Starbucks Capstone Project Proposal

Domain Background

The proposed project falls under the domain of Marketing & Recommendation systems. Data analysis and machine learning techniques have been widely adopted in the Marketing domain in the recent years to aid companies in making correct decisions regarding their Marketing goals and to propose the best offers for their customers. Through Data analysis companies study their customers dividing them into segments to deliver the best offers suitable for each segment. Machine learning techniques can further help companies predict their customers' attitude based on historical data collected through applications, websites, or databases in general. I chose that project since I'm interested in visualizing and studying information hidden within data. My goal is to challenge myself in correctly cleaning the data, finding the right insights, and building a useful machine learning model that might help a company like Starbucks deliver the best offers they could on their mobile application.

Problem Statement

The famous Starbucks company has provided a mobile rewards application for their customers. The goal of the mobile application is to deliver certain offers to the users to encourage them to buy products from Starbucks. Since Starbucks has a variety of customer segments it might be hard to select the best offer for each user based on his demographic data. So, the company provided simulated data that simulates the real users of the application given their demographic data, their transactions, and details about some of their provided offers. My goal is to use the demographic data and the offer info first to analyze the different customer segments and how they are affected by different offers then to build a machine learning model, a binary classifier, that predicts whether a user might accept an offer given the user's demographic data and the offer details.

Datasets and Inputs

The Dataset is provided by Starbucks to Udacity for the capstone project. It represents a simulation for the actions of real users on the mobile rewards app. The dataset Is divided mainly into three JSON files:

- **Profile JSON file**
  - Has demographic data about the users (age, gender, income, … etc).
  - The simulated data has 17000 rows of different users with different demographic data.
  - A quick analysis of the dataset shows that there is some missing data specially in the income and gender columns.
- **Portfolio JSON file**
  - Containing offer ids and metadata about each type of offer.

    - There are 3 main types of offers provided in the dataset.
        - Buy One Get One (BOGO): states that the user has to pay a certain amount for a product to receive a reward of the same price.
        - Discounts: states that the user has a certain discount if he pays a certain amount during the offer duration.
        - Informational: Just advertising a certain product to influence the user to buy that product.
    - For each offer we have other information such as the duration of the offer, the difficulty (amount that needs to be paid) and the reward.
    - There are 10 combinations of offer types, durations, difficulty, and rewards provided in the dataset – only a subset of what the actual app offers.
- **Transcript JSON file**
    - Has data about the transactions and events made by different users.
    - For each user we have mainly 4 types of events.
        - Offer received: recorded whenever a user receives a certain offer.
        - Offer viewed: recorded whenever a user views the offer received.
        - Transaction: recorded whenever a user pays an amount of money through the app.
        - Offer Completed: recorded whenever a user completes an offer within the duration of that offer even if the user wasn't influenced by that offer.

Solution Statement

The provided dataset would be used to build a machine learning model that can predict whether a certain user might accept an offer given his demographic information as well as the offer details. This might help the app assess whether to send a certain offer to a certain segment of customers.

Project Design

The following steps summarize my proposed solution:

- Loading the dataset files into data frames.
- Cleaning the Dataset by:
    - Imputing Values for the missing data or removing rows with missing data.
    - Removing the offer transactions that didn't influence the customer, i.e the customer didn't view them, whether he completed the offer or not.
- Creating Visualizations and performing EDA to view the different customer segments and get insights about their behavior while using the app.
- Creating a data frame that has all the required details to build the machine learning model where each row should represent an offer and have the following attributes:
    - Demographic data about the user.
    - Detail information about the sent offer.

    - Whether the user Completed the offer or not as well as the amount paid for the transaction.
- Splitting the dataset into train and test splits.
- Applying any suitable feature Engineering according to the performed EDA.
- Using Auto ML (such as autogluon) test different models on the given dataset where the target column is to predict whether a user has completed a certain offer.
- Pick the best Model by autogluon as a baseline model and begin further tunning its hyperparameters to get the best possible performance.
- Deploy the best model to an endpoint using AWS Sagemaker and test it.

Benchmark Model

Since I intend to use tabular data and the task is considered a binary classification task, a models' benchmark can be created using Auto ML (Such as autogluon) to try different model types with simple hyperparameter tuning then picking the best model as a benchmark. The models might include the famous tree models XGBoost, LightGBM, RandomForest as well as the SVM model.

Evaluation Metrics

Since the task is a binary classification task, I choose the following two evaluation metrics for my model:

- **Accuracy**: Measures how good the model performs given what the model predicts, it is calculated as follows:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives}$$

- **F1-Score**: Added to quantify the model accuracy in case of any imbalance in the dataset, the F1-score is calculated as the harmonic mean between precision and recall as follows:

$$precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$recall = \frac{True\ Positives}{True\ Postivies + False\ Negatives}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$