Assignment 1

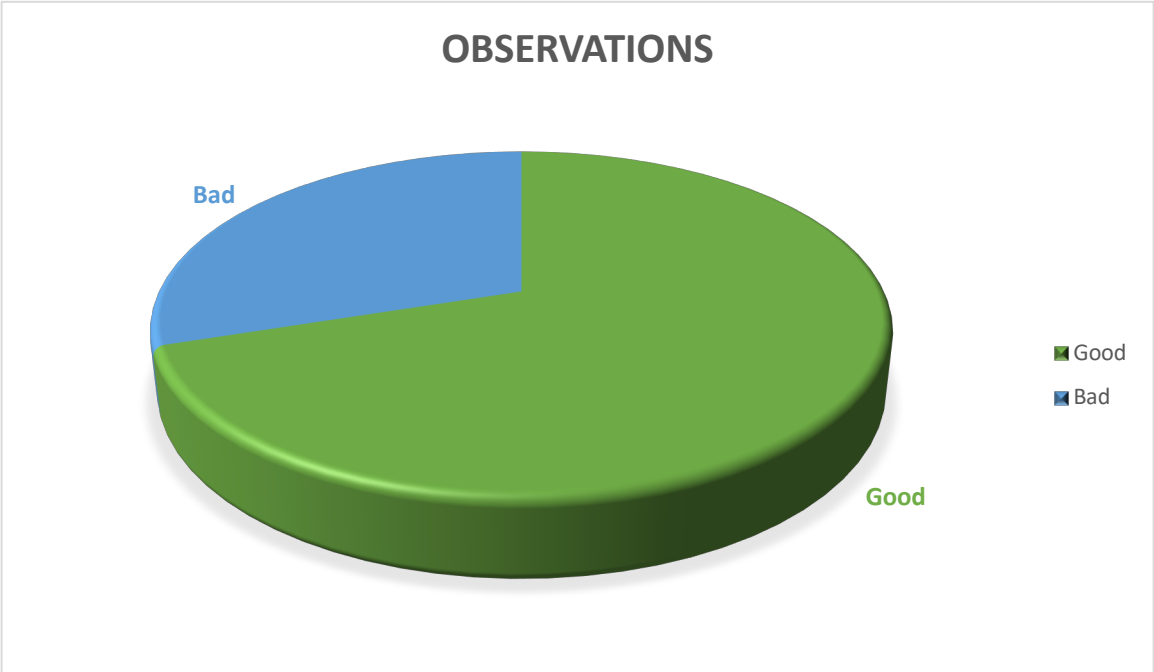
DECISION TREE FOR CREDIT EVALUATION

Gupta, Srija

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?

Answer:1

The data consists of 1000 observations. Out of these, the good cases constitute 70% of the data while the remaining 30% are bad cases. The proportion of good to bad is 7:3. This proportion is represented by the following graph.



Missing Values:

Out of the 31 columns, **7 had missing values**.

The 6 columns

NEW_CAR, USED_CAR, FURNITURE, RADIO.TV, EDUCATION, RETRAINING

contain binary values i.e. 0 for false and 1 for true. These columns contained the values only for the true cases. The rest of the rows contained NA. These NAs were replaced by 0 using the statement:

```
dataset$column_name[is.na(dataset$column_name)] <- 0
```

e.g.: `GCredit$NEW_CAR[is.na(GCredit$NEW_CAR)] <- 0`

The column AGE also contains null values.
These missing values were handled by computing the average age and replacing the null values by this average age.

After exploring the data, we concluded that the following variables are most relevant to the interested outcome:
History: it gives us the information about the customer past credit record. Which can help determine the credibility of the customer.
Employment: it gives the customer's employment status. Which is a strong factor in determining the customer's ability of pay back the credit.
Own_res: It indicates if the customer owns a residence
CHK_ACCT: Check for the customer's checking account
SAV_ACCT: Check for the customer's Saving account
AGE: Indicates Age
Amount: Gives the amount requested by the customer

Predictor Variables:

Variable	Min	Max	Range	Average	Std Deviation	Median
Duration	4	72	68	20.9	12.06	18
Amount	250	18424	18174	3271.26	2822.74	2319.5
Installment rate	1	4	3	2.97	1.12	3
Age	19	75	56	35.55	11.38	33
Num Credits	1	4	3	1.41	0.58	1
Num Dependents	1	2	1	1.16	0.36	1

Categorical variables:

Variables with types	# of Bad credits	% of Bad Credits	# of Good Credits	% of Good credits	Ratio of bad to good
Checking account Type 0	135	45	139	20	0.97
Type 1	105	35	164	23	0.64
Saving account Type 1	34	17	69	8	0.49
Employment < 1year	70	23	102	15	0.69
Present resident >4 years	124	41	289	41	0.43
Job	186	62	444	63	0.42

Employee/officials					
History Type1 all credits at bank paid back	28	4	21	1	1.33

From the above table, we can infer that the:

- People with checking account of type 3 have a good credit score compared to people with account of type 0 and 1.
- Saving account in category 1 as compared to the others has the highest bad to good ratio of credit score.
- The ratio of credit decreases with decrease in experience in the employment. But it is observed that the people with less than one year of experience have comparatively a worse credit score than the unemployed.

