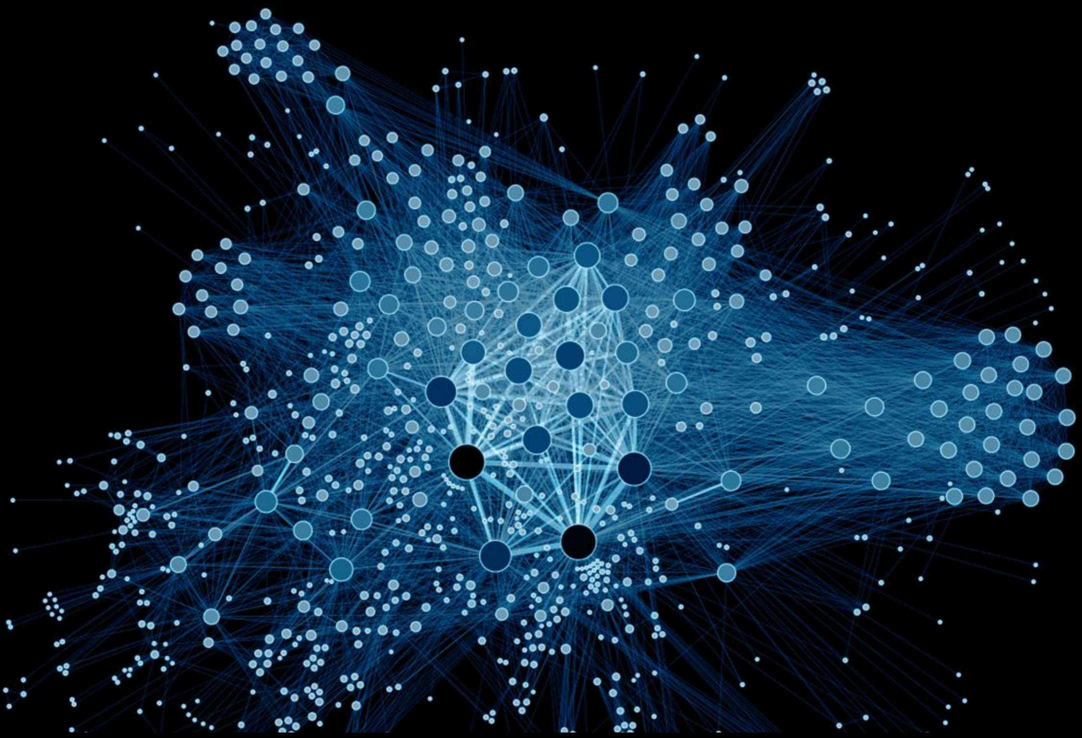
Health Evaluation and Linkage to Primary Care (HELP) Data Analysis



Farnoush Rashed	– 20%
Hemik Parikh	– 20%
Mayank Jaggi	– 20%
Rohit Lilhare	– 20%
Soniya Deshpande	– 20%

Contents

Table of Figures

1. Introduction

The primary purpose of this report is to identify attributes which affect enrollment into a healthcare plan. The Health Evaluation and Linkage to Primary Care study was a clinical trial for adult inpatients recruited from a detoxification unit. The study was based on how linkage to Primary care can be achieved and what attributes in life are responsible for deciding to join Primary care. Patients with no primary care physician were randomized to receive a multidisciplinary assessment and a brief motivational intervention or usual care, with the goal of linking them to primary medical care.

The study focuses on finding out key parameters contributing to the decision making of primary care and relating them to real life explanations. Data were collected by interview assessment of predisposing, enabling, and illness variables. Linkage was defined as self-report of at least one visit with a primary care clinics during follow-up.

The data collection was not part of the study. Questions selected to participate also not under review by this study, this study identifies Data analysis methods which can be used to determine critical predictors for the response of Link to the healthcare plan. The author of the dataset and whose contribution the team would like to acknowledge Dr. Vincent Arel-Bundock - a political scientist at the University of Montreal.

2. Data selection and Description

With the task set to find a Healthcare data with more than 100 attributes, the team researched multiple sources for data. The team was astounded by the work Dr. Arel-Bundock has done to produce 1147 data sets for the general public to perform Data analysis and find the underlying trends with vital predictors.

The dataset selected by the team consisted of 434 columns and 1472 rows. The data is collected based on the response to the questionnaire from different patients. These include information regarding various aspects in patient's life including medical history, present complaints, allergies, psychiatric health, habits related to drinking and drug usage and other details like financial condition and marital status. Each predictor has a question associated with it and separately cumulative predictors have been calculated which calculates the score of a section of questions. (e.g. Drugs and Drinking Habits). Appendix has a file attached where each column is described according to the question that it represents.

2.1 Response Selection

First major task was to identify a response variable. We chose response variable as enrollment to Primary Health care which opened up the avenue to identify factors and real-life scenarios which influences a person's decision to enrollment of Healthcare system. Presence of the response variable in the dataset makes this model a supervised learning problem.

2.2 Task

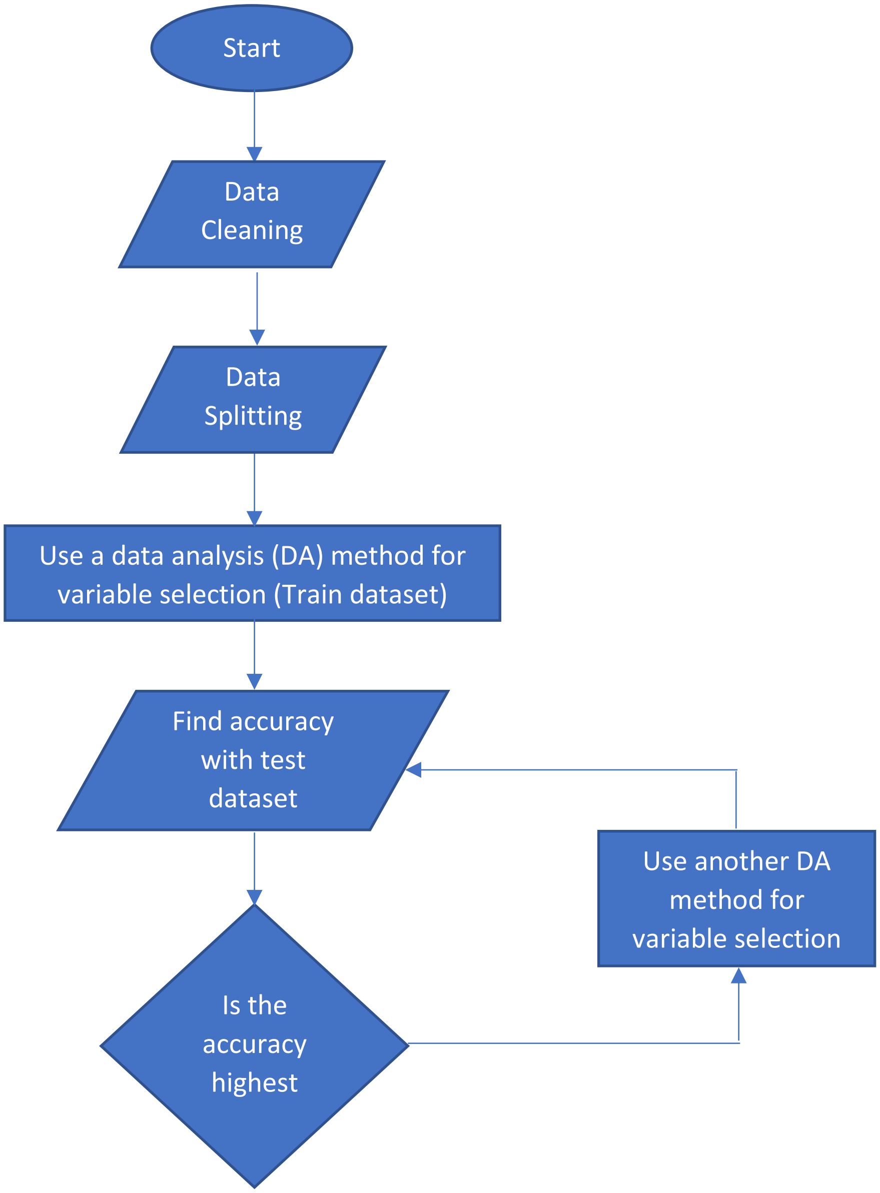
The aim of the analysis is to find whether a person should opt for healthcare plan. The data we have is a very comprehensive data as it considers various aspects in a person's life such as Demographics, Medical Status, Health Care use etc. which would affect this decision. As the response is yes / no this is a classification problem.

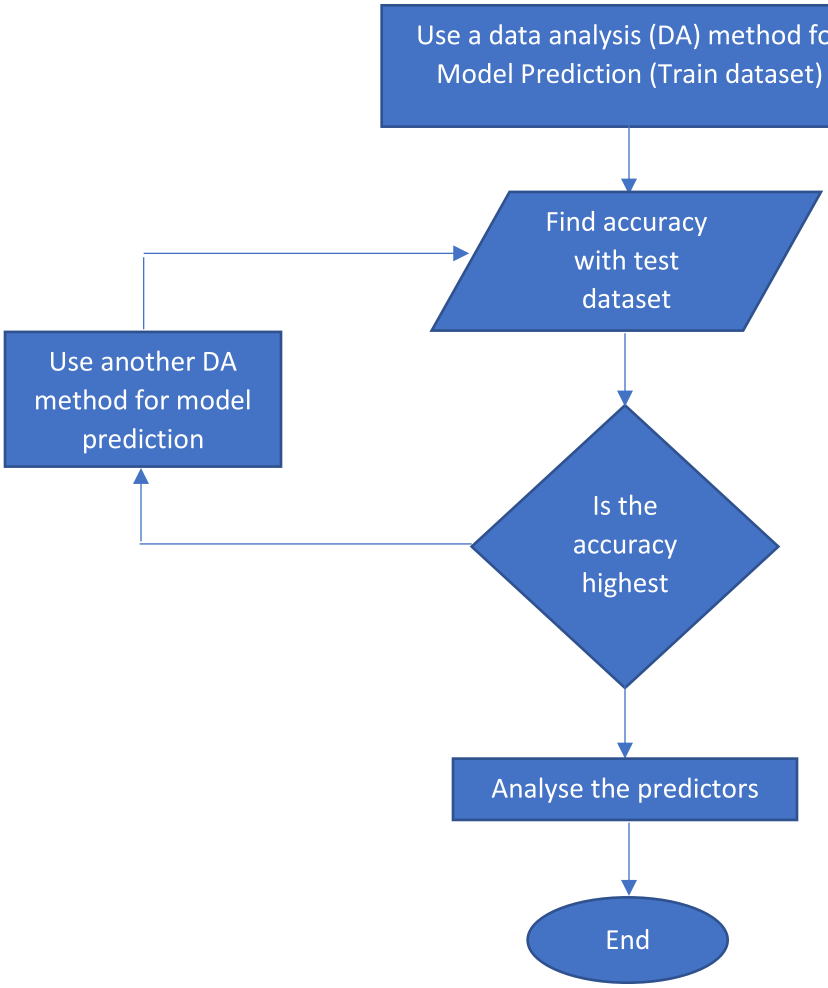
2.3 Approach

As there are currently 416 predictors so the interpretation would be difficult. Thus, the first part of analysis focuses on finding critical factors in this decision. This would be done by various variable selection methods mentioned in chapter 3. Based on the shortlisted predictors which have major role in the decision of opting for health care the model can be fit to achieve more accuracy.

To come up with a best possible solution we identified two loops. Since the data has huge number of attributes it is of utmost importance to identify key parameters which are relevant to response. To perform variable selection, we identified five selection methods and performed each of them. For the same amount of accuracy, we chose the model with least number of predictors.

2.4 Flowchart


Start
Data Cleaning
Data Splitting
Use a data analysis (DA) method for variable selection (Train dataset)
Find accuracy with test dataset
Is the accuracy highest
Use another DA method for variable selection


Use a data analysis (DA) method for Model Prediction (Train dataset)
Find accuracy with test dataset
Use another DA method for model prediction
Is the accuracy highest
Analyse the predictors
End

3. Implementation Method

3.1. Data Cleaning

Initially the data set was checked for inconsistencies. These include columns/ rows with missing data, columns with has no variance or very less variance (constant data). Such columns were removed using following commands. We calculated number of NAs in every column and every row with following code.

```
rowSums(is.na(HELPfull_Original))

colSums(is.na(HELPfull_Original))
```

After finding out the number at looking at the distribution we removed all the columns having more than 450 NA values, and after the operation we removed each row with having more than 3 NA values.

```
colvar<-colnames(HELPfull_Original)[colSums(is.na(HELPfull_Original))>450]

HELPFUL <- HELPfull_Original[,!names(HELPfull_Original)%in%colvar]
```

```
delete.na <- function(DF, n=3) { DF[rowSums(is.na(DF)) <= n,]}

HELPFUL1<-delete.na(HELPFUL)
```

We used a custom function to perform row reduction. After removal of rows and columns with NA value, we identified columns with less variability in the data and predictors which were "obvious" for the value of response. For example, a predictor which asks whether the subject has a regular doctor, for that predictor the distribution is highly correlated and hence these predictors needs to be removed from the data set. The table output is given below.

```
Var1 Var2 Freq


1    0    0  383
2    1    0  109
3    0    1    0
4    1    1  350
```

After removal of columns and rows the final dataset consists of 832 rows and 416 attributes including the response variable.

This data is mix of quantitative and qualitative predictors.

73 predictors are qualitative while other are quantitative. Out of which attributes marked in Red have no variance in their value i.e. only 1's or 0's and attributes marked in Green have very low variance. For the reason of better accuracy as well better interpretation.

C1M
E14F
H12_30
H12_RT
H9_PRB
H12_PRB

E8A3
E14E
H6_RT
H9_30
H9_RT
H11_RT

S1B
S1D
H6_PRB
H11_PRB

The following code was used to identify the variability of predictors.

```
stats <- apply(NewDataset,2,var)
```

Also, the output RCT_LINK is qualitative with two classes. Following code is run to identify columns of qualitative predictors as classes and not numeric value. The final data matrix is ISEN613 which is used for further analysis.

```
attach(NewDataset)
a=c("TIME","NUM_INTERVALS","INT_TIME1","A15A","A15B","A15C","D3","H1_30","H2_30","H3_30
","H4_30", “H5_30", "H6_30", "H7_30", "H8_30", "H9_30", "H10_30", "H11_30", "H12_30",
"H13_30", "H15A","H15B",   "H16A", "H16B", "H17A", "H17B", "ALCQ_30", "RAWPF", "PF", "RP",
"RAWBP", "BP", "RAWGH", "GH", "RAWVT",   "VT", "RAWSF", "SF", "RAWRE", "RE", "RAWMH",
"MH", "HT","PCS","MCS","CES_D","C_MS","C_AU","C_DU",   "RAW_RE", "DEC_RE", "RAW_AM",
"DEC_AM", "RAW_TS","DEC_TS","PHYS","PHYS2","INTER","INTRA","IMPUL",    "IMPUL2",
"SR","CNTRL","INDTOT","INDTOT2","PSS_FR","PSS_FA","DRUGRISK","SEXRISK","TOTALRAB",
   "RABSCALE","ANY_VIS","ANY_VIS_CUMUL")
abc=NewDataset[,!names(NewDataset)%in%a]
#View(abc)
abc[]=lapply(abc[],factor)
sapply(abc,class)
ISEN613=data.frame(TIME,NUM_INTERVALS,INT_TIME1,A15A,A15B,A15C,D3,H1_30,H2_30,H3_30,
H4_30,H5_30,H6_30,H7_30,H8_30,H9_30,H10_30,H11_30,H12_30,H13_30,H15A,H15B,H16A,H16B
,H17A,H17B,ALCQ_30,RAWPF,PF,RP,RAWBP,BP,RAWGH,GH,RAWVT,VT,RAWSF,SF,RAWRE,RE,RAW
MH,MH,HT,PCS,MCS,CES_D,C_MS,C_AU,C_DU,RAW_RE,DEC_RE,RAW_AM,DEC_AM,RAW_TS,DEC_
TS,PHYS,PHYS2,INTER,INTRA,IMPUL,IMPUL2,SR,CNTRL,INDTOT,INDTOT2,PSS_FR,PSS_FA,DRUGRISK,
SEXRISK,TOTALRAB,RABSCALE,ANY_VIS,ANY_VIS_CUMUL,abc)
```

3.2. Variable Selection

The final dataset contains 404 predictors. To improve prediction accuracy and interpretability we decided to implement variable selection methods. This helped to identify and exclude irrelevant predictors (predictors with less impact on response). Prediction accuracy is checked for each method. For predicting test accuracy, the data is split as 70% training data and 30% test data. This split along with set seed is kept constant for each method for easy comparison among the available methods. Later the predictors identified from variable selection methods are used for fitting the data with different models for classification.

