A Relational Model of Data for Large Shared Data Banks

Rahul Shamdasani
Graduate Student : M.S. C.S.
Indiana University, Bloomington
rshamdas@iu.edu

Shilpa Kumari
Graduate Student : M.S. C.S.
Indiana University Bloomington
shkumari@iu.edu

Richa Jha
Graduate Student : M.S. C.S.
Indiana University Bloomington
ricjha@iu.edu

ABSTRACT

In today's world, obtaining loans from a financial institution has become a common phenomenon, in this paper we will discuss the profitability of an institution from a business perspective and try to maximize the same, using various machine learning techniques. We will predict loan status based on applications and applicant's credit history. Loans taken by a borrower can be fully paid, default or charged off depending on a lot of factors out of which some are taken into account.This project will help solve the real world problem by working on the real world data. After this project we will not only have learned about the algorithms needed, but will also understand the importance of data preprocessing and various ways used to do the same. LendingClub is the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market.

1 MOTIVATION

LendingClub is the world's largest peer-to-peer lending platform. Lending club works as a mediator between investors and borrowers. It works in favour of investors and helps them identify potential borrowers for lending their money. Potential borrowers are the ones who are likely to pay their loans on time. These potential borrowers are identified based on the credit history of the borrower and other application specific factors like interest rates, term of loan etc. Our project will help solve this issue by using machine learning techniques.

2 RELATED WORK

Abhishek Bhagat. [1] have used Regression model for prediction and ended up with a precision value of 90 , we have compared the results for 3 Models as a part of our project and for all the three we are also getting accurate results of close to 90 but the point that should be noted is, we are able to get the best possible values for false positives even better than the above paper. So we can say that although our model might pass on some of the potential candidates of loans but it will be better in the terms of returns, which is beneficial for the organization.

There have been many studies on classification models predicting LendingClub loan default. Chang et al. [3] built Logistic Regression, Naive Bayes, and SVM classifiers, all of which are able to achieve a G-mean score of around 0.86, the geometric mean of true positive and true negative rates. However, we find it questionable that loans with a Current status were treated as positive examples, along with Fully Paid loans. Since current loans may become default in the future, this practice invariably labels some true negatives as positive. In light of this, we decide to restrict our dataset to finalized loans only.

Tsai et al. [4] also experimented with the three models above along with Random Forest, but with an emphasis on precision at the expense of recall and negative predictive value (i.e. precision for the negative class). They find that Logistic Regression achieves a greater precision than the other models; they also break down the metrics by LendingClub's assigned loan grades (A-G) and subgrades (e.g. A-1). We believe that precision for both classes and their recalls are equally important metrics to optimize for, as a naive model which always predicts positive already achieves a good precision since the majority of examples are positive, but its negative predictive value would be zero.

In addition to classification models that predict loan default, Gutierrez and Mathieson [5] built regression models that predict the annualized return of a given loan. The loan selection strategy derived from a combination of these models was able to achieve better investment performance as measured by the Sharpe ratio than the baseline. This encourages us to build regression models and evaluate an investment strategy that select loans with high enough annualized return predictions.

Pujun et al. [6] built classification and regression models, but the goal was to predict LendingClub loan approval and their assigned interest rates. They applied k-means clustering and PCA techniques to detect latent trends in LendingClub approved loans. One of their most interesting findings is that loan approval standard had been gradually relaxed over the years. This reaffirms the desirability and usefulness of developing an independent and effective model for evaluating credit risks.

3 PERFORMANCE MATRIX

To maximize the profitability of the organization, the best matrix for evaluation was precision of the model. Inside the precision we are more concerned about reducing the number of false positive results.

- **True Positive(TP)**: These are the candidates which are predicted as potential borrowers by our model and are actually potential borrowers.
- **False Positive(FP)**: These are the candidates which are predicted as potential borrowers by our model and have actually defaulted their loans.
- **True Negative(TN)**: These are the candidates which are predicted as defaulters by our model and are actually potential borrowers.
- **False Negative(TN)**: These are the candidates which are predicted as defaulters by our model and are actually defaulters.

Our main aim while training this model is to minimize the number of false positive(FP) results, since these are the customers which cause heavy losses to lending club.

