Machine Learning Engineer Nanodegree

Capstone Proposal

Stefan Langenbach
March, 27th, 2019

Proposal

Domain Background

The challenge to forecast customer behaviour is prevalent in many industries. This project will investigate a specific variant of this problem in the financial services industry, specifically in consumer banking: To forecast which customers will make a specific transaction in the future, irrespective of its amount. While machine learning techniques have been used in the financial industry extensively, i.e. to score consumer credit risk or to detect fraudulent transactions, there does not seem to be a lot of research regarding transaction forecasting.

For further information please consult the description of the corresponding *Kaggle Competition*, Santander Customer Transaction Prediction.

Problem Statement

The need for specific financial services and products is closely tied to customers' living situations, i.e. if they are planning to attend college, start a family or buy real estate. In order to gain insight into their customers financial situation, banks can use the history of their transactions. For the case at hand, the challenge is to **identify, which customers will make a specific transaction in the future**, not taking into account the amount of that transaction. This is a classical binary (0/1, yes/no) classification problem often found in the realm of data science.

Datasets and Inputs

The data to be analyzed is provided by Santander Bank through a Kaggle Competition (see above). It is anonymized but its structure is identical to the data the bank uses to solve similar problems internally. The data is split into training and testing set, each consisting of 200,000 observations and 202 (201 for the test set as dependent variable 'target' is omitted). As the classes for dependent variable 'target' are highly imbalanced (only roughly 10% are labeled 1, 90% are labeled 0), one will have to make appropriate adjustments (oversampling/downsampling of underrepresented/overrepresented class, using class weights for penalization, etc.) during the modeling process. Further, as no dedicated validation set is supplied by Kaggle, one will have to build it manually using a subset of the provided test set as the validation set - again taking into account class imbalance. Since the variables are anonymized it is hard to make to an educate guess on which variables might be useful input to the machine learning model prior to performing exploratory data analysis (see below).

For further information please consult the data section of the Santander Customer Transaction Prediction competition.

Solution Statement

A potential solution to the problem at hand can be obtained through the application of machine learning techniques. By training various algorithms with historical transaction of individual customers, one can obtain models that can predict the probability of a customer making a specific transaction.

Benchmark Model

Given the evaluation metric specified by the Kaggle Competition (area under the ROC curve; see section below), the benchmark for the solution is to better than random choice, i.e. reaching a ROC score > 0.5. The personal ambition of the author is to come up with a solution placing him in the top 50% of the Kaggle leaderboard, which translates into a ROC score of >= 0.89 (as of the time of writing this proposal).

Evaluation Metrics

The evaluation metric used for this project is the area under the ROC curve between the predicted probability of a customer making a specific transaction (calculated by a machine learning model) and the observed target, i.e. the actual transaction made by a customer in the past (available in the test dataset).

Project Design

In order to arrive at a potential solution for this project, the following steps are required:

1. Set up a working, Python-based development environment
2. Gather relevant data from the corresponding Kaggle competition
3. Perform exploratory data analysis, i.e. calculating summary statistics, visualizing shape and distribution of the data

4. Conduct data cleansing, i.e. filling missing values (for example using the mean of all observations of one particular variable), normalizing/scaling skewed data, etc.
5. Perform feature engineering, i.e. in an automated fashion using deep feature synthesis via feature tools
6. Apply machine learning techniques to build a model, specifically supervised approaches like Extreme Gradient Boosting available via XGBoost and lightGBM, as well as deep learning implementations found in fastai's Python library.
7. Evaluate model performance by submitting model predictions to Kaggle and comparing them with the leaderboard
8. Write a report summarizing the project