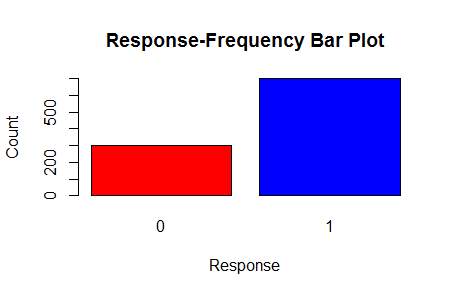
1. **Explore the data: What is the proportion of “Good” to “Bad” cases? Are there any missing values – how do you handle these? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-values attributes, frequencies of different category values. Examine variable plots. Do you notice ‘bad’ credit cases to be more prevalent in certain value-ranges of specific variables, and is this what one might expect (or is it more of a surprise)? What are certain interesting variables and relationships (why ‘interesting’)? From the data exploration, which variables do you think will be most relevant for the outcome of interest, and why?**

As observed in Response-Frequency bar plot, the proportion of the Good(blue) to Bad(red) cases in the data provided is 700:300 = 7:3.

****

Below is a table with the statistical information of the predictor (independent) variables:

Statistical description of predictors for real values attributes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Variable Type** | **Mean** | **Standard Deviation** | **Min** | **Max** |
| AGE | Numerical | 35.461 | 11.32187 | 19 | 75 |
| AMOUNT | Numerical | 3271.156 | 2822.625 | 250 | 18424 |
| DURATION | Numerical | 20.903 | 12.05881 | 4 | 72 |
| INSTALL\_RATE | Numerical | 2.973 | 1.118715 | 1 | 4 |
| NUM\_CREDITS | Numerical | 1.407 | 0.5776545 | 1 | 4 |
| NUM\_DEPENDENTS | Numerical | 1.155 | 0.3620858 | 1 | 2 |

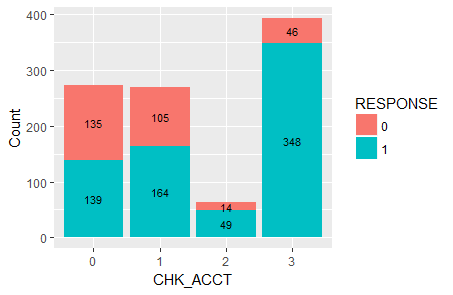
Frequency table for different categorical predictors

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Variable Type** | **Frequency** |
| CHK\_ACCT | Categorical | 1 (274), 1 (269), 2 (63), 3 (394) |
| CO-APPLICANT | Binary | 0 (959), 1 (41) |
| EDUCATION | Binary | 0 (950), 1 (50) |
| EMPLOYMENT | Categorical | 0 (62), 1 (172), 2 (339), 3 (174) , 4(253) |
| FOREIGN | Binary | 0 (963), 1(37) |
| FURNITURE | Binary | 0 (819), 1 (181) |
| GUARANTOR | Binary | 0 (948), 1 (52) |
| HISTORY | Categorical | 0 (40), 1 (49), 2 (530), 3 (88), 4 (293) |
| JOB | Categorical | 0 (22) , 1 (200), 2 (630), 3 (148) |
| MALE\_DIV | Binary | 0 (950), 1 (50) |
| MALE\_MAR\_WID | Binary | 0 (908), 1 (92) |
| MALE\_SINGLE | Binary | 0 (452), 1 (548) |
| NEW\_CAR | Binary | 0 (766), 1 (234) |
| OTHER\_INSTALL | Binary | 0 (814), 1 (816) |
| OWN\_RES | Binary | 0 (287), 1 (713) |
| PRESENT\_RESIDENT | Categorical | 1 (130), 2 (308), 3 (149), 4 (413) |
| PROP\_UNKN\_NONE | Binary | 0 (846) , 1 (154) |
| RADIO/TV | Binary | 0 (720), 1 (280) |
| REAL\_ESTATE | Binary | 0 (718), 1 (282) |
| RENT | Binary | 0 (821), 1 (179) |
| SAV\_ACCT | Categorical | 0(603), 1(103), 2(63), 3(48), 4(183) |

By observing the plots for all the independent variables, we decided to use CHK\_ACCT, DURATION, AGE, SAV\_ACCT, AMOUNT, OTHER\_INSTALL and JOB as good predictor (independent) variables.

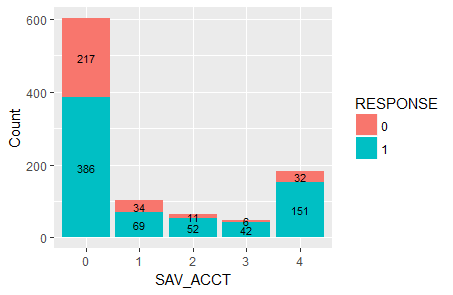
1. **CHK\_ACCT**

As we can see in the plot below, out of 700 good cases, 348 cases are the ones that do not have a checking account (categorical value 3) have their loan approved. Thus, this variable can be a good predictor.



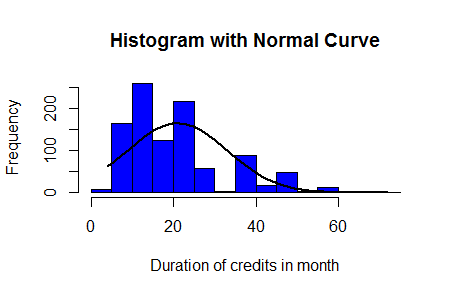
1. **SAV\_ACCT**

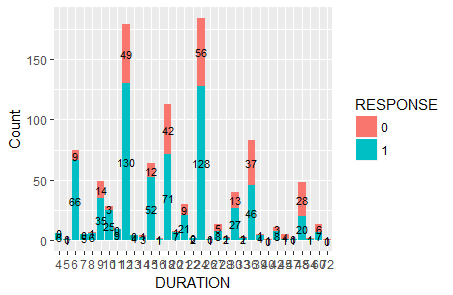
Below is the plot showing the frequency of the response variable with SAV\_ACCT variable. For the category 0 (average balance less than 100 DM), there is good proportion of good as well as bad cases, which can help us in getting a good model.



1. **DURATION**

Below is the distribution of DURATION variable which can be an interesting predictor as it has less P-value.





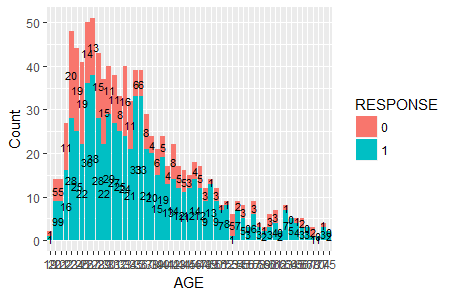
1. **JOB**

JOB can also be an important predictor variable which gives the information about the nature of the job of an individual. Below is the bar plot showing the frequency distribution of good cases and bad cases across different categories.



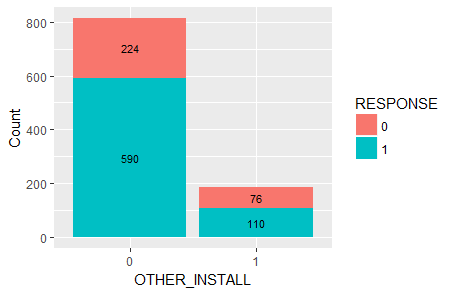
1. **AGE**

Below is the distribution of AGE variable which shows that this variable can be insightful in determining the credit risk for a new applicant.



1. **OTHER\_INSTALL**

It is also a good predictor which shows that the individuals who do not have other installment plans have good chances of getting the loan approved.



1. **We will first focus on a descriptive model – i.e. assume we are not interested in prediction.**

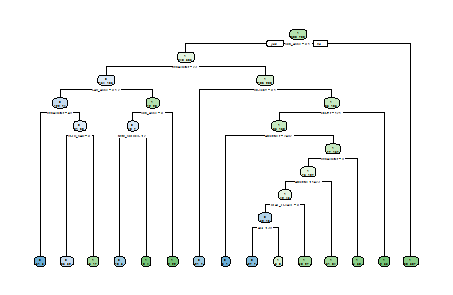
**(a) Develop a decision tree on the full data (using the *rpart* package).**

**What decision tree node parameters do you use to get a good model. Explain the parameters you use**

**(b)Which variables are important to differentiate “good” from “bad” cases – and how do you determine these? Does this match your expectations (from your response in Question 1)?**

**(c)What levels of accuracy/error are obtained? What is the accuracy on the “good” and “bad” cases? Obtain and interpret the lift chart. Do you think this is a reliable (robust?) description, and why.**

Our initial decision tree was a very simple model based on the Gini Index criterion and a minimum split of 10. Below is the decision tree used for the same.



Text for the model is as below:

print(rpModel1)

n= 1000

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 1000 300 1 (0.30000000 0.70000000)

2) CHK\_ACCT=0,1 543 240 1 (0.44198895 0.55801105)

4) DURATION>=22.5 237 103 0 (0.56540084 0.43459916)

8) SAV\_ACCT=0,1,2 196 74 0 (0.62244898 0.37755102)

16) DURATION>=47.5 36 5 0 (0.86111111 0.13888889) \*

17) DURATION< 47.5 160 69 0 (0.56875000 0.43125000)

34) USED\_CAR=0 137 52 0 (0.62043796 0.37956204) \*

35) USED\_CAR=1 23 6 1 (0.26086957 0.73913043) \*

9) SAV\_ACCT=3,4 41 12 1 (0.29268293 0.70731707)

18) CHK\_ACCT=0 17 7 0 (0.58823529 0.41176471)

36) NUM\_CREDITS< 1.5 13 3 0 (0.76923077 0.23076923) \*

37) NUM\_CREDITS>=1.5 4 0 1 (0.00000000 1.00000000) \*

19) CHK\_ACCT=1 24 2 1 (0.08333333 0.91666667) \*

5) DURATION< 22.5 306 106 1 (0.34640523 0.65359477)

10) HISTORY=0,1 28 7 0 (0.75000000 0.25000000) \*

11) HISTORY=2,3,4 278 85 1 (0.30575540 0.69424460)

22) OBS#>=120.5 241 83 1 (0.34439834 0.65560166)

44) AMOUNT>=7491.5 7 1 0 (0.85714286 0.14285714) \*

45) AMOUNT< 7491.5 234 77 1 (0.32905983 0.67094017)

90) DURATION>=8.5 200 73 1 (0.36500000 0.63500000)

180) AMOUNT< 1423 85 42 1 (0.49411765 0.50588235)

360) REAL\_ESTATE=0 48 16 0 (0.66666667 0.33333333)

720) AGE< 37.5 34 7 0 (0.79411765 0.20588235) \*

721) AGE>=37.5 14 5 1 (0.35714286 0.64285714) \*

361) REAL\_ESTATE=1 37 10 1 (0.27027027 0.72972973) \*

181) AMOUNT>=1423 115 31 1 (0.26956522 0.73043478) \*

91) DURATION< 8.5 34 4 1 (0.11764706 0.88235294) \*

23) OBS#< 120.5 37 2 1 (0.05405405 0.94594595) \*

3) CHK\_ACCT=2,3 457 60 1 (0.13129103 0.86870897) \*

true

pred 0 1

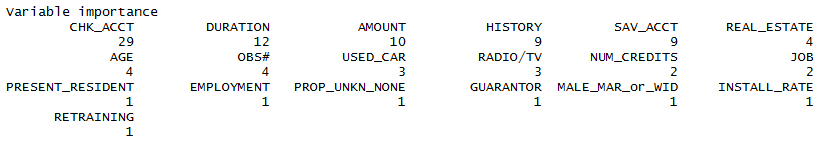
0 180 75

1 120 625

But, with this model we obtained a decision tree which is very big and it is difficult to read the decision tree efficiently. This makes the tree unreliable. We know that for a model to be reliable, it should be possible to validate it by performing statistical test on it. Also, it should be resistant to error i.e. it should perform well if its assumptions are violated by the true model from which the data were generated. In order to get a good model, below are the node parameters that result in a more optimized tree which has a better performance. To make our decision tree more robust and readable, we split the data into training and testing data.

**Variable Importance:**

The variables that are important to differentiate “good” cases from “bad” cases are CHK\_ACCT, AMOUNT, HISTORY, DURATION, INSTALL\_RATE based on the variable importance that we got from the model.

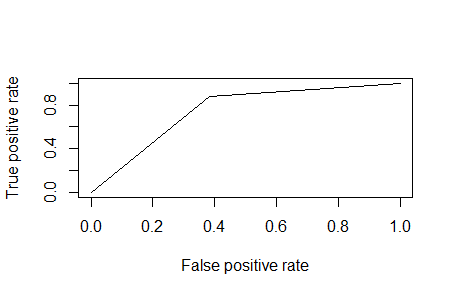


CHK\_ACCT is the root node of the decision tree obtained above. Hence it is the most important predictor variable. It splits the data in four categories as 0,1,2,3 depending on the status of checking accounts of the individuals.

DURATION is a numerical variable with the lowest P-value, hence is a good predictor.

This model provides an **accuracy of 80.5%** with a **sensitivity of 89.2%** and **precision 83.89%** for truly predicted cases.

**Lift curve for the model is as shown below for the model made for full data:**



3. **(a) Develop decision trees using the rpart package. What model performance do you obtain? Consider performance based on overall accuracy/error and on the ‘good’ and ‘bad’ credit cases – explain which performance measures, like recall, precision, sensitivity, etc. you use and why. Also consider lift, ROC and AUC.**

The some of the few important combination of parameters for the decision tree model are as mentioned below:

1. Minimal size for split
2. Minimal Leaf size
3. Minimal Gain
4. Maximal Depth of tree
5. Decision Criterion
6. Complexity parameter

Based on these parameters we have executed we have created 3 models and calculated the accuracy for the same.

Below table contains the parameters used and the accuracy for the 3 models which we have built by splitting the data into 50% as training data and 50% as testing data

TRAINING DATA:

|  |  |
| --- | --- |
| Parameters Used | OUTPUT: |
| **RPModel2** (Method Used: Class  Complexity parameter:0.01  Decision Criterion: Entropy) | Accuracy:0.848  Precision:0.75(0),0.88(1)  Recall:0.70(0),0.90(1) |
| **RPModel2a** (Minimal Size for the split:10  Method Used: Class  Complexity parameter:0.01  Decision Criterion: Information) | Accuracy:0.878  Precision:0.80(0),0.904(1)  Recall:0.75(0),0.926(1) |
| **RPModel2b** (Minimal Size for the split:86  Method Used: Class  Max Depth:30  Complexity parameter:0.01  Decision Criterion: Information) | Accuracy:0.77  Precision:0.602(0),0.835(1)  Recall:0.59(0),0.842(1) |

TEST DATA:

|  |  |
| --- | --- |
| Model | Output |
| **RPModel3** Method Used: Class  Complexity parameter:0.01  Decision Criterion: Entropy | Accuracy:0.828  Precision:0.786(0),0.84(1)  Recall:0.615(0),0.924(1) |
| **RPModel3a**Minimal Size for the split:10  Method Used: Class  Complexity parameter:0.01  Decision Criterion: Information | Accuracy:0.824  Precision:0.809(0),0.828(1)  Recall:0.57(0),0.938(1) |
| **RPModel3b**Minimal Size for the split:86  Method Used: Class  Max Depth:30  Complexity parameter:0.01  Decision Criterion:  Information | Accuracy:0.744  Precision:0.593(0),0.808(1)  Recall:0.57(0),0.822(1) |

Based on the output data of the models executed on the training data and test data using the performance measures such as accuracy, precision, recall, ROC and AUC, we have come to conclusion that **RPModel3a** is the best model.

The decision tree parameters that we found useful to pick the best model are:

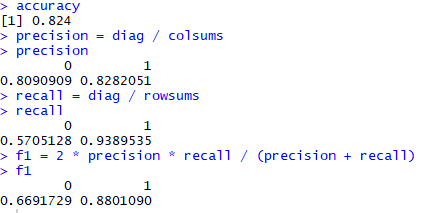
Minimal Size for the split:10

Method Used: Class

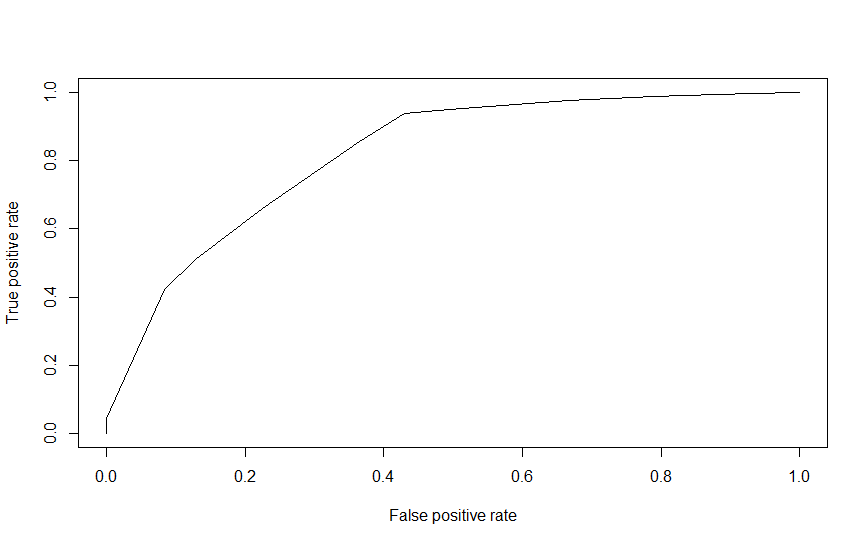
Complexity parameter:0.01

Decision Criterion: Information

**Output of accuracy, precision and recall for the best model rpmodel3a**



**ROC plot for the best model rpmodel3a**



**AUC output for the best model rpmodel3a**



**3b)**

CMOD Training Data

|  |  |
| --- | --- |
| **CMOD1**  Method Used: Class | Accuracy:0.916  Precision:0.88(0),0.92(1)  Recall:0.812(0),0.822(1) |
| **CMOD2**  Method Used: Class  No global pruning: FALSE  Subset: TRUE  Winnow: TRUE  Rules: TRUE | Accuracy:0.886  Precision:0.817(0),0.890(1)  Recall:0.70(0),0.957(1) |
| **CMOD3**  Method Used: Class  Early stopping: TRUE  CF: 0.00001  Winnow: FALSE  Rules: TRUE  Mini-case: 10 | Accuracy:0.886  Precision:0.871(0),0.89(1)  Recall:0.708(0),0.957(1) |

CMOD Test Data

|  |  |
| --- | --- |
| **CMOD1**  Method Used: Class | Accuracy:0.89  Precision:0.946(0),0.873(1)  Recall:0.685(0),0.982(1) |
| **CMOD2**  Method Used: Class  No global pruning: FALSE  Subset: TRUE  Winnow: TRUE  Rules: TRUE | Accuracy:0.862  Precision:0.948(0),0.841(1)  Recall:0.589(0),0.985(1) |
| **CMOD3**  Method Used: Class  Early stopping: TRUE  CF: 0.00001  Winnow: FALSE  Rules: TRUE  Mini-case: 10 | Accuracy:0.862  Precision:0.948(0),0.841(1)  Recall:0.589(0),0.985(1) |

Based on the output data of the models executed on the training data and test data using the performance measures such as accuracy, precision, recall, ROC and AUC, we have come to conclusion that **CMOD2** is the best model.

The decision tree parameters that we found useful to pick the best model are:

Method Used: Class

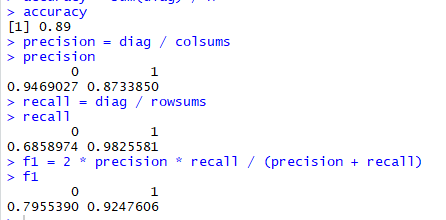
No global pruning: FALSE

Subset: TRUE

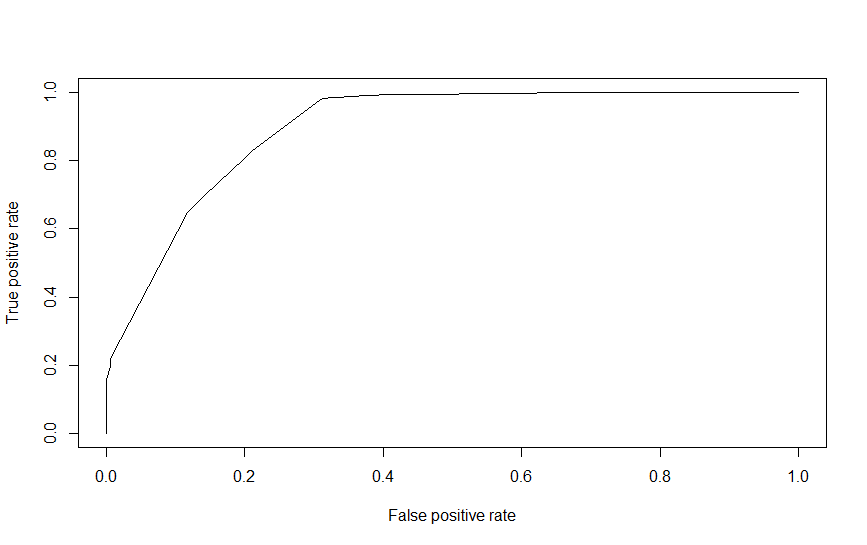
Winnow: TRUE

The models obtained from C5.0 decision tree differs from the RPart decision tree in terms of accuracy, precision, recall, ROC, and AUC as shown in the tables and the plots.

**Output of precison,accuracy and recall for best C5.0 Model**



**ROC output:**



**AUC Output:**

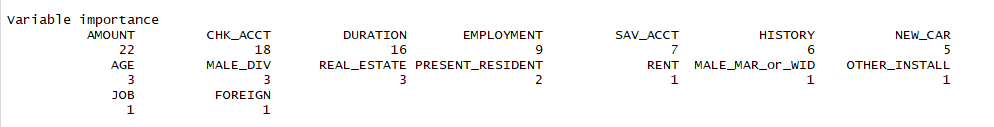


**3c)**

By interpreting the models and performance of the samples of different training and test data we found out that the performance of those samples varied. So we could say that the decision tree is unstable. As we change the training and test dataset, the outputs like accuracy, precision, recall changes significantly. Also, the variability of the models is visible while obtaining the outputs for the different models for different parameters.

**3d)**

Below is the screenshot showing the variable importance for the best model.



There is a difference between variable importance for the **RPModel1** and the best model **RPModel3a**. Based on the variable importance observed for the best model **RPModel3a**, CHK\_ACCT, SAV\_ACCT, AMOUNT, DURATION, HISTORY, MALE\_DIV,NEW\_CAR, EMPLOYMENT variables are important.

**4.a)Use the misclassification costs to assess performance of a chosen model from Q 2 above. Compare model performance. Examine how different cutoff values for classification threshold make a difference. Use 3 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?**

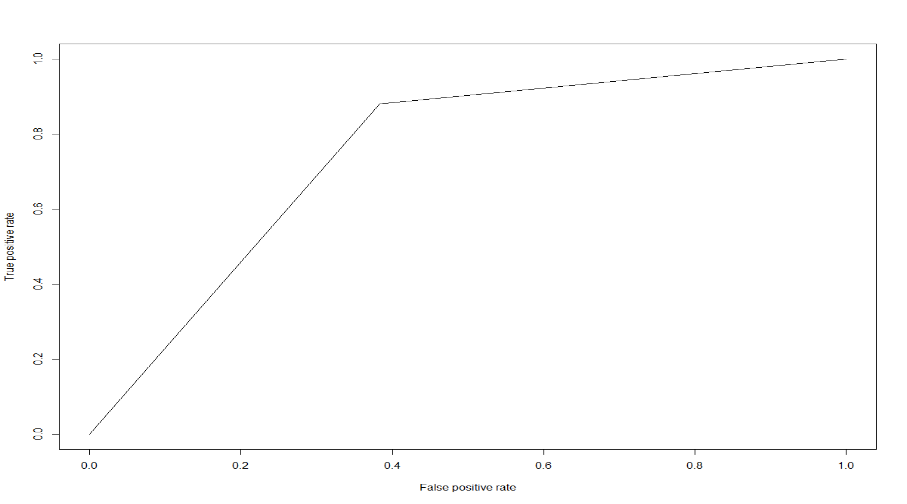
From the Q2 we chose Rpmodel1 as our best model and we executed the misclassification costs.

We could see that when the threshold value is varied  beyond a certain range from the default value which is 0.5 the accuracy reduced. Below table indicates the accuracy variation with respect to the threshold value

**Best Model from Q2: Rpmodel1**

|  |  |
| --- | --- |
| **Threshold value for Rpmodel1** | **Accuracy** |
| 0.6 | 0.801 |
| 0.75 | 0.784 |
| 0.9 | 0.33 |
| 0.3 | 0.75 |
| 0.1 | 0.62 |

Based on the table above we found our optimum threshold value to be 0.6 and the plot for the same is given below



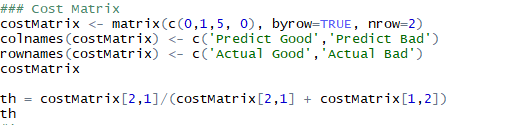
We also found the predicted cutoff values as below

**https://lh3.googleusercontent.com/3jBHfsLVdn04YNoOMBh-aSDwhQ2UiuXHThkjSGusVVlzclDCALkih_MQANSxTbOQ-Z6B3wm1fDeeSWM1FghHo6tGcOOH4k9a35zwriXD6Qi4GNpEAsvRgjkC3B0VDjabXBtB8TY2**

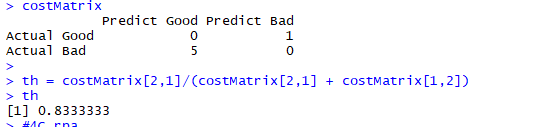
**4. B**

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

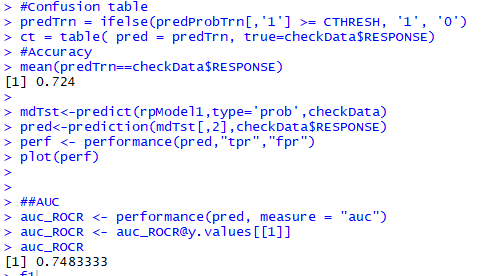
We calculated the theoretical threshold from the below formula

****

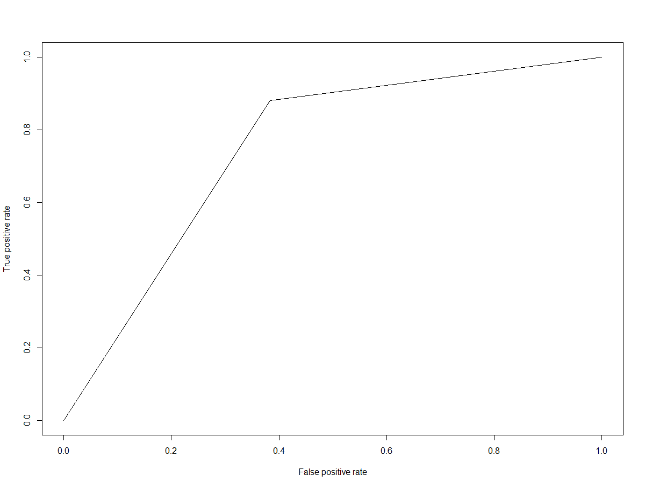
From this we found the theoretical cut off as **0.833**

****

When we applied this threshold value we found the below output



**Accuracy:0.724**

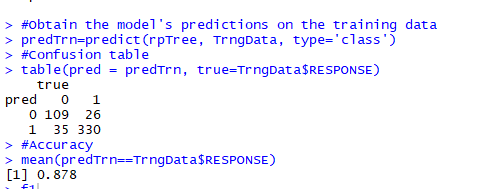
****

**4.C**

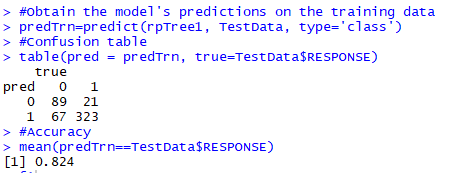
**Use misclassification costs in building the tree models (rpart and C5.0) – are the trees here different than ones obtained earlier? Compare performance of these two new models with those obtained earlier.**

We added the miscalculation costs in our best tree models in Rpart and C5.0 and the output is explained below

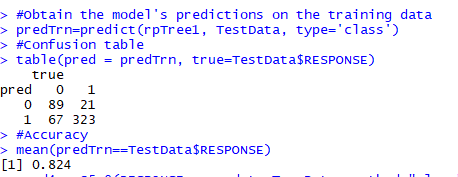
**Rpart output for training data:**



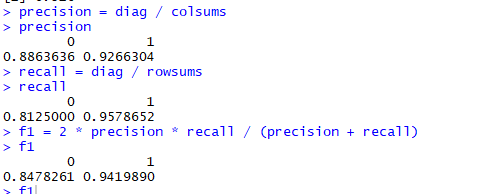
**Rpart output for Testing data:**

****

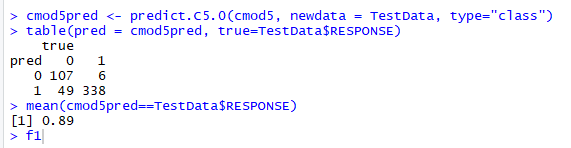
**C5.0 Output for Training data:**

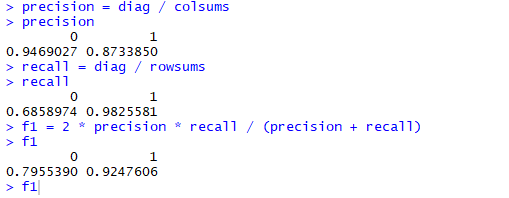


**Precision and recall value for the same**

****

**C5.0 Output for Testing data:**

****

****

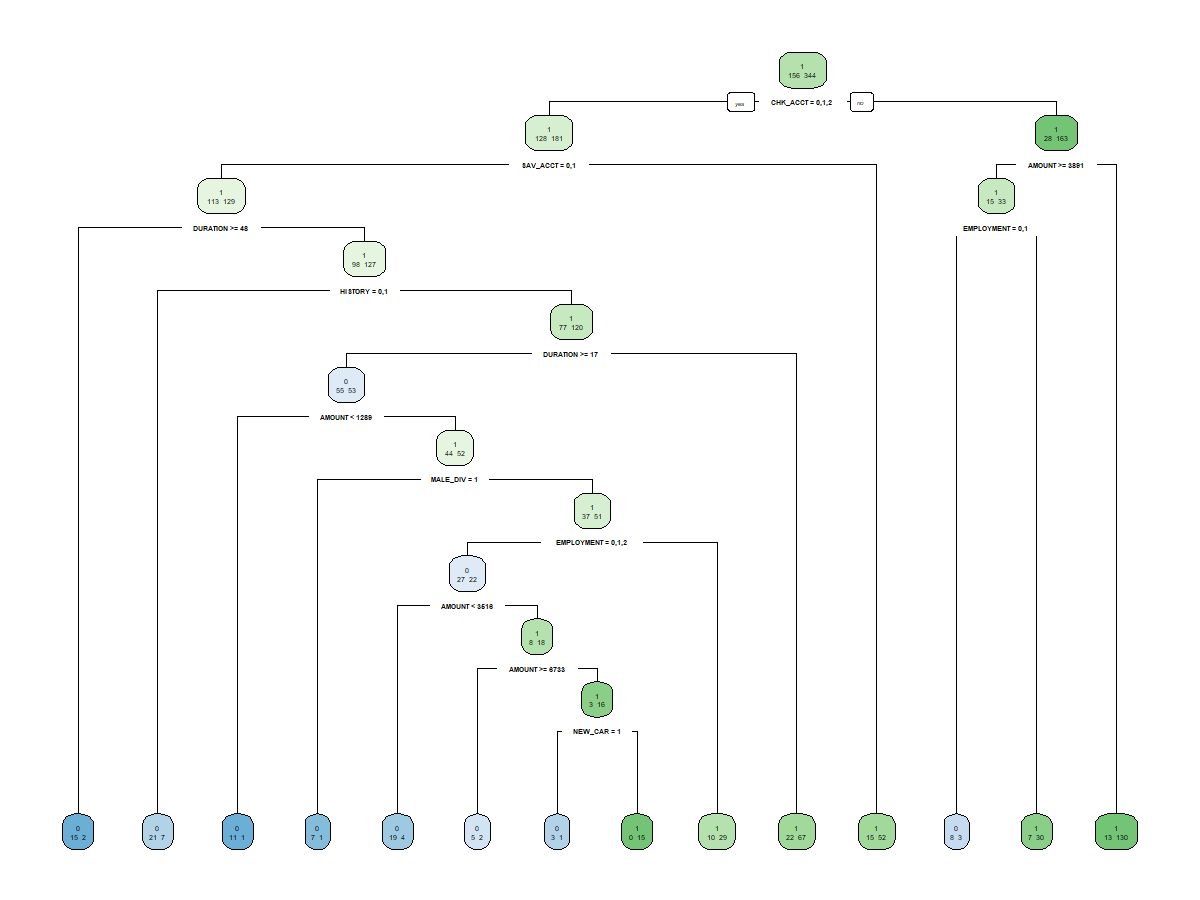
On comparing these two models from above the accuracy for the rpart and C5.0 has increased than the previous one.

