Final Project Report

Asrorbek Orzikulov, Meduri Venkata Shivaditya, Xinyu Hu

December 23, 2020

1 Abstract

For most commercial banks, the loan portfolio—all outstanding loans a bank has issued—is the most important source of revenue. Therefore, any bank does its best to have only reliable, performing loans in its loan portfolio. This is not an easy task, however, and credit officers at banks spend a significant amount of time to determine whether a potential client is creditworthy, i.e., able and willing to repay a granted loan timely and fully.

The objective of this project is to train an Extreme Gradient Boosting (XGBoost) algorithm as well as the logistic regression and the least-squares models to see their utility in credit analysis. For this research, we used the data provided by LendingClub that includes the loans the company issued over the 2007-2018 period. Our analysis shows that the XGBoost classifier could correctly classify 88% of good loans and 91% of bad loans. Similarly, XGBoost regression can predict, on average, the expected loss with 15% accuracy.

2 Introduction

Although the banking industry has undergone significant changes over the last decades, interest income from loans still remains the largest source of revenue for most commercial banks. However, issuing a loan always involves a risk, and being able to differentiate high-quality borrowers from potentially problematic ones is of great importance for the following reasons:

1. Borrowers can default and cause a loss of capital for the bank.
2. Banks are highly leveraged companies; therefore, losing their capital damages their financials more than the amount of gain they realize when borrowers successfully repay their loan.
3. Decreasing the number of bad loans relative to good loans allows a bank to remain financially strong and earn a decent income for its shareholders.

For many years, banks relied on traditional credit analysis that uses various qualitative and quantitative information about potential clients to accomplish this task. However, these methods are usually slow and need trained personnel to carry out. As customer loan applications are proliferating and the competition in this sector is increasing year by year, the banks are experiencing a need to speed up the loan issuance process and increase the number of borrowers they can serve. The first step towards this goal can be automating accept-reject decisions when a potential client applies for a loan. To do it effectively and minimize default-related losses, a bank should have robust models that can predict the loan default probability and the recovery rate—the percent of the loan amount that can be recovered upon default—for each loan application.

This is the main reason why banks can benefit from embracing machine learning algorithms that can be used either instead of or as a complement to credit analysts to take correct decisions. At the very least, machine learning models can be used as a good screening device to avoid obvious bad loans, so that credit analysts can devote their time to cases requiring expert opinion. Regardless of their use, we believe that financial institutions can take advantage of such models, and in this paper, we examine two models that commercial banks can consider using.

3 Literature Review

One of the earliest studies that tried to predict bank defaults was conducted by Zurada (2002). The research utilized data provided by a lending institution on 3,364 consumer loans issued between 2000 and 2002. The data was fairly imbalanced, with the proportion of default loans around 9%. Zurada (2002) employed a logistic regression, decision trees, and Multilayer Perceptron (MLP) model and achieved an overall accuracy of 90-94%. However, the performance of Zurada's models was boosted by the large non-default class, while it could correctly classify only 40-46% of default loans. When the author eliminated some correlated variables and build an ensemble model combining all three models, the overall accuracy rose to 80%.

Wang and Priestley (2018) also used decision trees and logistic regression for loan default prediction. Their data included more than 11 million observations and 300 parameters. However, they dropped 114 parameters that had a significant proportion of missing values and used median values to fill missing values in other columns. After dimensionality reduction using variable clustering, they ended up with 14 predictors for the final models. As did Zurada (2002), the authors focused on accuracy to evaluate model performance, and the logistic regression showed the accuracy of 95%, while the decision tree algorithm did marginally better (97%).

To predict credit defaults, Zhou and Wang (2012) adopted a different approach than the above two papers in two ways. First, it used the Random Forests algorithm, which is an ensemble model fitting multiple decision trees. Second, they explicitly dealt with the class imbalance problem by over-sampling the default class and under-sampling the non-default class. As a result, they could achieve the accuracy score of 94% in their data, which had 150,000 observations $\frac{1}{13}$ of which were bad loans.

Research by Odegua (2020) involved the study of 4,346 consumer loans, relying solely on the XGBoost model. The study included extensive data preprocessing and feature engineering steps at the end of which 15 parameters were selected. To find the optimal parameter values, the study used 5-fold cross-validation, and the best model exhibited the accuracy of 79%. However, the recall of bad loans was significantly better than in other studies: in the test set, 98% of default loans were classified correctly (at the expense of good loans).