- People who have stayed in/ been a resident for 4 years or more are observed to have a bad credit score, but this is not enough to form a strong opinion as the people who have been a resident for 1 to 2 years are also observed o have bad credit score.

- From the Job variable, it is observed that the people whose job is employee or an official have a bad credit history. Though their number of bad score is more, so is their good score. The percentage of bad score is similar to the percentage of good score.

- Also, from the past history variable, it is seen that the people who have paid back the credits to the bank fully have a bad credit score as compared to the others.

2. 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. What decision tree node parameters do you use to get a good model (and why?)

Parameters	Model-1	Model-2	Model-3
Minsplit	5	5	6
MaxDepth	15	15	20
MinBucket	2	6	5
Accuracy	0.78	0.773	0.773

#Inference: We played around with values of Minsplit,Maxdepth and minbucket. We kept 2 functions constant and manipulated the rest and we found out that changing the minbucket(the minimum number of observations in any terminal node), mostly affects the model accuracy. The best model obtained is the Model-1 with **Accuracy 0.78**.

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

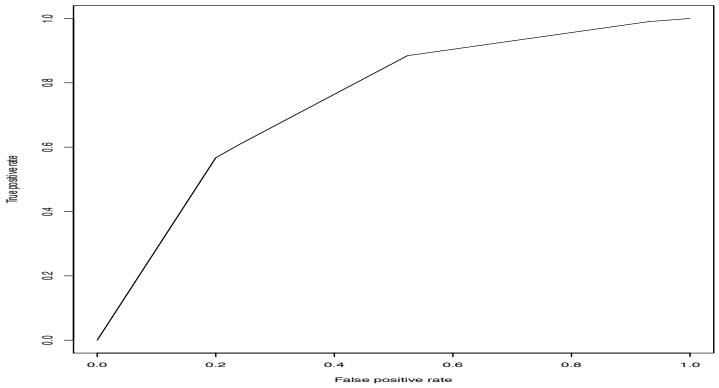
```
Variable importance
       CHK_ACCT         HISTORY        SAV_ACCT        DURATION          AMOUNT PRESENT_RESIDENT
             49              15              13              10               4               3
 PROP_UNKN_NONE      EMPLOYMENT     NUM_CREDITS     REAL_ESTATE             JOB
              2               1               1               1               1
```

We took out the summary of the whole model and found out that variables like **CHK_ACCT,HISTORY,SAV_ACCT,DURATION and AMOUNT** are the most important factors for constructing our model with respect to our dependent **RESPONSE** variable. These match with the expected variables were most relevant to the outcome of our dependent variable.

(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.

```
     true
pred   0   1
   0 147  67
   1 153 633
```

Good cases: (633/700)*100=90.42%
Bad case: (147/300)*100=49%
The accuracy of **Good cases is 90.42%** and the for the **bad cases is 49%**



From the lift curve we can see that we have obtained an area under the curve(AUC),but it is an overfit model and hence not reliable as we made a model using the entire dataset which may or may not work when we give it a set of other different data values.

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.
Is the model reliable (why or why not)?
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.
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.
Which decision tree parameter values do you find to be useful for developing a good model.

Parameters	Model-1				Model-2			
Minsplit	5				5			
MaxDepth	15				15			
MinBucket	2				6			
Cp value(For Pruning)	0.01	0.05	0.07	0.10	0.01	0.05	0.07	0.10
Accuracy_Train	0.84	0.73	0.70	0.70	0.82	0.73	0.70	0.70
Accuracy_Test	0.71	0.72	0.69	0.69	0.71	0.72	0.69	0.69

So we take two models, and play around with Minsplit,Maxdepth and minbucket and the cp value, We see that the model improves with increasing the complexity parameter upto a level (cp=0.07)and then there isn't any effect of changing CP value on the models. Now we try playing with the threshold values and predicting results.

