Lending Club Default Risk Prediction

Introduction

The Lending Club is a peer-to-peer lending service that allows lenders to buy portions of loans with an interest rate and loan amount that has been set. They provide historical loan data that includes features about the borrower, as well as the loan status. Here, I build a prediction system that takes a vector of features about a particular borrower and returns a probability of default. This method is intended to be an initial estimate of loan risk and can be extended to questions about what the loan interest rate and funded amount should be.

Methods

Cleaning, normalization, and feature analysis

Prior to modeling, all data were cleaned and normalized. The cleaning process was applied to (1) reduce problems of sparseness and (2) handle missing data. The data were first randomly split the data into training, validation, and test sets (70%, 10%, and 20%). All imputation and normalization procedures were learned on the training set and applied to the other sets. All feature analysis and selection was performed using the training set.

With respect to sparseness, the zip code and state address data were too diverse to be useful for modeling. To address this problem, I utilized IRS zip code data to assign regional economic statistics such as average incomes and average unemployment compensation to each loan. This substantially reduced the dimensionality of the data.

With respect to missing data, I applied mean imputation for most features where the data could be characterized as missing at random. However, many of the time related features (e.g., months since last default) had missing values that actually conveyed information about the borrower. That is, having no entry for “months since last default” means that there wasn’t a default. To account for this, I quantized the “months since” values into ten bins, based on the percentiles of the data. I then created new categorical variables for each bin. This allowed me to represent the lack of default, in this example, and preserve the information that the missing value conveyed.

After cleaning, I normalized the data by z-scoring each feature, setting the mean feature value to zero and the standard deviation of the feature values to one. This puts all of the numeric features on the same scale and leads to more stable model fits.

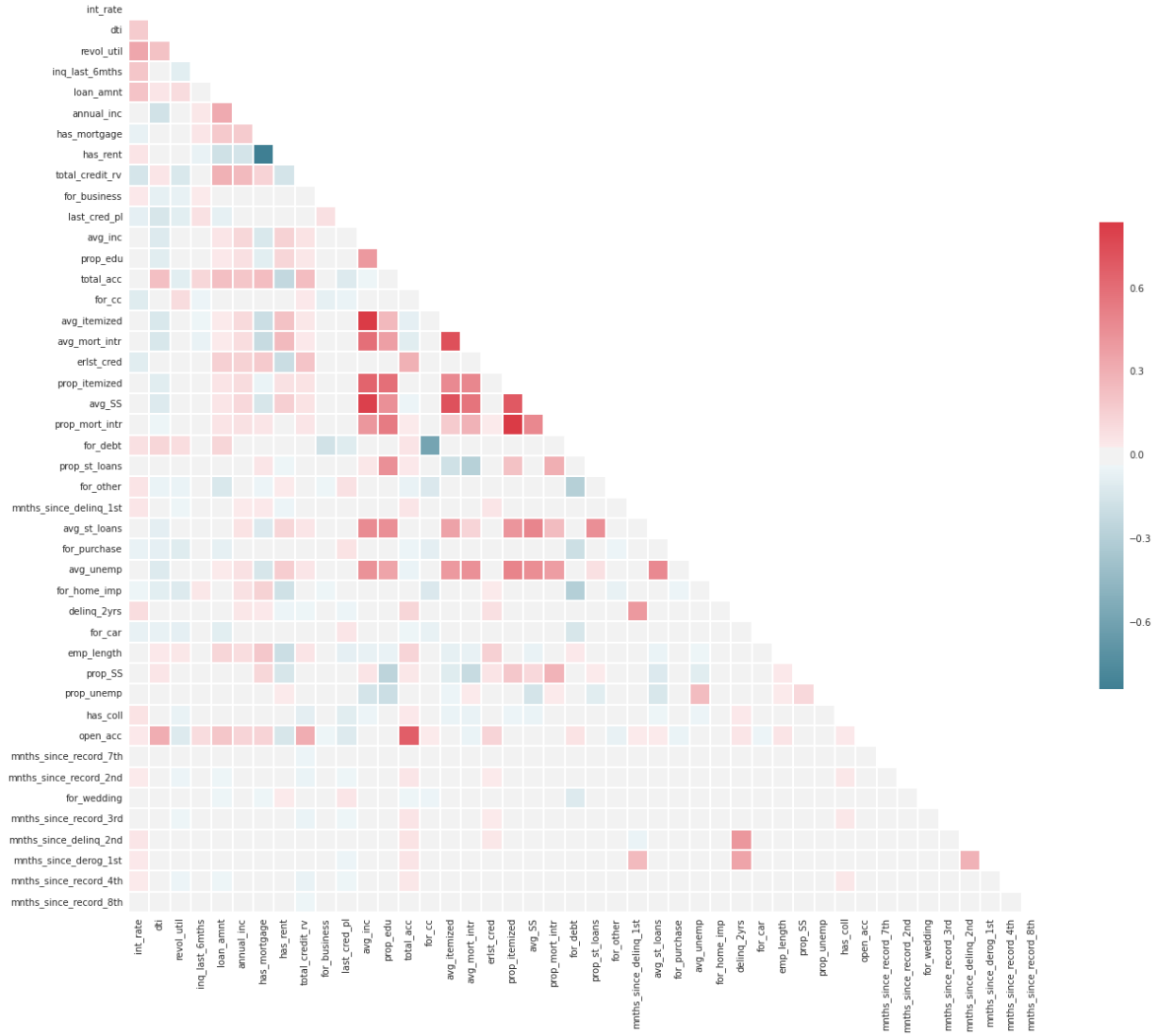
Once the data were cleaned and normalized, I analyzed the relationship between each feature and whether or not the borrower defaulted on the loan. For each feature, I used Cohen’s d as a measure of effect size and determined if the relationship was reliable at the 0.05 level, having applied a Bonferroni correction for multiple comparisons. Table One shows the F scores, reliability, and Cohen’s D for the five features with the largest effect sizes. Cohen’s D values are all in the modest to low range and only statistically reliable features were included in this analysis.

Table One

Feature	F Score	Reliable	Cohen's D
dti	2144.12	True	0.20
revol_util	1698.80	True	0.18
inq_last_6mths	667.88	True	0.10
annual_inc	521.50	True	0.09
has_mortgage	400.67	True	0.08

After running statistical tests and effect size analyses, I measured the correlations between features that had a reliable relationship with default. Figure One shows the correlation matrix for each feature by each feature. Most of the features are not reliably correlated. However, reliable correlations do exist between the variables in the IRS zip code data set. This is not surprising, as those features all serve as proxies for the economic health of the region.

Figure One



Once the features were cleaned, normalized, and selected based on reliability, I proceeded with the same model fitting procedure for all models considered. First, models were fit to the training set, adding one feature at a time from the largest Cohen's D to the smallest. Models were fit using 5-fold cross-validation and then used to predict the validation set. I measured performance using the area under the ROC curve for the classification task. For each model, I selected the number of features that yielded the best validation set performance. After selecting two model classes, a decision tree and a stacked ensemble, I analyzed performance on the test set.

To analyze test set performance, I used two metrics. The first metric was the area under the ROC curve. For the second metric, I looked at the return on investment (ROI) as a function of loan rejection rate for each model. I used each model to predict the probability of default for every loan in the test set and rejected the top X% of loans deemed to be the riskiest. Then I calculated the ROI on a $10,000 investment in the hypothetical portfolio of loans. I compared this to a portfolio in which X% of loans were randomly rejected. Note that if the model were perfect and only rejected loans that would default, the ROI on a $10,000 would be $1257.84.

Decision Tree Methods

The decision tree classifier was trained using a cost function that minimized the gini impurity for new splits. The maximum depth was set to six. However, a minimum of five samples was required for creating a new leaf in the tree.

Stacked Ensemble Methods

The stacked ensemble consists of a two-stage modeling process. In the first stage, I trained N classifiers using 5-fold cross-validation. For each classifier, and for each fold, I saved the predicted default probabilities for the held out fold. Those predicted probabilities were then used to train the second-stage model. For the second stage, a logistic regression classifier was trained on the probability. This enabled me to utilize a linear weighting of the stage-one classifiers that minimized mean squared error between the estimated class and the true class over the training data.

For the first stage classifiers, I trained two Random Forest classifiers, two Extra Tree classifiers, two AdaBoost classifiers, and one Gradient Boosting classifier. For the two-classifier pairs, one was trained with an entropy minimizing cost function while the other was trained with a gini impurity minimizing cost function. Both the Random Forests and the Extra Trees consisted of 100 estimators and had no limitation placed on the maximum depth. However, a minimum of five samples was required for creating a new leaf in any tree. For the AdaBoost classifiers, I used 100 estimators that took the form of decision trees with a maximum depth of three. The Gradient Boosting classifier was given 50 estimators with a maximum depth of six.

Results

Decision Tree Methods

Figure Two shows the ROC curve for the decision tree's performance on the test set. The ROC curve plots the true positive rate as a function of the false positive rate. The area under the curve (AUC) is a measure of the classification accuracy that is minimally biased by the imbalance in the classes. Additionally, the ROC curve can be

used to understand the trade-offs between increasing the true positive rate and the false positive rate.

The AUC is 0.68, which is on par with other publicly available risk models for this dataset. That said, I for this analysis, I have excluded the Lending Club's loan grades, the interest rate, and the loan amount. These features are highly predictive of default, but my goal is to develop a risk model that is independent of the Lending Club. Therefore, those features would provide an overly optimistic view of the AUC.

Figure Two

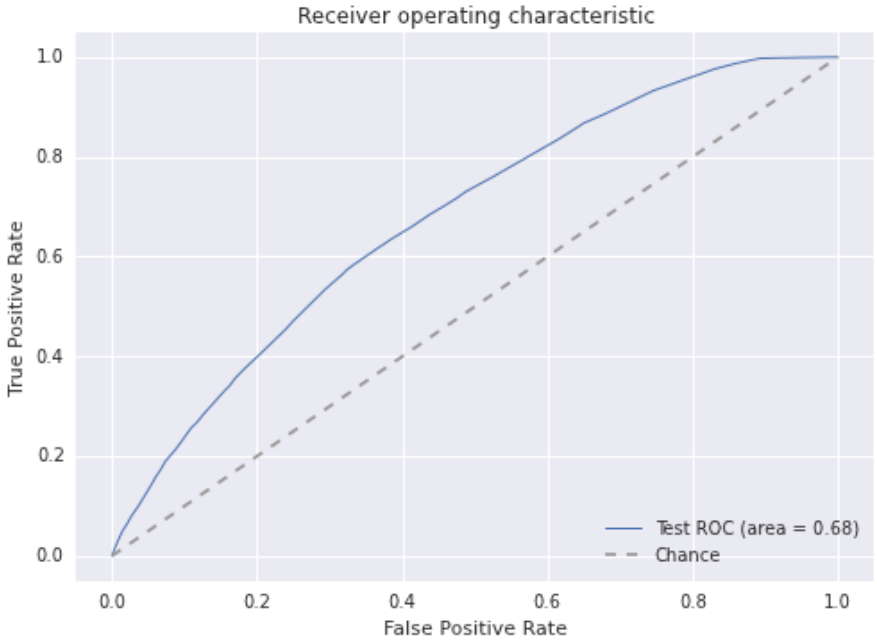


Figure Three shows the ROI on a $10,000 investment as a function of the loan rejection criterion, where loan criterion X means that loans are rejected if they are in the top X% with respect to risk. The green curve shows the model-based loan rejection, while the blue curve shows random rejection. The error bars represent the 95% confidence interval of the mean, estimate via bootstrapping.

As a starting point for the rejection rate, I will consider 15%. This may be an overly strict rejection criterion, but it is a reasonable point from which to compare the models and is still below the base default rate of 18%. It is important to note that risky loans will be part of any portfolio that expects to make substantial returns, as returns are strongly correlated with risk. For a 15% rejection rate, the ROI on a $10,000 investment was $1052.04. This is reliably larger than the chance rejection ROI of $1033.99. Note that perfect rejection would yield $1257.84.

I do not recommend including either the interest rate or the loan amount as features in any modeling, as both features are set using the Lending Club's internal risk metrics. These risk calculations should be used to set interest rates and loan amounts,

not make predictions from them. However, if loan amount were included, the AUC for the decision tree would increase to 0.69 and the ROI on a $10,000 investment at a 15% rejection rate would be $1056.78. If both loan amount and interest rate were included, the AUC would increase to 0.74 and the ROI would decrease to $1052.26. This reflects the fact that this model rejects loans with higher interest rates and loses some profit as a result of false positives.

Figure Three

One advantage of the decision tree is that it permits the analysis of feature importance. Specifically, features importance can be assessed on the basis of how much each feature is able to reduce the gini impurity. For this tree, the most important features are how long ago the last credit pull was, the debt-to-income ratio of the borrower, the annual income of the borrower, the revolving credit utilization of the borrower, the total credit of the borrower, and the number of credit inquiries in the last six months. These are features are often reported as important in the risk modeling community.

Additionally, whether or not the loan is for a business, if the borrower is a renter, and if the borrower has a mortgage are also important features for this decision tree. The business finding is interesting and probably reflects the fact that small businesses are risky.

Stacked Ensemble Methods

Figure Four shows the ROC curve for the stacked ensemble classifier. The AUC for this classifier is 0.72. This is reliably higher than the decision tree, demonstrating that this is a stronger classifier. This is not surprising as the stacked ensemble is composed of trees with qualities similar to those of the decision tree model. Additionally,

the boosting and bagging operations of the stage one classifiers reliably lead to stronger classifiers.

Figure Four

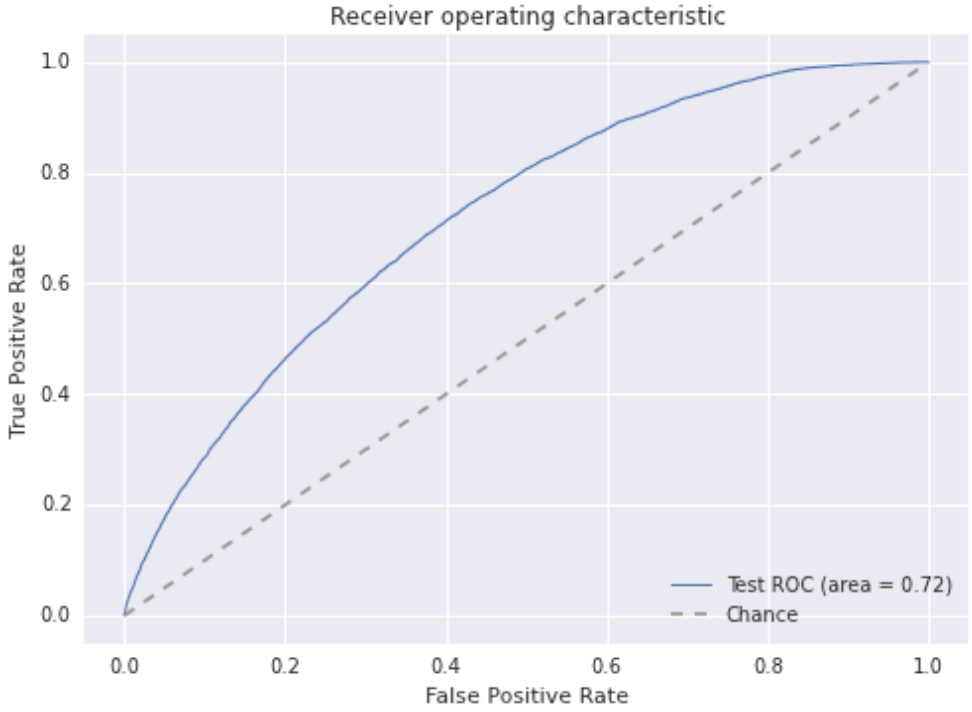


Figure Five shows the ROI as a function of the rejection rate for the stacked ensemble. For a 15% rejection rate, the ROI on a $10,000 investment was $1059.95. This is reliably larger than the chance rejection ROI of $1034.01. Note that perfect rejection would yield $1257.84.

As mentioned above, these analyses do not include the loan amount of the interest rate of the loan. If the loan amount were included, the AUC for the stacked ensemble would increase to 0.74 and the ROI on a $10,000 investment at a 15% rejection rate would be $1065.93. If both loan amount and interest rate were included, the AUC would increase to 0.76 and the ROI would decrease to $1064.74.
Whereas the decision tree lends itself to a simple understanding of the important features, the stacked ensemble is much less transparent. However, many of the same features are still important for reducing either entropy of gini impurity. That is, financial features about the borrower are strong predictors of default.

Figure Five

Discussion

These findings suggest that a stacked ensemble of trees is the best method for determining default risk. The AUC indicates of the strength of the stacked ensemble as a classifier relative to the decision tree. I conducted additionally analyses using logistic regressions, Naïve Bayes, and individual bagging and boosting approaches. The stacked ensemble is superior to all of those approaches with respect to AUC and should be utilized for risk prediction.

From a business perspective, the higher ROI of the stacked ensemble classifier is another good reason for implementing it. That said, the ROI analysis much come with several caveats. Essentially, this ROI analysis is highly idealized and should not be treated as a guaranteed profit prediction.

The first caveat is that the data used here have already passed through the Lending Club's risk models. As a result, I am looking at loans that have already been deemed profitable by that metric and am identifying its "mistakes." In that sense, the 15% rejection criterion discussed here may be overly conservative and potentially disruptive to the loan pool.

The second caveat is that the ROI analysis assumes that the interest rate is pre-determined. In reality, the default risk probability should be used to set interest rates and loan amounts. As a result, adjustments will need to take place in order to find the right interest rates to attract borrowers and remain profitable. While the ROI should give a sense of that, it is nevertheless idealized.

Future Research

As suggested above, future research should focus on how to translate these risk predictions into interest rates and loan amounts. While the stacked ensemble classifier

provides strong risk predictions, it does not tell us what the interest rate or loan amount should be. That is a different optimization problem.

Additionally, this model is limited by the set of features available to it. Due to the peer-to-peer nature of the Lending Club, privacy is a concern. As a result, credit scores, precise zip codes, and many features about credit reports are not available. These features are all potentially predictive of default risk and could make this model much stronger. Future work should focus on the relationship between those variables and default.

Finally, there are still additional data sources to consider. Fields such as loan description and job title were rendered useless by deprecation. Previous work suggests that those features do have predictive power. If those data sources can be exploited in the future then they could improve model performance and increase ROI.

Proposals

1) Adopt the stacked ensemble learner as your initial risk prediction method. This model is the strongest predictor and does not suffer from the potential issues of variance that the decision tree faces.

2) Begin collecting credit scores and precise zip codes. Geographical features do have predictive power in this area and that should be exploited. Additionally, credit scores have a rich history in this domain.