12-18-2019

Exploration of Variable Importance and Variable selection techniques in presence of correlated variables

Sailee Rumao
sxr9810@rit.edu

Rochester Institute of Technology

College Of Science

Department of Mathematical Sciences

Exploration of Variable Importance and Variable selection techniques in presence of correlated variables

Author:

Sailee Rumao

Supervisor:

Prof. Ernest Fokoue

A thesis submitted for the degree of

MS Applied Statistics

December 18, 2019

Abstract

Variable selection is of utmost importance in aviation safety where the data contains a large number of highly correlated predictors and flight safety has to be accurately predicted. Variable selection methods were not encouraged in medical research where the subject-matter knowledge is limited. For this reason, Genell, Anna Nemes, Szilard Steineck, Gunnar Dickman, Paul W. (2010) conducted simulated study to compare Bayesian Model Averaging and stepwise regression to motivate medical researchers to conduct automatic variable selection on their regression models and encourage them to take advantage of it. In this era of data science and Machine Learning, we have extended this comparative study by considering Machine learning algorithms. Various studies have shown that the Recursive feature elimination (RFE) algorithm reduces the effect of correlation on the variable importance measure and results in minimal prediction error. In this study, we compare RFE-RF, RFE-SVM and Bayesian Model Averaging (BMA) for simulated data in the presence of correlation by varying sample sizes (30,300) for 45 variables considering both cases n<p and n>p. Our results show that the percentage of selecting true predictors is highest for the RFE-RF model of all the three models. However, though the overall percentage of selecting true predictors is highest for RFE-RF, the estimated probability of selecting correlated true predictors is better for the Bayes in comparison to the other methods.comparison to the other methods.

Acknowledgements

I want to thank my advisor Dr. Ernest Fokoue for his constant motivation and support throughout my thesis research. I also want to thank Dr. Carol Marchetti and Dr. Robert Parody for being on my thesis committee and for their valuable time and feedback on my work. I want to extend special thanks to Shawna Hayes and all the staff members of the Statistics Department for preparing me to complete this endeavor and making my journey at Rochester Institute of Technology a memorable one. Last but not the least, I want to thank my parents and my entire family for their love and support towards my aspiration to get the masters degree.

Signature Page

APPROVAL SHEET

Signed by the Thesis committee

Name	**Signature**	**Date**
Dr.Ernest Fokoue (**Supervisor**)		
Dr.Robert Parody (**Committee member**)		
Dr.Carol Marchetti (**Committee member**)		

"Essentially, all models are wrong, but some are useful"

— George Box (1976).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

This paper is motivated by the importance of variable selection in presence of highly correlated predictors in the real-life problem of aviation safety that has been studied and published (Gregorutti, Baptiste, Michel, Bertrand, Saint-Pierre, Philippe,2017). It is a very sensitive issue to assure flight safety. This has to be done by evaluating all possible risks by studying the flight data parameters extremely minutely and faultlessly. The problem here is that the flight data recorders provide a large amount of raw data which contains a large amount of highly correlated variables. For this reason, it is of utmost importance to select the most important variables in order to get accurate predictions of any hazardous or unexpected events (Gregorutti, Baptiste, Michel, Bertrand, Saint-Pierre, Philippe,2017). Variable Importance Analysis and selection techniques were developed and studied independently in the fields of Statistics and Machine learning. The variable selection has a common problem of effect due to correlation which has also been studied independently. There has been extensive work done previously about the variable selection in the presence of highly correlated predictors. In this paper, I have explored and reviewed some of these works done previously and have re-addressed the issue of effect due to correlated predictors on variable importance. To the best of my knowledge, there has been no study to compare the effect of correlation on variable importance measured by the methods I have considered for the study. Our goal in this paper is to explore and compare the effect of correlation on different variable importance techniques to find if any of these methods give better results or is superior over other methods under any respective conditions. In part 1 of this paper, I have briefly discussed the theory of correlation and previous work in variable importance analysis and variable selection. Further, I have talked about the theory of the methods chosen for this study and how variable selection is conducted using these methods. In the next part, I have conducted a simulation-based comparative study to understand the effect of correlation on variable selection under different settings such as number of parameters (p), number of rows (n), correlation level and the level of signal-noise ratio (SNR) in the model output for each model that is built using the chosen Methods/techniques. This experiment is conducted using both Statistical and Modern Machine Learning techniques. The models are studied for their ability to select the true predictors in the presence and absence of correlation. The model performances are evaluated based on the root mean square error and visualized using boxplots.

1.2 Background

The modeling process is efficient and well-established if the variables are small, fixed and uncorrelated or mildly correlated. Variable importance analysis is conducted for feature selection to get precise reliable modeling results.
Let's try to understand basics of variable importance by understanding the answers to following questions:

- What is Variable Importance?
- How is Variable Importance Defined?
- What are the different types of Variable importance measures or what are the different techniques used for variables selection?
- Why is variable selection important? In what fields does it play a significant role?
- What does variable importance affect?

Variable selection is done for multiple reasons and can be useful in different ways. Some of these are listed below in section 1.2.1. The goal of any data related project is to extract information or knowledge from the available data and to deduce significant inferences. Some practical applications of this might be to find a cure for a disease, to find loan defaulters, improvise web-search methods, for insurance modeling, to make strategic business decisions or any other real-life problem associated with data depending on the field of study and the purpose of the project. But the solution to a problem is never straightforward. Many applications have a large number of variables with only a few significant variables having relevant information required for the prediction or study of the project. There are numerous varying reasons for these non-contributing variables. However, in such cases, it is of utmost importance to consider only the important variables to complete the task efficiently and get a reliable outcome.

1.2.1 Reasons to use variable selection

- Variable selection has a significant advantage in a fast-paced world where this process enables the machine learning algorithm to train faster leading to decreased time required for training the model.

- Fewer variables contribute to less storage required in the system. Storage and fast processing time lead to significantly decreased costs involved in the process.

- It makes the model easily interpretable by making it less complex and giving the advantage of conducting meaningful data visualization to gain and showcase insights to the audience.

- It plays a pivotal role in refining and boosting the performance of a model by choosing the most significant variables contributing to the predictive accuracy of the model under study.

- It helps to reduce over-fitting by eliminating redundant and irrelevant variables from the model.

- Variable selection also increases model generalization.

- By keeping only the significant variables in the model, it makes the analysis more understandable to gain knowledge about the process and help make important decisions quickly and confidently.

1.2.2 Definitions of Variable Importance

The variable selection process can be understood as the process of selecting a subset of most important variables which optimizes the objective function and minimizes the risk.

Variable Importance does not have a single consolidated definition. As these techniques were developed independently in multiple fields, it has been defined in different ways by these methods. In the review paper by Wei, P., Lu, Z., Song, J. (2015), they have classified all the Variable Importance measures into three categories and have defined it as follows:

- A calculated or estimated quantity that measures the change in the model performance with respect to the change in the predictor variables given to the model.

- A value that quantifies the variability contributed by one or more predictor variables to the variability of the model.

- A value that quantifies the intensity of association between the model outcome and the predictor variable individually or as a set.

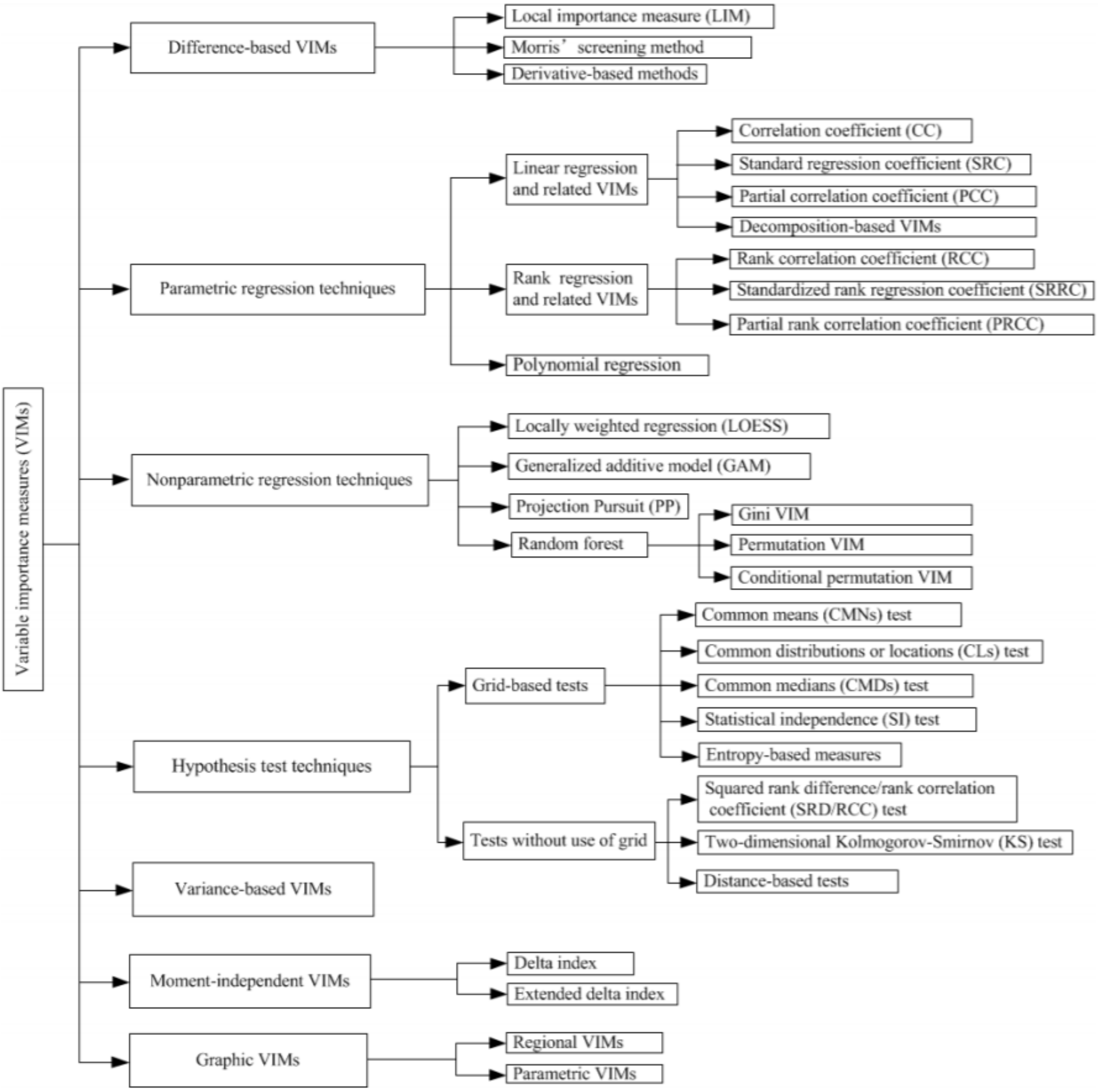

Variable importance measures (VIMs)
Difference-based VIMs
Local importance measure (LIM)
Morris' screening method
Derivative-based methods
Parametric regression techniques
Linear regression and related VIMs
Correlation coefficient (CC)
Standard regression coefficient (SRC)
Partial correlation coefficient (PCC)
Decomposition-based VIMs
Rank regression and related VIMs
Rank correlation coefficient (RCC)
Standardized rank regression coefficient (SRRC)
Partial rank correlation coefficient (PRCC)
Polynomial regression
Nonparametric regression techniques
Locally weighted regression (LOESS)
Generalized additive model (GAM)
Projection Pursuit (PP)
Random forest
Gini VIM
Permutation VIM
Conditional permutation VIM
Hypothesis test techniques
Grid-based tests
Common means (CMNs) test
Common distributions or locations (CLs) test
Common medians (CMDs) test
Statistical independence (SI) test
Entropy-based measures
Tests without use of grid
Squared rank difference/rank correlation coefficient (SRD/RCC) test
Two-dimensional Kolmogorov-Smirnov (KS) test
Distance-based tests
Variance-based VIMs
Moment-independent VIMs
Delta index
Extended delta index
Graphic VIMs
Regional VIMs
Parametric VIMs

Figure 1.1: Various Variable selection methods that follow either of the three definitions are reviewed in the paper(Wei, P., Lu, Z., Song, J. (2015))

1.2.3 Types of Variable selection Methods

With the on-going research in the past, there have been numerous methods developed in this area to solve problems with different conditions. As discussed in the introduction, methods have been developed independently in statistics, machine learning and different fields of study. These methods can be broadly grouped and classified in the section below. Studying each of these methods in depth is out of the scope of this study. I have listed the approaches that researchers have utilized to tackle this problem so far.

Supervised Methods:

1. Filter Method

This method is based on calculating a particular statistical measure for a variable and scoring it to rank the variables. Considering a particular threshold for the measure computed, those variables that are greater than or less than the threshold are then selected or eliminated based on these scores. Here it is assumed that the feature that is greater than the threshold contains more information and is more relevant in comparison to others. The only drawback of this method is that the measure computed for a particular variable does not take into consideration its association with output variables for the ranking purpose and is independent of the output variables or the model output change. This method is usually statistically univariate and measures intrinsic properties of the variable, unlike the wrapper method which selects those variable as important which optimizes the objective function and improve the model performance.

Some examples of filter method are as follows:1. Information gain

2. Chi-square test

3. Fischer score

4. Correlation coefficient

5. variance threshold

6. PCA

2. Wrapper Method

This method behaves like a search problem. In this method, every single possible combination of all the available variables is tried to compute the predictive model performance. Each of the models is scored based on the model accuracy and compared to choose the best model with the best model performance built with the optimal subset of variables. The wrapper algorithm takes into account the association between the variable subset search and model selection based on the model performance metric. It also has the potential to effectively deal with the correlated variables. (Sanz Hector,Valim Clarissa ,Vegas Esteban,Oller Josep M., Reverter Ferran (2018)).

The drawback of this method is that it has a higher risk of overfitting. Also, there might be a need to train and conduct cross-validation for each subset of variables to compare the model performance which is time-consuming and cost-expensive. Thus, the Filter method has an advantage over the wrapper method with regard to this aspect. This method can be methodological, stochastic or heuristic.

Some Examples of Wrapper Method are:

1. Forward selection method
2. Backward Elimination method
3. Stepwise Selection Method
4. Sequential Feature selection
5. Recursive Feature elimination

3. Embedded Method

This method works similarly to the wrapper method. However, embedded variable selection methods incorporate model learning using the performance measure in the process of variable selection. This method includes calculation of the change in the objective function along with the search for the best variable for each iteration in the modeling execution. It selects the variables which minimize the fitting error along with the modeling procedure.

Some examples of embedded method are:1.LASSO (L1 Regularization)

2.Ridge

3.Elastic Net

4.Decision Trees (ID3, C4.5, and CART)

5.Random Forest

Unsupervised Methods:

1. Matrix Factorization
2. Clustering

1.2.4 Problems associated with Variable selection

In addition to the benefits of variable importance analysis and variable selection discussed in section 1.2.1, there could be complications arising due to variable selection. These problems are statistically studied and are discussed in (Heinze, George Wallisch, ChristineDunkler, Daniela,2018). Refer to Figure 1.2. In this paper, they have also suggested possible solutions to these problems. They have mentioned some of these problems arising due to correlated predictors which are the center of our work.

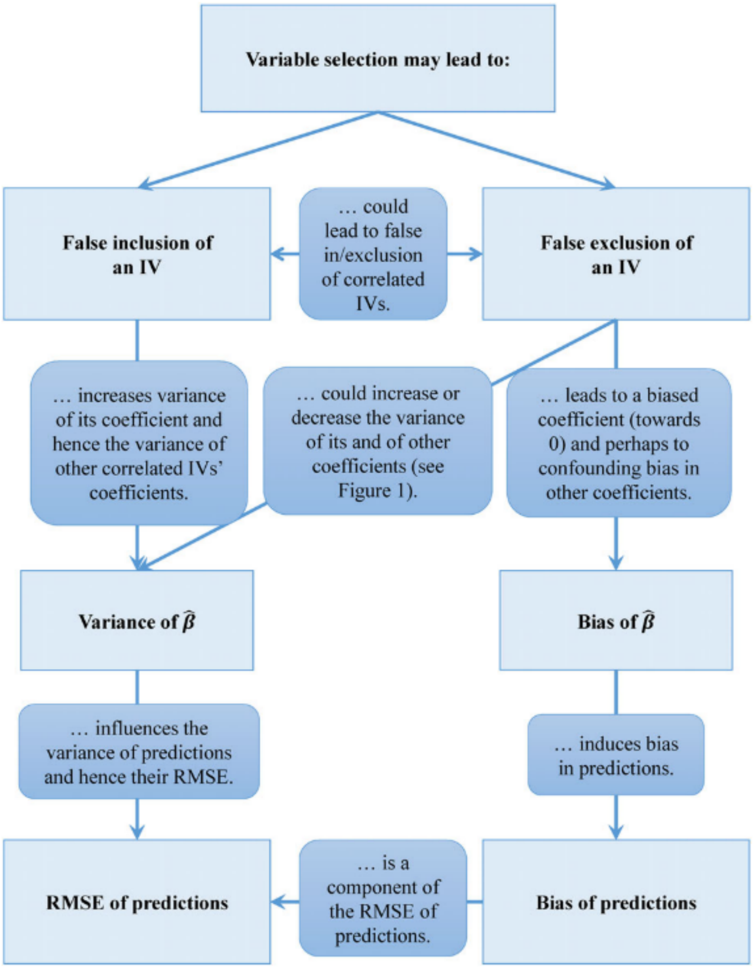

Variable selection may lead to:
False inclusion of an IV
… could lead to false in/exclusion of correlated IVs.
False exclusion of an IV
… increases variance of its coefficient and hence the variance of other correlated IVs' coefficients.
… could increase or decrease the variance of its and of other coefficients (see Figure 1).
… leads to a biased coefficient (towards 0) and perhaps to confounding bias in other coefficients.
Variance of $\hat{\beta}$
Bias of $\hat{\beta}$
… influences the variance of predictions and hence their RMSE.
… induces bias in predictions.
RMSE of predictions
… is a component of the RMSE of predictions.
Bias of predictions

Figure 1.2: Reprint from "Variable selection – A review and recommendations for the practicing statistician" by,Heinze,Georg Wallisch, ChristineDunkler, Daniela,(2018)

Some studies have also shown that sometimes the variable selection is not necessary and might not make a significant difference. Thus, before actually conducting variable selection, it is very important to answer questions like *"Should Variable selection be applied?" "Will it make a significant difference?" "Could there be any bad repercussions"*. Even after conducting variable importance analysis, it is crucial to conduct performance analysis, study Model sensitivity and test model stability to see if things have improved or had they worsened, in order to avoid hazardous implications of variable selection.

1.3 Thesis Scope

The scope of this study is limited to the linear regression framework for the two cases n<p and n>p where n is the number of observations and p is the number of variables. The number of observations for the case n<p is 30 and the number of observations for the case n>p is 300 and is fixed throughout the study. A crucial component of this research is to explore and analyze the behavior of the wrapper algorithms - Random forest and Support vector regression in the presence of correlation. Here the Pearson correlation coefficient is taken into consideration for the study and the correlation levels in the simulated data are set to three levels - uncorrelated variables, mildly correlated and highly correlated variables. Also, the experiments are conducted for a fixed low signal-noise ratio. Cross-validation to train the models under study is out of the scope of this study.

1.4 Organization

This thesis is organized in the following way. In Chapter 2, I have reviewed previous studies related to this area and introduced correlation and its types. In chapter 3, I have described the theory and mathematical concepts of all the models I have used in this study for comparison. In chapter 4, I have explained the simulation design procedure and have presented the results obtained from the experiments. In chapter 5, I have illustrated the choice of model performance metric and evaluated the results based on this metric. Finally, In chapter6, I have made conclusions from the study and made some remarks on the future scope for the experimentation.

1.5 Contributions

Although various studies have compared different variable importance measures and variable selection methods, there is no research work to particularly compare the wrapper algorithms against the Bayesian method of variable selection for a linear regression framework. In this study, in addition to exploring various variable selection techniques, I have comprehensively analyzed these methods and provided an assessment of their ability to conduct variable selection. I have primarily studied the models for their ability to choose the true predictors effectively. I have evaluated the model performance of these methods using a common metric root mean square (RMSE) which would be an appropriate metric in this simulated study. The study has paved the scope for more research in this direction further for classification and for non-linear cases.

Chapter 2

Literature Review

2.1 Previous Work

Various importance analysis has been widely conducted considering different constraints and its behavior has been studied under different problems of which variable selection in high dimensional data had taken a lot of attention in this era. It includes popular research on the high-dimensional Microarray gene data where the challenge is to gain accurate disease-related information from the data that contains an enormous number of genes as variables and noise. The variable selection has also been studied in other disciplines to study specific problems of univariate constraints, redundancy, categorical/nominal variables, supervised /unsupervised classification and/or regression machine performances. Here in this paper, our focus is to study the effect of correlation on Variable importance analysis and compare variable selection methods RF-RFE, SVM-RFE and Bayesian Model averaging. But before we dive into it, let's review some research papers to understand other work done in this area.

2.1.1 Synopsis of Variable importance and variable selection

Guyon, Isabelle De, Andre (2003) have discussed various methods of calculating variable importance measures and criteria for variable ranking. These Methods include statistical techniques like mutual information, correlation co-efficient, variable subset selection methods (forward and backward selection), Clustering and Matrix Factorization. They have also briefly discussed validation methods like Statistical t-tests. They have mentioned the issue with regard to the choice of data proportions for training purposes and for validation purposes.Further, they have also brought up statistical issues like variance, multi-class problems. They have also mentioned that the results of sophisticated embedded or wrapper methods are not always significant especially in high dimensional data sets. However, with decades of research work under these constraints, things have improved and models like random forests have given significant results. They have concluded the paper by setting a layout for further work to create a unified benchmark for different settings in variable selection and by conducting performance evaluation by comparison with respect to a baseline model. In this paper, we have addressed the issue of the effect of correlation on variable selection by conducting an experiment on simulated data and comparing the variable selection with the baseline models.

Correlation and VIM in random forest

Gregorutti, Baptiste Michel, Bertrand Saint-Pierre, Philippe (2017) have studied variable importance using the random forest in the presence of correlation combined with recursive feature elimination and compared it with Non-recursive feature elimination (NRFE).In their study, they have shown that recursive feature elimination (RFE) with Random forest performs better than NRFE. The following figure Figure 2.1 represents the first ten variables selected by RFE and NRFE algorithms from the Landstad dataset over 100 runs.

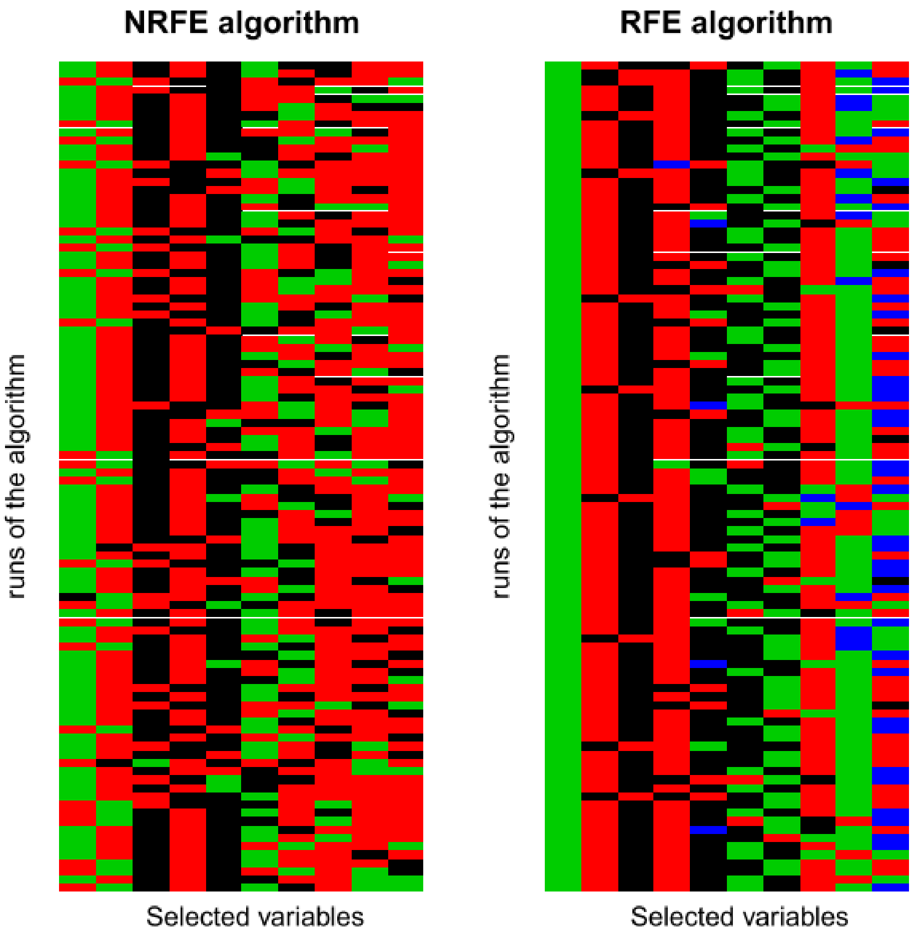


Figure 2.1: Gregorutti, Baptiste ,Michel, Bertrand, Saint-Pierre, Philippe,(2017)

The four colors in the figure represent four blocks of correlated variables. The horizontal line represents the block of variables selected and the vertical line represents the 100 iterations. From this study, they have concluded that the RFE algorithm is more consistent in selecting true predictors in comparison to the NRFE algorithm. These conclusions have motivated me to choose this model (Random forest - RFE) as one of the models for this comparative study for variable importance and variable selection.

Model selection in Medical Research: A simulation study comparing Bayesian Model Averaging and Stepwise Regression

Genell, Anna Nemes, Szilard Steineck, Gunnar Dickman, Paul W. (2010) have studied Bayesian Model averaging and stepwise regression for regression models both in the presence and absence of correlation. They have concluded that Bayesian Model averaging (BMA) performs better in selecting true predictors in comparison to the stepwise regression.

Numerous studies have been conducted to study variable importance using SVM-RFE for classification. Ishak, A Ben, inspired from the SVM-RFE algorithm has introduced a new norm which is a tweak to the existing SVM-based metric for variable selection and found that this SVM norm performs well in variable selection even for models with small n and large p and has also shown that it is computationally faster. Sanz, Hector Valim, Clarissa Vegas, Esteban Oller, Josep M. Reverter, Ferran (2018) have studied SVM-RFE for non-linear kernals. Limiting to the scope of this study, I wanted to experiment and analyze the behavior of the SVM-RFE algorithm for the regression problem framework under consideration.

2.2 Correlation and Variable Importance Analysis

Correlation is the change between two variables that go along with each other. In other words, if with the change in one variable, the other variable also changes concurrently then the two variables are said to be correlated. This proclivity of variation between the two variables at the same time is called a correlation and indicates some kind of relationship between them. This relationship can go in the same direction or opposite direction. If so, the variables are termed as positively correlated or negatively correlated respectively. If there is no relationship between the two variables then the variables are said to have a neutral correlation.

Correlation between two variables is quantitatively estimated which informs us of the level of connection or association among the two variables. There are different methods to compute this numeric index. These methods are abstracted below:

2.2.1 Correlation and its types

1. Pearson correlation coefficient (r):

   This is the most widely used metric to gauge correlation in statistics. This index is practiced to measure the correlation strength also known as the effect size, between two continuous and linear variables. It can also be used for dichotomous categorical variables that are encoded in binary format. It is not a suitable index for categorical variables with more than two levels. For this index, the random error of the data should be equally spread over all the variables (Homoscadesticity).

   The pearson correlation coefficient r is computed as follows:

$$r_{xz} = \frac{n\sum x_i z_i - \sum x_i \sum z_i}{\sqrt{n\sum {x_i}^2 - (\sum x_i)^2}\sqrt{n\sum {z_i}^2 - (\sum z_i)^2}} \tag{2.1}$$

   where,

   r_{xz} = Pearson correlation coefficient r between x and z

   n = Total number of observations.

   x_i = ith observation of variable x.

   z_i = ith observation of variable z.

   $-1 \leq r_{xz} \leq 1$

2. Kendall's correlation coefficient (τ):

   This index is used to measure the ordinal relationship between two variables. Thus it is the measure of the rank correlation of two variables. It is used as statistics for non-parametric tests to study if the two variables are dependent or independent.

$$\tau = \frac{n_c - n_d}{\frac{n(n-1)}{2}} \tag{2.2}$$

where,

$n_c =$ number of concordant pairs

$n_d =$ number of discordant pairs

3. Spearman's Rank correlation coefficient (ρ):

   This index is similar to the Kendall's correlation coefficient and is suitable for the ordinal data.

   It is calculated as follows:

$$\rho = 1 - \frac{6 \sum {d_i}^2}{n(n^2 - 1)} \tag{2.3}$$

Where,

$\rho =$ Spearman's rank correlation coefficient

$d_i =$ The difference between the ranks for each observation between the two variables.

$n =$ Total number of observations.

Since the scope of our study is limited to the continuous and linear dataset, we have used Pearson's correlation coefficient for this study.

Nicodemus, Kristin K. Malley, James D. (2009) have shown that the Random forest Gini index is biased towards correlated variables and permutation-based variable importance measures are mildly impaired by correlation. Degenhardt, Frauke Seifert, Stephan Szymczak, Silke (2019) have inferred that recomputing variable importance recursively as in the RFE algorithm results in better results for correlated variables. They have also declared that the random forest variable importance measure Mean decrease in Gini Index is biased and that permutation variable importance is not influenced by correlation. In addition to the above, Gregorutti, Baptiste Michel, Bertrand Saint-Pierre, Philippe (2017) have theoretically attested that permutation importance measure in the random forest is susceptible to correlation and should be recomputed after each variable is eliminated using the RFE algorithm. Therefore I have Incremental MSE (%IncMse) which is a permutation-based variable importance measure for the experimental study using the RFE algorithm.

Chapter 3

Methodology

3.1 Recursive Feature Elimination

Recursive feature elimination (RFE) and Non-Recursive feature elimination (NRFE) are types of Wrapper algorithms. Wrapper algorithms are succinctly described in section 1.2.3. These algorithms are computationally expensive but they do a full-scaled inspection with all possible combinations of the variables. This algorithm ranks variables according to the criteria and then traverses the best model based on it. The distinction in these techniques depends on two things:

- Computation of Variable importance measure.
- Computation of Predictive performance.

The differentiation between the two algorithms can be seen from the working of the two respective algorithms paraphrased below:

Algorithm 1 Non-Recursive Feature Elimination (NRFE)

```
Train the Model using all the variables
Rank the Variables using the variable importance measure
Calculate the Model Performance metric
Eliminate the least important variable
for all the remaining variables do
    Train the Model again
    Eliminate the least important variable
    Continue until convergence or no variables left
end for
Calculate Model performance metric for each Variable subset $V_i$
Choose the subset with best Model Performance as best subset of variables.
```

- **Note in this algorithm that the rank is not re-calculated or updated each time a variable is eliminated and the model is re-built.This is the primary difference between NRFE and RFE**.

Algorithm 2 Recursive Feature Elimination (RFE)

```
Train the Model using all the variables
Calculate the Model Performance metric (MSE)
Calculate the Variable importance measure
Eliminate the least important variable
for all the remaining variables do
    Train the Model again
    Calculate the Variable importance measure
    Eliminate the least important variable
    Continue until convergence or no variables left
end for
Calculate Model performance metric for each Variable subset $V_i$
Choose the subset with best Model Performance as best subset of variables.
```

3.1.1 Recursive Feature elimination (RFE) vs Non-Recursive Feature elimination (NRFE)

NRFE and RFE are compared in (Svetnik et al. ,(2004)) for real data sets where NRFE performs better. In (Gregorutti, Baptiste Michel, Bertrand Saint-Pierre, Philippe,2017) NRFE and RFE for Random forest are analyzed for simulated data in the presence of correlation. The results for this research show that RFE based on random forest performs more reliable than NRFE in the ubiquity of correlated predictors.

RFE algorithm updates the variable importance measure at each step of the backward elimination approach iteratively.RFE warrants that the ranking of variables is consistent throughout each of the models by re-calculating it in each of the iterations. (Gregorutti, Baptiste Michel, Bertrand Saint-Pierre, Philippe,2017). In this study, I am eliminating one feature at a time. The stopping criteria are until there are two variables left which would be regarded as the most important variables.

The variable selection has a very prevalent issue of model instability. This issue can be best administered by the usage of bootstrap samples. In their paper, Gregorutti, Baptiste Michel, Bertrand Saint-Pierre, Philippe,2017, they demonstrate that RFE performs better than NRFE for simulated data. Therefore we have adopted the RFE over NRFE to conduct this experiment to examine different methods of variable selection.

3.2 Random Forest

Introduction

Random Forest is an ensemble learning algorithm. Ensemble learning models strive to reduce the model prediction errors as a result of the Bias Variance decomposition, by aggregating the performance of say, k models. There are two types of ensemble learning algorithms: Bagging and Boosting. Random Forest operates on the mechanism of the Bagging method.

Consider $\mathbb{D} = \left\{\mathbf{Z_1}, \mathbf{Z_2}, \cdots, \mathbf{Z_n}\right\}$ where $Z_i = (X_i, Y_i)$ to be a learning data set where $X_i = (X_{i1}, \cdots X_{ip})$ is the p-dimensional input vector and $Y = (\mathbf{Y_1}, \mathbf{Y_2}, \cdots \mathbf{Y_n})$ is the response vector.

We know that the true prediction error R is not known in reality. For this reason, we estimate this error R using the sample $\mathbb{D}$ as

$R = \frac{1}{|\mathbb{D}|} \sum_{(X_i, Y_i) \in \mathbb{D}} (Y_i - \hat{f}(X_i))^2$

And we would have M different estimators of f, $\hat{f_1}(\cdot), \cdots \hat{f_m}(\cdot)$

where $m = 1, 2, \cdots M$ computed from the k samples of $\mathbb{D}$

Here each $\hat{f_m}(\cdot)$ is called the Base learner.

Our goal is to find an estimator of f with the smallest prediction error.

Assume f :function that models relationship between $\mathcal{X}$ and $\mathcal{Y}$

$\hat{f_1}(\cdot), \hat{f_2}(\cdot), \cdots, \hat{f_m}(\cdot)$: M different estimators of f

Mathematically,

$$\hat{f}^{(agg)}(\cdot) = \sum_{m=1}^{M} \alpha_m \hat{f_m}(\cdot) \quad i = 1, 2, \cdots M \tag{3.1}$$

Where,

α_m : Weights that measure relative importance of each estimator $\hat{f_m}(\cdot)$ m=1,2,3,..M

Working:

Before we jump into the working of Random forest, Lets understand the Out of Bag Error (OOBE)

Out of Bag Error

Consider $\mathbf{z} \in \mathbb{D}$ be a random pair drawn from $\mathbb{D}$ and
Consider bootstraped sample $\mathbb{D}^{(m)}$ of size n , then,

$Pr(\mathbf{z} \in \mathbb{D}^{(m)}) =$ Proportion of observations from $\mathbb{D}$ present in $\mathbb{D}^{(m)} = 1 - (1 - \frac{1}{n})^n$ ....(*)

$Pr(\mathbf{z} \notin \mathbb{D}^{(m)}) =$ Proportion of observations from $\mathbb{D}$ not present in $\mathbb{D}^{(m)} = (1 - \frac{1}{n})^n$ $=Pr[O_n]$ .....(**)

As $n \to \infty \quad , Pr[O_n] \to e^-1 = 0.37$

This implies that approximately one third of the training set is not used to build mth bootstraped base learner. This proportion is used to calculate the out of bag error.(used in random forest to estimate variable importance.)

Now let $\hat{f}^{(m)}(.)$ be the base learner from $\mathbb{D}^{(m)}$.
Then the out of bag error of $\hat{f}^{(m)}(\cdot)$ is ,

$$err_{OOB}(\hat{f}^{(m)}(\cdot)) = \frac{\sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \notin \mathbb{D}^{(m)})\, \ell(\mathbf{y}_i, \hat{f}^{(m)}(\mathbf{x}_i))}{\sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \notin \mathbb{D}^{(m)})} \tag{3.2}$$

where,
$\mathbb{1}(\cdot)$: Indicator function
$\ell(\cdot,\cdot)$:Loss function
Thus, the OOB error for the ensemble is,

$$err_{OOB}(\hat{f}^{(agg)}(\cdot)) = \frac{1}{M}\sum_{m=1}^{M} err_{OOB}(\hat{f}^{(m)}(\cdot)) \tag{3.3}$$

Random forest works on the foundation of bagging as described in the steps below:

The CART (decision tree for classification and regression) are known to be unstable for predictions subject to the training sample and may also tend to overfit the models. For these acumens, (Breiman,2001) developed the random forest as a significant improvement over the decision trees. The aggregation for the random forest is based on the predictions by all the trees built on all the bootstrap samples. The best split variable is determined based on the variable importance measure calculated for each variable in each bootstrap sample. The variable with optimal variable importance measure is chosen to be the split. This working can be well conceded from the steps below:

1. Bagging process

   (a) Subset the dataset $\mathbf{D_n}$ such that it has d variables from the original p-variables
   $(X_{j1}, X_{j2} \cdots, X_{jd})$: Subset of variables (where d < p).
   Note that here X_{jk} are chosen without replacement
   $\mathbb{D}^{(m)}$ is an nxd matrix. Note that here m stands for the number of trees grown in the process.

   (b) Pick bootstrap sample from $\mathbb{D}^{(m)}$ (This is done with replacement).

   (c) Construct base learner for $\mathbb{D}^{(m)}$

2. Decide and compute for the Variable Importance Measure for $\mathbb{D}^{(m)}$

   (a) Set aside the bag of sample of size S = $e^{-1}n$

   (b) After $\hat{f}(m)(\cdot)$ is built which is the estimate for the mth tree of the total ntrees to be grown, calculate out of bag error i.e $\hat{\mathbb{E}}_{OOB}$ (raw)

   (c) For k=1 to d,

      i. Perform permutation of column j_k in OOB sample (shuffle the entries in column j_k)

      ii. Compute $\hat{\mathbb{E}}^{(k)}_{OOB}$

      iii. Compute $\hat{\mathbb{E}}_{OOB} - \hat{\mathbb{E}}^{(k)}_{OOB}$

3. To comprehend the outcome based on all the Base learners, the algorithm employs the majority vote for classification. For regression, the average value is utilized.

Random forest algorithm Variable importance measure

There are numerous variable importance measures like Gini-index, entropy, information gain, etc.

For this research, I have used the package randomForest which computes these two variable importance measures IncMSE and IncNodePurity.Several studies have explicated that measures based on Gini-index are biased. (Degenhardt, Frauke Seifert, Stephan Szymczak, Silke (2019)) %IncMSE also known as the Mean decrease in Accuracy is said to be a very informative measure of importance. Thus, in this study, I have used IncMSE for this simulation study and correspondence.

- ***%IncMSE***

  This metric measures the increase in MSE of predictions when a variable j is permuted.It is also know as Mean decrease accuracy for classification problems.

  The algorithm works computes IncMse in the following steps:

  1. Grow regression forest.

  2. Compute OOB-mse and call it mse0.

  3. for 1 to j variables: permute values of variable j, Make predictions under this condition and compute OOB-mse(j)

  IncMSE of j'th is (mse(j)-mse0)/mse0 * 100

Random Forest RFE algorithm

Many times in the case of correlated variables, models with a small subset of variables give good prediction performance. Since RFE recalculates the variable importance measure at each step, it takes into consideration the magnetism of correlated variables and picks the subset of variables most efficient in prediction. (Gregorutti Baptiste, Michel Bertrand, Saint-Pierre, Philippe (2017))

Now let us understand the implementation of the Random forest algorithm in the Recursive feature elimination mechanism from the steps in the algorithm outlined below.

Algorithm 3 Random Forest -Recursive Feature Elimination (RF-RFE)

```
 1: Train the Model using the Random forest algorithm described above for all the variables
 2: Calculate the Model Performance metric (MSE)
 3: Calculate the Variable importance measure %IncMse
 4: Eliminate the least important variable based on the variable importance measure.
 5: for all the remaining variables do
 6:     Train the Model again as in step 1.
 7:     Calculate the Variable importance measure IncMse
 8:     Eliminate the least important variable
 9:     Continue until convergence or no variables left
10: end for
11: Calculate Model performance metric (MSE) for each Variable subset $V_i$
12: Choose the subset with best Model Performance as best subset of variables.
```

In this algorithm, I am eliminating one variable which is least significant at a time. Then train the model repeatedly for the remaining variables and the variable importance measure is re-computed. I use the test data to make predictions every time and store the resulting root mean square error (RMSE) for each subset of the variable. For this experiment, I ran this algorithm 100 times and averaged the root mean squared error (RMSE) over these 100 iterations for each of the two data processes - Baseline and correlated.

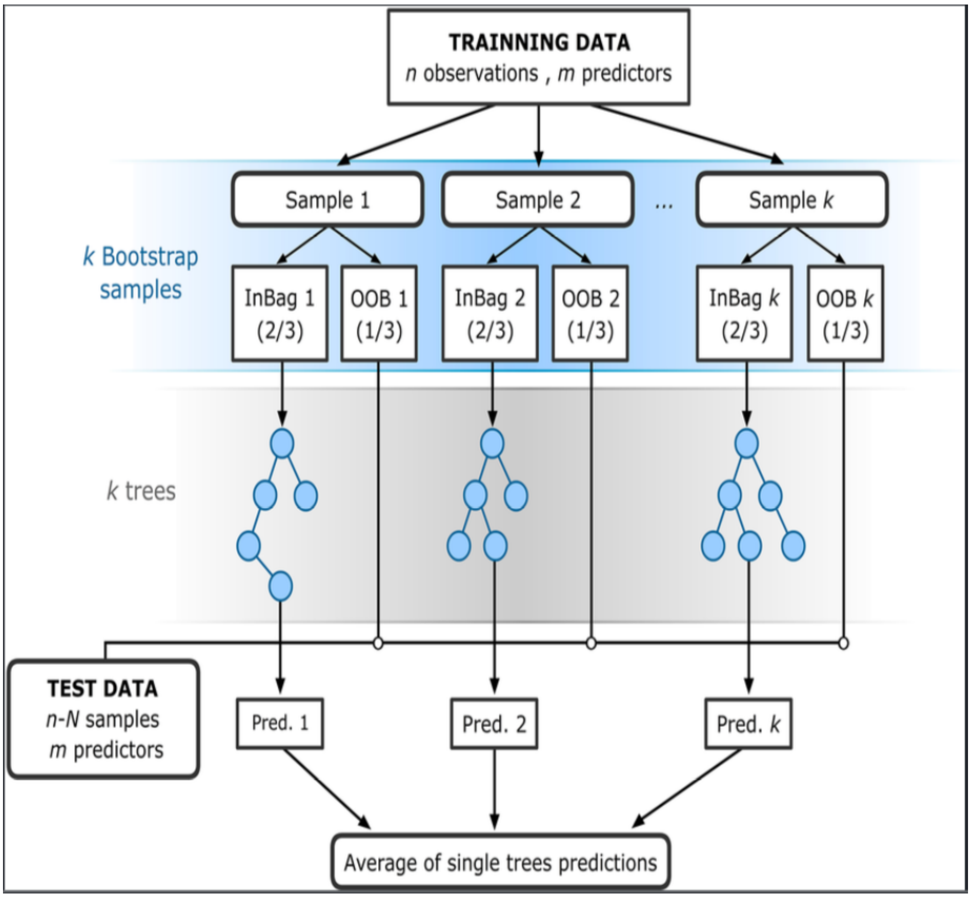


Figure 3.1: Rodriguez-Galiano et al.,(2015)

3.3 Support Vector Regression (SVR)

Introduction

Support vector machines were introduced by Vapnik (1963). The two categories of SVM are Support vector classification(SVC) and Support vector regression (SVR). Since the extent of this study is confined to the regression framework with a continuous-valued response variable, we will only look at Support vector regression. A support vector machine is the generalization of the well-known portrait algorithm. This algorithm allows the machine to learn and generalize the data that it has not seen ahead. Statistics is the establishment of support vector machines and they operate on quadratic constraints to minimize the objective function which incorporates the cost function and the regularization function. The difference between regression and Support vector regression is that unlike regression support vector regression intends to minimize the generalization error which is a combination of training error and the regularization term. The regularization function is a term that controls the complexity of the hyperplane. (Basak Debasish, Pal Srimanta, Patranabis Dipak Chandra,(2007)) Support vector machines are known to apportion well with sparse data and are also apprehended to handle the overfitting of the models.

Working

Support vector machines for classification are extended to support vector regression by adding an $\epsilon-$ insensitive region encompassing the optimization function which is called the $\epsilon-$ tube. The goal is to find the ϵ tube such that it maximizes the number of data points in this tube and minimizes the ϵ insensitive loss function.

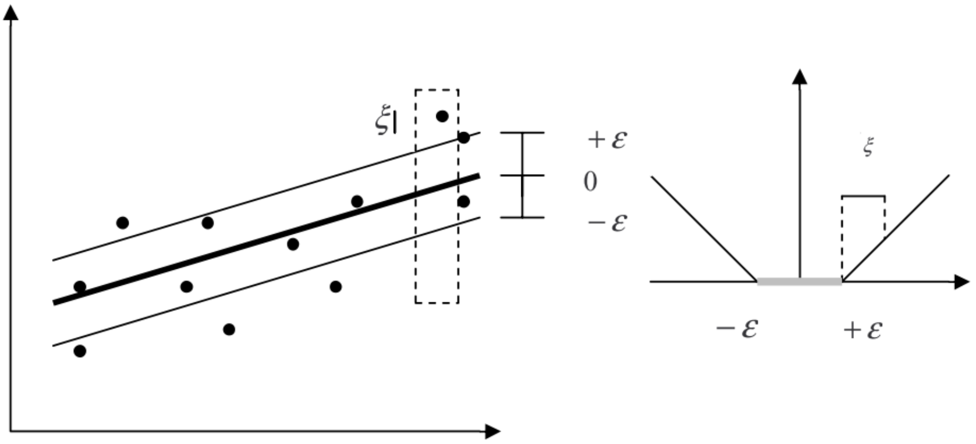


Fig. 2. The soft margin loss setting corresponds to a linear SV machine [11].

Figure 3.2: Basak Debasish,Pal Srimanta,Patranabis Dipak Chandra (2007)

Just like the SVM for classification, in SVR we reach the hyperplanes by using a complex algorithm to depreciate the errors and maximize the margins.

The optimization function to be approximated is mathematically reproduced as follows,

$$y = f(x) =< w, x > +b = \sum_{j=1}^{M} \mathbf{w_j x_j} + b \quad y, b \in \mathbb{R}, \mathbf{x}, \mathbf{w} \in \mathbb{R}^M \tag{3.4}$$

The preceding equation is for one-dimensional data. If we have multi-dimensional data X then we can augment X with 1 and the mathematical equation of multivariate regression in equation 3.4 can be rendered as,

$$f(x) = \begin{bmatrix} w \\ b \end{bmatrix}^\top \begin{bmatrix} x \\ 1 \end{bmatrix} = \mathbf{w}^\top \mathbf{x} + b \quad \mathbf{x}, \mathbf{w} \in \mathbb{R}^M \tag{3.5}$$

The goal of support vector regression to minimize the ϵ loss function. The constraint here is to minimize the error between the predicted values of the function and the actual values. This commences to the minimization of the objective function beneath:

$$min_{\mathbf{w}} \frac{1}{2} \|w\|^2 \tag{3.6}$$

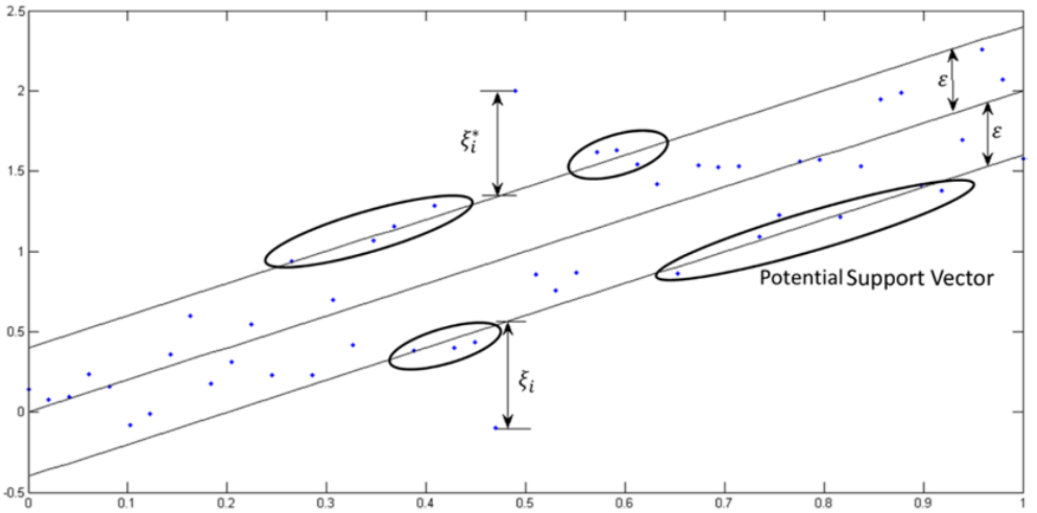


Figure 3.3: One-Dimension Linear SVR Source:Awad, Mariette Khanna, Rahul (2015)

In the Equation (3.6),
$\|w\|$ Magnitude of the vector to the plane that is being approximated. ω is also called as the weight vector.
SVR uses the ϵ loss function to penalize the data points that are beyond ϵ from the coveted output. The value of ϵ circumscribes the width of the tube. Lower width of ϵ indicates low error tolerance and a higher number of support vectors while a higher width of ϵ indicated vice-versa. The choice of the loss function depends on the antecedent information concerning the data, signal to noise ratio and training complexity. In this instance, I have acknowledged the linear loss function also known as the L1 hinge loss which is defined as follows:

$$L_\epsilon(y, f(x,w)) = \begin{cases} 0, & |y - f(x,w)| \le \epsilon; \\ |y - f(x,w)| - \epsilon; , & \textit{otherwise;}. \end{cases} \tag{3.7}$$

According to a few studies, asymmetrical loss function assists decrease the number of support vectors. As in the case of SVM, in order to take precaution against the outliers, slack variables $(\xi_i + \xi_i{}^*)$ can be supplemented by the following the soft margin approach. Furthermore, a regularization term C is added to the optimization function 3.6, Then the equation shifts to asymmetrical loss function that can be represented as follows:

$$min_{\mathbf{w}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*) \tag{3.8}$$

Under the following constraints,

$$y_i - \mathbf{w}^\top \mathbf{x_i} - b \leq \epsilon + \xi_i \qquad i = 1, 2, \cdots n \tag{3.9}$$

$$\mathbf{w}^\top \mathbf{x_i} - b - y_i \leq \epsilon + {\xi_i}^* \qquad i = 1, 2, \cdots n \tag{3.10}$$

$$\xi_i, {\xi_i}^* \geq 0 \qquad i = 1, 2, \cdots n \tag{3.11}$$

Based on all the constraints, the optimization function is modified as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \boldsymbol{\xi_i}, \boldsymbol{\xi_i^*}, \boldsymbol{\lambda}, \boldsymbol{\lambda^*}, \boldsymbol{\alpha}, \boldsymbol{\alpha^*}) = min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} (\xi_i + {\xi_i}^*) + \sum_{i=1}^{n} \alpha_{\mathbf{i}}^* (y_i - \mathbf{w}^\top \mathbf{x_i} - b - \epsilon - \xi_i) + \\ \sum_{i=1}^{n} \alpha_i (-y_i - \mathbf{w}^\top \mathbf{x_i} - b - \epsilon - \xi_i) + \sum_{i=1}^{n} (\lambda_i \xi_i + \lambda_i^* \xi_i^*)\end{aligned} \tag{3.12}$$

$\lambda_i, \lambda_i^*, \alpha_i, \alpha_i^*$ are non-zero real valued Langrange multipliers. (Awad, Mariette Khanna, Rahul,(2015))

Since the experimental data in this study is linearly separable, I am using linear kernal for this study such that,

$$f(x) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) k(\mathbf{x_i}, \mathbf{x}) + b \tag{3.13}$$

Where, $k(\mathbf{x_i}, \mathbf{x}) = \phi(\mathbf{x_i}).\phi(\mathbf{x})$ is the dot product denoting linear kernal.

