Project Background

This project is the Capstone project of the Udacity Machine Learning Engineer Nanodegree. The data set contain simulated data that mimics customer behavior on the Starbucks rewards mobile app. The data has been simplified so as to only contain one product. Once every few days, Starbucks sends out an offer to users of the 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. Not all users receive the same offer, and that is the challenge to solve with this data set.

Project Statement

The aim of this project is to create a model looking at customers who had either completed and offer without viewing or before they had viewed the offer and customers who did not complete the offer.

Once a model to predict whether a user would naturally have completed an offer has been created the evaluation can be extended and used to predict whether or not the users who had viewed and completed the offer would have done so naturally anyway and calculate whether the offers generated additional revenue.

Data Sets & Inputs

The data is contained in three files:

- portfolio.json - containing offer ids and meta data about each offer (duration, type, etc.)
- profile.json - demographic data for each customer
- transcript.json - records for transactions, offers received, offers viewed, and offers completed

Here is the schema and explanation of each variable in the files:

portfolio.json

This file describes the characteristics of each offer, including its duration and the amount a customer needs to spend to complete it (difficulty).

- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) -
- channels (list of strings)

profile.json

This file contains customer demographic data including their age, gender, income, and when they created an account on the Starbucks rewards mobile application.

- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

transcript.json
The third file describes customer purchases and when they received, viewed, and completed an offer.

- event (str) - record description (ie transaction, offer received, offer viewed, etc.)
- person (str) - customer id
- time (int) - time in hours. The data begins at time t=0
- value - (dict of strings) - either an offer id or transaction amount depending on the record

An offer is only successful when a customer both views an offer and meets or exceeds its difficulty within the offer's duration.

Solution Statement

The solution to this problem will be to create a few different classification models in find a model which best fits the data, the models will be enhanced by using hyperparameter optimization to further improve the results.

Benchmark Model

The benchmark model in this project will be extremely simple, we will look at competing against the naïve assumption that all offers are completed successfully. This of course may be changed later along if there is significant class imbalance in the model as it may prove to be lenient a benchmark.

Evaluation Metrics

The evaluation metrics will be standard evaluation metrics for binary classification, these being metrics such as accuracy, f1-score and also looking at the confusion matrix to further assess the true positive and negative results.

Project Design

The project will follow a pretty standard data science process flow, starting with data exploration before transforming the data and applying a data set to models for training and evaluation.

Explore Data

Explore the data sets to better understand the feature and structure of the data. There will be some visualisation e.g. for fields such as age and income to look into their distribution as well as just a lot of work in the IDE looking at the data in a more tabular form.

Clean Data

Based on what comes out of exploring the data, any anomalies will be cleaned, either by fixing the data or removing the errant data. Further to this, the data will be flattened in some cases a feature will be pivoted so that flags exist rather than the original value e.g. for offer type.

Create features & targets

After cleaning the data, we will be able to focus on defining the target variable as well beginning to create features, for instance looking at a customer's previous offer completion rate is likely to have some correlation to whether the complete the next offer. This will create a lot of work in order to convert the time series data in point in time snapshots but may ultimately provide useful predictive power.

Training/Test

Since there is a date element to this data usually you would create a cut off point you use the old data to predict future data however in this scenario, due to the fact we have a secondary objective to explore the effects on the promotions I will use the standard approach of shuffling and splitting the data. If the classes are not balanced then I will use the sci-kit learns inbuilt weight parameter as an optimisation technique in order to investigate the effect of balancing the classes. I will also use cross validation in order to try and minimise the risk of overfitting the model.

Models

I will use a few different classification models as well as using hyperparameter optimisation in order to find the best model based on testing. I look to use tree based methods such as Random Forests and XGBoost as well things such as Logistic Regression.

Evaluate

The models will be evaluated based on the metrics discussed in the previous section, the best model will then be used to explore the second part of the problem in which we investigate the scenarios in which the user viewed and then completed the offer how likely they would have been to complete the offer without viewing the offer to assess the real impact of the offers.