Starbucks Rewards Optimisation

June 13, 2021

1 Optimizing Starbucks rewards offers program

About Author Lionel has a BEng (Hons) in electronic engineering (N.U.S.T), Data Analyst, Data Scientist and Machine learning nanodegrees (Udacity). He is passionate about building data driven solutions for businesses and is always looking for opportunties to colloborate. He is currently open for work. You can find some of his work on github.com/msongi or email senatorlionel2@gmail.com

1.1 Introduction

Starbucks is one of the largest coffehouses in the world. They have been known for having done of the most vigorous digitalisaion strategies that has seen them grow to become titans in industry. In the first quarter of 2020, it witnessed a 16pc in year over year growth recording 18.9 million active users. It is infamous for its rewards program. According to starbucks, as of end of 2020, nearly a quarter of their transactions are done through the phone. This means that, a substantial amount of its revenue relies on the rewards app. In this project, we seek to analyse customer behavior of the starbucks rewards app so we can optimise profits by targetted marketing.

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. 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.

Not all users receive the same offer.

The objective is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type.

Every offer has a validity period before the offer expires. As an example, a BOGO offer might be valid for only 5 days. Informational offers have a validity period even though these ads are merely providing information about a product; for example, if an informational offer has 7 days of validity, you can assume the customer is feeling the influence of the offer for 7 days after receiving the advertisement.

Starbuks has provided transactional data showing user purchases made on the app including the timestamp of purchase and the amount of money spent on a purchase. This transactional data also has a record for each offer that a user receives as well as a record for when a user actually views the offer. There are also records for when a user completes an offer.

Keep in mind as well that someone using the app might make a purchase through the app without having received an offer or seen an offer.

1.1.1 Example

To give an example, a user could receive a discount offer buy 10 dollars get 2 off on Monday. The offer is valid for 10 days from receipt. If the customer accumulates at least 10 dollars in purchases during the validity period, the customer completes the offer.

However, there are a few things to watch out for in this data set. Customers do not opt into the offers that they receive; in other words, a user can receive an offer, never actually view the offer, and still complete the offer. For example, a user might receive the "buy 10 dollars get 2 dollars off offer", but the user never opens the offer during the 10 day validity period. The customer spends 15 dollars during those ten days. There will be an offer completion record in the data set; however, the customer was not influenced by the offer because the customer never viewed the offer.

1.1.2 Cleaning

This makes data cleaning especially important and tricky.

You'll also want to take into account that some demographic groups will make purchases even if they don't receive an offer. From a business perspective, if a customer is going to make a 10 dollar purchase without an offer anyway, you wouldn't want to send a buy 10 dollars get 2 dollars off offer. You'll want to try to assess what a certain demographic group will buy when not receiving any offers.

The "transaction" event does not have any "offer_id" associated to it, so we have to figure out which transactions are connected to particular offer and which ones are not (customer bought something casually).

Informational offer can not be "completed" due to it's nature, so we need to find a way to connect it with the possible transactions.

Some demographic groups will make purchases regardless of whether they receive an offer. Ideally we would like to group them separately.

A user can complete the offer without actually seeing it. In this case user was making a regular purchase and offer completed automatically. This is not a result of particular marketing campaign but rather a coincidence.

1.1.3 Implementation Steps:

- Data exploration
- Data cleaning
- Exploratory data analysis (EDA)
- Customer segmentation
- Exploring the resultant clusters
- Data Preprocessing
- Training the model
- Improving the model
- Choosing the best performing model
- Deploying the best model
- Exploring the prediction results

1.2 Data Exploration

1. DATA ASSESSMENT 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)

[14]:
```
portfolio.head()
```

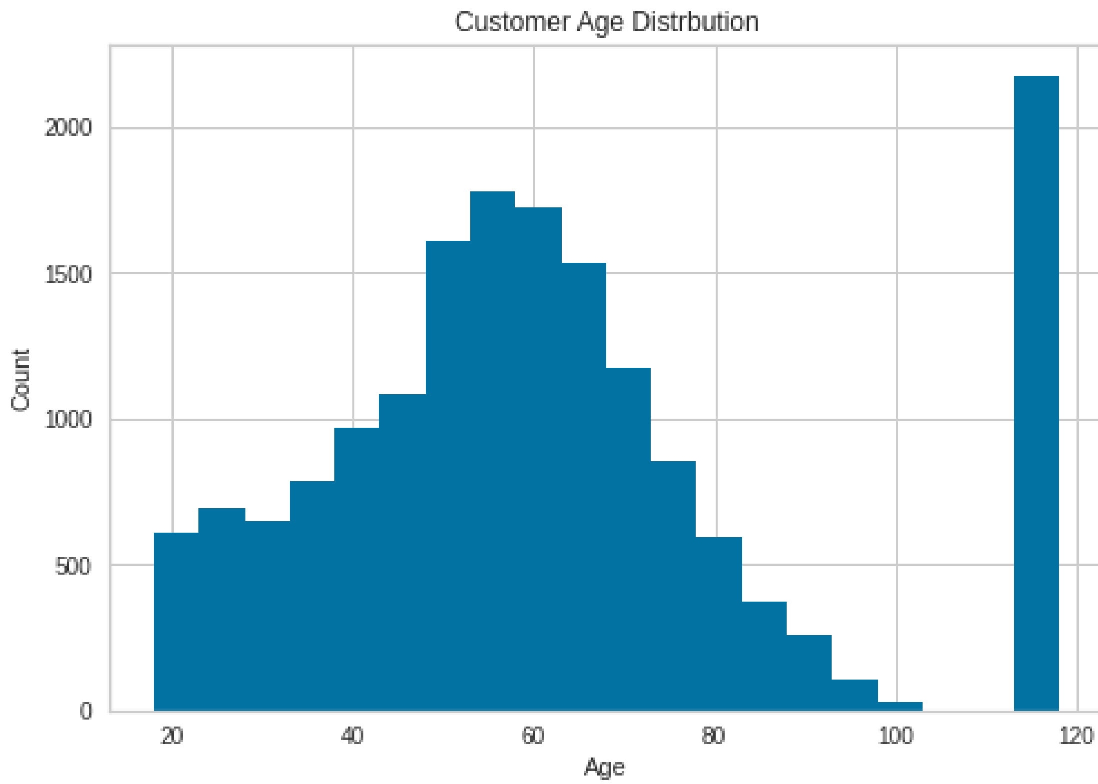
[14]:
```
   possible_reward                      channels  spent_required  \
0               10       [email, mobile, social]              10
1               10  [web, email, mobile, social]              10
2                0          [web, email, mobile]               0
3                5          [web, email, mobile]               5
4                5                  [web, email]              20

   offer_duration_days     offer_type                          offer_id
0                    7           bogo  ae264e3637204a6fb9bb56bc8210ddfd
1                    5           bogo  4d5c57ea9a6940dd891ad53e9dbe8da0
2                    4  informational  3f207df678b143eea3cee63160fa8bed
3                    7           bogo  9b98b8c7a33c4b65b9aebfe6a799e6d9
4                   10       discount  0b1e1539f2cc45b7b9fa7c272da2e1d7
```

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

[15]:
```
profile.head()
```

[15]:
```
  gender  age                                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
```

[20]:
```
display_customer_profile()
```



Age is normally distributed with a few outliers who inputted their age as 118. This is likely user input error, since this data was collected in 2018, we assume that 1900 was the default age in the app and these users put the default year of birth. Since we know that this age is unlikely, we replace this age with nan. display_customer_profile()

[16]:
```
profile.isnull().sum()[profile.isnull().sum()>0]
```

[16]:
```
gender    2175
income    2175
dtype: int64
```

There are 2175 customers with missing gender and income information, without these, we cannot factor them into the analysis because their demographics affect how they make transactions and perform with offers.

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 - offer id: (string/hash) not associated with any “transaction” - amount: (numeric) money spent in “transaction” - reward: (numeric) money gained from “offer completed”

[17]:
```
transcript.head()
```

[17]:

```
                                person           event  \
0  78afa995795e4d85b5d9ceeca43f5fef  offer received
1  a03223e636434f42ac4c3df47e8bac43  offer received
2  e2127556f4f64592b11af22de27a7932  offer received
3  8ec6ce2a7e7949b1bf142def7d0e0586  offer received
4  68617ca6246f4fbc85e91a2a49552598  offer received

                                               value  time
0  {'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}     0
1  {'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}     0
2  {'offer id': '2906b810c7d4411798c6938adc9daaa5'}     0
3  {'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}     0
4  {'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}     0
```

Data quality issues: * Null values for "gender" and "income" in profile.json * Age: missing value encoded as 118 in profile.json * Incorrect data format (int64 instead of datetime) in profile.json * Column names can be improved * Incompatible units - offer duration days vs. event time in hours * Different keys in transcript.json - "event id" and "event_id" * Channels is a list of strings that specify the channels used to send the offer to the customer, this needs to be in seperate encoded columns * Dictionary value in value columns has information about amount, transaction and offer id, inorder to make this information useful we need to extract it from the value column * Time needs to be in the correct format

2. CLEANING THE DATA The challenges with this dataset have been discussed above and can be summarised as: * Customers do not opt in on offers * Customers can complete offers without viewing them and can complete offers without receiving them * Informational offers can only be viewed, we need to find a way to connect viewing and completion to measure its success * Transactions can be done casually i.e with no connection to offers so we need to seperate transactions

Data cleaning implementation process:

- Convert date to datetime format
- Only keep customers whose profiles we have, drop the rest
- Add more convenient ids for 10 types of offer
- Create age, membership, income ranges to categorise customers in terms of their ages, incomes and membership types e.g adult, regular member, middle income
- Convert offer duration to days
- Decide what to do with missing values in individual tables
- Extract the features from the value column, make them into columns and drop the value column
- Encode "channels" using a one-hot encoding scheme
- Encode "event" using a one-hot encoding scheme
- Remove the extra "offer id" column (and transfer values to a correct column)
- Connect "transaction" with "offer completed"
- Connect "offer viewed" with "offer completed"
- Connect transaction with each type of offer and drop those transactions so that remaining columns are casual purchases not linked to any offers

- Make new event category for offers completed by accident: "auto completed" and connect it with offer viewed such that we know that all who just viewed the offers, just viewed them.
- Improve column naming

1.3 Exploratory Analysis

1.3.1 Create a new dataframe for offer earnings analysis

Build a dataframe with aggregated transaction, offer, and demographics data for customer behavior analysis. As we saw earlier, the same customer can receive the same offer multiple times, they can also receive other offers of the same type or of different types. We will build a new dataframe by: 1. Getting the offer type data per customer 2. Getting offer id data per customer 3. Building a dataframe by merging extracted variables, demographics, offers, and transaction data

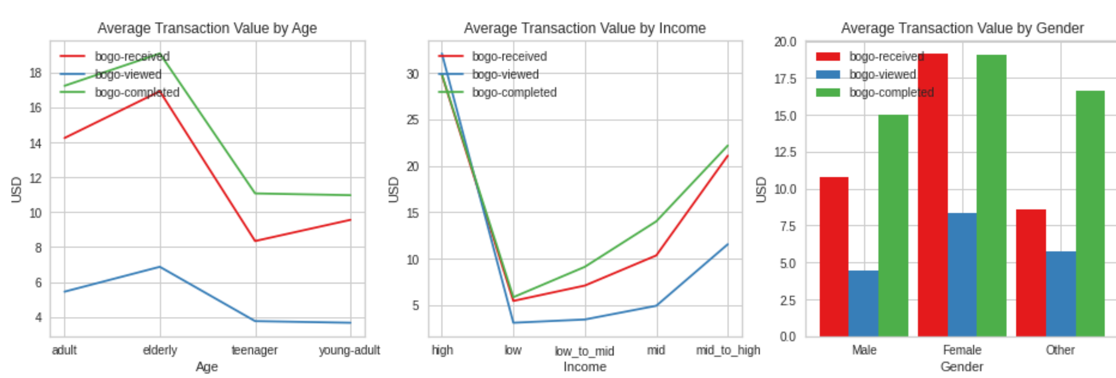
[140]:
```
merged_cust.head()
```

[140]:
```
                                  total_expense  total_transactions  received  \
0009655768c64bdeb2e877511632db8f         127.60                 8.0       5.0
0011e0d4e6b944f998e987f904e8c1e5          79.46                 5.0       5.0
0020c2b971eb4e9188eac86d93036a77         196.86                 8.0       5.0
0020ccbbb6d84e358d3414a3ff76cffd         154.05                12.0       4.0
003d66b6608740288d6cc97a6903f4f0          48.34                18.0       5.0

                                  money_gained  viewed  completed  \
0009655768c64bdeb2e877511632db8f           9.0     4.0        3.0
0011e0d4e6b944f998e987f904e8c1e5          13.0     5.0        3.0
0020c2b971eb4e9188eac86d93036a77          14.0     3.0        3.0
0020ccbbb6d84e358d3414a3ff76cffd          13.0     4.0        3.0
003d66b6608740288d6cc97a6903f4f0           9.0     4.0        3.0

                                  bogo_received  bogo_money_gained  \
0009655768c64bdeb2e877511632db8f            1.0                5.0
0011e0d4e6b944f998e987f904e8c1e5            1.0                5.0
0020c2b971eb4e9188eac86d93036a77            2.0               10.0
0020ccbbb6d84e358d3414a3ff76cffd            2.0               10.0
003d66b6608740288d6cc97a6903f4f0            0.0                0.0

                                  bogo_viewed  bogo_completed  …  \
0009655768c64bdeb2e877511632db8f          1.0             1.0  …
0011e0d4e6b944f998e987f904e8c1e5          1.0             1.0  …
0020c2b971eb4e9188eac86d93036a77          1.0             1.0  …
0020ccbbb6d84e358d3414a3ff76cffd          2.0             2.0  …
003d66b6608740288d6cc97a6903f4f0          0.0             0.0  …

                                  info_1_viewed  info_2_received  \
0009655768c64bdeb2e877511632db8f            1.0              1.0
0011e0d4e6b944f998e987f904e8c1e5            1.0              1.0
0020c2b971eb4e9188eac86d93036a77            0.0              1.0
```

```
0020ccbbb6d84e358d3414a3ff76cffd          0.0              1.0
003d66b6608740288d6cc97a6903f4f0          1.0              1.0

                                  info_2_viewed  gender  year  quarter  \
0009655768c64bdeb2e877511632db8f            1.0       M  2017        2
0011e0d4e6b944f998e987f904e8c1e5            1.0       O  2018        1
0020c2b971eb4e9188eac86d93036a77            1.0       F  2016        1
0020ccbbb6d84e358d3414a3ff76cffd            1.0       F  2016        4
003d66b6608740288d6cc97a6903f4f0            1.0       F  2017        2

                                    age_group  income_range  member_type  \
0009655768c64bdeb2e877511632db8f  young-adult   mid_to_high          new
0011e0d4e6b944f998e987f904e8c1e5        adult           mid          new
0020c2b971eb4e9188eac86d93036a77        adult   mid_to_high      regular
0020ccbbb6d84e358d3414a3ff76cffd  young-adult           mid      regular
003d66b6608740288d6cc97a6903f4f0  young-adult   mid_to_high          new

                                  net_expense
0009655768c64bdeb2e877511632db8f       118.60
0011e0d4e6b944f998e987f904e8c1e5        66.46
0020c2b971eb4e9188eac86d93036a77       182.86
0020ccbbb6d84e358d3414a3ff76cffd       141.05
003d66b6608740288d6cc97a6903f4f0        39.34

[5 rows x 59 columns]
```

The result is a merged customer dataset that shows the total expenses,net expense, total offers viewed, received,completed by each customer and which offer types and offer ids

[126]:
```
offer_earnings_plot(merged_cust,'bogo')
```

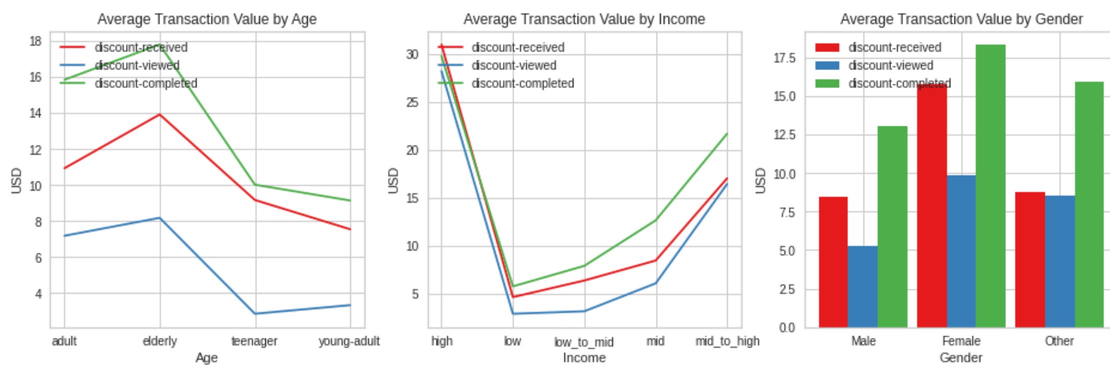


The pattern in terms of offers, received, viewed and completed is the same accross age, income and gender demographics.

- The elderly have the highest average transaction for BOGO offers. In general, as age increases, customers spend more on BOGO.

- High income earners spend the highest on BOGO followed by mid-to-high earners. In general we observe an increase in spending with an increase in income.
- Females spend the most on BOGO but in terms of view rate, those that did not specfy gender spent more on BOGO if we compare how many received vs how many completed BOGO.

[130]:
```
offer_earnings_plot(merged_cust,'discount')
```



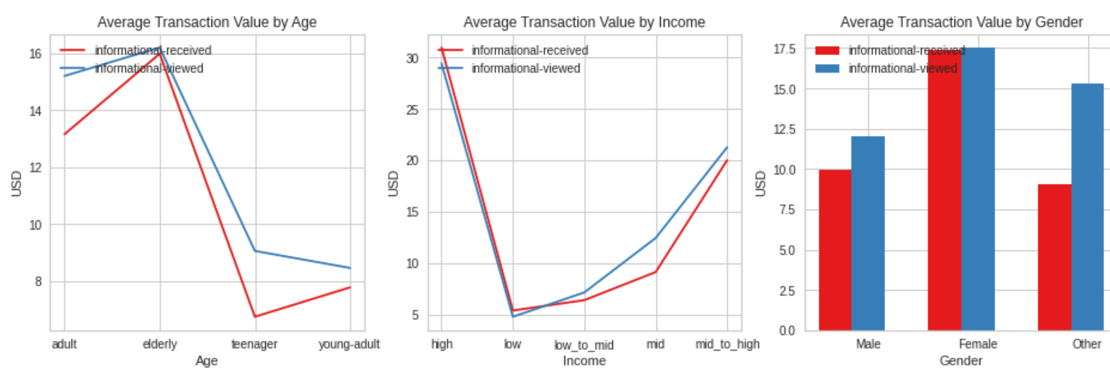
The same trends we observed for BOGO seem to apply here as well.

Transactions increase with an increase in age with those above 60 spending the most on discounts.

High income earners will spend an average of 30USD on discount offers. Mid to high earners will also spent a lot on discount

Females are also the highest spenders on discount earners and again we see those with unspecified gender have a higher conversion rate of all demographics

[131]:
```
offer_earnings_plot(merged_cust,'informational')
```



[132]:
```
offers = greatest_earnings(merged_cust, n_top=5)
```

[133]:
```python
offers
```

[133]:
```
([discount_1', 'bogo_1', 'bogo_4', 'bogo_3', 'bogo_2'],
 {'discount_1': 145.425,
  'bogo_1': 145.325,
  'bogo_4': 145.23000000000002,
  'bogo_3': 131.14,
  'bogo_2': 131.03,
  'discount_2': 124.99000000000001,
  'discount_3': nan,
  'discount_4': 138.07999999999998,
  'info_1': 124.17000000000002,
  'info_2': 102.96000000000001})
```

bogo 1 and 2 and discount 1 had the highest earnings

1.3.2 Create another merge for more analysis

Create a new dataframe that will combine transaction data with offer type such that the data about offers received, viewed and completed for each offer type are in one row.

