Norwegian University
of Life Sciences

Master's Thesis 2021 30 ECTS
Faculty of Science and Technology

Sequential and Orthogonalized Partial Least Squares Regression applied to healthcare data acquired from patients diagnosed with gastrointestinal carcinoma

Hemanth Babu Sana
Master of Science in Data Science

Preface

This masters thesis is written at the Faculty of Science and Technology at the Norwegian University of Life Sciences (NMBU) in 2021. The thesis is for a total of 30 ECTS credits and concludes the two-year masters degree in Data Science. The research for the thesis has been carried out with the Functional and Molecular Imaging research group at the Oslo University Hospital.

First of all, I would like to thank my supervisors, Associate Professors Kristian Hovde Liland and Oliver Tomic, Faculty of Science and Technology (REALTEK), NMBU for their excellent guidance and constant support throughout this process. Writing the thesis was a challenge in the middle of a pandemic. Fortunately, both Kristian and Oliver were always available when needed and willing to support in all situations.

Further, I would like to thank Henning Langen Stokmo and Mona-Elisabeth Rootwelt-Revheim, Functional and Molecular Imaging Research Group for taking time from their busy schedule to provide us feedback at different stages of the thesis.

Finally, I would like to thank my family, friends and fiance for their love, encouragement and support throughout my Master's program.

Ås, 31st May, 2021

Hemanth Babu Sana

Abstract

Gastrointestinal carcinoma are the cancers that affect the gastrointestinal tract and other organs that include esophagus, pancreas, stomach, colon, rectum, anus, liver and intestine. Gastrointestinal cancers account to 26% of global cancer incidence. They account to 35% of all cancer-related deaths. Being able to find the factors responsible for increasing the life span of patients adds significant value in the course of treatment for the doctors.

This Master's thesis explored the feasibility of employing two new state-of-art techniques- Repeated Elastic Net Technique (RENT) for feature selection and Sequential and Orthogonalized Partial Least Squares regression (SO-PLS). This study helped to (1) find features that are important for predicting the target using RENT and to (2) use the underlying dimensionality of the data blocks to explain the variance of the target using SO-PLS.

The feature selection using RENT proved to be useful by reducing the number of features from 57 to 7 in the first block and from 27 to 7 in the second block. By using these selected features from both blocks, SO-PLS regression achieved a cumulative calibrated explained variance of 76.4%. The score and loading plots from SO-PLS helped in identifying the features that explain the distribution of values in the target block.

These results indicate that RENT and SO-PLS have the potential in developing as useful techniques for clinicians in understanding the factors responsible for the longevity of patient life.

Contents

1 Introduction **1**
1.1 Background 1
1.2 Problem Statement 3
1.3 Structure of the thesis 4

2 Theory **5**
2.1 Repeated Elastic Net Technique for Feature Selection 5
2.1.1 Regularization 6
2.1.2 Selection Criteria 8
2.2 Principal component analysis 9
2.2.1 Understanding PCA 9
2.2.2 Main steps in PCA 10
2.3 Partial least squares regression 12
2.4 Sequential and Orthogonalized Partial Least Squares Regression . 13
2.4.1 SO-PLS Model 13
2.4.2 Choosing optimal number of components 14
2.5 Validation 15
2.5.1 Cross-validation 15
2.5.2 K-fold cross validation 16
2.5.3 Stratified K-fold cross validation 16
2.5.4 Repeated stratified k-fold cross validation 17

3 Materials **19**
3.1 First data block features 19
3.2 Second data block features 21
3.3 Response variable block 21

4 Analysis workflow **23**
4.1 Software . . . 23
4.2 Data Exploration and Preparation . . . 24
4.2.1 Data pre-processing . . . 24
4.2.2 Data visualisation . . . 27
4.2.3 Checking for Deviating observations using PCA . . . 30
4.2.4 Determining potential target variables . . . 31
4.2.5 Including features having missing values . . . 31
4.3 Feature Selection using the RENT workflow . . . 32
4.3.1 Using repeated stratified k-fold cross-validation with RENT 32
4.3.2 Applying RENT regression . . . 34
4.4 SO-PLS regression . . . 37
4.4.1 Selecting number of components in each block . . . 37
4.4.2 Final SO-PLS model . . . 38

5 Results **39**
5.1 Data Exploration and Preparation . . . 39
5.1.1 PCA on the first block . . . 39
5.1.2 PCA on the second block . . . 40
5.1.3 Preparing the target block . . . 41
5.2 Feature selection results . . . 43
5.2.1 Running RENT without features having missing values . . 43
5.2.2 RENT after including RESPONS1 and TIMETOTRM1 . . 45
5.2.3 RENT after removing samples having high average absolute error samples . . . 46
5.2.4 Selected features for SO-PLS modelling . . . 51
5.3 SO-PLS regression . . . 52
5.3.1 Cross validation . . . 52
5.3.2 SO-PLS model . . . 54

6 Discussion **57**
6.1 Dataset . . . 57
6.1.1 Features having irrelevant values . . . 57
6.2 Patients with high average absolute error . . . 57
6.3 Implications for SO-PLS model . . . 58
6.3.1 Explained variance . . . 58
6.3.2 Overfitting . . . 59
6.3.3 Correlation between the features . . . 59
6.4 Possible future work . . . 60

7 Conclusion **61**

Bibliography **i**

List of Tables

4.1 Software versions 23
4.2 Missing values for each data block 25

5.1 Features selected by RENT for each cross-validation split without including features having missing values and the performances for first block 43
5.2 Features selected by RENT for each cross-validation split without including features having missing values and the performances for second block 44
5.3 Features selected by RENT and performances for each cross-validation split with RESPONS1 and TIMETOTRM1 included for first block 45
5.4 Features selected by RENT and performances for each cross-validation split with RESPONS1 and TIMETOTRM included for second block 46
5.5 Average absolute value of 20 patients in a split 47
5.6 Average absolute value of 20 patients in another split 48
5.7 Features selected by RENT and performances for each cross-validation split after removing the samples with high absolute error for the first block 49
5.8 Features selected by RENT and performances for each cross-validation split after removing the samples with high absolute error for the second block 50
5.9 Number of times features are selected out of 8 splits for the first block 51
5.10 Number of times features are selected out of 8 splits for the second block 51

6.1 Table showing both calibrated and validated explained variance . . 59

List of Figures

1.1 Global cancer statistics 1

2.1 Schematic RENT 6
2.2 Sequential functionality of PLS and orthogonalization 14
2.3 Step wise SO-PLS 15
2.4 Cross-validation 16
2.5 K-fold cross validation 17

4.1 Figure showing skewness in data 28
4.2 Scatter plot example 30
4.3 Score and loading plot example 31
4.4 RENT workflow 33

5.1 Score plot for the two components in block 1 40
5.2 PCA explained variance for block1 40
5.3 Score plot for the two components in block 2 41
5.4 PCA explained variance for block2 41
5.5 Histogram of target 42
5.6 Histogram of target after box-cox 42
5.7 SO-PLS cross validation for first data block 52
5.8 SO-PLS cross validation for two blocks considering block1 first . 53
5.9 So-PLS cross validation for one block with second data block . . . 53
5.10 SO-PLS cross validation for two blocks considering block2 first . 54
5.11 SO-PLS PCP score plot 55
5.12 SO-PLS X loading plot 55
5.13 SO-PLS explained variance plot 56

6.1 BMI plot 58
6.2 SO-PLS correlation loadings plot 59

Chapter 1

Introduction

1.1 Background

According to World Health Organisation, cancer is the second leading cause of death in human beings globally. As of 2018, an estimate of 9.6 million people lost their lives due to cancer [1]. One in every six deaths in the world is caused by cancer. According to a research [2], it is estimated that 13·2 million cancer related annual deaths by 2030.

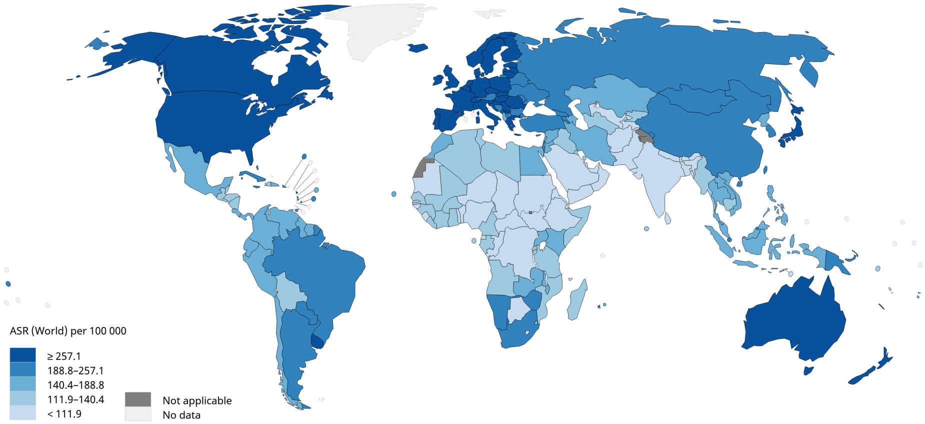

Figure 1.1: *Global estimated age-standardized incident rates of cancer in 2020 [3].*

Cancers are cells that grow uncontrollably which eventually form a mass of tissue. These cancerous cells interfere with the regular functionality of the tissues and organs in the human body [4]. In some cases, cancer spreads from one part of the body to other surrounding areas. There are different stages in cancer varying from stage-0 to stage-4 depending on the severity [5]. There are around 100 types of cancer depending on which organ or tissue the cancer cells form [6].

Various factors play vital roles in developing cancer in individuals like lifestyle, genetics, environment, exposure to radiation, ageing, etc [7]. In general, it is not possible to find out which factors are responsible for cancer in individuals. It is believed that interaction of many such factors together can result in changes in the cells which may lead to cancer.

Diagnosis of cancer in general, is done using different ways like physical exam, laboratory tests, imaging tests, biopsy [8] etc. In laboratory tests, urine and blood tests are performed. Imaging tests such as computed tomography scan (CT), magnetic resonance imaging (MRI), positron emission tomography scan (PET) allows to examine bones and internal organs. During biopsy, a sample of cells are collected to test in lab which is widely considered as a definitive way for cancer diagnosis.

There are different ways to treat cancer such as surgery, radiation therapy, chemotherapy, immunotherapy, hormonal therapy, stem cell transplant etc. [9]. The type of the treatment depends on the condition of patient, location, stage and grade of the cancer.

In the recent decades, many researchers around the globe have been doing impressive work on cancer research [10] [11]. With the growing cases every year it has become need of the hour for early detection and prognosis of cancer as it helps the doctors determining the course of the treatment. With large amounts of healthcare data available for the research, a lot of work is underway. Data analysis techniques such as convolutional neural networks(CNN) for segmentation of tumours, Multi-block regression techniques for prognosis etc. can be used for the right treatment determination.

Healthcare data is available in different forms such as images, tables etc. In our project we focus on data in table(traditional data container) and methods that can analyse such data. The data we work on is a multi-variate data and may come from different sources and can have different complexity. This kind of data needs methods that can handle such complexities.

Many multi block data analysis techniques have made their way in the research [12] based on multi block practice. In addition to principal component analysis [13], partial least squares regression [14], a number of techniques which are built on these have surfaced[15]. With these techniques, the inter dependency of the blocks and their contribution to the target blocks are studied. Sequential and orthogonalised partial least square regression (SO-PLS) is one such multivariate technique used to know the relation between predictor and target variables.

Feature selection techniques mainly focus on finding a subset of features which reduce the unnecessary noise or features while retaining the systematic information explained by the data[16]. The selected subset of features helps in easier interpretation with fewer features and prevents overfitting of the models. In addition to good prediction, these techniques are also used for fast and cost effective prediction [17]. By finding features that are important we can collect only needed data thereby reducing the cost incurred for collecting the data.

In multi block and multi variate data analysis, each block of data can represent one type of measurement or data from one instrument. Feature selection techniques can be used to find the subset of features in each block that explains most of the information contained in the block. It helps in reducing the size of the blocks considerably while retaining most of the information. These blocks when fed to the multi block models reduces the computational costs and time considerably.

1.2 Problem Statement

In cancer treatment, knowing which variables guide the treatment process helps clinicians to plan the course of the treatment. It is believed feature selection not only helps in treatment progress but also helps in prognosis and early detection of the disease. This helps clinicians in identifying the severity of patients' condition in addition to conventional medical techniques.

Feature selection helps in improving the predictive performance of the model and helps in getting more interpretable model. It also helps in knowing which data to collect to acquire relevant knowledge for future analyses.

In medicine, understanding the blocks of data which contains features collected using similar procedures helps in knowing variables to look for finding bio markers for treatment progress. Deriving components in the data blocks to see underlying patterns in data is useful to understand and predict new findings.

With the availability of healthcare data to the scientific research community, it has become need of the hour to use this data to train models using machine learning and data science techniques to find critical patterns which help in studying how the variables effect the progress of the treatment. A number of techniques have surfaced for multi block and multivariate data which is usually seen in the field of medicine.

In this thesis we propose two state of the art techniques - feature selection using repeated elastic net technique (RENT), sequential and orthogonalised partial least squares regression (SO-PLS) for cancer data.

SO-PLS is being used for the first time on these type of data. It belongs to a class of component based methods. With its interpretation tools and visualisation of the data and understanding of the data it would be interesting to see how it works on this data .The proposed techniques helps in identifying the variables which contribute to the target variable.

1.3 Structure of the thesis

The thesis starts with explaining the methods involved such as RENT, SO-PLS, PCA in chapter 2. In chapter 3, the details about the data are provided. The workflow of the thesis is discussed in chapter 4. Chapter 5 covers the results obtained which are then discussed in 6. The summary of the thesis is given in chapter 7.

