Predicting Home Credit Default Risk

Clayton Weidinger
Machine Learning Engineer Nanodegree Capstone Proposal
June 30, 2018

Domain Background

Home Loans enable long term wealth building for the borrower and a steady Return On Investment (ROI) for the lender, if the lender can avoid loans that are likely to default (not pay back their loan). Traditional methods of measuring default risk, such as the FHA insurability requirements, have created a gap between the credit haves and havenots. Those meeting FHA requirements can often find a loan that enables them to build wealth whereas those who don't are forced to waste money by renting or face the unreasonably high interest rates commonly referred to as predatory lending. I have long been concerned with the high price of housing for all. I once lived in my garage so that I could rent out more rooms in my house so that each renter could pay less each month. Now with my machine learning skills, there is something much more beneficial I could do to help those not able to meet FHA requirements.

An enterprise that can identify potential loans with low default risk using alternative, perhaps machine learned, criteria stands to profit as they will be able to earn a higher ROI than FHA approved loans and their borrowers can get a loan between the FHA approved loan market rate and the predatory lending market rate. The "Home Credit Group" has seen this opportunity and is leveraging the benefits of competition on kaggle.com to develop a better algorithm to predict home credit default risk.

Problem Statement

The "Home Credit Default Risk" competition on kaggle.com explains to problem I want to solve along with a quantification of what solved looks like. In short, the problem is to use financial data points to predict default risk ([0,1]). The "Home Credit Group" has provided training data that pairs a borrower id with the default risk (a 0 if they didn't default, a 1 if they defaulted) along with a host of other data that will serve as inputs to an algorithm. The solution will maximize the Area Under the Curve (AUC) of the Receiver Operator Characteristic (ROC). This is a specific way to calculate comparisons between algorithms that prefer True Positives and not False Positives.

Datasets and Inputs

The inputs I'll be using to predict default risk are those included in the kaggle "Home Credit Default Risk" competition (kaggle 2018). There are 221 columns in 7 separate datasets, so I will only touch on the datasets broadly and not specific columns.

Each loan has an application which contains self-report information about the client and the property the client is hoping to purchase (application_(train|test).csv). Previous_appilication.csv contains applications the client may have made to the "Home Credit Group" before the current application. The applications are relevant because they were designed to separate people with high and low risk of default. I intend to test each of these columns relationship to their default risk.

There is a corpus of data regarding the applicant's other credits with "Home Credit Group" (POS_CASH_balance.csv and credit_card_balance.csv) and their repayments (Installments_payments.csv). The rest of the data contain information about credits from other financial institutions (bureau.csv) and their balances over time (bureau_balance.csv). These corpora provide a quantitative insight into the applicant's financial transaction history. I intend to use machine learning techniques to build relationships between this information and the applications to determine default risk.

Solution Statement

This problem will be solved when the default risk for all the applications is generated from a model in a way the distribution of True Positives with respect to default risk is separable from the distribution of False Positives thus maximizing the AUC of the ROC (Area Under the Curve of the Receiver Operator Characteristic). Kaggle maintains the AUC of the ROC for 20% of the test data on the public leaderboard (Kaggle 2018). Although, there is no threshold for having solved the problem as defined by kaggle, my goal is to use what I have learned in Udacity's Machine Learning Nanodegree to do as well as I can.

Benchmark Model

My benchmark model is to classify default risk based upon FHA requirements. In practice, if a loan meets the FHA requirements, the rate the borrower obtains is significantly lower (because the loan can be insured by the FHA). This creates a positive feedback loop where the lower loan interest rate enables the borrower to have a lower monthly payment and lower total cost of borrowing. To implement this

benchmark model, I will assign a default risk of 0 to loans which meet the FHA requirements and a default risk of 1 to loans which do not meet the FHA requirements. I will submit this solution to kaggle to see what the AUC of the ROC is for the test data.

Evaluation Metrics

As mentioned previously, the kaggle competition uses the AUC of the ROC to rank solutions. This is an appropriate measurement because it rewards algorithms that correctly identify applications that will default (True Positives) and punishes algorithms that falsely predict that an application will default (False Positives). Theoretically, the Area Under the Curve is the integral of the function that creates the curve. The Receiver Operator Characteristic is the plot of True Positives (on the vertical axis) with the False Positives (on the horizontal axis) for various cutpoints in the target variable (in this case the default rate). A lender needs to know the model and the cutpoint to make decisions.

Project Design

I will follow the CRISP-DM steps that are relevant to this problem statement in a cyclical fashion. To get an understanding of the data, I will explore the distribution of values (for different data types) for each column in the dataset with respect to the target value (default risk). I will decide what transformation of the data is likely to improve predictability and when to include this variable into the model. After having done a few of these columns manually, I will make an automated process to do this work and make decisions on my behalf since making manual decisions for 221 columns is an onerous and error prone task.

Next I plan to choose a model (starting with a Gradient Boosting Machine) and prepare the datasets available to me for input into the model. I plan to slowly increase the model complexity both in terms of hyper parameters and inputs to the model to find a good fit (not under or over fit) as determined by comparing the AUC of the ROC with test and train data. To this end, I will split the application_train.csv data into many folds so that I do not have to post to kaggle in order to do my training.

References

Kaggle 2018, Home Credit Default Risk Data Description, viewed 30 June, 2018, <https://www.kaggle.com/c/home-credit-default-risk/data>.

Kaggle 2018, Home Credit Default Risk Public Leaderboard, viewed 30 June, 2018, <https://www.kaggle.com/c/home-credit-default-risk/leaderboard>.