Understanding Credit Worthiness of Customers

Predicting Customer Credit Rating

To study the credit worthiness of customers based on a training data set and to use that knowledge to predict the credit rating of unknown customers.

Grace Ramamoorthy

Student ID: 10278389

11/26/2010

Understanding Credit Worthiness of Customers

Predicting Customer Credit Rating

Overview:

In this project, I studied the credit worthiness of customers based on a training data set. I used this training set to create a Naïve Bayes model and a decision tree model. Then used the model on the test data to predict the customer type and decide if that customer should be given a loan. This model could be used by the bank to predict the customer class before servicing credit for them thereby improving their loan repayment pattern.

Objective:

The objective is to improve the credit collection of a bank by mining the available data on existing customers and use that to predict customer types. This will improve the banks credit policy by reducing bad debts and increasing customer repayment chances. This will also reduce the time taken to do background research on customers before servicing loans by restricting additional work only on Class B and C customers.

The objective is to improve the credit collection of a bank by mining the available data on existing customers and use that to predict customer types.

Our aim is to classify the customers as A, B, C and "No Credit" class. Class A customers have a good credit standing and hence could be given loans without much of background research on the customer. Class B and C are customers with good and OK standing and will require further background research or collaterals to support their loan request. "No Credit" class customers have reached their peak credit liability and are at-risk customers and therefore should NOT be given further loans. Based on the training set, our aim is to come up with a model that will make an accurate prediction of the class of an unknown customer.

Dataset Collection and Understanding:

I have 2500 examples training data and 50 example testing data set. Both of them have 94 attributes. The training set has a class label attribute for the customers. Out of the 95 attributes, some of them, such as Gender, Zip, Town, Street, Region, Country, Phone, Fax, Email do not contribute much to the objective of the classification. All of the data is of Nominal data type. From the dataset, we can find that the Salary range, rebalanced a recently overdrawn current account, FICO Credit score, credit refused in past attributes have an effect on classification. Most customers classified as "No Credit" class have salary values 1 or 2, category B or C customers have salary values 2 or 3. Class A customers have salary value 4 or 5. We can also notice that most customers classified as Class A or B have recently paid back overdrawn current account and have good FICO credit score. If any customer was refused credit in the past then mostly they are classified C or "No Credit" class. On the other hand if a customer is not refused credit in past they have good chance for further classification. The Savings in other accounts attribute is an additional value attribute. Most of the "No Credit" customers have less than 7500 in their account. However, most of the classification is not solely based on any one attribute. So if a customer have less than 1000 in their current account and if they have better score in other attributes, then they may still be classified as B or C.

Salary range, rebalanced a recently overdrawn current account, FICO Credit score, credit refused in past attributes have an effect on classification.

Another interesting fact is that most customers are not self-employed. Very few self-employed people are part of the customer set. We can also see a pattern in the customers who have bonds type A, B C or D. These have some effect in the classification. Based on all these factors, and the information gain from each of these attributes, I came up with a workable subset of attributes.

Techniques Used:

As the data set had 95 attributes, using information gain operation on the attributes, I picked out 30 most useful attributes. I could have used up to 50 attributes but initially I started with the decision tree model and I found the tree very large to work on. Therefore, I started the other way around and picked out 20 attributes and studied the

model. I kept adding more attributes to verify the effect on the performance of the model. When I reached 30 attributes, I got the peak model performance.

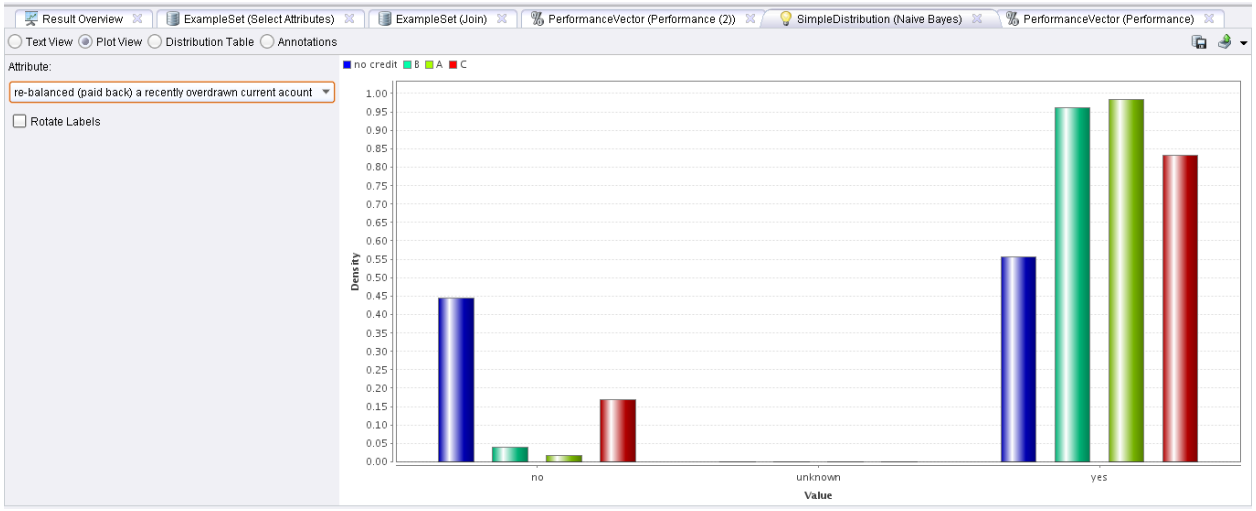
To start with I used the decision tree model, then I tried ID3 and later Naïve Bayes model. With the same set of attributes, the Naïve Bayes model showed a slightly better performance. So I decided to work with Naïve Bayes model. The model generated by the Naïve Bayes was applied on the test data set and the output from the unlabeled test data was evaluated against the labeled test data set.

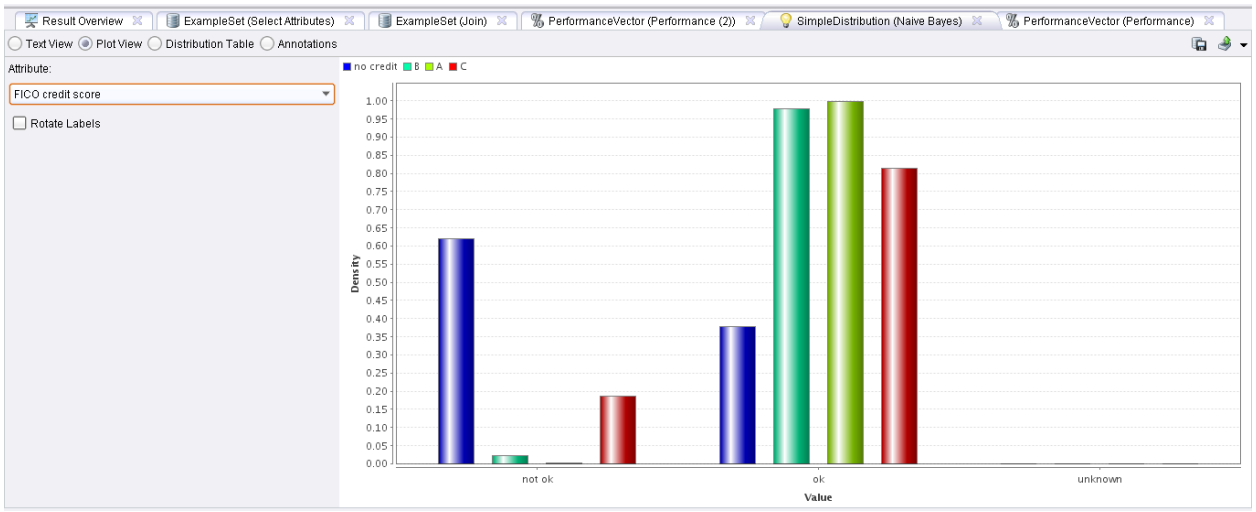
Current Project:

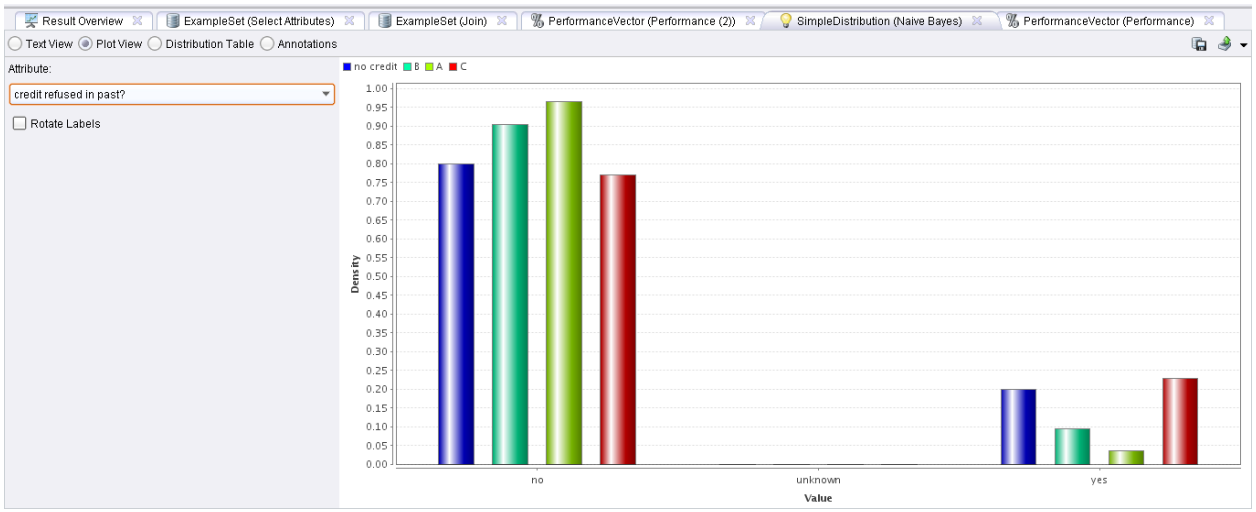
In the training data set of 2500 examples, the pattern was very predictable. Most customers are not self-employed (492 self-employed), have "OK" FICO credit score (2049 oks), recently paid back overdrawn current account- showing a good current economical standing (2129) and most were not refused credit in the past (2166). However, we also notice that some clients who were refused credit in the past are still classified as A, B or C based on other attribute values. Similarly, even if the FICO rating is "Not OK" for some clients, they are classified as A, B, or C. So a Naïve Bayes classification worked better on the model.

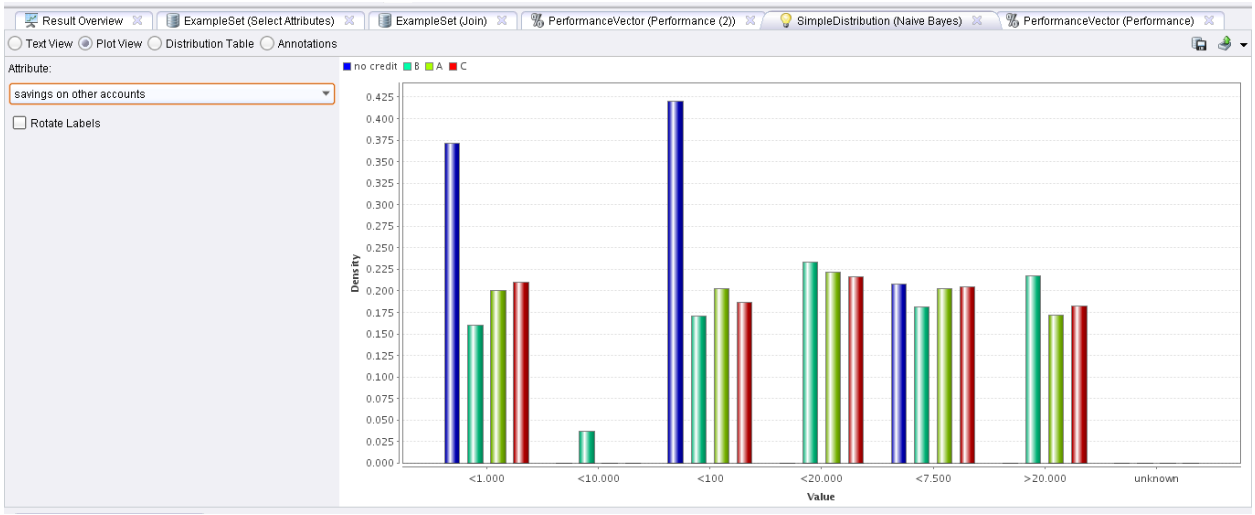
Role	Name	Type	Statistics	Range
regular	re-balanced (paid back) a recently overdrawn current account	binominal	mode = yes (2129), least = no (371)	yes (2129), no (371)
regular	FICO credit score	binominal	mode = ok (2049), least = not ok (451)	not ok (451), ok (2049)
regular	Credit refused in past?	binominal	mode = no (2166), least = yes (334)	yes (334), no (2166)
regular	savings on other accounts	nominal	mode =	20.000 (387),
regular	self-employed?	binominal	mode = no (2008), least = yes (492)	yes (492), no (2008)

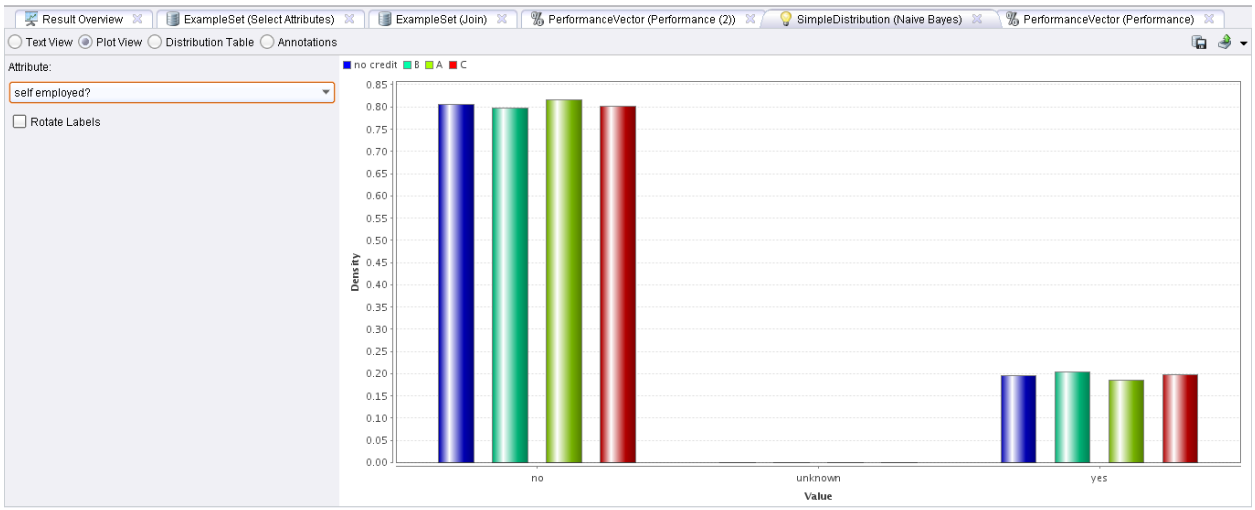
The graphs below show the distribution of data in the training set.

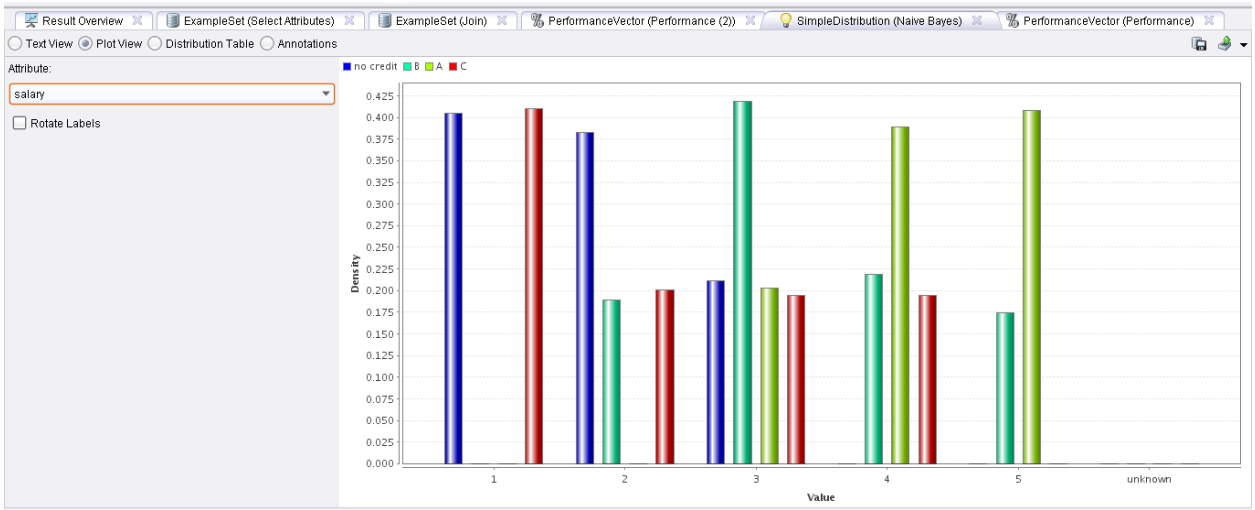






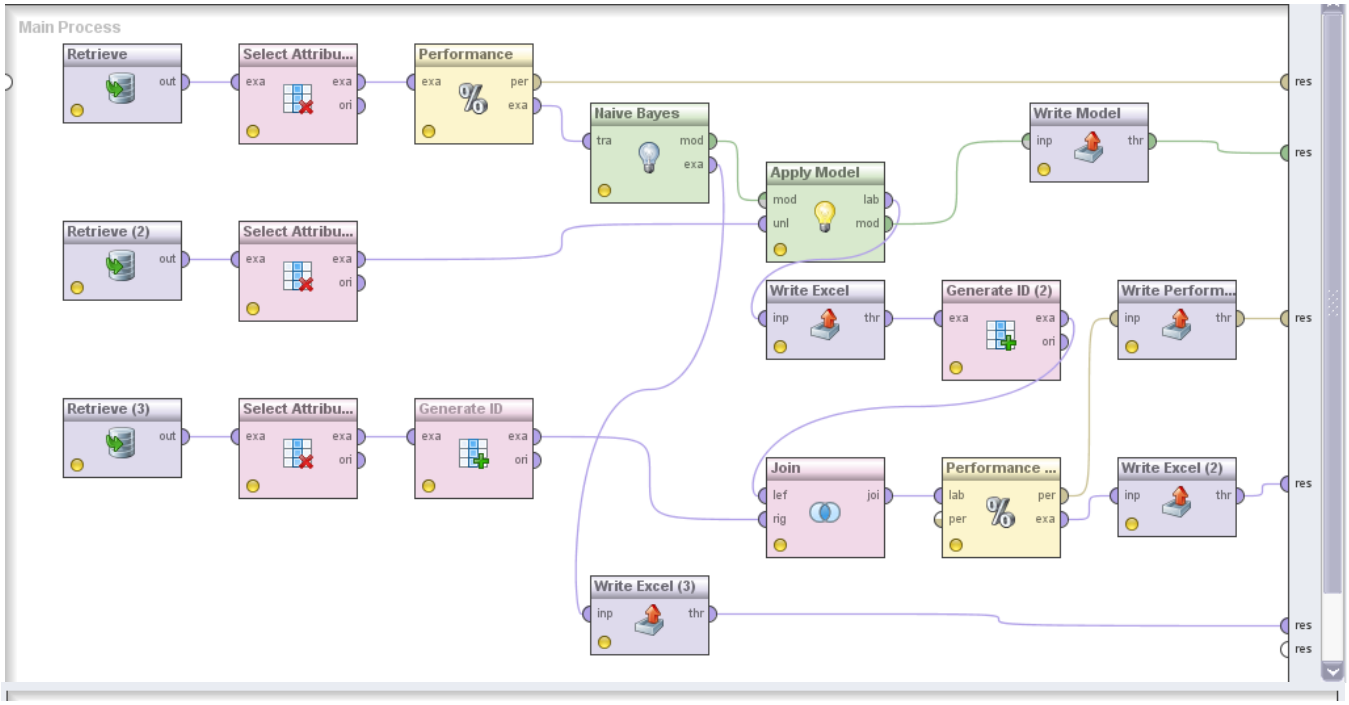

Result Overview
ExampleSet (Select Attributes)
ExampleSet (Join)
PerformanceVector (Performance (2))
SimpleDistribution (Naive Bayes)
PerformanceVector (Performance)
Text View
Plot View
Distribution Table
Annotations
Attribute:
savings on other accounts
Rotate Labels
no credit
B
A
C
Density
<1.000
<10.000
<100
<20.000
<7.500
>20.000
unknown
Value


Result Overview
ExampleSet (Select Attributes)
ExampleSet (Join)
PerformanceVector (Performance (2))
SimpleDistribution (Naive Bayes)
PerformanceVector (Performance)
Text View
Plot View
Distribution Table
Annotations
Attribute:
self employed?
Rotate Labels
no credit
B
A
C
Density
no
unknown
yes
Value


Result Overview
ExampleSet (Select Attributes)
ExampleSet (Join)
PerformanceVector (Performance (2))
SimpleDistribution (Naive Bayes)
PerformanceVector (Performance)
Text View
Plot View
Distribution Table
Annotations
Attribute:
salary
Rotate Labels
no credit
B
A
C
Density
1
2
3
4
5
unknown
Value

PROCESS:

1. On the training data, I used a “select attribute” operator, then “Naïve Bayes” operator. This created the necessary model.
2. I used the “Apply Model” operator to use the Naïve Bayes output on the unlabeled Test data set.
3. To record the performance of the model, I “Joined” the labeled test data set and the output of the “Apply Model” and used a “Performance” operator to check the performance of the above model. This compares the prediction attribute and the labeled class to list the performance statistics. The Join operation required an ID attribute which was not part of the dataset. So I used an Attribute generator to generate an ID for the data sets.
4. Naïve Bayes output is recorded in ***NaiveBayesModelOutput.xls***, the “Apply Model” output is recorded in ***NaiveBayesperformance.xls*** and ***NaiveBayesModel.mod.*** The Performance output is recorded in ***NaiveBayesResults.xls*** and ***NaiveBayesPerformanceOutput.per.***



SUITABILITY:

I tried various models before settling in for Naïve Bayes based on a slightly better performance. I have submitted results from Decision Tree classification as well as Naive Bayes for comparison sake. But my report analysis is based on Naïve Bayes output.

CONCLUSION:

Based on the training set, a Naïve Bayes model was generated and the unlabeled test data classification was predicted. I compared it with the class label on the labeled test data and got 78.43% accuracy with 90.91% accuracy for "No Credit" classification and 62.5% accuracy for Class A.

	true A	true B	true C	true no credit	class precision
pred. A	5	2	1	0	62.50%
pred. B	3	19	2	1	76.00%
pred. C	0	1	6	0	85.71%
pred. no credit	0	0	1	10	90.91%
class recall	62.50%	86.36%	60.00%	90.91%	

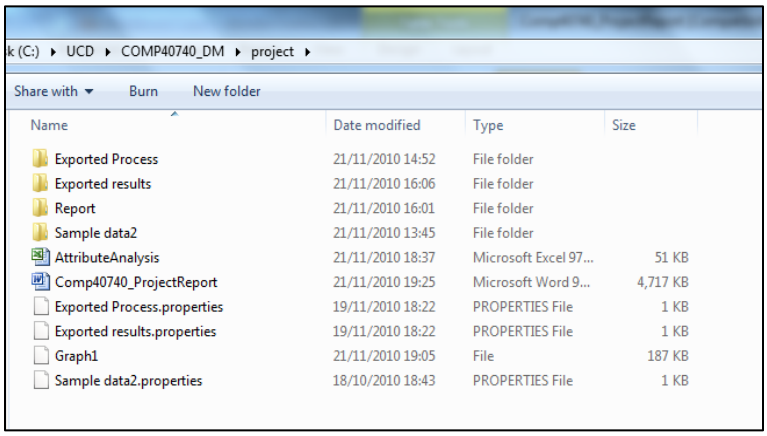
Performance Vector:				
accuracy:	**78.43%**			
Confusion Matrix:				
True:	**A**	**B**	**C**	**no credit:**
A:	5	2	1	0
B:	3	19	2	1
C:	0	1	6	0
no credit:	0	0	1	10
Classification error	**21.57%**			

ANALYSIS:

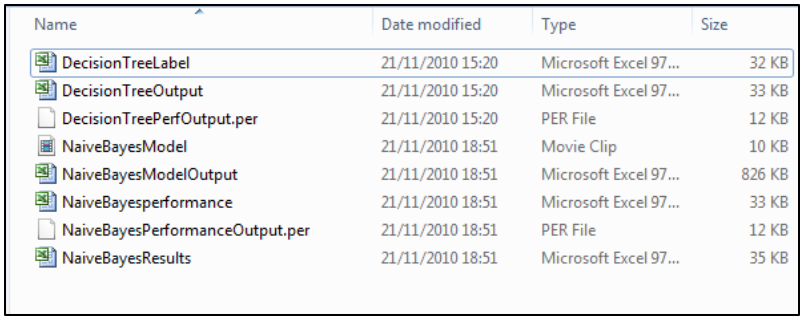
- Having a higher "No Credit" classification accuracy implies that the Bank will not provide a loan to a customer with bad loan standing thereby improving the Bank's credit collection.
- The slightly lower accuracy rate in the class A should not be a cause of concern because the Customer maybe improperly classified as class B requiring additional verification/ collateral. This would still be in the interest of the bank as the bank can provide loan to these customers based on the additional requirement. Therefore this output meets the objective well with a little additional background research.
- The 76% accuracy for Class B is a bit of concern as some of the customers are classified A. But I feel that with a larger training set and a little more study of the attributes, we will be able to overcome this hitch.

ATTACHMENTS:

The structure of the project files are listed as under:



1. Sample data2 :
   a. Data: Input data to Rapid Miner
      i. CreditData (Training Set)
      ii. testingdata(labelled testdata)
      iii. testingdata2 (unlabelled test data),
   b. Process: Rapid Miner Process generation
      i. Decision Tree
      ii. Naïve Bayes
   c. Results: Rapid Miner Result generation
      i. Decision Tree
   d. Naïve Bayes
2. Exported Process: Consists of exported process in XML from Rapid Miner
      i. Decision Tree
      ii. Naïve Bayes
3. Exported Results: Exported excel sheet outputs from Rapid Miner



Name	Date modified	Type	Size
DecisionTreeLabel	21/11/2010 15:20	Microsoft Excel 97...	32 KB
DecisionTreeOutput	21/11/2010 15:20	Microsoft Excel 97...	33 KB
DecisionTreePerfOutput.per	21/11/2010 15:20	PER File	12 KB
NaiveBayesModel	21/11/2010 18:51	Movie Clip	10 KB
NaiveBayesModelOutput	21/11/2010 18:51	Microsoft Excel 97...	826 KB
NaiveBayesperformance	21/11/2010 18:51	Microsoft Excel 97...	33 KB
NaiveBayesPerformanceOutput.per	21/11/2010 18:51	PER File	12 KB
NaiveBayesResults	21/11/2010 18:51	Microsoft Excel 97...	35 KB

4. Attribute Analysis: Worksheets with all the training set and test data output analysis from Rapid miner.

REFERENCES:

The following references were used to conduct this project and analyze the result:

- *http://www.statsoft.com/textbook/boosting-trees-regression-classification/?button=1*
- *http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf*
- *http://archive.ics.uci.edu/ml/datasets.html*
- *http://www.easydatamining.com/*
- *http://rapid-i.com/content/view/189/198/*