SVM Variable importance measure

Variable selection in Support vector machines works on the principles of wrapper algorithm where it searches through all the subset of variables for training the dataset to find the best subset based on the accuracy of the model. In our study, we further extend this by computing the model performance on the test dataset using the RFE algorithm.

In Support vector machines the weight vectors represent the hyperplane. These vectors are orthogonal to the hyperplane. The weight vectors are calculated using the dot product of the variable coefficients and the support vectors of the fitted model.

We have adopted these weight vectors to compute the variable importance measure for the support vector regression model in this research.

The SVM model is fit using the e1071 package in R Software. In this model, the epsilon is set to 0.9 and because our data is linearly separable, I have applied Linear Kernel.

SVM RFE algorithm

The SVM-RFE algorithm was proposed by Guyon et al. (2000) for selecting genes that are relevant for a cancer classification problem. The working of SVM-RFE is similar to that of RF-RFE and it contrasts in calculating the variable importance measure in addition to the model itself. It is illustrated in the subsequent steps below:

Algorithm 4 Support Vector Machines- Recursive Feature Elimination (SVM - RFE)

```
Train the SVM Model using all the variables
Calculate the Model Performance metric (MSE)
Calculate the Variable importance measure support vector weights
Eliminate the least important variable with the minimum weight.
for all the remaining variables do
    Train the Model again as in step 1.
    Calculate the Variable importance measure support vector weights
    Eliminate the least important variable
    Continue until convergence or no variables left
end for
Calculate Model performance metric for each Variable subset $V_i$
Choose the subset with best Model Performance as best subset of variables.
```

This algorithm also operates in an analogous way to that of RFE-Rf. The root means squared error (RMSE) is averaged over 100 iterations and plotted in the boxplots in the chapter for both the sample sizes (n=30,n=300) and both the data processes - baseline and correlated datasets. The minimum RMSE estimated in the process for each leave-one-out variable in the recursive feature elimination process additionally denotes the best variable subset size.

3.4 Bayesian Model Averaging

Bayesian Model Averaging follows undeviatingly from the Bayes Theorem. Bayesian Model Averaging (BMA) deals with Model skepticism. It scrutinizes all possible combinations of models in the model space $\mathcal{M}$. From Bayesian Model, we know that we need to describe the prior probability to get the posterior probability. In this case, we need to define the prior distribution on the model space $\mathcal{M}$ to get the posterior distribution $P(M_j|y)$ of each model M_j in the Model space $\mathcal{M}$

From the posterior model probabilities, the best model is determined.It is the one with highest posterior probability. Choosing one best model based on this criteria and despising all other model can give fallacious predictions. Thus,Bayesian model averaging (BMA) deems for the posterior probability of the models as weights and builds a model by averaging posterior results from all the individual models in the model space $\mathcal{M}$. (Steel, Mark F J (2011))

In other words, BMA resolves the predicament of Model uncertainty by estimating models for all possible combinations of the input variables X and by constructing a weighted average across all of them. For example,if X contains K input variables, then we would have 2^k variable combinations to be estimated.This accounts for a total of 2^K models. The model weights for this process are obtained from the posterior model probabilities.

To make inference using BMA,a non- model specific quantity Δ is calculated by averaging all the models. This can be mathematically manifested as follows:

$$p_{\Delta|y} = \sum_{j=1}^{\mathcal{J}} p_{\Delta|y,M_j} p(M_j|y) \tag{3.14}$$

In the above equation, $pM_j|y)$ is computed using bayes theorem and is as follows:

$$p(M_j|y) = p_y(M_j)p(\theta|M_j) \tag{3.15}$$

$p_y(M_j)$: the Marginal likelihood of M_j together with the prior probability of M_j dentoted as $p(\theta_j|M_j)$

And thus, the marginal likelihood is defined as follows:

$$p_y(M_j) = \int p(y|\theta_j, M_j)p(\theta_j, M_j)d\theta_j \tag{3.16}$$

Let $2^K = \mathcal{J}$. Since the total number of model $\mathcal{J}(j = 1, 2 \cdots \mathcal{J})$ can be very large and consequently it can be computational exhaustive to average over all the models, BMA performs simulations using Monte Carlo Markov Chains sampler to deal with the Model space $\mathcal{M}$ There are other sampler methods like coin-flip importance sampling algorithm, branch, and bound method, Bayesian Adaptive sampling approach (Steel, Mark F J(2011)) which are not used for our study.

Bayesian Variable importance measure

- For this research, I have adopted posterior inclusion probability (PIP) which is obtained by averaging across all the possible individual models.PIP is a measure that BMA calculates to indicate how likely it is for a variable to be included in the true model. Posterior inclusion probability is the mean of the posterior probabilities. This value calculated by the Monte Carlo Markov chain sampler in BMA.This measure gives us a reliable understanding of how well the data accommodates a particular variable and how much the variable is contributing to the response variable.

The Bayesian Model averaging (BMA) is concisely summarized in the following image.

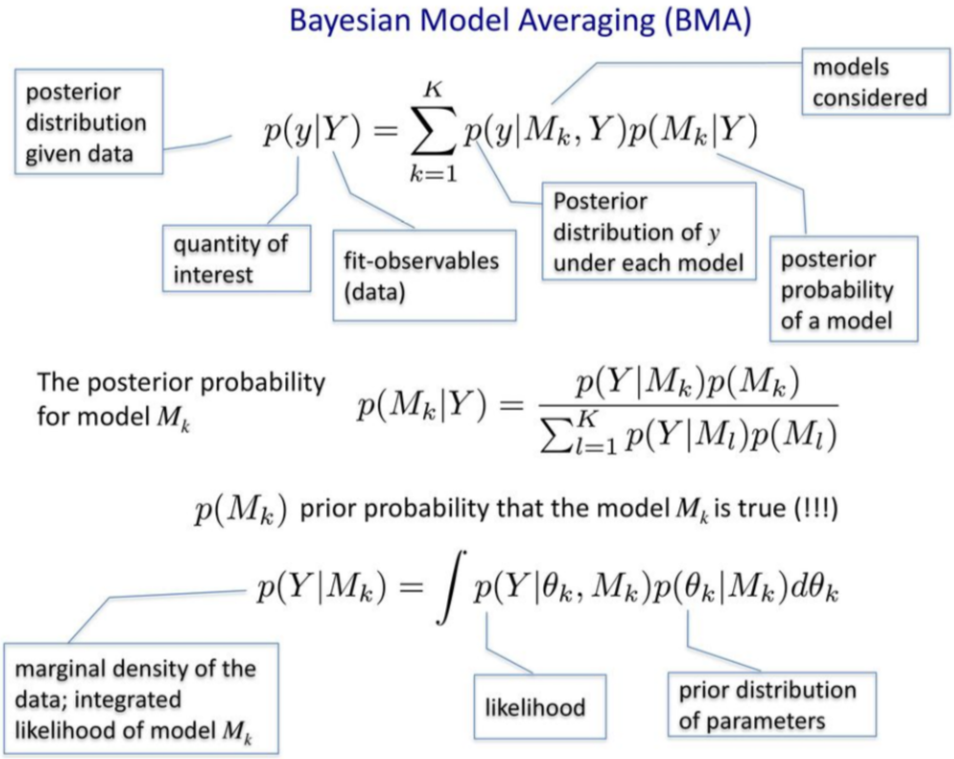

Bayesian Model Averaging (BMA)
posterior distribution given data
$p(y|Y) = \sum_{k=1}^{K} p(y|M_k, Y)p(M_k|Y)$
models considered
quantity of interest
fit-observables (data)
Posterior distribution of y under each model
posterior probability of a model
The posterior probability for model M_k
$p(M_k|Y) = \frac{p(Y|M_k)p(M_k)}{\sum_{l=1}^{K} p(Y|M_l)p(M_l)}$
$p(M_k)$ prior probability that the model M_k is true (!!!)
$p(Y|M_k) = \int p(Y|\theta_k, M_k)p(\theta_k|M_k)d\theta_k$
marginal density of the data; integrated likelihood of model M_k
likelihood
prior distribution of parameters

Figure 3.4: Susan Gardner,(2017)

This algorithm is independent of the recursive feature elimination concept (RFE). It is replicated over 100 iterations just like the other two methods and the appearance of the important variables in each instance is recorded. The important variables are selected from the ranked features by the posterior inclusion probability above a threshold of 0.5.

Algorithm 5 Bayesian Model Averaging

```
Compute Marginal Likelihood of each model.
Set the Model before distribution for each model.
Train the Bayesian model for each model and compute the posterior probability.
Perform MCMC simulations to sample the models with non-negligible posterior probability.
Fit the BMA model and Compute the posterior probability averaged over all the models.
Calculate the Model Performance metric (MSE).
Calculate the Variable importance measure posterior inclusion probability (PIP) that is obtained by averaging over all these models.
Eliminate the least important variable based on the posterior inclusion probability (PIP) by setting a threshold.
```

R package bms

- Used the default number of models: 500 to store the results from the best models and the default mcmc sampler in the package to experiment.

- I set the default model prior probability to the uniform distribution.

Chapter 4

Experimental Design and Analysis of Different Methods

In this chapter, I have explained the simulation process on the methods described in Chapter 3.I begin by explaining the Simulation design.Further I have explained the experimental data generating process for each method.

4.1 Simulation Design

In this experiment we simulate two data sets for the purpose of comparison as suggested in the conclusion of (Guyon, Isabelle De, André Elisseeff,(2003)):the Baseline design which has the orthogonal variables and the Correlated design which comprises of correlated variables.

A. Baseline data set

We simulate n=300 n=30 , p=45 i.i.d random normal and orthogonal variables from a multivariate Gaussian distribution

$$X \sim \mathcal{N}(\mu, \Sigma^{(0)}) \, .$$

Here,

μ is the mean and $\Sigma^{(0)}$ is the covariance matrix of the orthogonal variables.

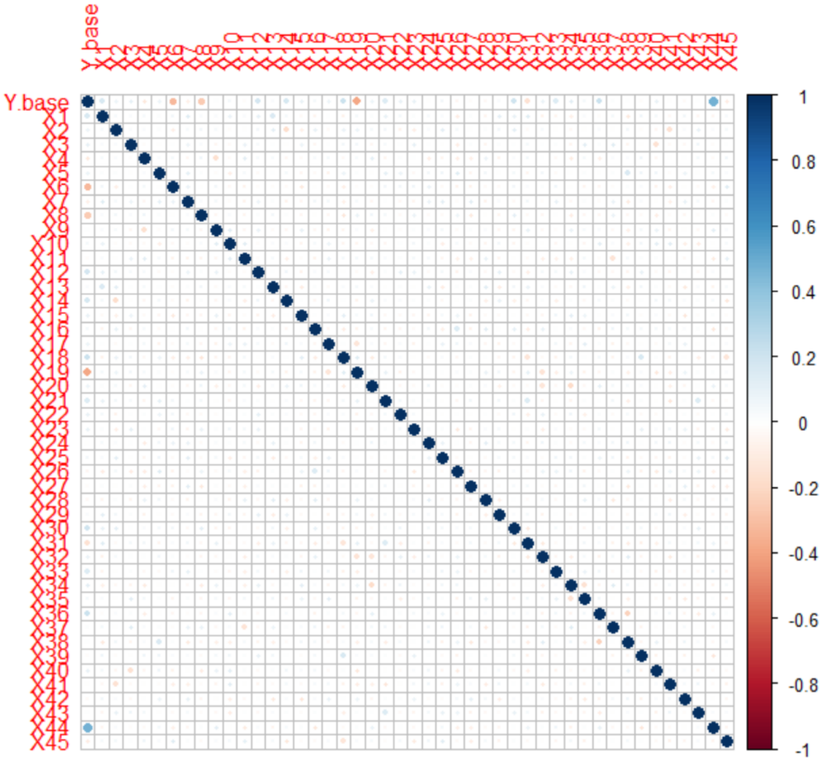


Figure 4.1: Correlation plot of the Baseline design, n= 300

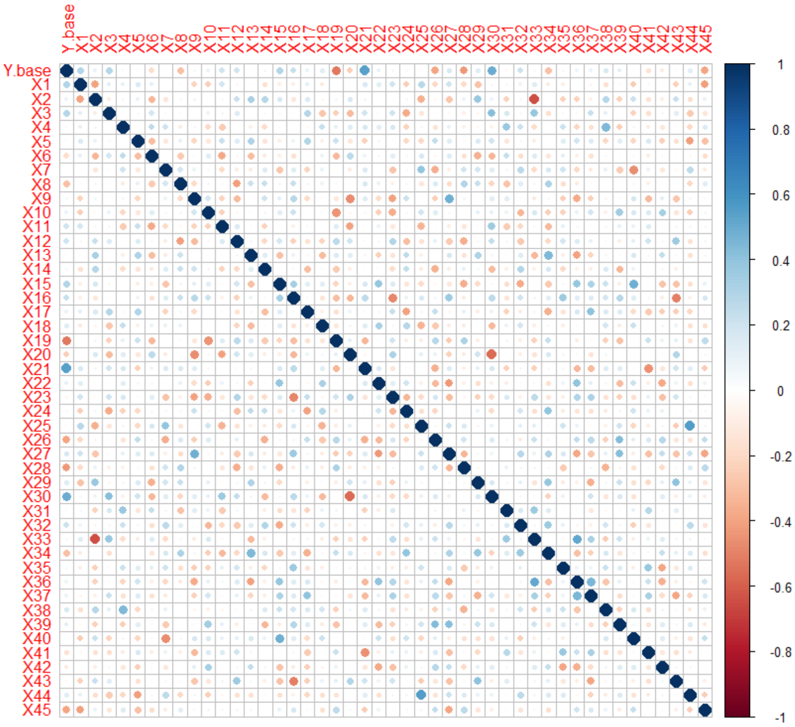


Figure 4.2: Correlation plot of the Baseline design, n= 30

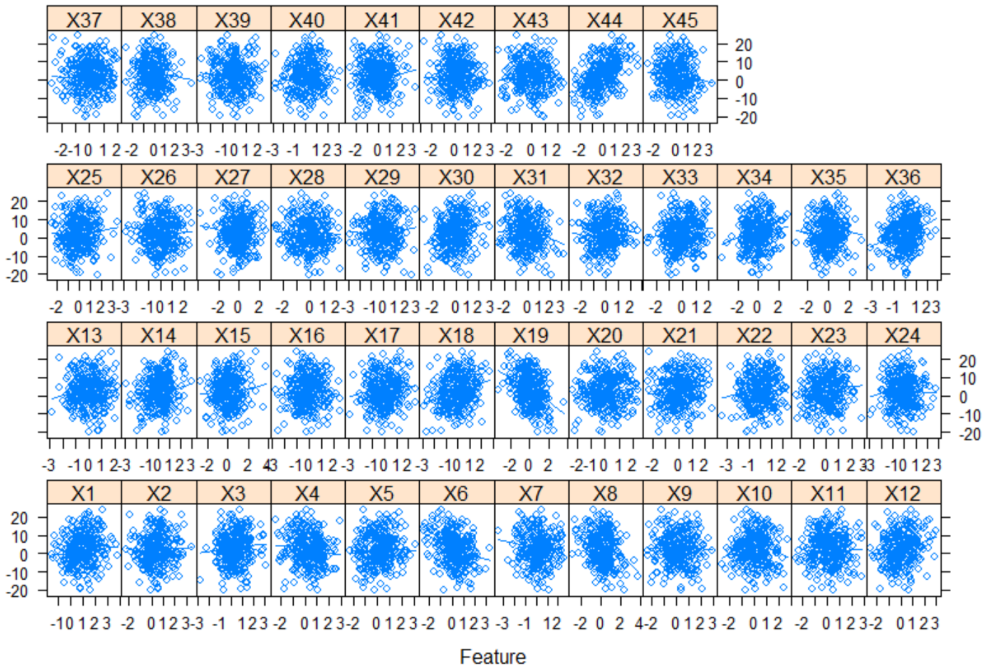


Figure 4.3: Scatter plot of the Independent variables with the dependent variables for the base dataset

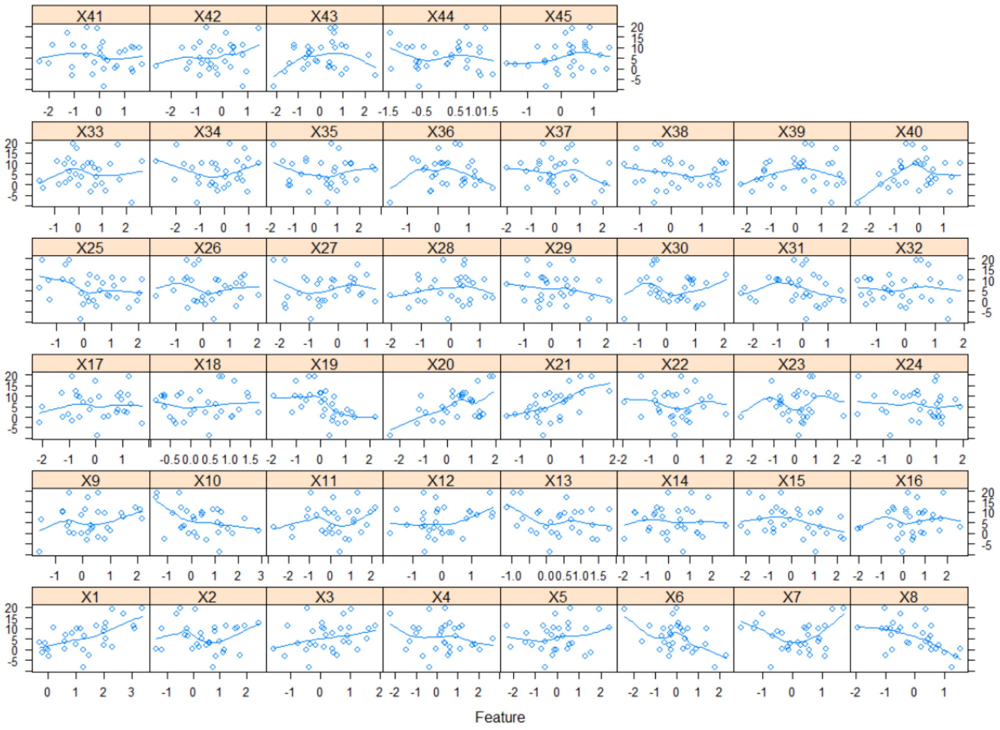


Figure 4.4: Scatter plot of the Independent variables with the dependent variables for the base dataset

B. Baseline data set with correlation induced Similarly for this Data set, I have simulate n=300,p=45 i.i.d random normal variates from a multivariate Gaussian distribution

$$X \sim \mathcal{N}(\mu, \, \Sigma) \, .$$

Where, $\Sigma =$ Block Covariance Matrix

such that the correlation levels are divided into three groups:

1. Orthogonal variables $({X_1}^{(0)}, {X_2}^{(0)}, {X_3}^{(0)}, \cdots {X_{p1}}^{(0)}) \sim$ N $(\mu, \Sigma^{(0)})$ with $\rho = 0$ and $p_1 = 15$

$$\mathbf{\Sigma^{(0)}} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{4.1}$$

2. Mildly correlated variables $({X_1}^{(0.5)}, {X_2}^{(0.5)}, {X_3}^{(0.5)}, \cdots {X_{p2}}^{(0.5)}) \sim$N $(\mu, \Sigma^{(0.5)})$ with $\rho = 0.5$ and $p_2 = 15$

$$\mathbf{\Sigma^{(0.5)}} = \begin{pmatrix} 1 & 0.5 & 0.5 & \cdots & 0.5 \\ 0.5 & 1 & 0.5 & \cdots & 0.5 \\ 0.5 & 0.5 & 1 & \cdots & 0.5 \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ 0.5 & 0.5 & 0.5 & \cdots & 1 \end{pmatrix} \tag{4.2}$$

3. Strongly correlated variables $({X_1}^{(0.9)}, {X_2}^{(0.9)}, {X_3}^{(0.9)}, \cdots {X_{p3}}^{(0.9)})$ ~N $(\mu, \Sigma^{(0.9)})$ with $\rho = 0.9$ and $p_3 = 15$

$$\mathbf{\Sigma^{(0.9)}} = \begin{pmatrix} 1 & 0.9 & 0.9 & \cdots & 0.9 \\ 0.9 & 1 & 0.9 & \cdots & 0.9 \\ 0.9 & 0.9 & 1 & \cdots & 0.9 \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ 0.9 & 0.9 & 0.9 & \cdots & 1 \end{pmatrix} \tag{4.3}$$

***Note* :** $p_1 + p_2 + p_3 = p = 45$

Then $X \in \mathbb{R} \sim N(\mu, \Sigma)$

Where, Σ is the Block-Covariance Matrix and is represented as follows:

$$\Sigma = \begin{pmatrix} \mathbf{\Sigma^{(0)}} & 0 & 0 \\ 0 & \mathbf{\Sigma^{(0.5)}} & 0 \\ 0 & 0 & \mathbf{\Sigma^{(0.9)}} \end{pmatrix} \tag{4.4}$$

Case: Regression

Based on the data matrix, the Response variable Y for the Model is generated as follows:

$Y = 1 + 2 * X_1 - 2 * X_8 + 7/8 * X_{12} + 2/5 * X_{13} + X_{14} + X_{16} + 2 * X_{18} - 3.0 * X_{19} + X_{20} + 2 * X_{21} + X_{23} + X_{29} + 1.5 * X_{30} - X_{31} + X_{3_2} + X_{33} + 2 * X_{34} - 0.25 * X_{35} + 0.85 * X_{36} + 4 * X_{44} + \epsilon$

Where $\epsilon \sim N(0, \sigma^2)$ is the random gaussian noise with Homoscadestic constant variance and is generated as $\epsilon = 3 * rnorm(nn)$

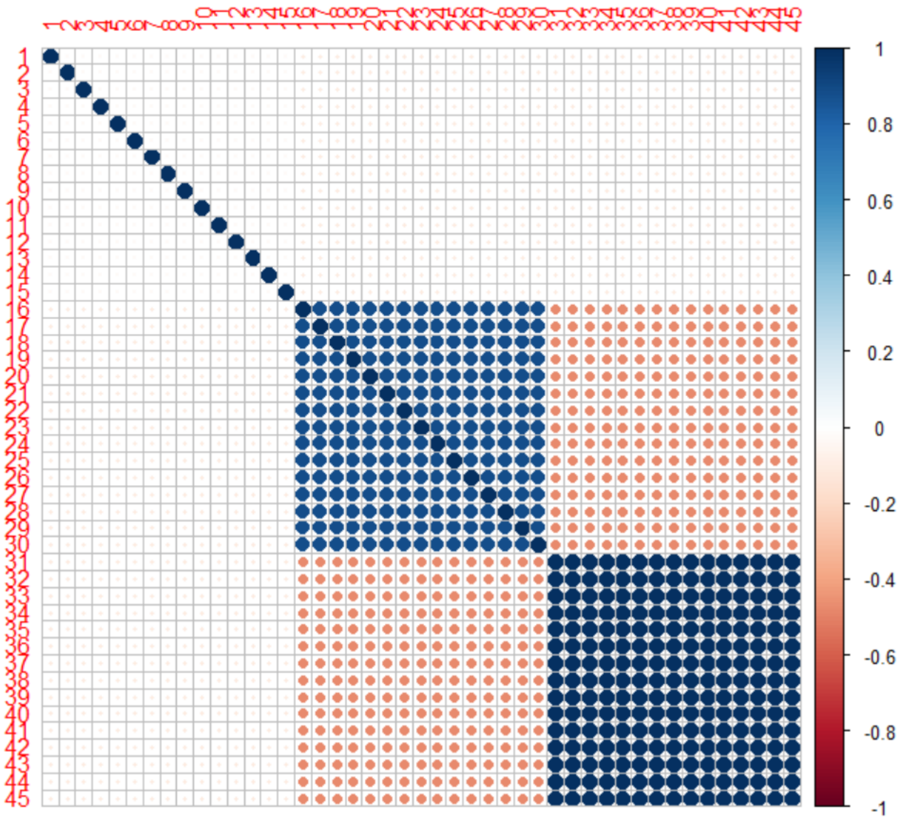


Figure 4.5: Correlation plot of the Block Covariance Matrix,n=300

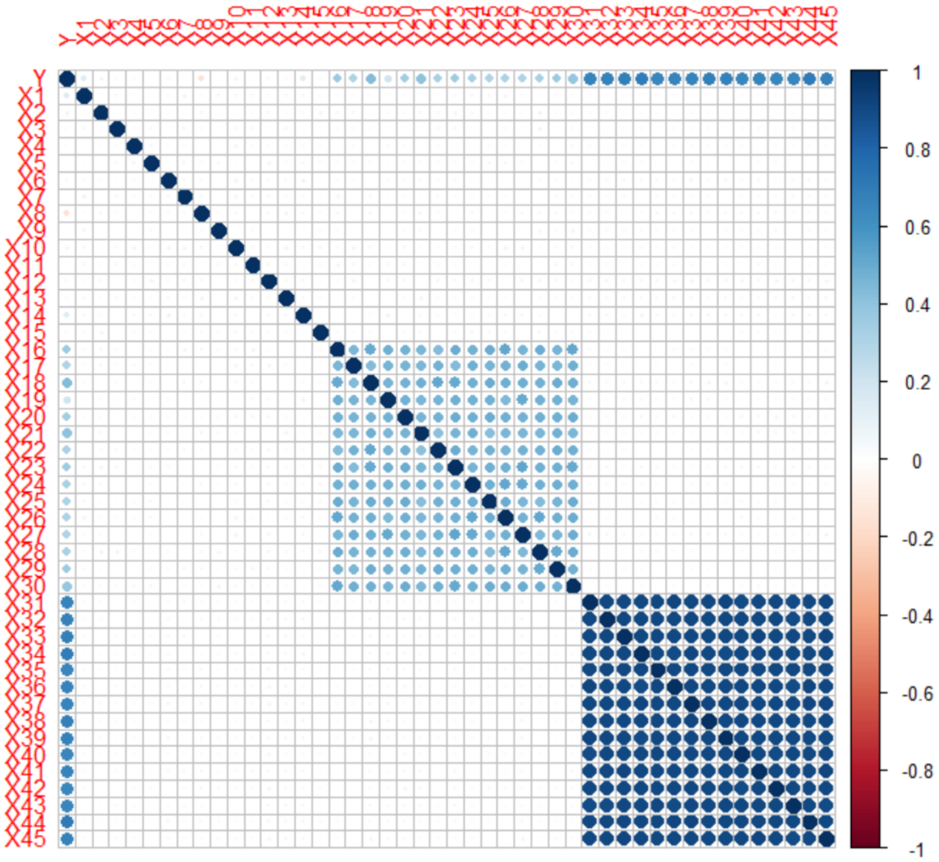


Figure 4.6: Correlation plot of the Data Matrix generated from the block covariance matrix,n=300

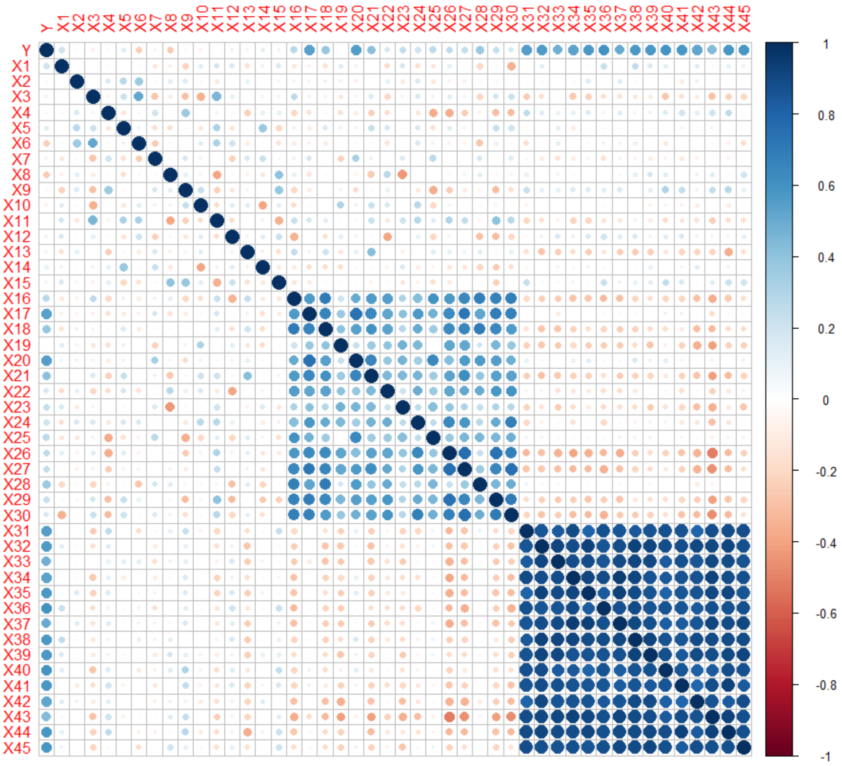


Figure 4.7: Correlation plot of the Data Matrix generated from the block covariance matrix, n=30

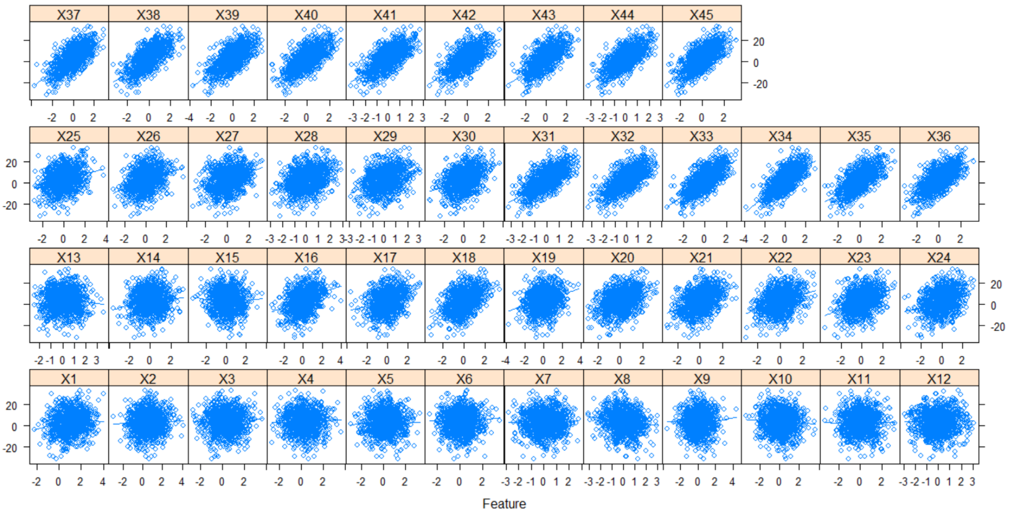


Figure 4.8: Scatter plot of the Independent variables with the dependent variables for correlated data

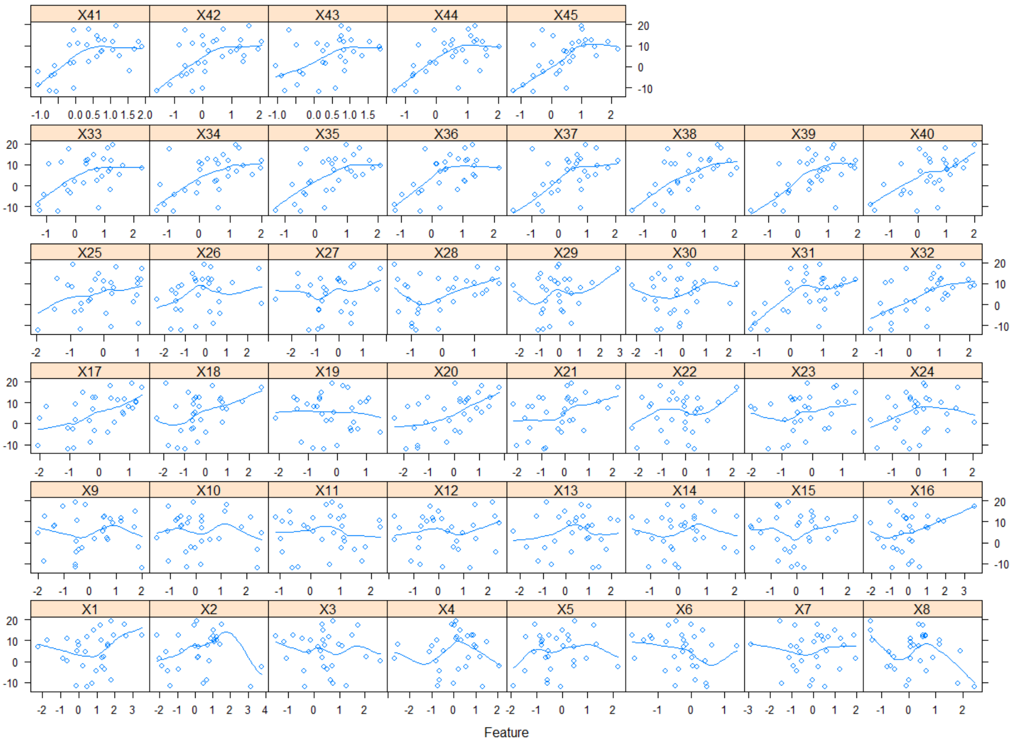


Figure 4.9: Scatter plot of the Independent variables with the dependent variables for correlated data

4.1.1 Signal to Noise Ratio

The signal to noise ratio is the ratio of the power of signal in data to the power of random noise.

Mathematically, For our Model

$$Y = f(x) + \epsilon \tag{4.5}$$

$$\textbf{Signal-noise ratio (SNR)} = \frac{V(f(x))}{V(\epsilon)} \tag{4.6}$$

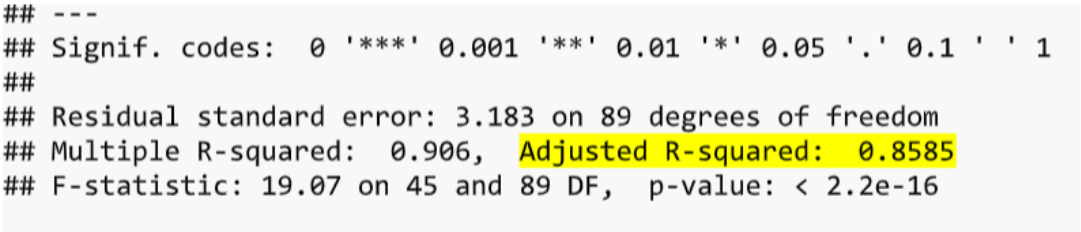
Here,

$V(f(x))$ = Variance of the data

$V(\epsilon)$ = Variance of the noise

For the experiment, $V(f(x))$ is randomly pre-defined as 3 to achieve a low signal-noise ratio (SNR). $V(\epsilon)$ is obtained from a normal distribution such that $\epsilon \in N(0, \sigma^2)$.

This furnishes us with a low SNR that is verified from the R-square of the linear model.R-square reveals the proportion of total variability in the model due to the model predictors. For our model, we confirmed to have a high R-square which implies a low Signal noise ratio. Thus, our model predictors contribute profoundly to the response variable. The impact of random noise on the model is controlled to be low on the response variable.



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.183 on 89 degrees of freedom
## Multiple R-squared:  0.906,  Adjusted R-squared:  0.8585
## F-statistic: 19.07 on 45 and 89 DF,  p-value: < 2.2e-16
```

Figure 4.10: R-square Adjusted

4.1.2 Experimental Data generating process

Using the simulation design described in the section above. I experimented to generate two datasets for both the Baseline and correlated design on the following two conditions:

1. **Data process 1 :** p»»>n i.e The dataset consists of 30 observations and 45 variables.This dataset was generated to understand the performance of the algorithms to the curse of dimensionality (Where the number of variables is less than the number of observations).

2. **Data process 2 :** p««<n i.e The dataset consists of 300 observations and 45 variables.

The experiment was conducted and studied for each of the variables. Selection methods on 4 different data processes that are compiled in the table below: Note that the number of variables in the entire experiment is 45 and is fixed.

Table 4.1: Summary of Data Processes

	Data Process 1	Data Process 2
Correlated	n = 30	n = 300
Baseline	n = 30	n = 300

Variables used to generate the response variables are defined as the true predictors for the model.

Experimental Design for RFE-RF and RFE-SVM

1. The data is split into train and test data.

2. Model is built using the RFE algorithm described in section 3.2 and 3.3 for each of the algorithms Random forest and Support Vector Regression respectively.

3. I used the IncMse in Random forest and the weight vectors in the Support vector machine respectively to remove the variable with the lowest value from the model.

4. Predictions are made on the test dataset and test error is recorded for each instance.

5. The algorithm in the first three steps in repeated over 100 iterations and the data is generated each time for all the 100 iterations.

6. The test error for model boxplot comparison calculated by averaging over these 100 iterations.

Experimental Design for Bayesian Model Averaging

The experimental design of this model is substantially similar to the one for RFE-RF and RFE-SVM. But, since the Bayesian Model averaging experiment was conducted independent of Recursive feature elimination, the modeling task was performed along with the 100 iterations.

1. The data is split into train and test data.

2. Model is built using the BMS package.

3. I used the posterior Inclusion Probability with a threshold of 0.5 to choose the variables to be kept in the final model.

4. Predictions are performed on the test dataset and test error is recorded for each instance.
5. These steps are repeated over all the 100 iterations.
6. The test error for model boxplot comparison is calculated by averaging over 100 iterations.

4.2 Results

The methods are being compared using probability in terms of selecting true predictors by each of the methods. The probability of true predictors is calculated as the proportion of selecting true predictors across all the selections made. The results are enumerated as percentages in the table below.

4.2.1 Percentage of selecting True predictors

Dataset with n = 300

Table 4.2: Method Comparison for Selecting True Predictors (n=300)

	Percentage (%) of selecting True predictors
Bayes correlated	53 %
Bayes base	57 %
RF correlated	92 %
RF base	99 %
SVM correlated	47 %
SVM base	47 %

Dataset with n=30

Table 4.3: Method Comparison for Selecting True Predictors (n=30)

	Percentage (%) of selecting True predictors
Bayes correlated	47 %
Bayes base	46.57 %
RF correlated	62 %
RF base	61 %
SVM correlated	47 %
SVM base	46.70 %

Observations:

The Bayesian model averaging with a threshold of 95 % on the posterior inclusion probability performed very poorly in selecting the true predictors in this case. So it is not acknowledged for comparative research. We can see that the Random forest performs admirably in selecting the true predictors and is somewhat affected in the presence of correlation. The same applied for Bayesian

Model averaging where it is slightly affected by correlation. However, SVM does not show any impression due to correlation as can also be seen in table 2 in section 3.2.

The proportion of variables selected other than the true predictors as registered in this table are redundant variables. Thus, we can see that the proportion of SVM selecting redundant variables is more distinguished in comparison to both other methods under study.

4.2.2 Examination of correlated True Predictors

The results are tabulated based on the data generating process 2 (Correlated Dataset).

Dataset with n=300

Table 4.4: Probability of true predictors selected by correlation level (n=300)

	Uncorrelated	Mildly correlated	Highly correlated
RFE Random Forest	0.84	0.00	0.07
RFE SVM	0.16	0.16	0.16
Bayes	0.16	0.10	0.28

Dataset with n=30

Table 4.5: Probability of true predictors selected by correlation level (n=30)

	Uncorrelated	Mildly correlated	Highly correlated
RFE Random Forest	0.3741	0.1223	0.1244
RFE SVM	0.1556	0.1556	0.1556
Bayes	0.1510	0.1582	0.17

4.3 Discussion

Observations:

Bayes model is slightly affected by correlation, particularly by the highly correlated variables. The probability of Random forest choosing uncorrelated true predictors is very high in comparison to other correlation levels.

A striking observation here is that the Random forest performs remarkably poorly in selecting the mildly correlated true predictors.

Chapter 5

Evaluation

5.1 Prediction and Test Error results

Performance evaluation is a significant component of any experimentation and plays a vital role in comparing different models.RMSE is a very popular performance error metric. There have been arguments that RMSE is inappropriate for comparing model performance for time series data (Armstrong and Collopy (1992)) and that it is an unreliable error measure to evaluate model performance (Willmott, Matsuura, and Robeson (2009)). However, (Chai, T. Draxler, R. R.(2014)) have shown that the RMSE is a more suitable measure to use when the error distribution is gaussian. They have also stated that root mean squared error has an edge over mean absolute error since it does not exercise the absolute value of the measure considering it is undesirable in mathematical calculations. As you can perceive in Figure (5.1) the simulated data used for experimentation in this study has errors following the normal distribution.

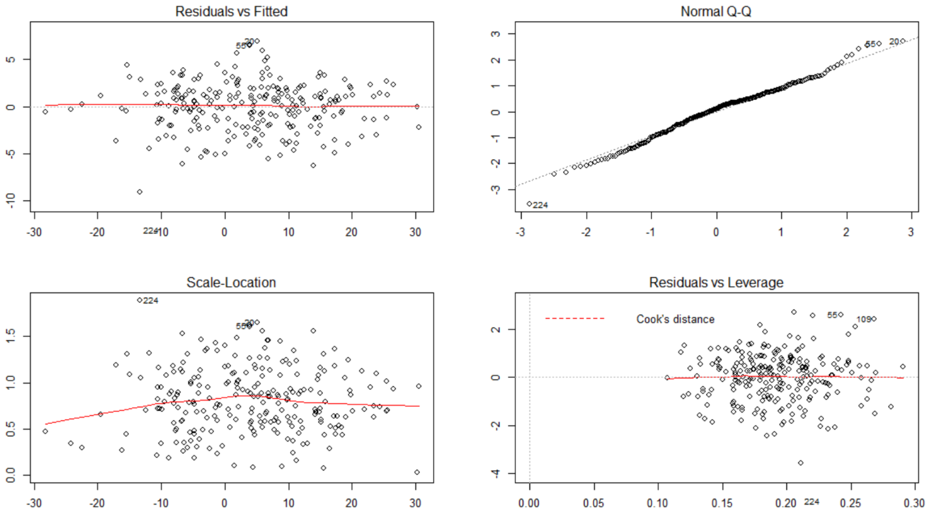


Figure 5.1: Residual plot to examine gaussian distribution of errors of the simulated data

Therefore, to evaluate the models, I have used the Root mean squared error (RMSE). RMSE is the quadratic metric that measures the error of model predictions. It is the average of the squared difference between the predicted and actual data points. It is mathematically delivered as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{5.1}$$

Where,

y_i = Actual data point (observation)

$\hat{y_i}$ = predicted data point (observation).

n= Total number of data points in the test data set.

RMSE is susceptible to outliers and this is one principal concern about this evaluation metric. But because our data is simulated, there are no outliers in the data. Thus, RMSE adequately evaluates our models. For real data sets, the outliers can be simply discarded to use this metric. Also, (Chai, T. Draxler, R. R.(2014)) has designated that for larger sample sizes, the distribution of the errors can be easily reconstructed. Also, note that since RMSE is a squared value, so it gives larger weights by penalizing the higher errors in comparison to the other performance error metrics. As (Chai, T. Draxler, R. R.(2014)) has stated, cost function to be minimized is often the squared term just like the RMSE. Therefore, penalty to the higher errors functions as the regularization for the incorrect predictions. Thus, RMSE is an accurate measure to compare the model performances and consequently spotlights the variations.

For the methods with recursive feature elimination, the RMSE is computed on test data set each time by fitting the model after eliminating each variable in the iteration. Thus we get a total of 45 RMSE values for each iteration denoting the error on the test data set in the absence of each of the variables one by one. I have repeated this process and computed the RMSE values as illustrated above each time for the test data set for all the 100 iterations of the experiment. To be more translucent, the 45 RMSE values obtained by the leave-one-out process in the recursive feature elimination methodology are averaged over 100 iterations. As mentioned above in the description of the methodology, the minimum RMSE also indicates the best subset size of variables. Note that lower RMSE denotes a more reliable performance of the model.

The two boxplots in Figure 5.2 and Figure 5.3 represent the distribution of root mean square errors on the test dataset for each of the models for the two sample size n = 30 and n=300 respectively.

5.1.1 Model Comparison n = 300

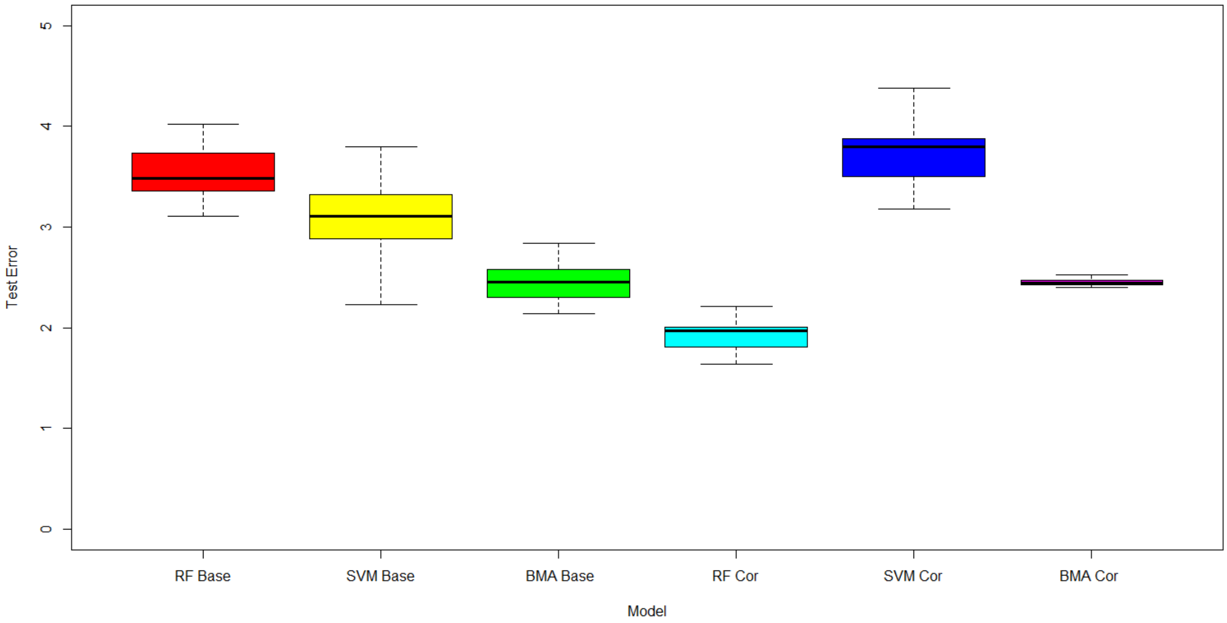


Figure 5.2: Comparative Boxplots (n=300)

5.1.2 Model Comparison n = 30

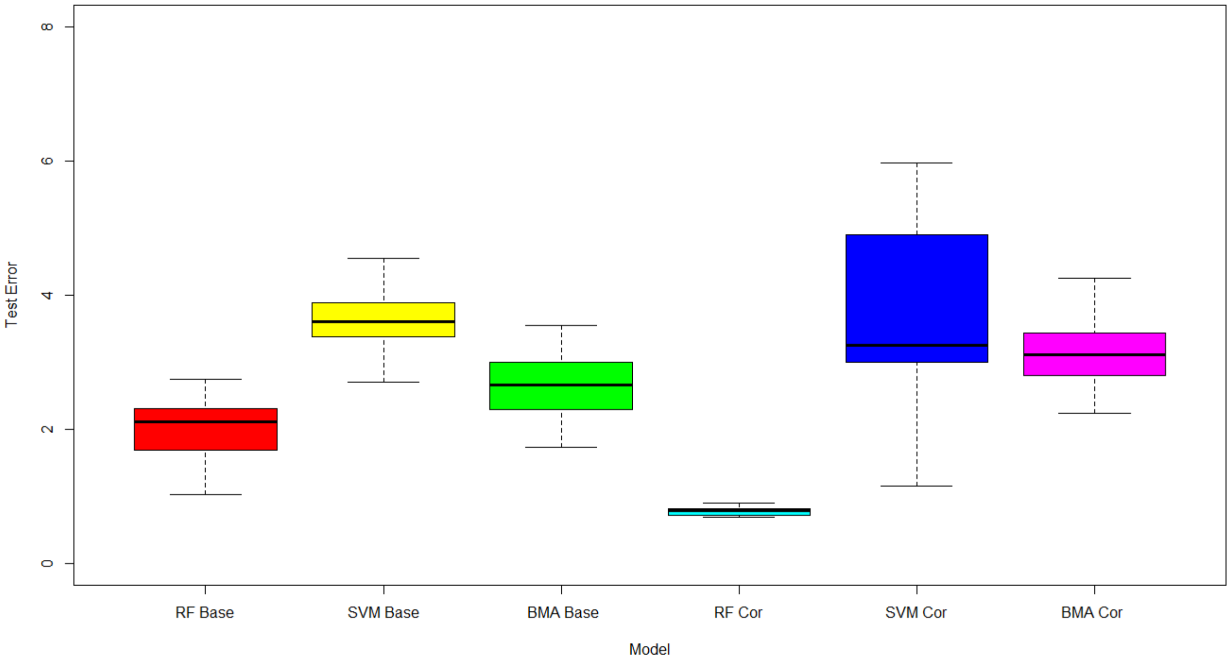


Figure 5.3: Comparative Boxplots (n=30)

In the figures above,the model name are abbreviated as follows:

- RF Base - Random forest baseline model
- SVM Base - Support Vector Machine (regression) baseline model
- BMA Base - Bayesian Model Averaging basleline model
- RF Cor - Random Forest correlated model
- SVM Cor - Support vector machine (regression) correlated model.
- BMA Cor - Bayesian Model Averaging correlated model.

5.1.3 Discussion

As we can see in both the figures above, the random forest model for the correlated data set has the minimum test error. This indicates that it is the best model of all models. This also reinforces our results in Table 4.2 and Table 4.3 indicating a higher percentage to choose true predictors of all the other models. One intriguing observation here is that, the test error for the random forest baseline model increase for larger sample size n = 300 in comparison to the smaller sample size n=30. Also, the variation in the test error for the support vector machine (regression) with a sample size n=30 is quite high in comparison to n=300. This could indicate model instability in SVM for a smaller sample size.

Chapter 6

Conclusion & Future Work

6.1 Conclusion

In this study, I investigated how variable importance measures in Random forest and SVM can be combined with recursive elimination and compared it with Bayesian Model Averaging. The effect of correlation on the predictors is on-going research for several years and is of great interest. Thus, I crammed variable importance and variable selection using these methods in the presence of correlation. Since a lot of research already exists on real data sets, I experimented on experimental data by simulating correlated data and modulating the signal-noise-ratio. The results betoken that the Recursive feature elimination with the Random forest method outperforms the other two methods. Also, the models perform better for n>p corresponding to n<p.

6.2 Future Work

This study is restricted to the linear regression framework only. Thus, this study can be extended considerably in quite assorted areas. This can also be inquired for non-linear regression problems.Besides,it can be repeated for classification problems using correlation types like kendall or spearman's correlation. Several other variable importance and variable selection methods like Lasso Ridge can also be incorporated in the study.Generalizing this idea of using the recursive feature elimination under different frameworks might entail thorough thought process and deep comprehension of all the conjectures. It would likewise be fascinating to compare the predictive performance metric RMSE with additional remarkable metrics like Mean Absolute error etc. (Botchkarev, Alexei,(2018))

Appendix A

First Appendix

I had conducted a preliminary Study to understand Variable importance measure on different methods:

Figure A.1,A.2,A.3 represents variable selection in Lasso,Ridge and Elastic net using the R package glmnet. Studies have found that in presence of correlation,the penalized shrinkage methods select only one variable and disregards all the other variables.(Bühlmann, P., Rütimann, P., van de Geer, S., Zhang, C.-H (2013)).For this reason, I did not choose the shrinkage methods for this comparative study.But it would be interesting to combine the shrinkage methods with recursive feature elimination and then analyse the variable selection in the presence of correlation.

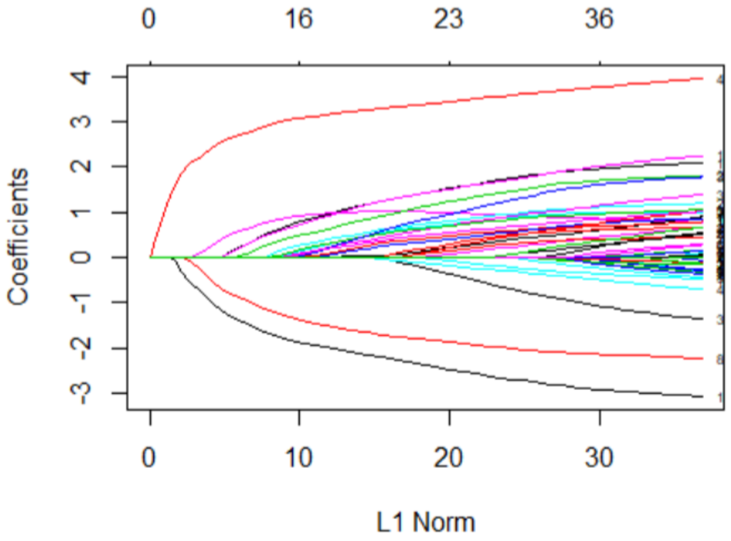


Figure A.1: Lasso Variable Importance

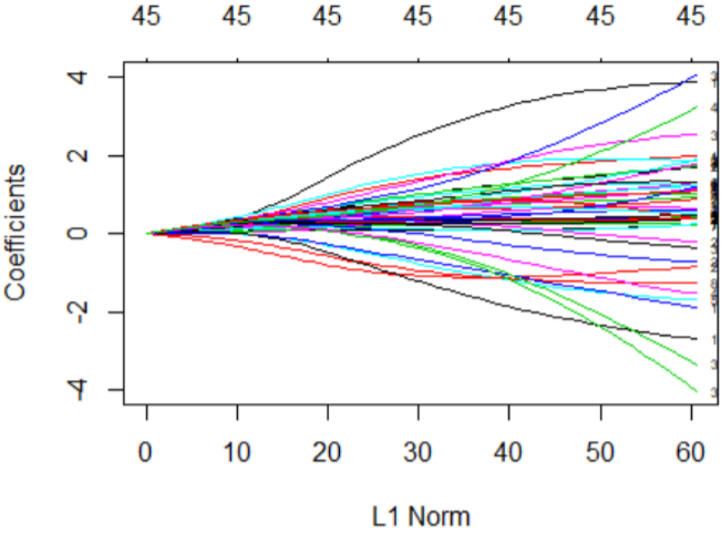


Figure A.2: Ridge Variable Importance

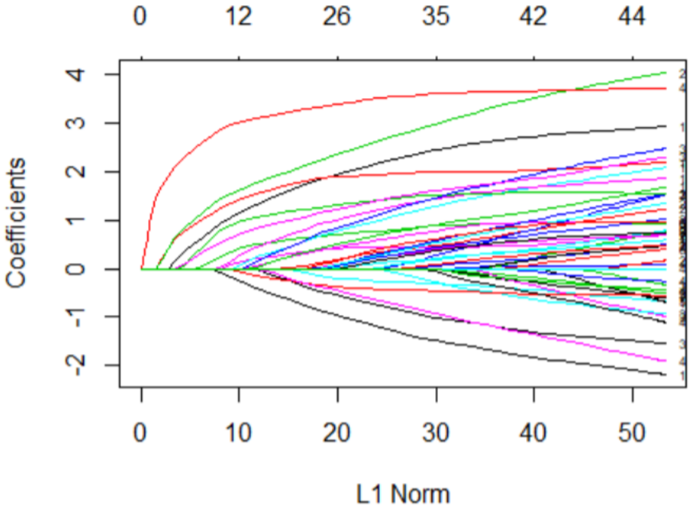


Figure A.3: Elastic net Variable Importance

Figure A.4 represents variable selection in Random forest using R package RandomForest.

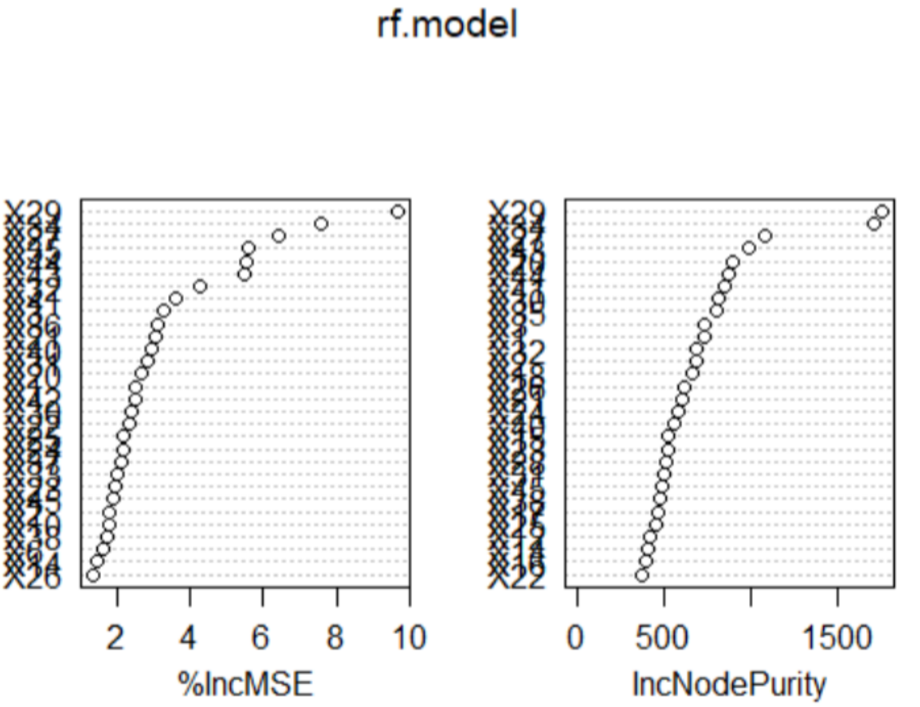
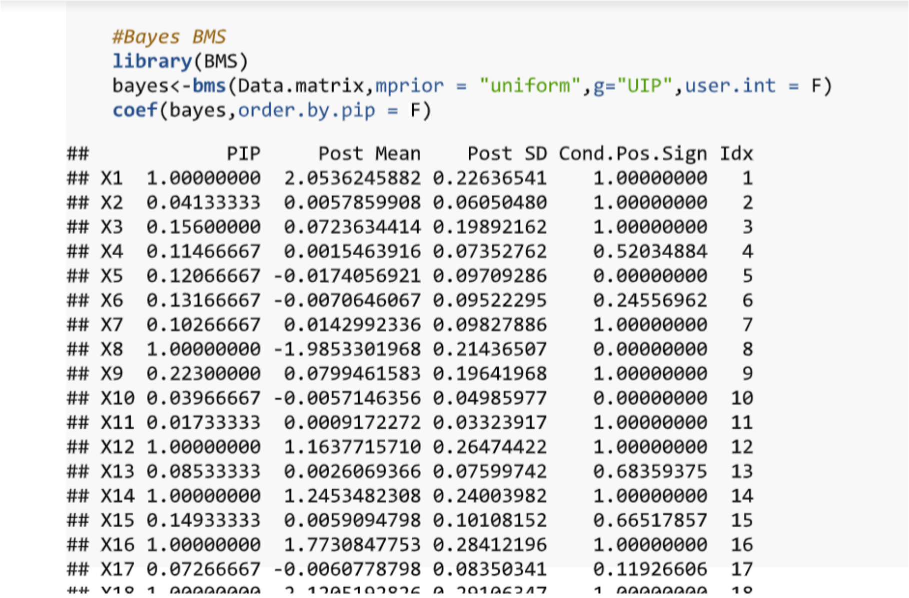


Figure A.4: Random Forest Variable Importance

Figure A.5 presents the results from a bayesian model averaging. Here the column PIP denotes the posterior inclusion probability(PIP). PIP has been used for variable selection in our study.



