Starbucks Capstone Project

Abdul Tamimi

20/12/2021

Project Definition

Project Overview

The Starbucks Capstone Project is about using experimental data to discover how customers respond to messages containing offers and rewards. The data is derived from the mobile app that Starbucks uses to interact with customers. The data mimics how customers make purchasing decisions and respond to offers. The aim of the project is to identify what are the offers that excite people.

The app is designed to understand what drives each individual to purchase Starbucks products. Certain customers will respond to certain offers in different ways, some positively, some negatively and some might not even respond at all. The challenge is to build a model to determine what offer should be sent based on demographics.

The offer types provided by Starbucks are: BOGO(buy one get one free), discount and informational.

The data provided by Starbucks contains the following three files:

- Portfolio.json: containing information about each offer type
- Profile.json: demographic data for each customer
- Transcript.json: records for transactions, offers received, offers viewed, and offers completed.

Problem Statement

.

Starbucks as well as many coffeeshop companies base their targets and marketing campaigns trying to understand what's best for the customer. It is very important for these companies to analyze the customer’s behaviour and reaction to initiatives, offers and rewards. By doing that, the company will ensure that the customer is provided a good service and will be interested in buying more of the company's products.

It is also very important for the company to understand that some customers are not as responsive as others. Some customers do not see the offers or rewards sent to them, this may be due to the channel used or simply that this customer is not active or a normal respondent. Other customers might receive and view the offers but don't act upon them. This may be either the customer is not satisfied with the offer type or simply because they just do not want the offer. If the customer views the offer and decides to not take advantage of it, then put it simply that this customer should either receive a different offer or should not be considered as a target. The case where a customer views and completes the offer is the one that should be considered and pursued. The customer is considered a target when he receives, views and completes the offer.

The problem this project is trying to solve is to identify which demographic groups respond best to which offer type. That is finding the offer that will lead the customer to buy that respective product or other Starbucks products.

Strategy

Since this is a classification problem, the strategy used was creating machine learning algorithm models to predict the best offer type for the customer. Before creating the model, the dataset in the files were preprocessed and the following steps were taken:

- Data Loading and Cleaning
- Data preprocessing
- Feature engineering
- Normalizing and Engineering data for Machine Learning
- Evaluate the model

Metrics

Evaluating the model is an essential part of this project. While preprocessing the files and training a model is a crucial step in creating a machine learning algorithm, measuring the performance of that model is equally important. Evaluation is done by using machine learning metrics to monitor and measure the performance of the model. The aim is to use these metrics to help understand how well the model is working. Another advantage metrics can provide, is improvements until achieving the best performance of the model. There are different metrics to evaluate the performance of machine learning algorithms. Since this is a classification problem, will use the following metrics to evaluate the model:

- Confusion Matrix: shows visualization performance of an algorithm. Confusion matrix is helpful in this problem in a sense that it helps us find how the predicted values are far from the truth values. Sense we want to predict how many customers will respond to offers, it useful to know
- Classification report: shows a report of various evaluation metrics (accuracy, f1 score, precision and recall). It is encouraging in this problem to have an insight about how different metrics respond to the model. Because 100% accuracy doesn’t necessarily mean that our model is precise or f1 score is 100%. Therefore, it's intriguing to see how the metrics will compare.

Exploratory Data Analysis

Portfolio Dataset

Data Exploration

The Schema and explanation of each variable in the portfolio file:

Portfolio.json

- id(string) - offer id
- offer_type(string)
- difficulty (int)
- reward (int)
- duration (int)
- channels (list of strings)

	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5
5	[web, email, mobile, social]	7	7	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	3
6	[web, email, mobile, social]	10	10	fafdcd668e3743c1bb461111dcafc2a4	discount	2
7	[email, mobile, social]	0	3	5a8bc65990b245e5a138643cd4eb9837	informational	0
8	[web, email, mobile, social]	5	5	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5
9	[web, email, mobile]	10	7	2906b810c7d4411798c6938adc9daaa5	discount	2

Offers are sent during the 30-day test period. There are three types of offers that can be sent: bogo, discount and informational. For a bogo offer to be valid, the customer needs to spend a certain amount to get a reward equal to that threshold amount. In a discount offer, the customer gains a reward to a fraction of the cost of something. An informational offer is just merely an advertisement. Offers can be delivered through various channels: email, mobile, social and web. Each offer has a validity period before it expires. For an offer to be completed, the customer must spend a minimum amount (duration). Reward is given to a customer after completing the offer.

According to the dataset, there are 63834 bogo offer types, 62311 discount offer types and 22660 informational offer types. The graph below shows how the three offer types vary within the dataset. There were no null values found in the dataset, for that not much was done during data exploration.

Data processing

- Rename ‘id’ column to ‘offer_id’.
- Create dummy variables from the ‘channels’ column using one-hot encoding
- Drop ‘channels’ column
- Re-index the dataset to a more representative meaning

Below is how the profile dataset looks like after data cleaning and reindexing.

	offer_id	offer_type	duration	difficulty	reward	web	email	mobile	social
0	ae264e3637204a6fb9bb56bc8210ddfd	bogo	7	10	10	0	1	1	1
1	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	5	10	10	1	1	1	1
2	3f207df678b143eea3cee63160fa8bed	informational	4	0	0	1	1	1	0
3	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	7	5	5	1	1	1	0
4	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	10	20	5	1	1	0	0
5	2298d6c36e964ae4a3e7e9706d1fb8c2	discount	7	7	3	1	1	1	1
6	fafdcd668e3743c1bb461111dcafc2a4	discount	10	10	2	1	1	1	1
7	5a8bc65990b245e5a138643cd4eb9837	informational	3	0	0	0	1	1	1
8	f19421c1d4aa40978ebb69ca19b0e20d	bogo	5	5	5	1	1	1	1
9	2906b810c7d4411798c6938adc9daaa5	discount	7	10	2	1	1	1	0

Profile Dataset

Data Exploration

The Schema and explanation of each variable in the profile file:

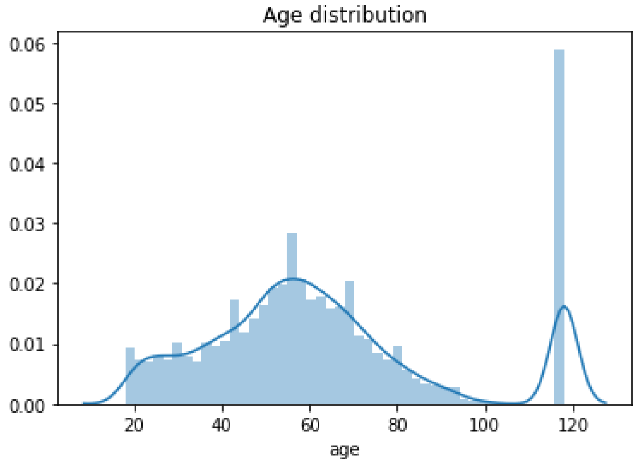
Profile.json

- age(int)
- became_member_on (int)
- gender(str)
- id(str)
- income(float)

	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN

The interpretation of the dataset above is simple. Inside the profile dataset, is more prolific information about the customers that buy from Starbucks. Information about age, income, gender and the start of the membership of each customer.

There are 8484 male customers, 6129 female customers and 212 other customers. During data exploration, 2175 null values have been found within gender and income columns. To understand how these null values affect the dataset and the accuracy in the future have decided to look at the age distribution and see the different age customers that have used the Starbucks app. Below is the graph for the age distribution. One thing to notice from the graph is the values coming from those at 118 years old.



Let's take a look at the entries related to age 118 in a more detailed way. When extracting some values related to age 118, have noticed that the values associated in gender and income columns are null values. The number of null values related to that age group is equal to the number of null values found in gender and income columns overall. Hence, can conclude that the null values in the profile dataset belong to customers who are 118 years old. These values will be dropped.

```
# dataframe for gender and income when age equal 118.
profile[['age','gender', 'income']][profile['age']==118].head(10)
```

	age	gender	income
0	118	None	NaN
2	118	None	NaN
4	118	None	NaN
6	118	None	NaN
7	118	None	NaN
9	118	None	NaN
10	118	None	NaN
11	118	None	NaN
17	118	None	NaN
23	118	None	NaN

Data processing

- Rename 'id' column to 'customer id'

- Drop customers with age = 118.
- Adjust the 'became member on' column to datetime
- Add a new column 'month_member', that will present the month at which the customer becomes a member
- Add a new column 'year_member', that will present the year at which the customer becomes a member.
- Create new columns 'AgeGroup' and 'IncomeGroup' that segments age and income columns for better visualization and further analysis.
- Drop age and income columns.
- Replace null values in Agegroup and Incomegroup columns with mode

For better understanding and visualization have grouped income and age into four segments. Also, have added two columns that describe the month and year to when the customer has stated their membership.

The 4 AgeGroups are:

- 18-40
- 41-60
- 61-80
- 81-101

The 3 IncomeGroups are:

- Low
- Medium
- High

Let's see how the profile dataset looks after data cleaning.

	customer_id	gender	AgeGroup	IncomeGroup	month_member	year_member
1	0610b486422d4921ae7d2bf64640c50b	F	41-60	high	7	2017
3	78afa995795e4d85b5d9ceeca43f5fef	F	61-80	high	5	2017
5	e2127556f4f64592b11af22de27a7932	M	61-80	medium	4	2018
8	389bc3fa690240e798340f5a15918d5c	M	61-80	medium	2	2018
12	2eeac8d8feae4a8cad5a6af0499a211d	M	41-60	medium	11	2017

Transcript Dataset

Data Exploration

The Schema and explanation of each variable in the transcript file:

Transcript.json

- event(str)
- person(str)
- Time (int)
- Value (dict of strings)

	event	person	time	value
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
2	offer received	e2127556f4f64592b11af22de27a7932	0	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}

There are 4 event types: offer received, offer viewed, transaction and offer completed. The value column contains a dictionary of strings depending on the event described. There are 4 value types: offer id, offer_id, amount and reward. Offer id and offer_id are the same and they represent random numbers and letters describing the id number of the offer. Any offers received and viewed by the customer are associated with an offer id. Amount is a number describing the transaction spent by the customer. Reward is a number the customer receives when completing an offer.