4 DATASET AND FEATURES

4.1 Dataset Overview

We used Lending Club's data for this analysis [1].The Model will be trained on a dataset having almost 4 Million entries and a total of 28 different parameters.It is often very hard to run analysis with all the variables and observations. So, we cleaned and processed this data .Such a huge dataset was helpful for our task.The following images are a part of the dataset. We split the data using a random (0.7, 0.3) split into training and test sets.

The majority of loans were under the "Fully Paid" category. The "Fully Paid" and "Charged Off" categories were the target for prediction.

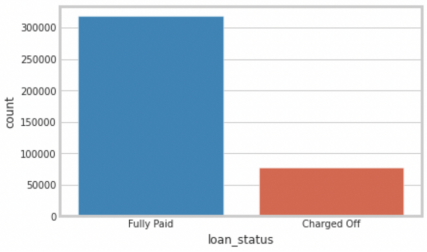

Figure 1: Count of loan status by type

4.2 Features

The following table consists the features of the dataset with their description:

	LoanStatNew	Description
0	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.
1	term	The number of payments on the loan. Values are in months and can be either 36 or 60.
2	int_rate	Interest Rate on the loan
3	installment	The monthly payment owed by the borrower if the loan originates.
4	grade	LC assigned loan grade
5	sub_grade	LC assigned loan subgrade
6	emp_title	The job title supplied by the Borrower when applying for the loan.*
7	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.
8	home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
9	annual_inc	The self-reported annual income provided by the borrower during registration.
10	verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
11	issue_d	The month which the loan was funded
12	loan_status	Current status of the loan
13	purpose	A category provided by the borrower for the loan request.
14	title	The loan title provided by the borrower
15	zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
16	addr_state	The state provided by the borrower in the loan application
17	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.
18	earliest_cr_line	The month the borrower's earliest reported credit line was opened
19	open_acc	The number of open credit lines in the borrower's credit file.
20	pub_rec	Number of derogatory public records
21	revol_bal	Total credit revolving balance
22	revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
23	total_acc	The total number of credit lines currently in the borrower's credit file
24	initial_list_status	The initial listing status of the loan. Possible values are – W, F
25	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
26	mort_acc	Number of mortgage accounts.
27	pub_rec_bankruptcies	Number of public record bankruptcies

Figure 2: Information on the dataset

5 EXPLORATORY DATA ANALYSIS

5.1 Feature Preprocessing

Columns with empty values for most of the rows as well as columns with the same values across all rows are dropped in order to have a cleaner dataset.For features with missing values, they are categorized into three cases and treated differently: mean-set, zero-set and max-set.For data preprocessing we performed following steps:

- Remove or fill any missing data.
- Remove unnecessary or repetitive features.
- Convert categorical string features to dummy variables.

Figure 3: Properties of dataset

Realistically there are too many unique job titles to convert this to a dummy variable feature. we removed that employment title column.Charge off rates are extremely similar across all employment lengths. So we are going to drop the employment length column.The title column is simply a string subcategory/description of the purpose column. So we are going to drop the title column.Categorical features, such as application Type,home ownership, are replaced with their one-hot representations. Normalization is then performed at the end on all features so they have zero mean and one standard deviation.We perform Min-Max scaling(normalization) to re-scale values between the range of 0 and We do this by subtracting the min value and dividing by max minus the min

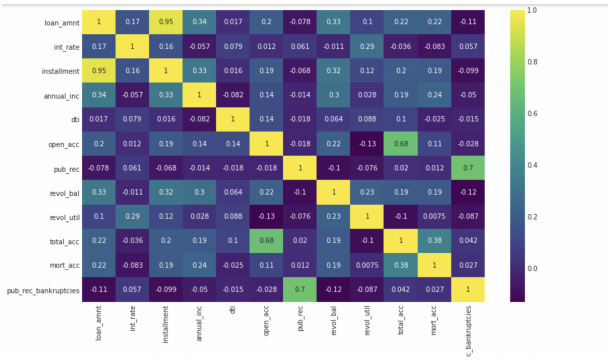

Figure 4: Feature correlation heatmap

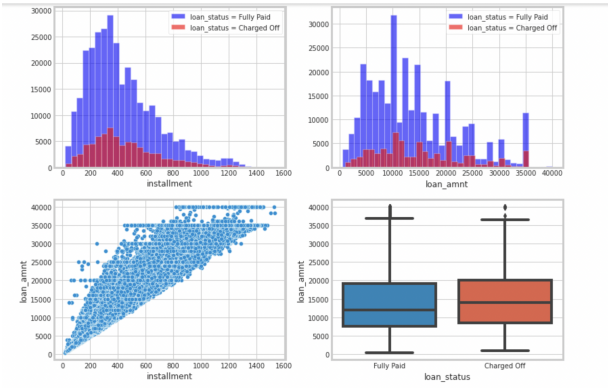

Figure 5: Loan amount and installment against loan status.

On plotting correlation among two sets of features of our dataset,we noticed almost perfect correlation between "loan amount" the "installment" feature. We'll explore this features further.

```
data.groupby(by='loan_status')['loan_amnt'].describe()
```

loan_status	count	mean	std	min	25%	50%	75%	max
Charged Off	77673.00	15126.30	8505.09	1000.00	8525.00	14000.00	20000.00	40000.00
Fully Paid	318357.00	13866.88	8302.32	500.00	7500.00	12000.00	19225.00	40000.00

Figure 6: statistics of loan status.

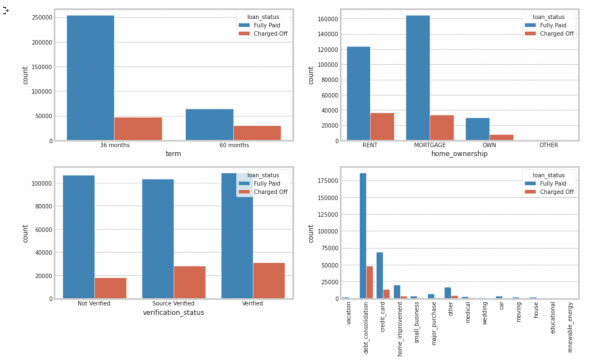

Figure 7: Count plot of term,home ownership,verification Status and purpose.

It seems that loans with a high interest rate are more likely to be unpaid.Higher the annual income the higher the likelihood of the loan being fully paid. However, if the annual income is low, there are chances of the loan to be Charged Off. Grades and subgrades

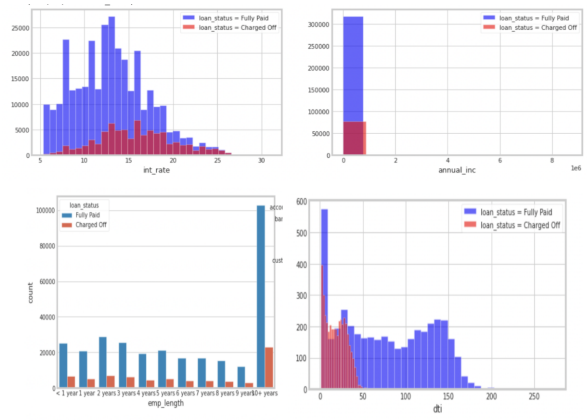

Figure 8: Count plot of interest rate,annual income,diti and employment length for loan status.

are given to a particular loan transaction based on the probability of getting the amount back, this probability is based on the credit history of the borrower. So it is clearly visible that defaulters will probably end up having a grade of F and G, Our goal is to reduce the number of defaulters who have good grades but still are likely not to pay the amount. Count of simplified loan status by employment

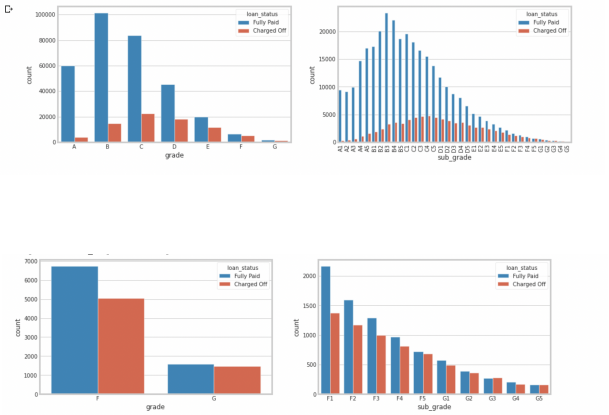

Figure 9: Count plot of grades and sub grades for loan status.

length,This indicates that the higher the employment lengths the more the fully Paid loans.

Graph shows that higher the debt to income ratio , higher the probability of the loan being Charged Off.Mean for accounts defaulter's account is lesser than the mean for accounts who paid fully.

- It seems that the smaller the dti the more likely that the loan will not be paid.
- Only 217 borrowers have more than 40 open credit lines
- Only 266 borrowers have more than 80 credit lines in the borrower credit file.

We notice that, there are broadly two types of features:

- Features related to the applicant (demographic variables such as occupation, employment details etc.),
- Features related to loan characteristics (amount of loan, interest rate, purpose of loan etc.)

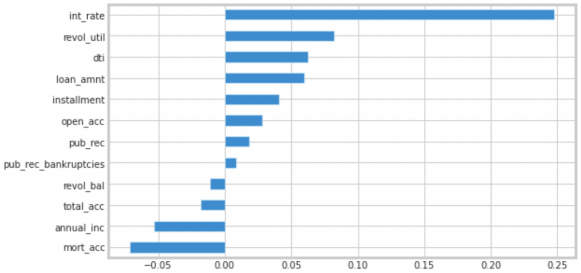

Figure 10: Numeric features correlate with the target variable(loan status).

6 CLASSIFICATION

Our classification goal is to predict which class the loan belongs to: either Default or Fully Paid. In the following sections, we will share and discuss our experiments using Neutral Networks and Random Forest for classification problem. For metrics to evaluate classification performance, we use confusion matrix whose columns represent predicted values and rows represent true values. We also measure precision, recall, f1-score (the harmonic mean of precision and recall) and weighted average as defined below:

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

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

$$F1 - score = \frac{2TP}{2TP + FP + FN}$$

$$Support = the number of true instances for each label$$

$$Weighted - avg metric = metric weighted by support$$

7 MODELS

7.1 Artificial Neural Network

As pearson's correlation of some of our attributes are not linear and ANN is more robust for the data being used.All neurons in one layer of the network are connected to all neurons within the next layer while the neural network is learning the relationships between the data and results, it is said to be training.The best ANN architecture for classification for the model was obtained using a trial-and-error approach by varying the number of neurons in the hidden layer, as described in model summary.We constructed a fully connected neural network with 3 hidden layers of shape (128, 56, 28),Relu activation for all input and hidden layers neurons and sigmoid activation function for output layer neurons.The main reason why we used sigmoid activation function for output layer is because our problem is Binary Classification.We arrived at these hyper-parameter values by experimenting with various settings.The generated model displays information about the neural network[fig 13].

```
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 78)                6162
_________________________________________________________________
dense_1 (Dense)              (None, 128)               10112
_________________________________________________________________
dense_2 (Dense)              (None, 56)                7224
_________________________________________________________________
dense_3 (Dense)              (None, 28)                1596
_________________________________________________________________
dropout (Dropout)            (None, 28)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 29
=================================================================
Total params: 25,123
Trainable params: 25,123
Non-trainable params: 0
```

Figure 11: Summary of ANN model

The final output of the network uses binary cross entropy (log loss) as loss function for the sequential model .To arrive at optimal parameters, the model iteratively updates weights within each layer using Keras Adam optimizers with a mini-batch size of 8 and epoch 25.

7.2 XG Boost

We are using extreme gradient boosting as one of the models for our data, XG boost is a tree based ensemble machine learning algorithms. We selected this algorithm since we have a large data set and data is a combination of categorical and numerical values, also we have only 77k samples for charges off case which is very low compared to 3.1M samples for fully paid data so XG boost observation weightage takes care of that part, as it make multiple trees of N random samples with replacement, after each iteration the observations are weighted so there might be some samples which get selected multiple times.

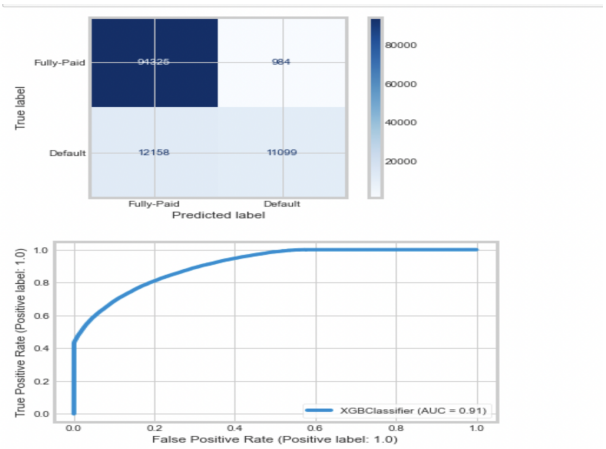

Figure 12: Confusion metrics and AUC Score generated for XGBoost Model

Using the XGBoost Classifier model in a case of predicting the default rate for the loans sanction by the lending club we were able to obtain accuracy of 89.58 percent on our training data, also gave us an accuracy of 88.92 on the test data. Also, this gives the highest score of 0.91.

Hyper Parameters

- **Booster**:GBTree
- **Step Size Shrinkage**:: 0.3
- **Alpha Alias**: Learning Rate

7.3 Random Forest:

A Random Forest is an ensemble of Decision Trees and performs various tasks ranging from regression and classification. Being an ensemble, it gives better performance and is more robust than decision trees. The training in Random Forest is achieved through Bootstrap Aggregation, also known as Bagging. Bagging is a way of creating multiple subset samples using a replacement for fitting the model.

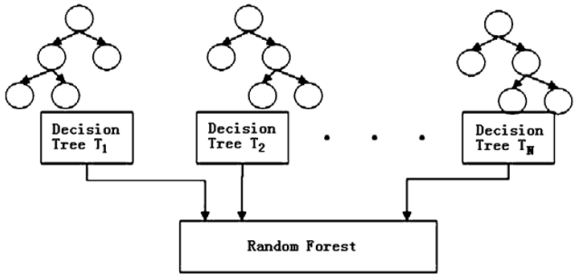

Figure 13: Random Forest Classifier

The advantage achieved is that we can use several instances to train our model. In the case of bagging in trees, also known as "feature-bagging," we make these subsets of training data and fit a decision tree to each of them and, in the end, aggregate the outcome. This contrasts with the top-down greedy approach to search for best predictors and partitioning of branches; instead, we create more randomness and diversity by applying the bagging method to the feature space. This helps reduce the variance in our prediction, albeit at the cost of an increase in bias. In the case of Random Forests, each instance does not need to go from the root node to the bottom until it has been classified; instead, it visits all the different trees in an ensemble that have grown until now using randomization. These functionalities differ a bit for Classification and Regression tasks. In the case of classification, we use the mode or most frequent class predicted, whereas, in regression, we use the mean value.

8 COMPARING MODELS:

We can see the prediction accuracy is similar for all the three models. However, we can see that XGBoost and ANN Model performs more robustly since there is very less variation between its train and test set accuracy, which goes on to show a good balance between under-fitting and overfitting of our model. However as mentioned earlier our primary objective is to identify the false positive correctly. As seen in the given slide the Random forest gives the best results in terms of false positives.

	Train data Error	Test Data Error
ANN Model	88.92%	88.75%
XGBoost Classifier	89.50%	88.92%
Random Forest Model	100.00%	88.88%

Figure 14: Accuracy on training and testing data

8.1 XGBoost Classifier:

Using the XGBoost Classifier model in a case of predicting the default rate for the loans sanction by the lending club we were able to obtain accuracy of 89.58 on our training data, also gave us an accuracy of 88.92 on the test data. Also, this gives the highest score of 0.91.

8.2 Neural network:

As we can see in the ANN outputs, the ANN has been trained well and gives us accuracy of 89 for both train and test data for classification of our target variables (Loan status). Also, the loss keeps reducing as the number of epochs increase, which is what we have expected.The outputs of epoch-by-epoch loss functions and accuracies at the end of each epoch of training for ANN model.

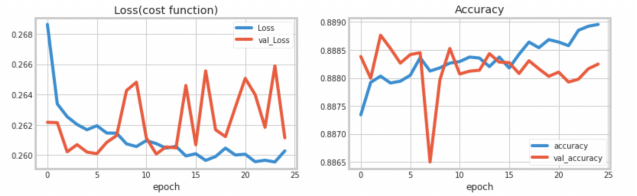

Figure 15: ANN Model Loss and Accuracy on training and testing data

8.3 Random Forest:

Upon using the random forest model in a case of predicting the default rate for the loans sanction by the lending club we were able to obtain accuracy of 88.88 on our training data, also gave us an accuracy of 100 on the test data. There is a huge variation in the test error and train error which basically means we have overfit our model on the training set. We can hypertune the parameters to reduce this gap and use cross validation to improve the performance. We can also see from the ROC curve that we were able to obtain an AUC score of 0.89.

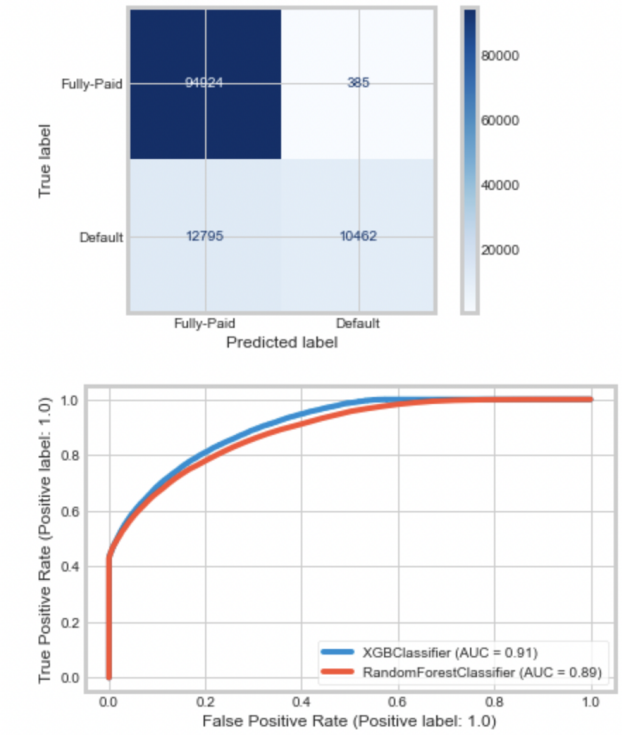

Figure 16: Confusion metrics and AUC Score generated for Random Forest Model

9 CONCLUSION:

We have successfully built a machine learning algorithm to predict the people who might default on their loans. We opted against using Principal Component Analysis for dimensionality reduction since we had a lot of categorical data. This can be further used by LendingClub for their analysis. For our purpose we used Artificial Neural Network, XGBoost and Random forest for our Modelling purpose. Upon doing our exploratory data analysis and using the linear measures of correlation between the predictors and the response, the most important variables for predicting loan defaulters are total mortgage accounts, annual income, installments and loan amount. During the course of our project, we realized some key points. Collection of more data points for loan defaulter (false positives) will help us classify them better. Also, we can use other pre-processing techniques or variables to improve the prediction power of the algorithm. We can also categories the loan defaulters based on how many times they have defaulted. This will help us minimize the false negative which in turn will help us maximize profit.

The biggest challenge in the prediction of loan defaulters was the data surrounding the limited number of people who defaulted on their loan. We can use recent data compare the number of current loans that were paid off or defaulted. We should also note the loans that were charged off. Then these new data points can be used for predicting them or even used to train the model again to improve its accuracy. As mentioned earlier, instead of PCA we can use different variable selection techniques like Multiple correspondence Analysis to further reduce the dimension. This will help us select the most important features. This will enhance the model fit since attributes which are not contributing much to our model will be removed. In the end, our model was able to predict the lending club default rate at a good accuracy and making few changes in pre-processing and reduction in dimensionality we can further improve the performance of our model.

10 REFERENCES

(1) Abhishek Bhagat "Predicting Loan Defaults using Machine Learning Techniques" Journal of Machine Learning Research
(2) F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
(3) S. Chang, S. D.-o. Kim, and G. Kondo, "Predicting default risk of lending club loans," 2015.
(4) K. Tsai, S. Ramiah, and S. Singh, "Peer lending risk predictor," CS229 Autumn, 2014.
(5) A. Gutierrez and D. Mathieson, "Optimizing investment strategy in peer to peer lending," 2017.
(6) B. Pujun, C. Nick, and L. Max, "Demystifying the workings of lending club,"
(7) "Lending club statistics — lendingclub." https://www.lendingclub.com/info/download-data.action. (Accessed on 12/08/2018).
(8) "Predict lendingclubs loan data." https://rstudio-pubs-static.s3.amazonaws.com/203258$_d$20c1a34bc094151a0a1e4f4180c5f6f.html. (Accessed on12/08/2018).