# Starbucks Capstone Challenge

**A capstone project for Data Science Nanodegree Program from Udacity**

## I. Project Definition

### a. Project Overview

Starbucks is one of the leading coffeehouse chains in the world with more than 30 thousand stores all over the world. Starbucks offers a free app that can be used to make orders, receive special offers, and get waiting times, etc. In this project, simulated data sets are provided by Starbucks that mimics customers' behavior on different kinds of rewards sent through the mobile app. Starbucks periodically sends various individual offers related to its products to its customers through its mobile app. An offer can be 'informational' about the product, or actual offers like buy-one-get-one-free (BOGO) and discount. These marketing campaigns have associated costs and the companies do not want to spend money sending offers to customers that are not likely to buy their products or attract new customers to buy their product. Therefore, companies must identify the target group of people who are more likely to respond to their offers. This is the challenge to address the given data sets.

### b. Problem Statement

In these simulated data, Starbucks has provided simulated data for only one product, whereas in reality, the company sells many products. We are provided with the demographic data of the customers and the transcripts of customers using the app. People respond to the offers in a different way; some prefer to make the best use of the offers while others do not entertain them. The appropriate offers are the ones that a customer views and completes. On the other hand, customers so buy products without any offer are also of interest to the company to minimize their advertising cost.

Out of several possible challenges that can be addressed through this data, our goals are:

1. To find the offer that a customer ends up in buying the Starbucks product. In other words, the project aims to find the most appropriate offer for each customer.
2. To find the customer who buys the product without the influence of the offer.

**These goals are achieved by adopting the following steps:**

- Exploring and analyzing data.
- Cleaning and preprocessing the data.

- Transforming the data through scaling and numerical features.
- Training several machine learning models.
- Evaluate and compare the models' performance using the chosen metric, in our case accuracy.
- Choosing the best model.
- Making predictions from the model.

### c. Metrics

We will use accuracy to evaluate our models' performance.

The choice of this metric is because of the nature of our problem. Accuracy directly gives what percentage of predictions compares with the actual values which is a good metric for a classification problem like ours.

# II. Analysis

## Data Exploration and Data Visualization

There are three data files:

1. **portfolio.json** — containing offer ids for different offer types and metadata about each offer type. This data set has 10 rows and 6 columns (10X6).
    i. reward (int): money awarded for completing the offer.
    ii. channels (list): web, email, mobile, social
    iii. difficulty (int): money required to be spent to get the reward.
    iv. duration (int): time in days for the offer to be open.
    v. offer_type (string): a type of offer (BOGO, discount, informational)
    vi. id (string): offer id

**Interesting observations on the exploration of portfolio data:**

- There are three types of offers that can be sent: BOGO, discount, and informational. Informational offer doesn't offer any reward, while BOGO and discount offers have some reward upon completion of the offer.

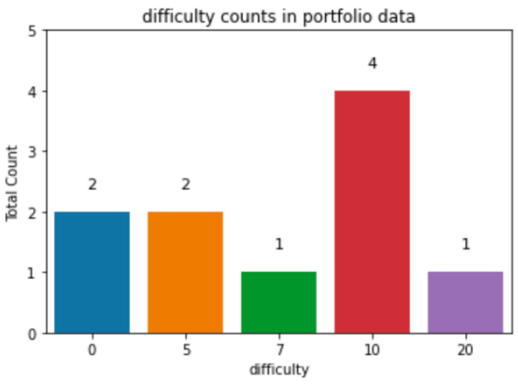
- The offers can be sent through different channels: web, email, mobile, and social. There are four offers each classified as BOGO and discount depending on the reward and difficulty. Two other offers are classified as informational.

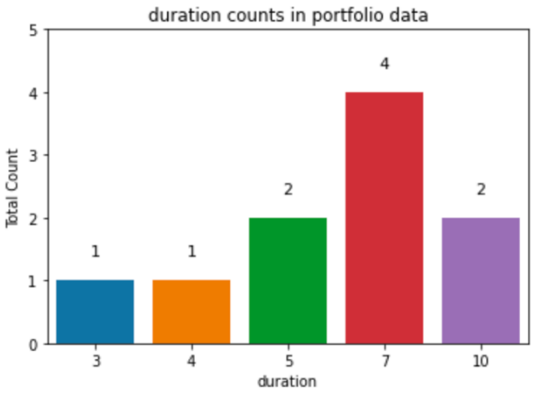
- There are 5 different reward amounts: 0 (no reward), 2, 3, 5, and 10. The greatest number of rewards sent is 5 (3 times).
- There are 5 different difficulty amounts: 0 (no difficulty), 5, 7, 10, and 20. The difficulty amount that dominant is 10 (4 times).
- There are 5 different durations in days: 3, 4, 5,7 and 10 days. The maximum duration of the offer being the 7days (4 times).