Data processing

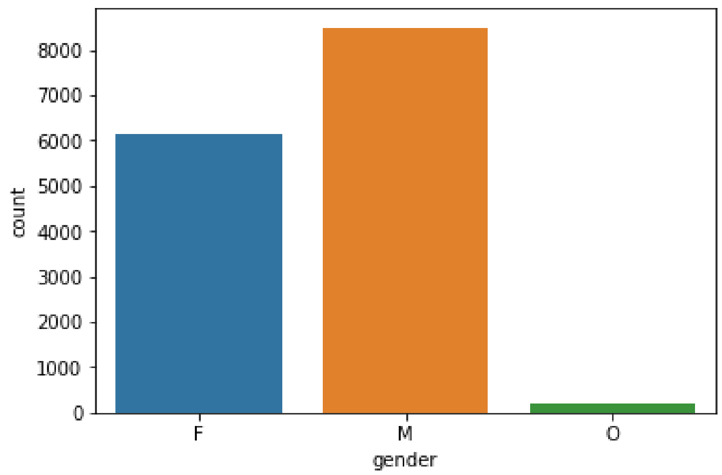
- Rename ‘person’ column to ‘customer_id’.
- Expand each key that exists in the ‘value’ column to a seperate column.
- Concatenate offer id and offer_id columns found after extracting value column into one column.
- Drop ‘value’ and ‘offer id’ columns
- Rename concatenated column to offer_id
- Replace null values with mean

	event	customer_id	time	reward	amount	offer_id
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	0.0	0.0	9b98b8c7a33c4b65b9aebfe6a799e6d9
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	0.0	0.0	0b1e1539f2cc45b7b9fa7c272da2e1d7
2	offer received	e2127556f4f64592b11af22de27a7932	0	0.0	0.0	2906b810c7d4411798c6938adc9daaa5
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	0.0	0.0	fafdcd668e3743c1bb461111dcafc2a4
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	0.0	0.0	4d5c57ea9a6940dd891ad53e9dbe8da0

Data Visualization

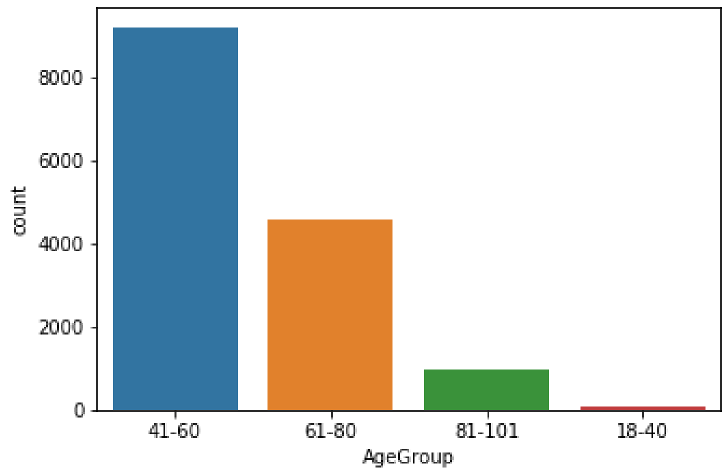
In this project we are trying to analyze what offer is sent to a customer. Lets first visualize the number of customers that we have in terms of males and females. All figures shown below are after exploring and cleaning the three datasets.

Demographic gender



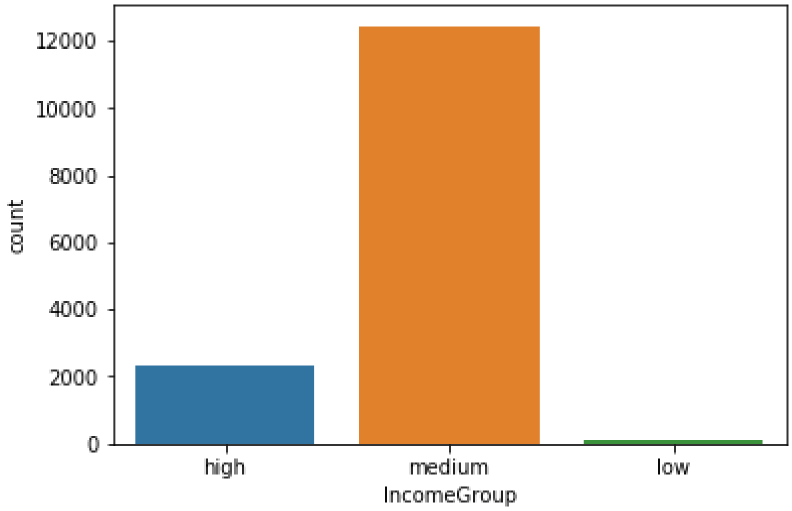
There are 8484 male customers, 6129 female customers and 212 customers who have not identified themselves.

Demographic age



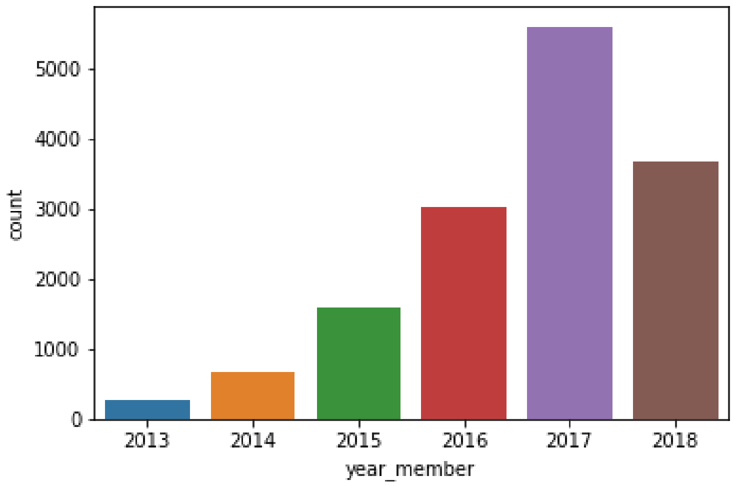
Most Starbucks customers are between 41 and 60 years old. According to the dataset, 9213 customers belong to that AgeGroup. Surprisingly, the second highest age group is 61-80 with 4556 customers in it. 986 customers belong to the 81-101 age group and only 70 customers are between 18 and 40 years old.

Demographic income



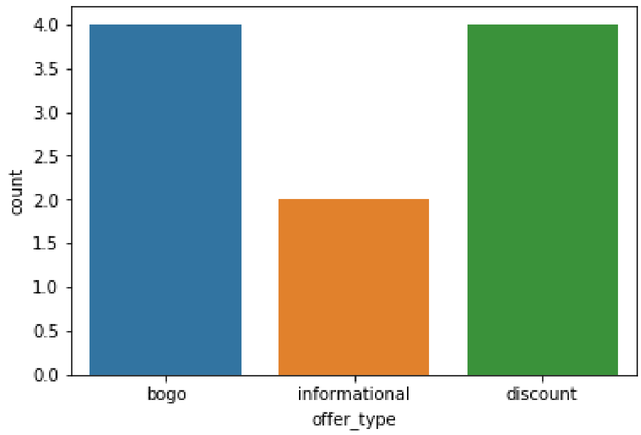
Most of the customers receive medium income. 12429 customers receive that income and only 2308 customers receive a high salary. Those who receive low income are about 88 customers and this shows that most of Starbucks customers are within the medium income group.

Demographic membership

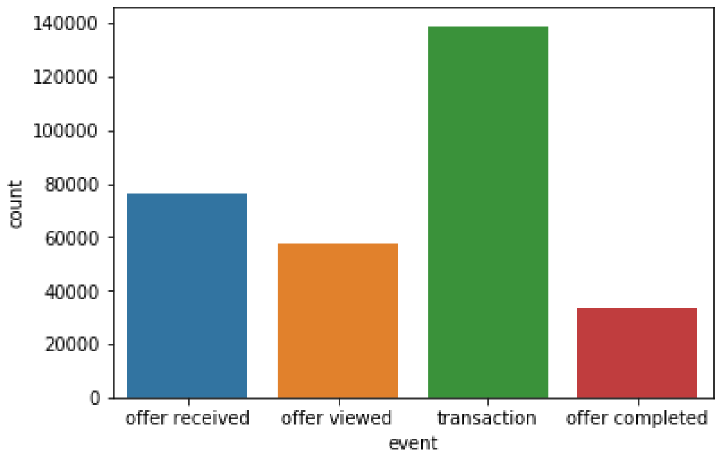


There is an increase every year in the number of people creating a membership from 2013 till 2017. There is a drop in the number of customers joining the app in 2018. In 2013, only 274 customers created a membership and in 2017 this number increased to 5599 customers. There is almost a 30% decrease in the number of customers who have created a membership from 2017 to 2018.

Offer types



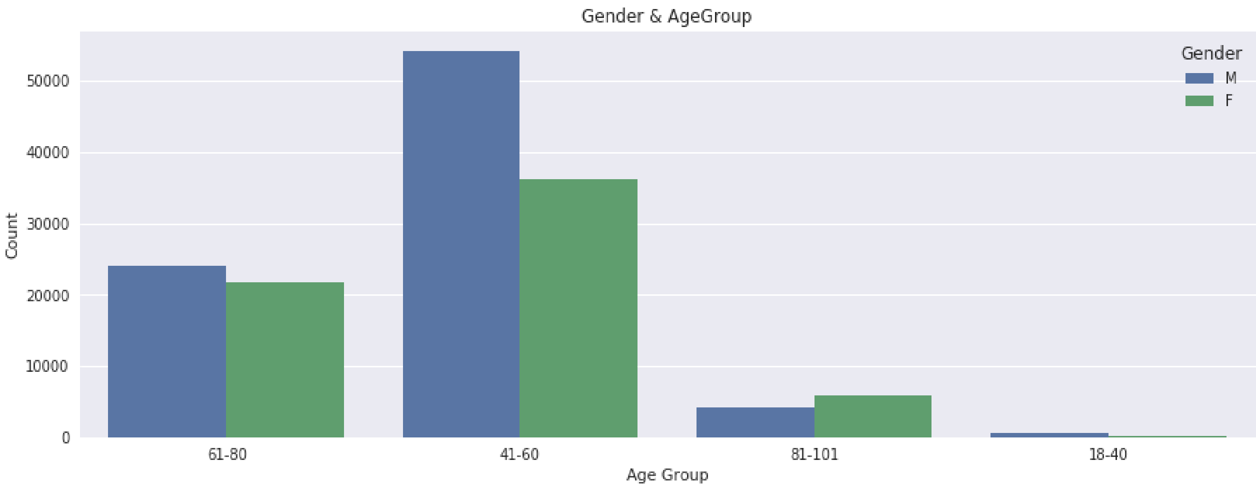
Events



138953 transaction events done by customers. There are a total of 76277 offers received by customers, 57725 offers are viewed and 33579 are completed. Almost half of the offers received are completed.

