ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

Subsampling Based Model Selection for Linear Predictive Models

Author:
Christoforos Mavrikakis

Supervisor:
Dr. Yiannis Kamarianakis

Department of Mathematics & Applied Mathematics
University of Crete
Heraklion, Greece
17 April 2024

Acknowledgements

I would like to express my gratitude to Dr. Yiannis Kamarianakis, for his patient guidance, valuable suggestions, and willingness to give his time generously, throughout the development of this thesis.

I would also like to extend my thanks to the examination committee of my thesis.

Finally, I wish to thank my family and my friends for their encouragement and support.

Contents

1 INTRODUCTION **5**

2 Lasso type estimators **6**
2.1 Lasso 6
2.2 Adaptive Lasso 8
2.3 LAD Lasso 9
2.4 Adaptive LAD Lasso 10

3 Subsampling and Bootstrap methods **12**
3.1 Bootstrap Sampling 12
3.2 Subsampling 13

4 Statistical Validation Techniques **15**
4.1 Confidence Intervals 15
4.2 Bootstrap Confidence Intervals 16
4.3 K-Fold Cross Validation 17

5 Application **19**
5.1 First Stage 19
5.2 Second Stage 29
5.3 Model Performance 31

6 Conclusions **39**

List of Figures

2.1 Lasso estimation illustration 7

5.1 Boxplots for each predictor 21
5.2 Correlation matrix plot 22
5.3 VIF before dropping $X50$ and after. 24
5.4 VIF after dropping both $X50$ and $X63$ 25
5.5 Correlation plot after VIF-filtering 26
5.6 Frequency of inclusion in L2 methods 29
5.7 Frequency of inclusion in L1 methods 30
5.8 Bootsrap based C.I for least squares coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators. . 32
5.9 Bootsrap based C.I for least absolute deviation coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators. 33
5.10 Bootsrap based C.I for least absolute deviation coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators. 34
5.11 RMSE for each method. 36
5.12 Closer look of RMSE for each method. 36
5.13 MAE for each method. 37
5.14 Closer look of MAE for each method. 38

List of Tables

5.1 Summary Statistics for Predictors 20
5.2 Coefficients through all methods 27
5.3 RMSE and MAE for each method. 35

Chapter 1

INTRODUCTION

Nowadays, we live in a data-driven society and statistical modeling stands as a keystone for making informed decisions across a wide range of fields. From applications in healthcare to directing business strategies, statistical models provide invaluable insights by uncovering patterns and relationships within data. The exponential growth of data volume and complexity, fueled by technological advancements, underscores the critical importance of robust model selection techniques in ensuring the reliability and accuracy of predictive models. The choice of predictors in Linear Regression models can significantly impact their predictive power and generalizability. Therefore, creating methods that simplify this process while maintaining accuracy is crucial.

The primary objective of this thesis is to evaluate a Subsampling-based model selection approach for Linear Regression. By utilizing recent advancements in Subsampling techniques and drawing inspiration from existing methodologies, we aim to improve the process of predictor selection while maintaining predictive accuracy. Ultimately, this study seeks to contribute to the advancement of statistical modeling practices by offering a systematic framework for efficient and effective model selection.

In conclusion, this thesis efforts to deal with the critical need for robust model selection techniques in the context of Linear Regression. By utilizing a Subsampling-based approach, we aim to provide researchers with practical tools to handle the challenges of selecting predictors in decision-making driven by data.

Chapter 2

Lasso type estimators

In statistical modeling, Lasso-type estimators have become indispensable tools for navigating high-dimensional data challenges and conducting variable selection. Lasso-type estimators excel in **minimizing coefficients** and **promoting sparsity** within regression models. This deliberate approach enables them to attain a balance, optimizing **predictive accuracy** while maintaining model simplicity.

2.1 Lasso

Lasso, short for Least Absolute Shrinkage and Selection Operator, is a widely used statistical tool in regression analysis. Initially designed as a regularization technique, it tackles overfitting and aids in selecting essential variables in linear regression models. Its primary aim is to enhance the model's **predictive capabilities** and **interpretability** by encouraging **sparsity** in coefficient estimates.

In the context of linear regression, the Lasso alters the objective function of the **least squared errors** by incorporating a penalty term derived from the sum of the absolute values of the coefficients.

The Lasso coefficients minimize the quantity :

$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} |y_i - \mathbf{X}_i^\top \boldsymbol{\beta}|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.1}$$

- n is the number of observations.
- p is the number of predictors.
- β_j represents the regression coefficients.
- β is a vector including all β_j coefficients.

- $\mathbf{X}_i$ vector of predictors for observation i.
- y_i is the response variable for observation i.
- λ is the regularization parameter.

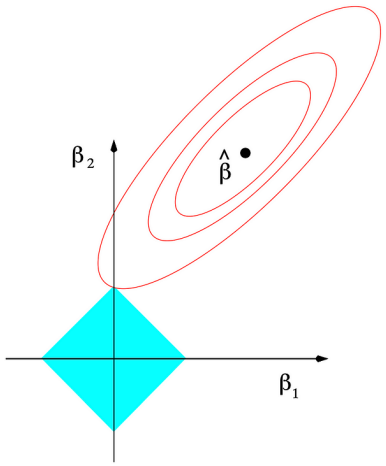

Figure 2.1: Lasso estimation illustration depicting contours of both the error and constraint functions. The solid blue regions denote the constraint areas, characterized by the conditions $|\beta_1| + |\beta_2| \leq s$. Concurrently, the red ellipses outline the contours of the Residual Sum of Squares (the figure is taken from 'Introduction to Statistical Learning (ISLR) chapter 6.2.2 page 222) [5].

Lasso employs an $l1$ **penalty term**, causing the shrinkage of coefficient estimates towards zero. Specifically, when the tuning parameter λ is sufficiently large, the $l1$ penalty of the lasso compels certain coefficient estimates to become precisely zero. The lasso contributes to variable selection, thereby improving the interpretability of the models it produces [5].

R DOCUMENTATION:

The **glmnet** package is a widely used tool for implementing Lasso regularization in the context of generalized linear models (GLMs). Developed by Friedman, Hastie, and Tibshirani (2010), this package offers a powerful and flexible approach to model fitting, particularly in scenarios with high-dimensional data and a need for variable selection. In the implementation of our model, we employed the **cv.glmnet(alpha=1)** function for conducting cross-validated Lasso regularization. This function, part of the **glmnet** package, allowed us to systematically select the optimal regularization parameter through cross-validation, enhancing the model's predictive performance.

2.2 Adaptive Lasso

In regression analysis, the Adaptive Lasso (ADLASSO) emerges as a refined regularization technique, building upon the foundations of the classical Lasso method. Designed to tackle limitations in variable selection and coefficient estimation, the Adaptive Lasso proves particularly valuable in scenarios where **predictors vary in importance.**

In the ADLASSO, the regularization parameter isn't fixed in a single mode. It changes **dynamically** based on the initial model's estimated coefficients. Unlike the fixed penalty in traditional Lasso, ADLASSO uses weights from those early coefficients. This flexibility allows the method to be choosy, giving a stronger push to specific coefficients, especially those **linked to more relevant variables**.

The ADLASSO minimizes the following objective function:

$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} |y_i - \mathbf{X}_i^\top \boldsymbol{\beta}|^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\} \tag{2.2}$$

- n is the number of observations.
- p is the number of predictors.
- β_j represents the regression coefficients.
- β is a vector including all β_j coefficients.
- $\mathbf{X}_i$ vector of predictors for observation i.
- y_i is the response variable for observation i.
- λ is the regularization parameter.
- w_j is the adaptive weight associated with predictor j.

The adaptive weights w_j are derived from the inverse of the absolute values of the preliminary coefficient estimates, $w_j = 1/|\hat{\beta}_j|$, where $\hat{\beta}_j$ is the estimate obtained from an **initial model**. The initial model typically involves fitting a regression model without any regularization, often using ordinary least squares regression.

The ADLASSO applies different penalties to each variable based on their predictive importance. This adaptability helps the model handle cases where some predictors are more important than others, improving the **accuracy of variable selection** and making the model easier to understand. This flexibility is particularly useful when predictors vary in importance and contribute differently to the model's predictive performance.

R DOCUMENTATION: In implementing Adaptive Lasso, we employed the **glmnet** package. The process involved fitting a Ridge regression model through cross-validation using **cv.glmnet(alpha=0)**, extracting coefficients at the optimal regularization parameter and computing weights for subsequent use in ADLASSO regularization.

2.3 LAD Lasso

The LAD Lasso (LADL), a fusion of the **Least Absolute Deviations** (LAD) regression and the **Lasso** method, stands as a powerful technique in statistical modeling. Unlike conventional regression techniques that prioritize minimizing the **sum of squared residuals**, LAD regression distinguishes itself by minimizing the **sum of absolute deviations.**

The LADL optimization problem is expressed as :

$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left| y_i - \mathbf{X}_i^\top \boldsymbol{\beta} \right| + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.3}$$

- n is the number of observations.
- p is the number of predictors.
- β_j represents the regression coefficients.
- β is a vector including all β_j coefficients.
- $\mathbf{X}_i$ vector of predictors for observation i.
- y_i the response variable for observation i.
- λ the regularization parameter, controlling the strength of the penalty term.

The objective function includes two key components: the sum of absolute deviations, reflecting the **robustness** of LAD regression to **outliers** and an $L1$ penalty term, controlled by the regularization (tuning) parameter λ, which induces sparsity in the model by encouraging certain coefficients to be precisely zero. In a practical context, the LADL balances the need for fitting the model to the observed data while promoting **simplicity** and **interpretability** through variable selection. The flexibility provided by λ enables researchers to adjust the model emphasizing either a closer fit to the data or a sparser representation. This adaptability is particularly valuable in scenarios where **noisy** or **high-dimensional** datasets are encountered. LADL addresses challenges associated with complex datasets, making it suitable for a range of applications in statistical modeling.

R DOCUMENTATION:

LAD (quantile) regression has gained popularity in recent years as a statistical method that extends traditional regression approaches. There are many R packages for penalized quantile regression like **rqPen**, **hqreg** and **quantreg**. In our study, we used the **rqPen** for LAD Lasso. The process begins with cross-validated quantile regression using **rq.pen.cv(penalty = "LASSO")** to find the best Lasso regularization parameter. Then, this optimal parameter is utilized to fit a quantile regression model using **rq.pen(penalty = "LASSO")**.

2.4 Adaptive LAD Lasso

The Adaptive LAD Lasso (ADLADL) stands out as a distinctive approach applied in regression analysis. It improves upon the **conventional LAD Lasso**, when we focus on sparse coefficients vectors, by introducing flexibility into the processes of variable selection and coefficient estimation. Differing from fixed penalty terms, this method dynamically adapts its regularization parameter, leveraging preliminary coefficient estimates derived from an initial model as a basis for adjustment.

The objective function for the ADLADL is given by:

$$\underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} |y_i - \mathbf{X}_i^\top \boldsymbol{\beta}| + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\} \tag{2.4}$$

- n is the number of observations.
- p is the number of predictors.
- β_j represents the regression coefficients.
- β is a vector including all β_j coefficients.
- $\mathbf{X}_i$ vector of predictors for observation i.
- y_i the response variable for observation i.
- λ is the regularization parameter, controlling the strength of the penalty term.
- w_j is the adaptive weight associated with predictor j.

The optimization process involves finding the values of $\boldsymbol{\beta}$ that minimize this combined objective function, striking a balance between fitting the model to the data and penalizing coefficients based on their adaptive weights. This adaptive approach enhances the method's capability to emphasize more **relevant predictors** in the modeling process.

R DOCUMENTATION:

For the implementation of Adaptive LAD Lasso, the package **rqPen** was also used. Initially, a Ridge regression model is fitted through cross-validated quantile regression **rq.pen.cv()**. This allows us to experiment with different regularization parameters while developing the model. Afterward, the coefficients associated with the optimal regularization parameter are extracted from the cross-validation results. Weights are computed for the Ridge coefficients, and an additional step is taken to handle potential infinite values in these weights. In such cases, they are substituted with a substantial finite value 10^{10}. Finally, the ADLADL model is fitted using cross-validated quantile regression **rq.pen.cv(penalty.factor = abs(w_ridge))**.

Chapter 3

Subsampling and Bootstrap methods

In this chapter, we'll introduce three essential statistical methods: **Bootstrap and Subsampling**. These tools are crucial in data analysis, offering **robust methodologies** to handle various statistical problems. Following this chapter, we'll demonstrate their utility in addressing **real-world scenarios**.

3.1 Bootstrap Sampling

The Bootstrap method has emerged as a fundamental technique in modern statistics. Essentially, it generates supplementary (pseudo) data derived from the original dataset, aiming to faithfully replicate the **true characteristics** of the sample while **substituting unknown model features** with estimates derived from the sample itself. Serving primarily as a tool for data analysis, Bootstrap facilitates the **evaluation** of statistical methods by repeatedly applying them to Bootstrap pseudo data or "**resamples**." This iterative process provides valuable insights into the **performance of statistical procedures**. As a result, data analytic approaches like Bootstrap offer distinct advantages in the **evaluation of statistical methods**[6].

Here's a more detailed analysis:

Let's say we have a dataset $X = \{x_1, x_2, \ldots, x_n\}$ consisting of n observations. We want to estimate a parameter, let's call it θ, based on this dataset.

The Bootstrap procedure involves the following steps:

1. **Sampling with Replacement:**
   We randomly draw n observations from the dataset X, with **replacement**. This means that each observation in the original dataset has an equal chance of being selected in each draw, and it's possible for the same observation to be selected **multiple times**.

2. **Parameter Estimation:**
   For each bootstrap sample, we calculate the parameter of interest, denoted as $\hat{\theta}_i$, using some estimation method (e.g. mean, median, regression coefficients, etc.). This gives us a collection of Bootstrap estimates $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_B$, where B is the number of bootstrap samples.

R DOCUMENTATION:

For constructing the Bootstrap samples ($B = 200$) the function **sample(replace=TRUE)** was used and then these samples were applied in each model that were presented before (LASSO , ADLASSO , LADL and ADLADL).

3.2 Subsampling

Subsampling is a powerful technique used in statistical analysis for assessing the **performance of a model**. Instead of analyzing the entire dataset, Subsampling involves repeatedly drawing random samples (subsamples) from the original dataset and performing analyses on these smaller samples. In this study, whenever we mention Subsampling, it **consistently refers to the method without replacement**.

Let's provide a more detailed analysis:

Let's consider a dataset $X = \{x_1, x_2, \ldots, x_n\}$, of n observations. Our objective is to estimate a parameter θ based on this dataset.

Here's how the Subsampling without replacement works:

1. **Sampling without Replacement:**

2. We randomly select m observations from the dataset X, where m is a proportion of n, without replacement. This means that each observation in the original dataset can be chosen **only once** in each draw, ensuring that each Subsample is **unique**.

3. **Parameter Estimation:**

   For each Subsample drawn without **replacement**, we compute the parameter of interest, denoted as $\hat{\theta}_i$, utilizing an appropriate estimation technique (e.g. mean, median, regression coefficients, etc.). This process yields a set of estimates $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_B$, where B denotes the number of Subsamples.

In our implementation, we adopted a similar resampling strategy to that used in the study conducted by De Bin et al.(2016). Specifically, they set the size of the Subsample (m) to approximately 63.2% of the total number of observations (n). This percentage is determined by the formula $m = \lfloor 0.632n \rfloor$ [2].

R DOCUMENTATION:

For constructing the Subsamples ($B = 200$) the function **sample(size = floor(0.632 * n), replace = FALSE)** was used and then these samples were applied in each of the four models.

Chapter 4

Statistical Validation Techniques

Before proceeding with the application, we examine fundamental statistical validation techniques, focusing on **Confidence Intervals** and **K-Fold Cross Validation**. These methodologies are essential in assessing the **reliability** and **robustness** of statistical models.

4.1 Confidence Intervals

Confidence intervals (C.I) are an essential tool in statistical analysis by providing essential insights into the **precision** and **reliability** of estimates derived from sample data. They offer a comprehensive understanding of the **uncertainty inherent in statistical inference**. Additionally, they assist in comparing different estimates and evaluating the **practical significance** of discoveries with greater ease. Overall, C.I offer reliable information and aid us in making critical decisions for our ongoing study.

To construct a C.I, we require both an upper and a lower bound that defines the potential range of values for the estimated parameter. This interval is determined based on available data and a specified **confidence level**, denoted as γ; this level indicates how confident we want to be that the **true value of the parameter falls within the interval**. Typically, γ falls between **0.9** and **0.95**. As the confidence level increases, the width of the interval also expands, thus enhancing the **likelihood of including the parameter**. However, a wider interval may not be preferred, as it introduces greater **uncertainty** regarding the parameter. Therefore, the preferred C.I. remains **narrow while maintaining a high level of confidence**.

When interpreting a confidence interval (A,B) at a confidence level γ for a parameter θ, consider the sample we observed as just one among many possible

samples we could have obtained. Each of these hypothetical samples would yield its own calculated confidence interval. We would anticipate that $100 \times \gamma$ of these intervals would **include the true value** θ. However, even if we obtained numerous such observed intervals, it remains **impossible** to determine definitively which intervals contain the true value and which do not [3].

In the context of **Hypothesis Testing**, we can modify the problem to incorporate the use of **confidence intervals**. Suppose that we aim to estimate a parameter θ against a specified value θ_0. We seek to test the following hypothesis at a **significance level** of α:

$$\begin{aligned} H_0 &: \theta = \theta_0 \\ H_1 &: \theta \neq \theta_0 \end{aligned} \tag{4.1}$$

Where:

- H_O : null hypothesis
- H_1 : alternative hypothesis

Afterward, we can calculate the confidence interval at a confidence level of $\gamma = 1 - \alpha$ and **reject the null hypothesis if θ_0 is not within the interval**.

In our study, we will approach to **evaluate the coefficients** obtained from a **Linear Regression Model** using the following Hypothesis Testing:

$$\begin{aligned} H_0 &: Coef_i = 0 \\ H_1 &: Coef_i \neq 0 \end{aligned} \tag{4.2}$$

Where:

- $Coef_i$: The predicted coefficient for the i_{th} predictor.

This modification operates as follows: if the i_{th} predictor (X_i) is considered **insignificant** for our model, then the coefficient ($Coef_i$) **equals 0**. Rejecting the null hypothesis is equivalent to concluding that X_i is considered **significant**. To reject H_0 we can construct a confidence interval for $Coef_i$ and examine whether **0 is within the interval**.

4.2 Bootstrap Confidence Intervals

One of the primary objectives of **Bootstrap Theory** is to automatically generate "good" C.I. The term "good" in this context refers to Bootstrap Intervals that closely resemble exact C.I in situations where **traditional statistical theory provides precise solutions**. Additionally, these Bootstrap intervals

should consistently yield **accurate coverage probabilities across all scenarios** [4].

The general procedure for constructing bootstrap confidence intervals involves fitting linear regression models using both lm R function (for $L2$ methods) and LAD R function (for $L1$ methods) on each bootstrap sample. This process yields coefficients for each β_j. To ensure symmetry and equal coverage of the confidence level γ on both sides of the distribution, we choose the $(1-\gamma/2) \times 100$th and $(1+\gamma/2) \times 100$th percentiles as the lower and upper bounds of the confidence interval, respectively.

Now, let's take a closer look at the analytical steps for building Bootstrap Confidence Intervals:

Let $X = \{x_1, x_2, ..., x_n\}$ denote the original dataset of n observations.

- **Resample with Replacement:** Generate B bootstrap samples denoted as $X_1^*, X_2^*, ..., X_B^*$ by randomly sampling n observations with replacement from X.
- **Compute Statistic:** For each bootstrap sample X_i^*, calculate the desired statistic θ_i^*, where θ represents the statistic of interest (e.g., mean, median, coefficients regression).
- **Determine Confidence Interval:** Choose a confidence level γ (e.g., 0.90 for a 90% confidence interval) and calculate the corresponding quantiles of the bootstrap distribution. Let $\theta^*_{(1-\gamma/2)}$ and $\theta^*_{(\gamma/2)}$ represent the $(1-\gamma/2)$ and $(\gamma/2)$ quantiles, respectively.

The confidence interval for the statistic θ is given by $(\theta^*_{(\gamma/2)}, \theta^*_{(1-\gamma/2)})$.

R DOCUMENTATION:

For constructing the C.I the fuction **apply(.. , quantile(x, c(0.05, 0.95)))** and **confint(.. , level = 0.90)** was used. The first one is for **stage 2** of our application and the second one is for **comparing previous results**. We'll provide a more detailed explanation in the following chapter.

4.3 K-Fold Cross Validation

The prediction error is a metric that measures **how accurately a model predicts future observation responses**. It's commonly used in model selection to favor models with the lowest prediction error among options. Cross-validation is a common approach for estimating prediction error [4]. Specifically, K-fold cross-validation (K-Fold C.V) is a robust technique used to **assess the performance and generalization ability of predictive models**. By dividing the dataset into multiple subsets and repeatedly training and testing the model. K-Fold C.V provides valuable insights into model reliability and helps in selecting the **best-performing algorithm**.

To perform K-Fold C.V, we follow these steps:

1. Split the dataset randomly into approximately K equal-sized subsets.
2. For each subset:
    - Use it as the test set.
    - Use the remaining K-1 subsets as the training set.
    - Train the model on the training set.
    - Evaluate the model on the test set and compute the performance metric (rmse, mae, etc.).
3. Repeat step 2 for each subset.
4. Compute the average performance metric across all K folds to obtain the final evaluation score.

The random separation of the dataset into training and testing sets introduces randomness, **reducing bias in model evaluation**. Evaluating model performance across various dataset splits, helps assess its ability to generalize to new data. Randomly shuffling the dataset ensures **fair distribution of samples**, preventing any systematic patterns or biases in evaluation.

Chapter 5

Application

In this chapter, we will adopt a similar method to the one employed by Anika Buchholz, Norbert Holländer, and Willi Sauerbrei in their 2008 study, where they utilized a two-step bootstrap technique for model averaging within a simulation study of linear regression models [1]. However, instead of employing **bootstrap-based filtering in the initial step** to reduce the number of predictors, we will utilize **subsampling**, as recommended by De Bin et al. [2], due to their findings suggesting that subsampling is more effective at retaining the most important predictors linked to the case. In the subsequent step, we will employ a **two-step bootstrap model averaging** approach similar to the one described previously, specifically using the model averaging step with only those variables meeting the specified condition (0.3). In this collection of predictors, we utilize the **lm** R function in the second step when LASSO and ADLASSO were utilized in the first stage. Similarly, the LAD R function is used in the second stage when LADL and ADLADL were implemented in stage 1. We employ bootstrap sampling to generate multiple samples, from which we calculate the **average coefficients**. Moreover, we construct **confidence intervals** using these coefficients to assess the **robustness** of the predictors that have been retained and to evaluate the effectiveness of **model averaging**. Finally, we compare these outcomes with those obtained when applying the same procedures to the predictors **without bootstrap sampling**.

5.1 First Stage

Our dataset originates from **Department of Respiratory Medicine, School of Medicine, University of Crete, Heraklion, Greece**. To ensure confidentiality, variables have been assigned coded names. The dataset comprises an X matrix of predictors, measuring 657 by 33 dimensions, and a Y vector representing the response variable, with dimensions 657 by 1. The response variable, denoted as Y, indicates the **impact of a therapy**, either positive or negative, on the patients' health. In our study, first, our objective is to determine the

predictors that significantly impact the overall health outcomes of patients and second, we seek to discover suitable models capable of predicting, with new data, the probability of a particular treatment yielding either positive or negative effects on the patient, depending on the values taken by their predictors. First, let's conduct a detailed analysis of the data.

Variable	**Mean**	**Median**	**Additional Information**
X12	10.67	11.00	Symmetrical distribution
X35.27	35.53	34.42	Rightward skew
X0	0.1553	-	Mostly 0-valued binary
X31.5	35.36	37.10	Rightward skew
X0.1	0.2344	-	Mixture of 0s and 1s
X0.2	0.9422	-	Mostly 1-valued binary
X1	0.3303	-	Skewed towards 0
X0.3	0.3501	-	Mixture of 0s and 1s
X0.4	10.88	9.00	Right-skewed distribution
X387	416.7	420.0	Wide range of values
X50	61.76	65.00	Slightly left-skewed
X194	258.2	265.0	Higher mean, potential shift
X129	118.5	108.0	Right skewness
X67	50.94	41.00	Right-skewed distribution
X348	0.9422	-	Mostly 1-valued binary
X88	90.75	91.20	Symmetrical distribution
X0	7.515	6.800	Right-skewed distribution
X11	9.297	8.800	Rightward skew