## Data visualizations for portfolio data as discussed above

| | reward | channels | difficulty | duration | offer_type | id |
|---|---|---|---|---|---|---|
| **0** | 10 | [email, mobile, social] | 10 | 7 | bogo | ae264e3637204a6fb9bb56bc8210ddfd |
| **1** | 10 | [web, email, mobile, social] | 10 | 5 | bogo | 4d5c57ea9a6940dd891ad53e9dbe8da0 |
| **2** | 0 | [web, email, mobile] | 0 | 4 | informational | 3f207df678b143eea3cee63160fa8bed |
| **3** | 5 | [web, email, mobile] | 5 | 7 | bogo | 9b98b8c7a33c4b65b9aebfe6a799e6d9 |
| **4** | 5 | [web, email] | 20 | 10 | discount | 0b1e1539f2cc45b7b9fa7c272da2e1d7 |





2. **profile.json** — The profile data contains demographic data for each customer. There are 17000 customers and 5 fields, so the shape is 17000 X 5.
    i. gender (categorical): customer's gender (M: male, F: Female, Others and None)
    ii. age (int): customer's age
    iii. id (str): customer id
    iv. became_member_on (date): the date when the customer created an app
    v. income (int): customer's income

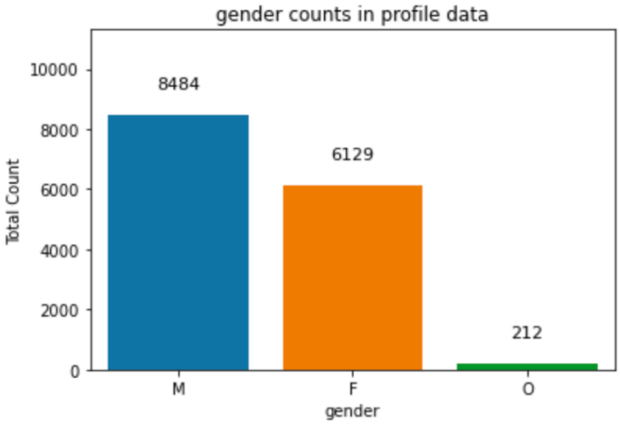
**Interesting observations on the exploration of profile dataset:**

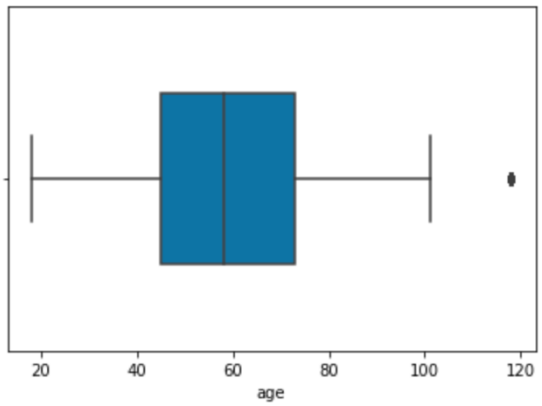
- There are no duplicate rows.
- There are 2175 null values in gender and income columns.
- All the missing income and gender have age 118.
- There are 8484 males, 6129 females, and 212 others.
- The minimum age reported is 18 and the maximum is 118. Most of the customers are within the age group of 40-80 years.
- Splitting income based on gender shows the number of men with income less than 80,000 is considerably higher than the number of females. The scatter plot of income versus age for different gender groups shows a strong correlation between some ranges of age and income.

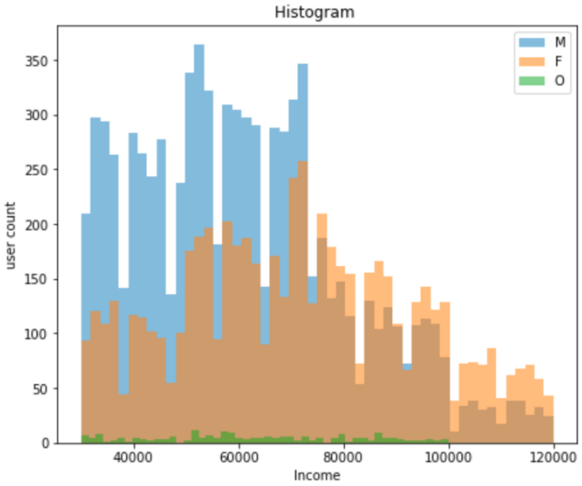
## Data visualizations for profile data as discussed above

