Starbucks Capstone Project

Project Overview

A Starbucks is one of the most well-known companies in the world. It strives to give his customers always the best service and the best experience, They have a mobile application in which the users can make orders online. The project aims at optimizing the customers experience using the app through user's behavior analysis.

Data Sets

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

Problem Statement

The goal that is to be achieved here is to best determine which kind of offer to send to each user based on their response to the previously sent offers. There are three different kinds of offers.

- Buy One Get One Free (BOGO)
- Discount
- Informational

Our goal is to analyze the historical data about the app usage and develop the algorithm that associates with the response of s customer to an offer.

Metrics

A model metric is needed to assess the quality of the approach and determine which model gives the best results**.** I have considered the F1 score as the model metric to assess the quality of the approach and determine which model gives the best results.It can be interpreted as the weighted average of precision and recall.

Data Exploration

There are three datasets available. We should explore the data sets individually to get a good idea of what features will be needed for the final input dataset and get ideas for any feature engineering.

Portfolio

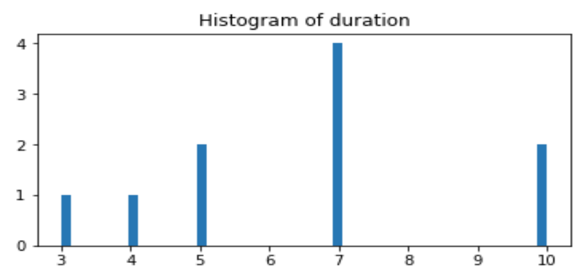
- id (string) - offer id
- offer_type (string) - type of offer ie BOGO, discount, informational
- difficulty (int) - minimum required spend to complete an offer
- reward (int) - reward given for completing an offer
- duration (int) - time for offer to be open, in days
- channels (list of strings)

```
In [3]: portfolio
```

Out[3]:

	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

- The channel type column describes the social media through the offer is sent to the customer
- The difficulty column describes the minimum required spend to complete an offer
- The duration column describes the time for offer to be open, in days
- The id column refers to the offer_id
- There are 3 types of offers available
  1. BOGO
  2. Discount
  3. Informational



Profile

- 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

In [9]:
```
profile.head()
```

Out[9]:

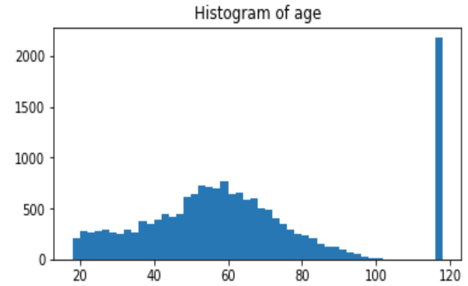
	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43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

In [10]:
```
print('(orig) rows,cols:',profile.shape)
```
```
(orig) rows,cols: (17000, 5)
```

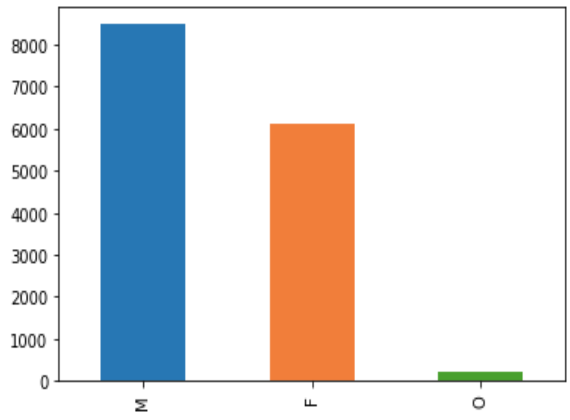
In [13]:
```
profile.isnull().sum()
```

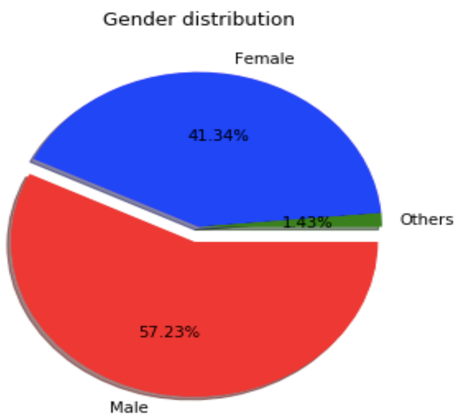
Out[13]:
```
age                    0
became_member_on      0
gender              2175
id                     0
income              2175
dtype: int64
```

There are 2175 values in profile whose age, gender and income values are missing. Those values are encoded with the age values as 118.



