Anu Rekha Thiruvenkadam

anurekha105@gmail.com

Abstract

This report is an attempt to analyze and predict the risk of a Lending Club loan being default based on past loan data and borrower credit history.

Predicting loan defaults of lending club loans

# Synopsis

LendingClub is an online marketplace for Peer-to-Peer lending and has issued over $20 billion in loans since it was first launched in 2007. Investors in LendingClub can choose to invest in Notes (fractions of individual borrower loans) of a diversified portfolio of loans. LendingClub assigns a loan grade between A and G based on borrower credit quality and underlying risk. Investors choose their portfolio based on the loan grades. One would assume that the higher loan grades would have practically nil or very low default rates. But data shows otherwise – Even Grade A, B loans have a default rate of 7-15%. With a well-diversified portfolio, defaults may not do much harm, but all the same we would like to minimize the defaults as much as possible. So, it would be of significant use for LendingClub and its investors to have a predictive model that will predict whether a given loan will default or not and use this prediction in the choice of loans to be funded.

We have used the loan data provided by LendingClub from 2012 to 2015 for building and testing a predictive model. Among the multiple machine learning algorithms that we tried, including a Voting Ensemble, we found that a model based on Extreme Gradient Boosting gave better results compared to others.

The implementation has been done using Python, pandas for data wrangling and scikit-learn for machine learning.

# Dataset & Labelling loans

The LendingClub has made its datasets of complete loan data open to the public at <https://www.lendingclub.com/info/download-data.action> . Each dataset contains the latest status for the loans originated in the stated period. For this study, three datasets containing loan data issued from 2012 through 2015 have been used – viz, Loans 2012-2013, Loans 2014 & Loans 2015.

There are 7 possible loan statuses – *Fully Paid, Current, In Grace Period, Late (16-30 days), Late (31-120 days), Default, Charged Off*

For this analysis & classification, we consider the statuses *Late (31-120 days), Default, Charged Off* as **Bad** loans & the status *Fully Paid* as **Good** loans. All other loan statuses (*Current, In Grace Period, Late (16-30 days)*) are ignored since these loans could end up either way and we do not want to use them with partial information in our supervised learning algorithms.

The 9-month loan status migration rate in the below chart as provided by LendingClub compels us to consider the *Late (31-120 days), Default* statuses as **Bad** loans, since at least 77- 90% of these loans end up being *Charged Off* in 9 months

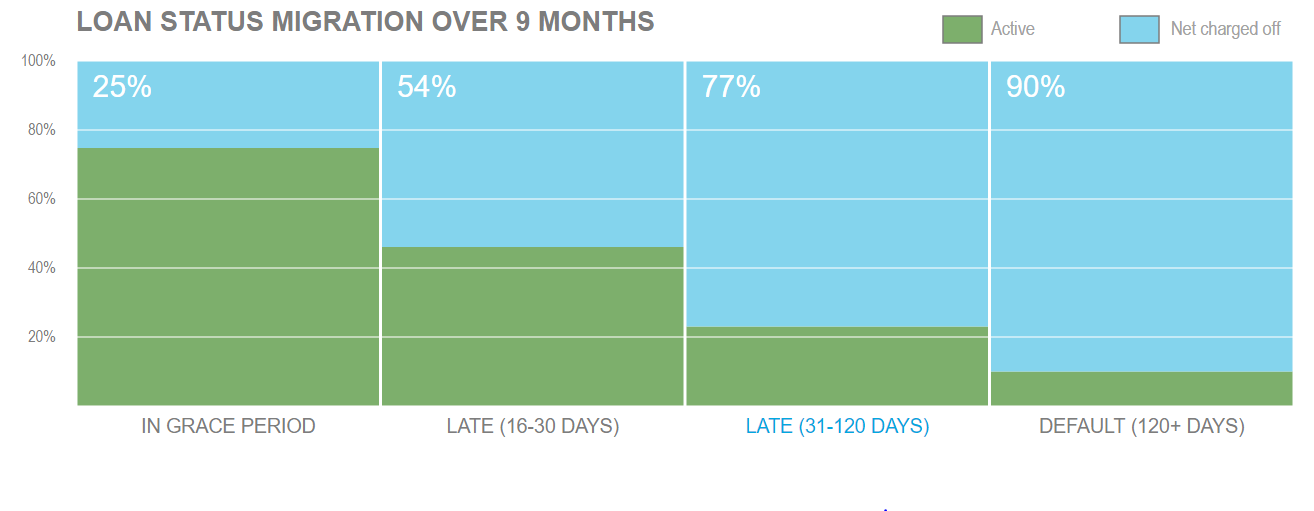
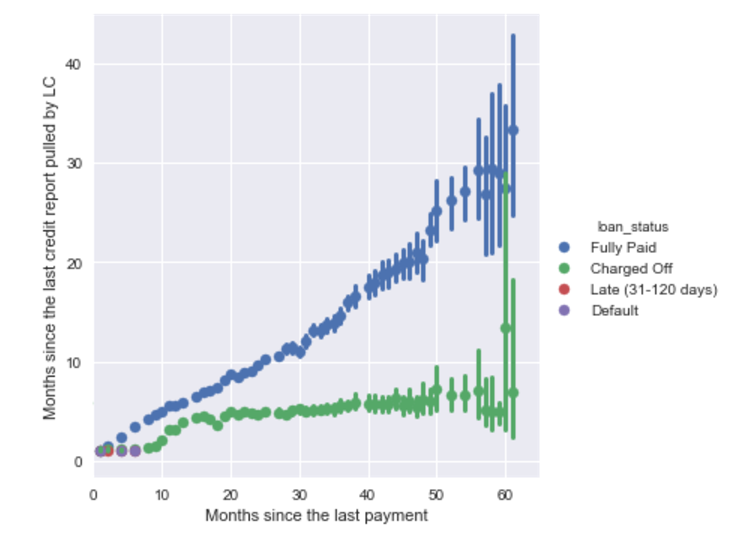


Figure 1 [https://www.lendingclub.com/info/demand-and-credit-profile.action]

# Analysis

## Data Cleanup & Transformations

1. Loans with statuses *Late (31-120 days), Default, Charged Off* , *Fully Paid* as mentioned above have been retained, removing all other statuses.
2. The older datasets may not have data for few features that are available in the later dataset of 2015. For eg, FICO score high and low range, total balance across all trades etc. Only the features that are available across all the datasets have been retained. The data dictionary for the remaining features is available at [this link](https://github.com/anurekhat/Capstone/blob/master/LendingClub/LCDataDictionary2014.xlsx).
3. The datasets across the different time-periods are merged into a single pandas dataframe.
4. A very less number of loans, a few hundred, were of ‘JOINT’ application type. These loans have been removed from analysis as a bulk of credit history features may not be fully applicable for these kinds of loans.
5. The dataset includes some features related to the loan repayment details, viz, *issue\_d , next\_pymnt\_d, last\_pymnt\_d , collection\_recovery\_fee , last\_pymnt\_amnt , out\_prncp , out\_prncp\_inv , recoveries , total\_pymnt , total\_pymnt\_inv , total\_rec\_int , total\_rec\_late\_fee , total\_rec\_prncp* . These features are clear giveaways of whether the loan is going good or bad and obviously will not be known at the time of approval or funding of the loan. So these features have been removed from the predictor variables list. The data dictionary in the link provided above has a column indicating whether a feature is a predictor variable or not.
6. One of the features that needed to be removed was the date of last Credit History pull by LC. For loans that were *Charged Off* or *Fully Paid* around the same time, one wouldn’t expect the time since last Credit pull to be different. However, the below chart shows that LC continues to pull Credit History for *Charged Off* loans but not for *Fully Paid* loans. So this feature was removed from the predictor list as this is an ‘after-the-fact’ feature that will not be known at the time of approval or funding of the loan.