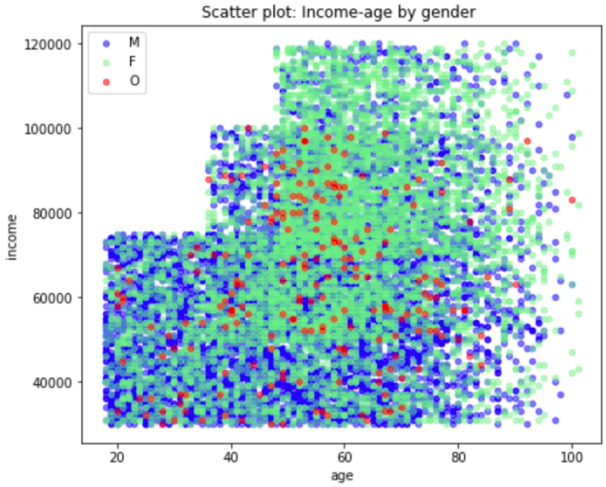
| | gender | age | customer_id | became_member_on | income |
|---|---|---|---|---|---|
| 0 | None | 118 | 68be06ca386d4c31939f3a4f0e3dd783 | 20170212 | NaN |
| 1 | F | 55 | 0610b486422d4921ae7d2bf64640c50b | 20170715 | 112000.0 |
| 2 | None | 118 | 38fe809add3b4fcf9315a9694bb96ff5 | 20180712 | NaN |
| 3 | F | 75 | 78afa995795e4d85b5d9ceeca43f5fef | 20170509 | 100000.0 |
| 4 | None | 118 | a03223e636434f42ac4c3df47e8bac43 | 20170804 | NaN |
| 5 | M | 68 | e2127556f4f64592b11af22de27a7932 | 20180426 | 70000.0 |
| 6 | None | 118 | 8ec6ce2a7e7949b1bf142def7d0e0586 | 20170925 | NaN |
| 7 | None | 118 | 68617ca6246f4fbc85e91a2a49552598 | 20171002 | NaN |
| 8 | M | 65 | 389bc3fa690240e798340f5a15918d5c | 20180209 | 53000.0 |
| 9 | None | 118 | 8974fc5686fe429db53ddde067b88302 | 20161122 | NaN |

Shape of profile data set: (17000, 5)




3. **transcript.json** — This dataset contains the event log for each customer. There are 306534 events and 4 fields in the dataset, so the size is 306534 by 4.
    i. person (str): customer id
    ii. event (str): offer received, offer viewed, offer completed and transaction.
    iii. value (dictionary of strings): different values: offer id, amount, reward.
    iv. Time (int): time in hours since the start of the test. The data begins at time t=0.

**Interesting observations on the exploration of profile dataset:**

- There are no missing values.
- There are four types of events in the event column: offer received, offer viewed, offer completed and transaction. Out of 76277 offers received, 57725 offers are viewed and 33579 offers are completed. There are 138953 total transactions.

- For analysis, the value column is split into four columns: offer id, amount, offer_id and rewards. ‘offer id’ is for the offer received and viewed while ‘offer_id’ is for the offer completed with a reward value. There is an amount associated with each transaction.
- All four unique events can happen to a person at the same time.

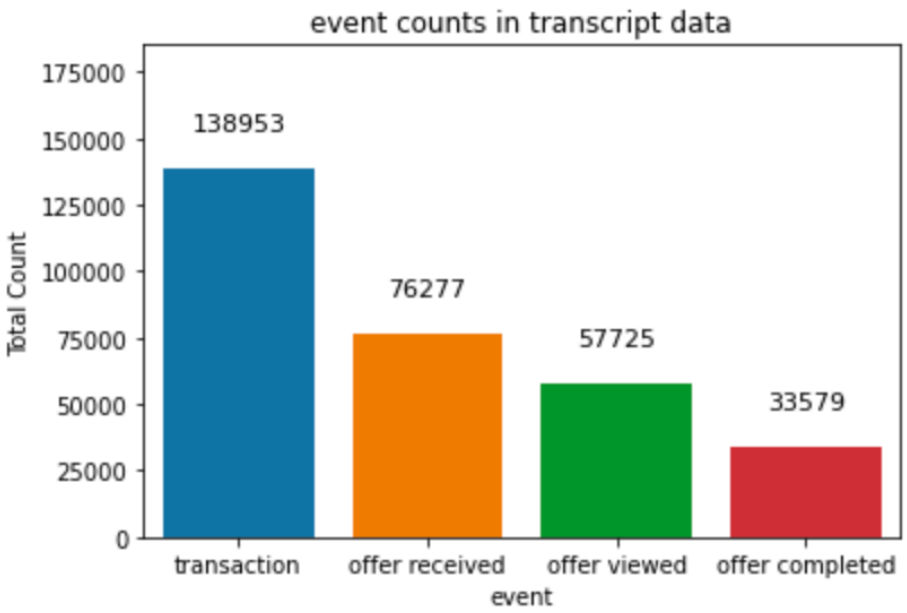
## Data visualizations for profile data as discussed above

| | person | event | time | offer id | amount | offer_id | reward |
|---|---|---|---|---|---|---|---|
| **0** | 78afa995795e4d85b5d9ceeca43f5fef | offer received | 0 | 9b98b8c7a33c4b65b9aebfe6a799e6d9 | NaN | NaN | NaN |
| **1** | a03223e636434f42ac4c3df47e8bac43 | offer received | 0 | 0b1e1539f2cc45b7b9fa7c272da2e1d7 | NaN | NaN | NaN |
| **2** | e2127556f4f64592b11af22de27a7932 | offer received | 0 | 2906b810c7d4411798c6938adc9daaa5 | NaN | NaN | NaN |
| **3** | 8ec6ce2a7e7949b1bf142def7d0e0586 | offer received | 0 | fafdcd668e3743c1bb461111dcafc2a4 | NaN | NaN | NaN |
| **4** | 68617ca6246f4fbc85e91a2a49552598 | offer received | 0 | 4d5c57ea9a6940dd891ad53e9dbe8da0 | NaN | NaN | NaN |

| | person | event | time | offer id | amount | offer_id | reward |
|---|---|---|---|---|---|---|---|
| **12650** | 389bc3fa690240e798340f5a15918d5c | offer viewed | 0 | f19421c1d4aa40978ebb69ca19b0e20d | NaN | NaN | NaN |
| **12651** | d1ede868e29245ea91818a903fec04c6 | offer viewed | 0 | 5a8bc65990b245e5a138643cd4eb9837 | NaN | NaN | NaN |
| **12652** | 102e9454054946fda62242d2e176fdce | offer viewed | 0 | 4d5c57ea9a6940dd891ad53e9dbe8da0 | NaN | NaN | NaN |
| **12653** | 02c083884c7d45b39cc68e1314fec56c | offer viewed | 0 | ae264e3637204a6fb9bb56bc8210ddfd | NaN | NaN | NaN |
| **12655** | be8a5d1981a2458d90b255ddc7e0d174 | offer viewed | 0 | 5a8bc65990b245e5a138643cd4eb9837 | NaN | NaN | NaN |

| | person | event | time | offer id | amount | offer_id | reward |
|---|---|---|---|---|---|---|---|
| **12654** | 02c083884c7d45b39cc68e1314fec56c | transaction | 0 | NaN | 0.83 | NaN | NaN |
| **12657** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | transaction | 0 | NaN | 34.56 | NaN | NaN |
| **12659** | 54890f68699049c2a04d415abc25e717 | transaction | 0 | NaN | 13.23 | NaN | NaN |
| **12670** | b2f1cd155b864803ad8334cdf13c4bd2 | transaction | 0 | NaN | 19.51 | NaN | NaN |
| **12671** | fe97aa22dd3e48c8b143116a8403dd52 | transaction | 0 | NaN | 18.97 | NaN | NaN |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **36** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | offer received | 0 | 2906b810c7d4411798c6938adc9daaa5 | NaN | NaN | NaN |
| **12656** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | offer viewed | 0 | 2906b810c7d4411798c6938adc9daaa5 | NaN | NaN | NaN |
| **12657** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | transaction | 0 | NaN | 34.56 | NaN | NaN |
| **12658** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | offer completed | 0 | NaN | NaN | 2906b810c7d4411798c6938adc9daaa5 | 2.0 |
| **27850** | 9fa9ae8f57894cc9a3b8a9bbe0fc1b2f | transaction | 42 | NaN | 21.55 | NaN | NaN |




# III. Methodology

## a. Data preprocessing

Data preprocessing is done on all three data sets.

### Portfolio data:

i. Column ‘id’ is renamed to ‘offer_id’.
ii. Column ‘offer_type’ contains categorical data which is converted into one-hot encoding.
iii. Similarly, ‘channel’ columns contain categorical data with different categories. Separate columns (emails, mobile, social, and web) are created for each category. The original channel column is dropped.
iv. Since the ‘email’ channel is used by all the offer types, the column is dropped.

