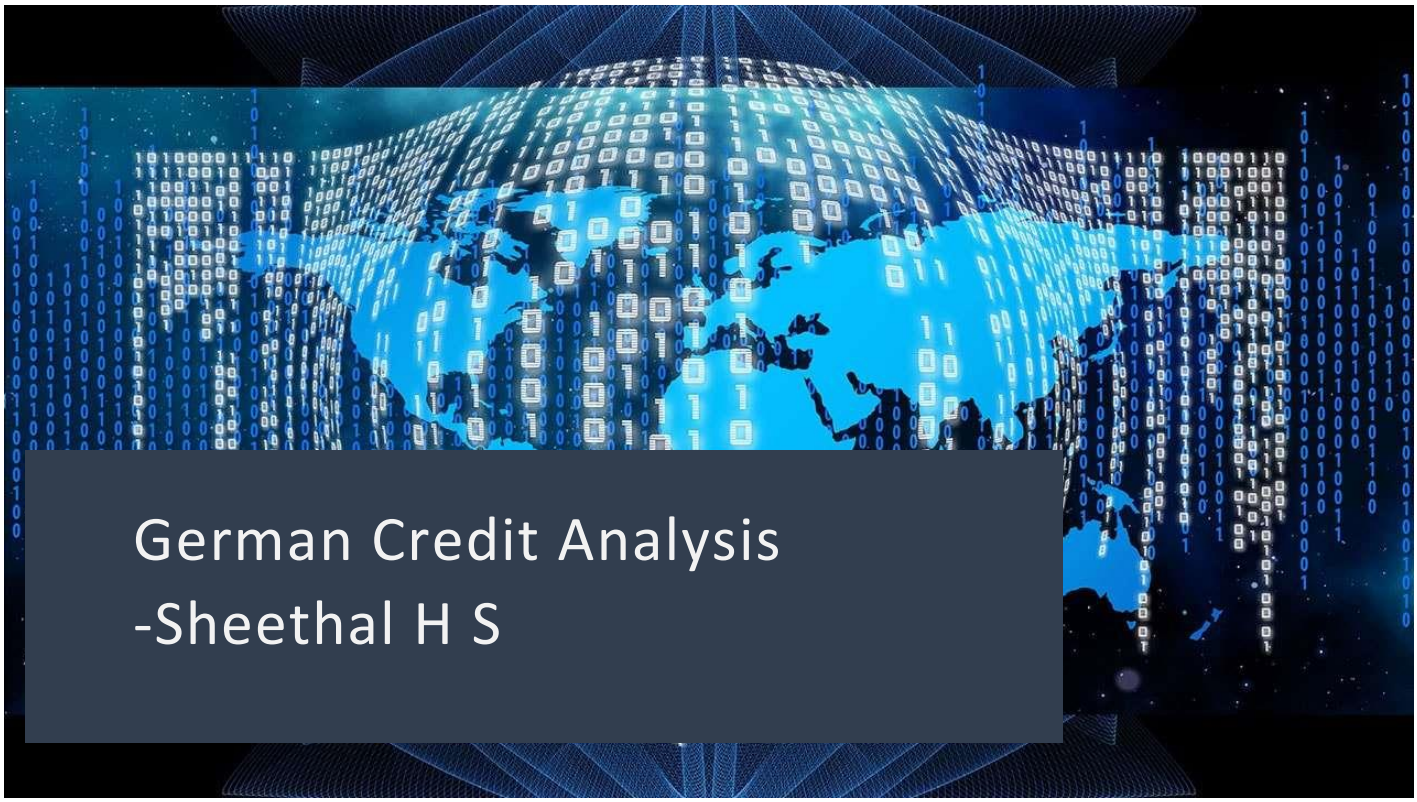


German Credit Analysis
-Sheethal H S

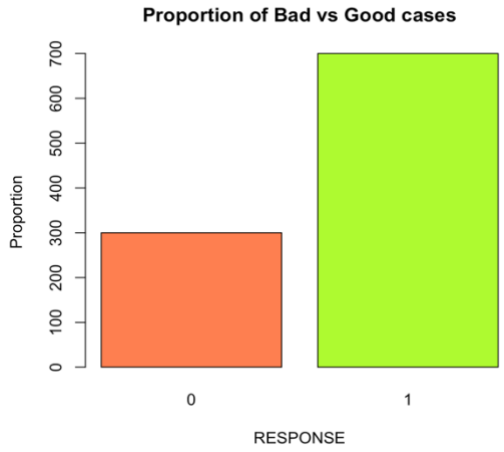
1. **Explore the data: What is the proportion of "Good" to "Bad" cases? Are there any missing values – how do you handle these? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-values attributes, frequencies of different category values. Examine 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 - explain why you think these are 'interesting'. From the data exploration, which variables do you think will be most relevant for the outcome of interest (and why)?**

DATA CLEANING

1. **Examining the data types and conversion to right types:** We first analyzed the data types of all the variables and then converted them to factor or numeric as we found them appropriate.
2. **Handling Missing Values:** While analyzing the descriptive statistics of the variables we realized that there were a lot of missing values/NAs.
   These are the variables which had missing values:
   New_Car, Used_Car, Furniture, Radio/TV, Education, Retraining
   The missing values have been **converted to 0** as the existing values were 1 and we considered it as a binary value.
   The other variables which have missing values are:
   - Personal _Status – For this we have defined number **"4" as a fourth variable category** in addition to the other category values. 4 would mean missing category.
   - Age – We have replaced the missing values with the **median value 33**.

3. **Proportion of good to bad cases: RESPONSE** variable gave us the proportion of good to bad cases.

Bad Cases (0)	Good Cases (1)
300	700
30%	70%



4. **Descriptions of the predictor variables**

Numeric Independent variables:

Variable	Mean	SD	Median	Min	Max	Skew
DURATION	20.9	12.06	18	4	72	1.09
AMOUNT	3271.16	2822.63	2319.5	250	18424	1.94
AGE	35.46	11.32	33	19	75	1.04

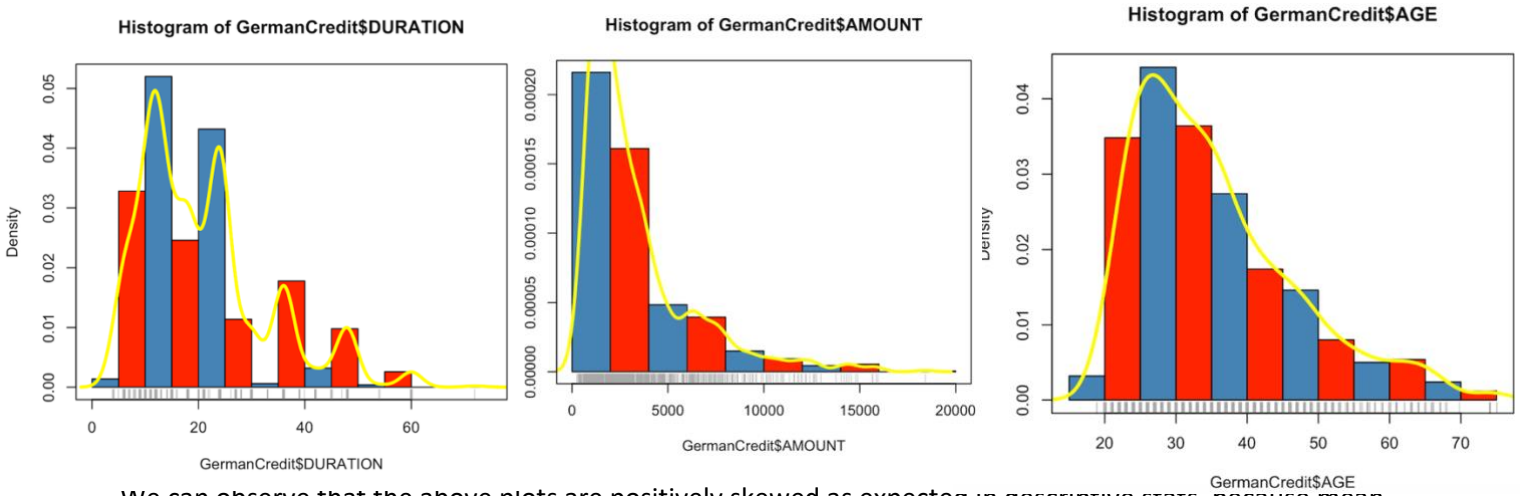
Please note that although we considered INSTALL_RATE, NUM_DEPENDENTS and NUM_CREDITS as numeric initially. As we plotted univariate graphs, we realized its appropriate if they are considered as factors.

