Lending Club Loan Data

Analyzing Lending Club's Issued Loans

Sejal | Jaiswal
September, 18th 2017

Table of Contents

1 Introduction

The goal is to analyze Lending Club's issued loans and to create prediction model using Machine Learning algorithms to predict clients who might default.
Default clients are the clients who have 'loan_status' variable as:

- Charged off
- Default
- Does not meet the credit policy. Status: Charged Off
- Late (31-120 days)

2 Data Understanding

The Leading Club data comprises of 'loan' table, which consists of 887383 rows (total number of loan takers) and 75 columns (features or information for the same).

Table	Total Rows	Total Columns	Columns
loan	887383	75	index, id, member_id, loan_amnt, funded_amnt, funded_amnt_inv, term, int_rate, installment, grade, sub_grade, emp_title, emp_length, home_ownership, annual_inc, verification_status, issue_d, loan_status, pymnt_plan, url, desc, purpose, title, zip_code, addr_state, dti, delinq_2yrs, earliest_cr_line, inq_last_6mths, mths_since_last_delinq, mths_since_last_record, open_acc, pub_rec, revol_bal, revol_util, total_acc, initial_list_status, out_prncp, out_prncp_inv, total_pymnt, total_pymnt_inv, total_rec_prncp, total_rec_int, total_rec_late_fee, recoveries, collection_recovery_fee, last_pymnt_d, last_pymnt_amnt, next_pymnt_d, last_credit_pull_d, collections_12_mths_ex_med, mths_since_last_major_derog, policy_code, application_type, annual_inc_joint, dti_joint, verification_status_joint, acc_now_delinq, tot_coll_amt, tot_cur_bal, open_acc_6m, open_il_6m, open_il_12m, open_il_24m, mths_since_rcnt_il, total_bal_il, il_util, open_rv_12m, open_rv_24m, max_bal_bc, all_util, total_rev_hi_lim, inq_fi, total_cu_tl, inq_last_12m

Figure 1: Loan Table information

2.1 Number of loans per state

The 'addr_state' feature helps visualize the total number of loans that were sanctioned in each state of the United States of America.

'LoansByState.html' is an interactive plot, and when hovered around, also shows the average loan amount sanctioned in every state, along with the highest and lowest interest rate.

It can be seen from Figure: 2, that California has the highest number of loans sanctioned, with average of $14655 and the highest Interest rate as 28.99% and the lowest as 5.32%.

The state with the lowest number of loans sanctioned is Idaho. The average loan amount here is $4362, highest interest rate as 15.45% and the lowest as 6.03%.

2.2 Number of missing values

Checking for NULL value in the data to identify the missing values. This information can be used in several ways:

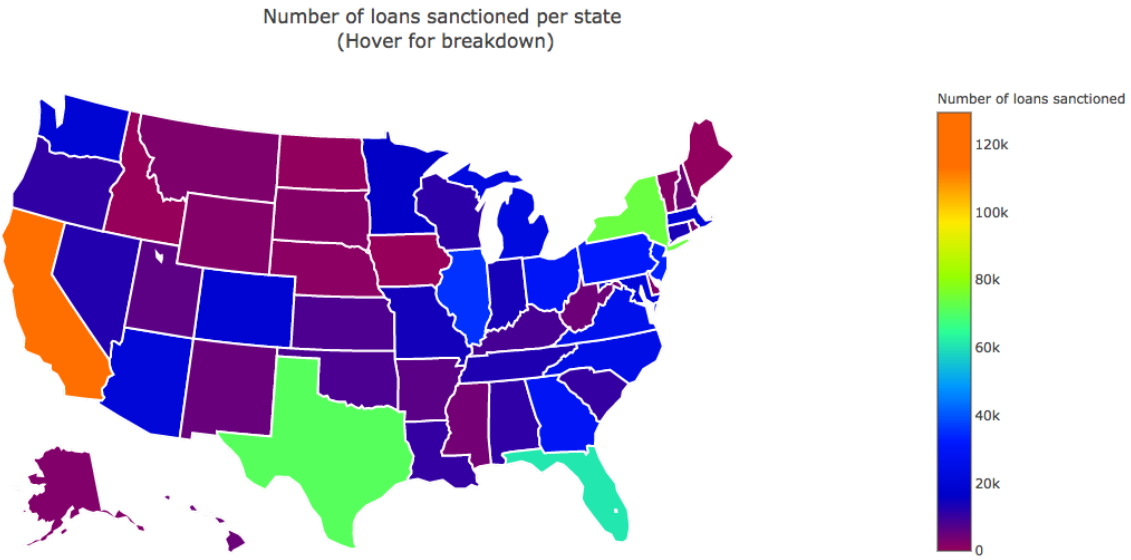

Figure 2: Number of loans sanctioned per State

- Supply the missing values with appropriate value (e.g. mean/median of the column) whenever possible.
- Could remove the column entirely if it is identified as unimportant information (for modelling purpose) or provides very less information.

The Figure: 3 shows the features with missing values.
The plot 'Freq_MissingValues.html' is an interactive plot, and when hovered around tells the exact number of missing values for the features.

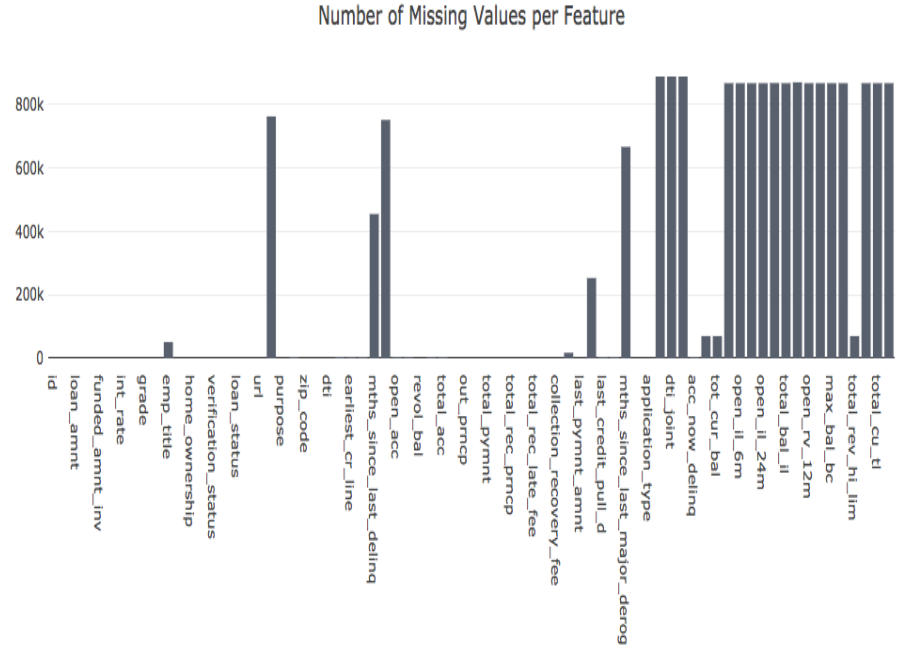

Figure 3: Number of missing values per column

2.3 Loan Status Distribution

The 'loan_status' value is the feature we are trying to predict. The different value for the features are:

- Charged Off (45248 values)
- Current (601779 values)
- Default (1219 values)
- Does not meet the credit policy. Status:Charged Off (761 values)
- Does not meet the credit policy. Status:Fully Paid (1988 values)
- Fully Paid (207723 values)
- In Grace Period (6253 values)
- Issued (8460 values)
- Late (16-30 days) (2357 values)
- Late (31-120 days) (11591 values)

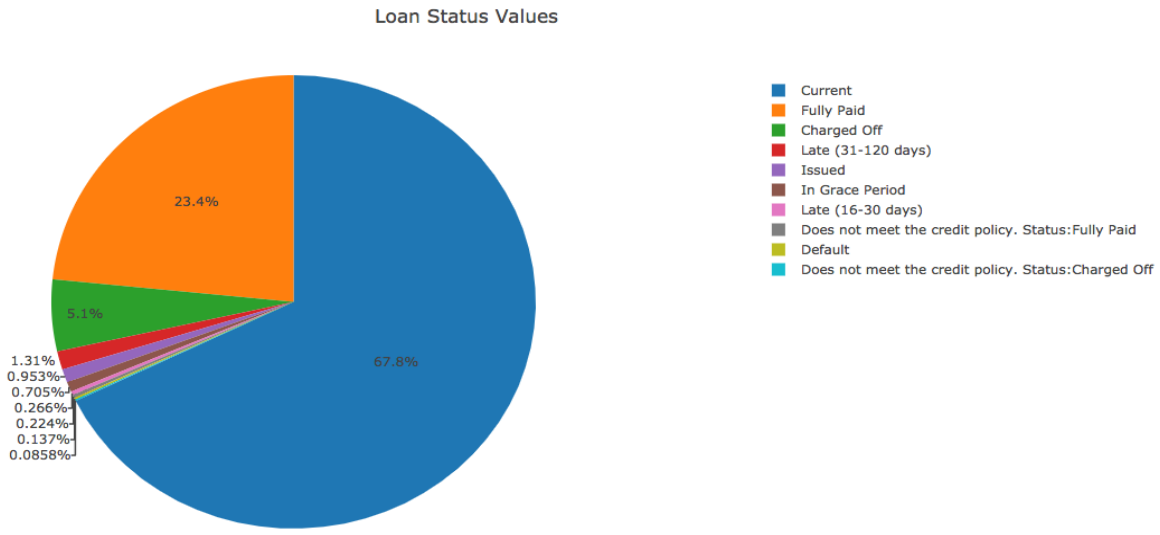

Figure 4: Loan Status distribution

The plot 'LoanStatus.html' is an interactive plot, and when hovered around tells the number of type of 'loan_status'.

The 'loan_status' feature when divided into defaulters and non-defaulters as described in the Introduction section, yields 58819 as 'defaulters' and 828560 as 'non-defaulters'.

Remark: The distribution clearly indicates the unbalanced nature of the data (Figure: 5).

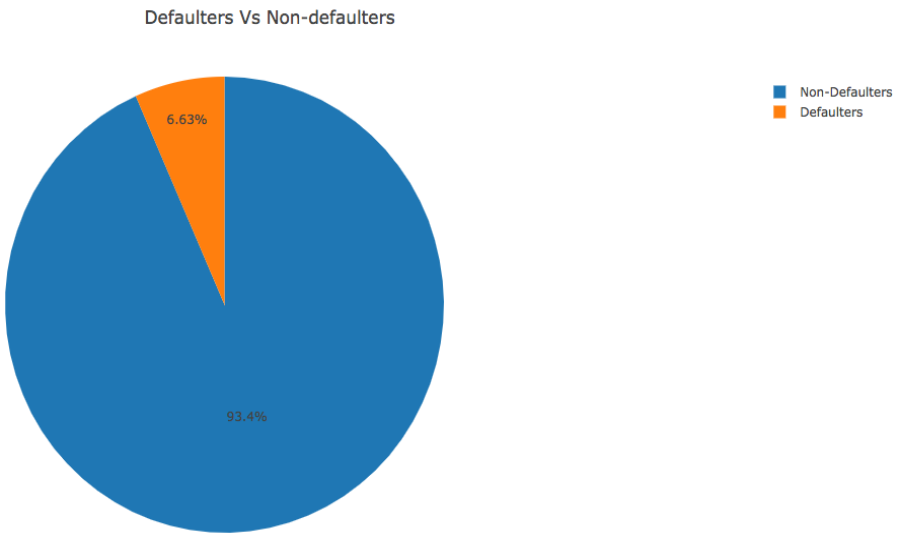

Figure 5: Defaulter VS Non-defaulters Distribution

2.4 Distribution by profession

'emp_title' is grouped according to profession and then sorted according to the group sizes. We then plot the 30 largest group of professions.
Teachers make up the largest percent of the distribution, with 16,394 in number. Followed by managers (11,240) and registered nurses (5,525).

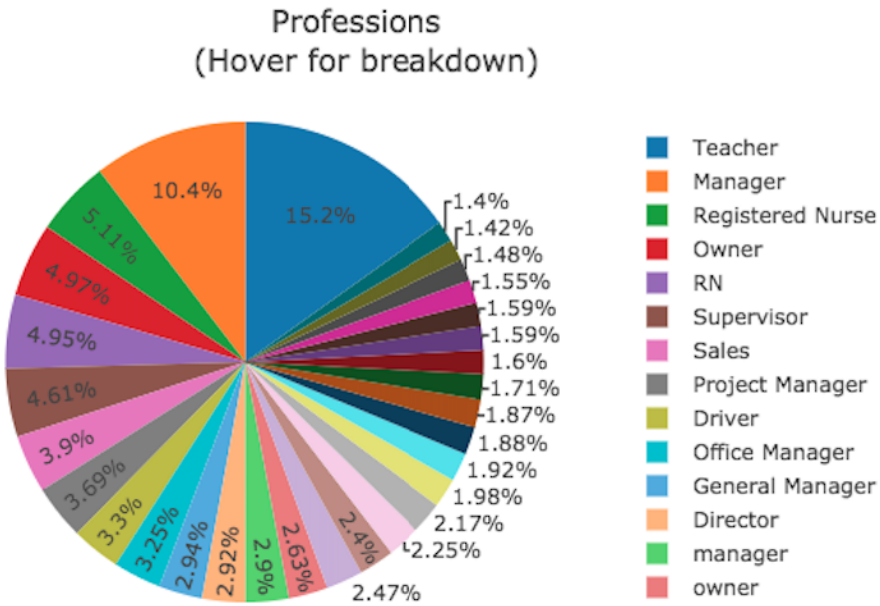

Figure 6: Distribution by Profession

The plot 'Professions.html' is an interactive plot, and when hovered around tells the number of different type of 'emp_title' (employment/profession title) present.

2.5 Reason for taking loan

The feature 'title' is analysed to display the main reason for taking loans. The words are reflected through a word-cloud.
The main reason for taking loans appears to be to pay off debt, credit card bills and other type of loans.

Figure 7: Reasons for taking loan

3 Data Preparation

3.1 Removing columns

- We remove the columns: 'acc_now_delinq', 'policy_code' since they do not provide any important or extra information as most of the values are majority 0 ('acc_now_delinq') and entirely 1 ('policy_code').
- We also remove the columns: 'url', 'desc', 'title'. These columns are useful and can provide extra information when using Natural Language Processing or Semantic Web Mining using the url. But this is out of the scope of the current project and hence removed.
- We remove the columns: 'emp_title', 'annual_inc', 'title', 'delinq_2yrs', 'earliest_cr_line', 'inq_last_6mths', 'mths_since_last_delinq', 'open_acc', 'pub_rec', 'revol_util', 'total_acc', 'last_pymnt_d', 'next_pymnt_d', 'last_credit_pull_d', 'collections_12_mths_ex_med', 'tot_coll_amt', 'tot_cur_bal', 'total_rev_hi_lim'.
  These columns have majority or some NULL values, although this is not a good practise to remove columns that have NULL value. However, we remove these for feature elimination, we shall reconsider this decision if we do not get good modelling results later on.

3.2 Data Type Conversion

We try to convert the datatype of columns into integer values, so that the columns can then be used to calculate feature importance.

- Converting 'int_rate' to integer data type by removing '%' sign.
- Converting 'emp_length' to integer data type by extracting the number from the value.
- We apply label encoding to these terms: 'grade', 'sub_grade', 'home_ownership', 'issue_d', 'verification_status', 'loan_status', 'pymnt_plan', 'purpose', 'zip_code', 'addr_state', 'initial_list_status', 'application_type'.
  The features do not have data that fits one-hard encoding technique. And although it does not make much computational sense to convert some features with many labels using 'label encoding'. However, the conversion is done strictly for converting the data-type and to be able to use it for feature importance calculation.

We also perform a data scaling to all the feature values.

3.3 Feature Importance Calculation

We use ExtraTreesClassifier method to calculate the importance of each feature. We drop the features that have importance value less than or equal to 0.01.
We then use SelectFromModel to transform the data according to the output of the 'ExtraTreesClassifier'.

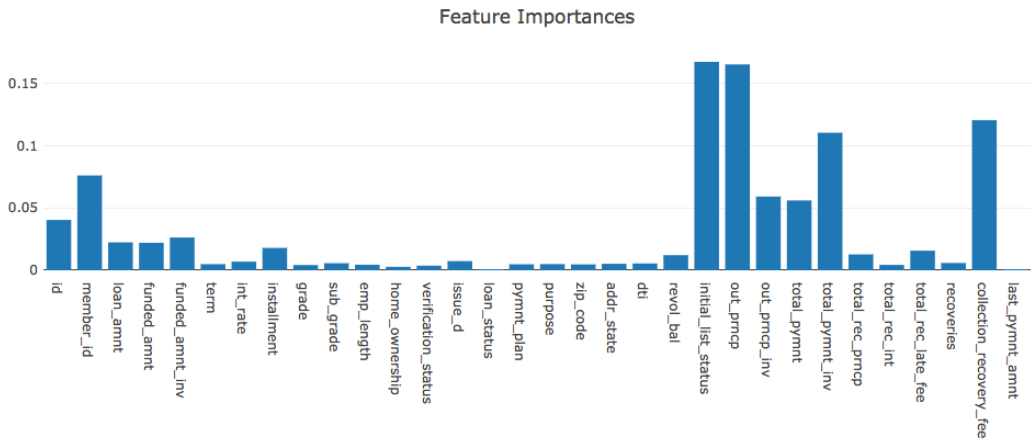

Figure 8: Feature Importance

We end up with a data having 15 features after the feature calculations.

4 Modeling

4.1 Model 1. Random Forest

The first model we apply is a simple Random Forest Classifier. We supply the class to be predicted (Y) as a binary sample, that is divided into 'defaulter' (Class 0) or 'non-defaulter' (Class 1) according to their 'loan_status'.

4.1.1 Evaluation

The best accuracy score (applying KFold iterator variant to the data) : 98.52%
The AUC score for the model is: 0.74

Real \ Predicted Value	0	1
0	3050	4043
1	524	214227

Figure 9: Confusion Matrix for Random Forest

The ROC plot shows that the False Positive Rate is very high. The same conclusion can be drawn from the confusion matrix. This depicts that the model is not able to learn well the 'Defaulter' Class and constantly makes error on the same. This could be attributed to the fact that the data given is highly unbalanced and thus further data augmentation could be done to better learn from the model.

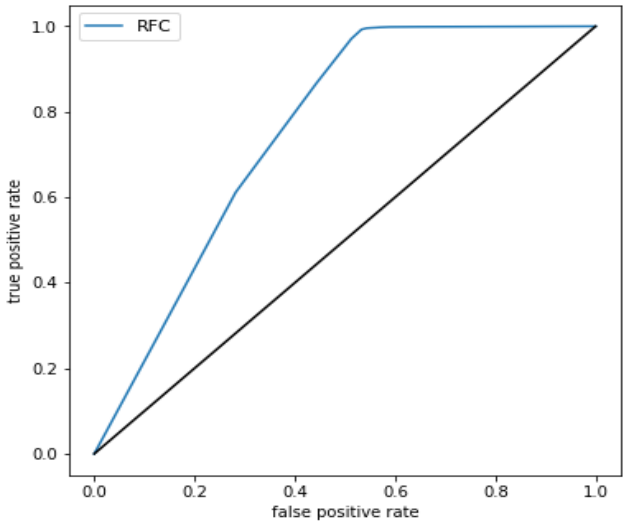

Figure 10: ROC model for Random Forest Model

4.2 Model 2: Gradient Boosting Regression

The second model we developed is Gradient Boosting Regression. This was done to check if a regression method is able to classify the data. We supply the class to be predicted (Y) as a binary sample, that is divided into 'defaulter' (Class 0) or 'non-defaulter' (Class 1) according to their 'loan_status'.

4.2.1 Evaluation

A score of 74.35% was obtained.
Since the score is no better than the first model, we try to model the data using a different methodology.

4.3 Model 3: Multi Layer Perceptron

The third model we developed is a simple Multi Layer Perceptron model. As earlier, we supply the class to be predicted (Y) as a binary sample, that is divided into 'defaulter' (Class 0) or 'non-defaulter' (Class 1) according to their 'loan_status'.

4.3.1 Evaluation

The best accuracy score (applying KFold iterator variant to the data) : 98.47%
The AUC score for the model is: 0.96

Real \ Predicted Value	0	1
0	6558	535
1	31937	182814

Figure 11: Confusion Matrix for Multi Layer Perceptron

The ROC plot and the confusion matrix show that the model has relatively low false positive rate as compared to Random Forest Classifier (Model 1).

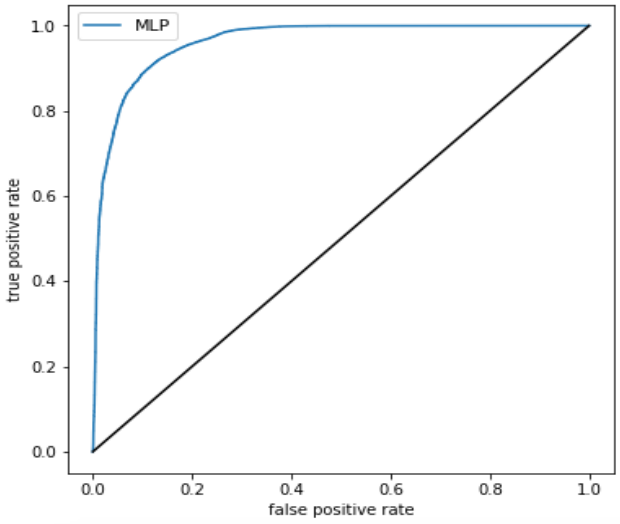

Figure 12: ROC model for Multi Layer Perceptron

5 Deployment & Conclusions

It is evident from the confusion matrix and the AUC graph that Model 3, the Multi Layer Perceptron works better for the given dataset as compared to the other models developed. Hence, we can select Model 3 as a predictive model for the dataset.