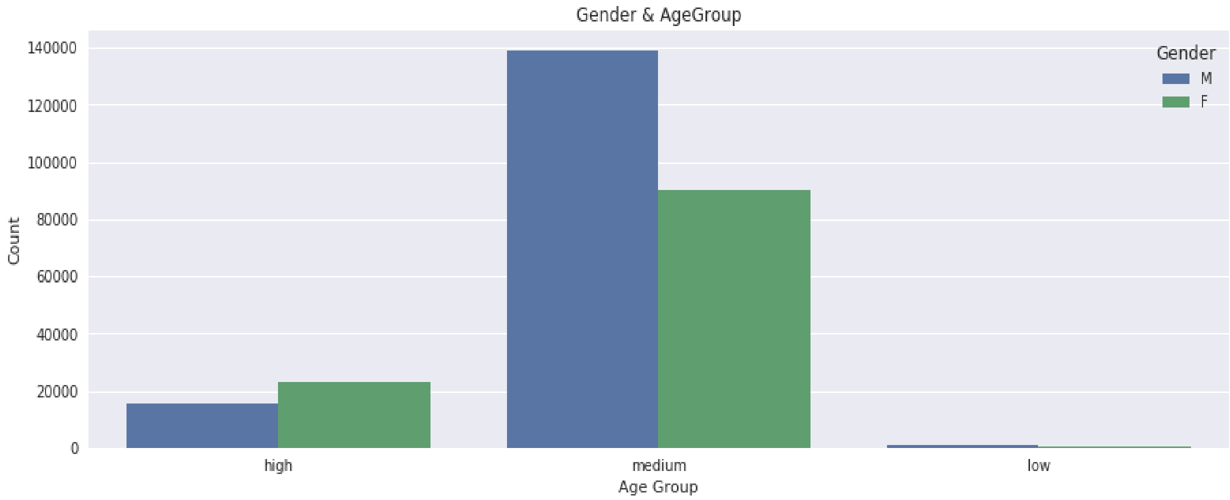
Now it's time to bring some different demographics together. To do that will merge the three cleaned datasets (Portfolio, Profile, Transcript) together into a single dataframe for better visualization. This is important for further analysis. The merged dataset

Distribution of Gender in each AgeGroup



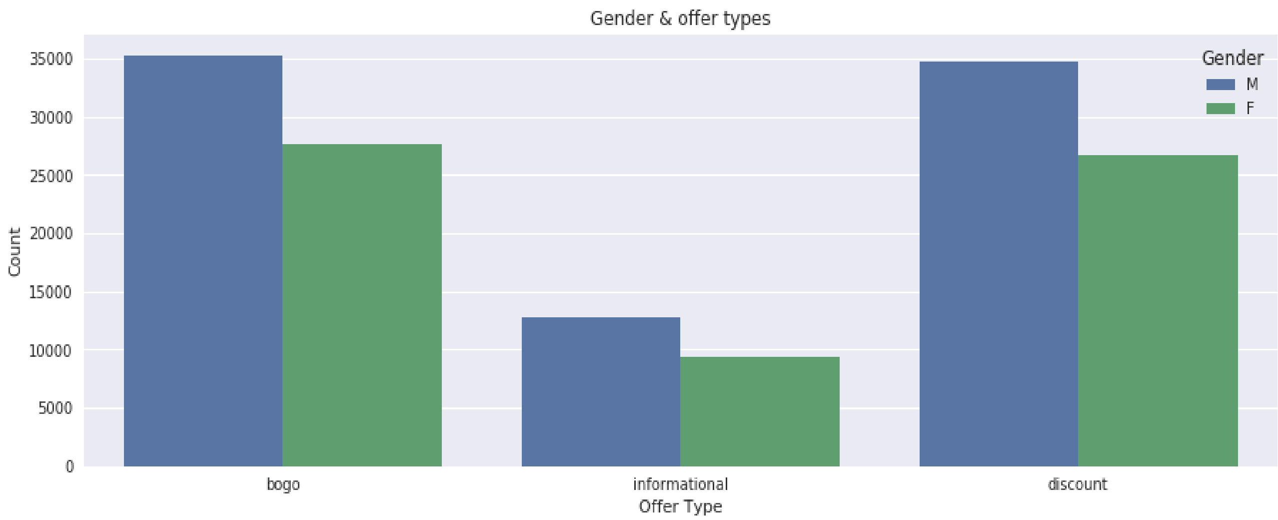
Since we have more males than females in the dataset it makes sense to find more males in each age group. That is true in the following age groups (18-40, 41-60 and 61-80), however there are more female customers between 81 and 101 years old.

Distribution of Gender in each IncomeGroup



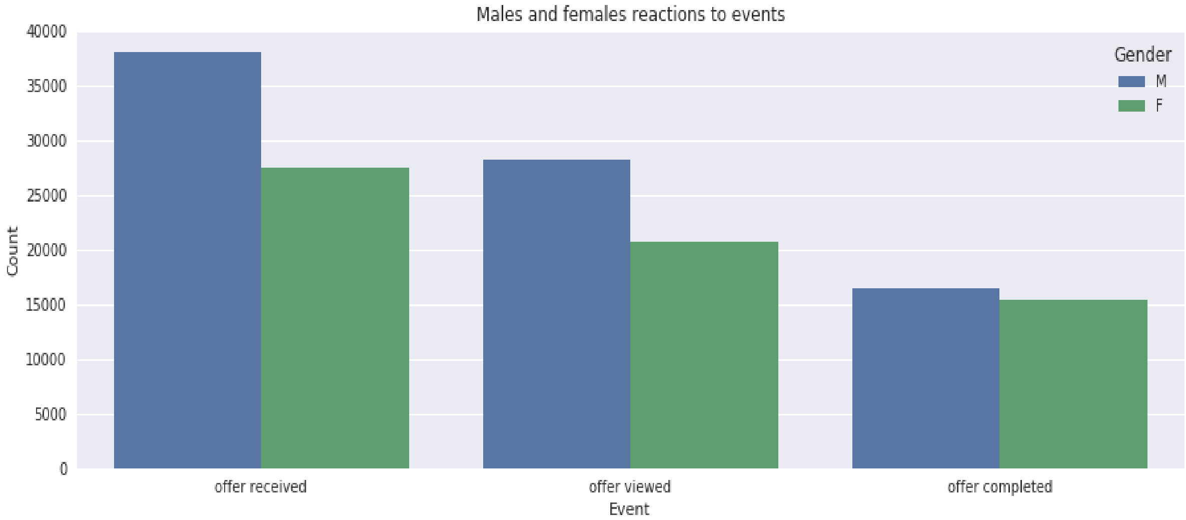
There are males than females that receive medium and low incomes, whilst there are more female customers located in the high income category. Hence can conclude that more than half of the male and females customers receive medium income.