Chapter 2

Theory

In this chapter we describe different methods used in this master thesis.

2.1 Repeated Elastic Net Technique for Feature Selection

In modern times, a lot of data is produced in a quick pace, not only does the samples but also number of features increases. This makes the computation and training of predictive models a difficult and time consuming process. In such data it is a common sight to face issues such as overfitting, correlation which makes training a model troublesome. These issues guided the necessity to select the features that are important for prediction [18] and not loose too much predictive performance.

Feature selection is reducing the number features when working with a predictive model. Reducing number of features helps in reducing the computational time and keeping the performance at the same mark or increasing it [19]. A variety of feature selection methods have been published and studied by many researchers [20] [21]. Most of the techniques concentrate on optimization of the selected feature subsets considering the performance of prediction.

Repeated elastic net technique for feature selection (RENT) is a feature selection technique that implements not just considering the frequency with which each feature is selected but also feature weight distribution of models using elastic net regularisation. RENT is built on central idea of ensemble models studied by Meinshausen and Bühlmann [22].

RENT trains a number of linear regression models on subsets of the training data. It uses elastic net for regularisation

RENT used three main criteria in selecting the features that use feature weight distribution across all the models:

1. Frequency with which a feature is selected
2. The degree of alternation for feature weights between positive and negative values
3. If feature weights are unequal to zero significantly

By using specific thresholds for each of the above criteria we can guide how aggressively RENT should reduce the number of select features. We can change the number of features selected by changing the threshold in each criterion. This kind of ensemble learning makes RENT a suitable feature selection technique in datasets where number of features are more than number of samples [19].

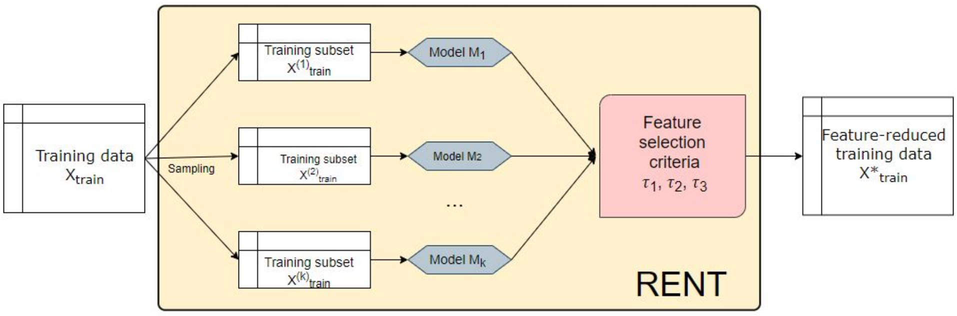

Figure 2.1: *Pipeline depicting feature selection using RENT*

Figure 2.1 is a schematic which depicts the pipeline in RENT. We sample the training data set X_{train} into k independent and identically distributed (i.i.d) subsets denoted as $X^{(k)}_{train} \subset X_{train}$. Now model evaluation is done on each of these subsets of the validation sets $X^{(k)}_{val} = X_{train} \setminus X^{(k)}_{train}$ where $\setminus$ is the set difference operator.

2.1.1 Regularization

A model is said to overfit if it learns very accurate patterns and noise from training data but fails to predict well on a new data. Regularization is a technique that helps in overfitting. It is done by reducing the generalisation error by appropriately fitting the function on a selected training set. This helps in preventing overfitting. Regularization penalizes those weights which are large by adding regularisation part to the loss function.There are two main regularization techniques:

1. Ridge regression also referred to as L2
2. Lasso regression also referred to as L1

Consider a simple linear regression which looks like this

$$Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m \tag{2.1}$$

In the above equation Y represents the relation learned and β's are the coefficients for the predictor variables X_1, X_2 etc.

The residual sum of squares(RSS) as loss function for the regression is calculated by using the below equation

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{2.2}$$

where n is number of predictor variables, β_j is the coefficient of j^{th} predictor variable, y_i is the i^{th} response variable.

It should be taken care to reduce the RSS. This is where regularization comes into account.

Ridge Regression

Ridge regression is also called L2 regularization as it uses L2 norm. it was proposed by Hoerl and Kennard in 1970 [23]. This regularisation modifies the RSS by adding square of magnitude of the coefficients as the penalty term. From the equation 2.2 by adding penalty term it becomes

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.3}$$

which is equal to

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.4}$$

In the above equations λ is the tuning parameter to decide how much we want to penalize.

Lasso Regression

Lasso regression was proposed by Robert Tibshirani [24] for least absolute shrinkage and selection operator. Lasso uses L1 norm so it is called L1 regularization. Lasso differs from ridge regression in a sense that it uses absolute value of the coefficients where as ridge uses squares of the coefficients. From the equation 2.2 by adding penalty term it becomes

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2.5}$$

which is equal to

$$RSS + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2.6}$$

Elastic Net Regularization

Elastic Net method is a combination of both ridge regression penalty term and lasso penalty term. This is a hybrid which uses both the penalties. The extent to which each penalty term should be used is regulated by α. Consider the penalty for lasso as $\lambda_1(\beta)$ and penalty for ridge as $\lambda_2(\beta)$. The elastic net becomes ridge regression if α is 0 and it becomes lasso regression if α is 1. Now the penalty for elastic net is derived as

$$\lambda_{enet}(\beta) = \gamma[\alpha\lambda_1(\beta) + (1-\alpha)\lambda_2(\beta)] \tag{2.7}$$

where γ is the regularization strength.

2.1.2 Selection Criteria

For all the models as shown in the figure 2.1 on page 6 we observe the weights that we train $\beta_{k,n}$ where k is the number of the model for each feature f$_n$ in X$_{train}$, where n = 1,...N. We get useful information from distribution of weights across the K models for all the features f$_n$ from β_n = ($\beta_{1,n}$, ..., $\beta_{K,n}$). All the vectors β_n, where n = 1,...,N form a matrix B of dimension (K × N). The average frequency $c(\beta_n)$ can be calculated by counting number of times a feature is selected in all the K models, it can be calculated as:

$$c(\beta_n) = \frac{1}{K}\sum_{k=1}^{K} 1_{[\beta_{K,n}\neq 0]} \tag{2.8}$$

The mean and variance of the weights of the features, $\mu(\beta_n)$ and $\sigma^2(\beta_n)$ are:

$$\mu(\beta_n) = \frac{1}{K}\sum_{k=1}^{K}\beta_{k,n}, \tag{2.9}$$

$$\sigma^2(\beta_n) = \frac{1}{K-1}\sum_{k=1}^{K}(\beta_{k,n} - \mu(\beta_n))^2. \tag{2.10}$$

Here, we consider three main criteria to select a feature f$_n$ in RENT :

1. Frequency of selection of f$_n$, $c(\beta_n)$ across K models.
2. The estimation of parameters has stable sign and does not change between postive and negative sign.
3. The feature has a reliably high parameter estimates which are not zero and has low variance across K models.

All the three criteria for selecting f$_n$ can be expressed as:

$$\tau_1(\beta_n) = c(\beta_n),$$

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

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

where t_{K-1} is the cumulative density function of the t-distribution with K-1 degrees of freedom [19].

The feature selection criteria from the metrics $\tau_1(\beta_n)$, $\tau_2(\beta_n)$, $\tau_3(\beta_n)$ can be defined by cutoff values t_1, t_2, $t_3 \in [0, 1]$. Any feature from the list of all features is added to the selected feature list if all the three criteria are satisfied : $\tau_i \geq t_i$, ∀i ∈ {1, 2, 3}. The user can regulate the feature selector by tuning the threshold values of t_1, t_2 and t_3. The number of selected features increases as we reduce the thresholds of these three values and decreases if we increase the threshold.

2.2 Principal component analysis

2.2.1 Understanding PCA

A dimensionality reduction method like principal component analysis (PCA), can be described as a technique prominently used to reduce dimension of a large data set to a smaller data set which retains most of the information present in the large data set. By reducing the dimension the noise also gets reduced which helps in extracting important information in the data. PCA is predominantly used in fields like image compression [25], facial recognition. It is also used in data mining, finding patterns in high dimensional data, bioinformatics, chemometrics, cancer study, etc.

PCA is considered as a feature extraction method in machine learning world. It achieves dimensionality reduction by projecting the data to a new space where the axes are orthogonal. The newly obtained variables after reducing the dimension are called principal components(PC). Some prominent properties of principal components are:

1. These are linear combinations of the original variables.
2. All the principal components are orthogonal to each other.
3. The first principal component has the highest variance, the second PC has the next highest value of variance and so on.
4. Principal components are uncorrelated.

2.2.2 Main steps in PCA

PCA involves the following steps:

1. Standardise the data.
2. Compute the covariance matrix
3. Compute eigen vectors and calculate the corresponding eigen values
4. Sort and pick k eigen vectors with k largest eigen values
5. Transform the original matrix from selected k principal components

Details on how we approached each step are given below.

Standardise the data

In PCA, if a feature has high variance it gets to dominate the first principal component more than the features having low variance in data. For standardisation we scale the data so that its mean is 0 and standard deviation is 1. By this we have all the features in the same scale.

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x} \tag{2.11}$$

Where $x^{(i)}$ is value of a particular sample, μ_x is the mean and σ_x is the standard deviation of the feature.

Compute the covariance matrix

Covariance can be described as a measure of how a feature changes with the change in an other feature. It is used to understand the relationship between two features.

A positive value in covariance indicates that if value of one feature increases the other feature's value increases. A negative value in covariance refers to a inversely proportional behaviour in both the features.

The covariance, σ_{jk} between two feature vectors x_j and x_k can be calculated by using:

$$\sigma_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

Where n is number of samples, $x_j^{(i)}$, $x_k^{(i)}$ are i^{th} values in each of the features *j* and *k*, μ_j and μ_k are mean of the features.

A covariance matrix is a symmetric matrix which contains the values of covariance between each pair of element in a given vector. A random vector with *n* elements has a covariance matrix with dimensions *n* × *n*.

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$$

The above matrix represents a covariance matrix, Σ of a random vector with 4 elements, with dimensions 4*4.

Compute eigen vectors and calculate the corresponding eigen values

The eigen vectors can be said to be the vectors in the direction of maximum variance for the covariance matrix. They can also be referred to as principal components of covariance matrix. Eigen values on the other hand are the magnitudes of the eigen vectors. An eigen vector having maximum eigen value corresponds to maximum variance.

For a matrix A, the eigen vector must hold the following equation:

$$A\vec{v} = \lambda\vec{v}$$

where $\vec{v}$ is eigen vector of unit magnitude and λ is eigen value associated with $\vec{v}$, and is a scalar. The above equation can further be written as

$$A\vec{v} - \lambda\vec{v} = 0$$

$$\vec{v}(A - \lambda I) = 0$$

where I is identity matrix.

Sort and pick k eigen vectors with k largest eigen values

Sort the eigen values calculated with respect to the eigen vectors in descending order. Select the k largest values from the list. The value of k depends on how many principal components we wish to compute, this determines the dimensionality we want to consider.

Transforming the original matrix from selected k principal components

In the last step we change the axes from the original axes to principal component axes. After transformation the new axes can be represented by principal components.

$$Transformed\, data = feature\, matrix * K\ eigenvectors$$

$$T = XP$$

2.3 Partial least squares regression

Notation used

Matrix X contains I observations and J predictor variables stored in an I x J matrix. Matrix Y contains has I observations and K dependent variables in an I x K matrix.

Partial least squares(PLS) regression was introduced in the year 1983 by Wold, H [26] as an extension to ordinary multiple regression. In multiple regression analysis we observe the effects on response variables from a set of predictor variables. PLS regression is used when the number of factors are more than that of number of observations or high multi-correlation between factors. In this condition multiple regression fits the model perfectly but most likely fail to predict new data.

In situations like this we can also use principal component regression, where we can find principal components which explain the maximum variance in the data. But this approach explains the information in the predictor variables but can fail to explain the predictor variables[27].

In the cases of over-fitting, there is chance of looking at latent variables that explain most of the variance in the response variables. These latent variables are linear combinations of predictor variables. The linear combinations are derived such that the dimensionality is reduced to a lower dimension which helps in deriving a relation between predictors and responses. In addition PLS regression maximises the covariance between them.

PLS regression comes in handy when (1) the number of observations is less than number of variables which results in overfitting as calculation of regression coefficients is not easy and (2) when the correlation between the predictor variables is high which results in wrong sign for the regression coefficients[28].

2.4 Sequential and Orthogonalized Partial Least Squares Regression

In recent times, a lot of multiblock data has surfaced in different disciplines. Multiblock data can be data organised into blocks by similar type of measurements. These blocks will have different level of variation within them and modelling all data together may not be optimal. Due to which the need for multi-block analysis techniques has increased over the time [15]. Many partial least squares regression based multi-block techniques have proved useful. One such regression technique used in this thesis is sequential and orthogonalized partial least squares(SO-PLS) regression.

SO-PLS handles the multi dimensionality present in the data without the need of any prior pre-processing i.e. it does not vary with the differences in the scale factor of the blocks. In addition it can handle different number of underlying components in blocks. This advantage of computing optimal number of components for each block gives SO-PLS an edge over most of the current state of the art multi-block algorithms.

2.4.1 SO-PLS Model

In this section we discuss about the working principle of the SO-PLS model. For easier explanation of the method we take two input blocks, let us call them X and Z. And the output data block, also called target block is Y. The common dimension for all the blocks is the number of rows. The linear regression model for this multiblock data can be presented as:

$$Y = XB + ZC + E \tag{2.12}$$

In the above equation X and Z are predictor blocks with dimensions $(N \times J)$ and $(N \times Q)$ respectively, whereas the response block Y has a dimension of $(N \times K)$. The regression coefficients for the linear model B and C will have dimensions $(J \times K)$ and $(Q \times K)$ respectively. E, being the residual matrix has a dimension which is equal to $(N \times K)$.

