IDS 572 Data Mining for Business (Spring 2016)

Assignment No. 1 Solution

Group Details:

Name	UIN
NagaShrikanth Ammanabrolu	676837954
Suresh Sappa	667192596
Sagar Kanchi	669850639

Question no. 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. Anything noteworthy in the data? Which variables do you think will be most relevant for the outcome of interest? (Why?)

Solution:

The given German Credit Dataset is first imported into the Rapidminer platform to perform some initial Exploratory Data Analysis. Understanding the Dataset is the first step in any Data Mining process and moreover, according to the CRISM-DM standards, understanding the data is an essential step before performing analysis on the data.

As per our initial understanding of the imported German Credit Dataset, the data represents cases of 1000 previous credit applicants and the objective is to develop a model to score the credit applicants based on the given factors. For the given 1000 credit applicant cases, there are a total of 30 variables, without considering the Observation Numbers and the Response, which are the ID and Label/Target Attribute respectively.

Proportion of 'Good' and 'Bad' Cases

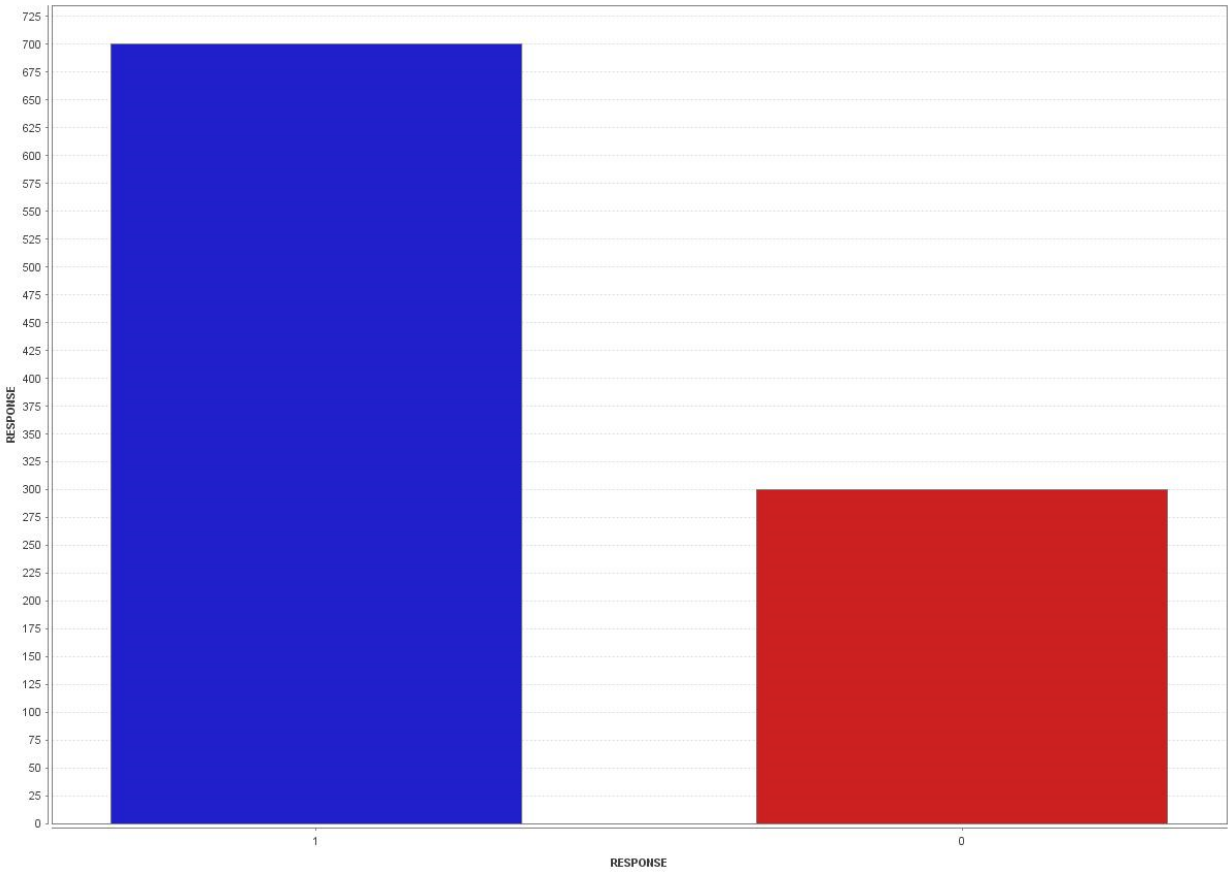
After the dataset is imported into Rapidminer through the 'Read Excel' or the 'Retrieve' repository operators, we can get a better data understanding through the 'Results' tab.

We first obtain the proportion of the 'Good' and 'Bad' cases by heading over to the 'Statistics' tab in the ExampleSet Results window. Here, the 'Good' cases represent the cases of credit applicants who has been approved and the 'Bad' cases represent the cases of credit applicants who have been rejected.

From the results, we conclude the following results tabulated below:

Response Value	Proportion of cases for the Value
0 (Bad Cases)	300
1 (Good Cases)	700

Therefore, we can conclude that the proportion of 'Good' to the 'Bad' cases is of the ratio of 700:300 or 7:3. The findings are summarized below:



Descriptive Analytics of the Dataset

Descriptive statistics form an integral part of Data Exploration and this is where we will obtain different statistical values for the given variables in the dataset. In the given dataset, from the total of the 30 variables, there are a total of 18 binary (of values 0 and 1) variables, 6 categorical (of the values ranging from 0 to 4) variables and 6 numerical (consisting of real and integer numbers) variables.

The descriptive statistics for all of the different types of categorical and numerical variables are described below.

Descriptive Statistics for the Numerical variables:

Role	Variable	Data Type	Observations			
			Minimum	Maximum	Average	Std Deviation
Attribute	Duration	Integer	4	72	20.903	12.059
Attribute	Amount	Integer	250	18424	3271.258	2822.737
Attribute	Install_Rate	Integer	1	4	2.973	1.119
Attribute	Age	Integer	19	75	35.546	11.375
Attribute	Num_Credits	Integer	1	4	1.407	0.578
Attribute	Num_Dependents	Integer	1	2	1.155	0.362

Descriptive Statistics for the Categorical variables:

Role	Variable	Data Type	Observations			
			Min	Max	Frequencies	
Label	Response	Binomial	0	1	0 (300)	1 (700)
Attribute	New_Car	Binomial	0	1	0 (766)	1 (234)
Attribute	Used_Car	Binomial	0	1	0 (897)	1 (103)
Attribute	Furniture	Binomial	0	1	0 (819)	1 (181)
Attribute	Radio/TV	Binomial	0	1	0 (950)	1 (50)
Attribute	Education	Binomial	0	1	0 (300)	1 (700)
Attribute	Retraining	Binomial	0	1	0 (903)	1 (97)
Attribute	Male_Div	Binomial	0	1	0 (950)	1 (50)
Attribute	Male_Single	Binomial	0	1	0 (452)	1 (548)
Attribute	Male_Mar_or_Wid	Binomial	0	1	0 (908)	1 (92)
Attribute	Co_Applicant	Binomial	0	1	0 (959)	1 (41)
Attribute	Guarantor	Binomial	0	1	0 (948)	1 (52)
Attribute	Real_Estate	Binomial	0	1	0 (718)	1 (282)
Attribute	Prop_Unkn_None	Binomial	0	1	0 (846)	1 (154)
Attribute	Other_Install	Binomial	0	1	0 (814)	1 (186)
Attribute	Rent	Binomial	0	1	0 (821)	1 (179)
Attribute	Own_Res	Binomial	0	1	0 (287)	1 (713)
Attribute	Telephone	Binomial	0	1	0 (596)	1 (404)
Attribute	Foreign	Binomial	0	1	0 (963)	1 (37)

Descriptive Statistics for the Categorical variables (Continued):

