Machine Learning Engineer Nanodegree

Capstone Proposal

Ekaterina Kravtchenko
November 17th, 2018

Proposal

Abstract

In this project, I propose to create a classification model which accurately predicts whether peer-to-peer loans are repaid, or not. To this end, I compare labeled historical Prosper loan data to the final repayment status predicted by my algorithm of choice (whether the loan was eventually repaid, or not). The dataset includes 113,937 loans, with 81 features per loan, including loan amount, interest rate, demographic information, and actual historical loan status. This information is particularly important for Prosper lenders, for the purpose of determining whether to lend money to a particular borrower, and for the company towards determining whether to offer someone a loan, which terms to offer, and what risk category to assign to a borrower.

Domain Background

One of the biggest problems in the lending business is determining whether to offer a given borrower a loan, what amount is safe to lend them, and what interest rate to charge them. Peer-to-peer lending companies offer loans, under company-determined terms, to potential borrowers, and then allow private individuals signed up with the service to choose whether to contribute to the loan (e.g., a private individual can choose to contribute 5% of the loan amount to a given borrower). These companies have the additional problem of determining how to accurately present loan risk (i.e., risk of non-payment or defaulting) to potential lenders.

In the first case, the company must determine, on the basis of background financial and demographic information, the loan amount, and other factors, whether to offer a potential borrower a loan in the first place. In the second case, potential lenders must decide whether they're willing to lend a potential borrower money. For these lenders, an accurate assessment of risk is very useful.

Given that we have a large database of historical information on which loans were in the end repaid, or not, this is fertile ground for building a supervised learning model to predict repayment status. A version of this model may assist the company in deciding who to offer a loan to in the first place, and on what

terms, and assist potential lenders in deciding who to risk lending money to. My personal motivation is to systematically investigate a rich and complex data set that I previously worked with, to see if I can get additional insight into qualitative patterns I observed, and to see if I can improve on Prosper's less formal predictions of repayment likelihood.

Problem Statement

The problem to be solved is to predict whether a loan will be repaid, or not. This can be done in terms of probability of repayment, or a categorical decision by a model. There are two possible solutions to this: either the features available at the time the borrower applies for a loan are used to predict repayment, for use by the company, or the features available to potential lender are used to build a lender-end predictive model of repayment status. Loans currently in repayment should be excluded, as their final status is not known.

At least two possible tasks are a classification task, where loans are categorized into repayment and non-repayment, and a regression task, which predicts the amount of the loan ultimately repaid by the borrower. For the purpose of this project, I will choose the former.

Datasets and Inputs

The original dataset is linked here: https://s3.amazonaws.com/udacity-hosted-downloads/ud651/prosperLoanData.csv

I previously cleaned and performed an exploratory analysis of this dataset for the Data Analyst Nanodegree. This is a very large historical dataset of peer-to-peer loans, including background financial and demographic information on borrowers, the risk status assigned by Prosper itself and likely lender yield, and most importantly, information on whether past loans were in fact repaid, or not. The dataset contains information on 113,937 loans, with 81 continuous or categorical features per loan.

I will use a slightly modified version of the dataset, which I already cleaned and organized, which is generated by the following R code: https://github.com/eskrav/udacity-data-analyst/blob/master/explore-and-summarize/explore-and-summarize.Rmd. It is possible that I will alter the code used to generate this modified dataset, as needed.

Solution Statement

I plan to build a classification model which predicts loan repayment status (whether the loan was in the end repaid, or not). The accuracy of the model should be relatively straightforward to evaluate, given that the data is clearly

labeled (see below for evaluation metrics). However, the sheer number of features available may result in overfitting, and I previously observed in my exploratory analysis that many seem to measure the same underlying factors, and/or are highly correlated with one other. In this case, dimensionality reduction, for instance using PCA, is likely appropriate. Some categorical variables will need to be re-coded for use in the models I use, and some continuous variables may need to be scaled.

The techniques I am most interested in exploring are:

1. Logistic Regression, which would provide a probabilistic estimate of likelihood of repayment, possibly allowing potential lenders to more carefully gauge their preferred level of risk.
2. Random Forest, which often produces highly accurate results, runs efficiently on large datasets, gives estimates of which variables are most important in classification, are less prone to overfitting than Decision Trees, and works naturally with categorical variables.
3. Possibly a Deep Learning Neural Network, if time allows. Deep Learning models are frequently overkill, and it is difficult to determine what they are doing under the hood (or why they may be rejecting certain customers), but they frequently produce state-of-the-art results.

A grid analysis will be used to determine optimal parameterization, where needed.

Benchmark Model

The benchmark model I am using would be the following analysis, which ended up using a Random Forest classifier: https://www.kaggle.com/jschnessl/prosper-analysis/notebook. I was not able to immediately locate any other analysis using the same dataset, or reasonably current Prosper loan data.

This model achieves a Recall score of 0.74, a Precision score of 0.44, and an F1 score of 0.55. As the author of this model points out, this model would result in a return of -23.25%, excluding additional fees, which would most likely be unacceptable in the real world (although, as I pointed out in my exploratory analysis of Prosper loan data, the predictive models, or other methods, that Prosper uses appear to grossly overestimate lender return).

Further, it is possible to compare the model results to the risk that Prosper itself assigns to the loans, in the form of loan ratings/scores, and estimated lender yield.

Evaluation Metrics

The evaluation metrics would be Recall (to ensure that a maximum number of non-repaid loans is correctly predicted), and secondarily the F1 score, which

takes into account Precision (how many of those loans predicted to not be repaid were, in fact, not repaid).

Project Design

Prior to using the data, I will need to ensure again, using my previous analysis and exploration scripts, that the data is clean, that missing data is identified and either excluded or imputed as appropriate, that categorical variables are appropriately coded, and that any additional features I previously created in the course of data exploration are appropriate and/or sufficient.

The general workflow would be to attempt the methods of interest (Logistic Regression, Random Forest, and possibly a Deep Learning model), using `scikit learn` and `Keras/Tensor Flow`, in the former case first attempting to reduce dimensionality through principal component analysis. Performance of the models, after moderate optimization/parameterization, will be compared, focusing on F1 scores (with a bias towards Recall, to ensure that all loans likely to not be repaid are identified) and model speed.

The most promising model will be chosen, depending on accuracy, runtime, usability of model results (repayment probability vs. binary classification), and potential for further optimization. This model will be further optimized, until it reaches the best performance I can achieve within the confines of this project. The aim is minimally to perform better than the Kaggle model linked above, on Recall and estimated Lender loss, and optimally to achieve at least 85% Recall.

Tools and libraries to be used: Python, R/RStudio for data preparation, Jupyter Notebook, `tidyverse` packages (R), `ggplot2` (R), `pandas`, `scikit learn`, `Keras` (possibly), `Tensorflow` (possibly), `matplotlib/seaborn`, others as necessary.