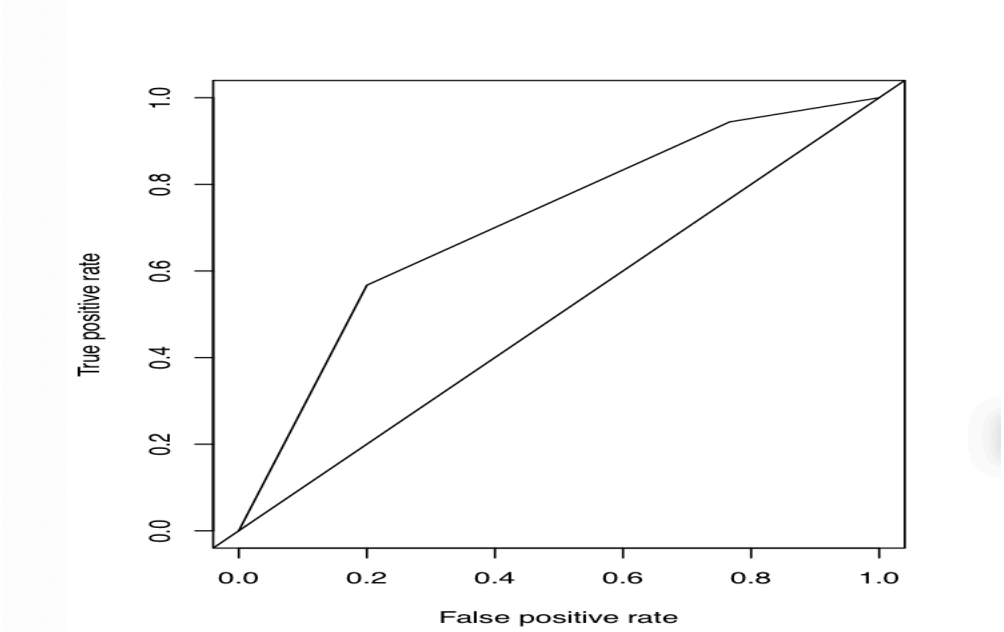
Parameters	Model-1		Model-2	
Minsplit	5		5	
MaxDepth	15		15	
MinBucket	2		6	
CTHRESH	0.5	0.3	0.5	0.3
Accuracy_Train	0.73	0.70	0.73	0.70
Accuracy_Test	0.72	0.69	0.72	0.69

```
> accuracy
[1] 0.726
> precision = diag / colsums
> recall = diag / rowsums
> f1 = 2 * precision * recall / (precision + recall)
> f1
        0         1
0.3381643 0.8272383
> precision
        0         1
0.6250000 0.7387387
> recall
        0         1
0.2317881 0.9398281
```

We obtain a model with precision and accuracy of 0.738 and 0.726 respectively.
In developing the above model, we considered various options for parameters:
1)**Minsplit**:Min number of observation that should be present in a node ,in order for a aplit to be done.
2)**MaxDepth**:Used for controlling over-fitting
3)**MinBucket:**It is the minimum number of observations in the terminal node

The following are the pruning methods taken into account to make this model:

1)**Compexity Parameter:** It is used to control the size of the decision tree and helps in selecting the optimal sized Decision tree. We see that the model improves with increasing the complexity parameter upto a level (cp=0.07)and then there isn't any effect of changing CP value on the models.
2)**Cthresh**: It is the probability above which a value is for good/bad credits. As we play around with cthresh, we realize that there is an improvement in the model upto a ceratin cthresh value(~0.6) and then the model performance remains constant.



This model seems robust as there is a significance area under the curve.Area Under the Curve is. 0.7117857.

(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?

Taking the whole dataset to calculate accuracy.

```
c50Tree = C5.0 (RESPONSE~., data = datas, rules = FALSE,control = C5.0Control (subset =
TRUE,bands = 0, winnow=FALSE, CF=0.25, minCases=2, noGlobalPruning=FALSE,
sample=0,fuzzyThreshold = FALSE,seed = sample.int(4096, size = 1) -
                                1L, earlyStopping = TRUE, label = "RESPONSE"))
```

First we made a decision tree using all the variables, taking out the summary of the model to find the variable importance, we get

```
> c50Tree = C5.0 (RESPONSE~., data = datas, rules = FALSE,control = C5.0Control (subset = TRUE,bands = 0, winnow=FALSE,
CF=0.25, minCases=3, noGlobalPruning=TRUE, sample=0,fuzzyThreshold = FALSE,seed = sample.int(4096, size = 1) -
+                                          1L, earlyStopping = TRUE, label = "RESPONSE"))
> C5imp(c50Tree)
                  Overall
CHK_ACCT            100.0
GUARANTOR            54.3
SAV_ACCT             50.0
OTHER_INSTALL        49.0
DURATION             44.3
FOREIGN              35.6
HISTORY              34.5
EDUCATION            30.2
RETRAINING           29.3
USED_CAR             24.3
INSTALL_RATE         19.5
REAL_ESTATE          11.1
NEW_CAR              10.8
MALE_MAR_or_WID      10.1
CO.APPLICANT          9.8
MALE_DIV              8.0
AMOUNT                7.1
PRESENT_RESIDENT      6.7
TELEPHONE             5.5
NUM_CREDITS           4.5
OBS.                  3.5
EMPLOYMENT            2.9
RADIO.TV              2.2
```

For the variables we see their overall importance, till 35.6 there isn't very huge difference in the values, but after 35.6 there's a huge difference, thus the important variable we are taking is **CHK_ACCT,GUARANTOR,SAV_ACCT,OTHER_INSTALL,DURATION,FOREIGN**
So there are a set of possible combinational parameters for C5.0 DT methods

CASE:1 CHECKING FOR THE WHOLE DATASET

Control parameters	Model 1	Model 2	Model 3	Model 4	MODEL 5
Bands	0	0	0	0	0
CF	0.25	0.25	0.50	0.25	0.25
minCases	3	3	3	10	1
noGlobalPruning	TRUE	TRUE	FALSE	FALSE	TRUE
Winnow	FALSE	FALSE	FALSE	FALSE	FALSE
fuzzyThreshold	FALSE	FALSE	FALSE	FALSE	FALSE
earlyStopping	TRUE	TRUE	TRUE	TRUE	TRUE
Subset	TRUE	FALSE	FALSE	TRUE	FALSE
ACCURACY_TEST	0.772	0.78	0.76	0.749	0.784

For the whole dataset,the maximum accuracy that we are getting is 0.894, And the major control parameter that is palying a role is "minCases" an integer for the smallest number of samples that must be put in at least two of the splits. ALSO SUBSET PLAYS A ROLE.

```
> table(pred=predict(rpModel1,mdData, type="class"), true=mdData$RESPONSE) #Confusion matrixx
    true
pred   0   1
   0  70  39
   1 230 661
```

CASE:2 Training=50% DATASET,TESTING=50%DATASET

Control parameters	Model 1	Model 2	MODEL 3

Bands	0		0		0	
CF	0.25		0.25		0.25	
minCases	3		3		1	
noGlobalPruning	TRUE		TRUE		TRUE	
Winnow	FALSE		FALSE		FALSE	
fuzzyThreshold	FALSE		FALSE		FALSE	
earlyStopping	TRUE		TRUE		TRUE	
Subset	TRUE		FALSE		FALSE	
ACCURACY_TEST	0.762		0.73		0.778	
ACCURACY_TRAIN	0.73		0.72		0.722	
ACCURACY	0.726		0.726		0.72	
PRECISION(0:1)	0.62	0.73	0.62	0.73	0.54	0.78
RECALL(0:1)	0.23	0.93	0.23	0.93	0.47	0.83
F1(0:1)	0.33	0.82	0.33	0.82	0.50	0.80

The maximum precision obtained is 0.78,with 0.72 accuracy

CASE:3 Training=60% DATASET,TESTING=40%DATASET

Control parameters	Model 1		Model 2		MODEL 3	
Bands	0		0		0	
CF	0.25		0.25		0.25	
minCases	3		3		1	
noGlobalPruning	TRUE		TRUE		TRUE	
Winnow	FALSE		FALSE		FALSE	
fuzzyThreshold	FALSE		FALSE		FALSE	
earlyStopping	TRUE		TRUE		TRUE	
Subset	TRUE		FALSE		FALSE	
ACCURACY_TEST	0.75		0.73		0.76	
ACCURACY_TRAIN	0.74		0.69		0.73	
ACCURACY	0.73		0.7375		0.73	
PRECISION(0:1)	0.61	0.75	0.61	0.75	0.52	0.82
RECALL(0:1)	0.22	0.94	0.22	0.94	0.57	0.79
F1(0:1)	0.33	0.83	0.33	0.83	0.55	0.81