Role	Variable	Data Type	Observations		
			Min	Max	Frequencies
Attribute	Chk_Account	Polynomial	0	3	0 (274), 1 (269), 2 (63), 3 (394)
Attribute	History	Polynomial	0	4	0 (40), 1 (49), 2 (530), 3 (88), 4 (293)
Attribute	Sav_Acct	Polynomial	0	4	0 (603), 1 (103), 2 (63), 3 (48), 4 (183)
Attribute	Employment	Polynomial	0	4	0 (62), 1 (172), 2 (339), 3 (174), 4 (253)
Attribute	Present_Resident	Polynomial	1	4	1 (130), 2 (308), 3 (149), 4 (413)
Attribute	Job	Polynomial	0	3	0 (22), 1 (200), 2 (630), 3 (148)

Noteworthy findings in the Dataset

For the total of 30 variables, excluding the Observation Numbers and the Response columns, below are the noteworthy findings:

1. 88.3% of the credit applicants without a checking account had a good credit rating
2. 59.85% of the credit applicants with good credit ratings have a credit duration in the range from 11 months and 1 week to 24 months.
3. 55% of the credit applicants with a good credit rating have no Deutsche Marks in their Savings account
4. 75.87% of the credit applicants with a good credit rating have an employment period of 4 or more years
5. Credit applicants between the ages 22-37 feature are the most likely (419 cases) ones to feature a good credit rating
6. 75% of the credit applicants who own a residence also feature good credit rating

Variables most likely to be of interest for predicting the outcome of 'Response'

1. Checking account status: This variable/entry allows us to get to know the account status of each applicant. This can be used as one of the best variable

to differentiate the good and bad cases. To support the statement provided, the data set shows that 85% of the applicants with no checking account have good credit ratings.

2. Credit history: Credit score shows whether the applicant have a history of financial stability and responsible credit management. The most important component of credit score looks at whether he/he can be trusted to repay money that is lent to him/her. The second-most important component of your credit score is how much he/she owe. A long history late payments and other negative items is always helpful in considering the applicant as a positive case.

3. Employment and nature of job: Source of income, stability of income and quality of income are important factors that decide the stability of income. For example, a salaried person with a reputed company can manage a higher loan amount as against a self-employed person purely due to a higher weightage on the stability of his income.

4. Age: Age is one of the important factors for the consideration of a loan application. Usually, financial institutions try to restrict the home loan tenure to a person's retirement age. Therefore, younger people can get a loan with a tenure of 20 to 25 years easily, while older people (50 years and above) find it difficult to get a longer term loan. In many cases, the bank does not give a loan to a single applicant above 50 years of age, or reduces the tenure based on other factors.

(Please Refer 'Appendix' for supporting information)

Question no. 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?)

Solution.

Obtaining the Decision Tree based on a Descriptive Model

For obtaining a decision tree based on a descriptive model of the given dataset, a decision tree is formed on the entire data. Here, for the descriptive model, splitting the dataset into Training, Validation or Testing sets are not required.

A decision tree is obtained on the entire available dataset for obtaining a descriptive model, which is not meant for the purposes of predicting future credit applicants.

Variables used to differentiate 'Good' from 'Bad' cases

Below listed are the list of Variables obtained from the Decision Tree obtained as mentioned previously, to differentiate the 'Good' from the 'Bad' cases:

1. Checking Account
2. Amount
3. Duration
4. Number of Dependents
5. Real Estate
6. Installment Rate
7. Other Installment Plan Credit
8. Employment
9. Age
10. Co-Applicant
11. Retraining
12. New and Used Cars

Levels of Accuracy and Errors for the Decision Tree obtained above

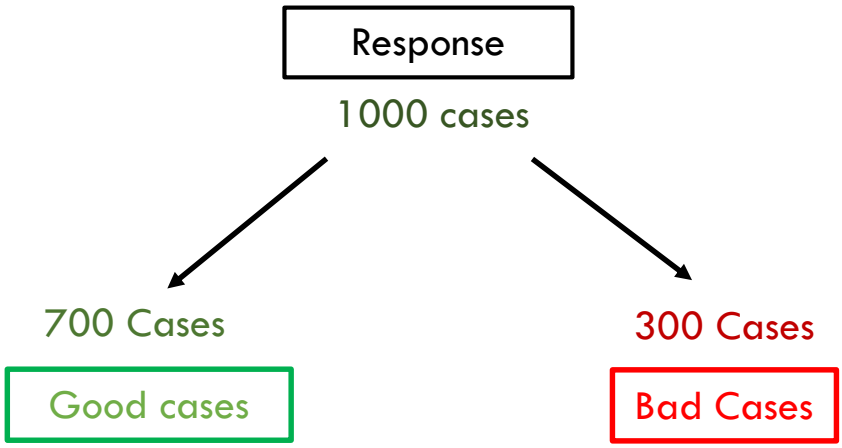
The performance metrics of the Decision Tree obtained on a Descriptive model given by the tree parameters as given above are listed out below:

Decision Tree Performance Metrics	Associated Value of the Metric
Misclassification Cost	74.000 +/- 0.000
Precision of the Good Cases	82.33%
Precision of the Bad Cases	54.14%
Overall Accuracy of the Model	72.80%

Analysis of the generated Decision Tree for the Descriptive Model

Below is a graphical breakdown of the total number of responses in the case of the decision tree developed for the entire data set.

Since, here the majority of the cases are 'Good' cases (700 cases), for a no-model prediction, we take all of the 1000 cases as 'Good' cases, yielding us the following accuracy and error rates.



Total accuracy of the system in the absence of a model is given by: 700/1000 (Since, 700 out of the 1000 cases can be accurately classified for the descriptive model).

Hence, total accuracy for no-model system is 70%

Similarly, error rate for the no-model system is 30%

The robustness of a model is defined as the accuracy with which it can predict or classify unseen data in the presence of noisy data, missing values, etc.

As seen from the analysis of the Decision Tree we created earlier, the total accuracy of the Decision Tree comes out to be 72.8%, which is better than the 70% accuracy obtained through the no-model system.

Hence, we can say that the Decision Tree model developed earlier to obtain a descriptive model is Robust.

Node Parameters used to obtain a Good model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on a Descriptive model:

Decision Tree Criterion Parameters	Associated Label or Value
Criterion	Gain Ratio
Maximal Depth	18
Apply Pruning	Checked
Confidence	0.01
Threshold	0.6
Apply Pre-Pruning	Checked
Minimal Gain	0.01
Minimal Leaf Size	4
Minimal size for Split	4
Number of Pre-Pruning alternatives	3

The most important Decision Tree node parameters for obtaining a good model are described below:

1. Minimal size for Split: Minimal size for Split is the parameter that mentions a decision tree when to split on the dataset, based on the number of cases in each class. If minimal size for split is 4, then the decision tree will not create a new branch when there are only three examples in the node. This prevents over-fitting. Also, on the other hand, a high minimal size for Split will lead to small trees, which aren't very useful for obtaining a good model and are often cases of under-fitting.
2. Minimal Leaf Size: When a decision tree features branches with single examples is not very useful, even though it yields better accuracy. Therefore, the minimum number of examples classified by a leaf in the tree can be set to 4 for the given dataset to yield a good model. This number of minimal leaf size does not yield leaves with a very small number of cases and hence helps in developing an efficient model.
3. Maximal Depth: The maximal depth of a decision tree is another important parameter which mentions up to what depth the tree must be allowed to grow. Too high a value for the maximal depth will yield in higher accuracy, but will lead to poorer and over fitted models. A maximal depth of 18 yields the best results in the model, without over-fitting the data, while also not affecting the accuracy of the model in the process.

(Please Refer 'Appendix' for supporting information)

Question no. 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)?**

Solution.

From the German Credit dataset imported into Rapidminer previously, and through the use of the 'Split Validation' operator of Rapidminer, we can partition the data into Training and Test sections.

For the first part, here we mention a Split Ratio of 0.5 and 'Shuffle Sampling' for Rapidminer to split the dataset into equal partitions of testing and training datasets.

Decision Tree Parameters used to obtain a 50%-50% Split model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on 50-50 Split model:

Decision Tree Criterion Parameters	Associated Label or Value
Criterion	Gini Index
Maximal Depth	10
Apply Pruning	Checked
Confidence	0.01
Threshold	0.5
Apply Pre-Pruning	Checked
Minimal Gain	0.07
Minimal Leaf Size	10
Minimal size for Split	5
Number of Pre-Pruning alternatives	5

Levels of Accuracy and Errors for the Decision Tree obtained above

The performance metrics of the Decision Tree obtained on a 50%-50% split model given by the tree parameters as given above are listed out below:

Decision Tree Performance Metrics	Associated Value of the Training Dataset	Associated Value for the Test Dataset
Misclassification Cost	59.600 +/- 0.000	86.000 +/- 0.000
Precision of the Good Cases	86.14%	79.25%
Precision of the Bad Cases	60.87%	54.25%
Overall Accuracy of the Model	78%	71.6%

Analysis of the generated Decision Tree for the 50%-50% Split Model

The decision tree model generated on the 50%-50% dataset split could be considered as reliable (robust), since it yields a better accuracy as compared to the no-model performance.

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?

Solution.

First, we mention a Split Ratio of 0.7 and 'Shuffle Sampling' for Rapidminer to split the dataset into 70%-30% partitions of training and testing datasets respectively.

Decision Tree Parameters used to obtain a 70%-30% Split model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on 70-30 Split model:

Decision Tree Criterion Parameters	Associated Label or Value
Criterion	Gini Index
Maximal Depth	10
Apply Pruning	Checked
Confidence	0.01
Threshold	0.72
Apply Pre-Pruning	Checked
Minimal Gain	0.085
Minimal Leaf Size	8
Minimal size for Split	5
Number of Pre-Pruning alternatives	5

Levels of Accuracy and Errors for the Decision Tree obtained above

The performance metrics of the Decision Tree obtained on a 70%-30% split model given by the tree parameters as given above are listed out below:

Decision Tree Performance Metrics	Associated Value of the Training Dataset	Associated Value for the Test Dataset
Misclassification Cost	63.000 +/- 0.000	82.000 +/- 0.000
Precision of the Good Cases	85.95%	78.29%
Precision of the Bad Cases	47.86%	55.20%
Overall Accuracy of the Model	70.71%	68.67%

Analysis of the generated Decision Tree for the 70%-30% Split Model

The decision tree model generated on the 70%-30% dataset split could not be considered as reliable (robust), since it yields a poorer accuracy as compared to the no-model performance, as well as the previously generated Decision tree.

Although, the precision for the Bad cases as improved with this model, it still yields poorer overall accuracy as well as poorer accuracy for the precision of Good cases. Hence, this model could be considered as 'Unreliable'.

Next, we mention a Split Ratio of 0.8 and 'Shuffle Sampling' for Rapidminer to split the dataset into 80%-20% partitions of Training and Testing datasets respectively.

Decision Tree Parameters used to obtain an 80%-20% Split model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on an 80-20 Split model:

Decision Tree Criterion Parameters	Associated Label or Value
Criterion	Gini Index
Maximal Depth	10
Apply Pruning	Checked
Confidence	0.01
Threshold	0.65
Apply Pre-Pruning	Checked
Minimal Gain	0.08
Minimal Leaf Size	6
Minimal size for Split	5
Number of Pre-Pruning alternatives	5

Levels of Accuracy and Errors for the Decision Tree obtained above

The performance metrics of the Decision Tree obtained on an 80%-20% split model given by the tree parameters as given above are listed out below:

Decision Tree Performance Metrics	Associated Value of the Training Dataset	Associated Value for the Test Dataset
Misclassification Cost	56.500 +/- 0.000	67.500 +/- 0.000
Precision of the Good Cases	87.13%	80.47%
Precision of the Bad Cases	56.95%	58.33%
Overall Accuracy of the Model	76.00%	72.50%

Analysis of the generated Decision Tree for the 80%-20% Split Model

The decision tree model generated on the 80%-20% dataset split could be considered as reliable (robust), since it yields a better accuracy as compared to the no-model performance and also a better performance as compared to the previous two decision tree models developed.

Here, the variance for the Misclassification costs and the overall model accuracy is also on the lower end. Hence, this could be considered as a 'Robust' model.

For further analysis of the data, we mention a Split Ratio of 0.6 and 'Shuffle Sampling' for Rapidminer to split the dataset into 60%-40% partitions of Training and Testing datasets respectively.

Decision Tree Parameters used to obtain a 60%-40% Split model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on a 60-40 Split model:

Decision Tree Criterion Parameters	Associated Label or Value
Criterion	Gini Index
Maximal Depth	10
Apply Pruning	Checked
Confidence	0.01
Threshold	0.5
Apply Pre-Pruning	Checked
Minimal Gain	0.07
Minimal Leaf Size	6
Minimal size for Split	4
Number of Pre-Pruning alternatives	3

Levels of Accuracy and Errors for the Decision Tree obtained above

The performance metrics of the Decision Tree obtained on a 60%-40% split model given by the tree parameters as given above are listed out below:

Decision Tree Performance Metrics	Associated Value of the Training Dataset	Associated Value for the Test Dataset
Misclassification Cost	57.500 +/- 0.000	80.000 +/- 0.000
Precision of the Good Cases	87.40%	79.67%
Precision of the Bad Cases	51.06%	54.55%
Overall Accuracy of the Model	73.17%	70.00%

Analysis of the generated Decision Tree for the 60%-40% Split Model

The decision tree model generated on the 60%-40% dataset split could be considered as reliable (robust), since it yields a similar accuracy as compared to the no-model performance and the variance in overall accuracy is relatively small.

From the above generated set of Decision Tree models, we would choose the 80%-20% split model, since it yields the best accuracy as well as low misclassification costs. Moreover, the variance for the misclassification costs and the overall accuracy are 2.25 and 3.25 respectively for the training and the testing dataset.

Hence, the 80%-20% model should be considered as a model with the best performance on unseen data and for its 'robustness'.

The effect of Pruning on the Decision Tree models

Based on the decision tree obtained from the modeling above, we take the 80%-20% split and use it for the analysis on models with and without pruning.

Decision Tree Performance Metrics	**Model obtained with Pruning**		**Model Without Pruning**	
	Values for the Training Dataset	**Values for the Test Dataset**	**Values for the Training Dataset**	**Values for the Test Dataset**
Misclassification Cost	56.500 +/- 0.000	67.500 +/- 0.000	0.875 +/- 0.000	104.500 +/- 0.000
Precision of the Good Cases	87.13%	80.47%	99.82%	74.29%
Precision of the Bad Cases	56.95%	58.33%	99.15%	51.67%
Overall Accuracy of the Model	76.00%	72.50%	99.62%	67.50%

From the obtained above results, we see that without pruning, the decision tree model over fits to the training dataset, hence obtaining poorer results when applied on unseen data.

Summarization of the Decision Tree parameters for different cases

Decision Tree Criterion Parameters	50%-50%	70%-30%	80%-20%	60%-40%
Criterion	Gini Index	Gini Index	Gini Index	Gini Index
Maximal Depth	10	10	10	10
Apply Pruning	Checked	Checked	Checked	Checked
Confidence	0.01	0.01	0.01	0.01
Threshold	0.5	0.72	0.65	0.5
Apply Pre-Pruning	Checked	Checked	Checked	Checked
Minimal Gain	0.07	0.085	0.08	0.07
Minimal Leaf Size	10	8	6	6
Minimal size for Split	5	5	5	4
Number of Pre-Pruning alternatives	5	5	5	3

Hence we find that Criterion, Maximal depth, Confidence to be the most important decision tree parameters which remain consistent throughout the different partitions of the dataset.

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. Do you see any performance differences across different types of decision tree learners?

Solution.

Decision Tree Parameters used to obtain a good 'CART' Decision Tree Model

Below listed are the details of the Parameters or the Decision Tree Criterion used to obtain a Decision Tree based on the CART Operator model:

Decision Tree Parameters (CART)	Associated Label or Value
S	1.0
D	Unchecked
M	2.0
N	5.0
U	Unchecked
H	Unchecked
A	Unchecked
C	1.0

Similarly, an adequate model using the W-J48 Operator for the Decision Trees is obtained using the following parameters:

Decision Tree Parameters (W-J48)	Associated Label or Value
U	Unchecked
C	0.25
M	10.0
R	Checked
N	Default
B	Unchecked
S	Checked
L	Unchecked
A	Checked
Q	Default

The performance metrics for the Decision tree using the 80%-20% split, CART and the W-J48 operators are listed out below:

Decision Tree Model	Accuracy (%) for the Training Dataset	Accuracy (%) for the Test Dataset
Decision Tree (80-20 Split)	76.00%	72.50%
CART Operator	72.40%	71.46%
W-J48 Operator	77.00%	75.20%

As seen from the performance table given above, we can conclude that the Decision Tree model obtained using the CART operator yields the best performance. The accuracy is high as well as the variance is relatively low as compared to the other Decision Tree models.

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?

Solution.

For verifying whether the developed Decision Tree model is stable or unstable, we change the values of the random seed to obtain different levels of performance accuracy. The decision tree with the CART operator has been used, since it yields the best accuracy and the least variance in terms of accuracy and performance accuracy.

The information is summarized below which is obtained for different values of the random seed:

Random Seed	Accuracy (%) for the Training Dataset	Accuracy (%) for the Test Dataset
1995	71.00%	70.00%
1992	75.60%	71.40%
2000	73.60%	71.20%
2200	74.20%	73.4%
2800	73.60%	70.6%

As seen from the performance accuracy above, the variance in the percentage accuracy for the training and the testing datasets is not too high for different values of the Random seed.

This illustrates the stability of the data and because of the very same reason, the variance tends to remain low even for different values of the Random Seed. Thus the model cannot be termed as 'Unstable' as with many other decision tree models applied on datasets.

For the decision tree obtained from different random seeds as inputted before, we see that the first three nodes remain consistent throughout. Hence, we can say that the Checking amount, Duration and the Amount are the most important variables used for predicting unseen datasets.

(Please Refer 'Appendix' for supporting information)

Question no. 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	
		Good	Bad
Actual	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?

Solution.

We have considered the optimal decision tree from the above cases with a split of 80:20 and calculated the misclassification costs at different threshold values.

Value of Threshold	Training Misclassification Cost	Testing Misclassification Cost	Calculated Misclassification Cost	Perf. Accuracy Test	Perf. Accuracy Training
0.5	78.375	103	$20600	73.00%	76.62%
0.6	57.75	83.50	$16700	72.5%	76.25%
0.7	55.00	75.50	$15100	72.5%	75.5%
0.8	52.75	74.00	$14800	70.0%	71.75%

The following results are evident from the above table.

1. The Training as well as testing misclassification costs goes on decreasing with the increase in threshold value.
2. The decrease in the misclassification costs follows a linear negative trend.
3. The overall loss which is calculated by (500*Number of False Positive + 100* Number of False Negative), decreases with the increase in threshold value.

4. By looking at the performance accuracy of the 80%-20% split Decision tree, we conclude that the threshold of 0.8 yields the lowest variance, although it features low accuracy.
5. In terms of pure performance accuracy, we conclude that the Decision Tree with a threshold of 0.5 yields the highest performance accuracy, both for the test and the training dataset.

Hence, we conclude that the Decision Tree model obtained at a split of 80%-20%, with a threshold of 0.5 yields the best accuracy and also a lower misclassification cost associated with it.

(Please Refer 'Appendix' for supporting information)

Question no. 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?

Solution.

By examining the decision tree models generated previously, we conclude that the 80%-20% model with a threshold of 0.65 yields the best results.

Following is a summarization of the description for the model described above:

Description	Associated Value for the Description
Tree Depth	7
Number of Nodes in the tree	9
Number of Nodes in the tree (Including Leaf Nodes)	24

List of the important Variables according to the selected Decision Tree Model:

1. Checking Account Status
2. History
3. Own Residence
4. Amount
5. Duration

6. New Car

Comparison of the Important Variables vs Variables found in Question 2:

Important Variables found with the new model	Important Variables as found in Question 2
Checking Account Status	Checking Account Status
History	Amount
Own Residence	Real Estate
Amount	Amount
Duration	Duration
New Car	New Car
	Number of Dependents
	Installment Rate
	Employment
	Age
	Co-Applicant
	Retraining

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

Solution.

By looking at the modeled Decision Tree, we obtain the following pair of relatively pure leaf nodes.

Variables with Pure leaf nodes	Probability of Good credit cases	Probability of Bad credit cases	Variable Class Type
Duration	95.4%	4.54%	1
New Car	85.71%	14.29%	1

(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).

Solution.

1. IF Checking Account = 0 AND History = 1 AND Other Installment = 1 AND Amount > 1746.5 THEN Class 1 (Good Credit)
2. IF Checking Account = 0 AND History = 3 THEN Class 0 (Bad Credit)

(Please Refer 'Appendix' for supporting information)

Question 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.

Solution.

Since the predicted probabilities can be used to determine how a decision tree model may be implemented, the data is sorted from high to low based on the predicted probability of 'Good' credit risk.

This sorting makes it easier to find the cutoff probability, above which could be considered as acceptable 'credit risk'.

After complete analysis of the model, a cutoff probability of 87.89% could be recommended for scoring future 'Good credit' applicants. A maximum cumulative benefit of 3000DM is obtained at this cutoff point.

(Please find attached Question 6 – Cutoff Probability Calculation.xlsx for Reference)

(Please Refer 'Appendix' for supporting information)

APPENDIX

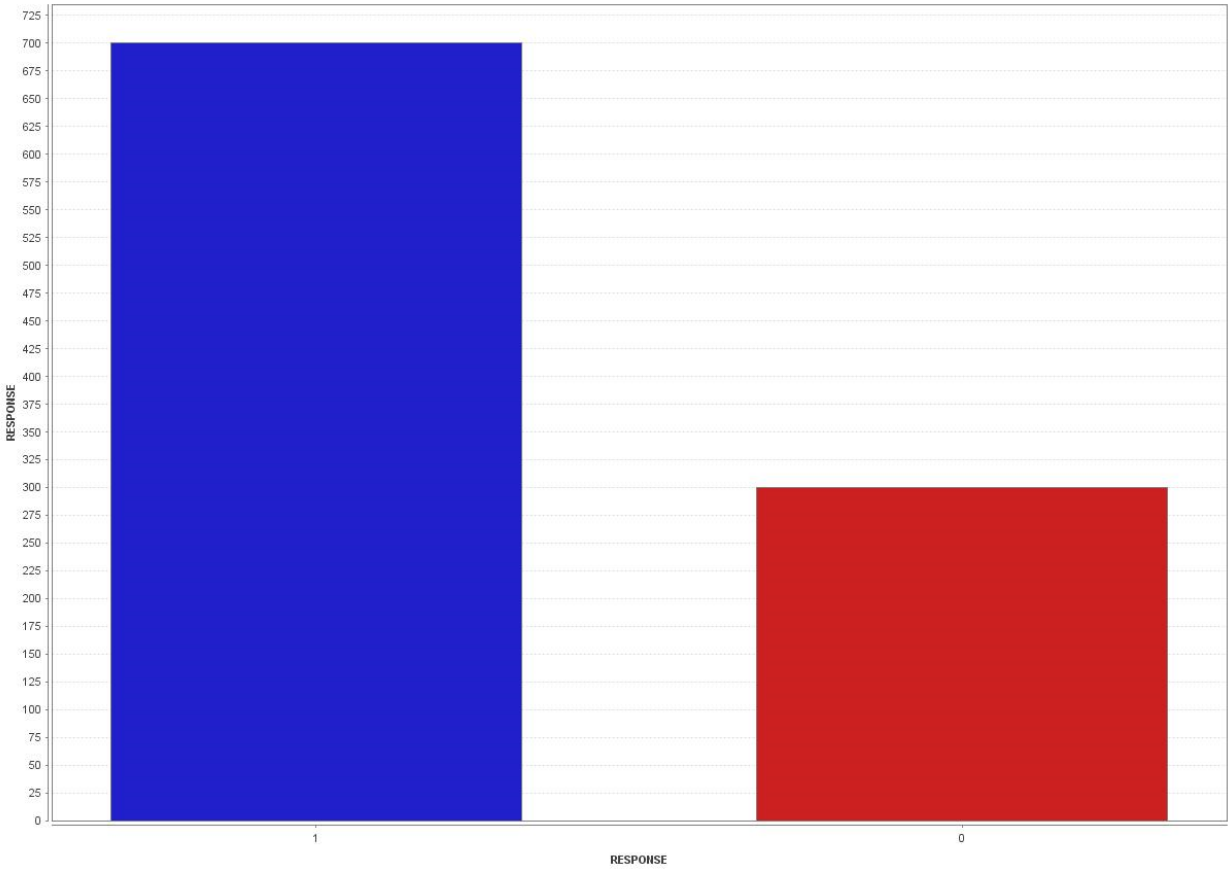


Fig 1.1. Proportion of Good and Bad Cases (Question 1)

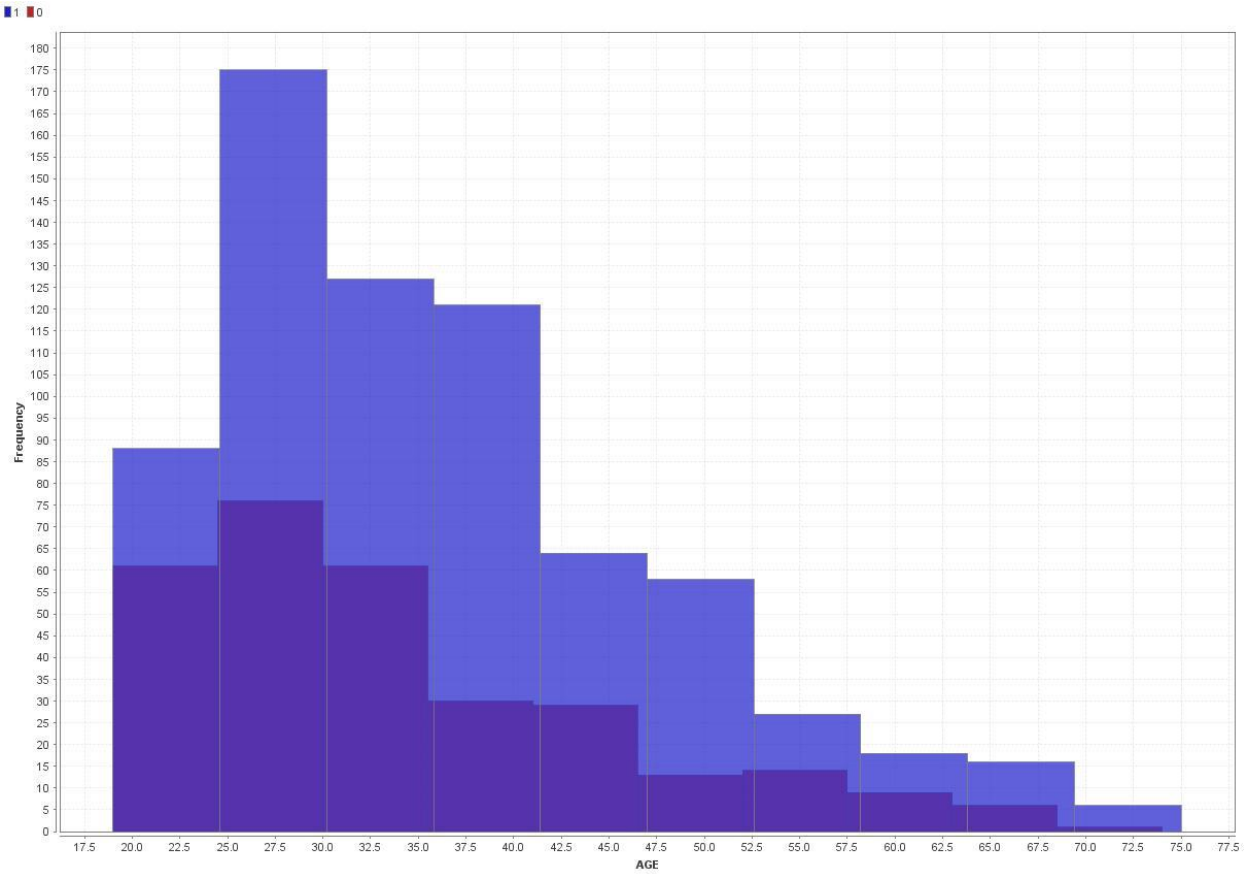


Fig 1.2. Comparison chart of Age vs Response (Question 1)

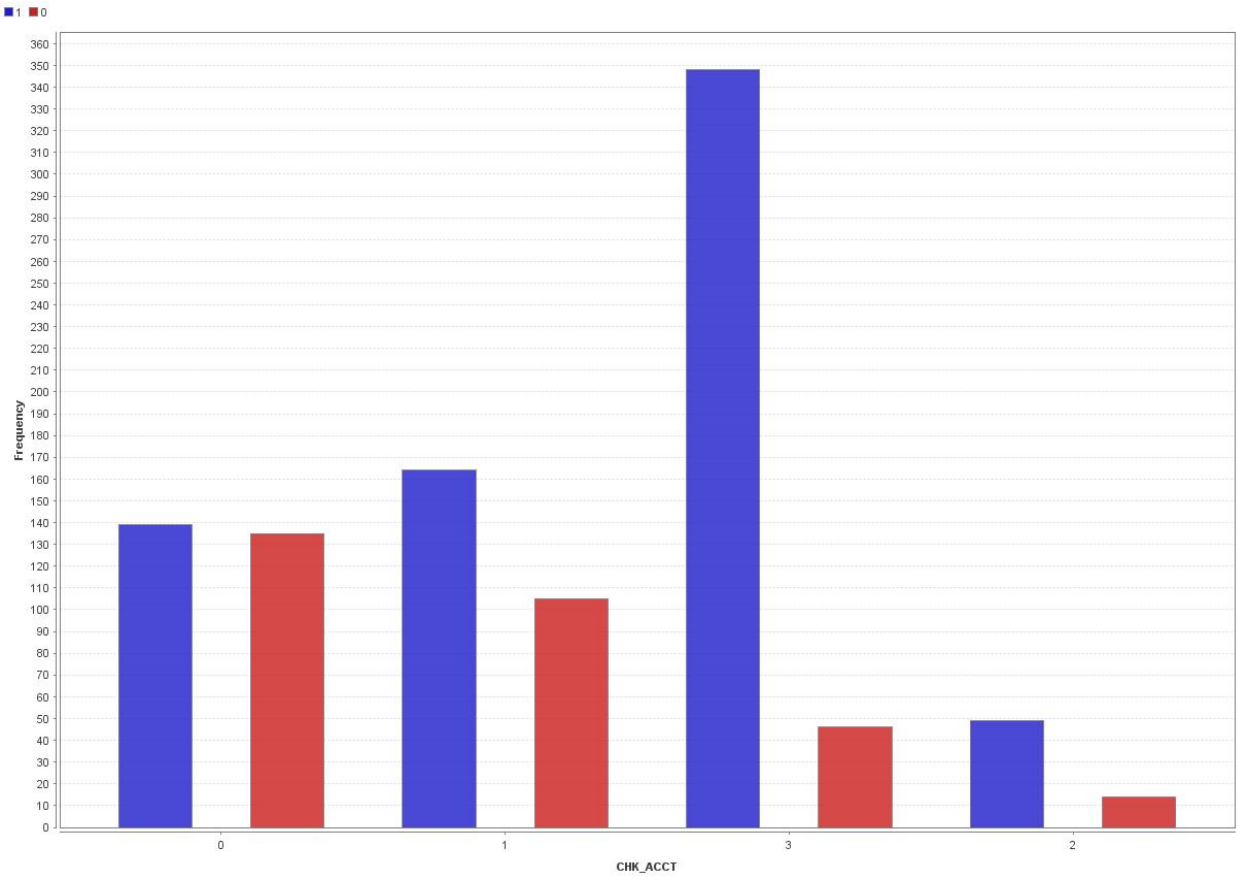


Fig 1.3. Comparison chart of Checking Account vs Response (Question 1)

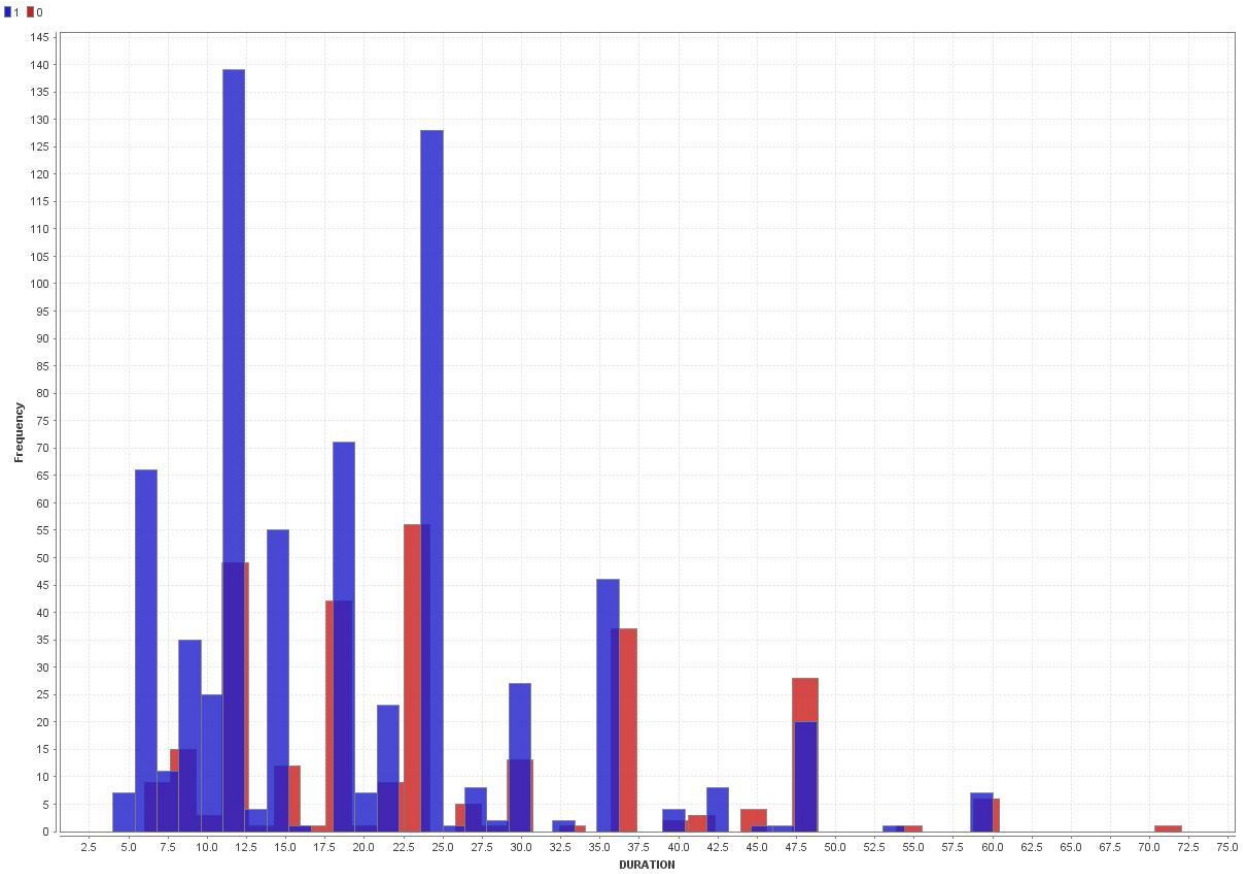


Fig 1.4. Comparison chart of Duration vs Response (Question 1)

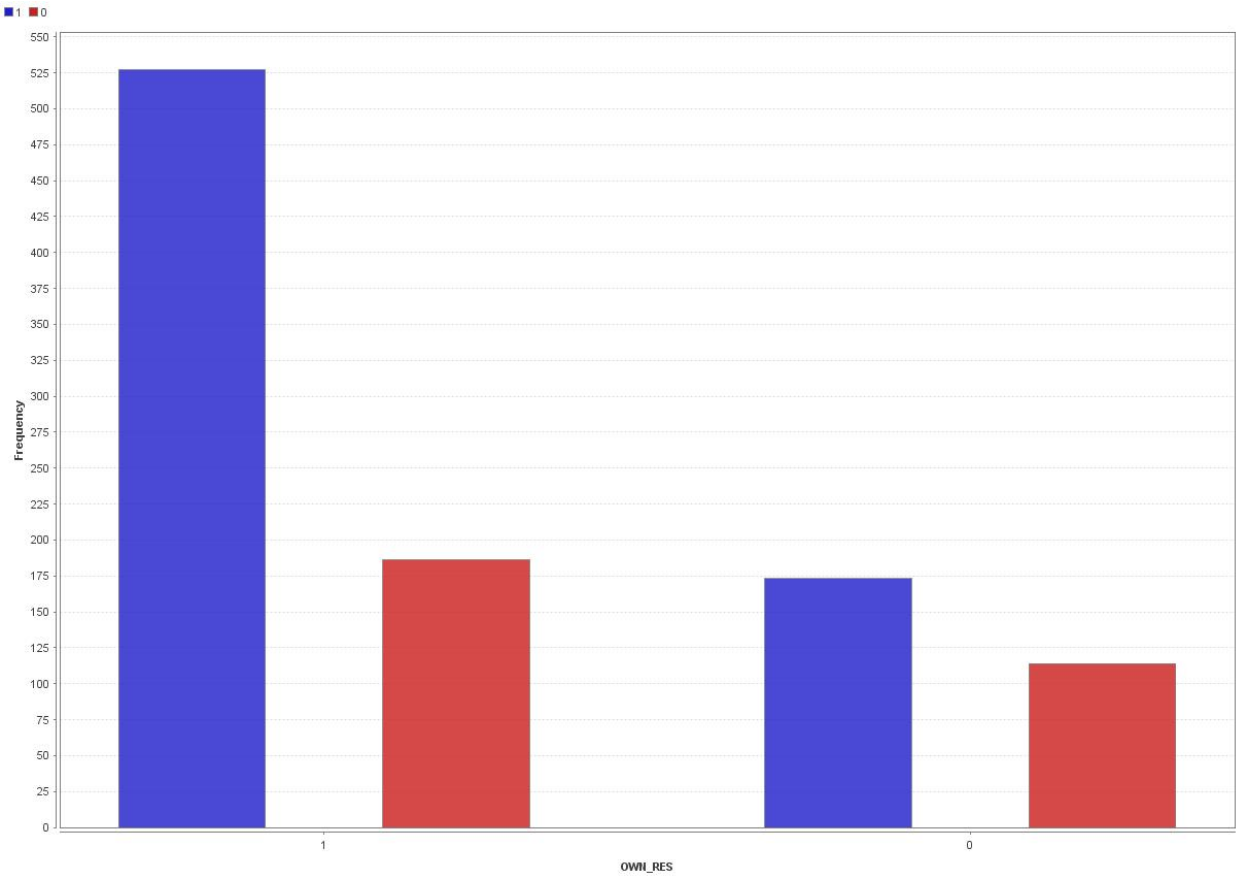


Fig 1.5. Comparison chart of Own Residence vs Response (Question 1)

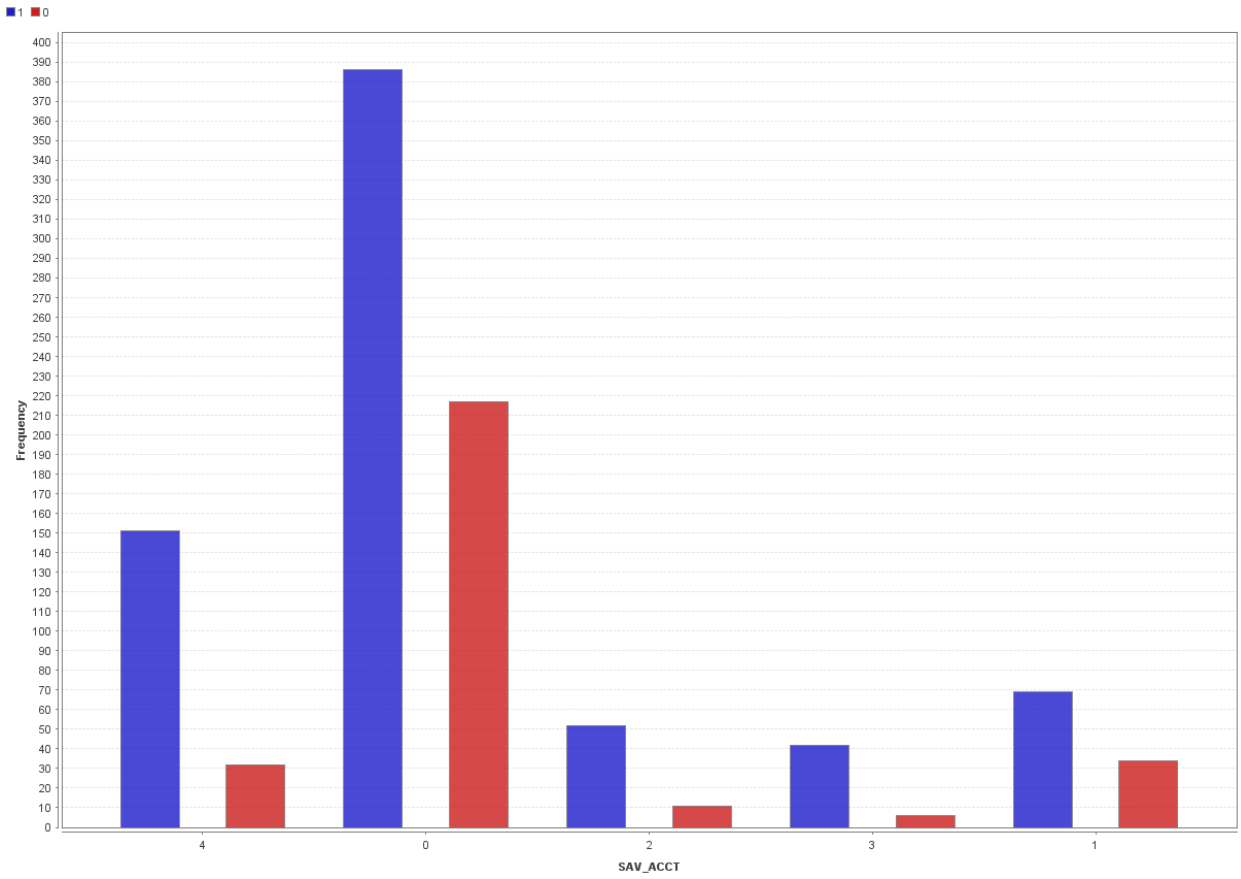


Fig 1.6. Comparison chart of Saving Account vs Response (Question 1)

accuracy: 72.80%

	true 1	true 0	class precision
pred. 1	545	117	82.33%
pred. 0	155	183	54.14%
class recall	77.86%	61.00%	

Fig 2.1. Decision Tree Performance Matrix (Question 2)

Decision Tree

```
CHK_ACCT = 0
|   AMOUNT > 438: 0 {1=134, 0=135}
|   AMOUNT ≤ 438: 1 {1=5, 0=0}
CHK_ACCT = 1
|   AMOUNT > 12296.500: 0 {1=0, 0=12}
|   AMOUNT ≤ 12296.500: 1 {1=164, 0=93}
CHK_ACCT = 2
|   DURATION > 7.500
|   |   NUM_DEPENDENTS > 1.500: 0 {1=1, 0=4}
|   |   NUM_DEPENDENTS ≤ 1.500
|   |   |   REAL_ESTATE = 0: 1 {0=4, 1=32}
|   |   |   REAL_ESTATE = 1
|   |   |   |   INSTALL_RATE > 3.500: 0 {1=1, 0=4}
|   |   |   |   INSTALL_RATE ≤ 3.500: 1 {0=2, 1=7}
|   DURATION ≤ 7.500: 1 {1=8, 0=0}
CHK_ACCT = 3
|   AMOUNT > 10924.500: 1 {1=4, 0=3}
|   AMOUNT ≤ 10924.500
|   |   OTHER_INSTALL = 0
|   |   |   CO-APPLICANT = 0
|   |   |   |   USED_CAR = 0
|   |   |   |   |   AMOUNT > 7829: 1 {1=4, 0=2}
|   |   |   |   |   AMOUNT ≤ 7829
|   |   |   |   |   |   AMOUNT > 4458.500
|   |   |   |   |   |   |   EMPLOYMENT = 1: 0 {1=2, 0=3}
|   |   |   |   |   |   |   EMPLOYMENT = 2: 1 {1=4, 0=4}
|   |   |   |   |   |   |   EMPLOYMENT = 3: 1 {1=8, 0=0}
|   |   |   |   |   |   |   EMPLOYMENT = 4: 1 {1=10, 0=0}
|   |   |   |   |   |   AMOUNT ≤ 4458.500
|   |   |   |   |   |   |   AGE > 22.500: 1 {0=10, 1=212}
|   |   |   |   |   |   |   AGE ≤ 22.500
|   |   |   |   |   |   |   |   INSTALL_RATE > 2.500: 0 {1=2, 0=3}
|   |   |   |   |   |   |   |   INSTALL_RATE ≤ 2.500: 1 {1=6, 0=0}
|   |   |   |   USED_CAR = 1: 1 {1=44, 0=0}
|   |   |   CO-APPLICANT = 1
|   |   |   |   AMOUNT > 2026.500: 1 {1=7, 0=0}
|   |   |   |   AMOUNT ≤ 2026.500: 0 {1=1, 0=3}
|   |   OTHER_INSTALL = 1
|   |   |   DURATION > 8
|   |   |   |   NEW_CAR = 0
|   |   |   |   |   RETRAINING = 0: 1 {0=6, 1=33}
|   |   |   |   |   RETRAINING = 1: 0 {1=5, 0=6}
|   |   |   |   NEW_CAR = 1: 0 {1=1, 0=6}
|   |   |   DURATION ≤ 8: 1 {1=5, 0=0}
```

(Question 2 Decision Tree)

accuracy: 71.60%			
	true 1	true 0	class precision
pred. 1	275	72	79.25%
pred. 0	70	83	54.25%
class recall	79.71%	53.55%	

Fig 3.1. Decision Tree (50%-50%) Testing Performance Matrix (Question 3)

accuracy: 78.00%			
	true 1	true 0	class precision
pred. 1	292	47	86.14%
pred. 0	63	98	60.87%
class recall	82.25%	67.59%	

Fig 3.2. Decision Tree (50%-50%) Training Performance Matrix (Question 2)

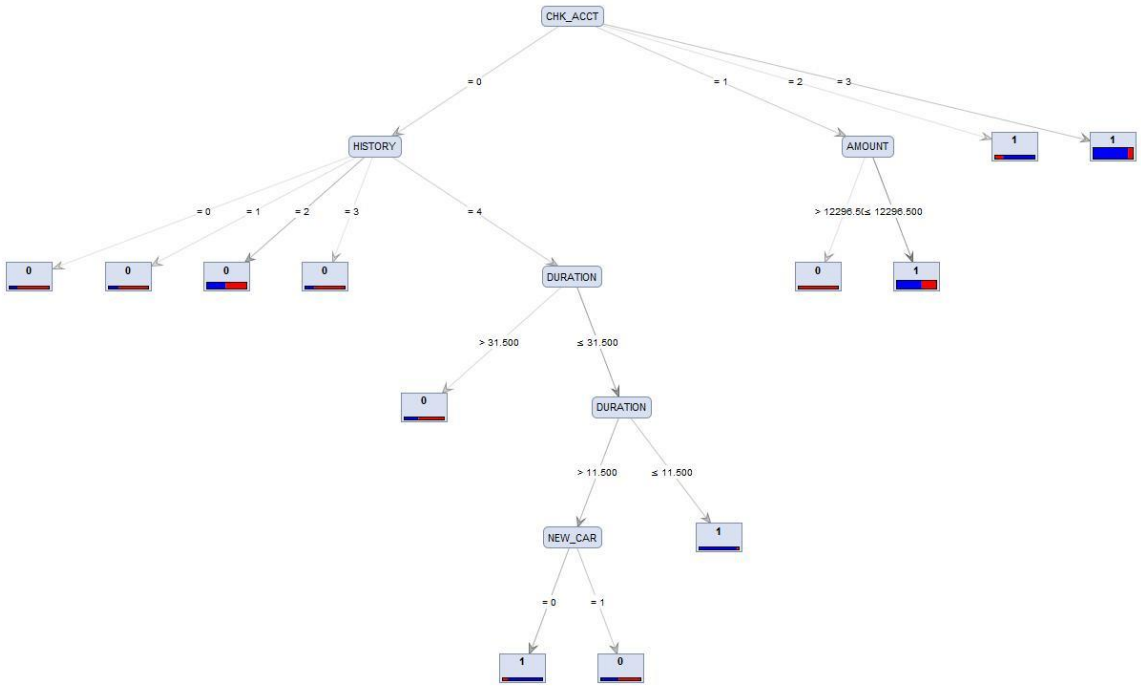


Fig 3.3. Decision Tree (50%-50%) Decision Tree Chart (Question 3)

accuracy: 68.67%			
	true 1	true 0	class precision
pred. 1	137	38	78.29%
pred. 0	56	69	55.20%
class recall	70.98%	64.49%	

Fig 3.4. Decision Tree (70%-30%) Testing Performance Matrix (Question 3)

accuracy: 70.71%			
	true 1	true 0	class precision
pred. 1	361	59	85.95%
pred. 0	146	134	47.86%
class recall	71.20%	69.43%	

Fig 3.5. Decision Tree (70%-30%) Training Performance Matrix (Question 3)

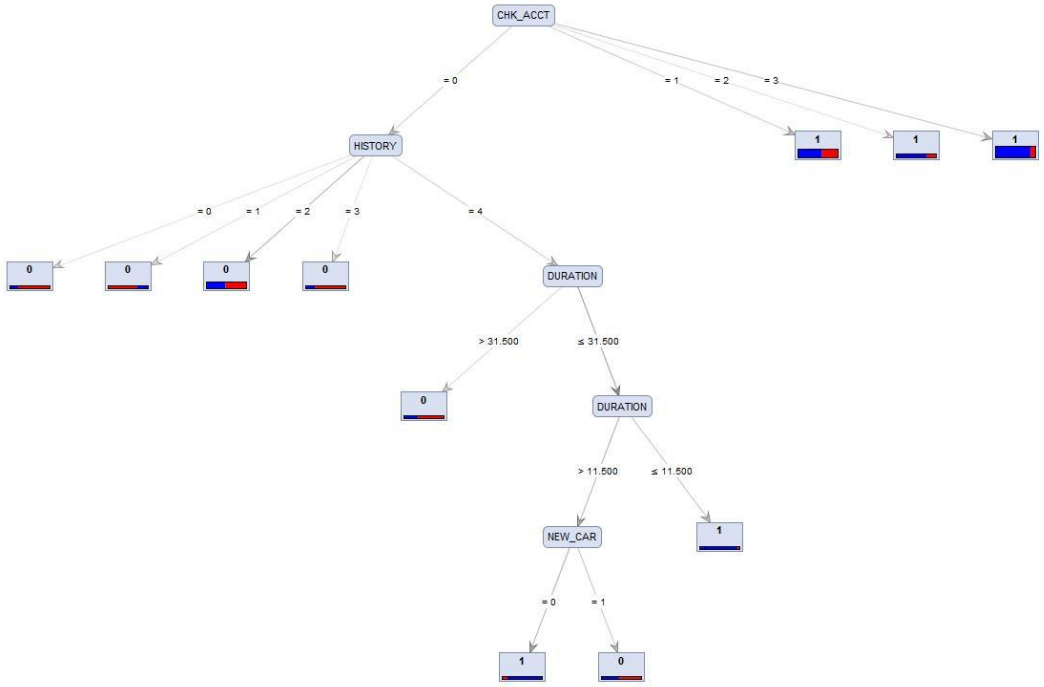


Fig 3.6. Decision Tree (70%-30%) Decision Tree Chart (Question 3)

accuracy: 73.00%

	true 1	true 0	class precision
pred. 1	117	38	75.48%
pred. 0	16	29	64.44%
class recall	87.97%	43.28%	

Fig 4.1. 0.5 Threshold Performance Matrix (Question 4)

accuracy: 72.50%

	true 1	true 0	class precision
pred. 1	106	28	79.10%
pred. 0	27	39	59.09%
class recall	79.70%	58.21%	

Fig 4.2. 0.6 Threshold Performance Matrix (Question 4)

accuracy: 72.50%

	true 1	true 0	class precision
pred. 1	102	24	80.95%
pred. 0	31	43	58.11%
class recall	76.69%	64.18%	

Fig 4.3. 0.7 Threshold Performance Matrix (Question 4)

accuracy: 70.00%

	true 1	true 0	class precision
pred. 1	95	22	81.20%
pred. 0	38	45	54.22%
class recall	71.43%	67.16%	

Fig 4.4. 0.8 Threshold Performance Matrix (Question 4)

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: 0 {1=78, 0=82}
|   HISTORY = 3: 0 {1=3, 0=9}
|   HISTORY = 4
|   |   DURATION > 31.500: 0 {1=4, 0=7}
|   |   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: 1 {1=164, 0=105}
CHK_ACCT = 2: 1 {1=49, 0=14}
CHK_ACCT = 3: 1 {1=348, 0=46}
```

Fig 5.1. Decision Tree (Question 5)

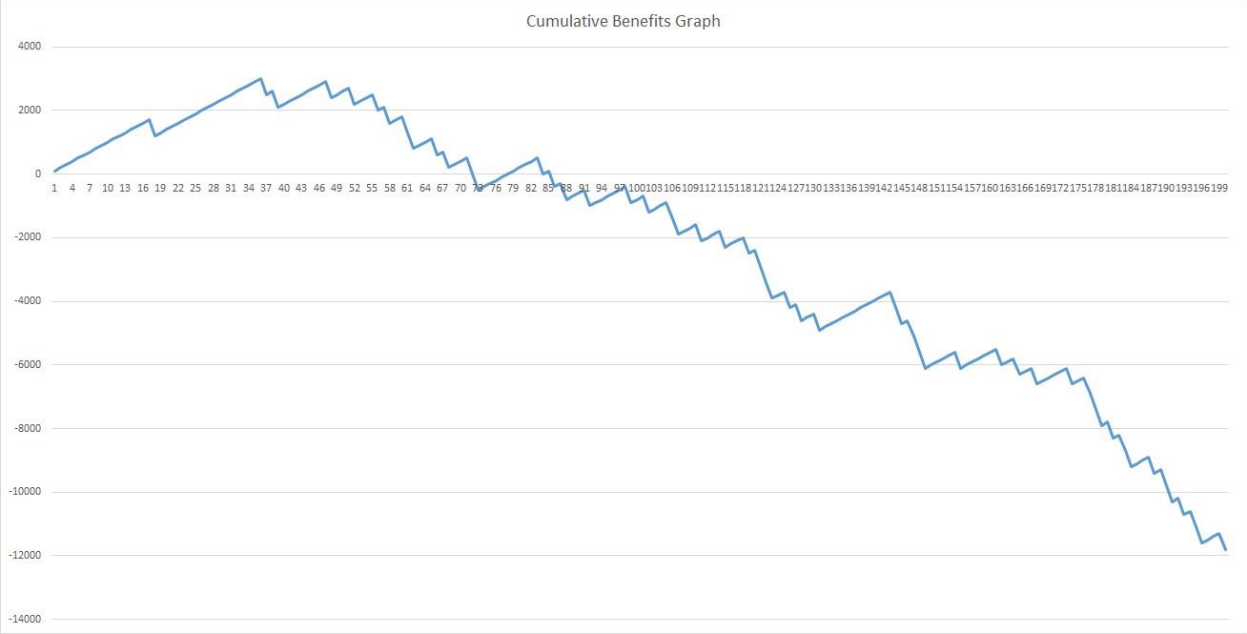


Fig 6.1. Cumulative Benefit Graph (Question 6)