### Profile data:

i. The ‘person’ column is renamed to ‘customer_id’.
ii. The ‘gender’ column consists of distinct categorical data: Male, Female, Others, and None (missing values). The missing gender is assigned to a separate category ‘N’. Then the column is converted into one-hot encoding.
iii. The income column comprises continuous variables; therefore, the missing values are imputed by mean.

iv. The ‘became_member_on’ column long is converted into a long format.

## Transcript data:

i. The ‘person’ column is renamed to ‘customer_id’.
ii. The ‘value’ column is split into its four values: ‘offer id’, ‘amount’, ‘offer_id’, and ‘reward’. The original value column is dropped.
iii. The two columns ‘offer id’ and ‘offer_id’ are merged.
iv. The offer was sent to almost everyone (16994 out of 17000 customers received the offer). So, at first, we filtered out the customers who received the offers.
v. The ‘amount’ and ‘reward’ columns corresponding to ‘offer received’ is NaN. We dropped those two columns. The final offer received data frame has the shape of 76277x10.
vi. As per the proposed problem, three labels are created: 1) label 0: No transactions 2) label 1: transaction under the influence of offering 3) label 2: transaction without using the offer. The labels are filtered through looping through the time for which the offer remains valid and the event performed by each customer in that time.
vii. After assigning the labels, the data frames are merged.
viii. In the merged data frame, the ‘income’ column is preprocessed to income groups. Every income is assigned to three income groups: 1) customers with income less than 60,000 are assigned to group 1, 2) customers with income between 60,000 and 90,000 are assigned to group 2, and 3) customers with income greater than 90,000 are assigned to group three. The column is named as ‘income_group’.
ix. Similarly, the ‘age’ column is also divided into three age groups. The customers with age less than 40 years are placed in group 1, the ones with age between 40 years and 80 years are placed in group 2 and the ones with age greater than 80 years are placed in group 3. The colum is named as ‘age_group’.
x. The columns ‘offer type’ and ‘gender’ are added to the merged data set for the analysis of combined data set.
xi. The final data frame is saved as a pickle file ‘merged_data.pkl’.
xii. To explore the demographic distribution of offers, a new data frame with event and gender columns are created and saved as pickle file ‘event_gender_data.pkl’

Final, merged data frame is shown below:

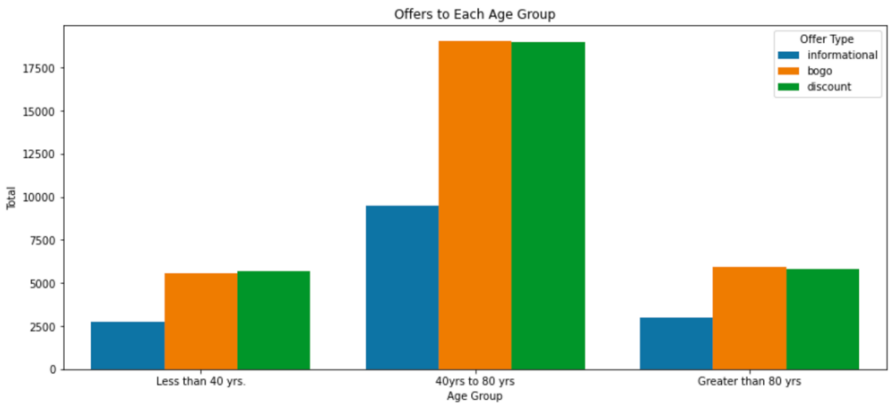
| | customer_id | time | event | offer_id | label | reward | difficulty | offer_duration_hrs | bogo | discount | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0009655768c64bdeb2e877511632db8f | 168 | offer received | 5a8bc65990b245e5a138643cd4eb9837 | 2 | 0 | 0 | 72 | 0 | 0 | ... |
| 1 | 0009655768c64bdeb2e877511632db8f | 336 | offer received | 3f207df678b143eea3cee63160fa8bed | 2 | 0 | 0 | 96 | 0 | 0 | ... |
| 2 | 0009655768c64bdeb2e877511632db8f | 408 | offer received | f19421c1d4aa40978ebb69ca19b0e20d | 0 | 5 | 5 | 120 | 1 | 0 | ... |
| 3 | 0009655768c64bdeb2e877511632db8f | 504 | offer received | fafdcd668e3743c1bb461111dcafc2a4 | 0 | 2 | 10 | 240 | 0 | 1 | ... |
| 4 | 0009655768c64bdeb2e877511632db8f | 576 | offer received | 2906b810c7d4411798c6938adc9daaa5 | 1 | 2 | 10 | 168 | 0 | 1 | ... |