Connect "transaction" with "informational" offer We want to connect informational offer with transaction done. Such that all the offer completed rows can show the offer type that was completed. Then all the remaining transactions will be for uncompleted offers. To find such transactions, we need to make sure that:

1. Offer type is "informational"
2. Customer saw the offer - "offer_viewed" is true
3. The same customer made a purchase within **offer validity period**

[127]:
```python
print("Number of informational offers that were viewed but not completed:")
len(master_df[(master_df.offer_type == "Informational") & (master_df.event ==␣
 ↪"offer viewed")])
```

```
Number of informational offers that were viewed but not completed:
```

[127]:
```
2778
```

Connect "transaction" with "bogo" offer We want to connect bogo offer with transaction done. Such that all the offer completed rows can show the offer type that was completed. Then all the remaining transactions will be for uncompleted offers. To find such transactions, we need to make sure that:

1. Offer type is "informational"
2. Customer received saw the offer - "offer_viewed" is true
3. The same customer made a purchase within **offer validity period**

[123]:
```python
print("Number of BOGO offers that were viewed but not completed:")
len(master_df[(master_df.offer_type == "BOGO") & (master_df.event == "offer␣
 ↪viewed")])
```

```
Number of BOGO offers that were viewed but not completed:
```

[123]: 3958

Connect "transaction" with "discount" offer We want to connect bogo offer with transaction done. Such that all the offer completed rows can show the offer type that was completed. Then all the remaining transactions will be for uncompleted offers. To find such transactions, we need to make sure that:

1. Offer type is "discount"
2. Customer received and saw the offer - "offer_viewed" is true
3. The same customer made a purchase within **offer validity period**

[125]:
```
print("Number of Discount offers that were viewed but not completed:")
len(master_df[(master_df.offer_type == "Discount") & (master_df.event == "offer␣
↪viewed")])
```

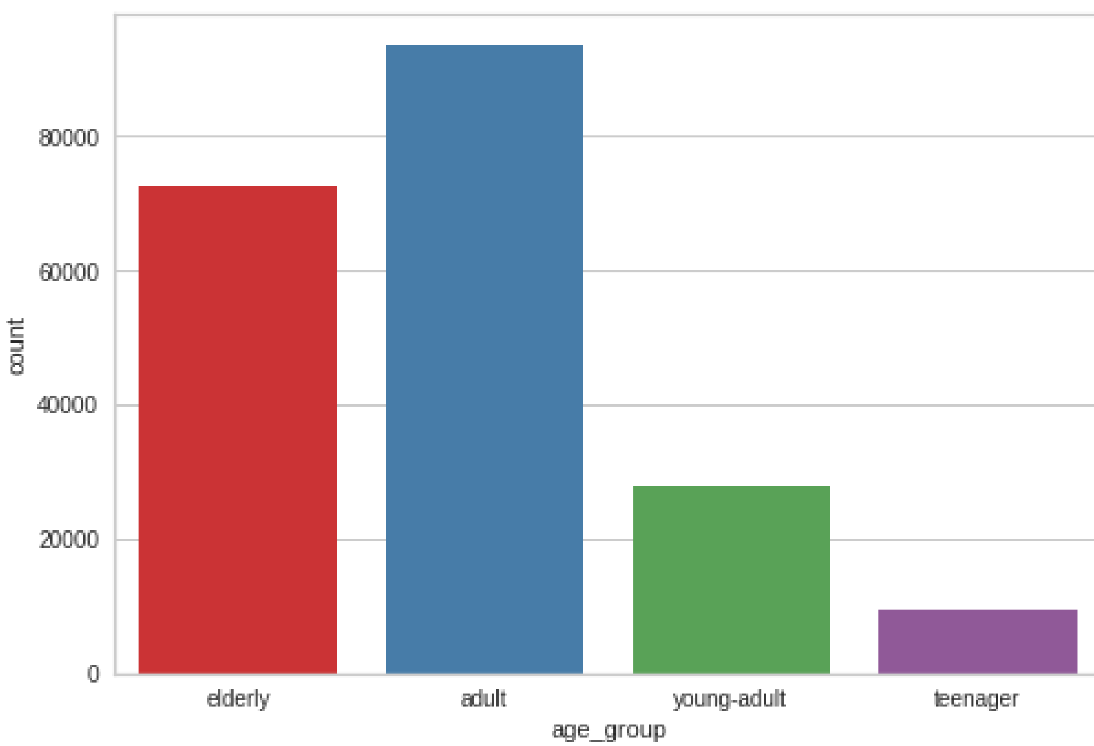
```
Number of Discount offers that were viewed but not completed:
```

[125]: 5460

1.3.3 Distribution of Age Groups among customers

[126]:
```
sns.countplot(master_df['age_group'])
```

[126]: <AxesSubplot:xlabel='age_group', ylabel='count'>



[127]:
```
master_df['age_group'].value_counts()
```

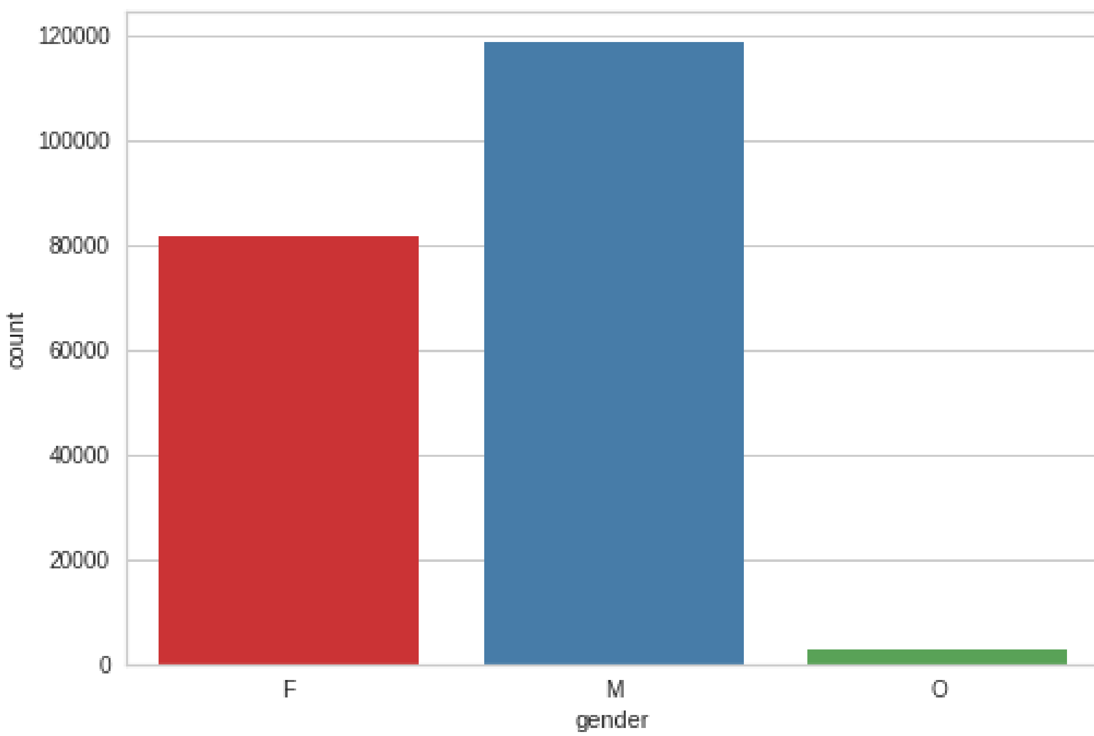
[127]:
```
adult           93596
elderly         72478
young-adult     27734
teenager         9355
Name: age_group, dtype: int64
```

The distribution of ages among starbucks customers is skewed towards the older group. The elderly are the largest population followed by the adults. Teenagers are the smallest group

1.3.4 Distribution of Gender among customers

[128]:
```
sns.countplot(master_df['gender'])
```

[128]: <AxesSubplot:xlabel='gender', ylabel='count'>



[129]:
```
master_df['gender'].value_counts()
```

[129]:
```
M    118718
F     81552
```

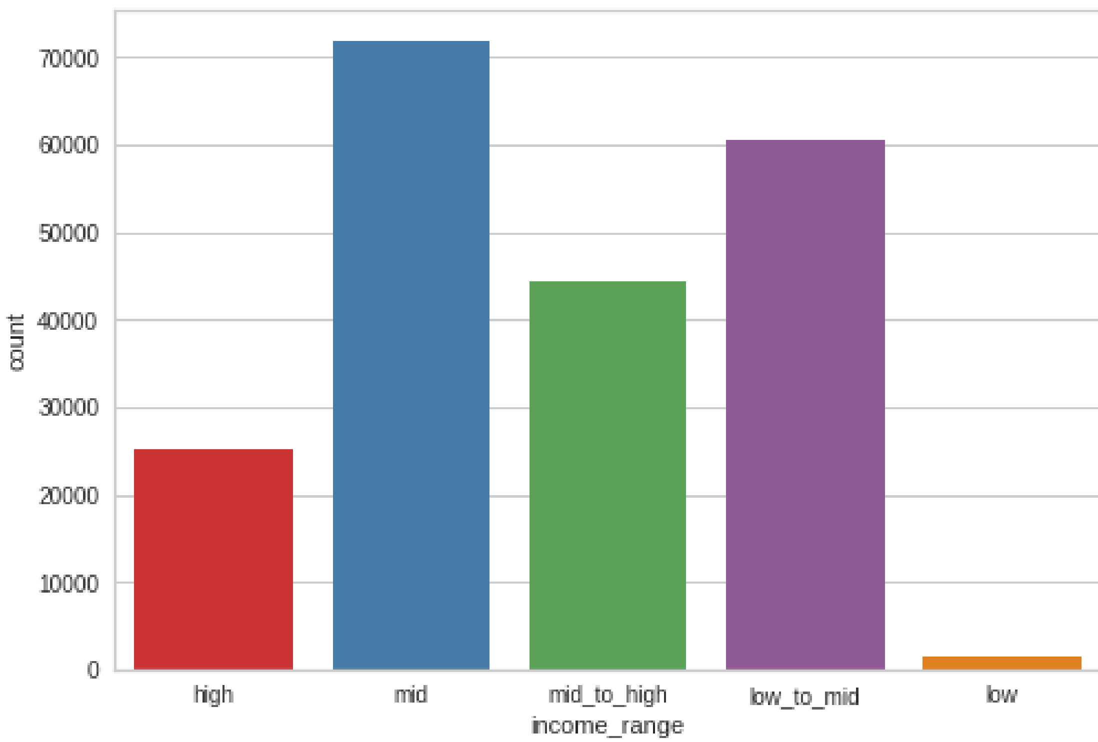
```
O       2893
Name: gender, dtype: int64
```

58.6 % of customers are male, 39.8% are female and 1.4% prefer not to specify gender

1.3.5 Distribution of income among customers

[130]:
```
sns.countplot(master_df['income_range'])
```

[130]:
```
<AxesSubplot:xlabel='income_range', ylabel='count'>
```



Middle income and low_to_mid income earners are occupy a huge proportion of the population, with mid income earners being the dorminant. Low earners are fewer, they are the least

Distribution of income by gender among customers

[132]:
```
master_df.groupby(['income_range','gender']).customer_id.count()
```

[132]:
```
income_range  gender
high          F         14980
              M          9945
              O           230
low           F           327
              M           953
              O            62
```

```
low_to_mid    F         17953
              M         41775
              O           726
mid           F         26124
              M         44576
              O          1170
mid_to_high   F         22168
              M         21469
              O           705
Name: customer_id, dtype: int64
```

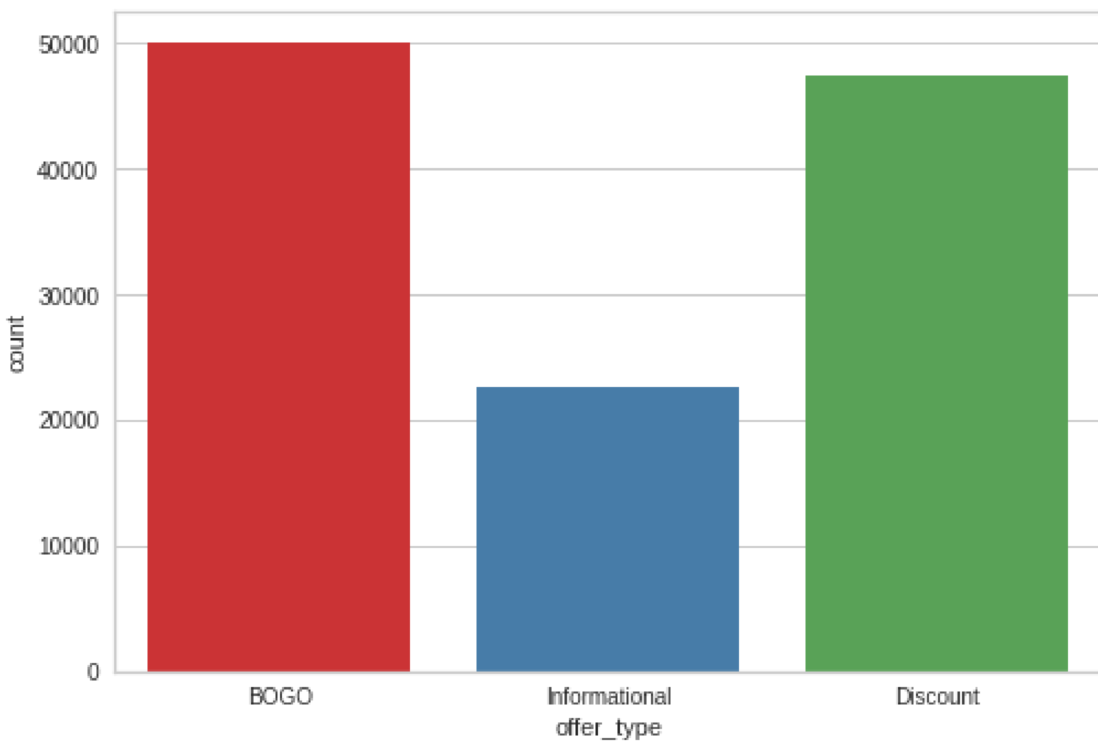
Females are the highest earners. Most men are low to mid and mid earners. There's an almost equal distribution of male and females among the mid to high bracket

1.3.6 Distribution of Offer Type and Offer ID during experiment

[133]:

```
```

[133]: <AxesSubplot:xlabel='offer_type', ylabel='count'>



[134]:

```
master_df['offer_type'].value_counts()
```

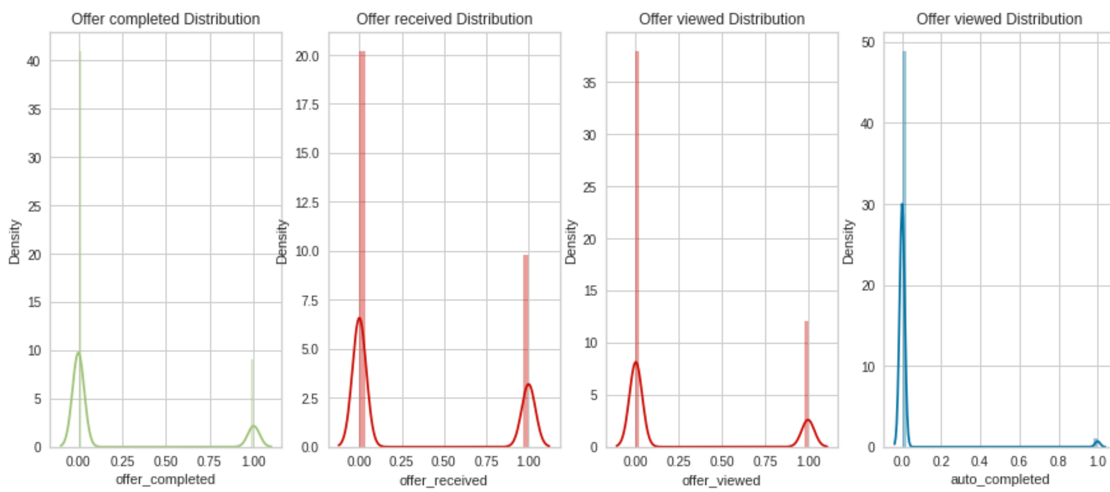
[134]:
```
BOGO             50077
Discount         47491
Informational    22660
Name: offer_type, dtype: int64
```

There are 3 types of offers presented to customers, BOGO, discount and informational. BOGO and discount were sent out more.BOGO was the most distributed offer, 25% of distributed offers were BOGO, 23% were Discount and 11.2% were Informational offers. There were 10 different offers from the 3 different offer types, there were 4 types of BOGO offers, 4 types of discount offers and 2 types of informational offers.

1.3.7 Distribution of offer received viewed and completed

[135]:

[135]:
```
Text(0.5, 1.0, 'Offer viewed Distribution')
```



The distribution of offer data follows the same pattern. There were a lot of offers not completed vs those that were completed. The same with offers received, viewed and auto completed

1.3.8 Distribution of Transactions for the offers

[136]:

[136]:
```
Text(0.5, 1.0, 'Offer type distribution - Transaction V.S. No Transaction')
```

The distribution of transactions done vs transactions not done for BOGO,Discount and Informational offers follows the same pattern. There was more transactions not done as compared to transactions completed

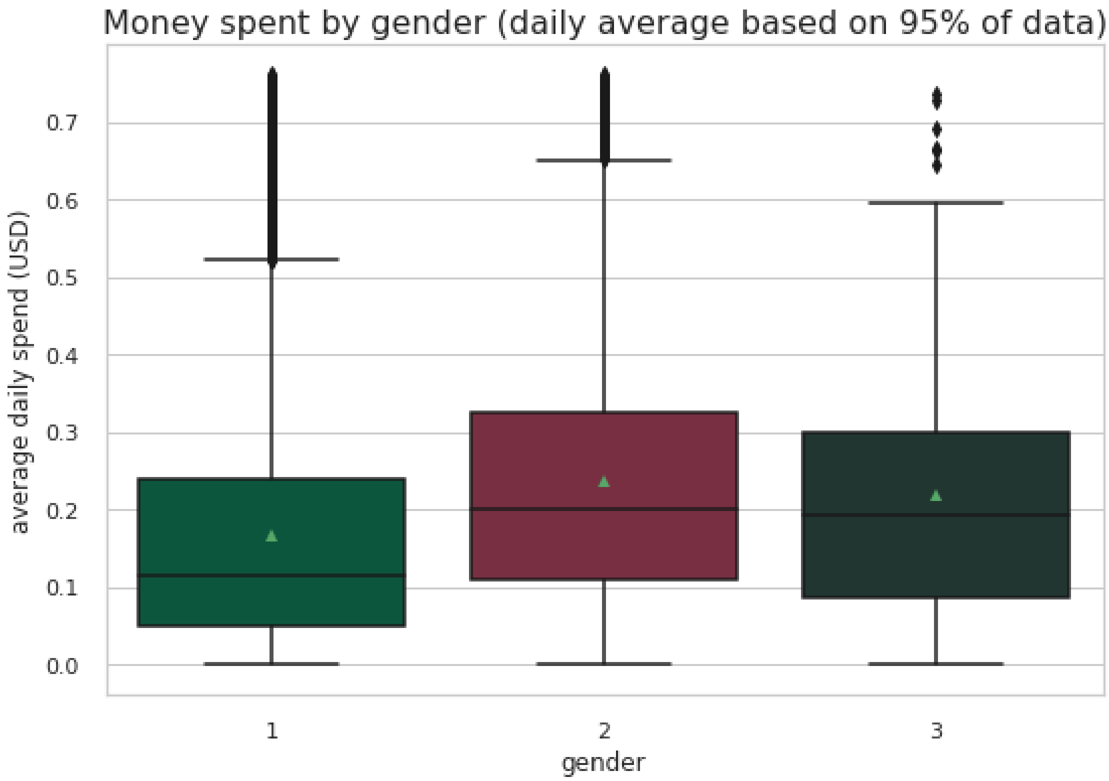
Create a column for daily avg spending Calculate how much a user spends or has spent on average since they became a member. Use the days of membership and became member on that we dropped before. We do this by creating a new df called data that groups how much money was spent by a customer by customer_id, gender, age group, income range and days of membership

[348]:
```
# group data by person, calculate summary spend:
data = master_df.groupby(['customer_id', 'gender', 'age_group', 'income_range',␣
↪'days_of_membership','member_type'], as_index=False)['money_spent'].sum()
```

[349]:
```
# Create new column "daily_spend_avg"
data['daily_spend_avg'] = data.money_spent / data.days_of_membership
data.sample(3)
```

[349]:
```
                            customer_id  gender  age_group  income_range  \
11965  ce9ff37fcd4a4f6180ee3df63d6ce6c0       1          4             3
12349  d53ea893dd774977ad75280d4be4c621       1          3             2
223    0409df7f3af74e8c813d975fbd4ceb02       1          4             4

       days_of_membership member_type  money_spent  daily_spend_avg
11965                 763     regular       106.77         0.139934
12349                 445         new        10.04         0.022562
223                  1108     regular        64.20         0.057942
```

