Data Mining Assignment 1
Decision trees

Submitted by-

Jaideep Adusumelli
Kaushik Kompella
Sushaanth Srirangapathi

Q1. 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?)

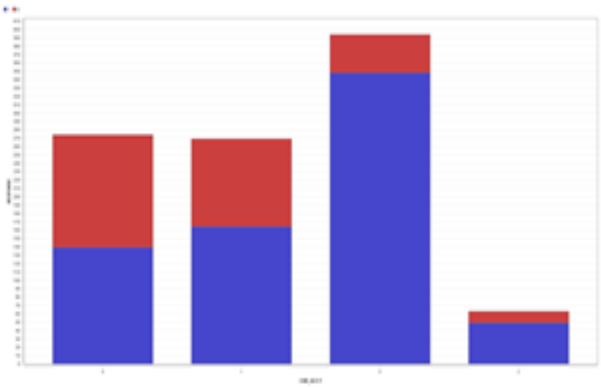
Proportion of "Good" to "Bad" cases is 70:30 for the given 1000 records. We think that, if the number of bed cases should have been more, that would have been very helpful in generating a more reliable and flexible model. Due to less number of bad cases provided, The highest accuracies that could be reached are limited to less than 80%.

We analyzed all the attributes given and found the following:

Checking Account:

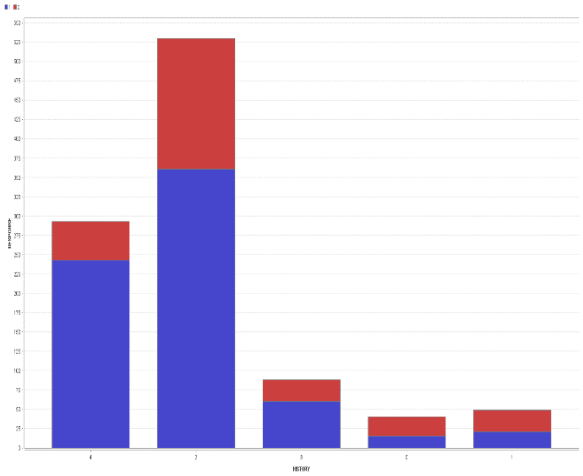
Checking Account Type	Total Count	Bad Credit Score	Good Credit Score	Ratio of Bad to Good Creditors
0	274	135	139	0.971
1	269	105	164	0.640
2	63	14	49	0.286
3	394	46	348	0.132

From the above table it's clear that most of the defaulters are the ones' who have a type "0" or type "1" checking accounts.



Credit History:

Credit History	Total Count	Bad Credit Score	Good Credit Score	Ratio of Bad to Good Creditors
0	40	25	15	1.667
1	49	28	21	1.333
2	530	169	361	0.468
3	88	28	60	0.467
4	293	50	243	0.206



From the above table we could easily figure out that there were a large number of bad creditors from the groups “0” and “1”. So this is one of the important attributes to classify the given data.

Duration:

Duration	Total Count	Bad Credit Score	Good Credit Score	Ratio of Bad to Good Creditors
0-6	82	9	73	0.123
7-12	277	67	210	0.319
13-18	187	56	131	0.427
19-24	224	66	158	0.418
25-30	57	19	38	0.5
31-36	86	38	48	0.792
37-42	17	5	12	0.417
43-48	54	32	22	1.455
49-54	2	1	1	1.000
55-60	13	6	7	0.857
61-66	0	0	0	0
67-72	1	1	0	0

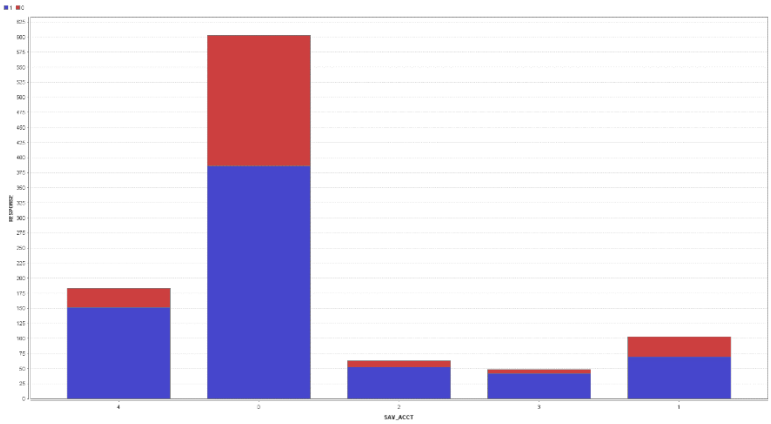
Average duration of credit: 20.903

Standard deviation of duration: 12.053

From the above data we can see that as the duration of credit increases, the ratio of bad creditors also increases, which makes this variable quite interesting. We will try to use this trend in further question.

Savings account:

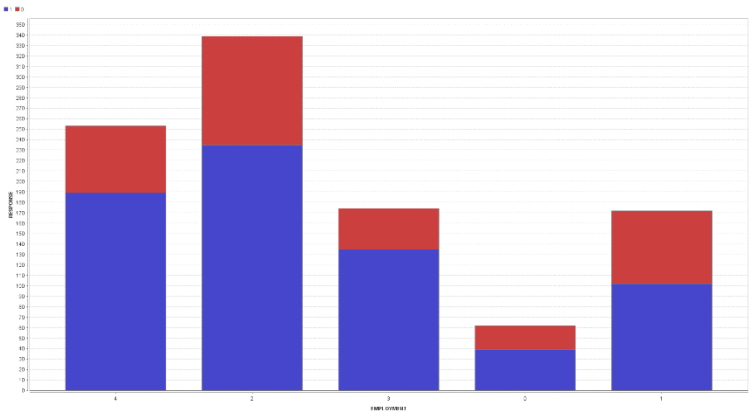
Savings account type	Total Count	Bad Credit Score	Good Credit Score	Ratio of Bad to Good Creditors
0	603	217	386	0.562
1	103	34	69	0.493
2	63	11	52	0.212
3	48	6	42	0.143
4	183	32	151	0.212



This attribute doesn't state anything strongly, but it weakly states that savings accounts "0" and "1" have more number of defaulters than the other accounts.

Employment:

Employment	Total Count	Bad Credit Score	Good Credit Score	Ratio of Bad to Good Credits
0	62	23	39	0.58974
1	172	70	102	0.68627
2	339	104	235	0.44255
3	174	39	135	0.28889
4	253	64	189	0.33862



Employment weakly forms a trend with the ratio of bad creditors as seen from the ratio of "0" and "1" which have a higher ratio of bad creditors than any other class.

Guarantor:

Guarantor	Total Count	Good Credit Count	Bad Credit Count	Ratio of bad to good credit count
0	948	658	290	0.44073
1	52	42	10	0.23810

From the above table it is almost clear that the people who have a guarantor are less likely to commit a fraud than those who didn't have a guarantor. But the relation is too weak to infer anything.

Remaining all the attributes are very weakly or not related to the credit scores of the people.

Q2. 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)?

Decision Tree generated without selecting any attributes:

Accuracy of the tree obtained on **Training dataset** using all attributes before pruning: 95.40%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	673	19	97.25%
Predicted Bad	27	281	91.23%
Class recall	96.14%	93.67%	

Accuracy of the tree obtained on **Training dataset** using all attributes after pruning: 92.50%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	661	36	94.84%
Predicted Bad	39	264	87.13%
Class recall	96.14%	93.67%	

Even though it has a very high accuracy but this model is not a reliable model as it is not flexible and since all the data is used it has also used the attributes which doesn't show any trend for credit rating, i.e. there is an overfit of data.

Hence this model is not a reliable model.

The variables that we have considered for generating this model based on the above analysis and the pre-analysis we did in the first question are:

1. Checking Account
2. History
3. Duration
4. Age
5. Present Resident
6. Job
7. Savings Account
8. Amount
9. Employment

Accuracy of the tree obtained on **Training dataset** using selected attributes before pruning: 85.90%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	645	86	88.24%
Predicted Bad	55	214	79.55%
Class recall	92.14%	71.33%	

Accuracy of the tree obtained on **Training dataset** using selected attributes after pruning: 84.60%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	637	91	87.50%
Predicted Bad	63	209	76.84%
Class recall	91.00%	69.67%	

As you can see the accuracy of data increases as we apply pruning that was because of the overfit that was caused due to the formation of the full tree and formation of sets with very few data points. Pruning generates a smooth curve which will be flexible.

This model has a good accuracy, but this model will be highly sensitive to the unseen data hence this is not a very robust model.

Q3. 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)?

Performance of model with **training data:** 95.20%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	348	17	95.34%
Predicted Bad	7	128	94.81%
Class recall	98.03%	88.28%	

Performance of model with **Validation data:** 72.20%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	288	82	77.84%
Predicted Bad	57	73	56.15%
Class recall	83.48%	47.10%	

This model doesn't seem to be very reliable due to the following reasons:

1. Accuracy of training data is very high, but doesn't hold the same accuracy for the validation data set, due to overfit of data.
2. The precision and recall for the prediction of good creditors is high, but as seen for the above data the model has a very poor precision and recall for the prediction of bad creditors.
3. Only 50% of the data was used to build the model, i.e. very less data about the bad creditors was available to build the model, which is not advisable.

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 and test partitions?

For data split in 70-30 ratio:

Accuracy of the model with the **training data set**: 87.00%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	426	35	92.41%
Predicted Bad	56	183	76.57%
Class recall	88.38%	83.94%	

Accuracy of the model with the **Validation data set**: 75.00%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	181	38	82.65%
Predicted Bad	37	44	54.32%
Class recall	83.03%	53.66%	

For data split in 80-20 ratio:

Accuracy of model with the **training dataset**: 86.25%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	508	63	88.97%
Predicted Bad	47	182	79.48%
Class recall	91.53%	74.29%	

Accuracy of model with the **Validation dataset**: 74.50%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	117	23	83.57%
Predicted Bad	28	32	53.33%
Class recall	80.69%	58.18%	

I would prefer the model built by splitting the data in 80-20 ratio due to the following reasons:

1. More amount of training data always gives you a more reliable model.
2. The precision is also better for the mode with 80% of training data than others. i.e. the class precision of good creditors has increased from 50-50 model to 80-20 model without compromising on the accuracy of prediction of bad creditors.

   [PS. Very less data was available for bad creditors, hence more data needs to be provided in order to increase the accuracy of predicting the bad creditors from an unlabeled data.]

3. The accuracy of this model is better than other models shown above.

Effect of various variables on the accuracy of the tree:

1. Minimal Gain:

   The optimal value of minimal gain was found to be at 1.

Minimal Gain	Performance on validation data

0.05	71.50%
0.1	67.00%
1.0	72.50%

2. Minimal Leaf size:

Minimal Leaf Size	Performance on validation data
2	75.00%
10	71.25%

3. Tree Depth

   The optimal tree depth is optimal at 6 as used in the models we generated.

Tree depth	Performance on validation data
5	72.5%
12	70.05%
25	69.5%

The parameter values have to be changed for each split ratio specially the tree depth.

The parameter values mostly used by us are:

Criterion: Information Gain

Confidence: 0.35

Tree depth: 6

Minimal Leaf size: 2

Minimal gain: 1.0

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?

For J-48 decision tree:

Performance for the model with the **training dataset**: 83.20%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	331	70	82.54%
Predicted Bad	14	85	85.86%
Class recall	95.94%	54.84%	

Performance of the model with the **validation dataset**: 74.80%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	315	86	78.55%
Predicted Bad	40	59	59.60%
Class recall	88.73%	40.69%	

For W-SimpleCART decision tree:

Performance of the model with the **Training Dataset**: 81.40%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	321	69	82.31%
Predicted Bad	24	86	78.18%
Class recall	93.04%	55.48%	

Performance of the model with the **validation Dataset**: 76.00%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	321	86	78.87%
Predicted Bad	34	59	63.44%
Class recall	90.42%	40.69%	

Comparison of performances:

Learner Type	Performance on the validation dataset
Decision Tree	75.20%
J-48 Decision Tree	74.00%
W-Simple CART Decision Tree	76.00%

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?

We changed the value of random seed keeping other parameters constant for the process for dataset split in the ratio of 80:20 for training and validation respectively and obtained the following results:

1. For **decision tree**:

Random seed value	Performance on the validation dataset
2001	70.00%
3500	67.50%
4600	76.00%
5500	73.25%

2. For **J-48 decision tree**:

Random seed value	Performance on the validation dataset
2001	71.50 %
3500	65.50%
4600	73.00 %
5500	75.00 %

3. For **W-simple Cart decision tree**:

Random seed value	Performance on the validation dataset
2001	66.50 %
3500	64.00%
4600	72.50%
5500	73.00%

In W-simple CART decision tree model we can see that the performance changed drastically on changing the random seed, whereas in J-48 and normal decision tree models there not much performance variation. This means they are quite stable than W-Simple CART decision tree model. But these changes are high enough to be taken into consideration.

The attributes at the top of the tree are the same for all the seed values for all types of decision trees, which shows that even though there is a variation in performance of the tree, the decision tree is quite stable.

Q4. 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

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?

The misclassification costs of the chosen model (decision tree with 80:20 split ratio) is:

Misclassification cost on training data	Misclassification cost on validation data
37.375DM	83DM

This value of misclassification changes as we change the threshold values, as changing the threshold value changes the way in which the good and the bad creditors are predicted.

Threshold value	Misclassification cost on validation data
0.5	83DM
0.75	47.5DM
0.25	147.5DM

Q5. 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)

Our best tree is based on the information gain criterion. As this criterion makes sure that the resulting leaf nodes have less entropy than the mother leaf, this model was found to be the best one by us.

Tree Depth: 6

Confidence: 0.35

Number of nodes: 76

Variables towards the top of the tree:

Branch 1:

Checking account – Depth 0

History – depth 1

Savings account – depth 2

Job – depth 3

Employment – depth 4

Branch 2:

Checking Account – Depth 0

History – Depth 1

Duration – Depth 2

Amount – depth 3

Branch 3:

Checking account – Depth 0

Amount – Depth 1

Saving amount – Depth 2

Duration – depth 3

Present resident – depth 4

Similarity to the best tree found in question 2:

The attributes towards the top of the tree were the same as the attributes listed above like:

The checking account, duration, amount, history, and the savings account.

The two relatively pure nodes are –

```
CHK_ACCT = 0
|   HISTORY = 4
|   |   DURATION ≤ 31.500
|   |   |   EMPLOYMENT = 1: 1 (65%)

CHK_ACCT = 1
|   AMOUNT > 12296.500: 0 (25%)
```

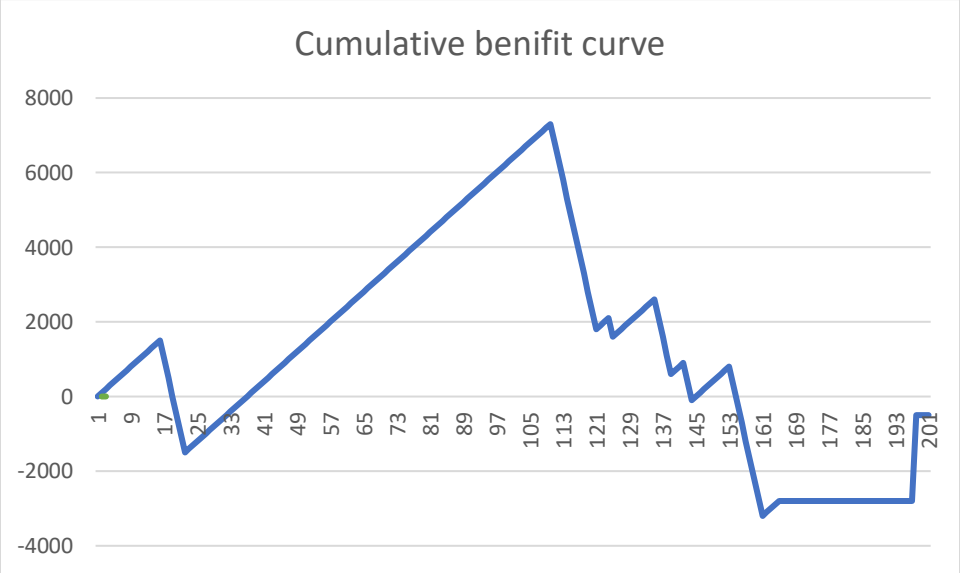
A few sample rules obtained from the tree are:

```
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 2 and EMPLOYMENT =
0 then 1  (2 / 1)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 2 and EMPLOYMENT =
1 then 1  (8 / 8)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 2 and EMPLOYMENT =
2 then 0  (7 / 13)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 2 and EMPLOYMENT =
3 then 0  (3 / 11)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 2 and EMPLOYMENT =
4 then 1  (4 / 4)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 3 and EMPLOYMENT =
0 then 1  (4 / 1)
if CHK_ACCT = 0 and HISTORY = 2 and SAV_ACCT = 0 and JOB = 3 and EMPLOYMENT =
1 then 1  (1 / 1)
```

Q6. 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.



As you can see in the above curve the maximum benefit (peak value) reached from the validation is at confidence level (1) =0.86 amounting to 7800DM but the other cases of same confidence level in the same seed will also be considered, hence the value (net benefit) at this level is 7800DM.

Considering the second highest peak 5500DM at confidence (1) level = 0.81 the total benefit is 2600DM. In case of net benefit, since the first peak is higher, we should consider its validation cut-off.

To score future credit applicants, confidence level (1) = cut-off 0. 86 value is recommended as after this point the chances of incurring a loss is more (by choosing a bad credit as good credit and good credit as a bad credit).

Appendix – 1

Model tested for split ratio 60:40 for training and validation datasets respectively:

Accuracy of the tree obtained on **Training dataset** using selected attributes before pruning: 84.33%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	415	88	82.50%
Predicted Bad	6	91	93.81%
Class recall	98.57%	50.84%	

Accuracy of the tree obtained on **Training dataset** using selected attributes after pruning: 70.75%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	246	84	74.55%
Predicted Bad	33	37	52.86%
Class recall	88.17%	30.58%	

This model is not robust and reliable because, it may have a high accuracy but the actual bads that are predicted to be good are very high and this may incur a very high loss to the organization as per the scenario given in question number 4.

Model tested for split ratio 45:55 for training and validation datasets respectively:

Accuracy of the tree obtained on **Training dataset** using selected attributes before pruning: 86.00%

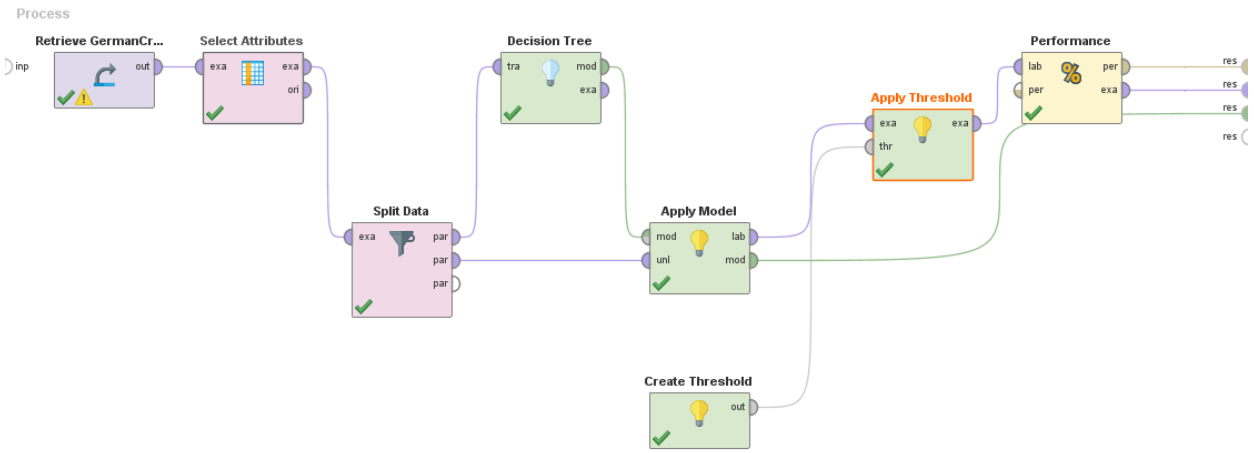
Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	313	59	84.14%
Predicted Bad	4	74	94.87%
Class recall	98.74%	55.64%	