5 rows × 25 columns

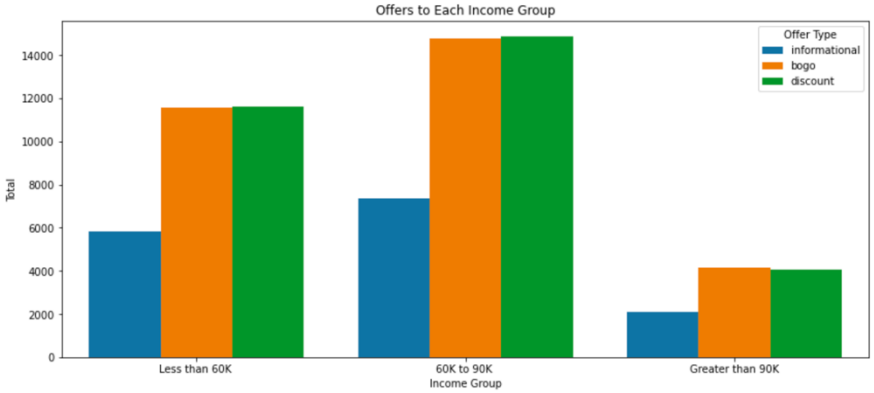
| became_member_on | income | F | M | N | O | income_group | age_group | offer_type | gender |
|---|---|---|---|---|---|---|---|---|---|
| 1492732800000000000 | 72000.0 | 0 | 1 | 0 | 0 | 2.0 | 1.0 | informational | M |
| 1492732800000000000 | 72000.0 | 0 | 1 | 0 | 0 | 2.0 | 1.0 | informational | M |
| 1492732800000000000 | 72000.0 | 0 | 1 | 0 | 0 | 2.0 | 1.0 | bogo | M |
| 1492732800000000000 | 72000.0 | 0 | 1 | 0 | 0 | 2.0 | 1.0 | discount | M |
| 1492732800000000000 | 72000.0 | 0 | 1 | 0 | 0 | 2.0 | 1.0 | discount | M |

## Analysis of final data set:

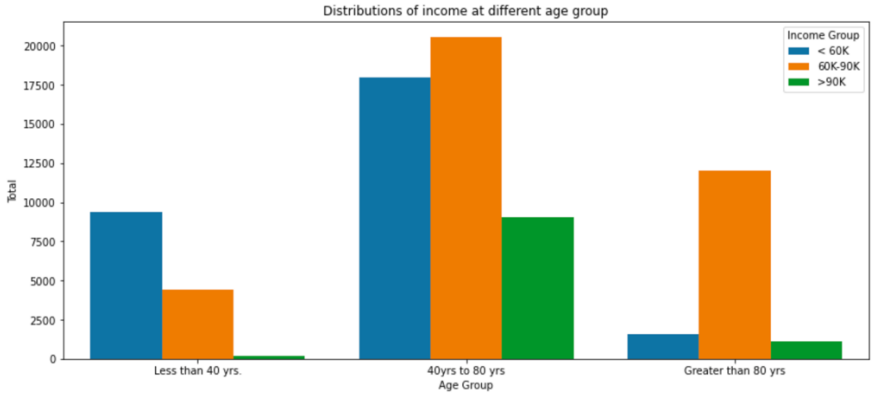
i. Exploring the offer sent to various age groups shows that the BOGO and discount are the most popular offer types for all age groups.



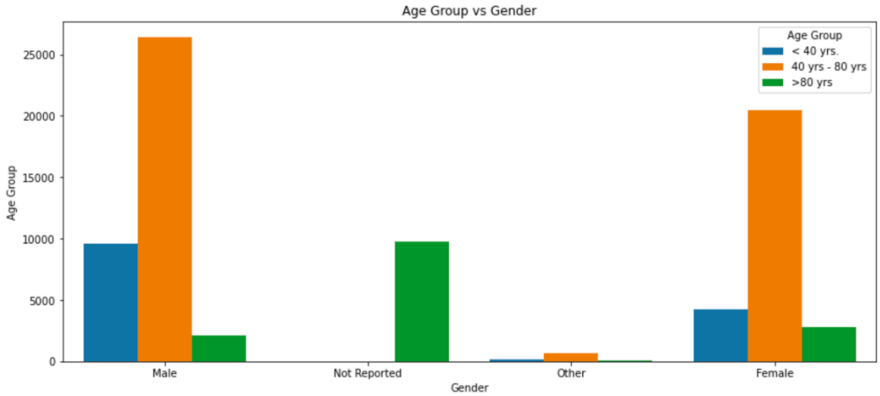

ii. Exploring the offer sent to various income groups shows the BOGO and discount as the most popular offer types for all income groups.



