

Mini Project 3 – Machine Learning

Submitted by

Sita K

BACP Batch (Dec 19-May 20)

Great Learning

April 4, 2020




Table of Contents

1. Project Objectives..... 4

2. Assumptions..... 4

3. Data Analysis – Approach..... 4

 a. Problem 1..... 4

 b. Problem 2..... 4

4. Problem 1 Responses..... 4

5. Problem 2 Responses..... 11

Appendix A..... 24

Table of Figures

| | |
|--|----|
| FIGURE 1: HISTOGRAM PLOTS | 5 |
| FIGURE 2: DENSITY PLOTS | 5 |
| FIGURE 3: DENDROGRAM | 6 |
| FIGURE 4: CLUSTER HEIGHT PLOT | 6 |
| FIGURE 5: CLUSTER DENDROGRAM | 7 |
| FIGURE 6: CLUSPLOT | 7 |
| FIGURE 7: AGGR FUNCTION OUTPUT | 8 |
| FIGURE 8: SILHOUETTE SUMMARY | 8 |
| FIGURE 9: SILHOUETTE METHOD | 9 |
| FIGURE 10: WSS PLOT | 9 |
| FIGURE 11: K MEANS CLUSPLOT | 10 |
| FIGURE 12: AGGREGATE – KMEANS | 10 |
| FIGURE 13: SIL SUMMARY - KMEANS | 11 |
| FIGURE 14: RPART PLOT | 11 |
| FIGURE 15: CART MODEL PLOT FOR DEV DATA | 12 |
| FIGURE 16: CART MODEL PLOT FOR HOLDOUT DATA | 12 |
| FIGURE 17: CART - CONFUSION MATRIX FOR TRAINING DATA | 13 |
| FIGURE 18: CONFUSION MATRIX FOR HOLDOUT DATA | 14 |
| FIGURE 19: RF MODEL PLOT | 15 |
| FIGURE 20: RF MODEL IMPORTANCE | 15 |
| FIGURE 21: TUNE RF PLOT | 16 |
| FIGURE 22: RF DEV DATA PLOT | 16 |
| FIGURE 23: CONFUSION MATRIX FOR DEV DATA – RF | 17 |
| FIGURE 24: HOLDOUT DATA RF PLOT | 18 |
| FIGURE 25: CONFUSION MATRIX FOR HOLDOUT – RF | 18 |
| FIGURE 26: ANN MODEL PLOT | 19 |
| FIGURE 27: ROCR PLOT FOR DEV DATA – ANN | 20 |
| FIGURE 28: CONFUSION MATRIX FOR DEV DATA – ANN | 20 |
| FIGURE 29: ROCR PLOT FOR HOLDOUT DATA- ANN | 21 |
| FIGURE 30: CONFUSION MATRIX FOR HOLDOUT DATA – ANN | 22 |
| FIGURE 31: COMPARE MODELS | 23 |
| FIGURE 32: VARIMPLOT – RF | 23 |
| FIGURE 33: IMPORTANCE VALUES | 24 |

1. Project Objectives

- a. The objective of this problem1 in this report is to explore the Bank Marketing dataset (bank_marketing_part1_Data.csv) in R and to build a clustering model and recommend different promotional strategies for each cluster.
- b. The objective of this problem2 in this report is to explore the Insurance dataset (insurance_part2_data.csv) in R and to build a CART, RF & ANN and compare the models' performances in train and test sets.

2. Assumptions

- The data provided is conclusive and contains the required data

3. Data Analysis – Approach

a. Problem 1

1. Environment data setup and data import
2. Calculating the required values using inbuilt functions
3. Apply scaling to the data
4. Apply hierarchical Clustering and plot dendrogram
5. Apply K-means clustering and wss plot
6. Describe the cluster profiles and various promotional strategies.

b. Problem 2

1. Environment data setup and data import
2. Split the data into test and train data
3. Create CART, RF & ANN models
4. Apply model to both train and test data

For environment data setup, R's inbuilt packages were used. Also for setting up working directory '`setwd()`' function was used. The given dataset is in .csv format, so we can use `read.csv` function to import the data. All the R commands are in Appendix A.

4. Problem 1 Responses

1. For Basic data summary we can use the function **summary** in R, which provides us with mean, median etc of each column. **Cor** function provides us with the correlation of each variable with each other. **Dim** function provides us with the dimensions of the data. **Str** function provides us with the structure of the data.
To check if any blank values are there we use **is.na** function.
Plot_histogram and **plot_density** show various behaviour of all the variables in graphical format.

From the plots we can see that most of the columns follow normal distribution except **spending** and **max spent in single shopping**.

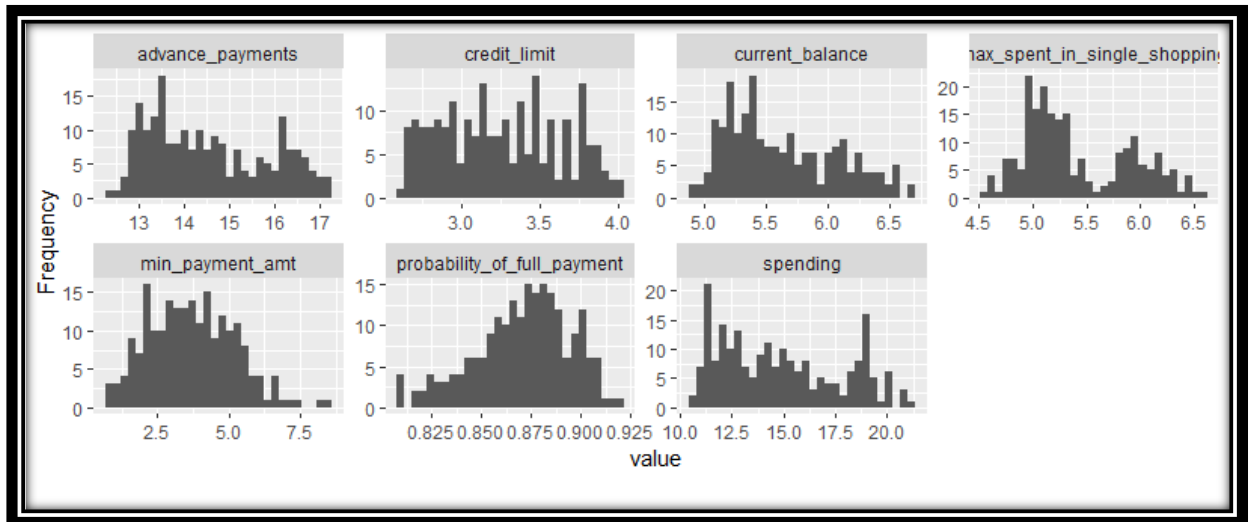


Figure 1: Histogram Plots

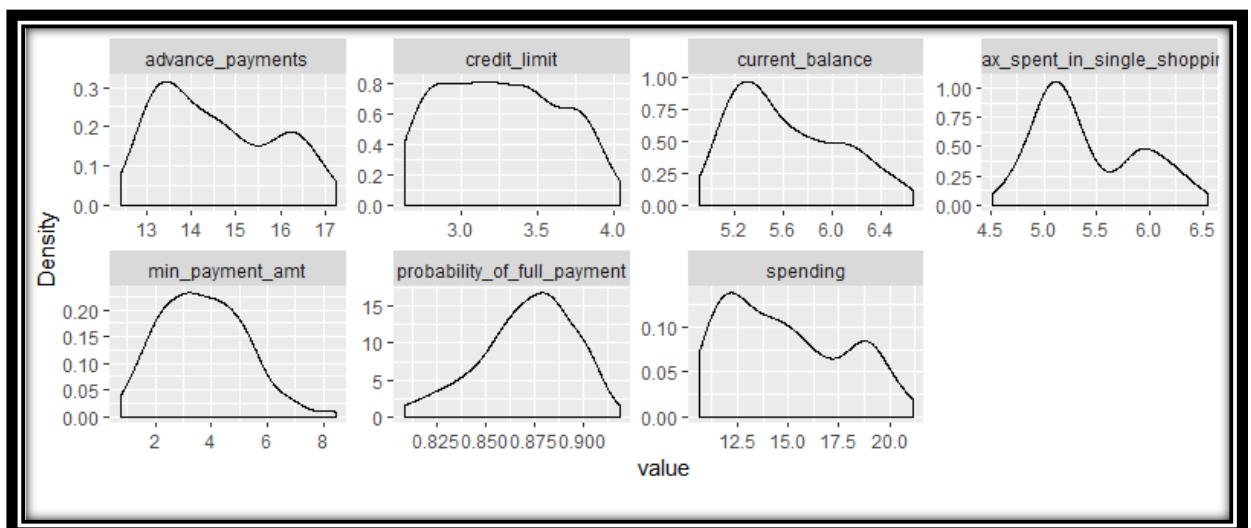


Figure 2: Density Plots

2. Scaling is necessary for this data since values of spending, advance payments etc are in larger amounts than other values. For applying clustering all column values must be uniform. We use **scale** function for the same.
3. For applying hierarchical clustering we need to use the Euclidean method to find the distance between clusters i.e. the **dist** function. Then we use the **hclust** function which uses complete linkage method and plot the dendrogram using the result.