The SO-PLS implementation involves iterative use of PLS regression and orthogonalization as represented in Figure 2.2

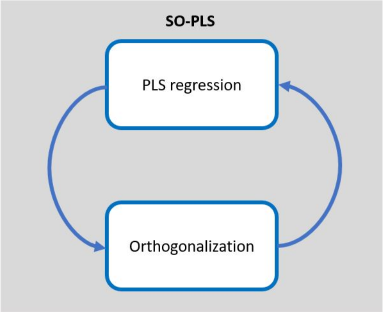

Figure 2.2: *Sequential use PLS regression and orthogonalization.*

The SO-PLS algorithm can be explained in four steps [29]:

1. Perform regression on the first block, i.e. Target block Y is fitted to predictor block X using PLS regression.
2. The second predictor block Z is orthogonalised with respect to the scores from the PLS regression (T_X) in step (1), from which we get Z_{Orth}.
3. The residual of Y from step 1 is fitted to Z_{Orth} by using PLS regression.
4. Now the predictions from both the PLS regressions in step (1) and (3) are summed and final prediction for Y is obtained.

As shown in the above Figure 2.3 on the next page the SO-PLS implementation involves two major steps, PLS regression and orthogonalization. If the number of blocks in the multi-block data is more than two, then the process of performing orthogonalization and PLS regression are repeated for all the models, i.e. we repeat the steps (2) and (3) before predicting the target block in step (4).

2.4.2 Choosing optimal number of components

To find the optimal number of components for a SO-PLS model we individually estimate the number of latent variables for each PLS model. The number of optimal components chosen for one block influences the number of components selected for the following blocks. In such cases using cross-validation proven to be a good practice when using PLS models[29].

In general two different types of approaches are used for choosing optimal number of components in SO-PLS - sequential approach and global approach[30]. In the sequential approach we select the optimal number of components in one block and then move to the other blocks in a sequential order. Here the number of components in a block stay fixed when choosing the components in the following block. In the

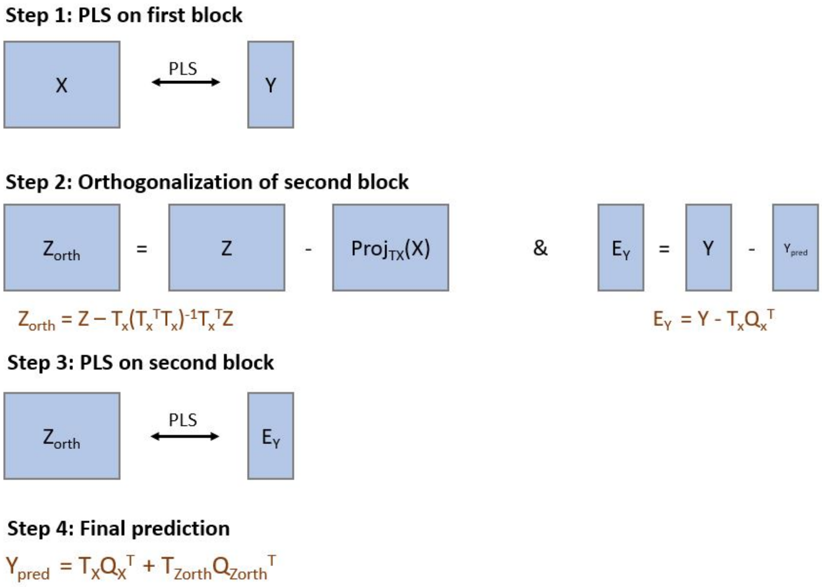
Step 1: PLS on first block
X
PLS
Y
Step 2: Orthogonalization of second block
Z_{orth} = Z - $Proj_{TX}(X)$
&
E_Y = Y - Y_{pred}
$Z_{orth} = Z - T_x(T_x^TT_x)^{-1}T_x^TZ$
$E_Y = Y - T_xQ_x^T$
Step 3: PLS on second block
Z_{orth}
PLS
E_Y
Step 4: Final prediction
$Y_{pred} = T_XQ_X^T + T_{Zorth}Q_{Zorth}^T$

Figure 2.3: *Step wise representation of SO-PLS.*

global approach we choose the number of components in each block based on best global performance.

In this thesis sequential approach is used as there is less chance of overfitting by chance. Though sequential approach of selecting number of components is more time consuming it guards better against overfitting.

The optimal number of features selected by the SO-PLS model can be selected using a graphical plot called ”Måge plot”. The horizontal axis in the plot has number of components and the vertical axis has root mean squared error of cross-validation(RMSECV). In the graph we can see the combination of components in each block. By finding the local minimum in the graph we get optimal components for the blocks.

2.5 Validation

2.5.1 Cross-validation

Cross-validation is a technique in statistics used for machine learning model evaluation. It is a process of repeatedly splitting the data into two subsets, one called training set and the other called test set. We split the data using sklearn’s train test split class. Then we train the model on training_set and test it on test_set. Figure 2.4 gives the schematic representation on how cross validation works.

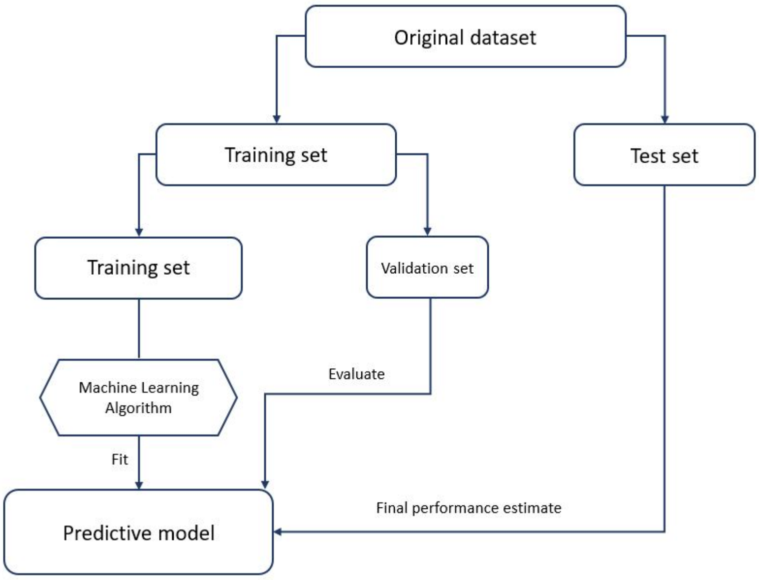

Figure 2.4: *Schematic representation of cross-validation*

Though cross validation helps in evaluating the model's performance it has drawbacks as we remove a part of the input data in training the model. This affects model capability to learn all underlying factors as we are not training the whole data. We risk losing important patterns which we left in the test data. This in turn introduced error by bias. To avoid this we use K-fold cross validation.

2.5.2 K-fold cross validation

In K-fold cross validation we divide the whole data set into k subsets. For every iteration we take one set as validation set and remaining subsets as training sets. By doing this we reduce the bias significantly as we use all the data for training and reduce the variance as we use almost all the data for validation at least once.

There is another type of K-fold cross validation which takes into consideration the distribution of the data. It is called stratified k-fold cross validation.

2.5.3 Stratified K-fold cross validation

Stratified k-fold cross validation uses stratification before dividing the data into k subsets. Stratification is a process in which data is rearranged such that every subset is a good representation of the total data. In this every data point gets tested only once and will be in the training set k-1 times. Stratified K-fold is generally used for classification problems as we wish to have approximately same class distribution in each fold.

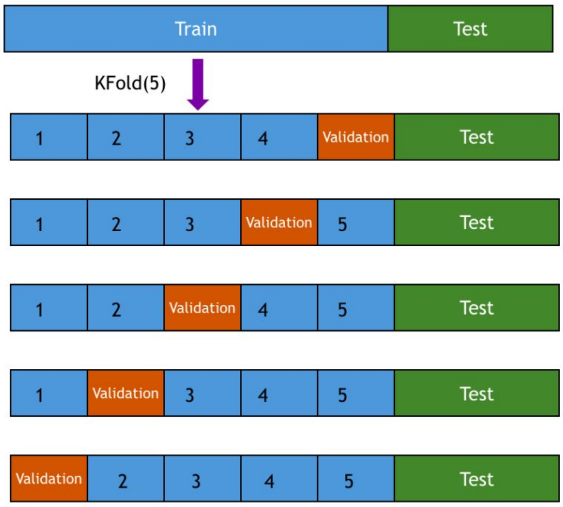

Figure 2.5: *Schematic representation of K-fold cross validation*

2.5.4 Repeated stratified k-fold cross validation

Repeated stratified K-fold performs stratified K-fold cross-validation a number of times specified by the user. It helps in improving the estimated performance of any machine learning model.

After the first stratified K-fold cross validation is done, the samples are reshuffled in stratified manner into the same number of folds and a new stratified K-fold is run. This process is repeated depending on the repeats specified by the user.

After performing the stratified K-fold cross validation the result is the mean of all the cross validation models.

Chapter 3

Materials

The data is obtained from researchers at Functional and Molecular Imaging research group at the Oslo University Hospital. The data contains different blocks which have the clinical properties of cancer patients.

In the below sections, we describe the data blocks and the features in them. We have two blocks of data which represent different characteristics of patients. The first data block has features about patient's clinical properties. The second data block contains information about the blood values of the patients.

3.1 First data block features

Below we give an overview of the features in first data block:

Column Name	Description	Feature type	Values
DATEBRTH	Age of the patient	Date	Date
DATEMET-DATEDIAG	Number of days between the date of diagnosis and date of metastasis	Numerical	Number of days
SEX	Gender	Nominal	Male/Female
PRIMTUM	Location of the primary tumor	Nominal	Colon, Esofagus, Gastric, Pancreas, Rectum, Others
PRTUMRES	If the primary tumour is resected	Nominal	yes/no

OPT	Other prior therapy	Nominal	RADTHRPY, STRPTCYT, SANDOSTN, INTRFERN, NONE, OTHRPRTH
SURGMET	Surgery of metastasis	Nominal	yes/no
SMOKHAB	Smoking habits	Nominal	Smoker, Ex-Smoker, Non-Smoker,
PROTHRCA	Prior other cancers	Nominal	yes/no
MORPH	Morphology	Nominal	Small cell carcinoma, Large cell carcinoma
KI67	Indicator of rate of cell growth	Numerical	percentage
CGA1	Cancer associated gene value	Nominal	Negative, Partly Positive, Strongly Positive
SYNAPTOF	Immunohistochemical factor	Nominal	Negative, Partly Positive, Strongly Positive
OCTREO	indicator for octreo scan	Nominal	Negative, Pos. <Liver, Pos. >Liver
SOM	Organ metastasis at the start of chemotherapy	Nominal	LIVER, LYMPHNDS, LUNG, BONE, OTHRORGM, BRAIN
PERFSTAT	WHO performance status	Nominal	WHO 0, WHO 1, WHO 2, WHO 3
BMI	Body mass index	Numerical	numeric values
HORMSYMP	Hormonal symptoms	Nominal	yes/no
CARSYNDR	Carcinoid syndrome	Nominal	yes/no
TIMETOTRM1	Days between diagnosis and first treatment	Numerical	Number of days
RESPONS1	How patient respond to treatment	Nominal	Complete Response(CR), Partial Response(PR), Progressive Disease(PD), Stable Disease(SD)

3.2 Second data block features

The second block of features represent the blood values as listed below.

Column Name	Description	Feature type	Values
HIAA	The 5-hydroxyindoleacetic acid test used to monitor carcinoid tumors.	Nominal	>2UNL, >Normal<=2UNL, Normal
CGA2	Chromogranin A test is used as tumor marker	Nominal	>2UNL, >Normal<=2UNL, Normal
HMGLBN	Hemoglobin values	Nominal	<11 g/dl, Normal
LACTDHDR	Lactate dehydrogenase	Nominal	>2UNL, >Normal<=2UNL, Normal
PLATELTS	Blood platelets count	Nominal	>400x10ˆ9 / L, Normal
WHITEBLD	White blood cells count	Nominal	>10x10ˆ9 / L, Normal
CRETININ	Describes if creatinine level is normal or not	Nominal	Normal, >Normal
ALKPHSPH	Alkaline phosphatase	Nominal	>3UNL, >Normal<=3UNL, Normal

3.3 Response variable block

The response block is a single dimensional variable. It is a continuous variable which is number of days between first diagnosis and the last observation of the patient. It can be interpreted as a measure of how long the patient lives after being diagnosed with cancer. This makes the problem a regression problem whose aim is to determine the factors responsible for larger value in the response variable.

Chapter 4

Analysis workflow

The workflow of the thesis is divided into the following sections:

- Data exploration and preparation
- Feature selection using RENT
- SO-PLS modelling

At the start of this chapter we start by an overview of the software used in this project. As part of data exploration we describe techniques used for data visualisation, handling missing data, checking for any deviating data points. In the next section we talk about feature selection using repeated elastic net technique(RENT). Finally we apply sequential and orthogonalized partial least squares regression.

4.1 Software

In this section, the software's used and their versions are presented in Table 4.1.

Software	**Version**
Python	3.7.4
Anaconda	4.9.1
Scikit-learn	0.22.1
Numpy	1.19.5
Pandas	0.25.3

Table 4.1: *Software versions*

4.2 Data Exploration and Preparation

Data exploration plays a very important role in having an initial look at the data. The data exploration section is presented in following steps:

- Data pre-processing.
- Data visualisation.
- Checking for Deviating observations using PCA.
- Determining potential target variables.
- Handling features having missing values.