[355]:
```
```



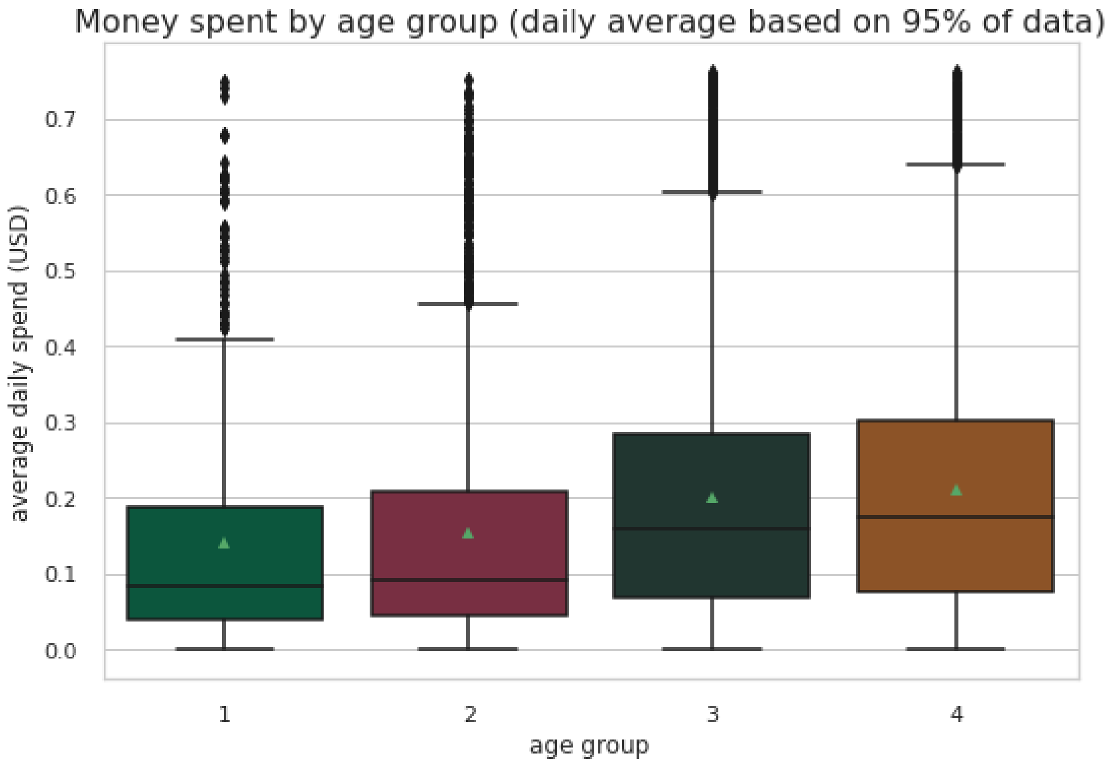

```
Mean values:
   gender  daily_spend_avg
0       1         0.167434
1       2         0.235465
2       3         0.219009
```

Females are the highest spenders, they spend on average 0.23USD per day followed by gender O who spend 0.21USD per day and least spenders are males who spent on average 0.16USD.

1.3.9 Daily average spend of customers by Age Group

[356]:

```
```



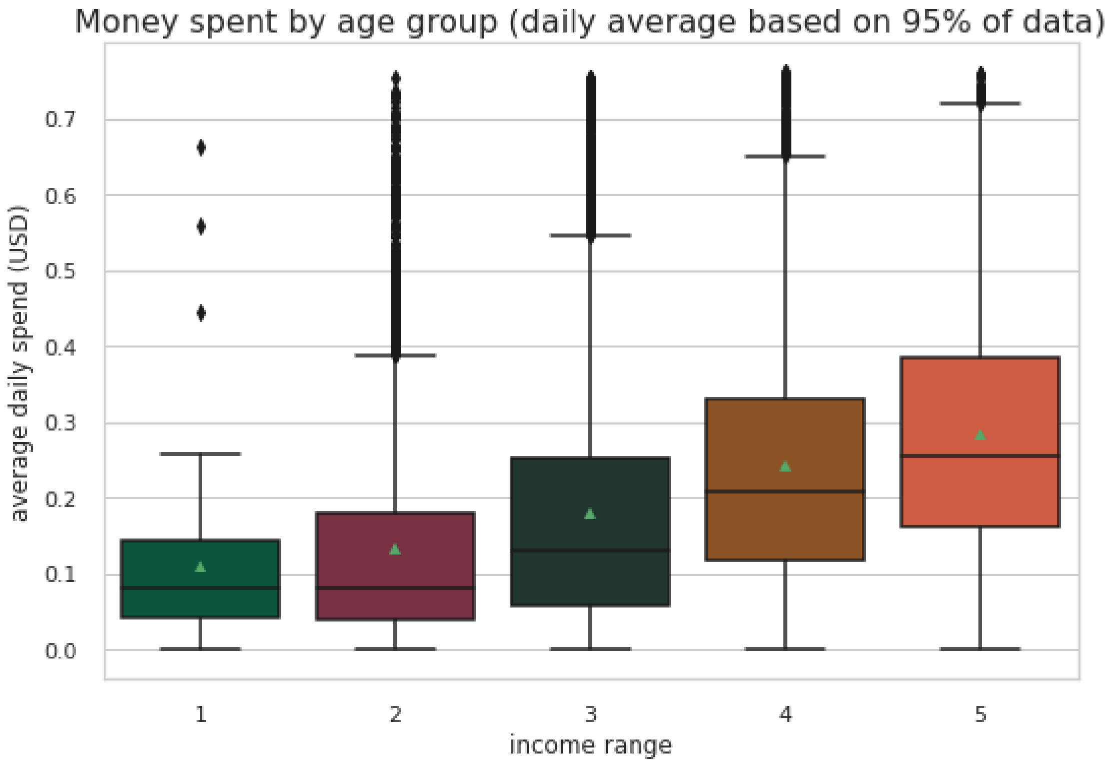

```
Mean values:
   age_group  daily_spend_avg
0          1         0.141821
1          2         0.153142
2          3         0.199117
3          4         0.211793
```

The elderly, that is those above 60 are the highest age group spending an average of 0.21US per day. Adults, those aged 35-60 are the second highest spenders on average, spending 0.19US per day. Ages 17-34 are the least spenders, spending roughly 0.14US per day

1.3.10 Daily average spend of customers by income range

[357]:

```
```



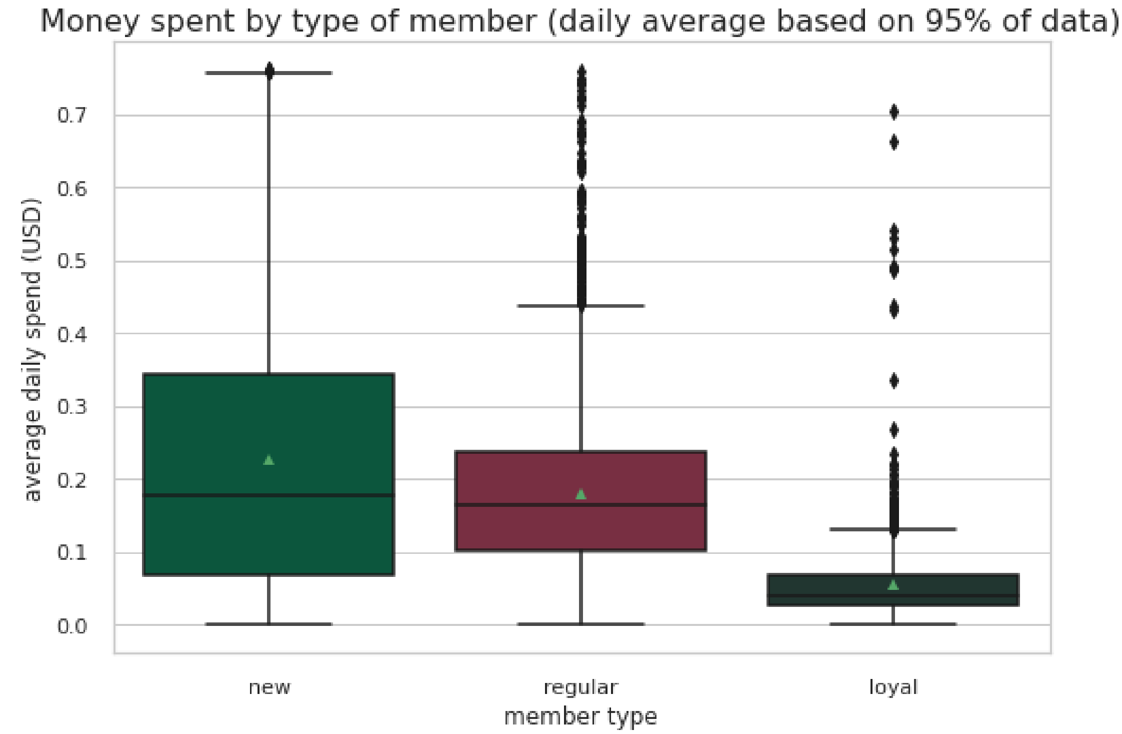

```
Mean values:
   income_range  daily_spend_avg
0             1         0.108308
1             2         0.133231
2             3         0.179468
3             4         0.240649
4             5         0.283618
```

As expected, high income earners have the highest average spend. Spending on average, 0.28US per day, mid to high earners also spend highly with 0.23US per day. Low and low to mid income earners spend right around the same amount. They spend about 0.11US per day

1.3.11 Daily Average spend by member type

[358]:

```
```



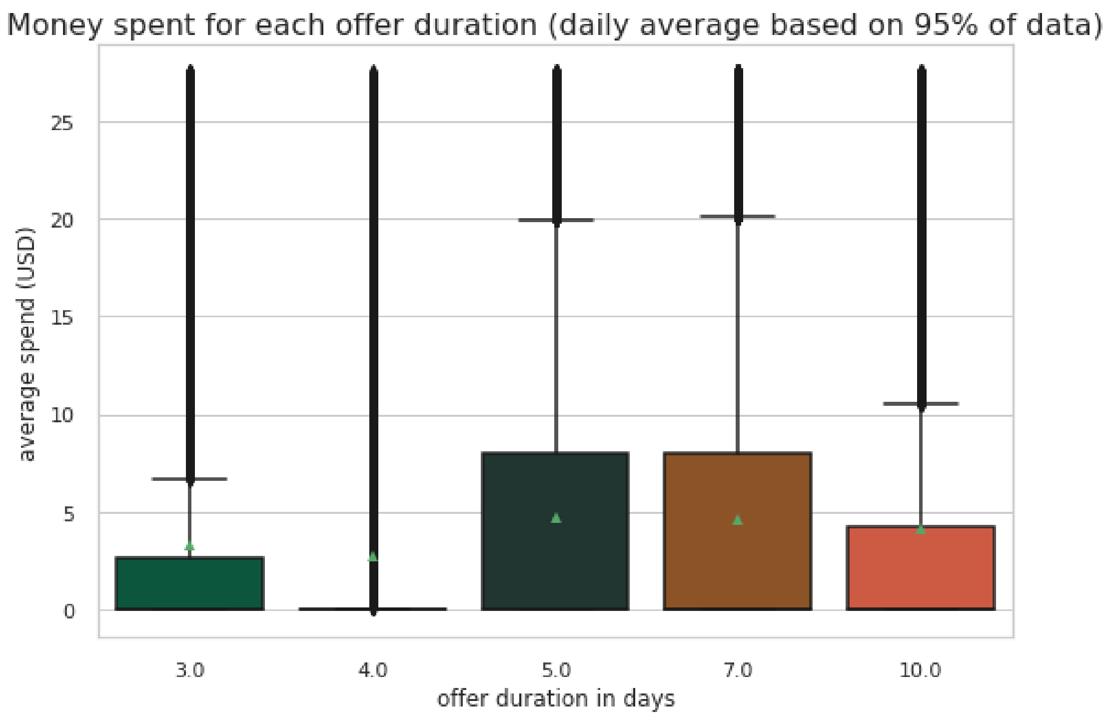

```
Mean values:
  member_type  daily_spend_avg
0       loyal         0.055500
1         new         0.225833
2     regular         0.178587
```

