****

**Predictive Analytics on Financial Data**

**IST 687 Applied Data Science**

**Professor Erik Anderson**

**Pankaj Bathija, Pratyush Kulwal**

**Fall 2015**

**Business Questions**

Interest area: With advent of internet, the rate of online transactions has exponentially increased which has lead to generation of huge amounts of transactional data. This data is being used by financial institutions for determining various hidden patterns and insights which helps financial institutions to take better financial decisions.

We are interested in performing predictive analytics using statistical models specially on financial data, as our interest area lies in Data Driven Finance.

Credit score plays an important role in order to acquire loan for an individual from financial institutions. For banks, it is crucial to determine whether an individual will be able to repay borrowed loan. Credit scoring is one of methods by which banks determine ability of an individual; whether the person should be granted a loan or not.

Following is the dataset we have chosen for this project.

<https://www.kaggle.com/c/GiveMeSomeCredit/data>

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| SeriousDlqin2yrs | Integer | Person experienced 90 days past due delinquency or worse |
| RevolvingUtilizationOfUnsecuredLines | Percentage | Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits |
| Age | Integer | Age of borrower in years |
| NumberOfTime30-59DaysPastDueNotWorse | Integer | Number of times borrower has been 30-59 days past due but no worse in the last 2 years. |
| DebtRatio | Percentage | Monthly debt payments, alimony,living costs divided by monthy gross income |
| MonthlyIncome | Real | Monthly income |
| NumberOfOpenCreditLinesAndLoans | Integer | Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards) |
| NumberOfTimes90DaysLate | Integer | Number of times borrower has been 90 days or more past due. |
| NumberRealEstateLoansOrLines | Integer | Number of mortgage and real estate loans including home equity lines of credit |
| NumberOfTime60-89DaysPastDueNotWorse | Integer | Number of times borrower has been 60-89 days past due but no worse in the last 2 years. |
| NumberOfDependents | Integer | Number of dependents in family excluding themselves (spouse, children etc.) |

With above collected data, we will be using various credit scoring algorithms and performing different exploratory analysis to find answers to following questions about above mentioned data:

1. The most important finding of this case study is to find out whether or not a person will experience a financial distress in next two years?
2. Among the defaulters, what is the salary-range which attracts more defaulters than others?
3. Which is major factor among this data that results in more defaulters than any other factor?
4. What is the relationship between defaulters and debtratio factor of each customer?
5. Does number of family members impact defaulter rate?
6. Out of three data variables of NumberOfTime30-59DaysPastDueNotWorse, NumberOfTime60-89DaysPastDueNotWorse and NumberOfTimes90DaysLate, which one will impact the most to defaulter list?
7. Does monthly income play an important role to defaulter list?
8. What is average monthly income of a person based on age bands?
9. How much does NumberOfOpenCreditLinesAndLoans variable impact debt ratio of person?
10. Does NumberOfOpenCreditLinesAndLoans variable has a positive effect on defaulter list?

We would like to follow 4 A’s approach to our plan at this point. We might change this format at later stage based on requirements of case study, but more or less these will remain the core structure for our project.

**Data Architecture:**

***Potential Stakeholders*:**

Banks, Financial Institutions providing loans, Credit card companies providing cards using above data, National agencies keeping track of debt ratios of citizens, agencies keeping track of defaulters, etc.

The main idea of data architecture is to help build a process which will organize the data and routed to support the analysis, visualization and presentation of data to above mentioned stakeholders.

In this case study, we have around 11 variables present in data. As the data needs no external transformations because of its comma delimited format, we will use the data directly without performing any *external* transformations to it.

For performing all the operations and analysis for this case study, we will mainly be using R-Studio powered by R programming language. Processes like cleaning, transformations, exploratory analysis, model implementation will be done in R. For visualizations, we will be using R and other visualization tools (depends on tools allowed and we can integrate with R).

In R, we will be using different R packages and functions which focus on plotting relations and answering our questions visually like ggplot2, histogram, bar-plot, etc.

**Data Acquisition:**

The role of data scientist in task of data acquisition is not only to acquire data but also to know how it will be represented before any analysis or visualizations.

Our data is provided by Kaggle, which is a platform for data scientist to tackle various datasets and participate in competitions based on data analytics and data science.

We will use using different variables for this case study and transforming results based on their data types. For example, our variable which shows whether a person is defaulter or not is **SeriousDlqin2yrs**, which can be converted to factor of classification answering in terms of YES or NO.

We will take machine learning approach for solving this case study to determine one of question of whether a person will be defaulter or not. We will train a sample of data with various models, and select best fitted model and then test the remaining sample of data based on the selected applied model.

**Data Analysis:**

We have included tasks of data cleaning, data mitigation, data analysis and data visualizations in this part.

***Data Cleaning***

The data is provided by Kaggle which is a real-word anonymous financial data. This data appears to be clean at first glance but on closer inspection we found that the input data had many discrepancies.

On observing we found that:

1. **Error in Data:** These errors are majorly typo errors in the user data. For example, we found that some of the ages were mentioned as 0 in the dataset which suggested that the values are not entered correctly.
2. **Missing Values:** Some entries in the data-set are marked as “N/A “, so there might be a need required to change or fill these values before running the prediction.

Columns such as *“NumberRealEstateLoansorLines”* and “*NumberofDependents”* had some “N/A” values.

1. **Coded Values:** Some of the quantitative values were actually coded for qualitative meanings. For example: Values 98 and 96 in the *“NumberOfTime3059DaysPastDueNotWorse*” were actually holding some qualitative meaning as it was representing the value "RevolvingUtilizationOfUnsecuredLines" as 0.999999 which is bringing the skewness in the data

***Data Mitigation***

Now, in order to make predictions better it is very important to clean the data and this choice of data imputation may highly change the prediction accuracy.

During our data modelling and predictions, we will be using 3 ways of data imputation:

1. **Leave the data as alone:** Here we will not be changing the data and will keep the data as it is and try to run our statistical model.
2. **Fill the missing values by median values:** This will be the second approach by us to handle the missing values in which we can substitute the missing values by mean or the median values.
3. **Code the missing values by a specific number:** Some statistical models do not take the “N/A” as an input value. (For example: Random forest method). In such cases, we can replace these N/A values by some specific number (For example: -1)

These are few of the data cleaning tasks, data mitigation and data transformations tasks we observed in data, we will add few more as we move ahead with project. Apart from these, we will also perform few basic data cleaning processes on data.

***Data Analysis***

For data analysis, we will be use machine learning approach as mentioned above. We will try to answer all the business questions mentioned above with different approaches. For prediction, we will try to train and test data based on popular models used in industry for credit scoring such as Random Forest.

For other operations, we will perform exploratory analysis using correlations, finding out how correlated the variables are with each other. To find out the causation of data and its effects on other factors. Also, find out various inferences using basic functions of R such as median, mean and representing data based in various formats such as tables.

Depending upon the statistical model we will be choosing the data imputation technique. We will also try and use association rules wherever they could be applied. Based on which we can find different set of rules, confidence, support and lift of these rules. Making inferences and assumptions based on these rules.

***Data Visualization:***

In the Analysis part, we will be also visualizing our data-set by plotting the data using R.

Some of the tentative visualizations would be:

* Plotting salary-range and calculating number of defaulters in that range (Using Training data).
* Make a table of number of dependents and defaulters and plot this data as a bar – graph
* Visualizing data of number of times (90 days late - range) and calculating number of defaulters in that range.
* Visualizing data of number of Loans and defaulters in that range.
* Visualizing age-range and defaulters of loans
* Visualization based on average salary for different ages.
* Representing effect of number of opened line variable on defaulter list.
* Plotting of number of family members and defaulters list.

**Data Archival:**

We will process this step based on later requirements and consultation with professor.

**Preliminary Project Report**

In this part of report, we are mainly focusing on data cleaning, data linking and data screening processes to be applied on our financial dataset.

As mentioned in our previous reports, we need to clean the data and use remove or mitigation strategies to improve the quality of dataset involved. We also need to transform the dataset variable names in order to maintain simplicity and understandability.

In below report, we will load the dataset in R and then perform various tasks mentioned above.

***Data Loading***

As we are using machine learning approach, we have the data in two parts: training and testing.

In order to build a model, we will be applying all the transformations on our training dataset; so we will load this training dataset in R.

We need to set our working directory on path where our training dataset is present.

**Code:**



**Explanation:**

We are loading our dataset in two data frames

1. “bank\_train” using read.csv function using dataset “cs-training.csv”
2. “bank\_test” using read.csv function using dataset “cs-test.csv”

***Data Viewing/Screening before cleaning***

We will try and analyze this dataset, first step is to screen the dataset and think of various cleaning techniques that can be used on this dataset.

For screening the dataset, we will use View command to look at the dataset

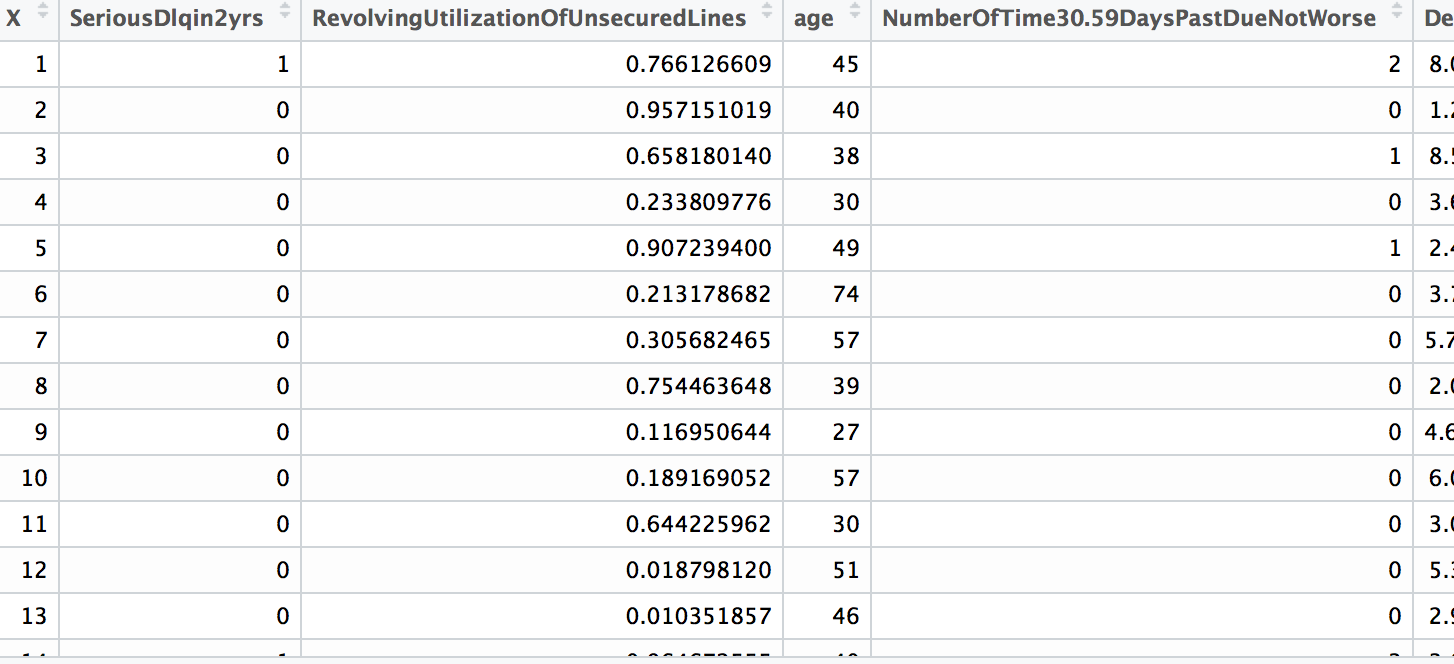
**Code:**



**Explanation:**

View command invokes a spreadsheet style data viewer on a matrix like R object

**Screenshot:**



Above screenshot is incomplete image of view command applied on our dataset.

We will try to view the structure of dataset “bank\_train” and analyze the number of rows and columns involved

**Code:**



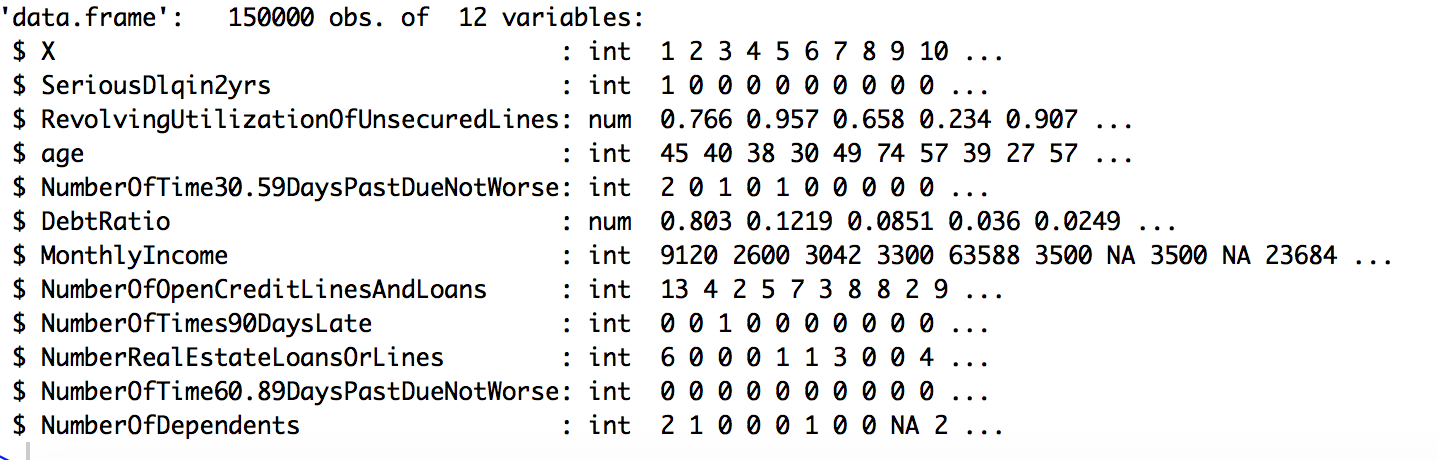
**Explanation:**

It provides the internal structure of dataset. it shows all the datatypes of all the variables involved. It also gives you the number of rows and columns involved in dataset.

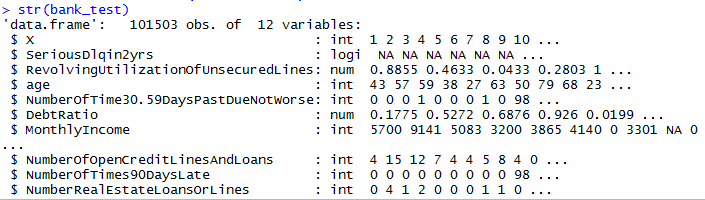
**Observation:**

As we can observe, there is a column “X” in dataset, which is just the index number in dataset.

There are 150000 observations of 12 variables involved in dataset.



Similarly, we will be viewing our testing dataset

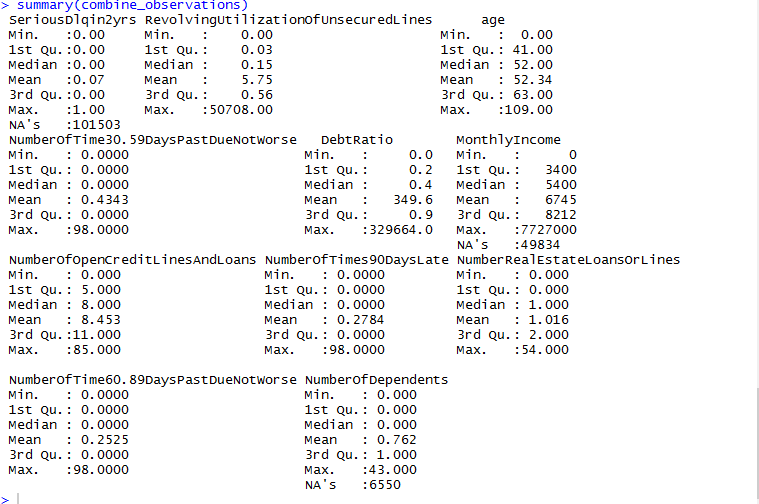


***Data Cleaning***

**Cleaning columns involved in dataset**

For cleaning our training and testing dataset, we will be combining both the datasets into one unified dataset called as “combine observations”.





We will analyze column by column based on above summary observation, thus we will be try to cleanse each column to better our results

***Variable:*** **RevolvingUtilizationOfUnsecuredLines**

* **Description:** Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits



* **Observation:** The data looks unevenly distributed as utilization of unsecured lines should be between 0-1, but in this case it is above 50000. So we will be removing such outliners from data.
* **Code:**



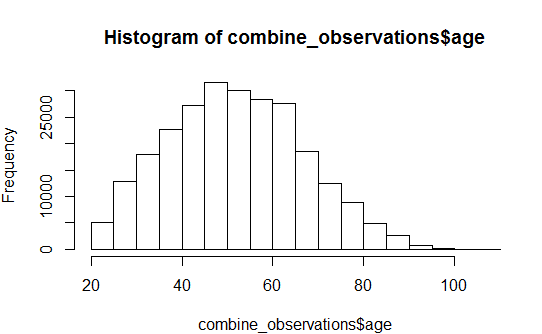
***Variable: Age***

* **Description:** Age of Borrowers in years



* **Observation:** The data looks fairly distributed when we run the histogram command. Our initial observations also reveal that was an observation with an age zero, therefore while cleaning the data we removed this observation.





***Variable:* NumberOfTime30-59DaysPastDueNotWorse*,* NumberOfTime60-89DaysPastDueNotWorse*, NumberOfTimes90DaysLate***

* **Description:**
* Number of times borrower has been 30-59 days past due but no worse in the last 2 years.
* Number of times borrower has been 60-89 days past due but no worse in the last 2 years.
* Number of times borrower has been 90 days or more past due.
* **Observation:** Using the histogram, boxplot and table function to observe the data for 30-59\_due\_default\_number. Some of the quantitative values were actually coded for qualitative meanings. For example: Values 98 and 96 in the “30-59\_due\_default\_number” were actually holding some qualitative meaning

**Code:**





We will be filling the NA values by using gradient boosting imputation, which we will be discussing in the later sections of the report.







Similarly, filling all the values by using gradient boosting imputation method.







***Variable: Monthly Income***

* **Description:** Monthly income
* **Observation:** On initial observation of the data, we observed that there were some missing values in our data therefore, we replaced these N/A values by the gradient boosting imputation based on the remaining data.



***Variable: Debt Ratio***

* **Description:** Monthly debt payments, alimony,living costs divided by monthy gross income
* **Observation:** As we considered monthly income as 200000 for an average person in United states. After researching on few articles, we found that average debt ratio we would take is 7000. This is assumption to better our model, which can be changed later.

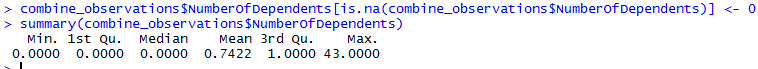


***Variable: Number of Dependent****s*

* **Description:**  Number of dependents in family excluding themselves (spouse, children etc.)
* **Observation:** On observing the data we can see that there are 3924 values as N/A values in the data frame. Now, we are replacing these 3924 values by 0 considering the fact, people with no dependents might have skipped or ignored this field.

Using Summary function, we found the following results





We have cleaned each and every variable involved in dataset and transformed wherever required.

**Understanding GBM Imputation**

Now, in order to fill the missing values in our dataset, we will be using gradient boosting imputation method.

Imputation using Boosted Trees Fill each column by treating it as a regression problem. For each column i, use boosted regression trees to predict i using all other columns except i. If the predictor variables also contain missing data, the gbm function will itself use surrogate variables as substitutes for the predictors.

It gradient boosting machines, an ensemble of decision trees, to fill missing values. GBM is particularly powerful because it can handle categorical data as well

**Why did we use this imputation?**

Since, we are working with lots of N/A values in our dataset, we did not want to lose a significant amount of our training and testing dataset.

Another reason to choose an imputation method was submission error on Kaggle. When we used our prediction model on the testing dataset, there was a submission error as our model was predicting NA as probability, which was clearly unacceptable.

**How did we use this imputation?**

Firstly, installing the prerequisite packages.



Now, in order to use this package, we had to fetch this package from the CRAN archive, as this package is no longer available in the CRAN repository.

The direct link to fetch the package is available as under:

<https://cran.r-project.org/src/contrib/Archive/imputation/imputation_2.0.1.tar.gz>

Once downloaded,



Source the package in your working directory and install the package.



Initializing the package.



Now, we had used an imputation on complete testing and training dataset.



In the above line, we are using gbmImpute technique which is basically boosting technique for imputation.

We have ignored our first column of the data, as we don’t want our response variables to get altered by any means as in the testing dataset, all SeriousDlqin2yrs are listed as ‘NA’.

Since our main aim is to predict defaulters on the testing dataset.

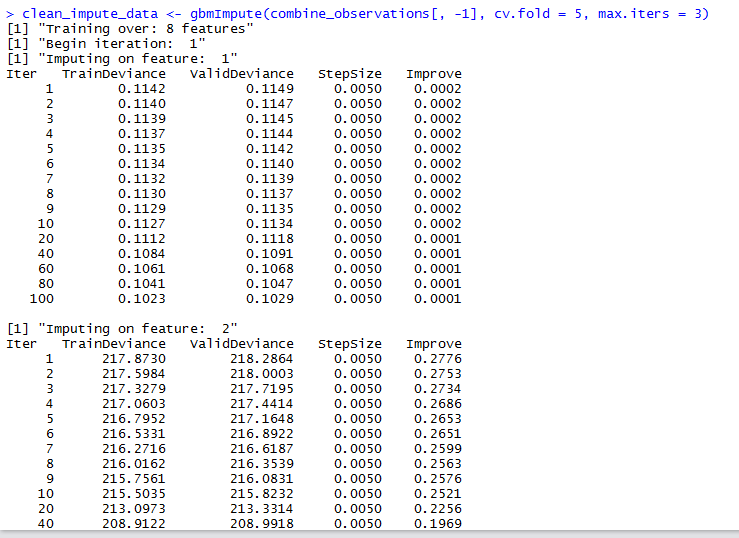
Other arguments used are as under:

**cv.fold**

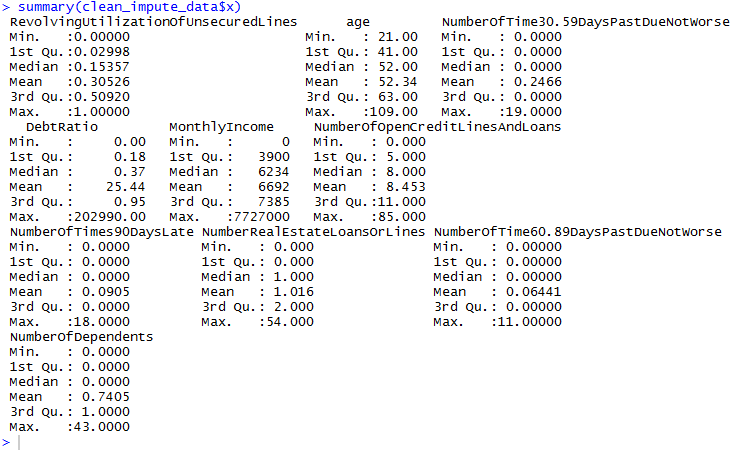
Number of folds that gbm should use internally for cross validation

**max.iters**

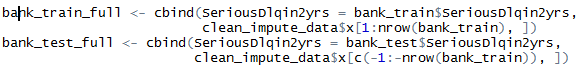
Number of times to iterate through the columns and impute each column with fitted values from a regression tree



Once we are done with imputation, we get cleaned imputed data



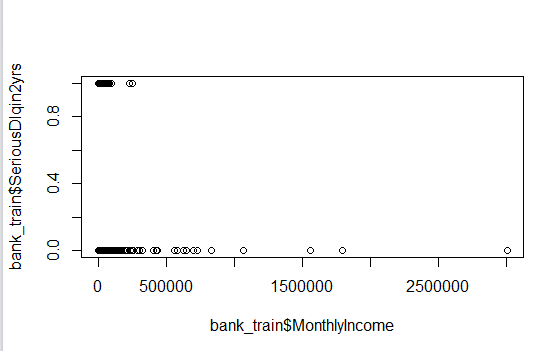
Dividing this imputed data again into training and testing set:



Now our data is cleaned, before working on prediction models we will be answering some of the business questions mentioned earlier using data visualization on the training data set.

* **Among the defaulters, what is the salary-range which attracts more defaulters than others?**

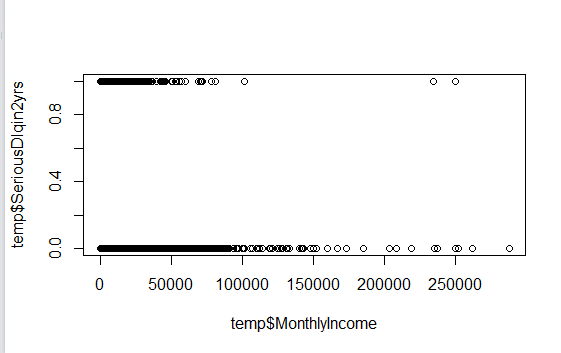




On a closer inspection, we find that most of defaulters have a salary range of

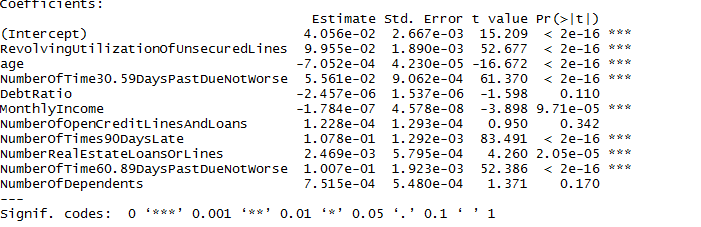
**$0 – $80,000**





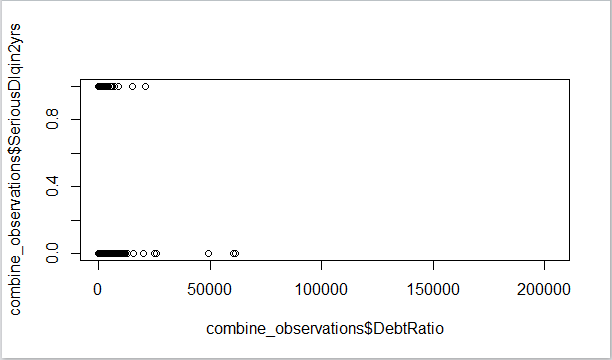
* **Which is major factor among this data that results in more defaulters than any other factor?**

Determination of factors which determine defaulters depends on the model we choose for our prediction. For example, when we use linear modelling for our prediction we can see the importance of independent variable in our model. ‘\*\*\*’ in front of an independent variable indicates that this independent variable is closely related to our dependent variable.



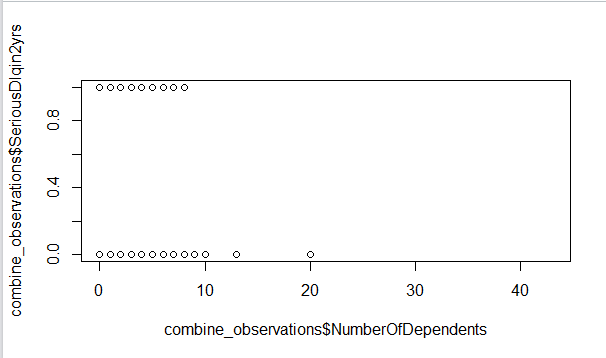
* **What is the relationship between defaulters and debtratio factor of each customer?**

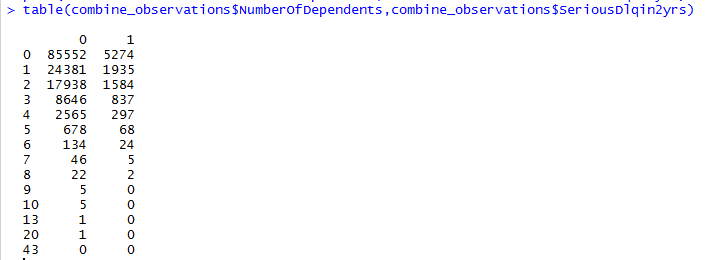




* **Does number of family members impact defaulter rate?**



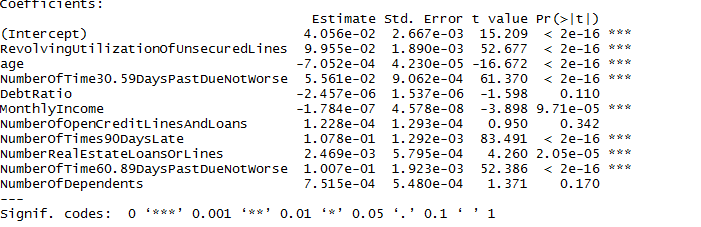




Looking at the plot and table function we were not able to determine any significant relationship between number of dependents and being a defaulter.

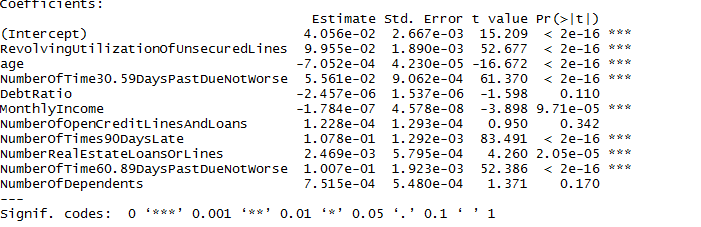
Therefore, in our models we may ignore this independent variable.

* **Out of three data variables of NumberOfTime30-59DaysPastDueNotWorse, NumberOfTime60-89DaysPastDueNotWorse and NumberOfTimes90DaysLate, which one will impact the most to defaulter list**?



As seen from the variable significance on summarizing a linear model, we found that **NumberOfTime30-59DaysPastDueNotWorse, NumberOfTime60-89DaysPastDueNotWorse** were pretty significant whereas **NumberOfTimes90DaysLate** was not significant according to our regression model.

* **Does monthly income play an important role to defaulter list?**

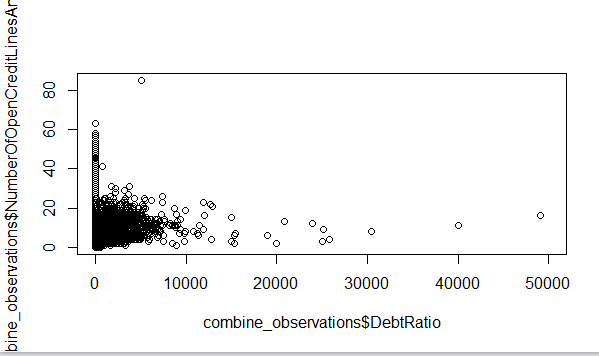


Clearly from the model, it is an important feature to train our model.

Apart from the regression model, we could also see in our previous visualization that salary range 0-80,000 are most of the defaulters.

* **How much does NumberOfOpenCreditLinesAndLoans variable impact debt ratio of person?**

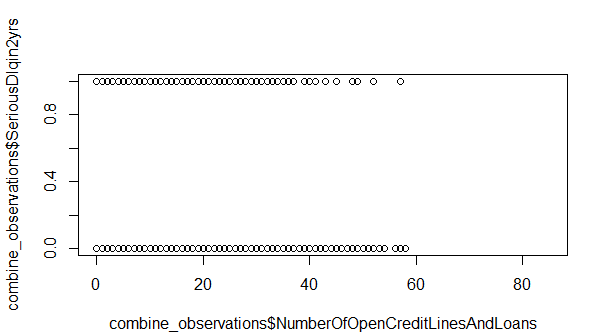




The General trend is that people have debt-ratio of 0 - 7,000 where average number of open credit lines tend to be in between 0-20.

* **Does NumberOfOpenCreditLinesAndLoans variable has a positive effect on defaulter list?**





As seen from the above visualization there is no significant relationship between defaulters and number of open credit lines.

These were some of the solutions to the business problems associated with our project.

Now, we will solving our major data question which is:

**Finding out whether or not a person will experience a financial distress in next two years?**

In order to solve this problem, we will be looking at many statistical and machine learning algorithms used directly in our problem.

Linear Regression

Let us first understand what is linear regression and weather this is applicable to our problem.

Linear regression analysis is the most widely used of all statistical techniques: it is the study of *linear*, *additive*relationships between variables.

 Let Y denote the “dependent” variable whose values you wish to predict, and let X1, …,Xk denote the “independent” variables from which you wish to predict it, with the value of variable Xi in period t (or in row t of the data set) denoted byXit.  Then the equation for computing the predicted value of Yt is:

http://people.duke.edu/~rnau/regintro_files/image002.png

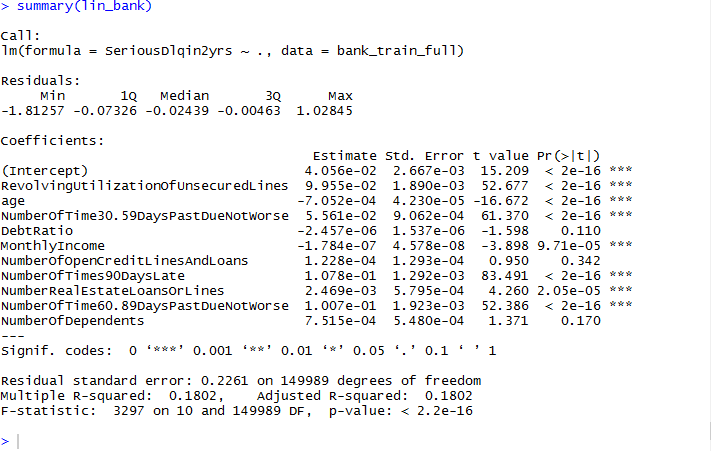
This formula has the property that the prediction for Y is a straight-line function of each of the X variables, holding the others fixed, and the contributions of different X variables to the predictions are additive.  The slopes of their individual straight-line relationships with Y are the constants **b1**, **b2**,…, **bk**, the so-called *coefficients* of the variables.

That is, **bi** is the change in the predicted value of Y per unit of change in Xi, other things being equal.  The additional constant **b0**, the so-called *intercept*, is the prediction that the model would make if all the X’s were zero (if that is possible).

The coefficients and intercept are estimated by *least squares*, i.e., setting them equal to the unique values that minimize the sum of squared errors within the sample of data to which the model is fitted. And the model's prediction errors are typically assumed to be *independently and identically normally distributed*

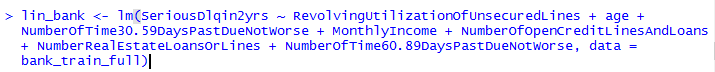
The flaw with linear regression is that we cannot assume our model to be simplistic and linear, and usually in the prediction problems mostly classification algorithms are used. Even with large R2 values, when we may assume that our model is predicting correctly it may not be predicting accurately it may bring overfitting and bias with it.

Despite of introduction to bias in our model, we tried running linear regression on our training data.

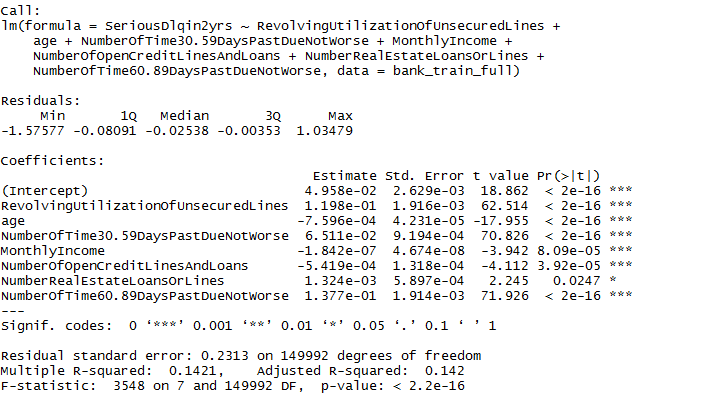


This gives us an R2 value of 0.2261 on the training data, which is very low and makes this statistical model insignificant.

But looking at the positive side, we were able to find some of the significant independent variables for future models.



To improve the accuracy of the linear model, we used the significant variables only and constructed a new model. We improved R2  value by 1%



Since, 0.2313 is a very small value for prediction model, therefore in order to improve our prediction model we will be working on some other models.

Now, we won’t be choosing only one particular model but will try to combine results from different models and calculate the probability of an event.

This process is known as “ensemblence” learning, which is very common for machine learning prediction models.

**Random Forest Algorithm**

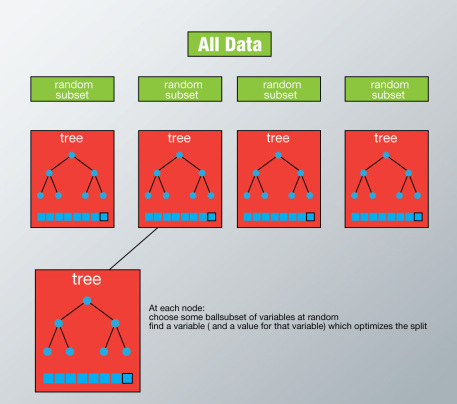
Random Forest is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration, and does a fairly good job. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

In Random Forest, we grow multiple trees as opposed to a single tree in CART model. To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

**How Random Forest algorithm works?**

In Random Forest, each tree is planted & grown as follows:

1. Assume number of cases in the training set is N. Then, sample of these N cases is taken at random but *with replacement*. This sample will be the training set for growing the tree.
2. If there are M input variables, a number m<M is specified such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while we grow the forest.
3. Each tree is grown to the largest extent possible till there is no pruning.
4. Predict new data by aggregating the predictions of the ntree trees (i.e., majority votes for classification, average for regression).



**Advantages of using Random-forests:**

* Random forests algorithm can solve both type of problems i.e. classification and regression and does a decent estimation at both fronts.
* One of benefits of Random forest is, the power of handle large data set with higher dimensionality. It can handle thousands of input variables and identify most significant variables so it is considered as one of the dimensionality reduction methods. Further, the model outputs **Importance of variable. We will be looking this by measuring performance of model as AUC score.**
* It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

**Disadvantages of using Random-forests:**

### It surely does a good job at classification but not as good as for regression problem as it does not give continuous output. In case of regression, it doesn’t predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy.

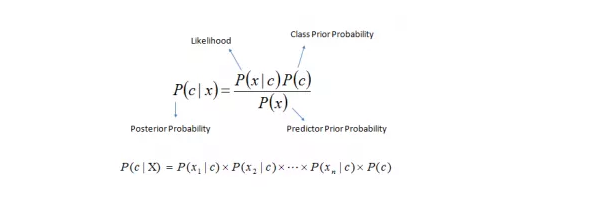
* Random Forest can feel like a black box approach for statistical modelers – you have very little control on what the model does. We can at best – try different parameters and random seeds!

**Naïve Bayes Algorithm**

It is a classification technique based on bayes- theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below



Above,

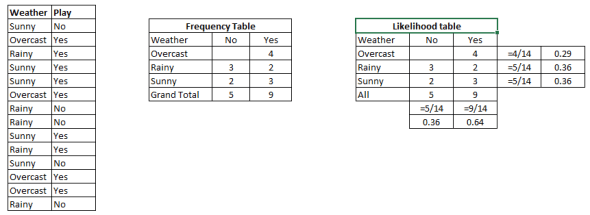
* *P* (*c|x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
* *P*(*c*) is the prior probability of *class*.
* *P* (*x|c*) is the likelihood which is the probability of *predictor* given *class*.
* *P*(*x*) is the prior probability of *predictor*.

## How Naive Bayes algorithm works?

Let’s understand it using an example. Below we have a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let’s follow the below steps to perform it.

**Step 1:** Convert the data set into a frequency table

**Step 2:** Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

[](http://i2.wp.com/www.analyticsvidhya.com/wp-content/uploads/2015/08/Bayes_41.png)

**Step 3:** Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

**Problem:**Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

**P (Yes | Sunny) = P (Sunny | Yes) \* P(Yes) / P (Sunny)**

**Here we have P (Sunny |Yes) = 3/9 = 0.33, P (Sunny) = 5/14 = 0.36, P( Yes)= 9/14 = 0.64**

**Now, P (Yes | Sunny) = 0.33 \* 0.64 / 0.36 = 0.60, which has higher probability**.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

## Applications of Naive Bayes Algorithms

* **Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
* **Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
* **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
* **Recommendation System:**Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

We will be using for naïve Bayes algorithm for making our recommendation system.

## Ensemble Learning

Ensemble is the art of combining diverse set of learners (individual models) together to improvise on the stability and predictive power of the model. In the above example, the way we combine all the predictions together will be termed as Ensemble Learning.

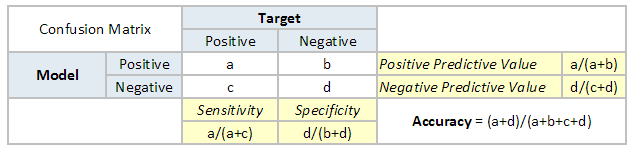
**AUC Scoring**

Now, while choosing parameters of Random forest or any other statistical model we will be determining our model by determining AUC scores.

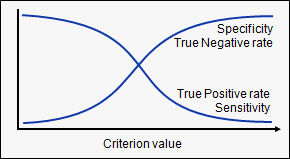
Let us understand what we mean by AUC Scoring, it is a popular metric used in industry to determine the performance of a model.

The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders. This statement will get clearer in the following sections.

Let’s first try to understand what ROC (Receiver operating characteristic) curve is. If we look at the confusion matrix below, we observe that for a probabilistic model, we get different value for each metric.



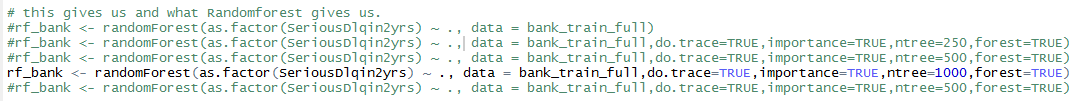
Hence, for each sensitivity, we get a different specificity. The two vary as follows:



Our aim is to make our model as accurate as possible. The higher the value is of ROC, the greater is the AUC (Area under the Curve).

In short, we must aim to achieve higher AUC and ROC scores.

**Developing our model:**



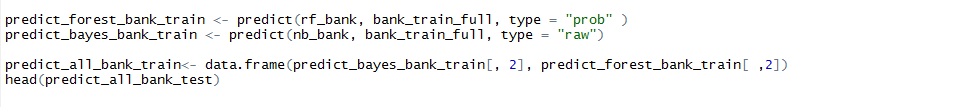
While developing our model, we had to use different combinations for random forests when combined with naïve Bayes theorem

Finally, we chose random forest algorithm and grew our number of trees up to 1000. The reason behind choosing was a better AUC score achieved by us.

Once random forest algorithm was chosen, we trained our model on the training dataset (bank\_data\_unskewed is basically training data, with sampling)



Once model has been trained, we have to test it. Firstly we are validating our data on the training data.   
Apart from validating the data, we will be using a genetic algorithm to determine the weight assigned to each statistical model used while ensemble learning.



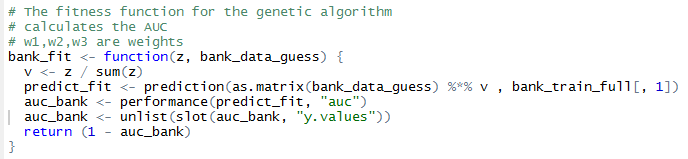
Here, we can see that we have stored our prediction results into predict\_all\_bank\_train dataframe.

We want the probability of an event happening, which is stored in second column of the dataframe(s).

We have prepared our model, now we have to assign weights to the predicted probability.In order to correctly assign weights to our model, we will be using genetic algorithm and fitness function which is used in genetic algorithms.

This Genetic algorithm is located in package “GA”





Here we have written a fitness function, which calculates AUC and helps in predicting the optimal weights.

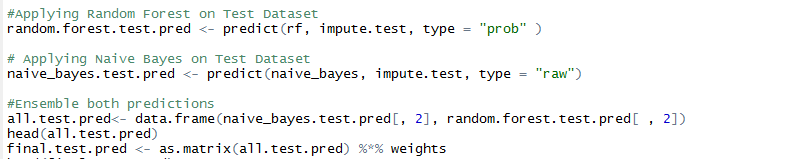


On Summarizing GA



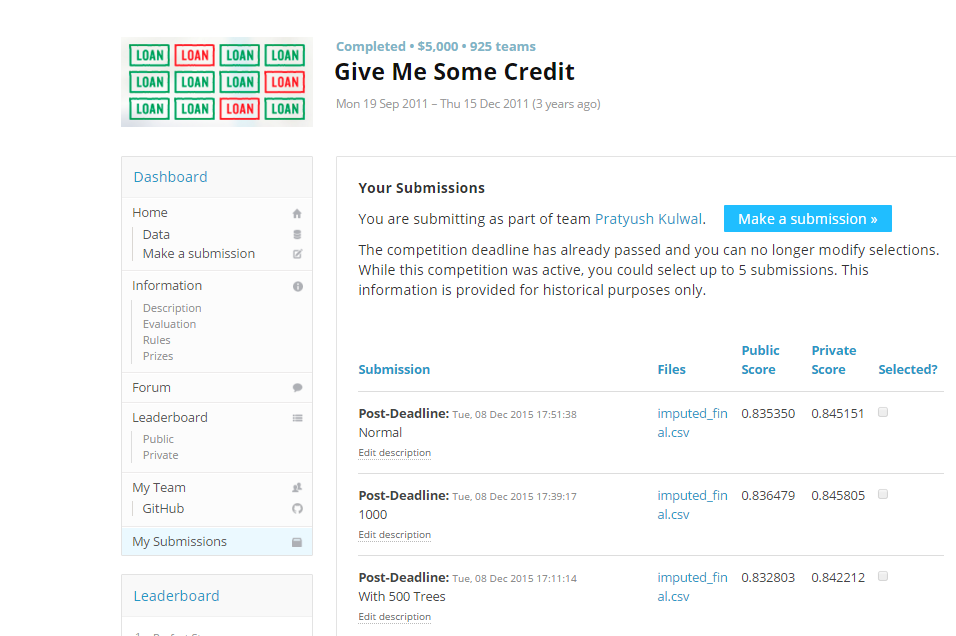
We get optimal weights, which we put in weights as a list for each model.

Finally applying prediction model on the testing dataset,

After that we will be writing our results in a file which we will be submitting on Kaggle in order to determine our test data predictions by AUC metric.



**Results:**



We got an accuracy of 84.5805 % , which is a good prediction accuracy score. The leaderboard i.e. the winner of this competion got a score of 86.9558 % .

**Scope of Improvement**

There is a scope of improvement for making a better model, based on further research for credit scoring algortihms we found that statistical models such as lasso and boosting could also be used. This might help our model to be more accurate and better.