4.2.1 Data pre-processing

As part of data pre-processing, we started with taking an initial look at raw data which was read by using the python package pandas. We started by figuring out how many samples and features each data block contains. This gave us an idea on how the next steps are to be performed as we would have a rough idea what the values are in each of the columns.

Data Description and preparation

The data received from the owners contain 4 data blocks. Each data block contains 80 samples and all together combined 99 features. The common axis across the blocks is the number of samples. As part of pre-processing and discussions with Functional and Molecular Imaging research group at the Oslo University Hospital, the number of features were reduced from 99 to 35 features. Identifying potential target variables will be discussed in the 4.2.4 on page 31.

The preparation of data was done in 4 steps :

1. Handling missing data.
2. Feature engineering and transformation.
3. Converting categorical variables to indicator variables.
4. Data experts advice.

Handling missing data

Once we read the data, it is necessary to make sure that the data has no missing values. This is because scikit-learn and SO-PLS code do not handle missing data. When checked for missing values we found that there are many features which have a considerable fraction of its samples missing. These features were removed from the data blocks so the remaining features have missing values which can be replaced by imputation techniques.

The following Table 4.2 shows number of features with missing values in each data block.

Block	**Total features**	Features with missing values
Block1	34	4
Block2	18	1

Table 4.2: *Missing values for each data block*

There is a trade-off by including seemingly important features having missing values, we lose some patients. These decisions have been made with the researchers and clinicians of the group. This is discussed in the next section.

In the remaining data which have columns free from more than 3 missing values, we have to impute the missing values. There are different missing data imputation techniques. In case of missing values we have to identify what type of missing it is. In general there are 2 main types of missing, missing at random (MAR) and missing not at random (MNAR).

In our data we see that the missing values can be classified as missing at random based on the feedback from clinicians. It means the probability of data point to miss is completely random. In cases like this, the most common type of simple imputation is mean or median imputation. In mean or median imputation the missing value is replaced with mean or median of the values in the feature.

The advantages of using mean or median imputation is that it is simple to implement. By using mean or median as a replacement we do not introduce any unwanted bias in the data [31] unless if the data is already biased.

In the remaining columns in the data, we had two features KI67, BMI which have one missing value each. By using mean imputation, the missing N/A values were replaced by the mean of their respective feature values. This imputation enables us to have these features for further models which helps in improved prediction capacity of the model. With further discussion with the clinicians and the feedback of their domain knowledge considering which features should be important for the model, we proceed with including these features.

Feature engineering and transformation

The next step after removing the columns with missing values is feature transformation. Feature transformation is a technique to modify or derive features from existing features while keeping the original information intact. This helps in making data readable by the model to which the data is used as input.

In our data there are features which cannot be used in any machine learning models and needed transformation. One such feature is ’DATEBRTH’, which has samples as type date. Generally, machine learning models do not take date type as input. We changed this feature by modifying the date type object to numerical value which is years and replaced in the place of DATEBRTH.

Similarly, there are features like Date of diagnosis, DATEDIAG and Date of metastasis, DATEMET. These features do not have any individual value for using them in data models. We have derived a new feature by subtracting DATEDIAG from DATEMET which gives the time takes for metastasis. We removed these two features and replaced them with a single feature which has numerical values, i.e. number of days taken for cancer metastasis.

Converting categorical variables to indicator variables

Categorical variables are those variables which take a limited number of possible values. Most of the machine learning models we use such as regression models, support vector machines requires the input to be numerical. For us to use these models categorical variables should be converted to numerical variables.

Categorical features are classified into two types - Nominal features and Ordinal features. Nominal features are those which do not have any particular order of precedence. Examples of nominal features are city names, sex, etc. On the contrary ordinal features have an order or scale associated with them. A feature like ’customer satisfaction survey’ can be a good example of ordinal feature which takes values ”not satisfied”, ”satisfied”, ”highly satisfied”.

In the data, we have many categorical features which needs to be converted to numerical features. There are many binary variables in the form of strings which need to be converted to numerical. For example there are variables which determine if there is an occurrence of cancer metastasis in lung, bone, etc. These variables take values Yes or No. It is necessary to convert them to numerical values for the model to include them. So we converted these kind of variables to binary that is 0 or 1.

Then, there are categorical variables which take more than 2 distinct values. For such variables we created dummies using pandas get_dummies() utility function. For example there is a variable called ’PRIMTUM’ which takes 6 distinct values. We converted this variables to 6 distinct dummy variables which take values according to the tumour type. This type of encoding enables us to use these variables in the regression models.

Data experts advice

As the data we use is real world data, it is necessary to consult colleagues with domain knowledge to make sure which features to include and prove to be important for the final model. As part of our initial data discussions with the data owners, we discussed on which data variables can be excluded.

This is done after we encoded the categorical variables to numerical variables. We excluded those variables which does not make any logical sense to the target. For example there are some samples such as 'HMGLBN' where the clinical tests for calculating hemoglobin were not done. So the value in such case is not done. In such cases including the value 'Not done' will be irrelevant.

The final stage of data preparation is to remove those features that are irrelevant to including. Once the discussions are done with the clinicians and data experts we have the input features in place, we proceed with the next step that is visualising the data.

4.2.2 Data visualisation

Histograms

A histogram is a visual representation of frequency distribution of the data with respect to the features in the data block. In a histogram the entire range of values in a feature are divided into equal intervals and then show the number of values that fall in each interval. The intervals in which the data is divided are called as 'bins'. Bins are generally of same size and adjacent to each other.

Once the data is divided into equal sized bins the number of values that fall under a particular bin are considered as part of the bin. These values in the bin are shown as a rectangle with height equal to the number of values in it.

To understand histograms we take a look at the mathematical representation of the histogram. Histogram can be seen as a function m_i which is used to count the number of samples that fall under a bin. Let us consider n samples which should be part of k bins. Then the histogram m_i can be shown as:

$$n = \sum_{i=1}^{k} m_i$$

$$k = \frac{max(x) - min(x)}{h}$$

where min(x), max(x) are the minimum and maximum values of the samples for a particular feature respectively and h is the bin width.

There are many different ways of deciding on how many bins can the samples be divided into. By using histogram utility function from pandas we can specify the number of bins as a value in the function arguments. If not specified, by default the number of bins will be 10.

There are many statistics which helps in analysis of a histogram. One such statistics which explains the distribution of the data points is skewness. Skewness is used to measure asymmetry in a histograms. We get a symmetric histograms if the distribution is normal. It means that when we consider the mean of the distributions, there will be same amount of data on both sides. Therefore the skewness is 0.

The direction of skewness is measured towards the tail of the distribution. The length of the tail is proportional to the magnitude of skewness. If the tail on the right side of the distribution is longer it is said to have positive skewness. The mean of the distribution will be to the right of its peak. If the tail is on the left side, it has negative skewness and its mean is to the left of its peak.

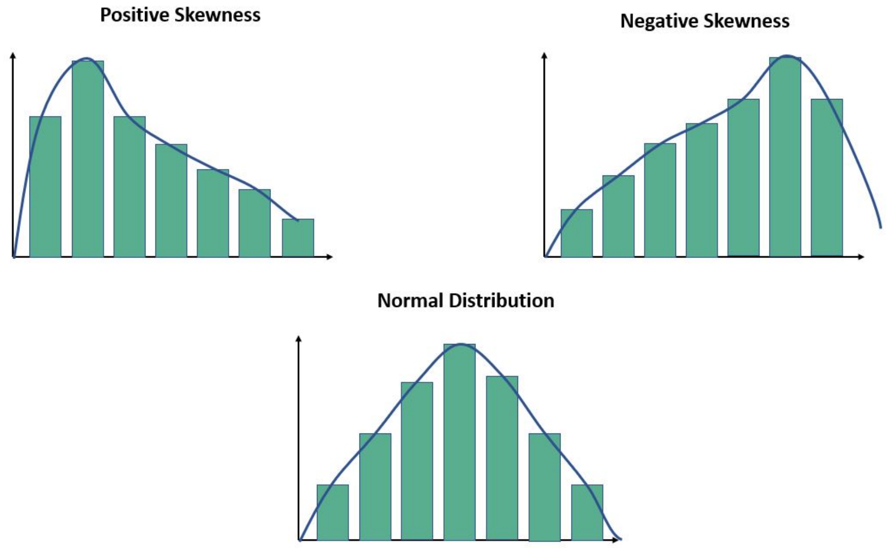

Figure 4.1: *skewed and normal distributions*

If a data is skewed that means it has outliers. Outliers are data points that are different from other data points significantly. Most of the regression models do not perform well on data having outliers. In a histogram with skewness the tail region acts as outliers. This makes skewness an unwanted criterion for performing regression analyses.

Skewness can be reduced by using data transformations. Data transformations make the data symmetric or nearly symmetric. This kind of data is is ideal for many statistical models to handle. Depending on the type of skewness we apply data transformation techniques.

The most common types of data transformations include logarithm transformation, square root transformation, Box-Cox transformation. All these transformations make data as normally distributed as possible. The equations 4.1, 4.2 are for log, square root respectively. In equation 4.3 we see the Box-Cox transformation, here λ varies from -5 to 5. All the values of lambda are considered and the optimal value is selected which gives the best approximation of the normal distribution.

$$y' = log_{10}(y) \tag{4.1}$$

$$y' = \sqrt{y} \tag{4.2}$$

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log y, & \text{if } \lambda = 0 \end{cases} \tag{4.3}$$

We used Box-Cox data transformation on the target variable because the distribution of the target is very skewed and it to make it follow normal distribution.

Correlation plots

Correlation is a term used to measure the strength of possible linear association between two continuous variables [32]. The correlation coefficient is used to represent the correlation strength. The correlation coefficient takes values between -1 and +1.

A correlation coefficient value of 0 indicates that the variables do not have any linear relation between them. The closer the value of correlation coefficient is to -1 or +1, the stronger the linear relation between variables. A correlation coefficient value of +1 indicates that there is perfect positive correlation, and -1 indicates perfect negative correlation.

There are mainly three types of correlation coefficients - Pearson correlation coefficient, Kendall correlation coefficient and Spearman correlation coefficient. Of these we use pearson correlation coefficient as it benchmarks the linear relationship between features. The pearson correlation coefficient is calculated by using

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^{n}(x_i - \bar{x})^2][\sum_{i=1}^{n}(y_i - \bar{y})^2]}} \tag{4.4}$$

where,
r is the correlation coefficient
n is the number of total samples.
x_i, y_i are values of x and y variables for the ith sample.
$\bar{x}$, $\bar{y}$ are mean values of x and y variables.

We use a table representation called a correlation matrix to see the correlation coefficient values between the variables. The correlations between the variables were also visually represented using correlation scatter plots. A scatter plot is used to display relation between two variables. Each sample gets plotted relative to the value in the two variables for which the scatter plot is generated. An example scatter plot is shown in Figure 4.2.

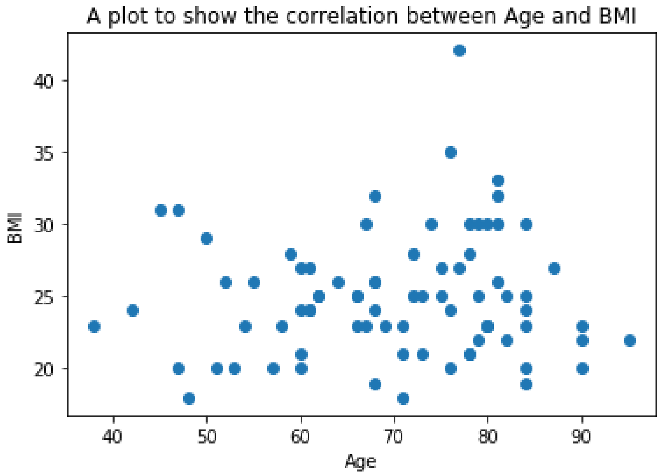

Figure 4.2: *Plot showing correlation between Age and BMI*

4.2.3 Checking for Deviating observations using PCA

The next step in data preprocessing was to find if there are any deviating samples in the data. There are several ways of finding deviating samples one such method we used is principal component analysis(PCA). PCA helps in projecting our existing high dimensional data to a lower dimensional sub-space. We visualise the data using the score plot provided by the PCA.

Using PCA data was visualised using its score and loading plots. By looking at the score plot we get to know how samples are distributed across the space spanned by any two components. By considering the loading plot we get to see if there are any patterns in the features. By superimposing score and loading plots we can identify which samples have higher of lower values of a feature.

The Figure 4.3 on the next page shows a sample superimposition of scores and loading plot. The points Obj1, Obj2, etc. are the samples and Var1, Var2, etc. are the features.

The main purpose we used PCA was to find the deviating observations by using the score and loading plots plotted by considering the more dominant principal components.

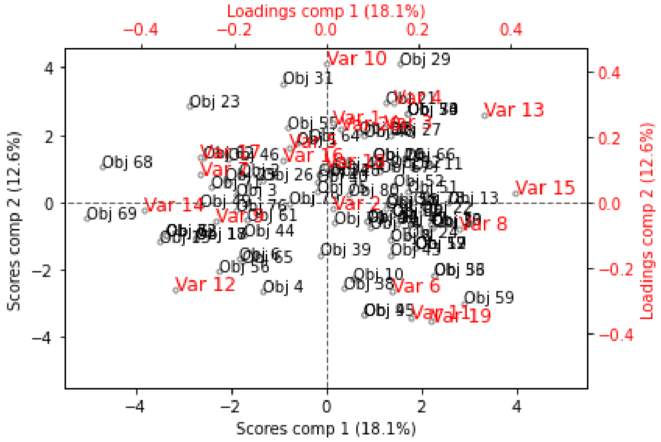

Figure 4.3: Plot showing superimposed plots and loading plots for PCA

4.2.4 Determining potential target variables

In a dataset where we have no core expertise, it is always important to know which variables can be potential target variables. We have considered a couple of potential targets and modelled them to find out how the model performs.