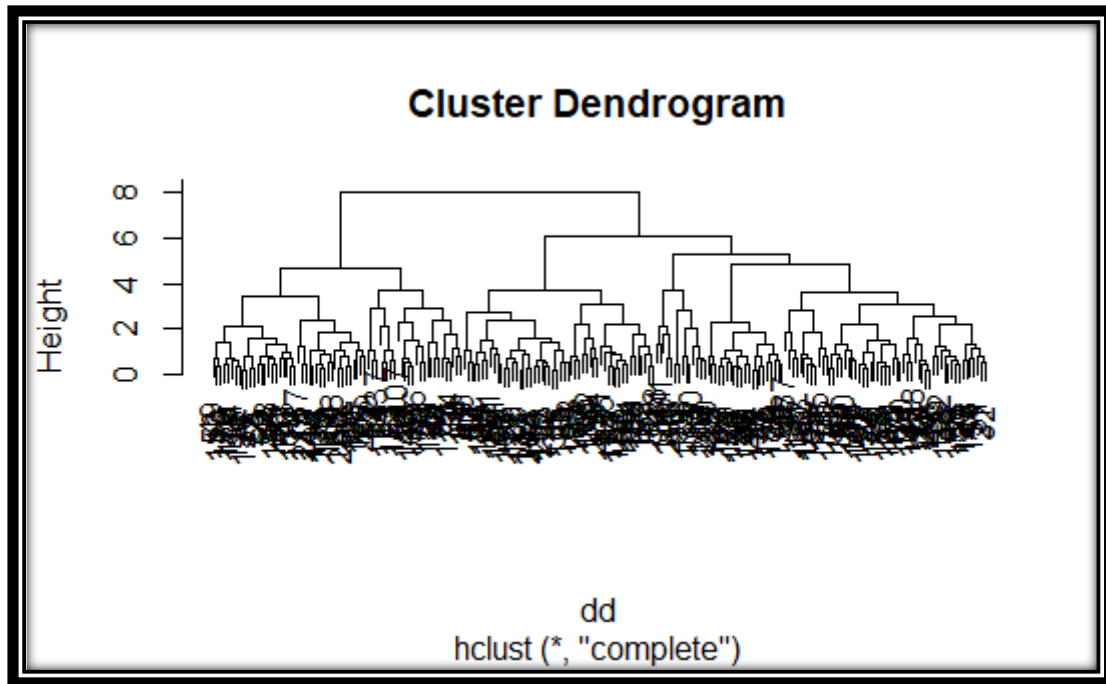


Figure 3: Dendrogram

To find the optimum number of clusters, we can plot the cluster height and see till where the height is dropping much.

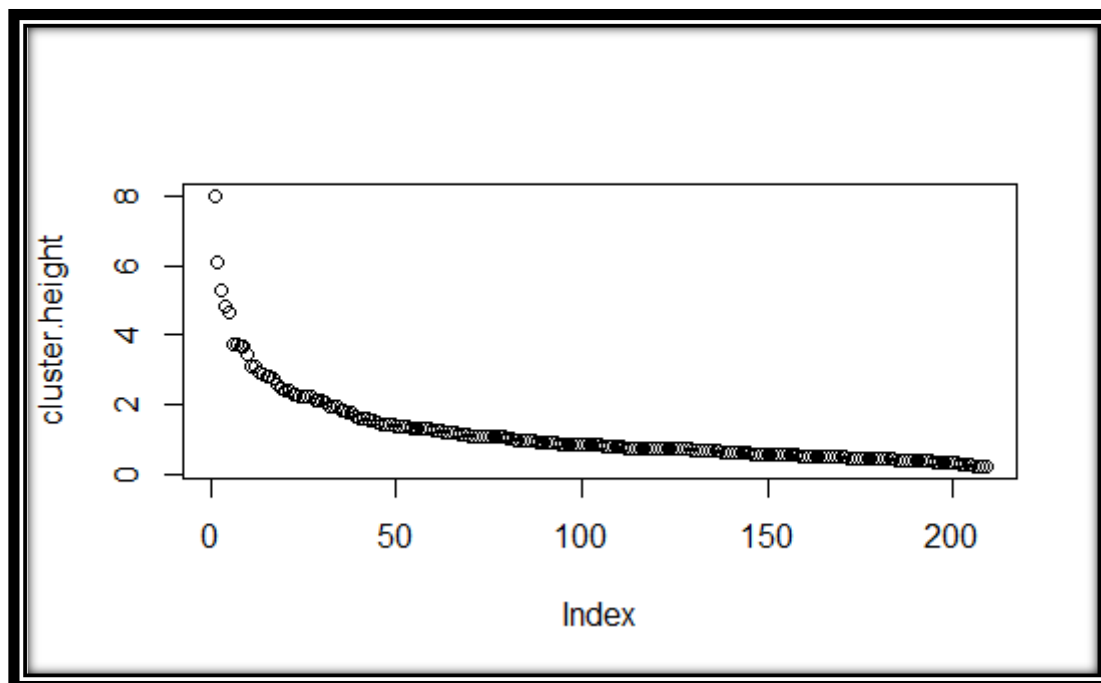


Figure 4: Cluster height plot

We can see after the third cluster the height is not dropping much. So we decide the optimum number of clusters as 3 and plot of dendrogram and cut it into 3 clusters.

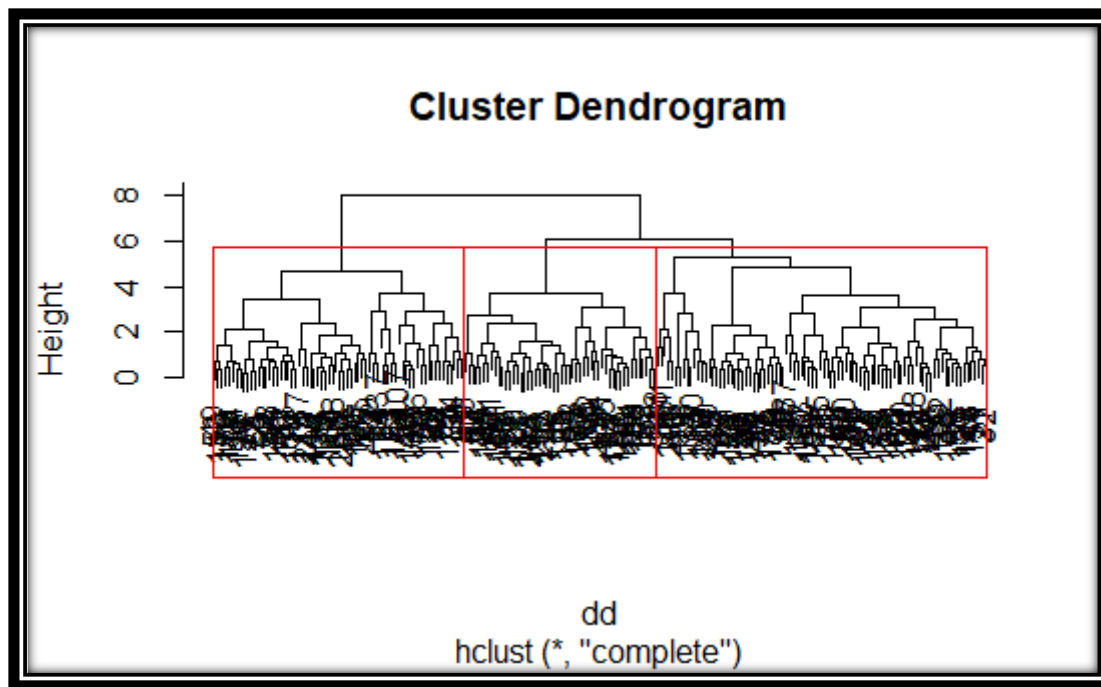


Figure 5: Cluster dendrogram

After this we can cut the data into 3 clusters using **cutree** function and bind the cluster data into the original dataset and group the data according to the clusters.

Then we can use **clusplot** function to plot the data with clusters. We can see 2 of the clusters are overlapping each other.

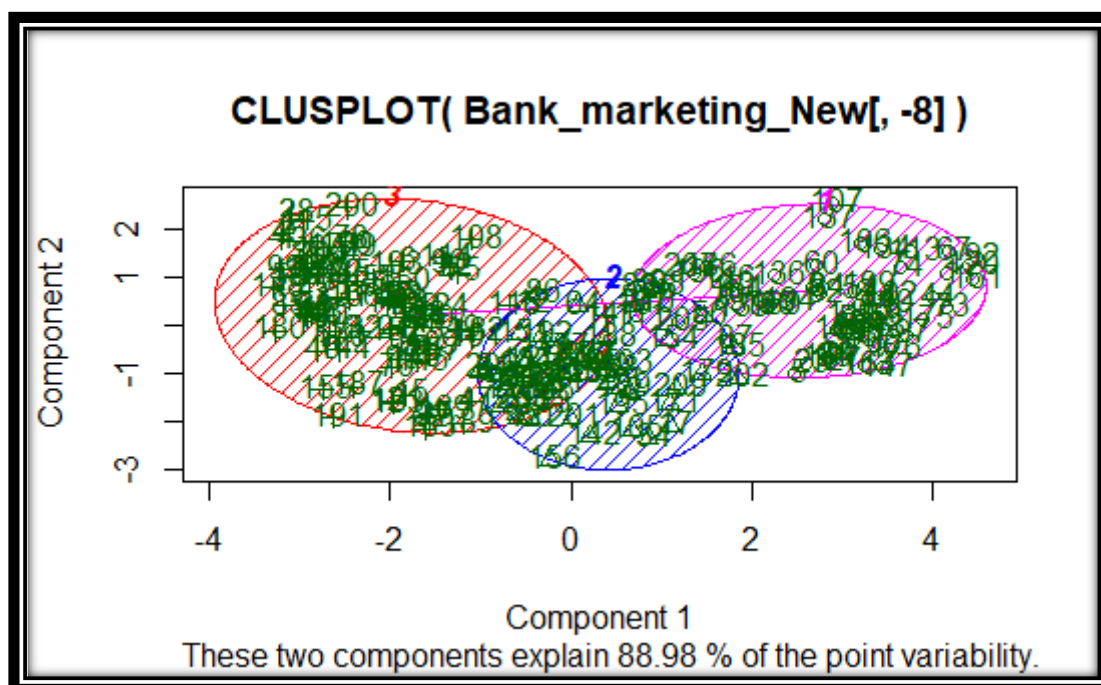


Figure 6: Clusplot

We can use the **aggr** function for profiling the clusters.

```

Group.1    spending advance_payments probability_of_full_payment current_balance
1         1  1.21893721      1.233132514      0.4950193      1.2277398
2         2  0.01545733     -0.002398137      0.4911367     -0.0887064
3         3 -0.92990568     -0.930314532     -0.6577825     -0.8763730
credit_limit min_payment_amt max_spent_in_single_shopping
1         1  1.1138930     -0.03217531      1.2820550
2         2  0.1601511     -0.81490533     -0.4377948
3         3 -0.9341398      0.49514443     -0.7157157

```

Figure 7: Aggr function output

From the output, we can conclude that

Cluster 1: Spending, Advance payments, current balance, credit limit and max spent in single shopping are more. So they are high income and high spending people.

