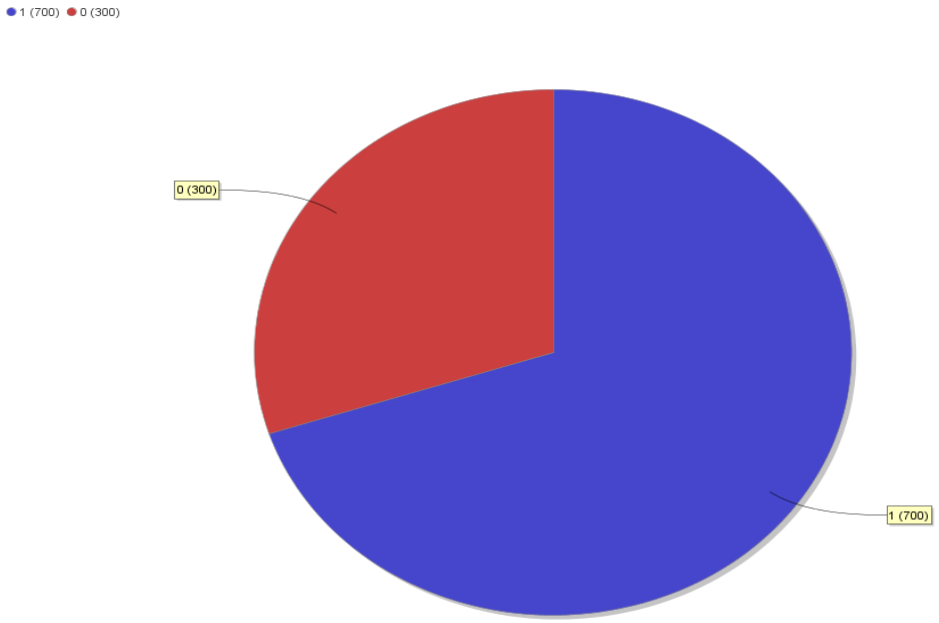
ASSIGNMENT 1: GERMAN CREDIT DATA
DECISION TREES

GROUP MEMBERS:
ASHUTHOSH GOWDA
LAAVANYA GANESH
AKSHAY MERCHANT

Q 1.

Explore the data: What is the proportion of "Good" to "Bad" cases? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-values attributes, frequencies of different category values. Examine variable plots. What are the interesting variables and relationships? Which variables do you think will be most relevant for the outcome of interest? (Why?)

A.1 : The proportion of 'Good' to 'Bad' cases is 7:3, with 700 having a Good Credit Rating and 300 having a Bad Credit Rating.



Description of Variables

Real Value Variables

Variable	Standard Dev	Mean	Meadian	Mode	Minimum	Maximum
Duration	12.059	20.903	18	24	4	72
Amount	2822.737	3271.258	2319.5	1262	250	18424
Install Rate	1.119	2.973	3	4	1	4
Age	11.375	35.546	33	27	19	75
Num_Credits	0.578	1.407	1	1	1	4
Num_Dependents	0.362	1.155	1	1	1	2

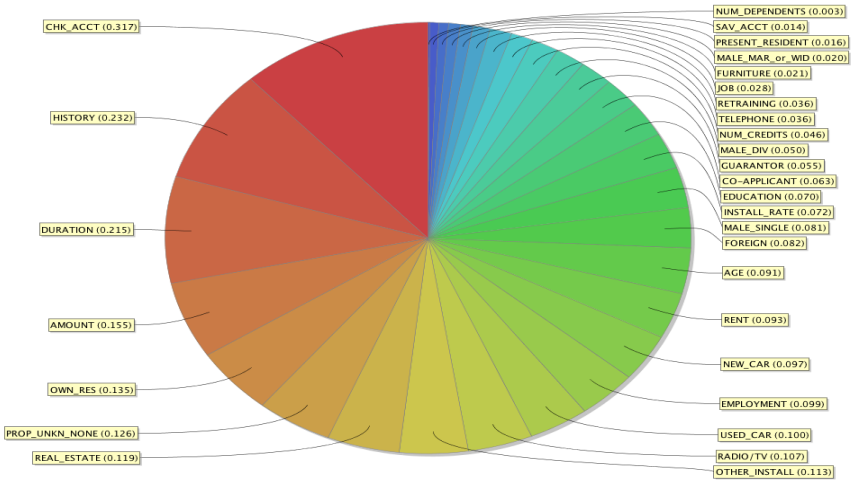
Categorical Variables(Frequency)

Variable	0	1	2	3	4
Chk_Acct	274	269	63	394	-
History	40	49	530	88	293
Sav_Acct	603	103	63	48	183
Employment	62	172	339	174	253
Present_Resident	-	130	308	149	413
Job	22	200	630	148	-

Binary Variables(Frequency)

Variable	1	0	Variable	1	0
New_Car	234	766	Co_Applicant	41	959
Used_Car	103	897	Guarantor	52	948
Furniture	181	819	Real_Estate	282	718
Radio/TV	280	720	Prop_Unknown	154	846
Education	50	950	Other_Install	186	814
Retraining	97	903	Rent	179	821
Male_Div	50	950	Own_Res	713	287
Male_Single	548	452	Telephone	404	596
Male_Mar_Wid	92	908	Foreign	37	963