The people of age 40-60 use the app more. The people of age above 80 doesn’t use the app more.





The above figures show that the usage of app is more by males compared to females and others. Out of 100 percent of people using the app males are about 57.23 percent, females are about 41.34 percent and the rest is contributed by the others.

Transcript

- 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

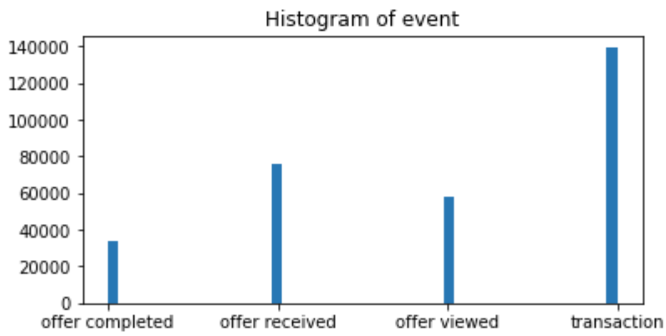
In [18]:
```
transcript.head(10)
```
Out[18]:

	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'}
5	offer received	389bc3fa690240e798340f5a15918d5c	0	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}
6	offer received	c4863c7985cf408faee930f111475da3	0	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}
7	offer received	2eeac8d8feae4a8cad5a6af0499a211d	0	{'offer id': '3f207df678b143eea3cee63160fa8bed'}
8	offer received	aa4862eba776480b8bb9c68455b8c2e1	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
9	offer received	31dda685af34476cad5bc968bdb01c53	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}

In [20]:
```
transcript.describe(include="all")
```
Out[20]:

	event	person	time	value
count	306534	306534	306534.000000	306534
unique	4	17000	NaN	5121
top	transaction	94de646f7b6041228ca7dec82adb97d2	NaN	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}
freq	138953	51	NaN	14983
mean	NaN	NaN	366.382940	NaN
std	NaN	NaN	200.326314	NaN
min	NaN	NaN	0.000000	NaN
25%	NaN	NaN	186.000000	NaN
50%	NaN	NaN	408.000000	NaN
75%	NaN	NaN	528.000000	NaN
max	NaN	NaN	714.000000	NaN

In [19]:
```
print('(orig) rows,cols:',transcript.shape)
```
```
(orig) rows,cols: (306534, 4)
```



Data Preparation and Cleaning

The three individual data sets that we analysed need to be combined into one to be used for the exploratory data analysis & model building. Before that can be done though, there's a lot of data wrangling we need to do. First, we will prepare the data for EDA, and later some more pre-processing for fitting it into the model.

Cleaning Portfolio

- Rename the column difficulty to offer_difficulty.
- Rename the column id to offer_id, duration to offer_duaration.
- Rename the column reward to offer_reward.
- Encode the channels column.
- Encode the offer_types column.

In [24]:
```
portfolio.head()
```

Out[24]:

	offer_difficulty	offer_duration	offer_id	offer_type	offer_reward	email	mobile	social	web	bogo	discount	informational
0	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10	1	1	1	0	1	0	0
1	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10	1	1	1	1	1	0	0
2	0	4	3f207df678b143eea3cee63160fa8bed	informational	0	1	1	0	1	0	0	1
3	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5	1	1	0	1	1	0	0
4	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5	1	0	0	1	0	1	0

Cleaning Profile

- Rename the columns id to customer_id, income to customer_income.
- Encode the age column into buckets by decade (10-20, 20-30, 30-40 and so on).
- Encode the gender column.
- Store the year the customer became a member on.

In [29]:
```
profile.head()
```

Out[29]:

der	customer_id	customer_income	membership_year	age[10-20]	age[20-30]	age[30-40]	age[40-50]	age[50-60]	age[60-70]	age[70-80]	age[80-90]	age[90-100]	ag
M	68be06ca386d4c31939f3a4f0e3dd783	65404.991568	2017	0	0	0	0	1	0	0	0	0	
F	0610b486422d4921ae7d2bf64640c50b	112000.000000	2017	0	0	0	0	1	0	0	0	0	
M	38fe809add3b4fcf9315a9694bb96ff5	65404.991568	2018	0	0	0	0	1	0	0	0	0	
F	78afa995795e4d85b5d9ceeca43f5fef	100000.000000	2017	0	0	0	0	0	0	1	0	0	
M	a03223e636434f42ac4c3df47e8bac43	65404.991568	2017	0	0	0	0	1	0	0	0	0	

Cleaning Transcript

- Rename person to customer_id.
- Encode events.
- Get ‘offer id’ from value column dictionary and place in new column clean_id.
- Get ‘amount’ from value column dictionary and place in new column money spent.