3.2.1. LASSO

LASSO is a type of the shrinkage methods which is used for variable selection. The model is fit on the complete data set. The penalty term is added which includes tuning parameter λ. This term shrinks the coefficient estimates towards zero and reduce the variance. With LASSO few of the coefficients are shrunk to zero, thus we get a model which is simple and easy for interpretation with lesser predictors.

Result: LASSO was run on the whole data set obtained after data cleaning. Cross validation is done to obtain best λ (Tuning parameter). The accuracy of the model obtained is 42.35%. The predictors were identified for which the coefficients shrink to zero. There were 25 predictors with non-zero coefficients. Thus, through variable selection LASSO selected 25 predictors which have more impact on response.

Following graph shows the shrinkage of coefficients with variation in tuning parameter (L1Norm)

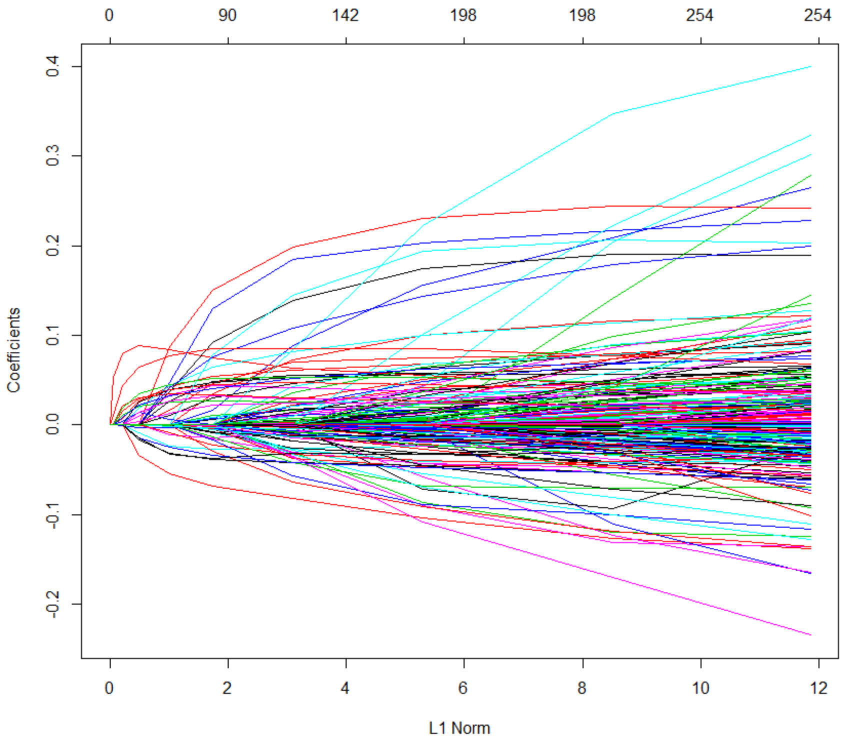


Figure 1 Coefficient vs L1 Norm

3.2.2. Random Forest

Decision trees could be used to predict the response. Decision trees are easy to interpret as they match human decision-making process. But the prediction accuracy is less with decision trees thus we are using random forest. Multiple trees are created but each tree is created using subset of total p predictors. We have selected m predictors which are equivalent to square root of p as the response is a categorical variable. This ensures that the trees are decorrelated. We have used variable importance plot for the dataset which shows decrease in Gini index. The predictors which have maximum decrease are most important parameters. We did not use bagging because there was no decorrelation which is necessary to get unbiased results.

3.2.3. Boruta

There have been numerical attempts at variable selection methods. One of the promising method has been Boruta, it gets rid of the bias created in Random Forest and performs multiple numerical analysis by working as a wrapper algorithm around random forest. It adds randomness to the given data by creating shuffled copies of all the features. Then, the random forest is trained on the extended dataset and applies a feature importance measure (which in case of classification problem is a decrease in Gini index) to evaluate the importance of each feature. A feature is removed if it has lower importance then the best of its shadow features. The iteration is continued till all the features are accepted or removed. The result is given in terms of Important and non-important features. [1] [2]

Boruta method was used here as an external method and as a replacement to Boosting which can reduce biased introduced by random forest and gives important variables. With different Seed combinations different attributes were identified in other methods as important ones. Hence it was necessary to reduce the biasness. Boruta gives us 25 variables which are deemed important. These variables have been used in further prediction methods.

The description of all 25 variables given by Boruta is given in Appendix A.

3.3. Modelling

As the response is categorical we are using classification methods to fit the data. Section 4.1 focusses on variable selection methods which helped to select predictors which have more effect on response. Based on those predictors different models are fit and the accuracy of the models are calculated. The model with best accuracy is chosen as the final model used for future prediction.

3.3.1. Logistics Regression

Logistic Regression is the appropriate regression analysis when the response (dependent variable) is dichotomous (binary). Logistic regression uses maximum likelihood to fit the logistic function and find the log-odds or logit. Therefore, the coefficients obtained from Logistic Regression is interpreted in terms of probability of occurring the response. [3] [4]

In this study, the dataset was split to training and test dataset to evaluate the performance of the method. The model was first fitted on the training data. Then prediction was conducted on the test data with the threshold set to 0.5. With logistic regression the accuracy is found to be 68%. Other details related to confusion matrix are in result table.

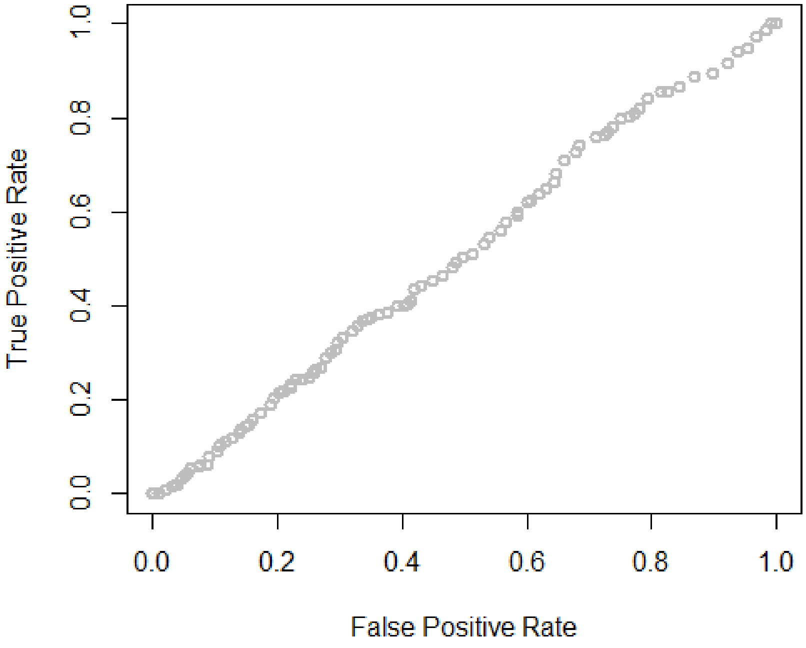


Figure 2 ROC Curve for Logistics Regression

3.3.2. LDA

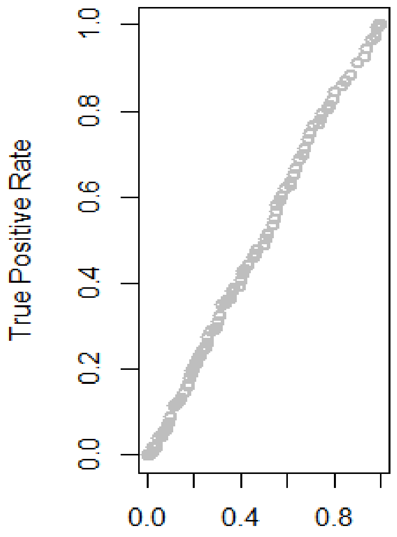


Figure 3 ROC curve for LDA

Linear Discrimination Analysis is dimensionality reduction technique with good class-separability to avoid overfitting. LDA assumes that each class is following a normal distribution with the equal variance. The advantages of LDA over Logistic Regression are as follows:

1- LDA appears to be more stable when the classes are well-separated.

2- LDA has a good performance on datasets with small number of observations.

3- LDA is popular in case of having more than two classes.

In this study, since the large number of observations was available, it could be assumed that they are following a normal distribution so LDA was performed.

3.3.3. QDA

Quadratic Discriminate Analysis is a classification algorithm which could be used if available classes of the response are more than two. In general, discriminate analysis models the distribution of predictors Xs in available predictor classes, and then make use of Baye's theorem to estimates the probability of response category for a given value of Xs. In Quadratic Discriminate Analysis (QDA) models are created using non-linear combination of the predictors.

Two main assumptions made in QDA modelling are:

1. Observations in each class drawn from gaussian distribution.
2. Predictor variable does not have a common variance across each of k levels in class Y.

Using above-mentioned assumptions QDA can capture the differing variance and provides more accurate non-linear classification decision boundary.

3.3.4. KNN

KNN is the simplest and best known non-parametric method. It provides more flexible and approach for performing regression analysis. It also provides an alternative approach against Bayes classifier, unattainable gold standard, to which compare classification results. It estimates the conditional probability for class j for a given X. and finally, KNN applies Bayes rule and classifies the test observation to a class with the largest probability.

The following graph shows the minimum MSE achieved at k=17. The k value was identified with for loop.

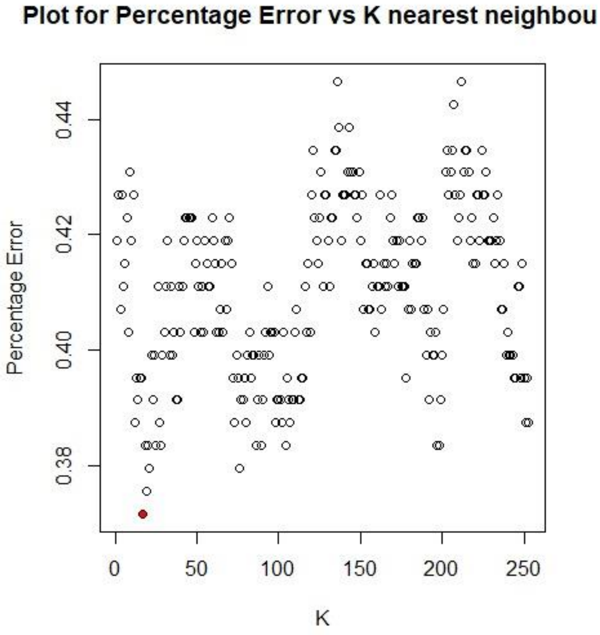


Figure 4 MSE vs K for KNN

3.3.5. Ridge Regression

Ridge Regression is a shrinkage method to reduce the size of predictors. This method is performed to analyze multiple regression data that faces multilinearity. Multilinearity results to obtain unbiased MSE but significantly increases the variance. Ridge Regression can solve this issue by adding some degrees of freedom. [5]

The disadvantage of this method is that it does not shrink the coefficients to zero, hence, the calculation would be computationally intensive. Also, it does not give model with less predictors hence it is difficult to interpret. [3]

The ridge regression is run on the reduced set of 25 predictors. The accuracy is 41.65%.

3.3.6. SVM

Support Vector Machine is a supervised machine learning algorithm which is mostly used for classification, although it can be performed on regression problems as well. This method enlarges the feature space by using kernels trick. Kernels are complex data transformation methods to find the optimal boundary the defined labels and responses. The advantage of this method is its ability to classify non-linear boundaries and capturing more complex relationship between data points. However, the calculation is computationally intensive compare to other methods.

Three types of kernels are used to transform data: linear, polynomial and radial. In this study, radial kernel is used to capture complex relation between data points. In addition, tune () function is performed to find the optimum values for gamma and cost.

In this study, radial kernel was used to enable the model to capture both linear and nonlinear boundaries.

3.3.7. Neutral Network

Neural network is a Machine learning process invented by Warren McCulloch and Walter Pitts in 1943. [6] Using threshold logic developed by Werbos (1975) [7] we use Neural net library to find the prediction model best with backpropagation. Neural network identifies all the possible linear models and trains itself to choose the best one among many. There have been numerous attempts of finding a best possible sub parameter selection that is number of nodes and layers, a paper by Stathakis (2009) [8] summarizes these approaches and results well however performing any of these methods is computationally extensive and requires significant mathematical knowledge. The rule of thumbs for Neural network suggests a better approach, selecting the number of nodes equal to the number of variables gives you significantly better result. Since more number of hidden layers are useful in case of nonlinear characteristics of response variable, data with high non-linearity requires more hidden layers. For our data set we increased number of hidden data layers till the time we reached improved accuracy.

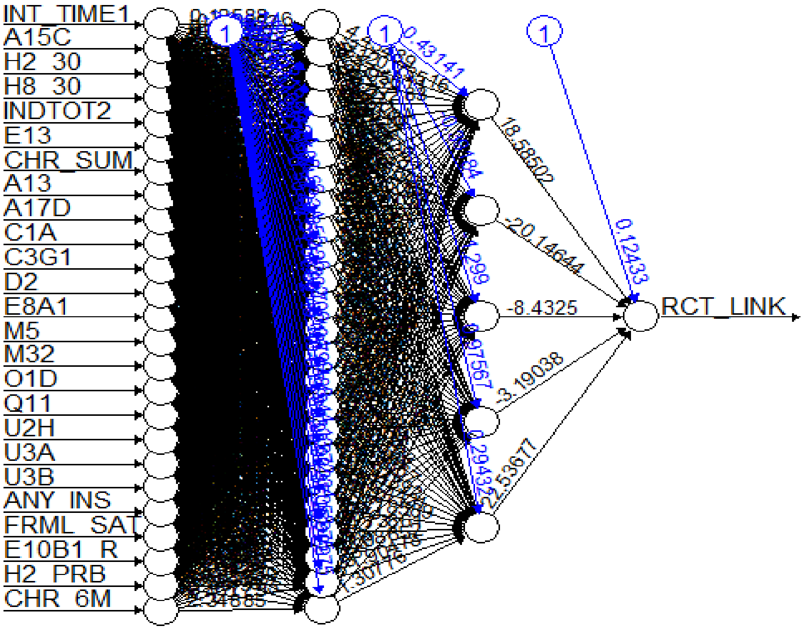


Figure 5 Neural Network

It is of utmost importance that the data is normalized before running neural network. We use the following code to normalize the data for calculation. Number of hidden layers are chosen to provide the highest accuracy and save computational time. As per the thumb rule number of intermediate nodes are selected as number of attributes being modelled.

The drawback of Neural network is that the method is a Blackbox and does not have a provision for interpretability. Even though 66% of accuracy was achieved by Neural Network it fails in identifying key attributes related to the response. The model accuracy is generally dependent on the threshold value used in neural network. But since the neural network is generated on Categorical inputs there is no difference in accuracy for changed threshold value, i.e. Neural network directly gives a categorical output.

