinations of ($\tau_1$, $\tau_2$) in each fold, regardless of whether RENT features or all available features were used. One possible explanation could be the disparity between each fold's training and test samples. This variation between the samples can impact the obtained results based on the relative ease or difficulty of classifying the samples. Hence, the average scores across all folds were employed to determine the optimal values for the ($\tau_1$, $\tau_2$) pairs in each model. A noteworthy observation about the gray lines in the SVM model, irrespective of utilizing the RENT features, is that the model consistently obtains an MCC score of zero for all ($\tau_1$, $\tau_2$) combinations. This indicates that the SVM's performance was no better than that of a random classifier when the SMOTE balancing method was not employed.

Table 4.1 provides a more detailed breakdown of the average values obtained by the models trained using the RENT-selected features considering whether the SMOTE balancing method is employed. In the logistic regression model, the best ($\tau_1$, $\tau_2$) pairs are determined as 0.3 for models utilizing the SMOTE method and 0.8 for models not employing it. In the Random Forest model, the optimal ($\tau_1$, $\tau_2$) value remains constant at 0.2, regardless of whether the SMOTE method is used. Lastly, in the support vector machine model, where the average score remains at zero when the SMOTE method is not used, the optimal ($\tau_1$, $\tau_2$) pair is 0.4, exclusively for the models that utilize SMOTE.

Table 4.1: Average MCC scores of logistic regression (LR), random forest (RF), and support vector machine (SVM) models trained with RENT-selected features, grouped by the utilization of SMOTE balancing method, across various ($\tau_1$, $\tau_2$) values.

| | | **MCC scores belonging to each ($\tau_1$, $\tau_2$) value** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **SMOTE** | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** |
| **LR** | Yes | 0.307 | 0.269 | **0.411** | 0.296 | 0.288 | 0.262 | 0.277 | 0.265 | 0.285 |
| | No | 0.267 | 0.243 | 0.207 | 0.143 | 0.211 | 0.243 | 0.258 | **0.273** | 0.255 |
| **RF** | Yes | 0.367 | **0.415** | 0.415 | 0.310 | 0.269 | 0.232 | 0.161 | 0.295 | 0.233 |
| | No | 0.369 | **0.367** | 0.215 | 0.251 | 0.161 | 0.152 | 0.115 | 0.231 | 0.080 |
| **SVM** | Yes | 0.378 | 0.339 | 0.424 | **0.444** | 0.328 | 0.254 | 0.353 | 0.412 | 0.412 |
| | No | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Similar to this section, Appendix A.1 presents the accuracy scores of the models in a comparable manner.

## 4.2 Classification Modelling and Evaluation

To this point, the evaluation has focused on the performance analysis of models trained using features selected by the RENT feature selection technique, both prior to and after undertaking the SMOTE class balancing method. Now that the best scores have been obtained from these models, it becomes feasible to compare them against both the scores of the random classifier as the first baseline model and the scores of the models that employed all available features to predict the treatment outcomes of the patients as the second benchmark.

As mentioned in Section 3.5, this study utilizes three random classifiers as baseline models. Considering the class distribution depicted in Figure 3.1, the value of $p$ is 0.63, while $(1-p)$ is 0.37. Table 4.2 presents the $q$ and $(1-q)$ values, which rely on the random classifier. According to the information provided in this table, random classifier #1 disregards the distribution of classes in the dataset. In contrast, random classifier #2 assigns samples randomly based on the existing class distribution. On the other hand, classifier #3 assigns all samples to the majority class.

Table 4.2: The possibility of predicted sample classes belonging to class 0 (denoted as $q$) and the probability of them belonging to class 1 (denoted as $1-q$).

| **Model** | $q$ | $1-q$ |
|---|---|---|
| Random Classifier #1 | 0.5 | 0.5 |
| Random Classifier #2 | 0.63 | 0.37 |
| Random Classifier #3 | 1 | 0 |

Using these values, the confusion matrix components were determined for each random classifier, and subsequently, the Accuracy, MCC score, and F1-Score were computed. The corresponding results are presented in Table 4.3. Consequently, any classifier that achieves higher scores than those obtained from these baseline models is favored over them.

Table 4.3: Performance metrics (Accuracy, F1-Score, and MCC score) of three random classifiers: Random Classifier #1 randomly assigns labels without considering class distribution, whereas classifier #2 accounts for the dataset's class distribution. Random classifier #3 assigns all samples to the class with the highest count based on the class distribution.

| **Model** | **Accuracy** | **F1-Score** | **MCC** |
|---|---|---|---|
| Random Classifier #1 | 50% | 0.56 | 0 |
| Random Classifier #2 | 53% | 0.63 | 0 |
| Random Classifier #3 | 63% | 0.77 | 0 |

As stated in Section 3.6, the machine learning models were initially trained without using the RENT and SMOTE methods on each fold's subset of data. This was done to establish a second benchmark for performance evaluation. As part of the training process during this stage, the optimal hyperparameters for each model were obtained from the tested range through GridSearchCV, as indicated in Table 4.4. These hyperparameter values were subsequently

utilized in training the models that incorporated the features selected by RENT. It is important to highlight that these values remained consistent across all folds.

Table 4.4: Optimal hyperparameter values chosen for logistic regression (LR), random forest (RF), and support vector machine (SVM) models.

| **LR** | | | | **RF** | | **SVM** | |
|---|---|---|---|---|---|---|---|
| penalty | alpha | l1_ratio | max_iter | criterion | n_estimators | C | kernel |
| elasticnet | 0.1 | 0.0 | 100 | entropy | 500 | 0.0001 | linear |

To assess the performance of the machine learning models, Figure 4.2 compares the MCC scores of the models trained using the complete feature set (referred to as "All") before and after balancing the classes with SMOTE, serving as the second benchmark. Additionally, the scores of the models trained using the RENT-selected features (referred to as "RENT"), corresponding to the optimal ($\tau_1$, $\tau_2$) pairs mentioned in the previous section, are included in the comparison. Notably, In each plot, the final set of columns (marked as Avg) displays the average scores achieved across 7 folds, the standard deviation of each category, and the highest average score is also indicated in the respective column. Notably, the plots in this figure do not include the random classifiers' MCC scores, which remain constant at zero.

According to Figure 4.2, the scores achieved by models in each category varied when using different training and test data in each fold. One noteworthy observation is that the support vector machine model outperformed the random classifiers (used as the baseline) only when the training samples were balanced using the SMOTE method. Otherwise, it always predicts the majority class, resulting in an MCC score of 0. Among the support vector machine models, the group trained with the RENT features and SMOTE method exhibited the best performance, with an average MCC score of 0.444.

On the other hand, within the Random Forest models, the classifiers in all four categories exhibited closer average scores compared to the other two models, namely logistic regression and support vector machine. Furthermore, the average scores of all four groups were significantly higher than zero, surpassing the performance of the initial baseline model. In addition, the performance of the Random Forest models also showed improvement after applying the SMOTE method. The models utilizing the RENT and SMOTE techniques achieved the highest average score of 0.415.

Lastly, in logistic regression, the group that utilized all the features and the SMOTE method displayed more fluctuation between positive to negative values across different folds. As a result, this group's average score was lower than the other categories and only slightly improved over the initial baseline models. In contrast, the remaining categories within this model exhibited significantly better average MCC scores than the random classifiers. Similar to the previous two machine learning models, in logistic regression, the group that employed features obtained from the RENT technique and the SMOTE method achieved the highest average score, with an MCC score of 0.411.

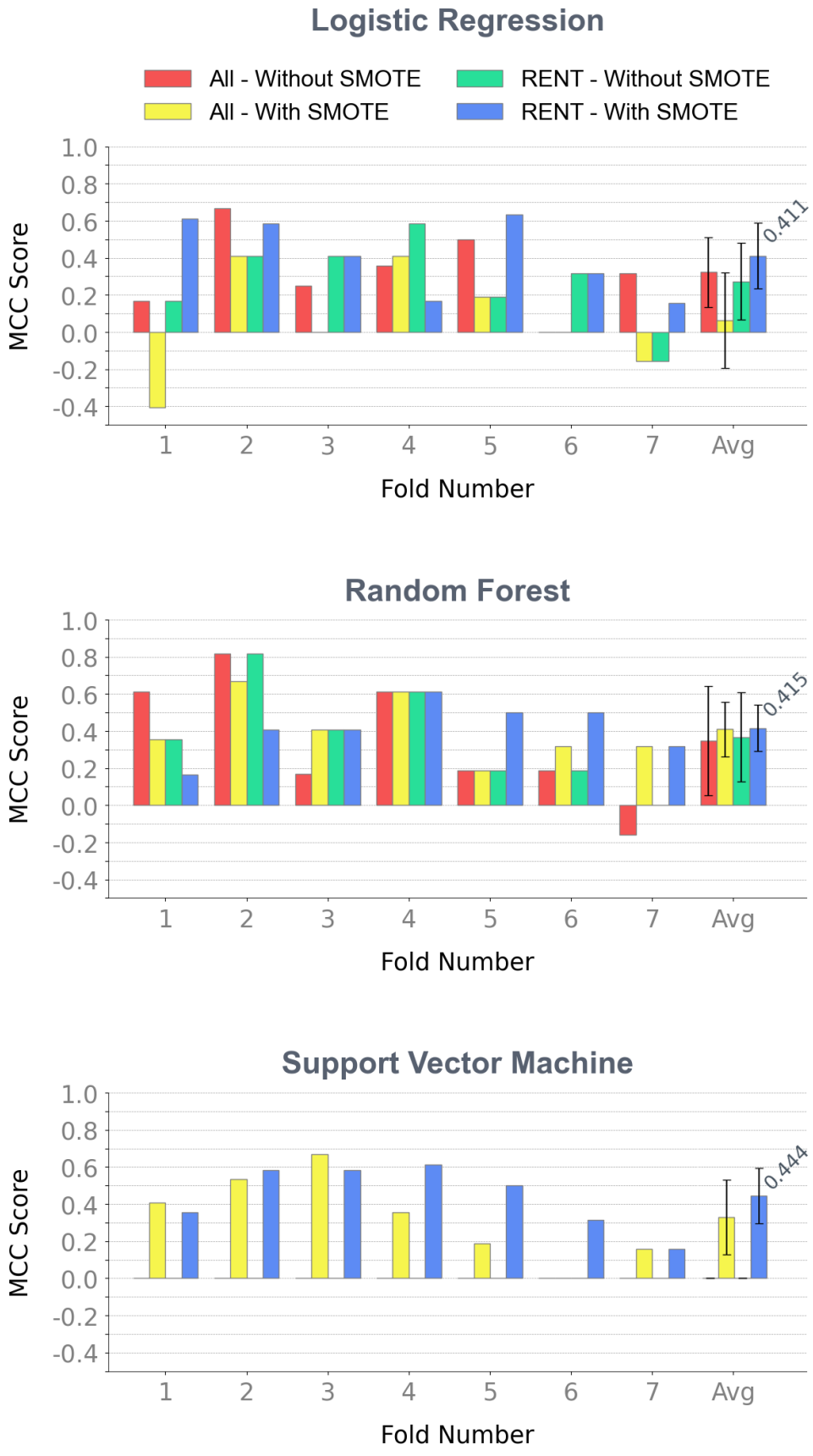



Figure 4.2: The classification MCC scores achieved by machine learning models in seven different folds. The models were color-coded based on whether they were trained using all available features (referred to as "All") or the features selected by the RENT method (referred to as "RENT"), with or without the utilization of the SMOTE balancing technique. The final set of columns in each plot displays the average scores across all folds, along with the standard deviations for each category. The column representing the highest average value is highlighted accordingly.

To facilitate comparison, Table 4.5 summarizes the data from the average score columns in Figure 4.2 and the percentage of changes observed after applying the RENT and SMOTE methods.

Table 4.5: MCC scores averaged across all seven folds for each model trained with all features and RENT-selected features, with and without applying the SMOTE method. The final two columns present the corresponding changes after applying the RENT and SMOTE methods.

| | **All Features** | | **RENT Features** | | **Change (after RENT)** | |
|---|---|---|---|---|---|---|
| **Model** | **No SMOTE** | **SMOTE** | **No SMOTE** | **SMOTE** | **No SMOTE** | **SMOTE** |
| **LR** | 0.322 | 0.062 | 0.273 | 0.411 | -15.2 % | +562.9 % |
| **RF** | 0.346 | 0.409 | 0.367 | 0.415 | +6.06 % | +1.46 % |
| **SVM** | 0 | 0.33 | 0 | 0.444 | 0 % | +34.5 % |

The results from Table 4.5 indicated that except for the logistic regression model with all features, where incorporating SMOTE had a negative impact on the results, the performance of all other models was enhanced using SMOTE. In addition, all three machine learning models outperformed all benchmarks when simultaneously employing RENT and SMOTE techniques. However, as the best scores of the models were relatively similar and the number of selected features varied based on different values of $\tau_1$ and $\tau_2$ in the RENT method, a statistical ANOVA (analysis of variance) test was employed to determine how each factor affects the achieved outcomes.

The test was performed with the null hypothesis that the means of the tested groups were the same. By selecting a significance level of 5%, any group with a p-value below 0.05 would reject this hypothesis. The independent factors examined in this 3-way test were the classifier with three values (LR, RF, SVM), SMOTE usage, and RENT usage, each with two values (0 for not using, 1 for using). In addition, the test incorporated interaction terms between the classifier and the other two factors, and the response variable was the MCC score. The test results are presented in Table 4.6. Also, to ensure that the test results were attributed to statistical significance rather than random variation between groups, both assumptions of the ANOVA test were examined. These assumptions involve verifying that the data adheres to a normal distribution and that the variances among the compared groups are roughly equal.

Table 4.6: The results derived from the ANOVA test

| | **sum_sq** | **df** | **F** | **PR(>F)** |
|---|---|---|---|---|
| C(Classifier) | 0.524243 | 2.0 | 5.606698 | 0.005458 |
| C(SMOTE) | 0.235821 | 1.0 | 5.044129 | 0.027779 |
| C(RENT) | 0.008502 | 1.0 | 0.181851 | 0.671060 |
| C(Classifier):C(RENT) | 0.008931 | 2.0 | 0.095516 | 0.909019 |
| C(Classifier):C(SMOTE) | 0.610876 | 2.0 | 6.533222 | 0.002470 |
| C(Classifier):C(SMOTE):C(RENT) | 0.299108 | 3.0 | 2.132605 | 0.103564 |
| Residual | 3.366109 | 72.0 | NaN | NaN |

Based on the p-values listed in the table's final column, it can be observed that the classifier, SMOTE, and their interaction reject the null hypothesis and have a statistically significant

impact on the MCC score, as their p-values are less than 0.05. On the other hand, the p-values associated with the RENT factor and its interactions were higher than 0.05, indicating no statistically significant interaction can be concluded at the present sample size. Investigating Figure 4.2 reveals that the utilization of RENT did not have a significant negative impact on the results either. However, this method proved highly effective in reducing the models' computational time and enhancing their interpretability by introducing the most significant features.

The ANOVA results indicated that employing different classifiers and the SMOTE method leads to a statistically significant different outcome. However, to identify which specific groups among these factors potentially exhibited significant differences compared to other groups, Tukey's HSD statistical test was employed in this study. This test served as a post hoc analysis following the ANOVA test to determine if there were significant statistical differences between the LR, RF, and SVM models and between these models when interacting with the SMOTE factor. Table 4.7 presents Tukey's HSD outcomes, explicitly assessing the dissimilarity between the classifiers. On the other hand, Table 4.8 displays the results of this test, examining the disparity between groups of the classifier factor when employing SMOTE versus not employing it.

Table 4.7: Results of Tukey's HSD test comparing the statistical differences among classifiers

| group1 | group2 | Diff | Lower | Upper | q-value | p-value |
|---|---|---|---|---|---|---|
| LR | RF | 0.117429 | -0.024353 | 0.259210 | 2.798602 | 0.124384 |
| LR | SVM | 0.073571 | -0.068210 | 0.215353 | 1.753382 | 0.435383 |
| RF | SVM | 0.191000 | 0.049218 | 0.332782 | 4.551984 | 0.005273 |

Table 4.8: Tukey's HSD test results investigating the statistical difference between various combinations of classifiers and the SMOTE utilization. In the first two columns, the first value within each pair represents the classifier name, while the second value indicates the SMOTE method's usage (1) or non-usage (0).

| group1 | group2 | Diff | Lower | Upper | q-value | p-value |
|---|---|---|---|---|---|---|
| (LR, 0) | (LR, 1) | 0.060929 | -0.178340 | 0.300197 | 1.054356 | 0.900000 |
| (LR, 0) | (RF, 0) | 0.059143 | -0.180125 | 0.298411 | 1.023454 | 0.900000 |
| (LR, 0) | (RF, 1) | 0.114786 | -0.124483 | 0.354054 | 1.986342 | 0.698841 |
| (LR, 0) | (SVM, 0) | 0.297643 | 0.058375 | 0.536911 | 5.150646 | 0.006480 |
| (LR, 0) | (SVM, 1) | 0.089571 | -0.149697 | 0.328840 | 1.550014 | 0.873690 |
| (LR, 1) | (RF, 0) | 0.120071 | -0.119197 | 0.359340 | 2.077810 | 0.662189 |
| (LR, 1) | (RF, 1) | 0.175714 | -0.063554 | 0.414983 | 3.040698 | 0.274119 |
| (LR, 1) | (SVM, 0) | 0.236714 | -0.002554 | 0.475983 | 4.096290 | 0.054192 |
| (LR, 1) | (SVM, 1) | 0.150500 | -0.088768 | 0.389768 | 2.604370 | 0.448044 |
| (RF, 0) | (RF, 1) | 0.055643 | -0.183625 | 0.294911 | 0.962888 | 0.900000 |
| (RF, 0) | (SVM, 0) | 0.356786 | 0.117517 | 0.596054 | 6.174100 | 0.001000 |
| (RF, 0) | (SVM, 1) | 0.030429 | -0.208840 | 0.269697 | 0.526560 | 0.900000 |
| (RF, 1) | (SVM, 0) | 0.412429 | 0.173160 | 0.651697 | 7.136988 | 0.001000 |
| (RF, 1) | (SVM, 1) | 0.025214 | -0.214054 | 0.264483 | 0.436328 | 0.900000 |
| (SVM, 0) | (SVM, 1) | 0.387214 | 0.147946 | 0.626483 | 6.700660 | 0.001000 |

With the same null hypothesis in Tukey's HSD test and setting a confidence level of 0.95, any

group with a p-value below 0.05 indicated a significant difference among the tested groups. Table 4.7 demonstrates no statistically significant distinction between LR and RF classifiers and between LR and SVM. However, RF and SVM exhibit a statistically significant distinction in obtaining MCC scores. These findings align with the standard deviations depicted in Figure 4.2.