Distribution of Gender in each Offer type



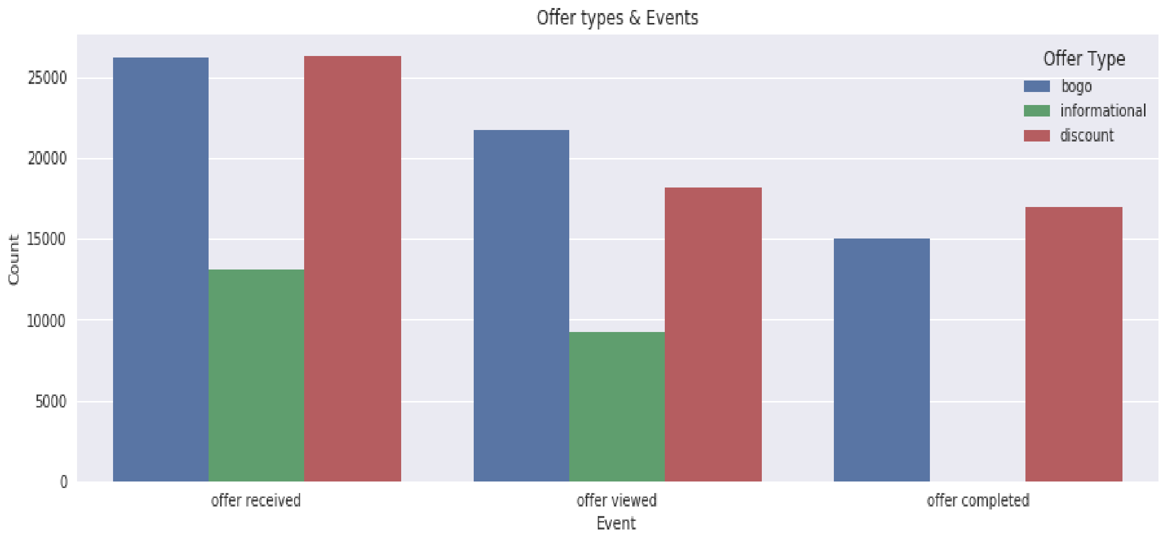
Again since we have more male customers, more of those customers are expected to choose more offer types. This is reflected in the above graph, more males have chosen all three offer types. Almost all bogo and discount offer types are equally chosen by females and males, with customers slightly choosing bogo over discount.

Distribution of Gender in each Event



The results of the above graph is similar to the one before, more males have received and viewed offers due to the difference in number of customers. More males have completed the offers, but the difference between males and females is not that big. If more females join the app, perhaps there would be an equal number of offer completion in both genders.

Distribution of Offer types in each event



Almost the same number of bogo and discount offer types have been received, with more bogo offer types viewed. However, more customers have completed discount offer types.

Distribution of income in offer types

The reason why this graph was constructed is to see whether or not there is a relation between the income of a customer and their preferred offer type. Those who receive below average income are not very active customers. There are almost equal preferences for the rest of the income groups when choosing between bogo and discount offer types. With customers choosing bogo slightly over discount offer types.

Data Modelling

After exploring, processing and visualizing the datasets, it is now time to create some machine learning models and make some predictions. This project is trying to create a model to predict whether a customer will complete an offer received. Viewing an offer means the customer will receive and view the offer. Completing an offer, on the other hand, means the customer will pass through the three stages (receiving, viewing and completing). For that, will include those transcripts with event '**offer completed'** only in training and testing sets.

To have the machine learning models perform better and give highest accuracies, more cleaning is needed to be done on the merged dataset. Below are the features in our master dataset.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 272762 entries, 0 to 272761
Data columns (total 19 columns):
difficulty      272762 non-null int64
duration        272762 non-null int64
offer_id        272762 non-null object
offer_type      272762 non-null object
reward_x        272762 non-null int64
web             272762 non-null int64
email           272762 non-null int64
mobile          272762 non-null int64
social          272762 non-null int64
event           272762 non-null object
customer_id     272762 non-null object
time            272762 non-null int64
reward_y        272762 non-null float64
amount          272762 non-null float64
gender          272762 non-null object
month_member    272762 non-null int64
year_member     272762 non-null int64
AgeGroup        272762 non-null object
IncomeGroup     272762 non-null object
dtypes: float64(2), int64(10), object(7)
memory usage: 51.6+ MB
```

Before creating any models and making predictions, there are some preprocessing steps that need to be done to certain features in the dataset. These steps include scaling, standardizing and transforming, which are important numeric feature engineering steps to skew and rescale features for modelling.

In order to create training and testing datasets, needed to do the following:

- Scale numerical input variables ('difficulty', 'duration', 'time', ' reward_x', 'reward_y' and 'amount').

- Fitting and transforming the above listed numerical variables.
- Convert Categorical features (‘gender’, ‘AgeGroup’, ‘IncomeGroup’, ‘offer_type’, ‘event’) into indicator variables.

Now that we have the merged dataset scaled and normalized, we can now split it into training and testing sets. 70% of the dataset will be allocated to the training set and 30% will be allocated to the testing set.

The machine learning algorithms used in this project are:

- Random Forest classifier (Default parameters)
- KNeighbours classifier (Benchmark model)
- Naive bayes (Default parameters)
- Decision Tree (Default parameters)
- Logistic regression (Default parameters)

Algorithms and Techniques

Classification is considerably a crucial aspect of supervised learning. In this section, I will discuss the various machine learning algorithms used in the problem. Additionally, I will discuss how these algorithms work and strengths and weaknesses of each.

Decision Tree

Works in a way where it orders classes in a precise way. The algorithm separates data points into further data points where they become categories within categories, allowing hierarchical steps to occur to help reach certain decisions. The technique is simple and easy to visualise. This algorithm can work on numerical as well as categorical data.

If there is a large dataset, generated decision trees can be complex to produce and this will lead to time consuming to train the model. This will also result in far more complex calculations compared to other algorithms.

Random forest is an expansion to decision trees. When decision trees are constructed and training time is taking long, forest classifiers are introduced to avoid over-fitting to the training dataset.