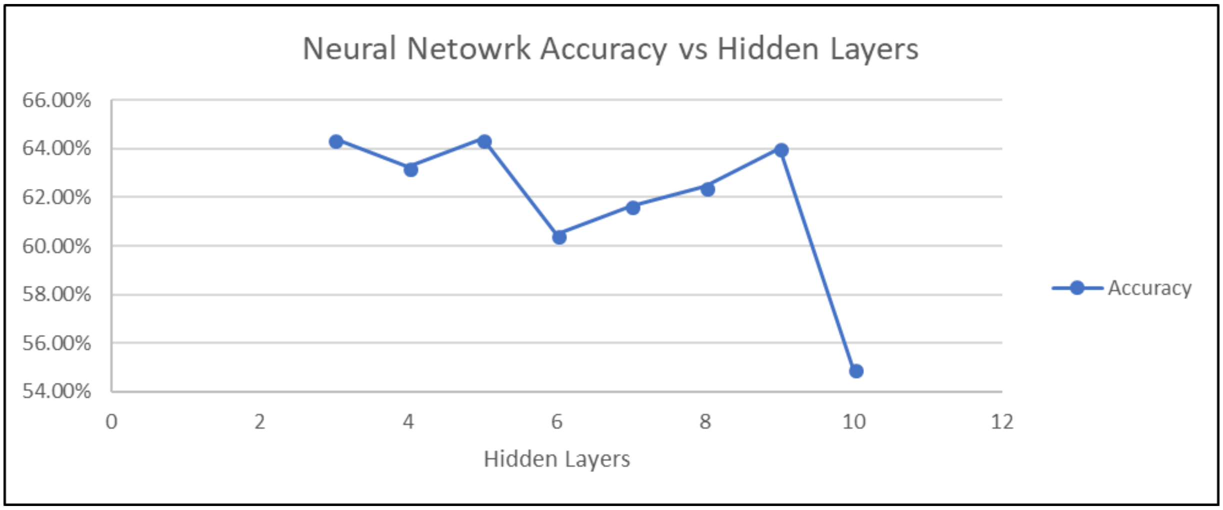


Figure 6 Neural Network Hidden Layer

3.3.8. Random forest

As it was mentioned in 3.1.2, Random Forest was performed on the dataset and the results are as follows: The decrease in Gini index is on the right side and mean error left side.

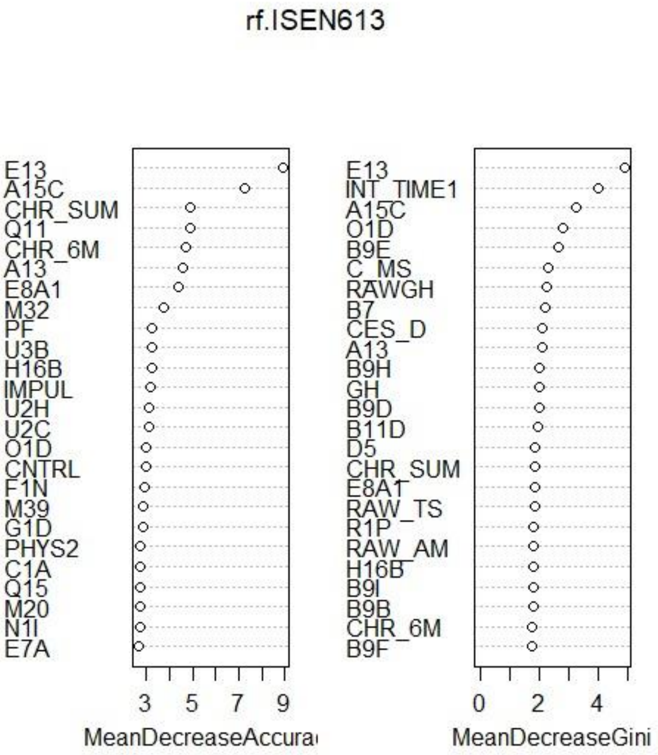


Figure 7 Random Forest Decrease in Gini Index

4. Results Comparison

Since the variables selected were identified for the maximum accuracy, all models below use only 25 predictors and their accuracy are determined accordingly. (All percentages are rounded up)

Method	Results	
Logistics Regression	Accuracy: 68 % Sensitivity: 61%	Error rate: 32 % Specificity: 72%
LDA	Accuracy: 68 % Sensitivity: 61%	Error rate: 32 % Specificity: 72 %
QDA	Accuracy: 67 % Sensitivity: 58 %	Error rate: 33 % Specificity: 73 %
KNN	Accuracy: 63 % Sensitivity: 54 %	Error rate: 37 % Specificity: 69 %
Ridge Regression	Accuracy: 42 %	
SVM	Accuracy: 64 % Sensitivity: 60 %	Error rate: 35 % Specificity: 72 %
Neural Network	Accuracy: 64 %	
Random Forest	Accuracy: 69 %	

Figure 8 Result Comparison

5. Conclusion

Through execution of this project we got exposure to implement various statistical learning methods. As the dataset was very large it contained missing data. Thus, we could learn data cleaning techniques. As the response was categorical we explored different classification techniques taught in the class room to the real time data set. Through literature survey we gained insight in other classification techniques such as Neural Network and Boruta.

On the Data analysis perspective, we could identify a different variable section method like Boruta for decreasing the biases of random forest and neural network as a computationally expensive but high accuracy method for prediction. Usage of Cross validation for finding parameters in parametric methods was explored fully. Bootstrap was also used when there was necessary to have more data points as our ration of n/p was not high.

Since the data was more compartmentalized it was understood that Random Forest would be giving the best accuracy for prediction model. One of the sample tree which can be generated with predicted 25 variables in given below. According to the values of predictors, one would be able to identify whether a person is enrolled in to a Primary care system or not.

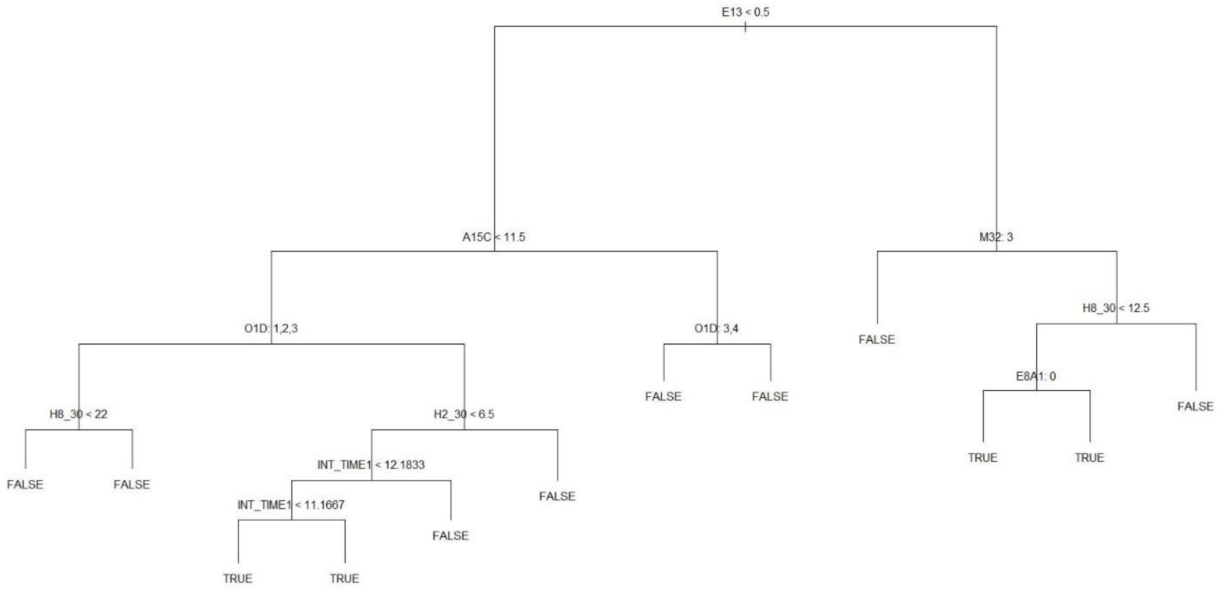


Figure 9 A sample tree

The results we get from this analysis state that the following variables are "important" for predicting whether a person is linked to Primary care right now.