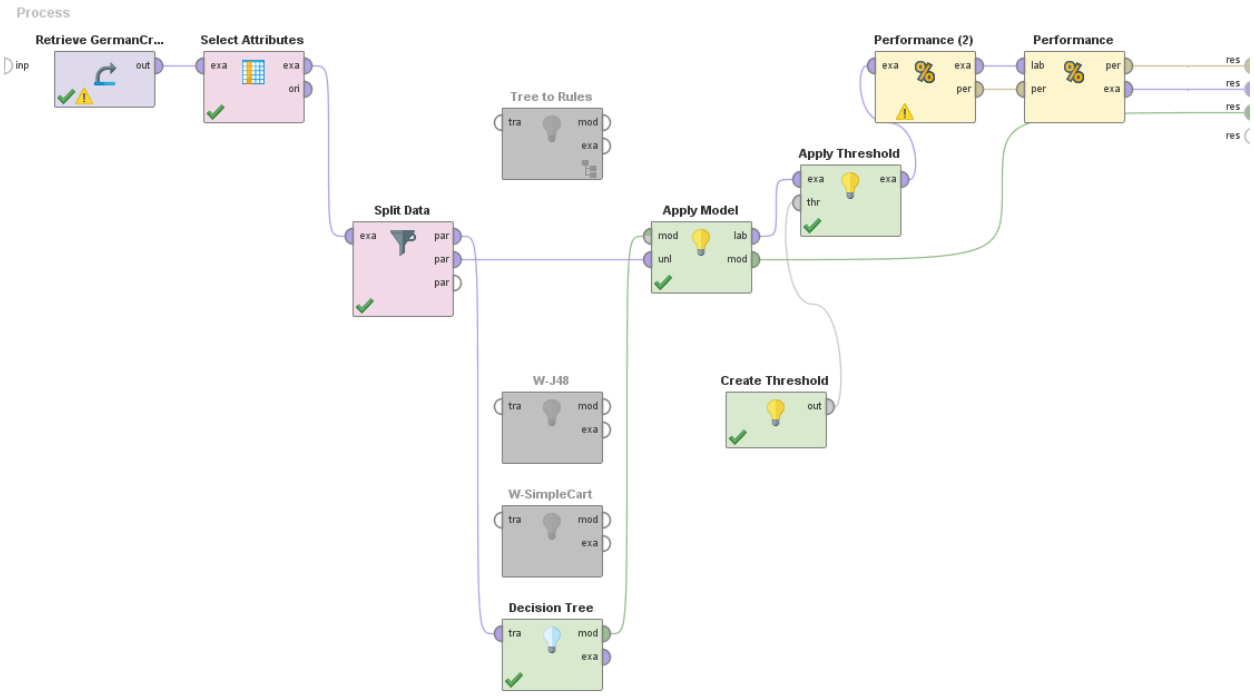
Accuracy of the tree obtained on **Training dataset** using selected attributes after pruning: 68.55%

Prediction / Actual	Actual Good	Actual Bad	Class precision
Predicted Good	325	115	73.86%
Predicted Bad	58	52	47.27%
Class recall	84.86%	31.14%	

Appendix 2

Process diagram:





Appendix 3

Decision tree Obtained:

```
CHK_ACCT = 0
|   HISTORY = 0
|   |   PRESENT_RESIDENT = 1: 1 {1=1, 0=0}
|   |   PRESENT_RESIDENT = 3
|   |   |   SAV_ACCT = 0: 0 {1=0, 0=1}
|   |   |   SAV_ACCT = 4: 1 {1=1, 0=0}
|   |   PRESENT_RESIDENT = 4: 0 {1=0, 0=6}
|   HISTORY = 1
|   |   AGE > 23
|   |   |   SAV_ACCT = 0: 0 {1=1, 0=11}
|   |   |   SAV_ACCT = 1: 1 {1=1, 0=0}
|   |   |   SAV_ACCT = 2: 0 {1=0, 0=1}
|   |   |   SAV_ACCT = 4: 0 {1=0, 0=2}
|   |   AGE ≤ 23: 1 {1=2, 0=0}
|   HISTORY = 2
|   |   SAV_ACCT = 0
|   |   |   DURATION > 16.500: 0 {1=20, 0=39}
|   |   |   DURATION ≤ 16.500
|   |   |   |   EMPLOYMENT = 0: 1 {1=3, 0=0}
|   |   |   |   EMPLOYMENT = 1: 1 {1=4, 0=4}
|   |   |   |   EMPLOYMENT = 2: 1 {1=12, 0=5}
|   |   |   |   EMPLOYMENT = 3: 0 {1=1, 0=3}
|   |   |   |   EMPLOYMENT = 4: 0 {1=2, 0=5}
|   |   SAV_ACCT = 1
|   |   |   EMPLOYMENT = 0: 0 {1=0, 0=1}
|   |   |   EMPLOYMENT = 1: 0 {1=0, 0=1}
|   |   |   EMPLOYMENT = 2
|   |   |   |   PRESENT_RESIDENT = 2: 0 {1=0, 0=1}
|   |   |   |   PRESENT_RESIDENT = 4: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 3: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 4: 1 {1=1, 0=0}
|   |   SAV_ACCT = 2: 1 {1=2, 0=0}
|   |   SAV_ACCT = 3: 1 {1=4, 0=0}
|   |   SAV_ACCT = 4
|   |   |   EMPLOYMENT = 1: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 2: 0 {1=1, 0=4}
|   |   |   EMPLOYMENT = 3
|   |   |   |   DURATION > 22: 0 {1=0, 0=2}
|   |   |   |   DURATION ≤ 22: 1 {1=2, 0=0}
|   |   |   EMPLOYMENT = 4: 0 {1=1, 0=3}
|   HISTORY = 3: 0 {1=3, 0=9}
|   HISTORY = 4
|   |   DURATION > 11
|   |   |   JOB = 0
|   |   |   |   DURATION > 30: 1 {1=1, 0=0}
|   |   |   |   DURATION ≤ 30: 0 {1=0, 0=1}
|   |   |   JOB = 1
|   |   |   |   EMPLOYMENT = 2: 1 {1=2, 0=0}
|   |   |   |   EMPLOYMENT = 3: 1 {1=1, 0=0}
|   |   |   |   EMPLOYMENT = 4: 0 {1=0, 0=2}
|   |   |   JOB = 2
|   |   |   |   AMOUNT > 2122.500: 0 {1=3, 0=8}
|   |   |   |   AMOUNT ≤ 2122.500: 1 {1=8, 0=0}
```

```
|   |   |   JOB = 3: 1 {1=8, 0=0}
|   |   DURATION ≤ 11: 1 {1=14, 0=0}
CHK_ACCT = 1
|   AMOUNT > 12296.500: 0 {1=0, 0=9}
|   AMOUNT ≤ 12296.500
|   |   SAV_ACCT = 0
|   |   |   DURATION > 20.500
|   |   |   |   PRESENT_RESIDENT = 1: 1 {1=5, 0=3}
|   |   |   |   PRESENT_RESIDENT = 2: 0 {1=5, 0=9}
|   |   |   |   PRESENT_RESIDENT = 3: 0 {1=0, 0=3}
|   |   |   |   PRESENT_RESIDENT = 4: 0 {1=6, 0=13}
|   |   |   DURATION ≤ 20.500: 1 {1=48, 0=20}
|   |   SAV_ACCT = 1
|   |   |   EMPLOYMENT = 0: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 1: 0 {1=3, 0=8}
|   |   |   EMPLOYMENT = 2: 0 {1=1, 0=6}
|   |   |   EMPLOYMENT = 3: 1 {1=8, 0=3}
|   |   |   EMPLOYMENT = 4: 1 {1=5, 0=1}
|   |   SAV_ACCT = 2
|   |   |   JOB = 0: 1 {1=1, 0=0}
|   |   |   JOB = 1
|   |   |   |   DURATION > 10.500: 0 {1=0, 0=2}
|   |   |   |   DURATION ≤ 10.500: 1 {1=1, 0=0}
|   |   |   JOB = 2: 1 {1=5, 0=1}
|   |   |   JOB = 3: 1 {1=1, 0=0}
|   |   SAV_ACCT = 3
|   |   |   DURATION > 10: 1 {1=7, 0=1}
|   |   |   DURATION ≤ 10: 0 {1=0, 0=2}
|   |   SAV_ACCT = 4