```
#Bayes BMS
library(BMS)
bayes<-bms(Data.matrix,mprior = "uniform",g="UIP",user.int = F)
coef(bayes,order.by.pip = F)
```

```
##            PIP     Post Mean     Post SD Cond.Pos.Sign Idx
## X1  1.00000000  2.0536245882 0.22636541    1.00000000   1
## X2  0.04133333  0.0057859908 0.06050480    1.00000000   2
## X3  0.15600000  0.0723634414 0.19892162    1.00000000   3
## X4  0.11466667  0.0015463916 0.07352762    0.52034884   4
## X5  0.12066667 -0.0174056921 0.09709286    0.00000000   5
## X6  0.13166667 -0.0070646067 0.09522295    0.24556962   6
## X7  0.10266667  0.0142992336 0.09827886    1.00000000   7
## X8  1.00000000 -1.9853301968 0.21436507    0.00000000   8
## X9  0.22300000  0.0799461583 0.19641968    1.00000000   9
## X10 0.03966667 -0.0057146356 0.04985977    0.00000000  10
## X11 0.01733333  0.0009172272 0.03323917    1.00000000  11
## X12 1.00000000  1.1637715710 0.26474422    1.00000000  12
## X13 0.08533333  0.0026069366 0.07599742    0.68359375  13
## X14 1.00000000  1.2453482308 0.24003982    1.00000000  14
## X15 0.14933333  0.0059094798 0.10108152    0.66517857  15
## X16 1.00000000  1.7730847753 0.28412196    1.00000000  16
## X17 0.07266667 -0.0060778798 0.08350341    0.11926606  17
## X18 1.00000000  2.1205192826 0.29106347    1.00000000  18
```

Figure A.5: BMA

The package BMS also has a nice function to visualize the most important features that are obtained by averaging over a set of models.

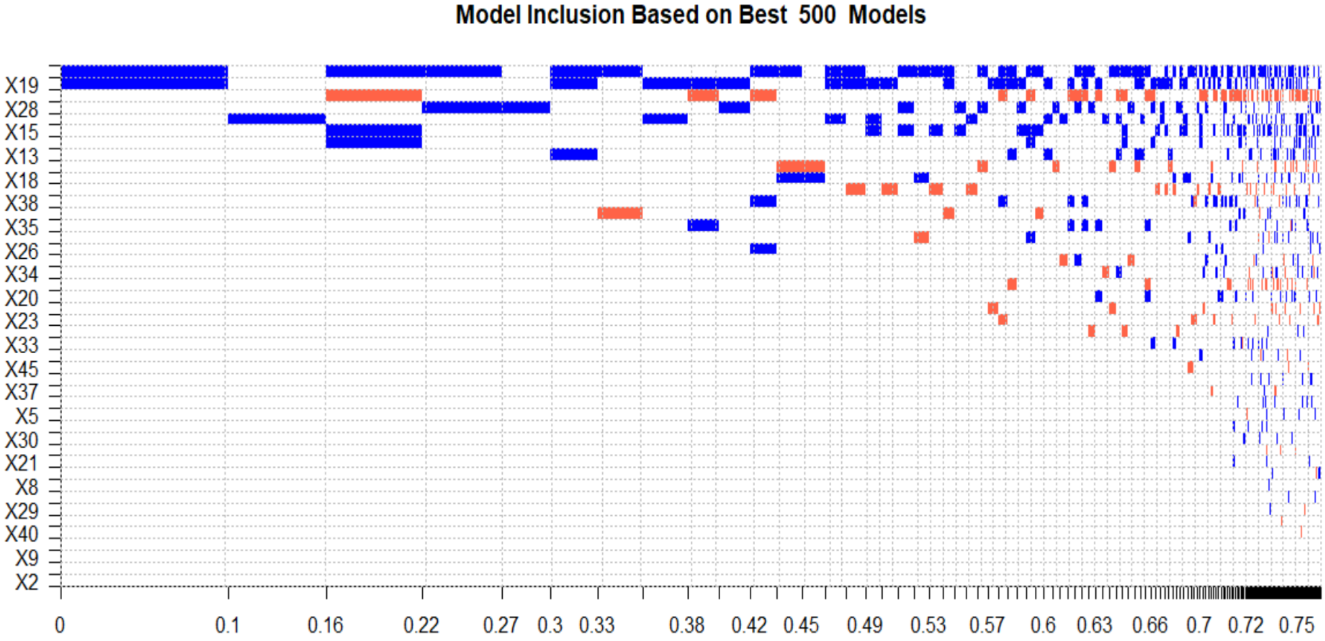
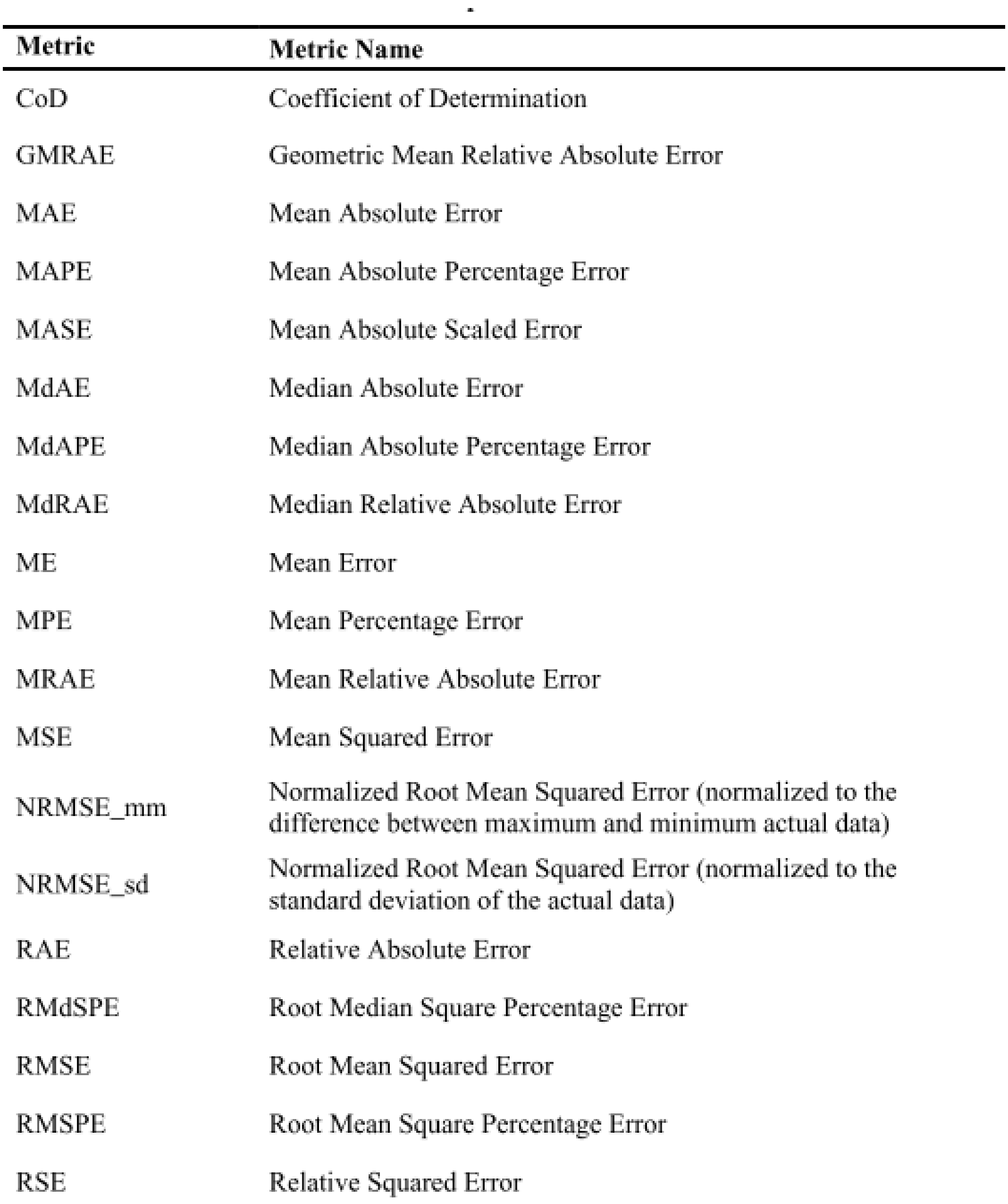


Figure A.6: BMS Model Ranking

Table A.7 is an list of different Model performance metrics that exists.



Metric	Metric Name
CoD	Coefficient of Determination
GMRAE	Geometric Mean Relative Absolute Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MdAE	Median Absolute Error
MdAPE	Median Absolute Percentage Error
MdRAE	Median Relative Absolute Error
ME	Mean Error
MPE	Mean Percentage Error
MRAE	Mean Relative Absolute Error
MSE	Mean Squared Error
NRMSE_mm	Normalized Root Mean Squared Error (normalized to the difference between maximum and minimum actual data)
NRMSE_sd	Normalized Root Mean Squared Error (normalized to the standard deviation of the actual data)
RAE	Relative Absolute Error
RMdSPE	Root Median Square Percentage Error
RMSE	Root Mean Squared Error
RMSPE	Root Mean Square Percentage Error
RSE	Relative Squared Error

Figure A.7: Error Metrics:Botchkarev, Alexei(2018)

Appendix B

Selectec R Code