After a series of discussions with the data experts we came up with a target variable which results in a regression problem. We used data transformation to create a new feature computed from two other features in the data.

A target variable called 'Days of survival' is derived from existing variables which can be used for regression problem. This is done by taking the difference of two variables 'DATELOBS' and 'DATEDIAG'. This gives number of days between the date of cancer diagnosis and the date of last observation before death. The values of this feature range from 4 to 3963 days.

As the distribution of values is large, we plotted the histogram of the new target and it is positively skewed. We tried multiple data transformation techniques to reduce skewness like log transformation, Box-Cox transformation so that we can make the distribution near to normal. Finally we fixed the new target as the Box-Cox transformation of the 'Days of survival'.

4.2.5 Including features having missing values

We have excluded the features having more than 3 missing values in them. There is a possibility that these features help in better prediction of the target. This was done by computing different models where we had a trade-off between inclusion of features and dropping patients out of the data to as few missing as possible.

As part of trial and error evaluation, first we build the models using the data blocks having features which do not have missing values and see which features are selected in RENT which is described in section 4.3. Then we ran the multi-block regression model SO-PLS and see the total explained variance as described in section 4.4. We registered the features selected and performance of the models using these features in SO-PLS.

Then we added a feature called 'TIMETOTRM1' which has 4 missing values. This was done at a cost of losing 4 patients. We removed the samples having missing values. We compromised on number of samples for model performance and number of features selected. Now we perform RENT and SO-PLS to see if this improves the number of features selected and explained variance. If there was a significant positive change it helps in increasing the prediction capacity of the model.

In the same way we tried this with multiple features in a trial and error fashion. One such feature which proved to be helpful is 'RESPONS1'. Though the cost of including the feature is more as we have to remove 11 patients in including both 'TIMETOTRM1' and 'RESPONS1'. This showed an increase in the accuracy of linear regression in RENT and also the explained variance in SO-PLS significantly.

Once these features were included we took expert advice in deciding if we can remove the samples to improve the model's performance. After we get a positive feedback we proceed to the next steps in the project. So we have 68 samples and a combined 35 features in both the blocks included.

4.3 Feature Selection using the RENT workflow

For selecting the features that contribute in predicting the target variable we use Repeated Elastic Net Technique for feature selection (RENT). Internally RENT trains an ensemble of unique models using elastic net regularisation to select features. Every model in the ensemble is trained with a randomly selected unique subset of the complete training data [19]. By training these data models we get weight distributions of each feature that contains information on feature selection stability. Further we can define adjustable classification criteria.

In RENT we sample the training set into different subset and train them individually. Apart from that we used repeated stratified k fold cross validation on the data to train data intensively to build robust RENT models. The methodology we used is explained in section 2.5 .

4.3.1 Using repeated stratified k-fold cross-validation with RENT

In our project we used feature selection RENT by combining it with cross-validation techniques. It is to increase the robustness of the model to select features consistently over different cross validated models. By running a simple RENT feature selection we get a set of selected features. If we run the RENT again with a different random state there is a chance we get a different set of selected features. To avoid this we decided to use cross validation techniques which helps in getting more robust estimate of the performance.

As the data set contains 80 samples as discussed earlier, we proceeded with running RENT by implementing 4-fold cross validation as shown in figure 2.5 on page 17. In such case we have 60 samples as training set and 20 samples as test set for every RENT model. These training and test sets were alternated for all the 4 RENT models. For every model we run we get a set of selected features for that selected training and test sets. By doing this we have all the samples in test set at least once. This helps the model to study all the underlying patterns.

To increase the robustness further, we applied repeated stratified k-fold cross validation on the data set. We applied two repeats in the process. By doing so we got a total of (4 × 2) that is 8 RENT feature selections. Here 4 is the cross validation folds and 2 is number of times we are repeating the cross validation. By having more number of models we get to regulate the features selected for the final model by the frequency of selection of features in these 8 models. A schematic representation of 4-fold 2-repeat RENT is shown in Figure 4.4.

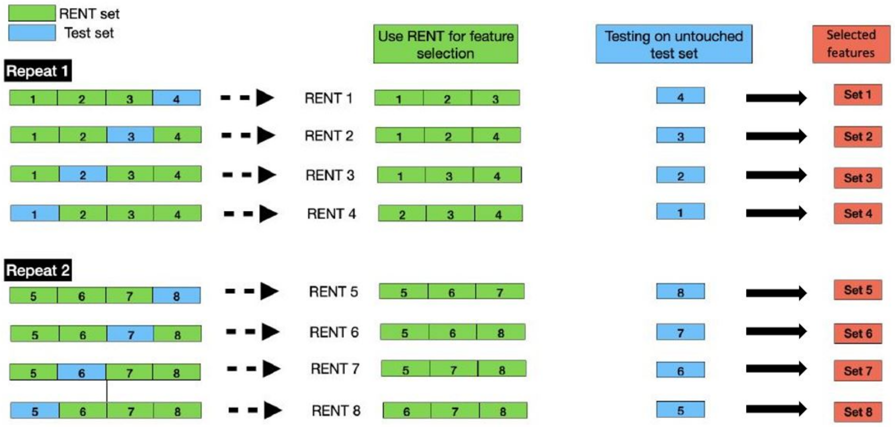

Figure 4.4: *Schematic representation of stratified 4-fold 2 repeat RENT implementation*

For example, if a feature A gets selected in 7 out of 8 models and another feature B gets selected only 2 times out of 8. We get to decide if we want to keep feature B by considering the performance of the final model. In our project we decided to keep the features that were selected at least once out of 8 models.

4.3.2 Applying RENT regression

In this section we describe how we used RENT for our project. As the machine learning problem we have is regression we use the RENT_Regression() class to apply the feature selector.

Parameters

A set of multiple parameters are required when we run the RENT_Regression() class. First of all we pass the training data in the data parameter. In the target parameter variable we pass the corresponding target variable for the train data we input earlier. Then comes the feat_names parameter where we pass the column labels for the training data. This is used to get the names of the selected features later.

The next two parameters we pass are the regularisation parameters for elastic net regularisation. The formula for elastic net regularisation is given by

$$\lambda_{enet}(\beta) = \gamma[\alpha\lambda_1(\beta) + (1-\alpha)\lambda_2(\beta)] \tag{4.5}$$

As the prediction performance of the elastic regularisation model depends on different combinations of γ and α as shown in equation 4.5, it is necessary to define these values we want to use in two different lists my_C_params and my_l1_ratios respectively. The parameter C in RENT represents inverse values for γ. And parameter l1_ratio represents the value of α. An l1_ratio value of 1 is equivalent to L1-regularisation and a value of 0 is equivalent to L2-regularisation. The values between 0 and 1 gives a mix of both L1 and L2 regularisation.

By setting the parameter autoEnetParSel=True we identify the best combination of γ and α from the lists, my_C_params and my_l1_ratios by using 5-fold cross validation. With these selected elastic net values RENT will compute an ensemble of models for selecting the features. If autoEnetParSel=False then we compute k models for all combinations of values in both the lists. This will take a high run time and processing capacity.

The input parameter poly is used if the user wants to include the squares of features and the interactions between them. It comes in handy if the data is non-linear, but the computational cost is high. Another input parameter called testsize_range gives the option to the user for more randomness in ensemble models by allowing the test set sizes to vary in the provided range (lower, higher). To have identical test set size keep both the values same.

Next comes the parameter K. K is number of ensemble models we wish to have in RENT. The higher the number of ensemble models the higher the stability of the model. This is because if we have a higher number of models it gives denser weight distributions which results in high feature selection stability.

Running RENT on our data

By looking at the performance of RENT, we have implemented three repeated stratified K-fold RENT models.

In the first model we applied RENT on the features having no missing values. We observed the selected features for this data blocks and also the prediction scores for linear regression model.

In the next RENT model we included a new feature named 'TIMETOTRM1' which is a transformed feature from 'DATETRM1' and 'DATEDIAG' which has values as days between the diagnosis and first treatment. This comes at a cost of removing 4 samples having missing values. This was done to test if the models' stability in selecting features is improved.

We also included another feature called 'RESPONS1' which is the patients response to treatment. This feature was included at a cost of 5 samples. We repeated the same procedure of running repeated stratified k-fold RENT and saw if the feature selection stability increases.

In the third RENT model we looked at the patients having high mean absolute error. We removed the 3 patient samples that have high mean absolute error. By recording the accuracy scores and RMSEP values we decided on which features to be selected further. The decision to remove the features was done collectively with the thesis supervisors and data experts.

Summary criteria

Once we ran the RENT model its time to define our requirements for selecting features. This was done by defining the selection criteria. As discussed in section 2.1.2 for the selection criteria to be set we need to define the values of t_1, t_2, t_3 using the select_features() utility method. It has three parameters tau_1_cutoff, tau_2_cutoff, tau_3_cutoff which range between 0 and 1. By setting these values we can regulate how aggressively RENT selects features.

We have given a value of 0.9 to tau_1_cutoff which means that we consider only those features that were selected at least 90% of the time across k ensemble models. A parameter value of 0.9 was given to tau_2_cutoff which means it selects features where a minimum 90% of parameter estimates have the same sign. A value of 0.975 was given to tau_3_cutoff which means we test if the weights of the features are high consistently with low variance in k models.

We can change this values to make sure how strict the RENT should select features. This was done by looking at the τ_1, τ_2, τ_3 values for each of the feature by using get_summary_criteria(). If we reduce the cutoff of these three values we select more features and vice versa.

To decide on whether we can compromise on the value of τ_1 we used an utility method provided by RENT: plot_selection_frequency(). This outputs a plot which gives a visual representation of how often a feature's weight was non-zero across all the ensemble models.

We have to be aware that a feature might get selected by elastic net, i.e. the weight is non-zero, but it might be very small and not influential. It can also happen that the sign of the weight might be alternating between positive and negative across models which indicated it is unstable. We used a RENT provided method called get_weight_distributions() to look at more detailed information of full distribution of weights of individual features across all the models. this returns an array containing weights for all ensemble models as rows and features as columns.

Checking the performance using the selected features

Now we have features selected by RENT based on distribution of weights of features across the models. The next step was to check the model performance on the unseen data by using the features we selected by RENT as shown in figure 4.4 on page 33. We used linear regression model to evaluate the performance with the selected features.

The metrics used to evaluate the performance are 'coefficient of determination' denoted as R^2 or 'R squared' and 'Root mean squared error(RMSE)'. 'R squared' is defined as the proportion of variance in one variable explained by the one or more variables. In regression models it is the variance in dependent variable explained by independent variables. The formula for R^2 is given by:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{4.6}$$

where RSS is sum of squares of residuals which is

$$RSS = \sum_{i=1}^{n} \left(y_i - \hat{y}i\right)^2$$

and TSS is total sum of squares given by

$$TSS = \sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2$$

where $\bar{y}$ is the mean of observations.

RMSE can be defined as the residual standard deviation. That is how far the data points are from the regression line. It is given by the below equation

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(y_i - \hat{y}i\right)^2}{n}} \tag{4.7}$$

where n is number of observations,
y_1, y_2,..., y_n are observed values,
$\hat{y}_1$, $\hat{y}_2$,..., $\hat{y}_n$ are predicted values.

4.4 SO-PLS regression

After working with RENT and done with feature selection, the next step was to use the data blocks with selected features in multi-block regression techniques. In this project we used sequential and orthogonalised partial least squares regression as the multi-block analysis method. From feature selection using RENT we have two blocks of features and a target block which is number of days.

In SO-PLS, we do partial least squares regression on each block in a sequential mode. This was done in two approaches to see in which approach the model performs better. In the first approach we started by fitting the target and the first block and then orthogonalise the second block and fit it using PLS regression.

In the second approach we did it in the opposite way. We started by fitting the second block with target using PLS regression. Then perform orthogonalisation on the first block and fit it to the target using PLS regression. There were many ways to see which block to fit to the target, but as there were just 2 blocks we went with this approach.

4.4.1 Selecting number of components in each block

To find the number of components in each block we performed SO-PLS cross validation on one block. By doing SOPLSCV() we can plot root mean squared error of prediction(RMSEP) against number of components. The components with smallest RMSEP value is the ideal number of components for that block.

The plot for RMSEP is given by using the utility method provided by SO-PLS plotRMSEP(). This method call takes parameter which is returned by method RMSECV(). By selecting components with smallest RMSEP that means by these components we can explain maximum variance in the target.

To find the unexplained variance in the target block after fitting the first block from the next block we used the number of components that we selected from first block and perform cross validation SO-PLS. We again consider the number of components for which the RMSEP is smallest. Once we find optimal number of components from each block, we now run the final model SOPLS().

4.4.2 Final SO-PLS model

As the final step we run SO-PLS model. Now we had everything needed in place - the predictor data blocks, target block, the number of ideal components so that the block has least RMSEP. The next step was to run the SO-PLS model and see how it performs on the data with the selected features.

In SOPLS(), we pass a numpy array converted target block, a list of predictor blocks numpy arrays in the order of their execution, number of components to be considered for each block as a list and whether we need to standardise predictor blocks and target block as parameters.

To see the performance of the model and how well it predicted the target we use plotSOPLS() to see different plots. This gives principal components of prediction(PCP) score plot, loading plot of both predictor blocks and target block and a plot showing global explained variance in the target block.

Finally the values of explained variance in the target block by the predictor data blocks is given by the methods Y_cumCalExplVar() and Y_cumValExplVar() which are cumulative calibrated explained variance and cumulative validated explained variance. These methods give explained variance from the first block and the final explained variance from both the data blocks after including the second block.