On the other hand, based on the p-values in Table 4.8, a statistically significant difference exists between SVM when not employing SMOTE and almost all the other combinations. Considering Figure 4.2, this is not unexpected as it demonstrates that SMOTE has the most substantial impact on SVM. Therefore, except for the combinations having (SVM, 0), the remaining interaction terms do not display any statistically significant differences. Thus, an ANOVA model without interactions would suffice for all other combinations. Lastly, it is worth mentioning that the p-value of 0.054 between the LR model with SMOTE and the SVM model without SMOTE is marginally higher than 0.05. While it does not reject the null hypothesis, it does indicate that significant differences might be detected in a larger sample size.

Appendix A.2 provides supplementary details for this section, including accuracy, precision, recall, and F1 scores for models before and after applying RENT and SMOTE techniques.

## 4.3 The Most Informative Features

A secondary objective of this study was to identify the most informative features from the available data for the early detection of treatment outcomes in patients with locally advanced cervical cancer using the RENT feature selection method. As explained in Section 3.4, various features were collected with different combinations of $(\tau_1, \tau_2)$ values to achieve this goal. Figures 4.3.a, 4.3.b, 4.3.c, and 4.3.d display the proportion and quantity of selected features from each available data block, along with the total number of features chosen in each subset across all seven folds, corresponding to the values 0.2, 0.3, 0.4, and 0.8 for the $(\tau_1, \tau_2)$ pair, respectively.

While the selected features are accessible for all tested values for the $(\tau_1, \tau_2)$ pair (as indicated in Section 3.4), only the features corresponding to the abovementioned values are presented in this section because the machine learning models achieved the highest average MCC score when utilizing the selected features associated with these specific values. Additional figures associated with the combinations of $(\tau_1, \tau_2)$ not presented in this section can be found in Appendix A.3.

Regarding Figures 4.3.a–d, it is essential to note that the number of selected features from the Clinical dataset considers preprocessing and applying the One-Hot Encoding technique. Therefore, the presented share may exceed this dataset's original number of features. Based on these figures, the total number of selected features is decreased by over 42%, even when opting for a value of 0.2 for the $(\tau_1, \tau_2)$ pair, which implies a relatively lenient restriction on the feature selection process. In order to evaluate the impact of this reduction in the number of features used in training each model, the training time for the models utilizing the best hyperparameter values was measured before and after applying the RENT and SMOTE methods[1]. The average training time, measured in seconds, for each model across seven folds is presented in Table 4.9.

[1]The outcomes were obtained through the execution of scripts on a Microsoft Windows 10 Pro device equipped with an Intel i7 CPU with two cores running at 1.8 GHz and 8 GB of RAM.

Table 4.9: Average training duration, in seconds, for logistic regression (LR), random forest (RF), and support vector machine (SVM) models across all seven folds, before and after employing the RENT method (referred to as All features or RENT features, respectively) and SMOTE technique.

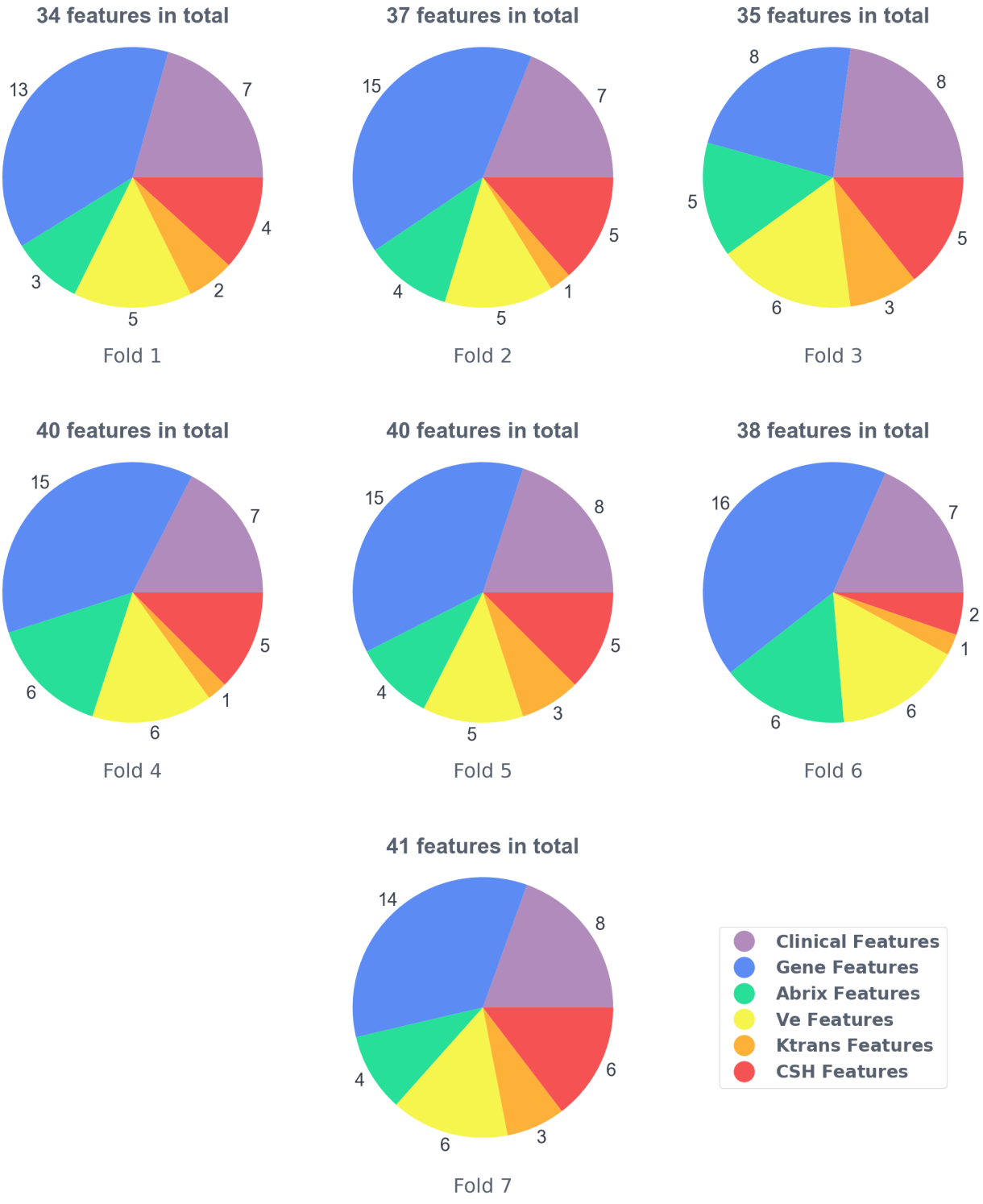
| | **All Features** | | **RENT Features** | |
|---|---|---|---|---|
| **Model** | **Without SMOTE** | **With SMOTE** | **Without SMOTE** | **With SMOTE** |
| LR | 1.49 | 8.29 | 0.01 | 0.03 |
| RF | 39.778 | 94.4 | 1.3 | 10.49 |
| SVM | 0.5 | 5.9 | 0.02 | 0.04 |

Based on the findings displayed in this table, reducing the number of features after employing the RENT method leads to a significant decrease in the average training time for all models, regardless of whether the SMOTE technique is utilized. However, it is worth noting that the training process for all models is considerably more time-consuming when the SMOTE technique is applied.

Upon further comparing Figures 4.3.a–d, it becomes evident that as the value of $\tau_1$ and $\tau_2$ increases, resulting in stricter constraints on the feature selection process features from the Clinical data block are prioritized over other data blocks. This is clear by the greater inclusion of features from the Clinical dataset among the total selected features. Simultaneously, the features from other data blocks are either partially or entirely eliminated in different folds. In addition to these figures, Tables 4.10 to 4.13 were presented to facilitate a comprehensive analysis of the chosen features categorized by each data block. These tables include the names of the selected features in the second column, the frequency with which each feature was selected for the given ($\tau_1$, $\tau_2$) value across all folds in the third column, and the specific folds in which each feature was chosen, outlined in the last column. Notably, if a feature was selected in all seven folds, the entry "All Folds" is indicated in the last column for that particular feature. Otherwise, the number of folds in which the feature was selected is stated.

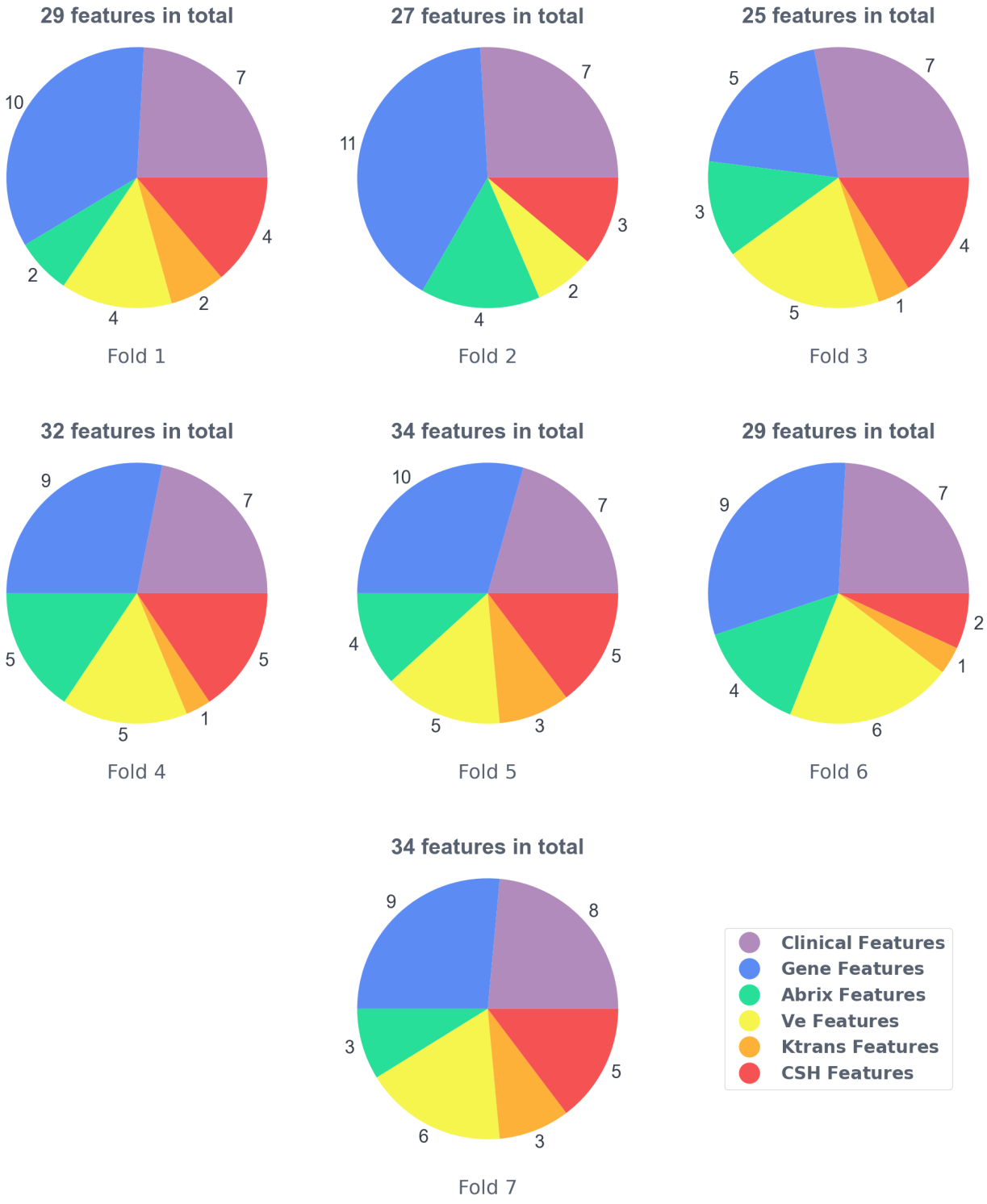
These tables serve as a complementary source of information to Figures 4.3.a–d, offering further insights. Similarly, they demonstrate that as the values of $\tau_1$ and $\tau_2$ increase, fewer features from each block surpass the imposed thresholds. For instance, Table 4.13 reveals that when selecting a value of 0.8 for the ($\tau_1$, $\tau_2$) pair, only three features from the Clinical dataset ('FIGO stage 2groups', 'FIGO stage 2B', and 'FIGO stage 3B') were consistently chosen in all seven folds. The remaining features exhibited lower frequency and stability throughout the feature selection process. Similar exploration can be conducted for other data blocks. In the gene scores block, when lower $\tau$ values are used, specific Hallmark scores like "APICAL SURFACE" or "PANCREAS BETA CELLS," "Dless MINUS Dmore" scores, and "ESTIMATE ImmuneScore" scores are selected in all or most folds. However, as the $\tau$ values increase, only Hallmark scores remain selected from this block. Regarding the pharmacokinetic parameters obtained from DCE-MRI images, it is evident that when the ($\tau_1$, $\tau_2$) pair holds a higher value, such as 0.8, features from the Ve and CSH blocks, specifically "Ve interval 6" and "CSH interval 6," are chosen more frequently compared to other features.

Detailed information regarding the selected features for other $\tau_1$ and $\tau_2$ values not provided in this section can be found in Appendix A.4.



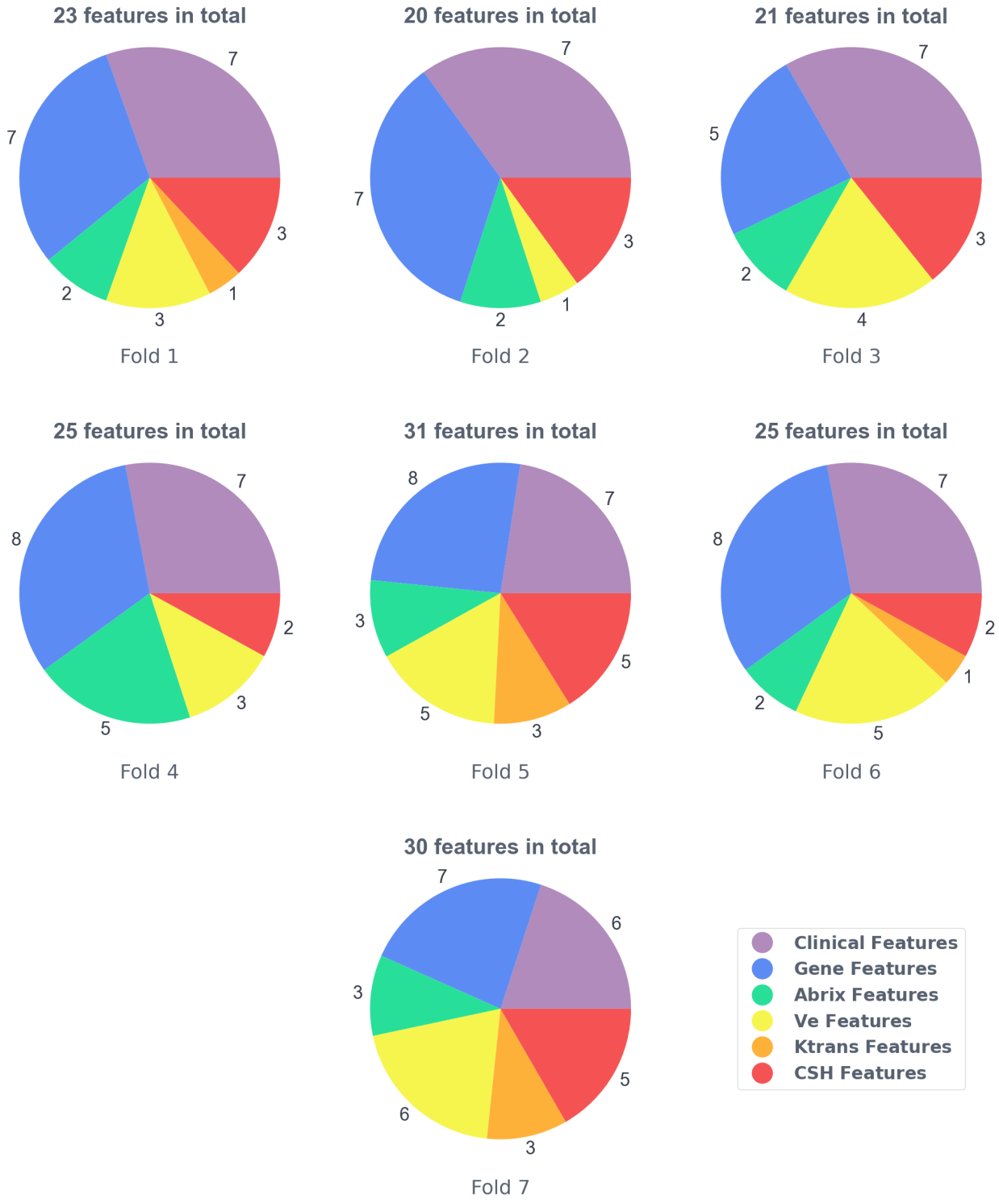

(a)

Figure 4.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2 = 0.2$ across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.



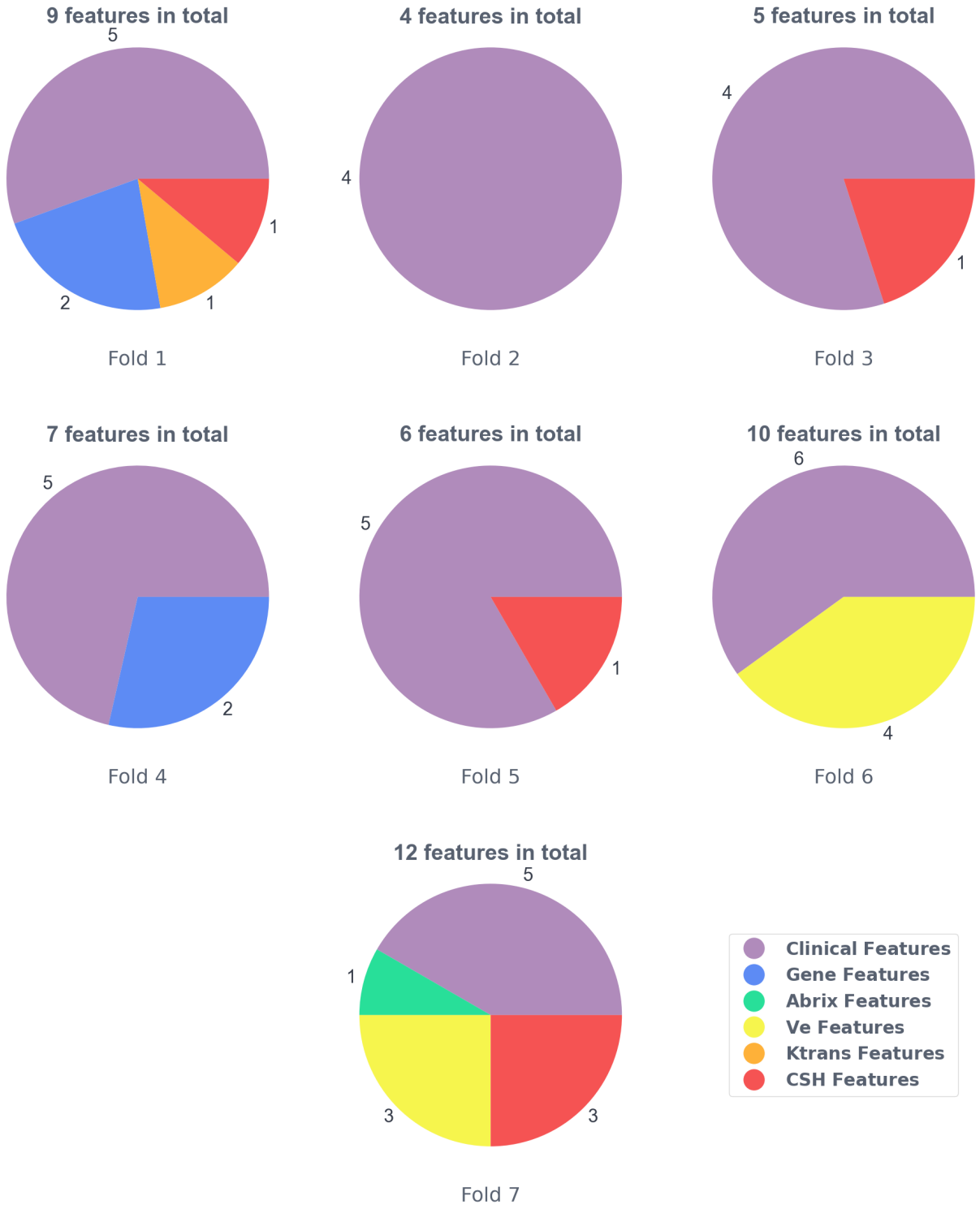

(b)

Figure 4.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2 = 0.3$ across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.




(c)

Figure 4.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2 = 0.4$ across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.




(d)

Figure 4.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2 = 0.8$ across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.

Table 4.10: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.2. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'Tumor volum mm3' | 7 | All Folds |
| | 'LN status' | 7 | All Folds |
| | 'FIGO stage 2groups' | 7 | All Folds |
| | 'n.voxels' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 7 | All Folds |
| | 'FIGO stage 2A' | 1 | 7 |
| | 'FIGO stage 1B1' | 1 | 5 |
| | 'FIGO stage 3A' | 1 | 3 |
| Abrix Features | 'ABrix interval 1' | 7 | All Folds |
| | 'ABrix interval 3' | 7 | All Folds |
| | 'ABrix interval 8' | 7 | All Folds |
| | 'ABrix interval 2' | 6 | 2,3,4,5,6,7 |
| | 'ABrix interval 7' | 3 | 3,4,6 |
| | 'ABrix interval 6' | 2 | 4,6 |
| Ve Features | 'Ve interval 6' | 7 | All Folds |
| | 'Ve interval 7' | 7 | All Folds |
| | 'Ve interval 8' | 7 | All Folds |
| | 'Ve interval 3' | 6 | 1,2,3,4,5,6 |
| | 'Ve interval 5' | 6 | 2,3,4,5,6,7 |
| | 'Ve interval 2' | 4 | 3,4,6,7 |
| | 'Ve interval 4' | 1 | 1 |
| | 'Ve interval 1' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 6 | 1,2,3,4,5,6 |
| | 'Ktrans interval 5' | 3 | 1,5,7 |
| | 'Ktrans interval 6' | 3 | 3,5,7 |
| | 'Ktrans interval 3' | 2 | 3,7 |
| CSH Features | 'CSH interval 1' | 7 | All Folds |
| | 'CSH interval 6' | 7 | All Folds |
| | 'CSH interval 2' | 6 | 1,2,3,4,5,7 |
| | 'CSH interval 7' | 5 | 2,3,4,5,7 |
| | 'CSH interval 5' | 4 | 1,4,5,7 |
| | 'CSH interval 3' | 3 | 2,3,7 |
| Gene Features | 'Score HALLMARK APICAL SURFACE' | 7 | All Folds |
| | 'Score HALLMARK PANCREAS BETA CELLS' | 7 | All Folds |
| | 'Dless MINUS Dmore' | 6 | 1,2,3,4,5,7 |
| | 'Score HALLMARK P53 PATHWAY' | 6 | 1,2,4,5,6,7 |
| | 'Score HALLMARK PI3K AKT MTOR SIGNALING' | 6 | 1,2,3,4,6,7 |
| | 'Score HALLMARK CHOLESTEROL HOMEOSTASIS' | 5 | 1.2.4.5.7 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 5 | 1,4,5,6,7 |
| | 'Score HALLMARK NOTCH SIGNALING' | 5 | 1,4,5,6,7 |
| | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 5 | 1,2,4,5,6 |
| | 'ESTIMATE ImmuneScore' | 5 | 2,4,5,6,7 |
| | 'Score HALLMARK MYC TARGETS V2' | 5 | 2,4,5,6,7 |
| | 'ESTIMATEScore' | 4 | 2,4,5,7 |
| | 'Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY' | 4 | 2,3,5,6 |
| | 'Score HALLMARK TNFA SIGNALING VIA NFKB' | 3 | 1,5,6 |
| | 'Score HALLMARK INTERFERON ALPHA RESPONSE' | 2 | 1,6 |
| | 'Score HALLMARK TGF BETA SIGNALING' | 2 | 1,5 |
| | 'ESTIMATE StromalScore' | 2 | 2,7 |
| | 'Score HALLMARK ALLOGRAFT REJECTION' | 2 | 2,5 |
| | 'Score HALLMARK COAGULATION' | 2 | 2,7 |
| | 'Score HALLMARK G2M CHECKPOINT' | 2 | 2,6 |
| | 'Score HALLMARK BILE ACID METABOLISM' | 2 | 3,4 |
| | 'Score HALLMARK APOPTOSIS' | 1 | 1 |
| | 'Score HALLMARK COMPLEMENT' | 1 | 3 |
| | 'Score HALLMARK HEME METABOLISM' | 1 | 3 |
| | 'Score HALLMARK DNA REPAIR' | 1 | 4 |
| | 'Score HALLMARK FATTY ACID METABOLISM' | 1 | 4 |
| | 'Score HALLMARK ANDROGEN RESPONSE' | 1 | 6 |
| | 'Score HALLMARK E2F TARGETS' | 1 | 6 |
| | 'Score HALLMARK MYC TARGETS V1' | 1 | 6 |
| | 'Score HALLMARK EPITHELIAL MESENCHYMAL TRANSITION' | 1 | 7 |

Table 4.11: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.3. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'Tumor volum mm3' | 7 | All Folds |
| | 'LN status' | 7 | All Folds |
| | 'FIGO stage 2groups' | 7 | All Folds |
| | 'n.voxels' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 7 | All Folds |
| | 'FIGO stage 2A' | 1 | 7 |
| Gene Features | 'Score HALLMARK PANCREAS BETA CELLS' | 7 | All Folds |
| | 'Score HALLMARK APICAL SURFACE' | 6 | 1,2,3,4,5,6 |
| | 'Score HALLMARK NOTCH SIGNALING' | 5 | 1,4,5,6,7 |
| | 'Score HALLMARK P53 PATHWAY' | 5 | 1,4,5,6,7 |
| | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 5 | 1,2,4,5,6 |
| | 'Dless MINUS Dmore' | 4 | 1,4,5,7 |
| | 'Score HALLMARK CHOLESTEROL HOMEOSTASIS' | 4 | 1,2,4,5 |
| | 'Score HALLMARK MYC TARGETS V2' | 4 | 2,5,6,7 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 3 | 1,5,6 |
| | 'Score HALLMARK PI3K AKT MTOR SIGNALING' | 3 | 1,3,7 |
| | 'ESTIMATEScore' | 3 | 2,4,7 |
| | 'ESTIMATE ImmuneScore' | 3 | 2,4,5 |
| | 'Score HALLMARK INTERFERON ALPHA RESPONSE' | 2 | 1,6 |
| | 'ESTIMATE StromalScore' | 2 | 2,7 |
| | 'Score HALLMARK COAGULATION' | 2 | 2,7 |
| | 'Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY' | 2 | 2,3 |
| | 'Score HALLMARK ALLOGRAFT REJECTION' | 1 | 2 |
| | 'Score HALLMARK HEME METABOLISM' | 1 | 3 |
| | 'Score HALLMARK G2M CHECKPOINT' | 1 | 6 |
| Abrix Features | 'ABrix interval 1' | 7 | All Folds |
| | 'ABrix interval 8' | 6 | 1,2,3,4,5,6 |
| | 'ABrix interval 3' | 6 | 2,3,4,5,6,7 |
| | 'ABrix interval 2' | 4 | 2,4,5,7 |
| | 'ABrix interval 6' | 1 | 4 |
| | 'ABrix interval 7' | 1 | 6 |
| Ve Features | 'Ve interval 8' | 7 | All Folds |
| | 'Ve interval 3' | 6 | 1,2,3,4,5,6 |
| | 'Ve interval 6' | 6 | 1,3,4,5,6,7 |
| | 'Ve interval 7' | 6 | 1,3,4,5,6,7 |
| | 'Ve interval 5' | 5 | 3,4,5,6,7 |
| | 'Ve interval 2' | 2 | 6,7 |
| | 'Ve interval 1' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 4 | 1,4,5,6 |
| | 'Ktrans interval 5' | 3 | 1,5,7 |
| | 'Ktrans interval 6' | 3 | 3,5,7 |
| | 'Ktrans interval 3' | 1 | 7 |
| CSH Features | 'CSH interval 6' | 7 | All Folds |
| | 'CSH interval 1' | 6 | 1,2,3,4,5,6 |
| | 'CSH interval 2' | 6 | 1,2,3,4,5,7 |
| | 'CSH interval 5' | 4 | 1,4,5,7 |
| | 'CSH interval 7' | 4 | 3,4,5,7 |
| | 'CSH interval 3' | 1 | 7 |

Table 4.12: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.4. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'LN status' | 7 | All Folds |
| | 'FIGO stage 2groups' | 7 | All Folds |
| | 'n.voxels' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 7 | All Folds |
| | 'Tumor volum mm3' | 6 | 1,2,3,4,5,6 |
| Gene Features | 'Score HALLMARK PANCREAS BETA CELLS' | 7 | All Folds |
| | 'Score HALLMARK APICAL SURFACE' | 6 | 1,2,3,4,5,6 |
| | 'Score HALLMARK P53 PATHWAY' | 5 | 1,4,5,6,7 |
| | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 5 | 1,2,4,5,6 |
| | 'Score HALLMARK NOTCH SIGNALING' | 4 | 1,4,5,6 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 3 | 1,5,6 |
| | 'Score HALLMARK PI3K AKT MTOR SIGNALING' | 3 | 1,3,7 |
| | 'ESTIMATE ImmuneScore' | 3 | 2,4,5 |
| | 'Score HALLMARK MYC TARGETS V2' | 3 | 2,6,7 |
| | 'Dless MINUS Dmore' | 3 | 4,5,7 |
| | 'ESTIMATEScore' | 2 | 2,7 |
| | 'ESTIMATE StromalScore' | 2 | 2,7 |
| | 'Score HALLMARK HEME METABOLISM' | 1 | 3 |
| | 'Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY' | 1 | 3 |
| | 'Score HALLMARK CHOLESTEROL HOMEOSTASIS' | 1 | 4 |
| | 'Score HALLMARK G2M CHECKPOINT' | 1 | 6 |
| Abrix Features | 'ABrix interval 8' | 6 | 1,2,3,4,5,6 |
| | 'ABrix interval 3' | 5 | 3,4,5,6,7 |
| | 'ABrix interval 1' | 4 | 1,4,5,7 |
| | 'ABrix interval 2' | 3 | 2,4,7 |
| | 'ABrix interval 6' | 1 | 4 |
| Ve Features | 'Ve interval 8' | 7 | All Folds |
| | 'Ve interval 7' | 6 | 1,3,4,5,6,7 |
| | 'Ve interval 3' | 5 | 1,3,4,5,6 |
| | 'Ve interval 6' | 4 | 3,5,6,7 |
| | 'Ve interval 5' | 3 | 5,6,7 |
| | 'Ve interval 1' | 1 | 7 |
| | 'Ve interval 2' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 3 | 1,5,6 |
| | 'Ktrans interval 5' | 2 | 5,7 |
| | 'Ktrans interval 6' | 2 | 5,7 |
| | 'Ktrans interval 3' | 1 | 7 |
| CSH Features | 'CSH interval 6' | 7 | All Folds |
| | 'CSH interval 1' | 6 | 1,2,3,4,5,6 |
| | 'CSH interval 5' | 3 | 1,5,7 |
| | 'CSH interval 2' | 3 | 2,5,7 |
| | 'CSH interval 7' | 3 | 3,5,7 |
| | 'CSH interval 3' | 1 | 7 |

Table 4.13: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.8. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| **Block** | **Feature Name** | **Count In Folds** | **Fold Details** |
|---|---|---|---|
| Clinical Features | 'FIGO stage 2groups' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 5 | 1,4,5,6,7 |
| | 'n.voxels' | 5 | 2,3,4,5,6 |
| | 'LN status' | 2 | 1,7 |
| | 'Tumor volum mm3' | 1 | 6 |
| Gene Features | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 2 | 1,4 |
| | 'Score HALLMARK NOTCH SIGNALING' | 1 | 1 |
| | 'Score HALLMARK PANCREAS BETA CELLS' | 1 | 4 |
| Abrix Features | 'ABrix interval 2' | 1 | 7 |
| Ve Features | 'Ve interval 6' | 2 | 6,7 |
| | 'Ve interval 3' | 1 | 6 |
| | 'Ve interval 5' | 1 | 6 |
| | 'Ve interval 7' | 1 | 6 |
| | 'Ve interval 2' | 1 | 7 |
| | 'Ve interval 8' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 1 | 1 |
| CSH Features | 'CSH interval 6' | 3 | 1,5,7 |
| | 'CSH interval 1' | 1 | 1 |
| | 'CSH interval 2' | 1 | 7 |
| | 'CSH interval 3' | 1 | 7 |
| | 'CSH interval 3' | 1 | 7 |
| | 'CSH interval 5' | 1 | 7 |

# Chapter 5

# Discussion

## 5.1 Data

Insufficient data poses challenges to the applications of machine learning in diagnosing, treating, and predicting treatment outcomes for cancer tumors. ML models require sufficient data to learn intricate patterns and develop precise predictive models. When data is scarce, it can lead to model overfitting, where the model performs well on the training data but fails to generalize to unseen cases. This compromises the accuracy and reliability of ML models. However, the availability of such data is often limited due to factors such as a small number of patients, issues with data consistency and confidentiality (as cross-border data sharing to generate larger databases is not easily feasible), data fragmentation across various systems and institutions, or the absence of standardized mechanisms for data collection. These limitations require more advanced methodologies to tackle the challenges that inadequate data presents.

This study utilized a multi-source dataset comprising pharmacokinetic parameters extracted from DCE-MR images, gene scores, and clinical data to address the limited available data. Using multiple data sources brought both advantages and disadvantages. On the positive side, pooling information from various sources offered a larger amount of data, enabling the training of models with better generalization capabilities to unseen data. However, it also posed challenges related to the diverse nature of the data sources. Firstly, the lack of accessible data for numerous patients across different data blocks led to the decision to exclude incomplete records. Secondly, by employing a multi-source dataset, quantifying the impact of each source became more complex, and the higher dataset dimensionality necessitated engaging in more complex data preprocessing.

By acknowledging these challenges associated with the multi-source nature of the data and the specific data types involved, this study aimed to mitigate potential limitations while leveraging the advantages of utilizing a comprehensive dataset. Furthermore, in this research, the prediction of tumor recurrence was examined regardless of whether it was locoregional or distant recurrence. Nonetheless, it is also possible to explore the prediction of these specific statuses individually.

## 5.2 Data Preprocessing

An imbalance or skewed distribution of classes or categories within a dataset can impose several problems on machine learning algorithms. Machine learning models tend to be biased toward the majority class and prioritize accuracy on the dominant class while performing poorly on minority classes. This can lead to poor detection or prediction performance for the minority class(es). Since medical data is often unbalanced, and here the preference of minority samples is sometimes even higher than the majority samples, it is essential to investigate how to reduce biased model performance caused by imbalanced data. This study found that out of the patients who underwent therapy, 63% fully recovered, while 37% experienced tumor relapse within a 5-year follow-up period. Instead of randomly duplicating samples from the minority class, the SMOTE method was employed as a more sophisticated approach to simulate minority samples. The results indicate that using this method significantly improves the performance of some machine-learning models. However, it is vital to acknowledge that the SMOTE method may have its limitations and issues.

Aside from the evident increase in model runtime shown in Table 4.9, SMOTE can introduce synthetic instances that are too similar to existing minority class samples, leading to overgeneralization. This can cause the model to become overly confident in its predictions, potentially compromising its ability to distinguish between the minority and majority classes accurately. By creating synthetic samples, SMOTE effectively increases the size of the minority class. Suppose the synthetic samples are not carefully generated. In that case, the risk of overfitting may increase, especially if the generated samples do not accurately represent the actual distribution of the minority class or if the minority class is inherently similar or overlaps with the majority class. Furthermore, SMOTE may amplify the impact of noise or outliers in the minority class. Since SMOTE generates synthetic samples by interpolating between existing minority class samples, noisy or outlier instances may also be used in the interpolation process, potentially creating misleading synthetic samples. In their study, Fernandez et al. [89] introduce several extensions of SMOTE that aim to examine and resolve the issues mentioned above and to create more robust ML models.

Furthermore, to better utilize the small available data, minimize bias in performance evaluation, and obtain reliable performance estimates, this study prioritized Stratified K-Fold Cross-Validation for dataset splitting rather than relying on a single Train-Test-Split. This way, by preserving the relative distribution of samples from each class in the original dataset, the study ensured that more training and testing subsets (specifically 7, in this case) were made available to the models. While implementing this approach alone can yield the benefits mentioned above, an additional possibility is to employ the Repeated Stratified K-Fold Cross-Validation technique, which involves repeating the partitioning process $N \times K$ times instead of $K$ times. On the one hand, this approach might offer additional advantages, including increased model robustness, enhanced evaluation of model stability, improved hyperparameter tuning, and even more efficient utilization of the available data. However, on the other hand, it may impact runtime due to training a higher number of elementary models.