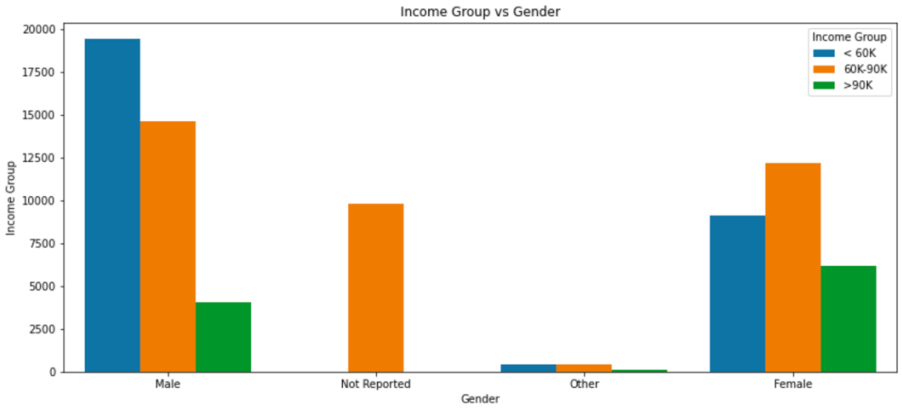

iii. Exploring the income for various age groups shows that the customers in the age group less than 40 years are dominated by the income of less than 60K. On the other hand, customers with the age group between 40 years and 80 years have an income range between 60K and 90K. There is a significant number of customers who fall in the high-income regime (>90K) with the age group greater than 80 years.



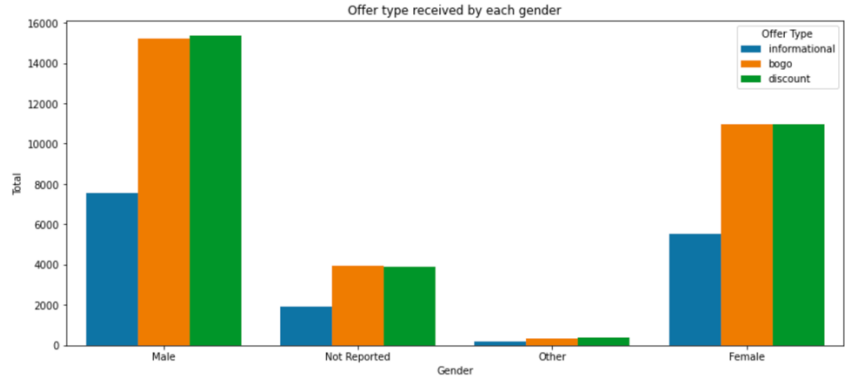

iv. All the genders are proportionately distributed among different age groups.



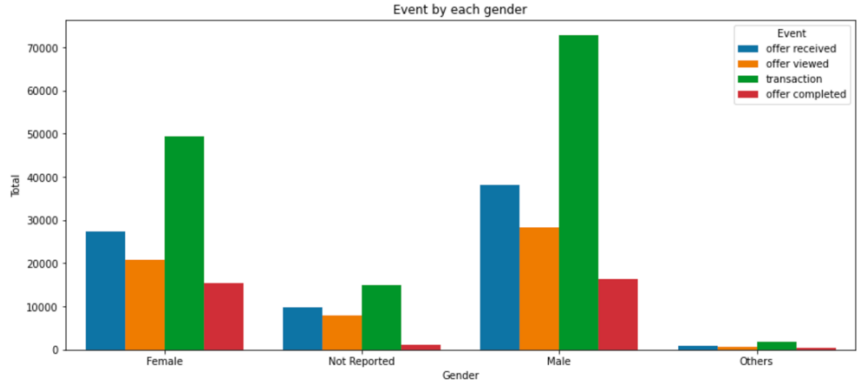

v. Customers with higher income (>90K) are mostly females and those with medium income (60k-90K) are male.




vi. All three types of offers are distributed in a similar way among all the genders. The dominant offer types distributed are BOGO and discount.




vii. All gender group performed higher transactions than the offer received. Significant number of offers received are completed.




## b. Implementation

As the first step of implementation, data exploration is carried out on the *Data_exploration* notebook. Some issues were discovered in all three data sets that were addressed in the *Data_cleaning* notebook. After cleaning the data and some feature engineering as explained above, final data analysis is carried out on the *EDA* notebook. This exploration provided further insight and connections

between the various features of the data sets. As a final step, we have trained some machine learning models to predict the appropriate offer to a customer on the basis of whether the customer will complete the transaction under the influence of the offer or not. The machine learning (ML) algorithms implemented are discussed below.

