IDS 572 - Data Mining for Business

Assignment 1: Decision Tree analysis

Team Members:

Mrunal Ghorpade (677441117)

Neha Chimata (655196210)

Tanvi Sethi (672107527)

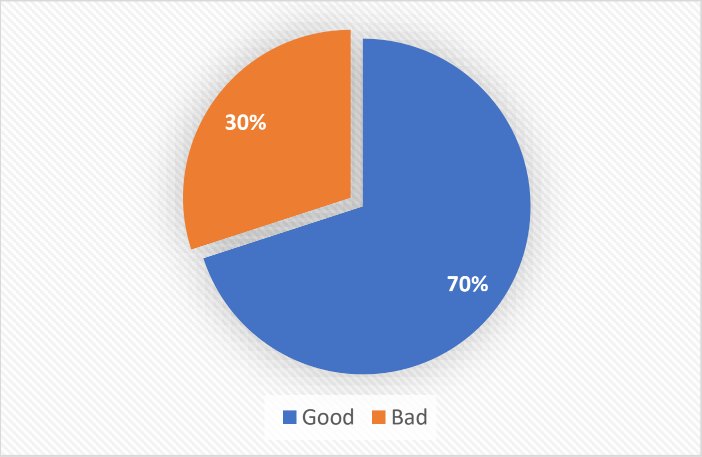
INTRODUCTION:

The German Credit dataset has data on 1000 past credit applicants, described by 30 variables. Each applicant is also rated as "Good" or "Bad" credit (encoded as 1 and 0 respectively in the Response variable). The GermanCredit.xls file contains the variable descriptions and the data. New applicants for credit can be evaluated on these 30 variables. We would like to develop a credit scoring rule that can be used to help determine whether a new applicant presents a good or bad credit risk. Here, we will attempt to obtain a decision tree based model to determine if new applicants present a good or bad credit risk. The original data has been transformed to ease analysis in this initial assignment. It is informative to compare the original and transformed data descriptions to see how different variables have been transformed. The original variables are given below.

ASSIGNMENT QUESTIONS:

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?

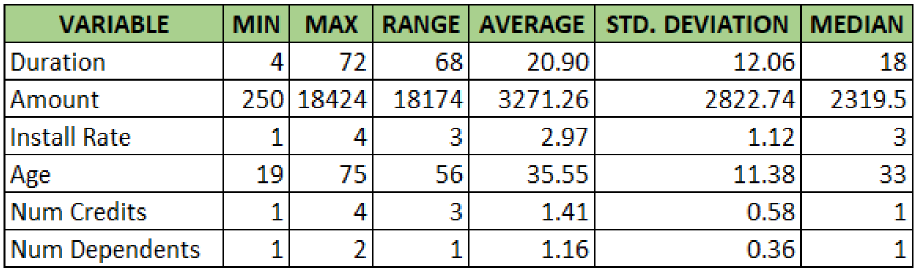
The proportion of "Good" to "Bad" cases is 7:3. Among the 1000 observations, 70% depict good credit cases and the remaining 30% depict bad cases and a graphical representation of the same is given below:



Missing Values:

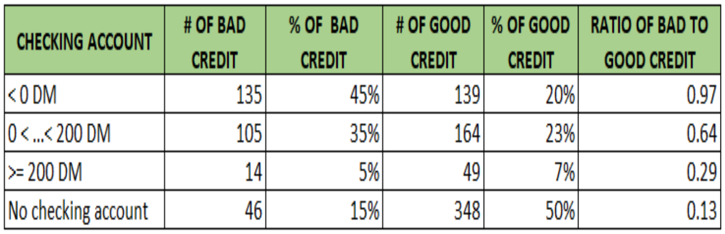
Yes, 7 columns out of 31 had missing values and are treated with the help of "Replace Missing Values Operator". Six of these columns (NEW_CAR, USED_CAR, RETRAINING, EDUCATION, FURNITURE, RADIO/TV) contain binamial values i.e 1 for true and 0 for false. Hence the missing values in these columns are replaced with 0 as data in these columns have been updated when the corresponding condition is satisfied. The other column which had missing values is AGE and the missing values in this column are replaced by the avaerage age.

Description of Predictor Variables:



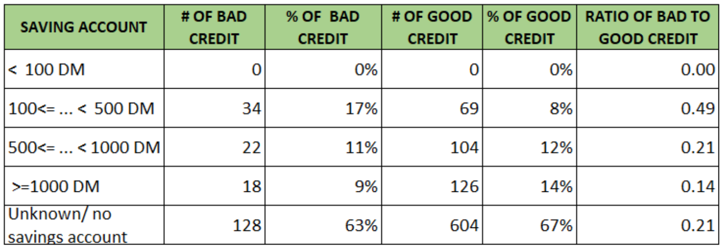
VARIABLE	MIN	MAX	RANGE	AVERAGE	STD. DEVIATION	MEDIAN
Duration	4	72	68	20.90	12.06	18
Amount	250	18424	18174	3271.26	2822.74	2319.5
Install Rate	1	4	3	2.97	1.12	3
Age	19	75	56	35.55	11.38	33
Num Credits	1	4	3	1.41	0.58	1
Num Dependents	1	2	1	1.16	0.36	1

Categorical Variables:



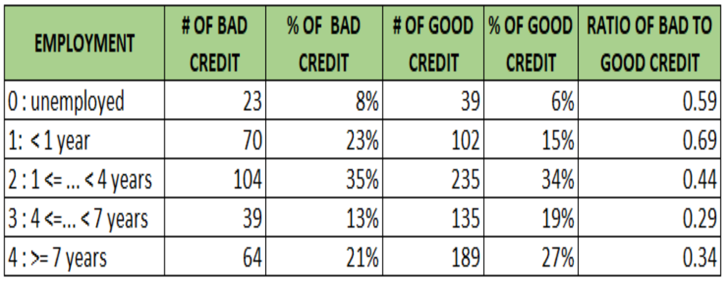
CHECKING ACCOUNT	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
< 0 DM	135	45%	139	20%	0.97
0 < ...< 200 DM	105	35%	164	23%	0.64
>= 200 DM	14	5%	49	7%	0.29
No checking account	46	15%	348	50%	0.13

- Checking account status of Type 0 and 1 have more people with bad credit score.
- Checking account status of Type 3 have highest percentage of people with good credit score.



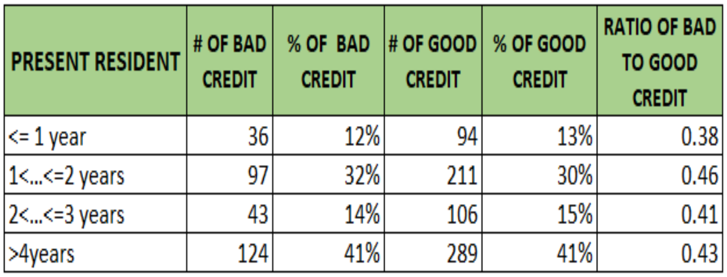
SAVING ACCOUNT	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
< 100 DM	0	0%	0	0%	0.00
100<= ... < 500 DM	34	17%	69	8%	0.49
500<= ... < 1000 DM	22	11%	104	12%	0.21
>=1000 DM	18	9%	126	14%	0.14
Unknown/ no savings account	128	63%	604	67%	0.21

- Savings account status in "Category 1" have highest ratio of bad to good credit score than any other category.



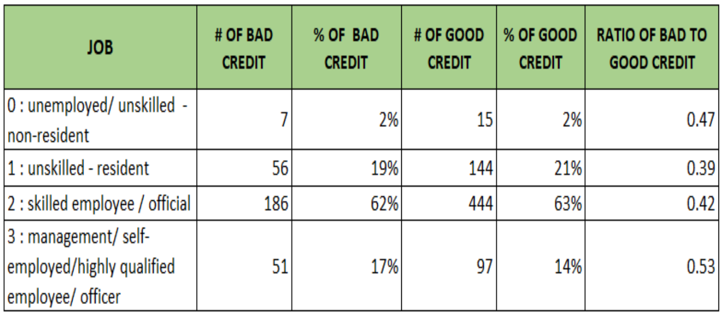
EMPLOYMENT	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
0 : unemployed	23	8%	39	6%	0.59
1: < 1 year	70	23%	102	15%	0.69
2 : 1 <= ... < 4 years	104	35%	235	34%	0.44
3 : 4 <=... < 7 years	39	13%	135	19%	0.29
4 : >= 7 years	64	21%	189	27%	0.34

- As expected, the ratio of bad tp good credit has decreased as the years of experience is increased. However, the unemployed have comparitively less ratio of bad to good credit ratio than the people with less than 1 years of experience.



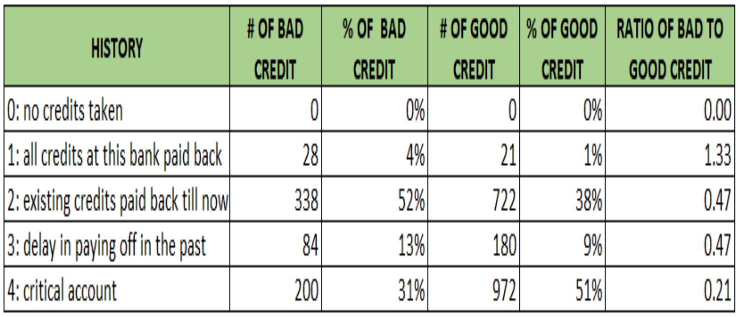
PRESENT RESIDENT	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
<= 1 year	36	12%	94	13%	0.38
1<...<=2 years	97	32%	211	30%	0.46
2<...<=3 years	43	14%	106	15%	0.41
>4years	124	41%	289	41%	0.43

- People who have been resident from past 4 years have bad credit history. But we cannot any relationship of credit rate from the present resident variable as even the people who are resident for 1- 2 years also have a bad credit history.



JOB	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
0 : unemployed/ unskilled - non-resident	7	2%	15	2%	0.47
1 : unskilled - resident	56	19%	144	21%	0.39
2 : skilled employee / official	186	62%	444	63%	0.42
3 : management/ self-employed/highly qualified employee/ officer	51	17%	97	14%	0.53

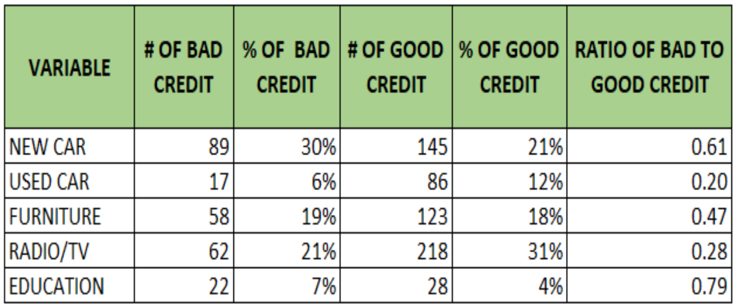
- In contratary to our belives Category 2 which consists skilled employee/officials are the once with maximum % of bad credit history, however its Bad to good credit ratio is not greater as the % of good credit score is similar to the % of bad credit score.



HISTORY	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
0: no credits taken	0	0%	0	0%	0.00
1: all credits at this bank paid back	28	4%	21	1%	1.33
2: existing credits paid back till now	338	52%	722	38%	0.47
3: delay in paying off in the past	84	13%	180	9%	0.47
4: critical account	200	31%	972	51%	0.21

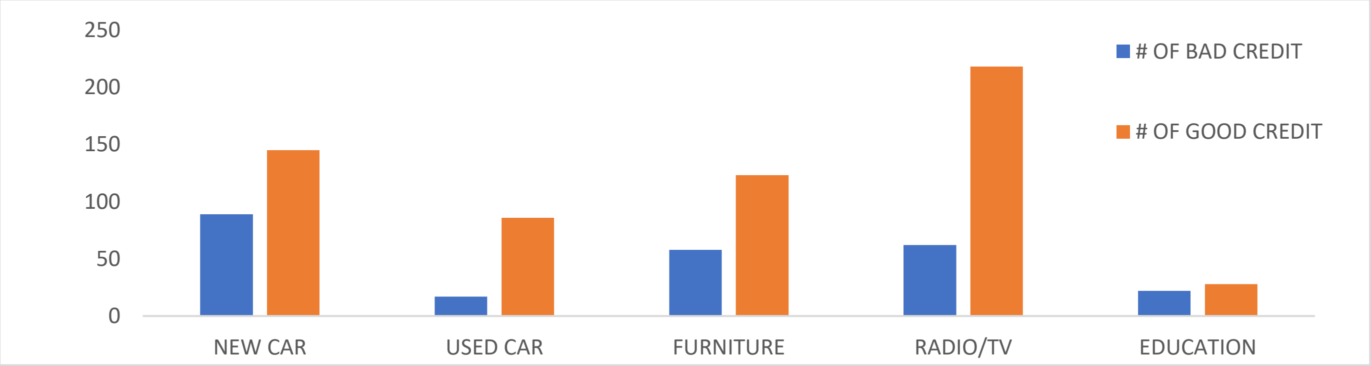
- Category 1 has the largest ratio of Bad to good credit which is a surprise as it correspends to the people who's all credits at this bank are paid back duly. Similarly, we can see that % of bad credit is maximum in Category 2 even it consists the people who have paid dues which are till date.

Binomial Variables:



VARIABLE	# OF BAD CREDIT	% OF BAD CREDIT	# OF GOOD CREDIT	% OF GOOD CREDIT	RATIO OF BAD TO GOOD CREDIT
NEW CAR	89	30%	145	21%	0.61
USED CAR	17	6%	86	12%	0.20
FURNITURE	58	19%	123	18%	0.47
RADIO/TV	62	21%	218	31%	0.28
EDUCATION	22	7%	28	4%	0.79

- The people who have taken loan for Radio/TV and New cars have higher good credit than the others.



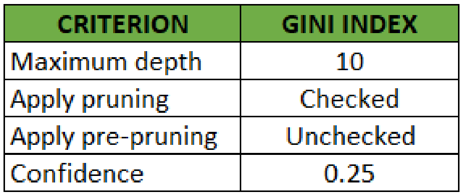
Variables, Most relevant to the outcome of interest:

CHK_ACCT – Tells us the capability of the customer in repaying the credit taken from the bank.
SAV_ACCT - Gives us the information about the savings of a customer.

AGE - Indicates the remaining customer's employement period.
HISTORY – Gives us information about the punctuality of the customer in repaying credits taken.
AMOUNT - Indicates how much credit a customer requested for.
EMPLOYMENT – Provides us the about the Employment status of the customer
OWN_RES - Indicates whether the customer owns a residence or not (asset)

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

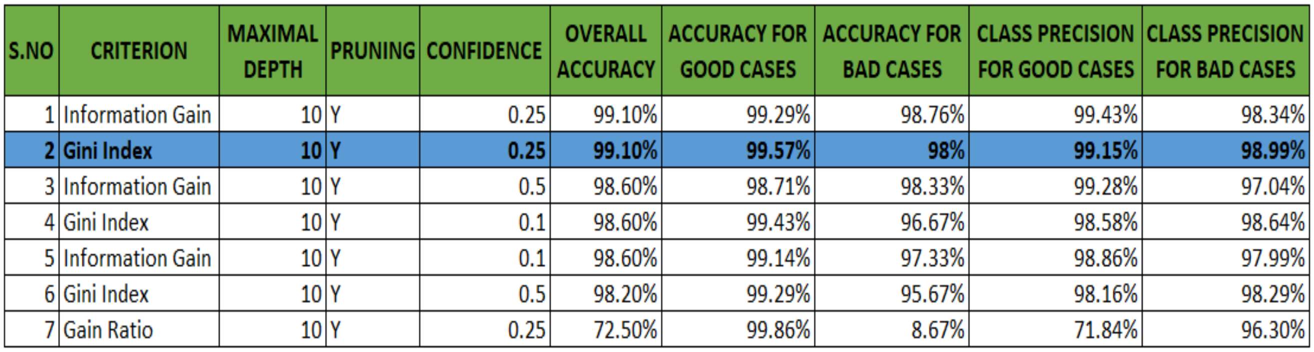
Decision tree was built on full data and the node parameters that were used to get a good model are as follows:



CRITERION	GINI INDEX
Maximum depth	10
Apply pruning	Checked
Apply pre-pruning	Unchecked
Confidence	0.25

The above parameters were chosen as that combination produced the maximum level of accuracy.

Different other combination of parameters which were tried before ariving at the above-mentioned node parameters and they are as follows:



S.NO	CRITERION	MAXIMAL DEPTH	PRUNING	CONFIDENCE	OVERALL ACCURACY	ACCURACY FOR GOOD CASES	ACCURACY FOR BAD CASES	CLASS PRECISION FOR GOOD CASES	CLASS PRECISION FOR BAD CASES
1	Information Gain	10	Y	0.25	99.10%	99.29%	98.76%	99.43%	98.34%
2	**Gini Index**	**10**	**Y**	**0.25**	**99.10%**	**99.57%**	**98%**	**99.15%**	**98.99%**
3	Information Gain	10	Y	0.5	98.60%	98.71%	98.33%	99.28%	97.04%
4	Gini Index	10	Y	0.1	98.60%	99.43%	96.67%	98.58%	98.64%
5	Information Gain	10	Y	0.1	98.60%	99.14%	97.33%	98.86%	97.99%
6	Gini Index	10	Y	0.5	98.20%	99.29%	95.67%	98.16%	98.29%
7	Gain Ratio	10	Y	0.25	72.50%	99.86%	8.67%	71.84%	96.30%

(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)?

Important Variables to differentiate good cases from bad cases:

According to the decision trees that were created using different criterions, the best decision trees were based on the following variables:

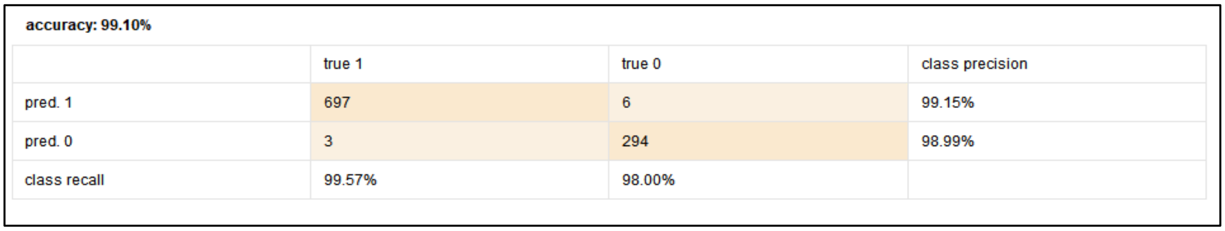
- Checking account status
- Credit Amount

- Application has other installment per credit.
- Credit History
- Present Residence
- Employment Since

These match with the expected variables that are most relevant to the outcome of interest from the previous question.

(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.

The Overall Accuracy of the model is 99.10%. The results obtained after the model was executed are as follows:



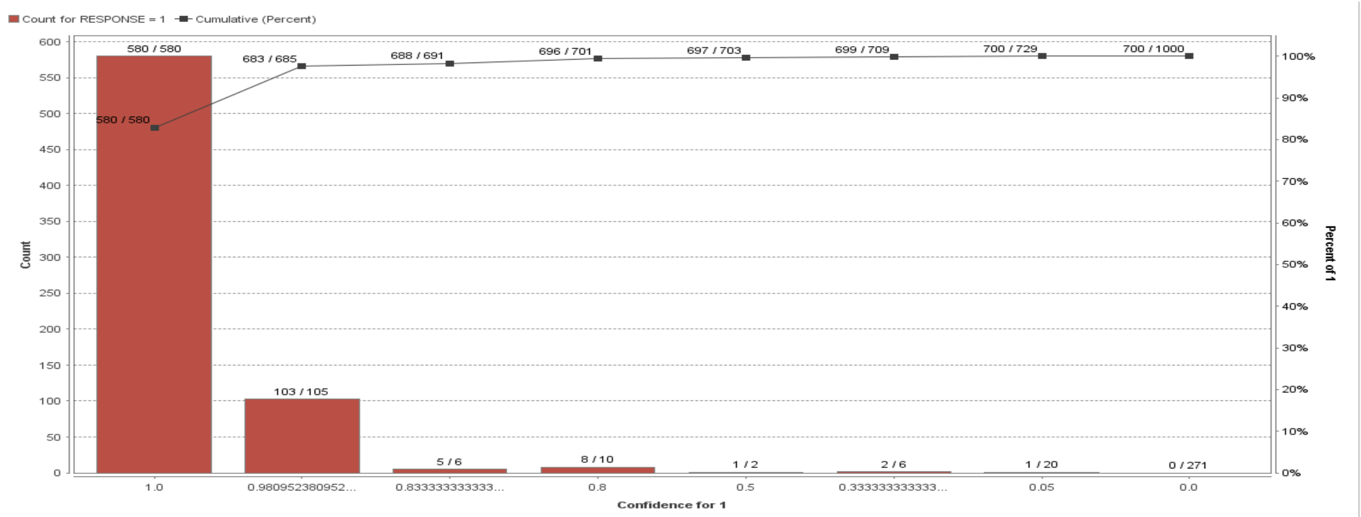
accuracy: 99.10%

	true 1	true 0	class precision
pred. 1	697	6	99.15%
pred. 0	3	294	98.99%
class recall	99.57%	98.00%	

The accuracy of the good cases (true 1) is 99.57% and bad cases (true 0) is 98%.

Lift Chart for Good Cases (True 1):

From the lift chart below, it is clearly seen that 58% of the people (580/1000) are predicted to have good credit scores with 100% confidence.



Robust:

This is an OVERFIT model and not a reliable(robust) model as training data contained the entire 100% data of the dataset. It might or might not work when tested on unobserved data.

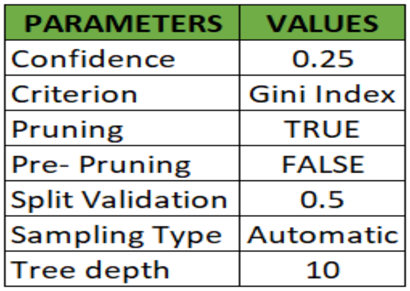
2.2 We next consider developing a model for prediction. For this, we should divide the data into Training and Validation sets.

(a) Consider a partition of the data into 50% for Training and 50% for Test. 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. Is the model reliable (why or why not)?

In developing the models above, change decision tree options as you find reasonable (for example, 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 2 parameters affect performance. Also, does pruning give a better model – please explain why or why not? Which decision tree parameter values do you find to be useful for developing a good model. Are they the same for different training-test partitions?

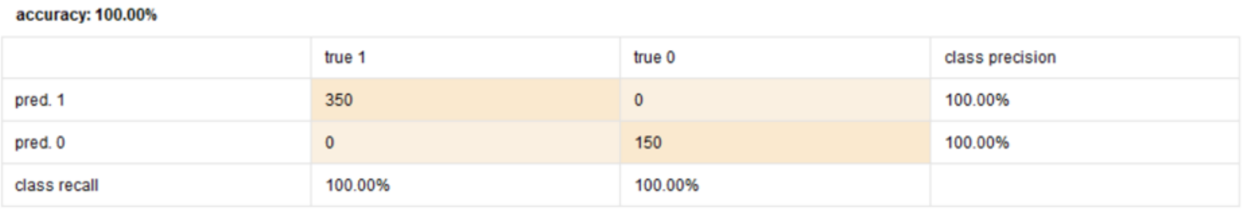
The training and validation datasets were obtained by splitting the datset using a 50:50 split ratio. The model returned best accuracy when the values for parameters *Criteria*, *Maximal depth*, *Apply Pruning*, *Confidence*, *apply pre-pruning and sampling* were as below.



PARAMETERS	VALUES
Confidence	0.25
Criterion	Gini Index
Pruning	TRUE
Pre- Pruning	FALSE
Split Validation	0.5
Sampling Type	Automatic
Tree depth	10

The tables below depict the results (accuracy of the model) obtained when a **50%-50%** split of training and validation data and the values of the parameters as mentioned above were considered.

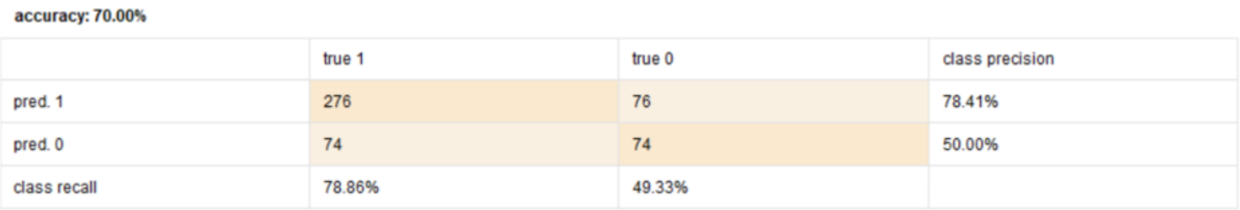
Training Data:



accuracy: 100.00%

	true 1	true 0	class precision
pred. 1	350	0	100.00%
pred. 0	0	150	100.00%
class recall	100.00%	100.00%	

Test Data:

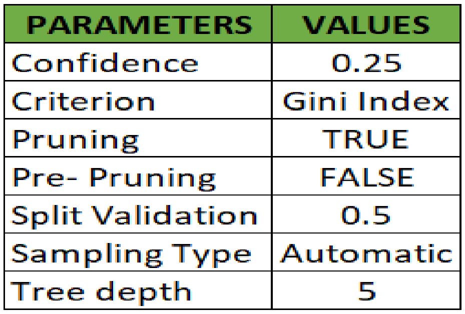


accuracy: 70.00%

	true 1	true 0	class precision
pred. 1	276	76	78.41%
pred. 0	74	74	50.00%
class recall	78.86%	49.33%	

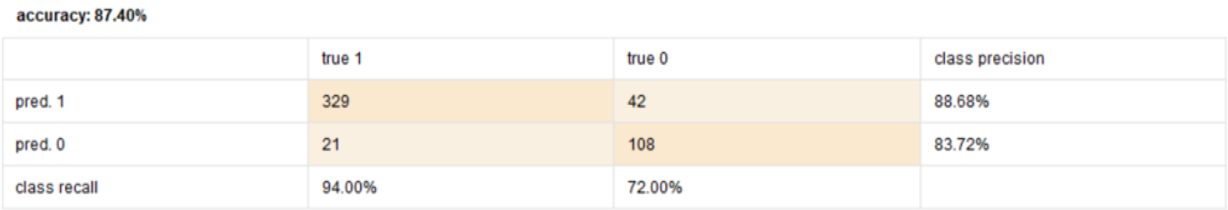
The Accuracy of test data decreases drastically while the accuracy of the training data is 100%, which shows that the model is overfitting. Hence it can be concluded that, this model is not reliable and deploying it wouldn't be beneficial.

This problem of overfitting can be decreased by using more data for training or by **pruning**. By altering the valus of certain parameters i.e by **decreasing the tree depth**, we are increasing the generalization. This in turn increases the accuracy on the test data by decreasing the problem of overfitting. The new set of parameters, used to yield best results and the results are as follows:



PARAMETERS	VALUES
Confidence	0.25
Criterion	Gini Index
Pruning	TRUE
Pre- Pruning	FALSE
Split Validation	0.5
Sampling Type	Automatic
Tree depth	5

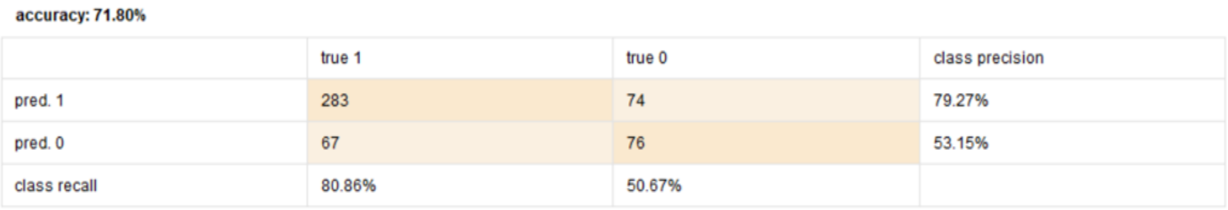
Training data:



accuracy: 87.40%

	true 1	true 0	class precision
pred. 1	329	42	88.68%
pred. 0	21	108	83.72%
class recall	94.00%	72.00%	

Test data:

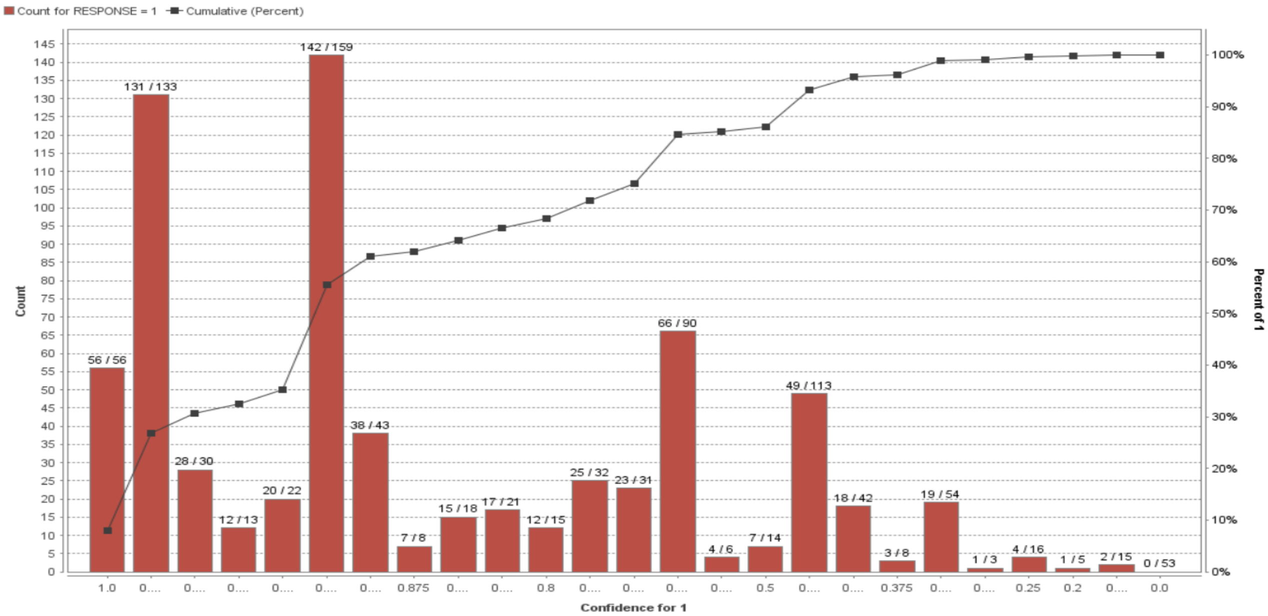


accuracy: 71.80%

	true 1	true 0	class precision
pred. 1	283	74	79.27%
pred. 0	67	76	53.15%
class recall	80.86%	50.67%	

Lift Chart for Good Cases (True 1):

From the lift chart below, it is clearly seen that 98% of the people (within the 95% confidence interval i.e 227/232) are predicted to have good credit scores.



The problem of overfitting was reduced on enabling **pre-pruning** as well. However, it was observed that by enabling the pre-pruning option, the decision tree was being trimmed even before it could grow. Though it increased the accuracy on the test data to **72.20%**, it decreased the accuracy on the training data to **75.20%**.

Parameters useful for developing a good model:

- Pruning and Pre-pruning: Solves the problem of Overfitting
- Criterion: Helps to decide the criteria on which the decision tree should be build.
- Tree Depth: Gives the length of the longest path from a root to a leaf.

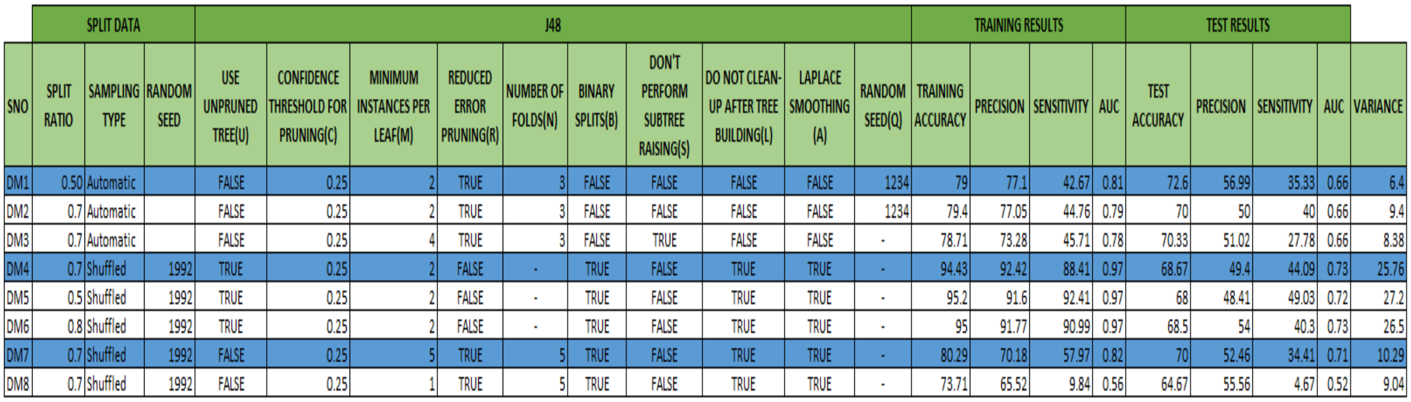
The same parameter setting was used for both the training and testing data sets inorder to uniformity in interpretation of the results of the model.

Performance measures:

The performance measure used is Accuracy as it helps us understanding the correctness of the predicted value i.e it helps in identifying the difference between the resulted value and the true value. And here, since we are focussing on developing a better model, accuracy of the model is given high priority.

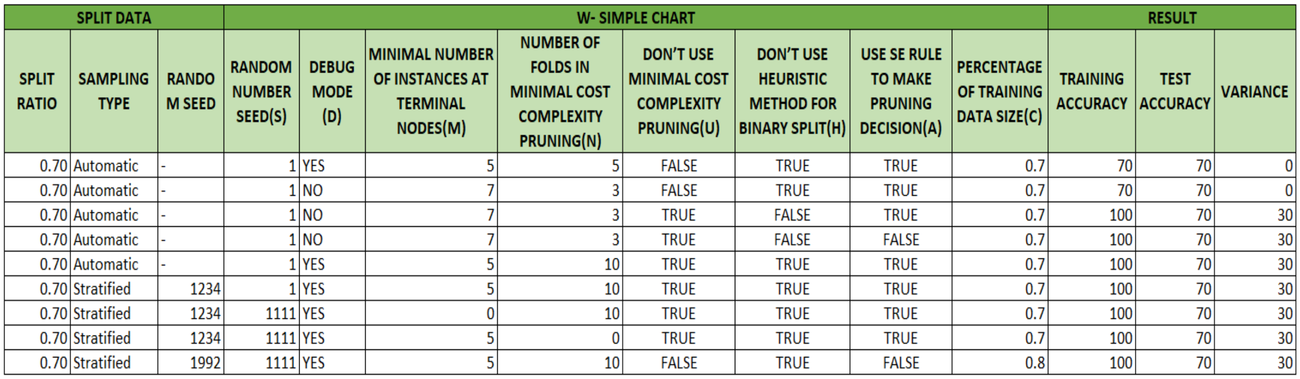
2.2 (b) Consider two other type of decision tree operators – for example, CART, J48 – experiment with the parameters till you get a 'good' model. Summarize the parameters and performance you obtain (using tables can be helpful for this) from the different decision tree operators. Does performance differ across different types of decision tree learners? Compare modelsusing accuracy, sensitivity, precision, recall, etc (as you find reasonable – you answer to Questions (a) above should clarify which performance measures you use and why). Alsocompare performance on lift, ROC curves and AUC.
How do the models obtained from these decision tree learners differ?
We used J48 and W-Simple Cart to build tree models using different parameters.

J48 decision operator:
Various parameters viz. number of folds for reduced error pruning, use of unpruned tree, minimum number of instances per leaf and use of binary splits have major effects on the model accuracy. Based on these parameters, we built the following models:



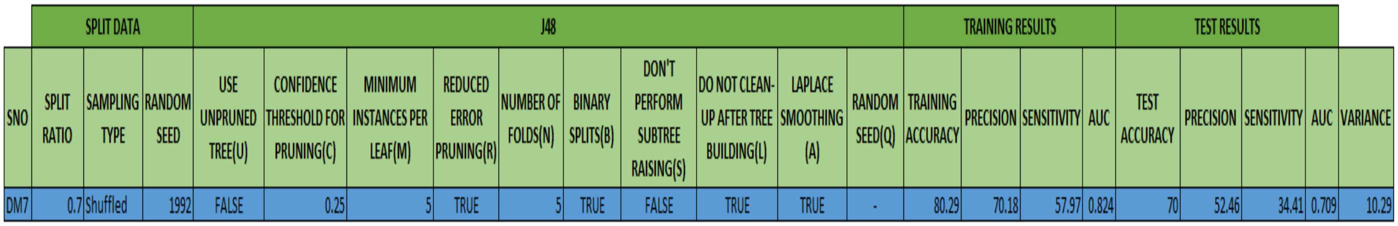
	SPLIT DATA			J48										TRAINING RESULTS				TEST RESULTS				
SNO	SPLIT RATIO	SAMPLING TYPE	RANDOM SEED	USE UNPRUNED TREE(U)	CONFIDENCE THRESHOLD FOR PRUNING(C)	MINIMUM INSTANCES PER LEAF(M)	REDUCED ERROR PRUNING(R)	NUMBER OF FOLDS(N)	BINARY SPLITS(B)	DON'T PERFORM SUBTREE RAISING(S)	DO NOT CLEAN-UP AFTER TREE BUILDING(L)	LAPLACE SMOOTHING (A)	RANDOM SEED(Q)	TRAINING ACCURACY	PRECISION	SENSITIVITY	AUC	TEST ACCURACY	PRECISION	SENSITIVITY	AUC	VARIANCE
DM1	0.50	Automatic		FALSE	0.25	2	TRUE	3	FALSE	FALSE	FALSE	FALSE	1234	79	77.1	42.67	0.81	72.6	56.99	35.33	0.66	6.4
DM2	0.7	Automatic		FALSE	0.25	2	TRUE	3	FALSE	FALSE	FALSE	FALSE	1234	79.4	77.05	44.76	0.79	70	50	40	0.66	9.4
DM3	0.7	Automatic		FALSE	0.25	4	TRUE	3	FALSE	TRUE	FALSE	FALSE	-	78.71	73.28	45.71	0.78	70.33	51.02	27.78	0.66	8.38
DM4	0.7	Shuffled	1992	TRUE	0.25	2	FALSE	-	TRUE	FALSE	TRUE	TRUE	-	94.43	92.42	88.41	0.97	68.67	49.4	44.09	0.73	25.76
DM5	0.5	Shuffled	1992	TRUE	0.25	2	FALSE	-	TRUE	FALSE	TRUE	TRUE	-	95.2	91.6	92.41	0.97	68	48.41	49.03	0.72	27.2
DM6	0.8	Shuffled	1992	TRUE	0.25	2	FALSE	-	TRUE	FALSE	TRUE	TRUE	-	95	91.77	90.99	0.97	68.5	54	40.3	0.73	26.5
DM7	0.7	Shuffled	1992	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	-	80.29	70.18	57.97	0.82	70	52.46	34.41	0.71	10.29
DM8	0.7	Shuffled	1992	FALSE	0.25	1	TRUE	5	TRUE	FALSE	TRUE	TRUE	-	73.71	65.52	9.84	0.56	64.67	55.56	4.67	0.52	9.04

W-Simple Cart:
Whereas in case of W-Simple Cart, the accuracy of the model changes with the effect of using cost complexity pruning and pruning decision. Almost all the models gave the same accuracy 100: 70, thus giving an overfit model.



SPLIT DATA			W- SIMPLE CHART								RESULT		
SPLIT RATIO	SAMPLING TYPE	RANDO M SEED	RANDOM NUMBER SEED(S)	DEBUG MODE (D)	MINIMAL NUMBER OF INSTANCES AT TERMINAL NODES(M)	NUMBER OF FOLDS IN MINIMAL COST COMPLEXITY PRUNING(N)	DON'T USE MINIMAL COST COMPLEXITY PRUNING(U)	DON'T USE HEURISTIC METHOD FOR BINARY SPLIT(H)	USE SE RULE TO MAKE PRUNING DECISION(A)	PERCENTAGE OF TRAINING DATA SIZE(C)	TRAINING ACCURACY	TEST ACCURACY	VARIANCE
0.70	Automatic	-	1	YES	5	5	FALSE	TRUE	TRUE	0.7	70	70	0
0.70	Automatic	-	1	NO	7	3	FALSE	TRUE	TRUE	0.7	70	70	0
0.70	Automatic	-	1	NO	7	3	TRUE	FALSE	TRUE	0.7	100	70	30
0.70	Automatic	-	1	NO	7	3	TRUE	FALSE	FALSE	0.7	100	70	30
0.70	Automatic	-	1	YES	5	10	TRUE	TRUE	TRUE	0.7	100	70	30
0.70	Stratified	1234	1	YES	5	10	TRUE	TRUE	TRUE	0.7	100	70	30
0.70	Stratified	1234	1111	YES	0	10	TRUE	TRUE	TRUE	0.7	100	70	30
0.70	Stratified	1234	1111	YES	5	0	TRUE	TRUE	TRUE	0.7	100	70	30
0.70	Stratified	1992	1111	YES	5	10	FALSE	TRUE	FALSE	0.8	100	70	30

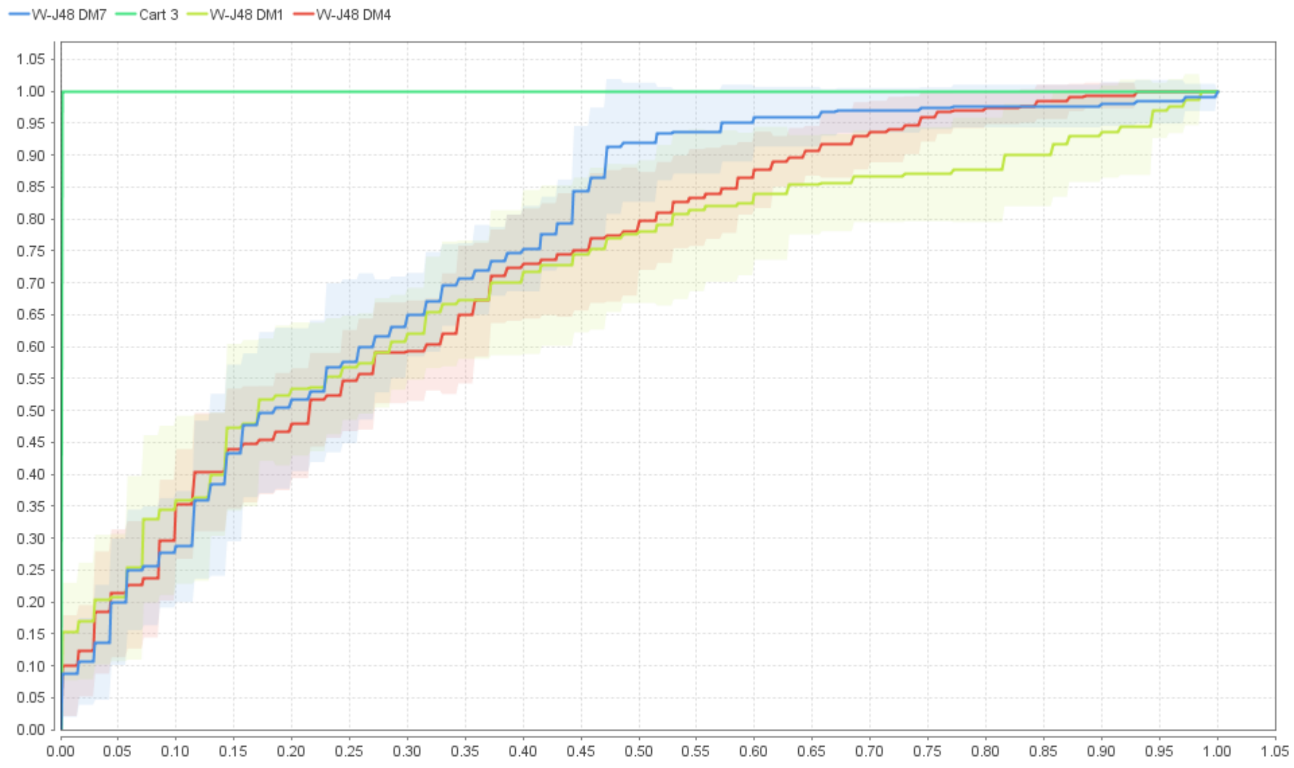
Based on the accuracy, sensitivity, precision, recall observed in the tables, the following model gave the best decision tree:



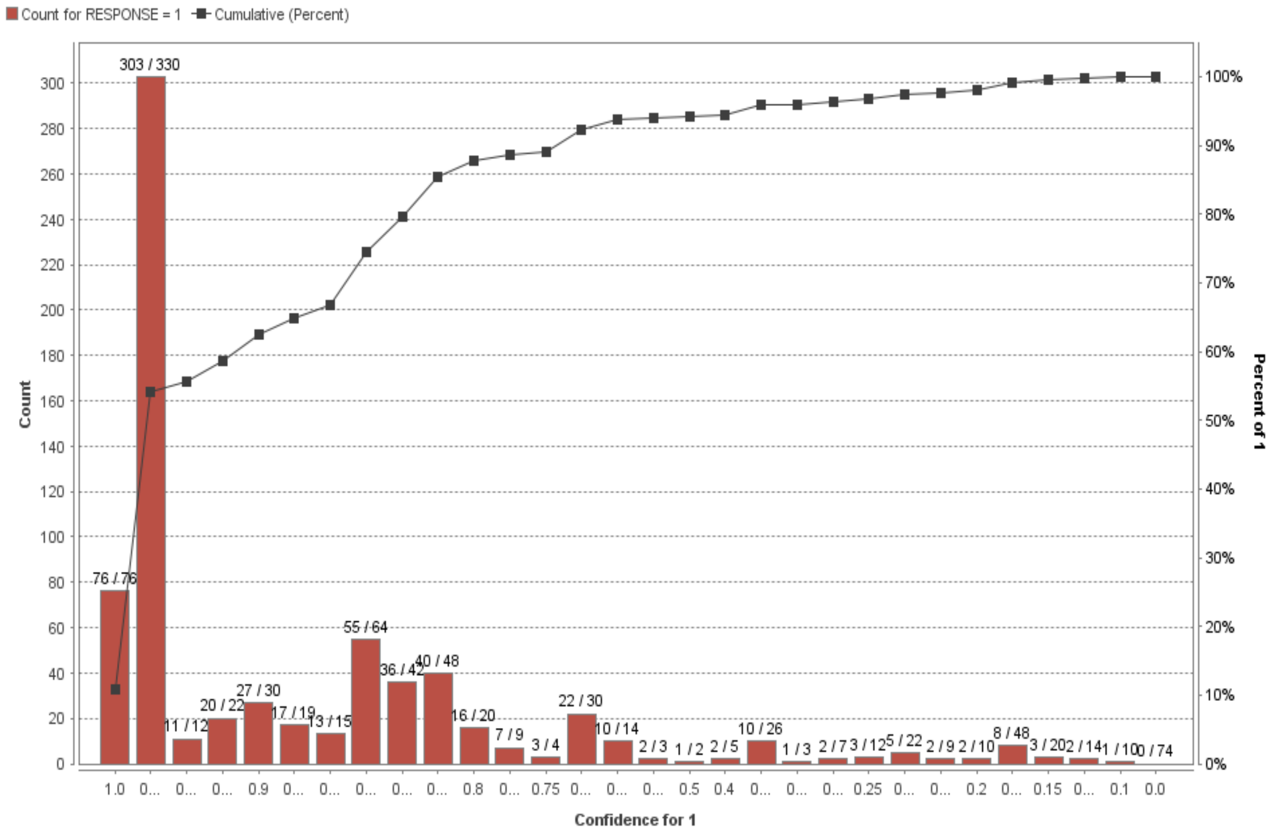
	SPLIT DATA			J48										TRAINING RESULTS				TEST RESULTS				
SNO	SPLIT RATIO	SAMPLING TYPE	RANDOM SEED	USE UNPRUNED TREE(U)	CONFIDENCE THRESHOLD FOR PRUNING(C)	MINIMUM INSTANCES PER LEAF(M)	REDUCED ERROR PRUNING(R)	NUMBER OF FOLDS(N)	BINARY SPLITS(B)	DON'T PERFORM SUBTREE RAISING(S)	DO NOT CLEAN-UP AFTER TREE BUILDING(L)	LAPLACE SMOOTHING (A)	RANDOM SEED(Q)	TRAINING ACCURACY	PRECISION	SENSITIVITY	AUC	TEST ACCURACY	PRECISION	SENSITIVITY	AUC	VARIANCE
DM7	0.7	Shuffled	1992	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	-	80.29	70.18	57.97	0.824	70	52.46	34.41	0.709	10.29

The ROC comparison and Lift charts support the best model derived from J48 and W-Simple Cart support the decision derived from descriptive measures.

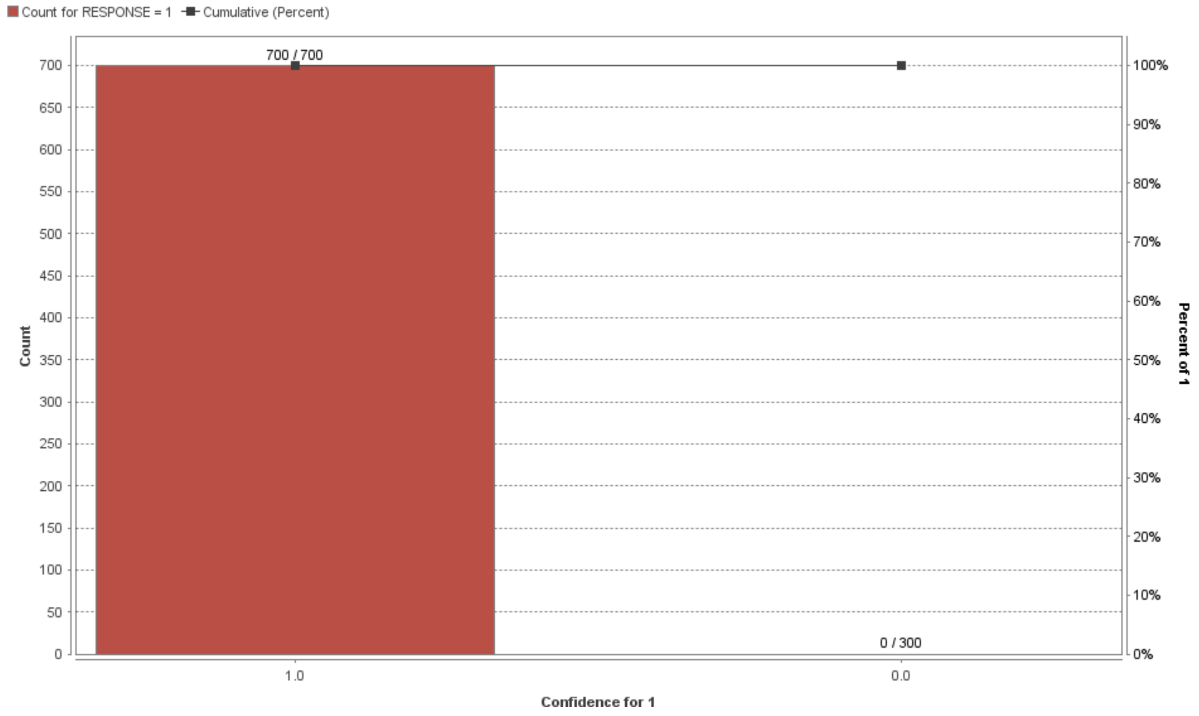
ROC Curve:



Lift Chart from J48 decision Operator:



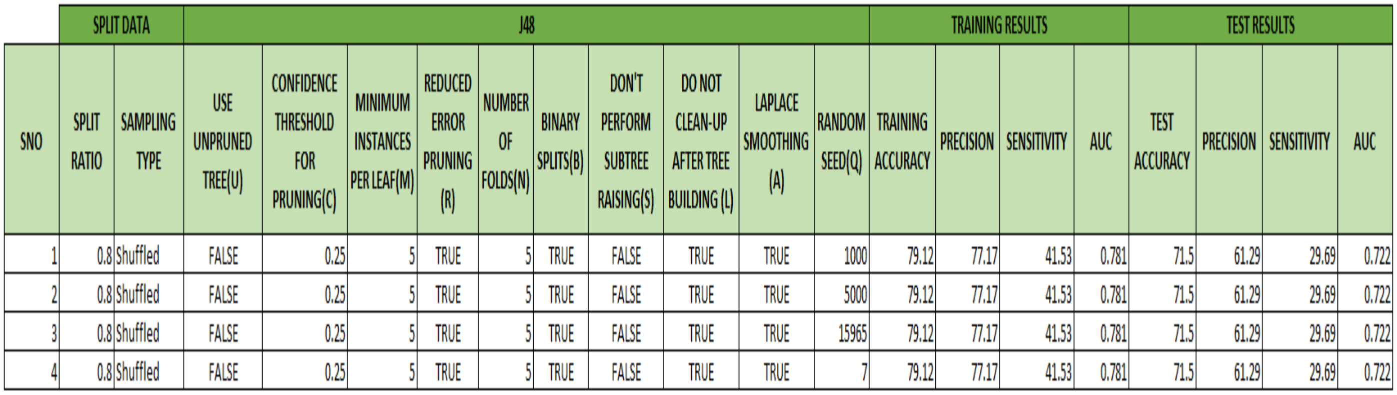
Lift Chart from W-Simple Cart:



The models from J48 and W Simple Cart decision operators differ in their accuracies depending on the parameters. W Simple Cart gave an overfit model whereas J48 gave many models with ranging accuracies and precisions.

2.2 (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?

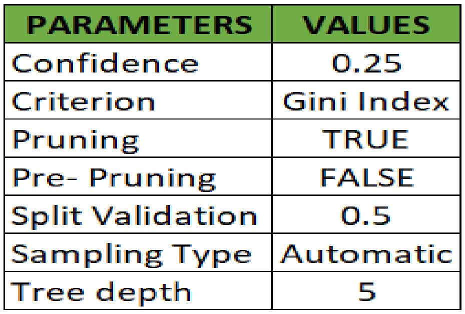
For the best models created from J48 operators, different seed values were allocated to the model to see any difference in the accuracy. It was observed that with different seed values there was no change in accuracy, precision or AOC level. On examining the trees for different seed levels, the node parameters did not change. Hence the model is relatively stable.



	SPLIT DATA		J48										TRAINING RESULTS				TEST RESULTS			
SNO	SPLIT RATIO	SAMPLING TYPE	USE UNPRUNED TREE(U)	CONFIDENCE THRESHOLD FOR PRUNING(C)	MINIMUM INSTANCES PER LEAF(M)	REDUCED ERROR PRUNING (R)	NUMBER OF FOLDS(N)	BINARY SPLITS(B)	DON'T PERFORM SUBTREE RAISING(S)	DO NOT CLEAN-UP AFTER TREE BUILDING (L)	LAPLACE SMOOTHING (A)	RANDOM SEED(Q)	TRAINING ACCURACY	PRECISION	SENSITIVITY	AUC	TEST ACCURACY	PRECISION	SENSITIVITY	AUC
1	0.8	Shuffled	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	1000	79.12	77.17	41.53	0.781	71.5	61.29	29.69	0.722
2	0.8	Shuffled	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	5000	79.12	77.17	41.53	0.781	71.5	61.29	29.69	0.722
3	0.8	Shuffled	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	15965	79.12	77.17	41.53	0.781	71.5	61.29	29.69	0.722
4	0.8	Shuffled	FALSE	0.25	5	TRUE	5	TRUE	FALSE	TRUE	TRUE	7	79.12	77.17	41.53	0.781	71.5	61.29	29.69	0.722

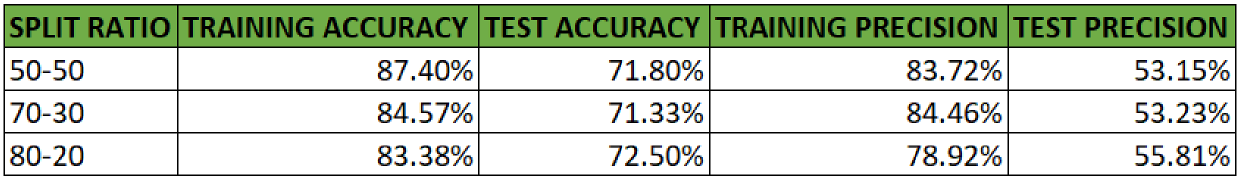
2.2 (d) Consider partitions of the data into 70% for Training and 30% for Test, and 80% for Training and 20% for Test and report on model and performance comparisons (for the decision tree learners considered above). In the earlier question, you had determined a set of decision tree parameters to work 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?

The training and test datasets are obtained by partitioning the actual dataset using different split ratios and used the following criteria as it provided best results on 50-50 splits.



PARAMETERS	VALUES
Confidence	0.25
Criterion	Gini Index
Pruning	TRUE
Pre- Pruning	FALSE
Split Validation	0.5
Sampling Type	Automatic
Tree depth	5

The summaries of our observation of accuracy values for the 3 models using different splits ratios for training and testing datasets is as follows:



SPLIT RATIO	TRAINING ACCURACY	TEST ACCURACY	TRAINING PRECISION	TEST PRECISION
50-50	87.40%	71.80%	83.72%	53.15%
70-30	84.57%	71.33%	84.46%	53.23%
80-20	83.38%	72.50%	78.92%	55.81%

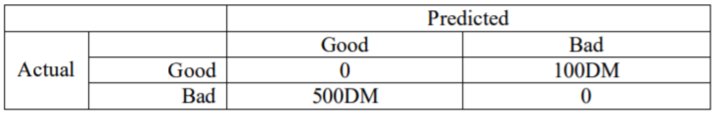
We tried different splits on J48 model which was choosed in question 2(b)

SPLIT RATIO	TRAINING ACCURACY	TEST ACCURACY	TRAINING PRECISION	TEST PRECISION
50-50	73.00%	69.00%	72.00%	46.67%
70-30	80.29%	70.00%	70.18%	52.46%
80-20	79.12%	71.50%	77.17%	61.29%

In all the models, it is observed that the top few nodes were split on the same variables as their Gini Index were higher. The root node is split on "Checking Account" and the second level splits were on "History", "Amount", "Real Estate" and "Other Installments". Then the 3rd split was on "Present Residence" or "Age" or "Guarantor" depending on the History Category for all the splits.

Model Preferred:
We would consider model based on Gini Index and spilt ratio as 80-20 as our best model as it has the maximum Accuracy and Precision.

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:

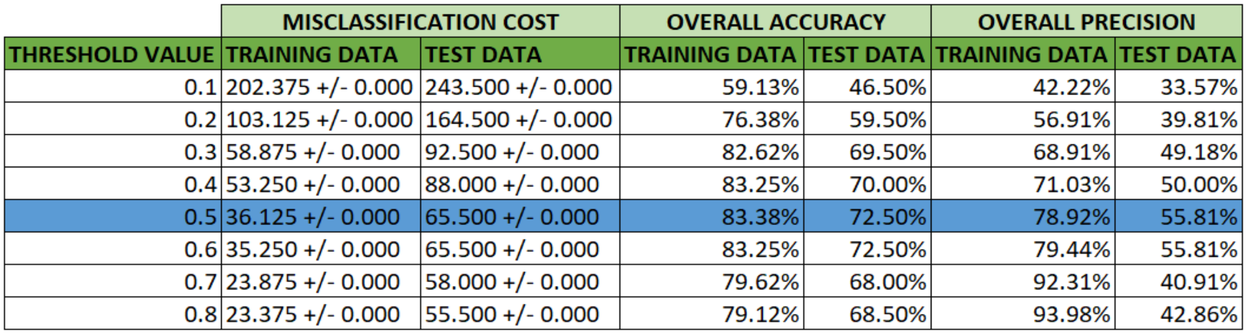


		Predicted	
		Good	Bad
Actual	Good	0	100DM
	Bad	500DM	0

Use the misclassification costs to assess performance of a chosen model from Q 2 above. Examine how different cutoff values for classification threshold make a difference – what do you find?

The best model used from the above question is Gini Index with 80-20 split.

The misclassification costs on both the training and test data by setting the threshold values between 0.1-0.8 is shown below:



	MISCLASSIFICATION COST		OVERALL ACCURACY		OVERALL PRECISION	
THRESHOLD VALUE	TRAINING DATA	TEST DATA	TRAINING DATA	TEST DATA	TRAINING DATA	TEST DATA
0.1	202.375 +/- 0.000	243.500 +/- 0.000	59.13%	46.50%	42.22%	33.57%
0.2	103.125 +/- 0.000	164.500 +/- 0.000	76.38%	59.50%	56.91%	39.81%
0.3	58.875 +/- 0.000	92.500 +/- 0.000	82.62%	69.50%	68.91%	49.18%
0.4	53.250 +/- 0.000	88.000 +/- 0.000	83.25%	70.00%	71.03%	50.00%
0.5	36.125 +/- 0.000	65.500 +/- 0.000	83.38%	72.50%	78.92%	55.81%
0.6	35.250 +/- 0.000	65.500 +/- 0.000	83.25%	72.50%	79.44%	55.81%
0.7	23.875 +/- 0.000	58.000 +/- 0.000	79.62%	68.00%	92.31%	40.91%
0.8	23.375 +/- 0.000	55.500 +/- 0.000	79.12%	68.50%	93.98%	42.86%

Observations:

- Misclassification cost decreases as the Threshold value is increased. At threshold value 0.3 we can see a sudden drop in misclassification cost.
- The precision is gradually increased as the threshold value is increased. Except when the Threshold value is 0.7, the precision of the test data is dropped while it is increasing for the training data.
- Accuracy is increased as the threshold is increased until a certain point i.e. till threshold = 0.5. If we further increase the threshold, the Accuracy of both training and test data decreases.

Hence, the Threshold value of "0.5" is optimal for our model as we get maximum accuracy and precision and misclassification cost on the test data is low.

4. Let's examine your 'best' decision tree model obtained.

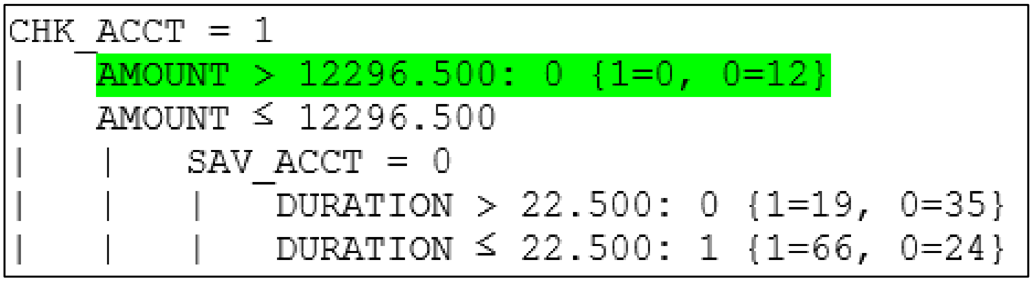
(a) What is the tree depth? And how many nodes does it have? What are the variables towards the 'top' of the tree, and are theysimilar to what you found in Question 2?

- **Tree depth** – 6
- **Number of Nodes** – 102
    - **Variables towards the 'top' of the tree:** Checking Account, History, Amount, Other Installment, Age, Guarantor and Present resident.
    - **Variables similar to the Q2:** Checking Account, History, Amount, Other Installment and Present resident

(b) Identify two relatively pure leaf nodes. What are the 'probabilities for 'Good' and 'Bad' in these nodes?

