Decision Tree Analysis

GERMAN CREDIT DATA

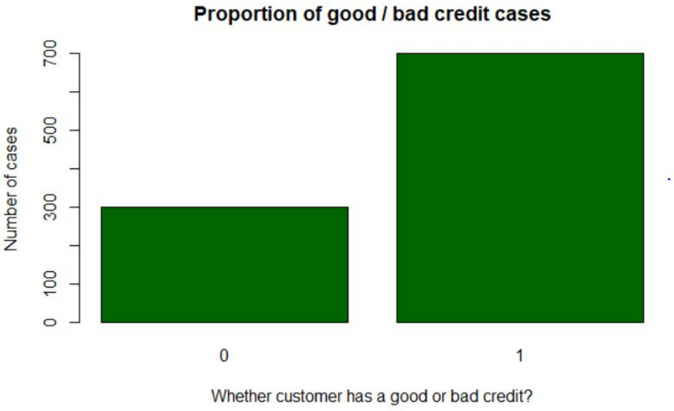
Honey Salve

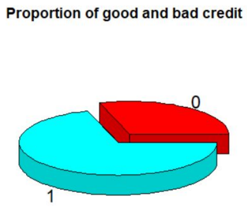
Question 1

Data Exploration:

➢ What is the proportion of "Good" to "Bad" cases?

By using '=COUNTIF (AD1: AD1002,1)' formula in the data file, we can see that there are 700 positive responses for credit score i.e. good cases whereas the rest 300 of the 1000 responses are negative i.e. bad cases.





As we can see, the proportion of good to bad cases among the 1000 applicants is:

700 / 300 = 2.33

➢ Are there any missing values – how do you handle these?

The variables with missing values are:

- **NEW_CAR (Binary):** If the customer has a new car (1-Yes, N/A – No)

sum(is.na(GCdata$NEW_CAR)) = 766 NAs - drop the variable.

- **USED_CAR (Binary):** If customer has a used car (1-Yes, N/A – No)

sum(is.na(GCdata$USED_CAR)) = 897 NAs - drop the variable.

- **FURNITURE (Binary):** If customer has furniture (1-Yes, N/A – No)

sum(is.na(GCdata$FURNITURE)) = 819 NAs - drop the variable.

- **`RADIO/TV (Binary):** If customer has Radio or TV (1-Yes, N/A – No)

sum(is.na(GCdata$`RADIO/TV`)) = 720 NAs - drop the variable.

- **EDUCATION (Binary):** If customer is educated (1-Yes, N/A – No)

sum(is.na(GCdata$EDUCATION)) = 950 NAs - drop the variable.

- **RETRAINING (Binary):** If customer had retraining (1-Yes, N/A – No)

sum(is.na(GCdata$RETRAINING)) = 903 NAs - drop the variable.

Inherently, we replaced the missing values for the above variables with 0.

- **PERSONAL_STATUS (Categorical):** Whether the customer is single, married or divorced (1-Single, 2-Married/Widowed, 3-Divorced)

sum(is.na(GCdata$PERSONAL_STATUS)) = 310 NAs

Since there are too many missing values to discard the variable altogether, we'll try adding an additional category, say 4 – Other.

- **AGE (Integer):**

sum(is.na(GCdata$AGE)) = 9 NAs – Since it's a very small chunk of missing data, we'll be deleting these 9 rows.

➢ Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-values attributes, frequencies of different category values.

Name	Data type	Min	Max	Mean	Median	sd	Frequency
CHK_ACCT	Numeric						0 - 274; 1 - 269; 2 - 63; 3 - 394
DURATION	Numerical	4	72	20.903	18	12.0588	
HISTORY	Categorical						0 - 40; 1 - 49 2 - 530; 3 - 88 4 - 293
NEW_CAR	Categorical						0 - 766; 1 - 234
USED_CAR	Categorical						0 - 897; 1 - 103
FURNITURE	Categorical						0 - 819; 1 - 181
RADIO/TV	Categorical						0 - 720; 1 - 280
EDUCATION	Categorical						0 - 950; 1 - 50
RETRAINING	Categorical						0 - 903; 1 - 97
AMOUNT	Numerical	250	18424	3271.15	2319.5	2822.6	
SAV_ACCT	Categorical						0 -603; 1 - 103; 2 - 63; 3 - 48; 4 - 183

EMPLOYMENT	Categorical						0 - 62; 1 - 172;
							2 - 339; 3 - 174;
							4 - 253

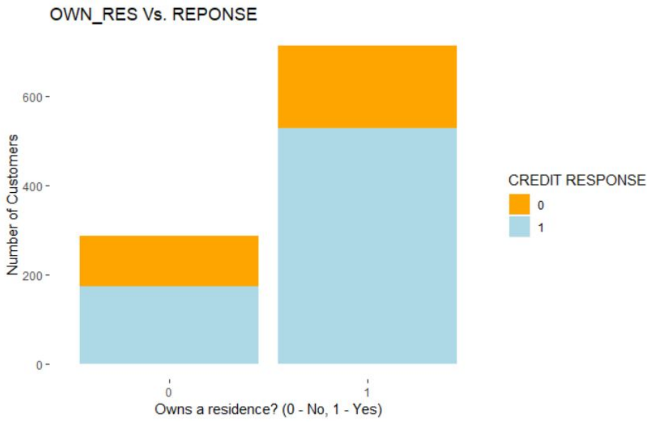
INSTALL_RATE	Numerical	1	4	2.973	3	1.118	

PERSONAL_STATUS	Categorical						0 - 310; 1 - 548; 2 - 92; 3 - 50
GUARANTOR	Categorical						0 - 948; 1 - 52
PRESENT_RESIDENT	Categorical						1 - 130; 2 - 308; 3 - 149; 4 - 413
REAL_ESTATE	Categorical						0 - 718; 1 - 282
PROP_UNKN_NONE	Categorical						0 - 864; 154
AGE	Numerical	19	75	35.47	33	11.32	
OTHER_INSTALL	Categorical						0 - 814; 1 - 186
RENT	Categorical						0 - 821; 1 - 179
OWN_RES	Categorical						0 - 287; 1 - 713
NUM_CREDITS	Numerical	1	4	1.4	1	0.577	
JOB	Categorical						0 - 22; 1 - 200; 2 - 630; 3 - 148
NUM_DEPENDENTS	Categorical						1 - 845; 2 - 155
TELEPHONE	Categorical						0 - 596; 1 - 404
FOREIGN	Categorical						0 - 963; 1- 37
RESPONSE	Categorical						0 - 300; 1 - 700

