Prediction of Bad Loans and Interest rates for Peer-to-Peer Lending

Domain Background

Peer-to-peer lending (P2P lending) is the practice of lending money to individuals or businesses through the any online platform which matches lenders with borrowers (https://en.wikipedia.org/wiki/Peer-to-peer_lending). P2P lending has been around for quite a while. While their volume is substantially low compared to standard bank-based loans, P2P lending has been growing at a very fast pace. From 2012 to 2016 it rose 15 times to $77 billion in 2016. The expectation in US is that this market will grow to reach $150 billion by 2026 (https://www.debt.org/credit/solutions/peer-lending/). Some of the major players in this field are Lending Club (https://www.lendingclub.com/), Prosper (https://www.prosper.com/) and SoFi (https://www.sofi.com/).

These loans require substantially less paper work, and the interest rates are lower. The lenders also benefit as the loan yields higher returns than traditional bank or bonds. It's a win-win situation for both the borrower and the lender.

I personally got involved in the Loans and lending when I was searching for our first house last year. And more so in the last month, when I was applying for a car loan. For my car loan, I applied to a P2P lending company also. I got the same rate as big national banks and hence, I didn't go through the P2P lending. But, my curiosity didn't die as to how do financial institutions judge your ability to repay a loan. This project is partly to cater my curiosity and partly for the fulfillment of the Udacity Nanodegree.

Problem Statement

When it comes to money lending business there is no such thing as risk-free lending. In the US, P2P lending is treated as investment and hence repayment in case of a default is not guaranteed by the Federal Govt. Despite the credit checks and other kinds of checks, the default rate for such loans has been high. In 2014, Lending Club's default rate was 8.7% and Prosper's was 3.6% (https://investorjunkie.com/9328/lending-club-vs-prosper/). In the first quarter of 2017, in a survey, 17% of US consumers said they were likely to default on a loan payment in the next year (http://www.financialexpress.com/economy/bad-loans-problem-in-us-trumps-america-is-facing-a-13-trillion-consumer-debt-hangover-a-new-record-high/704363/). Hence, the problem of loan defaulting is still very high, and it puts the individual lenders at risk of losing their money.

In terms of machine learning jargon this is a classification problem. The loans are to be classified as good loans and bad loans, where bad loan means it will be defaulted. In addition, I intend to do regression analysis to predict the interest rate of good loans.

Datasets and Inputs

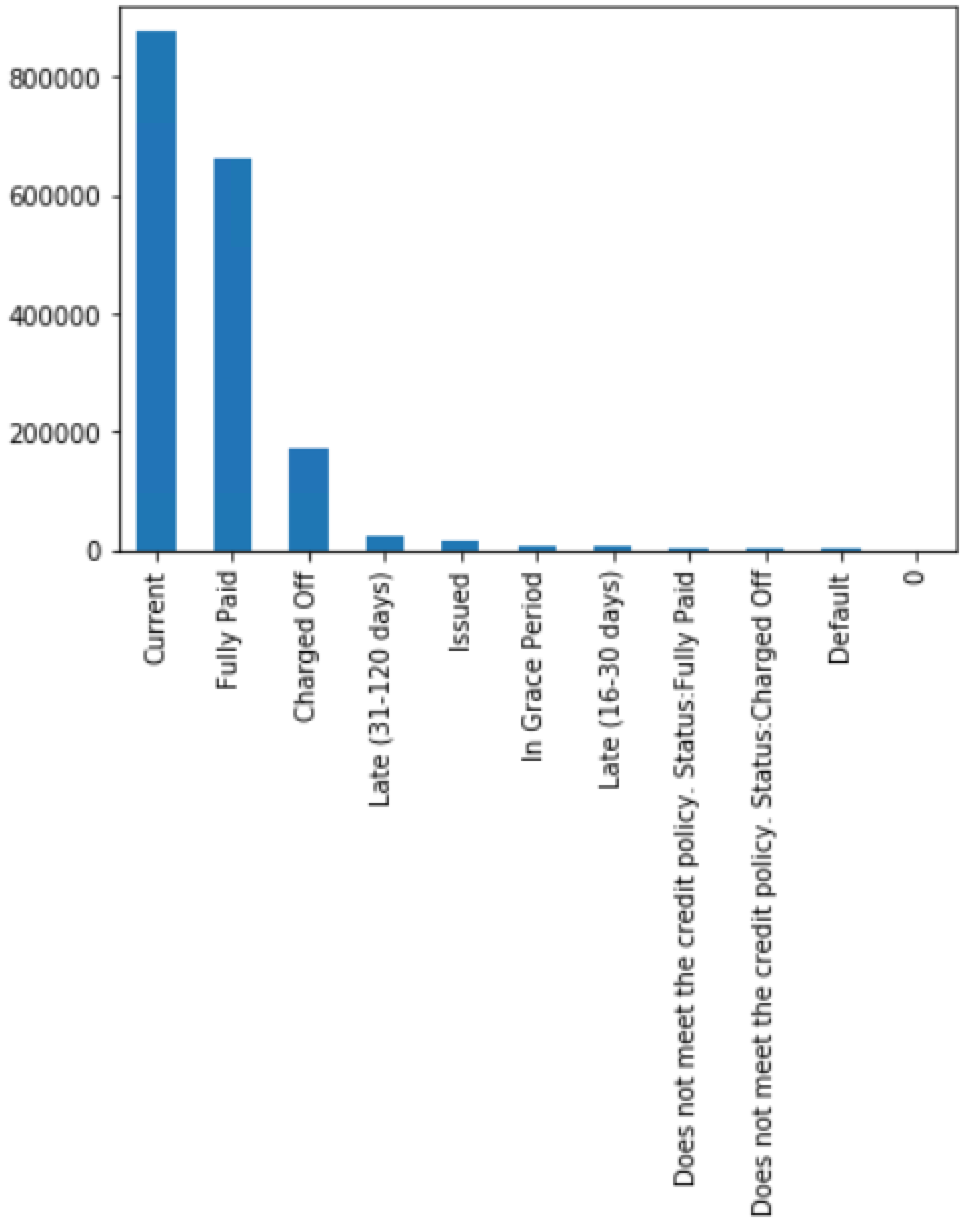
Lending club has shared their loan data from the year 2012 to 2017 at this webpage: https://www.lendingclub.com/info/download-data.action. For each loan, it gives the financial, employment, credit, demographic, bankruptcy and other decision-making information in a CSV format. https://resources.lendingclub.com/LCDataDictionary.xlsx gives the list of all the data attributes used. It then gives the interest rate which the loan was granted. And it also mentions if the loan was defaulted. In addition to the complete list of loans granted it has another set of CSV files which lists all the loans denied.

This is data is almost like gold for any P2P lending business. This data can be used for predicting loan interest rate by running machine learning training on the existing data. Similarly, it can be used to predict if a loan will result in default or not.

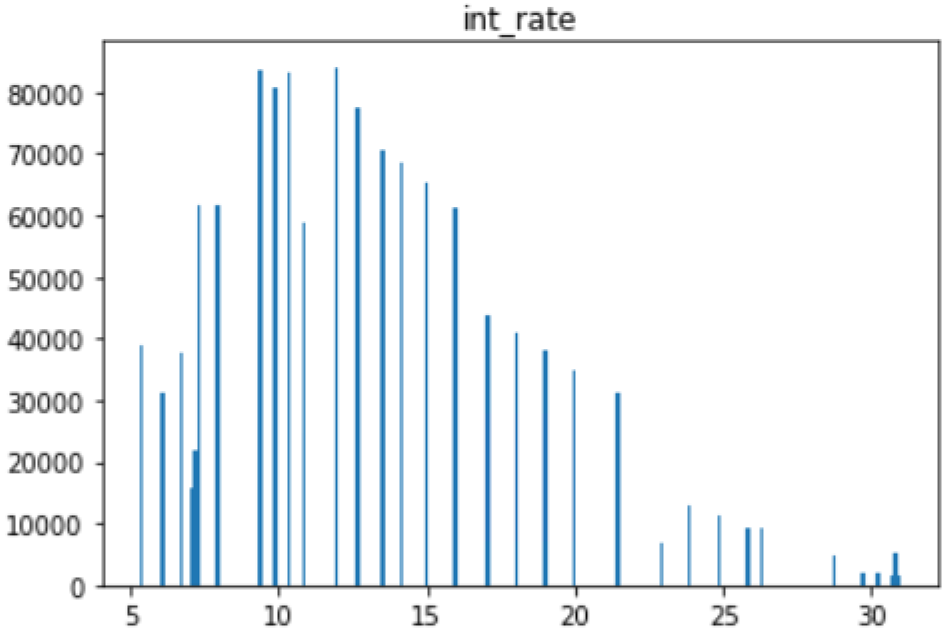
It must be noted that the data is not fully curated and clean. There are a lot of missing entries in the data. Appropriate methods would have to be used to fill those entries. The data is either numbers or text based on what kind of attribute is being described.

I also plan to augment the data. One example is the to see how close is the date to the nearest holiday like July 4th, or Labor day. This will give an idea about people's loan behavior closer to holidays. I also plan to use some NLP techniques to understand loan request description. This will give further insights to the loan will help in better prediction.

For predicting the loan as good or bad, the target variable to be trained for is "loan_status". It can take the following values: "Issued", "Current", "Fully Paid" and "Charged Off". "Charged Off" means the loan was defaulted. The distribution of loan_status is show below. As expected the number of loans defaulted is a relatively low number. The data is not balanced and only 9.7% loans end up in default. The plot also shows that some cleanup of data is required as there are more than 4 categories.



For the regression problem of predicting the interest, the target variable is “int_rate”. The int_rate distribution is shown below. There is nothing that stands out in this chart, which means the complication involved would be less. Note that this plot if for all the loans which includes the defaulted ones also.



Solution Statement

The solution to this problem is manifolds. First, it is important to identify bad loans. Here bad loan is any loan which is going to be defaulted before the loan term. By identifying such loans, the problem of loan defaulting can be drastically brought down, if not eliminated. On top of it, it is possible to predict the interest rate for a good loan. Sometimes by altering the loan rate it is possible to make the borrower pay it back. Hence, it is important to identify if the loan interest rate is optimal. This can be done by training on only the good loans. Thus, we can find the interest rate for good loans.

Before training for good loans, it makes sense to analyze the important features and top contributors to the good-loan classification. This will give clear indication on what factors contribute the highest towards a good loan. Also, it is important to note that there is no way to predict if the interest rate was different, the loan would not have defaulted. This information can only serve as a guidance to any financial institution.

Benchmark Model

1. Prior Work

Since, this data is freely available there has been multiple attempts to predict bad loans and interest rates in past. The following table shows a quick review of various attempts.

http://kldavenport.com/lending-club-data-analysis-revisted-with-python/	Predict interest rate.	Uses data until 2015. 76%

		accuracy in test set.
http://blog.yhat.com/posts/machine-learning-for-predicting-bad-loans.html	Predicts bad loans.	Creates an API for anybody to use. The data used is from 2011. The accuracy is not shown.
https://codyhatch.com/projects/lending-club-default-predictor.html	Predicts bad loans.	93% accuracy. Not much code is shared.
http://kldavenport.com/gradient-boosting-analysis-of-lendingclubs-data/	Simple analysis. No prediction.	Data is from 2013.
https://sites.google.com/site/2015pcsu/data-science/predicting-bad-loans-using-lendingclub-com-data	Predict bad loans.	Very limited analysis.

2. Benchmark

For benchmarking purpose, I will use Naïve Bayes classifier for finding bad loans and Logistic Regression for predicting the interest rates.

Here, since the data is not balanced, we can choose to predict that none of the loans are bad. In that case the accuracy score is 0.90 and the F-score is 0.92. When we consider the fact that loans with status "Current" doesn't mean they are good loan, the accuracy would further come down. Considering only "Fully Paid" and "Charged Off" loans for our calculation, the new accuracy score is 0.79 and F-score is 0.82.

Evaluation Metrics

The original data is divided into three parts – Train, Validate and Test data. The model is trained on the training set. Then we run the trained weights on the validation set. The train+validation is run multiple times until the validation accuracy over the validation set is remaining a constant. That is when it means that the model has learnt the best weights and there is no overfitting. At this point those weights are tried on the test dataset. Since, the model has never seen test dataset, the test data works like new real world data for the algorithm.

In this case, I plan to use 60% of the data for training, 20% for validation and the rest of the 20% for training. Not that, I will use cross-validation method where the train+validation set (a total of 80%) is used for both training and validation. There is no separate 20% of data separated for validation. Instead four different chunks of 20% of data are used for validation so that each data in the 80% is used for both training and validation.

1. Classifier evaluation

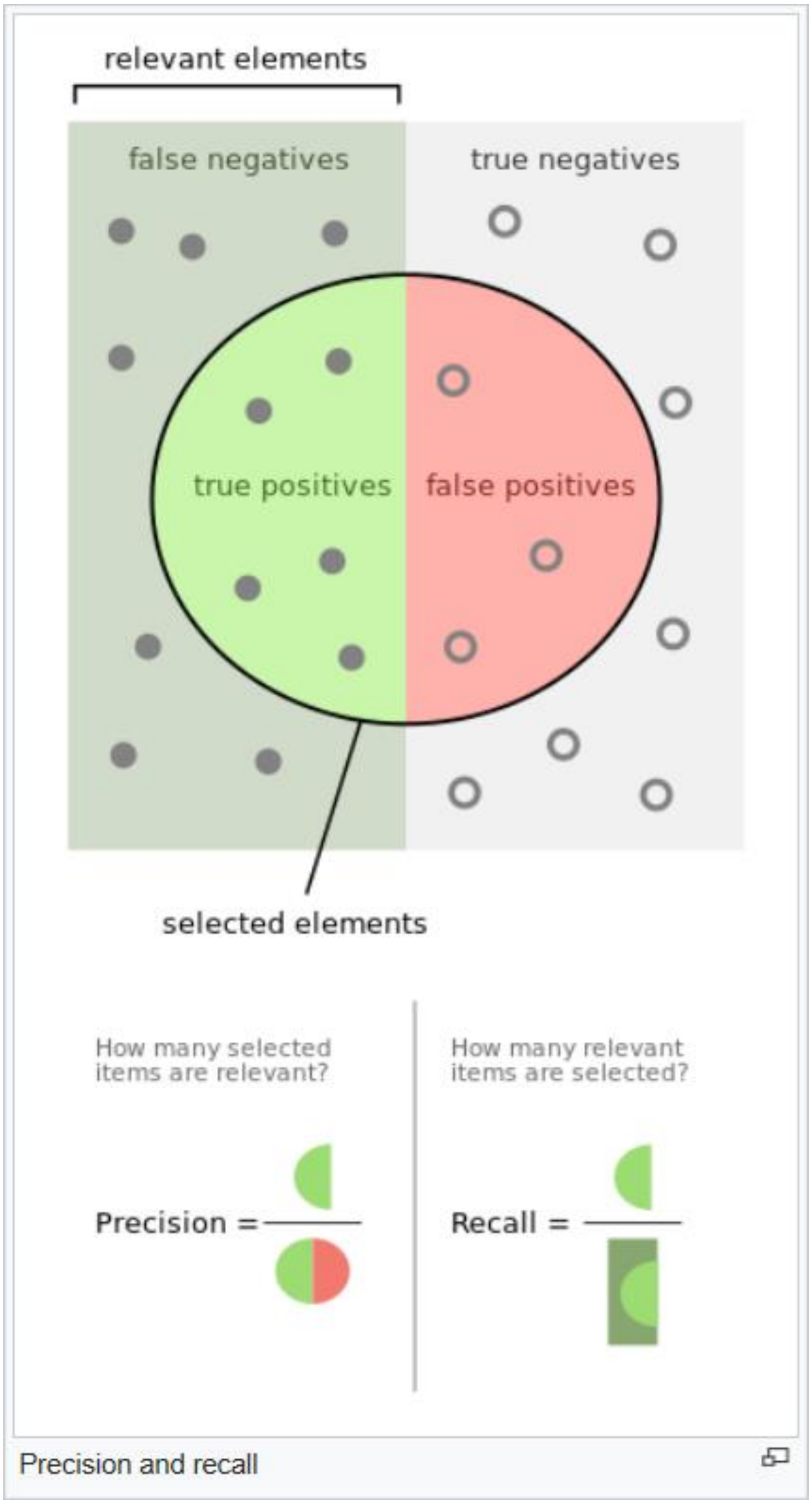
The metrics I will use for measuring the goodness of my model are the accuracy score and F1-scores. F1-score is a measure for the tests accuracy. It considers the precision and the recall.

Precision is the measure of how many of the selected items are relevant. In this case, if my predictor predicts 1000 bad loans, precision of is the measure of how many are bad loans in that 1000.

And Recall is the measure of how many relevant items are selected. In this case, if there were total 5000 loans, and there were 1200 bad loans, recall is the measure of how many of the 1200 did the predictor predict correctly.

The following snippet from Wikipedia page (https://en.wikipedia.org/wiki/Precision_and_recall) shows it pictorially.

Again, borrowing from Wikipedia (https://en.wikipedia.org/wiki/F1_score), in its simplest form, the F1-score is the harmonic mean of the Precision and Recall.



Precision and recall

Again, borrowing from Wikipedia (https://en.wikipedia.org/wiki/F1_score), in its simplest form, the F1-score is the harmonic mean of the Precision and Recall.

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Referring to the above Pie-chart, accuracy is the fraction of true-positives out of the whole dataset.

It is desirable to have a high accuracy score and a high F1-score.

2. **Regressor evaluation**

For the regression model, I will use the root-mean-square-error (RMSE). It measures the absolute different between each prediction and its real value. In math, it the following:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}}.$$

(RMSD is root-mean-square-deviation, which is same as RMSE).

$\hat{y}_t$ is the predicted value and y_t is the real value.

The goal is to minimize the RMSE value.

Project Design

The following is the workflow for this project.

1. Cleanup the data.
    a. Removed redundant columns like member_id as they don't provide any information about the loan.
    b. Fill all the empty cells with 0s or average values or by any other means.
    c. Make sure the datatypes are all correct.
        i. Sometimes numeric column can be read as strings (objects). For example, time period can be represented as "36 months". They need to be made 36. Interest rates (3.2%) will be read as strings. Zip codes are in the form of 951XX. They need to be made categorical data.
2. One hot encoding: For data which has finite number of values (but described as text), one hot encoding can be applied. If not, use them as categorical data.
3. Text data: Text data must be dealt with in one of the below methods.
    a. Omit it. Sometimes text data is too much varied that it is better to omit that column.

   b. Tokenization: Parse the text to quantitatively described the contents. The function available in SciKit Learn are: CountVectorizer, TfidfVectorizer and HashingVectorizer.
4. Data transformation: Apply one or more of the following methods based on the data. Use histogram or any other method to find the data distribution.
   a. Scaling: Scale the data up or down to bring all the data in a similar range. Taking logarithm of the data is often used (depending on the distribution).
   b. Normalization: Most of the machine learning algorithms work well with normalized data.
5. Feature reduction: To reduce the number of dimensions, to faster computing, use one or multiple of the following to reduce the features
   a. Correlation matrix: Identify the features which are too closely related.
   b. PCA: Use PCA to reduce the number of features.
6. Create train, validate and test data.
7. Apply classification for learning the “bad loans”.
   a. Start with SGD classifier
   b. Followed by kernel approximation
   c. Try XBBoost also
8. Apply regression to predict the interest rate.
   a. Start with SVR with linear kernel
   b. Followed by Ensemble Regressors
   c. Try SVR with “rbf” kernel also
   d. In the end try XGBoost
9. Run the regression / classification prediction on data from second half of 2017.
   a. Use confusion matrix on the actual vs predicted class in classification problem.
   b. Use coefficient of determination (R^2) to measure the regression problem.
10. Once, the above classification and regression are successful, identify the top contributors to the results and understand the relationship between those features and the labels.