In [33]:
```
transcript.head()
```

Out[33]:

	event	customer_id	time	offer_viewed	offer_received	offer_completed	money_gained	money_spent	offer
0	3	78afa995795e4d85b5d9ceeca43f5fef	0	0	1	0	0.0	0.0	9b98b8c7a33c4b65b9aebfe6a799e6
1	3	a03223e636434f42ac4c3df47e8bac43	0	0	1	0	0.0	0.0	0b1e1539f2cc45b7b9fa7c272da2e1
2	3	e2127556f4f64592b11af22de27a7932	0	0	1	0	0.0	0.0	2906b810c7d4411798c6938adc9daa
3	3	8ec6ce2a7e7949b1bf142def7d0e0586	0	0	1	0	0.0	0.0	fafdcd668e3743c1bb461111dcafc2
4	3	68617ca6246f4fbc85e91a2a49552598	0	0	1	0	0.0	0.0	4d5c57ea9a6940dd891ad53e9dbe8c

Merging of data Frames

Now, it’s time to merge all cleaned data frames so that all the features are contained within one data frame and then apply Exploratory Data

Out[39]:

	event	customer_id	time	offer_viewed	offer_received	offer_completed	money_gained	money_spent	offer
0	3	78afa995795e4d85b5d9ceeca43f5fef	0	0	1	0	0.0	0.0	9b98b8c7a33c4b65b9aebfe6a799e6
1	3	a03223e636434f42ac4c3df47e8bac43	0	0	1	0	0.0	0.0	0b1e1539f2cc45b7b9fa7c272da2e1
2	3	e2127556f4f64592b11af22de27a7932	0	0	1	0	0.0	0.0	2906b810c7d4411798c6938adc9daa
3	3	8ec6ce2a7e7949b1bf142def7d0e0586	0	0	1	0	0.0	0.0	fafdcd668e3743c1bb461111dcafc2
4	3	68617ca6246f4fbc85e91a2a49552598	0	0	1	0	0.0	0.0	4d5c57ea9a6940dd891ad53e9dbe8c

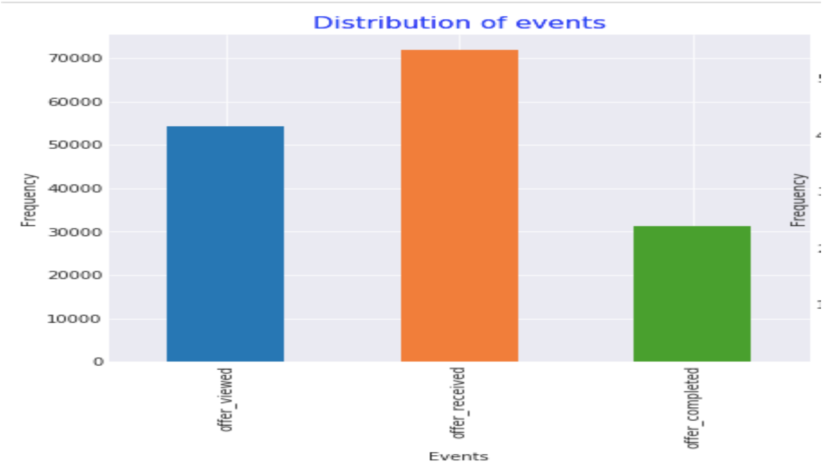
5 rows × 37 columns

Exploratory data analysis

To maximize insights into our cleaned data frame and find its interesting characteristic features and representations, we will cover key points for the combined population to the individual personalized level.