Cluster 2: Spending is average. Probability of full payment and credit limit are high. They are low income, but spend wisely on need basis people.

Cluster 3: Everything is low. Probability of spending, advance payments, full payment, credit limit and expenditure are low. So they are low income and low spending people

We can calculate the silhouette width of each cluster using **silhouette** function.

Then we can bind the sil_width to original data to form the final dataset.

```

> summary(sil)
silhouette of 210 units in 3 clusters from silhou
eting_New)) :
  Cluster sizes and average silhouette widths:
      68      52      90
0.4210697 0.4623712 0.3210709
Individual silhouette widths:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2260  0.3175  0.4553  0.3884  0.5178  0.6151

```

Figure 8: Silhouette summary

We can see most of the sil width are between 0 and 1 which indicates that most of the sample values are allocated to correct clusters.

For cluster 1, a typical strategy would focus certain promotional efforts for the high value customers.

For cluster 2, they are not spending enough also, maybe due to low income or other reasons — further analysis of these segments could lead to insights on the satisfaction / dissatisfaction of these customers.

For cluster 3, where both the income and annual spend are low, further analysis could be needed to find the reasons for the lower spend and price-sensitive strategies could be introduced to increase the spend using credit cards.

4. To apply K means clustering, we can either use the silhouette method or wss plot to know the optimum number of clusters.

First we have to scale the data using **scale** function since some of the values like spending and advance payments are much higher than the other values.

After scaling, we can use silhouette and kmeans methods and find the average silhouette score of each value of k. Then we can plot the values as given below.

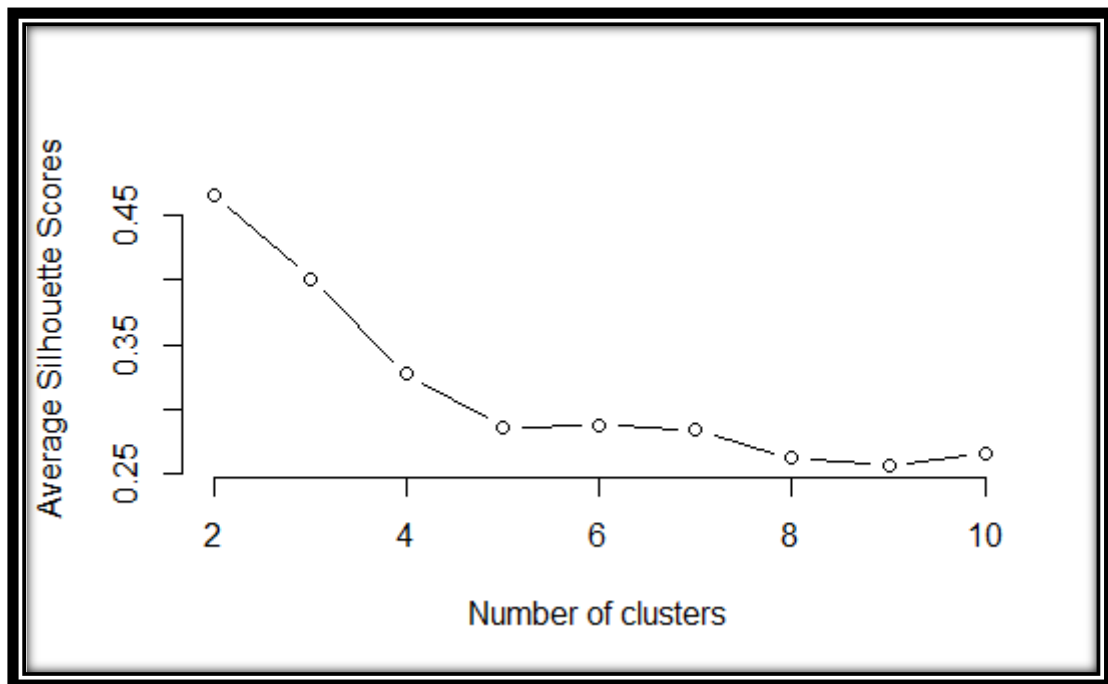


Figure 9: Silhouette Method

From the diagram we can understand the maximum average silhouette score is for $k=2$, so the optimum number of clusters according to this is 2.

To plot wss plot we will use the **wssplot** function and the output looks like below.

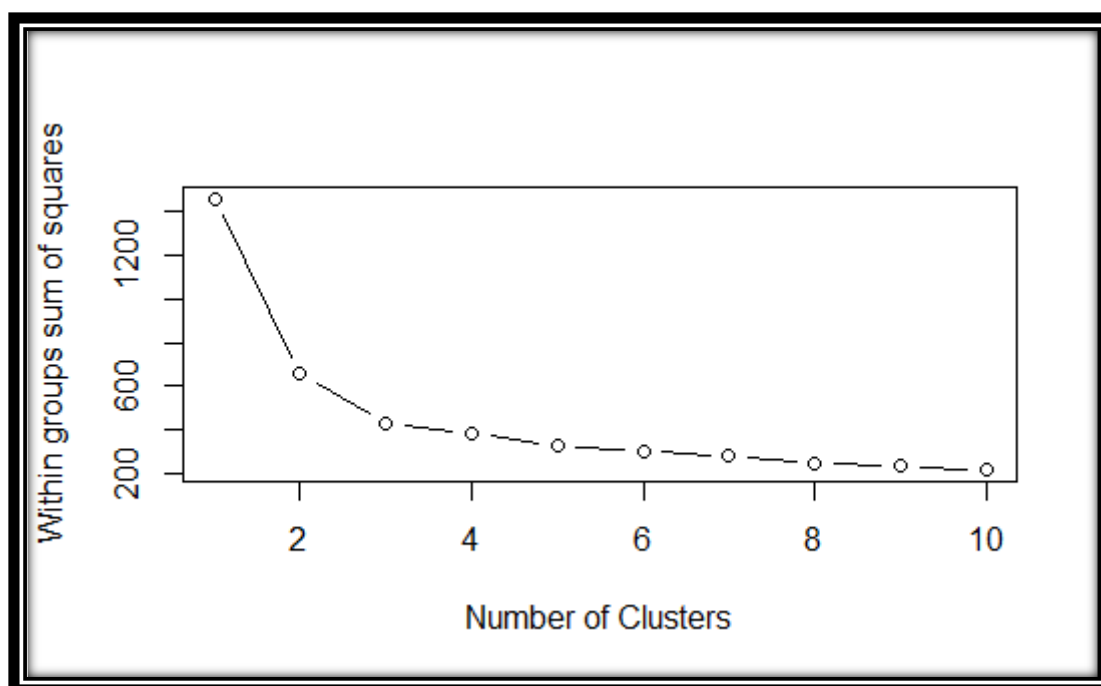


Figure 10: WSS Plot

From the diagram, we can see that the values doesn't change much from $k=2$, i.e. the bend of the knee, so the optimum number of clusters are 2.

We can create the clusters using the **kmeans** function and plot using **clusplot** function as given below.

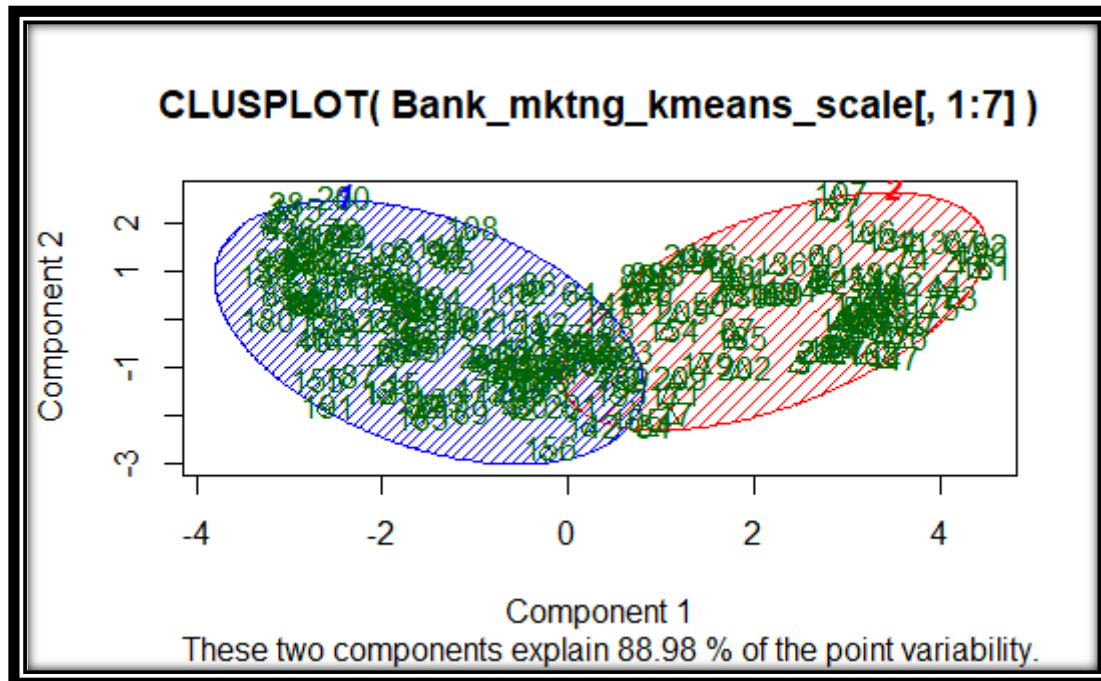


Figure 11: K means Clusplot

We can use the **aggregate** function to find the mean values of each cluster.

