Project Background

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. 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.

Problem Statement

I am interested to answer the following two questions:

- Which offer should be sent to a particular customer to let the customer buy more?
- Which demographic groups respond best to which offer type?

Datasets and 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

- 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) - time for offer to be open, in days
- channels (list of strings)

profile.json

- 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

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

Solution Statement

The solution proposed for the problem will be supervised learning models including at least logistic regression and random forest. I may try to include a gradient boosting model into the proposed solution depending on how time and resource intensive it may be to incorporate it.

Benchmark Model

A naïve predictor model that assumes every customer offer sent was successful will serve as a simple, benchmark model to evaluate the performance against the other models that will be used.

Evaluation Metrics

The metrics to quantify the performance of both the benchmark model and the solution models will be the accuracy and the F1-score.

The accuracy measures the fraction of predictions that the model got right and is defined below:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

The F1-score is also a measure of a test's accuracy except that it considers both the precision and the recall to compute the score. Prior to defining the F1-score however, we will need to define both the precision and recall.

Precision describes how precise the model is out of the predicted positive cases and is defined below:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall describes how many cases were actual positives through the model labeling it as positive and is defined below:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

With the Precision and Recall definitions in mind, the F1-score metric can be interpreted as the weighted average of the precision and recall and is defined below:

$$F1 = 2\ x\ \frac{Precision * Recall}{Precision + Recall}$$

Project Design

The solution and the results will be developed and obtained in the following manner:

- Download/Retrieve the datasets
- Explore/Visualize the data
- Pre-process the data
- Perform data analyses on cleaned pre-processed data
- Train the supervised learning models
- Test the supervised learning models
- Evaluate the models using the chosen metrics and select best model