

Machine Learning Engineer Nanodegree

Capstone Proposal

Chen Tong

Jun 23rd, 2019

Proposal

Domain Background

Starbucks is an American coffeehouse chain. Once every few days, Starbucks sends out an offer to users via different ways such as mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks. An important characteristic regarding this capstone is that not all users receive the same offer.

Problem Statement

I chose to build a model that predicts whether or not a customer will complete an offer, meaning that the customer will receive a offer, view the offer and finish the offer before expire day. Thus, to solve this problem, I will establish a binary classifier.

One potential solution has 3 steps:

1. Preprocess combined transaction, demographic and offer data. This dataset describes an offer's attributes, the user's demographic data and whether the offer was successful. Also, split datasets into train, valuation and test datasets.
2. Training XGBoost Model as baseline.
3. Training PyTorch deep learning model and improve the results.

The evaluation metrics is accuracy and F1-score. Accuracy means how will a model correctly predicts if an offer is complete. Since this dataset is imbalanced, F1-score is better choice than precision and recall because it is a weighted average of them.

Datasets and Inputs

The data is contained in three files:

- portfolio.json - containing offer ids and meta data about each offer, such as the duration and the amount a customer need spend to complete it for an offer.
- profile.json - demographic data for each customer including their age, gender and income.

- transcript.json - records for transactions, offers received, offers viewed, and offers completed.

These files will be joined into one table which contains demographic data of a customer, attributes of the offer and whether the customer complete the offer.

Solution Statement

This is a binary classifier. Whether a customer complete an offer or not is represented as 1 (complete) or 0 (incomplete). The metrics are accuracy and F1-score. The higher the better. We will use XGBoost and Deep Learning models in this solution.

Benchmark Model

XGBoost will be benchmark model here. The XGBoost is a boosting random tree model which is very classic way to solve binary classification problem. I will train a simple XGBoost using basic hyperparameters and set their metrics as baseline for solution model.

Evaluation Metrics

I chose two evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metrics are accuracy and F1 score. See reference for the definitions of them.

Project Design

The project will follow this workflow for approaching a solution:

1. Data Processing
 1. Data Cleaning
 - Profile: figure out how to deal with null values for age, gender and income. Simple way may discard these values. But if they make up a big portion of data, I may use another category to represent them.
 - Portfolio: split channels into its own columns.
 - Transcript: make an offer completion column or dataset based on customer's behaviors(events).
 2. Data Transformation: Join these tables into a data set. The first column is whether the offer is completed, and the rest are features from profile and portfolio.
2. Feature Engineering
 1. Exploratory analyze features to choose a small number of uncorrelated features.
 2. If features are too many, I may consider to reduce dimensionality by performing a PCA.
 3. Prepare final datasets: split dataset into train, valuation and test datasets.
3. Train XGBoost Model on SageMaker
4. Train Deep Learning Model on SageMaker
5. Hyperparameter Tuning Deep Learning Model on SageMaker

Reference

1. [Starbucks Wiki](#)
2. [Udacity Starbucks Project Overview](#)
3. [Accuracy](#)
4. [F1 score](#)
5. [Starbucks' Capstone Challenge](#)
6. [Starbucks Capstone Challenge Dataset Customer Offer Success Prediction](#)
7. [Who Might Respond to Starbucks' offer?](#)