Machine Learning Engineer Nanodegree

Capstone Proposal

Jose Mendez
July 2020

Domain Background

Where there is meaningful data, machine learning becomes a powerful tool, to improve revenues for business but also to offer better services. Nowadays with the smartphone revolution there is an app for virtually everything, and every service, and those apps collect insightful data about the users behaviours browsing the app that can be used to predict their needs and provide them the right offers. A great example that will be used in this project is Starbucks, as a case of study we are going to analyze simulated data provided by Starbucks and Udacity to make decisions to send offers to the client based on that data.
My goal is to leverage my experience in analysing this type of data, to apply the same ideas in other similar projects in production.

Problem Statement

The starbucks company wants to figure out an effective way of sending special offers to their customers through the in-app notifications. Our mission is to create a Machine Learning model that based on historical data we can predict if we should send a discount offer to the user or not.

Dataset and Inputs

From the capstone instruction, there are 3 files (datasource). The first is “portfolio”, it contains the list of offers sent to the customers. The offer can be a discount, a Buy One Get One (BOGO) or Informational (no real offer), and we’ve got the details about the discount, reward and period of the offer.
We also have a datasource “profile”, this is the list of all customers that interacted with the app. Each profile contains some personal information like age, gender, income, etc.
The third datasource is “transcript”, it contains the list of all actions on the app relative to special offers, plus all the customers’ transactions. For each record, we’ve got a dictionary of metadata, like offer_id and amount spent.

Solution Statement

This is a classification problem, my approach is going to be: create a Machine learning model to predict the best offer for the user, for BOGO or discount (we are going to leave out the informational offers, which have no real “conversion”)
There are the steps to follow:

- Fetching data
- Data Cleaning
- Data Preparation
- Data visualization and analysis
- Train Model
- Evaluate the model

Benchmark Model

We are going to explore several algorithm like:

- Random Forest Classifier
- Amazon Sage maker XG-Boost built in Algorithm.
- Decision Tree Classifier
- LightGBM Mode
- Etc.

We are going to compare the results and make some refinements using the Hyper parameter Tuning.

Evaluation Metrics

Since this is a classification problem we can use the following metrics to evaluate the model:

- **Precision** The proportion of positive cases that were correctly identified.
- **Recall** The proportion of actual positive cases which are correctly identified.
- **F1-score**, that combines the two previous measures.
- **Roc_auc_score** Area Under the ROC Curve (AUC), a measure that calculates the area under the Receiving Operating Characteristic Curve. This particular curve accounts that higher probabilities are associated with true positives and vice-versa.

Project design

During the **fetching data** process, we have to join all the different pieces of information coming from the 3 data sources, we need to fetch the data,

Data preparation. After analyzing the data, we transform the dataset through different stages missing imputation, categories encoding, data standardization, etc.

Clean and **Visualize** the data to understand it's content and analyze it, to find posibles outliers and remove them (if possible)

Train the model We can create 2 different ML models, one to predict the BOGO, the other to predict the Discount. For each model, we try different algorithms, such as **Random Forest Classifier, Amazon Sage maker XG-Boost built in Algorithm, Decision Tree Classifier, LightGBM Mode**
For each algorithm, we tune the hyper-parameters to find the one with the best performance. With the best two models, we can combine the results in order to obtain a single type of offer for the user.

Evaluate the model we measure and compare the performances of the models with the current benchmark, to learn if the proposed solution is viable for the offer attribution process.