Starbucks Capstone Challenge

Project Overview

This project is the Capstone project of the data scientist Nanodegree in Udacity. The given Data set. 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.

Problem Statement

The goal of the project is to identify whether customers will respond to the offer given transaction,demographic and offer data.
The steps to solve the challenge is:
Explore all the given data.
Preprocess and visualize data.
Data modelling
Training on different algorithms
Evaluating the metrics and choosing a better performing model.

Metrics

F1 Score is used to measure a test's accuracy
F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).
High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

F1 Score tries to find the balance between precision and recall.

- Precision : It is the number of correct positive results divided by the number of positive results predicted by the classifier.

- Recall : It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

Analysis

Starbucks has provided three different set of datasets as follows:
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

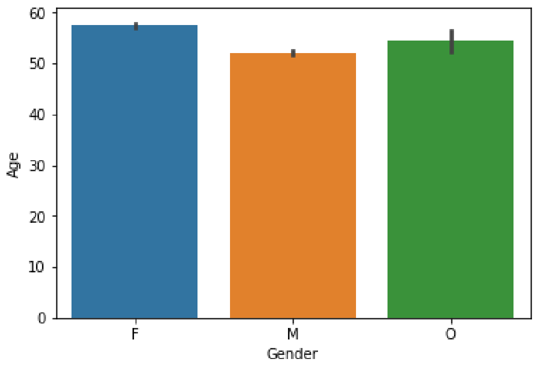
- age (int) - age of the customer
- became_member_on (int) - date when customer created an app account
- gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
- id (str) - customer id
- income (float) - customer's income

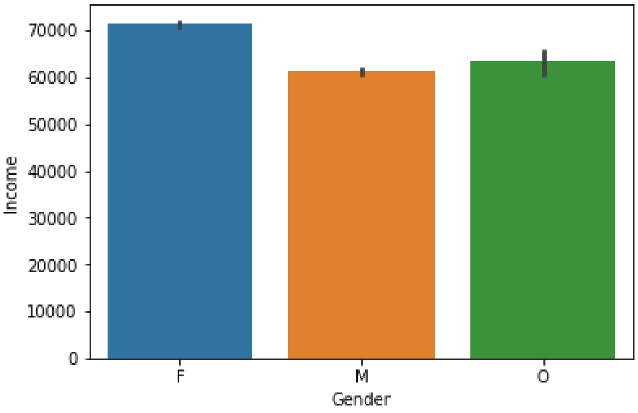
transcript.json

- 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

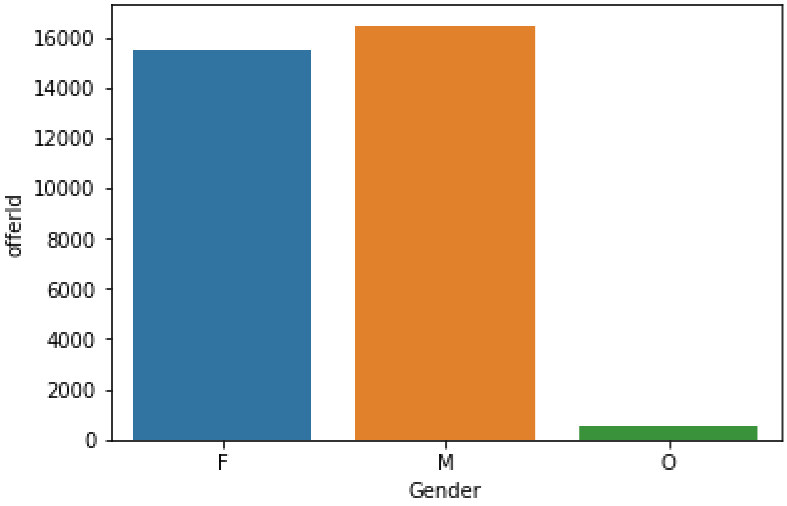
Exploration and visualization of datasets:

The unique customers in the dataset are 17000 and ages,genders and income comparison as follows:





- Only 44% customers completed the offer from who received.
- Offer completed by both Male and Female are almost the same



Methodology

Data Preprocessing

- One hot encoding for portfolio,offertype and channels columns ,since they are categorized.
- Split amount from transaction,Amount column and framing in Amount column.
- Renaming all column names to better format.
- Merging all three datasets to one dataframe.

Algorithms and Techniques:

The Algorithm is going to be a classifier because we are going to predict whether a customer completed an offer after receiving it.

The features considered are
MinAmountRequired,Reward,bogo,discount,informational,email,mobile,social,web

The target : Ground truth(Label) Offer completed (1 or 0)

Classification Algorithms used are as follows:
GaussianNB
KNeighborsClassifier
AdaBoostClassifier

Results

The benchmark was achieved by using AdaBoostClassifier which is 0.97 and I believe there is no room for improvement and it may be overfitting.
The metrics used is F1 score because it evaluated good accuracy by leveraging precision and recall in classification problems.

Improvements:

Support vector classifiers have large Time complexity which might give better accuracy,but avoided due to resource constraint.
The result which may overfit problem better feature engineering ang hyperparameter may result in great accuracy.