The maximum precision obtained is 0.82,with 0.73 accuracy

CASE:4 Training=70% DATASET,TESTING=30%DATASET

Control parameters	Model 1	Model 2	MODEL 3
Bands	0	0	0
CF	0.25	0.25	0.25
minCases	3	3	1
noGlobalPruning	TRUE	TRUE	TRUE
Winnow	FALSE	FALSE	FALSE

fuzzyThreshold	FALSE		FALSE		FALSE	
earlyStopping	TRUE		TRUE		TRUE	
Subset	TRUE		FALSE		FALSE	
ACCURACY_TEST	0.762		0.76		0.77	
ACCURACY_TRAIN	0.73		0.75		0.74	
ACCURACY	0.726		0.73		0.7433	
PRECISION(0:1)	0.62	0.73	0.67	0.73	0.57	0.81
RECALL(0:1)	0.23	0.93	0.22	0.95	0.58	0.81
F1(0:1)	0.33	0.82	0.33	0.83	0.58	0.81

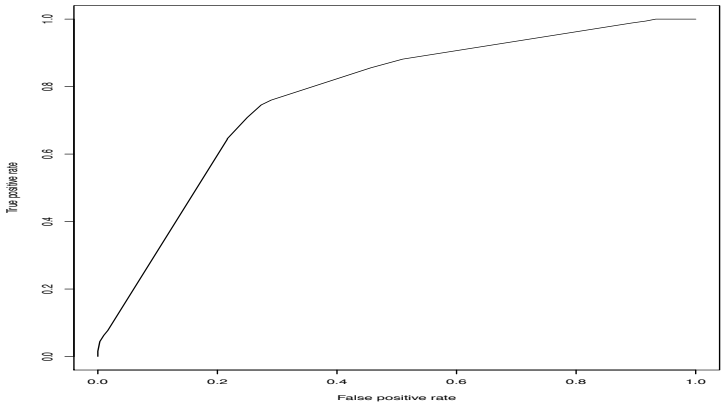
The maximum precision obtained is 0.81,with 0.7433 accuracy

CASE:5 Training=80% DATASET,TESTING=20%DATASET

Control parameters	Model 1		Model 2		MODEL 4	
Bands	0		0		0	
CF	0.25		0.25		0.25	
minCases	3		3		1	
noGlobalPruning	TRUE		TRUE		TRUE	
Winnow	FALSE		FALSE		FALSE	
fuzzyThreshold	FALSE		FALSE		FALSE	
earlyStopping	TRUE		TRUE		TRUE	
Subset	TRUE		FALSE		FALSE	
ACCURACY_TEST	0.78		0.76		0.77	
ACCURACY_TRAIN	0.74		0.73		0.76	
ACCURACY	0.75		0.745		0.76	
PRECISION(0:1)	0.78	0.74	0.78	0.74	0.61	0.80
RECALL(0:1)	0.24	0.97	0.24	0.97	0.51	0.86
F1(0:1)	0.37	0.84	0.37	0.84	0.56	0.83

The maximum precision obtained is 0.80,with 0.76 accuracy

Inference: The maximum precision obtained is 0.82 with 0.73 accuracy. The model which got the max precision and accuracy was with the 60-30 split.



ROC CURVE for the 60-30 split.

SPLIT(TRAIN-TEST)	AUC	Precision	Accuracy
50-50	0.7416167	0.7387387	0.726
60-40	0.7572571	0.8218182	0.73
70-30	0.7623714	0.8173077	0.743
80-20	0.7709548	0.7891156	0.735

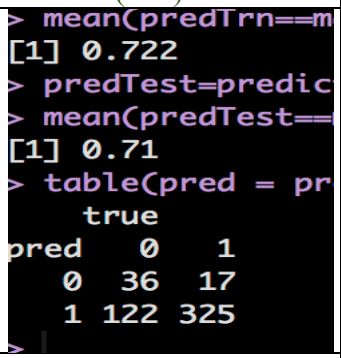
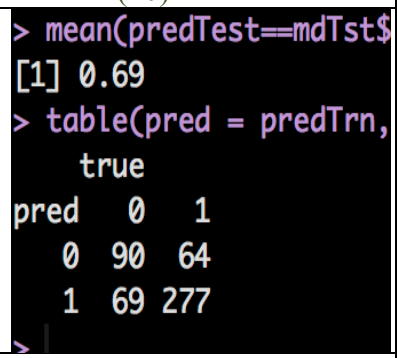
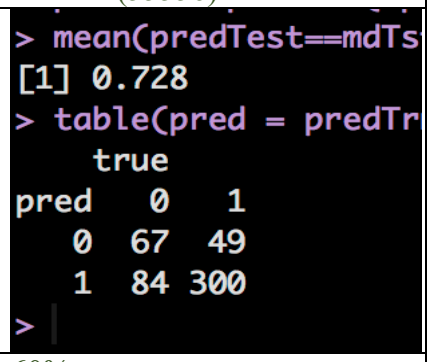
- As we go for higher percentage of data in the testing set, the AUC is increasing meaning that the model is getting over-fitted.
- We obtained the highest precision for our 60-30 split and its accuracy was also good.