```
# Simulating Correlation induced data
# Step 1: Generating covariance matrix
generate.cov <- function ( p =15, rho =0.0, tau =1, nn =60,
                           homogeneous = TRUE )
{
  # p is the dimensionality of the data
  # rho is the correlation coefficient
  # tau is the overall level of relatedness
  # nn is the sample size
  library ( MASS )
  pp <- 1: p
  m1p <- matrix (rep (pp , p ) , ncol =p , byrow = T )
  mp1 <- matrix (rep( pp , p ) , ncol =p , byrow = F )
  M.het <- abs( m1p - mp1)
  M.hom <- rep(1,p ) %*% t(rep(1,p ) ) # creating a matrix of all 1   s   .

  #%*% is used for matrix multiplication
  # Creating a covariance matrix .
  if ( homogeneous )
  {
    Sigmap <- rho ^ M.hom ;
    diag ( Sigmap ) <- rep(1,p )
  }
  else
  {
    Sigmap <- rho ^ M.het
  }

  12

  Sigmap <- tau * Sigmap
  return ( Sigmap )
}
# Simulating the three covariance matrices using the function created above
covariance_matrix_1 <- generate.cov( rho =0.0)
covariance_matrix_2 <- generate.cov( rho =0.45)
covariance_matrix_3 <- generate.cov( rho =0.9)
# Creating the Zero matrices required to create the Block Covariance matrix
     ( Intrinsic matrix ).
zero_1 <-mat.or.vec (30,15)
zero_2 <-mat.or.vec (15,15)
# Appending the matrices to create the covaraince matrix .
mat1 <- rbind ( covariance_matrix_1, zero_1)
mat21 <-rbind ( zero_2, covariance_matrix_2)
mat2 <-rbind (mat21, zero_2)
mat3 <-rbind ( zero_1, covariance_matrix_3)

# Creating the Block Covariance Matrix
Block_covariance_matrix <- cbind( mat1,mat2,mat3)

# Step 2: Generating data using the block covaraince matrix created above
p =45
X <- data.frame ( mvrnorm ( nn , mu =(1/1: p ) , Sigma = Block_covariance_
    matrix ) )
Y <- 1+ 2* X [ ,1] + X [ ,3] - 2.5 * X[ ,6] - 2* X [ ,8] + 7/8* X [ ,12]+ 2
    /5 * X [ ,13] + X [ ,14] + X [ ,16] + 2* X [ ,18] -3.0* X [ ,19] +
  X [ ,20] +2* X [ ,21] + X [ ,23] + 1.5* X [ ,30] - X [ ,31] + X [ ,32] +
  X [ ,33]+ 2* X [ ,34] - 0.25* X [ ,35]+ 0.85* X [ ,36] +4* X [ ,44]+ 3*
      rnorm ( nn )

Data.matrix <- data.frame (Y,X)
# Checking the correlation plot for the block covaraince matrix
```

```
corrplot (cor ( Block_covariance_matrix ) )
# checking the correlation plot for the Data matrix generated by inducing
    correlation .
corrplot (cor ( Data.matrix ) )


model.x <- as.matrix ( Data.matrix [ , -1])
model.y <- as.matrix ( Data.matrix [ ,1])
# Scatter Plots of the independent varaibles .
featurePlot ( x = Data.matrix [ , 2:46] ,
              y = Data.matrix [ ,1] ,
              plot = "␣scatter␣",
              type = c("p", "␣smooth␣") ,
              ## Add a key at the top
              auto.key = list ( columns = 3) )

#The relationship is mostly linear with almost all variables

# Partioning data into train and test datasets
set.seed (4545)
trainIndex <- createDataPartition ( Data.matrix $Y , p=0.8,list =
                                      F , times = 1)
Train_dataset <- Data.matrix [ trainIndex ,]
Test_dataset <- Data.matrix [ - trainIndex ,]
train.x <- as.matrix ( Train_dataset [ , -1])
train.y <- as.matrix ( Train_dataset [ ,1])
test.x <- as.matrix ( Test_dataset [ , -1])
test.y <- as.matrix ( Test_dataset [ ,1])



##Function for Random Forest Recursive Feature Elimination Algorithm


RF_RFE <- function(x=train.x.base,y=train.y.base,t.x=test.x.base,t.y=test.y
    .base)

{
```

```
remaining_features <- colnames(x)
Variables <- 0
mse <- Inf
Top_10 <- NULL
selected_predictors <- NULL
number_of_variables <- NULL
while(length(remaining_features) >= 3)
{

  Rf_reg <- randomForest(x[,remaining_features],y,ntree=1000,importance =
       T)
  #Prediction
  rf_predictions <- predict(Rf_reg,t.x[,remaining_features])
  rf_predictions <- as.data.frame(rf_predictions)
  rf_predictions <- cbind(rf_predictions,t.y)
  colnames(rf_predictions) <- c("prediction","test_y")
  rf_predictions <- as.data.frame(rf_predictions)
  rf_predictions <- rf_predictions %>% mutate(Test_error = (test_y-
      prediction)^2)
  #calculating relative error
  rf_predictions <- rf_predictions %>% mutate(Relative_test_error = abs((
      Test_error/test_y)))
  #Calculating mean Sqaured error
  Mean_Squared_error <- sum(rf_predictions[,3])/nrow(rf_predictions)
  #calculating mean of relative test error #Note that this is not error
      is not squared #NOT RMSE
  #RMSE can be calculted if required
  Mean_Relative_Test_error <- sum(rf_predictions[,4])/nrow(rf_predictions
      )
  #Calculate RMSE
  RMSE = sqrt(Mean_Relative_Test_error)
  #Variable Importance and eliminating features
  w <- importance(Rf_reg,type=1)
  w <- w[order(w[,1]),,drop=F]
  #Extracting top variables for minimum mse
  if(Mean_Squared_error <= mse)
  {
    mse <- Mean_Squared_error
```

```
      Top_10 <- tail(w,10)
      selected_predictors <- w
      number_of_variables = nrow(w)
      Best_subset_MSE <- rf_predictions[,3]
    }
    d <- rownames(w)
    w <- cbind(d,w)
    elim_feature <- w[1,1]
    #eliminating feature with lowest weight
    remaining_features <- remaining_features[remaining_features != elim_
       feature]
    Metrics <- cbind(nrow(w),Mean_Squared_error,RMSE,Mean_Relative_Test_
       error)
    Variables <- rbind(Variables,Metrics)
  }
  #best_subset <- list(number_of_variables,Top_5)
  #return(best_subset)
  Final_features <- (remaining_features)
  number_of_variables <- print(paste0("The␣best␣subset␣size␣is:", number_of
     _variables))
  #Top_5 <- print(paste0("The Top 10 Variables are : ",Top_5))
  mylist <- list(Variables,remaining_features,number_of_variables,selected_
     predictors,Top_10,mse,Best_subset_MSE)
  return(mylist)
  #returning the relative error values for the boxplots data
}

# Function to run the experiment over 100 Iterations Using the function
   created above.

Iterated <- function ()

{
  c <- NULL
  MSE = matrix(0,nrow = p-2,ncol = 100)
  for ( i in 1:100)
  {
    #Generating data for the Baseline Model
    p = 45
    nn = 300

    X.base <- data.frame(mvrnorm(nn,mu =(1/1:p),Sigma = diag(p)))
    Y.base <- 1 + 2* X.base[,1] + X.base[ ,3] - 2.5 * X.base[ ,6] -
      2* X.base[ ,8] + 7/8* X.base[ ,12] + 2/5 * X.base[ ,13] + X.base [ ,1
          4] + X.base[ ,16] + 2* X.base[ ,18] -3.0* X.base[ ,19]+
      X.base[ ,20] +2* X.base[ ,21] + X.base [ ,23] + 1.5* X.base[ ,30]
    - X.base[ ,31] + X.base[ ,32] + X.base[ ,33]+ 2* X.base[ ,34] - 0.25* X
        .base[ ,35]+ 0.85* X.base[ ,36] +4* X.base[ ,44]+  3* rnorm ( nn )
    Data.matrix.base <- data.frame ( Y.base, X.base )

    # Partioning data into train and test datasets
    set.seed (4545)
    trainIndex.b <- createDataPartition ( Data.matrix.base$Y.base ,
                                           p =0.8,list = F , times = 1)
    Train_dataset.b <- Data.matrix.base[ trainIndex.b,]
    Test_dataset.b <- Data.matrix.base[- trainIndex.b,]
    train.x.base <- as.matrix( Train_dataset.b[ , -1])
    train.y.base <- as.matrix(Train_dataset.b[ ,1])
    test.x.base <- as.matrix (Test_dataset.b[ , -1])
    test.y.base <- as.matrix ( Test_dataset.b[ ,1])

    a <- RF_RFE (x = train.x.base,y = train.y.base,t.x = test.x.base,t.y =
        test.y.base)
    MSE[,i] = (a[[1]][-1,3])
    b <- row.names ( a [[4]])
    c <- qpcR ::: cbind.na(c, b)
    print(i)
  }

  values = list(MSE,c)
  return (values)
}
```

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31]

Bibliography

[1] André Elisseeff Guyon Isabelle De. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[2] Song Jingwen Wei Pengfei, Lu Zhenzhou. Variable importance analysis: A comprehensive review. *Reliability Engineering and System Safety*, 142:399–432, 2015.

[3] Michel Bertrand Saint-Pierre Philippe Gregorutti, Baptiste. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, 2017.

[4] A. Hall, Mark. Correlation-based feature selection for machine learning. *Diss. The University of Waikato*, 322(10):1–5, 1999.

[5] Anne-Laure Kneib Thomas Augustin Thomas Zeileis Achim Strobl, Carolin Boulesteix. Conditional variable importance for random forests. *BMC bioinformatics*, 9, 2008.

[6] James D. Nicodemus, Kristin K. Malley. Predictor correlation impacts machine learning algorithms: Implications for genomic studies. *Bioinformatics*, 25(15):1884–1890, 2009.

[7] Mark F J Steel. Bayesian model averaging and forecasting. *Department of Statistics,University of Warwick, U.K.*, 44, 2011.

[8] Brkić K. Bogunović N. Jović, A. A review of feature selection methods with applications. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, pages 1200–1205, 2015.

[9] Annie Xue, Fei Qu. Variable selection for highly correlated predictors. 2017.

[10] Muralidhara S Radha R. Removal of redundant and irrelevant data from training datasets using speedy feature selection method. *International Journal of Computer science and Mobile Computing*, 5(7):359–364, 2016.

[11] Dasgupta Abhijit Malley James D. Molloy Anne M. Mills James L. Brody Lawrence C. Stambolian Dwight Bailey-Wilson Joan E. Szymczak Silke, Holzinger Emily. r2vim: A new variable

selection method for random forests in genome-wide association studies. *BioData Mining*, 9(1):1–15, 2016.

[12] Sander Oliver Lengauer Thomas Altmann André, Toloşi Laura. Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[13] Chunrui Zhang Yusen Liu Jiaguo Yu Bin Liu Xiaoping Dehmer Matthias Liu Shenghui, Xu. Feature selection of gene expression data for cancer classification using double rbf-kernels. *BMC Bioinformatics*, 19(1):1–14, 2018.

[14] Szilard Steineck Gunnar Dickman Paul W. Genell, Anna Nemes. Model selection in medical research: A simulation study comparing bayesian model averaging and stepwise regression. *BMC Medical Research Methodology*, 10, 2010.

[15] Srimanta Patranabis Dipak Chandra Basak, Debasish Pal. Support vector regression. *Journal of Machine Learning Research*, 11:203–224, 2007.

[16] Bernhard S C H Smola, Alex J Olkopf. A tutorial on support vector regression . *Journal of Machine Learning Research*, 2004.

[17] Chapelle Ponti Poggio Vapnik Weston, Mukherjee. Feature selection for svms. *Advances in Neural Information Processing Systems*, 13, 2000.

[18] Yichao Wang Lan Zhang, Xiang Wu. Variable selection for support vector machines in high dimensions. 2002.

[19] Clarissa Vegas Esteban Oller Josep M. Reverter Ferran Sanz, Hector Valim. Svm-rfe: Selection and visualization of the most relevant features through non-linear kernels. *BMC Bioinformatics*, 19, pages =, 2018.

[20] Roger Haake Anne Yacci, Paul Gaborski. Feature selection of microarray data using genetic algorithms and artificial neural networks. *Intelligent Engineering Systems through Artificial Neural Networks*, 2009.

[21] Andrew C. Merrill. Investigation of variable importance measures within random forest. *Utah State University*, 2009.

[22] Youping Chen Huann-sheng Tao Lin Sha Qiuying Chen Jun Tsai Chung-Jui Zhang Shuanglin Jiang, Hongying Deng. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 12, 2004.

[23] Bradley C. McCarthy Andrew J. Greenwell, Brandon M. Boehmke. A simple and effective model-based variable importance measure. 2018.

[24] Rahul Awad, Mariette Khanna. Efficient learning machines theories,concepts and applications for engineers and system designers. 2015.

[25] R. R. Chai, T. Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific Model Development*, 2014.

[26] Alexei Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. 2018.

[27] Mary M. Murray, Kim Conner. Methods to quantify variable importance: Implications for the analysis of noisy ecological data. *Ecology*, 2009.

[28] James M. Johnson, Jeff W. LeBreton. History and use of relative importance indices in organizational research. *Organizational Research Methods*, 2004.

[29] Matthias Emmert-Streib, Frank Dehmer. High-dimensional lasso-based computational regression models: Regularization, shrinkage, and selection. *Machine Learning and Knowledge Extraction*, 2019.

[30] Steffen Kuhn, Stefan Neumann. Learning methods for nmr prediction. *Not Known*, 2008.

[31] Bradley C. McCarthy Andrew J. Greenwell, Brandon M. Boehmke. A simple and effective model-based variable importance measure. 2019.