**Lending Club Loan Data:  
To default, or not, this is the question.**

Xiaohan Liu1, Weixin Liu2, Xinyu Xu3, Xuan Yin4, Haiyang He5

1Department of MSCS, University of Illinois at Chicago, Chicago, IL, 60607

2Department of Computer Science and Department of Physics, University of Illinois at Chicago, Chicago, IL, 60607

3Department of Computer Science, University of Illinois at Chicago, Chicago, IL, 60607

4Department of Economics, University of Illinois at Chicago, Chicago, IL60607

5Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL

60607

**Abstract:**

In this project, the data to be classified are the loan information of individuals from Lending Club. Our objective is to predict loan default status of a new loan applicant based on the given information. Thus, several classification algorithms, including Naïve Bayes (NB), Logistic Regression, Boosting and Random Forest are implemented to build classification models. The corresponding confusion matrix for each algorithm is obtained. The accuracy, precision, recall and F-score for each class are calculated. Based on our interpretation of the data, the F-score for the negative class is treated as the primary benchmark to evaluate the performance of the classifier and Random Forest achieves the best among all the classifiers.

**Keywords:** Lending Club, Naïve Bayes, Logistic Regression, Boosting, Random Forest

**1. Introduction**

In recent years, peer-to-peer lending platforms are becoming boomingly popular since they allow people to lend and borrow money conveniently and avoiding transaction costs which banks usually charge. Lending Club, the world’s largest online credit marketplace, has funded 22 billion dollars according to their statistics. They will assign different grades with corresponding interest rate for investors. However, high interest always accompanied with high risk. For example, according to the data from 2007-2012, for grade FG with an average interest rate 20.33%, only 61.41% of money is fully paid. And for grade A with an average interest rate 7.46%, 93.87% of the money is fully paid. So for investors, how to find those borrowers who are unlikely to default is the most important question [[[1]](#endnote-1)][[[2]](#endnote-2)].

In this report, our goal is to classify the loan status if given the information of a new borrower. We summarized these statuses and categorize them into two classes, TRUE or FALSE. We will utilize four binary classification models, Naïve Bayes, Logistic Regression, Model Based Boosting and Random Forest. We will use 5-fold cross validation to train and test on Lending Club’s data set. Since a loan default will cause both principal and uncollected interests losses, we want to focus on as high as possible default case recall rate. Furthermore, as the dataset is quite imbalanced, we will work on different techniques according various learning methods to overcome this problem. Finally, we will compare and discuss on performance of each classifier.

**2. DATA PREPARATION**

**2.1 Data Source**

Our original data is downloaded from Kaggle provided byLending Club. Following is the website: <https://www.kaggle.com/wendykan/lending-club-loan-data>.

**2.2 Data Preprocessing**

The original data’s dimension is 887383×75, which contains a lot of redundant and insignificant data as well as features. Thus, following preprocessing steps are carried out.

Step 1. Class definition. The loan status, which is our Y label, is redefined into True Class (those who are under “Current”, “Fully Paid”, “In Grace Period”, “Does not meet the credit policy. Status: Fully Paid”) and False Class (those who are under “Charged off”, “Late (31-120 days)”, “Late (16-30 days)”, “Default”, “Does not meet the credit policy. Status: Charged Off”) so that the original problem is simplified into a binary situation.

Step 2. Feature selection. The number of features is reduced in this step. We start by deleting all the goal-irrelevant features. For example, information for administrative purpose like “member\_id” and “issuded\_date”, “next\_due\_time” and “application\_type” are dropped. Columns like “grade” are deleted, since all the information is contained in subgrade. Secondly, features having the same values across all cases are dropped, like “policy\_code” and “pymnt plan.”. Features with more than 50% NA are discarded.

Step 4. Data cleaning. Rows with NA, including those have abnormal 999 and 9999 values, which statistically meant to represent NA, are removed. The “desc” feature, which contains the reason for the applicant’s application, is converted from string to the count of the string length.

Step 5. Numeric features conversion. Numeric features having high covariance (>0.95) as well as those who are responsible for off-the-roof (>100) VIF (Variance Inflation Factor) with caution are removed. Because some of our models are relative not sensitive to the correlations between features while the others are. We’d like to compare them as well on this front. Besides, all numeric data are normalized to 0-1 range for simplicity.

Additional treatments are performed for logistic regression and its boosting, all the categorical values are expended to design matrix of each sublevels with dummy variable to better cope with the package.

