Santender Customer Transaction Prediction

*Ankush Saha*

Table of Contents

[1 Introduction](#_Toc14220_WPSOffice_Level1) 3

[1.1 Problem Statement](#_Toc1723_WPSOffice_Level2) 3

[1.2 Data](#_Toc27464_WPSOffice_Level2) 3

[2 Methodology](#_Toc1723_WPSOffice_Level1) 4

[2.1 Preparing Data](#_Toc8826_WPSOffice_Level2) 4

[2.2 Data Pre-Processing](#_Toc17176_WPSOffice_Level2) 4

[2.2.1 Missing Value Analysis](#_Toc1723_WPSOffice_Level3) 4

[2.2.2 Outlier Analysis](#_Toc27464_WPSOffice_Level3) 4

[2.2.3 Feature Selection](#_Toc15990_WPSOffice_Level3) 4

[2.2.4 Feature Scaling](#_Toc28126_WPSOffice_Level3) 5

[2.2.5 Sampling](#_Toc8826_WPSOffice_Level3) 5

[3 Model development](#_Toc27464_WPSOffice_Level1) 6

[3.1 Logistic Regression](#_Toc19423_WPSOffice_Level2) 6

[3.2 Decision Tree](#_Toc29930_WPSOffice_Level2) 7

[3.3 KNN Model](#_Toc4693_WPSOffice_Level2) 7

[4 Model Evaluation](#_Toc15990_WPSOffice_Level1) 8

[4.1 Confusion Matrix](#_Toc27680_WPSOffice_Level2) 8

[4.2 AUC](#_Toc5508_WPSOffice_Level2) 8

[4.3 Precision 1](#_Toc20027_WPSOffice_Level2)0

[4.4 Recall 1](#_Toc14815_WPSOffice_Level2)0

[4.5 Accuracy 1](#_Toc22602_WPSOffice_Level2)0

[5 Model Selection 1](#_Toc28126_WPSOffice_Level1)1

[6 Steps To Run the code 1](#_Toc8826_WPSOffice_Level1)1

[6.1 In R 1](#_Toc12537_WPSOffice_Level2)1

[6.2 In Python 1](#_Toc17225_WPSOffice_Level2)2

[7 Summary 1](#_Toc17176_WPSOffice_Level1)3

[8 Visualization 1](#_Toc20765_WPSOffice_Level1)3

[8.1 Corellation Plot 1](#_Toc20142_WPSOffice_Level2)3

[8.2 Boxplot 1](#_Toc24375_WPSOffice_Level2)7

[8.3 Decision Tree 2](#_Toc26724_WPSOffice_Level2)1

# 1 Introduction

## 1.1 Problem Statement

At Santander , mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals.

we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted

## 1.2 Data

Our task is to build classification models which will classify whether a customer will make transaction or not base on given variables. Given below is a sample of the data set that we are using to predict chances of transaction:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID\_code | target | var\_0 | var\_1 | var\_2 | var\_3 | var\_4 | var\_5 | .... | var\_195 | var\_196 | var\_197 | var\_198 | var\_199 |
| train\_0 | 0 | 8.9255 | -6.7863 | 11.9081 | 5.093 | 11.4607 | -9.2834 | .... | -2.3978 | 7.8784 | 8.5635 | 12.7803 | -1.0914 |
| train\_1 | 0 | 11.5006 | -4.1473 | 13.8588 | 5.389 | 12.3622 | 7.0433 | .... | 2.0339 | 8.1267 | 8.7889 | 18.356 | 1.9518 |
| train\_2 | 0 | 8.6093 | -2.7457 | 12.0805 | 7.8928 | 10.5825 | -9.0837 | .... | 3.1417 | -6.5213 | 8.2675 | 14.7222 | 0.3965 |
| train\_3 | 0 | 11.0604 | -2.1518 | 8.9522 | 7.1957 | 12.5846 | -1.8361 | .... | -1.2706 | -2.9275 | 10.2922 | 17.9697 | -8.9996 |
| train\_4 | 0 | 9.8369 | -1.4834 | 12.8746 | 6.6375 | 12.2772 | 2.4486 | .... | -1.5121 | 3.9267 | 9.5031 | 17.9974 | -8.8104 |

Table 1: ***train.csv***

As we can see there are 200 independent variables and one target variable which is dependent on the 200 independent variables. We will make our model om this train data-set and predict on the test data-set

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID\_code | var\_0 | var\_1 | var\_2 | var\_3 | var\_4 | var\_5 | .... | var\_195 | var\_196 | var\_197 | var\_198 | var\_199 |
| test\_0 | 11.0656 | 7.7798 | 12.9536 | 9.4292 | 11.4327 | -2.3805 | .... | 2.4669 | 4.3654 | 10.72 | 15.4722 | -8.7197 |
| test\_1 | 8.5304 | 1.2543 | 11.3047 | 5.1858 | 9.1974 | -4.0117 | .... | 0.4773 | -1.4852 | 9.8714 | 19.1293 | -20.976 |
| test\_2 | 5.4827 | -10.3581 | 10.1407 | 7.0479 | 10.2628 | 9.8052 | .... | 2.1281 | -7.1086 | 7.0618 | 19.8956 | -23.1794 |
| test\_3 | 8.5374 | -1.3222 | 12.022 | 6.5749 | 8.8458 | 3.1744 | .... | 3.1656 | 3.9567 | 9.2295 | 13.0168 | -4.2108 |
| test\_4 | 11.7058 | -0.1327 | 14.1295 | 7.7506 | 9.1035 | -8.5848 | .... | -0.286 | -5.1612 | 7.2882 | 13.926 | -9.1846 |

Table 2: ***test.csv***

# 2 Methodology

Any predictive modeling requires that we look at the data before we start modeling. First we prepare the data-set and we prepare the Data-set to feed into our model. In data preparation we perform actions like missing value analysis, outlier analysis, feature scaling, feature sampling.

## 2.1 Preparing Data

Before any kind of analysis we prepare the data. In this case we make the Column “ID\_code” the row names of the data-set, and take the target variable from 1st column to last column for ease of making model

## 2.2 Data Pre-Processing

### 2.2.1 Missing Value Analysis

Here we check if there is any missing value cell in the data set or not. If there is any missing value cell in the data set it may effect the model, and our model may not be efficient enough to predict correct value. So we need to either empty or impute those empty cells. If the number of empty cells in a column is more than 30 percent we can drop that variable as it may not add much value in our model. And if it is less than 30 percent we can impute them with either of any basic statistical process (i.e. mean, median, mode ) or we can use KNN imputation method which one is best fit.

After our analysis we found out that In our data set there are no missing value present. So no need to apply any kind of imputation method.

### 2.2.2 Outlier Analysis

Sometimes in a column of data set there are some values which do not comply with general behaviour of other data. These data are called outliers. These values may manipulate the behaviour of the data set. And our model may not work accurately. To tackle this kind of situation we can either delete the row or the impute the values with mean,median,mode or KNN imputation.

In our Data set total number of missing values are less than 15%, which is very less compared to this large data set. So we can remove the outliers which will not much impact our model. After removing outliers the size is (175073 X 201)

### 2.2.3 Feature Selection

There might be some variables who are highly dependent to each other. So keeping both of them in our data set may crate some partiality towards some features. So we can remove one of the variables. To check the correlation of the variables we can plot them in a graph and we can visually see that how much they are dependent to each other.

In our data set as we can see there are no such variables who highly dependent to each other. So we can feed all the variables to our model. For the plots please refer to page 13.

### 2.2.4 Feature Scaling

Now every variable has their own scale of measurement. As we are considering only values and not the scales the values of the different variable may differ in very high range. It may cause to partiality towards higher valued variable. So we need to scale them up or town to a common scale.

Here in our data set we have made the each variable to be varied from 0 to 1.

To achieve this we have used below formula



### 2.2.5 Sampling

First we need to divide our data set into two parts. First part on which we will develop our model and second part on which we will test our model how accurate it will predict.

So for sampling we have used random sampling technique which will randomly separate data into train and test data set for given size and for our data set we have taken 70% for train data and 30% for test data

After sampling train data size is (122552 X 201)

And test data size is (52521 X 201)

# 3 Model development

For prediction we need to develop a model by which we will predict whether a customer will make a transaction or not. There are different types of model building technique available. For different kind of data set different techniques work accurately. So we will develop three models using three different techniques and will check which model is working more accurately, and we will feed our test data set to the best model and develop our prediction.

The three techniques we are gonna use are

1. Logistic regression
2. Decision tree
3. KNN

## 3.1 Logistic Regression

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be combined to model several classes of events such as determining whether an image contains a cat, dog, lion, etc... Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

It first makes a formula depending on the independent variables to calculate the dependent variable. The formula is as shown below



Where  = probability

= estimations of the nth independent variable.

So our model will calculate the values of co-efficient of the variables from the train data and dependent on them it will predict the probability of dependent variable. If the probability is greater than 0.5 we assume that the incident is going to happen and if its less than 0.5 we assume that the incident is not going to happen.

It also calculate std. Errors, t value and p value.

Std error provides estimated error associated with the estimations.

t-value is (estimation)/(std. Error).

The absolute value of t-value must be less than std error.

The p-value is a probability value. The p-value will be always between 0 and 1.

Lower p-value indicates statistically significant effect of predictor on dependent variable.

## 3.2 Decision Tree

In this model it examines one feature at a time and branch according to value of feature.

It adaptively choose next feature to examine. Please find the decision tree for 100 variables at page 21

## 3.3 KNN Model

KNN stands for K-Nearest Neighbor. It is a simple algorithm which store all available cases and classifies new cases based on a similarity measure. It is a very slow learning method.

It would pick a number of neighbors we want to use for our model. it will pick the neighbors by measuring the distance from the data and it will pick the data with majority.

# 4 Model Evaluation

For Model Evaluation We use various techniques. Lets use them on our model and see what result we get back from them.

## 4.1 Confusion Matrix

It is a matrix which indicates how many prediction our model has made correctly. It is not a evaluation technique, But we do other techniques refereeing to the matrix it look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matix | | Predicted Prediction | |
| 0 | 1 |
| Actual Prediction | 0 | TP | FN |
| 1 | FP | TN |

**TP:** True Positive

**FN:** False Negative

**FP:** False Positive

**TN:** True Negative

In our Models we git confusion matrixes like below

**Logistic regression:**

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matix | | Predicted Prediction | |
| 0 | 1 |
| Actual Prediction | 0 | 46697 | 647 |
| 1 | 3832 | 1346 |

**Decision Tree:**

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matix | | Predicted Prediction | |
| 0 | 1 |
| Actual Prediction | 0 | 43207 | 4264 |
| 1 | 4118 | 933 |

**KNN Model:**

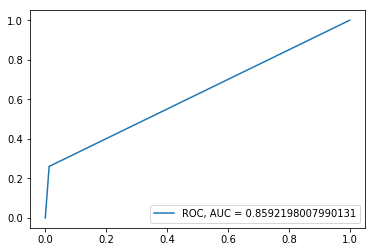
|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matix | | Predicted Prediction | |
| 0 | 1 |
| Actual Prediction | 0 | 43058 | 4338 |
| 1 | 4156 | 976 |

## 4.2 AUC

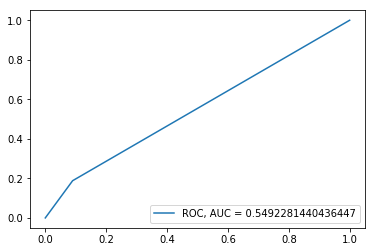
AUC stands for area under curve. So first we have to draw a curve of TP vs. FP on graph. This graph is called ROC graph or receiver operating characteristic curve. Then we get the area under the curve, and that value is our AUC. The value of AUC is range from 0 to 1. A model whose AUC is 0 has predicted 100% wrong on the other hand AUC 1 means model has predicted 100% correct.

Let’s check out our model’s ROC and AUC

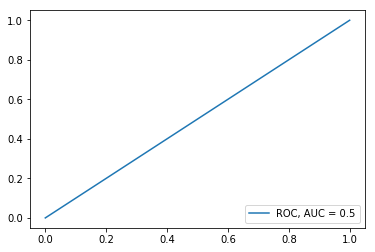
**Logistic regression:**



**Decision Tree:**

****

**KNN Model:**



## 4.3 Precision

Precision is a ratio of the number of true positives divided by the sum of the true positives and false positives. It is also known as True positive rate. It should be higher for any model.

Precision = True Positives / (True Positives + False Positives)

We take the Values from the confusion matrix and calculate the value of Precision.

In our models Precision values are:

**Logistic regression:** 68.61%

**Decision Tree:** 18.37%

**KNN Model:** 15.63%

## 4.4 Recall

Recall is calculated as the ratio of the number of true positives divided by the sum of the true positives and the false negatives. Recall is also known as sensitivity.high recall means that an algorithm returned most of the relevant results

Recall = True Positives / (True Positives + False Negatives)

We take the Values from the confusion matrix and calculate the value of Precision.

In our models Precision values are:

**Logistic regression:** 26.08%

**Decision Tree:** 6.9%

**KNN Model:** 15.62%

## 4.5 Accuracy

It is Calculated by below formula.

Accuracy: (True Positives + False Negatives)/(True Positives + False Positives + True Negatives + False Negatives)

Accuracy For our models

**Logistic regression:** 91.69%

**Decision Tree:** 89.83%

**KNN Model:** 88.57%

# 5 Model Selection

From all of the test we can clearly say that Logistic Regression is the best fit. As its accuracy is high, AUC is high, Precision and recall is also high.

So we will use Logistic regression model for predicting our test data.

# 6 Steps To Run the code

Below are the instruction to keep in mind to run the code.

1. Download the files and keep them in a folder and get th file path.
2. In R you have to press ctrl+Enter to execute a line of code. You can run multiple lines together by selecting the line and pressing ctrl+Enter.
3. In python you have to press shift+Enter to run a cell. You have to run a cell each at a time. The python program is written in Jupyter notebook.

Please follow the Below Steps to predict the test data. Before performing the steps.

## 6.1 In R

1. Open RStudio and open the file : “Project\_1.R”.
2. First run line 1 to 13 to install all the required library.
3. in 15th line of command you have to give your file path between the quotation ex: “C:/Users/Ankush Saha/Documents/Project\_1”.
4. Then run line 18&19 to load the train data and take a backup of it. ***You can run line 19 after every step to take the updated backup.***
5. Run line 23 to 28 to adjusting the columns.
6. Now you can test the data set for missing values by running line 33 to 38. As there were no missing values the codes are commented you can uncomment them by selecting the lines together and pressing ctrl+shift+c
7. Now for outlier analysis there are code written in line no. 47 to 57, they are also commented. So you can test them by uncommenting them and in 47th line you can give the range for which you want to plot. And in 57th line just give the number in i of gn’i’ and run 47 to 57th line. And run 58th to 62th line to remove outliers.
8. For feature selection you can test the corelletion of the variables by running line 67 to 80.
9. For scaling the data run line run line 94 to 98.
10. For dividing the data set into train and test data run line no. 101 to 104.
11. Now to develop Logistic Regression model rum 112th line. It will save the model in **lr\_model.**
12. And to get the predicted data run 114th and 115th line.
13. Now to get various evaluations of the model you can run the lines. Like for AUC: 122 and 123, for Confusion Matix 126&127, For MAPE 129th to 135th line. And for accuracy, precision and recall we can manually get the values from Confusion Marix and put it in the formula and get the values.
14. You can also develop develop Decision tree and KNN models. From this file. All the codes for these models are commented as we are using LR model to predict our Test data. You have to just uncomment those lines and run those lines as instructed in 13th point.
15. ***In any of these steps if you have done anything wrong just run line 20 to get the last backup.It will avoid rerunning the whole code from start, and it will save time.***
16. Now to test it on our test data we first have to load the test file and prepare it to feed it into the model. To achieve this run 188 to 190th line and ***take a backup by running line 191***
17. Now as we have made the model to feed a scaled data set we have to scale the test data, and by running line 195 to 199 we can acheve this.
18. Now run line 202 to 207 to predict the values and make it usable.
19. Now finally run line 208 to save the data set in csv format in your hard-disk. The predicted values are attached to this file.