Chapter 5

Results

This section contains the results for the sections we described in chapter 4. Similar to the previous chapter we divide the results into three sections:

- Data exploration
- Feature selection using RENT
- SO-PLS regression

5.1 Data Exploration and Preparation

In this section we present the results from data exploration and preparation as mentioned in the previous chapter. First we looked at the null values as mentioned in the previous chapter in section 4.2.1. Then as part of knowing the patterns in the data we use PCA on both the data blocks.

5.1.1 PCA on the first block

Here we present the results of principal component analysis on the first block of data. As part of it we have a score plot in Figure 5.1 on the next page and the explained variance for the number of principal components in Figure 5.2 on the following page.

From the score plot in Figure 5.1 on the next page for the first block we see that there is no particular pattern in the scores when we consider the two first principal components. It can also be seen that both the principal components explain just 9.8 + 6.9 = 16.7% of the variance in the block.

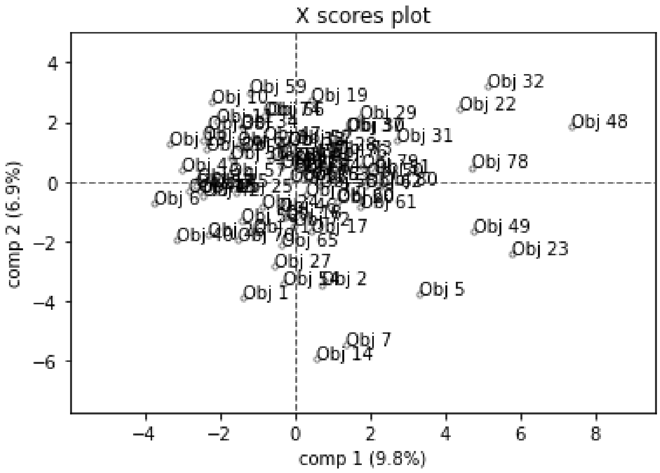

Figure 5.1: *Score plot for the two principal components in first block*

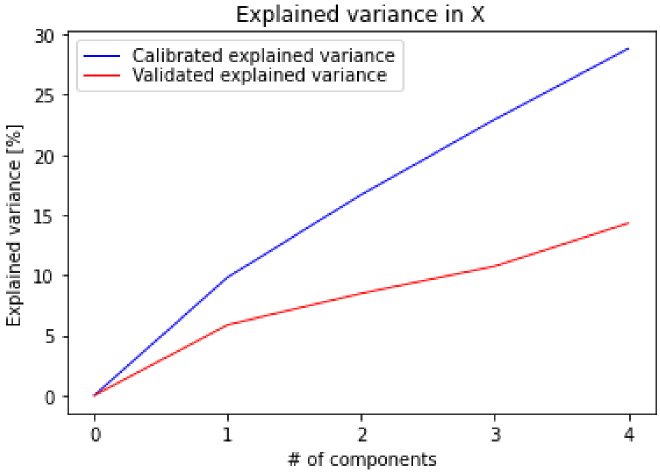

Figure 5.2: *Plot showing explained variance for the first block*

From Figure 5.2, we can see that explained variance is on the y-axis and number of components on the x-axis. It shows as we increase the number of principal components the explained variance increases. It can be seen that if we consider 4 principal components we have a calibrated explained variance of around 30%.

5.1.2 PCA on the second block

From Figure 5.3 on the next page, the score plot on the second block it can be inferred that the data points do not have any underlying patterns when we consider the first two principal components. And the two principal components explain a variance of 18.1 + 12.6 = 30.7% in the data.

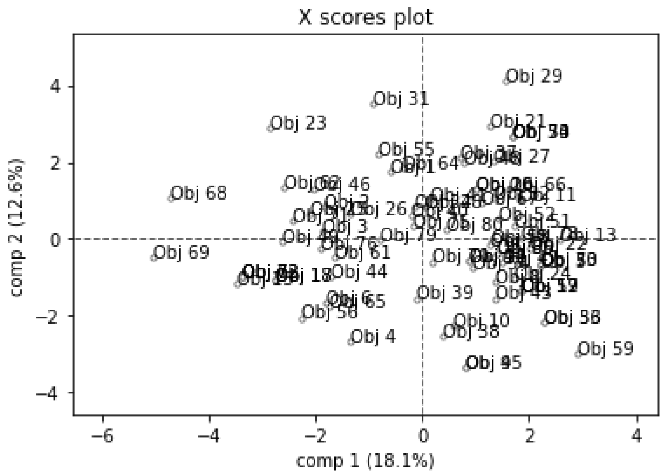

Figure 5.3: Score plot for the two principal components in the second block

Similar to the first block's explained variance plot, in second block we get an explained variance of 50% considering 4 principal components.

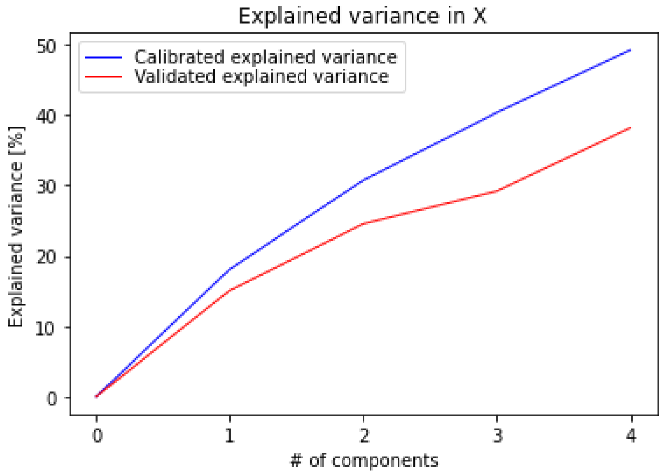

Figure 5.4: Plot showing explained variance for the second block

5.1.3 Preparing the target block

As part of seeing how data is distributed in the target block we plot the histogram of the target variable. This is to see if the data we are using as target variable is unevenly distributed.

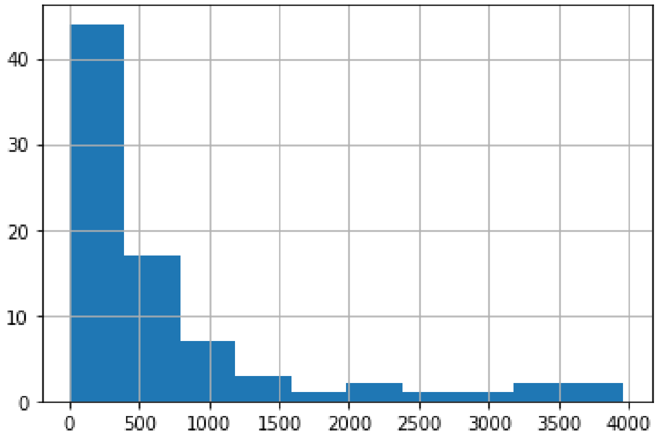

Figure 5.5: *Histogram of target variable*

From Figure 5.5, we can see that the data distribution is uneven. From the histogram we can see that it has positive skewness. We deal with the skewness using data transformation techniques.

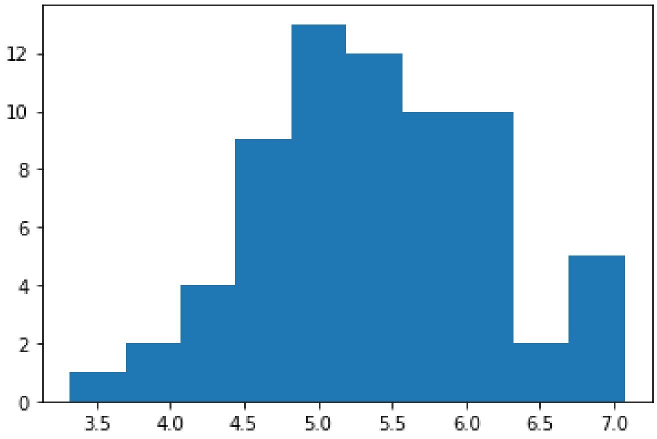

Figure 5.6: *Histogram of target variable after Box-Cox transformation*

Figure 5.6 shows the histogram of the target variable after Box-Cox transformation. It can be seen that the distribution of data values are now more similar to a normal distribution.

5.2 Feature selection results

In this section we have the results and findings from the repeated elastic net technique for feature selection (RENT) as discussed in 4.3 on page 32.

5.2.1 Running RENT without features having missing values

Split No	Selected features	R-squared	RMSEP
Split 1	Age, OCTREO_Negative	-0.32	1157.8
Split 2	CGA1_Negative, SOM_LUNG	0.04	901.36
Split 3	CGA1_Negative, OCTREO_Negative	-0.27	1113.4
Split 4	SURGMET	0.27	387.8
Split 5	AGE, SEX, PRIMTUM_Colon, SURGMET, KI67, SYNAPTOF_Negative	-2.9	866.6
Split 6	Age, SEX, OCTREO_Negative	-0.12	1302.1
Split 7	PROTHRCA_No	0.05	684.4
Split 8	PRTUMRES, SURGMET, MORPH_Other, CGA1-Negative	-0.5	1174.8

Table 5.1: *Features selected by RENT for each cross-validation split without including features having missing values and the performances for first block*

Split No	Selected features	R-squared	RMSEP
Split 1	CGA2_Normal, PLATELTS $\geq$ 400*10ˆ9	-0.08	1045.9
Split 2	HIAA_Normal, CGA2_Normal, LACTDHDR $\geq$ 2UNL, LACTDHDR_Not Done	-0.07	951.8
Split 3	HIAA $\geq$ Normal$\leq$2UNL, HIAA_Not Done, CGA2 $\geq$2UNL, CGA2_Normal, CGA2_Not Done, WHITEBLD_Normal, ALKPHSPH $\geq$3 UNL, ALKPHSPH$\geq$Normal$\leq$3 UNL, ALKPHSPH_Normal, ALKPHSPH_Not Done, TUMMARK1	-0.39	1162.7
Split 4	CGA2_Normal	-0.97	638.5
Split 5	CGA2_Normal	-0.24	484.7
Split 6	CGA2_Normal, LACTDHDR_Not Done	-0.001	1229.4
Split 7	CGA2 $\geq$ 2UNL, CGA2_Normal, CGA2_Not Done, LACTDHDR_Not Done, ALKPHSPH_Not Done, ALKPHSPH$\geq$Normal$\leq$3 UNL, ALKPHSPH_Normal	-1.4	1094.1
Split 8	CGA2_Normal, LACTDHDR$\geq$2UNL, LACTDHDR_Not Done	-0.04	964.5

Table 5.2: *Features selected by RENT for each cross-validation split without including features having missing values and the performances for second block*

From tables 5.1 on page 43 and 5.2 on the facing page it can be seen that the R-squared values are small or negative and the RMSEP values are too large for all the eight splits of RENT. It indicates that all the models have poor performance. We decided to include features having missing values which may or may not improve the performance of the models. We included the features RESPONS1 and TIMETOTRM1 at the cost of removing 8 patients by taking data expert's advice.

5.2.2 RENT after including RESPONS1 and TIMETOTRM1

Split No	Selected features	R-squared	RMSEP
Split 1	PRIMTUM_Colon, SURGMET, RESPONS1-Complete Response (CR)	0.26	712.6
Split 2	PRTUMRES	-0.31	979.1
Split 3	SURGMET, CGA1_Negative, RESPONS1_Complete Response (CR)	-0.22	807.5
Split 4	SURGMET, SMOKHAB_Unknown, CGA1_Negative, SOM_LIVER, RESPONS1_Complete Response (CR), TIMETOTRM1	0.44	872.8
Split 5	RESPONS1_Complete Response (CR)	0.46	405.3
Split 6	PRIMTUM_Colon, SURGMET, CGA1_Negative, SOM_LUNG, CGA1_Strongly Positive, TIMETOTRM1	0.56	728.3
Split 7	PRIMTUM_Colon, SURGMET, RESPONS1_Complete Response (CR)	0.38	422.4
Split 8	RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD)	0.07	1095.9

Table 5.3: *Features selected by RENT and performances for each cross-validation split with RESPONS1 and TIMETOTRM1 included for first block*

Split No	Selected features	R-squared	RMSEP
Split 1	CGA2_Normal, LACTDHDR_Not Done	-0.33	954.1
Split 2	HIAA_Normal, CGA2_Normal	-0.15	918.9
Split 3	CGA2_Normal, LACTDHDR_Not Done	-0.23	811.8
Split 4	CGA2_Normal, LACTDHDR ≥Normal ≤2UNL, ALKPHSPH≥Normal ≤3 UNL:	-0.03	1191.6
Split 5	CGA2_Normal	-0.94	768.8
Split 6	CGA2_Normal, LACTDHDR_Not Done	-0.01	1101.2
Split 7	CGA2_Normal, HMGLBN ≤ 11 g/dL, LACTDHDR ≥Normal≤2UNL	-0.9	741.8
Split 8	CGA2_Normal, LACTDHDR≥2UNL, WHITEBLD ≥ 10 x 10ˆ9 / L	-0.3	1297.6

Table 5.4: *Features selected by RENT and performances for each cross-validation split with RESPONS1 and TIMETOTRM included for second block*

After including the features RESPONS1 and TIMETOTRM1, some of the splits had substantial improvement in predictive performance, but not all, as seen in the Tables 5.3 on the previous page and 5.4. This prompted us to take a look at the average absolute error for each patient across all the ensemble models in the RENT models for all the splits. By observing the patients whose average absolute error is uncommon, we can remove them to improve the performances.

5.2.3 RENT after removing samples having high average absolute error samples

Sample	Test	Average abs error
0	177	722.48
2	186	411.9
6	182	978.4
11	175	646.0
12	181.0	1068.0
14	162	745.92
21	200	5736.7
26	170	1306.1
35	201	1060.3
36	183	1404.9
42	204	3742.9
43	204	3983.5
50	173	1068.3
56	191	1915.2
62	202	1376.0
64	187	1826.1
65	186	1329.5
67	170	564.2
68	203	659.0

Table 5.5: *Average absolute value of 20 patients in a split*

Tables 5.5 and 5.6 on the following page contain the average absolute error of 20 samples in the data for one split of the RENT model for both the blocks. We can see that the patients 21, 42 and 43 as highlighted have high absolute error. It seems that these three patients are not representative for the patient group. There might be other underlying factors that do not show in the data we have, but may be decisive for the response. After multiple discussions with thesis supervisors and data experts we decided to eliminate these three patients.

Sample	Test	Average abs error
0	177	871.9
5	186	994.9
6	188	825.8
11	180	1062.7
12	182	1207.9
14	182	679.1
21	176	4345.8
26	172	1184.8
35	203	1220.6
37	171	876.0
42	189	4133.1
43	209	3944.4
50	188	900.5
56	173	1927.9
62	175	1587.3
64	202	1884.9
65	187	1339.6
68	186	727.8
69	170	893.8

Table 5.6: *Average absolute value of 20 patients in another split*

Split No	Selected features	R-squared	RMSEP
Split 1	SURGMET, CGA1_Negative, RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), TIMETOTRM1	0.34	0.55
Split 2	PERFSTAT_WHO 0, CGA1_Negative, RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), TIMETOTRM1	0.30	0.67
Split 3	RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), TIMETOTRM1	0.22	0.58
Split 4	RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), SMOKHAB_Smoker	0.41	0.68
Split 5	CGA1_Negative, TIMETOTRM1, RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD)	0.67	0.42
Split 6	SURGMET, PERFSTAT_WHO 0, RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD)	0.03	0.86
Split 7	RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), TIMETOTRM1	0.48	0.53
Split 8	RESPONS1_Complete Response (CR), RESPONS1_Progressive Disease (PD), TIMETOTRM1, SMOKHAB_Smoker, PERFSTAT_WHO 0	0.27	0.62

Table 5.7: *Features selected by RENT and performances for each cross-validation split after removing the samples with high absolute error for the first block*

Split No	Selected features	R-squared	RMSEP
Split 1	CGA2_Normal, LACTDHDR $\geq$ 2UNL, HMGLBN $\leq$ 11 g/dL	-0.04	0.69
Split 2	CGA2_Normal, LACTDHDR $\geq$ 2UNL, ALKPHSPH_Normal	0.39	0.63
Split 3	CGA2_Normal, LACTDHDR $\geq$ 2UNL, HMGLBN $\leq$ 11 g/dL	0.37	0.52
Split 4	CGA2_Normal, LACTDHDR $\geq$ 2UNL, ALKPHSPH_Normal	0.37	0.71
Split 5	CGA2_Normal, LACTDHDR $\geq$ 2UNL, ALKPHSPH_Normal	0.18	0.66
Split 6	CGA2_Normal, LACTDHDR $\geq$ 2UNL, CGA2 $\geq$Normal$\leq$ 2UNL	0.02	0.86
Split 7	CGA2_Normal, LACTDHDR $\geq$ 2UNL, PLATELTS $\geq$ 40 x 10ˆ9/L	0.20	0.66
Split 8	CGA2_Normal, LACTDHDR $\geq$ 2UNL, HMGLBN $\leq$ 11 g/dL, ALKPHSPH_Normal, ALKPHSPH $\geq$3 UNL	0.18	0.66

Table 5.8: *Features selected by RENT and performances for each cross-validation split after removing the samples with high absolute error for the second block*

The performances of repeated stratified K-fold RENT for both the data blocks after removing the patients whose average absolute error is high are presented in Tables 5.7 on page 49 and 5.8 on the facing page. It can be seen that the values of R-squared and RMSEP have shown significant positive change from previous RENT models after we removed the patients which indicates improved performance of the feature selection models.

5.2.4 Selected features for SO-PLS modelling

Sl No	**Feature Name**	**No of times selected**
1	SURGMET	2
2	RESPONS1_Complete Response (CR)	8
3	RESPONS1_Progressive Disease (PD)	8
4	CGA1_Negative	3
5	TIMETOTRM1	6
6	PERFSTAT_WHO 0	3
7	SMOKHAB_Smoker	2

Table 5.9: *Number of times features are selected out of 8 splits for the first block*

Sl No	**Feature Name**	**No of times selected**
1	CGA2_Normal	8
2	LACTDHDR $\geq$ 2UNL	8
3	HMGLBN $\leq$ 11 g/dL	3
4	ALKPHSPH_Normal	4
5	CGA2 $\geq$Normal$\leq$ 2UNL	1
6	PLATELTS $\geq$ 40 x 10ˆ9/L	1
7	ALKPHSPH $\geq$3 UNL	1

Table 5.10: *Number of times features are selected out of 8 splits for the second block*

Tables 5.9 and 5.10 show number of times each feature got selected from RENT models in both the blocks. For SO-PLS modelling we select those features which are selected by RENT at least once across all splits.

5.3 SO-PLS regression

In this section we present the results and plots from SO-PLS model. The order of the blocks, i.e. which block should we consider first is determined by trying both the approaches.

5.3.1 Cross validation

Below we provide plots we obtained while doing cross validation to find the optimal number of components we use in each block. Here we try two approaches by considering which data block we include first.

Considering first block as the initial block in cross validation

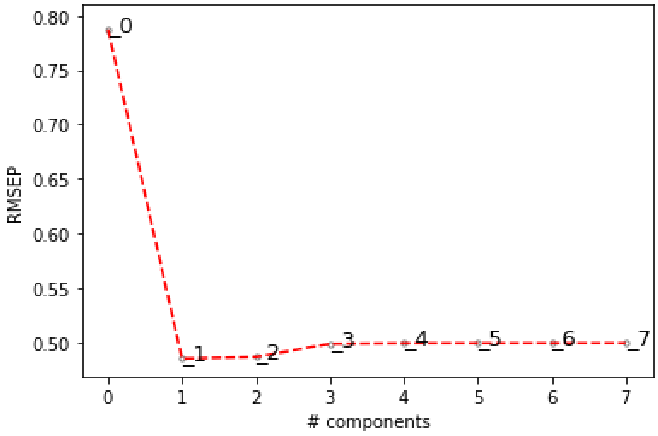

Figure 5.7: *SO-PLS cross validation for first block*

From Figure 5.7, it can be inferred that out of 7 components for just one component the RMSEP value is lowest. So we consider 1 component for the first block and include the next block for cross validation.

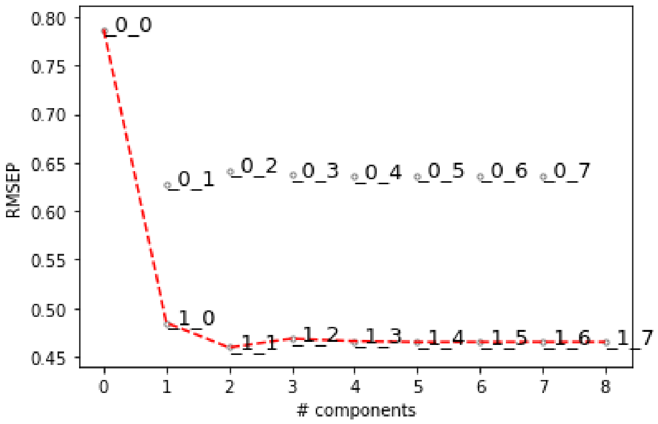

Figure 5.8: *SO-PLS cross validation for both the data blocks considering block 1 first*

Figure 5.8 shows the cross validation results of both the blocks included. It can be seen that from considering 1 component in the first block and 1 component in the second block we get an RMSEP of roughly 0.45 which is the lowest among all possible combinations.

Considering second block as the initial block in cross validation

Similarly, we repeat the same with considering the second block as the initial block for cross validation.

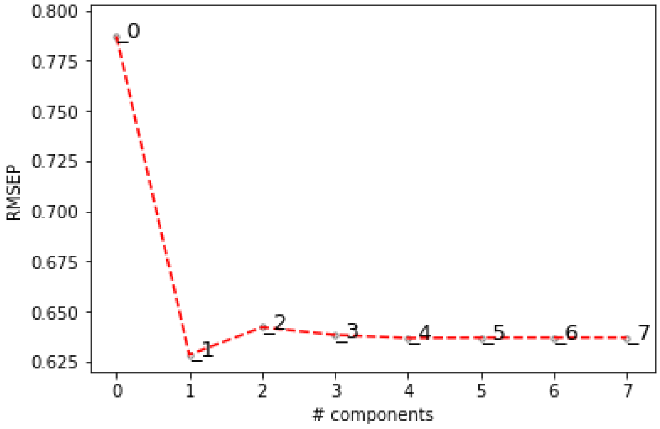

Figure 5.9: *SO-PLS cross validation for second block*

From cross-validation with only the second block as in Figure 5.9 we get that optimal number of components is 1 and gives an RMSEP of 0.625 which is lowest of all.

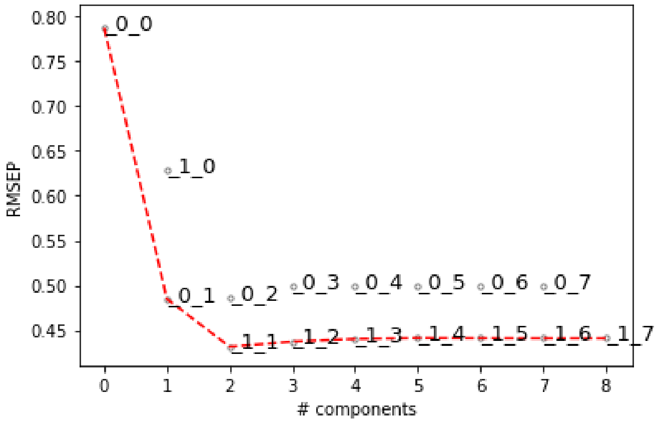

Figure 5.10: *SO-PLS cross validation for both the data blocks considering block 2 first*

Similar to previous subsection, we perform cross validation with both the blocks included having all possible combinations between them. Figure 5.10 gives that optimal number of components as 1 and 1 for both blocks.

5.3.2 SO-PLS model

In this section we have the results of the final SO-PLS model which is run with the optimal number of components for each block obtained from the results of cross validation in section 5.3.1 on page 52.

Order of the blocks	**Explained variance after adding initial block**	**Total explained variance after second block**
Block1, Block2	67.4%	73.9%
Block2, Block1	42.9%	76.4%

From Figure 5.11 on the facing page, we can see that around 100% of the explained variance is obtained from just one principal component. The second component (y-axis) has negligible values (1e-16) which is practically 0. The distribution of the points is from left to right, second component should be ignored.

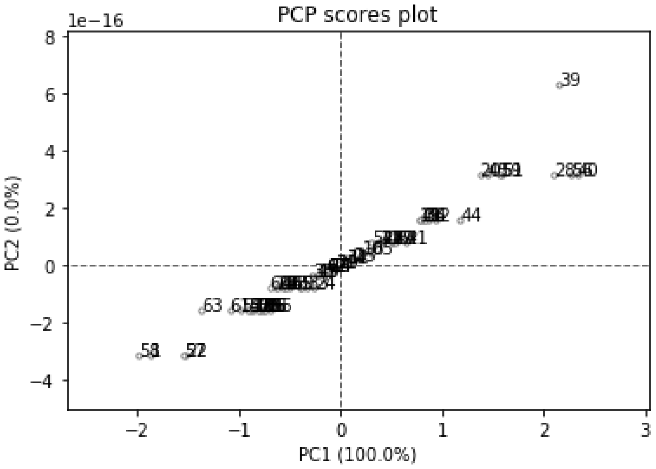

Figure 5.11: *SO-PLS principal component prediction score plot*

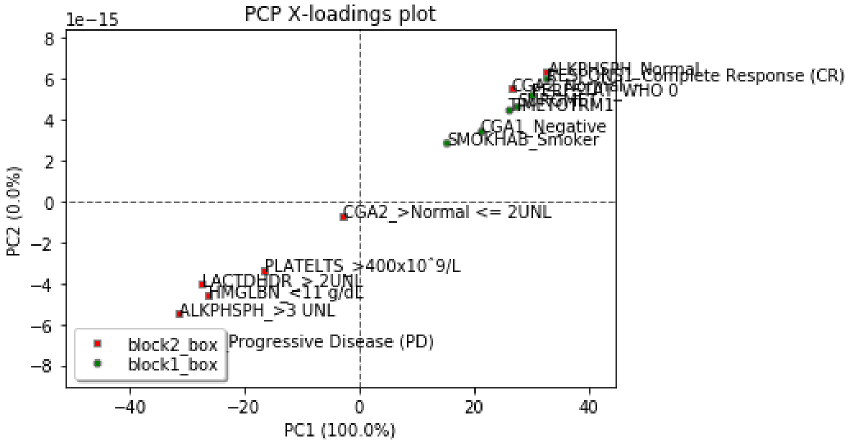

Figure 5.12: *SO-PLS X-loading plot showing the distribution of features*

The loading plot in 5.12 shows the pattern in which the features are distributed with respect to the principal components. By observing the similar patterns in both score and loading plots we can identify the features that are highly correlated with the target feature.

By superimposing the score and loading plot and manually checking the patient feature values, the patients who fall on top right corner in the scores plot live longer than those who fall on the bottom left. That is the values in the target are higher than their counterparts.

In the same way the features in the top right quadrant in the loading plot have a positive effect on the target values (live longer) whereas the features in the bottom left corner have an inverse effect on the target variable.

Explained variance plot for the final model

Plot 5.13 shows the explained variance by considering the optimal number of components in both the blocks.

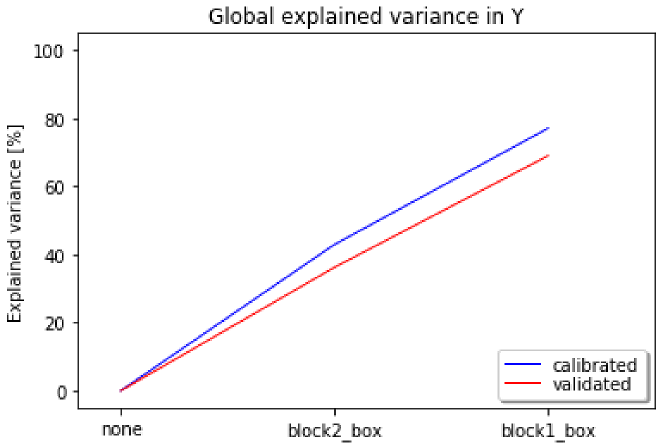

Figure 5.13: *SO-PLS explained variance plot with both blocks*

In the plot we can see the increase in explained variance value when we add the blocks sequentially. Initially the explained variance is 0. By including block 2 we see that the calibrated explained variance is 42.9% and validated explained variance as 36.3%. Further including block 1 it explained the variance that is not covered by block 2 thus increasing the value of calibrated explained variance to 76.4% and validated explained variance to 69.9%.

This indicated by considering just 7 features in both first block and second block we can explain 76.4% of the calibrated explained variance in the target variable using SO-PLS multi-block model.

Chapter 6

Discussion

6.1 Dataset

6.1.1 Features having irrelevant values

The data we used in the thesis was prepared by manually entering values of patient clinical characteristics from the case sheets available at the hospital. The case sheets or patient records contain clinical values recorded at the time of diagnosis or during the treatment of patients. The data entered in the case sheets can be specific to the patients conditions.

Different cancer types require different clinical tests to be performed on the patients. Due to this there are many clinical features which are not recorded for all the patients. They can be in the form of null values or just marked not done. For example if the platelets count test is not done for a patient the value is entered as 'Not Done'. While this might not be relevant for the diagnosis for the doctors, it might turn up to be important feature. Ideally such non-informative features should be removed even if RENT selects them.

The clinical values of patients are collected for the purpose of treatment and not considering the future data modelling chances. So data transformation and feature exclusion required data experts domain knowledge.

6.2 Patients with high average absolute error

From tables 5.5 on page 47 and 5.6 on page 48, the patients 21, 42 and 43 have high average absolute error. By removing these patients and proceeding to model without them proven to help the performance of the model.

An improved performance implies that predicted values of these patients are deviating significantly from the target value. This effects the overall performance of the model. These samples are removed after discussing with data experts and thesis supervisors.

For example, it can be seen in Figure 6.1 that for patient number 42 has an unusual value of 'BMI' compared to others. This may be a reason why the mean absolute error in RENT for patient 42 is high.

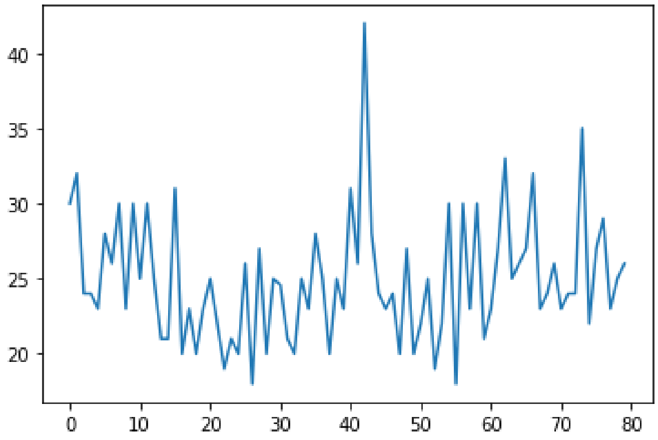

Figure 6.1: *Plot showing values of BMI for all the patients*

For the other two patients 21 and 43 , we could not identify any deviations in the data. This may be because of underlying features that do not show in the data we have.

6.3 Implications for SO-PLS model

6.3.1 Explained variance

The explained variance plot 5.13 on page 56 from the SO-PLS model shows the increase in the variance explained by adding each block to the model. By adding second block to the model the variance explained in target is increased by around 43%. And further adding first block will give a cumulative explained variance of 76.4%.

The increase in the explained variance percentage after adding first block is 33.4%. This is the the additional increase in variance that is not explained by the components extracted from the features in second block.

6.3.2 Overfitting

From Table 6.1 it is seen that the difference between the calibrated and validated explained variances are relatively low. This implies there is no indication of overfitting for the model.

Explained variance	Only second block	After adding first block
Cumulative calibrated explained variance	42.9%	76.4%
Cumulative validated explained variance	36.3%	69.9%

Table 6.1: *Table showing both calibrated and validated explained variance*

6.3.3 Correlation between the features

The X & Y correlation loadings plot shown in Figure 6.2 gives an insight of which features contribute more in explaining the variance in the target block. The features which are close to the target point in the plot have high positive correlation to the target.

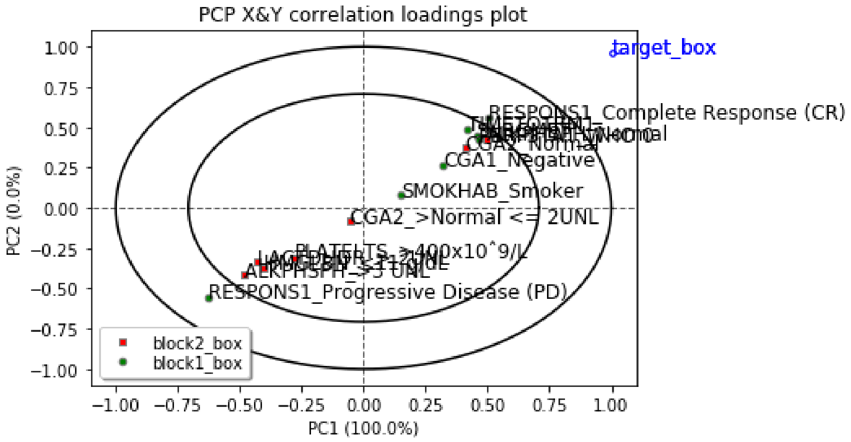

Figure 6.2: *SO-PLS correlation loading plot*

The rings in the plot helps in depicting how much variance of each feature was explained by the components. On the outer ring the percentage of explained variance is 100% and on the inner ring it is 50%. For example it can be interpreted that for the feature 'RESPONS1_Progressive Disease(PD)' roughly 60% of all the variation in this feature is explained by the component as we have only one component.

This implies a lot of information in this feature is relevant for the model.

The features that are closer to the target_box means they are highly correlated to target. That means high values for these features means high values for target block. In the same way the features that are farther from the target_box have very less correlation with the target block. So high values of these indicate the patients do not live longer.

It means that the clinicians having information of these features gets a good initial indication on whether a patient will live longer or shorter on the initial look.

6.4 Possible future work

We have used two blocks of data having multiple features in each. As a future work in the project we can further divide the data available into more blocks. By doing so we can get more insight on how much the features in each block contribute to explaining the variance in target. At this point we cannot guarantee by doing this if the performance of the model increases or not.

In RENT for feature selection we are using stricter values for the selection criteria. That is we are using values of τ_1, τ_2, τ_3 as 0.9, 0.9 and 0.95 respectively. By reducing the values of these cutoffs and rerunning RENT we can select more features for both the blocks. It can turn out these newly selected features will explain more unexplained variation in the target when used in SO-PLS model than we get now.

Also RENT uses linear regression to model linear relationships in the data. In further work one could take into account the feature interactions. This introduces non-linearity and may provide more information for modelling.

Chapter 7

Conclusion

The multiblock data analysis done on healthcare data acquired from patients diagnosed with gastrointestinal carcinoma has showed that sequential and orthogonalized partial least squares (SO-PLS) regression may be a potential tool for finding useful data insights. By exploiting the underlying dimensionality in the data it may help the clinicians in looking out for the features that contribute more to the required target.

Using state of the art feature selection method, repeated stratified K-fold RENT, proved that selecting features based on their performances in linear regression may help in increasing the explained variance of the target variable when using multiblock techniques.

It can be concluded that by using SO-PLS along with RENT feature selection has shown enough potential for further development in the field of health care, the next step of which would be dividing the data blocks to even smaller data blocks to see how important they turnout in knowing how long a patient lives.

Bibliography

[1] "Who cancer stats in 2018," https://www.who.int/, accessed: 2021-02-01.

[2] F. Bray, A. Jemal, N. Grey, J. Ferlay, and D. Forman, "Global cancer transitions according to the human development index (2008-2030): a population-based study," *Lancet Oncol*, vol. 13, pp. 790–801, 01 2012.

[3] "Global cancer stats," https://gco.iarc.fr/today/online-analysis-map, accessed: 2021-02-21.

[4] "What is cancer," https://www.nhs.uk/conditions/cancer/, accessed: 2021-02-02.

[5] "Different stages of cancer," https://www.nhs.uk/common-health-questions/operations-tests-and-procedures/what-do-cancer-stages-and-grades-mean/, accessed: 2021-02-01.

[6] "Types of cancer," https://www.cancer.gov/about-cancer/understanding/what-is-cancer, accessed: 2021-02-02.

[7] "Various factors of cancer," https://stanfordhealthcare.org/medical-conditions/cancer/cancer/cancer-causes.html, accessed: 2021-02-01.

[8] "Cancer diagnosis," https://www.cancer.gov/about-cancer/diagnosis-staging/diagnosis, accessed: 2021-02-02.

[9] "Cancer treatment," https://www.cancer.gov/about-cancer/treatment/types, accessed: 2021-02-02.

[10] H. Varmus, "The new era in cancer research," *Science (New York, N.Y.)*, vol. 312, pp. 1162–5, 06 2006.

[11] A. A. for Cancer Research, "Highlights from recent cancer literature," *Cancer Research*, vol. 81, no. 9, pp. 2257–2258, 2021. [Online]. Available: https://cancerres.aacrjournals.org/content/81/9/2257

[12] K. De Roover, E. Ceulemans, and M. Timmerman, "How to perform multi-

block component analysis in practice," *Behavior research methods*, vol. 44, pp. 41–56, 07 2011.

[13] T. Næs and H. Martens, "Principal components regression in NIR analysis: Viewpoints, background details and selection of components," *Journal of Chemometrics*, vol. 2, pp. 155–167, 1988.

[14] D. Pirouz, "An overview of partial least squares," *SSRN Electronic Journal*, 10 2006.

[15] J. Westerhuis, T. Kourti, and J. MacGregor, "Analysis of multiblock and hierarchical pca and pls models," *Journal of Chemometrics - J CHEMOMETR*, vol. 12, pp. 301–321, 09 1998.

[16] F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, pp. 16–28, 01 2014.

[17] I. Guyon and A. Elisseeff, "An introduction of variable and feature selection," *J. Machine Learning Research Special Issue on Variable and Feature Selection*, vol. 3, pp. 1157 – 1182, 01 2003.

[18] H. Liu, H. Motoda, R. Setiono, and Z. Zhao, "Feature selection: An ever evolving frontier in data mining." *Journal of Machine Learning Research - Proceedings Track*, vol. 10, pp. 4–13, 01 2010.

[19] A. Jenul, S. Schrunner, K. H. Liland, U. G. Indahl, C. M. Futsaether, and O. Tomic, "Rent–repeated elastic net technique for feature selection," *arXiv preprint arXiv:2009.12780*, 2020.

[20] C. Girish and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, pp. 16–28, 01 2014.

[21] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 12 2016.

[22] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society Series B*, vol. 72, pp. 417–473, 09 2010.

[23] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 04 2012.

[24] R. Tibshirani, "Regression shrinkage selection via the lasso," *Journal of the Royal Statistical Society Series B*, vol. 73, pp. 273–282, 06 2011.

[25] A. Arab, J. Harbi, and A. Abbas, "Image compression using principle component analysis," *Al-Mustansiriyah Journal of Science*, vol. 29, p. 141, 11 2018.

[26] S. Wold, H. Martens, and H. Wold, "The multivariate calibration problem in chemistry solved by the pls method," *Lect Notes Math*, vol. 973, 01 1983.

[27] R. D. Tobias *et al.*, "An introduction to partial least squares regression," in *Proceedings of the twentieth annual SAS users group international conference*, vol. 20. SAS Institute Inc Cary, 1995.

[28] L. Carrascal, I. Galván, and O. Gordo, "Partial least squares regression as an alternative to current regression methods used in ecology," *Oikos*, vol. 118, pp. 681–690, 05 2009.

[29] A. Biancolillo and T. Næs, "Chapter 6 - the sequential and orthogonalized pls regression for multiblock regression: Theory, examples, and extensions," in *Data Fusion Methodology and Applications*, ser. Data Handling in Science and Technology, M. Cocchi, Ed. Elsevier, 2019, vol. 31, pp. 157–177. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444639844000065

[30] T. Næs, O. Tomic, B.-H. Mevik, and H. Martens, "Path modeling by sequential pls regression," *Journal of Chemometrics*, vol. 25, pp. 28–40, 01 2011.

[31] H. Kang, "The prevention and handling of the missing data," *Korean journal of anesthesiology*, vol. 64, pp. 402–6, 05 2013.

[32] M. Mukaka, "Statistics corner: A guide to appropriate use of correlation coefficient in medical research," *Malawi medical journal : the journal of Medical Association of Malawi*, vol. 24, pp. 69–71, 09 2012.

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

Postboks 5003
NO-1432 Ås
Norway