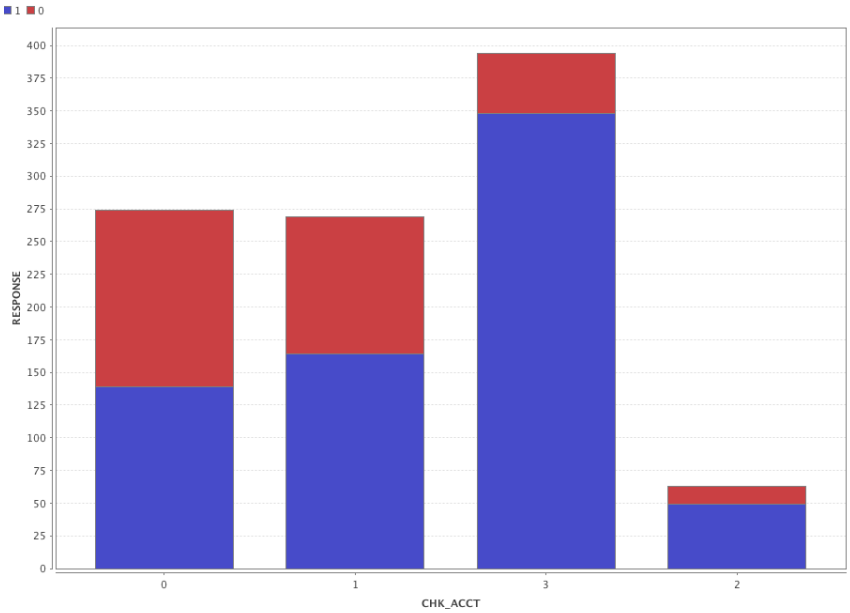
By correlating the above variables with respect to response, we made a pie chart comparing how much 'Response' depended on each of them.



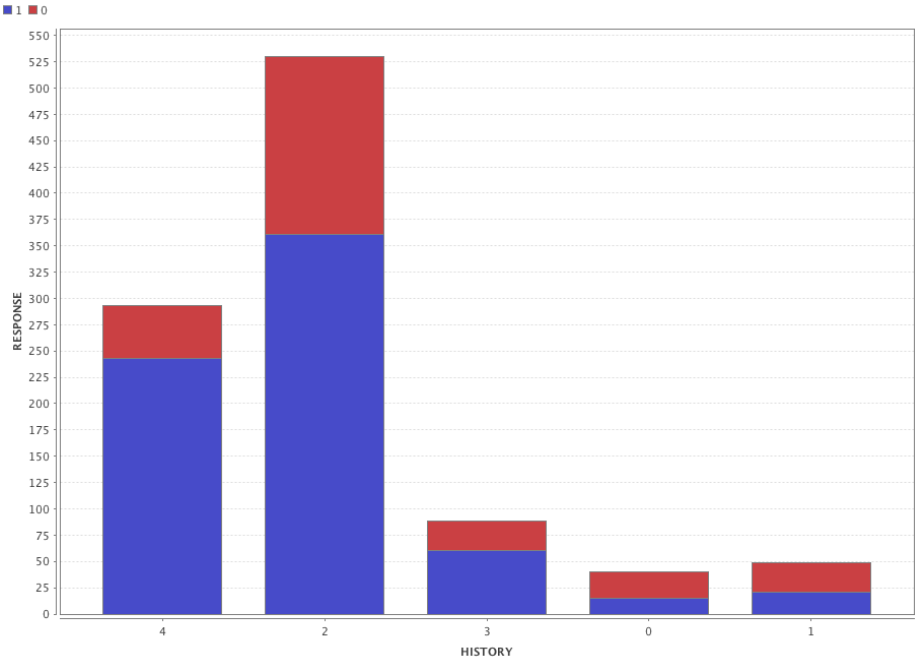
Correlating by weight, the most relevant variables for the outcome of interest and the variables interesting relationships with the response variable are:

Variable	Weight
Chk_Acct	0.317
History	0.232
Duration	0.215
Amount	0.155
Own_Res	0.135

Chk Acct(Checking Account) : Displays the status of the checking account for the applicants. The histogram below shows the proportion of applicants in each category who have a good credit rating(with 1=good rating, 0=bad rating)



History : Displays the credit history of the applicants. It describes whether a applicant has paid his credits on time or has delayed it. The histogram below shows the proportion of applicants in each category who have a good rating.



Duration : Displays the amount of time in months a applicant has taken to pay back the credits due. The tables below show the aggregate functions for applicants with Response=0/1.

For Response 0

Variable	Standard Dev	Mean	Meadian	Mode	Minimum	Maximum
Duration	13.26048	24.86	24	24	6	72

For Response 1

Variable	Standard Dev	Mean	Meadian	Mode	Minimum	Maximum
Duration	11.07165	19.207	18	12	4	60

This shows that applicants who have a good credit rating take a shorter duration to repay credits due.
Amount : Displays the credit amount due by the applicant. The tables below show the aggregate functions for applicants with Response=0/1.

For Response 0

Variable	Standard Dev	Mean	Meadian	Mode	Minimum	Maximum
Amount	3529.921	3938.127	2574.5	1282	433	18424

For Response 1

Variable	Standard Dev	Mean	Meadian	Mode	Minimum	Maximum
Amount	2399.756	2985.457	2244	1393	250	15857

This shows that applicants with a good credit rating have smaller dues than applicants with bad credit ratings.
Own_Res(Owns Residence) : Displays whether the applicant owns a residence or not. The frequency table below shows the proportion of applicants with Response=0/1 who own a residence

For Response 0

0	1
114	186

For Response 1

0	1
173	527

As shown in the table above, a majority of the applicants who own a residence have a good credit rating.

Q.2 We will first focus on a descriptive model – i.e. assume we are not interested in prediction. Develop a decision tree on the full data. Which variables are used to differentiate "good" from "bad" cases? What levels of accuracy/error are obtained? What is the accuracy for the "good" and "bad" cases? Do you think this is a reliable (robust?) description? What decision tree node parameters do you use to get a good model (and why?) Which variables are important for the outcome of interest (why)?

A.2:

Please find the decision tree that we have developed in the APPENDIX
We used the following decision tree node parameters to get a good model:

Criteria	Gini-Index
Maximal Depth	39
Confidence	0.2

Pruning and Pre-pruning both were applied using the following parameters:

Minimal gain	0.02
Minimal Leaf-size	5
Minimal Size for split	0.05
Number of pre-pruning alternatives	7

From the decision tree in the APPENDIX, the following variables are used to differentiate the "Good" from "Bad" cases and hence these variables are important for the outcome of interest:

1. **CHK_ACCT**: Shows the status and the credit-limit of the checking account
2. **AMOUNT**: Amount due to be paid, decides the chance of getting credit
3. **DURATION**: Duration of the applicant's loan in months
4. **HISTORY**: Shows the credit-risk with respect to past responses to paying loans
5. **OTHER_INSTALL**: whether or not the applicant has an another installment plan like bank/store

The above variables are top nodes or get repeated a lot in the decision tree obtained giving a better classification.

The confusion matrix showing the overall accuracy levels and the accuracies of the "Good" and "Bad" cases that we obtained is given below:

Overall Accuracy: 85.50%	**True 1**	**True 0**	**Class Precision**
Pred.1	655	100	86.75%
Pred.0	45	200	81.63%
Class recall	**93.57%**	**66.67%**	

The accuracy for "Good" and "Bad" cases is 93.57% and 66.67% respectively.

The decision tree obtained is a descriptive one since we are not interested in prediction. The current dataset contains only 1000 observations to train the model. It might not hold robust if there is a large dataset with noise.

Q.3.

3. 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? Is the model reliable (why or why not)?

b. 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. Feel free to experiment with other size partitions on the data. Is there any specific model you would prefer for implementation? In developing the models above, change some of the decision tree options and see if and how they affect performance (for example, the minimum number of cases at a leaf node, the split criteria). Also, does pruning give a better model – please explain why or why not ? Which parameter values do you find to be useful – are they the same for different training test partitions ?

c. Also, consider two other type of decision tree operators – for example, CART, J48 – play around with the parameters till you get a 'good' model. Describe any performance differences across different types of decision tree learners? 2 Comment on performance differences from use of different impurity measures (splitting criteria).

d. Decision tree models are referred to as 'unstable' – in the sense that small differences in training data can give very different models. After selecting a set of parameters which you find to work well, try building different models with different training samples (you can change the random seed for this). Do you find your models to be unstable? Are there similarities in, say, the upper part of the tree – and what does this indicate?

3 a)

To Develop this process, we used Validation Operator and split the Data into 50:50 as Train data and Validation data.

When we applied a model with Splitting criterion as Gini_index, Maximum Depth = 5, Confidence=0.25,Threshold = 0.35, we observe different Accuracies and Performance matrix depending on the Split Ratio we apply:

Split: 50Train-50 Validation

Validation Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 72.80%	True 1	True 0	Class Precision
Pred 1	288	78	78.69%
Pred 0	58	76	56.72%
Class Recall	83.24%	49.35%	

Train Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 86.0%	True 1	True 0	Class Precision
Pred 1	329	45	87.97%
Pred 0	25	101	80.16%
Class Recall	92.14%	69.18%	

The Validation Performance Matrix was obtained as shown above and, we observe an overall Accuracy percentage of 72.80%. This Model gives a moderately reliable accuracy, since this model gives a pretty high Class recall for True 1's and also has an even distribution of Class Precision.

3 b)

The Other Split Ratios we experimented with were: 60:40; 70:30; 80:20.
The following were the model Performance of the Above mentioned splits.

Split: 60 Train- 40 Validation:

Validation Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 68.50%	True 1	True 0	Class Precision
Pred 1	220	67	76.66%
Pred 0	59	54	47.79%
Class Recall	78.85%	44.63%	

Train Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 84.17%	True 1	True 0	Class Precision
Pred 1	391	65	85.75%
Pred 0	30	114	79.17%
Class Recall	92.87%	63.69%	

Split: 70 Train- 30 Validation:

Validation Data Performance Matirx

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 66.33%	True 1	True 0	Class Precision
Pred 1	168	62	73.04%
Pred 0	39	31	44.29%
Class Recall	81.16%	33.33%	

Train Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 83.75%	True 1	True 0	Class Precision
Pred 1	461	83	84.74%
Pred 0	32	124	79.49%
Class Recall	93.51%	59.90%	

Split: 80 Train- 20 Validation:

Validation Data Performance Matirx

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 68.50%	True 1	True 0	Class Precision
Pred 1	103	30	77.44%
Pred 0	33	34	50.75%
Class Recall	75.74%	53.12%	

Train Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35			
Accuracy: 82.75%	True 1	True 0	Class Precision
Pred 1	495	69	87.77%
Pred 0	69	167	70.76%
Class Recall	87.77%	70.67%	

Here, as we increase the Ratio of Train: Validation Data, we see a pattern of Decreasing Validation, and Train Data Overall Predicted Accuracy.

The Split Ratio of 56:44 on Train and Validation was found to have the Highest Overall Accuracy in Validation Data of 74.55%, further, we observed that a few nodes having 1 or 2 cases in them, hence suspecting the Overfit in such scenarios we gave a Minimum leaf size of 4 and found that the nodes we targeted, didn't split in this instance. This increased the accuracy slightly to give a best Overall Accuracy of 74.77%

Split: 56 Train- 44 Validation:

Validation Data Performance Matrix

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.25; Threshold=0.35 Minimal Leaf Size = 4			
Accuracy: 74.77%	True 1	True 0	Class Precision
Pred 1	255	61	80.70%
Pred 0	50	74	59.68%
Class Recall	83.61%	54.81%	

We experimented with the Minimal Gain Ratio in the Pre-pruning Option of the decision tree, we observed that even a minimum gain of 0.1, gave us just 1 level of split and ceased to split further.
A minimum gain of 0.05 was needed in order to let the tree grow. But, that did not give any improvement in the Accuracy.

c) W-J-48

The confusion matrix for training data showing the overall accuracy levels and the accuracies of the "Good" and "Bad" cases that we obtained is given below:

Overall Accuracy: 85.71%	**True 1**	**True 0**	**Class Precision**
Pred.1	374	60	86.18%
Pred.0	20	106	84.13%
Class recall	**94.92%**	**63.86%**	

The confusion matrix for testing data showing the overall accuracy levels and the accuracies of the "Good" and "Bad" cases that we obtained is given below:

Overall Accuracy: 74.77%	**True 1**	**True 0**	**Class Precision**
Pred.1	275	80	77.46%
Pred.0	31	54	63.53%
Class recall	**89.87%**	**40.30%**	

For W-J48 we used the following parameters:

J48	Pruning	Confidence	Minimum Leaf Size	Reduced Error pruning	Binary Splits only	No Subtree Raising	Do not clean up after tree is built	Training Performance	Validation performance
0.5:0.5	Y	0.25	2	N	N	N	N	86.69	70.40
0.7:0.3	Y	0.25	7	N	N	N	N	82	71.67
0.56:0.44	Y	0.19	2	N	N	N	N	85.71	74.77

W-SimpleCart

The confusion matrix for testing data showing the overall accuracy levels and the accuracies of the "Good" and "Bad" cases that we obtained is given below:

Overall Accuracy: 76.59%	**True 1**	**True 0**	**Class Precision**
Pred.1	277	74	78.92%
Pred.0	29	60	67.42%
Class recall	**90.52%**	**44.78%**	

Using these operators the performance of the model have increased over the simple Decision tree. The models are more 'robust' i.e., the differences between the performances of training and validation data set are relatively lower than that of decision tree operator

3d) Decision tree models are referred to as 'unstable' – in the sense that small differences in training data can give very different models. After selecting a set of parameters which you find to work well, try building different models with different training samples (you can change the random seed for this). Do you find your models to be unstable? Are there similarities in, say, the upper part of the tree – and what does this indicate?

A3 d). The best model we have got has an accuracy of 74.77%, with 56:44 proportion of train : test data. After applying pre-pruning with minimal gain=5, minimal leaf size=6, min size for split=12, number of prepruning alternatives=3, we get the following confusion matrix.

Split Criterion: Gini Index; Maximum Depth= 5; Confidence=0.35; Threshold=0.15			
Accuracy: 74.77%	True 1	True 0	Class Precision
Pred 1	255	7861	80.70%
Pred 0	50	74	59.68%
Class Recall	83.24%	54.81%	

Once this model was decided, we changed the random seed from values of 1 all the way to 10000, shown in the table below.

Seed	Train Accuracy	Test Accuracy	Difference
1	76.07	75.45	0.62
500	76.25	69.55	6.7
1000	84.46	71.59	12.87
1500	83.04	66.14	16.9
1992	85.5	74.55	10.95
2500	88.39	71.82	16.57
3000	84.11	67.05	17.06
5000	85.54	65.45	20.09
6000	76.96	70.23	6.73
10000	76.61	72.73	3.88

Initially, when the seed is 1, the accuracy for the test data is close to the accuracy for the test data. But as we start increasing the value of the seed, the difference in accuracy also starts increasing. The maximal difference is noticed seed value 5000, after which the difference in accuracy starts decreasing again.
This shows that the model is pretty unstable, this is due to the fact that when we change the seed values, the train accuracy moved from 75% to around 88%, and the test accuracy changed from 65% to 75%. This shows a lot of variation in the model (with the highest difference in the accuracies reaching 20%), and this is why the chosen model is unstable.

Q 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: Predicted Actual Good Bad Good 0 100DM Bad 500DM 0 Use the misclassification costs to assess performance of a chosen model from 3 above. Examine how different cutoff values for classification threshold make a difference – what do you find?

A.4)

Using the misclassification information, we can calculate the Net Profit Predicted by the Model, by using the Formula: NET PROFIT= Predicted True 1's*100 – Predicted False 1's*500.

For the model with the best accuracy, we get a net profit of 258*100 - 65*500= -6700. Clearly, this would not be a good model for Profit.

Accuracy: 74.55%	True 1	True 0	Class Precision
Pred 1	258	65	79.88%
Pred 0	47	70	59.83%
Class Recall	84.59%	51.85%	

However, when we experimented with several Decision Tree Parameters, we found many patterns:

1. Maximum Depth of 4 gives the Highest Profit.

a.

Maximum Depth= 3			
Accuracy: 62.20%	True 1	True 0	Class Precision
Pred 1	187	30	86.18%
Pred 0	159	124	43.82%
Class Recall	54.05%	80.52%	
Net Profit = 187*100-30*500 = **3700**			

b.

Maximum Depth= 4			
Accuracy: 62.40%	True 1	True 0	Class Precision
Pred 1	183	25	87.98%
Pred 0	163	129	44.18%

Class Recall	52.89%	83.77%	
Net Profit = 187*100-30*500 = **5800**			

c.

Maximum Depth= 5			
Accuracy: 65.00%	True 1	True 0	Class Precision
Pred 1	210	39	84.34%
Pred 0	136	115	45.82%
Class Recall	60.69%	74.68%	
Net Profit = 210*100-39*500 = **1500**			

d. Table: Maximum Depth vs Net Profit

Maximum Depth	Net Profit	Accuracy
3	3700	62.20%
4	5800	62.40%
5	1500	65.00%

2.Threshold of 0.1 gives the Best Net Profit Model.
Below is the recorded Observation for different Threshold Value

Table: Threshold vs Net Profit

Criterion = Gini_Index Maximum Depth = 4 Confidence = 0.35	Threshold	Net Profit	Overall Accuracy
	0.5	-13600	72.40%
	0.4	-8600	73.60%
	0.3	-1700	69.00%
	0.2	-900	65.80%
	0.15	1700	63.60%
	0.13	3000	64.00%
	0.1	**5800**	**62.40%**
	0.05	-1100	33.40%

We observe here that the Threshold can be altered to get better Cost Performances, the default Threshold of 0.5, gives a very high loss of 13600. As we reduce the threshold value, we notice that the Net Profit increases. It reaches a maxima of 5800 at the range 0.08-0.11.

Hence we can Conclude that 0.1 gives the Maximum Net Profit.
Optimal Model for Maximum Net Profit:
Creterion= Gini_index
Maximum Depth = 4
Confindence=0.35
Threshold=0.1

Accuracy: 74.55%	True 1	True 0	Class Precision
Pred 1	183	25	87.98%
Pred 0	163	129	44.18%
Class Recall	52.89%	83.77%	
Net Profit = 183*100- 25*500 = 5800			

Q.5 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 they similar to what you found in Question 2? (b) Identify two relatively pure leaf nodes. What are the 'probabilities for 'Good' and 'Bad' in these nodes? (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).

A.5

Please find the decision tree that we obtained for our "Best" model.
The confusion matrix for the "Best" model showing the overall accuracy levels and the accuracies of the "Good" and "Bad" cases that we obtained is given below:

Overall Accuracy: 74.77%	**True 1**	**True 0**	**Class Precision**
Pred.1	255	61	80.70%
Pred.0	50	74	59.68%
Class recall	**83.61%**	**54.81%**	

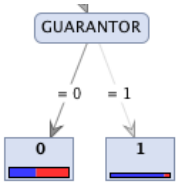
a) The maximal depth for our "Best" decision tree model obtained is 5. It has 41 nodes. The variables towards the top of the tree are as follows:
   1. CHK_ACCT
   2. HISTORY
   3. AMOUNT
   4. OTHER_INSTALL
   5. REAL ESTATE

   They are almost similar, except for REAL ESTATE which replaces DURATION from Question2. The REAL ESTATE variable shows how whether owning a real estate poses as a factor for evaluating the credit-risk of an applicant.

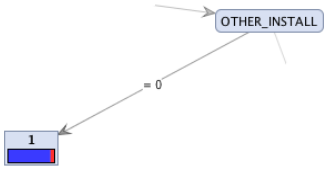
b) The two relatively pure leaf nodes from our "best" decision tree model along with the probabilities for "Good" and "Bad" cases are as follows:

   The following nodes have a higher proportion of 1's.

   - **GUARANTOR**
     Class Frequency of node (1=10, 0=1)
     P(Good Case) =0.909; P(Bad Case) =0.090



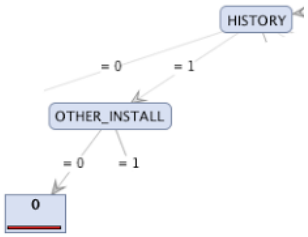
   - OTHER_INSTALL when CHK_ACCT=3
     Class Frequency of node (1=303, 0=27)
     P(Good Case) =0.918; P(Bad Case) =0.081



The following nodes have all the 0's

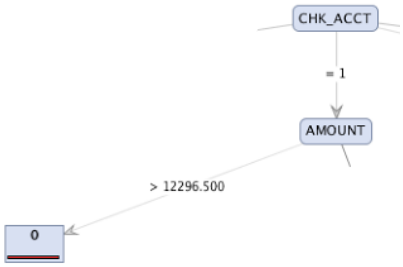
- OTHER_INSTALL when HISTORY=1

Class Frequency of node (1=0, 0=8)
P(Good Case) =1; P(Bad Case) =8



- AMOUNT

Class Frequency of node (1=0, 0=12)
P(Good Case) =0; P(Bad Case) =1



c)

Two sample rules obtained from the tree

1. IF CHK_ACCT = 1 AND AMOUNT > 12296.5 THEN RESPONSE = 0
2. IF CHK_ACCT = 0 AND HISTORY = 1 AND OTHER_INSTALL = 0 THEN RESPONSE =

Q.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.

A.6

The Model with the following Parameters, were developed and the Validation Data was exported to Excel.

Criterion: Gini_Index
Maximum Depth: 4
Confidence:0.35

In Excel, the Data Table was created by Applying the Miscalculation Information provided, in order to find the Optimal Threshold Value. The Cumulative Profit was calculated and the Last Cumulative Profit Values of each Unique Confidence(0) was filter and Tabulated as the following Table:

Confidence (0)	Cumulative Profit
0.000	-1100.00
0.056	-3200.00
0.082	**5800.00**
0.125	3000.00
0.143	1700.00
0.160	-900.00
0.250	-1000.00
0.267	-900.00
0.294	-1700.00
0.323	-8200.00
0.333	-8600.00
0.455	-11800.00
0.500	-13600.00
0.541	-26500.00
0.625	-26800.00
0.667	-27100.00
0.778	-28500.00
0.800	-30800.00
0.818	-31500.00
0.889	-35000.00

A Graph was plotted with Last Cumulative Profit vs Unique Confidence(0).

On comparing the Confidence(0) with the Cumulative Profit, we find that the Maximum Profit is obtained at 0.082 Threshold and the Profit obtained is 5800.

Hence, The Recommended Value of Cut-off would be 0.08 for the Confidence(0).
For the Confidence(1), the cut-off would be 1-0.082= 0.88

accuracy: 62.40%

	true 1	true 0	class precision
pred. 1	183	25	87.98%
pred. 0	163	129	44.18%
class recall	52.89%	83.77%	

On Applying the Calculated Threshold in our Model, We get an Exact Match of our highest Net Profit = 5800.

APPENDX

Q.2 Decision Tree

Tree

```
CHK_ACCT = 0
|  HISTORY = 0
|  |  OWN_RES = 0: 0 {1=0, 0=7}
|  |  OWN_RES = 1: 1 {1=3, 0=3}
|  HISTORY = 1
|  |  OTHER_INSTALL = 0: 0 {1=0, 0=8}
|  |  OTHER_INSTALL = 1
|  |  |  AMOUNT > 1746.500: 1 {1=5, 0=3}
|  |  |  AMOUNT ≤ 1746.500: 0 {1=1, 0=5}
|  HISTORY = 2
|  |  GUARANTOR = 0
|  |  |  INSTALL_RATE > 2.500
|  |  |  |  AGE > 54.500: 1 {1=6, 0=2}
|  |  |  |  AGE ≤ 54.500
|  |  |  |  |  DURATION > 22.500
|  |  |  |  |  |  USED_CAR = 0
|  |  |  |  |  |  |  MALE_SINGLE = 0: 0 {1=0, 0=17}
|  |  |  |  |  |  |  MALE_SINGLE = 1
|  |  |  |  |  |  |  |  AGE > 28.500: 0 {1=2, 0=13}
|  |  |  |  |  |  |  |  AGE ≤ 28.500: 1 {1=4, 0=1}
|  |  |  |  |  |  USED_CAR = 1: 1 {1=4, 0=2}
|  |  |  |  |  DURATION ≤ 22.500
|  |  |  |  |  |  AMOUNT > 1460: 1 {1=13, 0=7}
|  |  |  |  |  |  AMOUNT ≤ 1460
|  |  |  |  |  |  |  AMOUNT > 1209.500: 0 {1=0, 0=8}
|  |  |  |  |  |  |  AMOUNT ≤ 1209.500
|  |  |  |  |  |  |  |  REAL_ESTATE = 0: 0 {1=4, 0=10}
|  |  |  |  |  |  |  |  REAL_ESTATE = 1: 1 {1=5, 0=2}
|  |  |  INSTALL_RATE ≤ 2.500
|  |  |  |  DURATION > 16.500
|  |  |  |  |  AMOUNT > 2762
|  |  |  |  |  |  OWN_RES = 0: 1 {1=7, 0=1}
|  |  |  |  |  |  OWN_RES = 1: 0 {1=7, 0=8}
|  |  |  |  |  AMOUNT ≤ 2762: 0 {1=1, 0=6}
|  |  |  |  DURATION ≤ 16.500: 1 {1=15, 0=4}
|  |  GUARANTOR = 1: 1 {1=10, 0=1}
|  HISTORY = 3: 0 {1=3, 0=9}
|  HISTORY = 4
|  |  DURATION > 31.500
|  |  |  PROP_UNKN_NONE = 0: 1 {1=4, 0=2}
|  |  |  PROP_UNKN_NONE = 1: 0 {1=0, 0=5}
|  |  DURATION ≤ 31.500
|  |  |  DURATION > 11.500
|  |  |  |  NEW_CAR = 0: 1 {1=18, 0=3}
|  |  |  NEW_CAR = 1
|  |  |  |  AMOUNT > 2203: 0 {1=1, 0=5}
|  |  |  |  AMOUNT ≤ 2203: 1 {1=5, 0=2}
|  |  |  DURATION ≤ 11.500: 1 {1=21, 0=1}
CHK_ACCT = 1
|  AMOUNT > 12296.500: 0 {1=0, 0=12}
|  AMOUNT ≤ 12296.500
|  |  SAV_ACCT = 0
|  |  |  DURATION > 22.500
|  |  |  |  PRESENT_RESIDENT = 1: 1 {1=7, 0=3}
|  |  |  |  PRESENT_RESIDENT = 2
|  |  |  |  |  AMOUNT > 5958: 1 {1=5, 0=2}
|  |  |  |  |  AMOUNT ≤ 5958: 0 {1=1, 0=10}
|  |  |  |  PRESENT_RESIDENT = 3: 0 {1=1, 0=5}
|  |  |  |  PRESENT_RESIDENT = 4
|  |  |  |  |  AGE > 24.500: 0 {1=2, 0=13}
|  |  |  |  |  AGE ≤ 24.500: 1 {1=3, 0=2}
|  |  |  DURATION ≤ 22.500
|  |  |  |  EMPLOYMENT = 0: 1 {1=7, 0=1}
|  |  |  |  EMPLOYMENT = 1
|  |  |  |  |  DURATION > 8.500
|  |  |  |  |  |  RENT = 0: 1 {1=5, 0=3}
|  |  |  |  |  |  RENT = 1: 0 {1=1, 0=4}
|  |  |  |  |  DURATION ≤ 8.500: 1 {1=5, 0=0}
|  |  |  |  EMPLOYMENT = 2
|  |  |  |  |  FURNITURE = 0: 1 {1=20, 0=2}
|  |  |  |  |  FURNITURE = 1: 0 {1=2, 0=4}
|  |  |  |  EMPLOYMENT = 3: 1 {1=14, 0=1}
|  |  |  |  EMPLOYMENT = 4
|  |  |  |  |  NUM_DEPENDENTS > 1.500: 0 {1=1, 0=5}
|  |  |  |  |  NUM_DEPENDENTS ≤ 1.500: 1 {1=11, 0=4}
|  |  SAV_ACCT = 1
|  |  |  MALE_SINGLE = 0
|  |  |  |  DURATION > 27: 0 {1=0, 0=5}
|  |  |  |  DURATION ≤ 27
|  |  |  |  |  NEW_CAR = 0: 1 {1=5, 0=3}
|  |  |  |  |  NEW_CAR = 1: 0 {1=1, 0=5}
|  |  |  MALE_SINGLE = 1
|  |  |  |  PROP_UNKN_NONE = 0
|  |  |  |  |  AMOUNT > 6213: 0 {1=2, 0=3}
```

```
| | | | | AMOUNT ≤ 6213: 1 {1=11, 0=1}
| | | | PROP_UNKN_NONE = 1: 0 {1=4, 0=5}
| | SAV_ACCT = 2: 1 {1=8, 0=3}
| | SAV_ACCT = 3
| | | OTHER_INSTALL = 0: 1 {1=8, 0=1}
| | | OTHER_INSTALL = 1: 0 {1=2, 0=3}
| | SAV_ACCT = 4: 1 {1=38, 0=5}
CHK_ACCT = 2
| REAL_ESTATE = 0
| | AMOUNT > 1342: 1 {1=28, 0=2}
| | AMOUNT ≤ 1342
| | | MALE_SINGLE = 0: 1 {1=6, 0=1}
| | | MALE_SINGLE = 1: 0 {1=2, 0=3}
| REAL_ESTATE = 1
| | INSTALL_RATE > 3.500: 0 {1=1, 0=5}
| | INSTALL_RATE ≤ 3.500: 1 {1=12, 0=3}
CHK_ACCT = 3
| OTHER_INSTALL = 0
| | AGE > 30.500: 1 {1=205, 0=9}
| | AGE ≤ 30.500
| | | AMOUNT > 6465.500: 1 {1=5, 0=5}
| | | AMOUNT ≤ 6465.500
| | | | AGE > 22.500
| | | | | INSTALL_RATE > 3.500
| | | | | | DURATION > 19.500: 1 {1=23, 0=2}
| | | | | | DURATION ≤ 19.500
| | | | | | | AMOUNT > 1840.500: 0 {1=2, 0=3}
| | | | | | | AMOUNT ≤ 1840.500: 1 {1=11, 0=2}
| | | | | INSTALL_RATE ≤ 3.500: 1 {1=48, 0=1}
| | | | AGE ≤ 22.500
| | | | | INSTALL_RATE > 2.500: 0 {1=2, 0=4}
| | | | | INSTALL_RATE ≤ 2.500: 1 {1=7, 0=1}
| OTHER_INSTALL = 1
| | PRESENT_RESIDENT = 1: 1 {1=5, 0=1}
| | PRESENT_RESIDENT = 2
| | | RETRAINING = 0
| | | | JOB = 1: 1 {1=4, 0=1}
| | | | JOB = 2
| | | | | HISTORY = 2: 1 {1=4, 0=1}
| | | | | HISTORY = 4: 0 {1=1, 0=4}
| | | | JOB = 3: 1 {1=5, 0=0}
| | | RETRAINING = 1: 0 {1=1, 0=6}
| | PRESENT_RESIDENT = 3: 1 {1=9, 0=1}
| | PRESENT_RESIDENT = 4: 1 {1=16, 0=5}
```

Q.5 Decision tree for "Best" model

Tree

```
CHK_ACCT = 0
|   HISTORY = 0: 0 {1=3, 0=10}
|   HISTORY = 1
|   |   OTHER_INSTALL = 0: 0 {1=0, 0=8}
|   |   OTHER_INSTALL = 1
|   |   |   AMOUNT > 1746.500: 1 {1=5, 0=3}
|   |   |   AMOUNT ≤ 1746.500: 0 {1=1, 0=5}
|   HISTORY = 2
|   |   GUARANTOR = 0: 0 {1=68, 0=81}
|   |   GUARANTOR = 1: 1 {1=10, 0=1}
|   HISTORY = 3: 0 {1=3, 0=9}
|   HISTORY = 4
|   |   DURATION > 31.500: 0 {1=4, 0=7}
|   |   DURATION ≤ 31.500: 1 {1=45, 0=11}
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}
|   |   SAV_ACCT = 1
|   |   |   MALE_SINGLE = 0: 0 {1=6, 0=13}
|   |   |   MALE_SINGLE = 1: 1 {1=17, 0=9}
|   |   SAV_ACCT = 2: 1 {1=8, 0=3}
|   |   SAV_ACCT = 3: 1 {1=10, 0=4}
```

```
|   |   SAV_ACCT = 4: 1 {1=38, 0=5}
CHK_ACCT = 2
|   REAL_ESTATE = 0: 1 {1=36, 0=6}
|   REAL_ESTATE = 1
|   |   INSTALL_RATE > 3.500: 0 {1=1, 0=5}
|   |   INSTALL_RATE ≤ 3.500: 1 {1=12, 0=3}
CHK_ACCT = 3
|   OTHER_INSTALL = 0: 1 {1=303, 0=27}
|   OTHER_INSTALL = 1
|   |   PRESENT_RESIDENT = 1: 1 {1=5, 0=1}
|   |   PRESENT_RESIDENT = 2
|   |   |   RETRAINING = 0: 1 {1=14, 0=6}
|   |   |   RETRAINING = 1: 0 {1=1, 0=6}
|   |   PRESENT_RESIDENT = 3: 1 {1=9, 0=1}
|   |   PRESENT_RESIDENT = 4: 1 {1=16, 0=5}
```

Q.3 W-J48

W-J48

J48 pruned tree

```
CHK_ACCT = 0
|   GUARANTOR = 0
|   |   SAV_ACCT = 4
|   |   |   NUM_CREDITS <= 1
|   |   |   |   TELEPHONE = 1: 1 (4.0/1.0)
|   |   |   |   TELEPHONE = 0: 0 (9.0/1.0)
|   |   |   NUM_CREDITS > 1: 1 (3.0)
|   |   SAV_ACCT = 0
|   |   |   EDUCATION = 0
|   |   |   |   HISTORY = 4
|   |   |   |   |   DURATION <= 45
|   |   |   |   |   |   CO-APPLICANT = 0: 1 (22.0/1.0)
|   |   |   |   |   |   CO-APPLICANT = 1
|   |   |   |   |   |   |   AMOUNT <= 3416: 1 (2.0)
|   |   |   |   |   |   |   AMOUNT > 3416: 0 (2.0)
|   |   |   |   |   DURATION > 45: 0 (2.0)
|   |   |   |   HISTORY = 2
|   |   |   |   |   EMPLOYMENT = 4
|   |   |   |   |   |   MALE_SINGLE = 1: 0 (6.0)
|   |   |   |   |   |   MALE_SINGLE = 0: 1 (3.0/1.0)
|   |   |   |   |   EMPLOYMENT = 2
|   |   |   |   |   |   NUM_CREDITS <= 1
|   |   |   |   |   |   |   DURATION <= 30
|   |   |   |   |   |   |   |   TELEPHONE = 1: 0 (3.0/1.0)
|   |   |   |   |   |   |   |   TELEPHONE = 0
|   |   |   |   |   |   |   |   |   AGE <= 22: 0 (2.0)
|   |   |   |   |   |   |   |   |   AGE > 22: 1 (15.0/2.0)
|   |   |   |   |   |   |   DURATION > 30: 0 (3.0)
|   |   |   |   |   |   NUM_CREDITS > 1: 0 (2.0)
|   |   |   |   |   EMPLOYMENT = 3
|   |   |   |   |   |   USED_CAR = 0: 0 (6.0/1.0)
|   |   |   |   |   |   USED_CAR = 1: 1 (2.0)
```

```
|   |   |   |   |   EMPLOYMENT = 0: 1 (5.0)
|   |   |   |   |   EMPLOYMENT = 1
|   |   |   |   |   |   OTHER_INSTALL = 0: 0 (14.0/3.0)
|   |   |   |   |   |   OTHER_INSTALL = 1: 1 (2.0)
|   |   |   |   HISTORY = 3
|   |   |   |   |   DURATION <= 18: 1 (2.0)
|   |   |   |   |   DURATION > 18: 0 (2.0)
|   |   |   |   HISTORY = 0: 0 (7.0/1.0)
|   |   |   |   HISTORY = 1: 0 (9.0/2.0)
|   |   |   EDUCATION = 1: 0 (7.0/1.0)
|   |   SAV_ACCT = 2: 1 (2.0)
|   |   SAV_ACCT = 3: 1 (6.0)
|   |   SAV_ACCT = 1
|   |   |   NEW_CAR = 0: 0 (3.0)
|   |   |   NEW_CAR = 1: 1 (2.0)
|   GUARANTOR = 1
|   |   NEW_CAR = 0: 1 (10.0)
|   |   NEW_CAR = 1: 0 (2.0)
CHK_ACCT = 1
|   AMOUNT <= 12169
|   |   MALE_SINGLE = 1: 1 (67.0/17.0)
|   |   MALE_SINGLE = 0
|   |   |   MALE_DIV = 0
|   |   |   |   REAL_ESTATE = 1: 1 (24.0/4.0)
|   |   |   |   REAL_ESTATE = 0
|   |   |   |   |   JOB = 2
|   |   |   |   |   |   RADIO/TV = 1: 0 (7.0/2.0)
|   |   |   |   |   |   RADIO/TV = 0
|   |   |   |   |   |   |   DURATION <= 15: 0 (8.0/1.0)
|   |   |   |   |   |   |   DURATION > 15: 1 (14.0/1.0)
|   |   |   |   |   JOB = 1: 0 (7.0/2.0)
|   |   |   |   |   JOB = 3: 0 (10.0/2.0)
|   |   |   |   |   JOB = 0: 1 (1.0)
|   |   |   MALE_DIV = 1: 0 (7.0/3.0)
|   AMOUNT > 12169: 0 (8.0)
CHK_ACCT = 3: 1 (212.0/27.0)
CHK_ACCT = 2: 1 (36.0/6.0)

Number of Leaves  :     42

Size of the tree :      70
```