## Algorithms and Technique

Final data transformation is carried out on the notebook *Data_modeling* to train the ML models.

*i.* The correlation matrix for the features showed that the *age* and *age_group* features are correlated with other features in the same way. We dropped the age feature and considered only the age group.

ii. The numerical features (reward, difficulty, offer_duration_hrs, income and, became_memeber_on) are scaled using the minMaxScalar from the scikit.

iii. The data is split into 70% train set and 30% test set using the train_test_split from scikit.

We tried the following five models and considered the model accuracy score as the metric to evaluate the model quality.

i. **Neural Network:** A simple feed-forward neural network (NN) has been implemented to classify the response to the offers. This model is considered as a benchmark and all the other models' performance is compared with it.

ii. **Decision Tree Classifier**

iii. **K-nearest Neighbors**

iv. **Random Forest**

v. **Support Vector Machine**

The features used in training the models are a reward, difficulty, offer_duration_hrs, BOGO, discount, informational, mobile, social, web, age_group, became_member_on, income, income_group, F, M, N, O.

As described in the Metric section, accuracy is used as a metric to evaluate the model's performance.

**c. Refinement**

For the Random Forest and K-nearest neighbor, we did the grid search to refine the models, which will be further described in the Results section.

# IV. Results

## a. Model Evaluation and Validation

All the five models described above are trained on the training set and evaluated on the testing set. The size of the training set is 53393 and the testing set is 2284. The comparison of the models' accuracy is shown in the table below. The benchmark model neural network has a training and testing accuracy of ~64%. Decision Tree has a training accuracy of 94.64% and a testing accuracy of 57.77%. Similarly, K-Nearest Neighbor has the training and testing accuracy of 68.82% and 61.48%, respectively. Random Forest has a training accuracy of 94.64% and a testing accuracy of 59.76%. Support Vector Machine has a training accuracy of 60.90% and a testing accuracy of 61.39%.

| | Model | Train accuracy | Test accuracy |
|---|---|---|---|
| 0 | Neural Network | 0.642818 | 0.644206 |
| 1 | Decision Tree | 0.946435 | 0.577696 |
| 2 | K-Nearest Neighbor | 0.688218 | 0.614753 |
| 3 | Random Forest | 0.946435 | 0.597579 |
| 4 | Support Vector Machine | 0.609031 | 0.613922 |

Out of five models, I have considered Random Forest and K-Nearest for further refinements. Random Forest has high train accuracy and relatively lower test accuracy. This means the model is behaving differently for the training and testing sets. On the other hand, the K-Nearest Neighbor has both the training and testing accuracy in the same range.

After refinements, the Random Forest has a training accuracy of 65.40% and a testing accuracy of 64.70%. Thus, the tuned parameters behave in a similar way for both data sets.

Similarly, after refinements, the K-Nearest Neighbor has the training and testing accuracy of 73% and 60.8%, respectively.

Out of the two refined models, we take the Random Forest as the best classifier because of its better testing accuracy.

For the nature of the problem chosen and the given data sets, we are convinced that the final model reached a good accuracy. The customers' behavior for an offer can be varying and that could be a reason for the model to have medium accuracy.

### b. Justification

It can be seen that the benchmark model **Neural Network** and our best classifier **Random Forest** have similar training and testing accuracy with Random Forest being slightly better after refinement. Also, the remaining three models have slightly less testing accuracy, which I am pretty sure, will improve after optimizing the parameters.

## V. Conclusion

To conclude**,** the data from Starbucks are successfully analyzed, cleaned, transformed, and scaled. Five machine learning models are trained from the final data set, including a Neural Network which is considered as a benchmark. Two models, Random Forest and K-Nearest Neighbors are further refined using the GridSearchCV. After refinements, the **Random Forest** model is considered as the best classifier for our problem. The Random Forest model can predict the appropriate behavior shown to the offer by a customer; whether the customer does the transaction under the influence of the offer or not.

### a. Reflection:

It was a great pleasure working on this project. I enjoyed using all my data science skills learned in the course. It provided an opportunity to consolidate my machine learning and coding knowledge.

The most challenging parts of this project were to choose the problem statement and then data preprocessing. I have enjoyed every aspect of the project.

### b. Improvement:

Few suggestions for improvements are, the data set can be diverse. The simulated data have offers distributed to various demographic groups in proportion to their number. This is not the case in real life, where there are diverse demographic distributions. More diverse data can be used to predict the offer type for different demographic groups, which is not obvious from the given data.

More sophisticated models like the recurrent neural network can be used to predict the nature of the offer to be sent over time. The customers' behavior to offers can vary with time, and this can capture that behavior more accurately.