Machine Learning Engineer Nanodegree

Capstone Proposal

Randy Jackson
July 20, 2018

Proposal

Domain Background

My capstone project is based on the Home Credit Default Risk competition on Kaggle.

Many people struggle to get loans due to insufficient or non-existent credit histories and, unfortunately, this population is often taken advantage of by untrustworthy lenders. Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data--including telco and transactional information--to predict their clients' repayment abilities. (Kaggle, 2018).

Loans default can lead to huge losses for financial firms. Financial firms apply various methods to detect and predict default behaviors of their customers, considerable research and effort has been applied to this problem domain and entire courses are dedicated to this single problem.

https://medium.com/henry-jia/bank-loan-default-prediction-with-machine-learning-e9336d19dffa
https://nycdatascience.com/blog/student-works/kaggle-predict-consumer-credit-default/
https://towardsdatascience.com/predicting-loan-repayment-5df4e0023e92

Motivation

My personal motivation for selecting this project lies in the fact that I have spent the bulk of my career working in financial services and therefore I wanted to select a project that could be applied the financial services domain.

Problem Statement

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful. (Kaggle, 2018)

The objective of this competition is to use historical loan application data to predict whether or not an applicant will be able to repay a loan. This is a supervised binary classification task:

- Supervised: The labels are included in the training data and the goal is to train a model to learn to predict the labels from the features
- Classification: The label is a binary variable, 0 (will repay loan on time), 1 (will have difficulty repaying loan)

Some of the methods commonly used for binary classification are:

- Decision trees
- Random forests
- Bayesian networks
- Support vector machines
- Neural networks
- Logistic regression
- Probit model

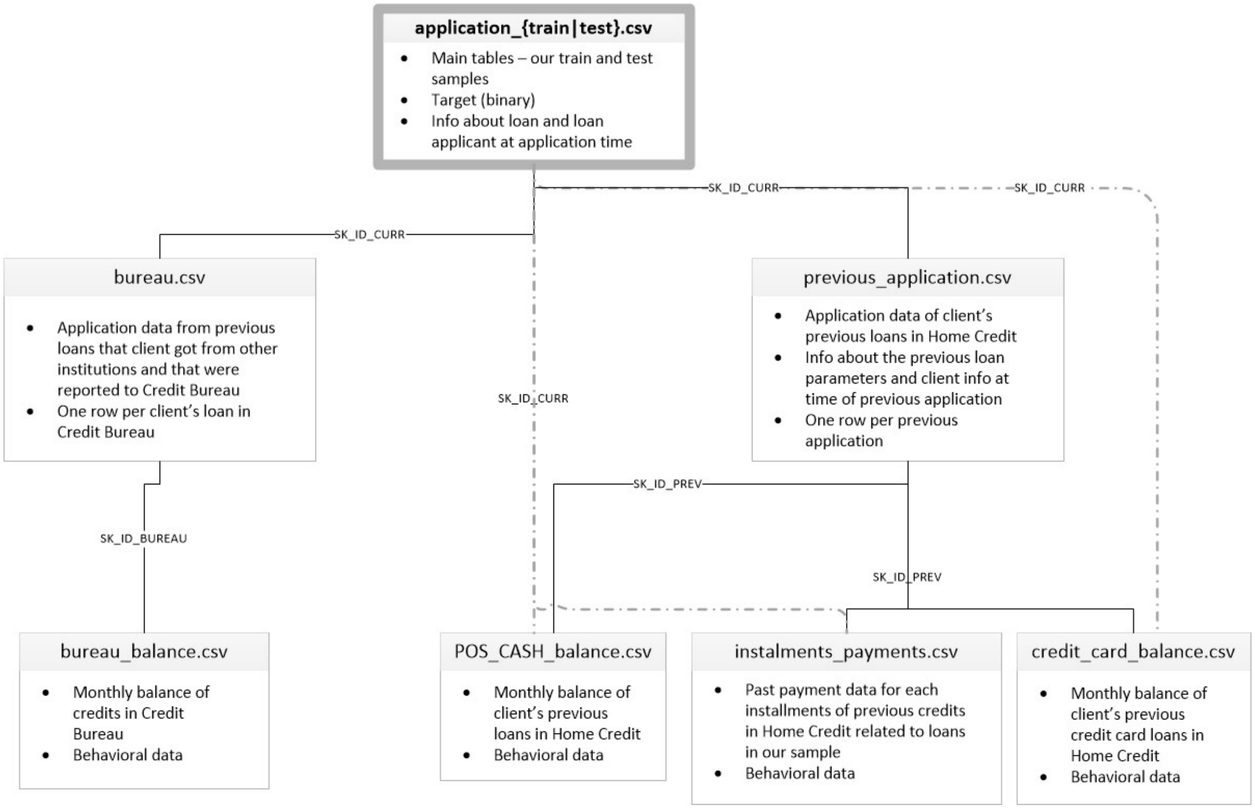
Datasets and Inputs

The data for this project is provided by Home Credit, the company hosting the competition. The training data has 307511 observations (each one a separate loan) and 122 features including the label we want to predict. The testing data has 48744 rows and 121 features. EDA has not yet been performed on the dataset. However, initial research of the dataset indicates that this is an imbalanced classification problem. Given that there are a much larger number of loans that were paid on time than those that were. This issue will need to be mitigated and cost function or sampling based mitigation options will be investigated and applied as part of the implementation of the project.

The following information was taken directly from Kaggle:

- **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 our 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 our sample).
  - For every loan in our 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 we 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 our sample – i.e. the table has (#loans in sample * # of relative previous credits * # of months in which we 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 we 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 our sample.
  - There is one row for each previous application related to loans in our data sample.
- **installments_payments.csv**
  - Repayment history for the previously disbursed credits in Home Credit related to the loans in our 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 our sample.
- **HomeCredit_columns_description.csv**
  - This file contains descriptions for the columns in the various data files.



application_{train|test}.csv

- Main tables – our train and test samples
- Target (binary)
- Info about loan and loan applicant at application time

bureau.csv

- Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
- One row per client's loan in Credit Bureau

previous_application.csv

- Application data of client's previous loans in Home Credit
- Info about the previous loan parameters and client info at time of previous application
- One row per previous application

bureau_balance.csv

- Monthly balance of credits in Credit Bureau
- Behavioral data

POS_CASH_balance.csv

- Monthly balance of client's previous loans in Home Credit
- Behavioral data

instalments_payments.csv

- Past payment data for each installments of previous credits in Home Credit related to loans in our sample
- Behavioral data

credit_card_balance.csv

- Monthly balance of client's previous credit card loans in Home Credit
- Behavioral data

Solution Statement

My goal is to create a predictive model to help determine a potential customer's probability of repaying their loan on time and determine the most important features found in the data. I plan to use multiple algorithms from the Scikit Learn Library which will help me develop better insights and ultimately decide on a target model for optimization via hyperparameters. My goal is to try multiple models including basic models like logistic regression and advanced models like gradient boosting.

Benchmark Model

A champion / challenger approach can be used when doing benchmarking. The approach uses the current model as the champion and current benchmark, which is then challenged by a new model. If the challenger model beats the champion in performance, then it can become the new champion and the new benchmark. This way, models are continuously challenged and further perfected. I will start by defining a naive solution as a very simple model as the initial champion and I will select challengers and compare their results as I investigate other models. I will likely start with a logistic regression or k-nearest neighbors as starting point because they are relatively easy to implement and then compare the results to ensemble methods or neural network designed for classification.

Evaluation Metrics

The Kaggle competition submissions are evaluated based on area under the ROC curve between the predicted probability and the observed target. I will use the ROC curve approach as the primary evaluation criteria. However, for the purpose of supplying a more comprehensive project I plan to compare results using at least two additional evaluation metrics covered in the MLND, such as Accuracy or F-score. **ROC is the metric to use when there is an imbalance and most samples are positive, so the best metric is already selected as a requirement of the project.** Accuracy has limitations with regard to unbalanced datasets. My goal will be to look at other methods and contrast the differences and include a discussion in the final workbook. Possible metrics for additional consideration or discussion are:

- True Positive Rate (TPR) or Hit Rate or Recall or Sensitivity = TP / (TP + FN)
- False Positive Rate(FPR) or False Alarm Rate = 1 - Specificity = 1 - (TN / (TN + FP))
- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Error Rate = 1 – accuracy or (FP + FN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- F-measure: 2 / ((1 / Precision) + (1 / Recall))
- ROC (Receiver Operating Characteristics) = plot of FPR vs TPR
- Kappa statistics
- **AUC (Area Under the Curve) – required for submission.**

Submission File

My desire for this project is to demonstrate my ability to do solid data science – not to win a Kaggle competition. However, I would like to l have the experience of participating in these competitions so I plan to formally submit my project results at least once before the competition ends.

Kaggle provides the follow format for the submission file: For each SK_ID_CURR in the test set, you must predict a probability for the TARGET variable. The file should contain a header and have the following format:

```
SK_ID_CURR,TARGET
100001,0.1
100005,0.9
100013,0.2
etc.
```

Project Design

Solution Architecture: Programming Language and Libraries

a. Python 3.
b. scikit-learn. Open source machine learning library for Python.

Project Design: The general sequence of steps are as follows-

a. Data Visualization: Visual representation of data to find the degree of correlations between predictors and target variable and find out correlated predictors. Additionally, we can see ranges and visible patterns of the predictors and target variable.
b. Data Preprocessing: Scaling and Normalization operations on data and splitting the data in training, validation and testing sets.
c. Feature Engineering: Finding relevant features, engineer new features using methods like PCA if feasible.
d. Model Selection: Experiment with various algorithms to find out the best algorithm for this use case.
e. Model Tuning: Fine tune the selected algorithm to increase performance without overfitting.
f. Testing: Test the model on testing dataset.

References

https://www.kaggle.com/c/home-credit-default-risk
https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction/
https://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it
https://en.wikipedia.org/wiki/Binary_classification
http://www.chioka.in/class-imbalance-problem/
https://towardsdatascience.com/what-metrics-should-we-use-on-imbalanced-data-set-precision-recall-roc-e2e79252aeba