Categorical Independent variables:

Was observed frequency using summary function. Please ignore the numeric variables in the snapshot.

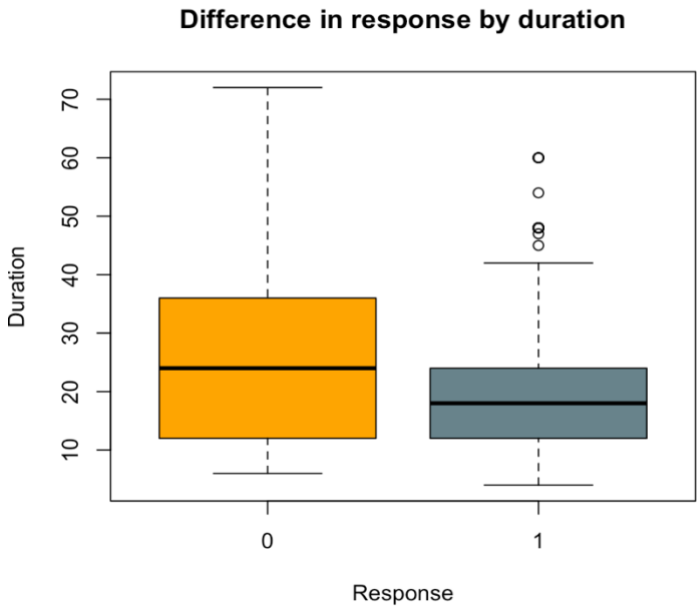
```
> summary(GermanCredit)
 CHK_ACCT    DURATION    HISTORY NEW_CAR USED_CAR FURNITURE RADIO/TV EDUCATION RETRAINING     AMOUNT      SAV_ACCT
 0:274    Min.   : 4.0   0: 40   0:766   0:897    0:819     0:720    0:950     0:903      Min.   :  250   0:603
 1:269    1st Qu.:12.0   1: 49   1:234   1:103    1:181     1:280    1: 50     1: 97      1st Qu.: 1366   1:103
 2: 63    Median :18.0   2:530                                                           Median : 2320   2: 63
 3:394    Mean   :20.9   3: 88                                                           Mean   : 3271   3: 48
          3rd Qu.:24.0   4:293                                                           3rd Qu.: 3972   4:183
          Max.   :72.0                                                                  Max.   :18424
 EMPLOYMENT INSTALL_RATE PERSONAL_STATUS CO-APPLICANT GUARANTOR PRESENT_RESIDENT REAL_ESTATE PROP_UNKN_NONE
 0: 62      1:136        1:548           0:959        0:948     1:130            0:718       0:846
 1:172      2:231        2: 92           1: 41        1: 52     2:308            1:282       1:154
 2:339      3:157        3: 50                                  3:149
 3:174      4:476        4:310                                  4:413
 4:253

      AGE        OTHER_INSTALL RENT    OWN_RES NUM_CREDITS JOB     NUM_DEPENDENTS TELEPHONE FOREIGN RESPONSE
 Min.   :19.00   0:814         0:821   0:287   1:633       0: 22   1:845          0:596     0:963   0:300
 1st Qu.:27.00   1:186         1:179   1:713   2:333       1:200   2:155          1:404     1: 37   1:700
 Median :33.00                         3: 28       2:630
 Mean   :35.46                         4:  6       3:148
 3rd Qu.:42.00
 Max.   :75.00
```

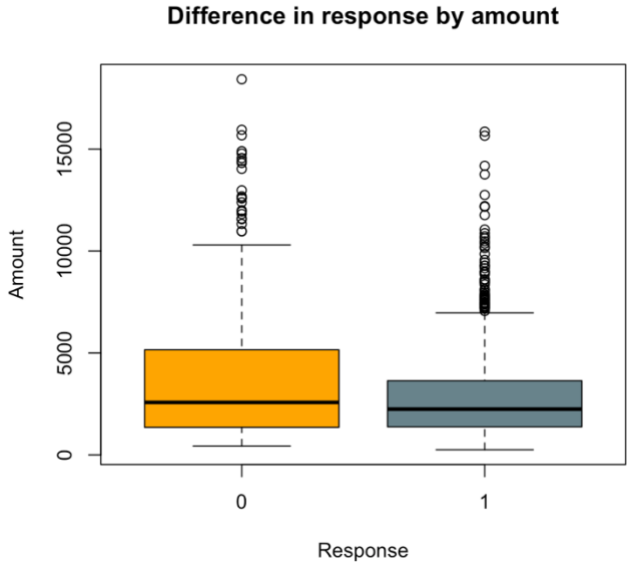
5. Univariate and Bivariate Plots:

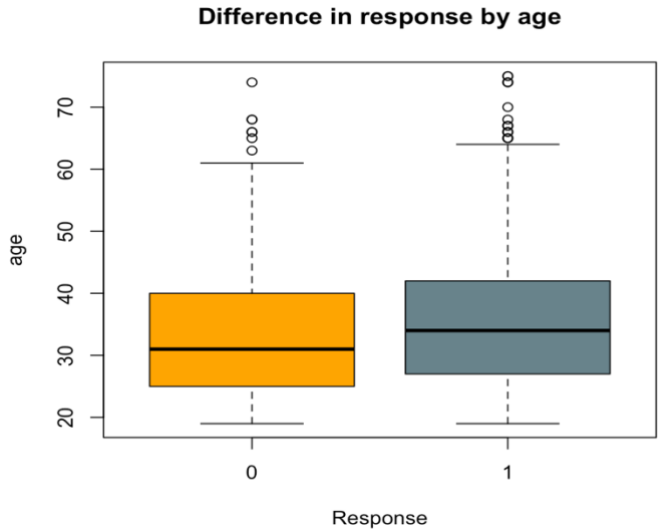


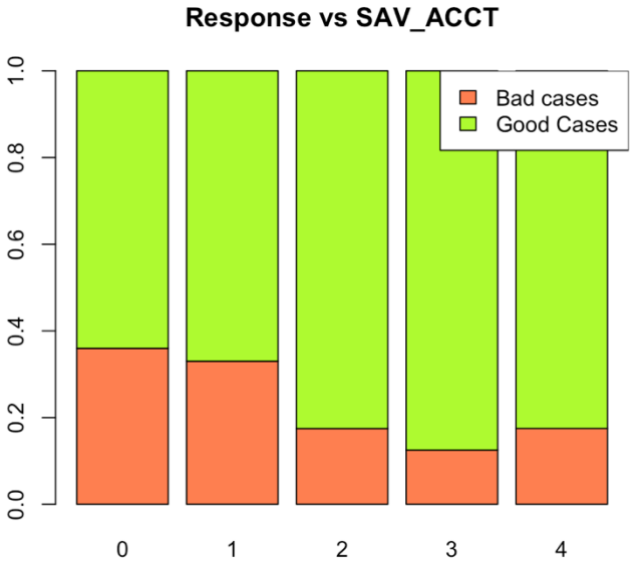
We can observe that the above plots are positively skewed as expected in descriptive stats, because mean was greater than median. This means most of the cases lie in the age of 20-40 yrs, duration of credit months are 1-20 months and credit amount are below 5000.

Please find the univariate analysis of all the remaining plots in the appendix.



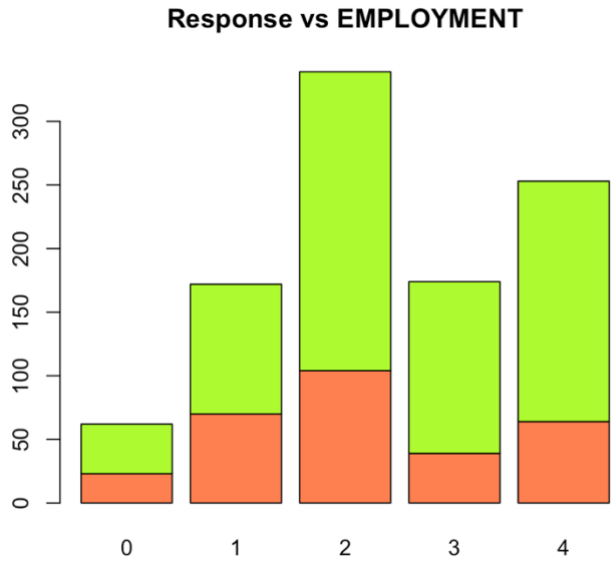






Response vs GUARANTOR

Response vs HISTORY



Interesting Observations and Anomalies:

RESPONSE vs HISTORY- Bad credit cases are higher for cases who have duly paid back their existing credits which is an anomaly. Whereas the bad credit cases should be higher for those people who delay in paying in the past or have a critical account. This is a weird relationship observed.

RESPONSE vs SAV_ACCT- Bad credit cases are higher for cases whose average balance in savings account is less than 100DM

RESPONSE vs CHK_ACCT- Bad credit cases are higher for <200DM cases. It is obvious that people with no checking account are good credit cases.

RESPONSE vs GUARANTOR – We observe that the bad credit cases are proportional for both applicant having or not having a guarantor.

After analyzing descriptive stats, univariate and bivariate plots we realized that these may be the important variables that could affect the RESPONSE variable:

AMOUNT [because the credit amount ranges from 0- 5000 are predicted to have bad cases, but 0-2000 is good cases as u see in the box plot],

EMPLOYMENT [The cases with employment< 2yrs are likely to have bad cases],

CHK_ACCT [As explained above in the interesting observations]

DURATION [The higher the number of months more likely to be a bad credit],

HISTORY and SAV_ACCT [As explained above in the interesting observations]

Question2.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

Approach: For a good model we used the following parameters:

- Split Type: Information, Gini Index
- Minbucket: minimum number of observations in a node
- Minsplit: minimum number of observations in a terminal node(leaf node)
- Cp: complexity parameter to control the size of the tree and select the optimal tree

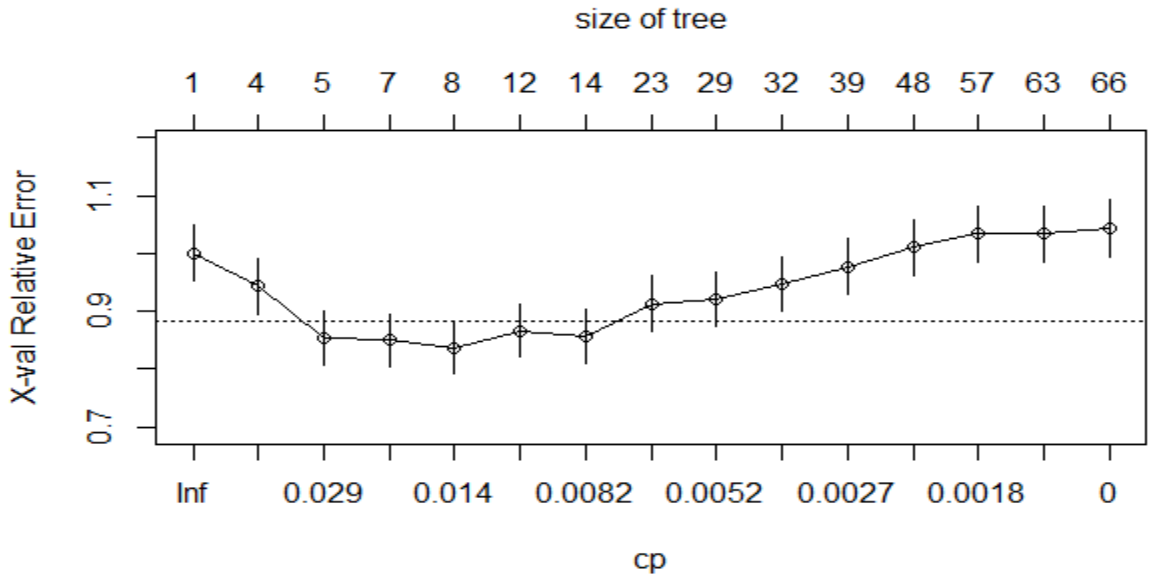
The table below describes all the parameters used and values observed for our best model

	PARAMETERS	VALUES	ACCURACY	RECALL	PRECISION
MODEL 1	INFORMATION		0.77	0.88	0.83
MODEL 2	GINI INDEX		0.794	0.88	0.833
MODEL 3	INFORMATION	cp=0 minsplit=20 minbucket=6	0.801	0.86	0.84
MODEL 4	INFORMATION	cp=0.01 minsplit=10 minbucket=3	0.801	0.87	0.844

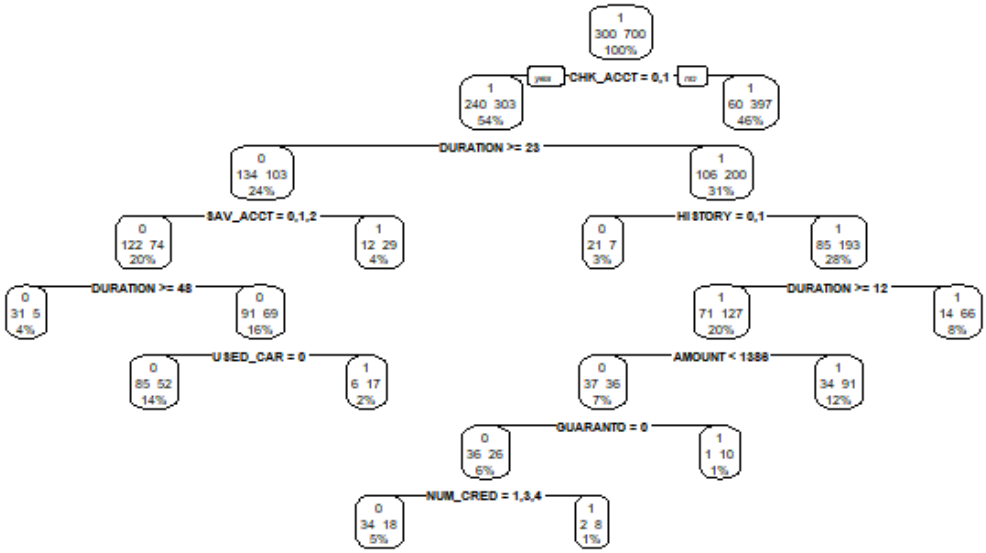
SUMMARY: MODEL 4 is the best model here with a positive rate of 87%. We used cp=0.01 as it gave back the optimal tree. In model4 with cp=0 there was a case of overfit. So to get the optimal value of cp we use printcp() .The cp with minimum xerror value is selected which is further used in our model 4.

	CP	nsplit	rel error	xerror	xstd
1	0.0516667	0	1.00000	1.00000	0.048305
2	0.0466667	3	0.84000	0.94333	0.047482
3	0.0183333	4	0.79333	0.85333	0.046003
4	0.0166667	6	0.75667	0.85000	0.045944
5	**0.0111111**	**7**	**0.74000**	**0.83667**	**0.045704**
6	0.0100000	11	0.68667	0.86667	0.046236
7	0.0066667	13	0.66667	0.85667	0.046062
8	0.0053333	22	0.59333	0.91333	0.047013
9	0.0050000	28	0.55667	0.92000	0.047120
10	0.0033333	31	0.54000	0.94667	0.047533
11	0.0022222	38	0.51667	0.97667	0.047976
12	0.0020000	47	0.49667	1.01000	0.048441
13	0.0016667	56	0.47667	1.03333	0.048751
14	0.0011111	62	0.46667	1.03333	0.048751
15	0.0000000	65	0.46333	1.04333	0.048880

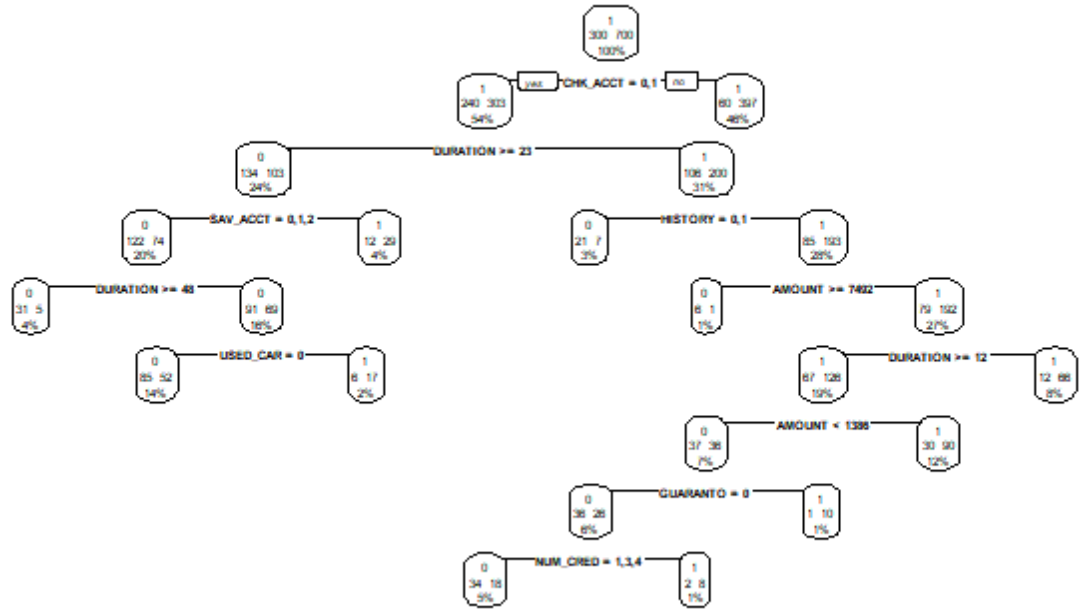
Below is the plot for different cp levels error:



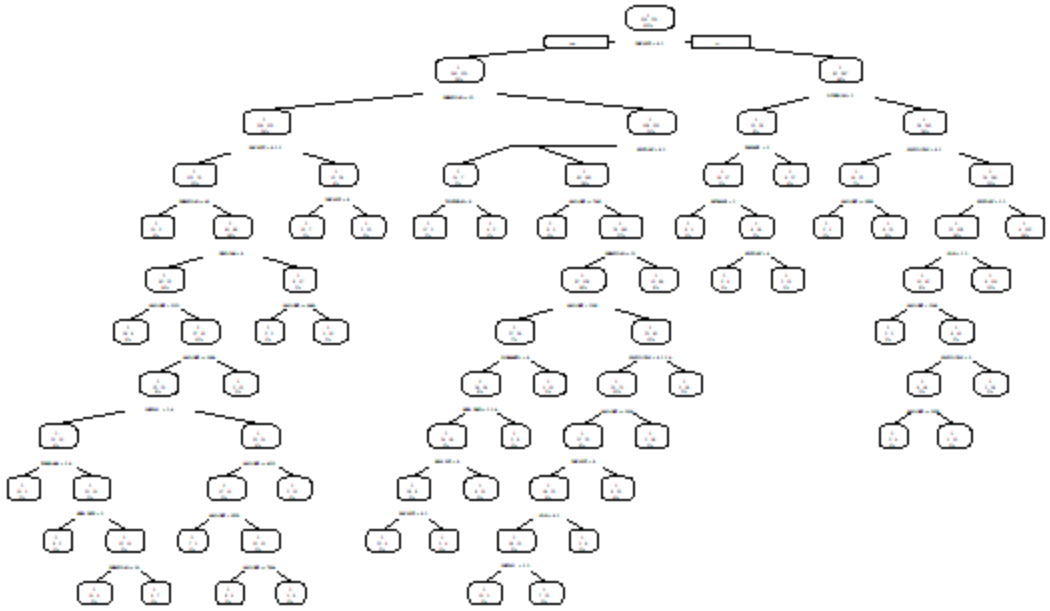
Model 1 – Information



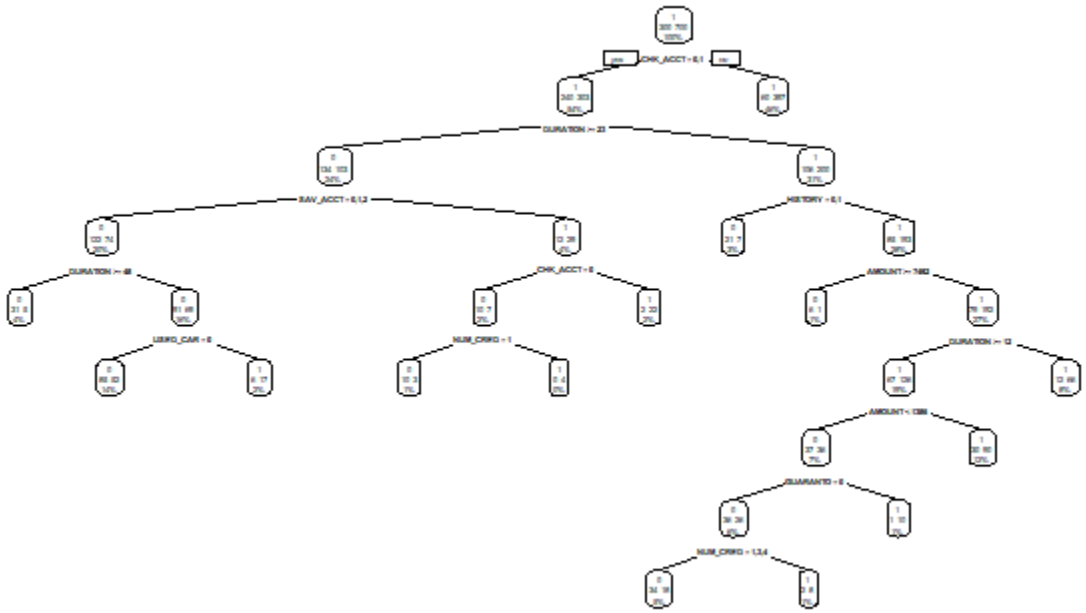
Model 2 – Gini Index



Model 3 - minsplit=20, minbucket = 6, cp=0,maxcompete = 0,maxsurrogate = 0))



Model 4 - minsplit=10,minbucket = 3,cp=0.01,maxcompete = 0,maxsurrogate = 0 (Accuracy - 0.801)



IMPORTANT VARIABLES:

For our model we get CHK_ACCT,DURATION,AMOUNT,HISTORY,SAV_ACCT,NUM_CREDITS as the important variables.It does match our expectation from question1

```
Model4$variable.importance
```

```
   CHK_ACCT    DURATION      AMOUNT     HISTORY    SAV_ACCT NUM_CREDITS    USED_CAR
  52.983269   21.985878   10.355052   10.040509    7.374515    7.074996    5.092387
  GUARANTOR
   4.481420
```

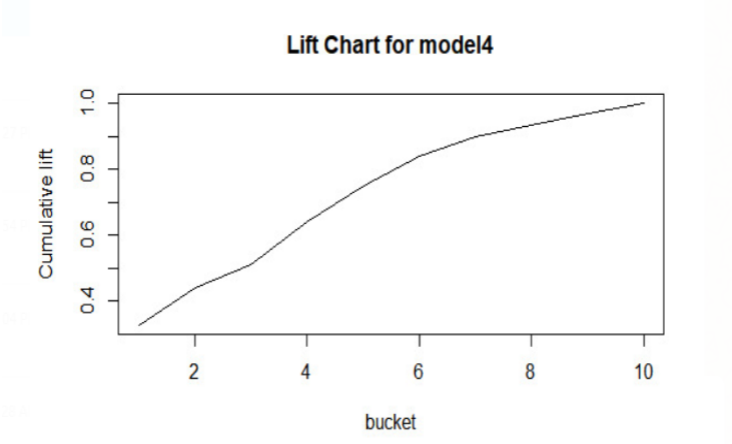
CONFUSION MATRIX

PREDICTION	DEFAULTER	NON-DEFAULTER
DEFAULTER	187	86
NON-DEFAULTER	113	614

The bad cases here is 199 and the good cases are 801.The accuracy of the model is 80%

LIFT CHART: It is a visual aid for measuring the model performance. It is an improvement of

prediction from random guess. The curve we obtained is not an ideal curve. From the lift chart we can infer its no t that reliable model since the plotting was done on complete dataset which had led to an overfit.



3. We next consider developing a model for prediction. For this, we should divide the data into Training and Validation sets. Consider a partition of the data into 50% for Training and 50% for Test (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.

I. In this part we try to divide it into several partitions with an intention to obtain the best model. So, first we consider the 50-50 partition and develop a model based on it.
The following are the parameters used and results obtained :