In the end, we have a ready-to-use input data with dimension 877,860 × 32 (877,860 × 141 for logistic regression and its boosting), the list of kept features are listed in Appendix A. The processed data is imbalanced, with approximately 95% true cases as dominating majority and 5% false cases as minority as shown in Figure 1.

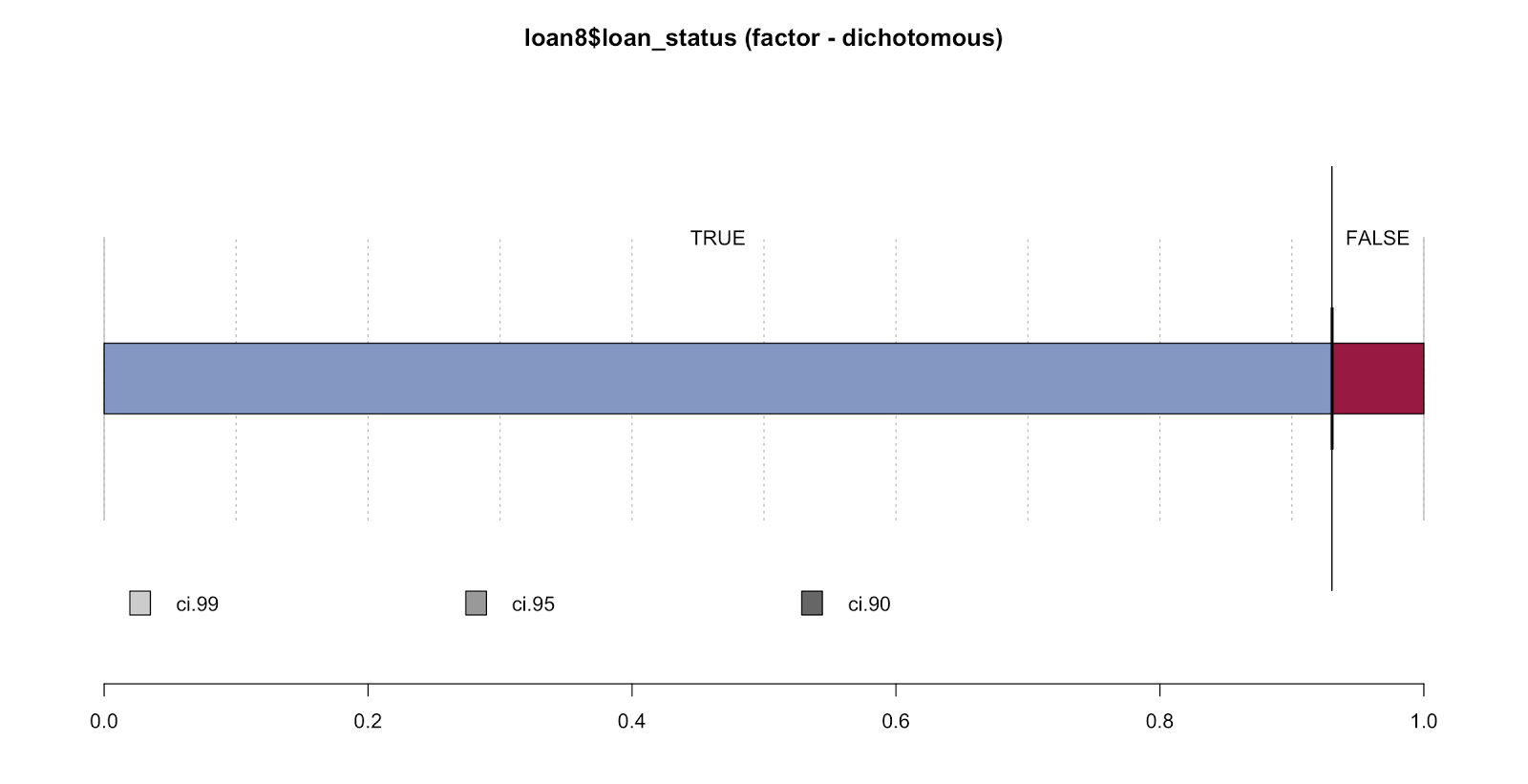


Figure 1. The distribution of true and false classes in the processed dataset.