Table 5.1: Summary Statistics for Predictors

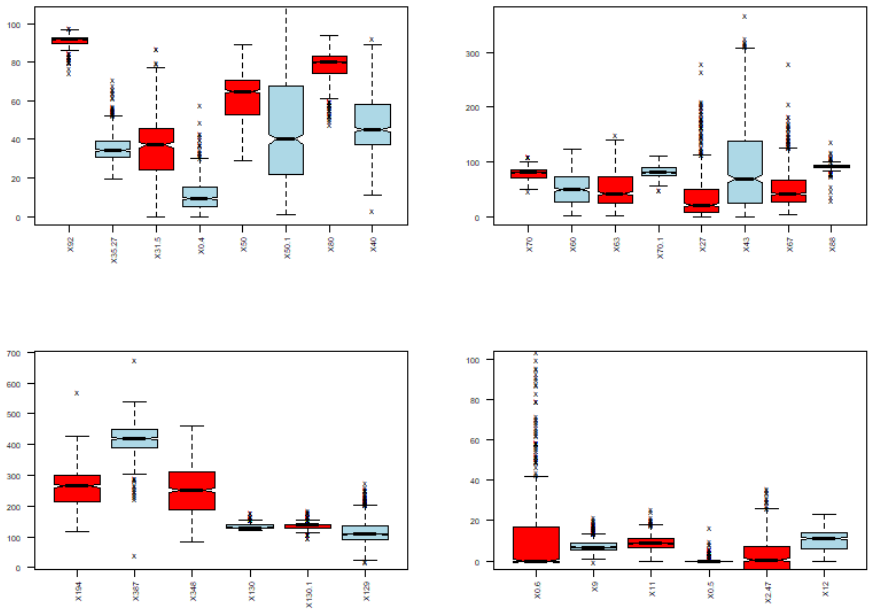

Figure 5.1: Boxplots for each predictor

Upon observing the boxplots for each predictor, clear patterns emerge. For instance, predictors such as $X12$ and $X88$ exhibit **symmetrical distributions**, with relatively balanced spreads of data around their respective means and medians. On the other hand, variables like $X35.27$ and $X11$ display noticeable **rightward skewness**, indicating a clustering of values towards the lower end of the distribution with a few higher **outliers**. Additionally, variables like $X194$ and $X129$ demonstrate a higher mean compared to their respective medians, hinting at potential **outliers or shifts** in the dataset towards higher values. At last variables like $X50$ and $X9$ display **slight skewness**, although in opposite directions, with $X50$ showing a **leftward skew** and $X9$ exhibiting a **rightward skew**. We omit the binary predictors ($X0$, $X0.1$, $X0.2$, $X1$, $X0.3$) from the Boxplots as they do not offer significant insights in this context.

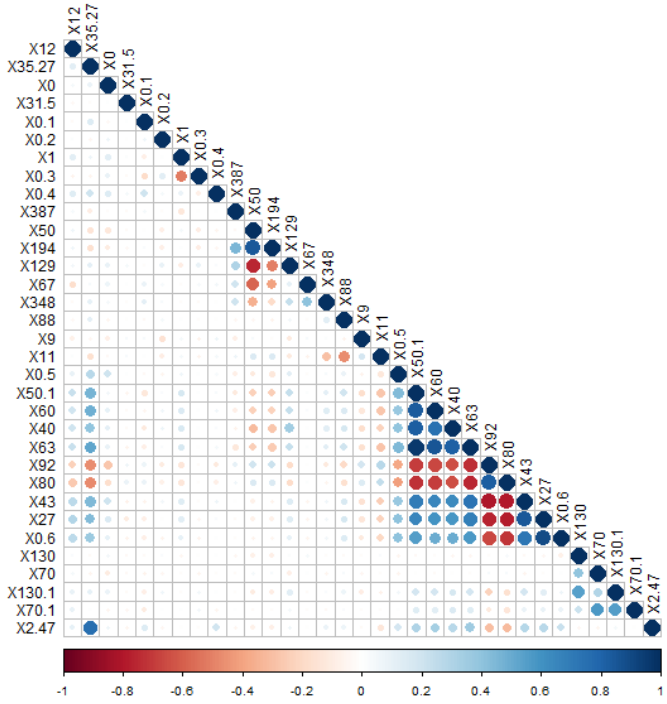

Figure 5.2: Correlation matrix plot

In the correlation matrix analysis conducted, a color bar was employed to visually represent the strength and direction of correlations between variables. The color bar provided a clear indication of the degree of correlation, ranging from deep red for **strong positive correlations** to deep blue for **strong negative correlations**, with shades of green representing **weaker correlations**. Notable results from the analysis revealed several significant relationships between variables. For instance, $X63$ and $X60$ exhibited an exceptionally **high positive correlation** of **0.854**, suggesting a strong association between them. Additionally, a strong **negative correlation** of **-0.515** was observed between $X1$ and $X0.3$, indicating an inverse relationship. Furthermore, moderate **positive correlations**, such as that between $X50.1$ and $X0.5$ (0.439), and **negative correlation**, such as between $X11$ and $X0.2$ (-0.579), were also identified. Notably, certain pairs of variables, such as $X31.5$ and $X0.5$, showed a low correlation of 0.002, suggesting relative **independence** from each other. These findings provide valuable insights into the **interrelationships** among predictors in the dataset. Upon noticing strong correlations among specific predictor variables, we speculated the potential existence of **multicollinearity**.

Multicollinearity:

In multiple linear regression, the occurrence of multicollinearity manifests as a phenomenon where two or more variables exhibit a discernible correlation. The strong correlation between these variables makes it difficult to separate their **individual effects on the response variable**. Due to the presence of this phenomenon, complications arise in the estimation of model coefficients, increasing the probability of encountering standard errors. To address this concern, we employed a commonly utilized method known as **VIF**.

Variance Inflation Factor(VIF):

In regression analysis, a paramount instrument for addressing **multicollinearity** among predictors is the Variance Inflation Factor (VIF). The VIF provides a quantitative assessment, offering insight into the extent to which the variance of an estimated regression coefficient is magnified as a result of correlation with other predictors.

For the computation of VIF we use this formula:

$$\mathrm{VIF}(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all of the other predictors. The minimum attainable value for the VIF is 1, indicating the absence of collinearity. However, this theoretical scenario is seldom realized in practical applications, as a certain degree of **collinearity** commonly exists among predictors. Empirically, when the VIF surpasses **thresholds** of 5 or 10, it signifies a notable degree of **collinearity**, prompting consideration for the potential removal of the associated predictors from the dataset [5].

Results from implementing VIF in dataset:

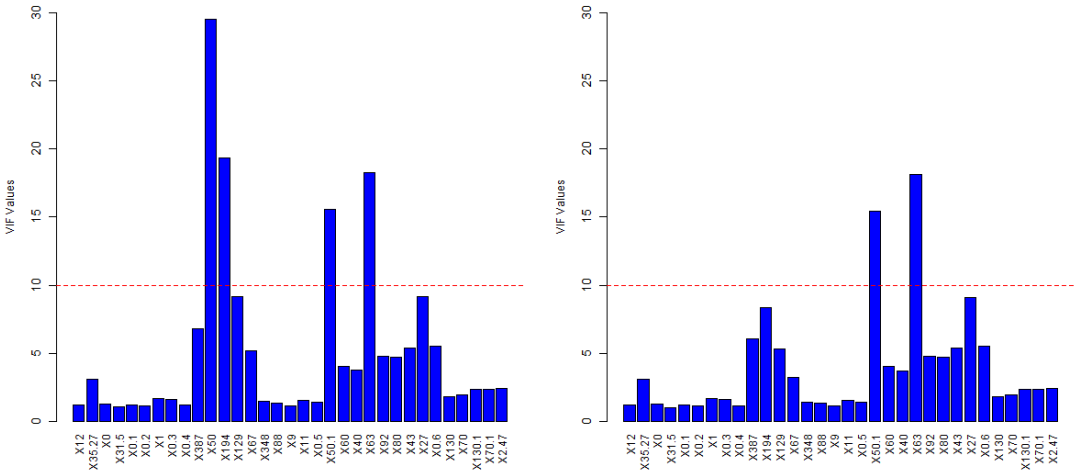

Figure 5.3: VIF before dropping $X50$ and after.

As evident from the two plots displayed above, the initial implementation of the VIF analysis led to the **exclusion** of predictor $X50$ due to the VIF value (**29.50819**) exceeding the empirically determined threshold of 10. In the second step of the VIF analysis, the predictor $X63$ was **excluded** with a VIF value of (**18.16223**).

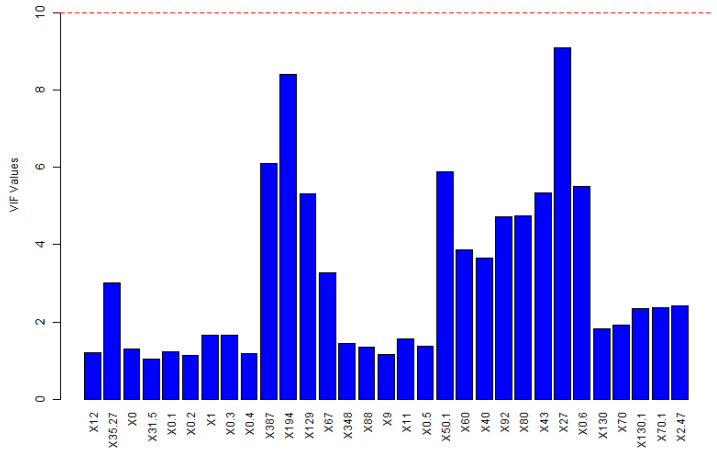

Figure 5.4: VIF after dropping both $X50$ and $X63$

Upon their removal, there was a **notable shift** in the VIF values of the remaining predictors, with **none surpassing** the threshold of 10. Consequently, the set of remaining predictors exhibited a **diminished level of multicollinearity**, indicating a potential improvement in the **accuracy** of the results obtained.

