Spring 2019

Credit Risk Analysis in Peer to Peer Lending Data set: Lending Club

Mohammad Mubasil Bokhari
Bard College, mb4740@bard.edu

Credit Risk Analysis in Peer to Peer Lending Dataset: Lending Club

Senior Project Submitted to
The Division of Science Mathematics And Computing
of Bard College

by
Mohammad Mubasil Bokhari

Annandale-on-Hudson, New York
May 2019

I would like to dedicate this project to my mother, Syeda Aisha Tahreem Gilani who dreamed to send her kid to America for education and I now have lived her dream.

Acknowledgements

I would like to thank my family and friends who have supported me over the last four years at Bard College. I would like to appreciate Manishka who has been a pillar of support ever since I was freshman. I would also like to thank my advisor Sven Anderson for his constant support and guidance. He kept pushing me, and I deeply appreciate all that he has done.

Abstract

This project studies the classification variable ‘default’ in Peer to Peer lending dataset known as Lending Club. The project improved on existing work in terms of accuracy, F-1 measure, precision, recall, and root mean squared error. We explored balancing techniques such as oversampling the minority class, undersampling the majority class, and random forests with balanced bootstraps. We also analyzed and proposed new features that improve the Learner performance.

Table of Contents

Introduction

Connor (2010) defines credit risk as to the "uncertainty about whether a counterparty will honor a financial obligation". Engle (2009) mentions the growth "in the volume and diversity" of credit derivatives over the past decade. Malik (2010) mentions the importance of credit modeling to develop a system that can correctly rank borrowers in terms of their default risk. Extensive research has been conducted but it can be synthesized to six major subfields of study:

- Default security pricing
- Default intensity modeling
- Comparative analysis of credit models
- Comparative analysis of credit markets
- Credit default swap
- Loan loss provisions.

It is important to measure credit risk, and thus researchers have developed methodologies to model credit risk. Saunders and Cornett (2011) group credit risk models into two groups: qualitative and quantitative models. Features such as reputation, financial leverage, earnings volatility, collateral, business cycle, and interest rates are employed in qualitative models. Quantitative models, however, aim at either producing a credit score, used to either determine the probability of default or classify borrowers into various default risk groups (Saunders 2011).

The rise in big data and available processing power over the past decade has resulted in the rise of implementing data-driven learning methods. Machine Learning (ML) has become a

vital part of credit risk modeling applications (Bacham 2017). Bacham mentions the reasons why modern models have shifted towards a machine learning learner from "statistical learning methods" as these methods assume a formal relationship between features whereas ML methods may learn from the data without requiring any "rules-based programming". This is evident from how learners are structured. An machine learning learner discerns the relationship between the features and the target variable through approximating a mapping function.

Kruppa (2013) presents the case for the use of machine learning methods such as Random Forests (RF) to estimate individual customer credit risk. An RF is simply multiple decision trees aggregated over the same training space. The results of each decision tree is pooled through voting which results in a final prediction. Kruppa shows that RF outperforms industry standard logistic regression. Khandani (2010) also shows that bootstrapped CART trees outperformed industry standard models to classify rates of credit-card-holder delinquencies and defaults. Stefan (2015) performed an exhaustive benchmarking of 41 different ML classifiers against the industry standard Logistic Regression (LR). He showed that several classifiers predict credit risk significantly better than LR. He concludes with a recommendation of further benchmarking models against RF and states that LR can no longer serve as a benchmark for future models. RF will be discussed in greater detail in the methods section.

Stefan's (2015) results indicate that different models perform best on particular types of credit data. There exists different types of credit: mortgage, student loan, credit card, individual loans, and thus it's imperative to differentiate these markets in model development. This paper

aims to model credit risk through a quantitative approach to classify credit risk in an emerging lending market known as Peer to Peer Lending (P2P).

Peer to Peer lending is a derivative of microcredit principles and has attracted widespread popularity within the last decade (Rajdeep 2008). Financial transactions are defined as P2P lending if they bypass conventional intermediaries by directly connecting the borrower to the lender. The financial crisis of 2008 played a key role in the expansion of this market (Havrylchyk 2018) within the United States. Fig 1 shows the rise of lending platforms in recent years. The most popular lending platforms in terms of the dollar amount of loans issues are Lending Club and Prosper.

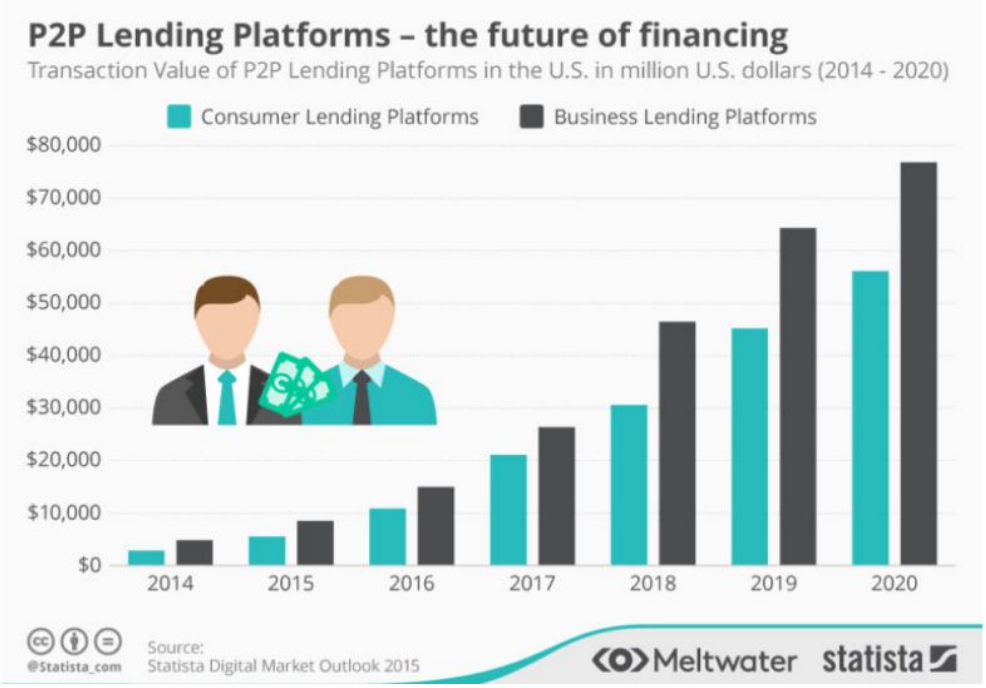

Fig 1 shows the rise of P2P lending in the last 4 year and it's projected growth to 2020 (SoFi).

The biggest P2P lending platform in the United States is an online website called "Lending Club". The platform cuts the middleman, a traditional financial institution, and connects multiple investors and potential borrowers to invest capital and to borrow credit. The borrower can put up a loan request which consists of a description of the loan purpose with their personal financial information. The investor then has the privilege to choose the amount of capital they would like to invest and also have the ability to choose the borrower. This market has its advantages and disadvantages.

The market allows borrowers who have a history of bad credit and who are faced with the option of no credit from a traditional financial intermediary or a high-interest rate loan the ability to not only receive credit but also to secure a lower interest rate. For the investor, this market provides the opportunity to receive a greater return on their capital as compared to depositing it in a savings account in a traditional bank. However, there is a great risk of the borrower defaulting and not repaying the loan with interest (Magee 2011). Therefore, there is a growing need to understand what characteristics make a borrower or loan id "bad" and enable investors to make informed decisions.

The body of literature around P2P lending has been growing ever since the formation of the first platform in 2005. Researchers have traditionally relied on the loan data provided by Prosper. Their data is structured to divide the information of borrowers into hard features and soft features. Hard features such as credit rating, loan amount, and debt to income ratio. Soft features may include information on the social network and social capital of borrowers.

Applications of ML principles in determining the default risk - the probability of a borrower to default - is in the implementation of a Neural Network [NN] with backpropagation (Zhang 2014). The NN scored an accuracy score of 78.6%. Freedman and Jin (2008) show that the credit rating of the borrowers is positively related to the success rate of loans. Fu (2017) experimented with combining tree methods such as RF with a NN. Milad (2005) explored the features such as loan grade (a score assigned by Lending Club for each borrower) and Fair Isaac Corporation scores (FICO) as indicators to default risk. Milad employed multiple learners including a cost based RF which achieved the highest 78.8% accuracy score. All these studies defined the problem of determining the risk of default as a classification problem. The classification variable has binary values: '0' as not default, '1' as default. Fig 2 shows this representation in a bar graph. A deeper look into these studies indicates that the classification variable, *loan_status* in most cases, contained an imbalance in instances. The instances in which a borrower would not default would be observed significantly greater than instances of a borrower to default. 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. This is shown by Chawla (2001).

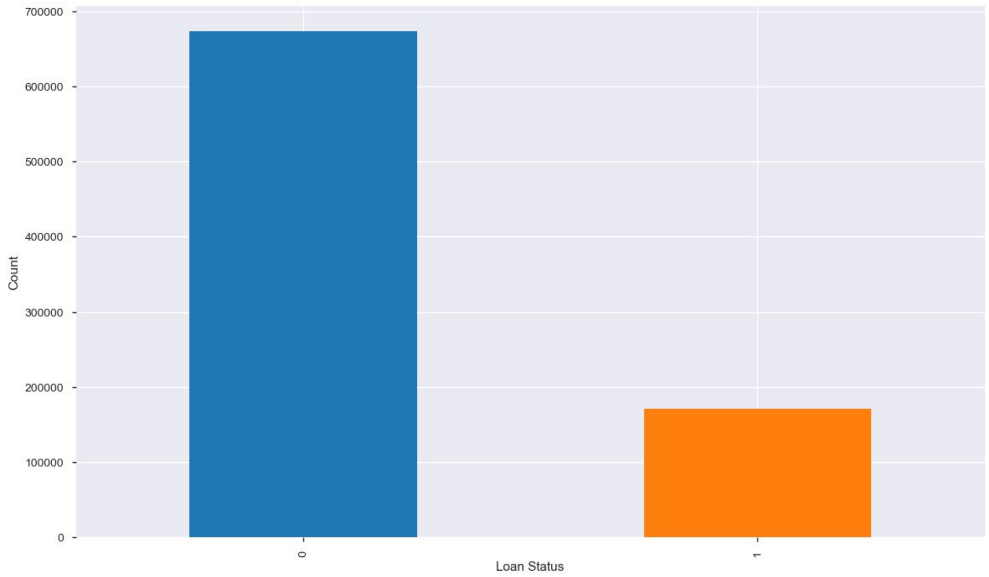

Fig 2: A bar graph showing the count of default. The graph illustrates the class imbalance in the classification variable

A framework of strategies have been proposed and adopted by researchers regarding the class imbalance problem. Kotsiantis (2006) presents two approaches currently in literature:

- Sampling techniques
  - Under-Sampling
  - Over Sampling
- Algorithmic approaches
  - Cost based models
  - Balanced Bootstrap models

The above balancing techniques are explained in the next section.

1.1 Sampling Techniques

One balancing strategy is to undersample the majority class until a desired ratio between the two classes is achieved. This technique is called Random Undersampling (RUS). This can be done either by randomly removing instances of the majority class or through some heuristic until the minority class becomes some specified percentage of the majority class. The major drawback of random undersampling is that this method can discard potentially useful data that could be important for the model. Furthermore, the goal of a machine learning classifier is to estimate the probability distribution of the target population. Since that distribution is unknown the goal is then to try to estimate the population distribution using a sample distribution. We know that a sample distribution drawn randomly may be used to estimate the population distribution. Thus, by learning the sample distribution the learner may effectively learn the target distribution. Once we perform undersampling of the majority class, however, the sample can no longer be considered random. We may, however, remove instances from the majority class that are outliers, and that are located far from the decision boundary as proposed by Kubat (1997). This ensures that we are balancing the dataset by removing instances from the majority class that shifts the learner to overfit the majority class.

Interjeet (2003) proposed an algorithm that built on top of Kubat's work called NearMiss (NM). First, the algorithm determines *n* closest majority class instances for each minority class instance and then removes the majority class instance that has the highest average distance from

the three closest minority instances. This solution ensures that only those instances are removed that are furthest from the minority class.

Similar to RUS to balance the dataset we could implement random oversampling of the minority class (ROS). This approach creates duplicates of the minority class instances. Japkowicz (2000) shows that oversampling does not significantly improve the recognition of the minority class. Work by Chawla (2001) suggests that new minority class instances can be created by interpolation. Chawla (2001) proposed an algorithm to tackle the classification problem by oversampling the majority class. His approach created synthetic instances of the minority class based on the distance between neighboring minority class instances. His approach is widely used and known as the SMOTE algorithm. The pseudocode is shown in Fig 4.

As shown by Fig 4 SMOTE over-samples the minority class by taking each minority class sample and introducing synthetic examples according to the line segments connecting any/all of the k minority class nearest neighbors. The number of k nearest neighbors is randomly chosen based upon the percentage of over-sampling required. Fig 3 provides a more intuitive explanation of the algorithm. Synthetic examples notated as x_{new} in Fig 3 and Fig 4 are calculated by first determining the difference in distance between the feature vector (sample) x_i under consideration and its *kth* nearest neighbor: this is called *dif*. The nearest neighbor is determined at random. The value of *gap* is multiplied with *dif* and then added to x_i. This approach effectively forces the decision region of the minority class to become more general.

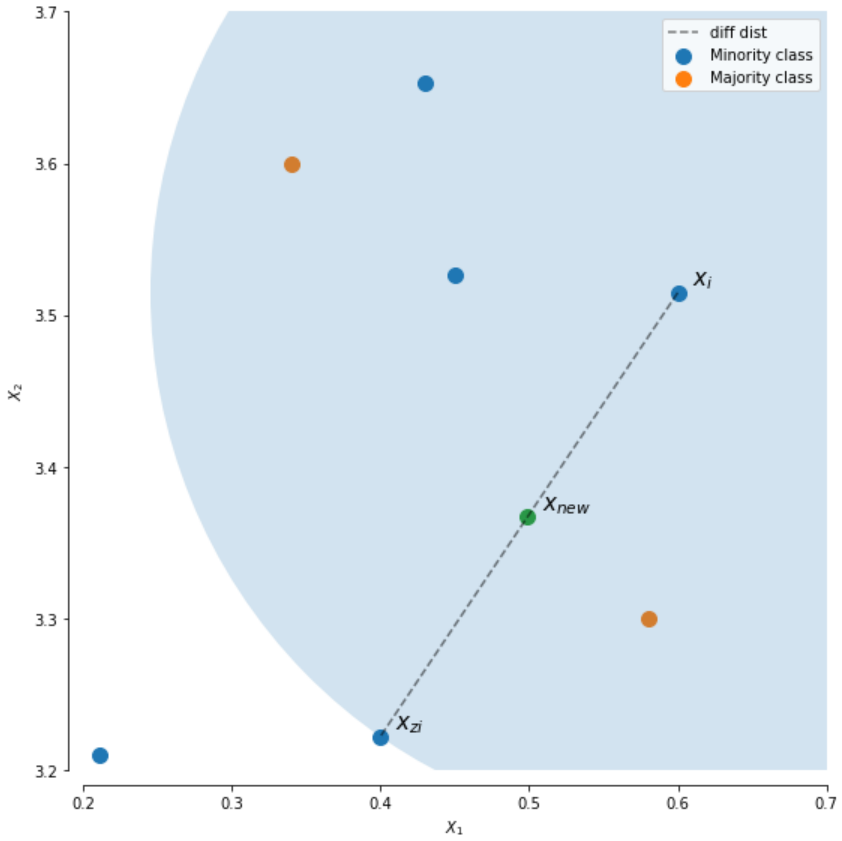

Fig 3: Demonstration of SMOTE: The line shows the distance between a minority class and its nearest neighbor. A synthetic instance of the minority class called X_{new}.

The selection of the nearest neighbor and the computation of the function dif have been further researched and different variants of SMOTE have been developed. One such variant is known as SMOTE-Borderline. Nguyen (2009) proposed SMOTE-Borderline which classifies each X_i to be one of the three:

- Noise being all nearest neighbours are form a different class from the one of X_i.
- Danger being at least half of the nearest neighbors are from the majority class than the minority class.
- Safe being all nearest neighbors are from the minority class: X_i.

The algorithm will use samples in 'danger' to generate new sample instances. The critical difference between SMOTE and SMOTE-Borderline is the instances selected as X_i. This is important as this difference affects what instances are used in creating new synthetic instances of the minority class. SMOTE picks X_i at random where SMOTE-Borderline would pick the instances on the border or referred to as 'Danger'. After selecting X_i the algorithm works identical to the original SMOTE algorithm as to how diff and X_{new} is used.

Algorithm 1: SMOTE ALGORITHM

Input
T = Number of Minority Samples ;
N = Percentage Amount of Synthetic Examples;
K = Number of Nearest Neighbors;
SMOTE(T,N,K)
for $i \leftarrow 1$ *to* T **do**
  Compute k nearest neighbours for i, and save the indices in nnarray
  Populate(N, *i*, nnarray)
end
Populate(N, i, nnarray)
while $N \neq 0$ **do**
  for *attribute* $\leftarrow$ *to numberofattributes* **do**
    $dif = X_{zi} - X_i$
    gap = random number 0-1
    $X_{new} = X_i * dif * gap$
  end
  N = N - 1
end

Fig 4: Pseudo Code for the SMOTE Algorithm

Next section we discuss the different Algorithmic approaches being studied to tackle the class imbalance problem.

1.2 Algorithmic Approaches

A learner trained on an imbalanced dataset can overcome its bias by employing two strategies: a cost function, and a balancing approach. Naively, the cost function is the error of the model. Through running the learner through a series of iterations we can optimize learners via learning from the error produced at each iteration. As the goal of learning from the error is to reduce it each iteration Learners being used for imbalanced data may increase the loss if the learner misclassified the minority class. This shifts the learner to learn better on the minority class at each iteration. The learner through this heuristic would be able to discern a mapping function with more caution as misclassifying the minority class would lead to a greater penalty. In this area of research different ways to evaluate the cost of misclassifying the minority class is developed. Within the realm of credit datasets, researchers have studied using the probability of the classification variable, and profit based models.

Another algorithmic approach as suggested by Chen (2004) is to combine sampling techniques with ensemble methods such as RF. Normally, each tree in a RF is constructed from a bootstrap sample of the training data, and thus there exists a significant probability that a bootstrap sample may contain few or even none of the minority class in an extremely imbalanced dataset. This results in RF learning poorly on the minority class. A simple solution to this problem is to use a stratified bootstrap. As noted by Chen (2004) this solution does not solve the problem. Thus, he proposed a solution to the problem and the algorithm is shown below:

- For each iteration in a random forest, draw a bootstrap sample from the minority class. Randomly draw the same number of cases, with replacement, from the majority class.
- Induce a classification tree from the data to maximum size, without pruning.

- Repeat the two steps above for the number of times desired. Aggregate the predictions of the ensemble and make the final prediction.

He compared his results to SMOTE and RUS. His model showed promise in some cases. We will be doing a similar comparison to see as mentioned in the next section.

1.3 Aims and Hypotheses

The study by Milad (2015) used the Lending Club dataset which is publicly available on their website. Their study shows that RF performs better than compared to K-Nearest Neighbor, NN, and LR. Their paper in the preprocessing section did not indicate that there exists a class imbalance in the classification variable, *loan_status.* Chawla (2004) shows that class imbalance may lead to an learner with a bias towards the majority class. The Milad study also failed to mention how they tended to the missing values in the dataset. As shown by Er (date) the treatment of missing values leads to better learners.

We hypothesize that the model implemented by Milad may be improved if we explore strategies to address the existing class imbalance. This paper will employ sampling techniques, and a balanced bootstrap ensemble approach and observe whether this improves the performance of the learner. Comparisons to Milad's learner are mentioned in Discussion and Results.

2. Methods

Fig 5 summarizes the structure of methodology adopted in this study. Fig 5 also highlights the structure of our data wrangling process. We first preprocess through a series of data wrangling steps. The processed dataset is then split into a training and testing set comprised of a 70/30 split. The training data is then fed into different sampling techniques such as oversampling the minority class and under-sampling the majority class. This results in a balanced dataset for each technique. The learner is then trained on each balanced dataset. We test the performance of each model with the testing set which has not been balanced. We also test machine learning approaches to the class imbalance problem through balanced bootstraps. The ensemble method which is built upon the multiple bootstrap samples is also then tested and evaluated using the testing dataset.

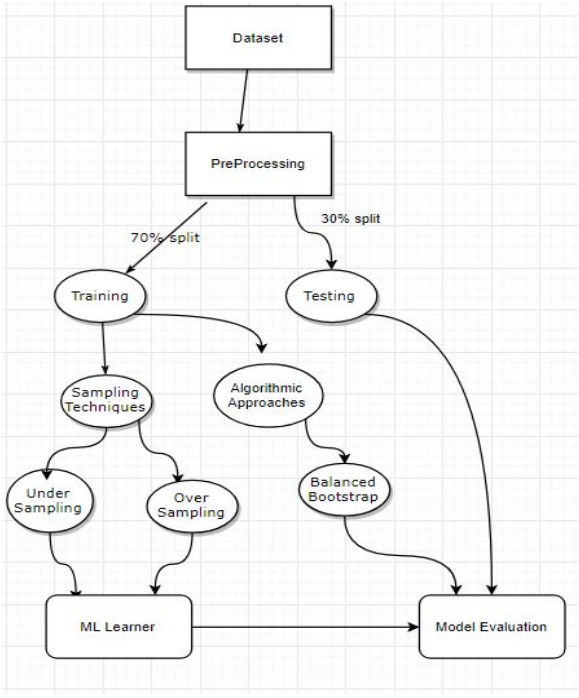

Fig 5: Illustrates the structure of the methodology adopted in this paper.

The data is processed in two ways one as carried by Milad (2015) so we can fairly compare our results. The second way would be our understanding of the dataset.

In each step of our methodology we employed the use of Python version 3.6. We choose Python because of the availability of extensive machine learning and data analysis libraries. We used Scikit-Learn, Pandas, Numpy, Imblearn, Seaborn, Jupyter-Notebook and Matplotlib in our study.

2.1 Data Preparation

This section we will describe the steps we took to preprocess the dataset. We first describe the dataset as used by Milad and then we propose our understanding of the Lending Club dataset.

2.1.1 Milad Data

We used the Lending Club dataset collected from January 2012 to September 2014 to ensure we make a fair comparison to the model proposed by Milad. The dataset is fairly big with the raw dataset containing 151 features and over 349666 observations. Following, the preprocessing steps outlined in Milad's study the data is reduced to 16 features as shown in Table 1.

Fig 6 shows the correlation of the features in Milads dataset. The heatmap shows that there are some features such as annual income and loan amount, and open account and total account that are correlated with each other. This could be because certain borrower with higher income require higher valued loans than compared to borrowers with lower annual income. Open account and total number of accounts are correlated because open account is a subset of total number of accounts. As we will mention in the next section highly correlated features may not concern our model building process.

In the next subsection, we will mention the steps we took to structure our processed dataset. We will also highlight where it differs from Milads.

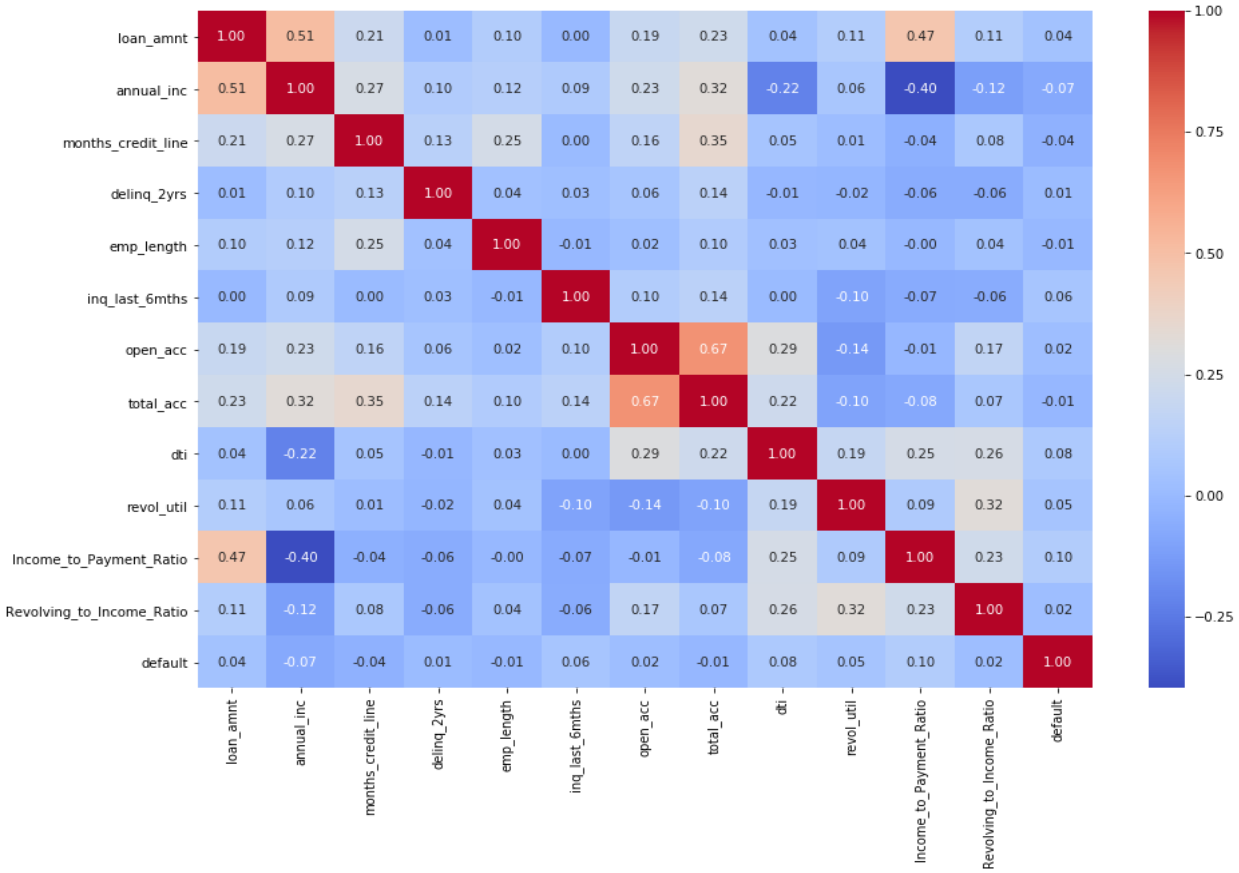

Fig 6: The correlation matrix for the Lending Club dataset as processed by Milad.

Feature Variable	**Description**
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
annual_inc	The self-reported annual income provided by the borrower during registration.
delinq_2yrs	The past-due amount owed for the accounts on which the borrower is now delinquent.
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
home_ownership	The homeownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
purpose	A category provided by the borrower for the loan request.
inq_last_6mths	The number of inquiries in the past 6 months (excluding auto and mortgage inquiries)
open_acc	Number of open trades in the last 6 months
total_acc	The total number of credit lines currently in the borrower's credit file
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
Income_to_Payment_Ratio	A ratio of the borrower's monthly income to their monthly installment.
Revolving_to_Income_Ratio	A ratio of the borrowers is the ratio of the borrowers revolving balance to monthly income.
months_credit_line	The number of months the borrower opened their first credit line from the issue date of the loan.

Table 1: Shows the features selected for the learner.

2.1.2 Our Data

Although Milad chose loans from January 2012 to September 2014, the Lending club website has data available from 2007 to the present. We preprocess our dataset using the entire dataset available on lendingclub.com. Since the purpose of our study is to improve the credit risk model proposed by Milad we structure our data to improve the default risk model. Below we summarize the preprocessing, and feature selection procedure.

We import the dataset into a pandas dataframe. The dataset contains current listings i.e. loans that are still active. These observations will be removed as we are attempting to understand why borrowers will default. The dataset also contains an extensive amount of missing values as shown in Fig 7. The figure suggests that there is a subset of features that are almost entirely missing, a set with about a quarter missing, a set with about 6% missing, and a set with no missing values.

Fig 7: This shows the percentage of missing values for the different features in the dataset.

The Lending Club dictionary provides the definition of the different features in the dataset. The dictionary gives insight as to why some features have missing values. The set of features that is almost entirely missing is not due to there being missing observations but to how the dataset is structured by Lending Club. The feature *hardship_flag* is a binary variable with values 'N' and 'Y'. These values indicate borrowers being in the hardship settlement program designed by Lending Club to help borrowers who are involved in an unexpected life event. Almost all of the borrowers are not on the hardship plan and thus have the value 'N' for their *hardship_flag*. Since there are 14 variables describing the hardship plan, and most borrowers are not on the plan these 14 features are almost entirely missing. This is one such example within the dataset that provides context to the missing values in the dataset. We drop all 15 features as they provide no relationship towards the default of a borrower.

The dataset also contains features that were not available to the investor at the time of the loan listing and was added later by Lending Club. We drop these features as the purpose of a default risk model is to determine whether a borrower would default before approving the loan. The dataset is reduced to 34 features. Table 2 shows the names and description of the features that were not included in Milad's preprocessing of the dataset. All of the numeric variables were standardized by removing the mean and scaling to unit variance. The categorical variables were converted into dummy variables as the software library Scikit-learn is not compatible with categorical variables.

To understand the relationship between the different features a correlation matrix is created in the form of a heatmap as shown in Fig 8. The figure suggests that some of the features are highly correlated with each other. Within linear models, this is a problem called multicollinearity, but since we will be employing a random forest we do not need to remove the correlated features.

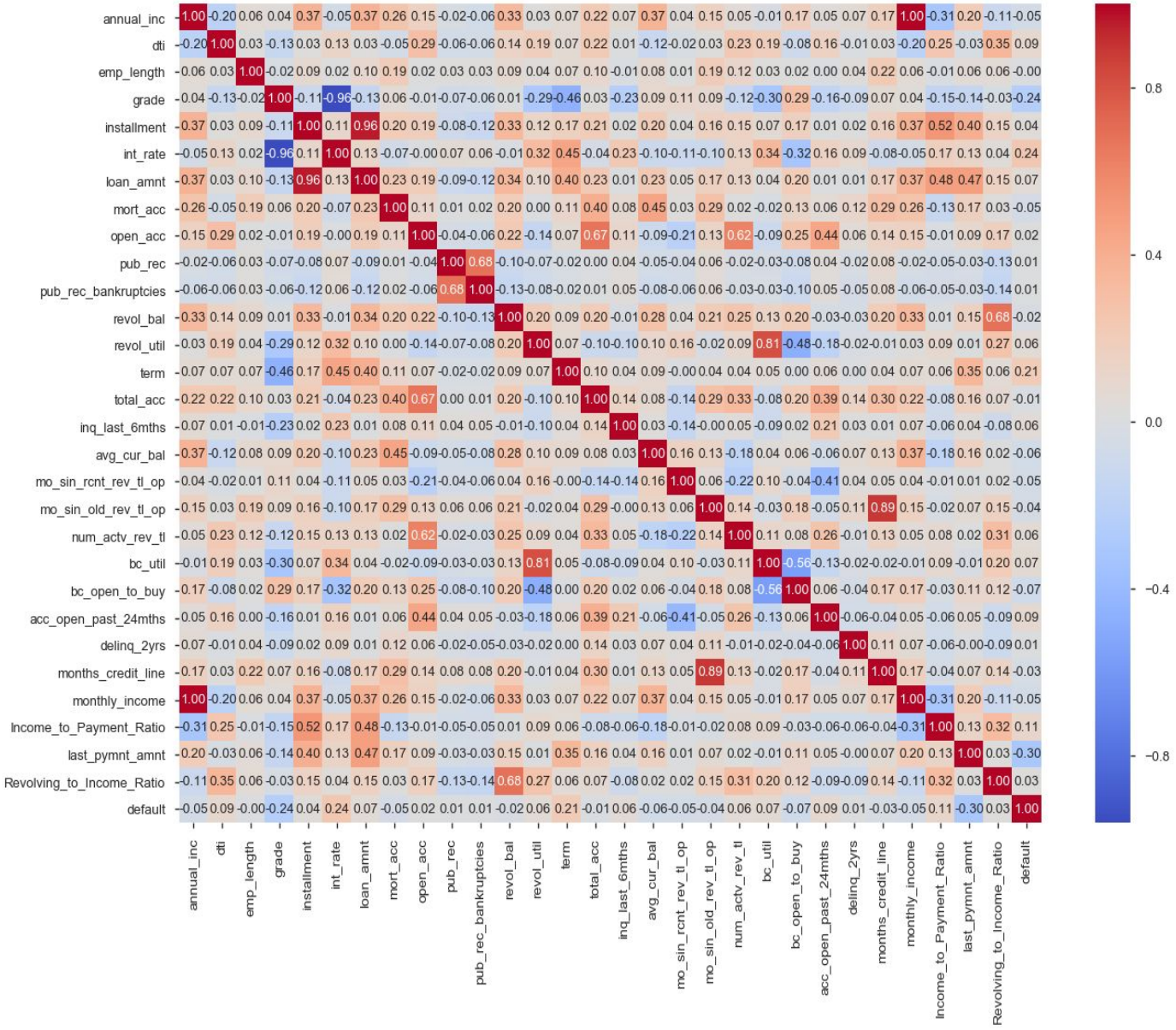

Fig 8: The correlation matrix for the Lending Club dataset as processed by us.

Feature Variable	**Description**
int_rate	Interest Rate on the loan
grade	LC assigned loan grade
acc_open_past_24mths	The number of trades opened in the past 24 months.
bc_open_to_buy	Total open to buy on revolving bankcards.
bc_util	The ratio of total current balance to high credit/credit limit for all bankcard accounts.
avg_cur_bal	The average current balance of all accounts
num_actv_rev_tl	Number of currently active revolving trades
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
mort_acc	The number of mortgage accounts.
revol_bal	Total credit revolving balance
mo_sin_old_rev_tl_op	Months since oldest revolving account opened
pub_rec	Number of derogatory public records
pub_rec_bankruptcies	Number of public record bankruptcies
last_pymnt_amnt	The last amount paid by the borrower for an account.

Table 2: Shows the additional features that will be deployed in the learner.

2.2 **Data Visualization**

In this section, we present some graphical representations of the Lending Club dataset. We hope it helps the reader to become familiar with the different features being employed in our learners.

Loan amount means the amount in U.S dollars the borrower received as a loan through Lending Club. The graph for this feature is shown in Fig 9. This feature has a range from 1200 to 40000. The value peaks at 10000 U.S dollars. From the box plot we can see that there exists some outliers for the loan amount when the feature is grouped by default. The box plot also shows that the value of loans at which borrowers default is higher than borrowers that do not default.

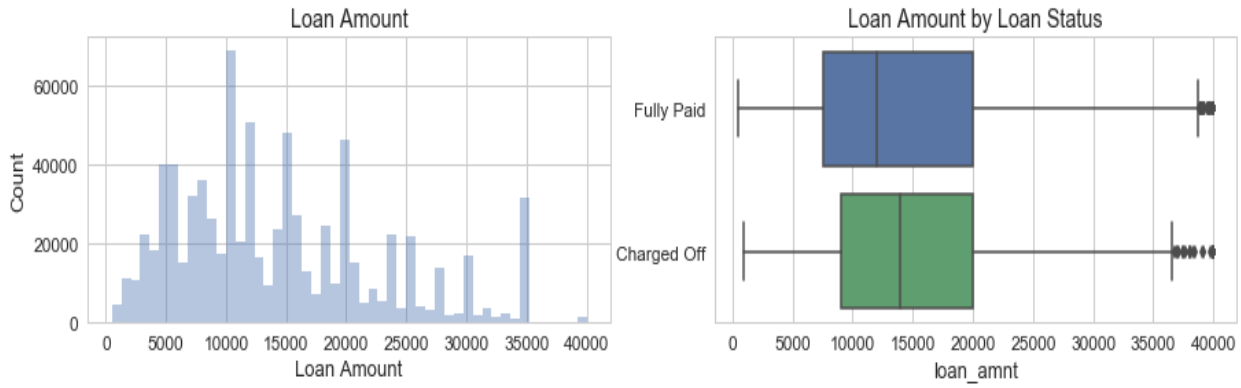

Fig 9: Illustrates the distribution of loan amount in the Lending Club dataset.

Interest rate means the rate agreed by the borrower to pay on the principal amount which is the loan amount. This rate is determined by the grade assigned by Lending Club to each borrower. The higher the grade the higher the interest rate. The heuristics used to assign these grades is unknown but Lending Club has stated they use the financial features of the borrowers in the assignment. The graph for Interest rate is shown in Fig 10, and the graph for grades is

shown in Fig 11. The boxplot shown in Fig 10 illustrates that higher interest rates lead to higher chances of the borrower defaulting. This behavior of borrowers can be correlated with the bar chart in Fig 11. The rise in grades show that the chance of a borrower would default also increases. It is interesting however that a borrower not paying their loan can occur in all grade assignments. Thus, a naive investing strategy of only investing in borrowers that have an assignment of 'A' would still in some cases observe the borrower in not being able to pay.

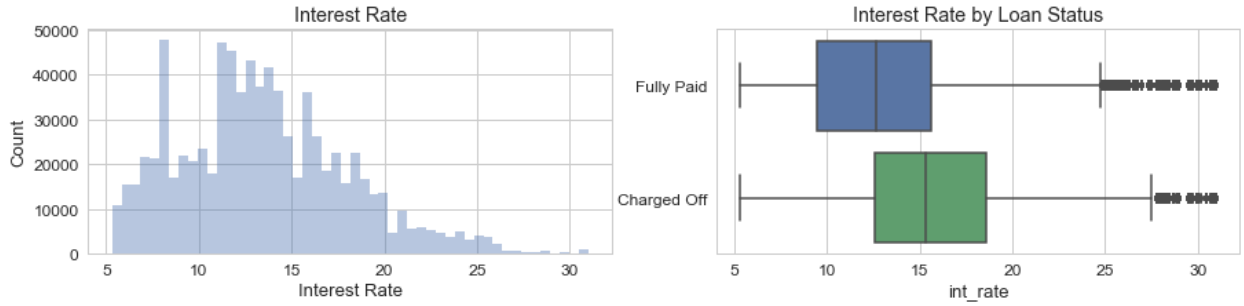

Fig 10: Illustrates the distribution of Interest rate in the Lending Club dataset.

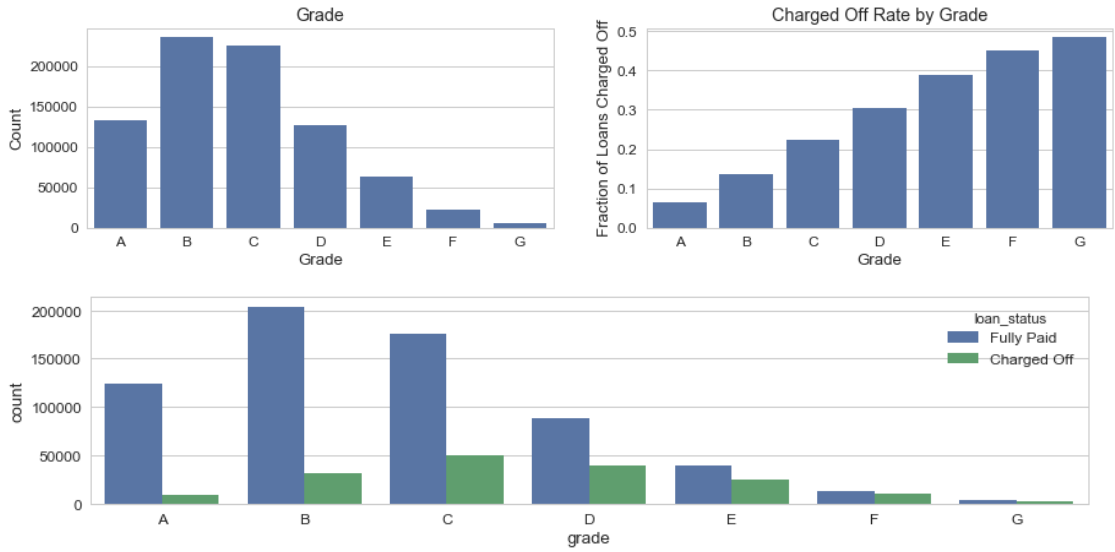

Fig 11: Illustrates the Count and then the Default Rate by Grade in the Lending Club dataset.

2.3 Learner

We will be using a RF in as our ML. For readers unfamiliar with RF we breakdown the model structure through first defining decision trees and then ensemble methods. The reader may skip this section and go to page 33 for the hyper parameters selected for the RF.

2.3.1 Decision Tree (DT)

A decision tree naively is a series of decisions undertaken through some form of information heuristic and stored in a tree like hierarchical structure. A more intuitive explanation can be shown through playing a game of Twenty Questions. Your opponent has secretly chosen a subject, and you must determine the subject. At each turn, you are allowed to ask a yes-or-no question, and your opponent must answer truthfully. Since, we have limited number of question we have to determine the value of each question asked so we are able to narrow down the space of possible subjects. If we draw the series of questions the resulting graph represents a tree with binary splits at each node. Each question is carefully crafted to provide the most information regarding the secret subject, and this is the intuition behind decision trees.

There have been several approaches to build decision trees in literature. We will use Classification and Regression Trees (CART) as proposed by Breiman et al (1984) in our model as this variant is supported by Scikit-Learn. CART determines the split based on the Gini Index as shown below.

$$G(X_i) \coloneqq \sum_{j=1}^{J} Pr(X_i = L_j)(1 - Pr(X_i = L_j)) = 1 - \sum_{j=1}^{J} Pr\,(X_i = L_j)^2.$$

For a candidate (nominal) split attribute X_i , denote possible levels as L_1 …, L_j . Once Gini Indices are computed for each candidate split attribute, the split is done on the attribute that has the highest Gini Index.

The CART algorithm recursively determines splits at each node until it determines that no further gain may be made or a pre-set stopping rule is satisfied. Next subsection we will discuss ensemble learning and how a decision tree model may be converted to forests.

2.3.2 Ensemble Learning

To provide an intuitive explanation of ensemble learning let us go back to the Twenty question game developed in the previous section. Now, suppose that you have asked your twenty questions. You are then offered the option to discuss your answer with a friend who also has asked twenty questions, but their questions have been asked independently from yours. You and your friend then collectively guess the subject. One might have heard the phrase 'two heads are better than one' and that translates to our modified game. This is the basic motivation of ensemble learning, where multiple learners can learn independently on the sample space and then pool in their predictions together. Fig 12 visualizes ensemble learning.

Ensemble learning can consist of different techniques but we will only discuss bagging. Bagging involves having each model in the ensemble vote with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set.

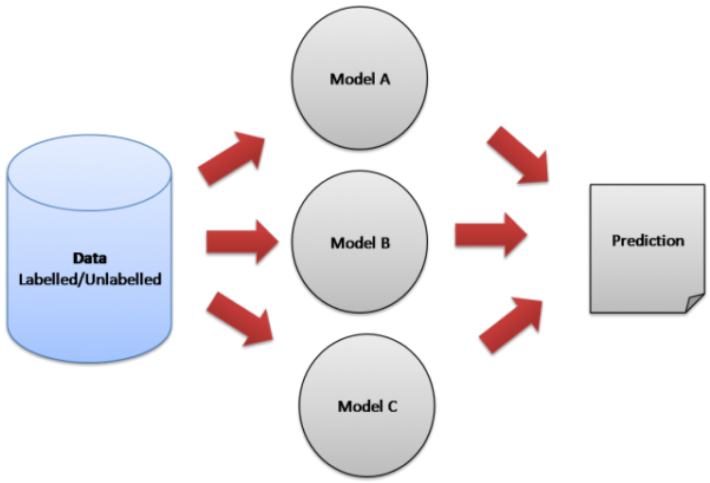

Fig 12: Ensemble learning is illustrated in this image. Multiple models may be developed and their results pooled in for a final prediction.

The next section we will pool in our discussion of DT and ensemble learning to convey the technique of a RF.

2.3.3 Random Forest

RF are multiple decision trees that are structured by using the ensemble learning technique called bagging. As discussed before this would result in each decision tree learning from a randomly drawn subset of the dataset. Splitting within each tree is done using the Gini Index at each tree node. The attribute that has the highest Gini Index is chosen for the nominal split. A key component of a RF that we have not discussed is what features are used to form the best split. Each decision tree randomly chooses *n* number of features where *n* is a hyperparameter of the model. A formal definition of a RF is given below:

Definition 1. A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input **x**.
Consequently, the results of the multiple DT are pooled in through majority voting.

As mentioned in Introduction we will also use RF with balanced bootstraps. This means that each tree in the RF will learn from a balanced random subsample of the dataset.

2.3.4 Model Hyperparameters

This paper implemented the model architecture proposed by Milad with the RF size being 80, attribute selection 5, and the tree depth of 25 is used. These numbers mean that for each tree the candidate split attribute are chosen by a random selection of 5 attributes from the full set of attributes. The split is only allowed to use one attribute out of the 5, and then a new set of attributes are selected. For each tree in the classifier, the tree is allowed to grow for a depth of 25.

3. Results

3.1 Evaluation Metrics

Accuracy, the percentage number of correctly classifies predictions, is one of the most intuitive ways to evaluate learner. However, the metric can be shown to be flawed when in the use of highly skewed data. For example, if the minority class was only 5 percent of the dataset, a learner could simply overfit to the majority class and would be able to achieve an accuracy score of 95%. Although, on its surface, an accuracy of 95% may show that we have a good learner, but that is further from the truth considering in certain cases a misclassification of the minority class is not acceptable. We observe this in fraud detection, anomaly detection, and information retrieval. Thus, it's imperative that the metrics evaluate our learner take into account the true class membership of each observation with the prediction of the classifier. To illustrate the alignment of predictions with the true distribution, a confusion matrix (Fig 13) can be constructed. Using the confusion matrix further metrics can be derived which have been used in literature for evaluating learners based on imbalanced data. We use sensitivity measures such as precision, recall, accuracy score, F-1 measure, and root mean squared error (RMSE) to evaluate our models. These metrics are defined below.

model prediction

actual loan status		no default (0)	default (1)
	no default (0)	TN	FP
	default (1)	FN	TP

Fig 13: This shows what sections of the confusion matrix is labeled as TN, FP, FN, TP.

Precision

This metric is defined as the number of true predictions of the borrower over the total number of predictions belonging to the positive class which is the sum of true positives and false positives. A precision value of 0.80 would be interpreted as the model predictions are correct 80% of the time.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

This metric is defined as the ratio of true positives divided by the total number of positive predictions. It is interpreted as what portion of actual positives was classified correctly. For example, if the recall value is 0.80 that means the learner correctly classified 80% of all loan status.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Accuracy Score

The percentage number of correctly classified predictions.

F-1 Measure

This measure is defined as the harmonic mean of precision and recall. The measure reaches its best score at 1 and the worst score at 0.

Root Mean Square Error

This metric simply the squared difference between the predictions by the learner and the observed values. The lower the value the better our learner.

3.2 Model Comparison

The tables shown below contain the results of our study. Table 3 is an exploration of improving the model proposed by Milad through balancing techniques. The dataset for all of the learners was prepared according to the specifications Milad laid out. The highest accuracy was achieved by SMOTE RF. The model also scored the highest F-1 measure, and Recall measure.

Classifier	Accuracy (%)	Precision	Recall	RMSE	F-1 Measure
RF (Milad)	78	0.72	0.63	0.42	0.72
RF-SMOTE	**84.0**	0.76	**0.84**	0.41	**0.77**
RF-NM	49.1	0.75	0.49	0.71	0.55
RF-RUS	61.0	**0.79**	0.61	0.63	0.66
RF-Balanced	83.7	0.76	**0.84**	**0.40**	**0.77**
RF-SMOTEBORDER	83.3	0.76	0.83	0.41	**0.77**

Table 3: Shows the different evaluation metrics for the different classifiers plus sampling techniques on Milad's processed dataset Jan 2012 - Sept 2014

Table 4 shows results and is an exploration to improve the model through the addition of more features. The data is from Jan 2012 to September 2014 so we can make a fair comparison to our own results in Table 3. RF with SMOTE as the balancing approach scored highest in all metrics except for precision. SMOTEBORDER is a close second in terms of the evaluation metrics as compared to SMOTE.

Classifier	Accuracy (%)	Precision	Recall	RMSE	F-1 Measure
RF (Milad)	78	0.72	0.63	0.42	0.72
RF-SMOTE	**87.0**	0.87	**0.87**	**0.36**	**0.87**
RF-NM	78.4	0.87	0.78	0.46	0.81
RF-RUS	79.2	**0.88**	0.79	0.46	0.81
RF-Balanced	83.7	0.81	0.83	0.40	0.84
RF-SMOTEBORDER	86.4	0.87	0.86	0.37	**0.87**

Table 4: Shows the different evaluation metrics for the different classifiers plus sampling techniques on our processed dataset Jan 2012 - Sept 2014

In the next section we will discuss our results, their implications, and provide future insight to researchers in this field.

4 Discussion

This study examined the classification variable in credit risk modelling within Peer to Peer lending market. We used the Lending Club dataset as a proxy to model the risk of default in P2P lending. We explored balancing techniques and the use of more features to improve the credit risk model suggested by Milad (2015).

Our results from Table 3 that draw a comparison to Milad (2015) model to the different balancing techniques suggest that there is evidence that supports our hypothesis that Milads learner was bias towards the majority class. We can observe this as the F-1 measure increases for the sampling techniques that oversample the minority class. Under sampling techniques do poorly on Milads dataset, and model. We do not know the reason why this occurs. We think that this may because that certain instances of the majority class is removed that are highly correlated with the classification variable but further exploration of feature variables present in Milads dataset is required to understand why that when undersampling of the majority class occurs the model greatly underperforms in all metrics.

We also proposed a series of new features to be added into the model as we felt from reading the Lending Club dictionary that these variables are relevant in determining the status of the loan. One such variable is last_pymnt_amnt which is the amount paid by the borrower on their last trade balance. This value represents the borrower making strides in repaying their credit which may be due on several accounts. The payment may not be related to Lending Club, but

because we have no way to confirm that a borrower only has one loan listing this value could also represent payment towards a Lending Club loan. We can observe this in Fig 14. These feature importances were determined by the Gini Index and is stored by the Scikit-Learn RF classifier.

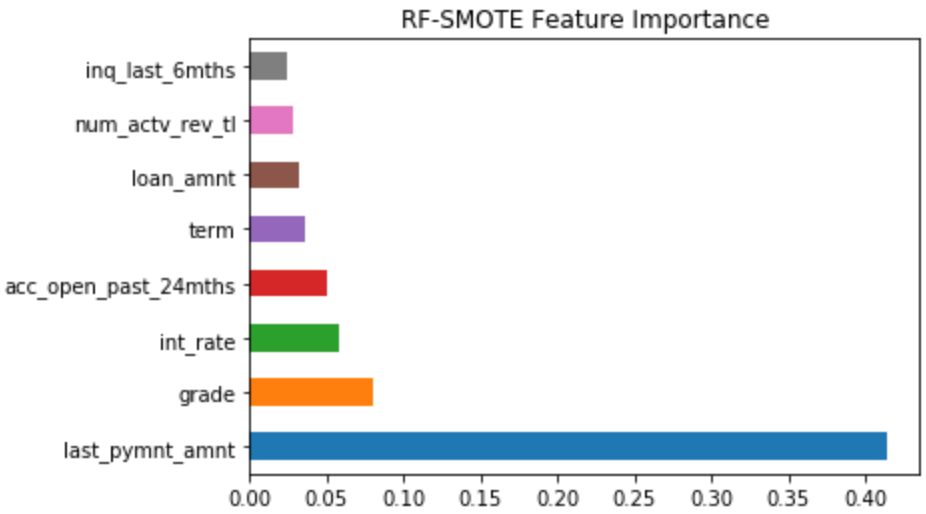

Fig 14: The top 8 most important features in RF-SMOTE

The variable last_pymnt_amnt is available to investors so we are not sure why such an important variable was left out in the analysis conducted by Milad. We can also observe other feature variables that we included in our model are in the top 8 features.

Currently, there are several limitations within our own analysis. Fig 15 shows the issuance of number of loans issued since 2007. We can observe that the market started to gain traction through the financial crisis of 2008 and as the economy improves so did the number of loans. We wonder whether the credit risk model could be further improved by incorporating the financial health of the economy per fiscal year. As this market is for borrowers in a lower income

group a further analysis of how national and global economic indicators may help in the model building process. We can see the importance of economic indicators because Lending Club offers its investors with a feature variable called ‘msa’ which is the Metropolitan Statistical Area of the borrower. This feature describes the economic wellbeing of the area in which the borrower lives. This feature , however, is not available in the dataset provided on their website although there is a feature for the state the borrower lives.

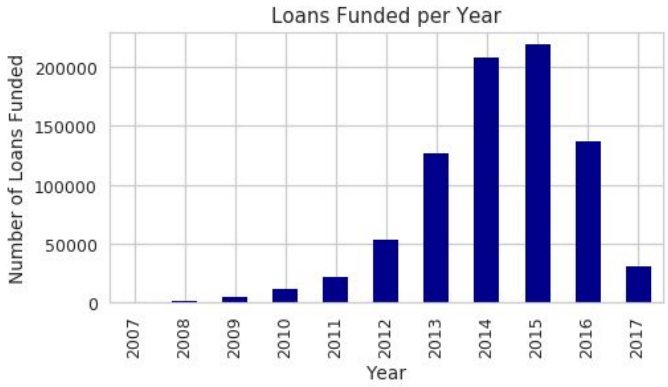

Fig 15: The number of loans issued and funded each year from 2007 to 2018

Although, we wanted to test our improved models on the entire dataset from 2007 to 2018 we first plot the RF-SMOTE learning curve as shown in Fig 16. The graph shows that the learner stops learning at about 9000 training samples. We could, however, simply randomly sample the entire dataset and then train our model, but we choose not to. Firstly, because we hypothesize that for an investor in 2019 borrower behaviors in the early stages of this market and in the financial crisis may mislead the learner. We feel an in depth analysis of economic indicators and the date if loan issuance should be further studied to ensure the learner that is learning may keep up with the changing behaviors of the borrower.

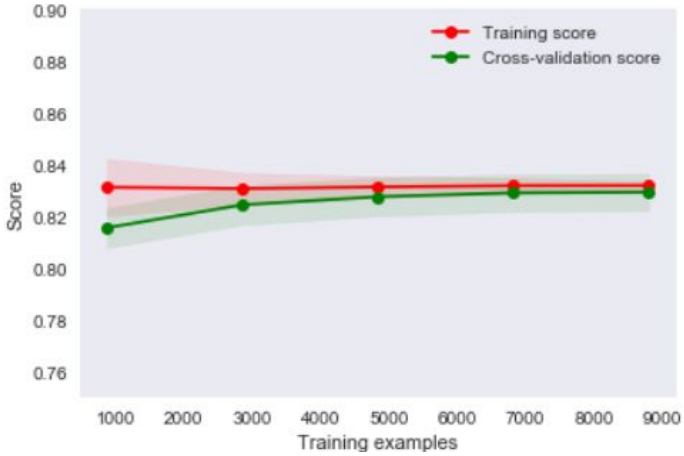

Fig 16: The plot shows the learning curve for RF-SMOTE

.

Bibliography

1. Connor, G., Goldberg, L., & Korajczyk, R. (2010). Credit Risk. In *Portfolio Risk Analysis* (pp. 212-240). Princeton University Press. Retrieved from http://www.jstor.org/stable/j.ctt7sm49.16
2. Engle, R. (2009). Credit Risk and Correlations. In *Anticipating Correlations: A New Paradigm for Risk Management* (pp. 122-129). Princeton University Press. Retrieved from http://www.jstor.org/stable/j.ctt7sb6w.13
3. Malik, M., & Thomas, L. (2010). Modelling Credit Risk of Portfolio of Consumer Loans. The Journal of the Operational Research Society, 61(3), 411-420. Retrieved from http://www.jstor.org/stable/40540268
4. Bacham, D., & Zhao, J., Dr. (2017, July). Machine Learning: Challenges, Lessons, and Opportunities in Credit Risk Modeling. Retrieved April 16, 2019, from https://www.moodysanalytics.com/risk-perspectives-magazine/managing-disruption/spotlight/machine-learning-challenges-lessons-and-opportunities-in-credit-risk-modeling
5. Rajdeep Sengupta & Craig P. Aubuchon, 2008. "The microfinance revolution: an overview," Review, Federal Reserve Bank of St. Louis, issue Jan, pages 9-30.
6. Havrylchyk, Olena and Mariotto, Carlotta and Rahim, Talal-Ur- and Verdier, Marianne, What has Driven the Expansion of the Peer-to-Peer Lending? (February 23, 2018). Available at SSRN: https://ssrn.com/abstract=2841316 or http://dx.doi.org/10.2139/ssrn.2841316
7. Peer-to-Peer Lending in the United States: Surviving after Dodd-Frank Notes & Comments: I. The Dodd-Frank Wall Street Reform and Consumer Protection Act Magee, Jack R. Page 139
8. Kruppa, J., Schwarz, A., Arminger, G., & Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. Expert Systems with Applications, 40(13), 5125-5131.
9. Khandani, A.E., Kim, A.J., & Lo, B.A. (2010). Consumer credit-risk models via machine-learning algorithms q.
10. Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, Lyn C. Thomas, Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, European Journal of Operational Research, Volume 247, Issue 1, 2015, Pages 124-136, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2015.05.030.
11. Milad Malekipirbazari, Vural Aksakalli, Risk assessment in social lending via random forests, Expert Systems with Applications, Volume 42, Issue 10, 2015, Pages 4621-4631, ISSN 0957-4174,
12. Chawla, Nitesh & Bowyer, Kevin & O. Hall, Lawrence & Philip Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. 10.1613/jair.953.

13. M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In Proceedings of the Fourteenth International Conference on Machine Learning, pages 179-186, Nashville, Tennessee, 1997. Morgan Kaufmann.
14. I. Mani, I. Zhang. "kNN approach to unbalanced data distributions: a case study involving information extraction," In Proceedings of the workshop on learning from imbalanced datasets, 2003
15. Chen, Chao, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data." The University of California, Berkeley 110 (2004): 1-12.
16. Kotsiantis, Sotiris & Kanellopoulos, D & Pintelas, P. (2005). Handling imbalanced datasets: A review. GESTS International Transactions on Computer Science and Engineering. 30. 25-36.
17. Zhang, DunGang, et al. "The Credit Risk Assessment of P2P Lending Based on BP Neural Network." *Industrial Engineering and Management Science*, 1st ed., CRC Press, 2014, pp. 90–94.
18. Fu, Y. (2017) Combination of Random Forests and Neural Networks in Social Lending. *Journal of Financial Risk Management*, **6**, 418-426. doi: 10.4236/jfrm.2017.64030.
19. Chawla, N.V., Japkowicz, N., and Kotcz, A.: 'Special issue on learning from imbalanced data sets', ACM Sigkdd Explorations Newsletter, 2004, 6, (1), pp. 1-6
20. Out of Sight, Not Out of Mind: Strategies for Handling Missing Data
21. H. M. Nguyen, E. W. Cooper, K. Kamei, "Borderline over-sampling for imbalanced data classification," International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4-21, 2009
22. Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. Classification and Regression Trees. New York: Chapman & Hall/CRC
23. LendingClub, Prosper, SoFi & OnDeck - Marketplace Lending Takeaways. (2016, May 10). Retrieved from https://www.valuewalk.com/2016/05/lendingclub-prosper-sofi-ondeck-marketplace-lending-takeaways/