Names: Kristy Bell (kab7kr), MacKenzye Leroy (zuf9mc), Yayi Feng (yf7qq)

DS 5110 Final Project: Predicting Late Loan Payments/Loan Default with Lending Club Data

I. Abstract

Lending money to strangers is inherently a risky decision. As a lender, you risk receiving late payments and being out of money for a short period of time, or worse, default by the borrower. There exists ways to mitigate lending risk by offering high interest rates or supplying loans with collateral terms. It is critical, therefore, that lenders understand the risk associated with offering a loan to a specific borrower, so that they can write adequate terms to reduce their exposure to risk. For our final project, we use attributes from 2,925,494 peer-to-peer loans provided through the LendingClub platform, to try to predict loan non-payment (late past the grace period, or loan default). Using the *ml* package in *pyspark*, we constructed four models including logistic regression, random forest, gradient boosted tree and linear support vector classification. We were able to predict loan non-payment with an Area Under the Curve of 0.95, 0.96, 0.97, and 0.95, respectively. Based on feature importance scores from the random forest classifier, it was evident that the borrower attribute that is most predictive of loan non-payment was FICO score.

II. Background

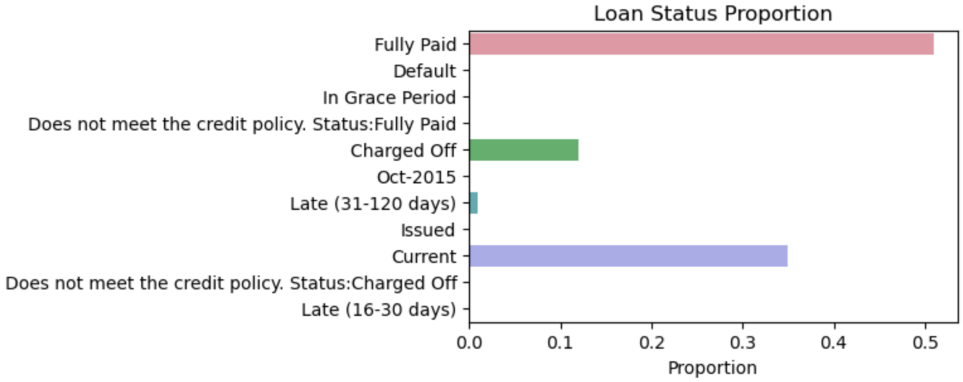
LendingClub is the world’s largest peer-to-peer (P2P) lending platform, and moreover, it was the first peer-to-peer lender to register its offerings as securities with the SEC and offer loan trading on a secondary market. LendingClub provides P2P lending, where investors are responsible for their own decisions on lending. Although there are many borrowers on this platform that meet the minimum requirements for receiving a loan, there still exists a large opportunity for risk of non-repayment. The objective of this project was to build predictive models to determine whether a borrower would default on a loan, or miss a payment, based on particular features and potentially assist lenders in making more informed decisions.

The dataset was obtained from Kaggle, which was originally collected from the LendingClub website. This dataset contains 2,925,494 loan cases, ranging from 2007-2020 including 142 variables. The personal information of borrowers were collected, such as their income, credit score, debt information, employment, etc. The dataset also includes other crucial information regarding the loan such as loan grade, loan amount, interest rate, etc. Our goal was to predict the loan status based on given features of the loan and borrower.

III. Data & Methods

Response Variable

The original data contained a variable entitled “loan_status” which had 11 different categories visualized in Fig 1. The “Oct-2015” was determined to be a data entry error with only one observation in that category; therefore, that observation, along with all null values were filtered out prior to grouping categories. Rather than deal with a multi-class classification problem with a severe imbalance between categories, we decided to combine categories into two classes: loan non-payment (1) or loan payments are current (0). The late money class consisted of the following categories from the original “loan_status” variable: default, charged off, late (31-120 days), and late (16-30 days). There remained a class imbalance after building a new response variable with 13% of loans having late payments/default, and 87% of loans being current.



Short money?	Count	Perc
Yes (1)	382614	13%
No (0)	2530788	87%

Fig 1. Distribution of original response variable, and the new response variable after class condensing.

Feature Description

After parsing through the 142 columns in the LendingClub dataset, we curated a list of categorical and numeric attributes that a lender would know prior to issuing a loan. Many of the selected variables required type casting to update the spark dataframe schema. We felt that the following categorical features may be predictive of loan default/late payment: purpose of the loan, term (36 month or 60 month), borrower’s employment length, borrower’s home ownership, loan verification status, and subgrade which is a quality score of a loan based on borrower’s information such as credit score. Interestingly, the purpose of most of the P2P LendingClub loans was debt consolidation, and nearly 50% of borrowers have mortgages while roughly 40% are renters (Fig 2A-B). There were a wide range of subgrades in this dataset, with roughly 70% and 30% being 36-month terms and 60-month terms, respectively (Fig 2C-D). Although the employment length variable was ultimately dropped due to excessive null values, we also noticed that most borrowers were young (employed for less than 10 years). All categorical features were one-hot encoded by first using the *StringIndexer*, then the *OneHotEncoder* in the *pyspark ml* package.

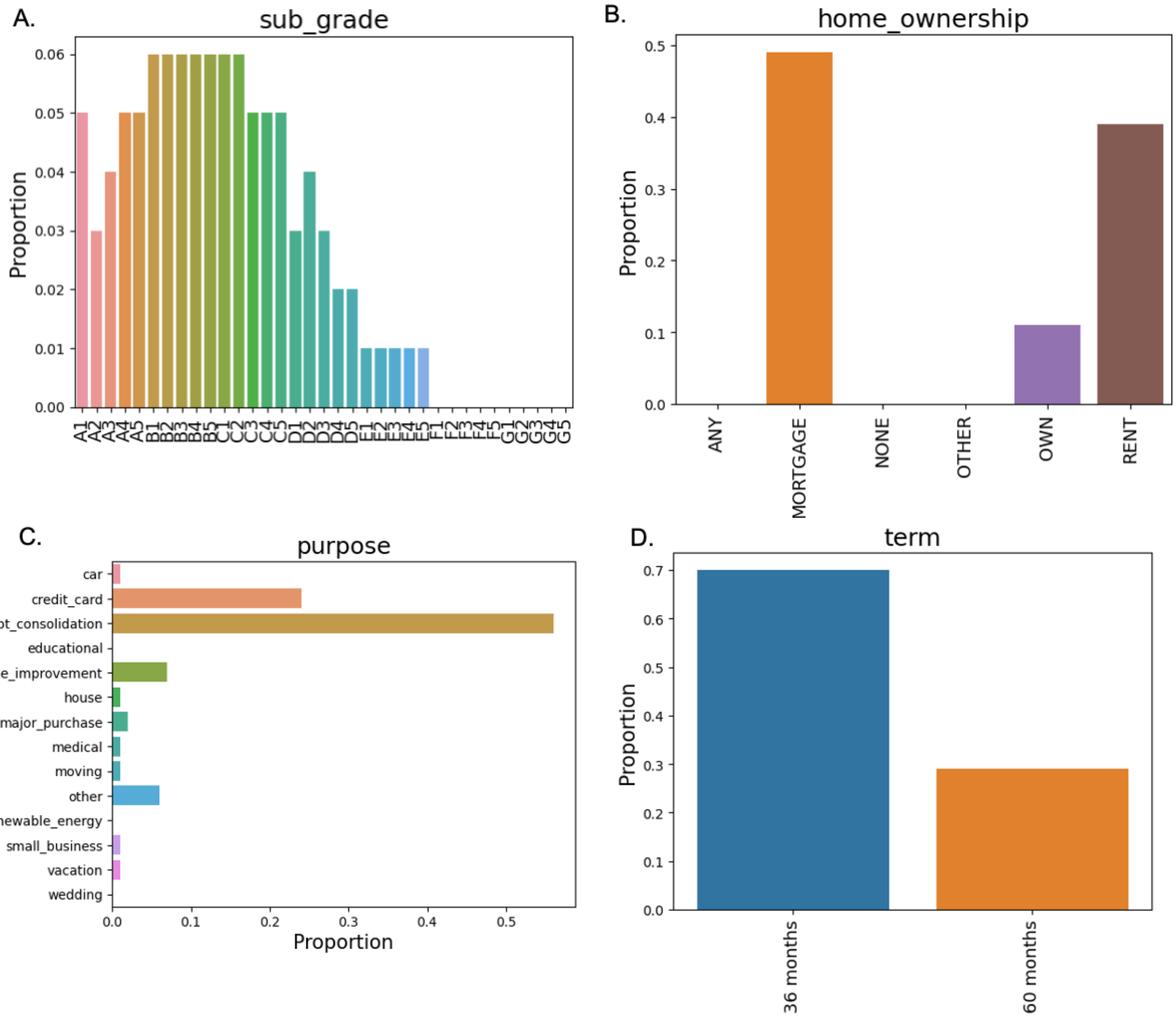


Fig 2. Distribution of a subset of categorical features including loan subgrade (A), borrower home ownership status (B), loan purpose (C), and term of the loan (D).

Numerical attributes that we suspected would relate to loan non-payment include loan amount loan, debt to income ratio (DTI), annual income, FICO credit score, total payment, funded amount, funded amount invested, and revolving balance. Many of the numerical variables were heavily skewed right such as DTI and annual income (histograms not pictured). All numerical features were combined with the one-hot categorical vectors into a singular feature vector using the *VectorAssembler*, then standardized using the *MinMaxScaler* in the *pyspark ml* package.

Model Building & Hyperparameter Tuning

Four models were constructed for this investigation using the *pyspark ml* package: logistic regression (LR), random forest (RF), gradient boosted tree (GBT), and linear support vector classification (LSVC). For each model, a variety of different parameters were tested using the *ParamGridBuilder* and 3-fold cross validation to select the best combination of parameters for each machine learning algorithm. Using the best parameters for each model as measured by AUC from the 3-folds, the final three models were built and compared by their performance on a holdout dataset (Fig 6). Due to the data imbalance, metrics such as Area Under Curve (AUC) as well as sensitivity were used to select the best final model. In addition, we examine the feature importance attribute from the random forest and gradient boosted tree models to examine which variables are most predictive of loan non-payment.

IV. Results

Logistic Regression Model

Since our problem was a binary classification, we started with a logistic regression model as a baseline. After some hyperparameter tuning, we were able to achieve an AUC of 0.95 (Fig 3) with a Ridge Regression regularization of .5 and max number of iterations of 10. While this is a respectable AUC, when we look closer we realize that this model even after hyperparameter tuning was simply selecting the majority class each time giving it an accuracy of 0.87 and specificity of 1, but a sensitivity of 0. Sensitivity is able to provide us with information about the ratio of true positives (actual loan cases predicted as 1s) to all the actual loan cases, which is an important metric to evaluate since we have an issue with class imbalance. Since our initial goal was to identify risky loans to avoid late payment or default, this would be problematic for lenders in terms of helping them to detect whether a particular loan would be riskier than others.

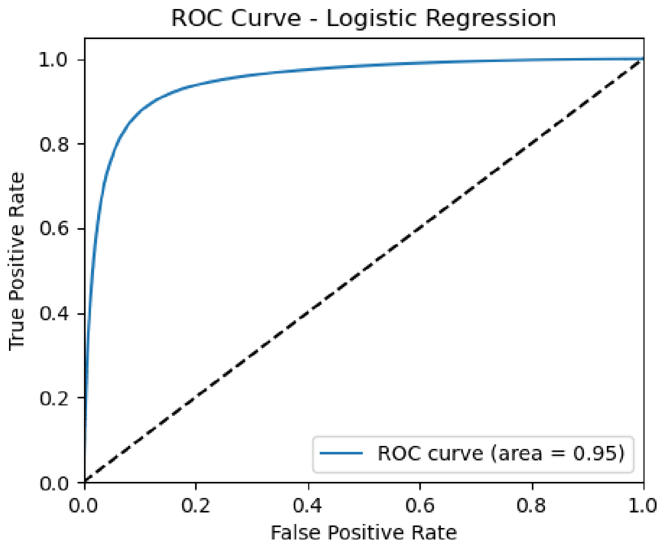


Fig 3. ROC Curve for optimized Logistic Regression. maxIter = 10, regParam = .5, elasticNetParam = 0 (L2/Ridge)

Random Forest Classifier

The next model we explored was a Random Forest. Our initial Random Forest model was able to achieve 92% area under the ROC without any prior hyperparameter tuning. However, the sensitivity of this model was very low (0.02), perhaps due to the issue of class imbalance because there are more non-default loan cases (0s) than default loan cases (1s). Thus, we proceeded to implement hyperparameter tuning to improve this issue. Due to memory constraints, the numTrees and maxDepth parameters were only tested for the range from 5 to 15, and best parameters after running three-fold cross validation were number of trees = 10 and max depth = 15, and the associated area under the ROC was 96%. Fig 4 below illustrates the plot of the ROC Curve of the model predictions with the best hyperparameters:

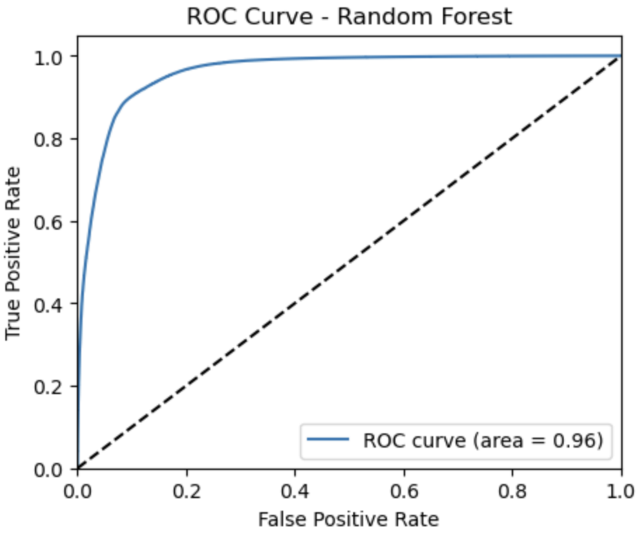


Fig 4. ROC Curve for optimized Random Forest. numTress = 10 & maxDepth = 15

The sensitivity for this model was much higher compared to the previous model (0.71), which means that it was able to predict 71% of the actual default loan cases as 1s out of all default loan cases proving it to be a much better solution to our initial problem than the logistic regression. Another advantage to using a Random Forest model was we were also able to explore which features were most important in our model. We found that FICO scores were by far the most important with a feature importance of 0.717.

Gradient Boosted Trees (MacKenzye)

The optimized Gradient Boosted Tree (GBT) was our best performing model in terms of Area under the ROC Curve (Fig 5) as well as accuracy, slightly edging out the Random Forest in both. Further, much like the Random Forest and unlike the Logistic Regression this model not only did well identifying non-risky loans, but also potentially risky loans boasting a sensitivity of 0.78 (Fig. 6). Our final optimized model featured a maxDepth of 10 and maxIter of 15. Similar to the Random Forest model, we explored the feature importances and again found FICO score to be the most important feature by far with a feature importance of 0.631.

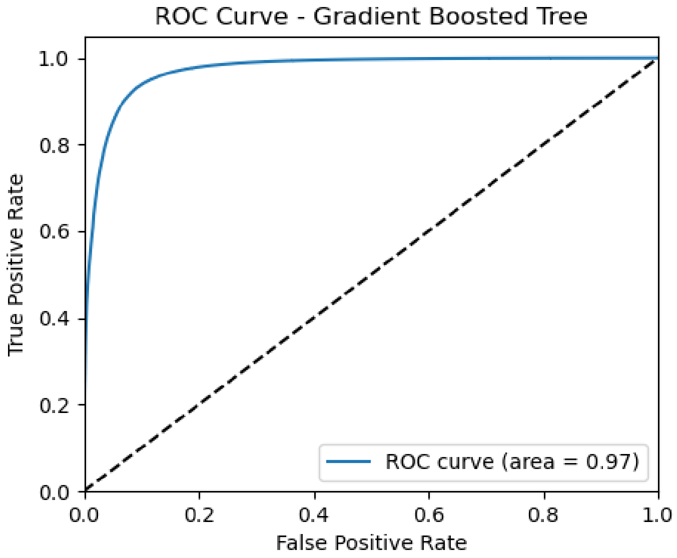


Fig 5. ROC Curve for optimized Gradient boosted Tree. maxIter = 15 & maxDepth = 10

Linear Support-Vector Classifier

The final model we fit was a Linear Support Vector Classifier (LSVC), which performed very similarly to our logistic regression model as seen in Fig 6. Like our other models we performed 3-Fold Cross Validation to find optimal hyperparameters for our LSVC. Much like the Logistic Regression though, the optimized LSVC defaulted to choosing the majority class each time resulting in an accuracy of .87 with a specificity of 1 but a sensitivity of 0. Much like the Logistic Regression, this model completely fails to identify the riskiest of loans-our initial goal. Further, with the logistic regression model, we could at least theoretically tinker with thresholds to increase our sensitivity at the cost of our specificity, but since LSVC does not output probabilities in PySpark, that's not an option of the LSVC making it on of the least practical models for our original goal of predicting risky loans. This also meant we couldn't plot a ROC curve for this model but we suspect it would look quite similar to the logistic regression.

Model	LR	RF	LSVC	GBT
AUROC	0.95	0.96	0.95	0.97
Accuracy	0.87	0.95	0.87	0.96
Sensitivity	0	0.7	0	0.78
Specificity	1	0.99	1	0.98
Precision	0.09	0.9	0.5	0.87

Fig 6. Final comparison of all models in terms of AUC, Accuracy, Sensitivity, specificity and precision

V. Conclusions

In conclusion, our Gradient Boosted Trees model had the best performance of all four models in terms of AUC, Accuracy and Sensitivity. In general, the tree models (Random Forest and Gradient Boosted Trees) had much better performance than the logistic regression and linear support vector classifier. With these tree-based models we were also able to explore feature importance and found FICO score to be the most significant in predicting the loan status in both models. Given more time and computational power, we would improve model performance by testing more hyperparameters to further tune our models, experiment with downsampling and upsampling the dataset to combat the issue of class imbalance, and explore a larger range of features.