**3 CLASSIFICATION ALGORITHMS**

Several classification algorithms are utilized in this project, as illustrated in this section. The algorithms are implemented with the help of R packages using Rstudio.

**3.1 Naïve Bayes**

Naive Bayes (NB) is a classic classification algorithm, which is effective in a lot of situations. However, it is sensitive to collinearity, two more features have to be deleted based on data interpretation. Because it assumes features are conditional independent given outcome, which makes it somehow unreliable. The classification results are shown in the table, the classifier performs well on the positive class but badly on the negative class, which might because of the high bias of the model [[[3]](#endnote-3)].

Table1. Classification results for Naïve Bayes

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Precision | 0.254 | 0.959 |
| Recall | 0.524 | 0.759 |
| F-score | 0.342 | 0.847 |

**3.2 Logistic Regression**

Logistic regression is a fast and popular classifier in practice. Binary logistic regression estimates the conditional probability that a characteristic is present given the values of explanatory random variables:

For this case, Yi=1means the prediction that the borrower will pay back the loan, Yi=0 means the prediction that the borrower will default.

We employed logistic regression with L1 regularization (LASSO). By L1 regularization, we could choose the features more smartly by avoiding overfitting. Logistic regression is sensitive to the imbalanced distribution of data.

As shown in Table 2, under default setting where cut-off probability 0.5, which means a probability higher than 0.5 will be classified as positive, and a probability lower than 0.5 will be classified as negative, the performance for the false case seems dreadful though it works beautifully for the true cases. By theory, this discrepancy in prediction powers for two cases are caused by the imbalance of our data. Although a more popular way to address this issue is to over sample the minority groups until it is as comparable with the majority group, given the limited time and resources allocated, a tuning over the discriminative cut-off threshold is a more feasible strategy for us.

Table 2. Test results for Logistic Regression using 5-fold cross validation

|  |  |  |
| --- | --- | --- |
|  | Cutoff Prob. = 0.5 | Cutoff Prob. = 0.924 |
| Precision | 0.111 | 0.397 |
| Recall | 0.052 | 0.808 |
| F-score | 0.071 | 0.532 |
| Accuracy | 0.904 | 0.899 |

Thus, an ROC (Receiver Operating Characteristic) [[[4]](#endnote-4)] curve plotting is created, which is shown in Figure 2. ROC is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. Ideally, when the false positive rate is 0, the true positive rate is already 1. From the curve, the closet point to (0,1) is taken as our optimal performance point, which corresponds to a discrimination threshold equal to 0.924. It can be seen that without any loss on performances regarding the true cases and overall accuracy, there is a considerable improvement on the prediction regarding the false cases due to the tuning of cutoff threshold.

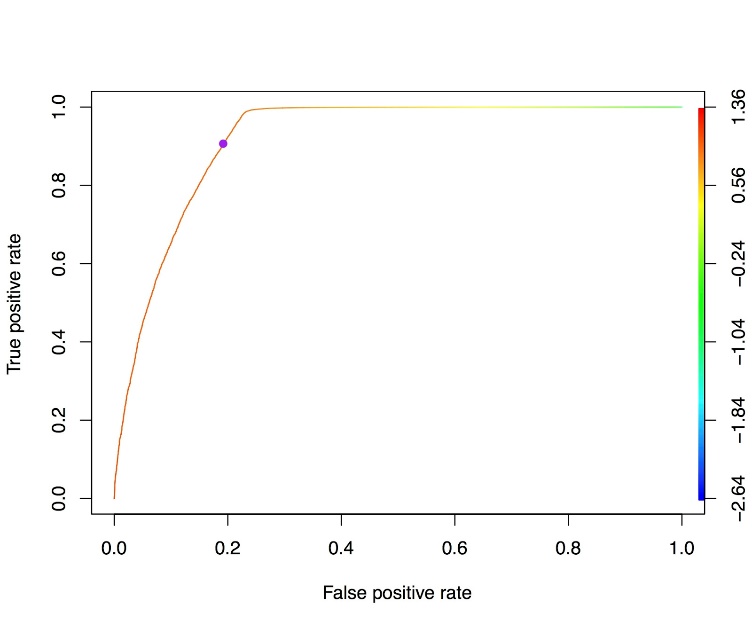


Figure 2. The ROC curve of tpr vs. fpr (The color scale represents the corresponding discriminative cut-off threshold value. The purple point shows the location of the closest point to (0,1) on the curve, which has a corresponding cut-off threshold value of 0.924.)

**3.3 Model Based Boosting**

One of the top challenges in statistical research is, we have little idea about the true data generating process. In logistic regression, we are not sure whether features entered into the model are in the form of linear combination. It is entirely plausible that there are some nonlinear functions of features entered into the model. It is also tough to decide which features should be included into the model [[[5]](#endnote-5)].

Stepwise selection, which is a classical technique for model building and variable selection, is known to be unreliable or even biased. LASSO improves prediction accuracy by restricting the coefficients within a constant, which performs variable selection. However, LASSO doesn’t directly provide solutions to how nonlinearity enters into the model. We could have made nonlinear functions of features, put them into the model, and use LASSO to make selection. But there should be a more advisable way to perform feature selection, nonlinear modeling, and accuracy improvement at the same time.

The solution we considered here is component-wise gradient boosting. It performs prediction accuracy optimization, nonlinear modeling, and functional form and variable selection at the same time. The resulting prediction rule is mathematically equivalent to General Additive Model, which is easy to interpret.

Specifically, logistic regression is chosen as the base model [[[6]](#endnote-6)]. For each continuous feature, we model it as a penalized spline and link the spline to the outcome using logistic regression. For all discrete features, we model them as a ridge regression term and link the term to the outcome using logistic regression. Gradient descend is utilized to minimize the negative binomial likelihood. In each step of gradient descend, the gradient of the model that minimizes the objective among all models in that step is chosen. So the gradients during the procedure may be from different models. 5-fold Cross Validation is performed to choose optimal stopping step to avoid overfitting (It is desirable to implement 10-fold Cross Validation or Boostrapping for optimal stopping. However, due to the high computational cost, 5-fold Cross Validation is used).

Table 3 shows the comparison of test results between logistic and boosting. In either case, boosting has higher accuracy than logistic. Since we focus on the detection of loan default to prevent loss, we pay attention to the false side statistics. When cutoff probability is 0.5, Boosting outperforms logistic in all the statistics. When cutoff probability is 0.924, Boosting generally has better performance except in Recall, 0.016 lower.

Table 3. Comparison of test results between Logistic regression and Boosting using 5-fold cross validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cutoff Prob. = 0.5 | | Cutoff Prob. = 0.924 | |
|  | Logistic | Boosting | Logistic | Boosting |
| Precision | 0.111 | 0.988 | 0.397 | 0.434 |
| Recall | 0.052 | 0.546 | 0.808 | 0.792 |
| F-score | 0.071 | 0.703 | 0.532 | 0.560 |
| Accuracy | 0.904 | 0.968 | 0.899 | 0.913 |

Similarly, ROC is used to determine the optimal cutoff probability. Due to high computational cost, only the statistics of 5 cutoff probabilities is calculated. Figure 3 shows the corresponding results. The optimal cutoff probability is ~ 0.8.

Figure 3. ROC curve for boosting

**3.4 Random Forest**

Random Forest [[[7]](#endnote-7)] is based on decision tree. Decision tree is non-parametric, easy to interpret, fast and scalable. The main disadvantage of Decision Tree is that they easily over fit, so we use random forest to average out and it is not necessary to care about pruning. It generates reasonably good performance on classification.

RF grows a bunch of classification trees. In each tree,

1. Bootstrapping data to reduce correlation

2. Select a few features to reduce correlation

3. Grow trees to the largest extent possible increase strength

The first step is parameter tuning. In related paper on random forests, it was shown that the forest error rate depends on two things: the correlation between trees and the strength of each tree. Increasing the correlation increases the forest error rate. Increasing the strength of the individual trees decreases the forest error rate. Pruning more trees to form the forest can reduce the correlation. From Figure 4, it can be seen that growing 50-100 trees gives similar performance. Since random forest is less likely to over fit the data, 100 trees are chosen for building the model.

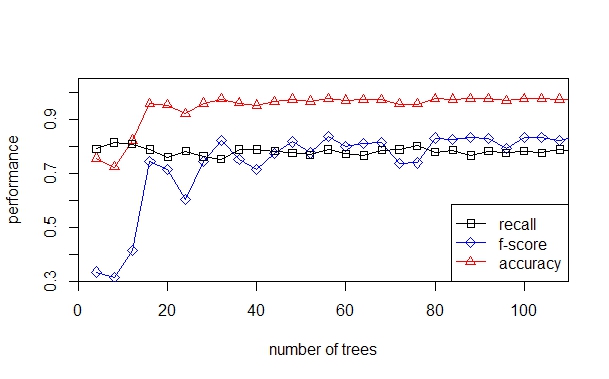


Figure 4. The relationship between number of trees and the classification performance

Increasing the number of features will increase the correlation, which increases the error. However, using too few features will make each tree too weak, which also increases the error. Thus, experiment is carried out to find out the optimum number of features. The relationship between number of features and the performance of the classifier is given in Figure 5. It can be seen that the optimum number of features per tree should be chosen as 5, which meets the conventional suggestions, square root of the total number of features 32.

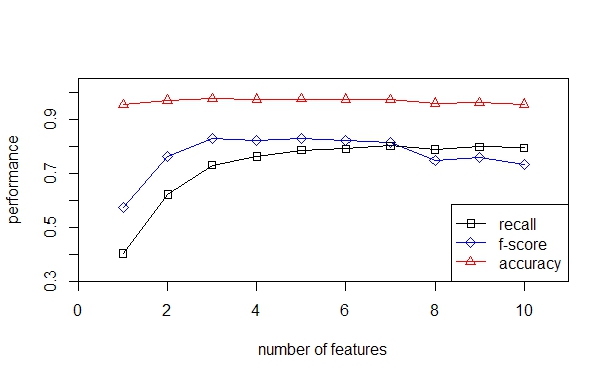


Figure 5. The relationship between number of features and the classification performance

To mitigate the influences of the imbalanced sample, we first tried to tune the cutoff, meaning how many P-voting trees are needed in 100 trees for a positive prediction. Experiment results shows that the default 50-50 has the best overall performance. In practice, RF is relatively immune to sample imbalance.

Similar to logistic regression approach, it is significant to carefully select an optimal cutoff point for RF. Cutoff for RF can be regarded as a critical number of the votes for a specific class. Intuitively, the ‘winning’ class for an observation is the one with the maximum ratio of proportion of votes to cutoff. Therefore, some tests were performed to find out this point, the corresponding results are shown in Figure 6. It can be seen that 60 seems to be the optimal.

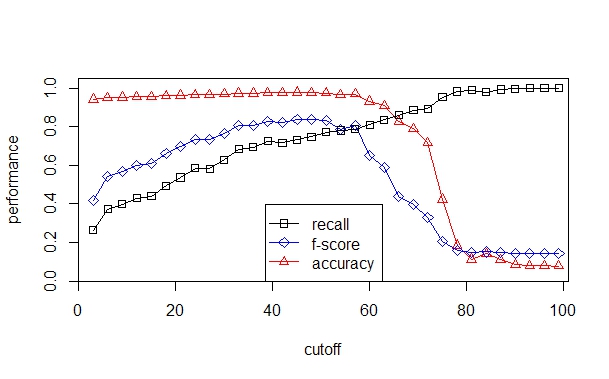


Figure 6. Relationship between cutoff point and classification performance

The classification results for RF is given in Table 4. It can be seen that Random Forest acquires satisfactory classification results.

Table 4. Classification results for Random Forest

|  |  |  |
| --- | --- | --- |
|  | Reference: False | Reference: True |
| Precision | 0.902 | 0.983 |
| Recall | 0.770 | 0.994 |
| F-score | 0.831 | 0.988 |

Based on the performance of each classifier, RF is the overall best by achieving optimum results in terms of F-score and accuracy. However, LR/LR-boosting tends to edge out slightly in terms of False recall.

**4 DISCUSSION AND CONCLUSION**

This project helps to enhance our understanding on classic ML models as well as some high level algorithms. During the process, we come up with some our own feelings on the pros and cons on different algorithms.

Following are our intuition about the algorithms we used. NB: computational not heavy, but too strong assumption. Logistic Regression: Lots of regularization, probabilistic outputs, online integrating new inputs, but mediocre performance. Model Based Boosting: a combination of accuracy improvement, nonlinear modeling, and functional form and variable selection, significantly improve the base model prediction but high computational cost. RF: Efficient for big data, nonparametric, top-notch performance, but memory-intense and annoying to tune. For logistic regression and boosting vs. RF: RF tend to beat out logistic regression in terms of accuracy and F-score given imbalanced sample, but logistic regression can be updated online and gives you useful probabilities. RF outperform model based boosting in classification, but in the case of continuous outcome, the performance comparison of these two still need to be explored. There is a trade-off between computational cost and model performance. Model Based Boosting and RF outperform the other two, but have high computational cost. For large data, logistic regression seems to be best choice if we take both computational cost and performance into consideration.

In conclusion, when data is huge, the performance gap among different algorithms is not that dramatic and it all comes down to the speed and easiness. Better data usually beats better algorithm, and good feature design gets you a long way.

**REFERENCES**

[1] Chang, Shunpo, Simon Dae-oong Kim, and Genki Kondo. "Predicting Default Risk of Lending Club Loans."

[2] Wu, Jiayu. "Loan Default Prediction Using Lending Club Data." (2014).

[3] Jordan, A. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." Advances in neural information processing systems 14 (2002): 841.

[4] Austin, Peter C., and Ewout W. Steyerberg. "Interpreting the concordance statistic of a logistic regression model: relation to the variance and odds ratio of a continuous explanatory variable." BMC medical research methodology 12, no. 1 (2012): 82.

[5] Hofner, Benjamin, Andreas Mayr, Nikolay Robinzonov, and Matthias Schmid. "Model-based boosting in R: a hands-on tutorial using the R package mboost." Computational Statistics 29, no. 1-2 (2014): 3-35.

[6] Hastie, Trevor J., and Robert J. Tibshirani. Generalized additive models. Vol. 43. CRC Press, 1990.

[7] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2, no. 3 (2002): 18-22.

**APPENDIX A**

|  |  |
| --- | --- |
| Feature names | Description |
| addr\_state | The state provided by the borrower in the loan application |
| annual\_inc | The self-reported annual income provided by the borrower during registration. |
| collection\_recovery\_fee | post charge off collection fee |
| collections\_12\_mths\_ex\_med | Number of collections in 12 months excluding medical collections |
| delinq\_2yrs | The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years |
| desc | Loan description provided by the borrower |
| 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. |
| 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. |
| funded\_amnt | The total amount committed to that loan at that point in time. |
| home\_ownership | The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER. |
| initial\_list\_status | The initial listing status of the loan. Possible values are – W, F |
| inq\_last\_6mths | The number of inquiries in past 6 months (excluding auto and mortgage inquiries) |
| installment | The monthly payment owed by the borrower if the loan originates. |
| int\_rate | Interest Rate on the loan |
| 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. |
| open\_acc | The number of open credit lines in the borrower's credit file. |
| out\_prncp | Remaining outstanding principal for total amount funded |
| pub\_rec | Number of derogatory public records |
| purpose | A category provided by the borrower for the loan request. |
| recoveries | post charge off gross recovery |
| revol\_bal | Total credit revolving balance |
| revol\_util | Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit. |
| sub\_grade | LC assigned loan subgrade |
| term | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| total\_acc | The total number of credit lines currently in the borrower's credit file |
| total\_pymnt | Payments received to date for total amount funded |
| total\_rec\_int | Interest received to date |
| total\_rec\_late\_fee | Late fees received to date |
| total\_rec\_prncp | Principal received to date |
| verified\_status\_joint | Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified |

1. [] Chang, Shunpo, Simon Dae-oong Kim, and Genki Kondo. "Predicting Default Risk of Lending Club Loans." [↑](#endnote-ref-1)
2. [] Wu, Jiayu. "Loan Default Prediction Using Lending Club Data." (2014). [↑](#endnote-ref-2)
3. [] Jordan, A. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." Advances in neural information processing systems 14 (2002): 841. [↑](#endnote-ref-3)
4. [] Austin, Peter C., and Ewout W. Steyerberg. "Interpreting the concordance statistic of a logistic regression model: relation to the variance and odds ratio of a continuous explanatory variable." BMC medical research methodology 12, no. 1 (2012): 82. [↑](#endnote-ref-4)
5. [] Hofner, Benjamin, Andreas Mayr, Nikolay Robinzonov, and Matthias Schmid. "Model-based boosting in R: a hands-on tutorial using the R package mboost." Computational Statistics 29, no. 1-2 (2014): 3-35. [↑](#endnote-ref-5)
6. [] Hastie, Trevor J., and Robert J. Tibshirani. Generalized additive models. Vol. 43. CRC Press, 1990. [↑](#endnote-ref-6)
7. [] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2, no. 3 (2002): 18-22. [↑](#endnote-ref-7)