Capstone Proposal

November 24, 2019

1 Machine Learning Engineer Nanodegree

1.1 Capstone Proposal

Júlia Tessler
November 25th, 2019

1.2 Proposal: Optimizing App Offers with Starbucks

1.2.1 Domain Background

The interest in optimizing (and even personalizing) offers to final customers is one of the main goals for business-to-consumer companies (B2C). This has been a rising field of interest in machine learning applications for a while now, since gathering, storing and processing data has become cheaper and easier than before. For companies, the usual goal is to sell more by making offers more appealing to consumers, whereas the consumers benefit from better user experience.

I've been working as a data scientist in Latin America's biggest foodtech companies, iFood. For iFood, the scenario as a B2C company is the one I mentioned above: selling more by providing a better consumer experience. Although my main focus has been improving search results, recommendation systems and ranking algorithms have been an interesting field for me recently, which is why I chose this project.

1.2.2 Problem Statement

The problem is simple: we want to make better purchasing offers to Starbucks' customers. For this, we can use customer's past behaviour to find patterns and try to be more assertive. As given by the Udacity's Starbucks Project Overview, 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. In other words, this is a classification problem where the model takes user behaviour data as input and produces a group as output (either previously defined or not).

1.2.3 Datasets and Inputs

As given by the Udacity's Starbucks Project Overview:

- The dataset comes from a program that simulates how people make purchasing decisions and how those decisions are influenced by promotional offers.
- 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 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

The `event` data in `transcript.json` can be used as label for a supervised learning classification algorithm, to measure for success of the offer. On the other hand, to use customer's data without labels in order to cluster clients with unsupervised learning algorithms might also be interesting. We'll discuss which approach seems more appropriated once we dive deeper into the data (during exploratory data analysis).

1.2.4 Solution Statement

We'll try to segment users based on the provided data. The goal is to improve responsiveness to the different types of offers. For this, we'll use clustering/classification algorithms. By the provided

data, it looks like we can use supervised learning models to find those clusters, then suggest an offer that's more effective for the customers who fall into that cluster.

1.2.5 Benchmark Model

I haven't been able to find benchmak models to this project, so I'll train a naïve classifier as baseline model. This can be achieved using `DummyClassifier` from Scikit Learn. The strategy to be used will be to predict the marjority class.

1.2.6 Evaluation Metrics

For this project, I'll use F1-Score as main evaluation metric.

F1-Score might be better suited for this problem since it's a weighted average of the precision and recall metrics, so it might be better to evaluate datasets that aren't very well balanced.

1.2.7 Project Design

The expected workflow to this project is as follows:

- Exploratory data analysis and defining which algorithms will be used.
- Feature engineering and selection. We'll need to combine the provided datasets to extract the most interesting data for modeling.
- Train a naive model as a baseline model.
- Modeling (training a few models to choose the best one). The decision between supervised or unsupervised learning models will me made during the EDA step, because we have to check if it makes sense to use some of the data as labels for this purpose.
- Refine model parameters.
- Reach a conclusion.

Note that this is highly based on CRISP-DM methodologies.

We expect the final model to be able to work for other customers who aren't in the provided dataset (that is, we expect the model not to overfit and be generalizible).

[ ]: