**IDS 572 – Data Mining**

**Assignment - 1**

**Submitted By:**

**Mikhil Pavu Paul**

**Akshat Verma**

**Amit Gupta**

1. Number of good credit cases is 700 and bad credit cases is 300.

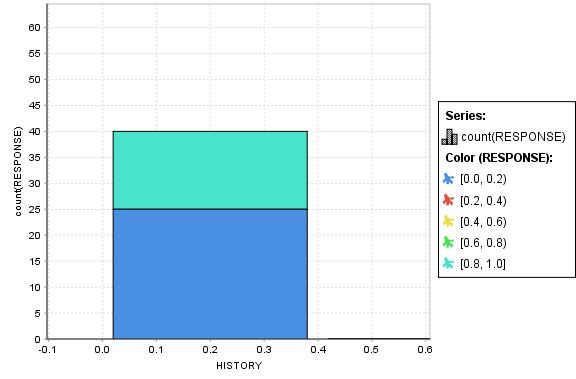
|  |  |
| --- | --- |
| **Missing Value** | **Corrective Action** |
| |  | | --- | | NEW\_CAR | | USED\_CAR | | FURNITURE | | RADIO/TV | | EDUCATION | | RETRAINING | | Replace null value with 0 |
| AGE | Replace null with average- such that valuable information is not discarded without affecting the overall average |

**Attribute Description**

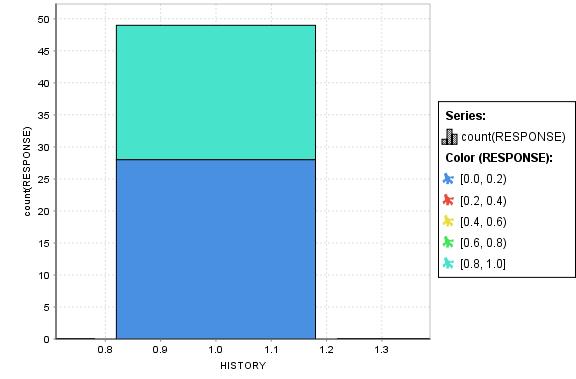
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Min. Value** | **Max. Value** | **Average** | **Std. Deviation** |
| DURATION | Duration of credit in months | 4 | 72 | 20.903 | 12.059 |
| AMOUNT | Credit amount | 250 | 18424 | 3271.15 | 2822.62 |
| AGE | Age of applicants in Yeats | 19 | 75 | 35.48 | 11.32 |

**Observed Bad Cases: -**

* People with credit History category 0 i.e., no credits taken, tends to have Bad response as shown below.



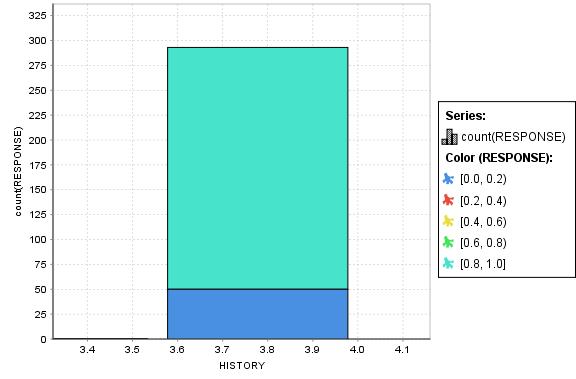
* As observed from below chart, there are more bad cases applicants who have paid back all credits duly (Category 1) which seems to be more of a ‘surprise’.



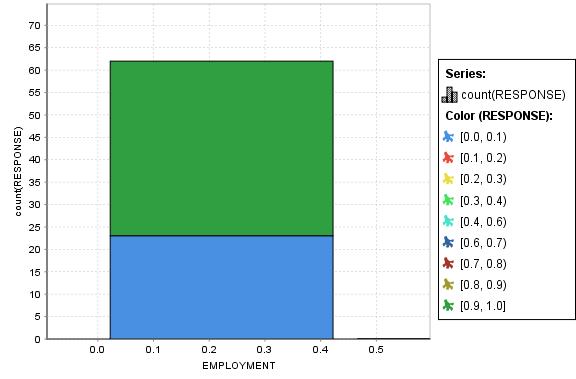
* Chances of getting a Bad credit response is more if the credit amount is greater than 12,000.

**Interesting Relationships: -**

* It is interesting to find that applicants whose accounts are in critical stage (HISTORY category 4) tend to get more ‘Good’ responses.



* Unemployed applicants also have a decent chance for getting Good credit response.



**Variables affecting Outcome of interest**

* Amount
* Age
* History
* Duration
* Savings Account
* Checking Account
* Job

During data exploration from various charts, it was observed that the above-mentioned variables best differentiated the Good and Bad cases.

**2)**

1. **Descriptive Model**
2. The following node parameters were used to get the best descriptive model.

|  |  |  |
| --- | --- | --- |
| **Node Parameter** | **Value** | **Description** |
| Criterion | Gini-Index | Gini-Index gave better split when compared to other split criterions. Some other criterions did not grow the tree beyond 1 level. |
| Maximal Depth | 10 | We found that 10 gives the right tree size with better accuracy. Adjusting them varied accuracy. |
| Prune Confidence | 0.25 | 25% confidence level for error calculation of pruning gave the correct pruned Decision Tree. |
| Minimal gain | 0.02 | Any value above 2% resulted in a decision stump and any value less than that gave consistent accuracy. |
| Minimal leaf size | 8 | Better accuracy for model and larger margin in lift chart were obtained with a minimal leaf size of 8 |
| Minimal size for split | 6 | It was found that value of 6 gave an optimal model. Also, higher value returned less accurate model. |
| Number of pre-pruning alternatives | 3 | Value of 3 gave an optimal model. Also, it was observed that beyond the value of 3 accuracy was constant. |

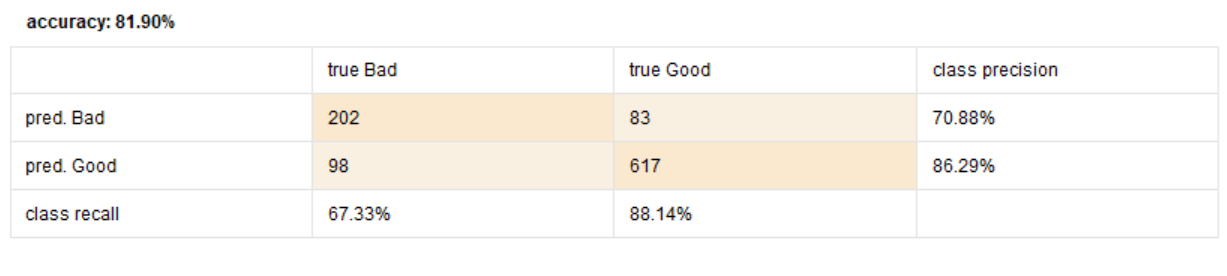
1. **Variables differentiating Good and Bad cases:**

* Amount
* Age
* Duration
* Checking Account

From the decision tree it was observed that the above-mentioned variables gave better split with almost pure leaf nodes. These variables do match with the findings from the question 1 except for Savings Account, Job and History.

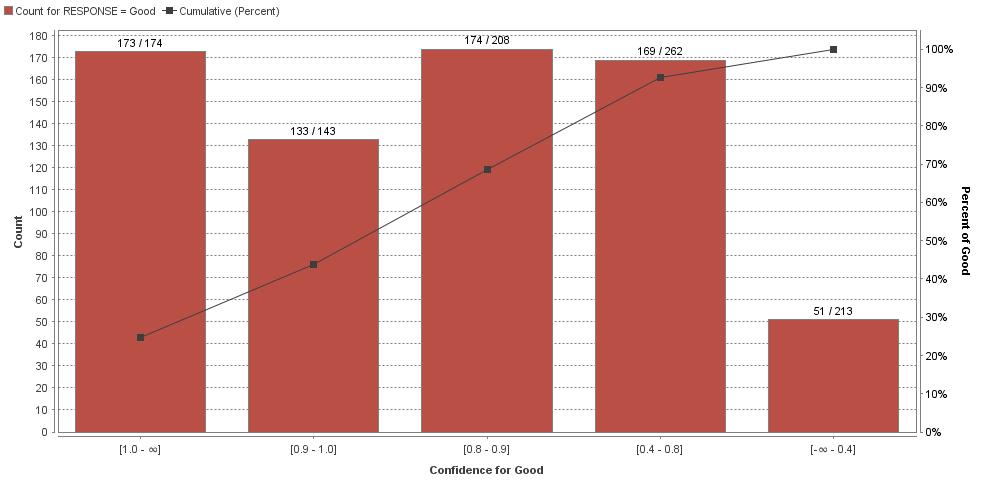
1. **Performance Metrics-Accuracy/Error/Lift Chart**

Below is the confusion Matrix for the descriptive model.



With Precision: **86.29%** and Recall: **88.14%**

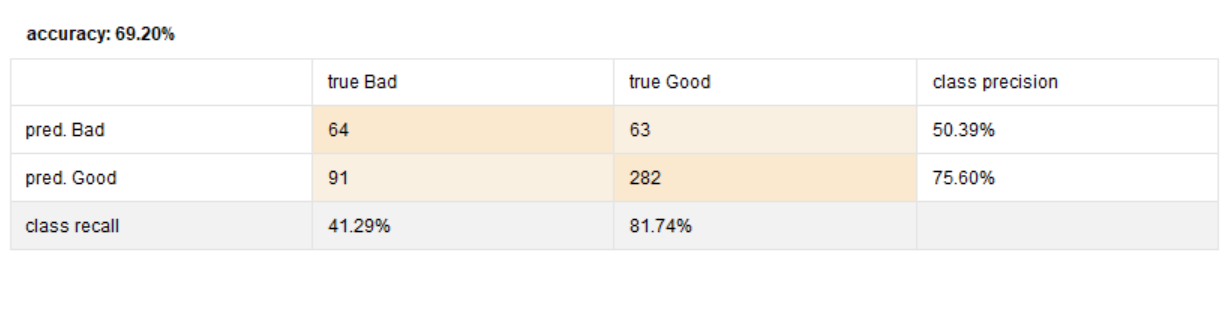
**Lift Chart**



We can observe a good Lift chart for the descriptive model as most of the predicted responses fall above the no model line.

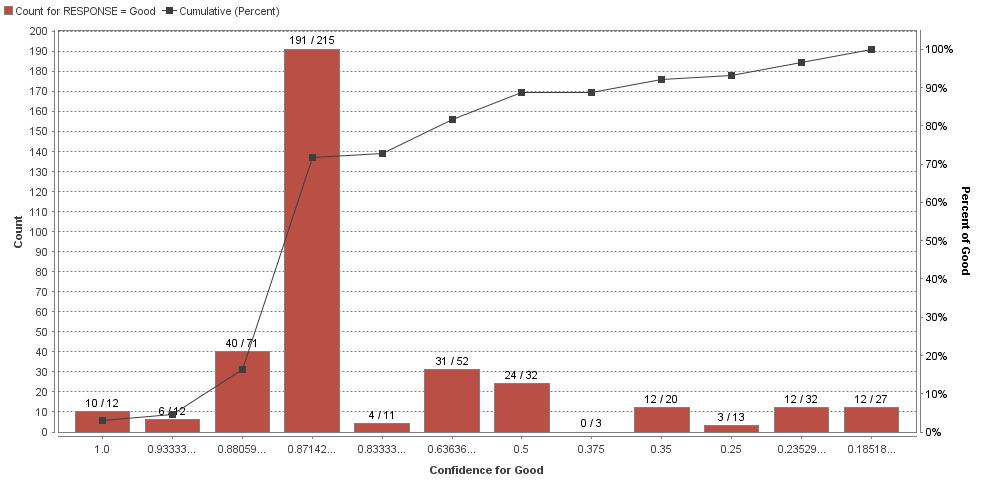
Yes from the overall accuracy and lift chart performance it can be concluded that this descriptive model is reliable.

1. **Predictive Model**



With a **Precision: 75.60%**and **Recall: 81.74%**

**Lift Chart:**

****

A decent lift chart performance with a good number of predicted responses falling above the no model line.

Unpruned decision tree gave high accuracy of 93.20% on training data and 73.20% on test data. But a poor performance was obtained from Lift chart hinting us that this drop in performance could be because of overfitting.

**Parameters for obtaining a Good Model**

* Split Criterion
* Maximal Depth
* Minimal Gain
* Minimal Leaf Size
* Minimal Size for Split

Adjusting these parameter values retrieved an optimal model.

Yes, these parameter values are the same across different training-test partitions.

**b)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Decision Tree Operator** | **Parameter** | **Description** | | **Value** | **Accuracy** |
| CART | M | Minimal number of instances at the terminal nodes | | 3.0 | 73.00% |
| N | Number of folds used in cost-complexity pruning | | 4.0 |
| J48 | C | Confidence threshold for pruning | | 0.25 | 73.20% |
| M | Minimum number of instances per leaf | | 4.0 |
| N | Number of folds for pruning | | 2 |
| S | Don’t Perform subtree raising | | Checked |
| A | Laplace smoothing | | Checked |
| Random Forest | Number of trees | Specifies the total number of decision trees to be created for the give training data | | 500 | 69.20% |
| Criterion | Gini-Index criterion gave better performance in lift chart | | Gini-index |
| Apply Pruning & Apply Pre- Pruning | Specifies if the tree is to be pruned or not | | Unchecked |
| **Altering other parameters did not make any difference to accuracy nor to the lift chart** | | | |
| AdaBoost (Decision Tree) | Iterations | | 10 | | 73.00% |

Performance do vary across different decision tree learners as it can be observed from different accuracies. It can also be observed that AdaBoost on Decision Tree learner has increased the performance significantly both in terms of accuracy and lift chart.

**Note: - AdaBoost meta algorithm did not show any significant increase or decrease in accuracies for CART, J-48 and Random Forest learners and thus the performances are not recorded in the above table.**

**Model Comparison**

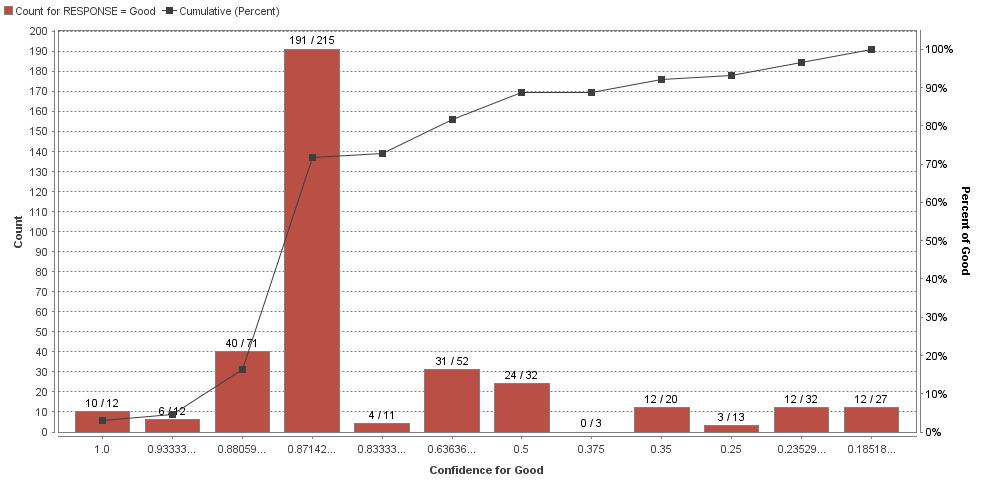
|  |  |  |  |
| --- | --- | --- | --- |
| **50-50 Split** | **Precision** | **Recall** | **Overall Accuracy** |
| **Decision Tree** | 75.60% | 81.74% | 69.20% |
| **J48** | 74.36% | 93.33% | 73.20% |
| **CART** | 80.52% | 80.29% | 73.00% |
| **Random Forest** | 69.14% | 100.00% | 69.20% |
| **AdaBoost (Decision Tree)** | 73.54% | 95.07% | 73.00% |

**Performance parameters used: -**

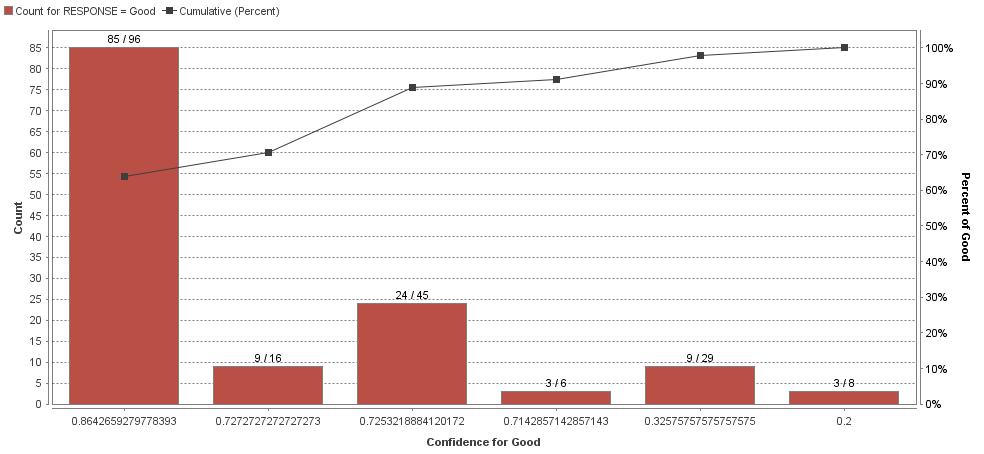
* Precision – Gives a percentage about predicted good vs actual good.
* Recall – gives the true positive rate.
* Accuracy – gives the overall performance of the model.
* Lift Chart – This performance measure gives us an estimation of how good the model will perform on unseen data with respect to ‘no model’ case.

**Lift Chart Comparison: -**

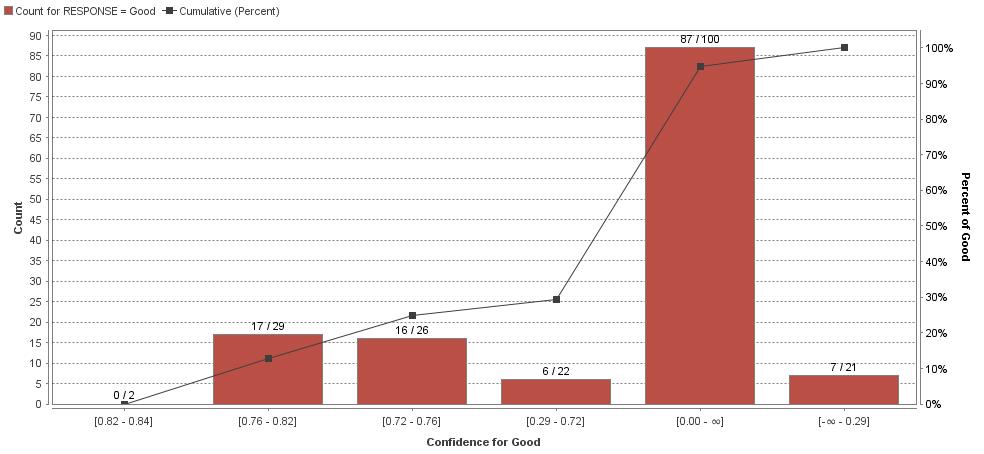
**Decision Tree**

****

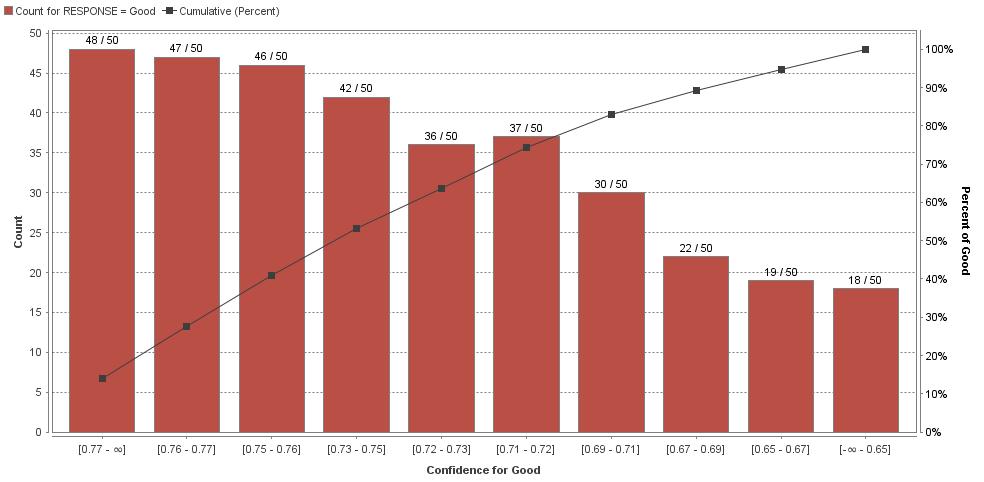
**CART**

****

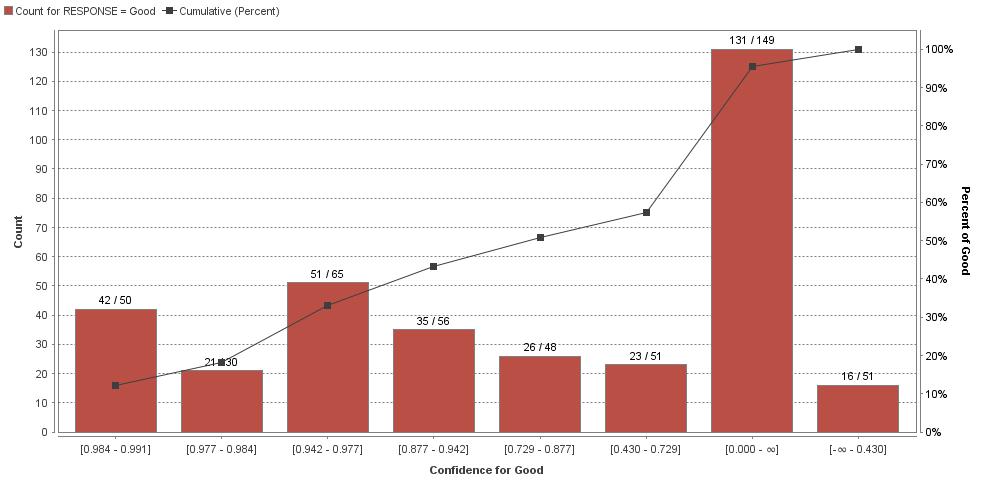
**J48**

****

**Random Forest**

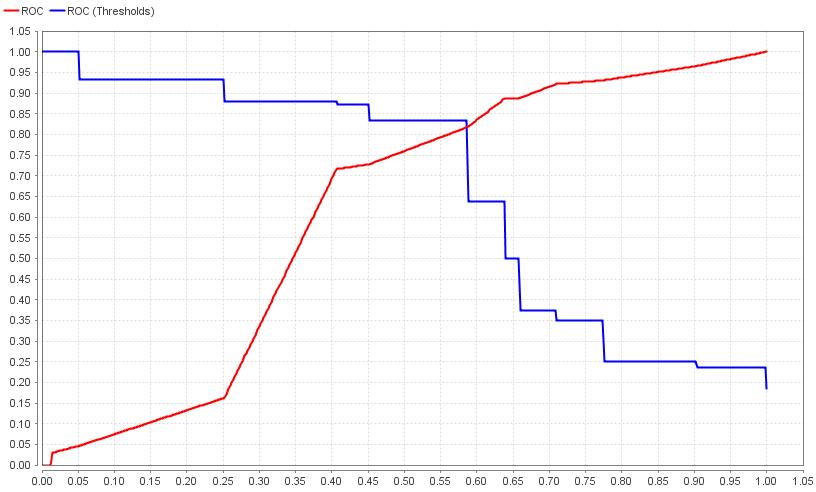
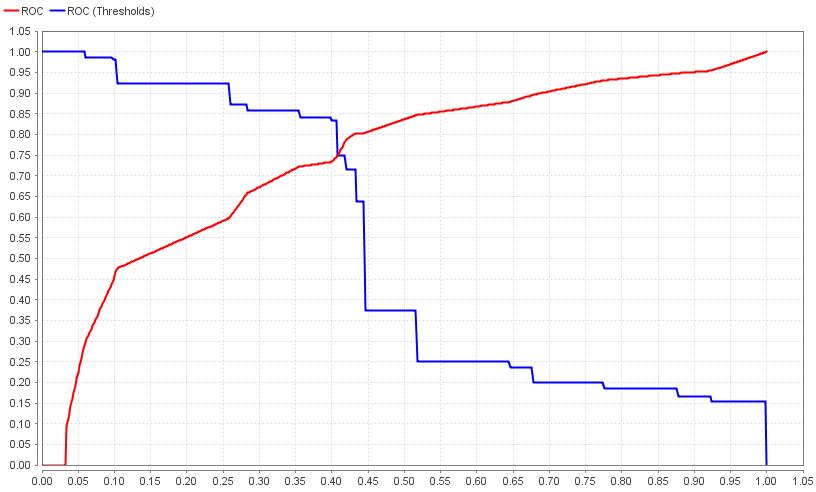
****

**AdaBoost (Decision Tree)**

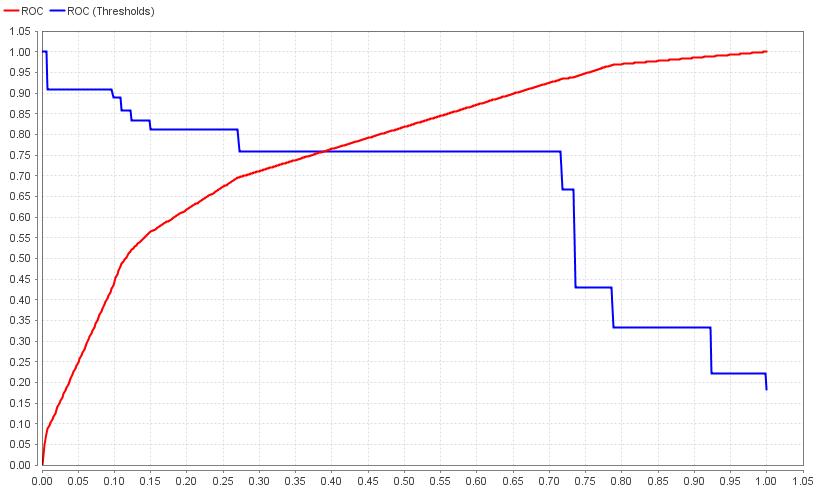
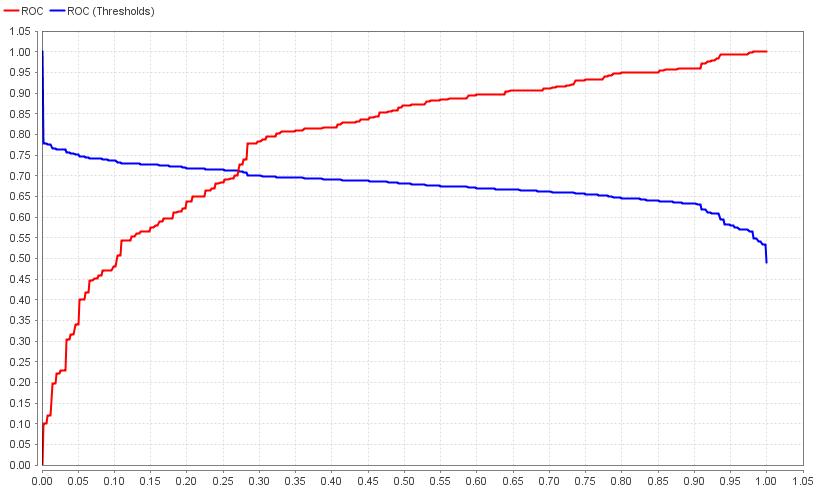
****

**AUC**

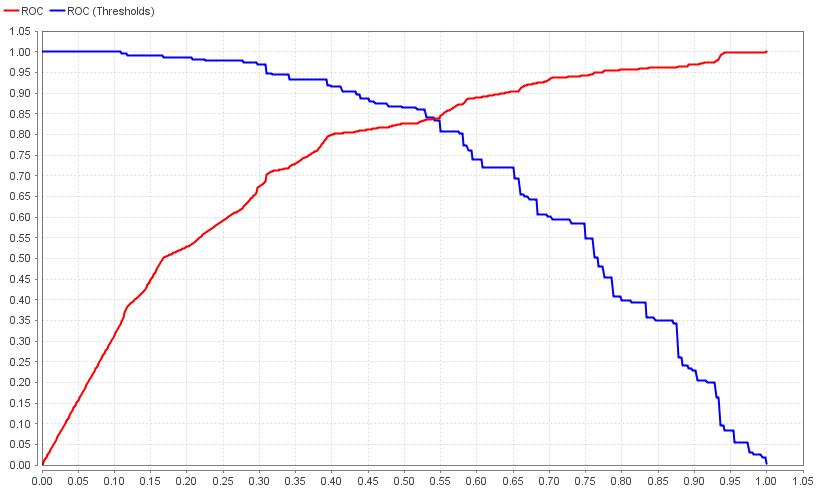
**Decision Tree – AUC: 0.613 CART – AUC: 0.740**

** **

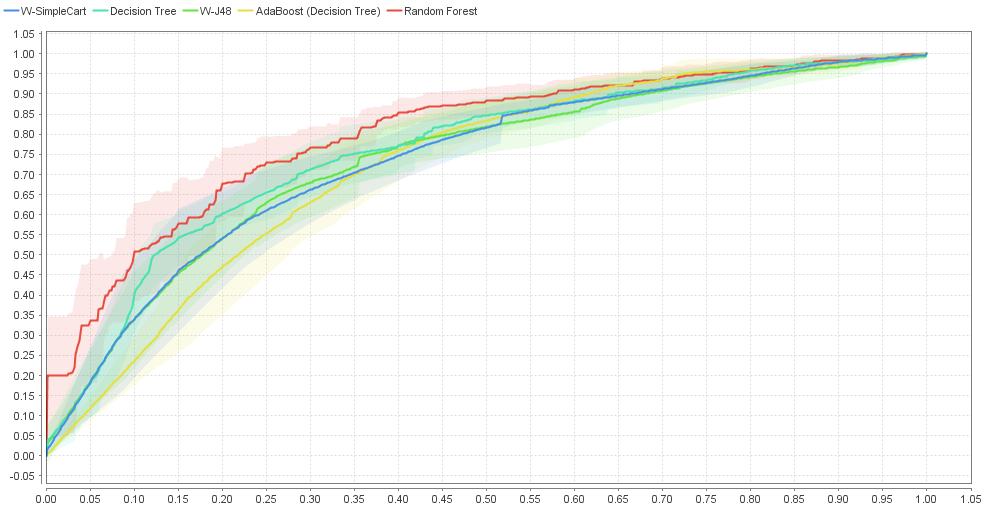
**J48 – AUC: 0.766 Random Forest – AUC: 0.790**

** **

**AdaBoost (Decision Tree) – AUC: 0.739**

****

**Comparison Using ROC**



* From the lift charts its observed that performance of ‘Random Forest’ dominates over other learners.
* From the AUC comparison, Random Forest, J-48 and AdaBoosted Decision Tree are the 3 learners with a decent chart and AUC score.
* Finally, from the ROC comparison graph its observed that ‘Random Forest’ and ‘Decision Tree’ has a better performance over other learners. ‘AdaBoosted Decision Tree’ also has a consistent increase in curve.
* Models differ mainly in their accuracies in predicting the correct classes. They also differ in tree sizes and also in their lift chart performances.

**c) Stability**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Seed value** | **Decision Tree** | **J48** | **CART** | **Random Forest** | **AdaBoost (Decision Tree)** |
| 1449 | 74.40% | 70.40% | 68.80% | 68.80% | 73.40% |
| 3559 | 69.60% | 69.00% | 72.20% | 69.20% | 72.40% |
| 1198 | 71.80% | 69.20% | 72.40% | 69.00% | 73.20% |

Its observed from the above table that with different seed values the ‘Decision Tree’ and ‘CART’ are found to be more unstable in their accuracies compared to other learners. ‘Random Forest’ and ‘AdaBoost’ are having almost consistent accuracies across different seed values.

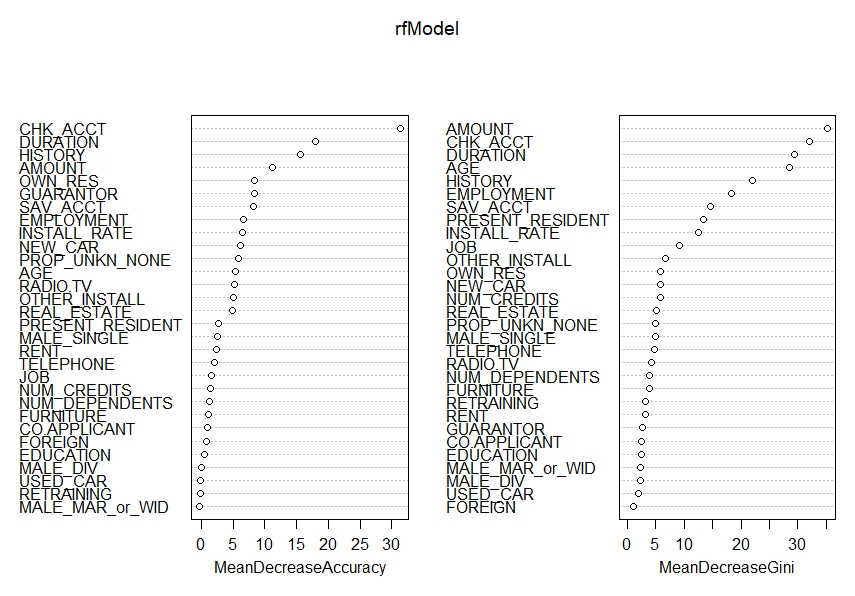
**d)**

**Testing with Different Partitions**

|  |  |  |  |
| --- | --- | --- | --- |
| **70-30 Split** | **Precision** | **Recall** | **Overall Accuracy** |
| **Decision Tree** | 72.52% | 83.42% | 69.00% |
| **J48** | 69.23% | 93.26 % | 69.00% |
| **CART** | 74.15% | 90.67% | 73.67% |
| **Random Forest** | 64.33% | 100.00% | 64.33% |
| **AdaBoost (Decision Tree)** | 72.52% | 83.24% | 69.00% |

|  |  |  |  |
| --- | --- | --- | --- |
| **80-20 Split** | **Precision** | **Recall** | **Overall Accuracy** |
| **Decision Tree** | 76.82% | 87.22% | 74.00% |
| **J48** | 74.69% | 90.98% | 73.50% |
| **CART** | 74.23% | 90.98% | 73.00% |
| **Random Forest** | 66.50% | 100.00% | 66.50% |
| **AdaBoost (Decision Tree)** | 76.82% | 87.22% | 74.00% |

* The same set of decision tree parameters were used in all the learners and it can be observed from the above table that accuracy tends to increase with more training data. Random Forest seems to be the only exception, as its observed that the accuracy tends to decrease with more data in the training set.
* It can be observed across models that all decision trees’ initial splits are made on the same attributes namely, CHECKING\_ACCOUNT, HISTORY and DURATION.



As can be seen from the graph above CHK\_ACCT, DURATION, HISTORY and AMOUNT are the variables which have a significantly positive effect on the accuracy of our model and the plot also syncs with our answer from the Q2 model.

**Conclusion: -**

From our analysis on 50-50 split performance Random Forest, Decision Tree and AdaBoost Decision Tree were found to be the best learners when compared to all other models. Although Random Forest model has the best lift chart performance and ROC comparison performance, the following drawbacks were also noted.

* + Precision decreases with increase in training data.
  + Accuracy on training data vs validation data is 71.20% and 69.20% which could result in overfitting unseen data.
  + 100% recall on all partitions of test data also shows that this model may result in overfitting.

Ada Boosted Decision Tree has

* Better precision, recall and overall accuracy and lift performance compared to normal Decision Tree.
* Accuracy tends to increase with increase in test data.
* Better stability for different seed values when compared to normal Decision Tree.

With these observations, we conclude that **Ada Boosted Decision tree** would be the ‘best’ model for implementation for the given data set.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Validation Data** | | | | | |
| **Positive Class** | **Threshold** | **Misclassification**  **costs** | **Accuracy** | **Precision** | **Recall** |
|  |  |  |  |  |  |
| **Good** | 0.6 | 50.00 +/- 0.000 | 74.00% | 75.16% | 90.98% |
| **Good** | 0.7 | 55.500 +/- 0.000 | 74.50% | 76.62% | 88.72% |
| **Good** | 0.8 | 87.000 +/- 0.000 | 73.00% | 77.24% | 84.21% |
|  |  |  |  |  |  |

Its observed that with higher threshold value precision increase with decrease in recall. Overall accuracy tends to fluctuate.

**4)**

**a)**

**Best Model:** AdaBoost Decision Tree

**Tree depth:** 9

**Number of Nodes:** 22

**Variables:** AGE, Amount, Duration, Retraining, Telephone, Checking Account, Used car, Own Residence

**b)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Probabilities of ‘Good’ and ‘Bad’** | | | |
| **AGE** | | **CHK\_ACCT** | |
| Good | Bad | Good | Bad |
| Bad:5/30 Good: 25/30 | Bad: 7/10  Good:3/10 | Bad:0/9  Good: 9/9 | Bad: 7/11  Good: 4/11 |

**c)**

**Rule 1: -**

IF (OWN\_RES > 0.5) AND (USED\_CAR> 0.500) AND (AGE <=47.5) THEN GOOD

**Rule 2: -**

IF (TELEPHONE>0.5) AND (RETRAINING<=0.5) AND (AMOUNT <= 4833) AND (AGE > 47.5) THEN GOOD

**5)** The cutoff value for the predicted probability is 92.29% to use this model to score future credit applicants.

The maximum net benefit that we are getting based on our model is 1000 DM