# Credit Scoring Data

## Background

Credit Scoring is a data set that contains the details of several people and goodness of their credit score. i.e. if their credit score is good enough to get bank loans etc. or not. This output is a categorical variable which means the outcome is either 1 indicating good credit score or 0 indicating bad credit score.

## Goal

The main aim of this report is to find a method with which we can estimate whether the credit score of a person is on par or not. i.e. to determine if a person has good enough credit score to approve bank loans etc. The performance of the solution is a key factor since the accuracy of the prediction is very important.

## Approach

In this report, we primarily concentrated on designing a Logistic Regression model and Classification Tree for predicting if credit score is good or not. We divided the data set into train and test data sets, built the model/tree on train data and validated it through the test data.

## Major Findings

Through our analysis in the report, we found that the goodness of credit score can be estimated either by using a logistic model or by a classification tree with considerably good accuracy. But by comparing some key parameters, we identified that regression tree gives some better results. Hence we feel that using Logistic Tree for predicting the median value of housing in Boston might yield good results.

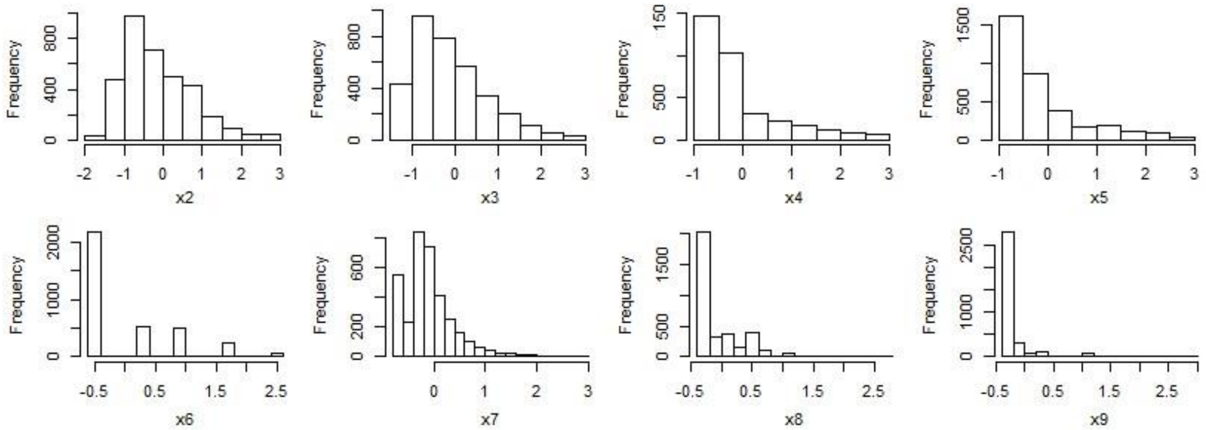
The below report has detailed analysis and approach we followed for this purpose.

**Credit Scoring Data- Exploratory Analysis**

The given training data is sampled into training data that contains 70% of original data and test data which contains remaining 30% of data. We consider training data for building a model and then test it using test data.

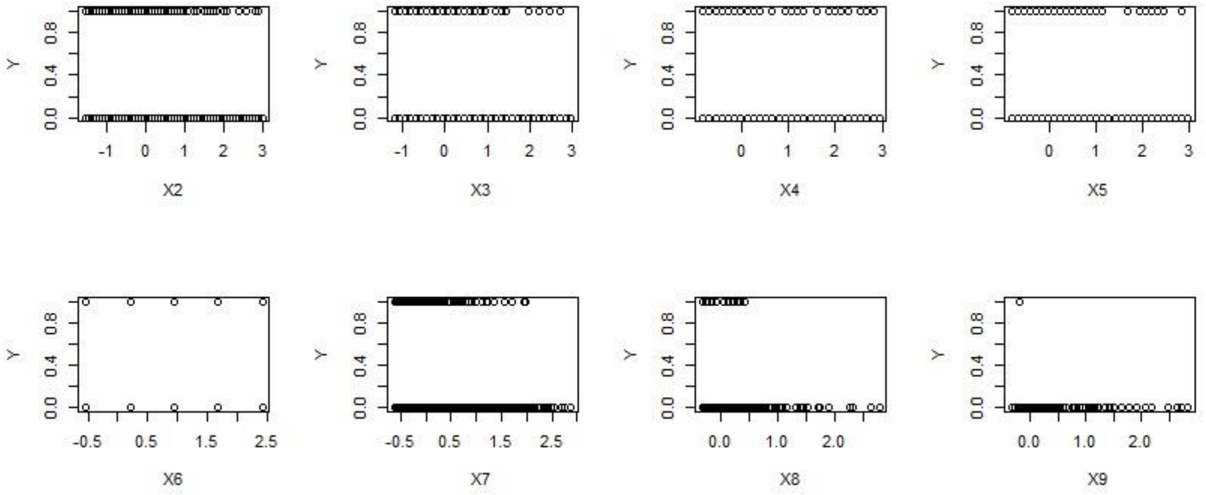
The data contains the variable id which is a unique id for each row, output variable Y which is a binary variable, independent ratio variables X2 to X9 and independent binary variables X10_2 to X24_2