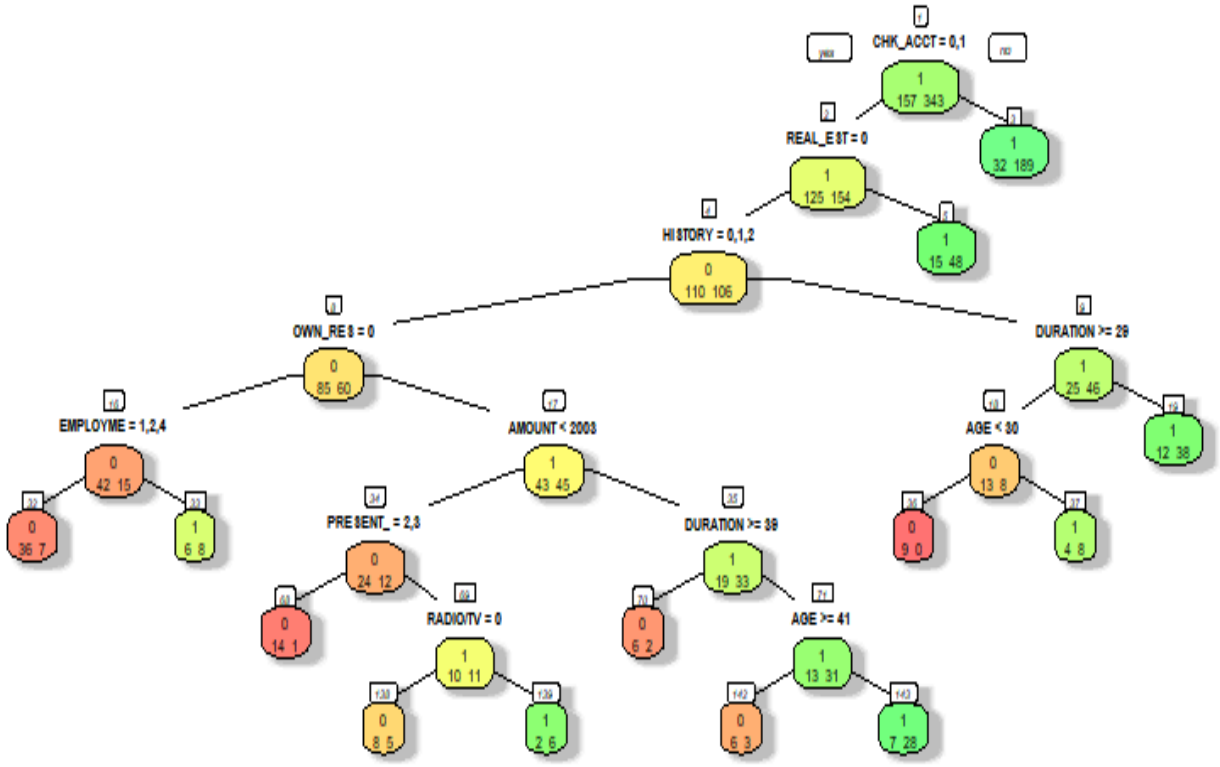
a) Model Performance – 80.8% (training set) | 74.6%(test set)
b) Decision Tree – Fig 1
c) Confusion Matrix

True / Pred	Defaulter (0)	non-defaulter (1)
Defaulter(0)	130	53
non-defaulter (1)	170	647

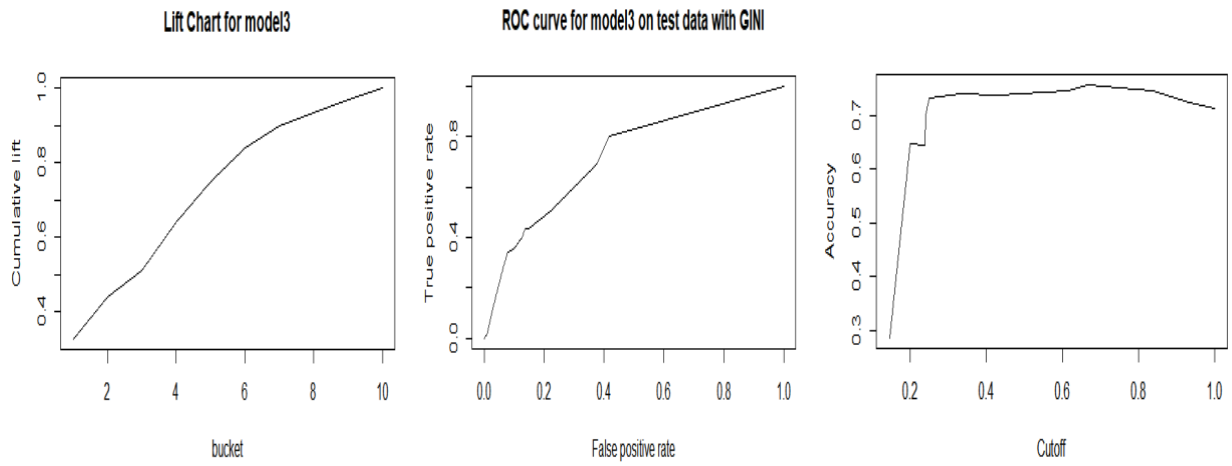
d) Levels of accuracy

Accuracy	Precision	Recall	F-score
76%	80%	90%	0.847

Good cases – 90% | Bad cases – 47%

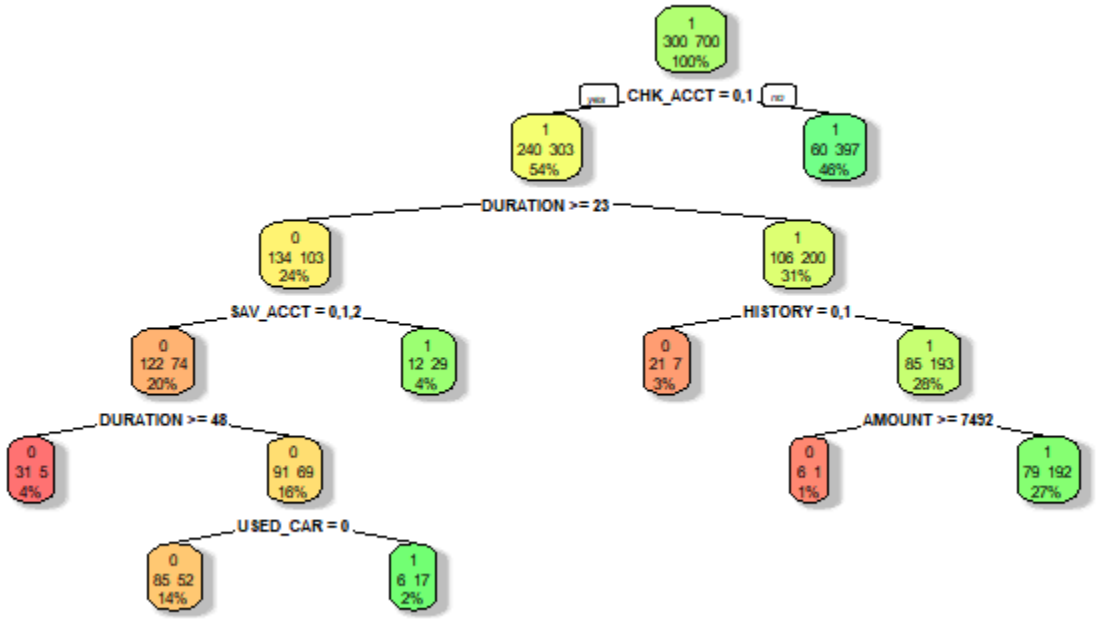


e) Lift Curve | ROC Curve | AUC Curve



In developing the models above, change decision tree options as you find reasonable (for example, complexity parameter (cp), the minimum number of cases for split and at a leaf node, the split criteria, etc.) - explain which parameters you experiment with and why. Report on if and how different parameters affect performance. Which decision tree parameter values do you find to be useful for developing a good model.

II. **Decision Tree with minsplit=10, minbucket = 3, cp=0.016**



f) Model Performance – 78.2% (training set) | 77.4%(test set)

g) Decision Tree – Fig 2

h) Confusion Matrix

True / Pred	Defaulter (0)	non-defaulter (1)
Defaulter(0)	143	65
non-defaulter (1)	157	635

