Machine Learning Engineer Nanodegree

Capstone Proposal: Loan Defaults

Scott R Smith
February 7, 2019

Proposal

Domain Background

Credit for loans has been a critical business problem for decades. Credit is usually evaluated based upon risk of the loan being paid back. The level of risk both determines if credit is granted as well as the cost of credit by the lending amount, interest rate and time to pay back. Higher credit risk results in lower lending amounts and higher interest rates. Currently, lenders such as banks and credit unions, use a FICO score and borrower financial information to determine if the borrower is a credit risk and has the cash flow to pay the loan back.

The age of machine learning has brought about a new breed of lender categorized as FinTech. FinTech is non-bank entities lending online using fully automated processes. Fintechs such as OnDeck, Prosper, LendingClub and Kabbage are offering unsecured business and consumer loans based on advanced, proprietary, ML algorithms[1]. One characteristic of these algorithms is they use supplemental data. For example, Kabbage uses UPS delivery data and social media as part of its credit score.[2] OnDeck uses accounting data and social data in addition to traditional credit data. These 'supplemental' features are used to further increase prediction accuracy.

The goal of this project is to determine if a borrower, granted credit, will default on the loan. Methods will focus on data cleaning, prep, and feature engineering along with model selection and tuning.

My interest in this area is two-fold. First, I have spent the last 4 years working for a software company providing software to FinTechs and banks working with some of these companies directly. My goal is to understand how FinTech uses ML to lower credit risk, explore using supplemental data to create new features, and how to prep data in ways to improve algorithm performance.

[1] https://lending-times.com/2018/08/15/lenders-bet-on-artificial-intelligence-for-credit-scoring/

[2] https://www.bankdirector.com/committees/lending/serving-up-kabbage-to-small-businesses-with-a-side-of-technology/

I also have an interest in exploring how Machine Learning can be used to solve common business problems in sales, marketing, and customer experience. A lot of my interest is on data cleaning and prep, as well as feature engineering. **For this aspect, I will be using a software library of my own development, created while taking this nanodegree, to streamline the process of taking raw data and managing the tasks data exploration, data cleaning, feature engineering, data prep, and training.**

Problem Statement

In this project, we will determine if an approved and funded borrower will default on a loan. We will take credit data that contains information about the borrower for loans that have defaulted and loans paid back. These are loans that were run through the Lending Club credit algorithms, graded and approved. While most are paid off, some default. This is by design since a certain percentage of loans are expected to default, improving the ability to determine if a loan will default impacts the bottom line.

This will be a supervised learning, binary classification problem with results that either predict default or not. Success will be measured by how well we predict default based upon a measure of the accuracy of our predictions. I will also use feature engineering and explore using 3rd party data to supplement the credit data to help with prediction.

Datasets and Inputs

I will be using the Lending Club loan dataset, from 2007 to Q2 2018, posted on Kaggle. This is a very large dataset, over 3 GB uncompressed, in two data files.
https://www.kaggle.com/wordsforthewise/lending-club

These two data files are accepted loans and rejected loans. Accepted loans have data for current active loans, completed loans (paid off) and defaulted loans. This dataset has the credit data used to make the initial, approved, credit decision, borrower information (state and zip code, loan purpose, etc.) as well as data on current payment status, default status, Lending Club scoring and loan terms. The second dataset is for rejected loans. This data set is of limited value since it does not contain all of the data to make the initial credit decision so it will not be used.

From the accepted loans file, we will only use data used for the initial credit decision, including credit score and borrower attributes and loan status (defaulted or completed). We will not use loans that are current. We will also limit our data to two years of data (2016 and 2017). This will make the file more manageable and help with feature engineering as we experiment with external data to improve accuracy.

Solution Statement

The solution to the problem is to build a machine learning classification model that correctly classifies a loan credit application for default. Since this is a supervised learning problem we can measure the accuracy of our predictions. (see evaluation metrics below.)

Benchmark Model

There are many benchmarks on lending data, credit scoring and the like. The most popular is FICO. This uses up to 1500 data points to make an analysis. Default rates run from 15% to 1% for FICO scores from 650 to 850.[3]

The Lending Club dataset is a very popular dataset on Kaggle with many kernels created to evaluate and explore the data. Kernels performing the equivalent analysis are using the AUC score. There are three kernels that we can use as a 'benchmark':

- https://www.kaggle.com/ionaskel/credit-risk-modelling-eda-classification
  AUC, using the trapezoidal rule, of 0.70 and accuracy of .79. Using logistic regression
- https://www.kaggle.com/pileatedperch/predicting-charge-off-from-initial-listing-data
  AUC score of 0.689. Using logistic regression
- https://www.kaggle.com/benesalvatore/predict-default-using-logisitic-regression
  AUC score of 0.7111916771536655. Using logistic regression

Of these. I will use an AUC score, using the trapezoidal rule, of .70 as the benchmark floor.

Evaluation Metrics

Credit Scores, like FICO, accurately predict delinquency rates. For FICO scores greater than 650 the average delinquency is around 4%[4]. The Lending Club data for 2016-2017 has an 8.3% default rate overall ALL of the data segments. Using the two target segments (paid v defaulted) the number of loans was 140,379 paid vs 38,413 defaulted and calculates out as a 21.48% default rate. With these numbers, I will use 21% as the baseline for actual default rates and 8.3% for predicted to create target metrics.

From these numbers I get the following confusion matrix:

Confusion Matrix N = 178792	**Predicted Paid (0)**	**Predicted Default (1)**
Actual Paid (0)	140,379 (True -)	23,573 (False +)
Actual Default (1)	0 (False -)	14,840 (True +)

Note on the confusion matrix numbers. We had a total of 178,792 loans with actual default of 38,413. Giving 140K True negatives and 14,840 actual defaults, with 23,573 false positives.

[3] http://www.wvasf.org/presentation_pdfs/John_Meeks_-_WV_Asset_Building_Charleston_102811.pdf

[4] http://www.wvasf.org/presentation_pdfs/John_Meeks_-_WV_Asset_Building_Charleston_102811.pdf

Since this will be a categorization problem, we want to look at precision and recall:

- Precision = Of the Defaults, how many were actual defaults
- Recall = Out of all the defaults calculated, how many were correct

The above confusion matrix gives the following scores:

Recall = 1.0, Precision = 0.386, f1 = 0.55, f-beta(β=.25) = .44

Lenders want to predict as many defaults as possible without turning away business. Thus we are interested in higher precision. That is, we want to catch as many defaults as we can to maximize our lending. (We are OK with false positives, just as we are with false negatives.) The goal is to beat the precision of .386 and the f-beta score of .485. (Lending Club Performance). Using these numbers and the AUC benchmark score we have the following targets:

- AUC score > .7
- Precision score of > .386
- f-Beta score > .44 (β=.25)

The goal of this project is to beat at least 2 out of the three metrics to match or beat Lending Club's performance. An f1 above 0.55 would also be nice.

Project Design

This project will go through the following design and execution phases.

1. Examine the data.
   For this project, I only want data that would be available in an application. Two reasons: One, we do not have complete original credit data for rejected applications. Second, we want to remove any Lending Club generated data especially data that is highly correlated to defaults (for example, higher interest rates correlate to default rates.)

2. Remove unneeded columns
   Remove columns related to post-loan acceptance. This consisted of highly correlated data to defaults as well as ongoing credit checks, late payment tracking, etc.

3. Prep data for initial, default training
   Clean the data. Removing or flag missing values, one-hot encode, remove out of range data, etc.

4. Train data for baseline
   Create the baseline for models. This will be based upon default settings for the various classification models. I will create a pipeline of the various ML Classification models

including Logistic Regression (L1 and L2), Random Forest, Gradient Boosting, Decision Tree, KNeighbors, SGD, Bagging AdaBoost, and GaussianNB.

From the initial training, pick the top 3 or 4 to further analyze

5. Feature Engineer
   This will include feature engineering with the existing data (re-bucketing, indicator variables), as well as trying to add external data. For example, the cost of living by state[5] or zip code[6], salary information by job type[7], unemployment rates[8], etc.

6. Run training
   As we feature engineer, I will track the changes to the scoring and run the training multiple times to gauge improvements (or not), to the default models.

7. Tune Hyperparameters. Tune hyperparameters using a GridSearch and cross-validation splits.

8. Evaluate results. Revisit steps 5 through 8 until the target evaluation metrics are exceeded.

[5] https://www.statista.com/topics/768/cost-of-living/,
https://www.numbeo.com/cost-of-living/region_rankings.jsp?title=2016&region=021

[6] https://www.irs.gov/statistics/soi-tax-stats-individual-income-tax-statistics-2016-zip-code-data-soi

[7] https://www.bls.gov/oes/2017/may/oes_nat.htm

[8] https://www.census.gov/econ/geo-zip.html