Machine Learning Engineer Nanodegree

Capstone Proposal

Chirag Chadha
July 2nd, 2018

Proposal

Domain Background

The project being proposed is currently a featured competition on Kaggle and focuses on predicting the ability of borrowers to repay loans. Home Credit Group are a non-bank financial institution that consider alternative factors such as monthly balance snapshots of point of sales, cash loans, and previous credit cards that the applicant had with the organization, in their predictions. They target the population segment that struggle to borrow from banks or other financial institutions due to insufficient or non-existent credit histories.

My personal motivation for selecting this project is based on a desire to begin competing in real Kaggle competitions with a problem that can likely be solved using a supervised learning approach.

Problem Statement

Home Credit Group are challenging Kagglers to predict the ability of their clients to repay loans based on alternative factors that banks and other financial institutions would not normally consider. In doing so, they can ensure that those capable of repayment are not rejected. This is a classification problem that will involve predicting whether an individual is capable of loan repayment based on a number of features.

Datasets and Inputs

The following data provided on Kaggle will be used as inputs for the project:

- application_{train|test}.csv
    - This is the main table, broken into two files for Train (with TARGET) and Test (without TARGET).
    - Static data for all applications. One row represents one loan in the data sample.
- bureau.csv
    - All client's previous credits provided by other financial institutions that were reported to Credit Bureau (for clients who have a loan in the sample).
    - For every loan in the sample, there are as many rows as number of credits the client had in Credit Bureau before the application date.

- bureau_balance.csv
    - Monthly balances of previous credits in Credit Bureau.
    - This table has one row for each month of history of every previous credit reported to Credit Bureau – i.e the table has (#loans in sample * # of relative previous credits * # of months where Home Credit have some history observable for the previous credits) rows.
- POS_CASH_balance.csv
    - Monthly balance snapshots of previous POS (point of sales) and cash loans that the applicant had with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in the sample – i.e. the table has (#loans in sample * # of relative previous credits * # of months in which Home Credit have some history observable for the previous credits) rows.
- credit_card_balance.csv
    - Monthly balance snapshots of previous credit cards that the applicant has with Home Credit.
    - This table has one row for each month of history of every previous credit in Home Credit (consumer credit and cash loans) related to loans in our sample – i.e. the table has (#loans in sample * # of relative previous credit cards * # of months where Home Credit have some history observable for the previous credit card) rows.
- previous_application.csv
    - All previous applications for Home Credit loans of clients who have loans in the sample.
    - There is one row for each previous application related to loans in the data sample.
- installments_payments.csv
    - Repayment history for the previously disbursed credits in Home Credit related to the loans in the sample.
    - There is a) one row for every payment that was made plus b) one row each for missed payment.
    - One row is equivalent to one payment of one installment OR one installment corresponding to one payment of one previous Home Credit credit related to loans in the sample.
- HomeCredit_columns_description.csv
    - This file contains descriptions for the columns in the various data files.

Each table will be used to create a comprehensive record of each individual (using inner joins on the tables) and this dataset will then be used to train and test the proposed model. The final csv file ('HomeCredit_columns_description.csv') is included simply as a description of the tables.

Solution Statement

Microsoft's LightGBM is a decision tree-based gradient boosted machine learning model that offers a significantly reduced time to train compared to XGBoost and Scikit-Learn's Random Forest model. As such, it can be used to quickly train and predict without the use of GPUs on local machines. This sort of model is well suited to classification problems, has been used extensively to win a number of Kaggle competitions, and can also handle missing values without the explicit need for removal or substitution of these values. The comprehensive dataset assembled from the above tables will be used as input into a LightGBM model (which can even be trained and tested in a Jupyter Notebook in a short amount of time) and parameters will be tweaked based on training and validation loss over a series of runs. The output from the model will be a probability for each individual based on their ability to repay a loan and these probabilities will be compared with the true values for the individuals in the test set using area under the ROC.

Benchmark Model

The benchmark model used to compare with the LightGBM model will be a Random Forest model provided by Kaggle that obtains an AUC score of 0.688. This benchmark will be especially suitable in this instance to compare the performance of two tree-based methods in LightGBM and Random Forest.

Evaluation Metrics

As mentioned above, the evaluation metric used in this scenario will be the area under the ROC as this is a useful metric for binary classification problems. Additionally, the benchmark model provided by Kaggle has already been evaluated using AUC and the challenge submissions are judged by Kaggle on their AUC also.

Project Design

The following workflow will be employed in approximating a solution to the outlined problem:

1. An initial exploratory data analysis stage with a view to find missing values, distributions of features, and correlation between features. This will involve creating visualizations of the distributions of features using scatter plots, bar charts, etc.

2. The creation of a validation set (split up into training and testing sets) that will provide a reflective AUC score for the LightGBM as though it were trained and tested on the full dataset. An existing kernel uploaded by other Kagglers can be used for this step as the public score of the kernel will be provided along with the model and we can compare our local score on the validation set using the same model with this public score.

   Since the full dataset contains millions of rows, it would take a lot of time to train and test our model over a series of runs where we would be tweaking hyperparameters

(even with the LightGBM model), so a validation set for quick training and testing will speed up the process of finding an optimal solution.

3. When an optimal solution has been approximated, the LightGBM model will be trained and tested on the full dataset (using the pre-split train and test data provided by Home-Credit).

4. The final model will be evaluated using area under the ROC curve.