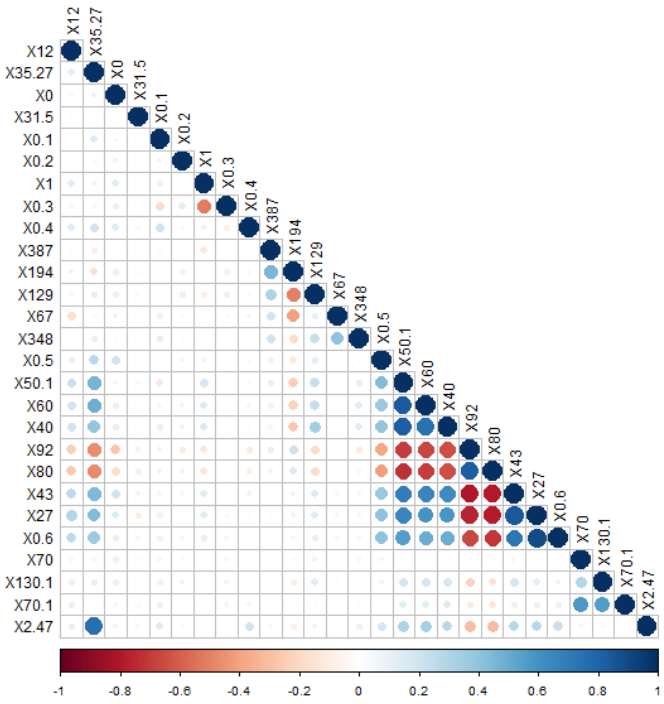

Figure 5.5: Correlation plot after VIF-filtering

Following the removal of problematic predictors from the dataset, we examine the correlation plot once more. The slight differences between figures 5.2 and 5.5 arise from disparities in their underlying principles. While correlation primarily detects linear relationships, VIF-based multicollinearity detection encompasses both linear and non-linear relationships. Correlation matrices might fail to capture more complex, especially non-linear, relationships, whereas VIF considers the collective effect of all other variables on each variable within the model.

Now, let's examine the results obtained by applying the full dataset to each method:

Predictors	**LASSO**	**ADLASSO**	**LADL**	**ADLADL**
$X12$	0.7305627	0.220002873	0.737490063	0.36782417
$X0$	-0.05098747	-0.023379902	-0.014497831	-
$X31.5$	0.05222455	-	0.061435745	-
$X0.1$	-0.003068241	-	-	-
$X0.2$	0.03864692	-	0.011394332	-
$X0.4$	-0.006619008	0.005079894	-	-
$X387$	0.0944155	-	0.006722837	-
$X67$	-0.02480468	-0.023622225	-0.022166338	-
$X11$	0.004339014	-0.008948004	-	-
$X0.5$	-0.02709775	-0.014928842	-0.046919577	-
$X60$	-0.006476320	-0.020002334	-	-
$X80$	0.04260858	-	0.072686315	-
$X43$	-0.03204111	-0.049334903	-	-0.02828112
$X130$	-0.06014917	-0.034415848	-0.118622562	-
$X70$	0.02655357	0.002497855	-	-
$X130.1$	-	0.001328634	-	-
$X70.1$	0.001918048	-	0.038562626	-
$X2.47$	0.004721250	-	-	-
$X35.27$	-	-	-0.005242465	-
$X194$	-	-0.007556494	-	-
$X88$	-	-0.018336876	-	-
$X0.6$	-0.09565026	-0.015530939	-0.094500711	-
$X27$	-	-0.028421890	-0.125210443	-0.05903720
$X40$	-	0.022898328	-	-
$X9$	-	-0.016837986	-	-
$X0.3$	-	-	-	-
$X129$	-	-	-	-
$X348$	-	0.003368764	-	-
$X50.1$	-	-	-	-
$X92$	-	-0.011500719	-	-
$X1$	-	-	-	-

Table 5.2: Coefficients through all methods

The results depicted in the table above illustrate **varying degrees of stringency** among different regularization methods applied to the filtered dataset. Lasso, known for its relatively permissive treatment of predictors, yields a **considerable** a number of non-zero coefficient values across the predictors. However, when the same dataset is subjected to ADLASSO, even though its characterized by it's heightened selectivity due to the incorporation of a weight parameter, the number of non-zero coefficients **increases by one**. Moving forward to LADL, which employs $L1$ regularization, the number of non-zero coefficients **decreases notably**, reflecting its more stringent approach in shrinking coefficients towards zero. Finally, ADLADL, combining both $L1$ regularization and the weight parameter, exhibits the most stringent criterion, resulting in the **smallest subset of predictors surviving** the regularization process.

The variables $X1$, $X0.3$, $X129$ and $X50.1$, were **not included** in any of the four models. Therefore, these variables may be considered the **least significant within our study**.

Interestingly, only one predictor $X12$ emerges as consistently **significant across all four methods**. This predictor is considered crucial for our study because it highlights its strength and significance in enhancing the predictive capability of the model.

In essence, the utilization of advanced regularization methods such as **Lasso variants** not only aids in **dimensionality reduction** but also facilitates the identification of key predictors essential for the model's performance. By prioritizing these influential variables, we can streamline our analysis and gain deeper insights into the underlying relationships within the dataset.

Choosing appropriate threshold:

Before proceeding to the next stage, it's crucial to clarify the criterion by which we will determine, **which predictors are considered significant and which are not**. In the study conducted by De Bin et al. (2016), it was found that the significance of predictors in **multivariable regression models** is determined by a threshold dependent on the balance between **relevance and stability** in the model. If there are numerous relevant variables and **consistent inclusion** is desired, a higher threshold (e.g., above 0.5) may be set. Conversely, if **variability** is expected due to correlation or weaker effects, a lower threshold (e.g., between 0.2 and 0.6) may be more appropriate[2]. Taking into account the characteristics of our dataset, it would be prudent to begin with a threshold set around **0.3**, given the presence of moderate to strong correlations identified in the correlation matrix analysis. This approach was chosen based on the findings and methodology outlined in their paper, which compared the effectiveness of **subsampling and bootstrapping in resampling-based model selection**.

5.2 Second Stage

After conducting a comprehensive exploratory analysis of our dataset, we have refined our set of predictors to remove any **problematic ones**. Now, our focus shifts to analyzing the results of applying our method to the data filtered during stage 1. Our goal is to **evaluate how well our approach performs on this refined dataset**. The results are depicted in figures 5.6 and 5.7.

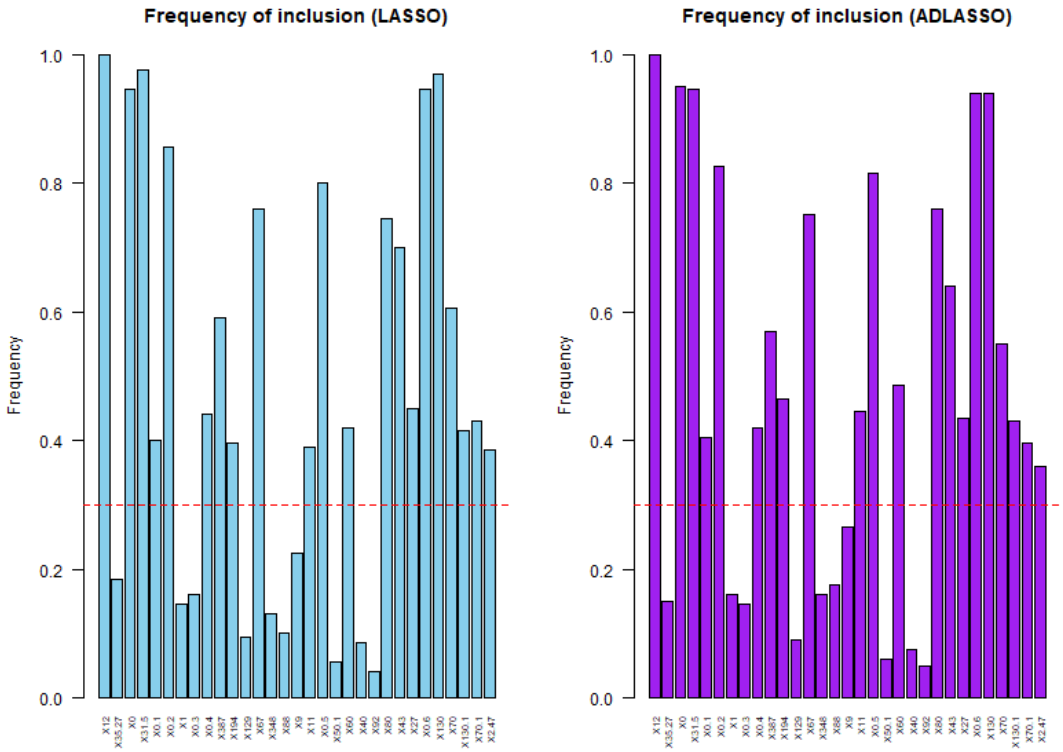

Figure 5.6: Frequency of inclusion in L2 methods

From the observations mentioned above, it is evident that even when employing solely the **LASSO** method without any additional penalization, numerous predictors are **excluded**. The remaining predictors deemed significant are as follows: $X12$, $X0$, $X31.5$, $X0.1$, $X0.2$, $X0.4$, $X387$, $X194$, $X67$, $X11$, $X0.5$, $X60$, $X80$, $X43$, $X27$, $X0.6$, $X130$, $X70$, $X130.1$, $X70.1$, and $X2.47$. Generally, the frequency of inclusion for each of the survived predictors **exceeds 0.3**. When employing the **ADLASSO** method, it is noteworthy that not only the **same number of predictors have survived** but also the **same predictors**, compared to the LASSO method. However, there exist slight **variances** in the

frequency of inclusion for each predictor. Generally, in ADLASSO, these frequencies tend to be **lower** compared to LASSO. Overall, the fact that the same total number of predictors survived through LASSO and ADLASSO, suggests a **consistent level of model complexity across both methods**.