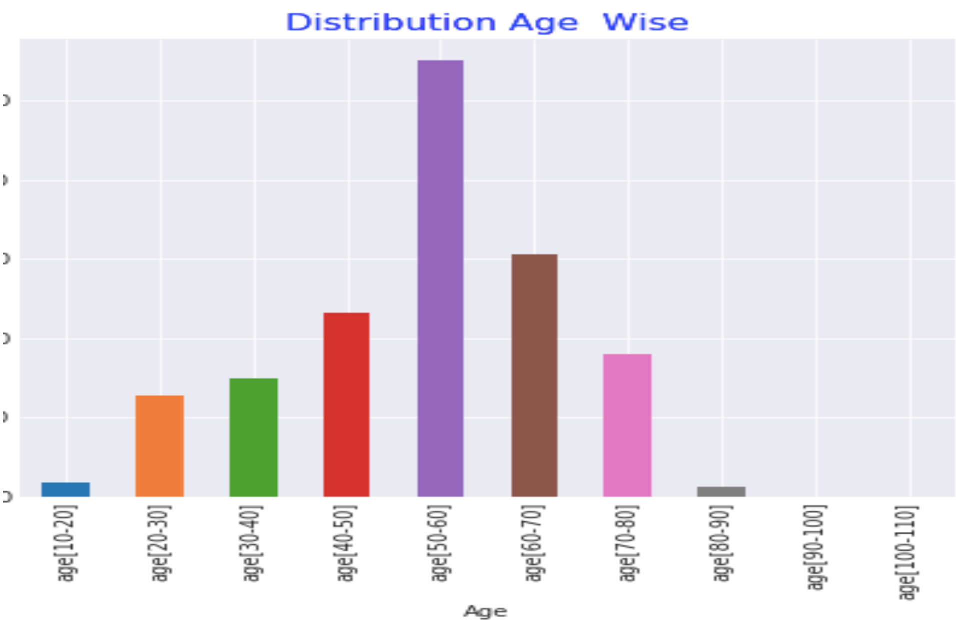
Distribution of Events

- The events are distributed as offer_viewed, offer_completed, offer_received.
- There are a smaller number of people who complete the offers compared to the people who view and ignore it.



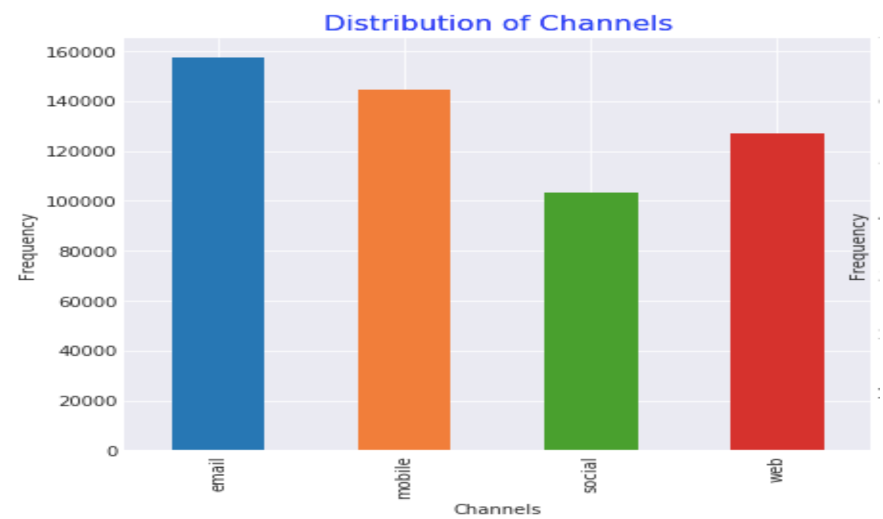
Distribution of Age

- The people of age between 50-60 are most likely to use the app
- The people of age above 90 are less likely to use the app



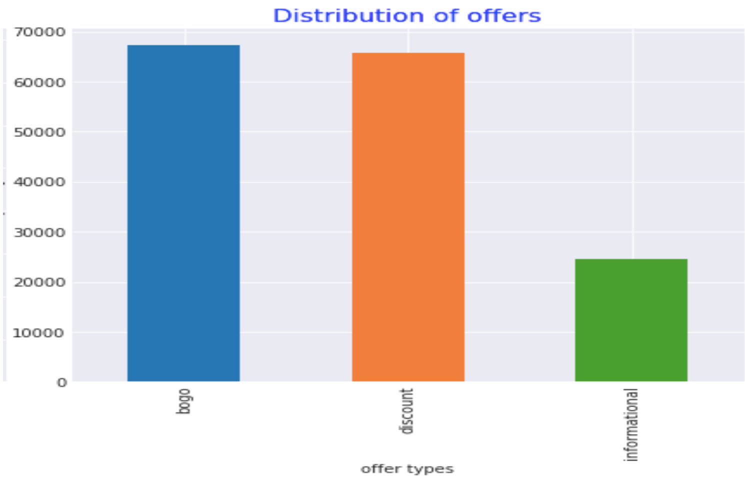
Distribution of Channels

- Every customer will receive the offer through the email
- The least used channel is social