Below are the summary statistics and exploratory analysis of the training data.



From the above histograms of ratio variables, most of the variables are concentrated at mostly one value.

Below are the scatter plots of each individual ratio variable with the outcome variable Y.



As we have did some analysis of variables, we need to build a model that can be used to predict Y in the future. For building a model we won’t consider **id** variable as it is clearly not an estimator of Y.

Now performing generalized linear regression for the credit data using different links. On comparing them, we can probably select a specific one for further analysis.

|  | AIC | BIC | Mean Residual Deviance |
|---|---|---|---|
| **Logit Link** | 1343 | 1725 | 0.36 |
| **Probit Link** | 1344 | 1723 | 0.35 |
| **Log-Log Link** | 1399 | 1781 | 0.25 |

By looking at the above comparison, we can go ahead using Logit link for our further analysis.

Model Selection

First, a model is built including all the ratio and binary variables. Then using stepwise variable selection another model is created. On comparing both models we can select a specific model for further analysis. Below is a table comparing both the models-

|  | AIC | BIC |
|---|---|---|
| **Model with all the variables** | 1343.44 | 1725.39 |
| **Model using stepwise with AIC** | 1310.04 | 1464.06 |
| **Model using stepwise with BIC** | 1338.09 | 1407.37 |

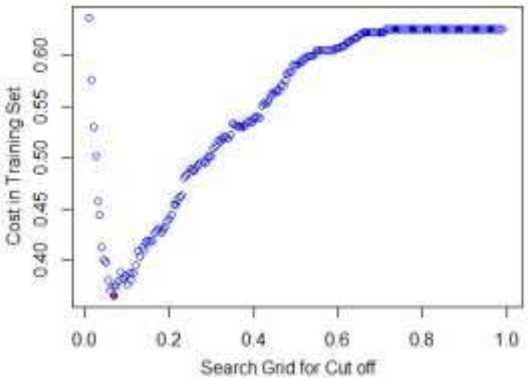
From above we select the model that came through Stepwise regression using AIC for further analysis. Below are the variables that are present in this model.

**Y ~ X3 + X8 + X9 + X11_2 + X13_2 + X15_4 + X15_6 + X16_2 + X17_3 + X17_5 + X17_6 + X18_4 + X18_5 + X18_7 + X19_3 + X19_7 + X19_8 + X19_9 + X20_3 + X21_3 + X22_2 + X22_9 + X22_10 + X23_3**

Finding cut off probability through grid search

For the selected model, let us find the optimal cut off probability so that we can have binary classification rule, to check performance on training and testing set and be able to plot ROC curve for further analysis.

We have selected the cut off probability based on minimizing the cost function in search grid. At a specific cut of probability, cost function of a given logistic model with the training data will have least value.



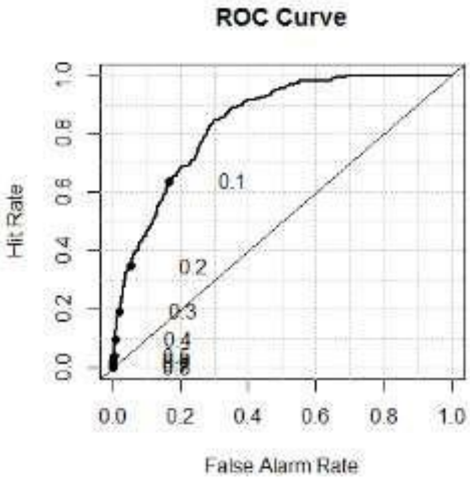
From above, the cost of the model for given training set is minimum at probability of 0.06. Therefore selecting the optimum cut off probability of **0.06.**

In sample performance of the model
Using this cut off probability and the training data set, the misclassification rate and confusion table are calculated as follows-

| **Predicted** | | |
|---|---|---|
| **Truth** | **0** | **1** |
| **0** | 2348 | 938 |
| **1** | 39 | 175 |

**Misclassification rate: 27.90%**
2348 rows of data with outcome as 0 are correctly predicted by model (**True Negative**).
175 rows of data with outcome as 1 are correctly predicted by model (**True Positive**).
938 rows of data which has actual outcome of 0 are predicted wrongly as 1 by the model (**False Positive**).
39 rows of data which has actual outcome of 1 are predicted wrongly as 0 by the model (**False Negative**).
Misclassification rate which is the percentage of outcomes that model predicted wrongly is 27.90%

From above, we identified the confusion table only for one probability which is the cut off probability. Through ROC curve, we can examine the same for the full range of cut off values from 0 to 1. With this for each possible cut off value we can create a confusion table and hence can calculate misclassification rate. Plotting the pairs of sensitivity and specificities (or, more often, sensitivity versus one minus specificity) on a scatter plot provides an ROC (Receiver Operating Characteristic) curve[1]. Below is the ROC curve for the selected model using the training data -



The ideal probability is where Hit Rate (TPR) is maximum with minimum False Alarm rate (FPR) and from above figure we can estimate that the optimum probability will be just below 0.1 which is our optimum cut off probability 0.06 which we calculated above.

The AUC for the above curve is **0.842** which indicates that 84.20% of the time a randomly selected pair of subjects will be correctly predicted by the model. Or the probability for correctly ordering a pair of subjects is 0.84.

Out of Sample performance

In the above chapter, we have developed a logistic model which is generated from the training data which is 70% of the entire Credit Scoring Data. We also checked the performance of the model using the same training data.

Now let us check the performance of the above model using the test data which is the remaining 30%
of the credit scoring data. This will be the Out of sample performance of the model.

Using the cut off probability of 0.07 which is the optimal probability, the confusion table for the logistic model with the test data is –

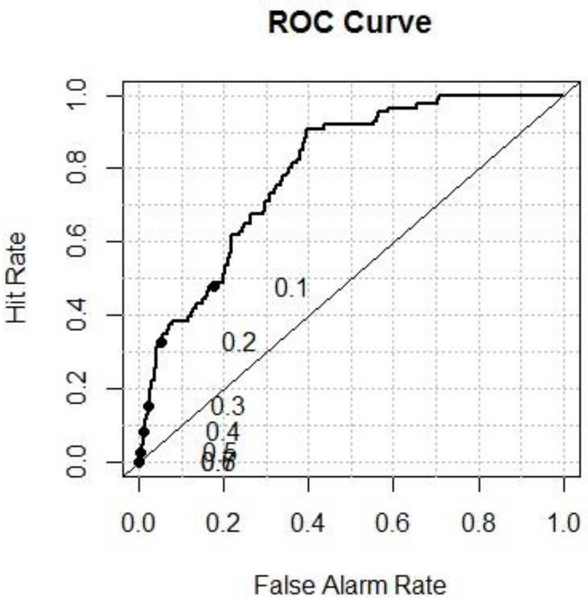
| **Predicted** | | |
|---|---|---|
| **Truth** | **0** | **1** |
| **0** | 1001 | 413 |
| **1** | 28 | 58 |

**Misclassification rate: 29.40%**

1001 rows of data with outcome as 0 are correctly predicted by model (**True Negative**).
58 rows of data with outcome as 1 are correctly predicted by model (**True Positive**).
413 rows of data which has actual outcome of 0 are predicted wrongly as 1 by the model
(**False Positive**).
28 rows of data which has actual outcome of 1 are predicted wrongly as 0 by the model (**False Negative**).
Misclassification rate which is the percentage of outcomes that model predicted wrongly is
29.40%

The misclassification rate of model using out sample data is 29.40% which is greater that the misclassification rate of model using in sample data. This is because the model is developed using in sample data and hence it has low error rate for it. But the out sample data is completely new to the model which results in slight increase in misclassification rate.

Now plotting the ROC curve for this model using the out of sample data i.e. test data



The ROC curve and calculated AUC in the above chapter are using the training data. So those values will be a bit good since the same training data is used for creating the model.

Above is the ROC curve using the test data and we can estimate that the optimum probability will be just below 0.1 which is our optimum cut off probability 0.06.

The AUC for the above curve is **0.795** which indicates that 79.5% of the time a randomly selected pair of subjects will be correctly predicted by the model. Or the probability for correctly ordering a pair of subjects is 0.79.

Below is a brief table comparing the AUC and Misclassification rates of model with in and out samples-

| | Misclassification Rate | AUC |
|---|---|---|
| **In Sample** | 27.90% | 0.842 |
| **Out Sample** | 29.40% | 0.795 |

It is clear that the AUC of test data is smaller than the AUC of training data. This is because, as the model is developed from training data, AUC yields better values since the model already knows the data. But the developed model does not know the test data and hence yields lower AUC value. Same is the case with Mis Classification rate also. Since model is generated from train data, mis classification rate will obviously be less on In Sample.

### Cross Validation of Logistic Model

In cross validating the Final Model, we check the performance of the Model in predicting the outcome variables. In above section, the model is generated on the train data and the MSE (or misclassification here) is calculated on the train data. The problem here is that-

Train data and the Test data are randomly split from the original data.
Because of this random split, both the data sets might not have proper conformity data.
I.e. data such as outliers might be concentrated in only one data set.

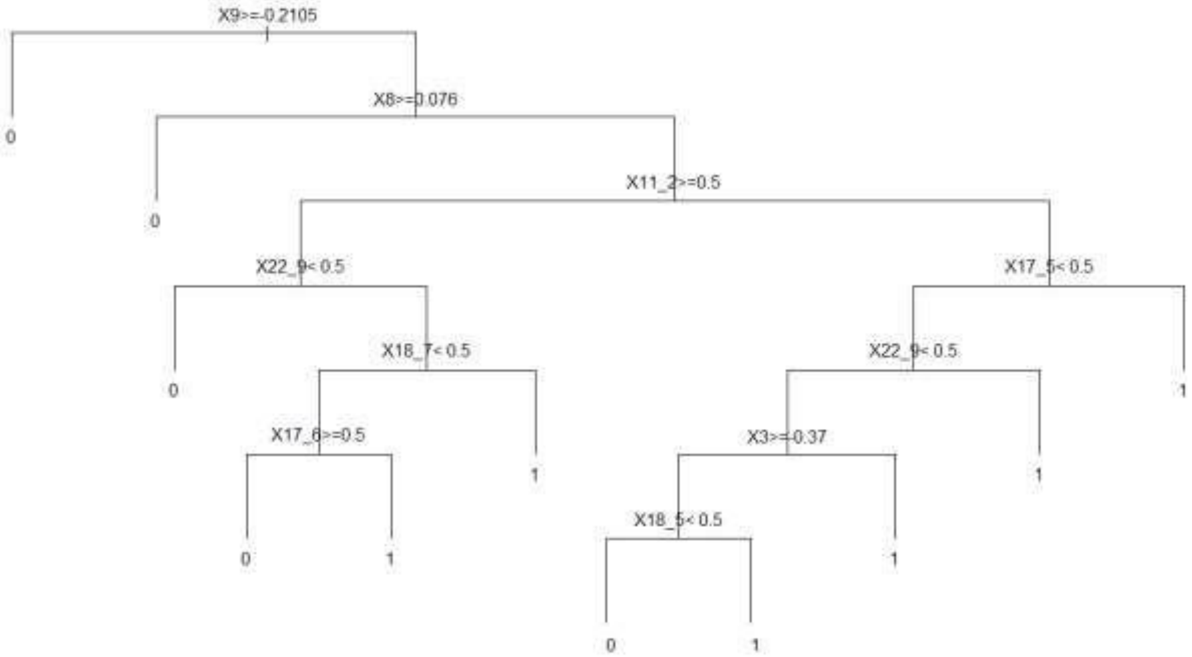
So to avoid the extent of these issues, we can calculate the MSE of our model through k-fold cross validation. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation[2].

The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross- validation is commonly used.
Misclassification rate is 11.20%
3025 rows of data with outcome as 0 are correctly predicted by model (**True Negative**).
83 rows of data with outcome as 1 are correctly predicted by model (**True Positive**).
261 rows of data which has actual outcome of 0 are predicted wrongly as 1 by the model (**False Positive**).
131 rows of data which has actual outcome of 1 are predicted wrongly as 0 by the model (**False Negative**).
Misclassification rate which is the percentage of outcomes that model predicted wrongly is 11.20%

The calculated MSE using 5-fold cross validation is 33.90 but the MSE from out sample is **29.40.** We can say that the MSE resulted from cross validation is a bit more accurate that the MSE from out sample since the cross validations takes 10 sub samples and calculates MSE for each one. Where as in Out sample we use only 1 sample of test data.

Classification tree

The classification tree for the credit scoring data is as follows-

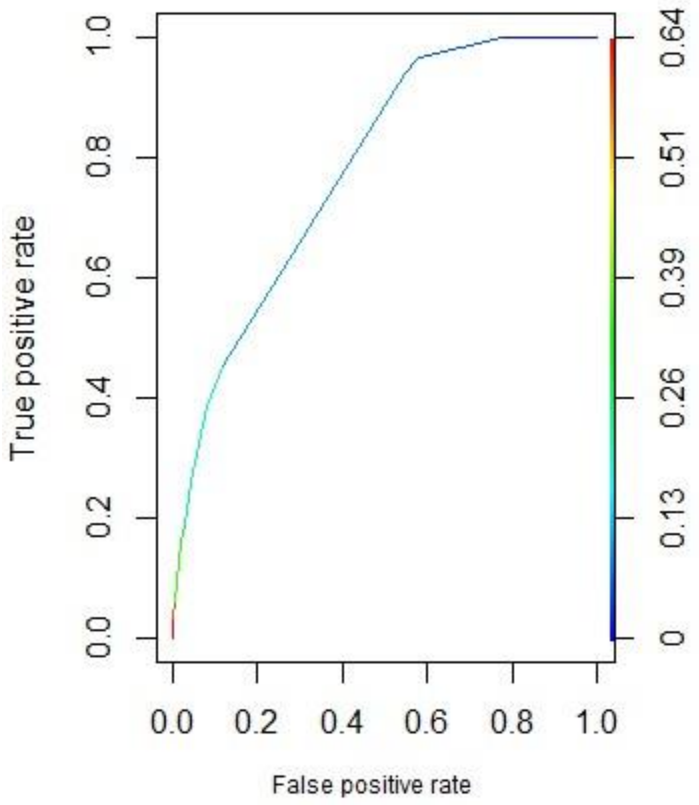


In sample performance of the classification tree

Using loss matrix (5:1) and the training data set, the misclassification rate and confusion table are calculated as follows-

| **Predicted** | | |
|---|---|---|
| **Truth** | **0** | **1** |
| **0** | 3025 | 261 |
| **1** | | 131 |

possible cut off value we can create a confusion table and hence can calculate misclassification rate. Plotting the pairs of sensitivity and specificities (or, more often, sensitivity versus one minus specificity) on a scatter plot provides an ROC (Receiver Operating Characteristic) curve[3]. Below is the ROC curve for the selected model using the training data -



The AUC for the above curve is **0.781** which indicates that 78.10% of the time a randomly selected pair of subjects will be correctly predicted by the model. Or the probability for correctly ordering a pair of subjects is 0.78.

Out of Sample performance of the classification tree
In the above chapter, we have developed a classification tree which is generated from the training data which is 70% of the entire Credit Scoring Data. We also checked the performance of the model using the same training data.

Now let us check the performance of the above tree using the test data which is the remaining 30%
of the credit scoring data. This will be the Out of sample performance of the model.

Using loss matrix (5:1) and the training data set, the misclassification rate and confusion table are calculated as follows-

**Predicted**

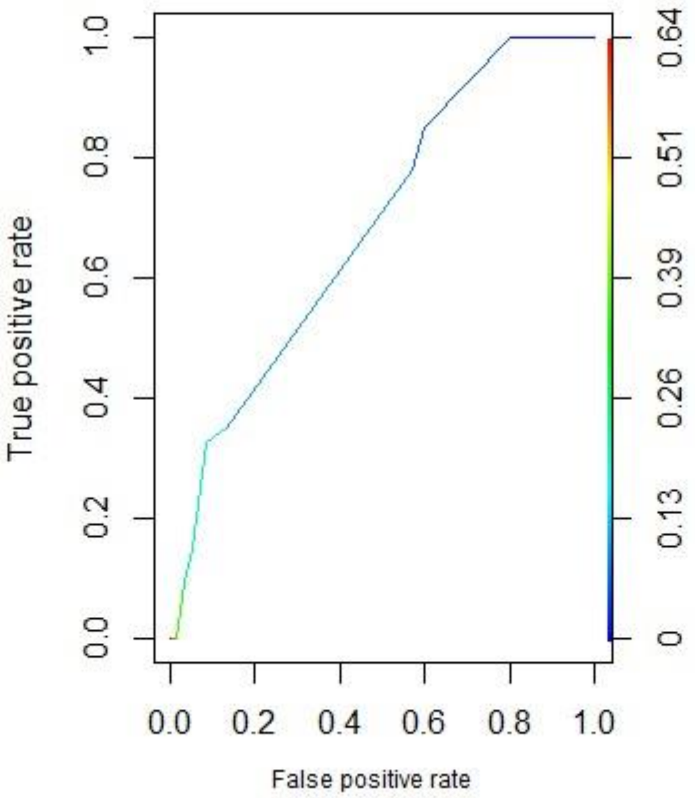
| **Truth** | **0** | **1** |
|---|---|---|
| **0** | 1293 | 121 |
| **1** | 58 | 28 |

**Misclassification rate: 11.90%**

1293 rows of data with outcome as 0 are correctly predicted by model (**True Negative**).
28 rows of data with outcome as 1 are correctly predicted by model (**True Positive**).
121 rows of data which has actual outcome of 0 are predicted wrongly as 1 by the model (**False Positive**).
58 rows of data which has actual outcome of 1 are predicted wrongly as 0 by the model (**False Negative**).
Misclassification rate which is the percentage of outcomes that model predicted wrongly is 11.90%

The misclassification rate of classification tree using out sample data is 11.90% which is greater that the misclassification rate of tree using in sample data. This is because the tree is developed using in sample data and hence it has low error rate for it. But the out sample data is completely new to the tree which results in slight increase in misclassification rate.

Now plotting the ROC curve for this model using the out of sample data i.e. test data



The AUC for the above curve is **0.713** which indicates that 71.30% of the time a randomly selected pair of subjects will be correctly predicted by the model. Or the probability for correctly ordering a pair of subjects is 0.71.

Below is a brief table comparing the AUC and Misclassification rates of model with in and out samples-

| | **MisClassification Rate** | **AUC** |
|---|---|---|
| **In Sample** | 11.20% | 0.78 |
| **Out Sample** | 11.90% | 0.71 |

It is clear that the AUC with test data is smaller than the AUC with training data. This is because, as the as the tree is developed from training data, AUC yields better values since the tree already knew the data.

But the developed tree does not know the test data and hence yields lower AUC value. Same is the case with Mis Classification rate also. Since tree is generated from train data, mis classification rate will obviously be less on In Sample.

Comparison of Classification tree with Logistic model

On simple comparison of out sample Misclassification rate, AUC and cost of generated Classification Tree and Logistic Model for the Credit risk data, it is clear that Classification Tree has better out of sample performance when compared with Linear Regression model.

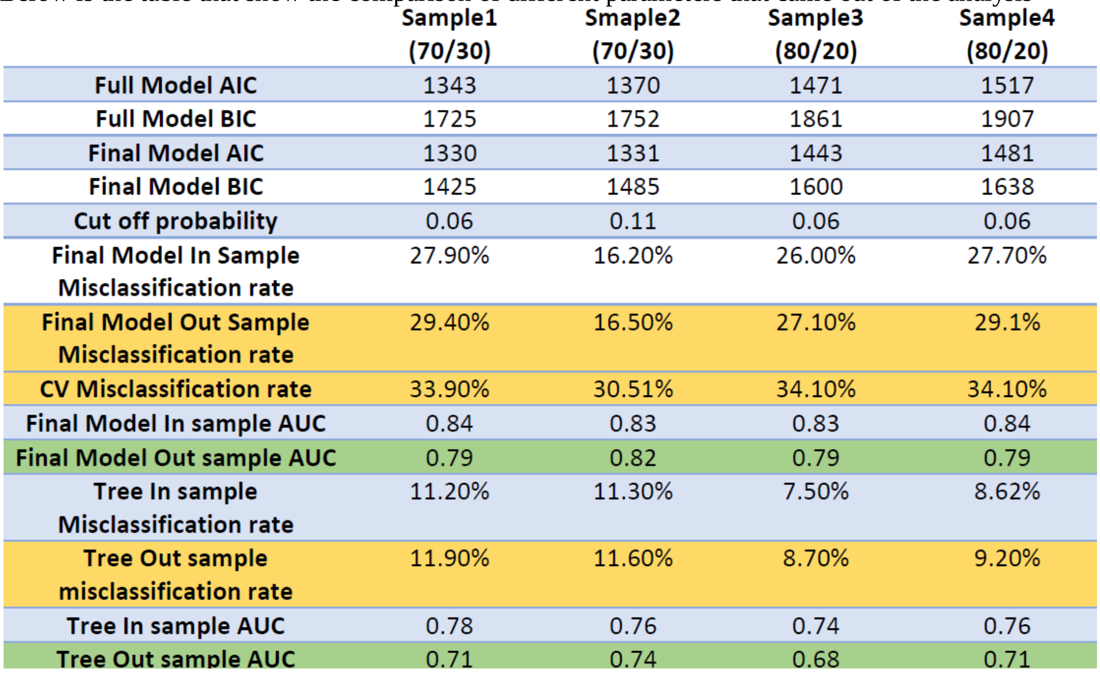
| | **MisClassification Rate** | **AUC** | **Cost** |
|---|---|---|---|
| **Logistic Model** | 29.40 | 0.79 | 0.46 |
| **Classification Tree** | 11.9 | 0.71 | 0.41 |

Our analysis shows that the logistic model approach is better suited than the classification and regression trees method for explaining the credit risk of banks. The quality of the performance of Classification tree out of the training data is good when compared with that of the Logistic model. Hence we can say that CART approach is worth being further investigated.

Comparing Different Samples

The entire analysis we did in the above chapters is based on single set of randomly sampled train and test data (Sample1). Now let us repeat the same analysis for different other randomly sampled train and test data from the credit data.

Below is the table that show the comparison of different parameters that came out of the analysis-



| | **Sample1 (70/30)** | **Smaple2 (70/30)** | **Sample3 (80/20)** | **Sample4 (80/20)** |
|---|---|---|---|---|
| **Full Model AIC** | 1343 | 1370 | 1471 | 1517 |
| **Full Model BIC** | 1725 | 1752 | 1861 | 1907 |
| **Final Model AIC** | 1330 | 1331 | 1443 | 1481 |
| **Final Model BIC** | 1425 | 1485 | 1600 | 1638 |
| **Cut off probability** | 0.06 | 0.11 | 0.06 | 0.06 |
| **Final Model In Sample Misclassification rate** | 27.90% | 16.20% | 26.00% | 27.70% |
| **Final Model Out Sample Misclassification rate** | 29.40% | 16.50% | 27.10% | 29.1% |
| **CV Misclassification rate** | 33.90% | 30.51% | 34.10% | 34.10% |
| **Final Model In sample AUC** | 0.84 | 0.83 | 0.83 | 0.84 |
| **Final Model Out sample AUC** | 0.79 | 0.82 | 0.79 | 0.79 |
| **Tree In sample Misclassification rate** | 11.20% | 11.30% | 7.50% | 8.62% |
| **Tree Out sample misclassification rate** | 11.90% | 11.60% | 8.70% | 9.20% |
| **Tree In sample AUC** | 0.78 | 0.76 | 0.74 | 0.76 |
| **Tree Out sample AUC** | 0.71 | 0.74 | 0.68 | 0.71 |

Since the test data is randomly generated every time, the out sample misclassification rate is little bit different of each set.

For cross validation, since we use the entire credit data, the CV misclassification rate is almost the same in each case.
When considering the misclassification rate, the performance of Classification tree is considerably very good when compared with that of the Logistic model.
When AUC is considered, the logistic model has little bit better values when compared with
Classification tree.