Machine Learning Engineer Nanodegree

Capstone Proposal

Jun Yan

Jan 9, 2017

Proposal ¶

Domain Background

Lendingclub.com is one of the leading online p2p lending platforms, where lenders are matched directly to borrowers. Since its inception in 2007, the total amount of loan reaches almost 8 billion US dollars. There are two ways to invest on lendingclub.com: automatic investing and manual investing. In automatic investing, investors specify the amount of risk they are willing to take, and lendingclub automatically invest their money in a mixed portfolio constructed using their algorithms given the risk. For manual investing, investors can browse the notes and the borrowers' information, and select which notes and how much amount they would like to invest. Furthermore, investors don't necessarily have to browser all the notes, instead they can download those data, use their own algorithm to identify which borrowers to lend their money and process them in a batch fashion. Lendingclub provides historical loan information so that one can build machine learning algorithms to predict loan performances. This project aims at identifying loans that may default so that borrowers can avoid those loans using manual investing.

There have been many similar efforts at predicting defaults using lendingclub data [1-7]. Although the goal of predicting default is the same, there are differences in feature selection, learning algorithms employed and metrics used for evaluating the model performance. In particular, some of the high performing models (for example, Ref [1]) incorrectly used fico scores that were produced after a loan is default, thus the results are highly misleading. Other analysis [2-7] use only numerical and categorical features but ignoring text. The most comprehensive feature engineering effort comes from Ref. [8], however, they use a different dataset and their goal is to report whether a loan defaults and how much is the loss if a loan defaults.

Problem Statement

In this project, I propose to predict whether a borrower will default so that investors can avoid those borrowers using manual investing feature provided by lendingclub. This, however, does not necessarily lead to highest return on investment (ROI) because by completely avoiding potential defaults, one also avoid riskier loans that may lead to higher ROI even though they default at some point in the future. In order to maximize ROI, one needs to optimize ROI instead. In this project, we work on the simpler problem, that is to predict loan defaults.

Predicting loan defaults is a binary classification problem: a borrower either default at some time during the loan term or finish payment. In reality, the majority of lendingclub loans are between default and full payment, that is, these loans are on-going. Since investors can only invest in lendingclub notes at initial stage, which means investor can not jump into on-going loans, those on-going loans are irrelevant to our discussion. A binary classification problem is a classic machine learning problem with multiple machine learning algorithms to choose from, has quantitative metrics, such as accuracy, precision, f1 score, etc. to measure the results, and is replicable with the same data and machine learning model.

Datasets and Inputs

The dataset comes from lendingclub website `DOWNLOAD LOAN DATA` section: https://www.lendingclub.com/info/download-data.action (https://www.lendingclub.com/info/download-data.action). These files contain complete loan data for all loans issued from 2007 through 2016 Q3. There are 115 columns in the dataset, including information such as borrowers' credit history, personal information (such as annual income, years of employment, zipcode, etc.), loan information (description, type, interest rate, grade, etc.), current loan status (Current, Late, Fully Paid, etc.) and latest credit and payment information. You can find the dictionary for the definitions of all data attributes here (https://resources.lendingclub.com/LCDataDictionary.xlsx).

Features

Among those columns, I plan to use loan information, credit history, personal information as features, and discard columns related to latest credit and payment information. Features include but not limited to (partly shown for simplicity):

- `funded_amnt`: funded amount
- `term`: term of the loan (36 or 60 months)
- `grade`: grade of the loan (A to G)
- `int_rate`: interest rate
- `emp_length`: employment length
- `home_ownership`: own, mortgage, rent, or other
- `annual_inc`: annual income
- `dti`: debt to income ratio
- `delinq_2yrs`: delinquency within last 2 years
- `fico_range_low`: fico score, lower one
- `fico_range_high`: fico score, higher one
- `inq_last_6mths`: inquiry within last 6 months
- `purpose`: purpose of the loan These and related features have been widely used for default prediction [1-8]. There are three types of data in the dataset: numerical, categorical and text. I plan

to use all types. Some numerical variables will be converted to categorical ones, and vice versa; text will be treated using bag-of-words representation.

Columns that shall not be used

In some of the previous work [1, 2], latest credit information such as `last_fico_range_low` and `last_fico_range_high` were used for prediction and lead to high model performance. However, these features are pulled recently according to the date specified in the `last_credit_pull_d` column. Thus the scores were obtained after a loan was fully paid or default, and their high predictive power is a false illusion because low fico score is the consequence of default but is not a predictor for default. As a result, I will exclude them from features.

Columns that have lots of NULL values

There are two types of columns that mainly consist of NULL values, the first type contains meaningful information such as column `annual_inc_joint` (if a borrower doesn't apply the loan jointly with someone else, this column is NULL); the second type is truly missing information, for example, compared to data in 2016, data from 2007 has more columns that are completely NULL. I will keep the first type but remove the second type of columns from features.

Labels

Column `loan_status` will be used as labels for classification task after some data cleaning. There are 10 different loan status in the raw data:

loan_status	Count
Charged Off	78609
Current	750682
Default	755
Does not meet the credit policy. Status:Charged Off	761
Does not meet the credit policy. Status:Fully Paid	1988
Fully Paid	337346
In Grace Period	8444
Issued	16049
Late (16-30 days)	5176
Late (31-120 days)	18491

As I mentioned above that investors can only invest in the initial period and can't jump in on-going loans, I will remove all on-going loans (with status Current, Issued). For the rest of the status, status (Fully Paid, and Does not meet the credit policy. Status:Fully Paid) will be categorized as good loan, and status (Charged Off, Default, Does not meet the credit policy. Status:Charged Off, In Grace Period, Late) will be categorized as bad loan.

Solution Statement

The solution to the problem is to train a binary classifier to predict whether a loan defaults or not. This is a classic supervised machine learning problem. There are a number of features related to personal information and credit history; as well as sufficient number of labels to train the classifier, as detailed in the Datasets and Inputs section. The results are measurable using the various metrics in the Evaluation Metrics section. There are also previous stuides to compare with, which are described in the Benchmark Model section.

Benchmark Model

There are a number of previous studies [1-8] that predict loan default using different datasets. Among them, Ref [1] incorrectly used `last_fico_range_low` and `last_fico_range_high` for prediction. Ref [2,7] has no details on the model results. Ref [8] predicts whether a loan will default, as well as the loss incurred if it does default; however it only has MAE (mean absolute error) as evaluation metrics, which I can't compare with because this project is a classification task. Ref [3-5] use lendingclub data to predict default, some of them use R and some of them use Python for analysis, but all have confusion matrix results as benchmark model to compare with. Ref [6] has ROC plot, area under curve, accuracy, sensitivity and specificity to compare with. I plan to compare my results with Ref[3-6] using confusion matrix and its associated accuracy, sensitivity, specificity.

Evaluation Metrics

Evalutate metrics will be confusion matrix and accuracy, sensitivity, specificity, the latter three can be computed from confusion matrix.

Confusion matrix is a 2×2 matrix containing true positive (TP), true negative (TN), false positive (FP) and false negative (FN)

Confusion Matrix	**Actual Postive**	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Accuracy, precision, sensitivity and specificity can be derived using confusion matrix.

$$\text{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{sensitivity} = \frac{\text{TP}}{P} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$\text{specificity} = \frac{\text{TN}}{N} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

Project Design

The workflow for this problem is to:

1. Data cleaning and pre-processing
   - handle missing values and columns that have lots of missing values;
   - convert categorical variables into numerical variables depends on whether the categorical variables are nominal or not. If they are nominal, use ordered number; and if they are not, use dummy variables;
   - convert text to numerical variables using bag-of-words representation;
   - convert other relevant variables to numerical ones, for example, convert `earliest_cr_line` which is year, to number of years with credit history;
   - remove outliers; for example, maximum value of `annual_inc` is over 9 million dolloars, and might screw the learning algorithm if it is distance based;
   - identify columns that have predictive power;
   - normalize features (machine learning algorithms that are distance based require features to be normalized)
2. Train classifier
   - perform 10 fold cross validation
   - calculate model performance as a function of training data size
   - compare a few classification models
   - Introduce new features if the model performance is poor and re-train the models
3. Compare results with benchmark models

References

1. http://cs229.stanford.edu/proj2015/199_report.pdf (http://cs229.stanford.edu/proj2015/199_report.pdf)
2. http://blog.yhat.com/posts/machine-learning-for-predicting-bad-loans.html (http://blog.yhat.com/posts/machine-learning-for-predicting-bad-loans.html)
3. https://rpubs.com/torourke97/190551 (https://rpubs.com/torourke97/190551)
4. https://res.cloudinary.com/general-assembly-profiles/image/upload/v1416535475/uwumooopprttsmpgu1goo.pdf (https://res.cloudinary.com/general-assembly-profiles/image/upload/v1416535475/uwumooopprttsmpgu1goo.pdf)

5. http://www.wujiayu.me/assets/projects/loan-default-prediction-Jiayu-Wu.pdf (http://www.wujiayu.me/assets/projects/loan-default-prediction-Jiayu-Wu.pdf)
6. https://rstudio-pubs-static.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html (https://rstudio-pubs-static.s3.amazonaws.com/203258_d20c1a34bc094151a0a1e4f4180c5f6f.html)
7. http://kldavenport.com/gradient-boosting-analysis-of-lendingclubs-data/ (http://kldavenport.com/gradient-boosting-analysis-of-lendingclubs-data/)
8. https://www.kaggle.com/c/loan-default-prediction (https://www.kaggle.com/c/loan-default-prediction)