Examining Variable Plots

- **OWN_RES Vs. RESPONSE**

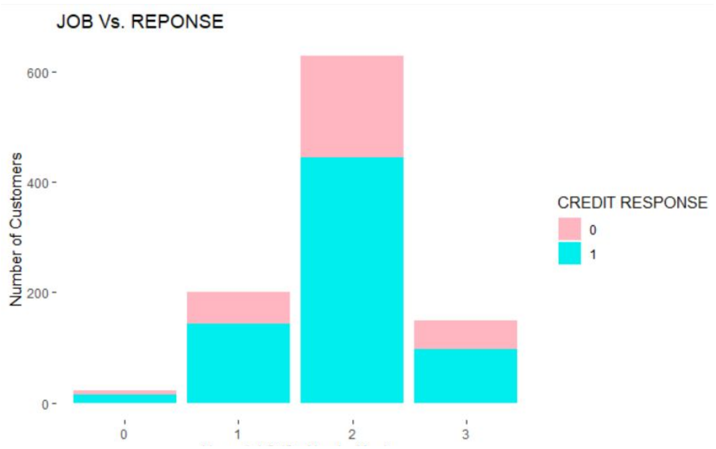
Customers who own their own residence present a good standing on their money payback guarantee. They can fall back on their houses as fixed assets when they fail to make payments for the credit loan.



As shown in the graph, such customers have a proportionately better amount of good credit responses as compared to ones who don't. Evidently, owning a residence has a significant impact on determining whether the customer has good or bad credit.

- **JOB Vs. RESPONSE**

An applicant's job/source of income weighs a ton on his/her final credit score. This variable categorizes them based on their skill level and type of employment as:



0: unemployed/ unskilled nonresident

1: unskilled - resident

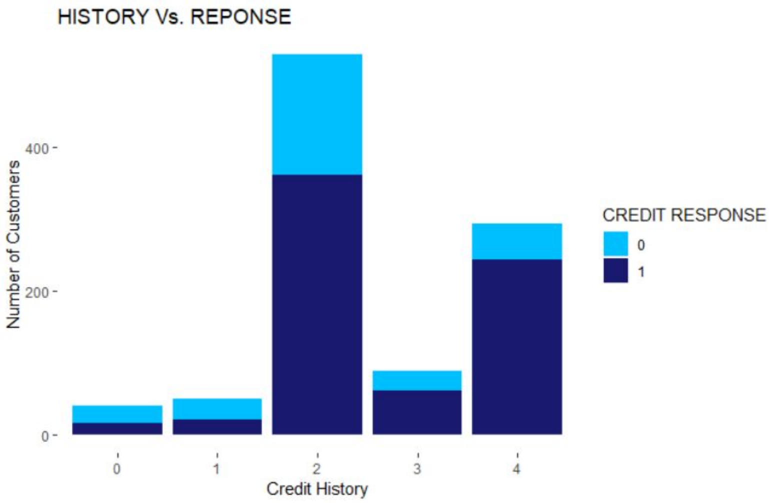
2: skilled employee/official

3: management/self-employed/highly qualified employee/ officer

Based on the graphical representation of the scenario, we can conclude that the number of good credits is much higher for skilled & employed applicants whereas it's the least for unemployed individuals. But surprisingly, as opposed to general belief, the proportion of bad credits is highest for skilled employees.

- **HISTORY Vs. RESPONSE**

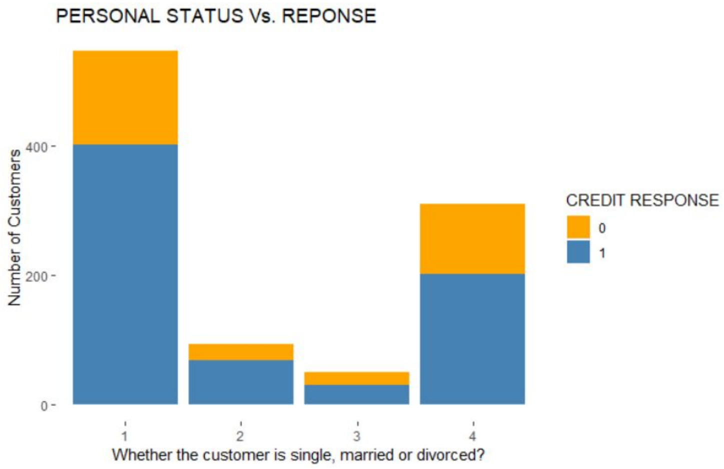
Credit history is something that defines your financial status and hence it'll most definitely impact one's credit score. Analyzing the variable with the output 'RESPONSE' led us to the conclusion as below:



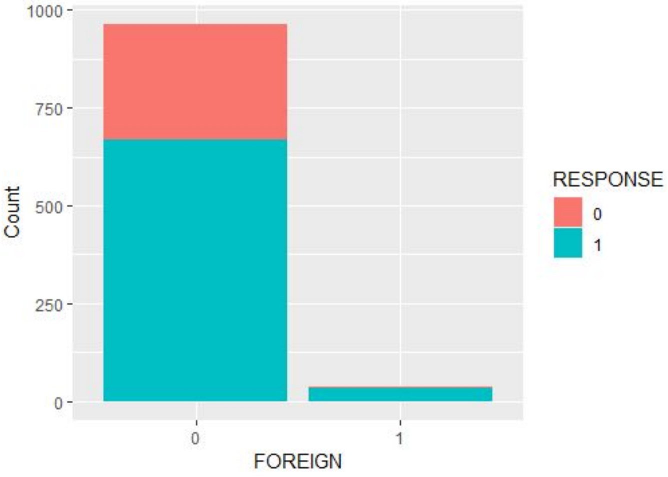
There are 4 types of credit history:
0: no credits are taken
This type has the least probability of having a good credit score.
1: all credits are paid
The proportion of bad to good responses is much higher i.e. 0.468 than that for a critical account i.e 0.205 (type 4). This is contradictory to the normal notion that paying back money on time keeps you in good books.

- **PERSONAL STATUS Vs. RESPONSE**

As per the graph showed, being single sure does pay off when it comes to having a good credit score. On the other hand, an interesting observation shows that the proportion of bad to good credit responses for divorced people (0.66) is higher than that for single & married ones (0.36).