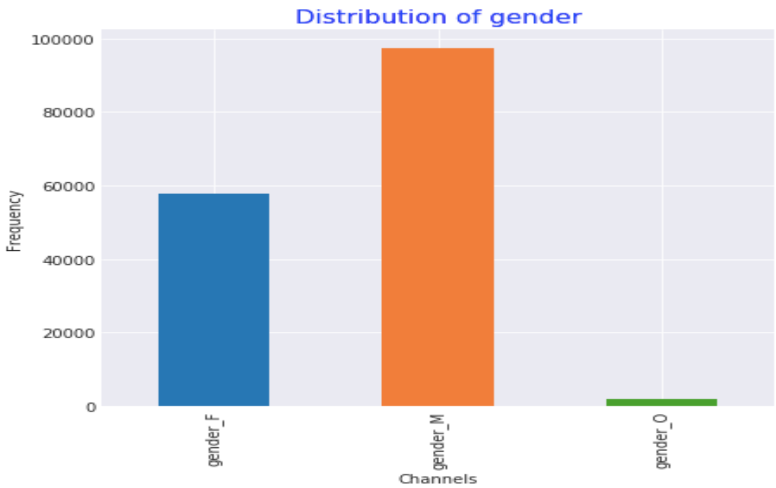


Distribution of offers

- The most commonly used offer is bogo.
- The least used offer is informational.



Distribution of gender



Build Machine Learning models

Revisiting our second objective, we are creating two different classification models to predict the effectiveness of an offer i.e., to predict response of a customer to an offer.

Data Preparation and Cleaning

Before building a model, we'll have to clean & prepare the data to fit into the model. To do this we will perform some tasks as:

- Encode categorical data such as gender, offer type, channel and age groups.
- Encode the 'event' data to numerical values.
- Scale and normalize numerical data.

Split train and test data

Final data is ready after tasks 1–5. We now have to split the data (both features and their labels) into training and test sets, taking 60% of data for training and 40% for testing.

Benchmark model

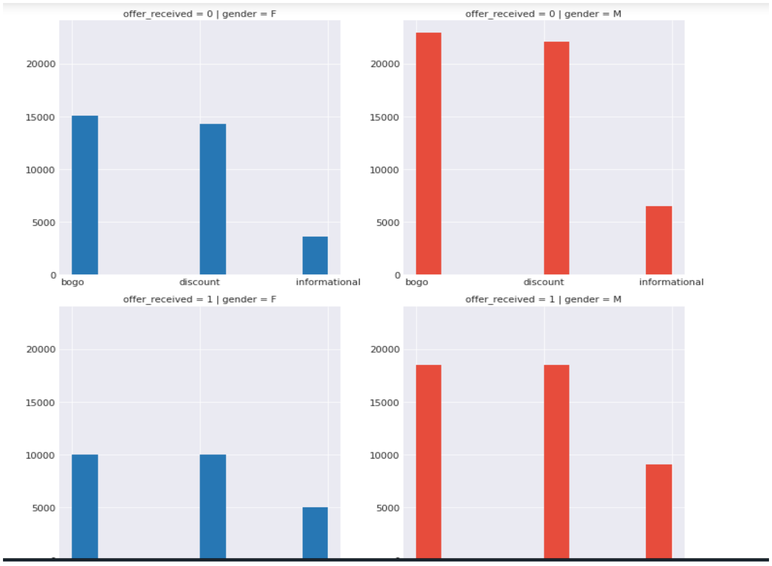
A quick and fairly accurate model can be considered as a benchmark. We'll use the KNeighborsClassifier to build the benchmark, as it is a fast and standard method for binary classification machine learning problems and evaluate the model result using the F1 score as the evaluation metric.