```
# create the model
dtree = tree.DecisionTreeClassifier()
# train the model
dtree.fit(X_train,y_train)
# predictions on model
y_train_pred = dtree.predict(X_train)
y_test_pred = dtree.predict(X_test)
# Training set performance
dtree_train_accuracy = accuracy_score(y_train,y_train_pred)
dtree_train_f1 = f1_score(y_train,y_train_pred,average='weighted')
dtree_train_cm = confusion_matrix(y_train,y_train_pred)
dtree_train_cr = classification_report(y_train,y_train_pred)
# Testing set performance
dtree_test_accuracy = accuracy_score(y_test,y_test_pred)
dtree_test_f1 = f1_score(y_test,y_test_pred,average='weighted')
dtree_test_cm = confusion_matrix(y_test,y_test_pred)
dtree_test_cr = classification_report(y_test,y_test_pred)
```

```
# perform random classifier model
rforest = RandomForestClassifier()
# train the model
rforest.fit(X_train,y_train)
# predictions on model
y_train_pred = rforest.predict(X_train)
y_test_pred = rforest.predict(X_test)
# Training set performance
rforest_train_accuracy = accuracy_score(y_train,y_train_pred)
rforest_train_f1 = f1_score(y_train,y_train_pred,average='weighted')
rforest_train_cm = confusion_matrix(y_train,y_train_pred)
rforest_train_cr = classification_report(y_train,y_train_pred)
# Testing set performance
rforest_test_accuracy = accuracy_score(y_test,y_test_pred)
rforest_test_f1 = f1_score(y_test,y_test_pred,average='weighted')
rforest_test_cm = confusion_matrix(y_test,y_test_pred)
rforest_test_cr = classification_report(y_test,y_test_pred)
```

Logistic regression

Logistic regression is a basic yet important classification algorithm. The technique is based on using single or multiple independent variables to determine an outcome. The technique is useful for determining and understanding the influence of independent variables on outcome variables. When creating this algorithm, I realised that it has low variance and is very efficient.

This technique can also be bad when handling large datasets with various categorical features. This technique assumes that the dataset is free of missing values [1].

```
# perform logistic regression
logreg = LogisticRegression()
# train the model
logreg.fit(X_train,y_train)
# predictions on model
y_train_pred = logreg.predict(X_train)
y_test_pred = logreg.predict(X_test)
# Training set performance
logreg_train_accuracy = accuracy_score(y_train,y_train_pred)
logreg_train_f1 = f1_score(y_train,y_train_pred,average='weighted')
logreg_train_cm = confusion_matrix(y_train,y_train_pred)
logreg_train_cr = classification_report(y_train,y_train_pred)
# Testing set performance
logreg_test_accuracy = accuracy_score(y_test,y_test_pred)
logreg_test_f1 = f1_score(y_test,y_test_pred,average='weighted')
logreg_test_cm = confusion_matrix(y_test,y_test_pred)
logreg_test_cr = classification_report(y_test,y_test_pred)
```

Naive Bayes

Naive Bayes algorithm is based on determining whether or not a data point belongs to a certain category. It is extracted from Bayes theorem which is based on the independence of each feature [2]. The classifier is very powerful and fast and solves multi class predictions Another good advantage is that this algorithm performs better than other algorithms on small training data.

Since the algorithm is based on independence, this assumption is difficult for anyone to find.