Independent variables used differed from one study to another. However, the length of employment, the total amount of other debt obligations, debt-to-income ratio, payment amount, and age were common to all. Indeed, these are also variables that traditional credit analysts study carefully, so their presence in all models should not be surprising. As to the unique variables, Zurada (2002) used property value and the number of credit lines in a borrower's credit history; Wang and Priestley (2018) included marital status, and Odegua (2020) controlled for the location of the borrower in their models to make loan default predictions.

Interestingly, all of the above papers used payment-related attributes (the last payment amount, the number of days since last delinquency) to make their predictions. However, these variables are not available when a credit manager reviews a loan application. (They become available only after loan issuance.) Therefore, our research tries to train a model that can differentiate between good and bad loans before their issuance. While doing so, we also compare how older models such as logistic regression and decision trees fare against newer models (e.g. XGBoost).

4 Research Objectives

4.1 Classifying a loan application

Our first aim is to build a classification model that predicts the probability of default for a given loan application and classifies it as either default or non-default. This model will help banks decide whether or not they should approve the application.

4.2 Predicting the Recovery rate at default

Our second aim is to use a regression model to predict the recovery rate if a customer defaults (0%-100% of the loan amount issued). If this metric is higher than the predetermined, bank-specific cut-off amount, the bank will issue a loan to the customer. It is worth noting that these two factors are the most important considerations while performing a credit analysis.

5 Feature Engineering

5.1 About the Dataset

The dataset used for the project is provided by LendingClub, an American peer-to-peer lending company headquartered in San Francisco, California. It includes 2,260,701 unsecured personal loans between $1,000 and $40,000 that LendingClub issued over the 2007-2018 period. 886,753 of those loans are ongoing loans that can ultimately turn out to be good or bad. Since their outcome is not clear, they will not be used in the analysis. About 1,373,915 observations, on the other hand, have an outcome (fully paid or default). Our analysis will be based on these data points.

The dataset includes 151 quantitative and qualitative features used by financial institutions to determine an applicant's creditworthiness. Some of those features are presented below.

1. Quantitative factors: loan amount, average transactions amount, assets, salary, age, etc.
2. Qualitative factors: marital status, education, employment sector, whether a person has a house, whether a person has a car, etc

5.2 Features with Missing Data

Through an exploratory data analysis, we found that the amount of missing data is significant. The following graph presents an overview of how much data was missing.

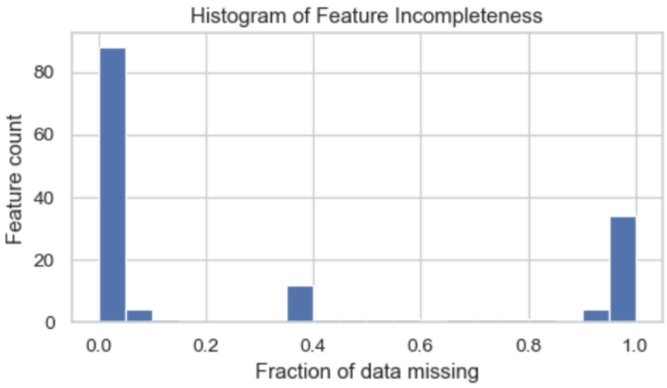

Figure 1: Visualization of missing data percentage

From the figure above, it is obvious that there is a gap between features lacking some data (less than 20%) and features missing a significant amount of data (less than 40%). Since filling in so many observations can significantly distort our results, we decided to drop features that missing in more than 20% of observations (overall 58 features).

5.3 Dropping Unavailable Features

The dataset tracks customer information from the point a customer applies for a loan until the loan is fully repaid (or defaulted). Therefore, there are many features that are unknown to the bank before a loan is issued, such as the last payment amount, total payments made by the customer, and the number of times a customer paid late. Although these are very useful features to predict loan defaults, we dropped them because the bank does not have access to them at loan origination.

5.4 Feature Selection

One of our group mates formerly worked as a credit analyst. Therefore, feature selection was done relying on the domain knowledge, prior studies and the relationship between features and the dependent variable.

Obviously, higher annual income means a borrower is less likely to default, while the presence of delinquencies in the credit file is a red flag. In addition, we defined a function to quantify and visualize the relationship between every feature and the dependent variable (loan_status). For example, the relationship between the interest rate and loan status is shown below.

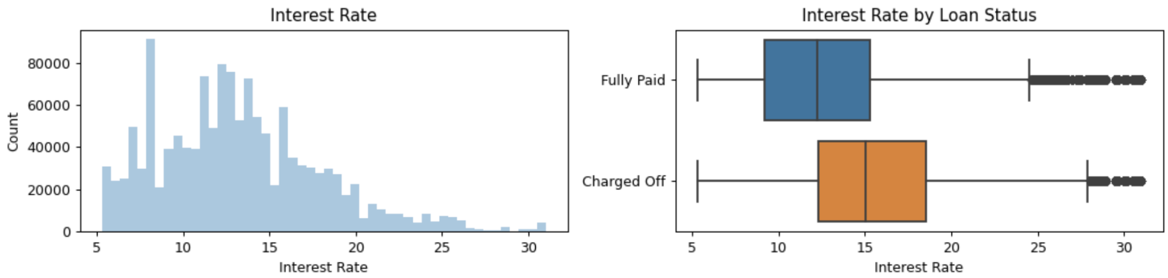

Figure 2: Relationship between Interest Rate and Loan Status

As the graphs show, a higher interest rate is associated with loans that have a higher risk. We can also see the relationship between the subgrade and loan status.

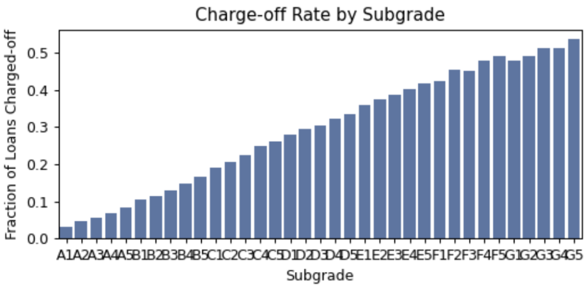

Figure 3: Relationship between subgrade and loan status

We explored all relationships between remaining features and the loan status to decide whether to keep them or not. After doing so, the following features were included into our models:

Chosen variables	**Variable Description**
annual_inc	The self-reported annual income provided by the borrower during registration.
delinq_2yrs	The number of delinquencies in the borrower's credit file in the past 2 years.
dti	A ratio of total monthly debt payments to self-reported monthly income.
emp_length	Employment length in years.
grade	Grade assigned to the borrower by LendingClub.
home_ownership	Categories: Rent, Own, Mortgage, Other.
int_rate	Interest rate suggested by the LendingClub's interest-rate model.
loan_amnt	The listed amount of the loan applied for by the borrower.
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due.
purpose	Purpose of the loan (educational, wedding, home improvement).
term	The requested loan term in months.
total_acc	The total number of credit lines currently in the borrower's credit file.

Apart from cleaning the above variables, we created a column fico_range_avg that is an average of fico_range_high and fico_range_low, upper and lower boundaries for the credit score provided by a credit bureau.

Since the data was gathered over 11 years, we did not fill missing values with mean, median or other values for 2 reasons. Firstly, the objective of our research is to analyze the performance of different models and to identify the best one. Filling missing values using some method will introduce unnecessary bias to results. Secondly, wide availability of data eliminates the need for incomplete observations. Therefore, we dropped observations with missing values and our final dataset contained 1,224,832 rows.

5.5 Create dummy variables

When the independent variable was a multi-category variable, we created a dummy variable for each category. In a regression model, each dummy variable gets an estimated regression coefficient, and it shows the difference between categories. Also, it makes the regression results easier to interpret.

5.6 Accounting for Class Imbalance

In real life, the number of people who repay their loans on time is higher than that of defaulters. This is also reflected in our dataset, and the ratio of the former to the latter is about 4:1. To account for this, we used over-sampling of default cases in the logistic regression and 80-20 stratified sampling in other classifiers (with the class weights parameter).

Our final set of independent variables contained 34 columns (14 when PCA was applied to keep 95% of the variation). We found that XGBoost classification worked best with the reduced dimensionality, while the PCA deteriorated performance for regression models.

6 Methodology

Loan Default Prediction

We need to create a binary classification model. We used Logistic Regression, Random Forests and XGBoost algorithms to create models which predict if a customer is likely to default or not. The models have been tuned and trained on the metric recall. Cross validation is used to measure model performance followed by selection. Grid Search is used for efficient hyper parameter tuning.

Training Features : ”annualinc”, ”delinq2yrs”, ”dti”, ”emplength”, ”grade”, ”homeownership”, ”installment”, ”intrate”, ”loanamnt”, ”loanstatus”, ”numacctsever120pd”, ”purpose”, ”term”, ”totalacc”, ”totalpymnt”, ”ficorangehigh”, ”ficorangelow”

Target Feature : ’loanstatus’(Boolean)

The categorical features are converted using One Hot Encoding

Logistic Regression

The parameters which are tuned for Logistic Regression are penalty(Regularization), C(Inverse Regularization), max_iter(number of iterations for convergence) and the Solver(Algorithm). The Solver used is LibLinear.

param_grid =’classifierPenalty’ : [’l1’, ’l2’],’classifierC’: np.logspace(-4, 4,20),’classifiermaxiter’=np.arange(100,200,25)

GridSearch is run on the above hyperparameter space to find the best model.

Random Forests Classifier

The parameters which need to be tuned for Random Forests are number of trees, maximum depth, no of features per estimator, samples required for leaf node and split.
’bootstrap’: [True, False], ’maxdepth’: [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None], ’maxfeatures’: [’auto’, ’sqrt’], ’minsamplesleaf’: [1, 2, 4], ’minsamplessplit’: [2, 5, 10], ’nestimators’: [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]

GridSearch is run on the above hyperparameter space to find the best model.

XGBoost Classifier

The parameters which need to be tuned for Random Forests are number of trees, maximum depth, no of features per estimator, samples required for leaf node and split.

'nestimators':[100, 125, 150, 175, 200],'maxdepth':[1, 3, 5, 7, 9],'learningrate':[0.01, 0.05, 0.1, 0.15], 'colsamplebytree':[0.1, 0.5, 0.7, 1],'subsample':[0.1, 0.3, 0.5, 0.7, 0.9, 1],'alpha':[0, 0.01, 0.05, 0.1], 'lambda':[0.05, 1, 1.5]

GridSearch is run on the above hyperparameter space to find the best model. The parameters for the best model are subsample=0.7, nestimators=200, maxdepth=7,learningrate=0.01, reglambda=1.5, colsamplebytree=0.7, alpha=0.01, scaleposweight=3

Model	**R**ecall
Logistic Regression	0.72
Random Forests	0.78
XGBoost Classifier	0.91

Recovery Rate Prediction

We need to create a Regression model to predict the recovery rate(100 * total pymnt/installment*term) for potential customers to estimate the risk and assign interest rate. We have used Linear Regression, Ridge Regression, Lasso Regression and Polynomial Regression to create models which predict the loss in case the customer defaults. The models have been tuned and trained on the metric Mean Squared Error. Cross validation is used to measure model performance followed by selection. MinMax Scaler and Standard scaler performance is checked to decide on the scaling method

Training Features : "annualinc", "delinq2yrs", "dti", "emplength", "grade", "homeownership", "installment", "intrate", "loanamnt", "loanstatus", "numacctsever120pd", "purpose", "term", "totalacc", "totalpymnt", "ficorangehigh", "ficorangelow"

Target Feature : 'recoveryrate'(float)

The categorical features are converted using One Hot Encoding

Linear Regression

Min Max Scaler is shown to give better results on Linear Regression. To remove the influence of noise PCA is performed to allow 95% variance and it is seen than model performance has dropped. So PCA is not performed.

e+2 and e-13 are the max and min order of coefficient values. The model does not seem to overfit a lot but to check regularization is applied in the next steps

The regression model is shown to give a square root of Mean Square Error of 15.2

Linear Regression with Regularization

Lasso-L1(absolute penalty) and Ridge-L2(Squared penalty) regularization methods are applied to see if the model performance improves.

The alpha values for L1 and L2 regularization are tuned using gridsearch and finding the best model.

The tuned alpha value for L1 is 0.0038.

The tuned alpha value for L2 is 0.1.

e+2 and e-2 are the max and min order of coefficient values.

The regression model with the best regularization technique(L2) is shown to give a square root of Mean Square Error of 15.19.

Polynomial Regression

Polynomial regression is done by creating polynomial coefficients of degree 2 in for the data set. For 40 columns the number of possible polynomial coefficients will be 780 which is not computationally feasible. So

to create a more computable model Principal Component analysis is performed saving 90% of the variance, which translates to 10 features. After extracting polynomial coefficients of degree 2 the number of columns is 45. A linear regression model is applied to predict the recovery rate.
The polynomial regression(with PCA) model is shown to give a square root of Mean Square Error of 24 which is a deterioration of previous model performance.
e+2 and e-2 are the max and min order of coefficient values, so no regularization is needed.

Model	**R**oot Mean Squared Error
Linear Regression	15.20
Lasso Regression	15.21
Ridge Regression	15.19
Polynomial Regression	24.00
XGBoost Regression	16.67

By comparison Linear Regression with L2 regularization is performing better than other models.

7 Evaluation method

For a bank, avoiding a default is far more important than missing a customer that will ultimately turn out to be creditworthy. The reason is a default usually entails weeks of legal proceedings and extensive reporting (especially to the Central Bank and Credit Bureaus). Therefore, we used recall as an evaluation metric for our classification models. In contrast, the performance of regression models (the OLS, Random Forest regression, XGBoost regression) was judged considering the Mean Squared Error (MSE) they produce.

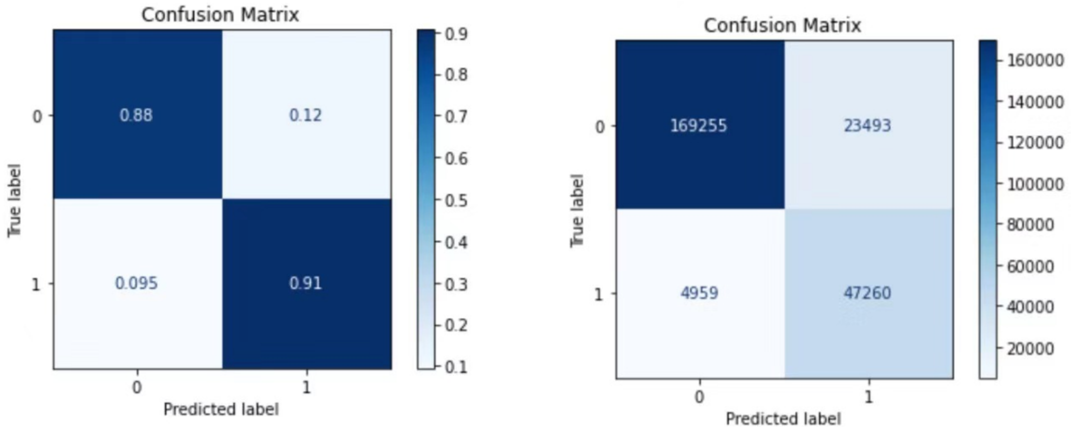

Figure 4: Classification results

As graphs illustrate, if LendingClub had trained XGBoost on the 80% of the data it had, then using this model on the other 20% of the loan applications could have brought significant savings for the company. It is true that the company could have missed 23,493 good loans. However, the company could have avoided lending 47,260 bad loans (91% of the default cases). When one considers that LendingClub earned, on average, 12% interest income on good loans and lost 59% of the lent amount on bad loans, he/she can see that using machine learning models for loan default prediction could have been very beneficial for this financial institution.

Also, the company could have estimated, on average, its recovery rate on potential problematic loans within 15% range. This could help the company's risk managers better manage the company's cash flows and estimate the approximate losses that LendingClub would incur in the near future.

8 Conclusion

XGboost Classifier showed the best performance among the classifiers tested for predicting if a potential customer is likely to default with a Recall Score of 91%

Linear Regression with L2 Regularization is the best among the regression models for predicting the loss given default for a potential customer with a RMSE of 15.19

Future researchers can experiment with CAT Boost and deep learning to achieve better recall and MSE in tasks 1 and 2, respectively.

9 References

- Lending Club Data (2018). Loan Default Dataset. Kaggle. Available from https://www.kaggle.com/wordsforthewise/len
club?select=accepted$_2$007$_t$o$_2$018*Q*4.*csv.gz*[*Accessed*23*December*2020].*Odegua*, *R*.(2020).*PredictingBankLoanDefault*
//*arxiv.org/abs*/2002.02011[*Accessed*23*December*2020].

- Wang, Y. and Priestley, J.L. (2018). Binary classification on past due of service accounts using logistic regression and decision tree, Grey Literature from PhD Candidates. Kennesaw State University. Available from https://www.researchgate.net/publication/317031021 [Accessed 23 December 2020].

- Zhou, L. and Wang, H. (2012). Loan Default Prediction on Large Imbalanced Data Using Random Forests,
TELKOMNIKA Indonesian Journal of Electrical Engineering, 10(6), 1519-1525. Available from https://www.researchgate.net
ValidatingLoan−GrantingDecisionsandPredictingDefaultRatesonConsumerLoans.TheReviewofBusinessInformation
84.*Availablefromhttps* : //*clutejournals.com/index.php/RBIS/article/view*/4563/4654[*Accessed*23*December*2020]

- https://www.investopedia.com/terms/l/lossgivendefault