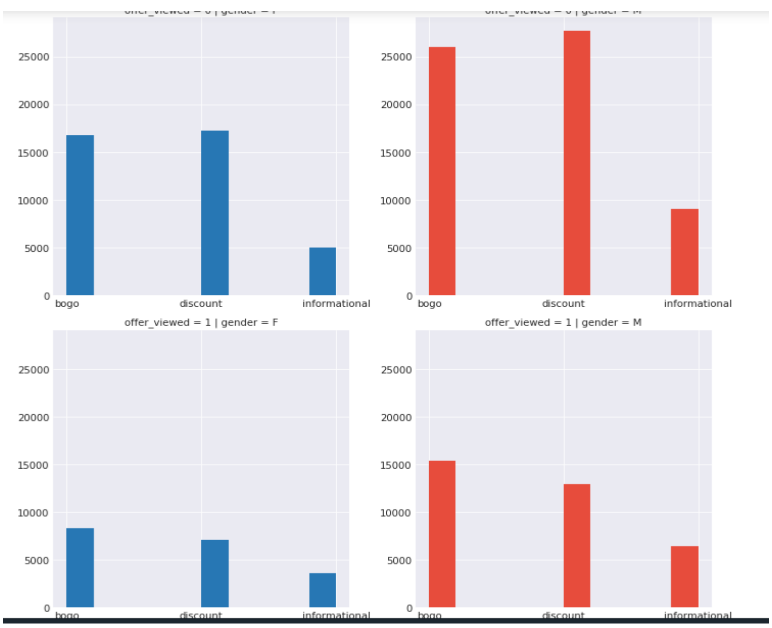
```
In [59]: benchmark
Out[59]:
```

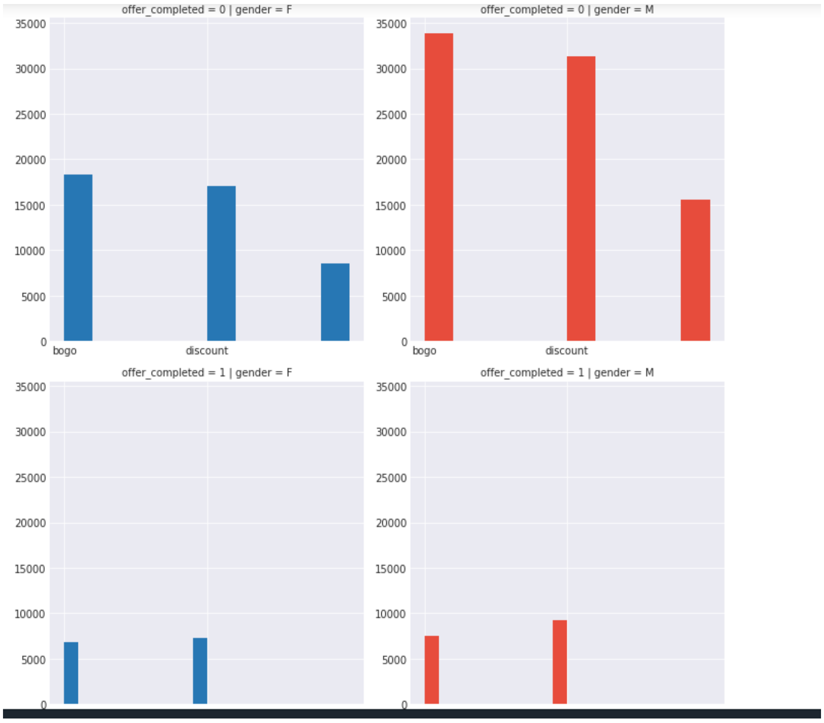
	Benchmark Model	train F1 score	test F1 score
0	KNeighborsClassifier	100.0	100.0

Conclusions

The males comprise 57.23% of the data and use the Starbucks app more than the females. Specifically, both males & females in the age group 50–60 use the app the most. Discount offers are more preferred by the customers. Also, there is less number of customers who actually complete the offer as compared to the ones who just view & ignore it. Males generally ignore offers more & offers are nearly equally completed by males & females. The ratio of males to females in each offer type is nearly the same, with male customers being more. We can look more at the figures & information in the Exploratory Data Analysis section more to best determine which kind of offers to send to the customers.







Model Evaluation

The problem that we chose to solve was to build a model that predicts whether a customer will respond to an offer. The strategy we followed has four steps. First, we combined offer portfolio, customer profile, and transaction data. Second, we did some more pre-processing to the combined data to fit into the model. Third, we assessed the F1 score of a benchmark KNeighborsClassifier model. Fourth, we compared the performance of RandomForestClassifier and DecisionTreeClassifier models to determine which model best represents our data on hand.

Out[66]:

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	100.0	100.0
1	RandomForestClassifier	100.0	100.0
2	DecisionTreeClassifier	100.0	100.0