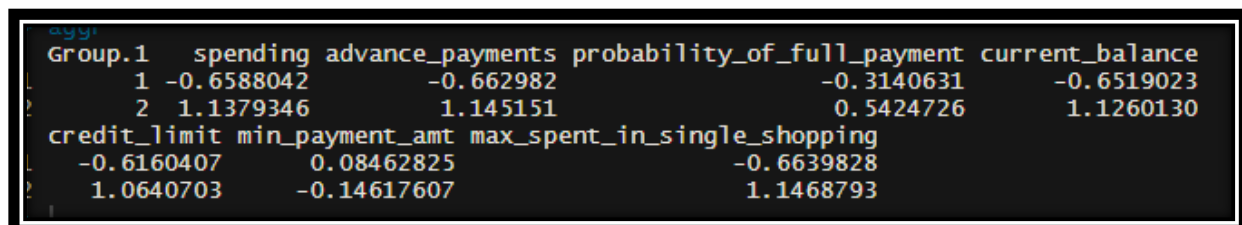


Figure 12: Aggregate – Kmeans

From the output we can understand that

Cluster 1: They are low income and low spending population. Here both the income and annual spend are low, further analysis could be needed to find the reasons for the lower spend and price-sensitive strategies could be introduced to increase their expenditure using credit cards.

Cluster 2: They are high income and high spending population. A typical strategy would focus certain promotional efforts for the high value customers.

Then we can find the silhouette coefficient of each cluster using the silhouette function and the summary. All the coefficients seems to be between 0 and 1, which indicates the clustering has been done perfectly.

```

> summary(sil)
Silhouette of 210 units in 2 clusters from silhouette
t(Bank_mktng_kmeans_scale) :
  Cluster sizes and average silhouette widths:
      133      77
0.4577194 0.5266485
Individual silhouette widths:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.05783 0.40164 0.53310 0.48299 0.59749 0.68470
> ##Final data
  
```

Figure 13: Sil Summary - Kmeans

Then we can form the final output using the original data, clusters and their sil widths.

5. Problem 2 Responses

- For Basic data summary we can use the function **summary** in R, which provides us with mean, median etc of each column. **Dim** function provides us with the dimensions of the data. **Str** function provides us with the structure of the data. To check if any blank values are there we use **is.na** function. The data has 3000 rows and 10 columns. Also none of the values are null. Except some fields like Age, duration, sales etc, other fields are all factors.
- We can use **createdatapartition** function to split the data by 70:30 ratio. Libraries **rpart** and **rpart.plot** are used to calculate CART model.

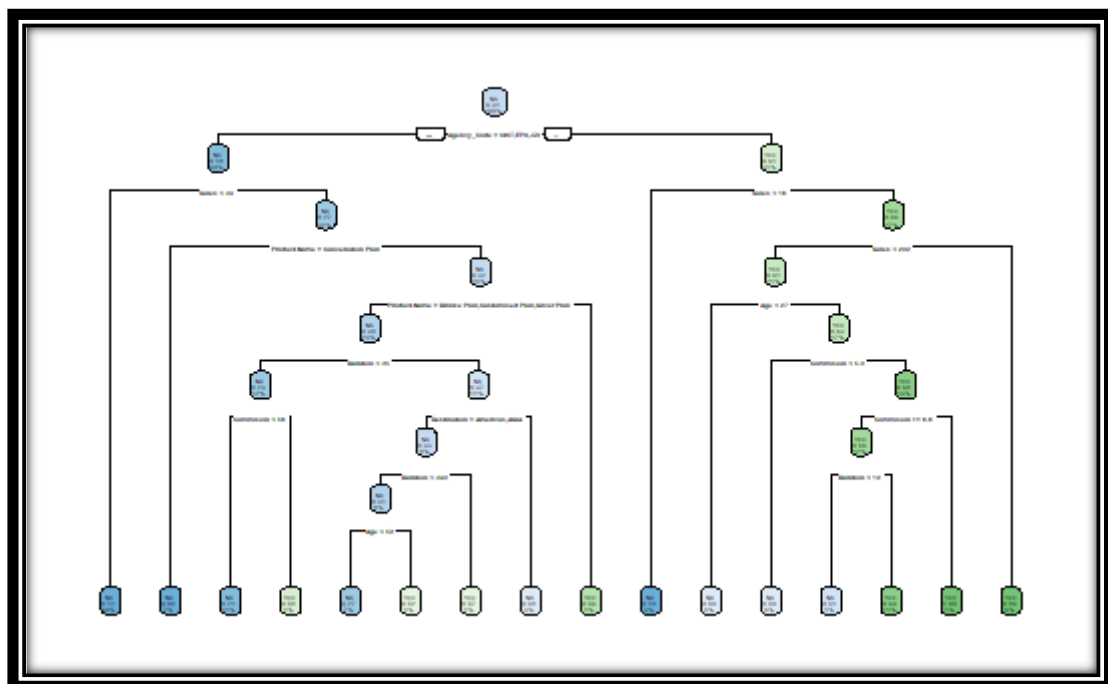


Figure 14: Rpart plot

After the model is formed, we will prune the data (because the xerror seems to be increasing) using **prune** function and predict the test and train data Claimed values (dependent variable) using this pruned tree.

We will use the **ROCR** library for calculating the performance of the model.

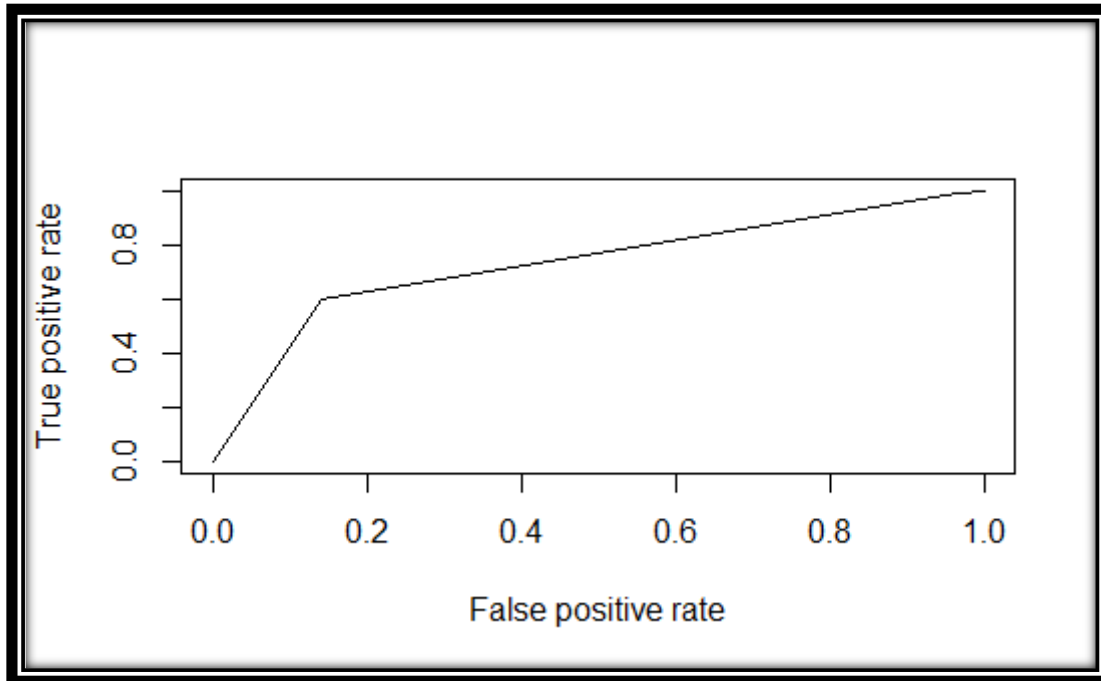


Figure 15: CART Model plot for dev data

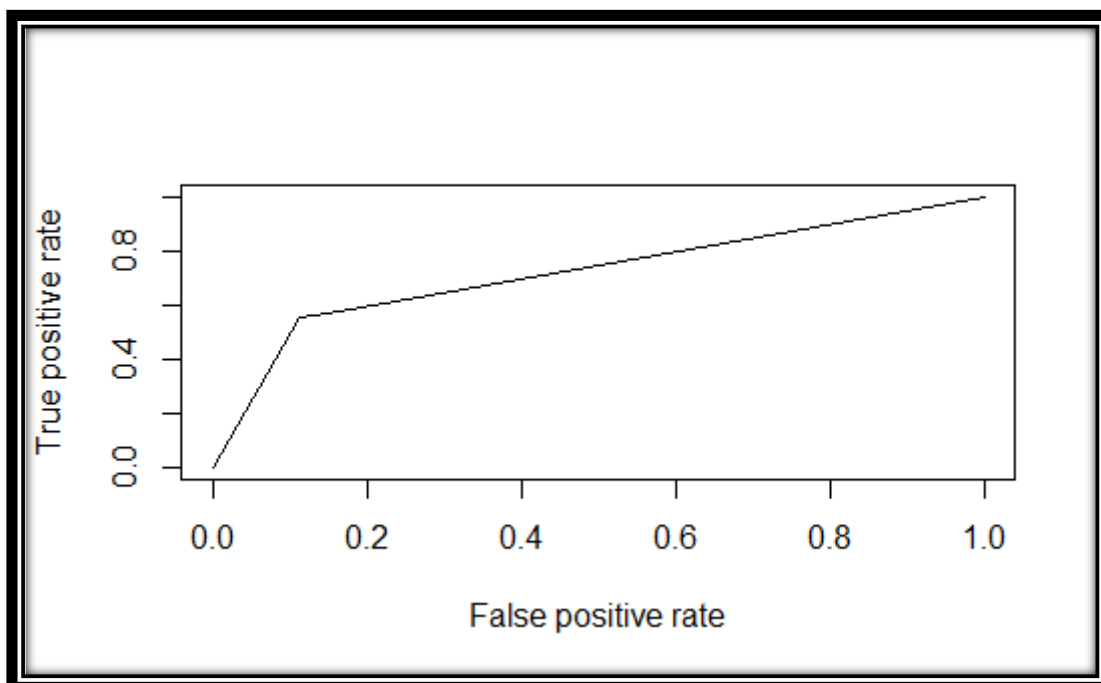


Figure 16: CART Model plot for holdout data

Area under the curve is calculated using **auc** function and it comes around 73% both data which is not that high.

```
Confusion Matrix and Statistics

      Reference
Prediction No  Yes
No      1247  259
Yes     207   388

      Accuracy : 0.7782
      95% CI   : (0.7598, 0.7958)
No Information Rate : 0.6921
P-Value [Acc > NIR] : < 2e-16

      Kappa : 0.4678

McNemar's Test P-Value : 0.01815

      Sensitivity : 0.8576
      Specificity : 0.5997
Pos Pred Value : 0.8280
Neg Pred Value : 0.6521
Precision : 0.8280
Recall : 0.8576
F1 : 0.8426
Prevalence : 0.6921
Detection Rate : 0.5935
Detection Prevalence : 0.7168
Balanced Accuracy : 0.7287

'Positive' Class : No
```