Key findings:

a) Subjects suffering from chronic diseases are more likely to be enrolled in to medical care plan which is intuitive and data analysis proves the intuition.
b) Subject's current employment status directly is related to their enrollment in to Primary Healthcare plant. This enables the plea to government that healthcare to poor and unemployed is still a necessity.
c) Substance abuse and alcohol abuse are big influencers for enrollment and reason behind this may be lack of awareness or financial ability to be enrolled in to a healthcare plan.
d) Subject who have talks with their doctors/MDs are more likely to be educated and hence enrolled in to Primary healthcare system.

In conclusion, we understand that prediction accuracy can be easily improved by utilizing more accurate data with more advanced technique.

6. Executive Summary

Figure 10 Summary

In this project, data analysis was performed on Health Evaluation and Linkage to Primary Care data set. The objective of this project is to determine important attribute from health care survey data set, which affects individual's enrollment in the health care plan. The selected dataset has predictors which cover various aspects such as Demographic, Medical Status, Drug/Alcohol use, Social support, Clinical history. The dataset has 432 predictors with 842 observation. These predictors are a combination of quantitative and qualitative predictors (with different factor levels). As a supervised Data analysis problem, Enrollment to Primary Healthcare at this point of time was chosen as a response variable. The model helps to identify whether the person should opt for health care plan based on various medical conditions. As the response is categorical this is a classification problem. After considering all the predictors the prediction accuracy is low. Thus, variable selection methods such as LASSO, Random Forest, Boruta are used to identify critical factors which would have maximum impact on the health care plan decision. It is found that Boruta gives a maximum accuracy of 68.7 %. It has selected 25 predictors which have a high impact on response. The model fitting with reduced predictors increases the prediction accuracy and model interpretability. For the selected 25 predictors, different classification models are used to fit the data. These include Logistic regression, LDA, QDA, SVM, Neural network. The accuracy of each method is calculated. It is found out that the best model to fit the given data is Random Forest with an accuracy of 69% whereas the logistic and LDA have fairly good prediction accuracy of 68%.

As per our analysis, the major contributing factors to the enrollment of Primary Healthcare system are current enrollment type, substance and alcohol abuse, chronic disease status, HIV status etc. Socio-economic status is the main predictor according to which a Healthcare enrollment is decided. It is also understood that person who actively talks and seeks guidance from a medical professional is more likely to be enrolled in a healthcare plan. There is a requirement for the healthcare system to mainly poor and unemployed which is intuitively proved via this analysis.

7. References

[1] Analystics Vidhya, "Select Important Variables Boruta package," Analystics Vidhya, 2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/. [Accessed 01 May 2018].

[2] M. Kursa and W. Rudnicki, "Feature Selection with the Boruta Package," *Journal of Statistical Software,* vol. 36, no. 11, pp. 02-105, 2010.

[3] J. Gareth, W. Daniela, H. Trevor and T. Robert, An Introduction to Stastical Learning with Application in R, New York: Springer, 2013.

[4] StasticalSolutions, "What is Logistic Regression?," 2018. [Online]. Available: https://www.statisticssolutions.com/what-is-logistic-regression/. [Accessed 01 May 2018].

[5] NCSS Stastical Software, "Chapter 335: Ridge Regression," NCSS, LLC., 2019. [Online]. Available: https://www.ncss.com/software/ncss/ncss-documentation/.

[6] W. McCulloch and P. Walter, "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics,* vol. 5, no. 4, p. 115–133, 1943.

[7] P. J. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Washington: Harvard University, 1975.

[8] D. STATHAKIS, "How many hidden layers and nodes?," *International Journal of Remote Sensing,* vol. 30, no. 8, pp. 2133-2147, 2009.

7. Appendix

Appendix A: Description of attributes for the HELP dataset.

A15C # months in jail-last 6 months

A13 Usual employment pattern in last 6 months (1=Full time, 2=Part time, 3=Student, 4=Unemployed,5=Control envir)

A17D Received EAEDC – past 6 months (0=No, 1=Yes)

ANY_INS Did you have health insurance in past 6 months (0=No, 1=Yes)

B3D Does health limit you in climb several stair flights (1=Limited a lot, 2=Limited a little, 3=Not limited)

C1A Tolf by MD had seix, epil, convuls (0=No, 1=Yes)

C3G1 Have you ever been tested for HIV/AIDS (0=No, 1=Yes)

CHR_6M Chronic medical conds/HIV – past 6m y/n (0=No, 1=Yes)

CHR_SUM Sum chronic medical conds/HIV ever

D2 Take prescription medicdation regularly for physical problem (0=No, 1=Yes)

E10B1_R Mental health treatment past 6m y/n (0=No, 1=Yes)

E13 Tlt # visits to MDs-last 2 weeks bef detox

E8A1 Saw MD/H care worker regarding alcohol/drugs-last 6 months (0=No, 1=Yes)

FRML_SAT Formal substance abuse treatment y/n (0=No, 1=Yes)

H1_30 # days in past 30 bef detox used alcohol

H2_30 #days in 3- bef detox use alcohol to intox

H8_30 # days in last 30 bef detox used cocaine

M32 Spent time in jail because of my alcohol/drug use (0=No, 1=Yes)

O1D # peop spend tx w/who supprt your abstin (1=None, 2= A few, 3=About half, 4= Most, 5=All)

Q11 # men had sex w/in past 6 months (0=0 men, 1=1 man, 2=2-3 men, 3=4+ men)

U2H My substance abuse interferes (0=No, 1=Yes)

U3A Has MD evr talked to you about drug use (0=No, 1=Yes)

U3B Has MD evr talked to you about alcohol use (0=No, 1=Yes)

Appendix B: R- code for Variable Selection

```
 #Converting to categorical


attach(NewDataset)
a=c("TIME","NUM_INTERVALS","INT_TIME1","A15A","A15B","A15C","D3","H1_30","H2_30","H3_30","H
4_30",

"H5_30","H6_30","H7_30","H8_30","H9_30","H10_30","H11_30","H12_30","H13_30","H15A","H15B",

"H16A","H16B","H17A","H17B","ALCQ_30","RAWPF","PF","RP","RAWBP","BP","RAWGH","GH","RAWVT
",

"VT","RAWSF","SF","RAWRE","RE","RAWMH","MH","HT","PCS","MCS","CES_D","C_MS","C_AU","C_DU"
,

"RAW_RE","DEC_RE","RAW_AM","DEC_AM","RAW_TS","DEC_TS","PHYS","PHYS2","INTER","INTRA","IM
PUL",

"IMPUL2","SR","CNTRL","INDTOT","INDTOT2","PSS_FR","PSS_FA","DRUGRISK","SEXRISK","TOTALRAB",
  "RABSCALE","ANY_VIS","ANY_VIS_CUMUL")


abc=NewDataset[,!names(NewDataset)%in%a]
#View(abc)
abc[]=lapply(abc[],factor)
sapply(abc,class)
ISEN613=data.frame(TIME,NUM_INTERVALS,INT_TIME1,A15A,A15B,A15C,D3,H1_30,H2_30,H3_30,H4_3
0,H5_30,H6_30,H7_30,H8_30,H9_30,H10_30,H11_30,H12_30,H13_30,H15A,H15B,H16A,H16B,H17A,H1
7B,ALCQ_30,RAWPF,PF,RP,RAWBP,BP,RAWGH,GH,RAWVT,VT,RAWSF,SF,RAWRE,RE,RAWMH,MH,HT,PC
S,MCS,CES_D,C_MS,C_AU,C_DU,RAW_RE,DEC_RE,RAW_AM,DEC_AM,RAW_TS,DEC_TS,PHYS,PHYS2,INT
ER,INTRA,IMPUL,IMPUL2,SR,CNTRL,INDTOT,INDTOT2,PSS_FR,PSS_FA,DRUGRISK,SEXRISK,TOTALRAB,RAB
SCALE,ANY_VIS,ANY_VIS_CUMUL,abc)
b=c("C1M","E14F","H12_30","H12_RT","H9_PRB","H12_PRB","E8A3","E14E","H6_RT","H9_30","H9_RT",
"H11_RT","S1B","S1D","H6_PRB","H11_PRB","ANY_VIS_CUMUL", "PC_REC",
"PC_REC7","PCS","MCS","ANY_VIS","E10B2_R","E10A","U5","U4","ANY_UTIL","REG_MD")     #with only
one categorical level and less variability and the obvious ones including cumulative

ISEN613=ISEN613[,!names(ISEN613)%in%b]        #now the dataframe has 404 variables and 842 obs
#View(ISEN613)



#data split
```

```
sum(is.na(ISEN613))
#attach(ISEN613)
#set.seed (3)
#Response=ifelse(RCT_LINK==0,"FALSE","TRUE")
#ISEN613=data.frame(ISEN613,Response)          #now its 405 variables
#View(ISEN613)


trainindex=sample (1: nrow(ISEN613),size=0.7*nrow(ISEN613), replace=FALSE)
train=ISEN613[trainindex,]
test=ISEN613[-trainindex,]


ytest=test$Response
ytrain=train$Response
ytest
length(ytest)
length(ytrain)
dim(ISEN613)
dim(test)
dim(train)
length(ytest)
View(ISEN613)


##########################################
#LASSO For variable Selection
##########################################

install.packages("glmnet")
library(glmnet)

x=model.matrix(RCT_LINK~.,ISEN613)[,-1]
y=as.double(ISEN613$RCT_LINK)

# 10 fold cross validation to find best lambda
k=10
set.seed(1)
folds=sample(1:k,nrow(ISEN613), replace = TRUE)

bestlamj=rep(0,10)
library(leaps)
for(j in 1:k){
  grid=10^seq(10,-2,length=100)
```

```
 train= folds!=j
 test=(-train)
 cv.outj =cv.glmnet(x[train,],y[train], alpha=1,lambda = grid)
 bestlamj[j]=cv.outj$lambda.min
}

bestlamj
bestlambda=bestlamj[j]
bestlambda

out=glmnet(x,y,alpha=1,lambda=grid)
plot(out)
lasso.coef=predict(out,type="coefficient",s=bestlambda)[1:434,]
lasso.coef[lasso.coef!=0] #Predictors with non zero coefficients using lasso

trainIndex1=sample(1:nrow(x),nrow(x)*0.7,replace = FALSE)
test2=(-trainIndex1)
y.test=y[test2]
lasso.mod=glmnet(x[trainIndex1,],y[trainIndex1],alpha=1,lambda = bestlambda)
plot(lasso.mod)
summary(lasso.mod)
set.seed(1)

lasso.pred=predict(lasso.mod,s=bestlambda,newx = x[test2,])
summary(lasso.pred)
newy=c(lasso.pred,y)
newy
mean((newy-y.test)^2)



#Randomforest on all attributes



set.seed(1)

library(randomForest)

rf.ISEN613=randomForest(Response~.-RCT_LINK,data=ISEN613,subset=trainindex,mtry=21,importance=TRUE)

summary(rf.ISEN613)

rf.ISEN613

yhat.rf = predict(rf.ISEN613 ,newdata =test)
```

```
l1=mean(yhat.rf==test$Response)          #test accuracy of random forest

l1

varImpPlot(rf.ISEN613,sort=TRUE,n.var=25)



#Bortua

library(Boruta)

set.seed(1)

boruta.train = Boruta(RCT_LINK~.-Response, data = train, doTrace = 2)

print(boruta.train)


plot(boruta.train, xlab = "", xaxt = "n")

lz=lapply(1:ncol(boruta.train$ImpHistory),function(i)

boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])

names(lz) = colnames(boruta.train$ImpHistory)

Labels = sort(sapply(lz,median))

axis(side = 1,las=2,labels = names(Labels),

     at = 1:ncol(boruta.train$ImpHistory), cex.axis = 0.7)


final.boruta = TentativeRoughFix(boruta.train)

print(final.boruta)

a=getSelectedAttributes(final.boruta, withTentative = F)        #this shows the selected attributes

a

plot(a,xlab = "", xaxt = "n")


boruta.df = attStats(final.boruta)

class(boruta.df)

print(boruta.df)

yhat.bortua = predict(boruta.train ,newdata =test)
```

```
sapply(boruta.train,class)


##Random forest on bortua attributes


set.seed(1)

library(randomForest)

rf.ISEN613_1=randomForest(Response~INT_TIME1 + A15C +     H2_30 +    H8_30 +    INDTOT2 +  E13 +
CHR_SUM   +A13    +   A17D +     C1A    +    C3G1 +     D2+       E8A1 +

                        M5    +    M32    +    O1D  +     Q11 +     U2H +     U3A    +   U3B  +     ANY_INS +  FRML_SAT
+ E10B1_R +  H2_PRB +   CHR_6M, data=ISEN613,subset=trainindex,mtry=5,importance=TRUE)

summary(rf.ISEN613_1)

yhat.rf1 = predict(rf.ISEN613_1 ,newdata =test)

l4=mean(yhat.rf1==test$Response)

l4

#RANDOM FOREST ON ATTRIBUTES THAT ARE FROM RANDOM FOREST 25 VARIABLES

set.seed(1)

library(randomForest)

rf.ISEN613_2=randomForest(Response~E13 + INT_TIME1 +A15C + O1D + B9E + C_MS + RAWGH + B7 +
CES_D
+A13+B9H+GH+B9D+B11D+D5+CHR_SUM+E8A1+RAW_TS+R1P+RAW_AM+H16B+B9I+B9B+CHR_6M+B9
F,data=ISEN613,subset=trainindex,mtry=5,importance=TRUE)

summary(rf.ISEN613_2)

yhat.rf2 = predict(rf.ISEN613_2 ,newdata =test)

l5=mean(yhat.rf2==test$Response)

l5

#Since l5<l4 this suggests that attributes from boruta have a higher prediction accuracy than that from
random forest. So, these variables are selected.

 #Tree plotting

 library(tree)

tree.ISEN613=tree(Response~INT_TIME1 + A15C +     H2_30 +    H8_30 +    INDTOT2 +  E13 +
CHR_SUM   +A13    +   A17D +     C1A    +    C3G1 +     D2+       E8A1 +
            M5   +   M32   +   O1D  +    Q11 +    U2H +    U3A   +  U3B  +    ANY_INS +  FRML_SAT +
E10B1_R +  H2_PRB +   CHR_6M ,data=train)

summary(tree.carseats)

plot(tree.ISEN613)

text(tree.ISEN613,pretty = 0)
```

Appendix C: R- code for Model Prediction

```
###################################
Logistic Regression
####################################
set.seed(1)
logistic.fit=glm(Response~INT_TIME1+A15C+H2_30+H8_30+INDTOT2+E13+CHR_SUM+A13+A17D+C1A+
C3G1+D2+E8A1+  M5+M32+O1D+Q11 +U2H
+U3A+U3B+ANY_INS+FRML_SAT+E10B1_R+H2_PRB+CHR_6M,data=ISEN613,
family=binomial,subset=trainindex)
summary(logistic.fit)
logistic.probs=predict(logistic.fit, newdata= ISEN613[-trainindex,], type="response")
logistic.pred=rep("FALSE",253)
logistic.pred[logistic.probs>0.5]="TRUE"
table(logistic.pred, ISEN613[-trainindex, ]$RCT_LINK)
mean(logistic.pred==ISEN613[-trainindex, ]$RCT_LINK)
table(logistic.pred, Response[-trainindex])
mean(logistic.pred==Response[-trainindex])
 #########################################
 ROC Curve Logistic
 ########################################
LR.pred=predict(logistic.fit,type="response")
#LDA.pred=lda.prob$posterior[,2]
roc.curve=function(s,print=FALSE)
{
  Ps=(LR.pred>s)*1
  FP=sum((Ps==1)*(Response=="FALSE"))/sum(Response=="FALSE")
  TP=sum((Ps==1)*(Response=="TRUE"))/sum(Response=="TRUE")
  if(print==TRUE)
  {
    print(table(Observed=Response,Predicted=Ps))
  }
  vect=c(FP,TP)
  names(vect)=c("FPR","TPR")
  return(vect)
```

```
}
threshold=0.5
roc.curve(threshold,print=TRUE)

roc.curve=Vectorize(roc.curve)
M.roc=roc.curve(seq(0,1,by=0.01))
plot(M.roc[1,],M.roc[2,],col="grey",lwd=2,xlab="False Positive Rate",ylab="True Positive Rate")

#############################################################
QDA
#############################################################
set.seed(1)
qda.fit=qda(Response~INT_TIME1+A15C+H2_30+H8_30+INDTOT2+E13+CHR_SUM+A13+A17D+C1A+C3
G1+D2+E8A1+ M5+M32+O1D+Q11 +U2H +U3A+U3B+ANY_INS+FRML_SAT+E10B1_R+H2_PRB+CHR_6M,
data=ISEN613,subset=trainindex)
qda.fit
qda.class=predict(qda.fit,ISEN613[-trainindex,])$class
table(qda.class,ytest)
mean(qda.class==ytest)
summary(qda.fit)

#############################################################
KNN
#############################################################
set.seed(1)

library(class)

e=c("RCT_LINK", "Response")

train1=train[,!names(train)%in%e]

test1=test[,!names(test)%in%e]

n=length(ytest)

dim(train1)

dim(test1)

mean.knn.pred=c()

error.knn.pred=c()

for (i in 1:n){

  knn.pred1=knn(train1,test1,ytrain ,k=i)

  mean.knn.pred[i]=mean(knn.pred1==ytest)
```

```
 error.knn.pred[i]=1-mean.knn.pred[i]

 }

mean.knn.pred

max(mean.knn.pred)

min(error.knn.pred)

f=which(error.knn.pred==min(error.knn.pred))        #value of k at which we get minimum error

which(mean.knn.pred==max(mean.knn.pred))

plot(mean.knn.pred,xlab = "K", ylab="Percentage Accuracy", main="Plot for Percentage Accuracy vs K
nearest neighbours")   #Accuracy plot

plot(error.knn.pred,xlab = "K", ylab="Percentage Error", main="Plot for Percentage Error vs K nearest
neighbours")      # Error plot

points(f,error.knn.pred[f],col="red", pch=20)


knn.pred17=knn(train1,test1,ytrain,k=17)

table(knn.pred17,test$Response)


##########################################################
SVM
##########################################################
set.seed(1)
library(e1071)
svmfit=svm(Response~INT_TIME1+A15C+H2_30+H8_30+INDTOT2+E13+CHR_SUM+A13+A17D+C1A+C3
G1+D2+E8A1+
    M5+M32+O1D+Q11 +U2H +U3A+U3B+ANY_INS+FRML_SAT+E10B1_R+H2_PRB+CHR_6M,
data=ISEN613,subset=trainindex,kernel="radial",gamma=1,cost=1)
summary(svmfit)
#for gamma=1,cost=ee5
svmfit=svm(Response~INT_TIME1+A15C+H2_30+H8_30+INDTOT2+E13+CHR_SUM+A13+A17D+C1A+C3
G1+D2+E8A1+
      M5+M32+O1D+Q11 +U2H +U3A+U3B+ANY_INS+FRML_SAT+E10B1_R+H2_PRB+CHR_6M,
data=ISEN613,subset=trainindex,kernel="radial",gamma=1,cost=1e5)
summary(svmfit)
tune.out=tune(svm,Response~INT_TIME1+A15C+H2_30+H8_30+INDTOT2+E13+CHR_SUM+A13+A17D+C
1A+C3G1+D2+E8A1+
       M5+M32+O1D+Q11 +U2H
+U3A+U3B+ANY_INS+FRML_SAT+E10B1_R+H2_PRB+CHR_6M,data=ISEN613[trainindex,],kernel="radial"
,
        ranges=list(cost=c(.1,1,10,100,1000),gamma=c(.5,1,2,3,4)))
summary(tune.out)
table(true=ISEN613[-trainindex,"Response"],pred=predict(tune.out$best.model,ISEN613[-trainindex,]))
pred=predict(tune.out$best.model,ISEN613[-trainindex,])
mean(pred==ISEN613[-trainindex,"Response"])


##########################################
#Ridge Regression for reduced parameters
##########################################

install.packages("glmnet")
library(glmnet)

x=model.matrix(RCT_LINK~INT_TIME1 + A15C +    H2_30 +   H8_30 +   INDTOT2 +  E13 +     CHR_SUM
+A13    +   A17D +    C1A    +    C3G1 +     D2+       E8A1 +
          M5    +    M32    +    O1D  +     Q11 +     U2H +     U3A    +   U3B  +     ANY_INS +  FRML_SAT +
E10B1_R +  H2_PRB +   CHR_6M  ,ISEN613)[,-1]
y=as.double(ISEN613$RCT_LINK)

set.seed(1)

# To find best lambda
k=10
set.seed(1)
folds=sample(1:k,nrow(ISEN613), replace = TRUE)

bestlamjr=rep(0,10)
library(leaps)
for(j in 1:k){
 grid=10^seq(10,-2,length=100)
 train= folds!=j
 test=(-train)
 cv.outjr =cv.glmnet(x[train,],y[train], alpha=0,lambda = grid)
 bestlamjr[j]=cv.outjr$lambda.min
}

bestlamjr
bestlambdar=bestlamjr[j]
bestlambdar

# Using best lambda on training data set to find prediction accuracy on test data set
trainIndex1r=sample(1:nrow(x),nrow(x)*0.7,replace = FALSE)
test2r=(-trainIndex1r)
```

```
y.testr=y[test2r]
ridge.mod=glmnet(x[trainIndex1r,],y[trainIndex1r],alpha=0,lambda = bestlambdar)
summary(ridge.mod)
set.seed(1)

ridge.pred=predict(ridge.mod,s=bestlambdar,newx = x[test2r,])
summary(ridge.pred)
newyr=c(ridge.pred,y)
newyr
mean((newyr-y.testr)^2)

############################################
#Neural Network
############################################

dim(newdata)
attach(newdata)
set.seed(1)
ideal = class.ind(newdata$RCT_LINK)
n1 = nrow(newdata)
trainIndex=sample(1:n1, size=round(0.7*n1), replace = FALSE)
DA.train=newdata[trainIndex,]
dim(DA.train)
DA.test=newdata[-trainIndex,]
n <- names(DA.train)
#f <- as.formula(paste("RCT_LINK ~", "A15C + H1_30 + H2_30 + H8_30 + A13 + A17D + B3D + C1A + C3G1
+ D2 + E8A1 + E13 + M32 + O1D + Q11 + U2H + U3A + U3B + CHR_SUM + ANY_INS + FRML_SAT +
E10B1_R + CHR_6M"))
nn <- neuralnet(RCT_LINK ~ INT_TIME1 + A15C +     H2_30 +    H8_30 +    INDTOT2 +  E13 +      CHR_SUM
+A13    +   A17D +     C1A    +    C3G1 +     D2+       E8A1 +
               M5    +    M32   +    O1D  +     Q11 +     U2H +     U3A    +   U3B  +     ANY_INS +  FRML_SAT +
E10B1_R +  H2_PRB +   CHR_6M,data=DA.train, hidden = c(24,10), linear.output=F)
pr.nn <- compute(nn,DA.test[,1:25])
#plot(nn)
pr.nn$net.result <- ifelse(pr.nn$net.result<0.4,0,1)
mean(pr.nn$net.result==DA.test$RCT_LINK)
table(pr.nn$net.result==DA.test$RCT_LINK)
```