**Lending Club Loan Data: Analyze Lending Club Issued Loan’s**



LendingClub is one of the world’s largest online marketplace that connects borrowers and investors. Lending Club have reinvented the way people access credit. LendingClub makes credit more affordable and investing more rewarding. They strive to provide a better financial future for everyone by building a marketplace that keeps costs low and investor opportunity high.

**Goal: Analyze LendingClub data to predict whether a borrower will default**

**Problem:**

LendingClub performs underwriting in a way that is different from a traditional bank. LendingClub screens the borrowers through the information that they have provided and evaluates the loan decision based on that information. LendingClub then assigns a rating to the loan and determines the interest rate on the loan. LendingClub then puts the loan on the marketplace where investors are able to evaluate the loan and invest. The main problem that we are trying to solve is to improve the credit scoring system of LendingClub using machine learning algorithms to predict whether a borrower will default on the loan. It is essential for LendingClub to access the validity of a loan to assign an accurate rating so that investors are provided with an accurate estimation. Otherwise, investors would be unlikely to invest on the platform. Additionally, credit risk is something that all investors must carefully consider when making investment decisions. The risk of default is the result of the borrower failing to make principal payments and interest.

The loan is one of the most important products of the banking. Banks are trying to figure out effective business strategies to persuade customers to apply their loans. However, some customers behave negatively after their application is approved. To prevent such a situation, banks and P2P lending platforms have to find some methods to predict customers’ behaviors.

Lending Club makes money through origination and service fees. Borrowers pay a one-time origination fee of 1.11% to 5% of the total loan amount, depending on the loan grade and term. Meanwhile, investors pay a service fee of 1% of each payment received from a borrower. Therefore, it is imperative and in good business sense to help the investors to make more investments in the platform while trying to help them reduce their credit risk.

The business objective is to build a model to predict whether or not a loan will default to prevent an investor from losing money so that they can continue using the LendingClub platform. A loan default will result in loss of both principal and interest.

**Dataset:**

LendingClub provides the ability to download the data from their website which includes complete loan data for all loans issued and latest payment information. However, the same dataset is available in Kaggle which was previously used for a competition that I am planning to use. The data contains loan data for all loans issued through 2007-2015, including the current loan status and latest payment information. It also contains present information on loans issues in last quarter. Aside from LendingClub data, it also contains various other features like credit scores, number of credit inquiries, address, total collections ever owed etc. The csv file contains 890,000 observations with 75 features. Additionally, a data dictionary has been provided to understand the features.

**Approach**:

Our goal is to predict whether the borrower will default on the loan or not. Given that our data is labeled, and our predictions will be placed in various categories. We will be tackling a supervised classification problem.

Exploratory Data Analysis

The data provided in this dataset is fairly clean and formatted, however, we will need to deal with null and empty values. We will also need to look for outliers or transform categorical values. Dates would need to be converted into datetime objects as well. The next step would be to do some early data exploration analysis as well data visualizations to understand what a bad borrower looks like and some commonalities shared by them. Furthermore, it will also help to understand how the data is distributed.

Feature Selection

With more than 75 features, it is not necessary to use all of them. Feature selection would need to be done to discover important features that are indicative of a borrower defaulting or not. Moreover, it will help us understand the significant features that lead to defaults.

Algorithms

Since this is a supervised classification problem, the algorithms that will be tried to build the model will be:

🡪Logistic Regression

🡪K-Means

🡪SVM

🡪Decision Trees

🡪Random Forests

🡪Gradient Boosting

🡪AdaBoost

Evaluation

It is important to use a test set that has been set aside at the start of the project. A confusion matrix will be used to describe the performance of the classification. Accuracy will be an important metric to understand however, since we have slightly imbalanced class it is better to use other metrics. Recall & ROC will be the performance which will help evaluate our classifier. The reason that we will need to use recall is that we are focused on minimized our false negatives. We want to we predict default and they actually were people that paid off is not as big of a concern as people that we predicted to paid off, but they default. We want to make sure that we can minimize this.

**Data Wrangling Report**

**Lending Club Loan Data: Analyze Lending Club Issued Loan’s**



The Lending Club dataset contains 74 features and 887378 instances in a csv file. The target variable is loan\_status. The two goals of the project are to be able to predict whether or not an individual will default and also to understand the features of a person that defaults.

The dataset is derived from Lending Club a subset of which was posted on Kaggle for competitions. The dataset itself is relatively clean and has good quality thus, external sources will not be required to achieve the objectives. The data contains loan data for all loans issued through 2007-2015, including the current loan status and latest payment information. It also contains present information on loans issues in last quarter. Aside from LendingClub data, it also contains various other features like credit scores, number of credit inquiries, address, total collections ever owed, etc.

The file was read in through the pandas csv function and an early look at the data showed the data is a mixture of object, int and float datatypes. A look at the histograms of the features show that dti, loan amount, grade, total accounts and interest rate seem to be normally distributed. Some features are very top heavy distributed, this may make it hard for the machine learning algorithm to detect patterns. Another thing to consider is to make sure that we have enough data and it is representative of the whole population. For example, we want to make sure that loan\_amnt is not capped to any amount as new data may get generalization errors.

Our target variable is loan\_status. We filtered out loans whose statuses are not yet final, such as “current”, “Late (less than 30 days)”, “In Grace Period”. Our positive label was “Paid Off” which represents loans that were fully paid off. Our negative label required a merge of “Default” and “Charged Off”. Loans that are in "Default" are loans for which borrowers have failed to make payments for an extended period of time. Charge off typically occurs when a loan is 120 days or more past due and there is no reasonable expectation of sufficient payment to prevent the charge off.

The next step was to look at missing values. There were 15 features that contained a significant amount of missing data. We removed such features because they only 2.4% of the data such as inq\_fi, total\_cu\_tl, dti\_join, annual\_inc\_joint etc. Since we had more than 800k instances, it was viable to also remove loans that were missing data for any loan without losing any significant information as it represented only about 3% of the loans in the dataset.

The data contained a few fields that are not immediately usable for a learning model like zip code and loan description. The description contains freeform text by the loan requestor and may contain keywords that can give information to defaulting or non-default loans. These were removed as they were out of scope of this project. However, for future work, TF-IDF can be used on the descriptions to gather top 20 keywords and be used to create binary features. Similarly, census data by zip code could be added to the dataset which could be helpful in determining whether certain areas are more prone to defaulting. For purposes of this project, they were removed.

Similarly, some features columns had either the same feature or features that were unique for every instance that would lead to overfitting in the models. Features like loan\_id, member\_id, application\_type, initial\_list\_status and policy\_code were removed. Purpose and Employee\_title required a bit of cleaning as various categories are repeated like ‘Teacher’,’teacher’,’school teacher’ all are the same and needed to be merged using simple regex cleaning.

The next objective is to scale the data and make it into a machine-readable format. We have two datatypes in our datatype: float32 and object. The categories can be converted using one hot encoding including our target variable so that the data is in a usable format. A standard scaler is used to scale the data to get the variances in the same range.

Finally, the last thing was to look at outliers in the continuous data. Building a scatter plot of the default data with loan amount and interest rate. It was easy to see that most annual amount of for the defaulted category are in the range of 0-40k however a single instance of 120k stands out which is an outlier and can add noise in the data. Outlier detection is tricky in a credit risk model because outliers contain data that is needed to predict whether or not an individual will default.

**Exploratory Data Analysis**

**Lending Club Loan Data: Analyze Lending Club Issued Loan’s**



Exploratory Data Analysis Outline:

1. Introduction

* General Distributions

1. Good Loans vs Bad Loans

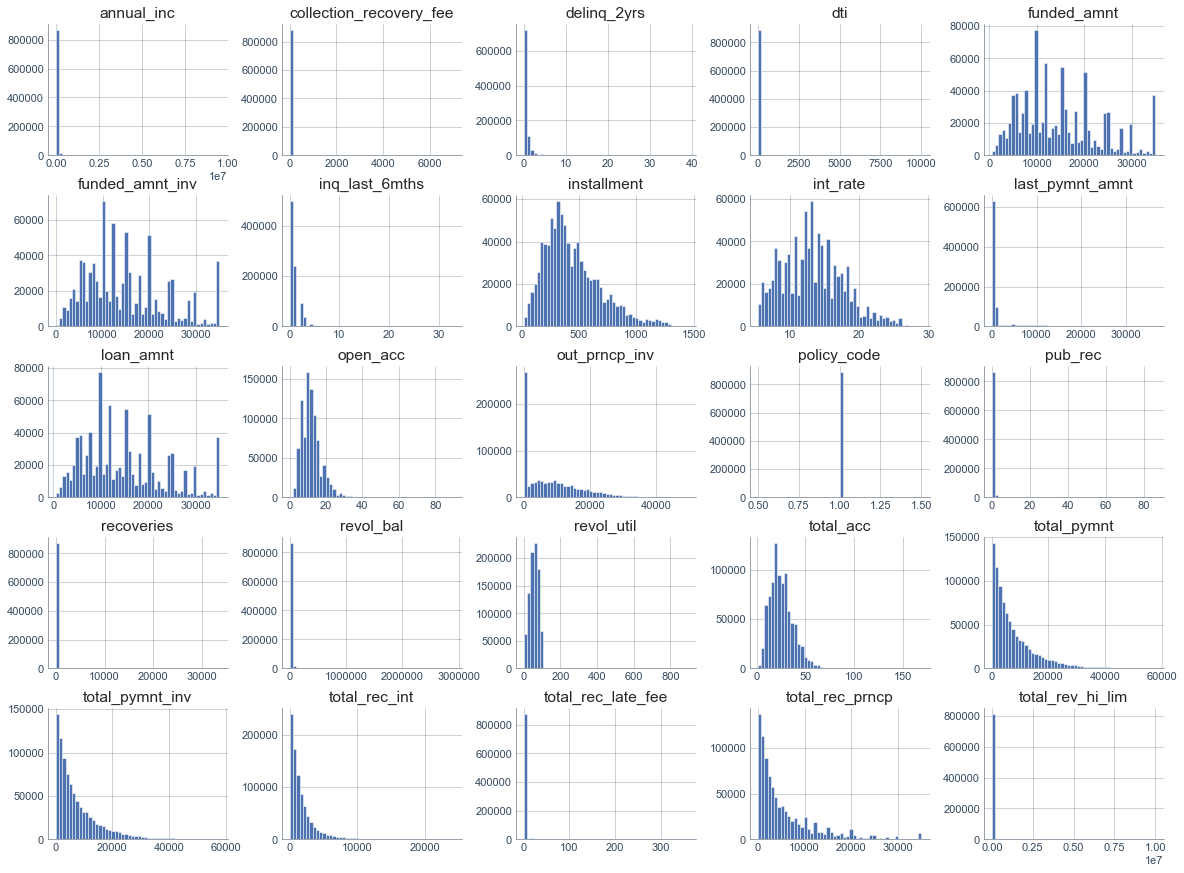
* Types of Loans
* Bad Loans

Introduction

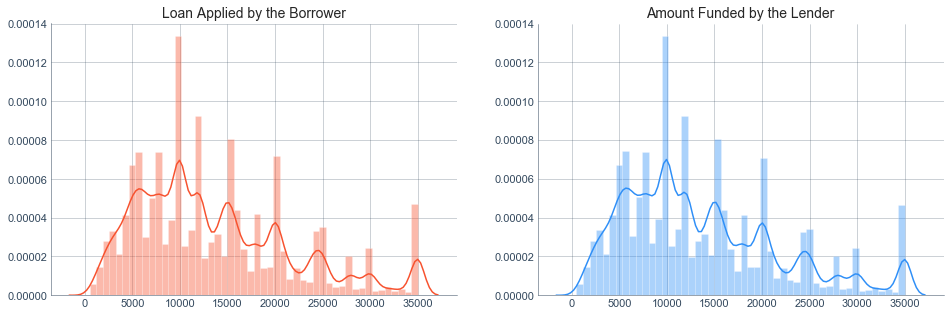
Each row represents a singular loan. There are 74 features with 887,379 instances. The features were a mixture of three datatypes- objects, float and int. More specifically, 49 features were float64, 2 features were integers and 23 features were objects.

A quick look at the distribution of the continuous data showed that dti, loan amount, grade, total\_acc, int\_rate seem to be normally distributed and intuitively are important factors that we will validate later. A lot of the features have very different scales. Some features are top heavy, this may make it hard for machine learning algorithms to detect pattern. Another thing to consider is we have enough data like is the loan\_amnt capped and is it representative of the whole population.

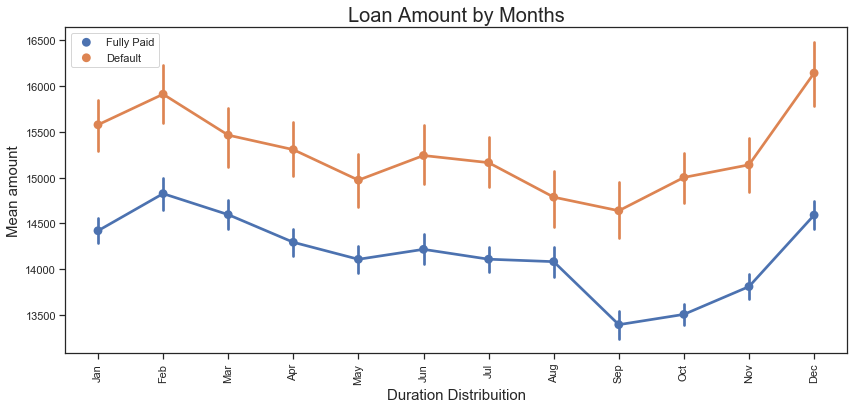
Using pandas, we can build histograms of the continuous features that can give us a feel for the data that we are working with and also the distribution of the data.

A dirty look at the data at hand:

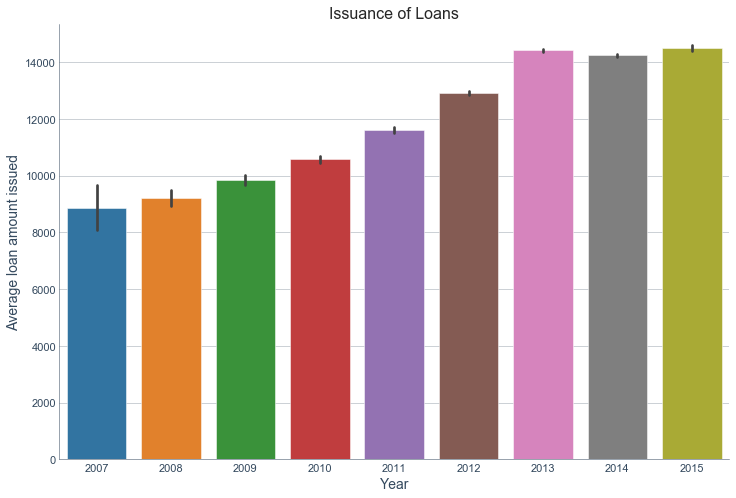
Some distributions that we are really interested in are loan\_amnt for the borrowers and the amount that investors are funding.



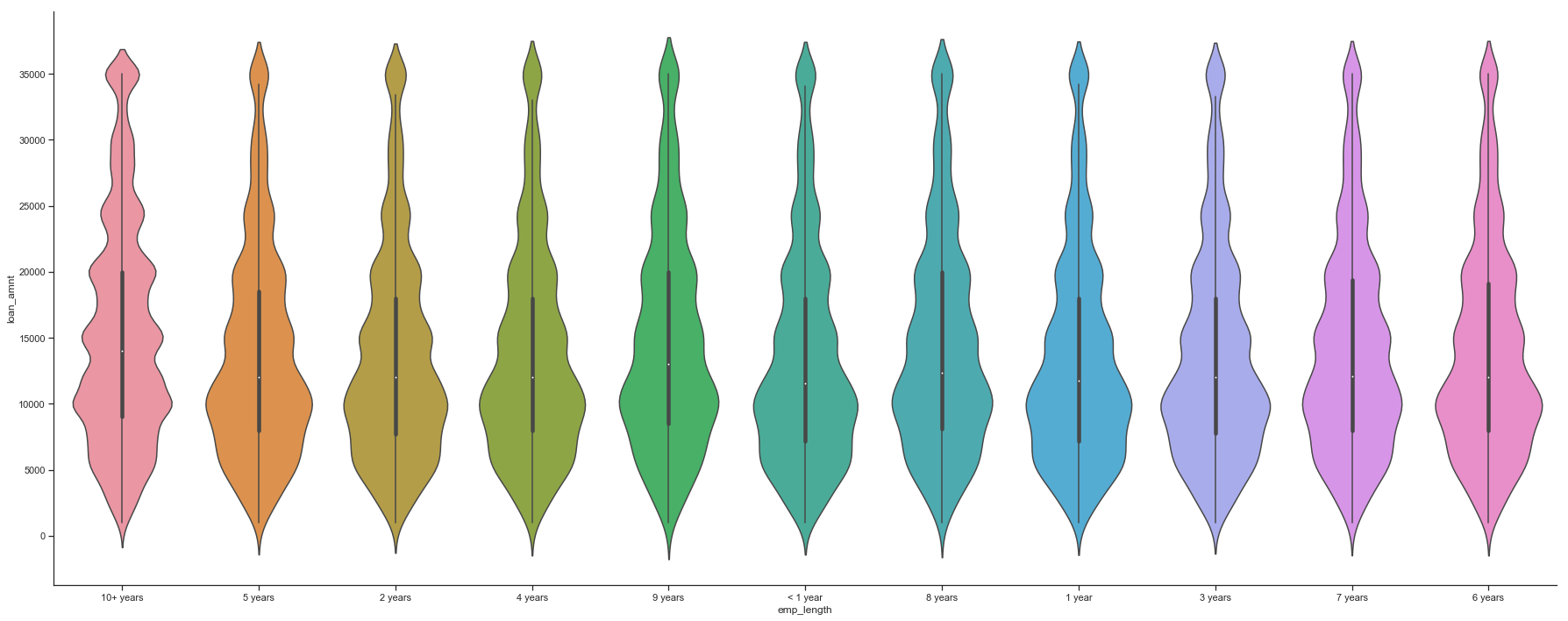
Most of the loans issued were in the range of 10,000 to 20,000 USD. The loans applied by potential borrowers, the amount issued to the borrowers and the amount funded by investors are similarly distributed, meaning that it is most likely that qualified borrowers are going to get the loan they had applied for. This is later validated by our correlation analysis which shows they are highly correlated. Finally, quick thing to notice is that loans are issued in increments which give the peaks and lows at the thousands.



Very interesting. It is surprising that people who default borrow more than people who pay off their loans. Another thing to look for is the borrowing increases the months of October, November and December. A reason for the spike at the end of the year could be because of the holiday season and people consolidating their debt heading in the new year.

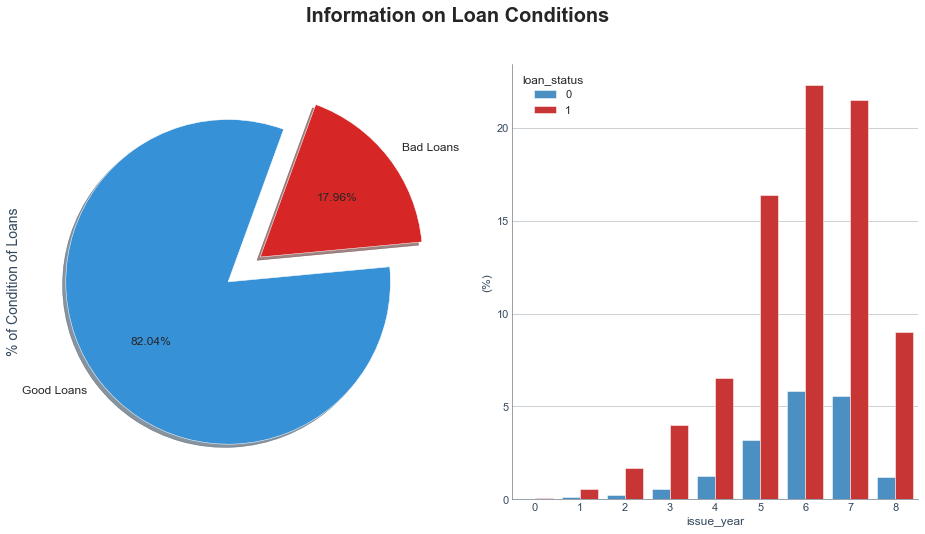


A look at the yearly distribution for the issuance of loans. Clearly, they have been issuing more loans every year. 2015 was the year were the highest amount of loans were issued.



Also, interesting to see the distribution of the amount that is borrowed compared to how long someone has worked. People who have worked more than 10+ years tend to have fatter tails meaning they borrow more because they have more income. People who have been employed for lesser time tend to have fatter bottoms for the same reason.

II. Good Loans vs Bad Loans



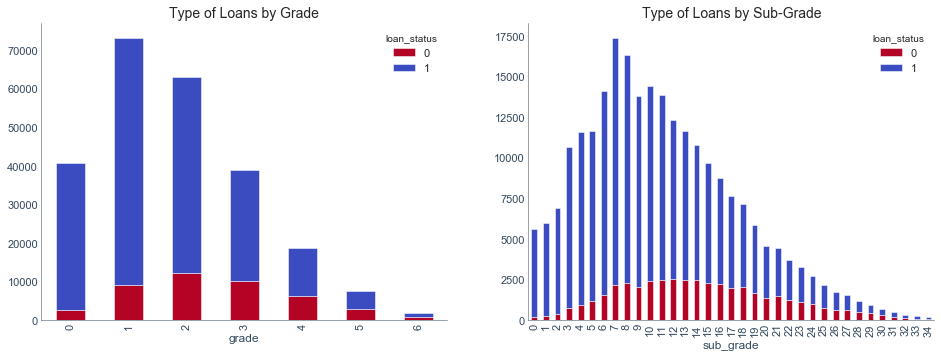
A loan becomes “Charged Off” when there is no longer a reasonable expectation of further payments.  Charge Off typically occurs when a loan is 120 days or more past due and there is no reasonable expectation of sufficient payment to prevent the charge off. Thus, to predict whether or not a person will default we can merge charged off and default.

The first thing to count is the number of people that fully paid off their loan vs the number of people that defaults.

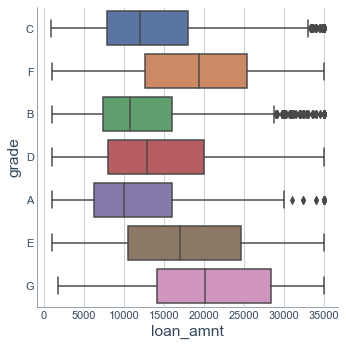
Fully Paid 148052

Default 33833

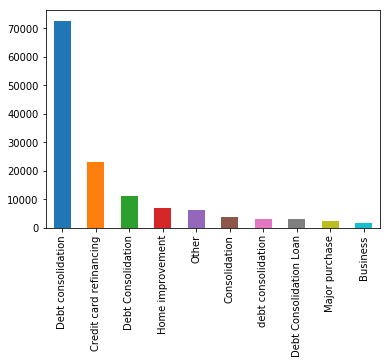
Therefore, 18.6% of all loans in our dataset are bad loans as compared to loans that been fully paid off, however, we have to remember that there are still current loans that are ongoing that have the possibility of becoming bad loans.



Type of Loans by Grade help visualize how the proportion of default changes as the grades change. Majority of loans defaulted lie in the C and D range however, the proportion of paid-off to default changes drastically and at higher grades almost ranges in 50% default rate. Thus, grade is an important indicator of loan default and will be an important feature in our ML models.

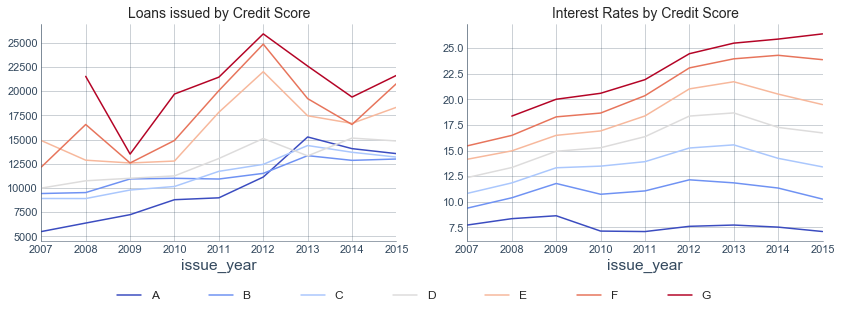


The top 10 title provided by borrowers. As we can see the main reason that people using Lending Club is for debt consolidation. A reason for that can be that people that resort to Lending Club is because they have exhausted the traditional means of borrowing.

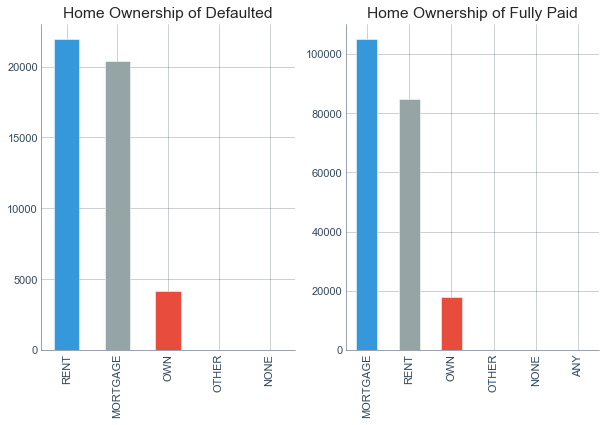


Due to most of the loans being of B and C grade the majority of the loans are of the shorter term of 36 months as they are less risky because they represent a shorter time for a loan to be paid off.

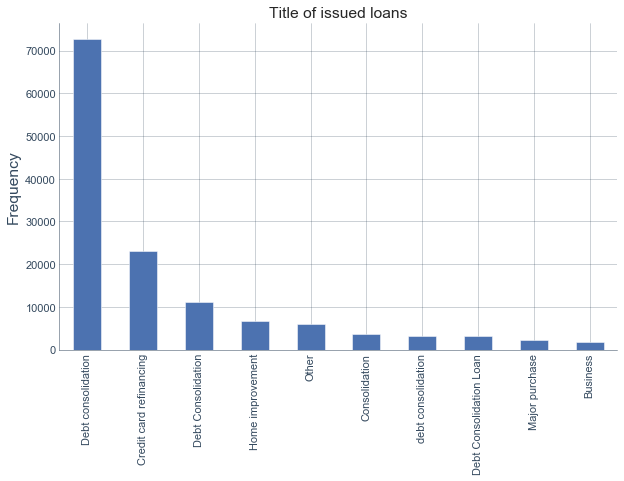




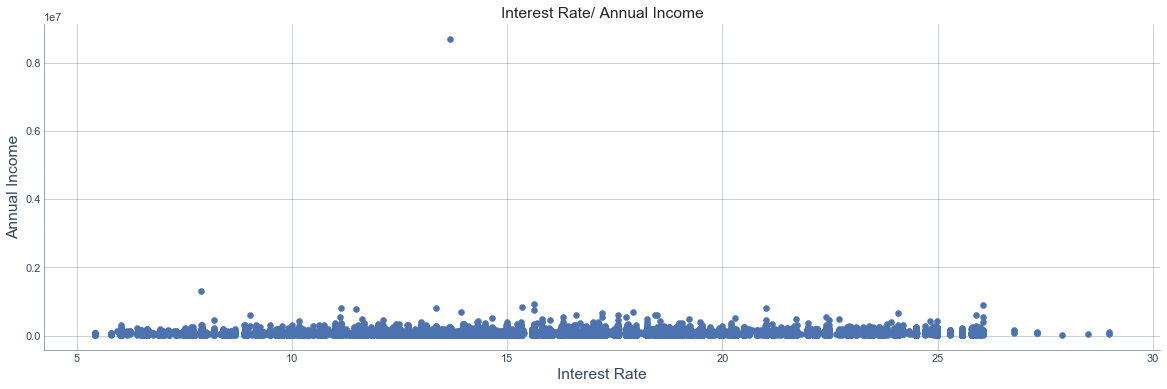
Credit scores are important metrics for assessing the overall level of risk. In this section we will analyze the level of risk as a whole and how many loans were bad loans by the type of grade received in the credit score of the customer. The scores that has a lower grade received a larger amount of loans (which might had contributed to a higher level of risk). Logically, the lower the grade the higher the interest the customer had to pay back to investors. Interestingly, customers with a grade of "C" were more likely to default on the loan



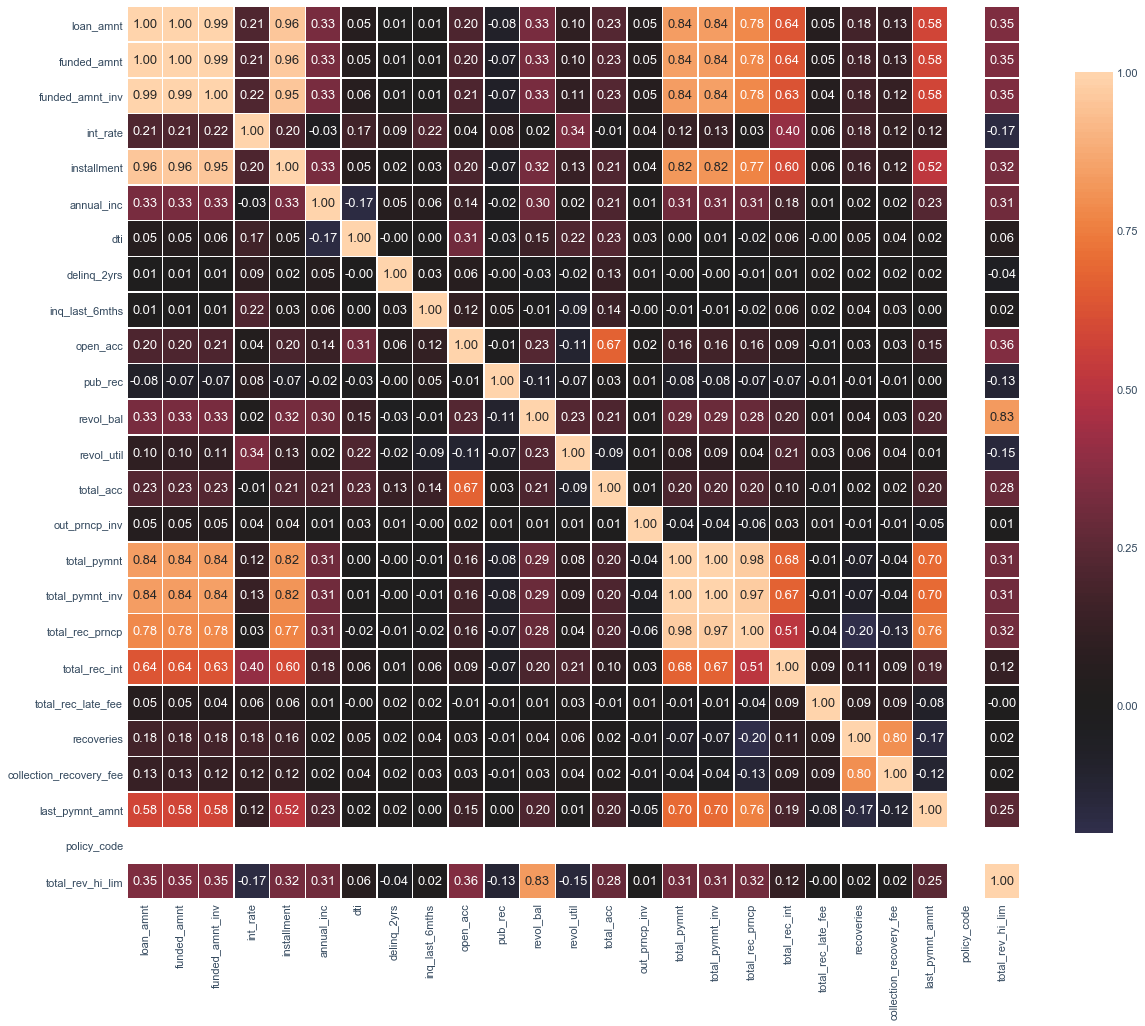
Mortgage and Rent are the two main categories of the home ownership on Lending Club. It is interesting to see how people who default mainly people who rent their homes, but it is closely followed by people with mortgages. Moreover, people who have paid off their loans it is easier to see are people with mortgages. People who have a mortgage (depending on other factors as well within the mortgage) are more likely to ask for. Obviously, we see less people who own houses as they are financially better off and don’t need loans.



Unsurprisingly, debt consolidation is the most common reason for borrowing. The greatest advantage of peer-to-peer lending is the low cost. Loans issued by LC usually charge lower interest rates compared with money provided by traditional banks. Most consumers choose to consolidate debt to enjoy lower borrowing costs. It helps give an overall view of the relationship between purposes of loans and the volume and amount of loans.



Just looking at the interest rate vs the annual income we can see that there is a clear outlier that will skew our results and something that we will need to remove. We do not see such high-income people apply for loans on Lending Club and thus it is a clear outlier will need to be removed.



 Correlation map is one of the most important ways to figure out the multicollinearity among dependent variables. There are a few features that are very highly correlated and would need to be removed. Features like loan\_amnt,funded\_amnt, funded\_amnt\_inv and installment would need to be removed as they add extra noise to our algorithms.

**Prepare the data for ML**

Having eliminated all redundant features or those that provide no information or lead to overfitting. I used a StandardScaler to change the values so that the distribution of standard deviation from the mean equals one. It outputs something very close to a normal distribution. Many machine learning algorithms perform better or converge faster when features are on a relatively similar scale and/or close to normally distributed. It also allows the data to be in a digestible form.

The next step was to convert categorical text data into model-understandable numerical data, we use the Label Encoder class. LabelEncoder encode labels with a value between 0 and n\_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to as assigned earlier.

The final dataset contained 254,190 rows and 28 columns.

**Training Machine Learning Algorithms**

The first thing we did was to split the data into training and test data. We used a test size of 0.3 to split the data as we have a good amount of data.

Our split results:

Training Data:(170821, 27)

Testing Data :(73209, 27)

The number of instances is 244030

The number of default class are 43827

The number of fully paid class are 200203

To check whether the percentage of default class is roughly the same in both training and testing class.

% Default: 17.959677088882515

Train - size: 170821 , %Default: 17.857289209172176

Test - size: 73209 , %Default: 18.198582141539973

Therefore, the dataset is slightly imbalanced which would require us to be careful when using performance measures.

Logistic Regression – Our best model

Logistic Regression takes in a list of features as input and outputs the Sigmoid of a linear combination of features weighted by learned parameters θ, i.e. hθ(x) = g(θ T x) = 1 1+e−θT x . To derive optimal parameters, the model iteratively updates weights by minimizing the negative log likelihood with L2 regularization. To tackle the class imbalance problem (only 19% of our dataset are negative examples), we used balanced weight for class labels, which is inversely proportional to class frequencies in the input data.

With a 5-fold cross validation we get the results:

[0.93908971 0.93850252 0.93955626 0.93645358 0.93888303]

0.938497019730057

Average of 5-fold CV: 0.938497019730057

Initially we get an accuracy of.

Accuracy of logistic regression classifier on test set: 0.94

We are trying to reduce our False Negatives. Our primary performance metric that we are concerned with is recall. Whenever it is default but often is it correctly predicted to be default. The reason for that is if we predict default and they actually were people that paid off is not as big of a concern as people that we predicted to paid off but they default. We want to make sure that we can minimize this.

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 12,358 | 683 |
| True Paid Off | 3776 | 56,392 |

precision recall f1-score support

0 0.76 0.95 0.84 13369

1 0.99 0.93 0.96 59840

micro avg 0.93 0.93 0.93 73209

macro avg 0.87 0.94 0.90 73209

weighted avg 0.95 0.93 0.94 73209

Since our dataset is imbalanced, we would need to change the probability threshold. We could use ROC curve to find the right threshold and we have to keep in mind that 80% paid off class which should give us an idea for our threshold.

Changing our threshold to 0.8 which changes our decision boundary, we get the following results:

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 12,669 | 670 |
| True Paid Off | 4102 | 55,738 |

precision recall f1-score support

0 0.76 0.95 0.84 13369

1 0.99 0.93 0.96 59840

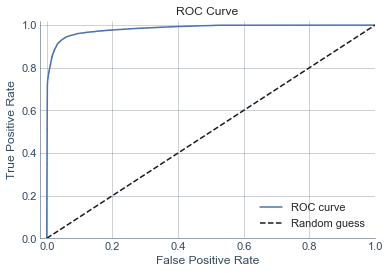
micro avg 0.93 0.93 0.93 73209

macro avg 0.87 0.94 0.90 73209

weighted avg 0.95 0.93 0.94 73209

As we can see, Logistic Regression is doing fairly well compared to naive models that blindly predict positive for all examples, or randomly guess positive and negative with 50% chance. Thanks to L2 regularization, we did not observe overfitting issues. One thing that we noticed, and we would like to improve upon, if possible, is the recall for default class. Although we used balanced class weights to offset data imbalance, changing the threshold has resulted in better results.

This can be seen by the ROC curve as well which shows that the classifier is performing very well.



Bagging

precision recall f1-score support

0 0.94 0.95 0.95 13032

1 0.99 0.99 0.99 60177

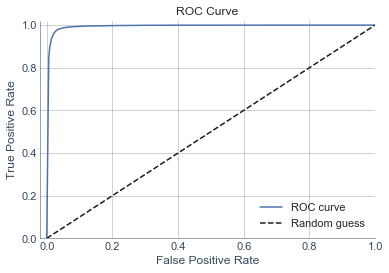
micro avg 0.98 0.98 0.98 73209

macro avg 0.97 0.97 0.97 73209

weighted avg 0.98 0.98 0.98 73209

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 12,336 | 705 |
| True Paid Off | 713 | 59,455 |

Bagging, however, performs very well in both predicting default and paid-off. Logistic Regression slightly outperforms for predicting the negative class. This suggests that it captures some non-linear relationships that are not captured by logistic regression.



Random Forest Classifier:

precision recall f1-score support

0 0.94 0.92 0.93 13041

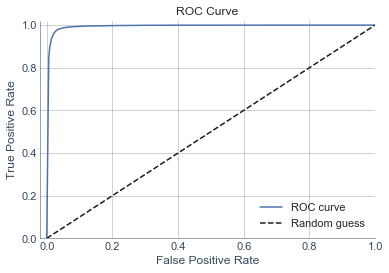
1 0.98 0.99 0.98 60168

micro avg 0.97 0.97 0.97 73209

macro avg 0.96 0.95 0.96 73209

weighted avg 0.97 0.97 0.97 73209

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 12,002 | 1039 |
| True Paid Off | 807 | 59,361 |



Gradient Boosting Classifier:

precision recall f1-score support

0 0.90 0.87 0.89 13041

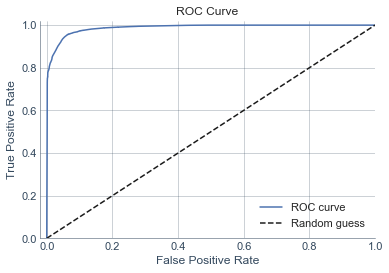
1 0.97 0.98 0.98 60168

micro avg 0.96 0.96 0.96 73209

macro avg 0.94 0.93 0.93 73209

weighted avg 0.96 0.96 0.96 73209

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 11,365 | 1,676 |
| True Paid Off | 1197 | 58,971 |



AdaBoost:

precision recall f1-score support

0 0.91 0.86 0.88 13041

1 0.97 0.98 0.98 60168

micro avg 0.96 0.96 0.96 73209

macro avg 0.94 0.92 0.93 73209

weighted avg 0.96 0.96 0.96 73209

|  |  |  |
| --- | --- | --- |
|  | Predicted Default | Predicted Paid Off |
| True Default | 11,162 | 1,879 |
| True Paid Off | 1,146 | 59,022 |