New members have the highest average dailly spend followed by regular members. Loyal members, those who have been members for over 3 years are no longer daily spenders

Money spent for duration of offer

[369]:
```
new_data = master_df[master_df.money_spent < master_df.money_spent.quantile(0.
↪95)]
```

[371]:
```
```

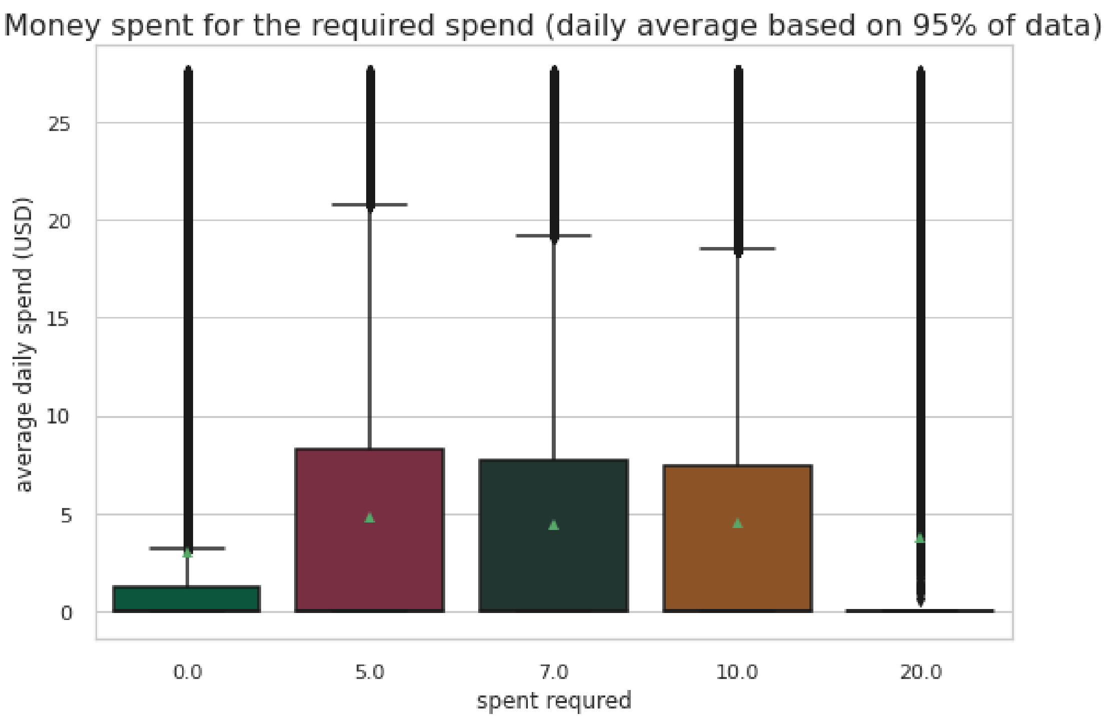



```
Mean values:
   offer_duration_days  money_spent
0                  3.0     4.402707
1                  4.0     4.005438
2                  5.0     6.907186
3                  7.0     6.593053
4                 10.0     6.539640
```

More money was spent for offers that lasted longer, on average 2USD more was spent on offers that lasted more than 5 days and more

Summary of average daily spending has revealed that: * High earning elderly females who are new members have the highest daily average spend i.e Female members who are over 60 years earning over 90K and have been members for at least 1200 days

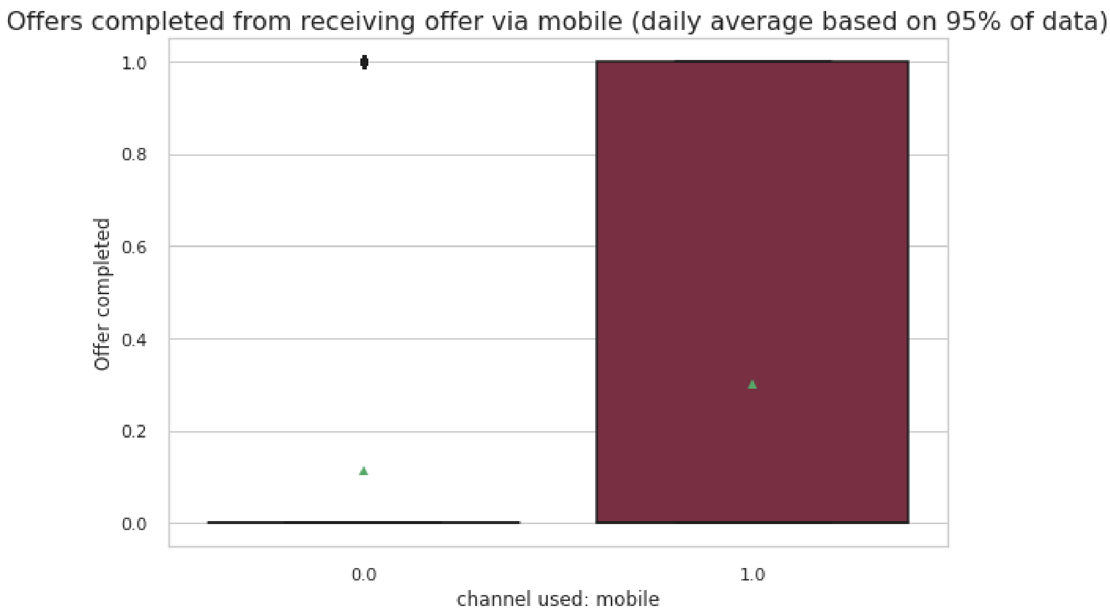
[372]:




```
Mean values:
   spent_required  money_spent
0             0.0     4.224865
1             5.0     6.736110
2             7.0     6.279434
3            10.0     6.762536
4            20.0     6.492396
```

The more spent required the more money was spent but it capped at 6.76USD

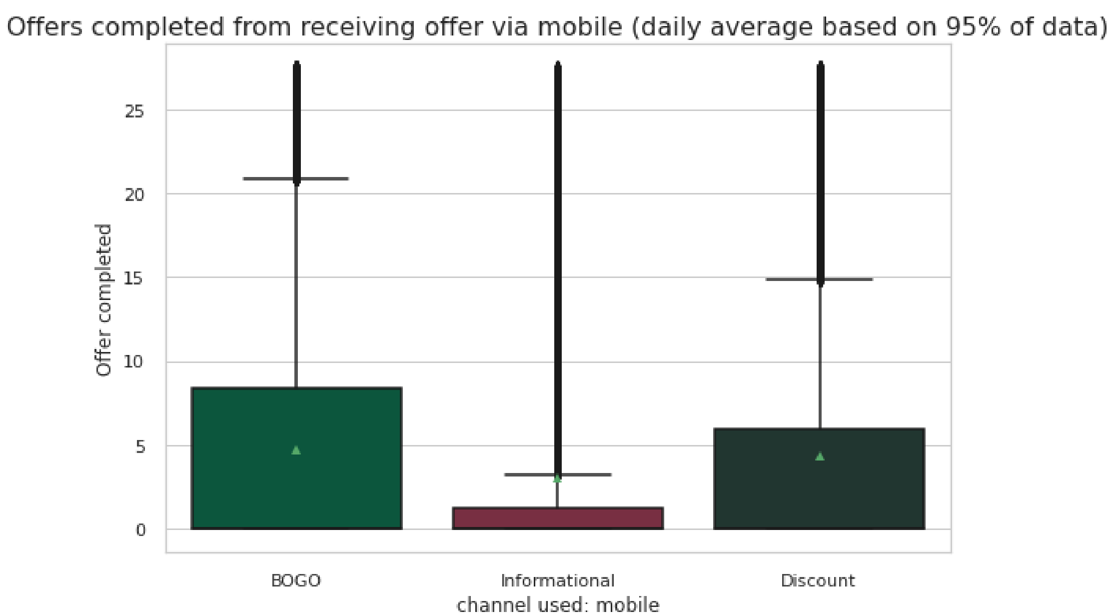
[376]:




```
Sum of values:
   channel_mobile  offer_completed
0             0.0             1367
1             1.0            35532
```

Most people who received offer via mobile completed it. It seems that mobile is the most important channel

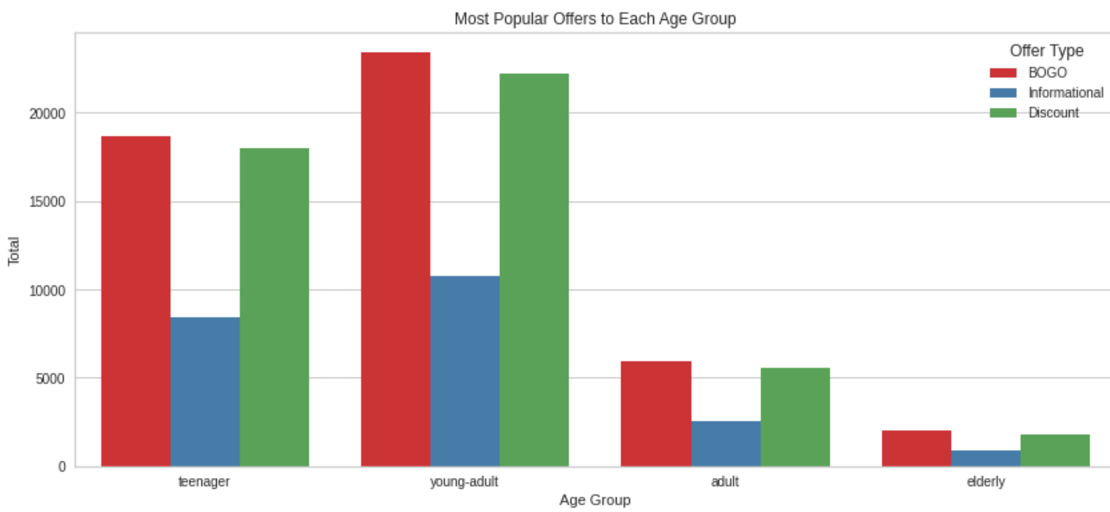
[377]:



```
Sum of values:
      offer_type  money_spent
0           BOGO     6.851173
1       Discount     6.464640
2  Informational     4.224865
```

1.3.12 Distribution of offers among age groups

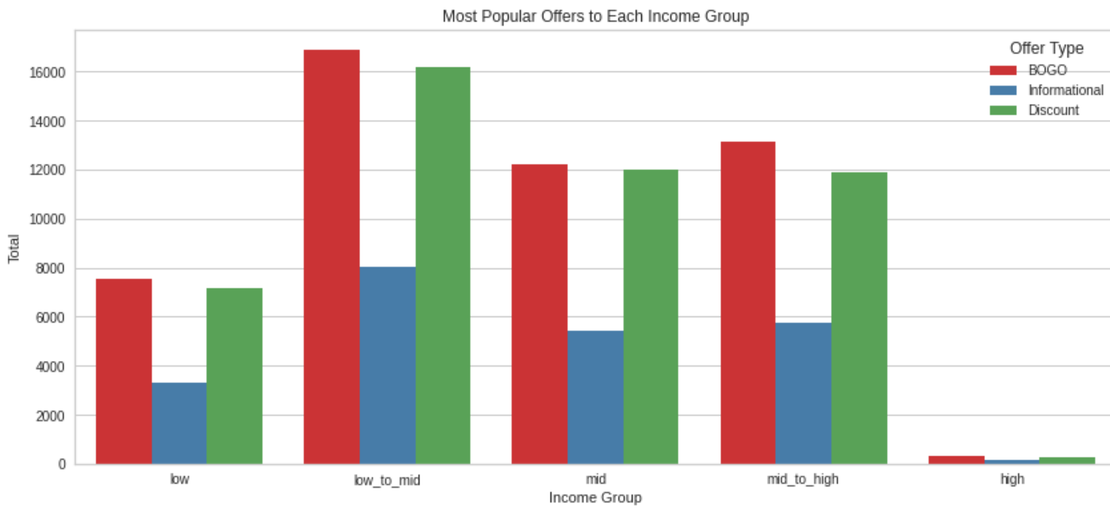
[157]:



Overall across all age groups, BOGO is the most occuring type of offer. The distribution of offers follows the same pattern accross all age groups. BOGO being most popular closely followed by discount with the informational offers being the least by a margin.

1.3.13 Distribution of offers by Income Group

[159]:

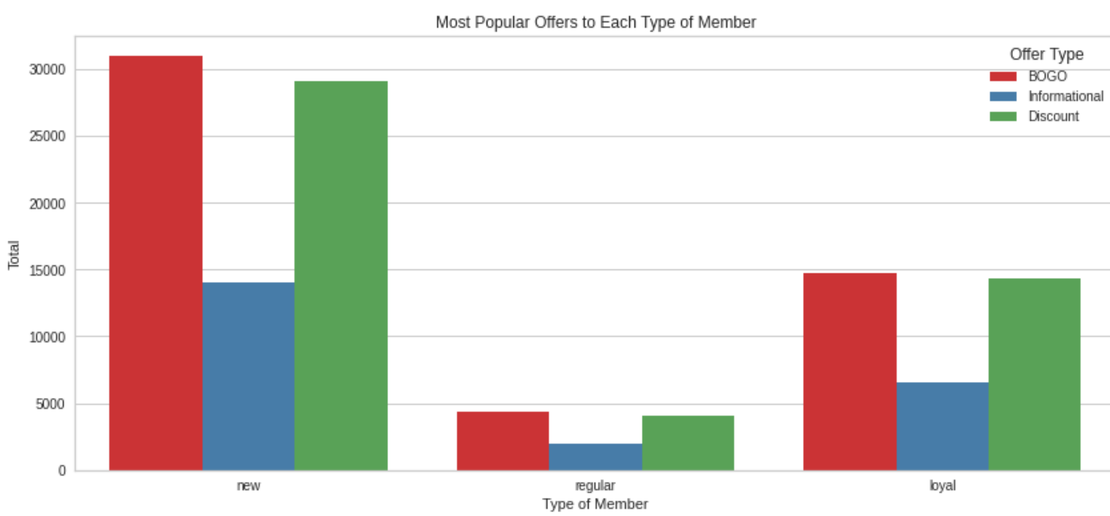


Most offers are concentrated around low to mid income earners. The second highest number of offers were sent to mid and mid to high income earners. High earners received the least offers. In terms of pattern, the offers follow the same pattern accross all income groups

1.3.14 Distribution of offers by Member Type

[161]:

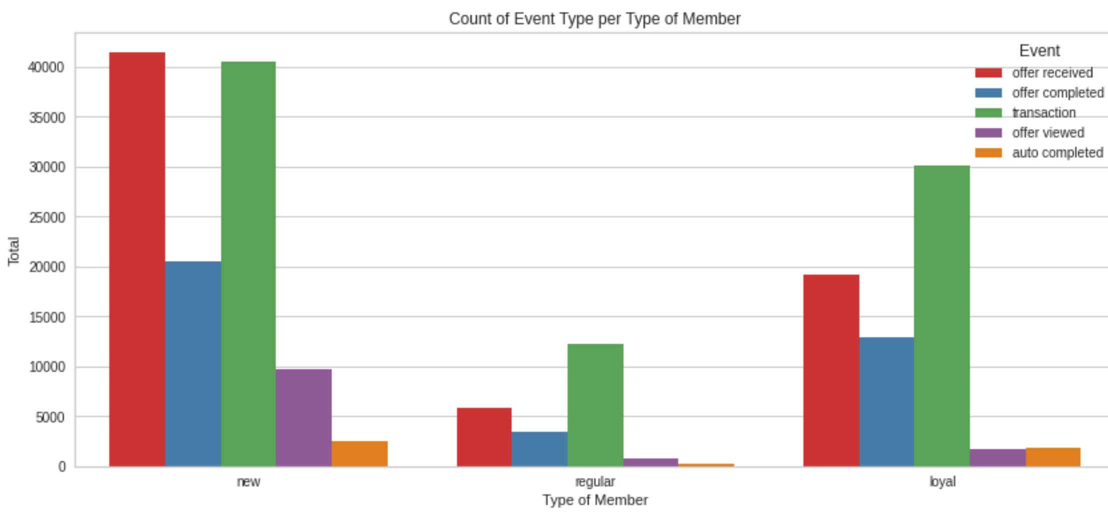
```

```



The highest number of offers were sent to new members followed by loyal members with regular members receiving the least amount of offers. In terms of popularity of offers, the same pattern consistently emerges of BOGO being highest with discount close by and low informational offers

1.3.15 Distribution of Events per Type of Member

[163]:



41 395 new customers received offers. Of those that received offers, only 41.5% viewed the offer and 37.5% completed the offer.

40 539 new customers made transactions on the app, new members made the highest number of transactions of all the members. Regular members also made a sizeable amount of transactions on the app, they made 30 036 transactions. Loyal members had the least number of transactions, making 12 299 transactions

Regular members were the second highest to receive offers, receiving 19 223 offers but only 24% viewing the offers and 60% completing the offers. Most regular members completed the offers without viewing them.

5785 loyal customers received offers, of this number, 38.2% viewed the offers and 41% completed the offers.

An indepth analysis of the types of offers given to customers shows interesting observations.

On new customers:

39.88% of new customers received BOGO offer, 41.4% of those that received the BOGO viewed it, which constituted 39.8% of all offer views by new customers. 45% of those new customers that received the BOGO completed it, accounting for 48.8% of all new customers that completed offers.

40% of new customers received the discount offer, 27.5% of those that received this offer viewed the offer, accounting for 26.6% of all offer views by new customers. 47.8% of new customers who received the discount offer completed it, which accounts for 51.1% of all offers completed by new customers.

20% of new customers received informational offers, 69.5% of them viewed it, which accounts for 33.5% of views by new customers.

None of the new customers completed informational offers.

On regular customers:

39.77% of regular customers received the BOGO offer, of these only 18% viewed it. This means that 29.4% of all views of BOGO offer were done by regular customers. 74% of regular customers who received the BOGO offer completed it. This accounts for 48.6% of all offers completed by regular customers

40.3% of regular customers received the discount offer, of these only 7% of them viewed it. This means that 12.1% of offer views by regular customers were viewing discount offer. 77.1% of regular customers who received discount offer completed it. This means 51.4% of all offers completed by regular customers was on discount offers.

19.91% of regular customers received the informational offer, of these 71.6% of them viewed it. 23.5% of all views made by regular customers were on informational offers.

None of the informational offers were completed.

On loyal customers:

40.3% of loyal customers received the BOGO offer, of these 46% of them viewed it. This means that, 48.85% of all loyal customer views were on BOGO. 40.5% of loyal customers who received BOGO completed it. This accounts for 39.86% of all offers completed by loyal customers.

39.6% of loyal customers received discount offer, 13.2% viewed it. This means that 13.7% of all views by loyal customers was on discount offer.62.28% of loyal customers who received the discount offer completed it. This accounts for 60% of all offers completed by loyal customers.

[167]:
```
percent_success
```

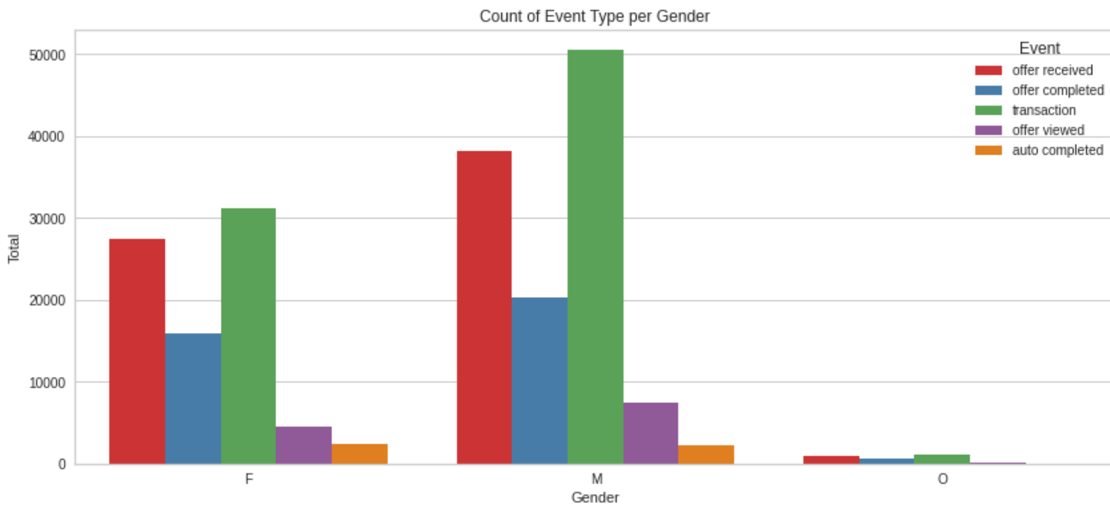
[167]:

	offer_id	count	percent_success
0	bogo_3	14372	29.139994
1	discount_4	14002	27.931724
2	fafdcd668e3743c1bb461111dcafc2a4	18062	27.699037
3	discount_1	12327	27.468159
4	discount_2	17920	27.265625
5	bogo_2	16989	24.150921
6	bogo_1	16241	22.517086
7	bogo_4	16232	20.391819
8	info_1	10144	0.000000
9	info_2	12516	0.000000

The bogo_3 is the most succesful offer with 29% success. The discount offers perfomed pretty well averaging 27%.

