# Loan Default Evaluation and Analysis

Isha Chawla

## Abstract:

Credit Scoring is the core of financial institutions and is a widely accepted standard used to determine a method for evaluation of the riskiness involved with giving out a loan to a customer. Credit Scores play a detrimental role when it comes to Loan Evaluation and deciding if a loan has a higher probability of defaulting or not.

In the past 5-6 years Lending industry has seen a 30% increase in loans that are charged off or default in each fiscal year, Lending Club is a P2P platform for borrowers and investors and they alone have seen a considerable increase in the charged off loans-which forms the premise of my research and the aim of this paper is to find an algorithm [1]that does a better job than the current algorithm.

## Introduction:

A detailed research and analysis of Credit Score and Loan Evaluation helped in identifying the factors that Lending Club or any bank while giving out loan use to make that decision, what I aim to do in this project is identify these factors in play and implement a classification algorithm that does a better job at classifying loans than the one being used by organizations now.

The main aim of this paper is to use the same attributes used by Lending Club or banks and devise an algorithm that can classify people in two categories Fully Paid or Default and the algorithm to have a higher accuracy in classifying Default Loans in the correct bucket.

## Background:

### How the credit score is calculated?

Traditionally, this is the most commonly adapted and used method for credit score evaluation.

Credit Score is divided into 5 categories[2] and each category has a weight associated with it:

1. **Payment History: 35%**

This is a log of all the credit you owed for maybe mortgage or student loans and how much have you repaid these loans. It also considers the number of accounts you have and if you have any if you have missed any payments on them.

2. **Used Credit vs Available Credit: 30%**

This is just a measure of the amount you owe on your revolving accounts. If, your credit limit is $5000 and you owe $1000 on it, the revolving utilization will be (1000/5000) = 0.2 and the revolving utilization will be 20%. The same approach applies if a user has multiple accounts.

3. **Type of credit used: 15% - 10%**

One of the key factors in determining credit scores is the different type of credits you have (installment, trade, gas station credit, student loans, etc.). This does not have a negative impact on the credit score but someone looking at improving their credit score can do it by having multiple types of credit accounts.
Though opening too many new accounts in a short duration of time can affect the score adversely.

4. **New credit – 10%**

The score is affected by the different types of accounts available. Every time a new account is opened an inquiry is made(hard-pull), which stay in your credit history for the period of 2 years, and they can adversely impact the score if the user has opened multiple accounts very close to each other.

5. **Length of oldest credit – 5%**

The oldest and the latest account opened, the oldest account is important to give creditors an idea that how well the user has handled credit card accounts over a course of time.

The traditional method for calculating credit score can be found in **Appendix A**.

Each category has a weight associated with it or the importance of each factor and these factors come together and form the FICO score. This score plays a very important role in the decision-making abilities, specially identifying loans and if a person is suitable to given out a loan out or not.

Most of the organizations use Logistics Regression, Stochastic Gradient Regressors or more intricate algorithms like Neural Nets to classify people and give them a credit score. What I will do here is take the factors that are mentioned in Appendix-1 and put people in one of the two categories: Fully paid or Default. Anyone in the Default category shouldn't be given a loan, one of the other important factors to consider here is that the cost of misclassification Default or Fully Paid Loans. If we misclassify Fully Paid loans the organization just ends up losing the interest that they will make, but if we misclassify Default Loans then the borrower loses the money they invested + the interest that they were supposed to earn.

---

[1] https://github.com/ishachawla/Big-Data-Systems-and-Intelligence-Analytics/tree/master/Lending%20Club

[2] Data collected from Transunion and Equifax

The goal of this project is to use various predictive algorithms to get to efficiently classify bad customers i.e. bad loans, from good customers and allowing platforms like Lending Club to better understand their overall risk exposure.

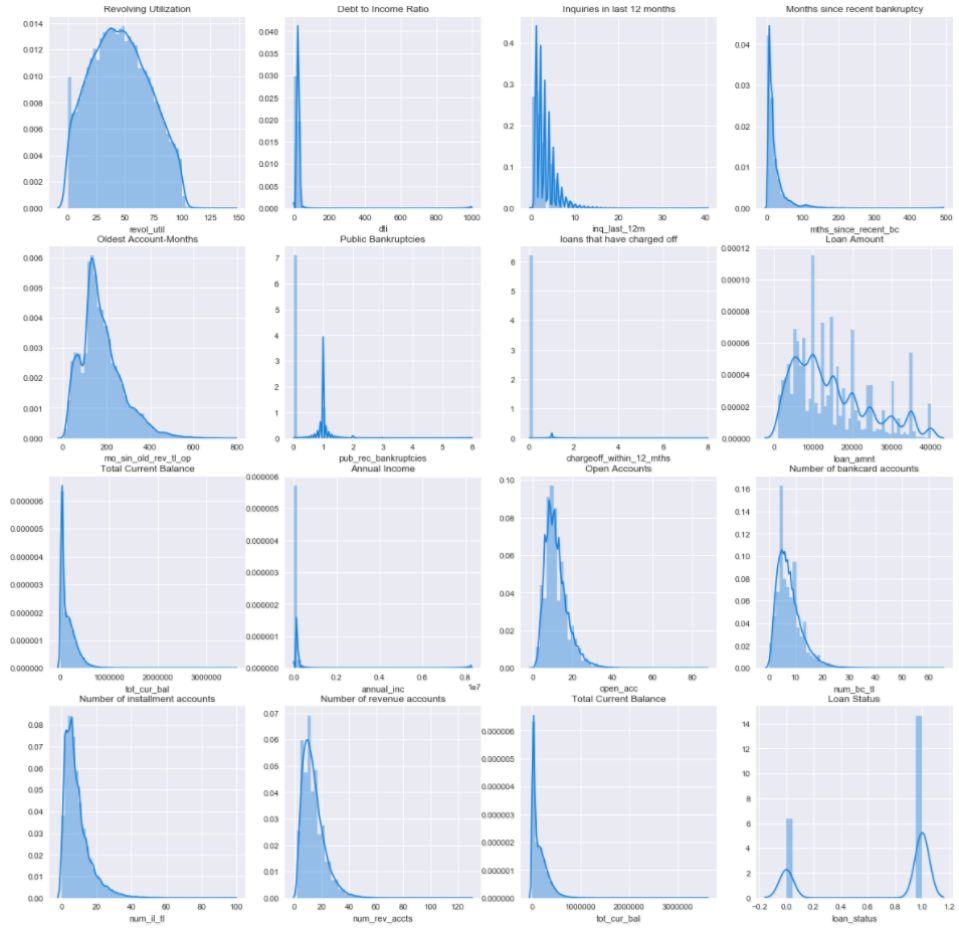
## Analysis:

### 1. Data Cleaning and Data Imputation

The data source for this evaluation is Lending Club. This data is available publicly through their website (https://www.lendingclub.com). One of the things we need to focus initially is that a lot of data has missing values and a lot of these columns are derived from other columns we need the values we need to make an informed decision (Appendix A).

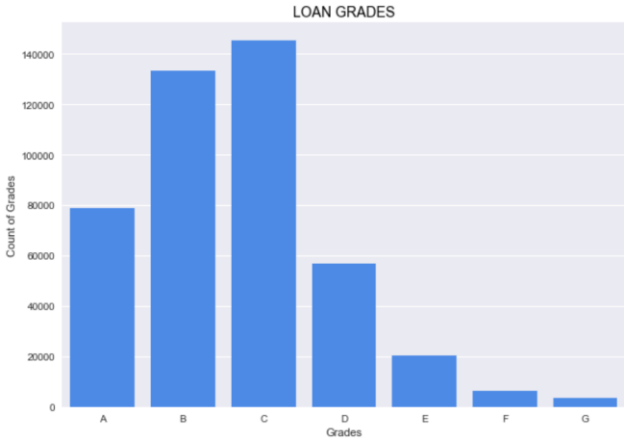
Almost 30% of the columns have more 50% of the data missing, hence we remove these columns completely from our dataset. We look at the distribution at some of the key variables to better understand how to complete the data. We can see that most of the variable are normally distributed around its median from the graphs below, hence we can impute the missing values with median values.

DTI or Debt to Income ratio is an important attribute in our existing dataset. DTI is a derived column; therefore, it should have a linear relationship with our other predictor variables. We build a linear regression model to complete the missing DTI values in our dataset.

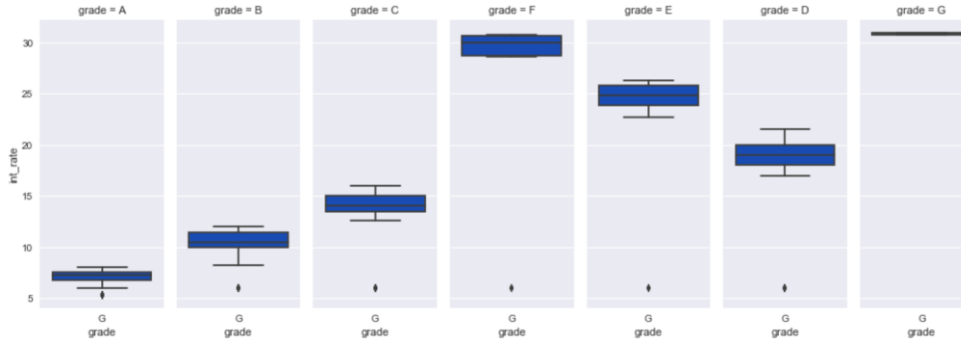


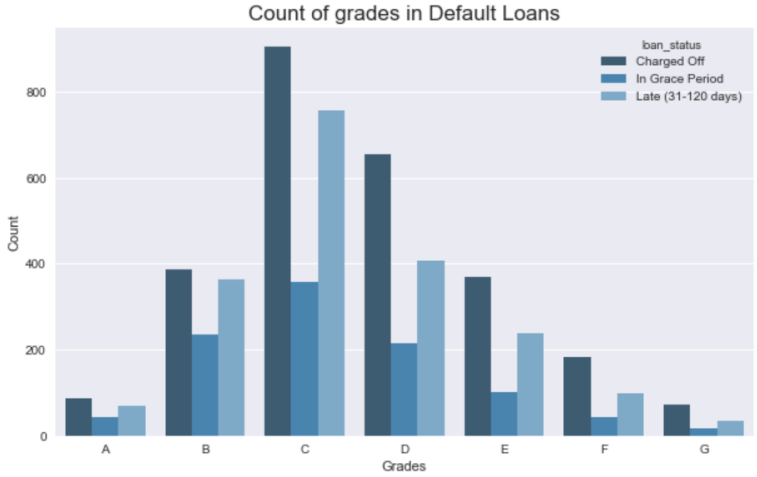

### 2. Exploratory Data Analysis

Lending Club classifies its loans into 7 categories from A-G, where A is the least riskiest in terms of repayment and G being the most risky. We can see that most of the loans have grades A-C, where most of the loans issued fall in category C.




Every grade has a minimum and a maximum interest rate associated with it, as we talked about before the less risky the loan, less the interest rate. This gives us an idea how the interest rate varies, across all the grades




We plot a graph to understand what grades has the most number of default loans are in category B and C but we see a considerable number of loans that were charged off defaulted in category A which has a considerably less interest rate, this is one more case where the current algorithm is doing a considerably bad job.

### 3. Data Transformation

We transform our existing dataset to best fit the problem we're trying to solve. The target variable in our dataset is Loan_Status which has 7 factors 1) Charged Off, 2) In_Grace Period, 3) Late (31-120 Days), 4) Current, 5) Default, 6) Fully Paid and 7) Late(16-30 Days). Based on our problem, all loans Late of in Grace Period should be considered as bad loans hence we reclassify these loans to Default status. This leaves us with only 3 factors 1) Default 2) Fully Paid and 3) Current.

We also normalize the complete dataset.

### 4. Algorithms/Techniques Used

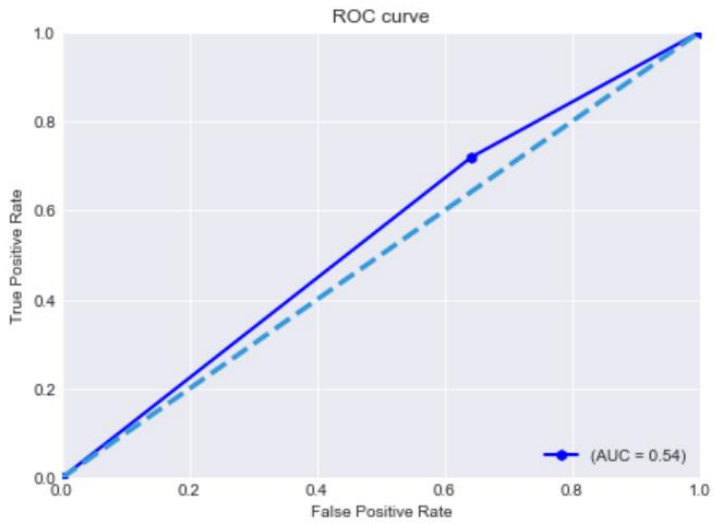
The main aim here is to have an algorithm that can efficiently determine by learning and looking at the various values and successfully segregate the loans that will be Fully Repaid to the one that won't. Our problem is primarily a supervised learning classification problem and we use multiple algorithms to get the best algorithm to that classifies the loans efficiently.

- Decision Trees: Decision Tree is one of the most commonly used classification techniques, which does a very good job at classifying people as people who will repay the loan or people who will default.

  Confusion Matrix of our model:

| | 0 | 1 |
|---|---|---|
| 0 | 1499 | 2653 |
| 1 | 2683 | 6786 |

  The area under the curve for the decision tree is 54% and we get an accuracy of 60%.




The model accuracy is considerably low, so we do parameter tuning and cross validation to improve the accuracy and reduce the misclassification error.

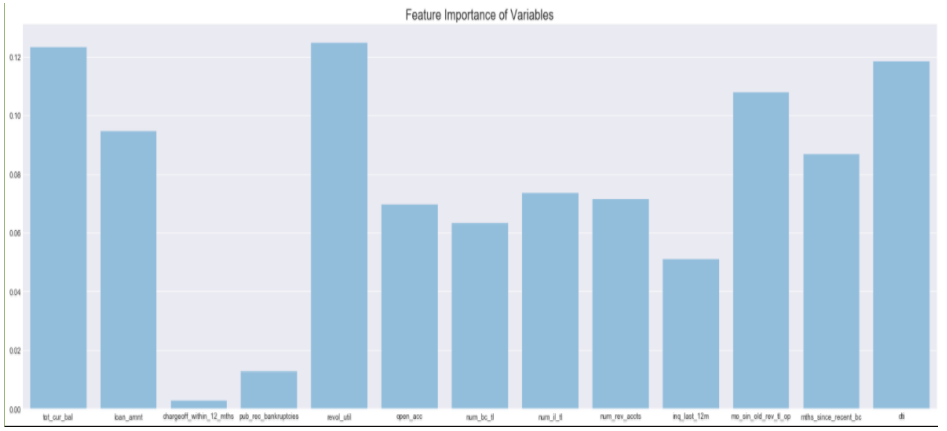
**GridSearchCV:**

Grid Search exhaustively searches a list of parameters grids and decides what parameters suits the models best and help it in achieving getting high accuracy. It also performs cross-validation.
It is one of the best methods for Hyperparameter tuning.

Some of the most common parameters to be tuned in Decision tree are:
max_depth: Maximum depth of the tree.
min_samples_split: Minimum number of nodes required to split an internal node.
min_samples_leaf: Minimum number of nodes required to be a leaf node.
max_leaf_nodes: Grow tree in best-first fashion.
criterion: gini or entropy for Information Gain
random_state: seed used by random number generator.

The GridSearch Decision Tree classifier gives us an accuracy of 69% but the area under the curve is 50%.

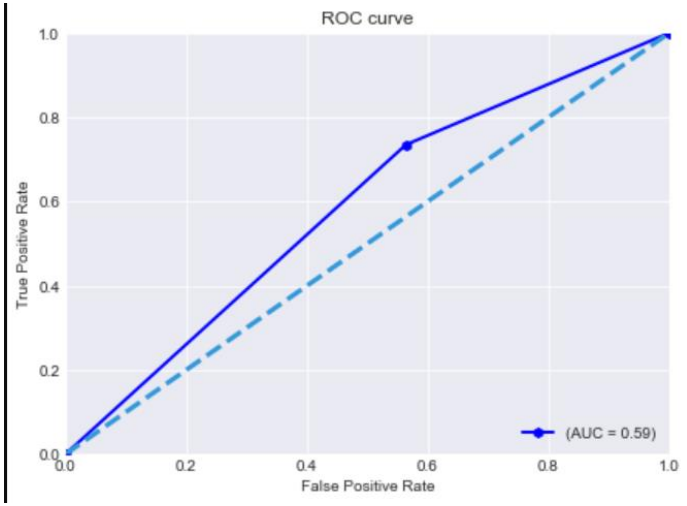
- **Random Forest Classifier:**
Random Forest is an ensemble method used for both Classification and Regression. The base model is created based on the top predictor variables and the following models are built on this base model.




We can get a feature importance in our dataset based on our model. Based on weights of predictor variables, we can see the most important variables for our prediction problem. Total Current Balance, Revol_util and DTI are the most important predictors as we can see from the graph above.

We optimize the parameters through a grid search and we achieve an accuracy of 70.07%.

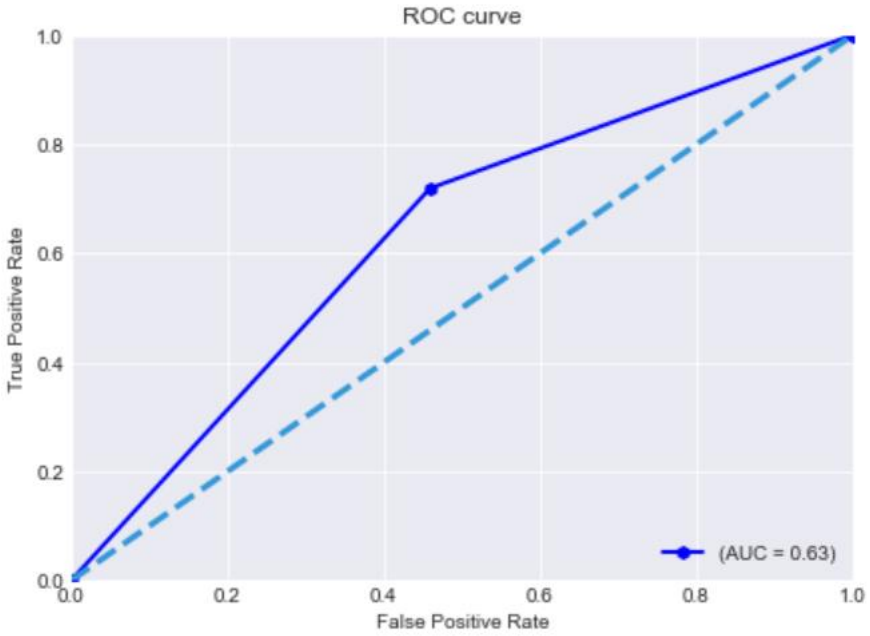
| | 0 | 1 |
|---|---|---|
| 0 | 190 | 114 |
| 1 | 3962 | 9355 |



- **XGBoost**

XGBoost (Extreme Gradient Boosting) is a scalable implementation of gradient boosting trees which help to improve the computing powers of decision trees. So far, we have seen an improving accuracy in our decision tree models. We try to implement XGBoost which should improve performance and speed of our overall predictive model.

Our model has an accuracy of 70.26% and AUC of 0.63 from the ROC curve.



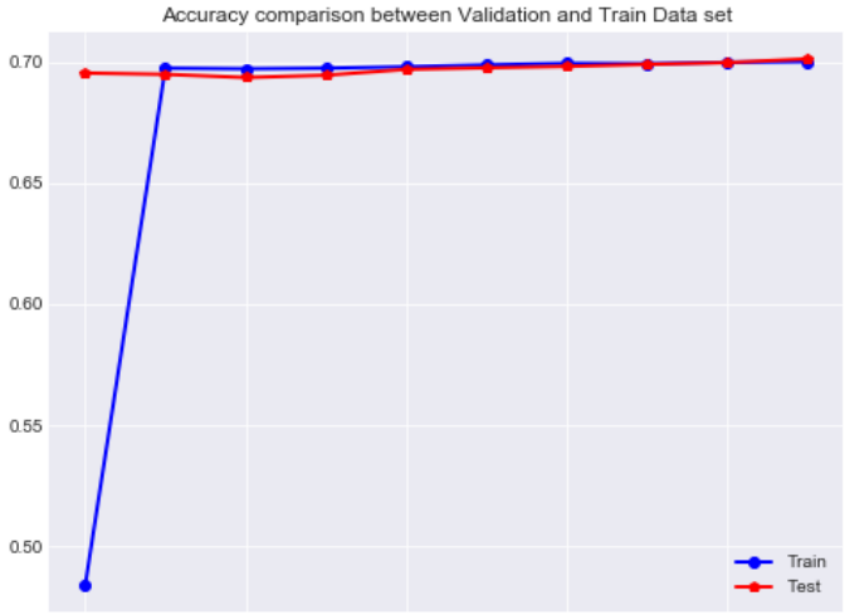

| | 0 | 1 |
|---|---|---|
| **0** | 675 | 3477 |
| **1** | 573 | 8896 |

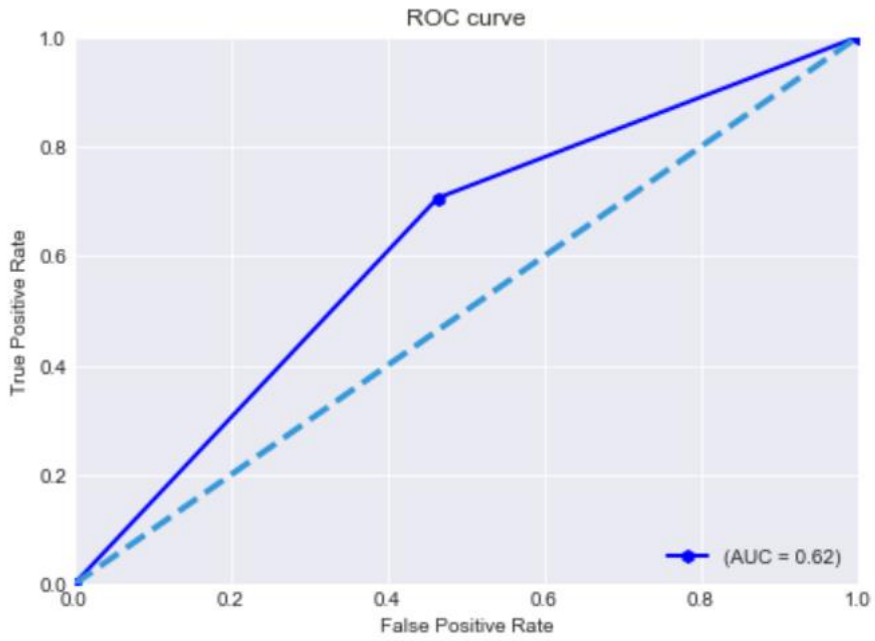
- **Neural Net – Multi Layer Perceptron**

I create a Feed Forward Neural Network using Keras for Binary Classification. The model of the accuracy is 69% and the area under the curve is 62%.

| | 0 | 1 |
|---|---|---|
| **0** | 310 | 268 |
| **1** | 3842 | 9201 |

Keras allows you to save accuracy and accuracy on the validation set on every iteration.






**Model with Hyperparameter Tuning:**

Activation Function: tanh,sigmoid, reLu

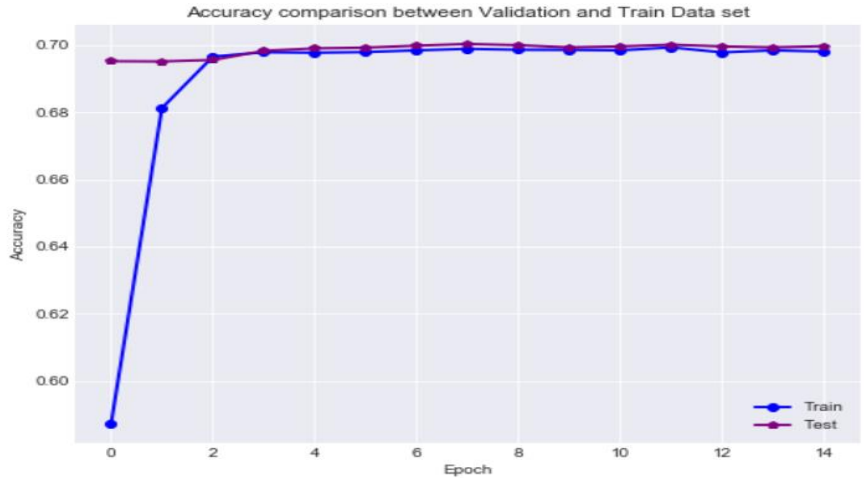
Kernel Initialization: random normal, None

Adding a regularization layer- Dropout

Optimizer- adam, adamax

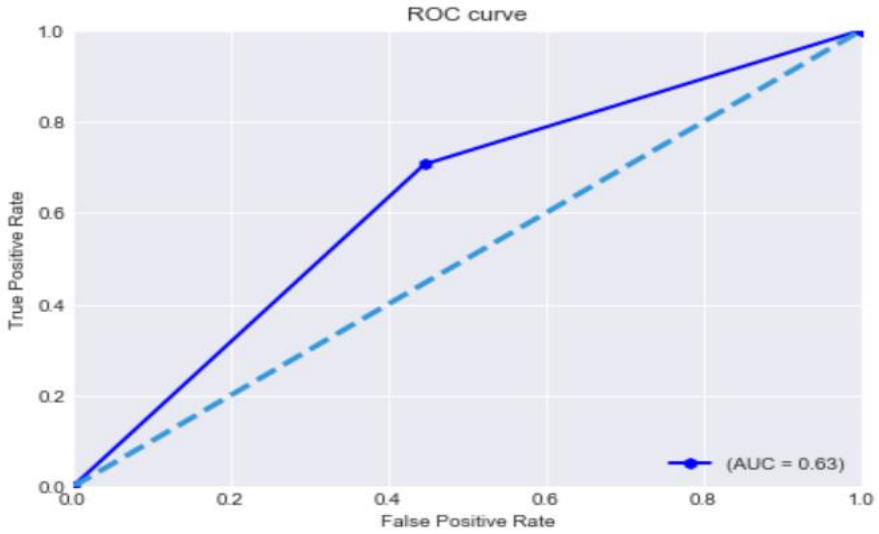
Epochs- number of iterations

Batch-size- number of rows being trained at one time




The accuracy of the model is 69.97% and the AUC is 63%

| | 0 | 1 |
|---|---|---|
| **0** | 312 | 251 |
| **1** | 3840 | 9218 |




**Results:**

| Model | Accuracy | Missclassified Loans Percentage |
|---|---|---|
| Decision Tree | 60% | 19% |
| Random Forest | 66% | 20.85% |
| Random Forest RandomizedSearchCV | 70% | 29% |
| Random Forest GridSearchCV | 70.14 % | 28.58% |
| XgBoost | 69.79% | 24.19% |
| Neural Network with 1 layer | 69% | 28% |
| Neural Network with multiple layers | 70.04% | 28.47% |

The two important factors in deciding if an algorithm does a good job is checking the accuracy and also making sure it has the lowest misclassification rate of default loans put in the Fully Paid category.

If we look at the results, we can see that Random Forest Grid Search CV has the highest accuracy but we cannot use it for multiple reasons:

1. High misclassification error.
2. High processing time.

Our next best option is XGboost which though has a lower accuracy compared to some other models but it has a considerably low misclassification error too, so this algorithm can be a good model for the organization but we should not forget the fact that a low accuracy signifies that it misclassified some Fully Paid Loans which can become a problem for the organization, because it can get very rigid and may lead to Lending Club or any other organization losing money on the interest they make on the loans that are completely paid.

So, the safest bet considering all options is Neural Network with multiple layers which has an accuracy of 70.04% and misclassifies default loans around 28.47% time which is better than the job, the current algorithm is doing by misclassifying 30% of the loans that will default.

## Conclusion:

In conclusion, I would like to say that the detailed analysis of using different classification algorithms we can say the Multi Layer Perceptron does a good job at classifying people in one of the two buckets and if Lending Club was to use this algorithm they would see a considerable decrease in the number of loans that default.

## Appendix A:

| **Category 1** | Payment History | Description |
|---|---|---|
| tot_cur_bal | 137118 | Total Balance Owed on the accounts |
| chargeoff_within_12_mths | 0 | Loans that have been charged off(defaulted) |
| pub_rec_bankruptcies | 0 | Public Bankruptcies in the last 7 years |
| | | |
| **Category 2** | Revolving Credit | |
| revol_util | 62% | Total money owed on all accounts/Total credit limit on all accounts. |
| | | |
| **Category 3** | Type of credit | |
| open_acc | 10 | Total number of active accounts |
| num_bc_tl | 10 | Total number of bankcard accounts |
| num_il_tl | 2 | Total number of installment accounts |
| num_rev_accts | 12 | Total number of revolving accounts |
| | | |
| **Category 4** | New Credit | |
| inq_last_12m | 4 | Inquiries in the last 12 months |
| | | |
| **Category 5** | Length of oldest credit | |
| mo_sin_old_rev_tl_op | 327 | Oldest Account - Months |
| mths_since_recent_bc | 6 | Most Recent Account opened- Months |

**References:**


https://keras.io/getting-started/sequential-model-guide/

http://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting

https://www.lendingclub.com/info/demand-and-credit-profile.action

http://cs229.stanford.edu/proj2011/JunjieLiang-PredictingBorrowersChanceOfDefaultingOnCreditLoans.pdf