(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?

Answer:2-c)

Comparing the above models:

Okay so for this we compared various set.seed values over DT using both C5 and rpart
For the last model we made using rpart

	Set.seed(124)	Set.seed(10)	Set.seed(55556)
			
GOOD	65%	55.4%	60%
BAD	7.2%	18%	13.4%

So, we see that by putting random values of set.seed, our results are completely different and it the value of set.seed has no relation with the TRUE POSITIVE or TRUE NEGATIVES of the cases.

(d) Which variables are important for separating 'Good' from 'Bad' credit? Determine variable importance from the different 'best' trees. Are there similarities, differences?

So while using the algorithm rpart, we got the variable importance as

```
Variable importance
       CHK_ACCT         HISTORY         SAV_ACCT         DURATION           AMOUNT PRESENT_RESIDENT
             49              15               13               10                4                3
 PROP_UNKN_NONE      EMPLOYMENT     NUM_CREDITS      REAL_ESTATE              JOB
              2               1                1                1                1
```

CHK_ACCT,HISTORY,SAV_ACCT,DURATION and AMOUNT are the most important factors for constructing our model with respect to reponse variable.

While using the C5.0 algo, we got the variable importance as:

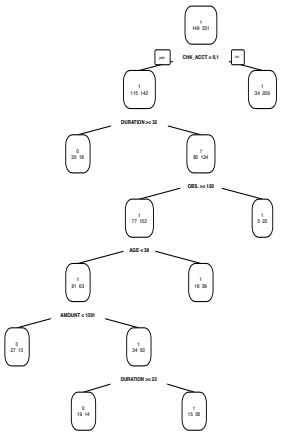
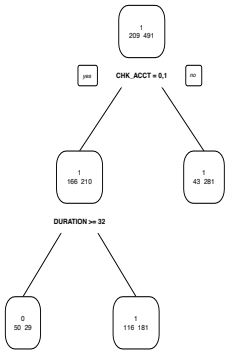
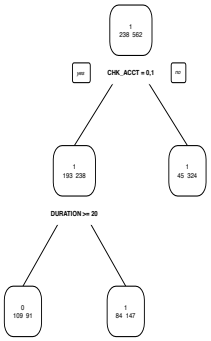
```
> c50Tree = C5.0 (RESPONSE~., data = datas, rules = FALSE,control = C5.0Control (subset = TRUE,bands = 0, winnow=FALSE,
CF=0.25, minCases=3, noGlobalPruning=TRUE, sample=0,fuzzyThreshold = FALSE,seed = sample.int(4096, size = 1) -
+                                          1L, earlyStopping = TRUE, label = "RESPONSE"))
> C5imp(c50Tree)
                 Overall
CHK_ACCT           100.0
GUARANTOR           54.3
SAV_ACCT            50.0
OTHER_INSTALL       49.0
DURATION            44.3
FOREIGN             35.6
HISTORY             34.5
EDUCATION           30.2
RETRAINING          29.3
USED_CAR            24.3
INSTALL_RATE        19.5
REAL_ESTATE         11.1
NEW_CAR             10.8
MALE_MAR_or_WID     10.1
CO.APPLICANT         9.8
MALE_DIV             8.0
AMOUNT               7.1
PRESENT_RESIDENT     6.7
TELEPHONE            5.5
NUM_CREDITS          4.5
OBS.                 3.5
EMPLOYMENT           2.9
RADIO.TV             2.2
```

CHK_ACCT,GUARANTOR,SAV_ACCT,OTHER_INSTALL,DURATION,FOREIGN

So, for both the algorithm, there is a difference in the important variables used for predicting the response variable.

(d) 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 what does this indicate?
Is there any specific model you would prefer for implementation?

	50-50	70-30	80-20
Testing	0.774	0.731	0.725
Training	0.722	0.73	0.73
Accuracy	0.722	0.73	0.73
Precision_Good CASE	0.7900552	0.7370370	0.7837838

			
Comments	The top variable is CHK_ACCT,DURATION. Also, it has the least accuracy as compared to the rest two but the highest precision.	The top variable is CHK_ACCT,DURATION. Also, it has the least precision as compared to the rest two but the highest precision.	The top variable is CHK_ACCT,DURATION. Also, it has the highest accuracy as compared to the rest two and second highest precision.

We see that the priority variables(in the upper part of the tree) is same across every split. So, variable importance for determining a factor isn't affected by splitting the data.