This research employed the One-Hot encoding approach for preprocessing categorical features. This method represents each category as a binary feature, effectively avoiding the introduction of ordinality or numerical relationships. However, it can lead to a high-dimensional and sparse representation, mainly when dealing with categorical features with numerous unique values. Other alternative options like Count Encoding and Label Encoding could be considered to

assess if they improve classification performance. Count Encoding provides information on category frequency but may not be suitable when categories have similar counts, as they would be encoded with the same value. Label Encoding offers a compact representation by assigning a unique numerical label to each category. However, it introduces an arbitrary ordinal relationship between categories that can mislead the model by assuming a meaningful order. Exploring these alternative encoding techniques could help to determine if they yield better classification performance in this study.

## 5.3 Feature Selection

Feature selection serves various objectives, including reducing complexity, enhancing interpretability, facilitating efficient data processing, or improving model performance in machine learning applications. This study employed the RENT feature selection technique, which falls under the embedded feature selector category and utilizes Elastic Net regularization to perform feature selection. In this study, RENT helped accomplish some of these objectives while others did not achieve. Figure 4.3.a illustrates that employing even a small RENT criteria value proved effective in detecting and eliminating over 42% of unnecessary or redundant features. This led to a streamlined and more manageable set of features. By reducing the dataset dimensionality, the subsequent modeling algorithms' complexity was significantly decreased, resulting in considerably faster training times.

RENT also facilitated a deeper comprehension of the underlying factors contributing to predictions. By referring to Tables 4.10–4.13, it was revealed that by segregating the influential features based on individual data blocks, the analysis and explanation of each block's impact on the model's outcomes were made more accessible. Specifically, the findings indicated that the clinical data block played a more substantial role in predicting patients' treatment outcomes than other blocks. This was evident since clinical features were consistently chosen over others as the $\tau$ values increased. However, regarding enhancing model performance, the post hoc analysis demonstrated that RENT did not exhibit a statistically significant impact on the obtained results. Nevertheless, it was observed that RENT did not have a detrimental effect on the results either.

Despite its strengths, RENT also has limitations. Its efficacy can be restricted when dealing with strongly non-linear problems due to its reliance on linear or logistic regression for feature selection. The limitations of these methods stem from their inherent assumptions and functional forms, which primarily allow them to capture linear or additive effects between variables. Consequently, when confronted with complex relationships, linear and logistic regression may struggle to adequately capture the underlying patterns and relationships in strongly non-linear problems. In addition, there are situations where expert knowledge can be highly advantageous in selecting the most crucial features, particularly in domains like medicine or life sciences that prioritize model interpretability. However, the RENT method, similar to numerous other feature selection techniques, does not incorporate the input or expertise of domain experts during the feature selection process.

The User-Guided Bayesian Framework for Feature Selection (UBayFS) method is a potential alternative—a recently developed ensemble feature selection technique operating within a Bayesian statistical framework. This method takes into account two sources of information: data and domain knowledge. It constructs an ensemble of feature selectors based on a multi-

nomial likelihood model derived from the data. Additionally, the user can guide UBayFS by assigning weights to features and imposing penalties on feature blocks or combinations using a Dirichlet-type prior distribution, thereby incorporating domain knowledge. The specific functionality and potential advantages of utilizing UBayFS can be explored in detail within the official paper [97]. However, it is essential to note that this method may also introduce particular challenges. For instance, the confidence level of experts in their applied opinions during the feature selection process and the resolution of potential conflicts between domain knowledge and data-driven information are important considerations to address.

This research noted that the hyperparameters chosen for the models during cross-validation were also employed to train the models that utilized the features selected by RENT. It is important to highlight from Table 4.4 that a l1-ratio value of zero indicates that this regularization method, which can also serve as a feature selection technique, was not employed again. Consequently, the features were not further limited or filtered by LR. However, the Entropy criterion in RF can be viewed as an additional feature selector or constraint on top of the features selected by RENT. In addition to the benefits the Entropy criterion brings to the RF classifier, such as maintaining diversity and information richness across the ensemble of decision trees by encouraging balanced splits and preventing overfitting, it also complicates the interpretability of RF during this study. This is because, by utilizing this criterion, RF tends to prioritize features that offer the highest information gain, possibly making the previously selected features even more filtered.

Feature correlation is another problem that can make it challenging to determine the true importance of each feature in a machine-learning model. When two or more features are highly correlated, they provide similar information to the model. As a result, the model may assign similar weights or importance to these features, making it difficult to distinguish their contributions. This redundancy can lead to overemphasizing certain features while neglecting others. Since one of the goals of this study was to identify the most informative indicators for predicting the patients' treatment outcomes, the research aimed to incorporate all available features into the ML models and the RENT feature selection technique. Nevertheless, it may be worthwhile exploring the potential impact of excluding highly correlated features on the classification performance of the models in subsequent investigations.

## 5.4 Outcome Prediction using Machine Learning

Given that this study focuses on a binary issue, the selection of classifiers may significantly impact the outcomes. The initial choice was to utilize logistic regression due to its noteworthy advantages. Logistic regression is a widely used algorithm that is relatively straightforward and comprehensible compared to more intricate classification algorithms. The coefficients in the logistic regression model can be interpreted as the impact of each feature on the likelihood of belonging to a specific class, resulting in more easily understandable outcomes. This model can handle noisy data and outliers reasonably, thanks to the sigmoid activation function that compresses extreme values towards the boundaries of 0 and 1. Considering Table 4.9, logistic regression also exhibits computational efficiency, as it demonstrates the second shortest average training time when utilizing all features and achieves the fastest runtime after reducing features using the RENT method. However, LR assumes a linear relationship between the independent variables and the log-odds of the target variable. If the actual relationship is highly non-linear, logistic regression may not capture it effectively, leading to suboptimal performance.

The subsequent option was the random forest, a popular ensemble learning algorithm known for its various strengths. By combining multiple decision trees and averaging their predictions, the random forest algorithm tends to achieve high accuracy. It also exhibits robustness against overfitting and outliers due to its independent construction of each decision tree, resulting in less influence from outliers on the overall prediction because of the averaging effect. This characteristic may explain its relatively higher stability in performance when faced with diverse subsets of data in each fold, distinguishing it from the other two models (as shown in Figure 4.2). Furthermore, random forests can handle complex datasets with non-linear relationships and capture a wide range of patterns. However, it has its limitations. Interpreting the random forest can be challenging compared to simpler models like logistic regression. Each decision tree within the random forest learns from different random subsets of features and observations, making it more intricate to comprehend the combined impact of multiple trees and their interactions. Additionally, random forests can be computationally expensive, particularly when involving hyperparameter tuning. Table 4.9 illustrated that random forest was noticeably the slowest model.

Support vector machine was the third choice because it generally performs well on short-wide datasets, where the number of features is larger than the number of samples. This is commonly referred to as the "curse of dimensionality," SVMs address this issue using the kernel trick. Outliers in the data have less impact on SVMs because they aim to maximize the margin between classes, reducing the influence of individual data points. By utilizing the regularization parameter (C), SVMs control the trade-off between achieving maximum margin and minimizing classification errors, preventing overfitting by penalizing misclassified samples. Nevertheless, SVMs require careful tuning of various parameters, including the choice of kernel and the regularization parameter (C). The model's performance is sensitive to these settings, and improper tuning can lead to suboptimal outcomes. Additionally, SVM may encounter challenges when working with imbalanced datasets. Since SVMs prioritize correctly classifying samples near the decision boundary, they may focus more on the majority class and neglect the minority class in imbalanced datasets. Consequently, the classification performance for the minority class tends to be poor, as observed in this study. When SMOTE was not employed as a class balancing technique, the SVM models assigned all samples to the majority class, resulting in MCC scores of zero.

The chosen evaluation metric in this study, the MCC Score, has several advantages. It addresses the issue of imbalanced datasets by considering the balance between positive and negative class samples. The MCC score considers both correct and incorrect predictions across all four categories (TP, TN, FP, FN), and it is specifically designed for binary classification tasks. However, a limitation of the MCC score is that it needs to be adapted or extended for multi-class classification tasks. Additionally, interpreting the MCC score can be challenging without context or a baseline for comparison, as there is no clear definition of what constitutes a good or bad score.

Apart from Precision, Recall, and F1-score, alternative evaluation metrics, such as the Area Under the ROC Curve (AUROC), exist. The AUROC measures the trade-off between the true positive rate (TPR) and the false positive rate (FPR) at different classification thresholds, as represented by the receiver operating characteristic (ROC) curve. It provides a single-value metric that summarizes the integral of this curve. AUROC is particularly useful when focusing on overall classification performance and the model's ability to distinguish between positive and negative classes, regardless of the chosen threshold. However, AUROC does not address the class imbalance or provide insights into the model's performance at a specific threshold.

Overall, given the limited sample size, the developed models demonstrated remarkable performance. Based on the information discussed in the preceding and current chapters, as well as the evaluations and statistical analyses, it appears that the SVM model is the preferred choice compared to the other two models. This is due to the SVM model achieving the highest MCC score and not having significant drawbacks in the other two models. Unlike logistic regression, the SVM model overcomes the limitation of capturing relationships in non-linear data by utilizing the kernel trick. Furthermore, compared to the random forest model, the computational cost of SVM is considerably lower. When employing SVM, it is crucial to apply the SMOTE method to address the issue of classifying minority class samples and employ the RENT technique to enhance the interpretability of the models by introducing the most significant features rather than improving classification performance. The research findings also indicate that among the various data blocks incorporated into the models, the clinical data block contains the most influential features for predicting patients' treatment outcomes. Ultimately, it should be noted that complete reliance on the models' results without human supervision is still not feasible. Continued research is necessary to enhance the data quality used for model training, explore different advanced ML algorithms, and improve overall performance in predicting treatment outcomes.

# Chapter 6

# Conclusion

Machine learning's abilities to discover intricate data relationships, identify patterns in data of various dimensions, continuously learn and adapt to new research findings, and process information quickly for real-time decision support have rendered it valuable in medical research. Several research studies have been conducted to investigate the applications of machine learning in early detection, imaging analysis, prognosis, and outcome prediction of cervical cancer, one of the prevalent cancers affecting women globally. This research aimed to create three distinct machine learning models —logistic regression, random forest, and support vector machine— to predict the overall treatment outcomes of patients diagnosed with locally advanced cervical cancer within a 60-month timeframe. In addition, the impact of employing the RENT feature selection method and the SMOTE class balance technique on enhancing the performance and interpretability of the models was also explored.

The training data for the machine learning models consisted of diverse sources, including pharmacokinetic parameters extracted from DCE-MR images, gene scores, and clinical data. A dataset comprising 67 patients and 97 features was obtained after establishing a data preprocessing pipeline. The models underwent multiple training iterations: one before implementing the RENT and SMOTE techniques and another after applying these methods individually or in combination. The results obtained from the developed models were initially compared with the outcomes of three random classifiers, serving as the first baseline. Subsequently, the results from the models trained without utilizing the RENT and SMOTE methods were used as the second benchmark for the more complex models that incorporated these techniques.

Given the imbalanced class distribution, the MCC score was chosen as the classification performance metric. The findings revealed that all the models developed in this research (except for SVM when the SMOTE method was not employed) outperformed the three random classifiers. The best results were achieved among the developed models when the RENT and SMOTE methods were utilized simultaneously, with an MCC score of 0.411 for LR, 0.415 for RF, and 0.444 for SVM. These scores demonstrate enhancements of 27.6% for LR, 19.9% for RF, and an impressive 44,300% for SVM when compared to the models not employing the RENT and SMOTE techniques. However, as the best scores were comparable, post hoc statistical analysis employing ANOVA and Tukey's HSD tests was conducted and indicated that between the classifier, SMOTE, and RENT, the first two factors and their interaction exhibited statistically significant effects on the MCC scores, with p-values of 0.005, 0.02, and 0.002, respectively. Furthermore, according to Tukey's HSD test, the SVM model demonstrated significant differences

from other models when the SMOTE method was not utilized. This indicates that the SMOTE method had a more substantial impact on the SVM model than the others.

Overall, while the obtained MCC scores are not the highest, they hold significance given the limited number of samples. This suggests that the presented models possess the potential for further investigation, improvement, and utilization as auxiliary tools for medical professionals. Furthermore, to enhance the interpretability of the models in predicting treatment outcomes, the study employed the RENT feature selection method to introduce the most influential features. The findings demonstrated that as the RENT criteria values increased, implying more significant restrictions on the feature selection process, the clinical block features such as "FIGO_stage_2groups," "FIGO_stage_3B," "FIGO_stage_2B," and "n.voxels" were more frequently selected compared to features from other data blocks. This suggests that clinical data block features are vital in predicting patient treatment outcomes.

# Bibliography


[1] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] Fahmi Khalifa, Ahmed Soliman, Ayman El-Baz, Mohamed Abou El-Ghar, Tarek El-Diasty, Georgy Gimel'farb, Rosemary Ouseph, and Amy C. Dwyer, "Models and methods for analyzing dce-mri: A review," *Medical Physics*, vol. 41, no. 12, pp. 124301, 2014.

[3] 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*, vol. 9, pp. 152333–152346, 2021.

[4] CS Fjeldbo, *Data for block-analyses: description of files and variables*, Oslo University Hospital, November 2019.

[5] Robert A. Weinberg, "How cancer arises," *Scientific American*, vol. 275, no. 3, pp. 62–70, 1996.

[6] Harold Varmus, "The new era in cancer research," *Science*, vol. 312, no. 5777, pp. 1162–1165, 2006.

[7] Chris Wild, Elisabete Weiderpass, and Bernard W Stewart, *World cancer report: cancer research for cancer prevention*, International Agency for Research on Cancer, 2020.

[8] Ruchit Shah, Chizoba Nwankwo, Youngmin Kwon, and Shelby L. Corman, "Economic and humanistic burden of cervical cancer in the united states: Results from a nationally representative survey," *Journal of Women's Health*, vol. 29, no. 6, pp. 799–805, 2020, PMID: 31967943.

[9] 72nd session Regional Committee for Europe, "Seventy-second regional committee for europe: Tel aviv, 12–14 september 2022: roadmap to accelerate the elimination of cervical cancer as a public health problem in the who european region 2022–2030," p. 8 p., 2022.

[10] Lisa Barbera and Gillian Thomas, "Management of early and locally advanced cervical cancer," *Seminars in Oncology*, vol. 36, no. 2, pp. 155–169, 2009, Gynecologic Cancer Update.

[11] Nicola Ielapi, Michele Andreucci, Noemi Licastro, Teresa Faga, Raffaele Grande, Gianluca Buffone, Sabrina Mellace, Paolo Sapienza, and Raffaele Serra, "Precision medicine and precision nursing: The era of biomarkers and precision health," *International Journal of General Medicine*, vol. 13, pp. 1705–1711, 2020.

[12] Mohammad H. Bagheri, Mark A. Ahlman, Liza Lindenberg, Baris Turkbey, Jeffrey Lin, Ali Cahid Civelek, Ashkan A. Malayeri, Piyush K. Agarwal, Peter L. Choyke, Les R. Folio, and Andrea B. Apolo, "Advances in medical imaging for the diagnosis and management of common genitourinary cancers," *Urologic Oncology: Seminars and Original Investigations*, vol. 35, no. 7, pp. 473–491, 2017, Seminar on Updates in Management of Urothelial Carcinoma.

[13] Xin Hou, Guangyang Shen, Liqiang Zhou, Yinuo Li, Tian Wang, and Xiangyi Ma, "Artificial intelligence in cervical cancer screening and diagnosis," *Frontiers in Oncology*, vol. 12, 2022.

[14] Charis Bourgioti, Konstantinos Chatoupis, and Lia Angela Moulopoulos, "Current imaging strategies for the evaluation of uterine cervical cancer," *World J Radiol*, vol. 8, no. 4, pp. 342–354, Apr. 2016.

[15] Bariş Türkbey, David Thomasson, Yuxi Pang, Marcelino Bernardo, and Peter L Choyke, "The role of dynamic contrast-enhanced MRI in cancer diagnosis and treatment," *Diagn Interv Radiol*, vol. 16, no. 3, pp. 186–192, Nov. 2009.

[16] A Fyles, M Milosevic, D Hedley, M Pintilie, W Levin, L Manchul, and R P Hill, "Tumor hypoxia has independent predictor impact only in patients with node-negative cervix cancer," *J Clin Oncol*, vol. 20, no. 3, pp. 680–687, Feb. 2002.

[17] Mark A Zahra, Li Tee Tan, Andrew N Priest, Martin J Graves, Mark Arends, Robin A F Crawford, James D Brenton, David J Lomas, and Evis Sala, "Semiquantitative and quantitative dynamic contrast-enhanced magnetic resonance imaging measurements predict radiation response in cervix cancer," *Int J Radiat Oncol Biol Phys*, vol. 74, no. 3, pp. 766–773, Nov. 2008.

[18] Jung Jae Park, Chan Kyo Kim, Sung Yoon Park, Arjan W Simonetti, Eunju Kim, Byung Kwan Park, and Seung Jae Huh, "Assessment of early response to concurrent chemoradiotherapy in cervical cancer: value of diffusion-weighted and dynamic contrast-enhanced MR imaging," *Magn Reson Imaging*, vol. 32, no. 8, pp. 993–1000, June 2014.

[19] Turid Torheim, Aurora R Groendahl, Erlend K F Andersen, Heidi Lyng, Eirik Malinen, Knut Kvaal, and Cecilia M Futsaether, "Cluster analysis of dynamic contrast enhanced MRI reveals tumor subregions related to locoregional relapse for cervical cancer patients," *Acta Oncol*, vol. 55, no. 11, pp. 1294–1298, Aug. 2016.

[20] Nina A Mayr, Jian Z Wang, Dongqing Zhang, John C Grecula, Simon S Lo, David Jaroura, Joseph Montebello, Hualin Zhang, Kaile Li, Lanchun Lu, Zhibin Huang, Jeffery M Fowler, Dee H Wu, Michael V Knopp, and William T C Yuh, "Longitudinal changes in tumor perfusion pattern during the radiation therapy course and its clinical impact in cervical cancer," *Int J Radiat Oncol Biol Phys*, vol. 77, no. 2, pp. 502–508, Sept. 2009.

[21] Bariş Türkbey, David Thomasson, Yuxi Pang, Marcelino Bernardo, and Peter L Choyke, "The role of dynamic contrast-enhanced MRI in cancer diagnosis and treatment," *Diagn Interv Radiol*, vol. 16, no. 3, pp. 186–192, Nov. 2009.