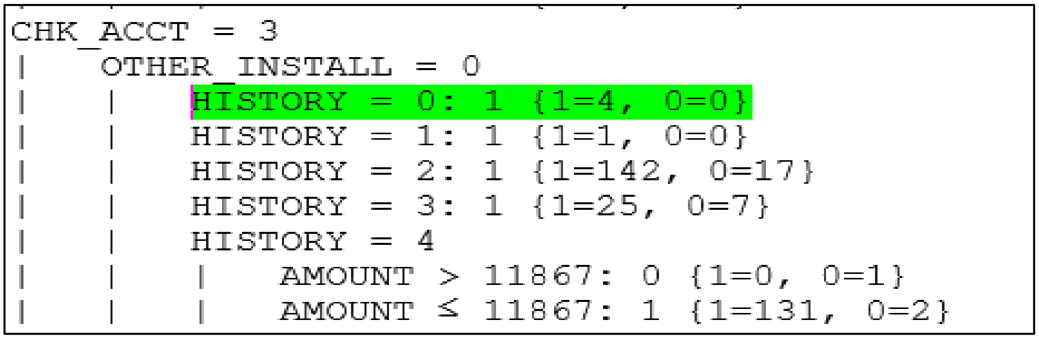
Two pure leaf nodes are given below:

1. For checking account = 1 and Amount > 12296.5
   - Number of bad credit scores i.e. 0 are 12 and number of good credit scores i.e. 1's is 0.
   - Probability of bad prediction is 1 and Probability of good prediction is 0.



```
CHK_ACCT = 1
|    AMOUNT > 12296.500: 0 {1=0, 0=12}
|    AMOUNT ≤ 12296.500
|    |    SAV_ACCT = 0
|    |    |    DURATION > 22.500: 0 {1=19, 0=35}
|    |    |    DURATION ≤ 22.500: 1 {1=66, 0=24}
```

2. For checking account = 3, other installment = 0 and History = 0
   - Number of good credit scores i.e. 1's is 4 and number of bad credit scores i.e. 0's is 0.
   - Probability of good prediction is 1 and Probability of bad prediction is 0.



```
CHK_ACCT = 3
|    OTHER_INSTALL = 0
|    |    HISTORY = 0: 1 {1=4, 0=0}
|    |    HISTORY = 1: 1 {1=1, 0=0}
|    |    HISTORY = 2: 1 {1=142, 0=17}
|    |    HISTORY = 3: 1 {1=25, 0=7}
|    |    HISTORY = 4
|    |    |    AMOUNT > 11867: 0 {1=0, 0=1}
|    |    |    AMOUNT ≤ 11867: 1 {1=131, 0=2}
```

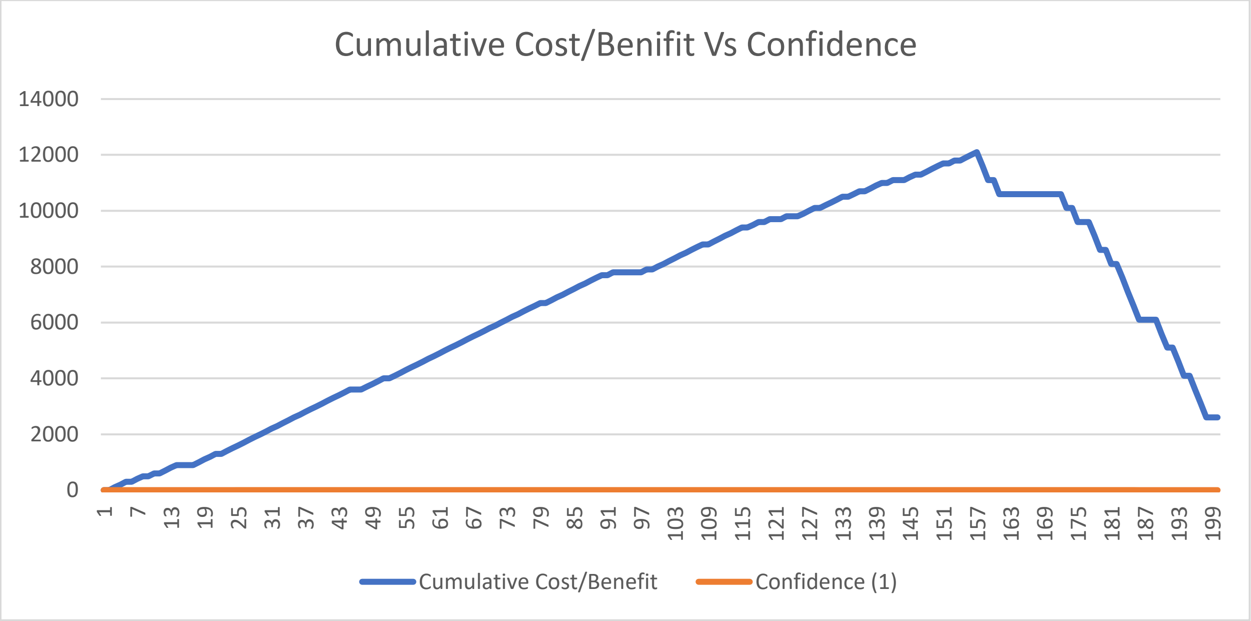
(c) The tree can be used to obtain rules – give two sample rules obtained from the tree. (Rules will be of the form IF condition AND condition AND.... THEN classification).

The two sample rules obtained are as follows:

1. IF Checking Account = 0 AND History = 0 AND Permanent Resident =4 THEN 0.
2. IF Checking Account = 0 AND Real Estate = 1 AND Install rate <= 3.5 AND Job =2 THEN 1.

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.

The below graph is the plot of Cumulative Cost/Benefit for good cases.



The maximum cumulative benefit is $12100 at 157th row (observation 633) and the confidence interval for good cases is 0.5. After which the Cumulative Cost starts dropping as we start incurring losses at row 160. In using this model to score future credit applicants, what cutoff value for predicted probability

In Question 3 we had observed that Accuracy for the test data was greatest at threshold value of "0.5".

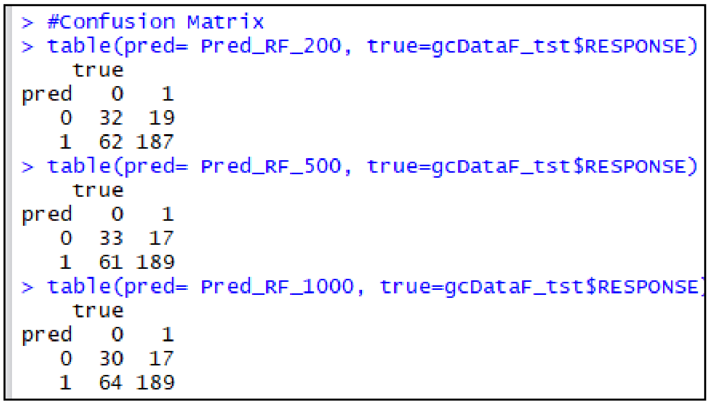
Therefore, for using this model to score future credit applicants, we would recommend 0.5 as the cutoff value for predicted probability.

RANDOM FOREST:

Random Forest was performed for different number of trees (200,500,700) The Error rate and Accuracy obtained are as follows:

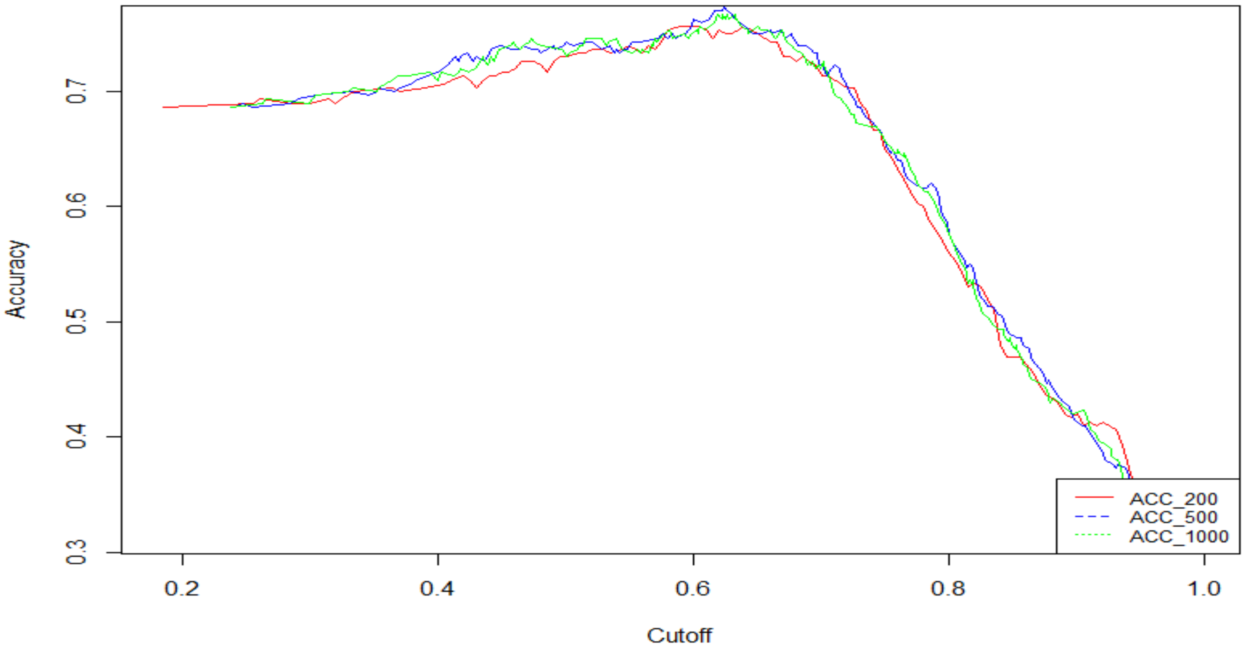
NUMBER OF TREES	ERROR RATE	ACCURACY
200	0.27	0.73
500	0.26	0.74
1000	0.27	0.73

The confusion matrix for the above:



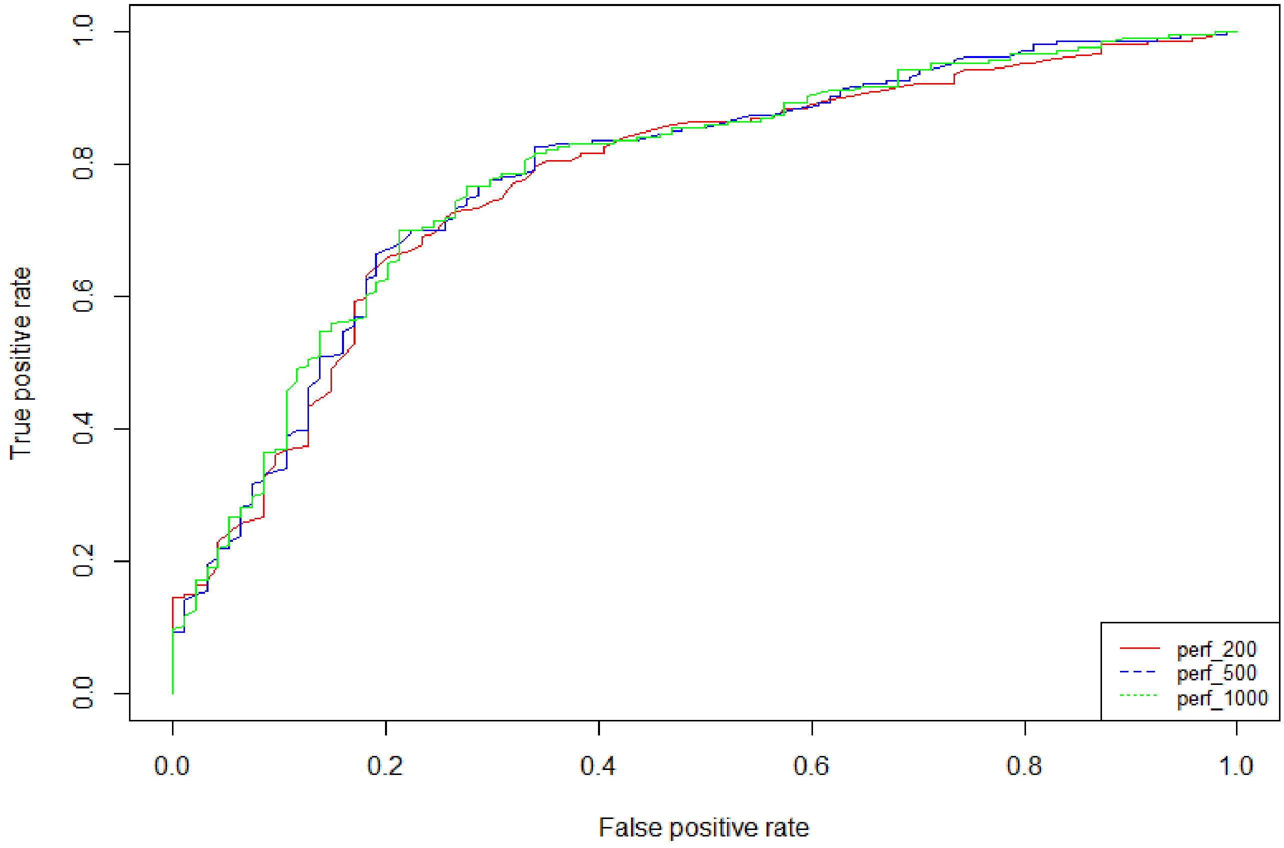
```
> #Confusion Matrix
> table(pred= Pred_RF_200, true=gcDataF_tst$RESPONSE)
    true
pred   0   1
   0  32  19
   1  62 187
> table(pred= Pred_RF_500, true=gcDataF_tst$RESPONSE)
    true
pred   0   1
   0  33  17
   1  61 189
> table(pred= Pred_RF_1000, true=gcDataF_tst$RESPONSE)
    true
pred   0   1
   0  30  17
   1  64 189
```

The below plot also shows Random Forest with 500 Trees has maximum accuracy performance and the cutoff value is 0.6 approximately.



ROC

The performance of all the three cases (i.e For 200, 500 and 1000 trees) looks similar and the ROC for the same is below.



AUC

The AUC for the different tree parameters is as follows:

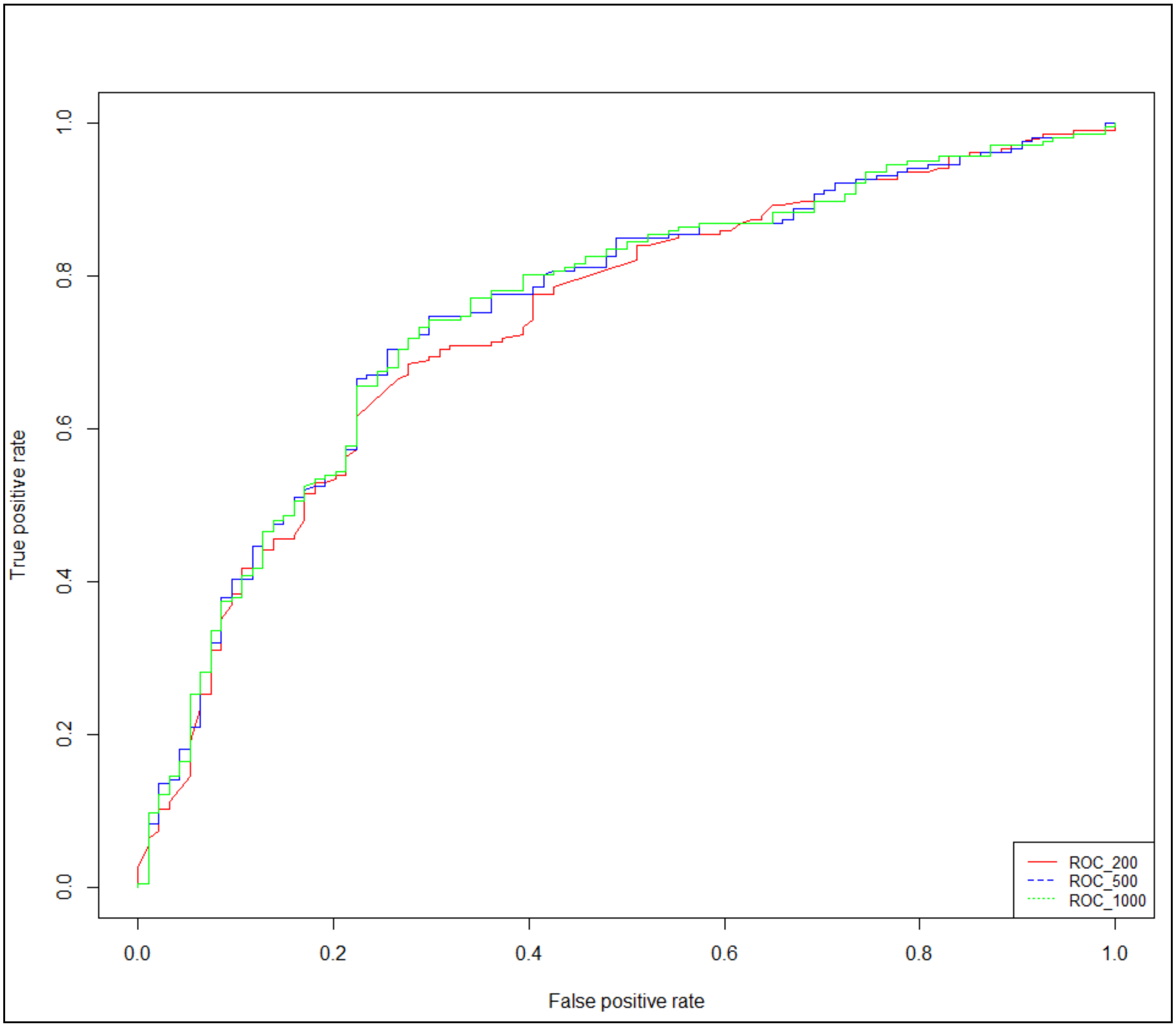
AUC for 200: 0.7730583
AUC for 500: 0.7818891
AUC for 1000: 0.7850134

As observed above, the AUC for 1000 trees is more than 500 trees by approximately 0.4% only and since accuracy of Random forest model with 500 trees is higher, we choose the Random Forest model with 500 trees.

ADA BOOST:

ROC

The performance of all the three cases (i.e For 200, 500 and 1000 trees) looks similar and the ROC for the same is below.



AUC

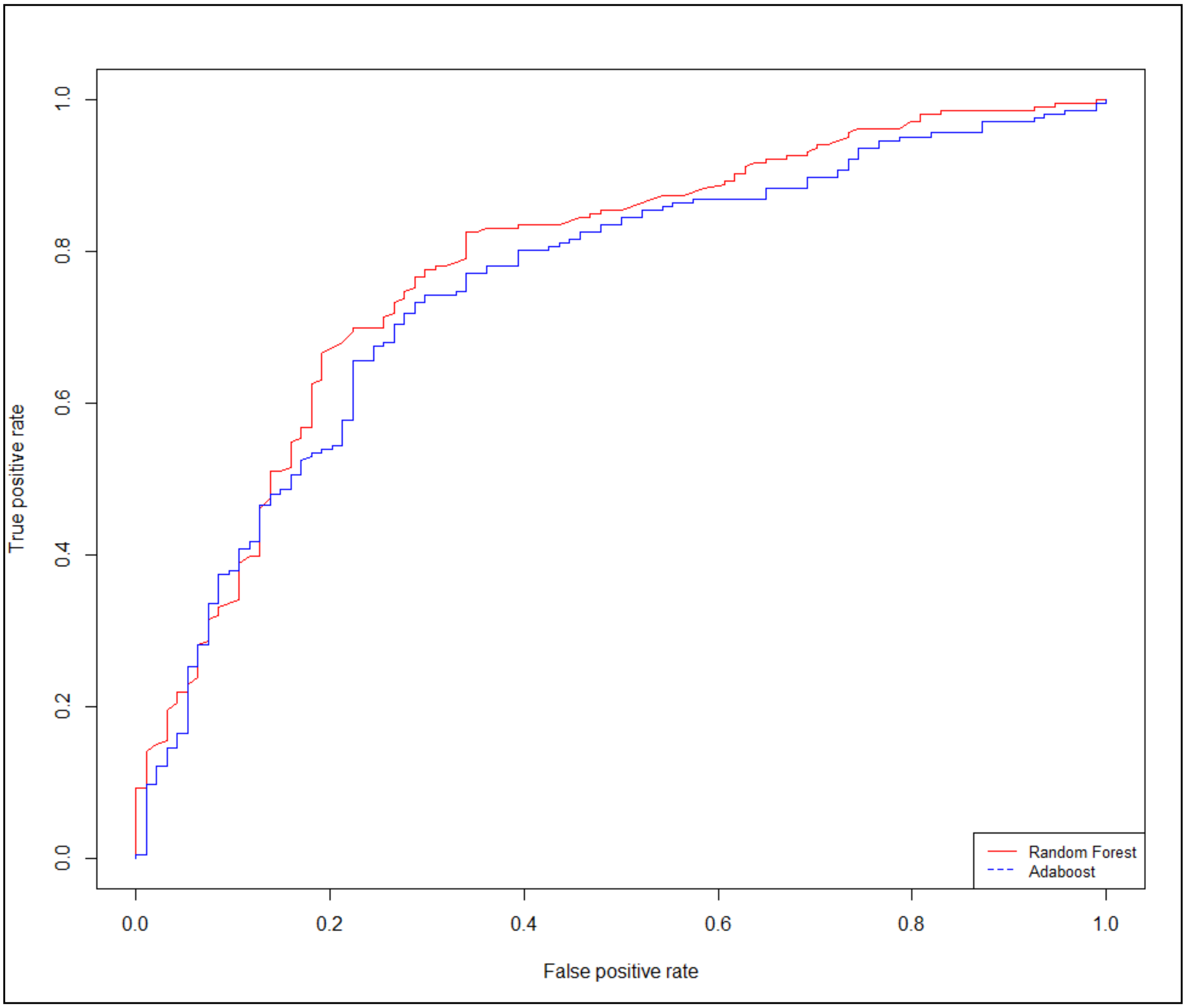
The AUC for the different tree parameters is as follows:

AUC for 200: 0.7386387
AUC for 500: 0.7511878
AUC for 1000: 0.7526596

As observed above, the AUC for Adaboost model with 1000 trees is better than the other two models. And henc e we chos the adaboost model with 1000 trees.

Comparision of Adaboost and Random Forest:

The comparision ROC is given below:



From the above ROC, its clear that Randomforest model is better than Adaboost as the true positive rate is higher for Random Forest than Adaboost.