3. 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
This information can be used to determine the following costs for misclassification:
Predicted
Actual
Good Bad
Good 0 100DM
Bad 500DM 0

(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 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?

(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.

(c) Use misclassification costs to develop 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).

USING RPART

Calculating the misclassification cost for this model. Going with the 60-40 model.

Threshold value	Misclassification number		Accuracy		Precision(Good Credits)	
	TRN[1,2][2,1]	TEST[1,2][2,1]	TRN	TEST	TRN	TEST
0.5	23,141	16,89	0.726667	0.7375	0.7354597	0.7513966
0.8	176,39	127,21	0.641667	0.63	0.859122	0.8826816

USING C5.0
Calculating the misclassification cost for this model. Going with the 60-40 model.

Threshold value	Misclassification cost		Accuracy		Precision(Good Credits)	
	TRN[1,2][2,1]	TEST[1,2][2,1]	TRN	TEST	TRN	TEST
0.5	83,64	59,49	0.755	0.7375	0.73	0.8218182
0.8	104,39	78,35	0.745	0.7175	0.8638889	0.8553719

th = costMatrix[2,1]/(costMatrix[2,1] + costMatrix[1,2])

CALCULATIONS:
1)FOR RPART:

Threshold value	Misclassification number		COST		th
	TRN[1,2][2,1]	TEST[1,2][2,1]	TRN	TEST	TRAINING
0.5	23,141	16,89	(2300-70500)=-68200	(-44500+1600)=-42900	141*(-500)/(2300-70500)=1.033
0.8	176,39	127,21	(17600-19500)=-1900	(12700-10500)=2200	39*(-500)/(17600-19500) =10.26

FOR[1,2]=+100DM,FOR [2,1]=-500DM

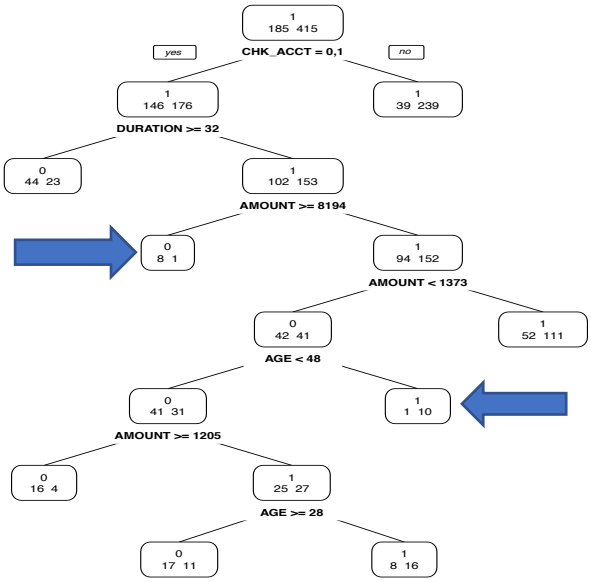
FOR C5.0:

Threshold value	Misclassification number		COST		TH
	TRN[1,2][2,1]	TEST[1,2][2,1]	TRN	TEST	TRAINING
0.5	83,64	59,49	(8300-32000)=-23700	(-24500+5900)=-18600	64*(-500)/(8300-32000)=1.35
0.8	104,39	78,35	(10400-19500)=-9100	(7800-17500)=-9700	39*(-500)/(10400-19500)=2.14

#**INFERENCE:**

- Talking about precision,accuracy and AUC values, we have better models for cthresh 0.8.
- We get more precision and accuracy with C5.0 as compared to rpart.
- We have more theoretical thresholsvalue for cthresh=0.8, using rpart.

4. 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?



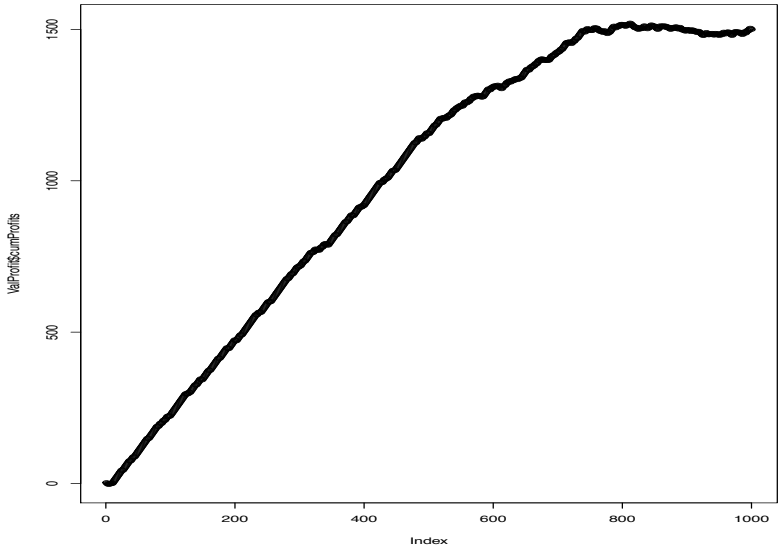
TREE DEPTH: 6
NUMBER OF NODES: 179
IMPORTANT VARIABLES: AMOUNT+CHK_ACCT+DURATION+AGE
RELATIVELY PURE NODE:

```
Node number 45: 11 observations
  predicted class=1  expected loss=0.09090909  P(node) =0.01833333
    class counts:     1    10
   probabilities: 0.091 0.909

Node number 10: 9 observations
  predicted class=0  expected loss=0.1111111  P(node) =0.015
    class counts:     8     1
   probabilities: 0.889 0.111
```

1)Node 45: If age>48,probability of him/her being “good credit scorer is” is 0.909 and “bad” is 0.091
2)Node 10: If amount<=8194,probability of him being a “good credit scorer” is 0.111 and “bad” is 0.889

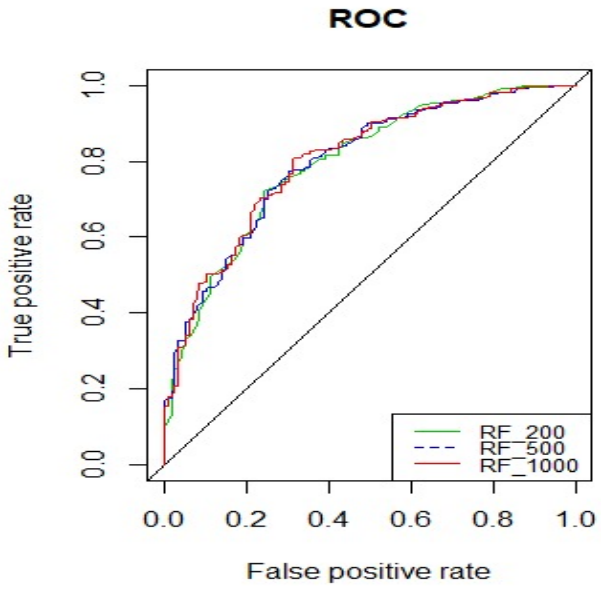
5. The predicted probabilities can be used to determine how the model may be implemented. We can sort the data from high to low on predicted probability of “good” credit risk. Then, going down the cases from high to low probabilities, one may be able to determine an appropriate cutoff probability – values 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.
provide maintenance



#INFERENCE: We obtain a maximum profit of 1521.00 and ValScore of 0.3396

RANDOM FOREST

Calculated Random forest using 200,500,1000



From the graph we see that the performance for all three combinations is almost the same.

AUC for 200	0.77
AUC for 500	0.781
AUC for 1000	0.785

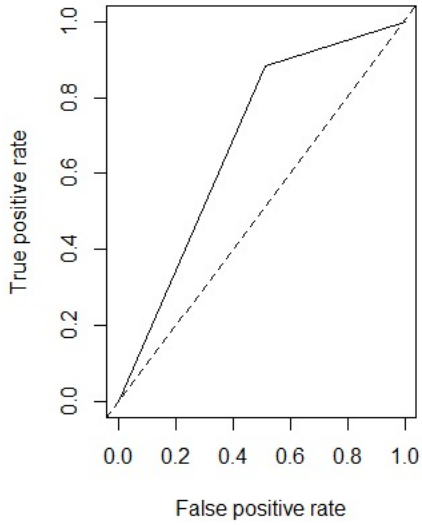
As observed , the AUC for 1000 trees is slightly more than that of 500 trees by approximately 0.4% only

Accuracy for 200	0.78
Accuracy for 500	0.77
Accuracy for 1000	0.775

since accuracy of Random forest model with 1000 trees is highest, we choose the Random Forest model with 1000 trees.

BOOSTED TREE MODEL

Accuracy on Training Data: 100% Accuracy on test data: 77% AUC: 0.685



#INFERENCE: With the accuracy of 78% using Random Forest Model as compared to 77% accuracy of Boosting Tree and higher area under the curve for Random Forest, it can be inferred that random tree perform better than boosting trees .Also, there is a less chance of over-fit, since Random forest uses different variables while building a new tree as opposed to boosting trees where the same data is used over and over again to build Decision Trees.