[22] N. Arteaga-Marrero, C. B. Rygh, J. F. Mainou-Gomez, K. Nylund, D. Roehrich, J. Heggdal, P. Matulaniec, O. H. Gilja, R. K. Reed, L. Svensson, N. Lutay, and D. R. Olsen, "Multimodal approach to assess tumour vasculature and potential treatment effect with dce-us and dce-mri quantification in cwr22 prostate tumour xenografts," *Contrast Media & Molecular Imaging*, vol. 10, no. 6, pp. 428–437, 2015.

[23] Craig B. Markwardt, "Non-linear least squares fitting in idl with mpfit," 2009.

[24] Toru Chikui, Makoto Obara, Arjan W. Simonetti, Masahiro Ohga, Shoichi Koga, Shintaro Kawano, Yoshio Matsuo, Takeshi Kamintani, Tomoko Shiraishi, Erina Kitamoto, Katsumasa Nakamura, and Kazunori Yoshiura, "The principal of dynamic contrast enhanced mri, the method of pharmacokinetic analysis, and its application in the head and neck region," *International Journal of Dentistry*, vol. 2012, pp. 480659, Oct 2012.

[25] Steven P. Sourbron and David L. Buckley, "On the scope and interpretation of the tofts models for dce-mri," *Magnetic Resonance in Medicine*, vol. 66, no. 3, pp. 735–745, 2011.

[26] Erlend K.F. Andersen, Knut Håkon Hole, Kjersti V. Lund, Kolbein Sundfør, Gunnar B. Kristensen, Heidi Lyng, and Eirik Malinen, "Pharmacokinetic parameters derived from dynamic contrast enhanced mri of cervical cancers predict chemoradiotherapy outcome," *Radiotherapy and Oncology*, vol. 107, no. 1, pp. 117–122, 2013.

[27] Erlend K. F. Andersen, Gunnar B. Kristensen, Heidi Lyng, and Eirik Malinen, "Pharmacokinetic analysis and k-means clustering of dcemr images for radiotherapy outcome prediction of advanced cervical cancers," *Acta Oncologica*, vol. 50, no. 6, pp. 859–865, 2011, PMID: 21767185.

[28] Scott I.K. Semple, Vanessa N. Harry, David E. Parkin, and Fiona J. Gilbert, "A combined pharmacokinetic and radiologic assessment of dynamic contrast-enhanced magnetic resonance imaging predicts response to chemoradiation in locally advanced cervical cancer," *International Journal of Radiation Oncology*Biology*Physics*, vol. 75, no. 2, pp. 611–617, 2009.

[29] Miriam Y. Salib, James H. B. Russell, Victoria R. Stewart, Siham A. Sudderuddin, Tara D. Barwick, Andrea G. Rockall, and Nishat Bharwani, "2018 figo staging classification for cervical cancer: Added benefits of imaging," *RadioGraphics*, vol. 40, no. 6, pp. 1807–1822, 2020, PMID: 32946322.

[30] Suzanne M Bleker, Shandra Bipat, Anje M Spijkerboer, Jacobus van der Velden, Jaap Stoker, and Gemma G Kenter, "The negative predictive value of clinical examination with or without anesthesia versus magnetic resonance imaging for parametrial infiltration in cervical cancer stages IB1 to IIA," *Int J Gynecol Cancer*, vol. 23, no. 1, pp. 193–198, Jan. 2013.

[31] Nilu Malpani Dhoot, Vinay Kumar, Atul Shinagare, Amal Chandra Kataki, Debabrata Barmon, and Utpal Bhuyan, "Evaluation of carcinoma cervix using magnetic resonance imaging: correlation with clinical FIGO staging and impact on management," *J Med Imaging Radiat Oncol*, vol. 56, no. 1, pp. 58–65, Feb. 2012.

[32] Zdenko Kraljević, Klaudija Visković, Mario Ledinsky, Dijana Zadravec, Ivan Grbavac, Marijana Bilandzija, Hrvojka Soljacić-Vranes, Krunoslav Kuna, Ksenija Klasnić, and Ivan Krolo, "Primary uterine cervical cancer: correlation of preoperative magnetic resonance imaging and clinical staging (FIGO) with histopathology findings," *Coll Antropol*, vol. 37, no. 2, pp. 561–568, June 2013.

[33] T V Prasad, S Thulkar, S Hari, D N Sharma, and S Kumar, "Role of computed tomography (CT) scan in staging of cervical carcinoma," *Indian J Med Res*, vol. 139, no. 5, pp. 714–719, May 2014.

[34] Perry W Grigsby, "The prognostic value of PET and PET/CT in cervical cancer," *Cancer Imaging*, vol. 8, no. 1, pp. 146–155, July 2008.

[35] Paulina Sodeikat, Massimiliano Lia, Mireille Martin, Lars-Christian Horn, Michael Höckel, Bahriye Aktas, and Benjamin Wolf, "The importance of clinical examination under general anesthesia: Improving parametrial assessment in cervical cancer patients," *Cancers (Basel)*, vol. 13, no. 12, June 2021.

[36] Jin-Rong Qu, Lei Qin, Xiang Li, Jun-Peng Luo, Jing Li, Hong-Kai Zhang, Li Wang, Nan-Nan Shao, Shou-Ning Zhang, Yan-Le Li, Cui-Cui Liu, and Hai-Liang Li, "Predicting parametrial invasion in cervical carcinoma (stages ib1, ib2, and iia): Diagnostic accuracy of t2-weighted imaging combined with dwi at 3 t," *American Journal of Roentgenology*, vol. 210, no. 3, pp. 677–684, 2018, PMID: 29323549.

[37] Weifeng Zhang, Chunlin Chen, Ping Liu, Weili Li, Min Hao, Weidong Zhao, Anwei Lu, and Yan Ni, "Impact of pelvic mri in routine clinical practice on staging of ib1–iia2 cervical cancer," *Cancer Management and Research*, vol. 11, pp. 3603–3609, 2019, PMID: 31118782.

[38] Katharina Hancke, Volker Heilmann, Peter Straka, Rolf Kreienberg, and Christian Kurzeder, "Pretreatment staging of cervical cancer: Is imaging better than palpation?," *Annals of Surgical Oncology*, vol. 15, no. 10, pp. 2856–2861, Oct 2008.

[39] Hedvig Hricak, Constantine Gatsonis, Dennis S. Chi, Marco A. Amendola, Kathy Brandt, Lawrence H. Schwartz, Susan Koelliker, Evan S. Siegelman, Jeffrey J. Brown, Robert B. McGhee, Revathy Iyer, Kenneth M. Vitellas, Bradley Snyder, Harry J. Long, James V. Fiorica, and Donald G. Mitchell, "Role of imaging in pretreatment evaluation of early invasive cervical cancer: Results of the intergroup study american college of radiology imaging network 6651–gynecologic oncology group 183," *Journal of Clinical Oncology*, vol. 23, no. 36, pp. 9329–9337, 2005, PMID: 16361632.

[40] Tord Hompland, Christina Sæten Fjeldbo, and Heidi Lyng, "Tumor hypoxia as a barrier in cancer therapy: Why levels matter," *Cancers*, vol. 13, no. 3, 2021.

[41] Saskia E Rademakers, Paul N Span, Johannes HAM Kaanders, Fred CGJ Sweep, Albert J van der Kogel, and Johan Bussink, "Molecular aspects of tumour hypoxia," *Molecular oncology*, vol. 2, no. 1, pp. 41–53, 2008.

[42] Constantinos Koumenis and Bradly G Wouters, "" translating" tumor hypoxia: unfolded protein response (upr)-dependent and upr-independent pathways," *Molecular Cancer Research*, vol. 4, no. 7, pp. 423–436, 2006.

[43] Michael Hockel and Peter Vaupel, "Tumor Hypoxia: Definitions and Current Clinical, Biologic, and Molecular Aspects," *JNCI: Journal of the National Cancer Institute*, vol. 93, no. 4, pp. 266–276, 02 2001.

[44] Christina S. Fjeldbo, Cathinka H. Julin, Malin Lando, Malin F. Forsberg, Eva-Katrine Aarnes, Jan Alsner, Gunnar B. Kristensen, Eirik Malinen, and Heidi Lyng, "Integrative Analysis of DCE-MRI and Gene Expression Profiles in Construction of a Gene Classifier for Assessment of Hypoxia-Related Risk of Chemoradiotherapy Failure in Cervical Cancer," *Clinical Cancer Research*, vol. 22, no. 16, pp. 4067–4076, 08 2016.

[45] Amato J Giaccia, "Molecular radiobiology: the state of the art," *Journal of Clinical Oncology*, vol. 32, no. 26, pp. 2871, 2014.

[46] E David Crawford, Mark C Scholz, Ashok J Kar, Jeffrey E Fegan, Abebe Haregewoin, Rajesh R Kaldate, and Michael K Brawer, "Cell cycle progression score and treatment decisions in prostate cancer: results from an ongoing registry," *Current medical research and opinion*, vol. 30, no. 6, pp. 1025–1031, 2014.

[47] Yasuto Naoi and Shinzaburo Noguchi, "Multi-gene classifiers for prediction of recurrence in breast cancer patients," *Breast cancer*, vol. 23, pp. 12–18, 2016.

[48] Cathinka Halle, Erlend Andersen, Malin Lando, Eva-Katrine Aarnes, Grete Hasvold, Marit Holden, Randi G Syljuåsen, Kolbein Sundfør, Gunnar B Kristensen, Ruth Holm, Eirik Malinen, and Heidi Lyng, "Hypoxia-induced gene expression in chemoradioresistant cervical cancer revealed by dynamic contrast-enhanced MRI," *Cancer Res*, vol. 72, no. 20, pp. 5285–5295, Aug. 2012.

[49] Hafsa Habehh and Suril Gohel, "Machine learning in healthcare," *Curr Genomics*, vol. 22, no. 4, pp. 291–300, Dec. 2021.

[50] Ethem Alpaydin, *Introduction to machine learning*, MIT press, 2020.

[51] Mohamed Alloghani, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain, and Ahmed J. Aljaaf, *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*, pp. 3–21, Springer International Publishing, Cham, 2020.

[52] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[53] Tien T. Tang, Janice A. Zawaski, Kathleen N. Francis, Amina A. Qutub, and M. Waleed Gaber, "Image-based classification of tumor type and growth rate using machine learning: a preclinical study," *Scientific Reports*, vol. 9, no. 1, pp. 12529, Aug 2019.

[54] Davood Karimi, Jurriaan M. Peters, Abdelhakim Ouaalam, Sanjay P. Prabhu, Mustafa Sahin, Darcy A. Krueger, Alexander Kolevzon, Charis Eng, Simon K. Warfield, and Ali Gholipour, "Learning to detect brain lesions from noisy annotations," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1910–1914.

[55] Peter D Chang, Tony T Wong, and Michael J Rasiej, "Deep learning for detection of complete anterior cruciate ligament tear," *J Digit Imaging*, vol. 32, no. 6, pp. 980–986, Dec. 2019.

[56] Uran Ferizi, Harrison Besser, Pirro Hysi, Joseph Jacobs, Chamith S Rajapakse, Cheng Chen, Punam K Saha, Stephen Honig, and Gregory Chang, "Artificial intelligence applied to osteoporosis: A performance comparison of machine learning algorithms in predicting fragility fractures from MRI data," *J Magn Reson Imaging*, vol. 49, no. 4, pp. 1029–1038, Sept. 2018.

[57] Jianbo Shao, Zhuo Zhang, Huiying Liu, Ying Song, Zhihan Yan, Xue Wang, and Zujun Hou, "Dce-mri pharmacokinetic parameter maps for cervical carcinoma prediction," *Computers in Biology and Medicine*, vol. 118, pp. 103634, 2020.

[58] Turid Torheim, Eirik Malinen, Knut Håkon Hole, Kjersti Vassmo Lund, Ulf G. Indahl, Heidi Lyng, Knut Kvaal, and Cecilia M. Futsaether, "Autodelineation of cervical cancers using multiparametric magnetic resonance imaging and machine learning," *Acta Oncologica*, vol. 56, no. 6, pp. 806–812, 2017, PMID: 28464746.

[59] Xue Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, pp. 022022, feb 2019.

[60] Raschka Sebastian and Mirjalili Vahid, *Python Machine Learning - Second Edition : Unlock Modern Machine Learning and Deep Learning Techniques with Python by Using the Latest Cutting-edge Open Source Python Libraries.*, vol. 2nd ed, Packt Publishing, 2017.

[61] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou, "A comparison study of different kernel functions for svm-based classification of multi-temporal polarimetry sar data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-2/W3, pp. 281–285, 2014.

[62] Mohammad Hossin and Md Nasir Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, pp. 1, 2015.

[63] Jan Brabec and Lukas Machlica, "Bad practices in evaluation methodology relevant to class-imbalanced problems," 2018.

[64] Davide Chicco and Giuseppe Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 6, Jan 2020.

[65] Tord Hompland, Knut Håkon Hole, Harald Bull Ragnum, Eva-Katrine Aarnes, Ljiljana Vlatkovic, A Kathrine Lie, Sebastian Patzke, Bjørn Brennhovd, Therese Seierstad, and Heidi Lyng, "Combined mr imaging of oxygen consumption and supply reveals tumor hypoxia and aggressiveness in prostate cancer patients," *Cancer research*, vol. 78, no. 16, pp. 4774–4785, 2018.

[66] Tiril Hillestad, Tord Hompland, Christina S. Fjeldbo, Vilde E. Skingen, Unn Beate Salberg, Eva-Katrine Aarnes, Anja Nilsen, Kjersti V. Lund, Tina S. Evensen, Gunnar B. Kristensen, Trond Stokke, and Heidi Lyng, "MRI Distinguishes Tumor Hypoxia Levels of Different Prognostic and Biological Significance in Cervical Cancer," *Cancer Research*, vol. 80, no. 18, pp. 3993–4003, 09 2020.

[67] Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo, "The molecular signatures database (MSigDB) hallmark gene set collection," *Cell Syst*, vol. 1, no. 6, pp. 417–425, Dec. 2015.

[68] Kosuke Yoshihara, Maria Shahmoradgoli, Emmanuel Martínez, Rahulsimham Vegesna, Hoon Kim, Wandaliz Torres-Garcia, Victor Treviño, Hui Shen, Peter W. Laird, Douglas A. Levine, Scott L. Carter, Gad Getz, Katherine Stemke-Hale, Gordon B. Mills, and Roel G.W. Verhaak, "Inferring tumour purity and stromal and immune cell admixture from expression data," *Nature Communications*, vol. 4, no. 1, pp. 2612, Oct 2013.

[69] 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*, vol. 12, pp. 2825–2830, 2011.

[70] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56 – 61.

[71] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[72] Michael L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, pp. 3021, 2021.

[73] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[74] Abdulhamit Subasi, "Chapter 2 - data preprocessing," in *Practical Machine Learning for Data Analysis Using Python*, Abdulhamit Subasi, Ed., pp. 27–89. Academic Press, 2020.

[75] Rima Houari, Ahcène Bounceur, Abdelkamel Tari, and M Kecha, "Handling missing data problems with sampling methods," 06 2014, pp. 99–104.

[76] Olawale F Ayilara, Lisa Zhang, Tolulope T Sajobi, Richard Sawatzky, Eric Bohm, and Lisa M Lix, "Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry," *Health Qual Life Outcomes*, vol. 17, no. 1, pp. 106, June 2019.

[77] Kang Hyun, "The prevention and handling of the missing data," *kja*, vol. 64, no. 5, pp. 402–406, 2013.

[78] Hakan Demirtas, "Flexible imputation of missing data," *Journal of Statistical Software*, vol. 85, pp. 1–5, 2018.

[79] Yiran Dong and Chao-Ying Joanne Peng, "Principled missing data methods for researchers," *SpringerPlus*, vol. 2, no. 1, pp. 222, May 2013.

[80] Yen-Chi Chen, "Pattern graphs: a graphical approach to nonmonotone missing data," 2020.

[81] A. Rogier T. Donders, Geert J.M.G. van der Heijden, Theo Stijnen, and Karel G.M. Moons, "Review: A gentle introduction to imputation of missing values," *Journal of Clinical Epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[82] Noor Bariah Mohamad, An-Chow Lai, and Boon-Han Lim, "A case study in the tropical region to evaluate univariate imputation methods for solar irradiance data with different weather types," *Sustainable Energy Technologies and Assessments*, vol. 50, pp. 101764, 2022.

[83] John T. Hancock and Taghi M. Khoshgoftaar, "Survey on categorical data for neural networks," *Journal of Big Data*, vol. 7, no. 1, pp. 28, Apr 2020.

[84] Sikha Bagui, Debarghya Nandi, Subhash Bagui, and Robert Jamie White, "Machine learning and deep learning for phishing email classification using one-hot encoding," *Journal of Computer Science*, vol. 17, no. 7, pp. 610–623, 2021.

[85] Md Manjurul Ahsan, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, 2021.

[86] Calpephore Nkikabahizi, Wilson Cheruiyot, and Ann Kibe, "Chaining zscore and feature scaling methods to improve neural networks for classification," *Applied Soft Computing*, vol. 123, pp. 108908, 2022.

[87] Neelam Rout, Debahuti Mishra, and Manas Kumar Mallick, "Handling imbalanced data: A survey," in *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, M. Sreenivasa Reddy, K. Viswanath, and Shiva Prasad K.M., Eds., Singapore, 2018, pp. 431–443, Springer Singapore.

[88] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, aug 2019.

[89] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[90] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018.

[91] Josefa Díaz Álvarez, Jordi A Matias-Guiu, María Nieves Cabrera-Martín, José L Risco-Martín, and José L Ayala, "An application of machine learning with feature selection to improve diagnosis and classification of neurodegenerative disorders," *BMC Bioinformatics*, vol. 20, no. 1, pp. 491, Oct. 2019.

[92] Sarah Nogueira, Konstantinos Sechidis, and Gavin Brown, "On the stability of feature selection algorithms," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6345–6398, 2017.

[93] 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*, vol. 6, no. 63, pp. 3323, 2021.

[94] V. Roshan Joseph and Akhil Vakayil, "Split: An optimal method for data splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022.

[95] Zuzana Reitermanova et al., "Data splitting," in *WDS*. Matfyzpress Prague, 2010, vol. 10, pp. 31–36.

[96] Tzu-Tsung Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.

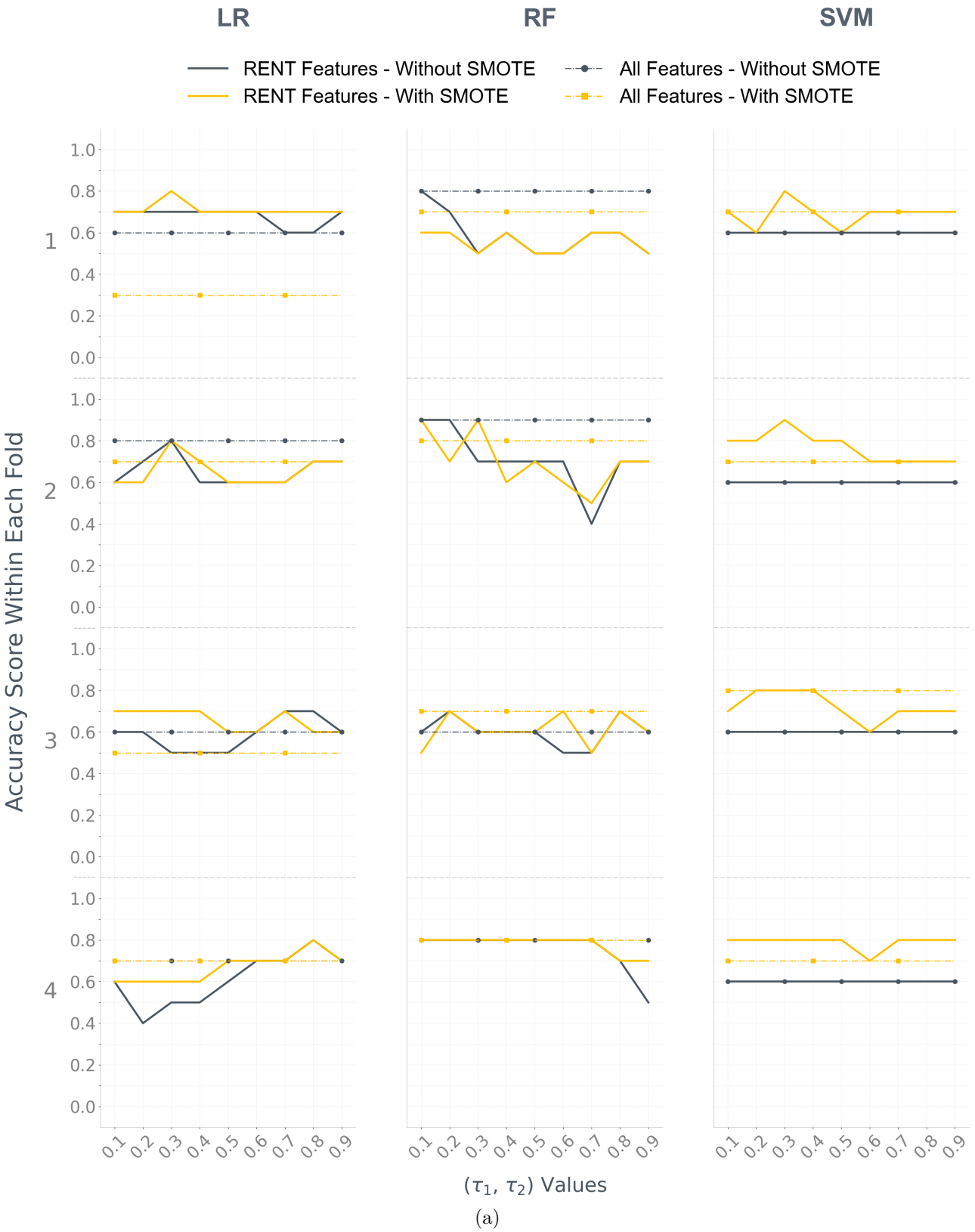
[97] Anna Jenul, Stefan Schrunner, Jürgen Pilz, and Oliver Tomic, "A user-guided bayesian framework for ensemble feature selection in life science applications (ubayfs)," *Machine Learning*, vol. 111, no. 10, pp. 3897–3923, Oct 2022.

# Appendix A

# Additional Results

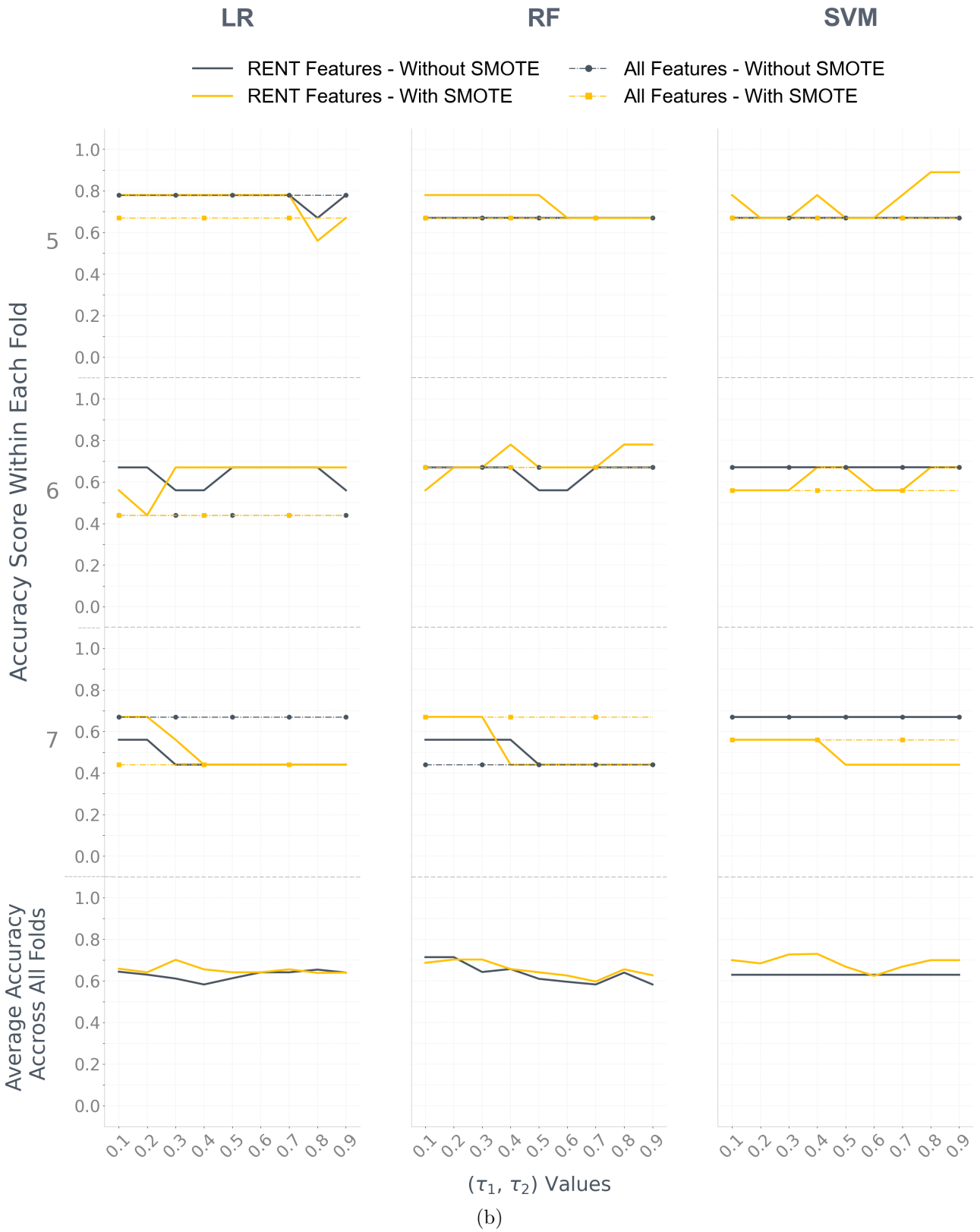
## A.1 RENT Hyperparameter Selection - Accuracy Scores

In addition to the details shared in Section 4.1, this appendix presents the accuracy scores achieved by logistic regression (LR), random forest (RF), and support vector machine (SVM) models in each fold, as well as the average scores across all folds. These results provide further evidence of the effectiveness of the values selected in Section 4.1 for the ($\tau_1$, $\tau_2$) pairs, as demonstrated by the average accuracy obtained.




(a)

Figure A.1: Classification Accuracy scores obtained from Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) models with different values for the ($\tau_1$, $\tau_2$) pair in the initial four data folds. Solid lines represent the scores of the models trained with the RENT features, while dashed lines with markers depict the scores of the models trained with all available features. The markers on the dashed lines are purely for visual convenience and do not indicate specific data points.




(b)

Figure A.1: Classification Accuracy scores obtained from Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) models with different values for the ($\tau_1$, $\tau_2$) pair in the last three data folds. The average scores across all folds are presented in the final row of the figure. Solid lines represent the scores of the models trained with the RENT features, while dashed lines with markers depict the scores of the models trained with all available features. The markers on the dashed lines are purely for visual convenience and do not indicate specific data points.

## A.2 Classification Performance - Other Metrics

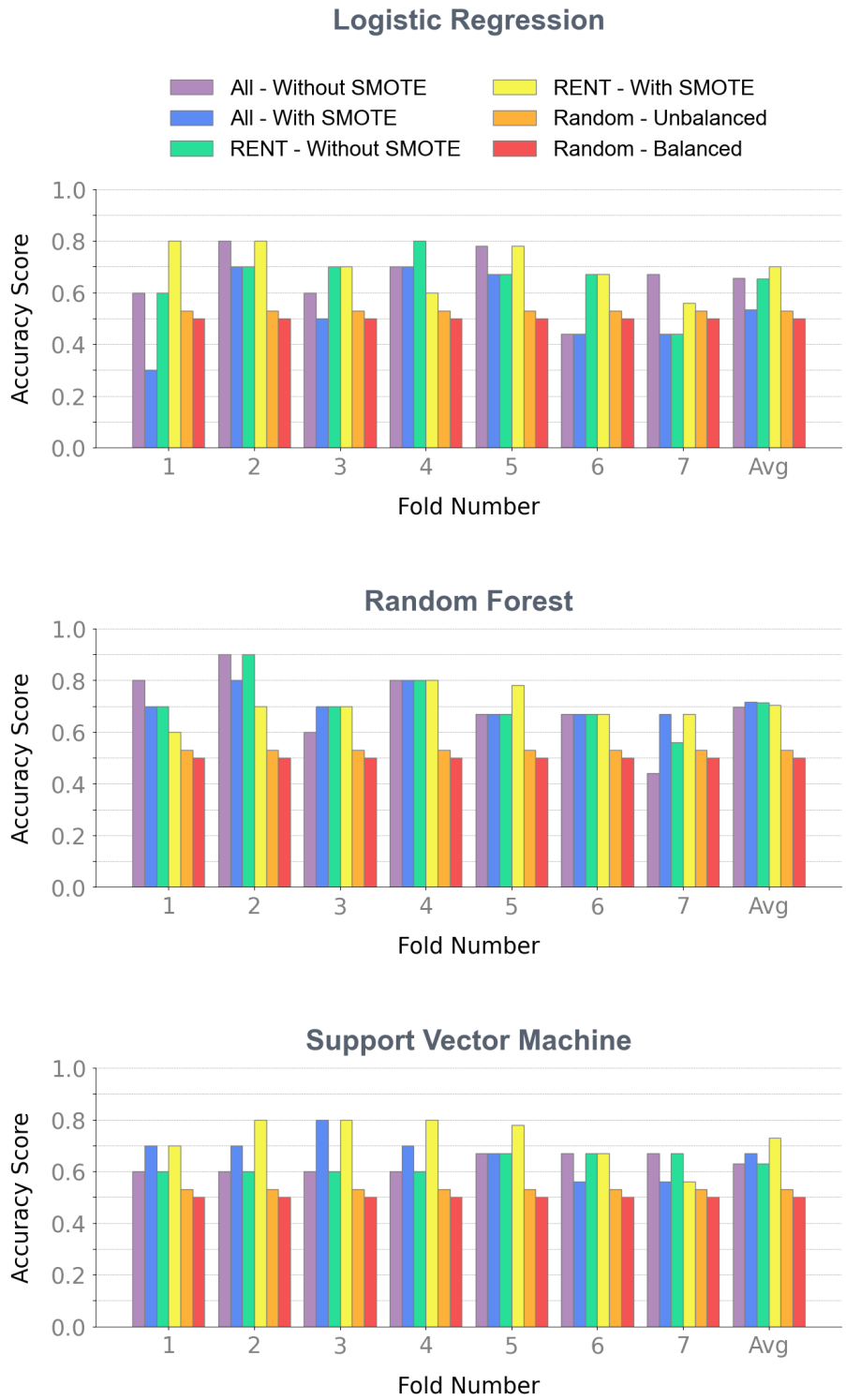



Figure A.2: The classification MCC scores achieved by machine learning models in seven different folds. The models were color-coded based on whether they were trained using all available features (referred to as "All") or the features selected by the RENT method (referred to as "RENT"), with or without the utilization of the SMOTE balancing technique. The last group of columns in each plot showcases the average scores across all folds. Furthermore, the accuracy score of the initial baseline (referred to as Random) is also presented, taking into account whether the samples are balanced or unbalanced. The $\tau_1$ and $\tau_2$ values utilized in RENT correspond to the same values chosen in Section 4.1.

Table A.1: Performance measures of logistic regression (LR), random forest (RF), and support vector machine (SVM) models before and after applying the SMOTE method using all available features: Precision (PRE), recall (REC), F1 score (F1 ) for each class (0=Cured, 1=Relapsed) within each fold, along with the average scores (Avg) across all folds.

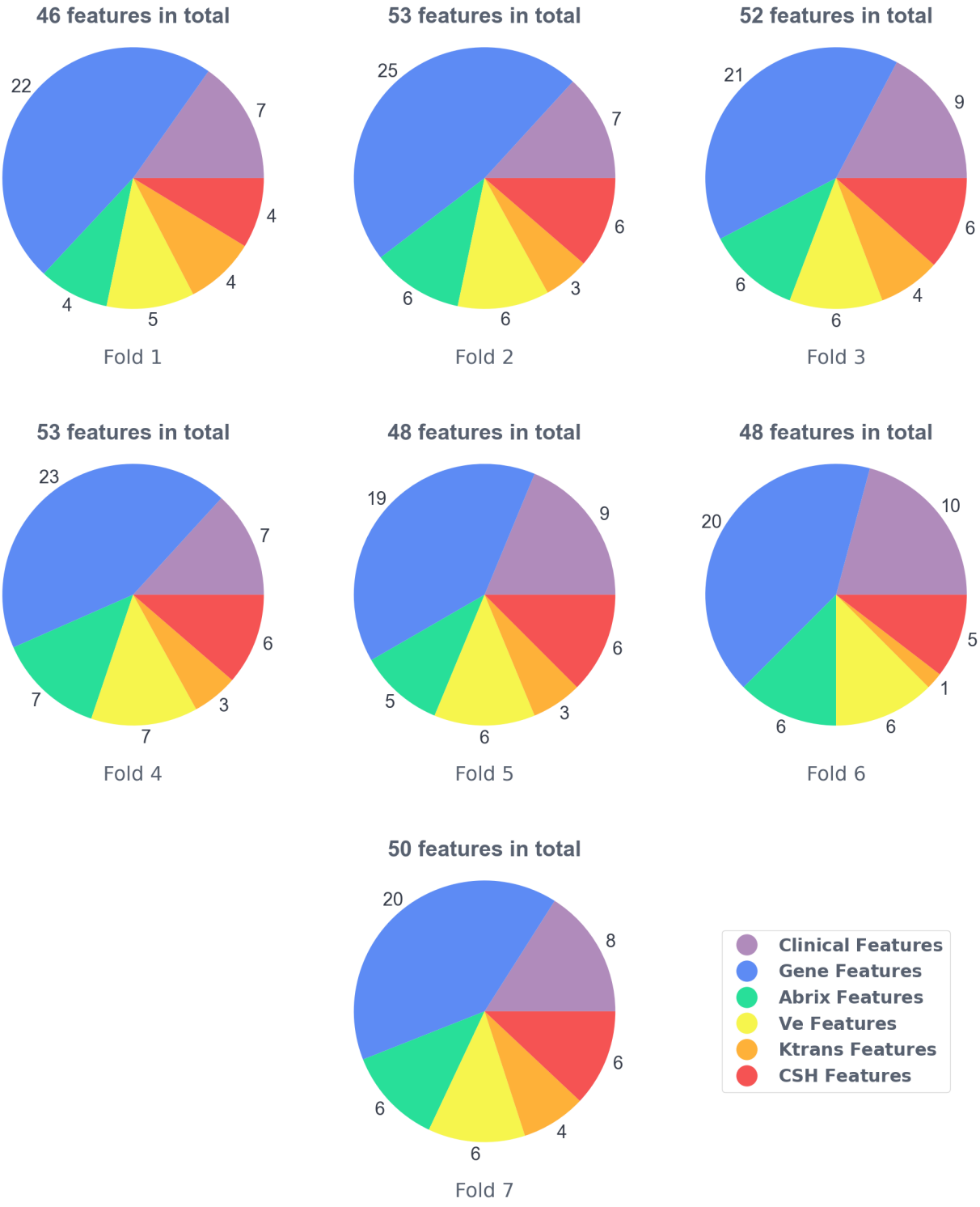
| | | **All Features - Without SMOTE** | | | | | | | | | **All Features - With SMOTE** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **LR** | | | **RF** | | | **SVM** | | | **LR** | | | **RF** | | | **SVM** | | |
| **Fold** | **Class** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** |
| 1 | 0 | 0.67 | 0.67 | 0.67 | 0.75 | 1 | 0.86 | 0.6 | 1 | 0.75 | 0.4 | 0.33 | 0.36 | 0.71 | 0.83 | 0.77 | 0.8 | 0.67 | 0.73 |
| | 1 | 0.5 | 0.5 | 0.5 | 1 | 0.5 | 0.67 | 0 | 0 | 0 | 0.2 | 0.25 | 0.22 | 0.67 | 0.5 | 0.57 | 0.6 | 0.75 | 0.67 |
| 2 | 0 | 1 | 0.67 | 0.8 | 1 | 0.83 | 0.91 | 0.6 | 1 | 0.75 | 0.8 | 0.67 | 0.73 | 1 | 0.67 | 0.8 | 1 | 0.5 | 0.67 |
| | 1 | 0.67 | 1 | 0.8 | 0.8 | 1 | 0.89 | 0 | 0 | 0 | 0.6 | 0.75 | 0.67 | 0.67 | 1 | 0.8 | 0.57 | 1 | 0.73 |
| 3 | 0 | 0.75 | 0.5 | 0.6 | 0.67 | 0.67 | 0.67 | 0.6 | 1 | 0.75 | 0.6 | 0.5 | 0.55 | 0.8 | 0.67 | 0.73 | 1 | 0.67 | 0.8 |
| | 1 | 0.5 | 0.75 | 0.6 | 0.5 | 0.5 | 0.5 | 0 | 0 | 0 | 0.4 | 0.5 | 0.44 | 0.6 | 0.75 | 0.67 | 0.67 | 1 | 0.8 |
| 4 | 0 | 0.71 | 0.83 | 0.77 | 0.75 | 1 | 0.86 | 0.6 | 1 | 0.75 | 0.8 | 0.67 | 0.73 | 0.75 | 1 | 0.86 | 0.71 | 0.83 | 0.77 |
| | 1 | 0.67 | 0.5 | 0.57 | 1 | 0.5 | 0.67 | 0 | 0 | 0 | 0.6 | 0.75 | 0.67 | 1 | 0.5 | 0.67 | 0.67 | 0.5 | 0.57 |
| 5 | 0 | 0.83 | 0.83 | 0.83 | 0.71 | 0.83 | 0.77 | 0.67 | 1 | 0.8 | 0.71 | 0.83 | 0.77 | 0.71 | 0.83 | 0.77 | 0.71 | 0.83 | 0.77 |
| | 1 | 0.67 | 0.67 | 0.67 | 0.5 | 0.33 | 0.4 | 0 | 0 | 0 | 0.5 | 0.33 | 0.4 | 0.5 | 0.33 | 0.4 | 0.5 | 0.33 | 0.4 |
| 6 | 0 | 0.67 | 0.33 | 0.44 | 0.71 | 0.83 | 0.77 | 0.67 | 1 | 0.8 | 0.67 | 0.33 | 0.44 | 0.8 | 0.67 | 0.73 | 0.67 | 0.67 | 0.67 |
| | 1 | 0.33 | 0.67 | 0.44 | 0.5 | 0.33 | 0.4 | 0 | 0 | 0 | 0.33 | 0.67 | 0.44 | 0.5 | 0.67 | 0.57 | 0.33 | 0.33 | 0.33 |
| 7 | 0 | 0.8 | 0.67 | 0.73 | 0.6 | 0.5 | 0.55 | 0.67 | 1 | 0.8 | 0.6 | 0.5 | 0.55 | 0.8 | 0.67 | 0.73 | 0.75 | 0.5 | 0.6 |
| | 1 | 0.5 | 0.67 | 0.57 | 0.25 | 0.33 | 0.29 | 0 | 0 | 0 | 0.25 | 0.33 | 0.29 | 0.5 | 0.67 | 0.57 | 0.4 | 0.67 | 0.5 |
| Avg | 0 | 0.77 | 0.64 | 0.69 | 0.74 | 0.8 | 0.77 | 0.63 | 1 | 0.77 | 0.65 | 0.54 | 0.59 | 0.79 | 0.76 | 0.77 | 0.8 | 0.66 | 0.71 |
| | 1 | 0.54 | 0.68 | 0.59 | 0.65 | 0.49 | 0.54 | 0 | 0 | 0 | 0.41 | 0.51 | 0.44 | 0.63 | 0.63 | 0.6 | 0.53 | 0.65 | 0.57 |

Table A.2: Performance measures of logistic regression (LR), random forest (RF), and support vector machine (SVM) models before and after applying the SMOTE method using the RENT-selected features: Precision (PRE), recall (REC), F1 score (F1 ) for each class (0=Cured, 1=Relapsed) within each fold, along with the average scores (Avg) across all folds. The specified values for $\tau_1$ and $\tau_2$ used in RENT for each model are enclosed in parentheses.

| | | **RENT Features - Without SMOTE** | | | | | | | | | **RENT Features - With SMOTE** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **LR ($\tau_1$,$\tau_2$ = 0.3)** | | | **RF ($\tau_1$,$\tau_2$ = 0.2)** | | | **SVM ($\tau_1$,$\tau_2$ = 0.4)** | | | **LR ($\tau_1$,$\tau_2$ = 0.3)** | | | **RF ($\tau_1$,$\tau_2$ = 0.2)** | | | **SVM ($\tau_1$,$\tau_2$ = 0.4)** | | |
| **Fold** | **Class** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** | **PRE** | **REC** | **F1** |
| 1 | 0 | 0.67 | 1 | 0.8 | 0.71 | 0.83 | 0.77 | 0.6 | 1 | 0.75 | 0.75 | 1 | 0.86 | 0.67 | 0.67 | 0.67 | 0.71 | 0.83 | 0.77 |
| | 1 | 1 | 0.25 | 0.4 | 0.67 | 0.5 | 0.57 | 0 | 0 | 0 | 1 | 0.5 | 0.67 | 0.5 | 0.5 | 0.5 | 0.67 | 0.5 | 0.57 |
| 2 | 0 | 0.75 | 1 | 0.86 | 1 | 0.83 | 0.91 | 0.6 | 1 | 0.75 | 0.83 | 0.83 | 0.83 | 0.8 | 0.67 | 0.73 | 0.83 | 0.83 | 0.83 |
| | 1 | 1 | 0.5 | 0.67 | 0.8 | 1 | 0.89 | 0 | 0 | 0 | 0.75 | 0.75 | 0.75 | 0.6 | 0.75 | 0.67 | 0.75 | 0.75 | 0.75 |
| 3 | 0 | 0.6 | 0.5 | 0.55 | 0.8 | 0.67 | 0.73 | 0.6 | 1 | 0.75 | 0.8 | 0.67 | 0.73 | 0.8 | 0.67 | 0.73 | 0.83 | 0.83 | 0.83 |
| | 1 | 0.4 | 0.5 | 0.44 | 0.6 | 0.75 | 0.67 | 0 | 0 | 0 | 0.6 | 0.75 | 0.67 | 0.6 | 0.75 | 0.67 | 0.75 | 0.75 | 0.75 |
| 4 | 0 | 0.67 | 0.33 | 0.44 | 0.75 | 1 | 0.86 | 0.6 | 1 | 0.75 | 0.67 | 0.67 | 0.67 | 0.75 | 1 | 0.86 | 0.75 | 1 | 0.86 |
| | 1 | 0.43 | 0.75 | 0.55 | 1 | 0.5 | 0.67 | 0 | 0 | 0 | 0.5 | 0.5 | 0.5 | 1 | 0.5 | 0.67 | 1 | 0.5 | 0.67 |
| 5 | 0 | 0.83 | 0.83 | 0.83 | 0.71 | 0.83 | 0.77 | 0.67 | 1 | 0.8 | 1 | 0.67 | 0.8 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 | 0.83 |
| | 1 | 0.67 | 0.67 | 0.67 | 0.5 | 0.33 | 0.4 | 0 | 0 | 0 | 0.6 | 1 | 0.75 | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 | 0.67 |
| 6 | 0 | 0.67 | 0.67 | 0.67 | 0.71 | 0.83 | 0.77 | 0.67 | 1 | 0.8 | 0.8 | 0.67 | 0.73 | 1 | 0.5 | 0.67 | 0.8 | 0.67 | 0.73 |
| | 1 | 0.33 | 0.33 | 0.33 | 0.5 | 0.33 | 0.4 | 0 | 0 | 0 | 0.5 | 0.67 | 0.57 | 0.5 | 1 | 0.67 | 0.5 | 0.67 | 0.57 |
| 7 | 0 | 0.6 | 0.5 | 0.55 | 0.67 | 0.67 | 0.67 | 0.67 | 1 | 0.8 | 0.75 | 0.5 | 0.6 | 0.8 | 0.67 | 0.73 | 0.75 | 0.5 | 0.6 |
| | 1 | 0.25 | 0.33 | 0.29 | 0.33 | 0.33 | 0.33 | 0 | 0 | 0 | 0.4 | 0.67 | 0.5 | 0.5 | 0.67 | 0.57 | 0.4 | 0.67 | 0.5 |
| Avg | 0 | 0.68 | 0.69 | 0.67 | 0.76 | 0.8 | 0.78 | 0.63 | 1 | 0.77 | 0.8 | 0.71 | 0.74 | 0.8 | 0.71 | 0.74 | 0.78 | 0.78 | 0.78 |
| | 1 | 0.58 | 0.47 | 0.48 | 0.63 | 0.53 | 0.56 | 0 | 0 | 0 | 0.62 | 0.69 | 0.63 | 0.62 | 0.69 | 0.63 | 0.68 | 0.64 | 0.64 |

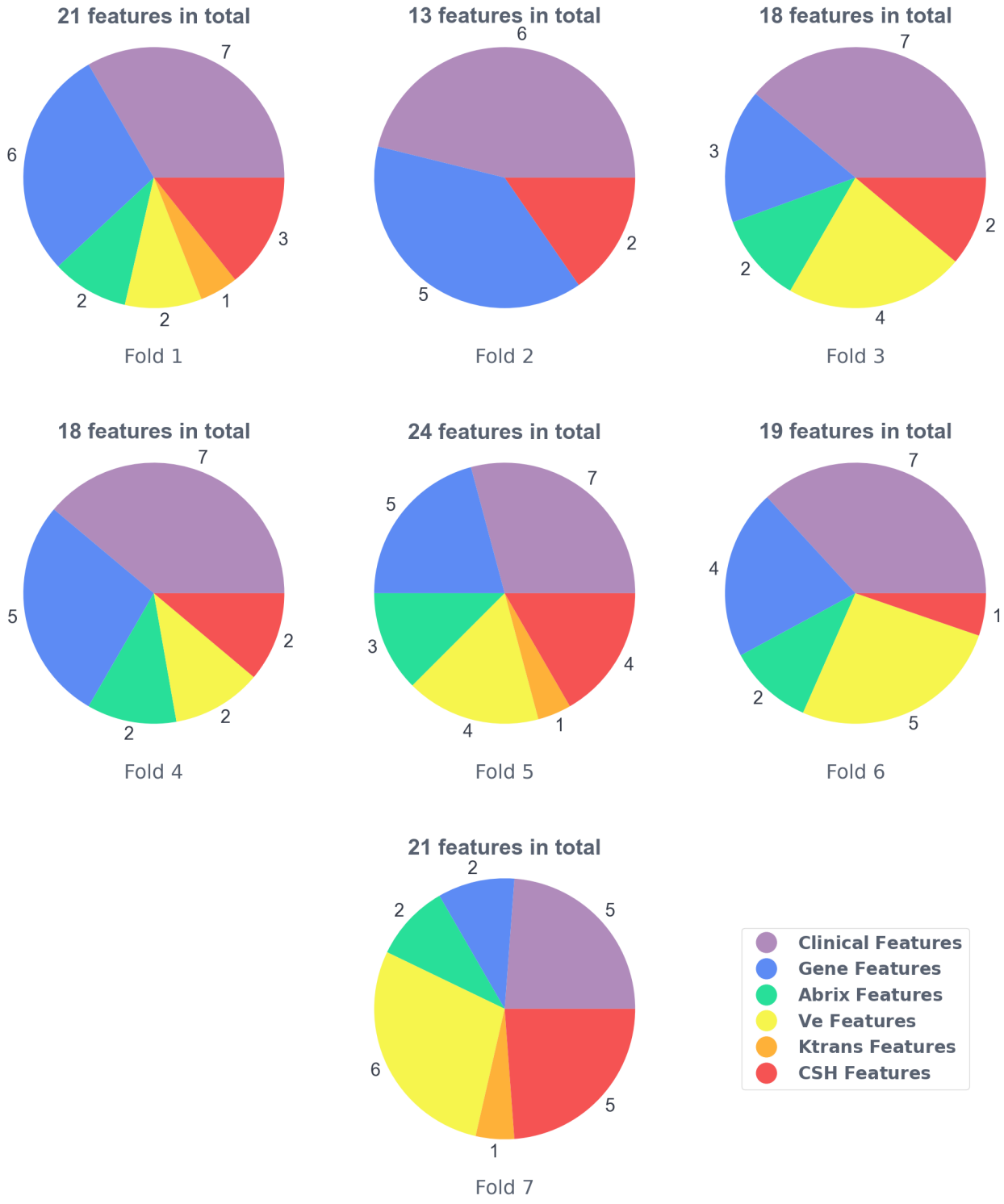
## A.3 Datasets' Shares in the RENT-Selected Features

This appendix offers additional details for Section 4.3, specifically regarding the distribution of selected features across available datasets for the following values of the ($\tau_1$, $\tau_2$) pair: 0.1, 0.5, 0.6, 0.7, and 0.9.



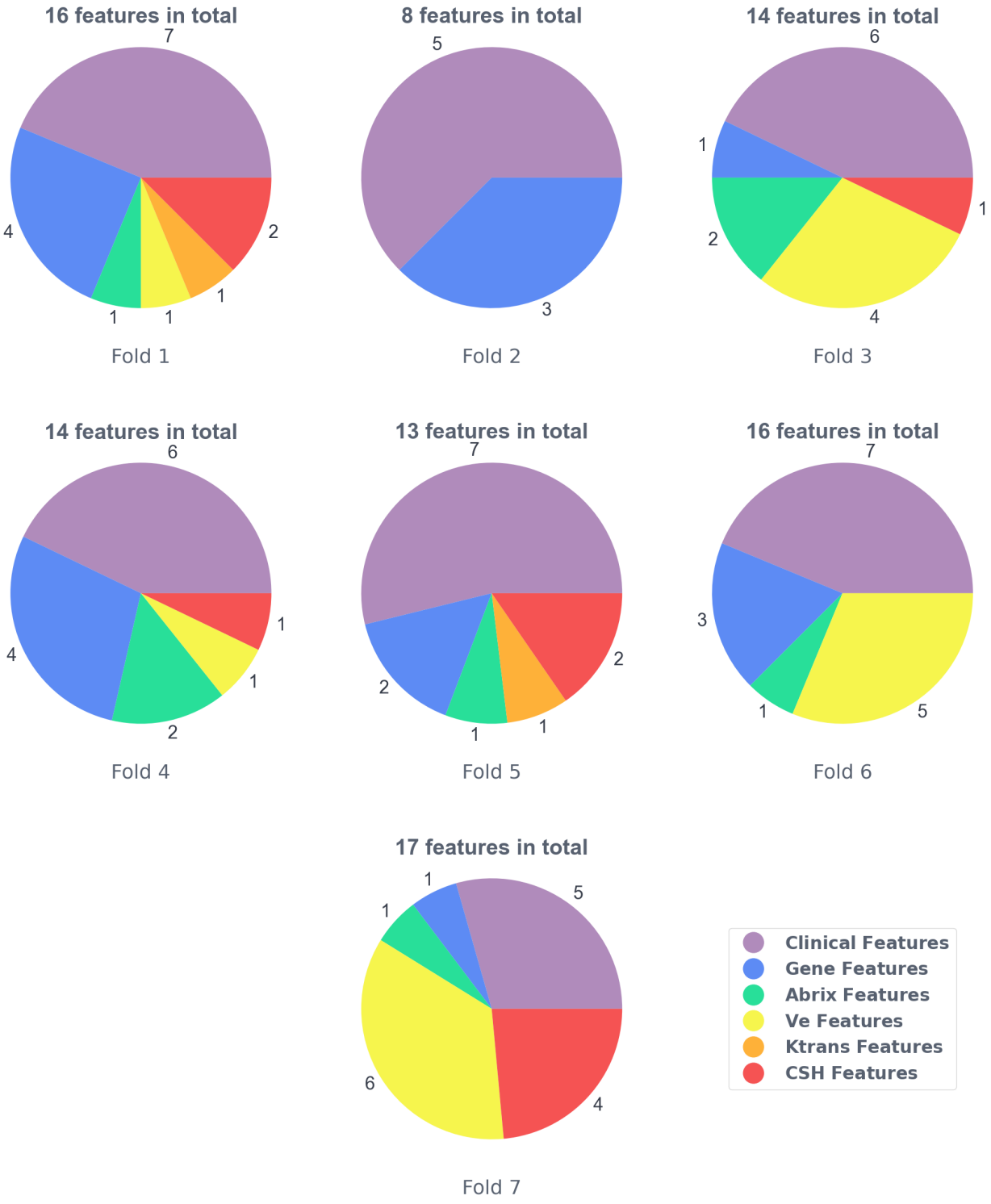

(a)

Figure A.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2$ = 0.1 across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.



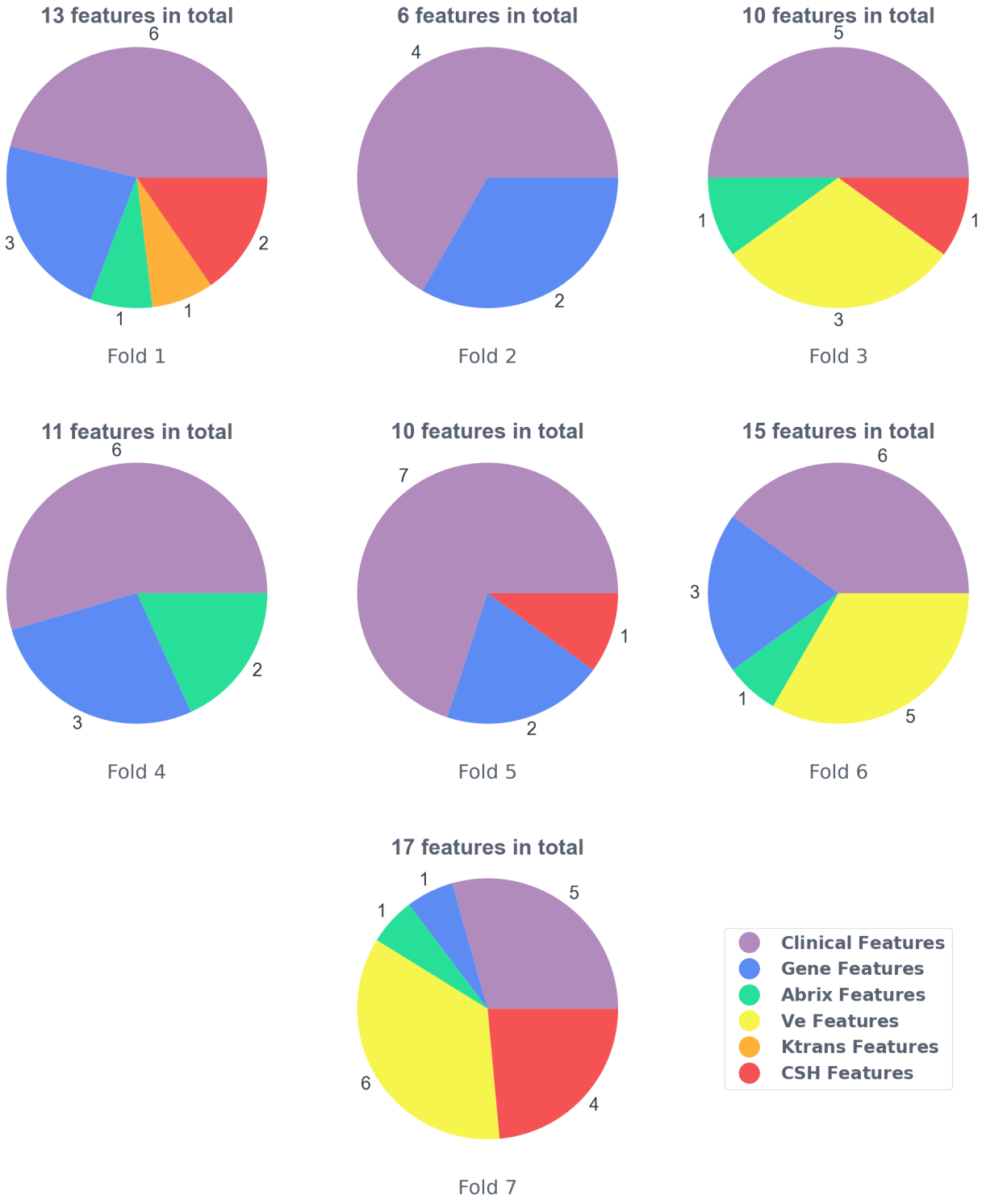

(b)

Figure A.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2$ = 0.5 across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.