|   |   |   PRESENT_RESIDENT = 1
|   |   |   |   EMPLOYMENT = 0: 0 {1=0, 0=1}
|   |   |   |   EMPLOYMENT = 1: 1 {1=1, 0=0}
|   |   |   |   EMPLOYMENT = 2: 0 {1=0, 0=1}
|   |   |   |   EMPLOYMENT = 3: 1 {1=2, 0=0}
|   |   |   PRESENT_RESIDENT = 2
|   |   |   |   AMOUNT > 1295.500: 1 {1=11, 0=1}
|   |   |   |   AMOUNT ≤ 1295.500: 0 {1=0, 0=1}
|   |   |   PRESENT_RESIDENT = 3
|   |   |   |   EMPLOYMENT = 1: 0 {1=0, 0=1}
|   |   |   |   EMPLOYMENT = 3: 1 {1=1, 0=0}
|   |   |   PRESENT_RESIDENT = 4: 1 {1=14, 0=0}
CHK_ACCT = 2
|   AMOUNT > 2878
|   |   PRESENT_RESIDENT = 1: 1 {1=5, 0=0}
|   |   PRESENT_RESIDENT = 2
|   |   |   EMPLOYMENT = 2
|   |   |   |   AMOUNT > 4061.500: 0 {1=0, 0=1}
|   |   |   |   AMOUNT ≤ 4061.500: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 3: 1 {1=1, 0=0}
|   |   |   EMPLOYMENT = 4: 1 {1=1, 0=0}
|   |   PRESENT_RESIDENT = 3: 1 {1=1, 0=0}
|   |   PRESENT_RESIDENT = 4: 1 {1=6, 0=0}
|   AMOUNT ≤ 2878
|   |   DURATION > 7.500
|   |   |   JOB = 0
|   |   |   |   HISTORY = 1: 0 {1=0, 0=1}
|   |   |   |   HISTORY = 4: 1 {1=1, 0=0}
```

```
|   |   |   JOB = 1
|   |   |   |   AGE > 40.500: 1 {1=4, 0=1}
|   |   |   |   AGE ≤ 40.500: 0 {1=0, 0=4}
|   |   |   JOB = 2: 1 {1=11, 0=3}
|   |   |   JOB = 3
|   |   |   |   HISTORY = 0: 0 {1=0, 0=1}
|   |   |   |   HISTORY = 2: 1 {1=3, 0=0}
|   |   |   |   HISTORY = 3: 0 {1=0, 0=1}
|   |   DURATION ≤ 7.500: 1 {1=5, 0=0}
CHK_ACCT = 3
|   EMPLOYMENT = 0
|   |   DURATION > 27: 0 {1=0, 0=2}
|   |   DURATION ≤ 27
|   |   |   AMOUNT > 552.500: 1 {1=11, 0=1}
|   |   |   AMOUNT ≤ 552.500: 0 {1=0, 0=1}
|   EMPLOYMENT = 1
|   |   AMOUNT > 5523
|   |   |   HISTORY = 1: 1 {1=1, 0=0}
|   |   |   HISTORY = 2: 0 {1=0, 0=1}
|   |   |   HISTORY = 3: 0 {1=0, 0=2}
|   |   |   HISTORY = 4: 0 {1=0, 0=1}
|   |   AMOUNT ≤ 5523: 1 {1=33, 0=5}
|   EMPLOYMENT = 2
|   |   DURATION > 9.500
|   |   |   DURATION > 43.500
|   |   |   |   SAV_ACCT = 0: 1 {1=2, 0=0}
|   |   |   |   SAV_ACCT = 1: 0 {1=0, 0=1}
|   |   |   |   SAV_ACCT = 2: 0 {1=0, 0=1}
|   |   |   |   SAV_ACCT = 4: 1 {1=1, 0=1}
|   |   |   DURATION ≤ 43.500: 1 {1=75, 0=11}
|   |   DURATION ≤ 9.500: 1 {1=24, 0=0}
|   EMPLOYMENT = 3
|   |   AGE > 22.500: 1 {1=51, 0=1}
|   |   AGE ≤ 22.500
|   |   |   PRESENT_RESIDENT = 1: 0 {1=0, 0=1}
|   |   |   PRESENT_RESIDENT = 2: 0 {1=0, 0=1}
|   |   |   PRESENT_RESIDENT = 3: 1 {1=1, 0=0}
|   |   |   PRESENT_RESIDENT = 4: 1 {1=2, 0=0}
|   EMPLOYMENT = 4
|   |   HISTORY = 0: 1 {1=2, 0=0}
|   |   HISTORY = 1
|   |   |   SAV_ACCT = 0: 0 {1=0, 0=1}
|   |   |   SAV_ACCT = 1: 1 {1=1, 0=0}
|   |   |   SAV_ACCT = 2: 1 {1=1, 0=0}
|   |   HISTORY = 2
|   |   |   AMOUNT > 723: 1 {1=38, 0=2}
|   |   |   AMOUNT ≤ 723: 0 {1=0, 0=1}
|   |   HISTORY = 3
|   |   |   AGE > 46: 0 {1=0, 0=2}
|   |   |   AGE ≤ 46: 1 {1=3, 0=0}
|   |   HISTORY = 4
|   |   |   AGE > 33.500: 1 {1=37, 0=0}
|   |   |   AGE ≤ 33.500
|   |   |   |   SAV_ACCT = 0: 1 {1=4, 0=0}
|   |   |   |   SAV_ACCT = 2: 1 {1=1, 0=0}
|   |   |   |   SAV_ACCT = 4: 0 {1=0, 0=2}
```