Figure 17: CART - Confusion Matrix for training data

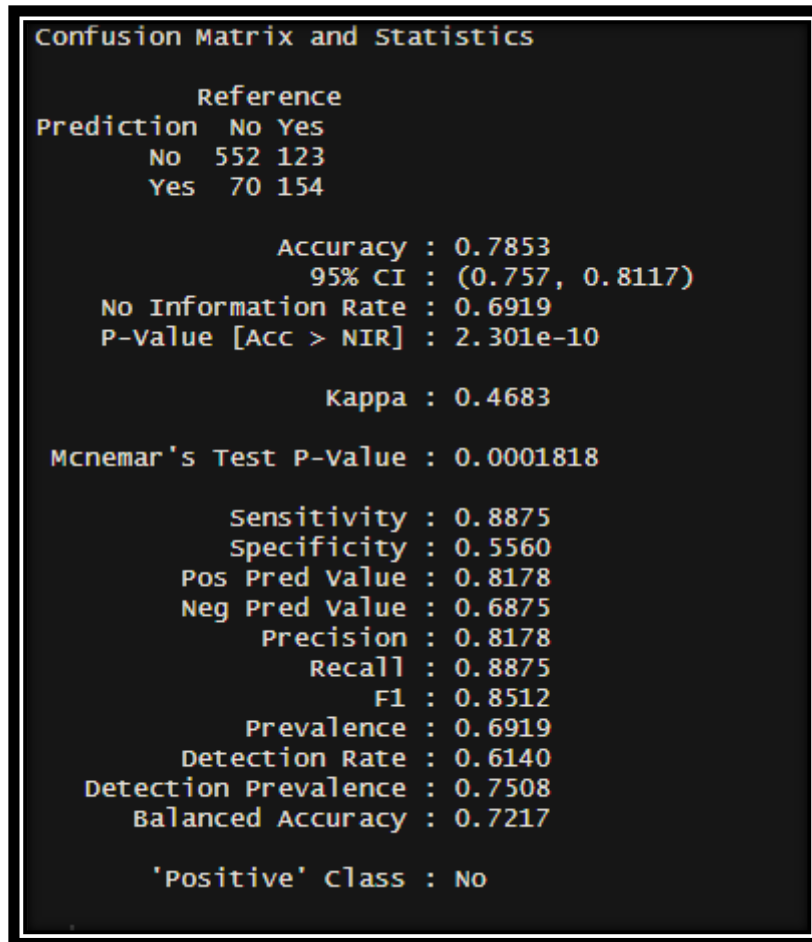


Figure 18: Confusion Matrix for holdout data

From the confusion matrix, we can see false positives and negatives are high leading to Sensitivity and Precision being high. Also specificity and accuracy is not that good leading to indicate that the model is not that good.

For Random forest model, we use the function **randomForest** to form the model and the RF plot looks like below. The error is given around 23% by the model.

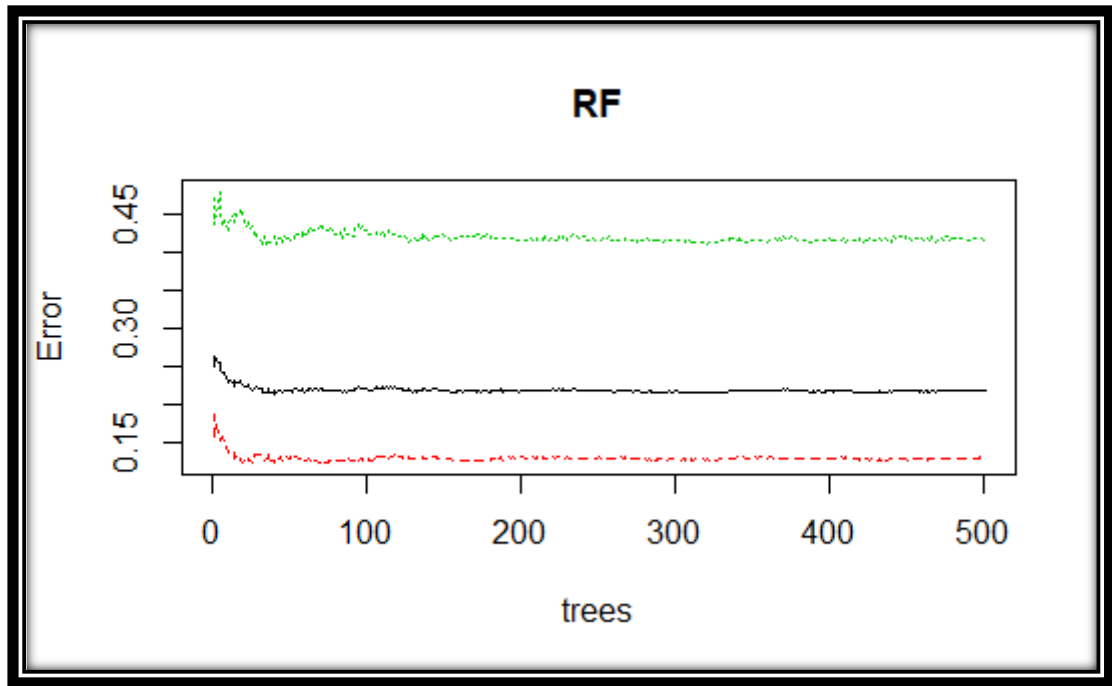


Figure 19: RF Model Plot

We can calculate importance of the variables using **importance** function.

```
> importance(RF)
```

| | No | Yes | MeanDecreaseAccuracy | MeanDecreaseGini |
|--------------|------------|-------------|----------------------|------------------|
| Age | 10.6023026 | -10.6362663 | 2.725390 | 81.085949 |
| Agency_Code | 4.4778847 | 34.1764588 | 26.338030 | 92.877855 |
| Type | -2.3803637 | 6.2678903 | 4.845326 | 5.594110 |
| Commision | -0.3219155 | 22.7132092 | 21.545583 | 64.640605 |
| Channel | 13.7854194 | 8.4524537 | 16.710357 | 5.569226 |
| Duration | -6.2671623 | 30.7653705 | 24.338931 | 117.647622 |
| Sales | -3.4336006 | 41.8919587 | 38.788176 | 120.612568 |
| Product.Name | 9.3139454 | 30.4742276 | 36.012304 | 83.826852 |
| Destination | 1.4768335 | 0.3877856 | 1.847242 | 11.964402 |

Figure 20: RF Model Importance

Sales has the max mean decrease in accuracy, so that's the most important variables in this data. Destination is the least important variable here.

Now we have to tune the RF model, using the **tuneRF** function. After tuning the plot looks like below.

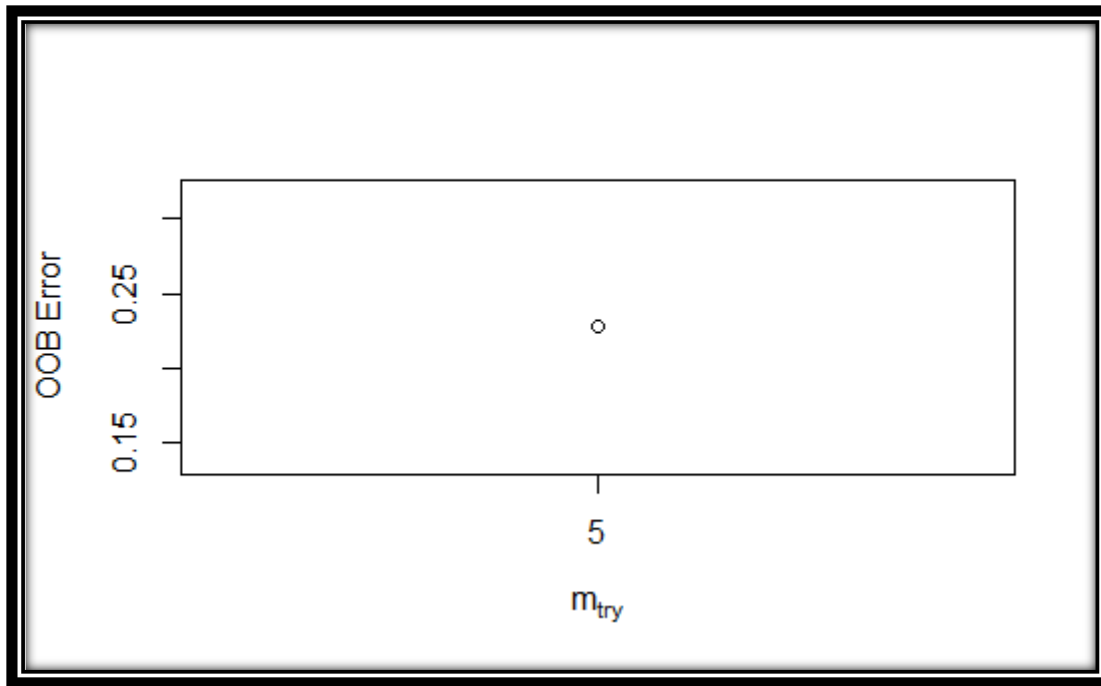


Figure 21: TuneRF Plot

After this we can predict and plot the graph for RF model. For test data the graph looks like below.

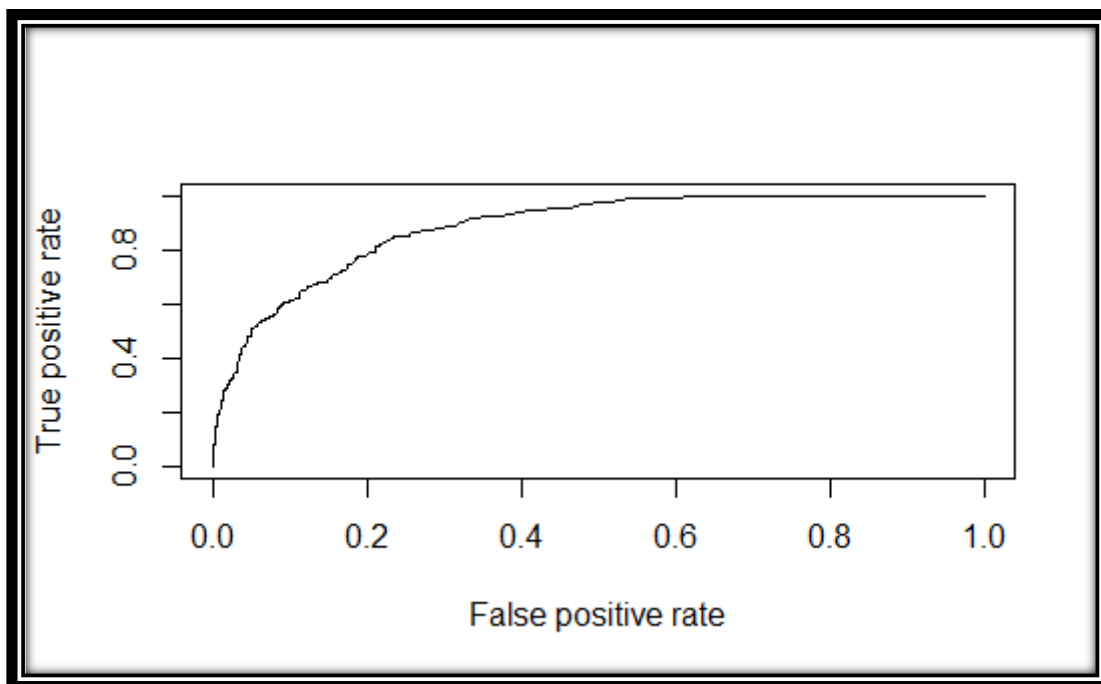


Figure 22: RF dev data plot

And auc value comes around 88.34% which is good for the model. The confusion matrix is

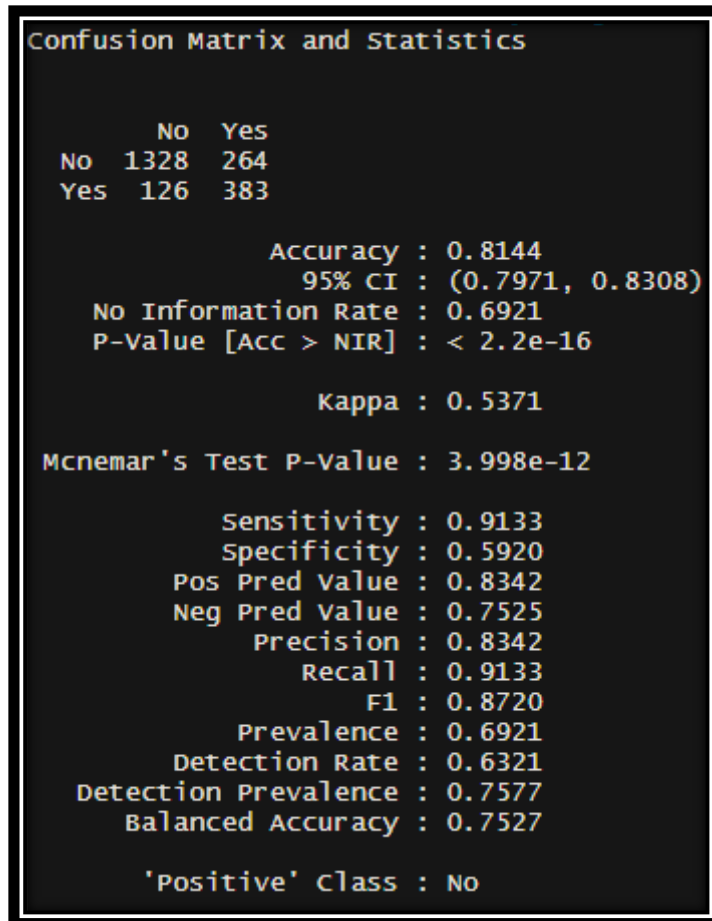


Figure 23: Confusion Matrix for dev data – RF

From confusion matrix we can Sensitivity and precision are high because false positives and negatives are high. Also specificity is low and Accuracy is medium. For training data the plot comes like below.

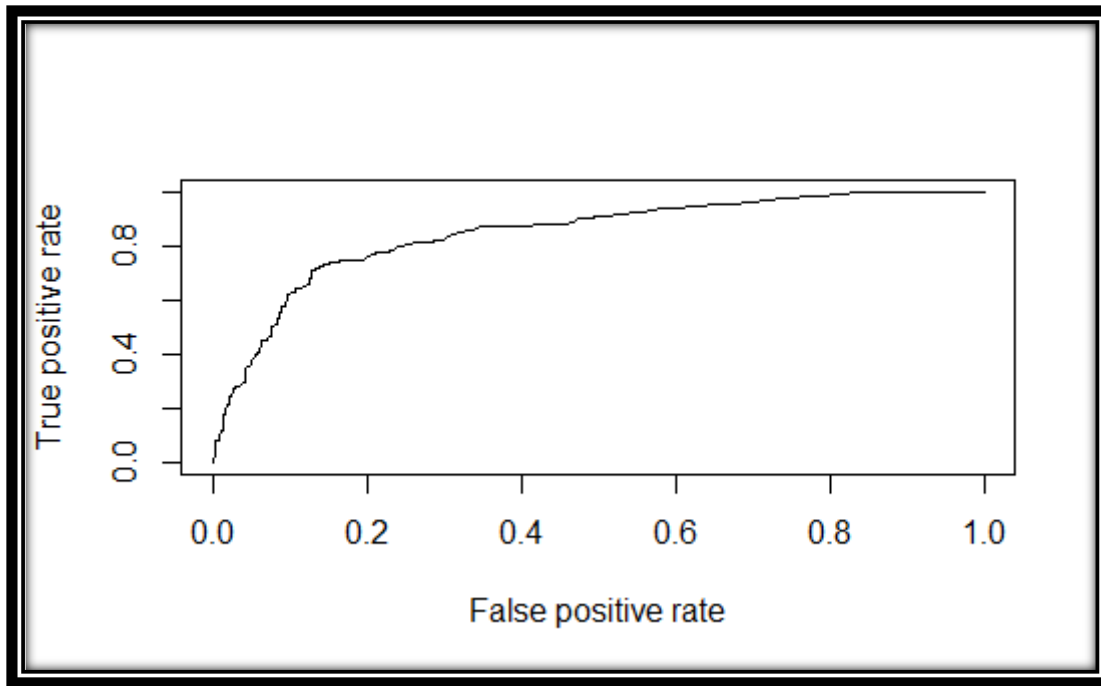


Figure 24: Holdout data RF plot

AUC value comes around 84.5% which is good. Confusion matrix looks like below.

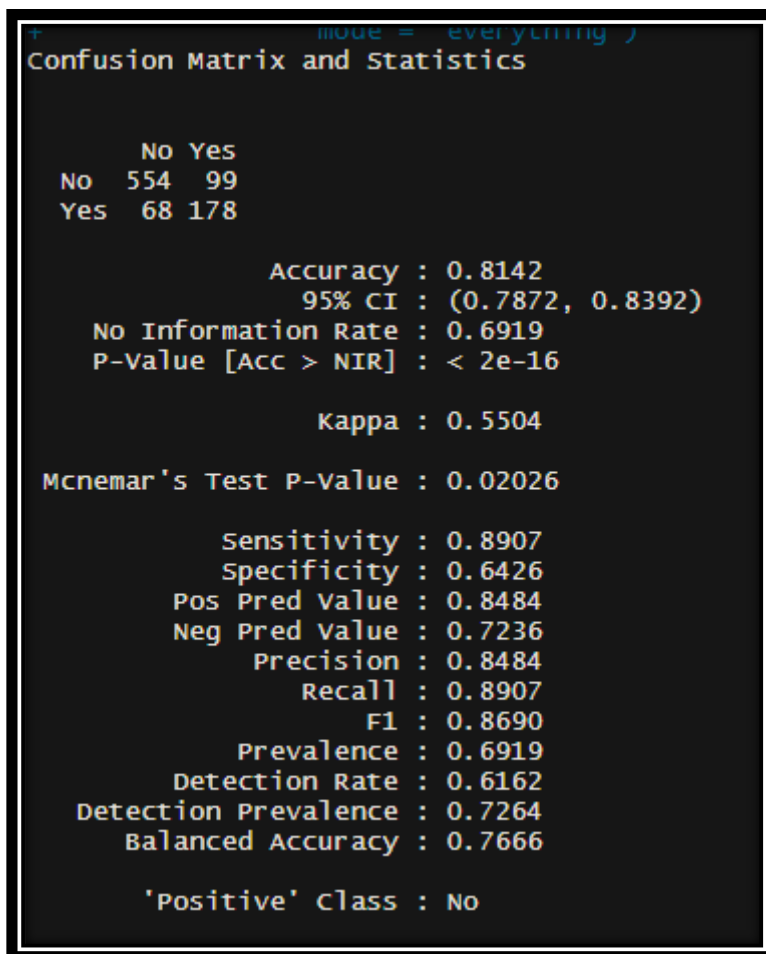


Figure 25: Confusion Matrix for holdout – RF

The specificity, sensitivity, accuracy and precision values seems similar to training data.

For ANN Model, we had to scale the training and test data first using **scale** function. We use the library **neuralnet** for the calculations. To apply ANN, all the variables should be in integer/numerical form, so we use **model.matrix** function to do the same.

Then we use **neuralnet** function to form the model. After plotting the model, it looks like this

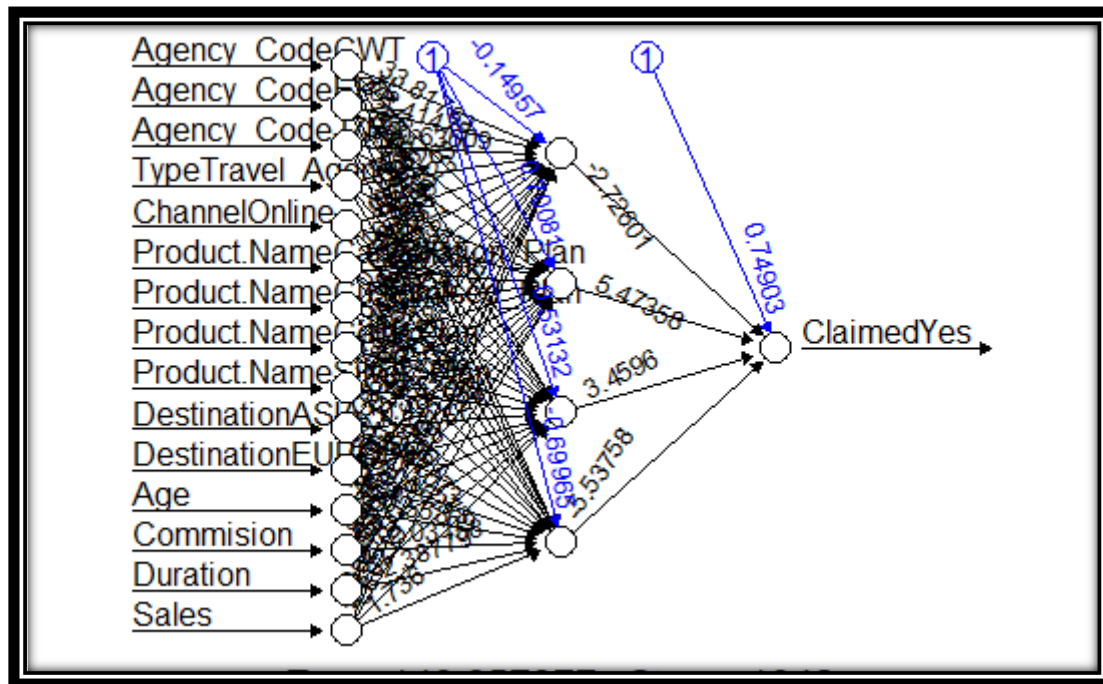


Figure 26: ANN model plot

We will assign probabilities to the dev sample and use the **quantile** function to see the distribution of estimated probabilities. Then we can assign 0 or 1 according to whether the probability is greater than 0.5 or not.

We will apply the same in test dataset also.

Then we will calculate the prediction and performance and plot the graph of dev data as below.

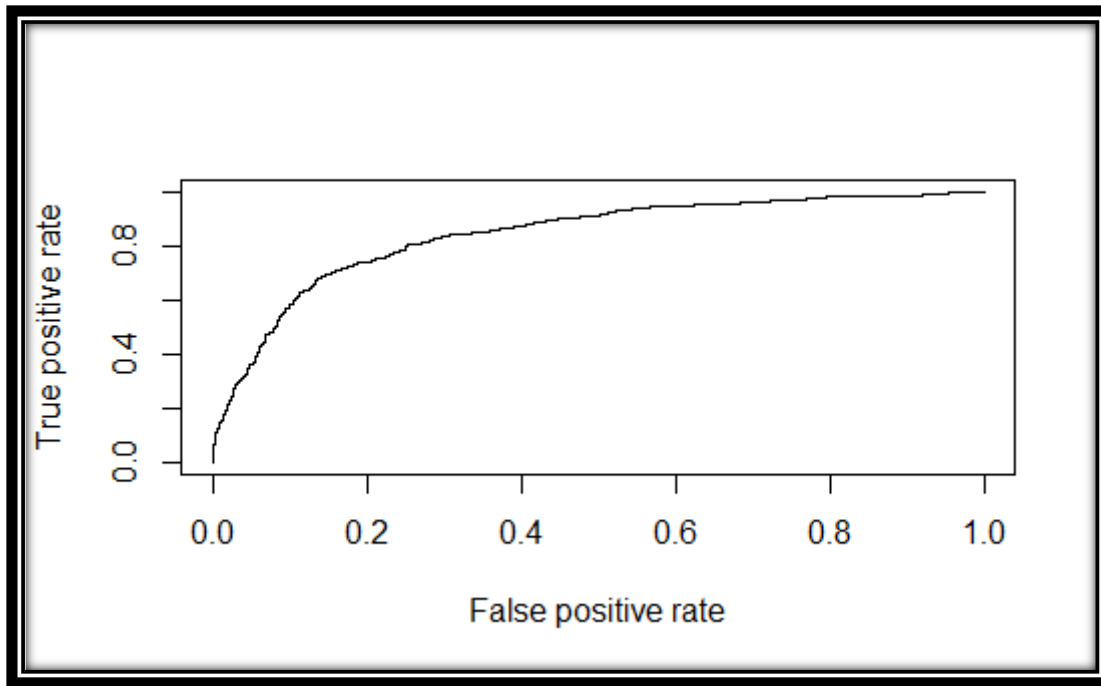


Figure 27: ROC plot for dev data – ANN

The auc value comes around 84% which is good. The confusion matrix for dev data is

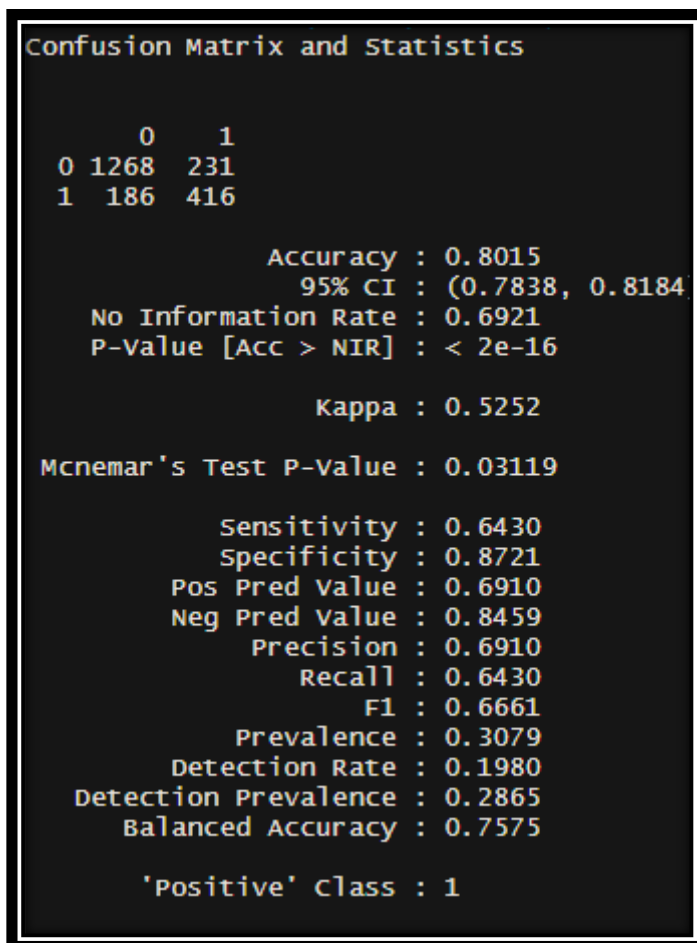


Figure 28: Confusion matrix for dev data – ANN

Here we can see the sensitivity and precision are low and accuracy and precision are high compared to other models.

We can apply the same for holdout data also.