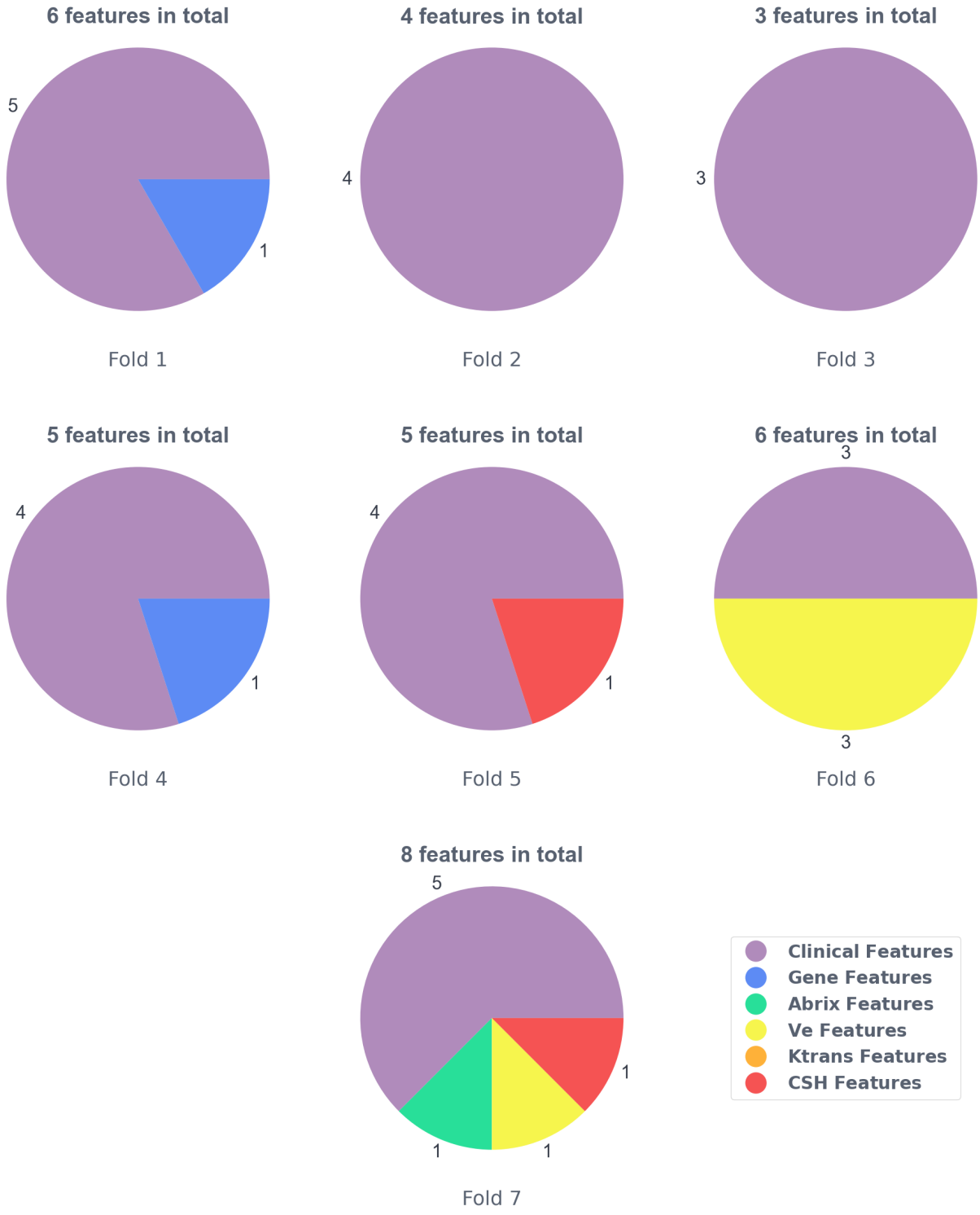

(c)

Figure A.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2$ = 0.6 across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.




(d)

Figure A.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2$ = 0.7 across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.




(e)

Figure A.3: Distribution of dataset shares within the RENT-selected features for $\tau_1$ and $\tau_2$ = 0.9p across all seven folds. The number displayed above each color represents the count of features from that dataset selected in the corresponding fold.

## A.4 Further Details Regarding the Most Informative Features

This appendix provides supplementary information to expand upon the content discussed in Section 4.3. Specifically, the feature names within each dataset and the frequency of their selection across different folds. The feature names and their respective selection counts are displayed for each dataset, considering various ($\tau_1$, $\tau_2$) values such as 0.1, 0.5, 0.6, 0.7, and 0.9.

Table A.3: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.1. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | Tumor volum mm3 | 7 | All Folds |
| | LN status | 7 | All Folds |
| | FIGO stage 2groups | 7 | All Folds |
| | n.voxels | 7 | All Folds |
| | FIGO stage 2B | 7 | All Folds |
| | FIGO stage 3B | 7 | All Folds |
| | FIGO stage 4A | 7 | All Folds |
| | FIGO stage 1B1 | 3 | 3, 5, 6 |
| | FIGO stage 3A | 2 | 6, 7 |
| | FIGO stage 2A | 2 | 3, 6 |
| | Dless MINUS Dmore | 1 | 5 |
| Abrix Features | ABrix interval 1 | 7 | All Folds |
| | ABrix interval 2 | 7 | All Folds |
| | ABrix interval 3 | 7 | All Folds |
| | ABrix interval 8 | 7 | All Folds |
| | ABrix interval 6 | 6 | 2,3,4,5,6,7 |
| | ABrix interval 7 | 4 | 2,3,4,6 |
| | ABrix interval 5 | 2 | 4,7 |
| Ve Features | Ve interval 6 | 7 | All Folds |
| | Ve interval 7 | 7 | All Folds |
| | Ve interval 8 | 7 | All Folds |
| | Ve interval 3 | 6 | 1,2,3,4,5,6 |
| | Ve interval 2 | 6 | 2,3,4,5,6,7 |
| | Ve interval 5 | 6 | 2,3,4,5,6,7 |
| | Ve interval 1 | 2 | 4,7 |
| | Ve interval 4 | 1 | 1 |
| Ktrans Features | Ktrans interval 1 | 6 | 1,2,3,4,5,6 |
| | Ktrans interval 6 | 6 | 1,2,3,4,5,7 |
| | Ktrans interval 5 | 5 | 1,3,4,5,7 |
| | Ktrans interval 7 | 2 | 2,7 |
| | Ktrans interval 3 | 2 | 3,7 |
| | Ktrans interval 4 | 1 | 1 |
| CSH Features | CSH interval 1 | 7 | All Folds |
| | CSH interval 2 | 7 | All Folds |
| | CSH interval 5 | 7 | All Folds |
| | CSH interval 6 | 7 | All Folds |
| | CSH interval 7 | 6 | 2,3,4,5,6,7 |
| | CSH interval 3 | 5 | 2,3,4,5,7 |

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Gene Features | Score HALLMARK APICAL SURFACE | 7 | All Folds |
| | Score HALLMARK MITOTIC SPINDLE | 7 | All Folds |
| | Score HALLMARK NOTCH SIGNALING | 7 | All Folds |
| | Score HALLMARK P53 PATHWAY | 7 | All Folds |
| | Score HALLMARK PANCREAS BETA CELLS | 7 | All Folds |
| | Score HALLMARK PI3K AKT MTOR SIGNALING | 7 | All Folds |
| | Dless MINUS Dmore | 6 | 1,2,3,4,6,7 |
| | ESTIMATE ImmuneScore | 6 | 1,2,4,5,6,7 |
| | Score HALLMARK ANDROGEN RESPONSE | 6 | 1,2,3,4,6,7 |
| | Score HALLMARK CHOLESTEROL HOMEOSTASIS | 6 | 1,2,3,4,5,7 |
| | Score HALLMARK WNT BETA CATENIN SIGNALING | 6 | 1,2,4,5,6,7 |
| | Score HALLMARK MYC TARGETS V2 | 6 | 2,3,4,5,6,7 |
| | Score HALLMARK INTERFERON ALPHA RESPONSE | 5 | 1,2,5,6,7 |
| | Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY | 5 | 1,2,3,5,6 |
| | ESTIMATEScore | 5 | 2,4,5,6,7 |
| | Score HALLMARK FATTY ACID METABOLISM | 4 | 1,3,4,5 |
| | Score HALLMARK TNFA SIGNALING VIA NFKB | 4 | 1,4,5,6 |
| | Score HALLMARK ALLOGRAFT REJECTION | 4 | 2,4,5,7 |
| | Score HALLMARK COMPLEMENT | 4 | 2,3,4,7 |
| | Score HALLMARK ADIPOGENESIS | 3 | 1,3,4 |
| | Score HALLMARK PEROXISOME | 3 | 1,3,5 |
| | ESTIMATE StromalScore | 3 | 2,4,7 |
| | Score HALLMARK COAGULATION | 3 | 2,3,7 |
| | Score HALLMARK G2M CHECKPOINT | 3 | 2,5,6 |
| | Score HALLMARK APOPTOSIS | 2 | 1,6 |
| | Score HALLMARK IL2 STAT5 SIGNALING | 2 | 1,3 |
| | Score HALLMARK PROTEIN SECRETION | 2 | 1,3 |
| | Score HALLMARK TGF BETA SIGNALING | 2 | 1,5 |
| | Score HALLMARK DNA REPAIR | 2 | 2,4 |
| | Score HALLMARK EPITHELIAL MESENCHYMAL TRANSITION | 2 | 2,7 |
| | Score HALLMARK HEME METABOLISM | 2 | 2,3 |
| | Score HALLMARK MYC TARGETS V1 | 2 | 2,6 |
| | Score HALLMARK BILE ACID METABOLISM | 2 | 3,4 |
| | Score HALLMARK XENOBIOTIC METABOLISM | 2 | 3,4 |
| | Score HALLMARK INFLAMMATORY RESPONSE | 1 | 1 |
| | Score HALLMARK IL6 JAK STAT3 SIGNALING | 1 | 2 |
| | Score HALLMARK HYPOXIA | 1 | 4 |
| | Score HALLMARK E2F TARGETS | 1 | 6 |
| | Score HALLMARK SPERMATOGENESIS | 1 | 6 |
| | Score HALLMARK MTORC1 SIGNALING | 1 | 7 |

Table A.4: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.5. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'LN status' | 7 | All Folds |
| | 'FIGO stage 2groups' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 6 | 1,3,4,5,6,7 |
| | 'Tumor volum mm3' | 6 | 1,2,3,4,5,6 |
| | 'n.voxels' | 6 | 1,2,3,4,5,6 |
| Gene Features | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 5 | 1,2,4,5,6 |
| | 'Score HALLMARK PANCREAS BETA CELLS' | 5 | 2,4,5,6,7 |
| | 'Score HALLMARK APICAL SURFACE' | 3 | 1,2,5 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 3 | 1,5,6 |
| | 'Score HALLMARK NOTCH SIGNALING' | 3 | 1,4,5 |
| | 'Score HALLMARK P53 PATHWAY' | 2 | 1,6 |
| | 'Score HALLMARK PI3K AKT MTOR SIGNALING' | 2 | 1,3 |
| | 'ESTIMATEScore' | 2 | 2,7 |
| | 'Score HALLMARK MYC TARGETS V2' | 1 | 2 |
| | 'Score HALLMARK HEME METABOLISM' | 1 | 3 |
| | 'Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY' | 1 | 3 |
| | 'Dless MINUS Dmore' | 1 | 4 |
| | 'ESTIMATE ImmuneScore' | 1 | 4 |
| Abrix Features | 'ABrix interval 8' | 5 | 1,3,4,5,6 |
| | 'ABrix interval 3' | 4 | 3,4,5,6 |
| | 'ABrix interval 1' | 3 | 1,5,7 |
| | 'ABrix interval 2' | 1 | 7 |
| Ve Features | 'Ve interval 8' | 6 | 1,3,4,5,6,7 |
| | 'Ve interval 3' | 5 | 1,3,4,5,6 |
| | 'Ve interval 6' | 4 | 3,5,6,7 |
| | 'Ve interval 7' | 3 | 3,6,7 |
| | 'Ve interval 5' | 3 | 5,6,7 |
| | 'Ve interval 1' | 1 | 7 |
| | 'Ve interval 2' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 2 | 1,5 |
| | 'Ktrans interval 3' | 1 | 7 |
| CSH Features | 'CSH interval 6' | 6 | 1,2,3,4,5,7 |
| | 'CSH interval 1' | 4 | 1,4,5,6 |
| | 'CSH interval 5' | 3 | 1,5,7 |
| | 'CSH interval 2' | 3 | 2,5,7 |
| | 'CSH interval 7' | 2 | 3,7 |
| | 'CSH interval 3' | 1 | 7 |
| | 'CSH interval 3' | 1 | 7 |

Table A.5: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.6. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'FIGO stage 2groups' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 4A' | 6 | 1,3,4,5,6,7 |
| | 'n.voxels' | 6 | 1,2,3,4,5,6 |
| | 'LN status' | 5 | 1,2,5,6,7 |
| | 'Tumor volum mm3' | 5 | 1,3,4,5,6 |
| Gene Features | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 4 | 1,2,4,6 |
| | 'Score HALLMARK PANCREAS BETA CELLS' | 4 | 2,4,5,7 |
| | 'Score HALLMARK APICAL SURFACE' | 3 | 1,2,5 |
| | 'Score HALLMARK NOTCH SIGNALING' | 2 | 1,4 |
| | 'Score HALLMARK P53 PATHWAY' | 2 | 1,6 |
| | 'Score HALLMARK REACTIVE OXYGEN SPECIES PATHWAY' | 1 | 3 |
| | 'Dless MINUS Dmore' | 1 | 4 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 1 | 6 |
| Abrix Features | 'ABrix interval 8' | 4 | 1,3,4,5 |
| | 'ABrix interval 3' | 3 | 3,4,6 |
| | 'ABrix interval 2' | 1 | 7 |
| Ve Features | 'Ve interval 8' | 5 | 1,3,4,6,7 |
| | 'Ve interval 6' | 3 | 3,6,7 |
| | 'Ve interval 7' | 3 | 3,6,7 |
| | 'Ve interval 3' | 2 | 3,6 |
| | 'Ve interval 5' | 2 | 6,7 |
| | 'Ve interval 1' | 1 | 7 |
| | 'Ve interval 2' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 2 | 1,5 |
| CSH Features | 'CSH interval 6' | 5 | 1,3,4,5,7 |
| | 'CSH interval 1' | 2 | 1,5 |
| | 'CSH interval 2' | 1 | 7 |
| | 'CSH interval 3' | 1 | 7 |
| | 'CSH interval 5' | 1 | 7 |

Table A.6: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.7. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| Block | Feature Name | Count In Folds | Fold Details |
|---|---|---|---|
| Clinical Features | 'FIGO stage 2groups' | 7 | All Folds |
| | 'FIGO stage 2B' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'n.voxels' | 6 | 1,2,3,4,5,6 |
| | 'FIGO stage 4A' | 5 | 1,4,5,6,7 |
| | 'Tumor volum mm3' | 4 | 3,4,5,6 |
| | 'LN status' | 3 | 1,5,7 |
| Gene Features | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 4 | 1,2,4,6 |
| | 'Score HALLMARK PANCREAS BETA CELLS' | 4 | 2,4,5,7 |
| | 'Score HALLMARK APICAL SURFACE' | 2 | 1,5 |
| | 'Score HALLMARK NOTCH SIGNALING' | 2 | 1,4 |
| | 'Score HALLMARK MITOTIC SPINDLE' | 1 | 6 |
| | 'Score HALLMARK P53 PATHWAY' | 1 | 6 |
| Abrix Features | 'ABrix interval 8' | 3 | 1,3,4 |
| | 'ABrix interval 3' | 2 | 4,6 |
| | 'ABrix interval 2' | 1 | 7 |
| Ve Features | 'Ve interval 7' | 3 | 3,6,7 |
| | 'Ve interval 8' | 3 | 3,6,7 |
| | 'Ve interval 3' | 2 | 3,6 |
| | 'Ve interval 5' | 2 | 6,7 |
| | 'Ve interval 6' | 2 | 6,7 |
| | 'Ve interval 1' | 1 | 7 |
| | 'Ve interval 2' | 1 | 7 |
| Ktrans Features | 'Ktrans interval 1' | 1 | 1 |
| CSH Features | 'CSH interval 6' | 4 | 1,3,4,7 |
| | 'CSH interval 1' | 1 | 1 |
| | 'CSH interval 2' | 1 | 7 |
| | 'CSH interval 3' | 1 | 7 |
| | 'CSH interval 5' | 1 | 7 |

Table A.7: The selected features, among all available features, by the RENT feature selection technique for $\tau_1$ and $\tau_2$ equal 0.9. The table also provides information on how frequently each feature was selected across seven experiment folds and the specific folds each feature was chosen, indicated in the Count In Folds and Fold Details columns, respectively.

| **Block** | **Feature Name** | **Count In Folds** | **Fold Details** |
|---|---|---|---|
| Clinical Features | 'FIGO stage 2groups' | 7 | All Folds |
| | 'FIGO stage 3B' | 7 | All Folds |
| | 'FIGO stage 2B' | 6 | 1,2,3,4,5,7 |
| | 'n.voxels' | 4 | 2,4,5,6 |
| | 'LN status' | 2 | 1,7 |
| | 'FIGO stage 4A' | 2 | 1,7 |
| Gene Features | 'Score HALLMARK NOTCH SIGNALING' | 1 | 1 |
| | 'Score HALLMARK WNT BETA CATENIN SIGNALING' | 1 | 4 |
| Abrix Features | 'ABrix interval 2' | 1 | 7 |
| Ve Features | 'Ve interval 3' | 1 | 6 |
| | 'Ve interval 5' | 1 | 6 |
| | 'Ve interval 6' | 1 | 6 |
| | 'Ve interval 2' | 1 | 7 |
| CSH Features | 'CSH interval 6' | 2 | 5,7 |

Thank You.

**Norges miljø- og biovitenskapelige universitet**
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway