Machine Learning Engineer Nanodegree

Capstone Proposal

Devin McCormack

Proposal – Lending Club Loan Picker

Domain Background

Lending Club (LC) is a peer-to-peer (P2P) lending platform, which matches borrowers with one or multiple lenders who “crowd-fund” the loan. P2P lending is touted as a way for everyday people to diversity their total investment profile with loans, and crowd-funding allows people to further mitigate risk by only contributing small amounts to individual loans. Additionally, they operate with low overhead, which allows them to be highly competitive with interest rates for borrowers. Although the returns of direct unsecured lending are potentially higher than either investing in pure-play loan businesses or using a CD or savings account (where the bank does the loaning “for you” and passes on an interest rate), the risk is potentially higher.

However, we know that banks make money through loaning, and there are methods for determining the risk of a loan. Beyond commonly used services such as FICO credit scores, many institutions have their own, proprietary algorithms to determine the riskiness of loaning money to a person as well as a required return interest rate, if they loan is worth investing in at all. P2P services have different criteria for assigning interest rates. Often, they want to beat traditional loans to draw more debt consolidation borrowers (almost 60% of the loans on LC are for debt consolidation). Additionally, as individual lenders hold the risk, LC simply needs to fulfill as many loans as possible, between people searching for the lowest interest rate (borrowers) and potentially inexperienced lenders. They facilitate this by having an Auto-Investor, which automatically invests money in loans based on an individual investor’s desired return, using the LC-assigned interest rates of loans.

The problem here is that LCs motivations are not wholly aligned with the people bearing the risk of defaulted loans. Long term, LC needs to have interest rates enticing enough to both lenders and borrowers – as well as meet risk tolerance expectations of lenders – but they are encouraged to apply a much more liberal borrower filter than traditional banks. Smart investors cannot depend on the Auto-Investor, and must be more stringent in the selection of loans.

Problem Statement

The problem is that LC – and therefore LC's Auto-Investor – does not screen loans as well as it can, and is potentially passing on more risk than necessary to "naïve" investors. My project intends to create a loan screener that will preferentially select loans with lower than average default rate at the given interest rate I will use machine learning to attempt to regress or classify loans in order to improve potential gains of a LC portfolio. Using the metrics provided by LC, I will develop an algorithm that will be better than random choice (Auto-Investor) at selecting loans at a given interest rate that do not default.

Datasets and Inputs

Lending Club allows all investors access to a large number of metrics on every loan; they include descriptive values such as interest rate, loan length, and loan amount, as well as potentially predictive metrics such as credit history, loan purpose, and borrower income and work history. Additionally, they periodically release outcomes of funded loans; giving an ideal dataset for developing a supervised machine learning algorithm. LC allows download of these datasets at .csv files on their website, at https://www.lendingclub.com/info/download-data.action.

I created a regression metric called "buy rate" which calculates the total return value of the loan, weighing in early payment and default. I additionally created a classification metric defining late and defaulted loans as "bad" and current and fully paid loans as "good". Both of these are more fully outlined in the Data Cleaning python notebook. These will be used as the labels of the loans, for regression and classification respectively.

Features that will be included in the algorithm include the loan amount, the income of the borrower, the length of employment, and other things such as revolving debt balances.

Solution Statement

A solution to this problem will be a model that takes in new loan data and spits out a list of loans that are less likely than average to default. At it's simplest, the solution will give list of loans that it has classified as more similar to loans that eventually are fully paid.

Benchmark Model

LC's Auto-investor simply picks loans randomly from the set of loans based on the portfolio balance you wish to maintain. For example, if you wish to set the auto-investor to pick a "platform mix" of loans, it will attempt to create a portfolio

that is 17% A rated loans, 28% B, 35% C, 12% D, 5% E, 2%F, and 1% G (with each letter indicating increasing interest rate as well as default risk); this roughly corresponds to the loan mix of the platform as a whole. However, it is unclear if it has an algorithm to pick loans within the rating groups beyond choosing randomly. Therefore, I will compare my model to a naïve classifier – essentially comparing my model's default rate to the platform total default rate.

Evaluation Metrics

As this data set is highly skewed toward "good" loans (~79% of loans are fully paid) it is most important to reduce false positives. One extremely simple yet effective evaluation metric that penalizes false positives is precision, defined as true positives/(true positives + false positives). This is additionally good, as missing a "good" loan is much less damaging than selecting a "bad" loan.

Project Design

My workflow will be divided between a few jupyter notebooks, each with a specific goal in mind. The first notebook will be dedicated to cleaning, and selecting data, the second notebook will be model selection/tuning, and the third will be applying the model to a new dataset.

In the cleaning notebook (Data Cleaning.ipynb), I will explore the data and clean it. Some things that will be necessary include replacing/removing NaN values, converting values to numeric when necessary (for example, the interest rate of loans include a percent, making it an object when it should be numeric), and dealing with date-time issues. Additionally, I will set up my labels and regression metrics in this notebook. Finally, I will narrow my feature space a bit and then export it for my model selection notebook.

The model selection notebook (LC_MLtrain.ipynb) will look at both regression and classification of the data, using several techniques. First, I will transform the data by using log to scale skewed data, and One-Hot encoding non-numeric features. Then I will test various ML methods, and select the best model. After selection, I will further tune to model to try to beat the naïve model. Finally, I will export my model to for usage in the final notebook.

The final notebook (Predicting new loans.ipynb) will be short, and will just take in a new set of loans, and apply the tuned model. It will then give suggestions for loans to pick for investment. Although the veracity of the model cannot be immediately tested – as outcomes of new loans are still unknown – this will be the working interface for selection new loans in the future.