Data Mining for Business – Assignment 1

Course : IDS572

Member #1 : Anshu Pathak

Member #2 : Srishti Jaju

Member #3 : Sriram Rajan

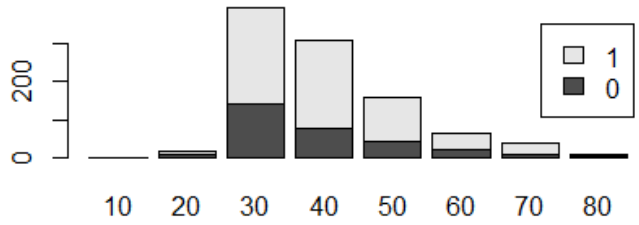
Course: IDS572	**Assignment #**: 1	**Course Name:** Data Mining

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?

Proportion of Good to Bad cases

GOOD CASES(1)	BAD CASES(0)
700	300

Column 'Age' has 9 missing values. Out of which 5 are Good case observation and 4 are bad cases. Distributing the missing values into the plot as shown below is not affecting the inference of the graph. So even if we do not handle these values, there won't be changes to the proportion of Good & Bad cases.



Predictor variables - DURATION, AMOUNT, INSTALL_RATE, AGE, NUM_CREDITS, NUM_DEPENDENTS

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis
DURATION	1	1000	20.903	12.058815	18	19.4725	8.8956	4	72	68	1.0909038
AMOUNT	2	1000	3271.156	2822.625	2319.5	2754.5325	1627.1535	250	18424	18174	1.9438389
INSTALL_RATE	3	1000	2.973	1.1187147	3	3.09125	1.4826	1	4	3	-0.5297551
AGE	4	991	35.48335	11.37077	33	34.09206	10.3782	19	75	56	1.0285522
NUM_CREDITS	5	1000	1.407	0.5776545	1	1.33375	0	1	4	3	1.2687608
NUM_DEPENDENTS	6	1000	1.155	0.3620858	1	1.06875	0	1	2	1	1.90372

Frequencies of different categorical variables

CHK_ACCT

	<0 DM	0<…<200 DM	=>200	No Checking A/c
Bad	135	105	14	46
Good	139	64	49	348

Course: IDS572	**Assignment #:** 1	**Course Name:** Data Mining

HISTORY

	No Credits	All Credits Paid Back Duly	Existing credits paid back duly	Delay in paying off in the past	Critical A/c
Bad	25	28	169	28	50
Good	15	21	361	60	243

EDUCATION

	No	Yes
Bad	278	22
Good	672	28

SAV_ACCT

	<100 DM	100<=…<500 DM	500<=…<1000 DM	=>1000 DM	Unknown/No Savings Acct
Bad	217	34	11	6	32
Good	386	69	52	42	151

EMPLOYMENT

	Unemployed	<1yr	1<=..<4 Years	4<=..<7 Years	>=7 Years
Bad	23	70	104	39	64
Good	39	102	235	135	189

PRESENT_RESIDENT

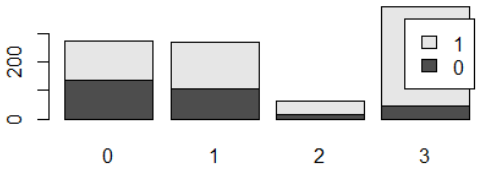
	<=1 Year	1<..<=2 Years	2<..<=3 Years	>4 Years
Bad	36	97	43	124
Good	94	211	106	289

JOB

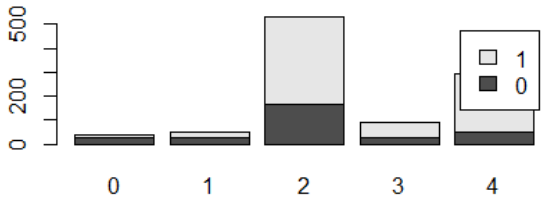
	Unemployed/Unskilled-Non-Resident	Unskilled-Resident	Skilled Employee/Official	Management/Self-Employed
Bad	7	56	186	51
Good	15	144	444	97

While examining variable plots following are some of the observation

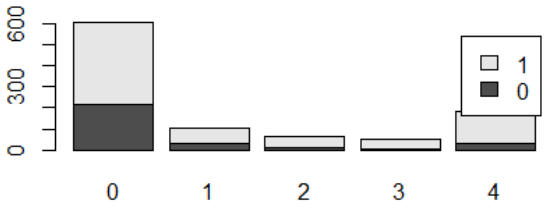
CHK_ACCT - We can see from the variable plot of CHK_ACCT that customers who don't have any checking account(3) have larger proportion of Good cases which is something not expected.



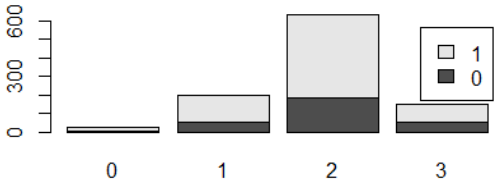
HISTORY: The Credit History variable plot below indicates that customers with existing credits paid back duly(2) have larger proportion of Good Cases. Also customers who have critical account(4) have larger proportion of Good Cases compared to Bad Cases.



SAV_ACCT - When average balance in savings account is less than 100DM(0) and when the savings account is unknown or there is no savings account(4) the proportion of Good cases is higher which is odd.



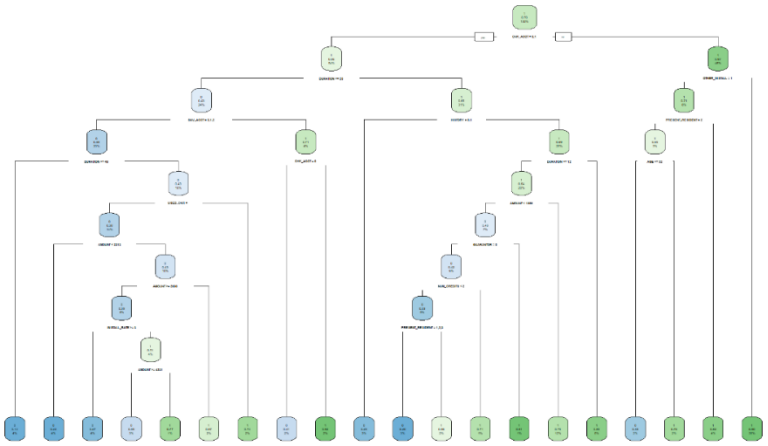
JOB - When nature of job is skilled employee/official(2) the proportion of Good cases is high as compared to customers with unskilled jobs.



Most relevant variables for the outcome of interest is as following based on correlation coefficient)

	0.3508475
HISTORY	0.2287847
DURATION	-0.2149267
SAV_ACCT	0.1789427
AMOUNT	-0.1547584

2. 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.



Critical parameters in building the model are: a) Minsplit – minimum number of splits at a node in order for a split b) Cp – complexity parameter which helps to reduce the complexity of the tree.

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)?

Parameters	Overall
HISTORY	55.94
CHK_ACCT	55.43
AMOUNT	47.73
DURATION	41.63
SAV_ACCT	36.69
REAL_ESTATE	27.31
USED_CAR	13.82
PRESENT_RESIDENT	13.47
INSTALL_RATE	13.38
EMPLOYMENT	13.17
GUARANTOR	11.50
AGE	8.30

Variable importance in a model can be determined with the help of **VarImp(model)** function in R.

Important variables in building the tree obtained from the function is very similar to the important variable list generated from manual inspection of variables plotted against Response. CHK_ACCT, DURATION, AMOUNT, HISTORY, EMPLOYMENT, SAV_ACCT etc. are the parameters that have been commonly identified as important from the decision tree and manual inspection.

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.

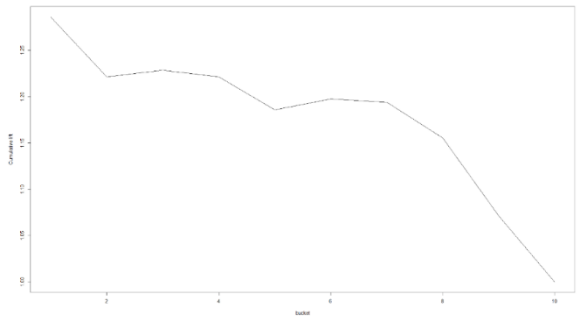
Confusion Matrix (Training data):

		Actual	
		0	1
Prediction	0	173	62
	1	127	638

Performance Evaluation Metrics (Training data):

Parameters	Values
Accuracy	81%
TP rate	91%
FP rate	58%

Lift Chart:



Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.29	1.22	1.23	1.22	1.18	1.19	1.19	1.15	1.08	1

Lift chart tells that top 10% of the cases have almost **30% higher chances** of being classified as 'Good' credit risk rather than choosing random target of 10% of cases. 10th decile lift is 1, indicating that targeting all the cases will have no difference compared to our full set of cases.

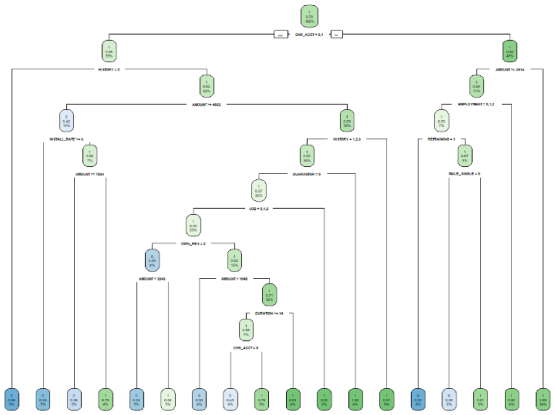
Decision trees are not robust because the model is extremely sensitive to data: any small change in data can result in a completely different tree. Moreover, complexity of the model tend to increase with the size of dataset.

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.

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. 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?

Course: IDS572	**Assignment #**: 1	**Course Name:** Data Mining



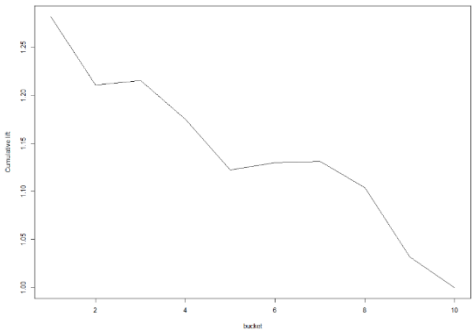
Confusion Matrix:

		Actual	
		0	1
Prediction	0	73	56
	1	76	295

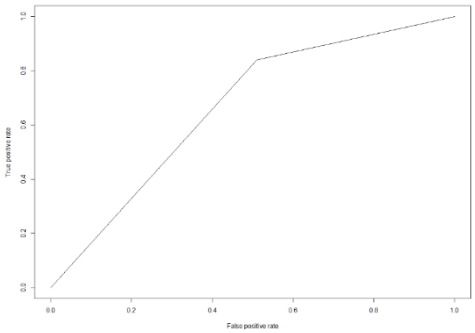
Performance Evaluation Metrics:

Parameters	Values
Accuracy	74%
Sensitivity	84%
Specificity	49%
Precision	80%

Lift Chart:



ROC curve:



Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.28	1.2	1.2	1.18	1.13	1.13	1.14	1.11	1.03	1

AUC (Area Under Curve): 0.67

Critical parameters which have been useful in deriving good model performance has been the following:

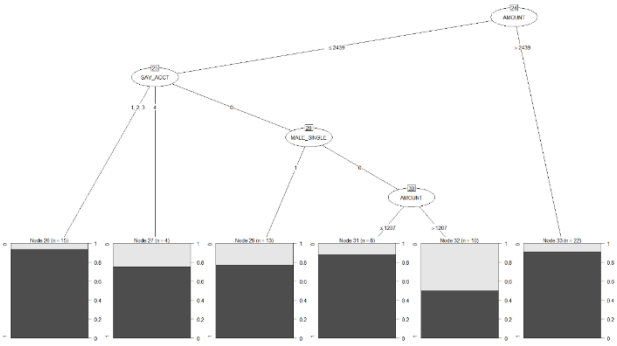
1. Split type – 'Gini' or 'Information' which helps to decide the purity of the node.
2. Minsplit – Minimum of cases to split at a node. Higher the number, smaller the tree size as there are more number of cases required to split the root node
3. Cp (Complexity parameter) – Helps to determine the complexity or size of the tree. As the number increases, size of the tree becomes smaller and hence impacts the overall accuracy of the outcomes.

Complexity parameter plays a crucial role in developing a good model performance. It helps to determine how the cost of the tree is compensated by number of terminal nodes, resulting in normalized

cost. Small cp results in bigger tree and hence resulting in overfitting. Larger cp results in smaller tree and hence underfitting.

(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?

Node = 24 (refer to code for the complete tree)



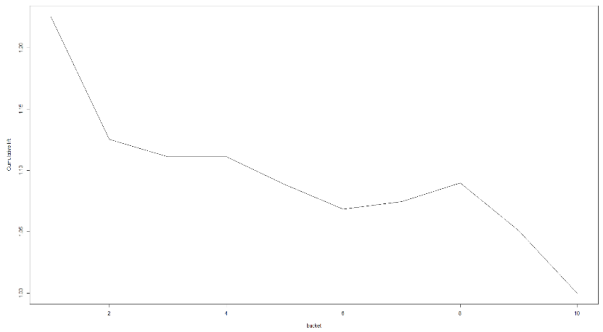
Confusion Matrix:

		Actual	
		0	1
Prediction	0	55	42
	1	94	309

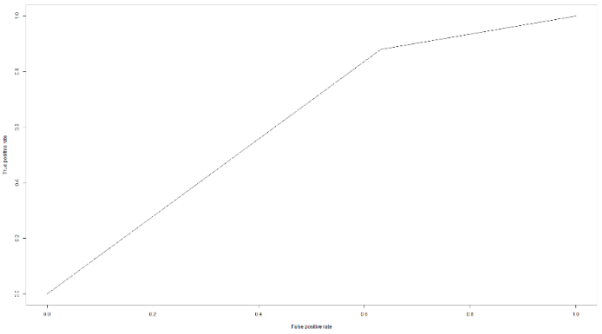
Performance Evaluation Metrics:

Parameters	Values
Accuracy	73%
Sensitivity	88%
Specificity	37%
Precision	77%

Lift Chart:



ROC Curve:



Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.22	1.13	1.11	1.11	1.09	1.07	1.08	1.10	1.05	1

AUC (Area Under Curve): 0.63

C5.0 Rules:

*not exhaustive, refer to the code for the complete list of rules

Rules:

Rule 1: (10, lift 3.0)
CHK_ACCT in {0, 1}
SAV_ACCT = 0
AMOUNT > 10875
-> class 0 [0.917]

Rule 2: (8, lift 3.0)
CHK_ACCT in {0, 1}
SAV_ACCT = 0
MALE_SINGLE = 0
GUARANTOR = 0
REAL_ESTATE = 1
AMOUNT > 1207
AMOUNT <= 2439
-> class 0 [0.900]

Rule 3: (19/5, lift 2.4)

CHK_ACCT in {0, 1}
HISTORY in {1, 2, 3}
SAV_ACCT = 1
REAL_ESTATE = 0
NUM_DEPENDENTS <= 1
-> class 0 [0.714]

Rule 4: (14/4, lift 2.3)

CHK_ACCT in {0, 1}
HISTORY = 4
GUARANTOR = 0
REAL_ESTATE = 0
AMOUNT > 4594
-> class 0 [0.688]

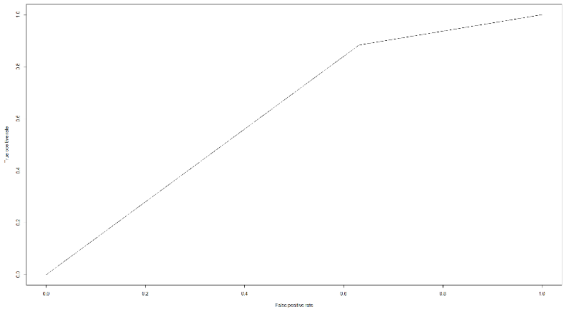
Confusion Matrix:

		Actual	
		0	1
Prediction	0	55	41
	1	94	310

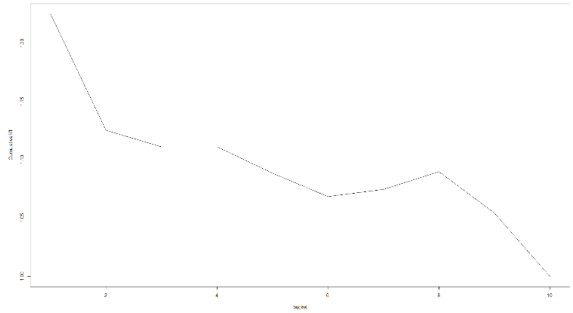
Performance Evaluation Metrics:

Parameters	Values
Accuracy	73%
TP rate / Specificity	88%
FP rate / Sensitivity	37%
Precision	77%

ROC Curve:



Lift Chart



Decile	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Lift	1.22	1.13	1.11	1.11	1.09	1.07	1.08	1.10	1.05	1

AUC (Area Under Curve): 0.63

Comparison of Performance

Parameters	Rpart	C5.0(Tree)	C5.0(Rules)
Overall Accuracy	74%	73%	73%
Sensitivity	84%	88%	88%
Specificity	49%	37%	37%
Precision	80%	77%	77%
Top Decile Lift	1.28	1.22	1.22
Area Under Curve	0.67	0.63	0.63

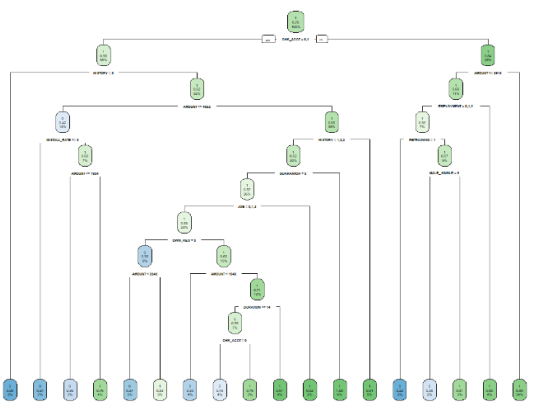
For the same set of training and test data, applying rpart and C5.0 earns very close results. Model obtained by C5.0 uses Information Gain as splitting criteria while rpart uses Gini index as splitting criteria. For rpart, pruning is based on Cost – Complexity factor to remove redundant branches but affects the accuracy. But C5.0 uses Binomial confidence limit to reduce the size of the tree and hence there is no loss in accuracy. Moreover, with rpart pruning does not happen at bottom of tree while C5.0 are properly pruned at all portions of the tree.

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?

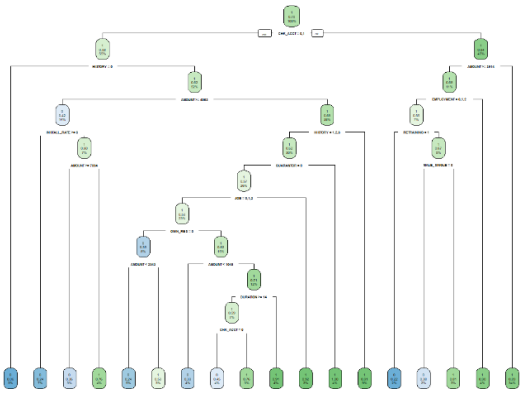
Comparison of Performance

Parameters	Random seed 1 (1006)	Random seed 2 (1234)
Overall Accuracy	74%	70%
Sensitivity	84%	84%
Specificity	49%	35%
Precision	80%	74%
Top Decile Lift	1.28	1.21
Area Under Curve	0.67	0.60

Random seed 1 – Model



Random seed 2 - Model



Based on the comparison, by changing the random seed, model and performance of the tree are completely different, keeping the parameters exactly the same. Since the decision tree is sensitive to data, by changing the random seed, training data set varies resulting in different model and hence a varied performance.

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

Variable Importance:

Using rpart

Parameters	Value
AMOUNT	53.92472
DURATION	30.909232
HISTORY	28.724536
OWN_RES	25.095145
PROP_UNKN_NONE	24.615467
SAV_ACCT	20.132758
CHK_ACCT	19.403457
GUARANTOR	14.537021
REAL_ESTATE	14.397315
AGE	12.501265
EMPLOYMENT	11.420922
NEW_CAR	7.267137
RETRAINING	7.166651
MALE_SINGLE	5.985569
OTHER_INSTALL	5.449124
NUM_CREDITS	5.134918
JOB	5.019453
INSTALL_RATE	4.587894
RENT	1.74434

Using C5.0

Parameters	Value
CHK_ACCT	100.0
HISTORY	94.0
GUARANTOR	45.6
REAL_ESTATE	19.8
SAV_ACCT	19.0
JOB	11.6
AMOUNT	6.2
NUM_DEPENDENTS	3.8
MALE_SINGLE	1.6

Variable importance in building models using rpart and C5.0 have common variables. But the importance value assigned to those variables between best models are different.

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

Performance Comparison (keeping the parameters the same)

Using rpart

Parameters	50 – 50	70 – 30	80 - 20
Overall Accuracy	74%	76%	74%
Specificity	84%	89%	85%
Sensitivity	49%	48%	46%

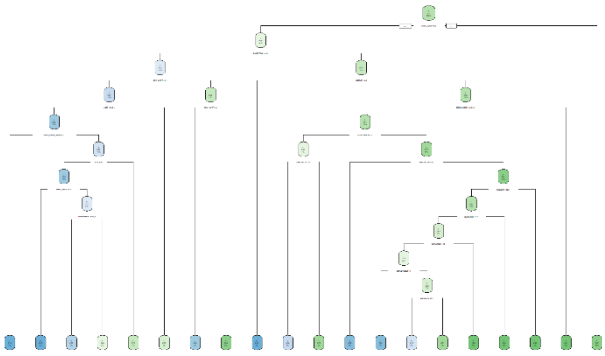
Precision	80%	78%	80%
Top Decile Lift	1.28	1.43	1.39
Area Under Curve	0.67	0.69	0.65

Using C5.0

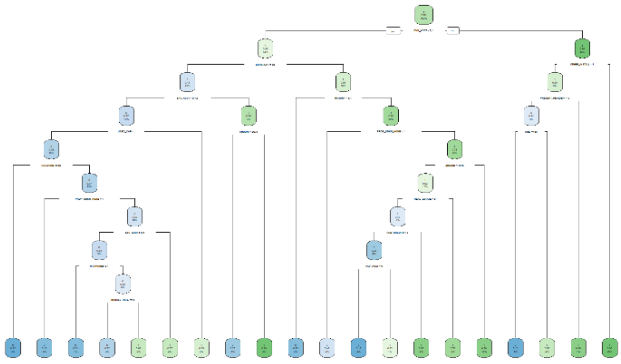
Parameters	50 – 50	70 – 30	80 - 20
Overall Accuracy	73%	72%	72%
Sensitivity	88%	90%	85%
Specificity	37%	35%	38%
Precision	77%	74%	78%
Top Decile Lift	1.22	1.38	1.32
Area Under Curve	0.63	0.63	0.62

Keeping the decision tree learner parameters the same, it is evident that performance and model varies with the size of training and test data. Performance metrics keeps varying indicating that for different size of the training data, parameters have to changed accordingly in order to obtain the best model.

r-part (70/30)



r-part (80/20)



Considering the models with different size of the training data, we can infer the upper part of the tree are similar. Predictors with higher information gain / purity / correlation clearly explains the distribution of the dependent variable. Hence they are less affected by the size of training data. So they tend to be root nodes or non-leaf nodes below the root nodes. While predictors with lower information gain / purity have lesser details on distribution of the dependent variable. So they sit at the bottom of tree. With increase in size of the data, impurity increases and hence predictors sitting at the bottom of the tree are subjected to change.

Ideally, 70 – 30 combination of training and test data would be a preferred combination for building a model. This helps to build a stable model and provides sufficient data to test the model and gain a higher confidence level.

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

This information can be used to determine the following costs for misclassification

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

Comparing different threshold values on a model developed on full data to reduce FP count

Parameters	Threshold 0.5	Threshold 0.6	Threshold 0.7
Overall Accuracy	81%	81%	80%
Sensitivity	58%	60%	62%
Specificity	91%	89%	88%
Precision	83%	84%	85%
Top Decile Lift	1.28	1.29	1.29
Area Under Curve	0.74	0.75	0.75
FP count	127	119	114

It is clearly seen from the table that with change in threshold value, accuracy of the model reduces but at the same time, False Positive counts also reduce indicating that overall misclassification cost will go down.

Theoritical threshold = 500 / (500+100) = 0.81

Using r-part (on full data with threshold as 0.8)

Parameters	Without Costs
Overall Accuracy	71%
Sensitivity	79%
Specificity	68%

Confusion Matrix (with threshold as 0.8)

		Actual	
		0	1
Prediction	0	237	222
	1	63	478

Precision	89%
Top Decile Lift	1.31
Area Under Curve	0.74
FP count	63

Using misclassification costs in building tree models yields a completely different tree from the ones generated without the costs. Below is the summary of performance for tree models with 50% data with cut-off value as 0.5

Using r-part

Parameters	Without Costs	With costs
Overall Accuracy	74%	69%
Sensitivity	84%	65%
Specificity	49%	71%
Precision	80%	83%
Top Decile Lift	1.28	1.34
Area Under Curve	0.67	0.68

Confusion Matrix (with costs)

		Actual	
		0	1
Prediction	0	98	102
	1	51	249

Using C5.0

Parameters	Without Costs	With costs
Overall Accuracy	73%	58%
Sensitivity	88%	81%
Specificity	37%	48%
Precision	77%	85%
Top Decile Lift	1.22	1.28
Area Under Curve	0.63	0.64

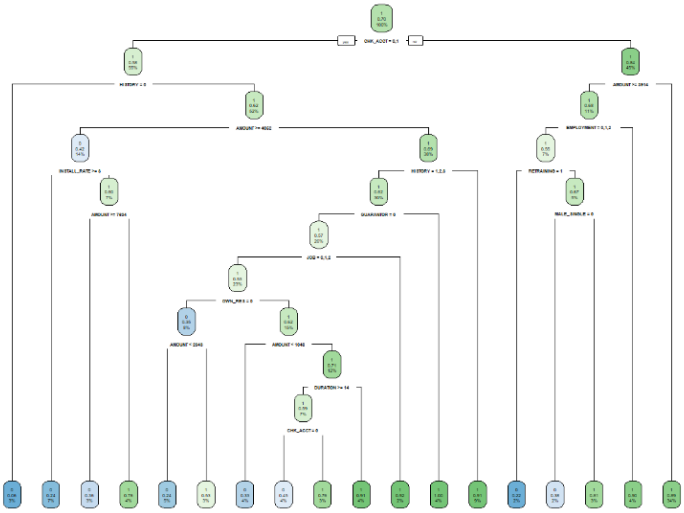
Confusion Matrix (with costs)

		Actual	
		0	1
Prediction	0	120	182
	1	29	169

5. Let's examine your 'best' decision tree model obtained. What is the tree depth? And how many nodes does it have? What are the important variables for classifying "Good' vs 'Bad' credit? Identify two relatively pure leaf nodes. What are the 'probabilities for 'Good' and 'Bad' in these nodes?

Best tree considered here is the model built with **rpart package with 50% training data**

Course: IDS572	**Assignment #**: 1	**Course Name:** Data Mining



Tree depth: 10

Number of nodes: 35

Variable Importance:

Parameters	Value
AMOUNT	53.92472
DURATION	30.909232
HISTORY	28.724536
OWN_RES	25.095145
PROP_UNKN_NONE	24.615467
SAV_ACCT	20.132758
CHK_ACCT	19.403457
GUARANTOR	14.537021
REAL_ESTATE	14.397315

Parameters	Value
AGE	12.501265
EMPLOYMENT	11.420922
NEW_CAR	7.267137
RETRAINING	7.166651
MALE_SINGLE	5.985569
OTHER_INSTALL	5.449124
NUM_CREDITS	5.134918
JOB	5.019453
INSTALL_RATE	4.587894
RENT	1.74434

Pure Leaf Nodes:

Following rules extracted from the model terminates at a relatively pure leaf nodes:

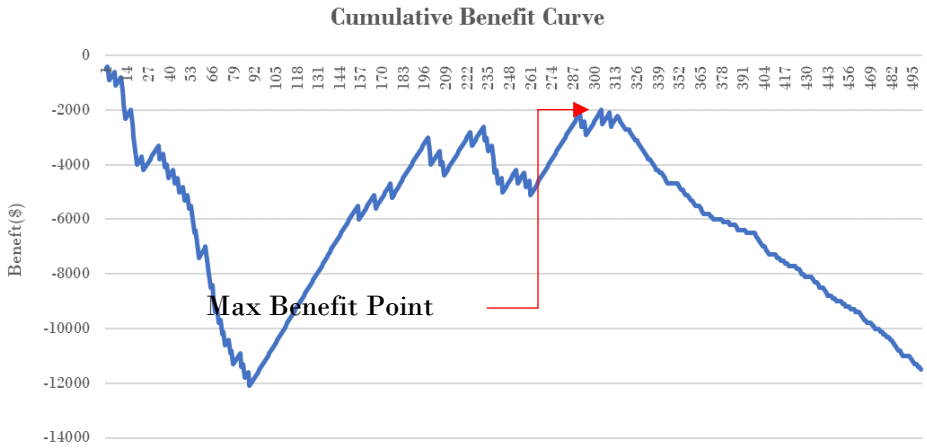
1. CHK_ACCT = 0,1 & HISTORY = 0
   P(Good) = 0.06, P(Bad) = 0.94
2. CHK_ACCT = 0,1 & AMOUNT < 4052 & HISTORY = 1,2,3 & GUARANTOR =1
   P(Good) = 1, P(Bad) = 0

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.

Best tree considered here is the model built with **rpart package with 50% training data**

Course: IDS572	**Assignment #:** 1	**Course Name:** Data Mining

Maximum Net Benefit Point



Maximum benefit point occurs at 304th observation. Hence, **maximum benefit point** occurs after traversing through **61% of the observations**. Probability value corresponding to the observation is **0.8125.**

Cut-off Value Vs Cumulative Net Benefit

Cut-off Value	Cumulative Net Benefit ($)
0.5	-14,100
0.6	-12,600
0.8	-11,500
0.8125	-12,300
0.9	-41,800

Based on the above table, ideal **cut-off value** to maximise the cumulative value is **0.8**.

Confusion Matric for cut-off value of 0.8

		Actual	
		0	1
Prediction	0	93	93
	1	56	258

Performance Metrics for cut-off value of 0.8

Parameters	50% training data
Overall Accuracy	70%
Sensitivity	62%
Specificity	74%
Precision	82%
Top Decile Lift	1.31
Area Under Curve	0.68