**Final Report**

# **Prediction of LendingClub loan defaulters**

**About LendingClub:**

LendingClub is a US peer-to-peer lending company, headquartered in San Francisco, California. It was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market. Lending Club is the world's largest peer-to-peer lending platform.

Lending Club enables borrowers to create unsecured personal loans between $$1,000 and $ 40,000. The standard loan period is three years. Investors can search and browse the loan listings on Lending Club website and select loans that they want to invest in based on the information supplied about the borrower, amount of loan, loan grade, and loan purpose. Investors make money from interest. Lending Club makes money by charging borrowers an origination fee and investors a service fee.

## About the Dataset

These files contain complete loan data for all loans issued through the 2007-2015, including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. The file containing loan data through the "present" contains complete loan data for all loans issued through the previous completed calendar quarter. Additional features include credit scores, number of finance inquiries, address including zip codes, and state, and collections among others. The file is a matrix of about 890 thousand observations and 75 variables. A data dictionary is provided in a separate file. k

## Purpose of this analysis

We will go step by step for building a machine learning algorithm for the prediction of loan defaulters based on certain variables present in the dataset. Our main goal is to correctly identifying defaulter's (True positives) so that lending club can decide whether a person is fit for sanctioning a loan or not in the future.

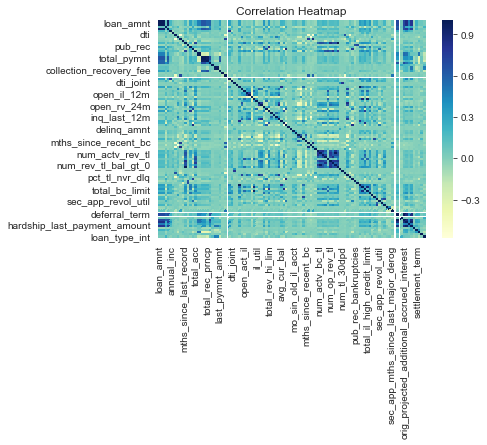
# **Data Cleaning**

As you can see, there are a lot of columns which have huge chunk of data missing. These columns are not necessary for our analysis. The following part will drop any columns where 80% or more data is missing. This will help us clean the Dataset a little bit.

Now that we explored the whole dataframe easily, we will now select the columns that are necessary for our analysis.

### **Finding the correlation between variables**

We will now look at the correlation structure between our variables that we selected above. This will tell us about any dependencies between different variables and help us reduce the dimensionality a little bit more

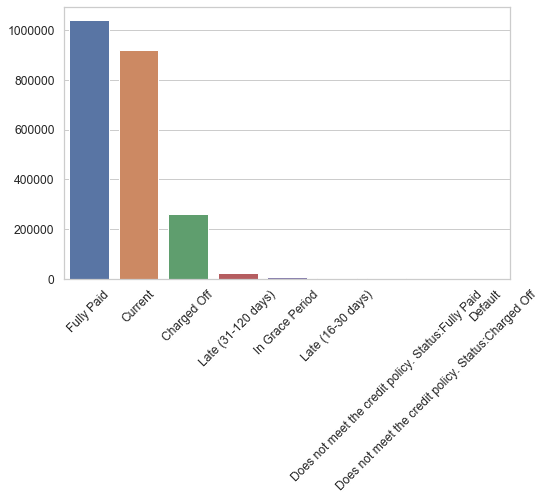


It can be seen from the plot above that loan amount and installment have a very high correlation amongst each other (0.94). This is intuitive since a person who takes a large sum of loan would require extra time to repay it back. Also, interest rate, sub grade and grade have a very high correlation between them. This is obvious since interest rate is decided by grades once the grades are decided, a subgrade is assigned to that loan (leading to high correlation).

Let's drop the three categories along with term and verification status (since it doesn't provide any valuable info) for further analysis.

# **Distribution of the loan status values**

Let us now see how the values in the status column are distributed. We will plot an histogram of values against count of times the status appears on the data frame.



As you can see, we have a lot of loans which are current with fair amount of fully paid loans. other categories (including) default have a really low number. This means the data is imbalanced and we might need to do something about this later in the analysis. For now, we will drop all the columns except 'Fully Paid', 'Default' and 'Charged off'. We will also merge 'Charged off' and 'Default' together meaning that anyone who fell into this category defaulted their loan. The following two parts tries to implement this.

We will now encode the two categories listed above as 0 or 1 for our analysis. This will help us in predicting whether a person defaulted their loan or not. 0 means he deaulted and 1 means he paid off his loan.

## Transformation

Before training the data, we would first transform the data to account for any skewness in the variable distribution. Various transformation techniques ranging from log transform to power transformation are available. For our analysis, we'll be using Box-cox transformation. It is used to modify the distributional shape of a set of data to be more normally distributed so that tests and confidence limits that require normality can be appropriately used.

# **One Hot Encoding**

Since we have some categorical variables for the analysis and the machine learning algorithms doesn't take categorical and string variables directly, we have to create dummy variables for them. We can either encode them using label encoder available for python, but it would be wrong in our analysis since a lot of these variables have multiple categories. Just using weights can cause discrepancies in the algorithm. Instead, we will one hot encode these so that we have a 1 wherever that category turns up and 0 otherwise. This will also create separate columns for each level of category. Also, we'll be dropping one of the categories so that we have N-1 columns instead of N.

**Modeling**

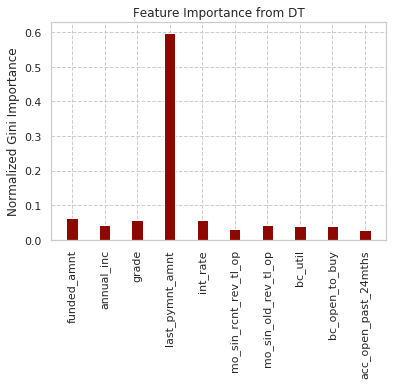
**Random Forest**

Random forest is a **Supervised Learning algorithm** which uses ensemble learning method for **classification and regression**.

**Random forest** is a **bagging** technique and **not a boosting** technique. The trees in **random forests** are run in parallel. There is no interaction between these trees while building the trees.

It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Random forest when implemented with randomized search we got the best accuracies and minimum false negatives (predicting borrower will not default even though he will. This might impact on the credibility of the company). We used the randomized search to find the best hyper parameters for the model.



We have a feature **‘last\_pymnt\_amnt’** as is shown above, has an importance of more than 50%. Random Forests select a subset of features in each of its decision trees thereby reducing the bias of the model because of high importance of single feature.

|  |  |
| --- | --- |
|  |  |

# **Logistic Regression**

Like all regression analyses, the logistic regression is a predictive analysis.  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Logistic Regression when implemented with grid search we got the best accuracies and minimum false negatives. We used the randomized search to find the best hyper parameters for the model.

|  |  |
| --- | --- |
|  |  |

**SVM**

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they’re able to categorize new text.

SVM when implemented with grid search, we got the best accuracies and minimum false negatives. We used the Grid search to find the best hyper parameters for the model. Later we used this value to find the predictions and plot the ROC curve.

|  |  |
| --- | --- |
|  |  |

**K-Nearest Neighbors**

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

KNN when implemented with grid search, we got the best accuracies and minimum false negatives. We used the Grid search to find the best hyper parameters for the model. Later we used this value to find the predictions and plot the ROC curve.

|  |  |
| --- | --- |
|  |  |

## Multi-Layer Perceptron Classifier

A multilayer perceptron (MLP) is a feedforward artificial neural network. An MLP consists of at least three layers of nodes. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. We got our highest accuracy for MLP classifier (little higher than Logistic Regression with Grid Search CV).

|  |  |
| --- | --- |
|  |  |

**Comparsion of models**

This is a summary of the models that have been applied to the data.

False negative rate is the best metric to evaluate the models as it shown in each model. Lower the number of false negative, better the model is. In this project, False negative is when model predictiing a borrower will not default a loan.

Although all the algorithms except KNN have almost the same accuracy, their False Negative rates differ which is our main evaluation metric.

From the confusion matrices, we can infer that Logistic Regression model has the least False Negative rate.

|  |  |
| --- | --- |
|  |  |