**5.** **Let’s examine your ‘best’ decision tree model obtained. What is the tree depth? And how many nodes does it have? What are the important variables for classifying “Good’ vs ‘Bad’ credit?**

**Identify two relatively pure leaf nodes. What are the ‘probabilities for ‘Good’ and ‘Bad’ in these nodes?**

**Best Model**

rpTree1 = rpart(RESPONSE ~ .,minsplit=10, data=TestData, method="class",cp=0.01, parms = list( prior = c(.70,.30), loss = costMatrix, split = "information"))



Number of nodes = 27

Depth of the tree = 11

**Path of the depth of the tree:**

CHK-ACCT(0,1) ->SAV\_ACCT- >DURATION(>=48)-> HISTORY->DURATION(>=17) -> AMOUNT(1289) -> MALE\_DIV->EMPLOYMENT ->AMOUNT(<3516) -> AMOUNT(>=6739)-> NEW\_CAR

**Pure Leaf nodes**

1(Good credit)- NEW\_CAR - 0 15

Probability of getting a good case is 15/15=1 and a probability of gettig a bad case is 0/15=0

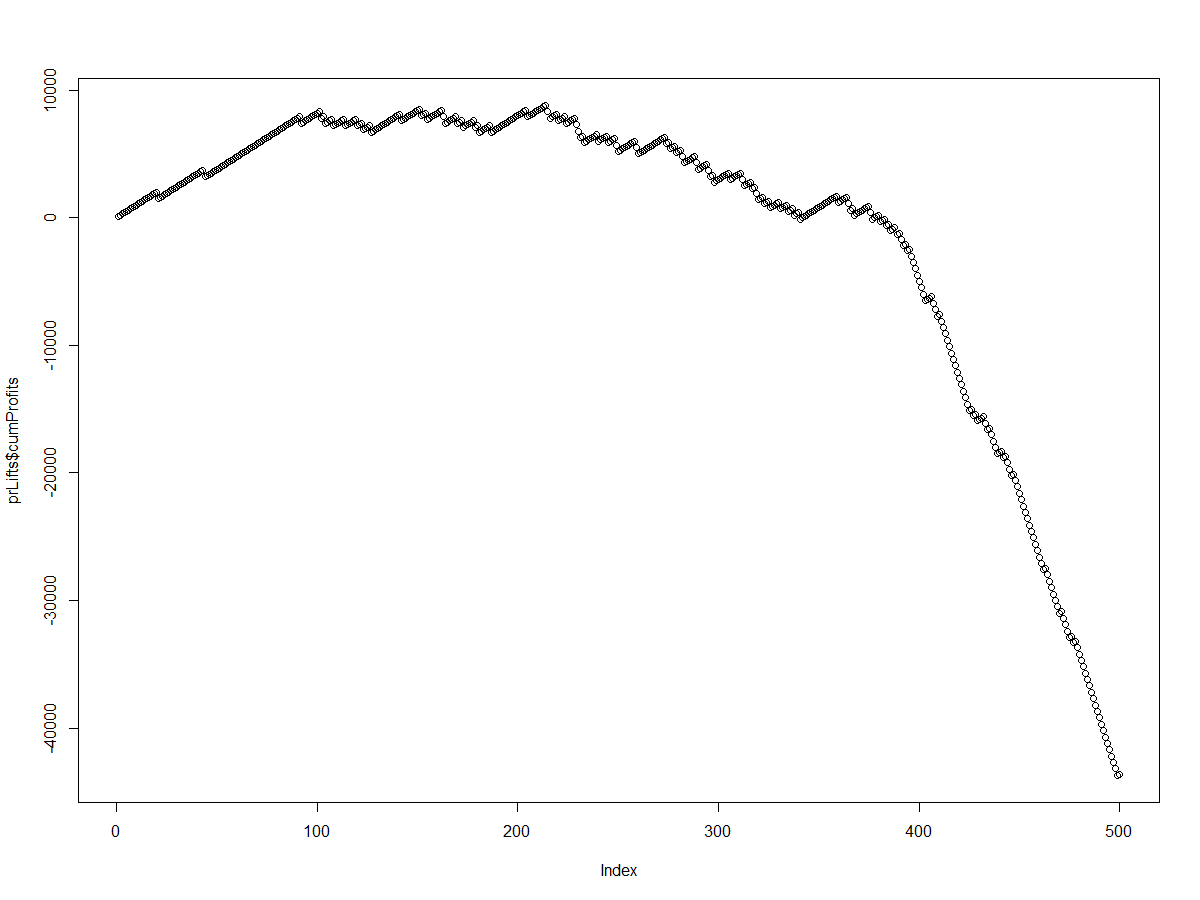
0(Bad credit) - AMOUNT(<1289) - 11 1

Probability of getting good case is 1/12=0.083 and probability of getting bad case is 11/12=0.91

**6.How far into the validation data would you go to get maximum net benefit? In using this model to score future credit applicants, what cutoff value for predicted probability would you recommend? Provide appropriate performance values to back up your recommendation**

We used probability method to determine how the model may be implemented. We predicted the data based on the probability on “ good” credit risk. We then sorted in descending order i.e,from high to low probabilities.

Please find the plot below



Based on the graph using the guidelines given in the question,assuming a 'profit' value for correctly predicting a 'good' case is 100, and a 'cost' for mistakes is -500,with this we calculated the profits and the cumulative profits.we obtained that the maximum net benefit is achieved at the 249th record of the validation data after which the cumulative cost starts decreasing.In order to score future credit applicants, we would recommend a cut-off value of 329th record of the validation data for predicted probability.