- **FOREIGN Vs. RESPONSE**

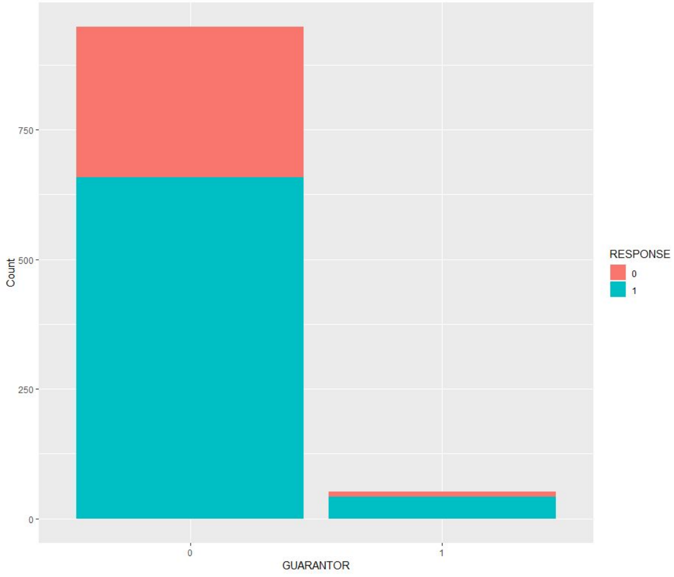


The percentage of local applicants with a poor credit rating is higher than that of foreign workers. Therefore, the probability of a local applicant defaulting on their loan is higher than that of a foreign applicant. However, the data set of foreign applicants is too small for this to be conclusive.

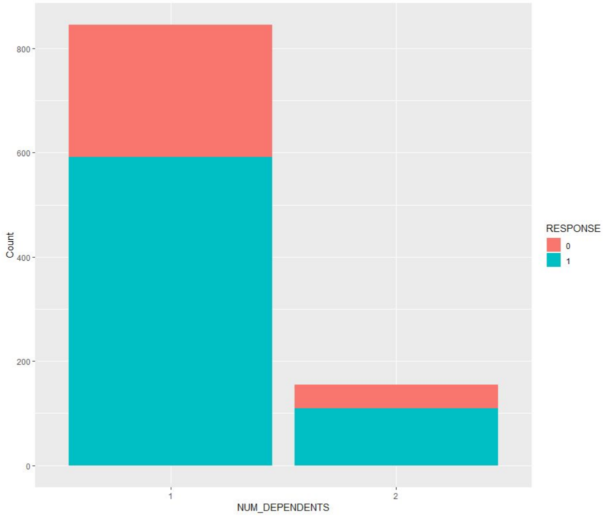
- **GUARANTOR vs. RESPONSE**

A good amount of people who do not have guarantors have good credit. We see this similar trend in the people who have guarantors where the number of people with good credit is high.

However, the number of people with no guarantors is much higher (948) than the people with a guarantor (52).



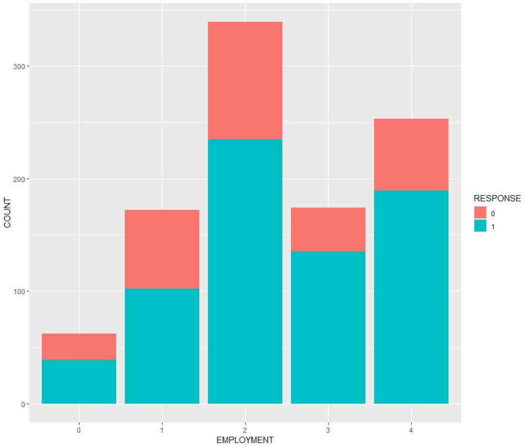
- **NUM_DEPENDENTS vs. RESPONSE**



The number of people with 1 dependent is much higher (845) than people with 2 dependents (155). Both of these categories have a high number of good credit customers.

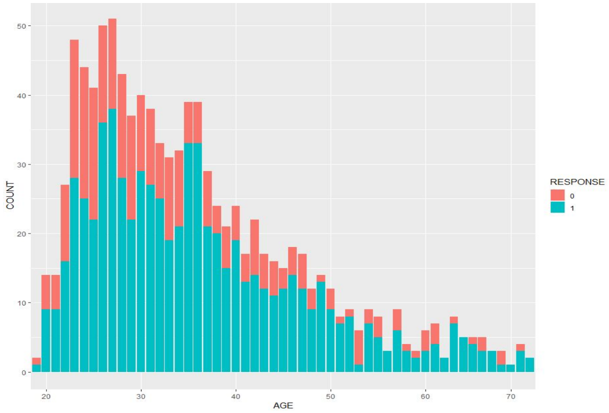
- **EMPLOYMENT vs. RESPONSE**

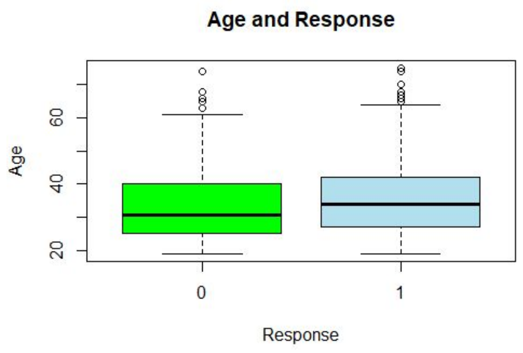
The number of good credit customers is fairly higher than those with bad credit in all categories of employment. Due to this similar trend in all categories, this variable may not be suitable for our model.



- **AGE vs. RESPONSE**

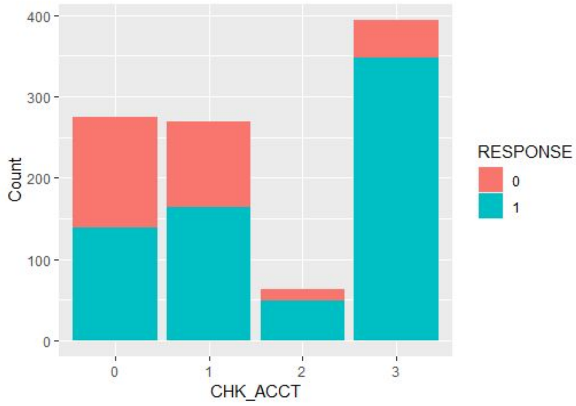
Majority of the customers are between the ages 25 to 40. We also see that a high number of bad credit customers between the ages of 20 and 35 as compared to older age groups, where the





majority of the customers qualify as good credit customers.

- **RESPONSE VS CHECKING ACCOUNT STATUS**

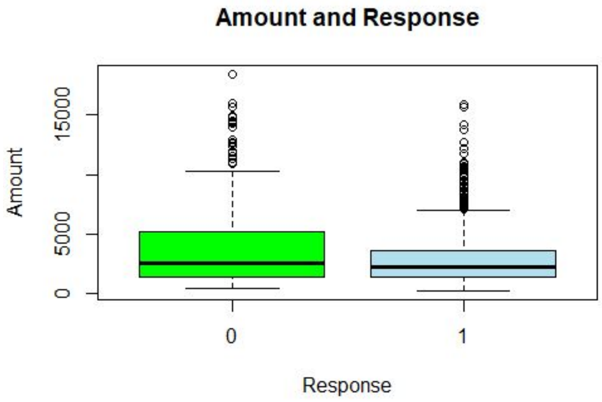


For applicants with Checking Account Status as “0”: The percentage of applicants with Credit Rating, ‘Good’ and ‘Poor’ are equal.
For applicants with Checking Account Status as “1”: The percentage of applicants with Credit Rating as ‘Good’ is slightly greater than that of ‘Poor’.
For applicants with Checking Account Status as “2”: The percentage of applicants with Credit Rating as ‘Good’ is significantly greater than that of ‘Poor’ however, the number of applicants in this category is relatively lower (approx overall 60).
For applicants with Checking Account Status as “3”: The percentage of applicants with Credit Rating as ‘Good’ is significantly greater than that of ‘Poor’ and the number of applicants in this category is high (approx overall 390).

The highest number of applicants with Credit Response as “Good” is from the Checking Account Status “3” category. (Around 340).

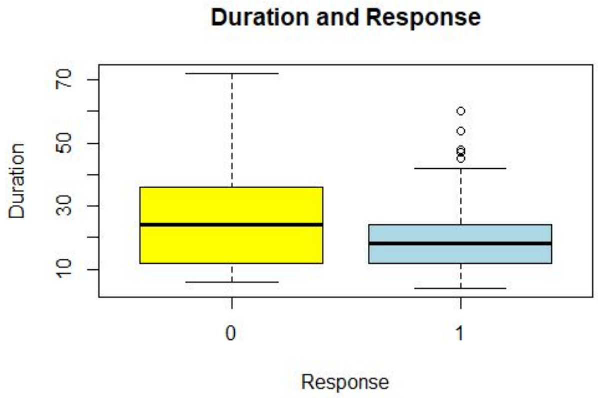
- **RESPONSE VS AMOUNT**

Half the applicants with “Poor” credit rating have credit amount due between $2000 and $5000 with median at $2500. 50% of applicants with “Good” credit rating have amount due between $2000 and $3000. Also, both the variables have a lot of outliers. Hence, the data is very spread.



- **RESPONSE VS DURATION**

50% of applicants with “Poor” credit rating have average duration of credit (in months) between 12 and 36 months, with a median value 24 months. While, half the applicants with credit rating “good” have average duration of credit (in months)



between 12 and 20 months, with median value of 16 months.

From initial data exploration, we can say the following about these parameters :

- CHK_ACCT: Checking account status is responsible for making the creditor Good or Bad
- DURATION: Higher duration results in lower credit score and hence, bad credit
- HISTORY: Credit history has significant relationships with the Response
- AMOUNT: Higher the amount, higher the chances of being having Bad credit
- SAV_ACCT: Higher the value in savings account, higher chances of having good credit
- REAL_ESTATE: People who own Real Estate tend to have good credit
- AGE: Higher the age, higher the chances of being a good creditor
- RENT and OWN_RES : Customers who rent usually have bad credit as compared to people who have their own residences
- JOB: Higher skilled job holders generally have good credit
- NUM_DEPENDENTS: People with more dependents tend to have bad credit
- FOREIGN: Foreign nationals generally have good credit

Question 2

➢ (a)

Model1 - Decision tree on full data using rpart:

In order to get a good model, we need to use the following arguments:

- minsplit
- minbucket
- maxdepth

Model2 - with Information Gain

Model3 - Information gain with minsplit = 10, minbucket = 3

Model4 - Information gain with minsplit = 20, minbucket = 10

Model5 - with Gini Index

Models with parameters	Accuracy	Sensitivity
Information Gain (70% threshold)	78.9%	88.6%
Information gain with minsplit = 10, minbucket = 3 (50% threshold)	68.8%	57.6%
Information gain with minsplit = 10, minbucket = 10 (50% threshold)	67%	55%
Gini Index (50% threshold)	79.4%	88.43%

As we can evidently see above, the decision tree model built using “Gini Index” is the best among these with the highest accuracy of 79.4%. Hence, we can build a decision tree using the Gini Index parameter with a max depth of 10.

➢ (b)

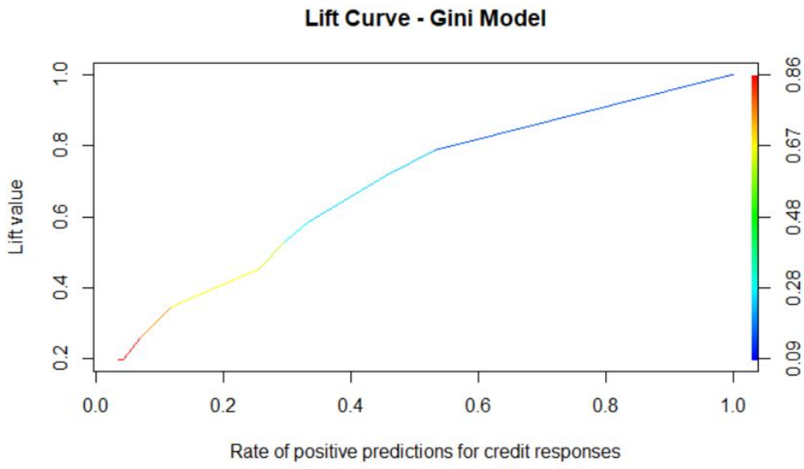
The impactful variables can be found out as: ‘Model$variable.importance’, This shows that the variables - ‘CHK_ACC’, ‘DURATION’, ‘HISTORY’, ‘AMOUNT’ and ‘SAV_ACCT’ are very important when it comes to set the good and bad cases apart.

This the exact same list of variables that we estimated earlier. Hence, it does match the expectations that we built in the previous question.

➢ (c) As discussed above, the accuracy of our model is 79.4%.

The Confusion Matrix, Precision & Recall:

	TRUE 0 (No)	TRUE 1 (Yes)	Precision
Pred. 0	175	81	68.4%
Pred. 1	125	619	83.2%
Recall	58.3%	88.4%	



Lift chart: The lift chart helps us predict how much more likely we are to receive good credit responses if we look at a random sample of applicants. This lift chart, the rate of positive predictions shows us that all the predictions in this model have been made with the highest confidence level of 0.86 but it starts with the lowest confidence level of zero. Also, the threshold at 0.7 succeeds with a positive linear growth rate in the number of good credit scores.

Finally, we don't think this model is the most effective / robust in predicting the credit responses because:

- This model has been built on the whole data set hence, even a small change in the data will slightly, if not completely change the decision tree. In order to prevent this, we believe we must divide the data into two parts - training and testing data which we'll be doing further.
- The decision tree built on the training data will then be tested on its performance with the testing data.

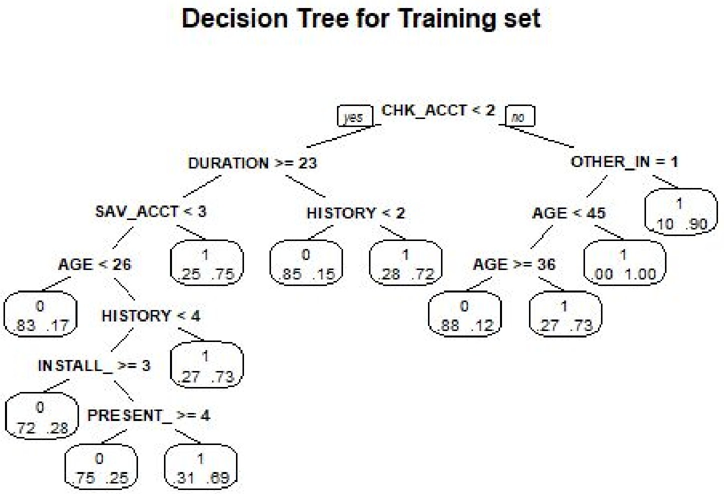
Question 3

➢ (a) Build models

(a) Model 1 : 50 - 50 Data Split with no other conditions and "Gini Split"(default cp - 0.01, default minsplit=20)

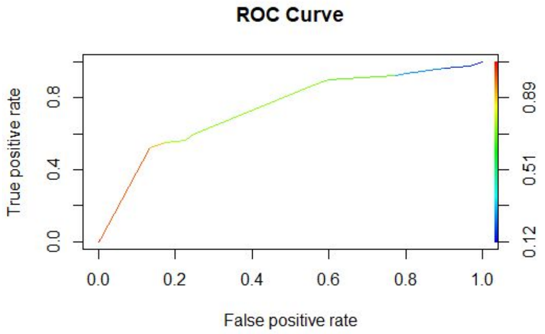
Confusion Matrix for 50-50 Data Split :

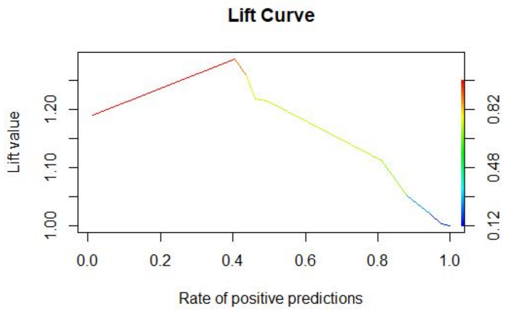
prediction	**Good Cases**	**Bad Cases**
Good Cases	60	35
Bad Cases	90	315



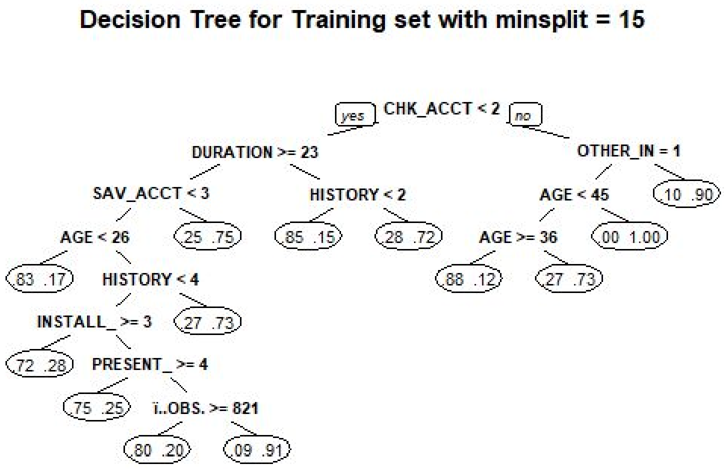
Accuracy Levels for 50 - 50 Data Split :

Accuracy	**Precision**	**Recall**	**F1 Score**
0.75	0.7047	0.65	0 - 0.498 1 - 0.8344





(b)Model 2 : 50 - 50 Data Split with minsplit = 15 "Gini Split" (cp - 0.018)



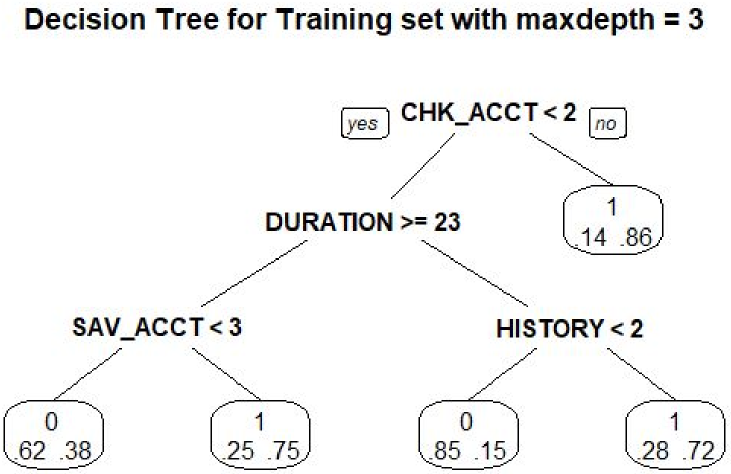
Confusion Matrix :

prediction	Good Cases	Bad Cases
Good Cases	61	35
Bad Cases	89	315

Accuracy Levels :

Accuracy	Precision	Recall	F1 Score
0.77	0.7076	0.6534	0 - 0.4959 1 - 0.8355

(c) Model 3 : 50 - 50 Data Split with maxdepth = 3 "Gini Split" (cp- default, minsplit - default)



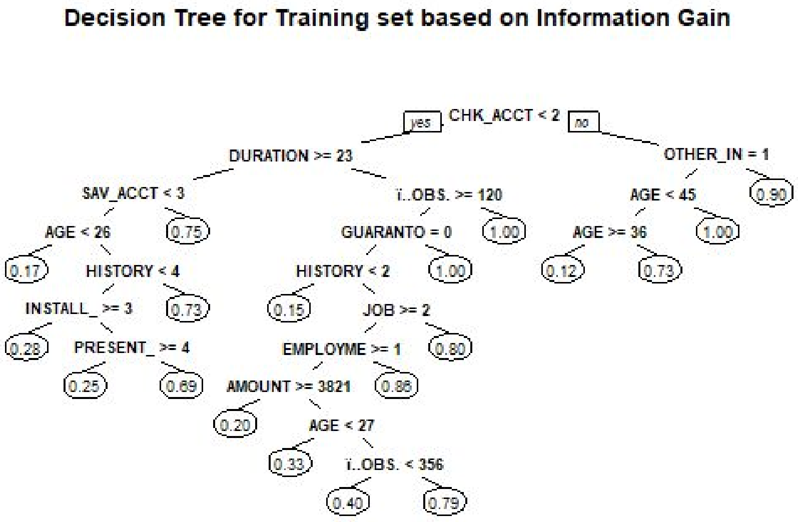
Confusion Matrix :

prediction	Good Cases	Bad Cases
Good Cases	69	40
Bad Cases	81	310

Accuracy Levels :

Accuracy	Precision	Recall	F1 Score
0.73	0.7279	0.6728	0 - 0.5328 1 - 0.8367

(d) Model 4 : 50 - 50 "Information Split" (cp- default, minsplit - default)



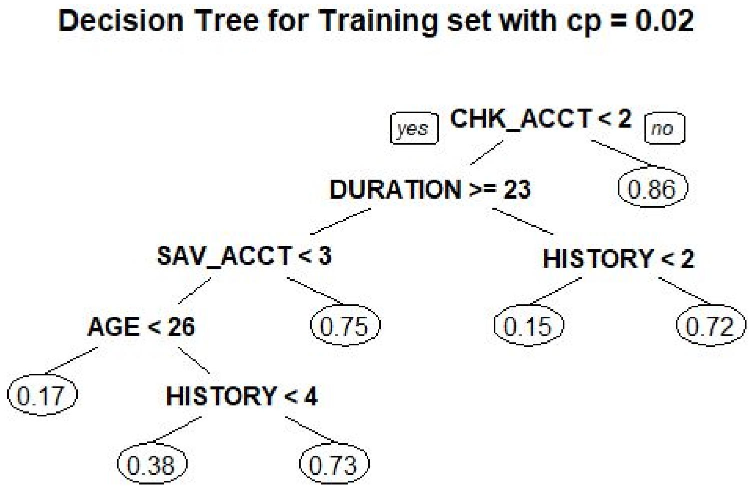
Confusion Matrix :

prediction	Good Cases	Bad Cases
Good Cases	75	60
Bad Cases	75	290

Accuracy Levels :

Accuracy	Precision	Recall	F1 Score
0.73	0.675	0.6643	0 - 0.5263 1 - 0.8112

(e) Model 5 : 50 - 50 "Information Split" with cp - 0.02 (minsplit - default)



Confusion Matrix :

prediction	Good Cases	Bad Cases
Good Cases	61	36
Bad Cases	89	314

Accuracy Levels :

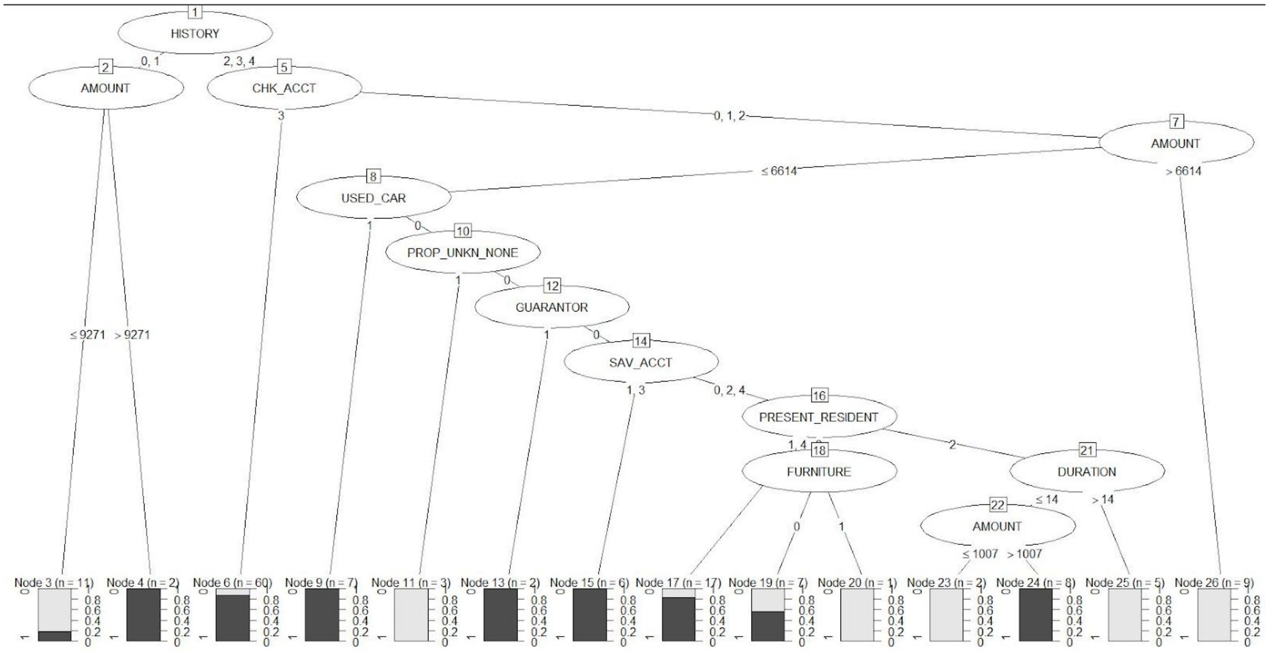
Accuracy	Precision	Recall	F1 Score
0.74	0.70	0.6519	0 - 0.4939 1 - 0.8342

Out of the 5 models implemented, the highest accuracy (77%) was observed in Model 2. The parameters of Model 2 were: Minsplit = 15 and cp = 0.018. The precision and recall of the model was 70% and 65% respectively. We have used cost complexity pruning by selecting the value of cp from cp table, which is calculated as 0.018. We chose this value as the error stops reducing after this value, since the data is not that significant.

Part 2 : C5.0 Models

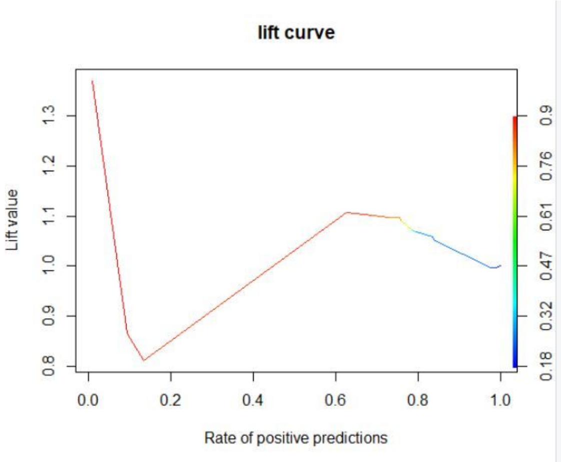
Models with parameters	Tree Size		Error		Accuracy	
	Train Data	Test Data	Train Data	Test Data	Train Data	Test Data
50 - 50 Information and Gini Split	44	35	13.2%	16.4%	0.7072	88.6%
70 - 30 Information Split and Gini Split	54	10	14.9%	15.7%	85.1%	84.3%
80 - 20 Information and Gini Split	46	14	12.5%	14.6%	87.5%	85.4%

C5.0 Plot for 80 - 20 Information Split

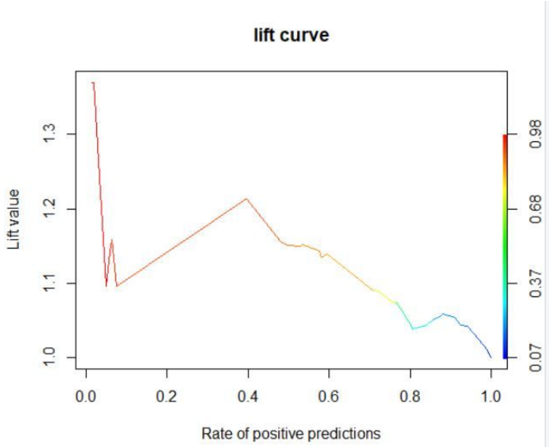


As summarised in the table, we developed 6 C5.0 Models using different training and test data splits and found out that accuracy improves the most in the 80 - 20 splits.

Lift curves :

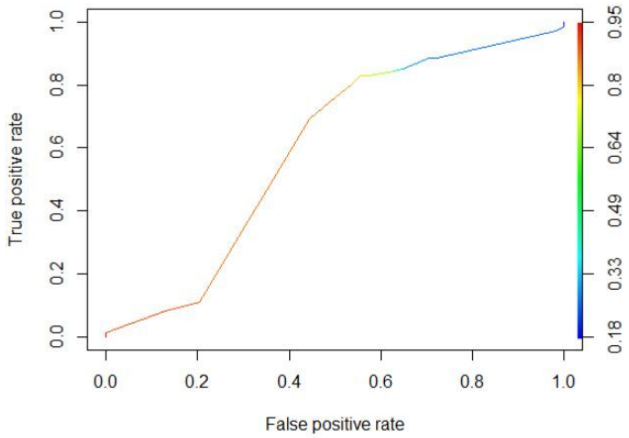


Lift Curve for rPart Model



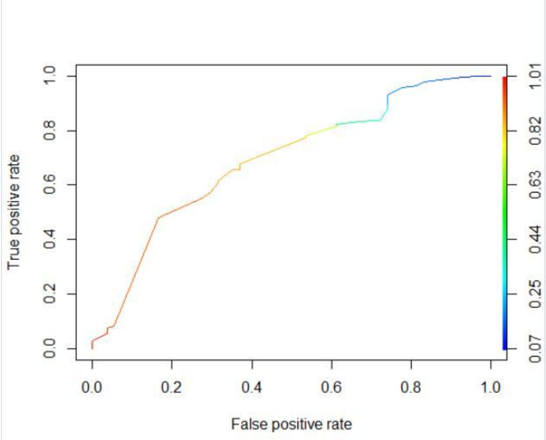
Lift Curve for C5.0

ROC curves :



ROC Curve for rPart Model

Area under the curve is maximum for a cutoff threshold of 0.7



ROC Curve for C5.0

The ROC curve covers much greater area at the threshold ~0.52

(c) Accuracy Levels in Training and Testing Datasets :

Seed	Split Used	Accuracy		
		Training	Testing	
663657	Gini	0.738	0.713	Decision trees are unstable because training a tree with a slightly different sub-sample causes the structure of the tree to change. As highlighted by the above table, the accuracies of the trees are affected by changing datasets, by choosing random seed values. The accuracy level varies across trees with different split criteria for the same seed value, as well.
69420	Gini	0.705	0.673	
663657	Information	0.73	0.719	
69420	Information	0.684	0.665	

(d)

From RPart models discussed in question 3(a), the model with minsplit = 15, cp = 0.018 (Model 2) gave the highest accuracy. In the summary of this model CHK_ACCT, DURATION, SAV_ACC, HISTORY, AGE were the parameters of highest importance.

In the C.50 Models, the model with an 80 - 20 Information Split Model saw the highest change in accuracy out of the 6 models tested. In the summary of this model HISTORY, AMOUNT and CHK_ACCT were the parameters of highest importance.

For rpart models, the importance for a predictor is simply the number of rules that involve the predictor. By default, C5.0 measures variable importance by determining the percentage of training set samples that fall into all the terminal nodes after the split.

e)

The best model is given by splitting the dataset 80-20 into Training and Test and applying the C5.0 algorithm. Also, by comparing the model performances, we observe that the C5.0 algorithm works better than the ‘rpart’ algorithm, across all splits in the datasets.

The 3 most important variables given by this tree were: History, Account and CHK_ACCT. These variables were observed in the upper part of the decision tree. This indicates that the tree is initially split on these variables, hence the high values of variable importance.

Question 4

Consider the net profit (on average) of credit decisions as: Accept applicant decision for an Actual "Good" case: 100DM, and Accept applicant decision for an Actual "Bad" case: -500DM

➢ Use the misclassification costs to assess performance of a chosen model. Compare model performance. Examine how different cutoff values for classification threshold make a difference. Use the ROC curve to choose a classification threshold which you think will be better than the default 0.5. What is the best performance you find?

Based on what the question indicates, every prediction of a customer's credit score as "Bad", causes us to loose 100DM whereas if we predict a customer a 'Bad' creditor as 'Good', we loose 500DM.

Confusion matrix for the best model at threshold value of 0.65:

	Predicted	
Actual	0	1
0	43	70
1	11	76

Net profit for validation data = 43*100 + 11*(-500) = -1200 DM

Opportunity cost = 70*100 + 11*500 = 12500 DM

Split	Threshold Value	Misclassification Cost Testing data	Accuracy of Training data	Accuracy of Testing data
0.8	0.5 0.65 0.8	54500 49200 4100	78.8% 78.7% 76.25%	81% 80% 76.5%

With a considerable change in threshold, the accuracy seems to decrease and so does the misclassification cost for the testing data. The ROC curve optimal cut-off with cost matrix is 0.76 and the area under the curve is 0.7. Thus, 0.65 is the best threshold earlier and now too.

➢ Calculate and apply the 'theoretical' threshold and assess performance – what do you notice, and how does this relate to the answer from (a) above.

Theoretical threshold = 500 / (500+100) ~ 0.83

When we apply this threshold to our model, the accuracy obtained is 76%. Though this is higher than the obtained threshold, it in fact reduces the accuracy of our model.

- ➢ Use misclassification costs in building the tree models (rpart and C5.0) – are the trees here different than ones obtained earlier? Compare performance of these two new models with those obtained earlier (in part 3a, b above)

After applying misclassification costs to the models, we can say that the model improves at every threshold by a slight amount.

The important variables before applying cost matrix: CHK_ACCT, DURATION, SAV_ACC, HISTORY, AGE

After cost matrix: CHK_ACCT, DURATION, SAV_ACC, HISTORY, USED_CAR

The C5.0 model with 0.8 split:

We used the thresholds 0.5, 0.65 and 0.8 to test the performance of the model and among these, 0.65 is still the best threshold with an improved accuracy of 85% with misclassification costs and 78% without.

