Machine Learning Engineer Capstone Project

Starbucks Capstone Challenge

Udacity Submission

Ashley Blanche

1. Background

Starbucks was founded in 1971 in Seattle, Washington, by three partners: Jerry Baldwin, Zev Siegl, and Gordon Bowker where initially high-quality coffee beans and equipment were sold rather than brewed coffee. In 1982, Howard Schultz joined the company and, inspired by Italian coffeehouses, transformed Starbucks into a coffeehouse chain serving espresso-based drinks. Schultz acquired the company in 1987, expanding rapidly beyond Seattle, including international markets. Starbucks went public in 1992, fuelling further growth and innovation, e.g. the Frappuccino and in-store Wi-Fi and mobile apps. Today, Starbucks operates over 30,000 stores worldwide, known for its premium coffee, commitment to sustainability, and continuous innovation.

The Starbucks mobile app enhances customer convenience by allowing users to place orders in advance, make payments, and customise their drinks. Integrated with the Starbucks Rewards program, the app lets users earn and redeem stars for free items. It also features a store locator, provides menu and nutritional information, and offers personalised promotions. 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.

2. Project Overview

The goal is to determine which offer should be sent based off of previous responses provided to previous offers. Not every single user receives the same offers. To resolve this problem, the data provided by Starbucks i.e. the json files will be utilised and I will then construct a machine learning model in order to provide predicts to the responses for customer offerings.

3. Data Sets

The data for the Starbuck's challenge 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)

The portfolio.json dataset does contain columns such as the 'offer_type' column that describes the following offers which Starbuck's may send to customers:

1. BOGO - When an additional same, free item is provided at no additional cost. There is a threshold spend to unlock reward.
2. Discount – Percentage off of Starbuck items/products.
3. Information – Details on Starbuck products.

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

4. Project Design Flow

1. Opening a workspace environment utilising AWS Sagemaker to do so.
2. Cleaning Starbuck's data provided within the portfolio, profile and transcription .json files.
3. Utilising cleaned data to build different models and evaluate metrics.
4. Using a model to benchmark
5. Summarise findings and project work in blog post.

5. Benchmark Modelling and Evaulation Metrics

KNeighboursClassifier was used as a benchmarking model due to it's speed and accuracy, ease of implementation as well as versatility being both a binary and multi-class which can be applied to classification problems. Utiliising this model I was able to model results for F1 scores with an evaluation metric.

The F1 score has been utilised as an evaluation metric to determine which model provides the best results and provides a weighted average for precision. The F1 scores reach a value between one and zero, where the closer to one you are, the better the metric score.

6. Data Analysis and exploration

As per section 3 above, the data provided consist of three different .Json files these are the portfolio.json file, the profile.json file and the transcript.json file.

Exploration of the portfolio.json file showed that there is three kinds of offer type:

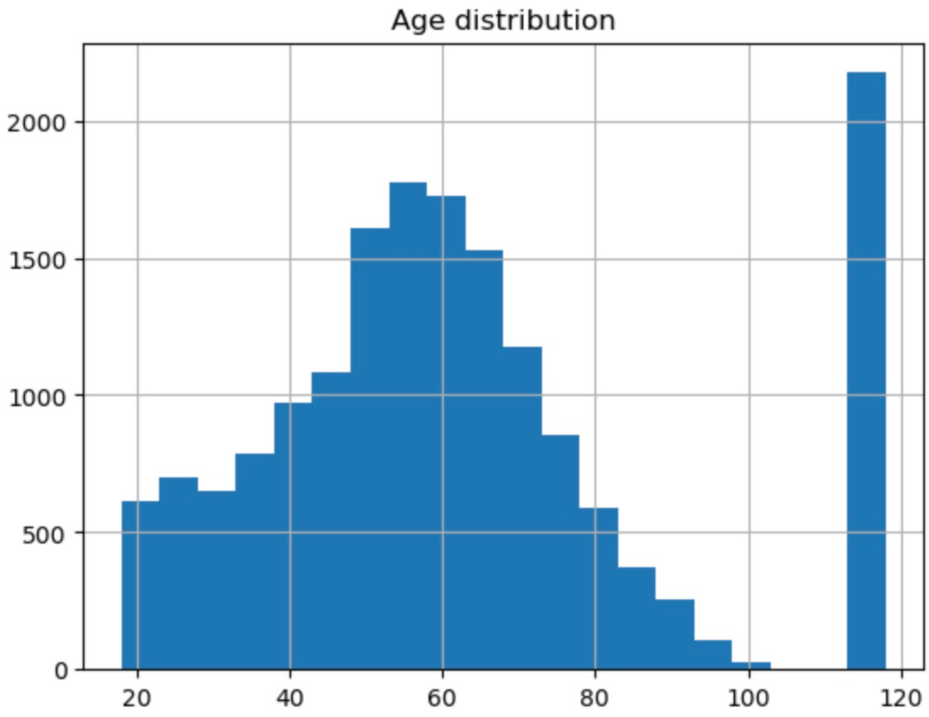
- BOGO
- Informational
- Discount

For channels, we have:

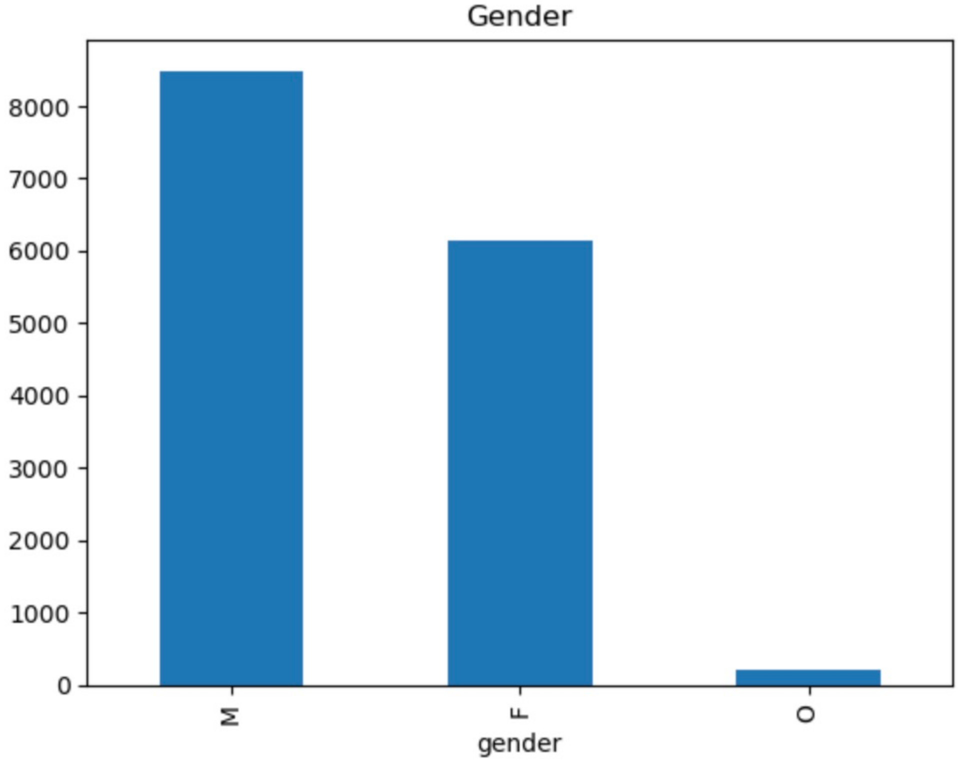
- Web
- Email
- Mobile
-
- Social

It is expected that different channel might have different influence on users. However, given the data set has multiple variables, I will try to bundle them together and not look at a specific variable.

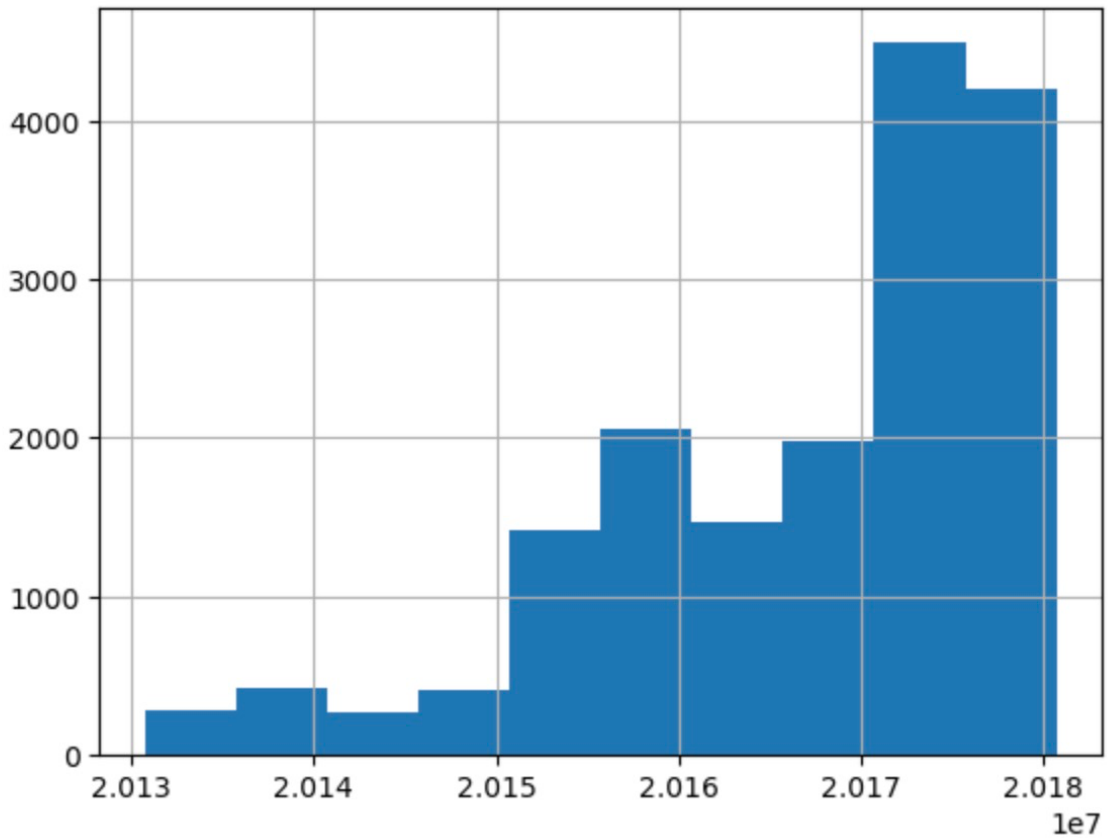
When exploring the Profile.json file this showed a number of customers who were 118 years old, as per the plot below:



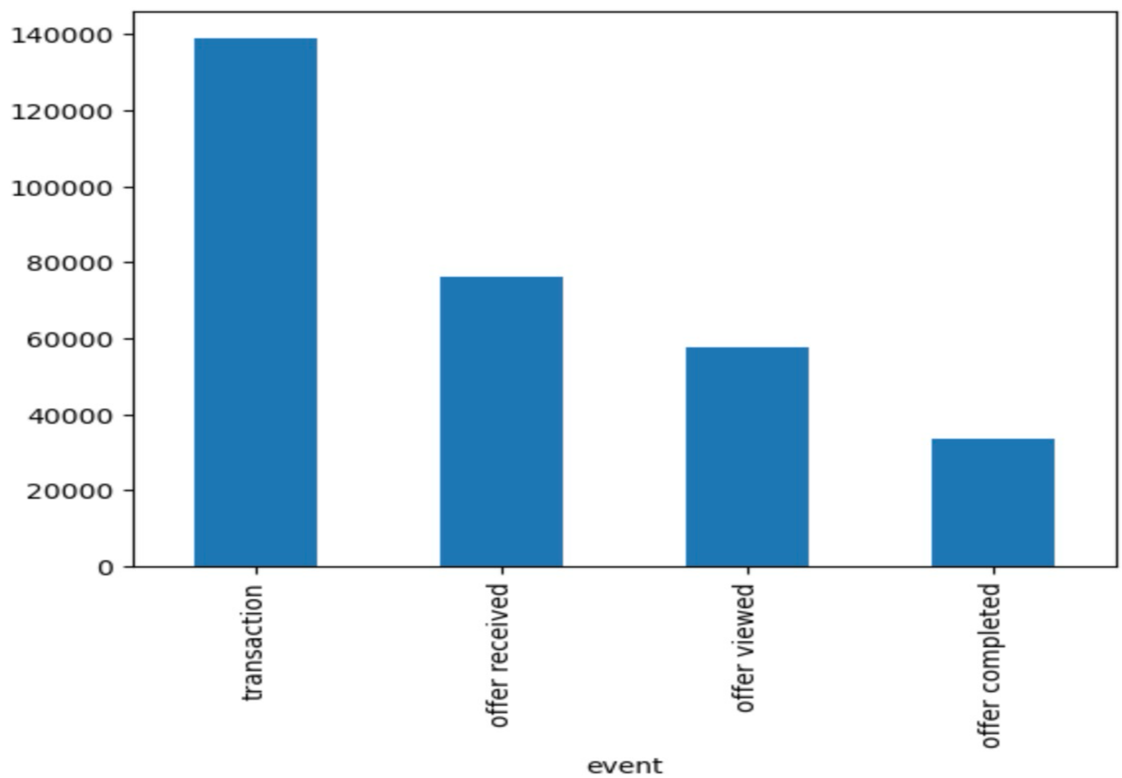
The breakdown is as follows below on gender:



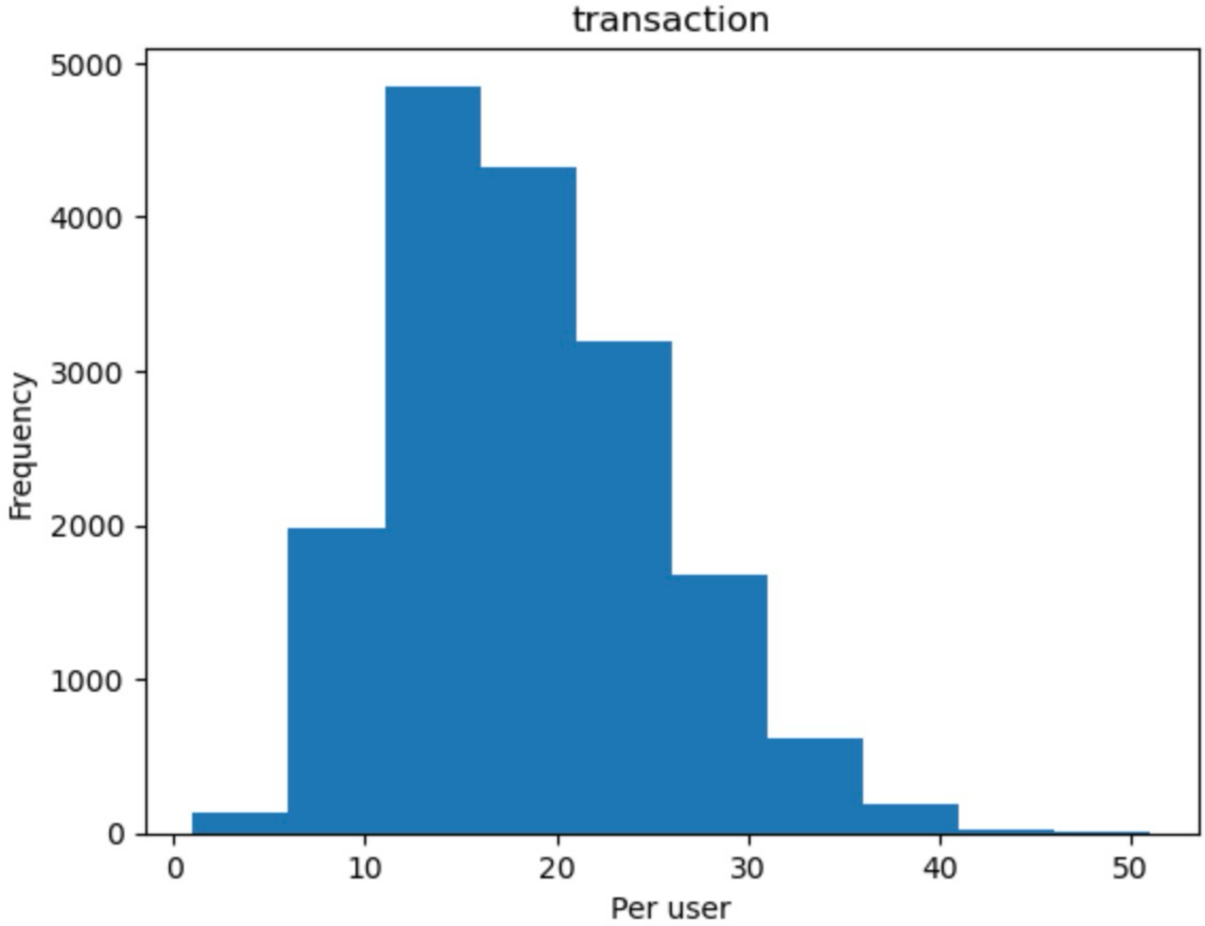
The plot below shows the number of people becoming members increasing year on year:



When exploring the transcript.json file, there is four types of events. Transaction, offer recieved and offer completed. They are reasonably balanced as represented in the graph below:



The number of transactions per customer has been also represented below in graphical form:



7. Data Preparation and Cleaning

Before building the model the data had to be prepared and cleaned in order to fit the model. All three data sets were cleaned and merged, this involved the following steps:

- Encoding te category data e.g. gender, channel, age groups etc.
- Encoding event data to numerical values i.e. 'offer received' =1, 'offer viewed' = 2 and 'offer completed' = 3
- Ecoding offer id and customer id as well as scaling and normalising data.

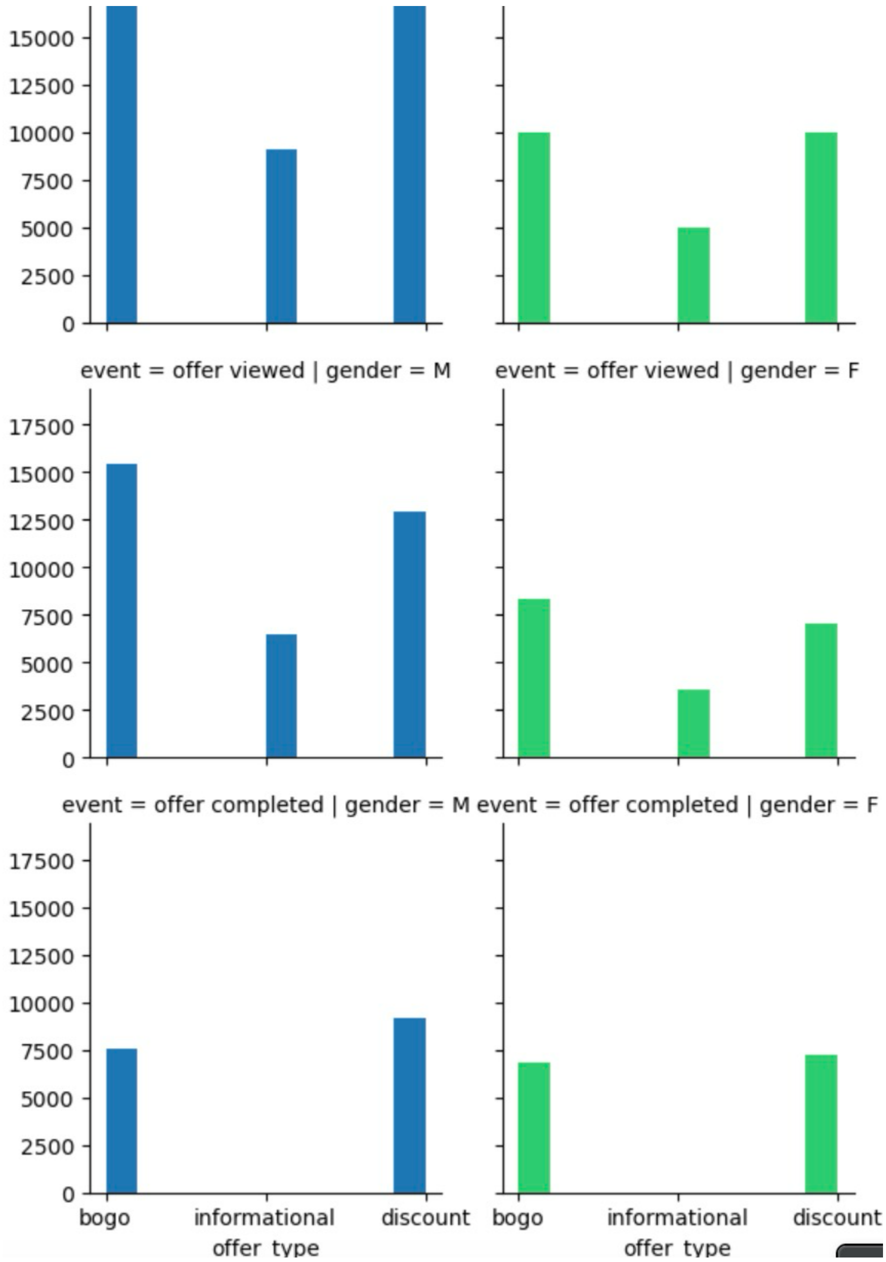
8. Traning and Testing

The F1 score is a model metric for which model provides the better scores. The K-Nearest Neighbors algorithm was utilised as a benchmark, and aided in the evaluation of the models via F1 results. The other algorithms utilised were the Random Forest Classifier and Decision Tree Classifier

9. Conclusion

The male Gender represented 63% of the Starbucks data. This 63% utilise the Starbucks app more than the female gender. Men and women in age range 41-60 utilise the Starbuck's app more than the other age ranges. It was also found that discounts were the event of choice by

Starbucks users. There is a lesser amount of Starbuck's customers who go through the whole offer to those who just view and/or ignore the offers. This can be seen in graphical form below:



The validation set (test data set) was utilised in order to provide an overview of the model. Both Random Forest Classifier and Decision Tree Classifier are better than the benchmark K-Nearest Neighbors. The highest score was the DecisionTreeClassifier model, with an F1 score of 84.67, which is greater than the KNeighborsClassifier. The RandomForestClassifier model scored well compared to the KNeighborsClassifier, with an F1 score of 70.07. in order to solve our problem we required a high F1 score which has been obtained, as such the scores are sufficient for the classification requirement to predict if a customer will respond to a Starbuck's offer.

10. References

- https://www.starbucks.com/about-us/
- https://pandas.pydata.org/docs/user_guide/reshaping.html#get-dummies-and-from-dummies
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/autopilot-metrics-validation.html
- https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/im-metrics-use.html#im-f1-metric
- https://en.wikipedia.org/wiki/F-score
- https://seaborn.pydata.org/generated/seaborn.FacetGrid.html#seaborn-facetgrid
- https://stackoverflow.com/questions/45759891/correct-use-of-map-for-mapping-a-function-onto-a-df-python-pandas
- https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html