06:00
Save!
LendingClub
LendingClub screens borrowers, facilitates the transaction, and services the loans
Earn More!
$100
Borrowers earn savings and financial control
Investors earn solid returns

Lending Club Loan Data Analysis

Group 1

Benjamin Allen beallen@syr.edu

Harper He xhe128@syr.edu

Abhiram Gopal agopal@syr.edu

Table of Contents

Abstract

The Lending Club is an online peer-to-peer lending company that offers people the ability to take out and give loans to their peers. The Lending Club published the 2007-2018 subset of its lending data a few years ago. This data is available on Kaggle, and it is what we will be using for our final project. This dataset consists of 145 interrelated columns, and roughly 2.26 million rows. The fields are different features cases of loan applications, and related variables. Some examples of fields include the loan_amnt (amount of money requested by the borrower), the funded_amnt (total amount committed to that loan at that point in time), and the funded_amnt_inv (total amount committed by investors for that loan at that point in time).

A critical business risk for the Lending Club is borrowers failing to pay their loans. Our team wishes to understand what factors are related or cause the bad loan. We would also like to predict if a loan would be charged off or not based on the features of each loan. After collecting the data, our team will first prepare the data, including coping with missing values, removing variables that are irrelevant to our question, winsorizing and standardizing numeric variables, encoding categorical variables. We will then explore the data using visualization techniques. To further reduce the number of variables, feature selection methods will be applied. We will compute a correlation matrix for all the numeric variables to drop the variables that contain similar information. After the correlation analysis, we will build decision trees using random forest, with the loan status (charged off or not) as the target variable. We are particularly interested in using some of the following methods to predict if a loan will be charged off or not:

- Logistic Regression
- Random Forest
- Gradient Boosting Trees

Finally, we will evaluate and compare the performance of our models. We will use AUC, Recall Precision and F1 score as metrics to evaluate our models. To compare the models, we will use "Recall" in terms of our business problem because it shows the ability of the model to find all cases where the loan is Charged Off. In this way, investors and Lending Club can detect the risky behaviors in the early stages and conduct the corresponding actions to reduce the possible loss.

Introduction

P2P lending has quickly gone from a market niche to a major player in the loan industry because it offers both borrowers and lenders significant efficiencies over the traditional banking model, including easy application, reasonable bank-like interest rates, fast funding, diversified investment options.

Lending Club is the first and largest online Peer-to-Peer (“P2P”) lending platform that enables borrowers to obtain a loan, and investors to purchase notes backed by payments made on loans. Lending Club provides the "bridge" between investors and borrowers so that individuals can lend and borrow money directly from each other.

Although peer to peer lending can look mighty tempting for investors, it is exposed to high credit risks. Many borrowers who apply for P2P loans possess low credit ratings that do not allow them to obtain a conventional loan from a bank. That risk is even greater because the loans are generally unsecured, so there is no collateral to go after in the event of default. Therefore, an investor should be aware of the default probability of his/her borrowers. It is also Lending club’s responsibility to screen borrowers’ information to make sure that only qualified applicants can receive loan offers (Writer, 2019).

Objective

In this project, we will attempt to predict if a loan will be charged off or not to help deter unprofitable investments in high-risk notes. A loan becomes “Charged Off” when there is no longer a reasonable expectation of further payments. Different from default, which are loans for which borrowers have failed to make payments for an extended period, having a loan charged off is one of the worst cases. A charged off loan will result in a loss of both principal and interest for investors and harm the reputation and credibility of Lending Club. Our team is devoted to examining and finding some of the relevant factors to predict if the current loans would be charged off to help avoid loss.

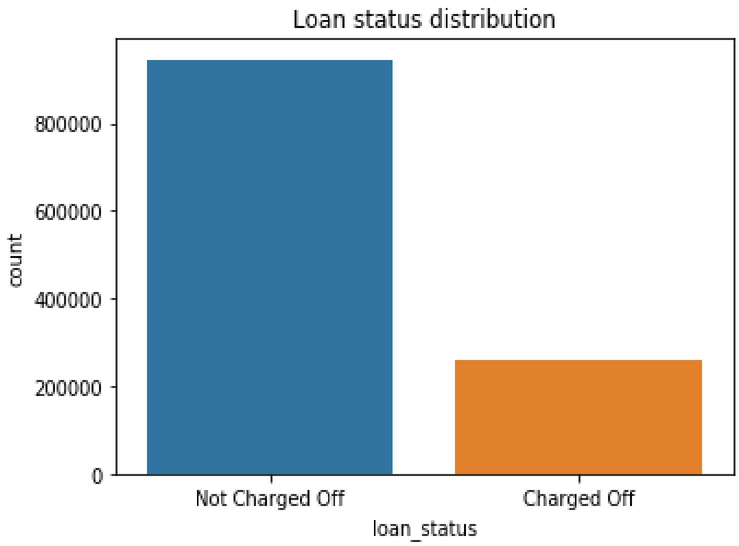
Data

The Lending Club loan data contains complete loan data for all loans issued through the 2007-2018, including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. The Lending Club dataset contains about 2.26 million loans and 145 features for each loan originated. Most of the variables fall into one of four categories – borrower's demographic information, borrower's past credit history, latest payment information and current loan information. Loan data contains variables such as the loan amount, interest rate, payment term and loan purpose. Borrower's demographic information has employment history, income, home ownership and so on. Borrower's past credit history has number of credit lines, missed payments, debt, credit verifications and criminal records.

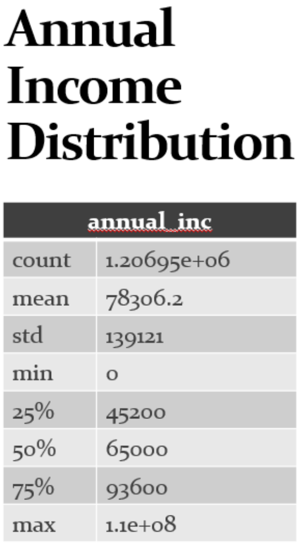
The target variable in our dataset is 'loan_status' which shows the status of the loan. It has 9 different values – 'Fully Paid', 'Default', 'Charged Off', 'Late', 'Current' and so on. Since our project goal is to predict whether a borrower will charge off the current loan, we are considering this observation. Then we grouped the 'Current', 'Late' and 'Default' observations as "Not Charged Off" class.

Descriptive Analysis

In this section, we present some graphical representations of the Lending Club dataset. The graphics help us to gain insight about the features.

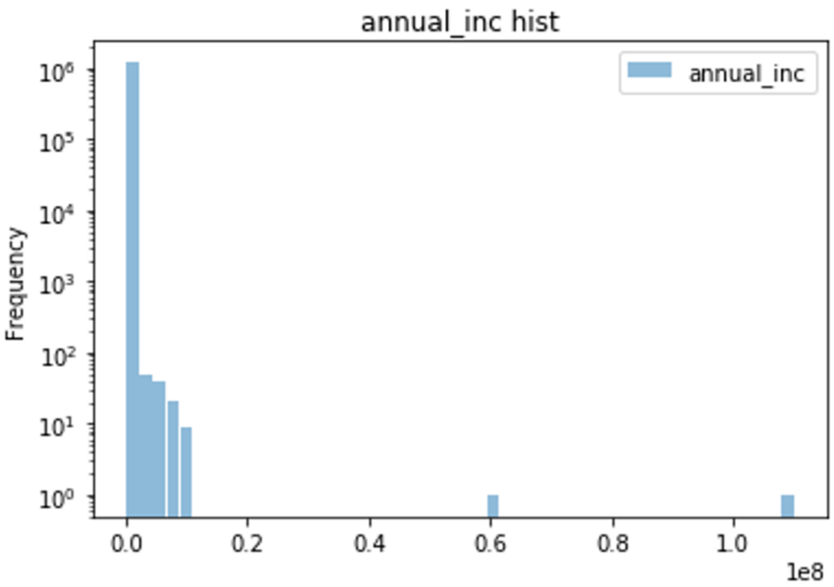


The bar graph shows the count of “Charged Off” and “Not Charged Off” classes. It illustrates that loan_status is imbalanced. The existence of this class imbalance is problematic for classification models as they tend to become bias to the majority class, and hence resulting in the model overfitting.



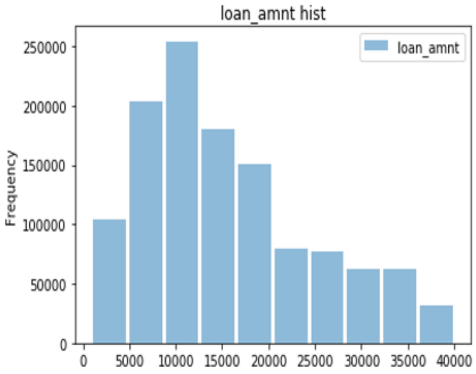
Annual Income Distribution

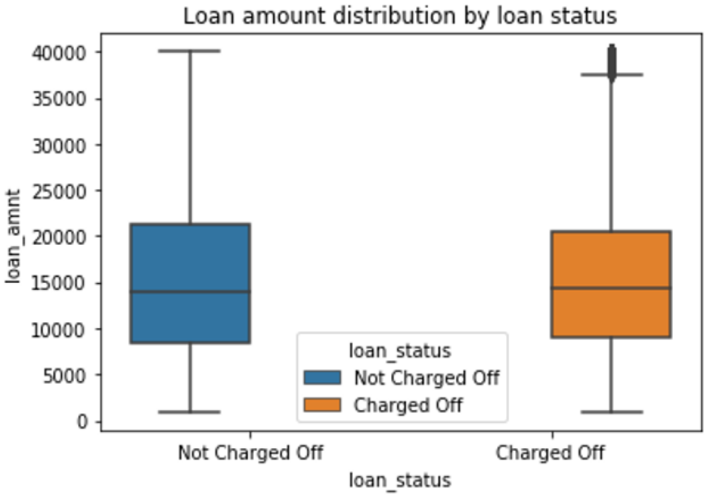
	annual_inc
count	1.20695e+06
mean	78306.2
std	139121
min	0
25%	45200
50%	65000
75%	93600
max	1.1e+08



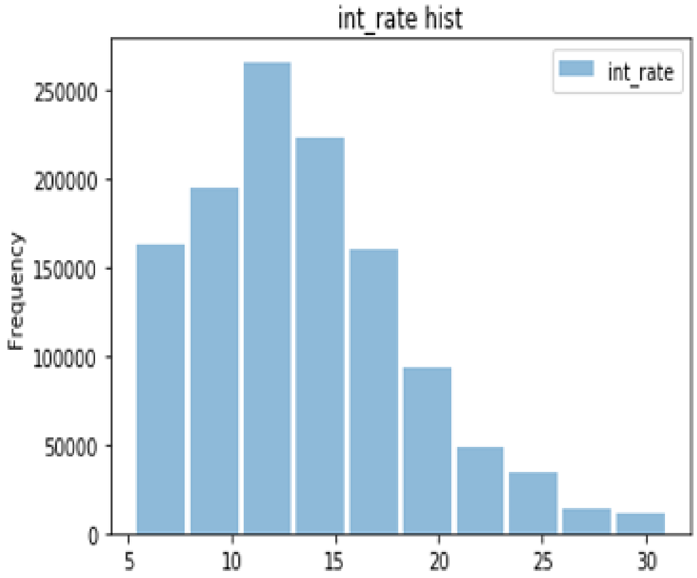
This chart shows the Annual Income of the borrowers. It is right skewed. There is a huge gap between high income groups and others. Though 75% people make less than 100,000 dollars each year, there are some borrowers who earn millions each year.

Loan Amount Comparison



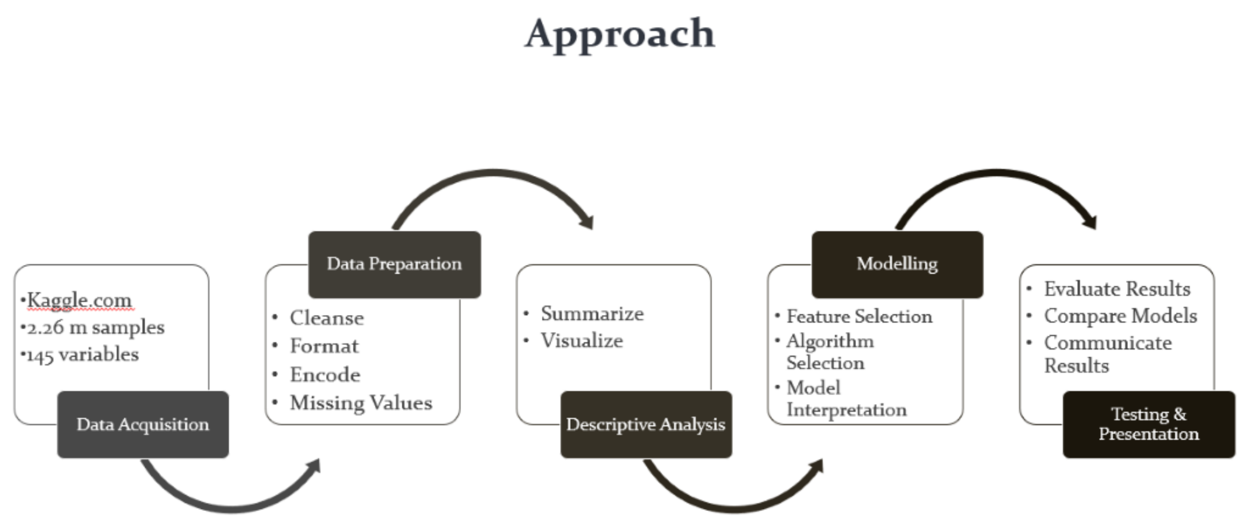


Loan amount represents the amount in dollars the borrower received as a loan through Lending Club. This feature has a range from 900 dollars to 40,000 dollars. The value peaks at 10,000 dollars. From the box plot we can see that there are some outliers for the loan amount when the observation is grouped as Charged Off. One of assumptions before we started is that the loan amount between ‘Charged Off’ and ‘Not Charged Off’ might be different. However, the boxplot shows that there is no significant difference between these two classes, so the loan amount might not be a good indicator to predict if a loan will be charged off.



Interest rate is the rate agreed by the borrower to pay on the principal amount which is the loan amount. This histogram shows the most common interest rate is 10% to 15%, followed by 8% to 10%. Personal loans at commercial banks range from 9.5% to 12.5%. This is interesting to compare because it appears Lending Club adjusts their interest rates for loans which may be an important business criterion in their model.

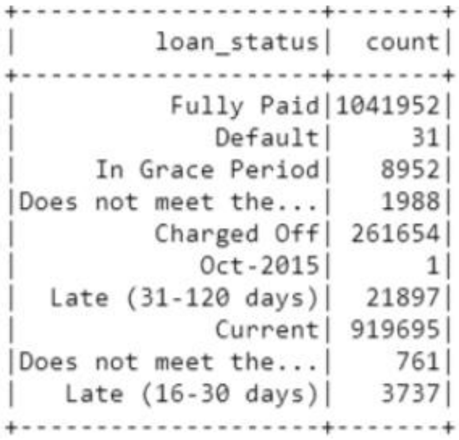
Methodology



Business Question Explained

Borrowers are listed on the Lending club platform and investors can see all the details about the borrowers before lending money to them. It is Lending club's responsibility to screen borrowers, facilitate the transaction with appropriate interest rates, and service the loan. Only qualified applicants can receive loan offers. However, there is always a risk of the borrowers failing to pay the loan. We are planning on understanding and constructing a method for

which Lending Club can assess potential customers to give them loans.



```
+--------------------+-------+
|         loan_status|  count|
+--------------------+-------+
|          Fully Paid|1041952|
|             Default|     31|
|     In Grace Period|   8952|
|Does not meet the...|   1988|
|         Charged Off| 261654|
|             Oct-2015|      1|
|  Late (31-120 days)|  21897|
|             Current| 919695|
|Does not meet the...|    761|
|   Late (16-30 days)|   3737|
+--------------------+-------+
```

From here we knew that our intent was to run a classification algorithm that would only consider the difference between two categories. Thus, we categorize the loans as either “Charged Off” or “Not Charged Off”. To do this we eliminated the records that were “Fully Paid”, “Does not meet...”, “Oct... “, “Default”, and “In grace period”. Those that were “Late...” were relabeled as “Not Charged Off” and the others went into “Charged Off”.

Data Wrangling

Our first point of interest was to check our data for missing values. We ran code to calculate the amount of missing data by each column. Being that our dataset has millions of records our threshold for removing columns that had missing data was 50%. Upon doing this we removed 40 columns. Our objective from here was to fill in remaining missing values.

We evenly split the columns up between ourselves. The standard we went about filling numeric columns was filling in missing values with the mean. For categorical missing values we use the value that appeared the most (mode) in the column to fill in with. Now, there are exceptions to some columns that needed to have their values filled.

In processing summary statistics, we found some columns had skewed data points.

To handle this, we used a method of winsorizing the data to fill with the mean. Winsorized mean is a method of averaging that initially replaces the smallest and largest values with the observations closest to them. This is done to limit the effect of abnormal extreme values, or outliers, on the calculation (Hargrave, 2020). We only had to fix large values.

There were also some columns that had values that were unlike the majority data type for that column. We converted the data types for these respective columns and by default the values that could not convert data type were identified to be incorrectly inputted, thus they were filled with the appropriate value. This resulted in a data frame of 1.2 million observations and 72 variables.

Preprocessing Data

Next, with a clean dataset, for us to run a classification algorithm we put our columns into a vectorized format and represented categorical features as a numerical input. The four tools used to do this were Standard Scaler, String Indexer, One Hot Encoder, and Vector Assembler.

The main idea of standard scaler is to make each column have a mean equal to zero and standard deviation of one. The String Indexer has the purpose of handling categorical columns that are ordinal and giving them a numerical representation. Using string indexer in conjunction with one hot encoder maps a categorical feature, represented as a label index, to a binary vector with at most a single one-value indicating the presence of a specific feature value from among the set of all feature values. Simply put, categories that are not ordinal in nature are transformed into a numerical representation. Finally, the last stage is to use the vector assembler. This is a transformer that combines a given list of columns into a single vector column. It is useful for combining raw features and features generated by different feature transformers into a single feature vector, in order to train machine learning models. In layman's term, it takes all the columns values in a data frame and transforms them into one column of features that will be used in the algorithm.

About 78% of people pay their loans while 22% of people failed to pay their loans. Standard classifier algorithms like Decision Tree and Logistic Regression have a bias towards majority class. Thus, there is a high probability of misclassification of the minority class as compared to the majority class. There are two main approaches to address the imbalance data. One is data level approach using resampling techniques. For example, we can increase the frequency of the minority class using over-sampling or decrease the frequency of the majority class using under-sampling in order to obtain approximately the same number of instances for both the classes.

The other is an algorithm level approach. Some existing classification algorithms using bagging and boosting are appropriate for imbalanced data sets. So, we used Gradient Boosting Trees which combine weak learners to create a strong learner that can make accurate predictions as one of our classifiers. Relevant evaluation parameters that can give us more insight into the accuracy of the model than traditional classification accuracy was also considered during the model evaluation and comparison, such as Recall, Precision and F-1 score.

Feature Selection

As described, we removed variables with more than 50% missing values. Also, we removed variables that were highly related at a threshold of 75%. Another useful step was understanding the business context of the variables in our dataset, so we removed unhelpful variables based on our business understanding. Even after applying these steps to reduce the dimensionality of the data, we had 72 variables present in our model. In order to get a smaller set, we ran our Random Forest model and extracted the feature importance. We then choose to use the top 20 features for the rest of our modeling since these variables captured a significant amount of variance in the data, thus improving run time and overfitting to some degree. We believe choosing random forest to get the most important features to run our models was an intriguing strategy because the algorithm chooses features that have little-to-no correlation. It also handles a big number of data fields well and is ideal for classification problems.

Models and Hyper Parameters

For the models, we choose logistic regression, random forest and gradient boosting since this is a binary classification problem. Logistic regression models have a certain fixed number of parameters that depend on the number of input features, and they output a categorical prediction. Our problem used a decision boundary to differentiate probabilities into “Not Charged Off” class and “Charged Off” class. It did this by computing the sigmoid function of X, thus we get a probability of an observation belonging to one of the two categories. Using the elastic net param we mixed alpha = 0, the penalty is an L2 penalty. For alpha = 1, it is an L1 penalty. This caused our coefficients to have different values. Using the regParam gave the algorithm additional information in order to prevent overfitting.

The Random Forest model consists of many individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. Uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction (Yiu, 2019).

For tuning parameters, we used max depth and max bins. MaxDepth is the depth of the tree meaning length of tree you desire. Larger tree helps us convey more info whereas smaller tree gives less precise info. So, depth was set large enough to split each node our desired number of observations. MaxBins is the number of bins used when discretizing continuous features. Increasing maxBins allows the algorithm to consider more split candidates and make fine-grained split decisions. However, it also increases computation and communication. Since we had a lot of categories for some features, we had to set this to be at least the maximum number of categories.

The max depth and max bins tuning parameters also apply to our final algorithm which is Gradient Boosting. Gradient Boosting trains many models in a gradual, additive and sequential manner. Gradient Boosting Algorithm identify the shortcomings of weak learners (e.g. decision trees) by using gradients in the loss function (*y=ax+b+e , e needs a special mention as it is the error term)*. The loss function is a measure indicating how good are model’s coefficients are at fitting the underlying data. Our goal is to classify loans as “Charged Off", so the loss function is a measure of how good our predictive model is at classifying bad loans. One of the biggest motivations of using gradient boosting is that it allows one to optimize a user specified cost function, instead of a loss function that usually offers less control and does not essentially correspond with real world applications (Singh, 2018).

Results

Model Evaluation

Our top 20 features provided by Random Forest are mentioned in this figure.

Feature	Description
revol_bal	Total credit revolving balance
pub_rec	Number of derogatory public records
open_acc_6m	Number of open trades in last 6 months
tot_cur_bal	Total current balance of all accounts
out_prncp	Remaining outstanding principal for total amount funded
total_bal_il	Total current balance of all installment accounts
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
mths_since_rcnt_il	Months since most recent installment accounts opened
last_pymnt_amnt	Last total payment amount received
open_il_24m	Number of installment accounts opened in past 24 months
total_acc	The total number of credit lines currently in the borrower's credit file
open_il_12m	Number of installment accounts opened in past 12 months
collection_recovery_fee	post charge off collection fee
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
bc_open_to_buy	Total open to buy on revolving bankcards.
tot_coll_amt	Total collection amounts ever owed
mo_sin_old_il_acct	Months since oldest bank installment account opened
open_rv_24m	Number of revolving trades opened in past 24 months
grade	LC assigned loan grade
open_act_il	Number of currently active installment trades

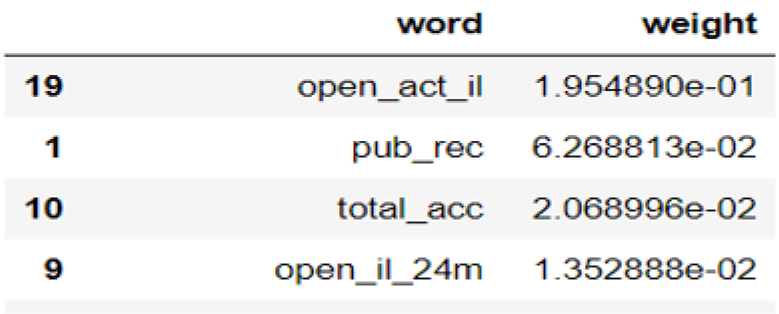
To address overfitting, we choose two main methods. The first was choosing a model with a reduced number of variables so the model could not learn on all the possible data points. This strategy improved overfitting. We contemplated using some other methods to change the parameters but realized that would not be easily attainable given the high volume of our data. We also believed our method to be enough because when we tuned our parameters for Logistic Regression using elastic net and regParam and saw a decrease in the performance of the model. Our second way for reducing overfitting was to rely on our models recall score. This evaluation metric is important because it puts more emphasis on customers who will be in the "Charged Off" group.

Scoring Metrics	**Logistic Regression**	**Random Forest**	**Gradient Boosting**
AUC	0.84	0.86	0.90
F1			
Not Charged Off	0.90	0.91	0.92
Charged Off	0.52	0.65	0.68
Recall			
Not Charged Off	0.95	0.93	0.93
Charged Off	0.42	0.61	0.65
Precision			
Not Charged Off	0.86	0.90	0.91
Charged Off	0.69	0.71	0.71

Overall, our goal was to find out how good can we predict whether a customer will be charged off. In our random forest model, you can see the recall prediction works on "Not Charged Off" very well, which means it can confidently tell you who are good customers. By contrast, it has a low recall when predicting the loan "Charge Off" behaviors. In laymen's terms, recall means how many cases are predicted correctly among all the true conditions. For the logistic regression and gradient boosting we can see that our model performs well for the "Not Charged Off" group but is not completely the best at predicting "Charged Off". We think that in a real-world scenario being able to predict whether 66% of people to be "Charged Off" will lead to a margin of profit however, continued parameters being tested or new metrics for loan evaluation may improve our model. An increase in processing power would go a long way in solving this problem but given the limitations of our personal computers, we believe our results to be useful.

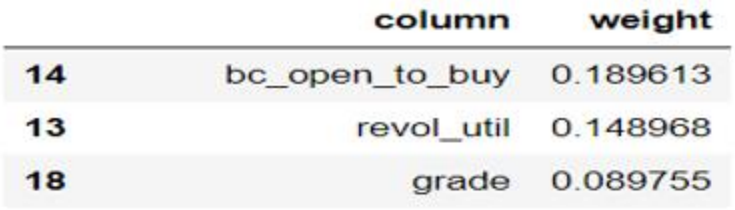
Inference

In solving for logistic regression, we thought it was important to find out how our algorithm weighted the features because we can extract usable business context (inference).



	word	weight
19	open_act_il	1.954890e-01
1	pub_rec	6.268813e-02
10	total_acc	2.068996e-02
9	open_il_24m	1.352888e-02

“open_il_24m” = Number of installment accounts opened in past 24 months. We see that “open_il_24m” = 1.35 * 10^-2; This indicates that an increase in number of installment accounts in the last 24 months is associated with an increase in the probability of default (Charged Off). To be precise, a one-unit increase in number of installment accounts is associated with an increase in the log odds of default by 0.0135 units.



	column	weight
14	bc_open_to_buy	0.189613
13	revol_util	0.148968
18	grade	0.089755

In doing gradient boosting an important feature in the GBT modelling is the variable importance. bc_open_to_buy = Total open to buy on revolving bankcards. The open-to-buy variable is information that the company accounts for in their budget to account for current and potential loans. This company metric is used for planning process and provides guidance on how much to buy, and provides benchmarks for evaluating progress, and adjusting future plans for the company. It was interesting to see this variable in the model because it gave us insight that loans are given also on what the company position to risk.

Conclusion

Outcome

Identifying the risky loans is crucial for the functioning and reputation of Lending Club. Our experiments show that the most important features to predict if a loan will be charged off or not are “Borrower’s past credit history” and “Latest payment information” ones. Attributes of Borrower’s demographic information like income and current loan information like loan amount and interest rate are not very important in the prediction of a charged off loan.

For predicting charged off loans, we established classification model using logistic regression, random forest and gradient boosting trees. Lending Club might apply the attributes identified in these models to assign the grade to borrowers and decide if a borrower is qualified for a loan. Moreover, Lending Club and investors can take advantage of the loan status prediction modeling discussed in this project to make smarter decisions when monitoring loan status. Identifying risky loans using recall as an evaluation metric can prevent financial loss.

Future Work

To address the imbalance data, we used some existing classification algorithms and appropriate evaluation metrics in this project. We would like to try out multiple methods to figure out the best-suited techniques for the dataset since there is no one-stop-solution to improve the accuracy of the prediction model. We are also interested in finding an efficient classifier and resampling strategy for the loan status prediction in social lending markets.

One area for future research may consider the principal component analysis as a feature engineering method. PCA helps us to reduce the number of variables without completely removing variables from consideration and it ensures all the variables are independent of one another. However, it will make our independent variables less interpretable than the original ones, especially in the financial field where we are not familiar with.

Additionally, we had more than 1 million observations of data, which took a lot of time for training and made it hard to do the cross-validation. We would like to implement a learning curve for our data to get the appropriate size of the dataset as some algorithms stop learning after a certain number of observations. After downsizing our data, we will be able to use cross-validation and possibly get a more accurate estimation of our models' performance.

References and Citations

Hargrave, M. (2020, January 29). How to Use the Winsorized Mean. Retrieved from https://www.investopedia.com/terms/w/winsorized_mean.asp

Kan, W. (2016, May 2). Lending Club Loan Data. Retrieved from https://www.kaggle.com/wendykan/lending-club-loan-data/version/1

MLlib - Decision Tree. (n.d.). Retrieved from https://spark.apache.org/docs/1.2.0/mllib-decision-tree.html

Peer-to-Peer Lending - Overview, How It Works, Pros & Cons. (2020, February 14). Retrieved from https://corporatefinanceinstitute.com/resources/knowledge/finance/peer-to-peer-lending/

pyspark.ml package¶. (n.d.). Retrieved from https://spark.apache.org/docs/latest/api/python/pyspark.ml.html#pyspark.ml.classification.LogisticRegression

Singh, H. (2018, November 4). Understanding Gradient Boosting Machines. Retrieved from https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab

Writer, A. B. F. S. (2019, May 2). Peer to Peer Lending - Types & Advantages. Retrieved from https://www.debt.org/credit/solutions/peer-lending/

Yiu, T. (2019, August 14). Understanding Random Forest. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2