Question 5

The most optimum tree has a maximum depth of 6 and 11 leaf nodes based on the 80-20 dataset.

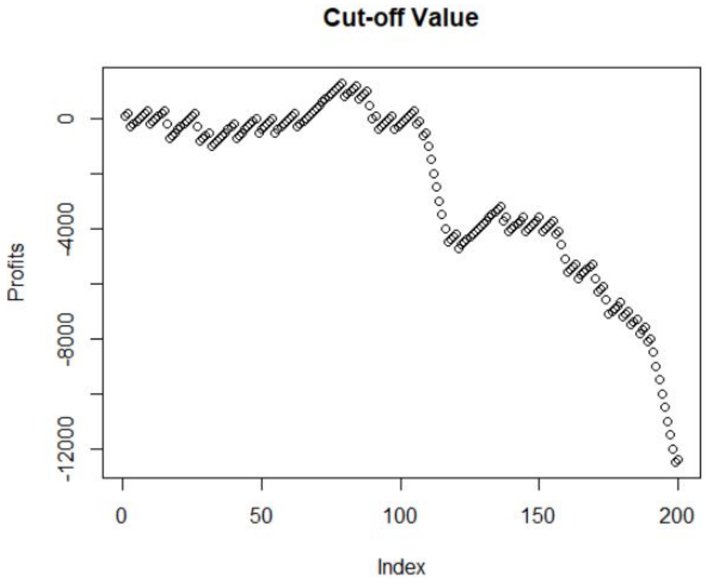
Top 5 important variables for classifying "Good" and "Bad" credit are as follow: CHK_ACCT, SAV_ACCT, DURATION, HISTORY, AMOUNT

Two relatively pure nodes are listed below: One is formed at the split criteria EMPLOYMENT=0,4, which contains 9 good credits (probability=1.00) cases and 0 bad credit cases(probability=0.00). Another is formed at the split criteria SAV_ACCT=2,3,4, which includes 8 good credit cases (probability=1.00) and 0 bad credit cases (probability=0.00)

Question 6

Our classification is to predict if the customer will default the credit or not.

We have introduced a new dataframe (prLifts) where we are storing scoreTst, RESPONSE, profits and cumulative profits. These values are sorted in descending order of scoreTst. For each correct prediction we gain 100DM and for each misclassification we lose 500DM. After sorting our data based on the predicted probability, we see that after a cut-off of 0.77, we get into losses due to increasing number of incorrect predictions.



Based on this , we see maximum revenue gain at 79th observation with a value of 0.77 where net profit is 13,000 DM which is the cumulative net benefit expected. After this value, the net profit starts dipping due to increasing number of misclassifications.

Therefore, it's reliable to give them credit if credit score is above this value.

Question 7

- ➢ Develop a random forest model (using a 70:30 training: test data split). What random forest parameters do you try out, and what performance do you obtain?

Random Forest:

Since the dependent variable - 'RESPONSE' is a factor, the random forest is assumed to be of classification type.

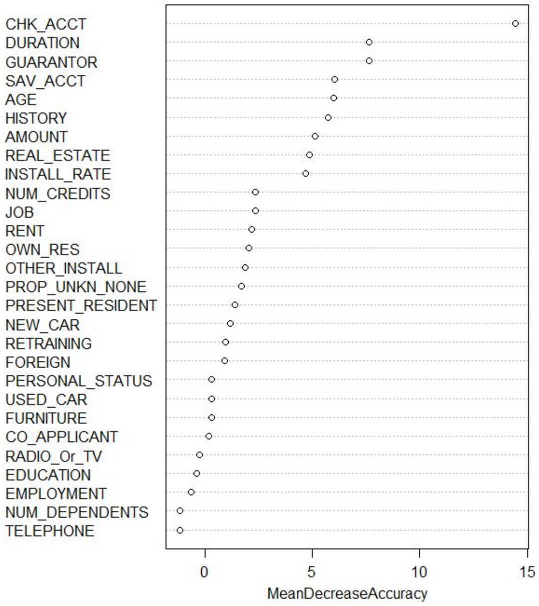
We split the data into training and testing parts (70-30%) and create a random forest model with

ntree = 200: Achieved OOB error rate is 24.71% i.e. the misclassification rate.

Performance evaluation for this random forest model: The number of random variables used in a tree is given by mtry. Now, we need to find the best value of mtry with the least OOB error rate using the tuneRF() function. This gives mtry = 5 with OOB = 0.240

The best Random Forest model with mtry = 5 gives the below variable importance plot:

Higher the mean decrease accuracy / mean decrease Gini value, higher the importance of a variable in the model. Here, CHK_ACCT is the most important variable.



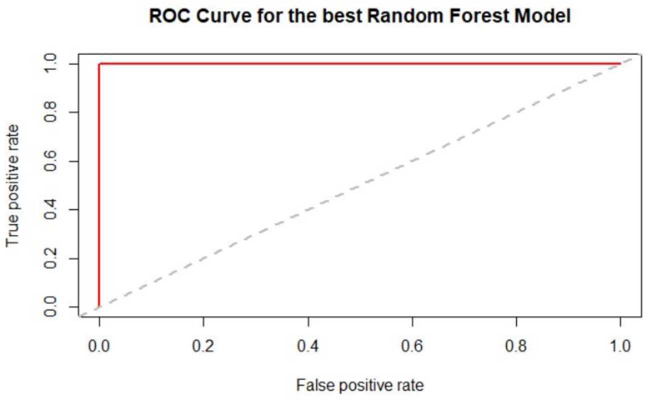
- ➢ Compare the performance of the best random forest and best decision tree models – show a ROC plot to help compare models, and also the maximum net benefit.

Performance evaluation of Random Forest model and the best decision tree model :

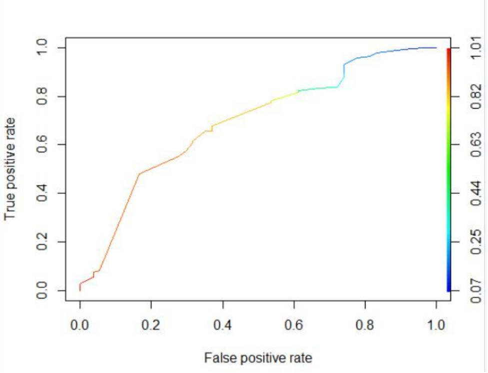
MTry Values	OOB Error
4	0.249
5	0.240
7	0.242

The accuracy for Random forest model is 1 while that for our best decision tree model is 0.77.

Performance evaluation for the best random forest model:



ROC for decision tree


True positive rate
False positive rate