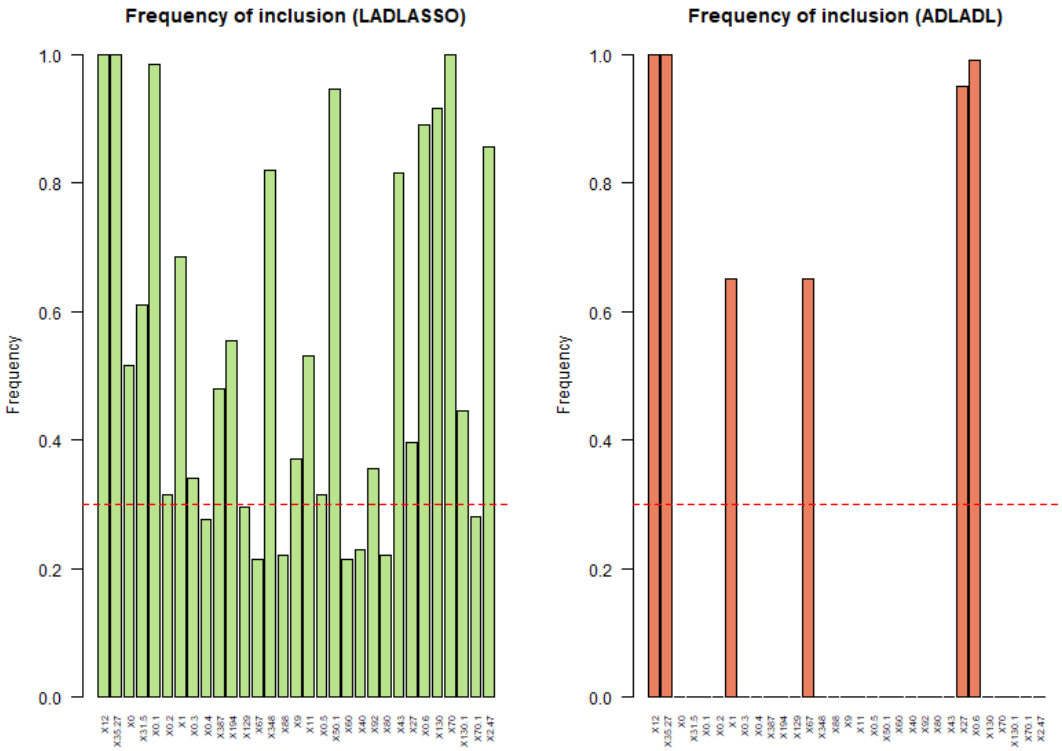

Figure 5.7: Frequency of inclusion in L1 methods

When the **LADL** method is utilized, the number of non-zero coefficients **are even more than before**. Notably, $X8.762886598$ is close to the threshold, with the value of 0.310. Generally, the frequencies of the remaining predictors are significantly higher than 0.3. There are 22 predictors considered significant, which are: $X12$, $X35.27$, $X0$, $X31.5$, $X0.1$, $X1$, $X0.3$, $X387$, $X194$, $X129$, $X348$, $X9$, $X11$, $X50.1$, $X92$, $X43$, $X27$, $X0.6$, $X130$, $X70$, $X130.1$, and $X2.47$. Upon examination in **ADLADL**, it's evident that **all of the predictors except from 6 have close to zero or exactly zero frequency of inclusion**. Among those with non-zero frequencies, four of them have a significant amount of inclusion close to 1. The lowest frequency stands at 0.65 for $X1$ and $X67$. The significant predictors are: $X12$, $X35.27$, $X1$, $X67$, $X27$, $X0.6$. This outcome was expected from this method due to its strict penalization of variables.

The examination of identifying significant predictors across different methodologies provides insight into the complex processes involved in **model selection** and **variable inclusion**. Initially, the LASSO method, without any additional penalization, showcases its ability to exclude numerous predictors. Remarkably, although the ADLASSO and LASSO techniques adopt different strategies, they both resulted in an **equal number of noteworthy predictors being identified**. However, there were discrepancies in the particular predictors singled out by each method. The LADL method didn't **reduce** the number of non-zero coefficients as expected. Finally, the utilization of the ADLADL method shows that the majority of predictors demonstrate **unremarkable inclusion frequencies**.

5.3 Model Performance

Following an extensive model analysis, our attention now turns to evaluating model performance. We aim to identify models that **demonstrate greater robustness and reliability in predicting the impact of therapy on patients' health**. To achieve this objective, we will utilize confidence intervals (CI) obtained through Bootstrap sampling and conventional methods, along with K-fold cross-validation (CV) as detailed in Chapter 4. Initially, we will present the confidence intervals obtained with and without Bootstrap sampling to compare the performance assessment. After that, we will showcase the results of a 10-fold CV for each model, using specific metrics: root mean squared error (RMSE) for L2 methods and mean absolute error (MAE) for L1 methods.

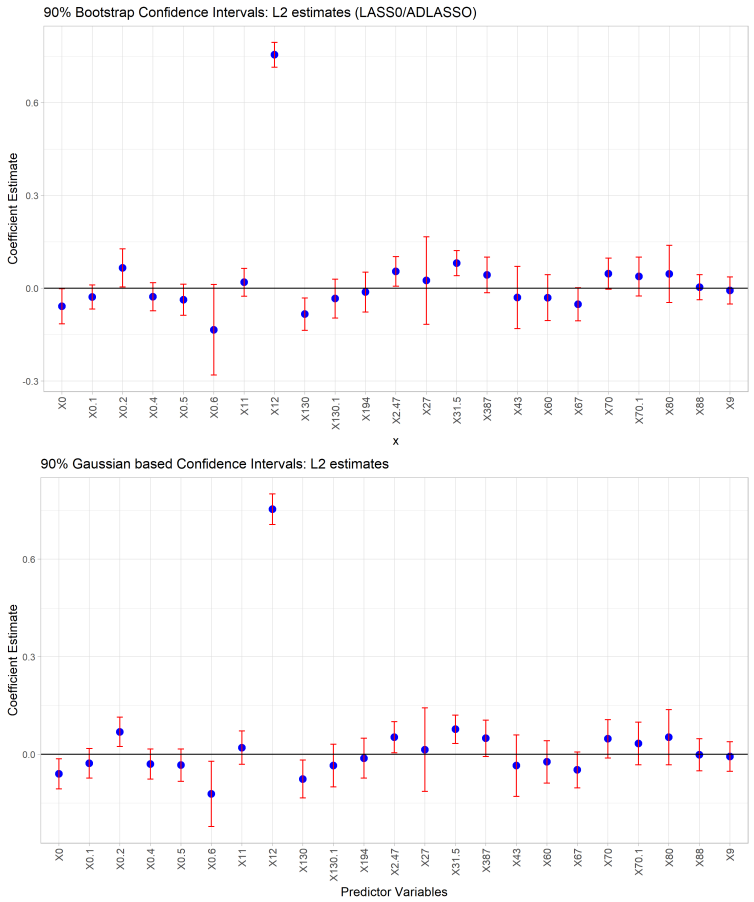

Figure 5.8: Bootsrap based C.I for least squares coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators.

As both the LASSO and ADLASSO methods yielded **identical predictor selections**, and the linear regression (lm) model was applied for evaluating **model selection** in both cases, the resulting C.I were also consistent. The plotted C.I indicates that the majority of intervals **include zero and are narrow**, except $X0.6$, $X43$, $X80$ and $X27$. Therefore, there is **limited evidence** to reject the null hypothesis for most predictors, except for $X0$, $X0.2$, $X0.6$, $X12$, $X2.47$, $X387$, $X130$, and $X31.5$, where **significance** is suggested. While for predictors like $X0$ and $X2.47$, the null hypothesis can be rejected with caution due to their close positioning above/below zero, $X12$ has a clearer difference from zero, ensuring its inclusion in the model. Moreover, $X0.4$, $X67$, $X70$, $X387$ and $X70$ hardly include zero, so we can reject the null hypothesis with caution too. Despite these findings, the overall **reliability of the results appears questionable**.

When solely employing the lm **without sampling** for LASSO and AD-LASSO, slight variations are observed in the width of the C.I (they are wider) and their proximity to zero. Some confidence intervals, such as those for variables $X130$, $X2.47$, $X387$, $X67$, and $X70$, are **either closer or include zero** when the lm model is applied without sampling. Again $X12$ notably remaining **distant from zero in both cases**.

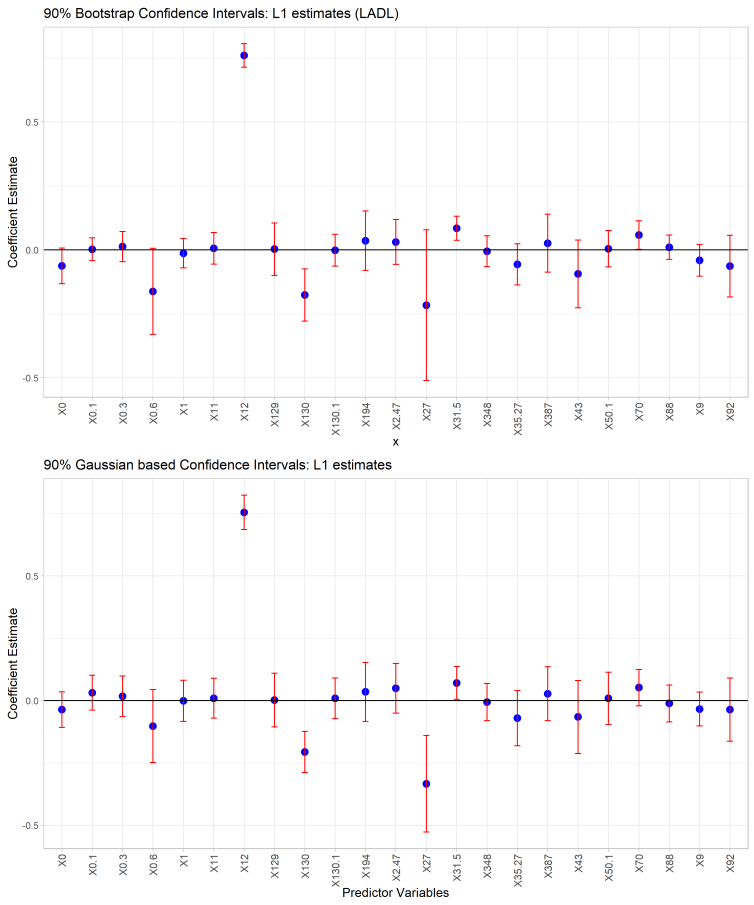

Figure 5.9: Bootsrap based C.I for least absolute deviation coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators.

All the C.I for predictors include zero, except for $X12$, 31.5, $X0.6$ and $X130$. However, we observe **wider C.I** for predictors $X0.6$ and 27 in this case. Notably, the interval for $X70$ barely includes zero. Once again, $X12$ stands out as its interval doesn't include zero and lies **significantly** above it, suggesting its significance and allowing us to **reject the null hypothesis**. We can also

reject the null hypothesis for $X0.6$ still, we should proceed with caution given its proximity to zero. For the remaining predictors, some of them have values of zero and wide C.I. Thus, we **cannot reject the null hypothesis**, indicating **insufficient evidence** to decide whether to include them in our model.

When comparing with the **conventional LAD model** (without sampling), we observe that the **almost the same C.I include zero** except from $X31.5$ and $X0.6$ which previously did not include zero. Most of them are remarkably **wider** than those obtained when using bootstrap. This indicates less precision in its estimates, as the **range of possible values** for the predictor estimates is greater than when using **Bootstrap samples**.

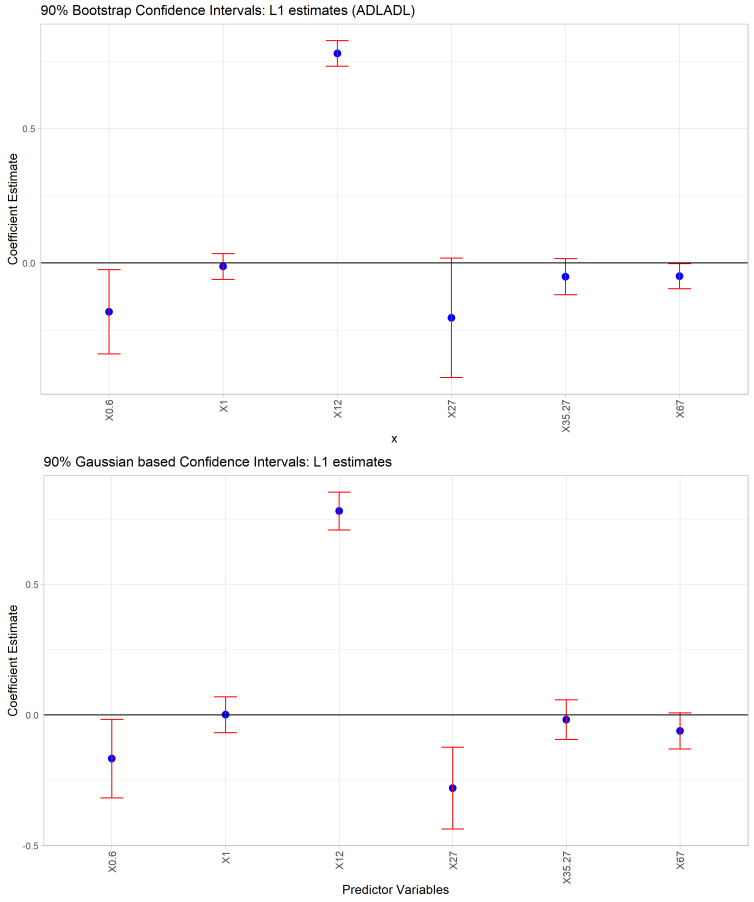

Figure 5.10: Bootsrap based C.I for least absolute deviation coefficients. Predictors were filtered via Subsampling coupled with LASSO type estimators.

In the last approach, it's evident that **two C.I barely include zero**. The

intervals are quite narrow except from $X0.6$ and $X27$, which indicates precision in the estimates. Despite $X35.27$ and $X27$ that **barely includes zero**, the null hypothesis should be **rejected with caution**. Notably, $X12$ stands out as significantly distant from zero, this underscores its importance and supports the decision to reject the null hypothesis and add it to our model. While these findings are important, it's essential to continue with caution when considering the **significance of predictors and deciding whether they should be included in the model**.

In this case, **notable differences** emerge when solely employing **LAD without sampling**. Primarily, the C.I are remarkably wider, indicating less precision in the estimates. Moreover, $X12$ and $X0.6$ are **less distant** from zero. However, the remaining C.I exhibit **minor changes compared to those obtained previously**.

Lets move forward to K-Fold C.V. The results from conducting K-Fold C.V are depicted in table 5.3 and figures 5.11, 5.12, 5.13 and 5.14.

First, we conducted a 10-Fold C.V and we repeated it 100 times, as we want **reduced variance for the estimated errors**. Simultaneously, we stored the results from each iteration and at the end, we averaged them using the median. For each model, we applied different relative accuracy. For $L2$ models we applied RMSE and for $L1$ models MAE. After all that, we generated Boxplots for further examination and to inspect the error distribution for each model.

Method	**RMSE**	**MAE**
LASSO/ADLASSO	0.719166	0.5784453
LADL	1.268047	1.018226
ADLADL	1.264769	1.014868

Table 5.3: RMSE and MAE for each method.

Because of the discrepancy in their values when depicted together, combining them doesn't offer a clearer visualization. Hence, figures 5.12 and 5.14 serve for more detailed examination.

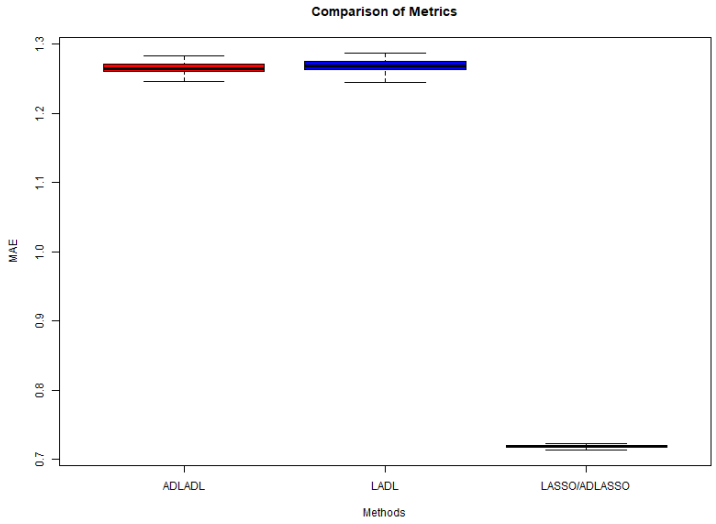

Figure 5.11: RMSE for each method.