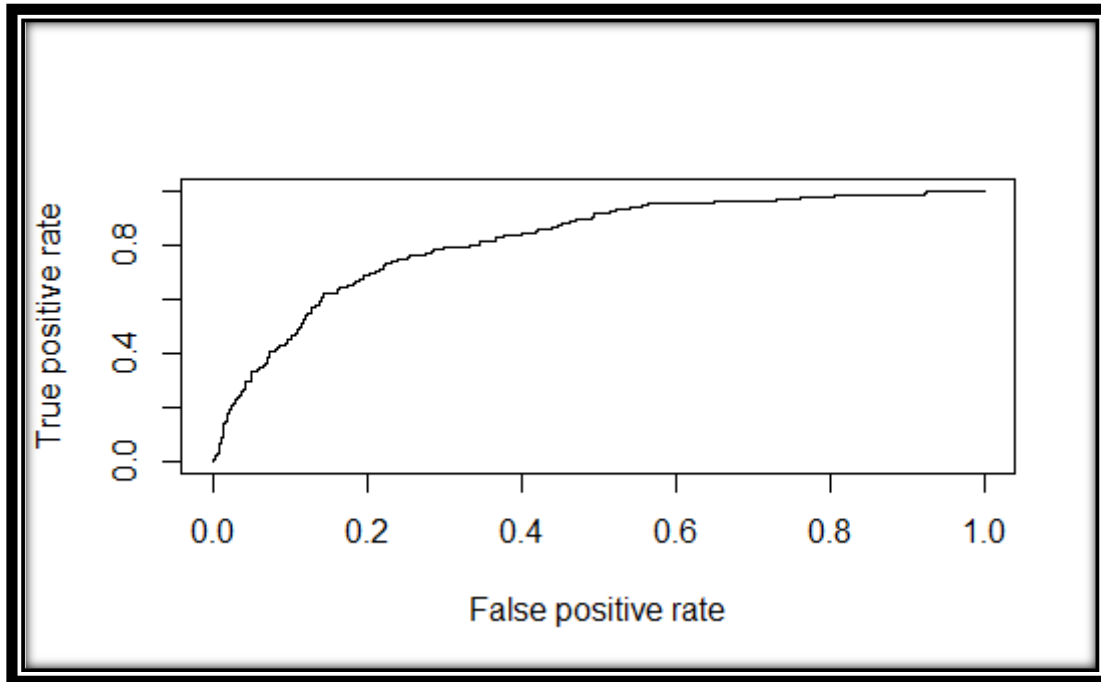


Figure 29: ROCR plot for holdout data- ANN

The auc is 81.4% which is good. The confusion matrix is

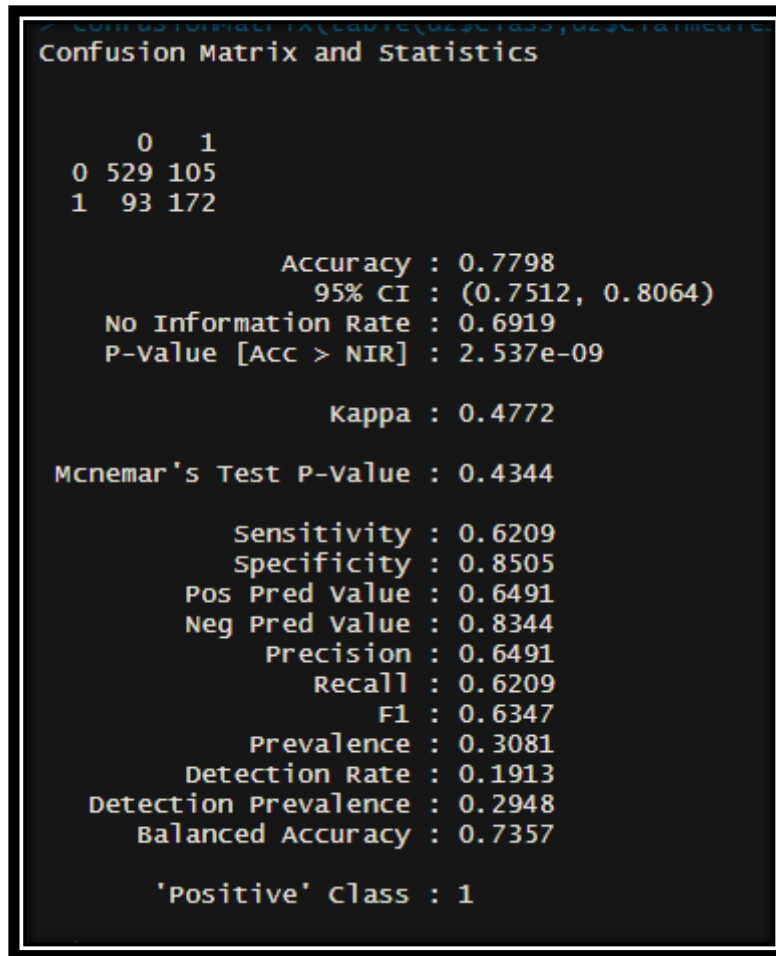


Figure 30: Confusion Matrix for holdout data – ANN

Here as similar to dev data, sensitivity and precision is low indicating less false positives and negatives. Also specificity and accuracy is good compared to other models.

3. The performance and confusion matrix are covered in the above section.
4. For comparing all the models, we can plot the accuracy of each model for dev data using a histogram and the output looks like below.

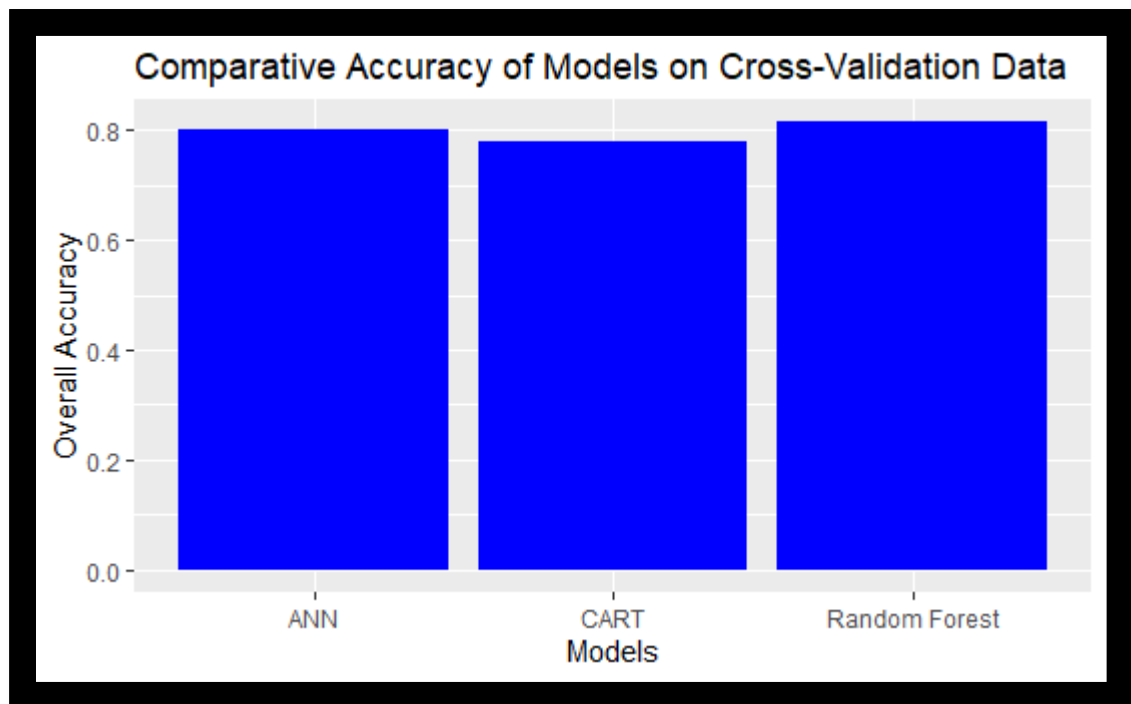


Figure 31: Compare Models

From the diagram, we can understand that random forest is slightly better than other two in accuracy.

- Using the random forest model, we can plot the importance of each variable using the function **varImpPlot**. The output is

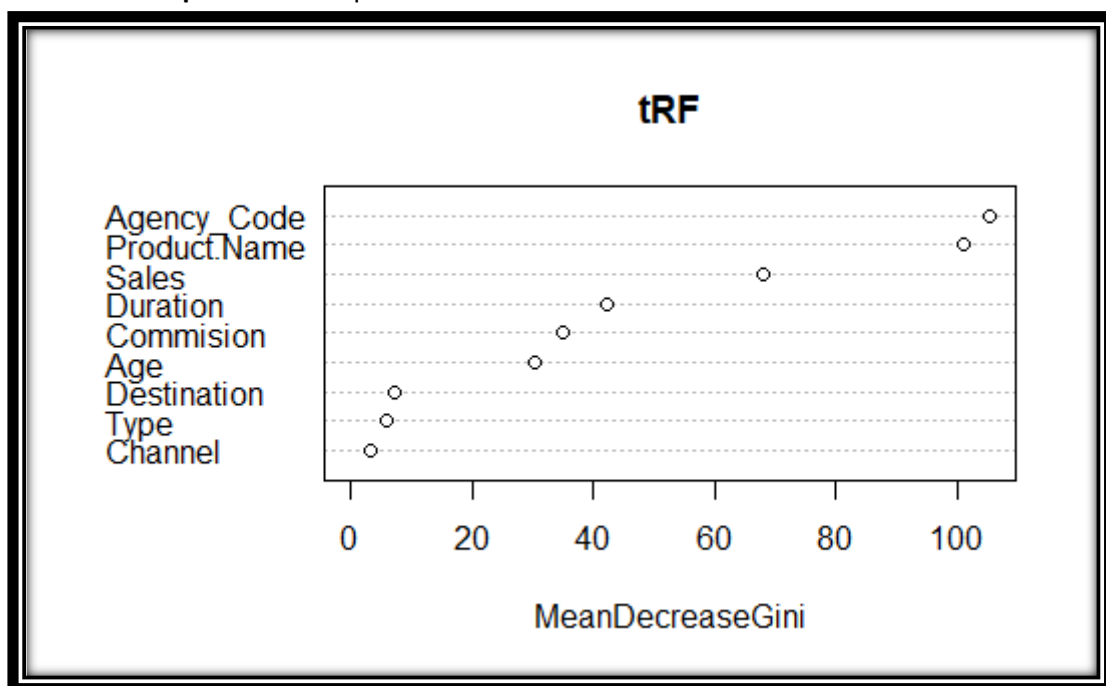


Figure 32: VarImpplot – RF

The values are given below

```
> varImp(trf)
      overall
Age      30.291896
Agency_Code 105.411191
Type      5.867619
Commision  35.065416
Channel    3.242510
Duration   42.228493
Sales      67.861912
Product.Name 101.118169
Destination  7.181903
```

Figure 33: Importance values

From the output, we can understand that agency code and product name are the most important variables of the data which helps in predicting the claim status.

Appendix A

R code is attached along with the report.