## 6.2 In Python

1. Open Jupyter notebook and open “Project\_1.ipynb” from the folder.
2. Run cell 1 to install required libraries.
3. Run cell 2 by giving your file location ex: “C:/Users/Ankush Saha/Documents/Project\_1”. to set your location.
4. Run cell 3 & 4 to load the train data set and take a backup. ***You can run cell 4 after every step to take the updated backup.***
5. You can check your data by running cell 5.
6. Run cell 6 to prepare your data.
7. You can run cell 7 to check missing value.
8. Run cell 8 to plot boxplot for variables, you can give the variable name for boxplot of that variable. Run cell 9 to remove outliers.
9. For feature selection you can test the corelletion of the variables by running cell 10.
10. For scaling the data run line run cell 11.
11. For dividing the data set into train and test data run cell 12.
12. Now to develop Logistic Regression model running cel 13. It will save the model in **logit.**
13. And to get the predicted data run cell 14.
14. Now to get various evaluations of the model you can run the cells. Like for AUC: 15, for Confusion Matix 16,. And for accuracy, precision and recall cell 17 & 18.
15. You can also develop develop Decision tree and KNN models. From this file. . You just have to do as instructed in 14th point for cells under them.
16. ***In any of these steps if you have done anything wrong just run cell 4 to get the last backup by commenting the first line uncommenting the second line. It will avoid rerunning the whole code from start, and it will save time.***
17. Now to test it on our test data we first have to load the test file and prepare it to feed it into the model. To achieve this run cell 29 and ***take a backup by running cell 30***
18. Now as we have made the model to feed a scaled data set we have to scale the test data, and by running cell 31 we can acheve this.
19. Now run cell 32 & 33 to predict the values and make it usable.
20. Now finally run cell 34 to save the data set in csv format in your hard-disk. The predicted values are attached to this file.

# 7 Summary

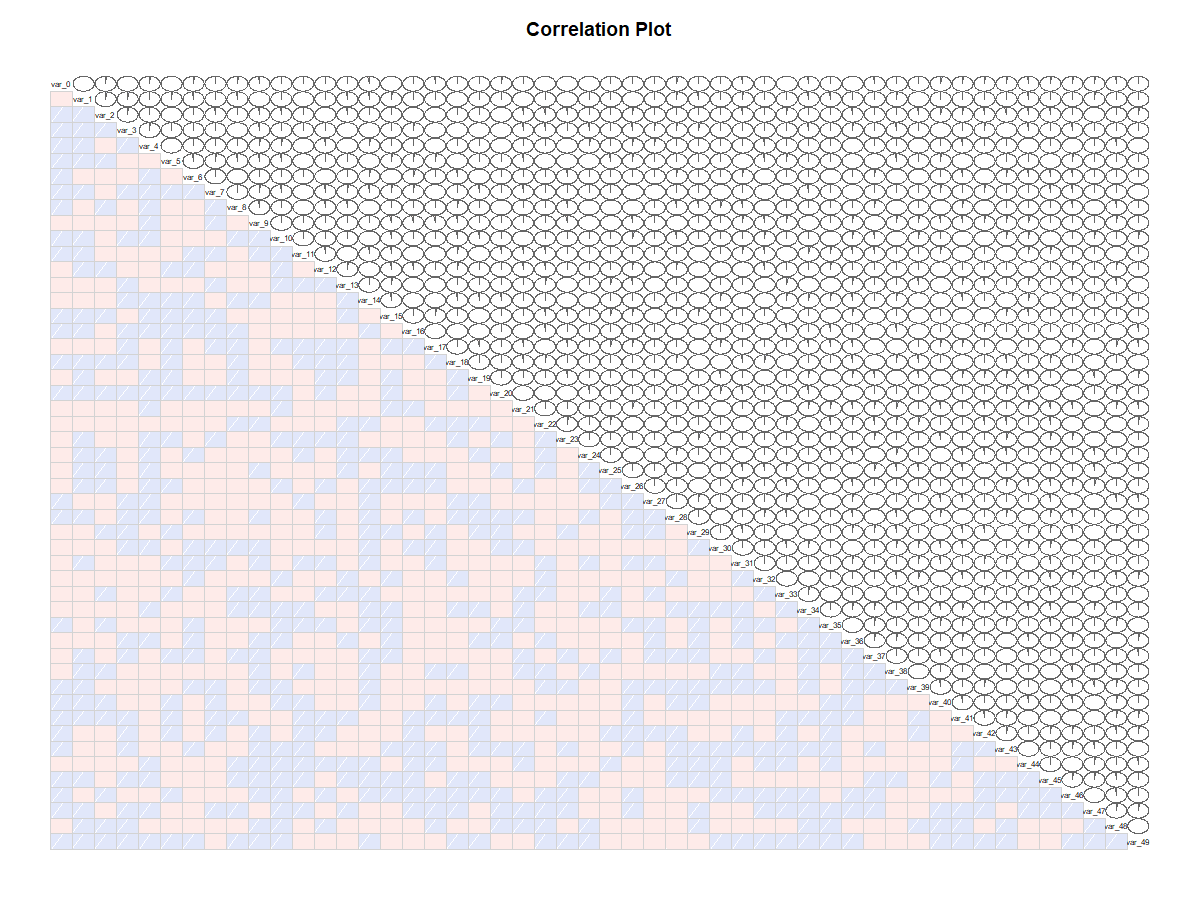
In this project we have built a model which will help business to predict which customer is gonna make a transaction. The model’s accuracy rate is up to 92%, which is very high.

In the test data set there are 200000 customers, and using our model we have predicted that around 16730 customers who are gonna make a transaction. So if the business approach those customer plus 10000 to 20000 more customers it will save bank’s lots of time, money and effort. This model will increase the efficiency of business.

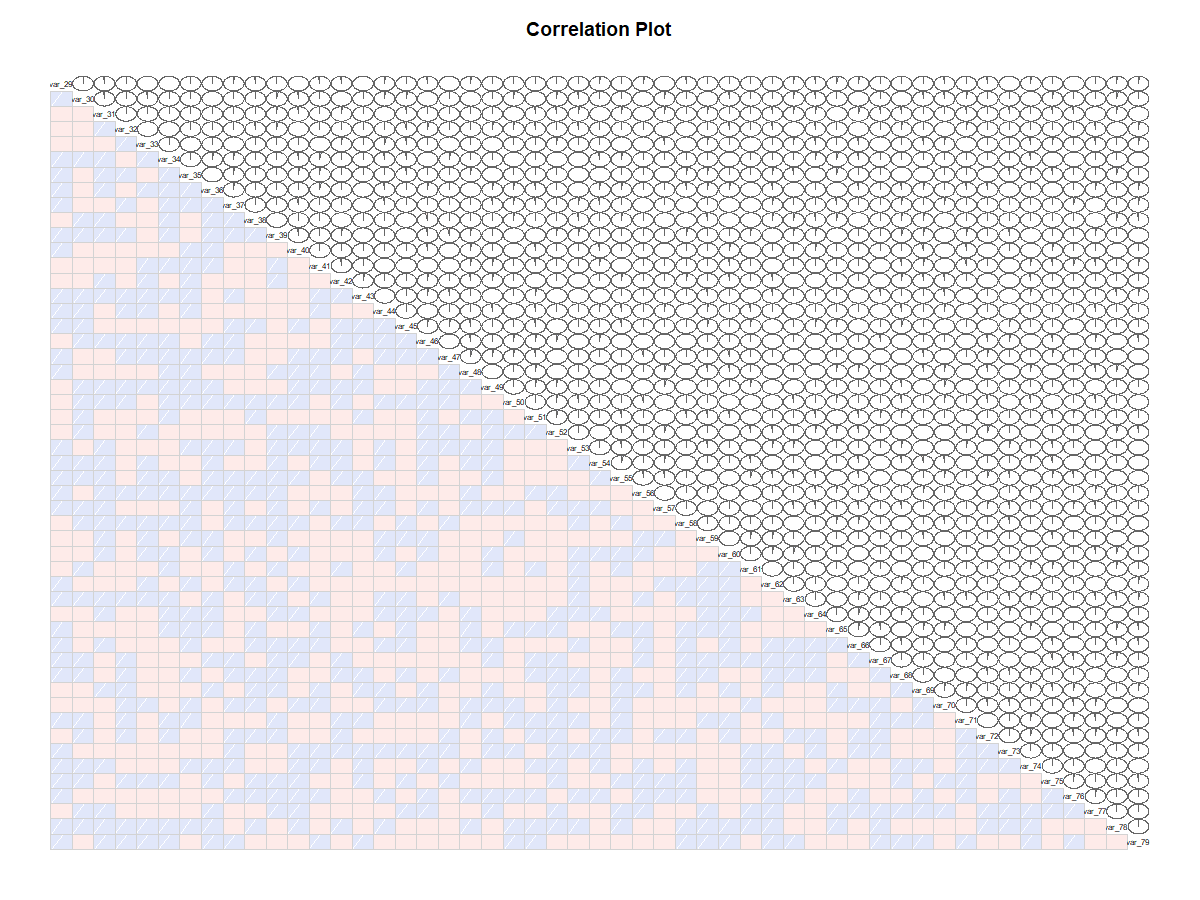
# 8 Visualization

## 8.1 Corellation Plot

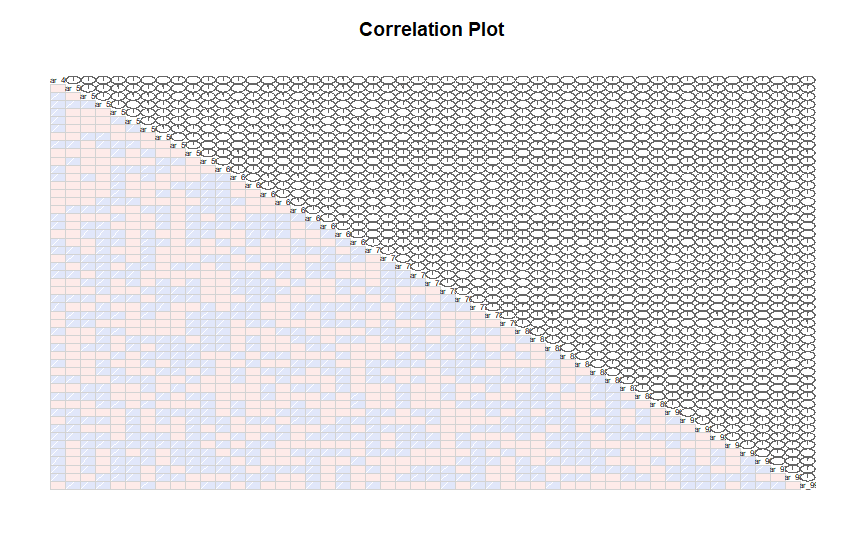
Variable 0 to 49:



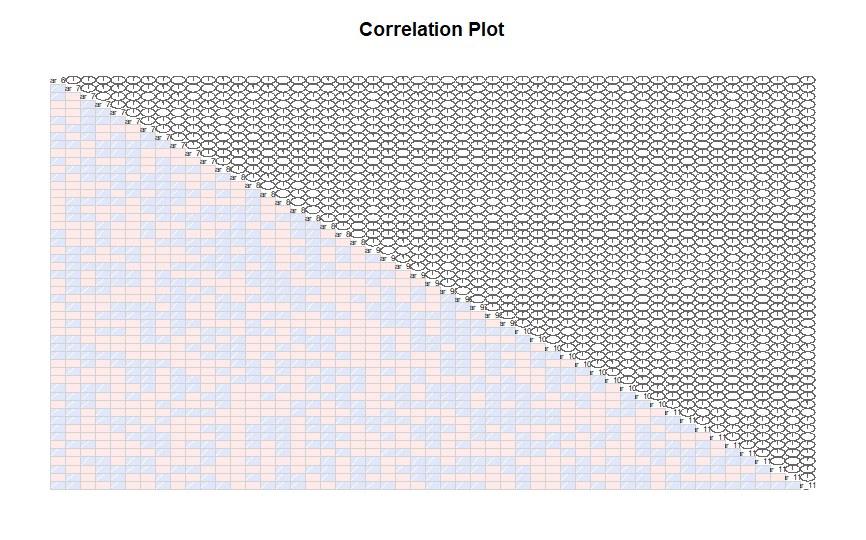
Var 29 to 79:



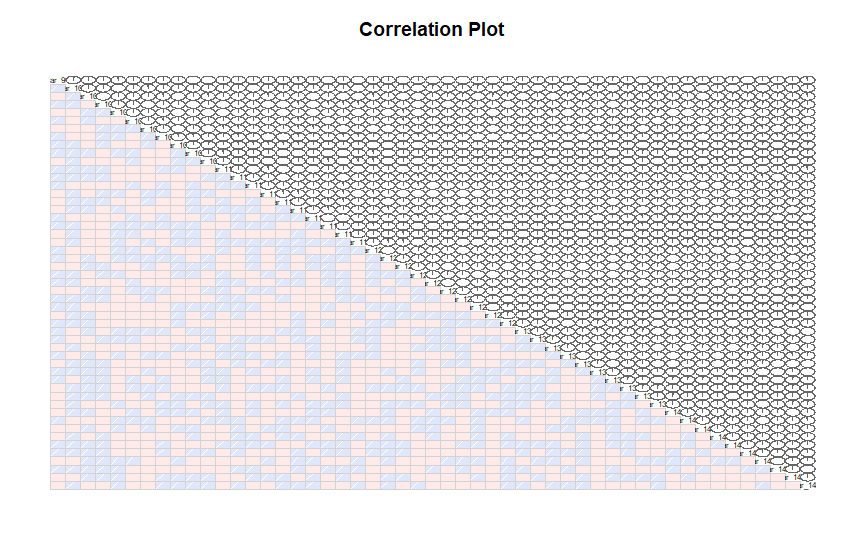
Var49 to 99:



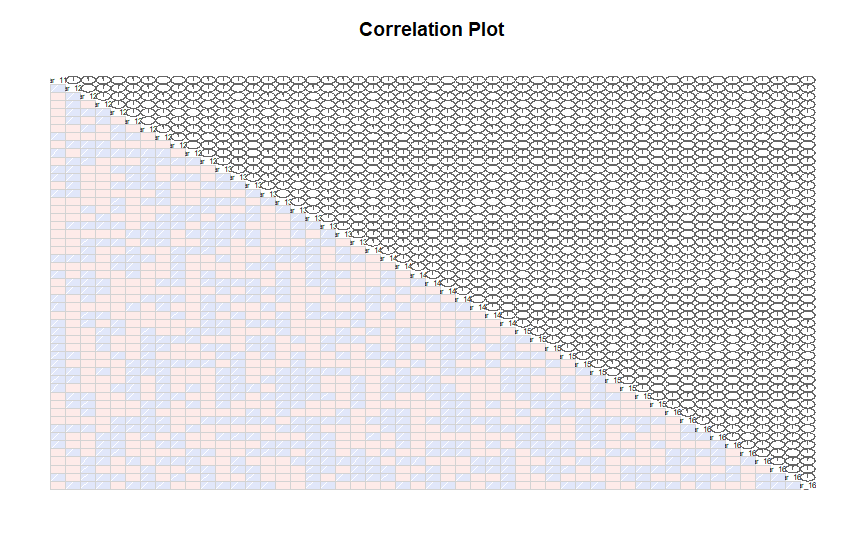
Var 69 to 119:

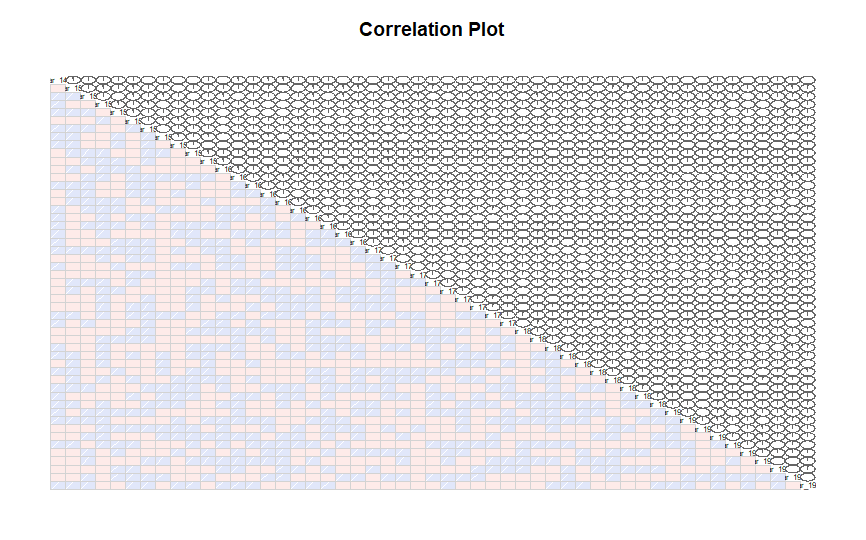


Var 99 to 149:



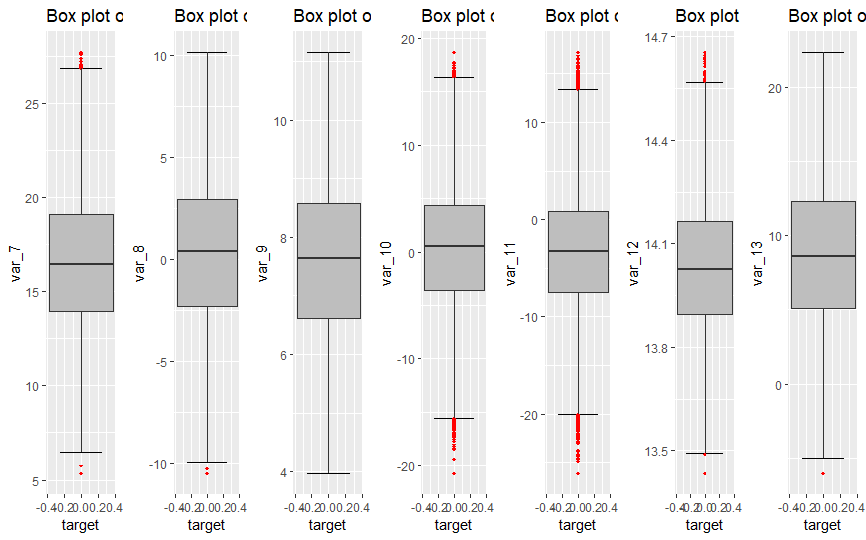
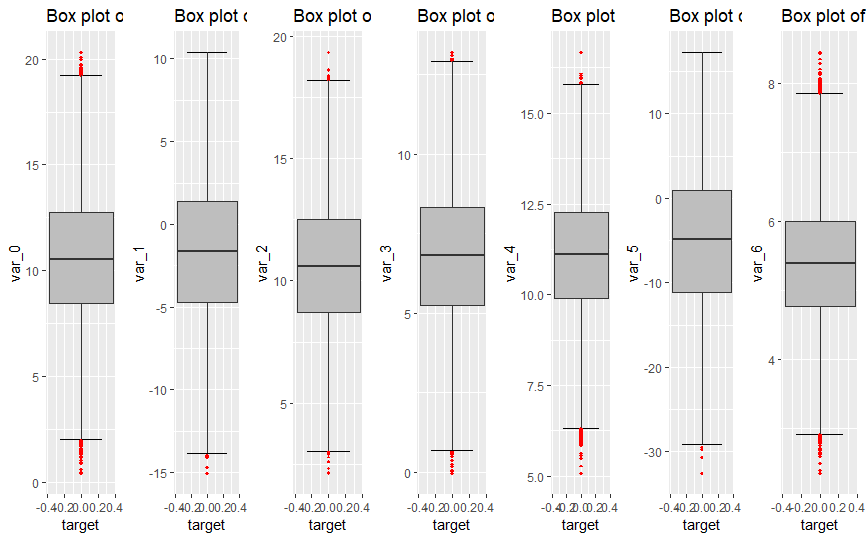
Var 119 to 169:

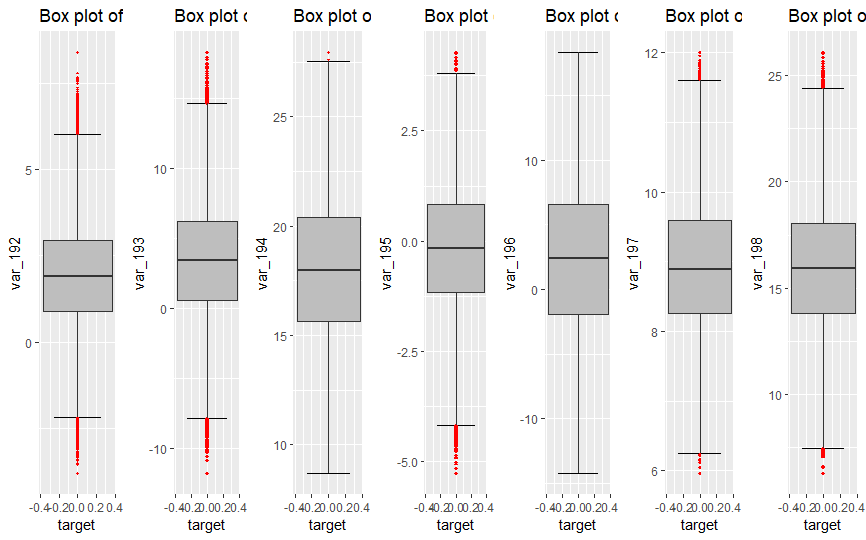
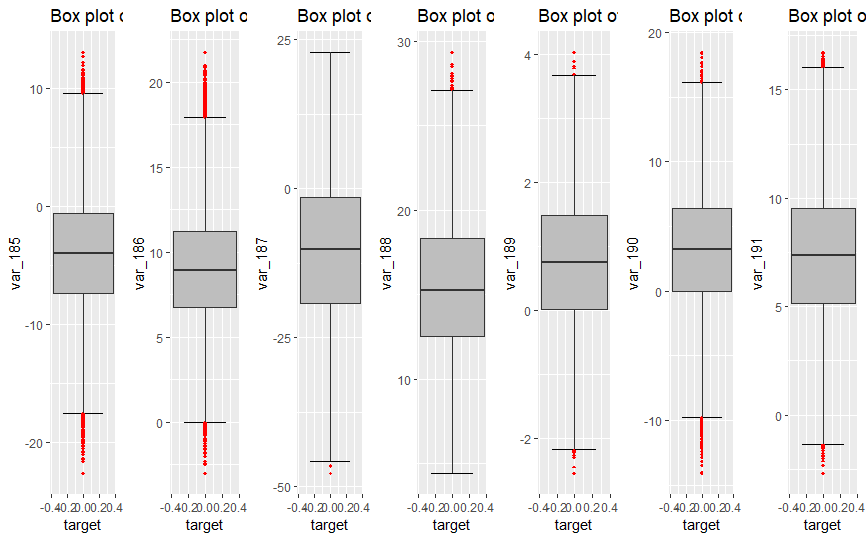
Var 149 to 199:



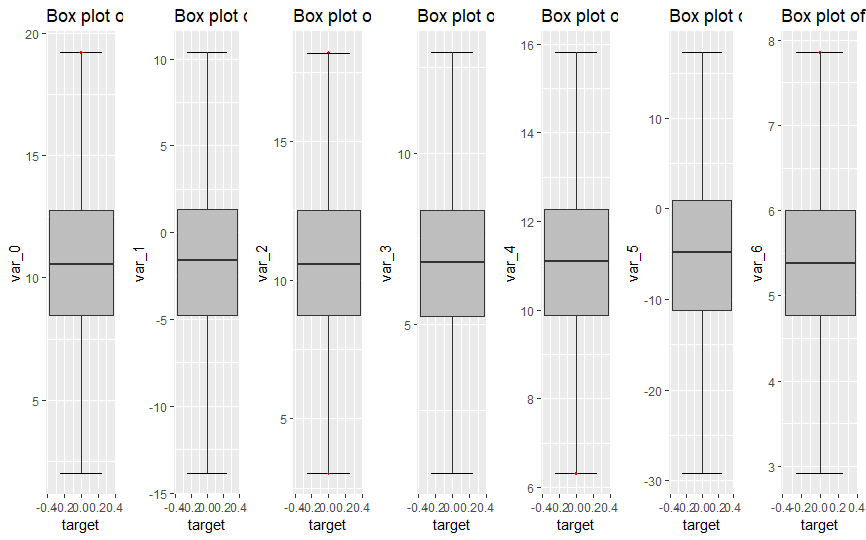
8.2 Boxplot

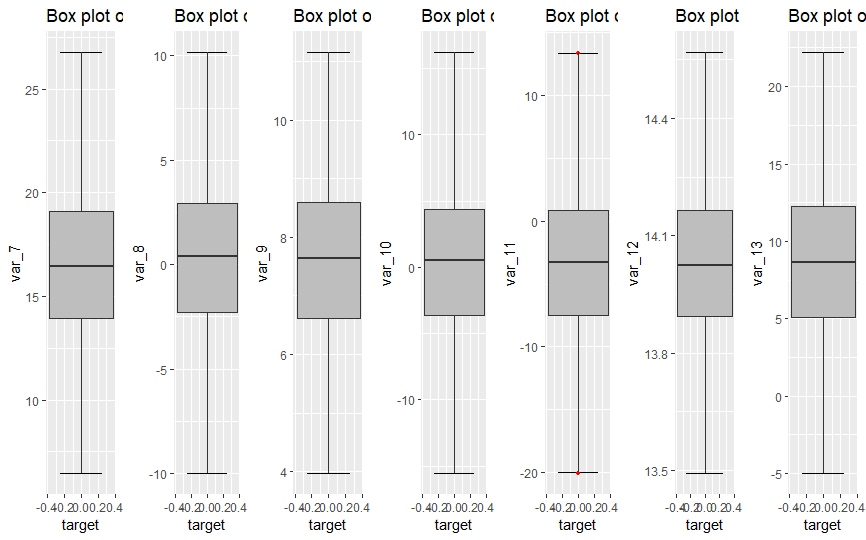
**Boxplot analysis before removing outliers:**

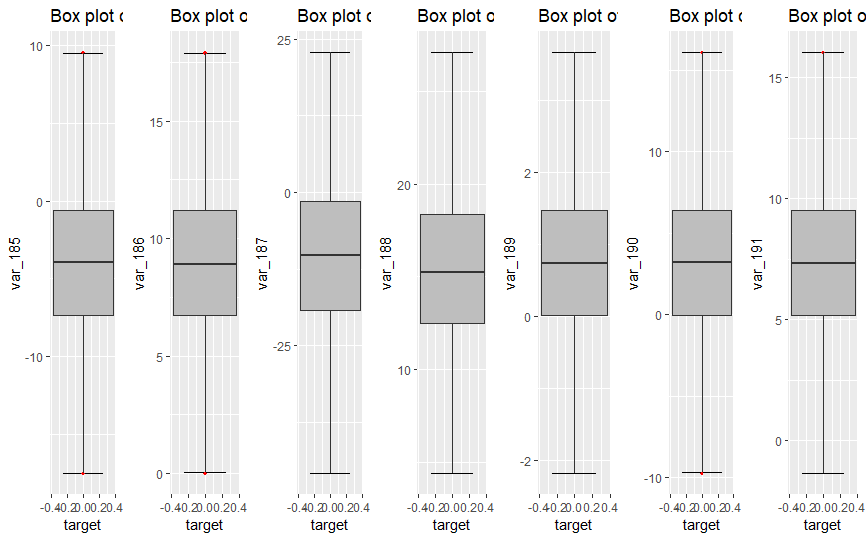


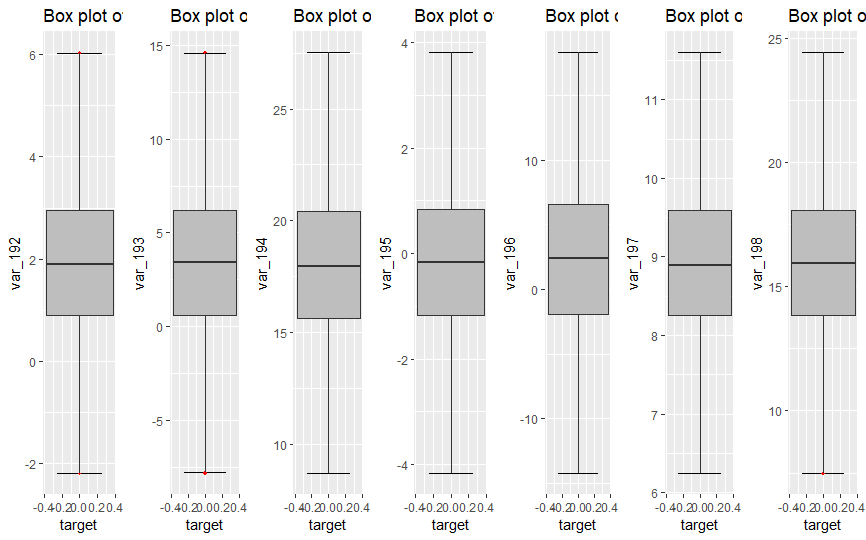


**Boxplot analysis before after removing outliers:**









## 8.3 Decision Tre