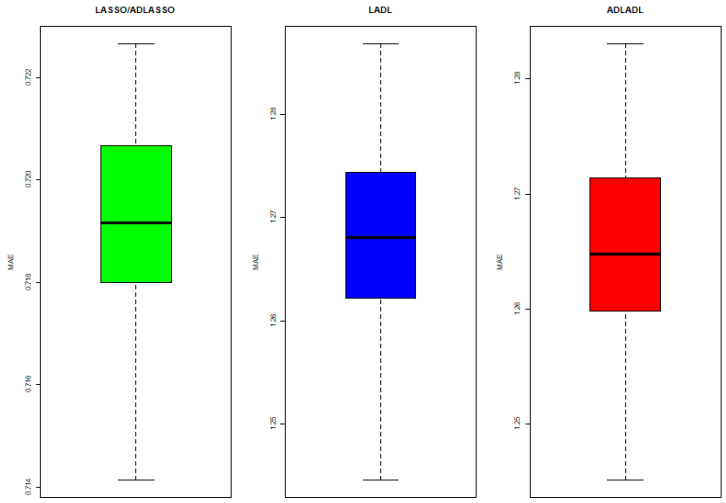

Figure 5.12: Closer look of RMSE for each method.

The boxplots illustrates the distribution of RMSE values across the 10 folds of cross-validation for the four models: LASSO, ADLASSO, LADL, and ADLADL. The width of the Interquartile Range ($IQR = Q3$ - $Q1$) and the length of the whiskers suggest that the spread of the RMSE values is relatively consistent across the 10 folds of cross-validation. This also indicates less variance in the estimated errors, as desired. As we see there are no outliers in the error distributions, suggesting that the RMSE values are within IQR, without extreme values skewing the distribution. Overall, the noticeable difference between LASSO/ADLASSO and the other methods indicates that LASSO and ADLASSO are performing better in this case.

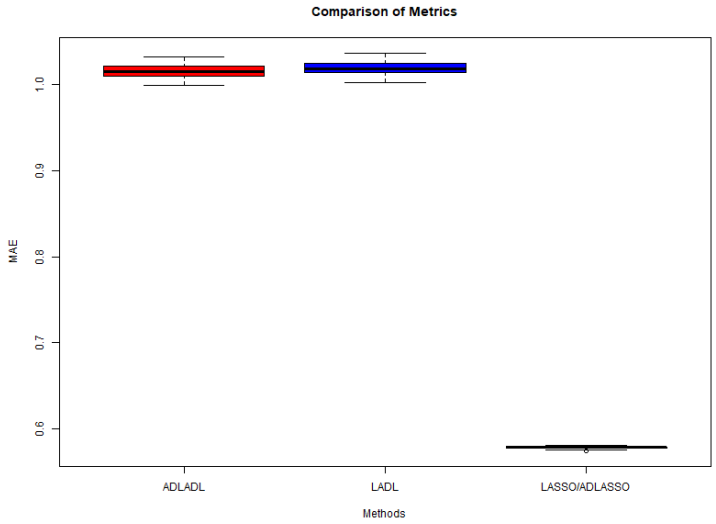

Figure 5.13: MAE for each method.

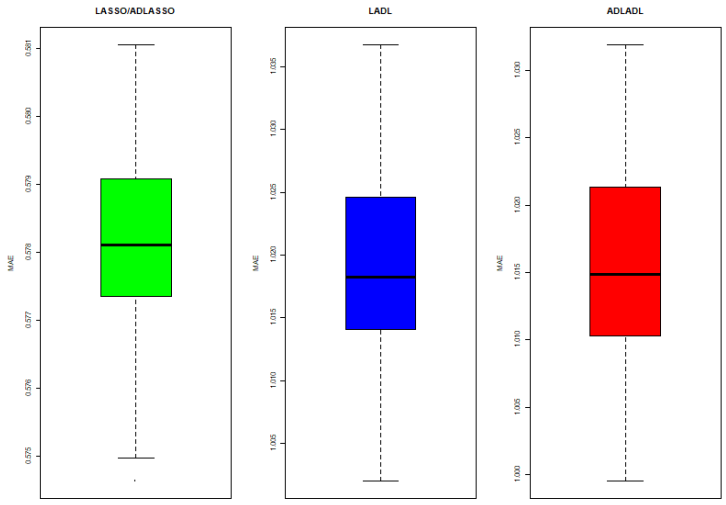

Figure 5.14: Closer look of MAE for each method.

The median MAE values for all the models are quite similar, as indicated by the horizontal lines inside the red boxes. The width of the IQR is similar across $L1$ models. As for LASSO/ADLASSO, the width is smaller than the rest. Once more, as evident, there are no outliers in the error distributions, suggesting that the MAE values are within IQR, without extreme values skewing the distribution. Again, its noticeable that the difference between LASSO/ADLASSO and the other methods indicates that LASSO and ADLASSO are performing better in this case.

Chapter 6

Conclusions

In this thesis, we aimed to construct linear predictive models using various regularization methods such as LASSO, ADLASSO, LADL, and ADLADL. Our objective was to develop a predictive model aimed at estimating the key factors that impact the overall health outcomes of patients receiving therapy. Through all that, we observed distinct patterns in predictor selection and model performance.

Initially, we conducted exploratory data analysis, including the creation of Boxplots to inspect the distribution of each predictor and the correlation matrix for predictors. The correlation matrix suggests that there is multicollinearity in the dataset. So we applied VIF, and predictors like $X50$ and $X63$ were found to have high VIF values and were consequently removed from the dataset, leading to a more reliable and robust model.

After we removed the problematic predictors from the dataset, we implemented each of the four methods (LASSO, ADLASSO, LADL, and ADLADL) on the updated dataset. As expected, the L1 methods (LADL, ADLADL) substantially reduced the number of predictors compared to the L2 methods (LASSO, ADLASSO). Notably, predictor $X12$ emerged as consistently influential across all models.

Moving forward to the second stage of the application, we applied Subsampling to reduce the number of explanatory variables, resulting in quite notable results. The $L2$ methods ultimately kept the same predictors, though with slight variations in their inclusion frequencies. On the other hand, LADL showed an increase in the number of non-zero coefficients, with several predictors exhibiting higher inclusion frequencies. At last ADLADL, demonstrated sparse predictor sets, with only a few predictors showing significant inclusion frequencies.

The main part of our study was to evaluate the predictive accuracy of each model obtained from the four methods. For model evaluation, we used Bootstrap and conventional Confidence Intervals (C.I) and 10-fold cross-validation. Bootstrap Sampling enhanced model performance, revealing consistent error distributions and providing insights into predictor significance. While $L2$ models showed consistent error distributions across folds, $L1$ models exhibited similar median error values with narrower interquartile ranges (IQR), suggesting comparable central tendency and spread.

Our findings indicate that while $L1$ methods were anticipated to be better suited for predicting factors affecting the overall health outcomes of patients undergoing therapy, the results suggest that $L2$ methods are more suitable. This discrepancy may arise from variations in the stability of each algorithm across different R libraries. Consequently, this suggests that our focus should be directed towards improving model evaluation methods and investigating additional regularization techniques to enhance predictive accuracy in linear regression analysis.

Bibliography

[1] Anika Buchholz, Norbert Holländer, and Willi Sauerbrei. On properties of predictors derived with a two-step bootstrap model averaging approach—a simulation study in the linear regression model. *Computational Statistics & Data Analysis*, 52(6):2778–2793, 2008.

[2] Ruggero De Bin, Silke Janitza, Willi Sauerbrei, and Anne-Laure Boulesteix. Subsampling versus bootstrapping in resampling-based model selection for multivariable regression. *Biometrics*, 72(3):274–275, 2016.

[3] Morris DeGroot and Mark Schervish. *Probability and Statistics*. Addison-Wesley, 4th edition, 2011.

[4] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1994.

[5] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *Introduction to Statistical Learning with Applications in R*. Springer, 4th edition, 2017.

[6] Enno Mammen and Soumendra Nandi. Bootstrap and resampling. In James E. Gentle, Wolfgang K. Härdle, and Yuichi Mori, editors, *Springer Handbook of Computational Statistics*, pages 499–500. Springer, 2nd edition, 2012.