```
# creating a model
nbayes = GaussianNB()
# train the model
nbayes.fit(X_train,y_train)
# predictions on model
y_train_pred = nbayes.predict(X_train)
y_test_pred = nbayes.predict(X_test)
# Training set performance
nbayes_train_accuracy = accuracy_score(y_train,y_train_pred)
nbayes_train_f1 = f1_score(y_train,y_train_pred,average='weighted')
nbayes_train_cm = confusion_matrix(y_train,y_train_pred)
nbayes_train_cr = classification_report(y_train,y_train_pred)
# Testing set performance
nbayes_test_accuracy = accuracy_score(y_test,y_test_pred)
nbayes_test_f1 = f1_score(y_test,y_test_pred,average='weighted')
nbayes_test_cm = confusion_matrix(y_test,y_test_pred)
nbayes_test_cr = classification_report(y_test,y_test_pred)
```

K Nearest Neighbour

K-nearest neighbour is based on a simple principle and that is classifying data. This algorithm stores all instances corresponding to training data in n-dimensional space. This algorithm can be used to solve a lot of different problems so it's flexible. The algorithm is effective if training data is large and robust to noise training data.

Since this algorithm is based on determining a value for K, computation cost can be large and this will slow the model down.

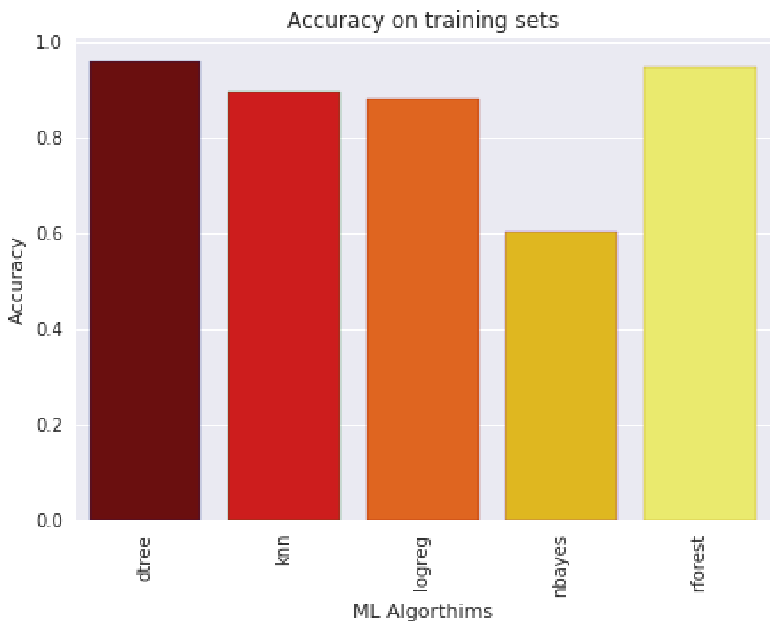
The K Nearest Neighbour will be used as our Benchmark model and will be compared to other models to see how far has our model worked.

```
# define the kneighbour classifier
knn = KNeighborsClassifier()
# train the model
knn.fit(X_train,y_train)
# predictions on model
y_train_pred = knn.predict(X_train)
y_test_pred = knn.predict(X_test)
# Training set performance
knn_train_accuracy = accuracy_score(y_train,y_train_pred)
knn_train_f1 = f1_score(y_train,y_train_pred,average='weighted')
knn_train_cm = confusion_matrix(y_train,y_train_pred)
knn_train_cr = classification_report(y_train,y_train_pred)
# Testing set performance
knn_test_accuracy = accuracy_score(y_test,y_test_pred)
knn_test_f1 = f1_score(y_test,y_test_pred,average='weighted')
knn_test_cm = confusion_matrix(y_test,y_test_pred)
knn_test_cr = classification_report(y_test,y_test_pred)
```

Justification

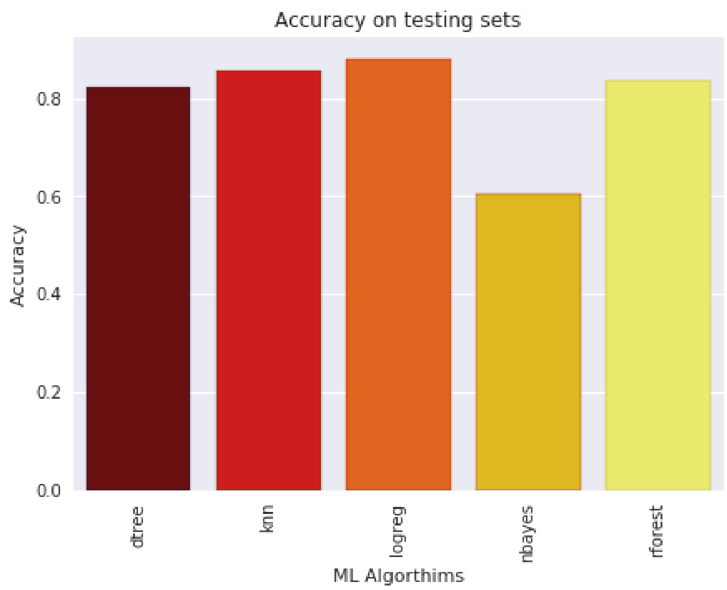
Evaluation results on training set

	Accuracy	F1Score
dtree	0.957943	0.954536
knn	0.894261	0.874757
logreg	0.881356	0.825775
nbayes	0.600912	0.667704
rforest	0.948364	0.946206



Evaluation results on testing set

	Accuracy1	F1Score1
dtree	0.821457	0.817295
knn	0.855748	0.829124
logreg	0.880348	0.824329
nbayes	0.603160	0.669292
rforest	0.836171	0.823515



From the above tables, we can see that on the training set, decision tree and random forest classifiers have performed better than our benchmark model with 95% and 94% accuracy. Whereas, the benchmark model performed better than the other model. On the testing set, our benchmark model performed as the second best model. Logistic regression was the best model with an accuracy of 88%.

Conclusion

There are more males than females. Most of the customers are between 41 and 60 years old. Most of the customers receive medium income. Customers tend to choose the bogo offer type slightly over discount offer. Since there are more males, they have received, viewed and completed more offers than females. Customers have received the same number of bogo and discount offer types, have viewed more bogo offer types but completed more discount offer types.

References

[1]https://www.analyticsvidhya.com/blog/2021/05/5-classification-algorithms-you-should-know-introductory-guide/

[2]https://data-flair.training/blogs/machine-learning-classification-algorithms/