1. Some features that are manual text entries / informational have been removed, viz *emp\_title, desc, title, addr\_state, zipcode*
2. The date attribute *earliest\_cr\_line* has been converted to the appropriate duration feature
3. Imputing missing values – Some features have a definitive meaning when the value is not present. For eg, *mths\_since\_last\_delinq* , months since last delinquent payment by the borrower will be null when the borrower has never been delinquent before. So the missing values for these features (*mths\_since\_last\_delinq, mths\_since\_last\_record* , *mths\_since\_last\_major\_derog* , *mo\_sin\_old\_il\_acct* , *mths\_since\_recent\_bc\_dlq* , *mths\_since\_recent\_inq*  , *mths\_since\_recent\_revol\_delinq* ) have been imputed to -1 to set these apart.
4. Some loan records do not have data for some of the critical credit history features (*tot\_hi\_cred\_lim, tot\_cur\_bal, num\_tl\_120dpd\_2m, revol\_util, bc\_open\_to\_buy, bc\_util, mths\_since\_recent\_bc, pct\_tl\_nvr\_dlq, num\_rev\_accts* ). Since this kind of information is highly subjective, it maybe misrepresentation of facts if we impute this missing data with the median or mean value. So it was decided to drop records with values missing for any of these features.
5. One Hot Encoding for categorical features – The categorical features (*home\_ownership* , *verification\_status* , *pymnt\_plan, purpose, initial\_list\_status, term*)were one-hot-encoded.
6. Some of the features like( *tot\_cur\_bal* , revol\_bal, *avg\_cur\_bal, bc\_open\_to\_buy, delinq\_amnt, total\_bal\_ex\_mort, total\_bc\_limit, total\_il\_high\_credit\_limit*) carry absolute dollar value. However these values make a true meaning only in comparison to the borrower’s annual income. So these features were converted as a fraction of annual income.
7. The loan grade and sub-grade categorical features were converted to numerical values considering that the grade and sub-grade are ordered. So Grade A gets a higher value compared to Grade B and sub-grade A1 gets a higher value compared to sub-grade A2.
8. The loans were labelled as Good (0) or Bad (1) as mentioned in the above section and the label was marked as the target variable.

## Exploratory Analysis

Note that all analysis and predictions are based on *Fully Paid, Charged Off, Default & Late (31-120 days)* loans and we have excluded all current loans and loans that are up to 30 days late

### 3.2.1 Loans by Loan Status

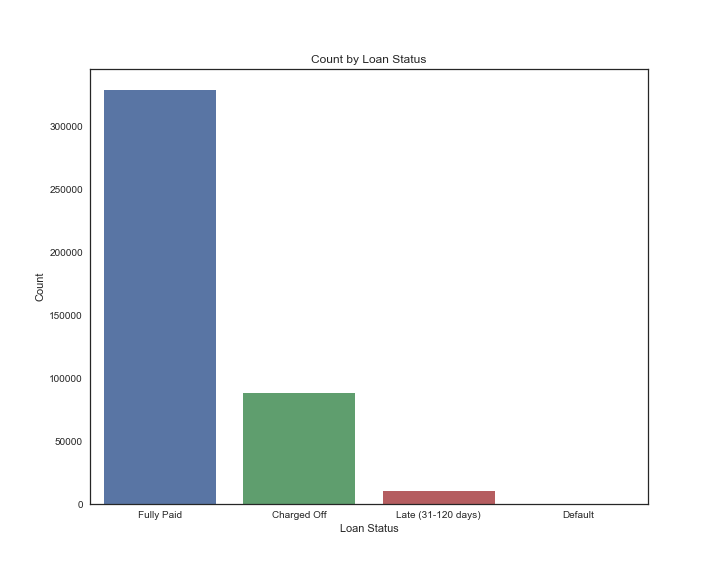


Figure 2

### 3.2.2 Loans by Loan Label (Good / Bad)

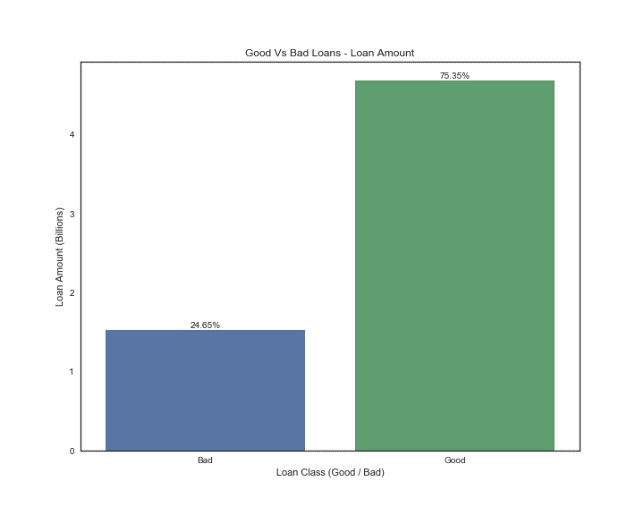
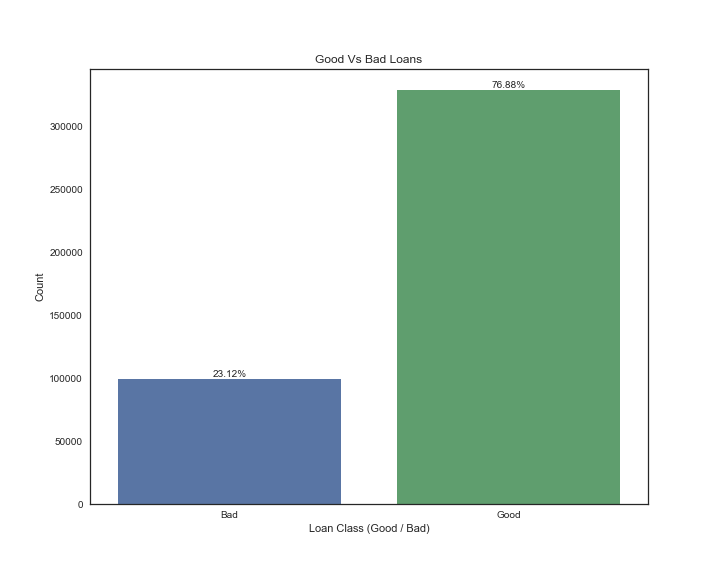
We can see that the data set is slightly imbalanced. 23.12% of loans are labelled as ‘Bad’

Figure 3

### 3.2.3 How does the Default rate vary across the loan grades

LC assigns loan grades based on the quality of borrower credit history. So higher grade loans are expected to have lower default rates. Let’s see the trend below.

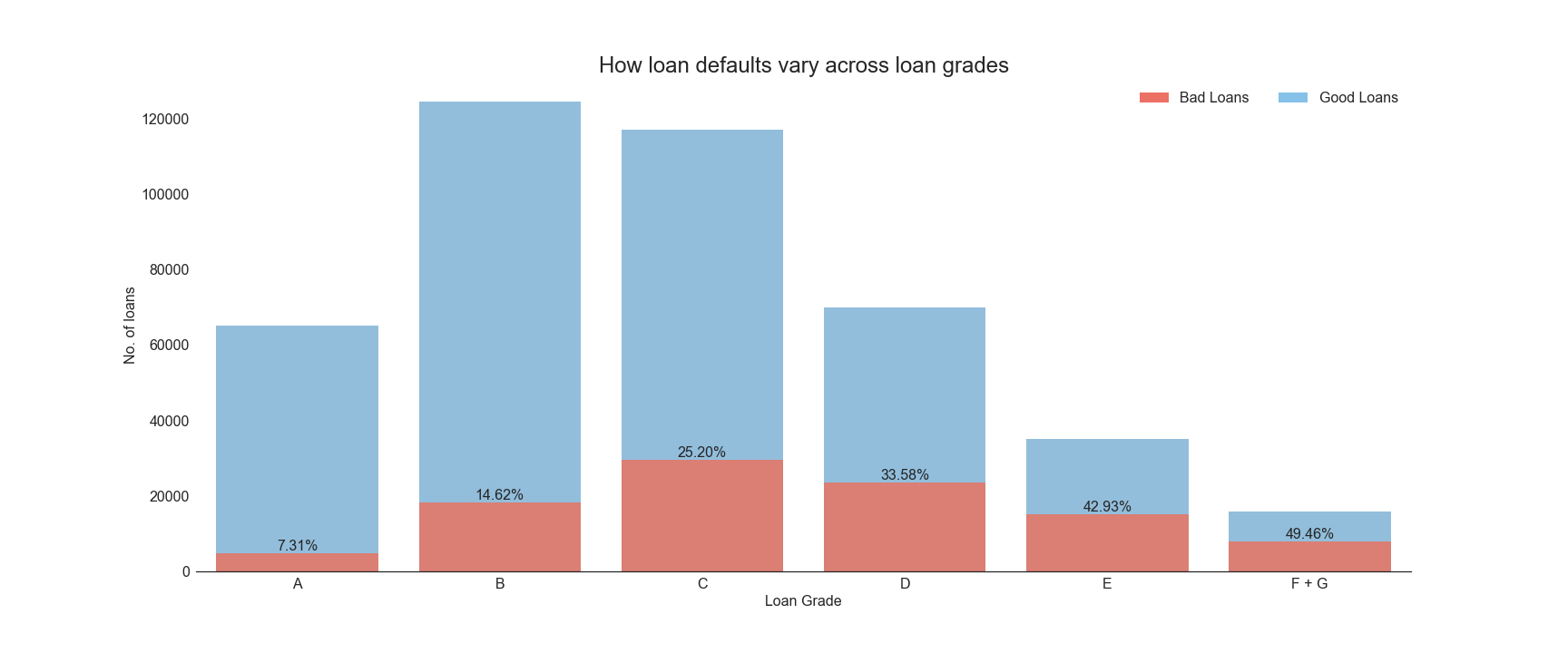


Figure 4

This trend of loan default rate can complement the insight provided by LC on the expected annualized return rate across grades. Please refer Figure 5 below, originally provided by LendingClub.

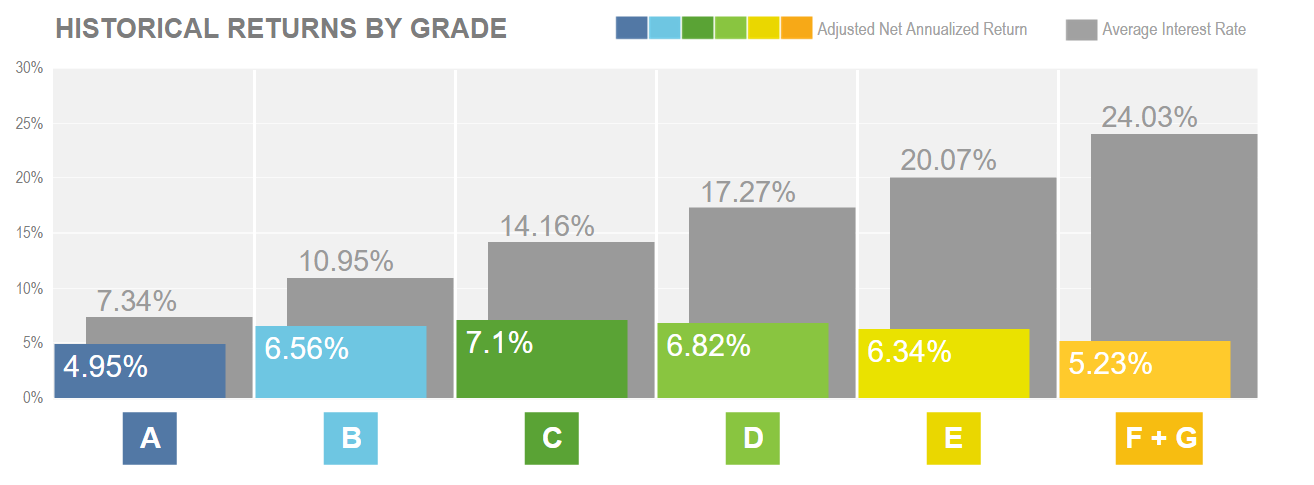


Figure 5 (Ref https://www.lendingclub.com/info/demand-and-credit-profile.action)

For eg: we can infer that although Grade E loans have an interest rate of 20.07%, the annualized return rate is only 6.34%, one of the main reason being that 42.93% of Grade E loans are at risk of default.

Conversely, Grade A loans charge an interest rate of 7.34% and is able to provide annualized return of 4.95%, as the risk of default is less at 7.31%.

### 3.2.4 How do the predictor variables vary for the Good / Bad loans

The below chart captures the density plot for all the predictor variables plotted separately for the Good & Bad loans. Here we see that there is considerable overlap in the value ranges for the predictor variables between the Good and Bad loans. There are some predictors that show a slight difference in the value trends, viz *dti* (debt to income ratio), *int\_rate, annual\_inc* etc.

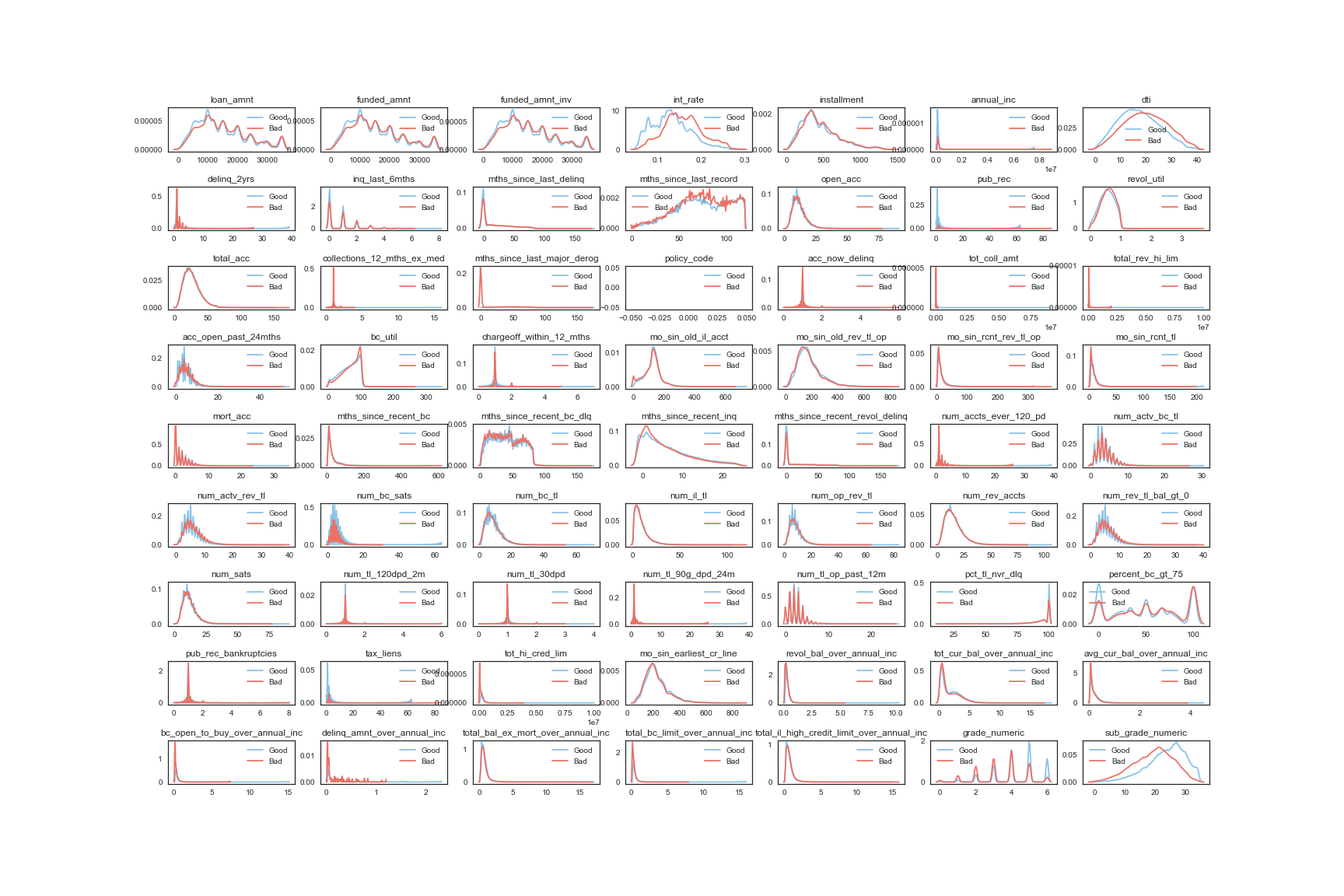


Figure 6

# Predictive Model

## Algorithms & Performance Metrics

The loan defaults or in general financial credit risk prediction is a tough task since there is no easily visible patterns. The dataset is imbalanced as one would expect, the default or ‘Bad’ class being at 23% of the completed loans (excluding ‘Current’ loans as mentioned previously). The dataset having 427K total records, out of which around 99K are labelled ‘Bad’ is split into train-test datasets at 80-20 ratio.

The idea is to train multiple models using different algorithms and build an ensemble of these models for the final prediction. The below models have been used to build the individual models.

1. Random Forest Model
2. Extreme Gradient Boost Model
3. Light Gradient Boost Model
4. Voting Ensemble

Cross Validation has been used for hyper parameter tuning for the individual models. A smaller dataset of 60K records has been used for the train- test splits for cross validation.

A probability threshold of 0.4 was used instead of the default 0.5 when labelling the predictions so as to pick up border cases of possible defaults as learned by the algorithms.

Since the dataset is imbalanced, we cannot rely on accuracy alone for evaluating the model performance. We concentrate on the ROC & Precision-Recall values and particularly the AUC score. A loan being default or ‘Bad’ is the positive case here and we are interested in the True positives and False positives predicted by the model.

We will also do a comparison of these metrics for all the models.

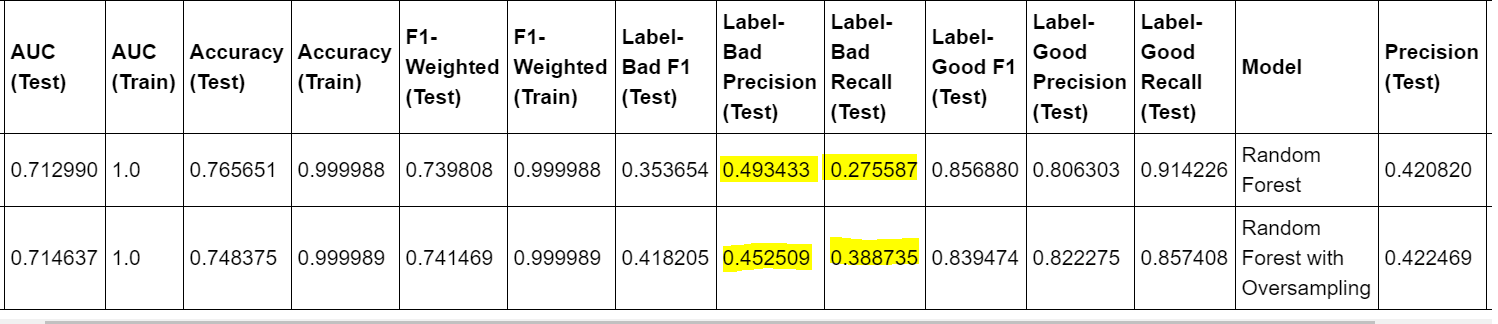
We have also trained a variation of each model using a SMOTE oversampled training dataset.

## Random Forest Model

Based on Cross validation, it was concluded that training more than 200 trees doesn’t bump up the results too much. Similarly, increasing the max\_features parameter to be more than the default value of square root of total features also doesn’t help much. A random Forest model with 200 trees, the sklearn default values for max\_depth and max\_features was trained on the training dataset. The parameter class\_weight was set to ‘balanced’ so that the class imbalance is addressed.

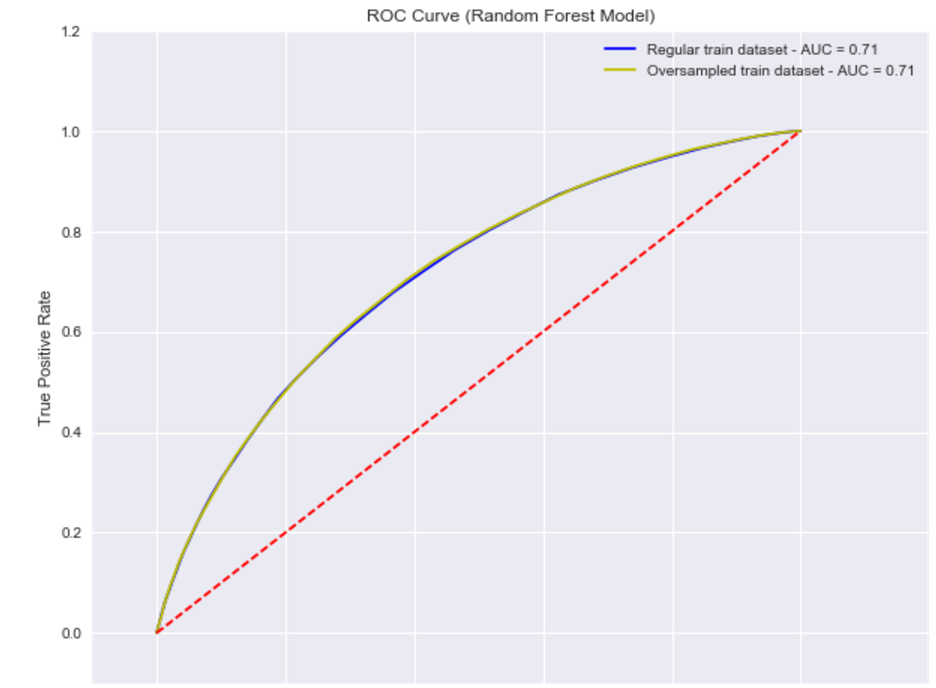
Below is a summary of the results & a plot of the ROC curve.

### Performance Metrics



Here we can see that using oversampled learning dataset has helped improve the recall rate for the ‘Bad’ class from 27% to 39%, though affecting the precision to go down to 45% from 49%. This is when a threshold of 0.4 is used to label the ‘Bad’ loans.

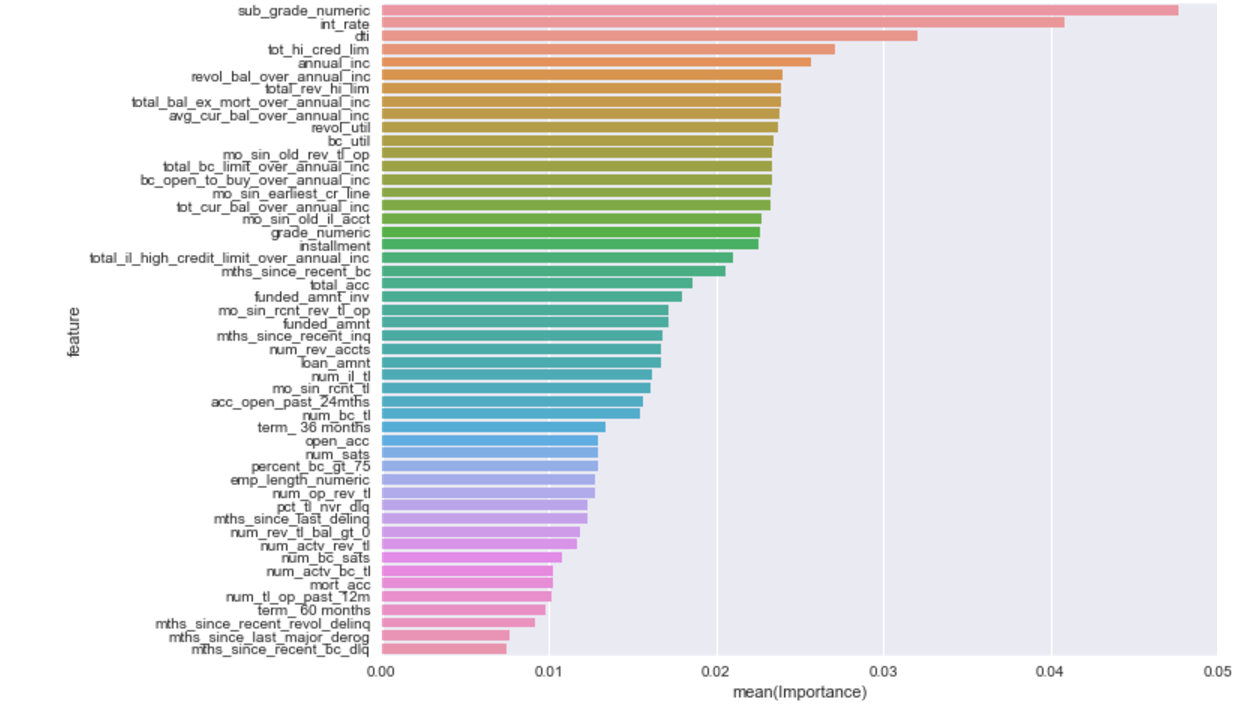
### ROC Curve



Oversampling doesn’t make much difference to the model in terms of the Sensitivity & Specificity

### Feature Importances

The below chart shows the top 20 most important features as per the Random Forest model.



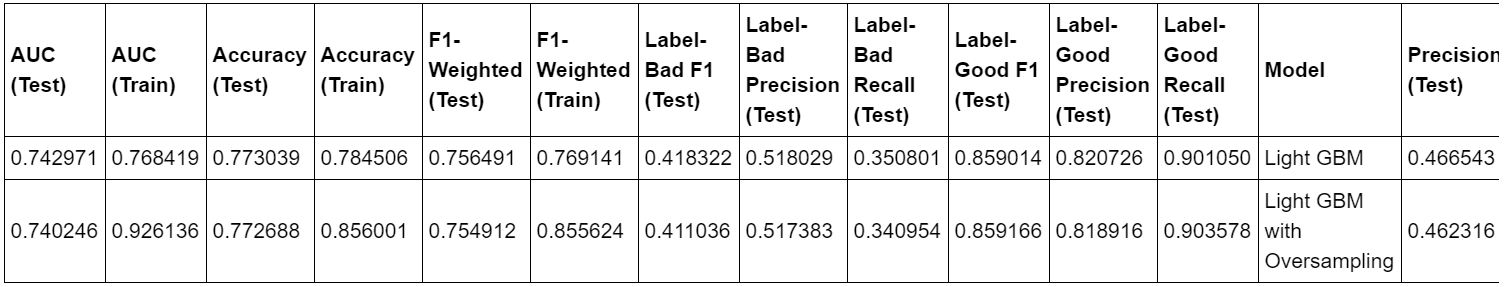
## Light Gradient Boost Model

The Light GBM model is a high performance variant of Gradient Boost model which splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth-wise or level-wise. Here the parameter num\_leaves determines the extent of growth of the trees. Max\_depth parameter serves as a means to limit the tree from growing beyond the defined level. Based on cross validation results, a Light GBM binary tree with 300 leaves and a low learning\_rate of 0.05 was trained. The evaluation metric for the model was set to ‘AUC’ and ‘F1\_Weighted’ scores so that the class imbalance is addressed.

We noticed that the Light GBM model indeed performs faster in comparison to XG Boost and gives comparable scores as well.

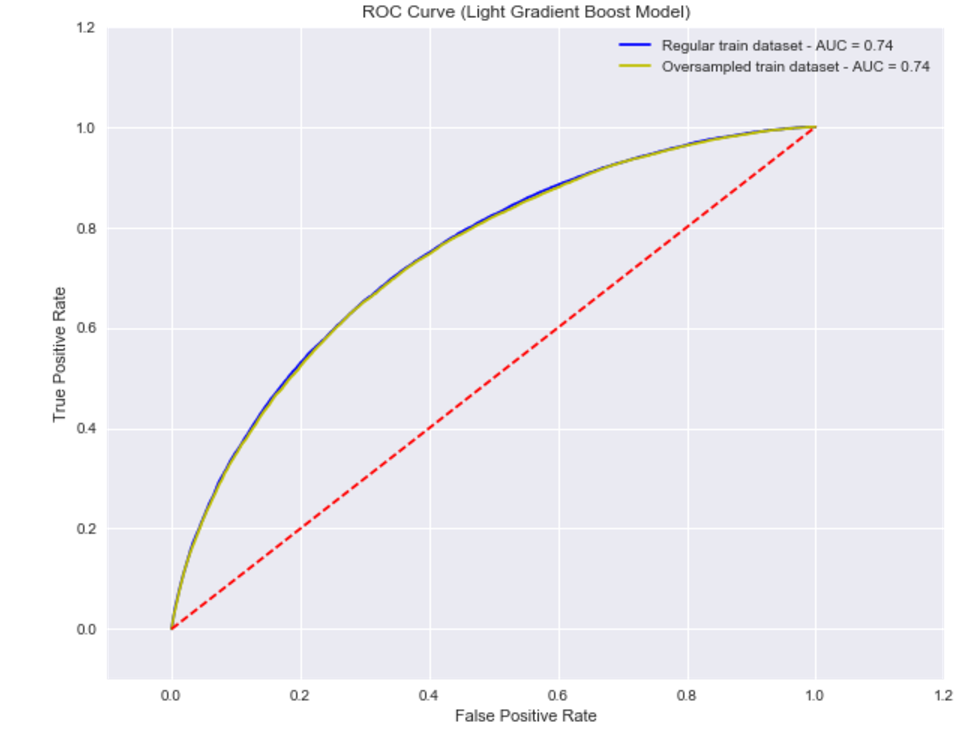
Below is a summary of the results & a plot of the ROC curve for the models trained using the regular dataset and the oversampled data set.

### Performance Metrics



Here we see that the performance metrics on test set remains the same between the models. The oversampled model has trained better, but doesn’t really translate to results on the test set.

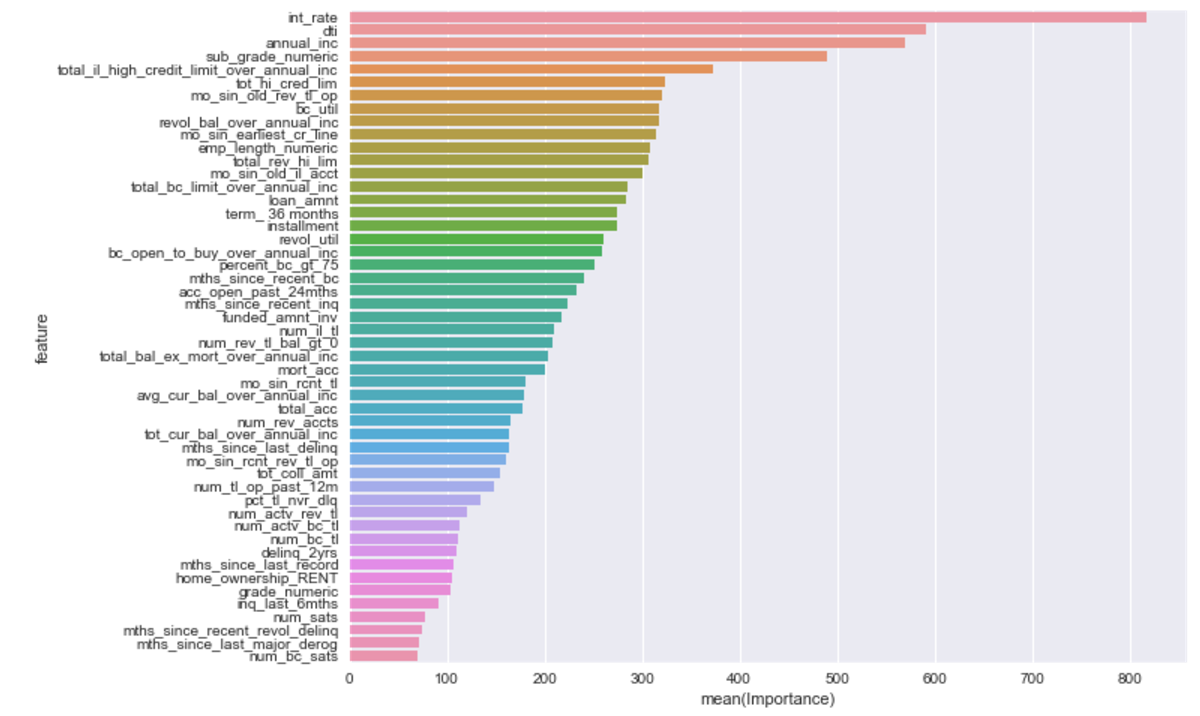
### ROC Curve



Oversampling doesn’t make much difference to the model in terms of the Sensitivity & Specificity

### Feature Importances

The below chart shows the top 20 most important features as per the Random Forest model.

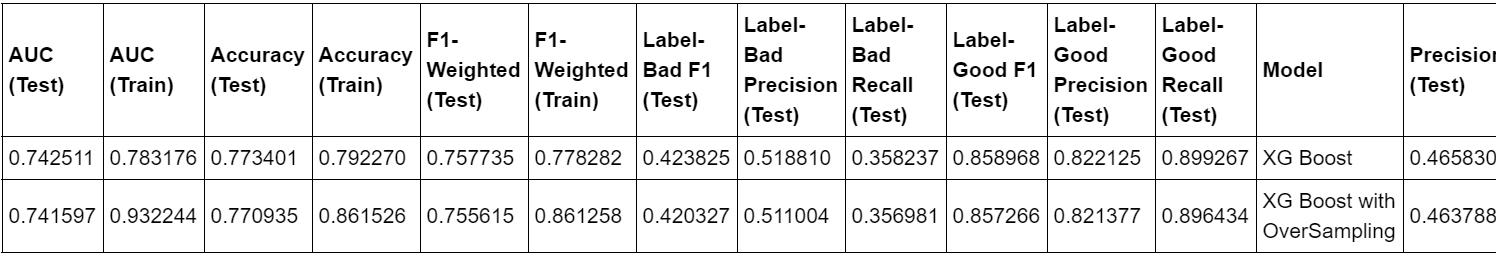


## Extreme Gradient Boost

The XGBM model was implemented with the ‘gbtree’ booster and tuned parameters from cross validation. A learning\_rate parameter value of 0.15 and 300 trees gave better results with the full dataset. The model was trained using AUC as the evaluation metric.

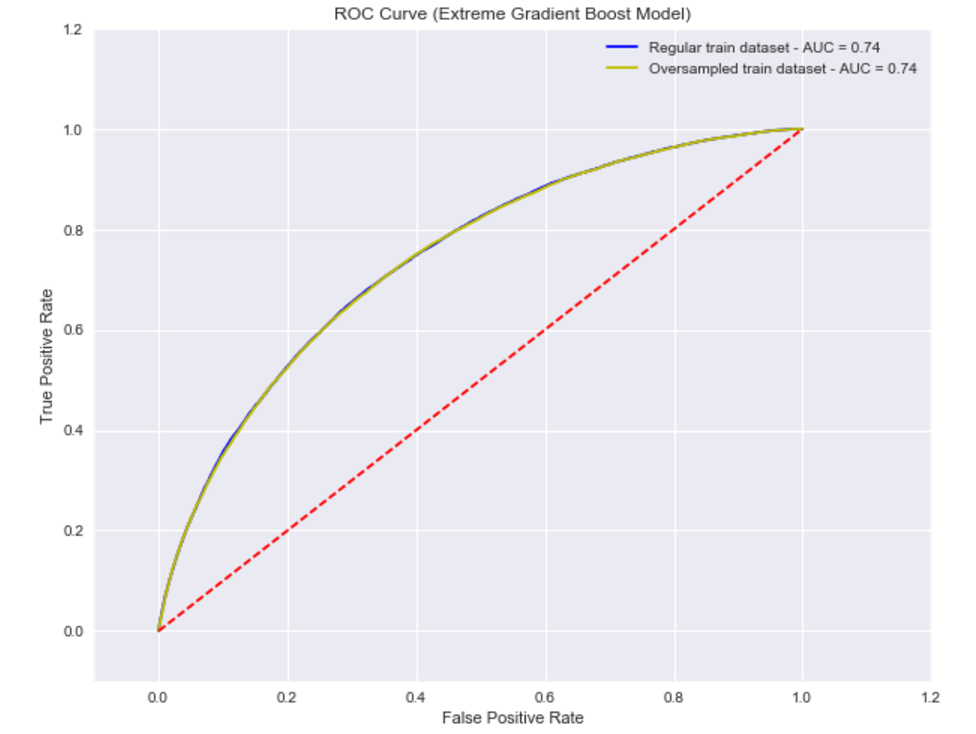
Below is a summary of the results & a plot of the ROC curve for the models trained using the regular dataset and the oversampled data set.

### Performance Metrics



Here we see that the performance metrics on test set remains the same between the models. The oversampled model has trained better, but doesn’t really translate to results on the test set.

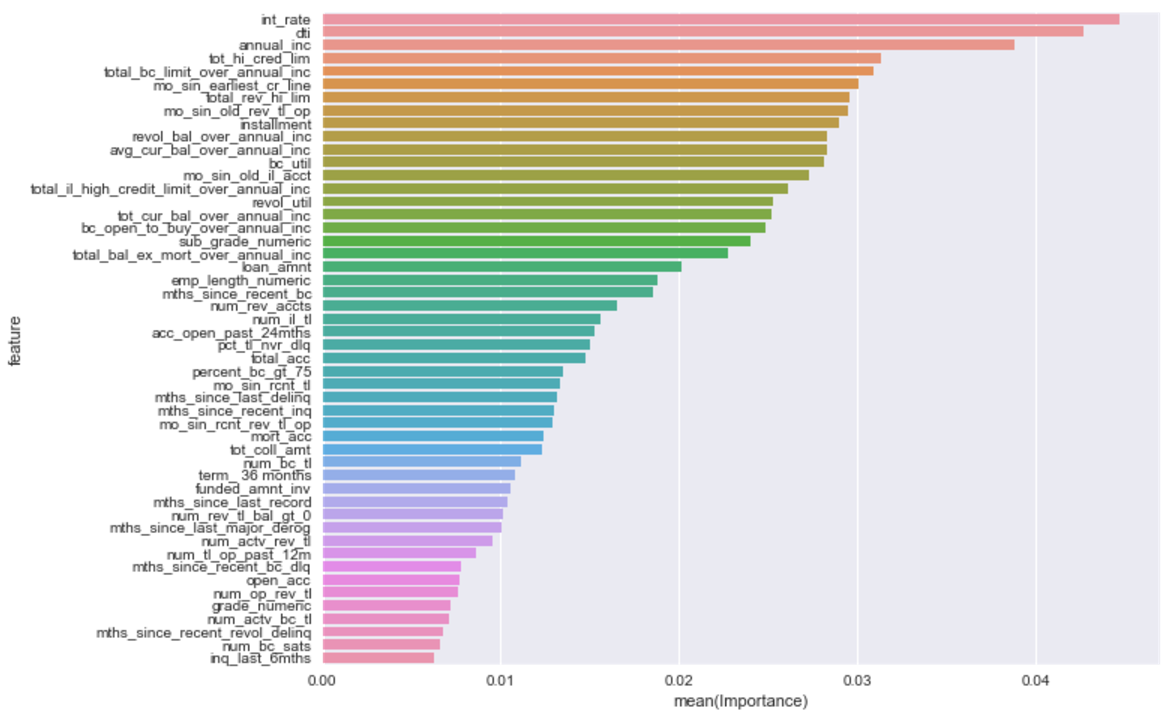
### ROC Curve



Oversampling doesn’t make much difference to the model in terms of the Sensitivity & Specificity

### Feature Importances

The below chart shows the top 20 most important features as per the XG Boost Model.



## Voting Ensemble

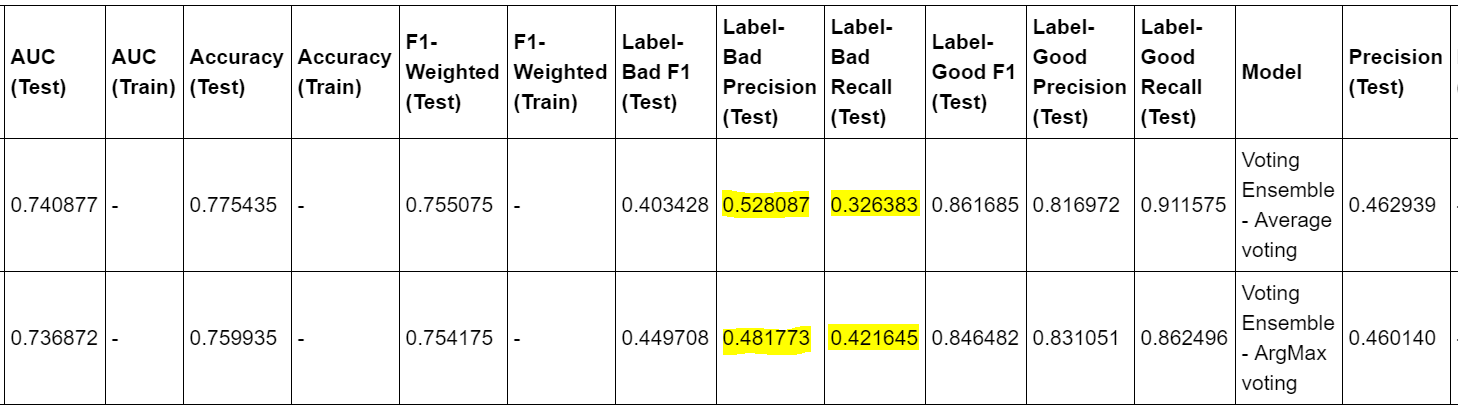
A Voting Ensemble was implemented using the above three models and the performance of the results from this ensemble model was recorded. Two voting models were implemented with two types of voting –

* 1. a ‘soft’ voting which will return an average of the predicted probabilities
  2. a ‘soft argmax’ voting which will return the maximum value of predicted probabilities for the positive (Bad) class from among the three classifiers.

Note: sklearn’s VotingClassifier ensemble could not be used here as the Light GBM model does not follow the sklearn Classifier methods and a custom implementation was needed. Also, that allowed for the implementation of the ‘Argmax’ voting which is not supported in the sklearn VotingClassifier

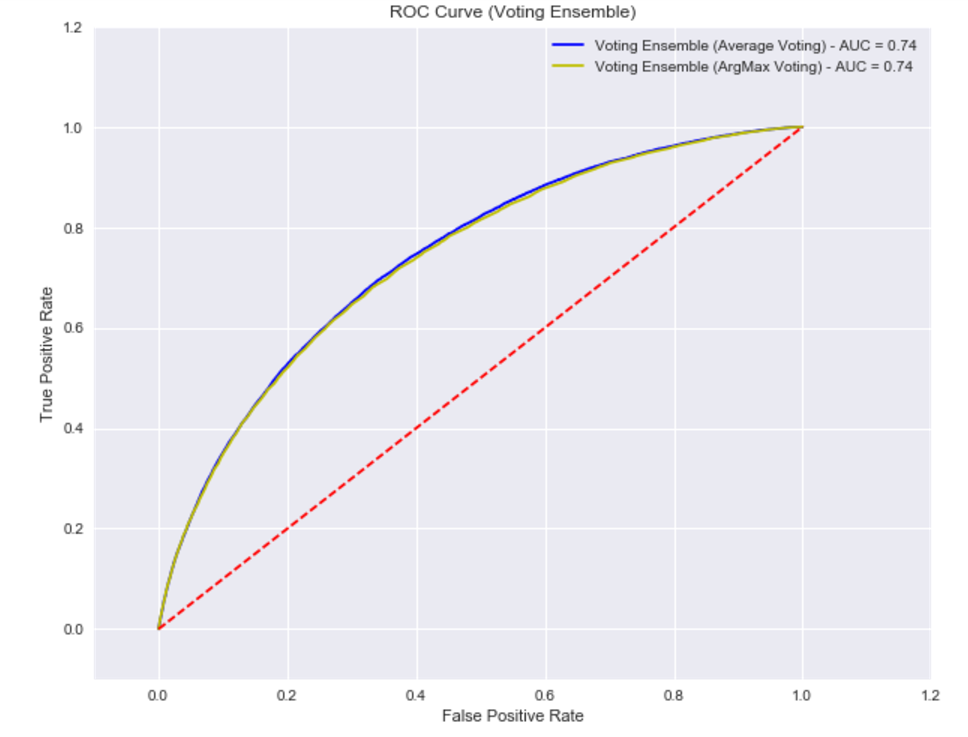
Below is a summary of the results & a plot of the ROC curve for the models trained using the regular dataset and the oversampled data set.

### Performance Metrics



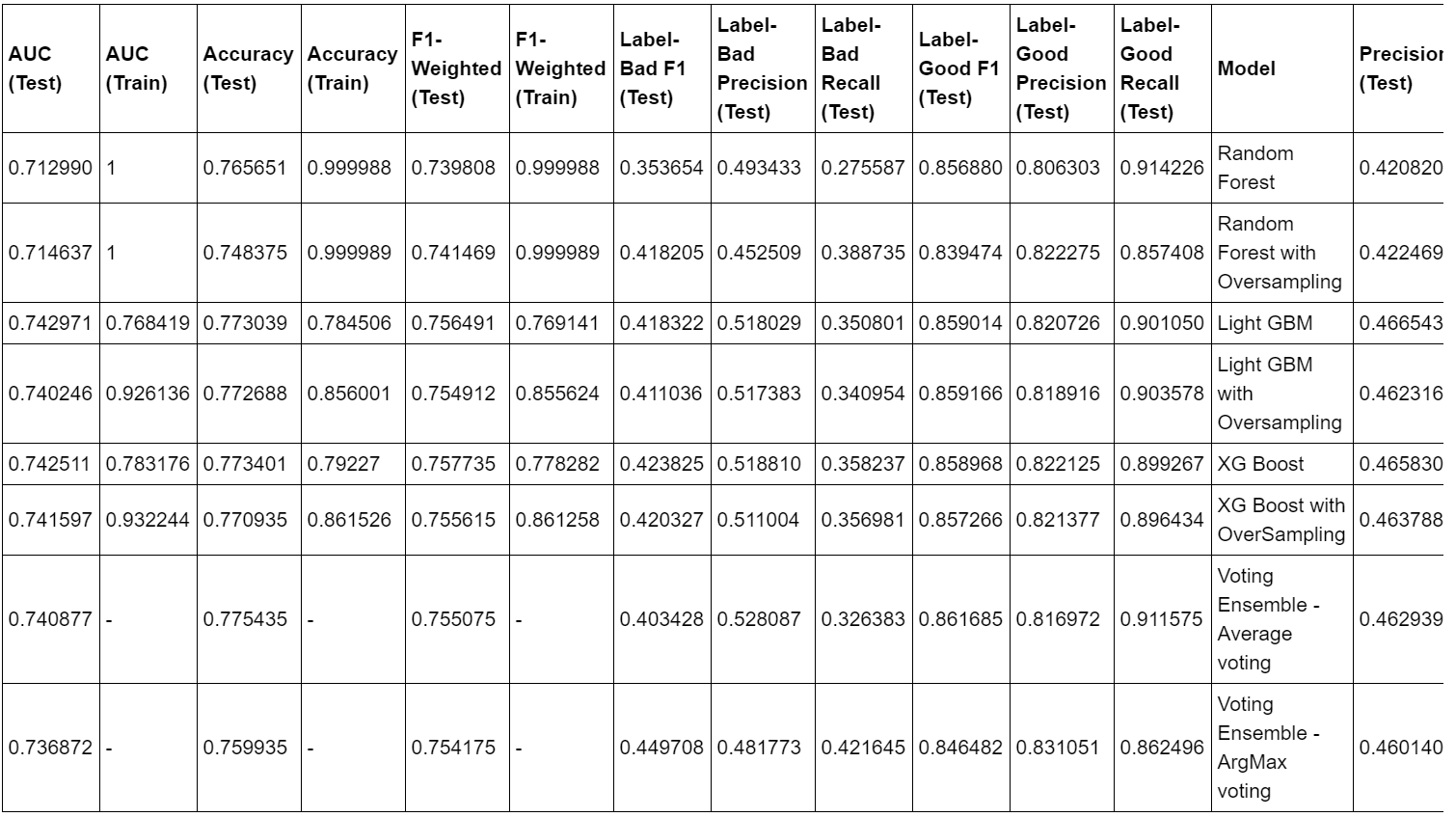
With the Argmax voting, the recall rate for Bad class gets a boost, but at the cost of the precision.

### ROC Curve



## Model Comparison

Below is a summary of all models that we saw in detail above.



In terms of AUC score, we’ve seen above that all the models, including the Voting Ensemble models gave the same AUC score – 0.72 or 0.74. So we will focus on the Precision – Recall values for the Bad loan class for model comparison

### Precision – Recall for ‘Bad’ loans

### Features

# Results & Discussion

The overall results from all the Predictive Models shows that there’s room for improvement. The AUC score of 0.74 & F1 weighted weighted score of 0.76 is not really ideal, but at the same time, the prediction made by this model can certainly be an input for the standards and procedures for the loan approval and funding process at LendingClub.

The performance has been fairly consistent

# Future Work

# Addendum – Code repository