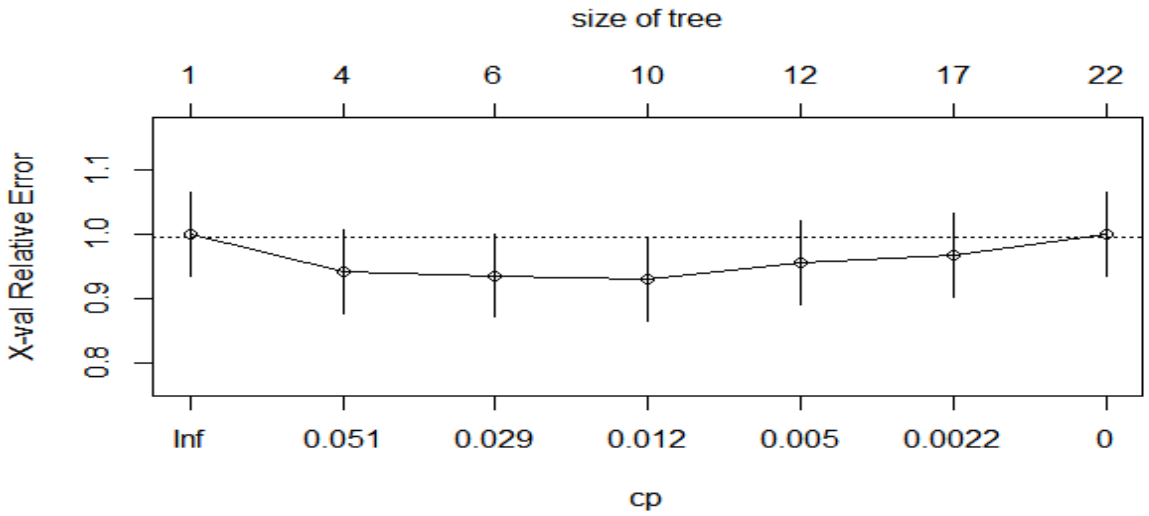
i) Levels of accuracy

Accuracy	Precision	Recall	F-score
77.8%	80%	91%	0.851

Good cases – 91% | Bad cases – 48%

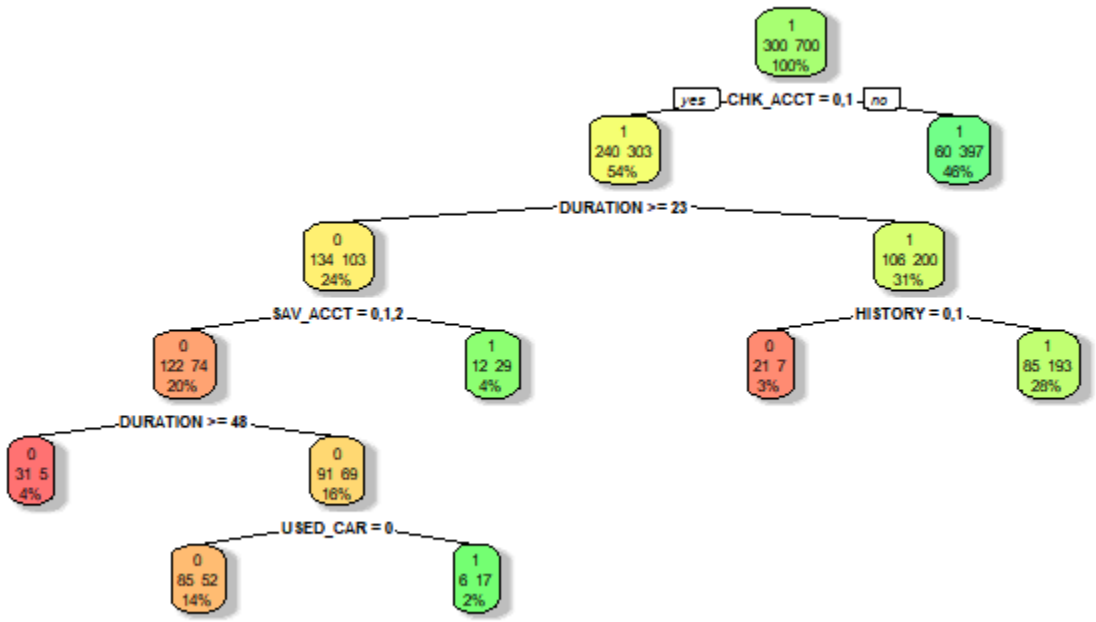
III. CP Table

CP	nsplit	rel	error	xerror	xstd
1	0.0683761	0	1.00	1.00	0.066
2	0.0384615	3	0.79	0.94	0.065
3	0.0213675	5	0.72	0.93	0.065
4	0.0064103	9	0.63	0.96	0.065
5	0.0012821	16	0.59	1	0.066



We used cost complexity pruning by selecting the value of cp from cp table, which is calculated as 0.0119. We ch ose this value as the error stops reducing after this value, since the data is not that significant. After performing c p, we get the optimal depth of the tree as 5.

IV. **Decision tree with minsplit=25, minbucket = 10, cp=0.01199**



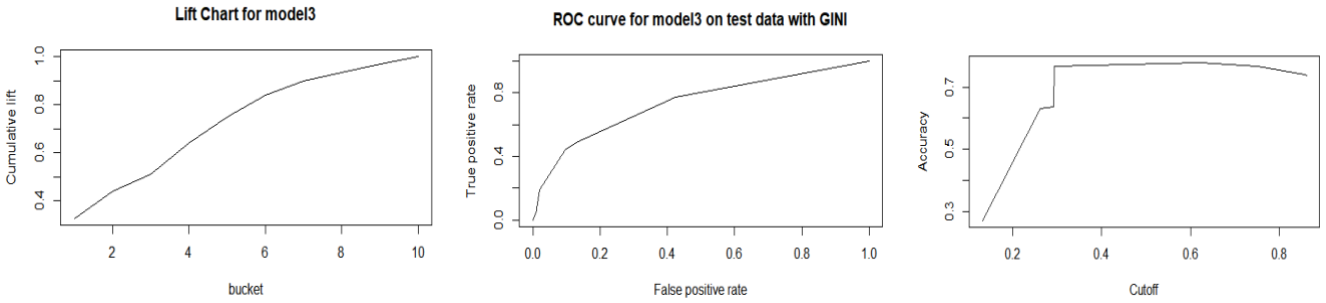
j) Model Performance – 78.2% (training set) | 77.4%(test set)

k) Decision Tree – Fig 2

l) Levels of accuracy

Accuracy	Precision	Recall	F-score
77.8%	81%	91%	0.851

Good cases – 91% | Bad cases – 48%



The curve moves towards 1.0 and area under the curve is high, so we can say that our model is good (not considering other models and parameters). Lift is the proportion of expected responses we get using our model, to the expected response we get without using any model. The cumulative gain here is greater than one, our model is better than random selection.

Describe the pruning method used here. How do you examine the effect of different values of cp, and how do you select the best pruned tree?
Explain how you use different performance measures to determine your best model.

Split & Node Parameters	Data Spit	Accuracy	precision	recall	F-score	ROC & AUC
minsplit=25 cp = 0.01199	50-50%	78%	80%	91%	85.1%	AUC = 0.737 has the least AUC among all the data splits
Minsplit=10 Cp = 0.016	50-50%	78%	80%	90%	85%	AUC = 0.732
minbucket=3	70-30%	67%	81%	85%	82%	AUC = 0.7281
No node parameters	80-20%	76%	80%	90%	85%	AUC= 0.7765 best of all three data splits

Analysing the table above we can see that the model with the 80-20 split gives better results compared to the other models.

(b) Consider another type of decision tree – C5.0 – experiment with the parameters till you get a 'good' model. Summarize the parameters and performance you obtain.

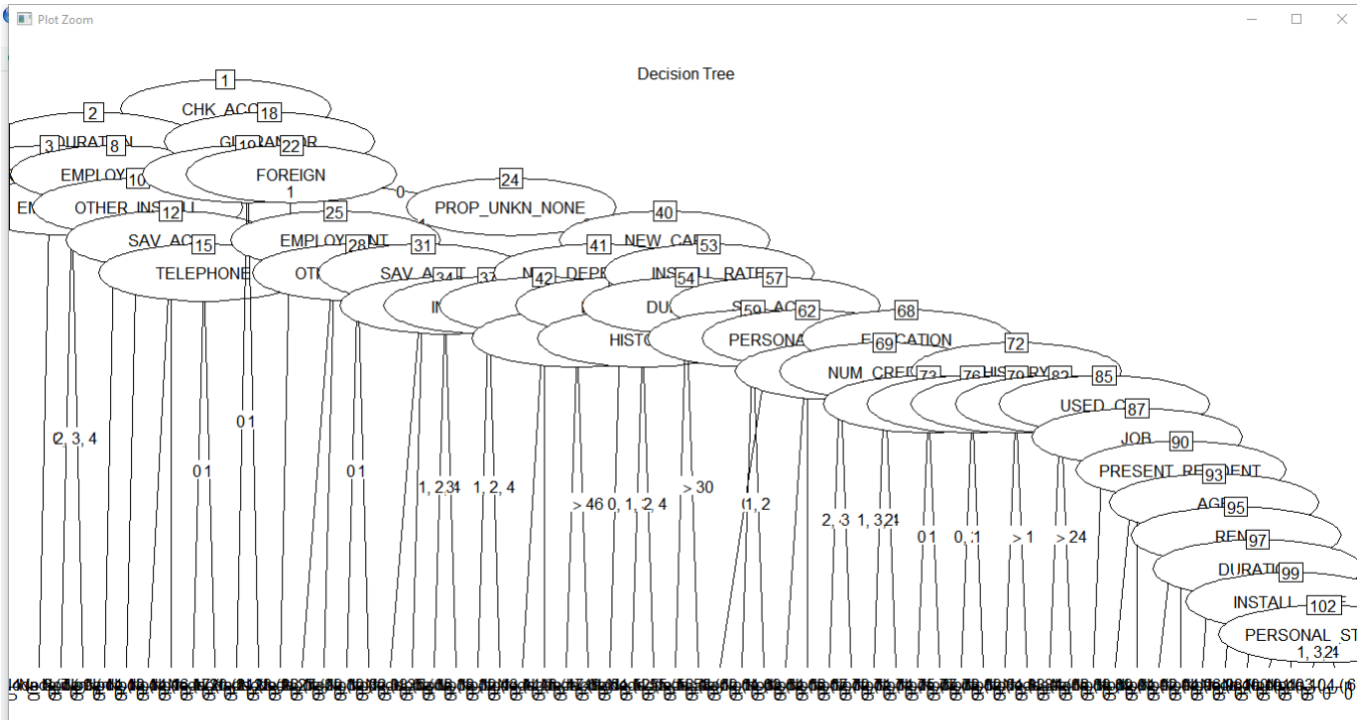
Also develop a set of rules from the decision tree and compare performance.

Does performance differ across different types of decision tree learners? Compare models using accuracy, sensitivity, precision, recall, etc (as you find reasonable – you answer to questions (a) above should clarify which performance measures you use and why). Also compare performance on lift, ROC curves and AUC.

How do the models obtained from these decision tree learners differ?

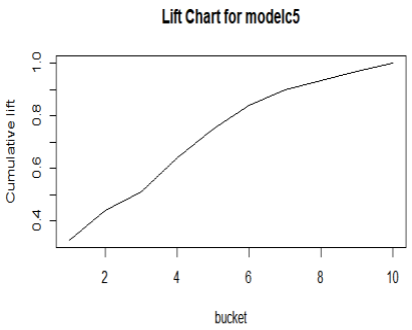
ROC curves tell you how well your model separates the two classes, no matter where the threshold value is. Accuracy is a measure which works well usually when classes keeps the same balance on train and test sets, and when scores are really probabilities.

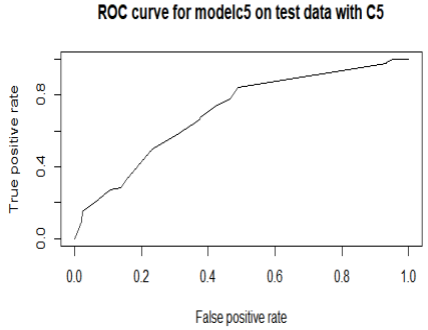
Resulted in a model with more than 20 splits, the tree depth has been limited to 6 to prune the tree to make it robust on unseen data.

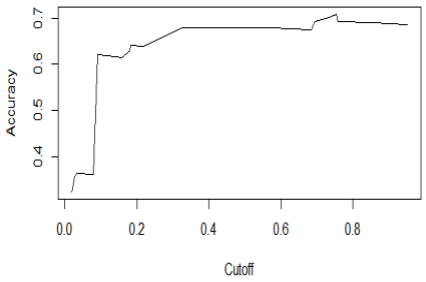


Node Parameters	Data Split	Accuracy	precision	recall	F-score	ROC & AUC
minCases =6, noGlobalPruning = TRUE	50-50%	75.8%	78.9%	89.2%	83%	AUC = 0.7717
minCases=6, winnow = TRUE	50-50%	77.8%	77.8%	90%	83.6%	AUC = 0.7717 best of all three data splits
minCases=6, winnow = TRUE	70-30%	78.8%	77.8%	89%	83%	AUC = 0.7643 better than 80-20%
minCases=6, winnow = TRUE	80-20%	79%	78.9%	90%	83.8%	AUC= 0.6856 has the least AUC among these three data splits

The best model seems to be the one with 80-20 split and minCases=6 . So, this is considered to be a good model.







c) Decision tree models are referred to as 'unstable' – in the sense that small differences in training data can give very different models. Examine the models and performance for different samples of the training/test data (by changing the random seed). Do you find your models to be unstable – explain.

C5 Model with information gain

Seed	**Accuracy test**	**Accuracy train**
02475	68	80
456	65	78

C5 Model with Gini Index

Seed	**Accuracy test**	**Accuracy train**
02475	66	82
456	65	76

70-30 split of training and validation

Seed Value	Accuracy(Training)	Accuracy(Validation)	Difference
02475	83%	76.33%	6.67%
456	76%	73%	3%

80-20 split of training and validation

Seed Value	Accuracy(Training)	Accuracy(Validation)	Difference
02475	82%	73.5%	8.5%
456	80%	71%	9%

The 70-30 split model is unstable, because when we changed the seed value in training data, there is a drastic change in the differences of accuracies. Whereas 80-20 split model is more stable with an average difference of 8%.

d) Which variables are important for separating 'Good' from 'Bad' credit? Determine variable importance from the different 'best' trees. Are there similarities, differences?
Explain how variable importance is determined (for rpart and C5.0 models)

We have carried out chi square test, to determine the variable importance. After analysis, we have selected both models, built on 80-20 data split, to be the best.

Model Considered as "Best"	Important Variables (Top 4)
80-20 split, rpart(no control parameters)	CHK_ACCT, DURATION, AMOUNT, EMPLOYMENT
80-20 split, C50(Control Parameters: minCases=3, winnow = TRUE)	CHK_ACCT, REAL_ESTATE, HISTORY, DURATION

- From C5.0 model, the important variables are CHK_ACCNT, DURATION, HISTORY, SAV_ACCNT, NEW_CAR
- From the model with Gini Index split, the important variable are CHK_ACCNT, DURATION, HISTORY, SAV_ACCNT, AMOUNT

The similarity found between the models is that both checking account and duration are important variables across models. Both models have picked Checking Account and Duration to be important variables. The other important variables seem to vary like for example NEW_CAR

(e) Consider partitions of the data into 70% for Training and 30% for Test, and 80% for Training and 20% for Test and report on model and performance comparisons (for the decision tree learners considered above).
In the earlier question, you had determined a set of decision tree parameters to work well. Do the same parameters give 'best' models across the 50-50, 70-30, 80-20 training-test splits? Are there similarities among the different models ….in, say, the upper part of the tree, and in variable importance – and what does this indicate?

Is there any specific model you would prefer for implementation?

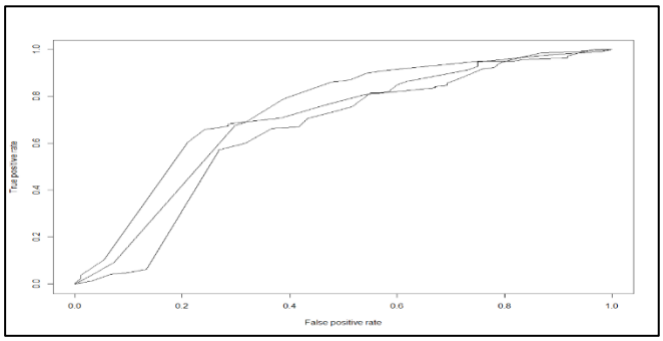
Split Ratio	Training Data	Test Data
50:50	0.828	0.76
70:30	0.825	0.74
80:20	0.8325	0.805

Chosen Parameters : **minsplit=10, minbucket = 3, cp=0.01199, split=GINI, threshold=0.8**

As the three models are compared as shown above, the conclusion is that the model with 80% -20% partitioning of data into training and testing is better than the other two for implementation because accuracies on test data replicates the model's performance on train data

Referring to that, we can say that the best model considered here is C5.0 decision tree with parameters winnow= TRUE and minCases=3. The accuracy levels for each data split has been given above. The best model is when the data split is 80-20%. The similarity is that, in all three data splits the root node is CHK_ACCT, but the number of splits are different. Even if 80-20% data model has a bigger tree than other data splits, it results in the more homogenous sub-nodes and greater accuracy.

The graph here shows the ROC curves for all three data splits for C5.0 decision tree.



4. Consider the net profit (on average) of credit decisions as: Accept applicant decision for an Actual "Good" ca se: 100DM, and Accept applicant decision for an Actual "Bad" case: -500DM This information can be used to de termine the following costs for misclassification: Predicted Actual Good Bad Good 0 100DM Bad 500DM 0 3 (a) Use the misclassification costs to assess performance of a chosen model from Q 3 above. Compare model perfo rmance. 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 perfor mance you find? (b) Calculate and apply the 'theoretical' threshold and assess performance – what do you noti ce, and how does this relate to the answer from (a) above. (c) Use misclassification costs in building the tree m odels (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).

MODELS	ACCURACY	AUC
80:20 split	0.78	0.77
80:20 split with misclassification	0.775	0.715

We can observe that the misclassification cost is less AUC values.

