Starbucks Capstone Project Proposal

Egor Makhov
george.mahoff@gmail.com

Udacity, MLE Nanodegree — May 23, 2020

Domain background

This Starbucks Capstone project is part of the Udacity Machine Learning Engineer Nanodegree. Udacity partnered with Starbucks to provide a real-world business problem and simulated data mimicking their customer behaviour.

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. As part of marketing strategy, we always want to figure out if a customer will spend more by giving a sound offer. Providing right offer to right customer could help build loyalty of the brand and product and as a result increasing sales margins in the long run

Problem statement

The problem we are looking to solve here is conceptually easy to understand, albeit difficult to answer. We are looking to best determine which kind of offer to send to each customer segment based on their purchasing decisions. We'll touch more on what these offers are and data we'll be utilizing down in the next section. We will leverage traditional evaluation metrics to determine which model is most appropriate for our dataset. These evaluation metrics will be discussed in an upcoming section.

Datasets and inputs

As given by the Udacity's Starbucks Project Overview:

- The program used to create the data simulates how people make purchasing decisions and how those decisions are influenced by promotional offers.
- Each person in the simulation has some hidden traits that influence their purchasing patterns and are associated with their observable traits. People produce various events, including receiving offers, opening offers, and making purchases.
- As a simplification, there are no explicit products to track. Only the amounts of each transaction or offer are recorded.
- There are three types of offers that can be sent: buy-one-get-one (BOGO), discount, and informational. In a BOGO offer, a user needs to spend a certain amount to get a reward equal to that threshold amount. In a discount, a user gains a reward equal to a fraction of the amount spent. In an informational offer, there is no reward, but neither is there a requisite amount that the user is expected to spend. Offers can be delivered via multiple channels.
- The basic task is to use the data to identify which groups of people are most responsive to each type of offer, and how best to present each type of offer.

The data is divided in 3 files:

- profile.json: Rewards program users (17000 users x 5 fields)
  - gender: (categorical) M, F, O, or null
  - age: (numeric) missing value encoded as 118
  - id: (string/hash)
  - became_member_on: (date) format YYYYMMDD
  - income: (numeric)

- portfolio.json: Offers sent during 30-day test period (10 offers x 6 fields)
  - reward: (numeric) money awarded for the amount spent
  - channels: (list) web, email, mobile, social
  - difficulty: (numeric) money required to be spent to receive reward
  - duration: (numeric) time for offer to be open, in days
  - offer_type: (string) bogo, discount, informational
  - id: (string/hash)

- transcript.json: Event log (306648 events x 4 fields)
  - person: (string/hash)
  - event: (string) offer received, offer viewed, transaction, offer completed
  - value: (dictionary) different values depending on event type
  - offer id: (string/hash) not associated with any "transaction"
  - amount: (numeric) money spent in "transaction"
  - reward: (numeric) money gained from "offer completed"
  - time: (numeric) hours after start of test

Solution statement

To solve this, I will use multiple supervised learning models to determine the propensity for a customer to complete an offer.

Proposed models are Logistic Regression (benchmark model), Support Vector Machines, and Neural Network.

Logistic Regression is a common technique used for binary classification problems. Propensity models are a form of binary classification, since they are concerned whether a customer is likely to respond to an offer or not.

Support Vector Machines (SVMs) attempt to find the best dividing hyperplanes in the data to determine whether to send an offer or not. I will use the kernel method to add non-linearity to the SVM model.

Neural Networks are universal function approximators. This means they can approximate any smooth polynomial function, allowing them to better find the boundaries in high-dimensional data. I'll build a simple neural network with 3 layers:

- The first layer is the input layer and will contain nodes for each feature.
- The second layer is the hidden layer and I will test different numbers of hidden nodes during hyperparameter tuning.
- The final layer is the output layer and will consist of a single node using a sigmoid activation function. This will output a value between 0 and 1, where 1 if the customer is likely to use the offer, and 0 otherwise. I will determine during my testing where the threshold should be on determining whether to send the offer or not.

For hyperparameter tuning I'll use a grid search.

Benchmark model

Logistic regression is quite efficient in dealing with the majority of business scenarios for propensity modelling, so it will be used as a benchmark model.

Evaluation metrics

False negatives are the worst kind of error we can make for this project. If we produce a False Positive, the user will likely just ignore our marketing effort and result in possibly some wasted effort on our part. In extreme cases, the user could view False Positives as harassment and be turned off by our brand. Because of this extreme case, False Positives are still important but not as important as False Negatives.

False Negatives result in a missed opportunity to market to a receptive customer. This can result in the business not making sales that they would have otherwise made.

Precision is used when the cost of False Positives is high. Recall is used when the cost of False Negatives is high. In our case we want to consider the cost of both but focus more on False Negatives. To do this we can use the F2 score, which puts more emphasis on recall.

The F2 score is defined as:

$$F_2 = (1 + 2^2) \cdot \frac{precision \cdot recall}{(2^2 \cdot precision) + recall} \tag{1}$$

Project design

The project will have the following steps:

- Data loading and exploration: Load files and present some data visualization in order to understand the distribution and characteristics of the data, and possibly identify inconsistencies.
- Data cleaning and pre-processing: Having analyzed the data, handle data to fix possible issues found.
- Feature engineering and data transformation: Prepare the data to correspond to the problem stated and feed models. The transcription records must be structured and labeled as appropriate offer or not.
- Splitting the data into training, validation, and testing sets.
- Defining and training the benchmark model.
- Defining and training the proposed model.
- Evaluating and comparing model performances: Comparison between the accuracy of both models to verify each one is more suitable to solve the problem stated.
- Preparing final report.