Loan Default Prediction Using Decision Tree, Logistic Regression and Random Forest

Final Project for Machine Learning I (DATS 6202)
Project Group 5: Liwei Zhu, Wenye Ouyang, Xiaochi Li
Data Science, George Washington University

Abstract

Loan default is always the threat to any financial institution and should be predicted in advance based on various features of the applicant. This study aims at applying machine models, including decision tree, logistic regression and random forest to classify the applicants with and without loan default from a group of predicting variables, and evaluate their performance. Comparison between using unbalanced training set and balanced training set suggests that balancing the data is the key to improve model performance. The study also finds out that regression is the best model to classify those applicants with loan default, and the recall score can be up to 65% with balanced data.

Introduction

Default is the failure to meet the legal obligations of a loan, for example, failed to pay the interest of the loan on time. The default is a threat to the performance and stability of any financial institution, and one way to reduce the financial risk of default is to predict the default possibility of applicants and reject the high-risk applicants in advance.

To predict whether an applicant is likely to default or not based on a combination of the applicant's features, our team will build several machine learning models, such as random forest, decision tree and logistic regression using the lending club data.

Methods and Ideas

Since the target variable (loan_status) is dichotomous, our problem can be considered as a classification problem under supervised learning. We decided to use decision tree, logistic regression and random forest in our project.

We realized that the target variable is highly unbalanced in the data. Only 15% of the target variable are 1 (Default). As we've learned, these classifiers will be influenced by highly unbalanced data and be more likely to fail to classify the minority label in the test set. However, in the real-life scenario, these default loan (labeled as 1) will be more harmful to the financial institution. So, we decided to use SMOTE (Synthetic Minority Over-Sampling Technique) to over-sample the minority group in the data and make both labels occupied 50% of the training set. We will evaluate the performance of the classifiers trained with unbalanced data and balanced data.

We also conducted another experiment with the random forest classifier. Since some hyper-parameter influences the performance of the classifier, we made some changes on these hyper-parameters and evaluate the performance of these classifiers.

Experimental Results and Analysis

Dataset

The "Default" dataset is one of the most popular machine learning practice dataset. Provided by lending club, a p2p lending institution, it contains complete loan data for all loans issued through the 2007-2015, including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. The file containing loan data through the "present" contains complete loan data for all loans issued through the previous completed calendar quarter. Additional features include credit scores, number of finance inquiries, address including zip codes, state, and collections among others. The dataset is a matrix of about 890 thousand observations and 75 variables. A data dictionary is provided in a separate file.[1]

Data preprocessing

Although the dataset contains 890 thousand observations and 75 variables, it doesn't mean that we can directly feed the dataset into the machine learning models. The model will have a low performance or even return errors if the data hasn't been preprocessed. That's the reason why we need to do some observations and preprocessing with the data set so that we can optimize the performance of our models.

Too many NAs in each variable or observation can be a problem to our model. We removed the variables with more than 70% NAs in them and the observations that has more than 30 NA variables in them.

Too few unique data in the variable can be a problem and we deal with this problem with two methods. When there is only one value in that variable or most of values in the variable (more than 90%) are in one value, we will remove that variable. When majority of the variable has one value and there are many other values that only occupy very few percentage, we will transform this variable to a dichotomous variable, for example, we will change the majority value in that variable to one label and all the other values to another label.

[1] https://www.kaggle.com/wendykan/lending-club-loan-data

Some variables are in the format that can't be used directly by the model, for example, "36 months" or "20%". We will transform these variables and make them usable.

Finally, we removed the variables that are not relevant or need NLP to analyze.

We use one-hot transformation for the categorical variables and use 70% of the data as train set. We standardized the training set and test set separately afterwards.

Table 1. The features used in prediction and the definition

Feature	Definition
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.
amt_difference	Whether investor pays fewer than requested by the borrower
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
installment	Number of currently active installment trades
grade	LC assigned loan grade
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
annual_inc	The self-reported annual income provided by the borrower during registration.
verification_status	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
purpose	A category provided by the borrower for the loan request.
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.
delinq_2yrs_cat	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
inq_last_6mths_cat	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records
pub_rec_cat	Categorical transformation of pub_rec
acc_ratio	How many loans accounts (ratio to total account) does the applicant have now
initial_list_status	The initial listing status of the loan.
loan_status	Status of the loan

Auto Train/Test/Evaluate Process

We use three functions to automate the model training, testing and evaluation process.

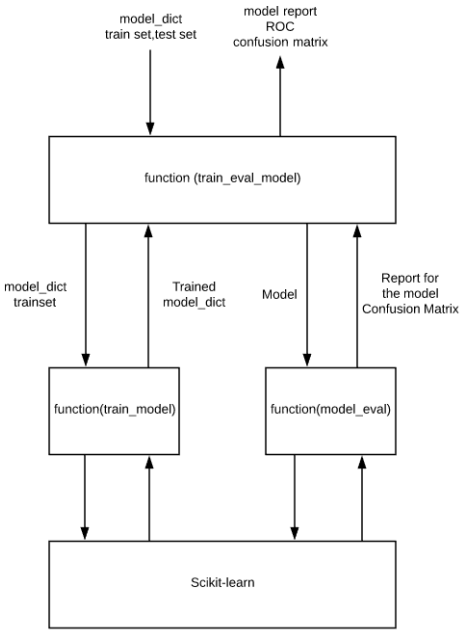

Figure 1. Illustration of the model training, testing and evaluation process

We only need to call the higher-level function (train_eval_model) and pass in the training set and test set and all the models in dictionary format and we can get the report of the models in the format of pandas data frame, which is easy to visualize with matplotlib.

Model comparison metrics

Table 2. Confusion Matrix

		Ground Truth	
		Positive	Negative
Prediction	Positive	True Positive(TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative(TN)

Table 3. The metrics, the definition and meaning

Metric	Definition	Meaning
Precision	TP/(TP+FP)	The ability of the classifier not to label a sample that is negative as positive
Recall	TP/(TP+FN)	The ability of the classifier to find all the positive samples
F1 Score	F1 = 2 * (precision * recall) / (precision + recall)	Weighted average of the precision and recall
roc_auc_score	area under the curve(AUC)	Determine which of the used models predicts the classes best

We used four metrics to evaluate the model performance as shown in Table 3.

In the real-life scenario, the financial institutions are more aware of the risk of misclassifying a bad loan applicant to a good one because these misclassifications will be more harmful to them comparing with losing some good applicants. Thus, we decide to mainly focus on recall score because it reflects the model's ability to find all the positive samples, which means the default applicant in our scenario.

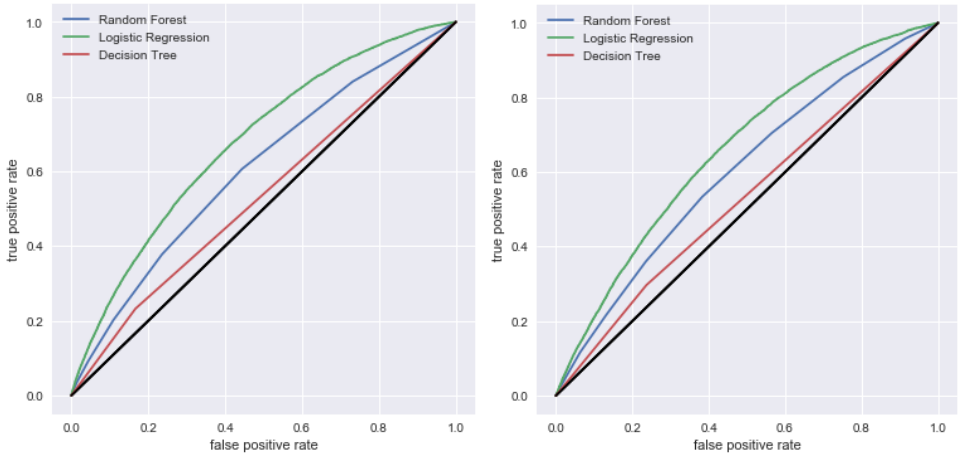

Figure2. ROC of three models with unbalanced data(left) and balanced data(right)

Table 4. Performance comparison with unbalanced data and balanced data

	Unbalanced Data				Balanced Data			
Model	AUC	Precision	Recall	F1	AUC	Precision	Recall	F1
Random Forest	0.606063	0.320000	0.036162	0.064981	0.613564	0.264678	0.186352	0.218714
Logistic Regression	0.677632	0.589744	0.001677	0.003344	0.677789	0.239949	0.646617	0.350014
Decision Tree	0.532962	0.210165	0.232429	0.220737	0.530720	0.199955	0.258895	0.225639

From table 4, we can see that although the overall performance (AUC) of the models doesn't improve significantly before and after we balance the data, the recall score improves significantly after we use SMOTE to balance the train set. We can see that the recall score of logistic regression improves from 0.167% to 64.66%, which means we can use this model to classify the 65% of the bad loans. The recall score of random forest and decision tree also improves, but not as significant as the logistic regression does.

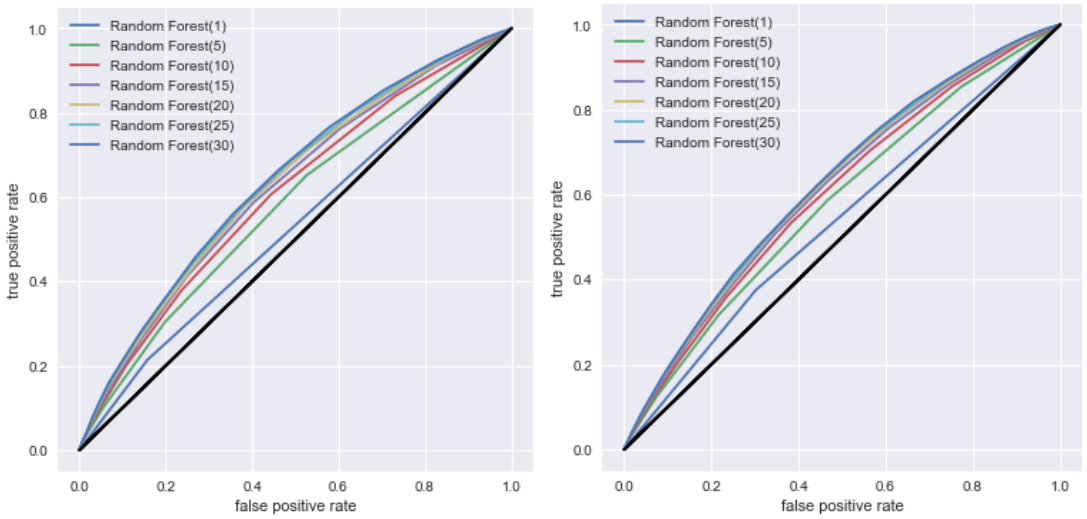

Figure 3. ROC of Random Forest contains different number of trees
With unbalanced data(left) and balanced data(right)

Table 5. Performance of Random Forest with different number of trees
With unbalanced data and balanced data

	Unbalanced Data		Balanced Data	
N of trees	AUC	Recall	AUC	Recall
1	0.528303	0.214202	0.546420	0.361184
5	0.580098	0.099883	0.593825	0.274278
10	0.606063	0.036162	0.613564	0.186352
15	0.620836	0.040901	0.622800	0.227107
20	0.628378	0.023330	0.628204	0.179717
25	0.634525	0.027851	0.631632	0.207568
30	0.638832	0.019758	0.634935	0.181467

From Figure 3 we can find that the marginal improvement of over-all performance (AUC) didn't change greatly when number of trees reaches 15. We also find out that while the recall score improves by using the balanced data, it decreases when the number of trees increases. And the performance of random forest still can't exceed logistic regression in Table 4.

Conclusion

This project focused on building different machine learning models and evaluating the performance of random forest, decision tree and logistic regression with unbalanced train data and balanced train data. We found that after oversampling the minority class by Synthetic Minority Over-Sampling Technique (SMOTE) in the training set, the recall score improved for every model, especially logistic regression. The recall score indicates that the logistic regression can classify 65% of the applicants that will be default. However, we can't find a random forest model that works better than logistic regression.

Perlich, Provost, Simonoff (2003) state that logistic regression is likely to perform best when the signal to noise ratio is low, which in technical terms means that if the AUC of the best model is below 0.8, logistic very clearly outperformed tree induction.[2] We think the complexity of our data, in other words, low signal to noise ratio can be the reason why logistic regression performs best.

We also think that the value of features in the dataset makes the decision boundary more like a linear boundary, which suits better to logistic regression instead of tree models.

Besides the high recall score with balanced data, logistic regression also has other benefits such as easy to interpret by viewing the parameters, which is the reason why we would recommend the financial institutions to use logistic regression for default prediction.

[2] Perlich C, Provostm F, Simonoff J.S. (2003). Journal of Machine Learning, 211-255, http://www.jmlr.org/papers/volume4/perlich03a/perlich03a.pdf