MODELS	THRESHOLD VALUE	MISCLASSIFICATION FOR TRAINING SET	MISCLASSIFICATION FOR TEST SET	ACCURACY (TRAINING)	ACCURACY (TEST)
80:20	0.8	505	152	82.37	78
	0.65	537	165	67	66
	0.5	543	175	64	60

So when we increase the threshold misclassification cost will decrease

THEORITICAL THRESHOLD CALCULATION:

Misclassification cost in building tree model:

Applying the misclassification cost gives us the following important variable for Rpart

WITHOUT COST MATRIX	WITH COST MATRIX
CHK_ACCNT	CHK_ACCNT
DURATION	NEW_CAR
HISTORY	DURATION

So we find a new important variable after applying cost matrix

MODEL(C5.0)	THRESHOLD	WMC(TRAIN)	WMC(TEST)	MC(TRAIN)	MC(TEST)
80:20 split					

**WMC=With Misclassification Cost

**MC=Classification Cost

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? Calculate the smoothed values for these 'probabilities' for "Good" and "Bad" cases in these nodes – calculate the Laplace smoothing and m-estimate smoothing values.

Approach: The best model for us is a 80:20 split.

Depth of a decision tree is the length of the longest path from the root to the leaf.To calculate the tree depth we can use the function

tree.depth which gives us the tree depth value as:

```
nodes <- as.numeric(rownames(GermanCreditModel2$frame))
max(rpart:::tree.depth(nodes))
```

Note:Also we can use node.depth function to get the depth.

For the number of nodes: (N^(L-1)/(N-1)

Each node has N subnodes and the tree is L levels deep

The important variables for classifying “Good” vs “Bad” credit

CHK_ACCNT

EMPLOYMENT

DURATION

HISTORY

AMOUNT

SAV_ACCNT

Pure leaf nodes :a/b

a=no. of records with class1

b=no of records with class 0

For smoothening the probabilities we use Laplace smoothening and m-estimate smoothening.

Laplace correction: prob(class k) = (nk + 1)/ (N + C)

N: # cases at the node

C: # classes among cases at the node

nk : # cases in class k

m-estimate:

prob = (k+bm)/ (n + m)

where b: base rate, m: controls how much to shift toward the base rate (Guide: given b, use m such that bm=10)

6. The predicted probabilities can be used to determine how the model may be implemented. We can sort the data from high to low on predicted probability of "good" credit risk. Then, going down the cases from high to low probabilities, one may be able to determine an appropriate cutoff probability – values above this can be considered acceptable credit risk. The use of cost figures given above can help in this analysis. For this, first sort the validation data on predicted probability. Then, for each validation case, calculate the actual cost/benefit of extending credit. Add a separate column for the cumulative net cost/benefit. 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.

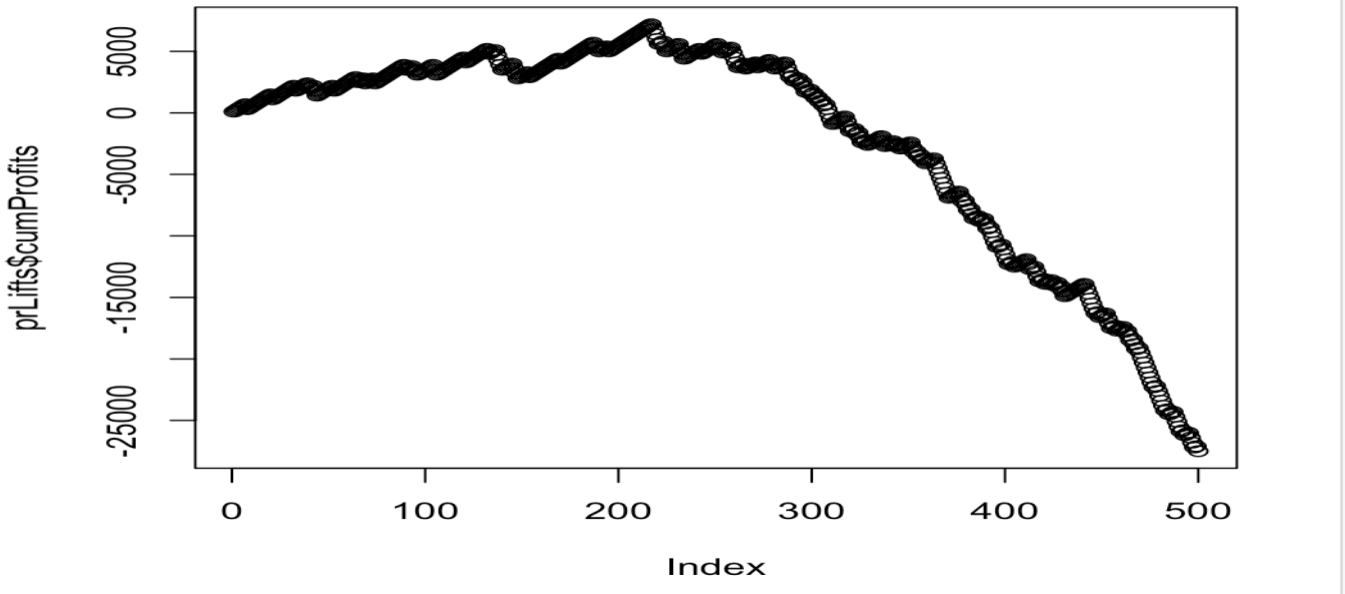
For every correct prediction we gain 100DM and for a misclassification we lose 400DM.

When we sort the data based on the predicted probability by ordering scoreTst. We obtain our maximum benefit to be 7200DM and our cutoff value of scoreTst is 89.03.

We have calculated the cumulative costs associated with the above assumptions.

Thus, we recommend keeping 89.03% (0.8903) as the cutoff for future credit applicants.

Actual Response	Predicted Response	Profit/Loss Cost	Cumulative Cost
0	0	100	100
1	1	100	200
1	0	-400	-200
0	1	-400	-600



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? 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 (as in question 6).

These are the important parameters to be modified to get better performance: **mtry & ntree**

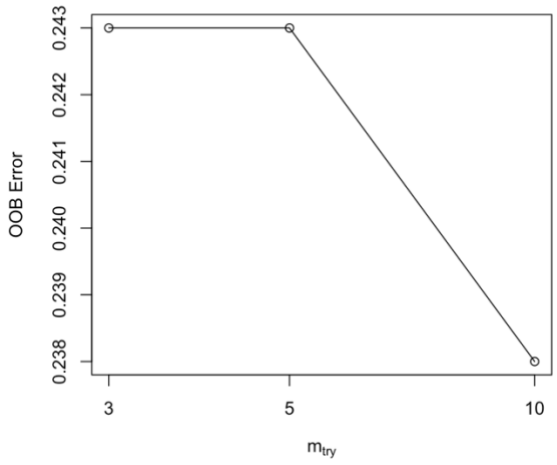
mtry	ntree	OOB Estimate
5	200	25.71
8	500	24.71
5	500	24.29
4	500	24.71
10	500	23.86

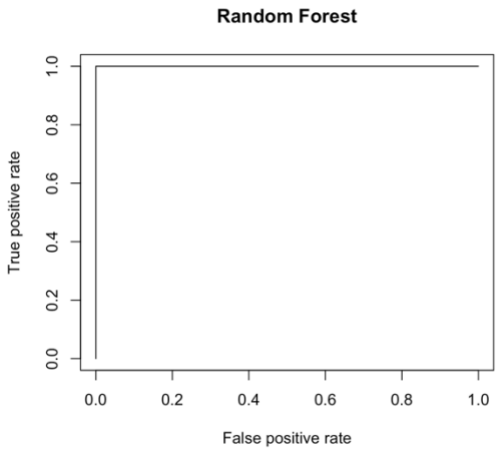
We find that the the oob estimate of error rate is least for mtry=10 i.e.23.86

Also we can use the below syntax to get the minimun OOB estimate sqrt(M)~m

```
> mtry_opt <- rf1[,"mtry"][which.min(rf1[,"OOBError"])]
> print(mtry_opt)
10.OOB
    10
```

NOTE: Also we can use the function tuneRF() in place of the randomForest() function to train a series of models with different mtry values and examine the the results. It will give us the best mtry value.Below is the plot for mtry value with the lowest OOB estimate:





We observe here that mtry=10 has the lowest OOB Error

The following plot gives us the variable importance for the Testing data and training data rfmodel. We observe CHK_ACCT is the most important variable followed by HISTORY and DURATION.

Training Data: OOB estimate of error rate: 24.86%

Testing Data: OOB estimate of error rate: 25.33%