1.3.16 Distribution of Events per Gender

[172]:
```

```



[174]:
```
print("Number of BOGO offers that were received by females but not completed:")
len(master_df[(master_df.offer_type == "BOGO") & (master_df.gender == "F") &␣
↪(master_df.event == "offer received")])
```

```
Number of BOGO offers that were received by females but not completed:
```

[174]: 10975

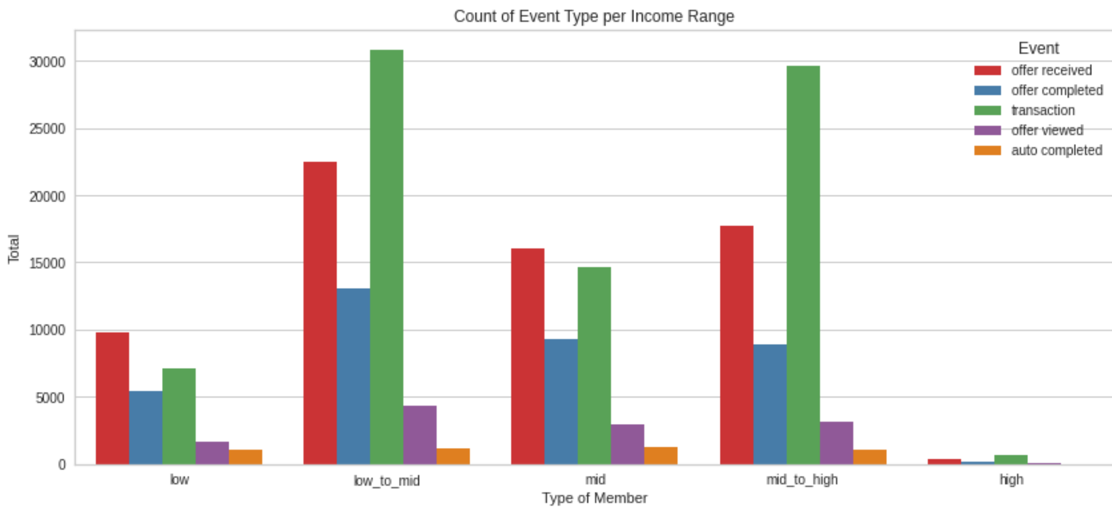
Female BOGO offer: * 10 975 females received the BOGO offers, 1368 viewed it but did not complete it. 7566 received and completed it. Only 995 of them auto completed it. **Female Discount offer**: * 10 943 females received the Discount offers, 1267 viewed but did not complete. 6369 received and completed it.1360 of them auto completed it. **Female Informational offer**: * 5538 females received the informational offer, 1242 viewed but did not complete the offer. 2668 received and completed the offer.

Male BOGO offer: * 15 208 males received the BOGO offer. Of these 2534 viewed the offer but did not complete it. 9785 received and completed the offer. 963 completed the offer automatically. **Male Discount offer**: * 15 354 males received the discount offer. Of these 2201 males only viewed the offer. 8049 received and completed the offer. 1271 auto completed the offer **Male Informational offer**: * 7567 males received the informational offer. 1480 males viewed the offer without completing it. 3809 received and completed it

Gender unspecified BOGO offer: * 354 of these customers received the offer. 56 only viewed the offer. 253 received and completed the offer while 20 auto completed it.

1.3.17 Distribution of Events per Income range

[175]:
```

```

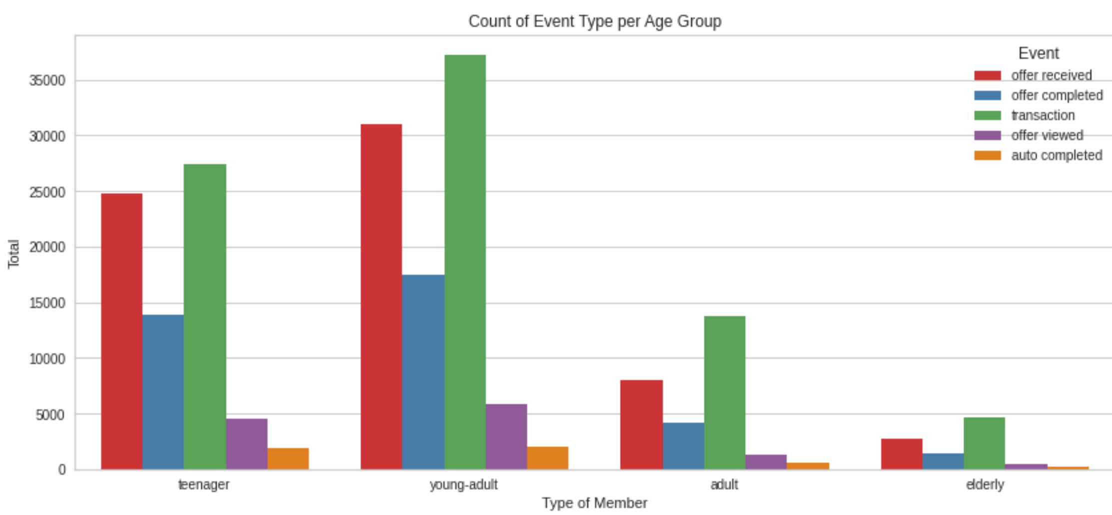


Mid income earners received and completed the most offers and made the most transactions.

Low to mid income earners were the second highest in terms of offers received and completed.

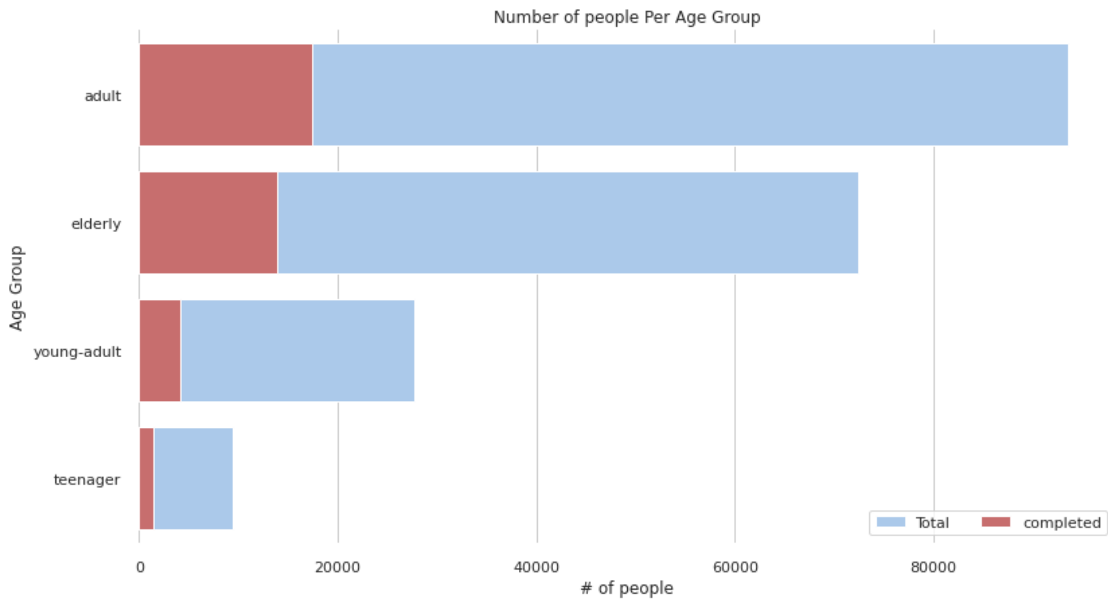
1.3.18 Distribution of events by age group

[178]:



Analysis of those who completed the offer by age group

[181]:



There are 3 clusters of those that completed the offer those that were older and had high income. Those that were older and had mid income and those that were middle aged and middle income earners. We will do an advanced customer segmentation

Age distribution of those who completed the offer

[185]:

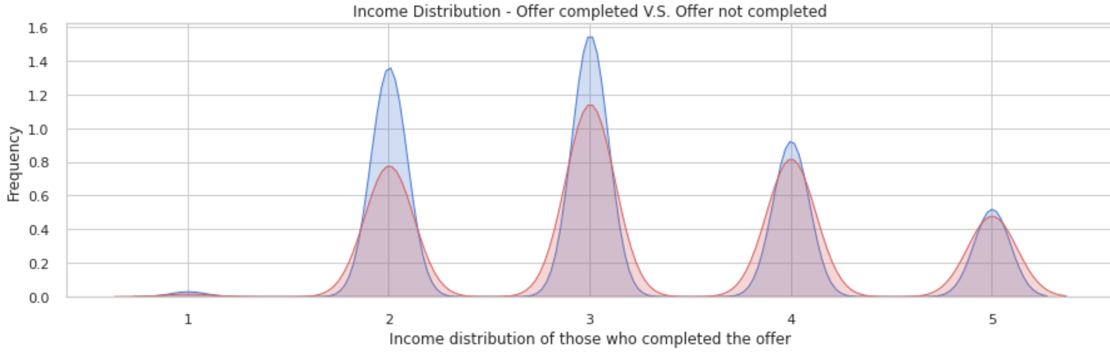
[185]: Text(0.5, 1.0, 'Age Distribution - Offer completed V.S. Offer not completed')

Most of the customers that completed offers were adults and the elderly. The least to complete offers were the teens and young adults

Income distribution of those who completed the offer

[186]:

[186]: Text(0.5, 1.0, 'Income Distribution - Offer completed V.S. Offer not completed')



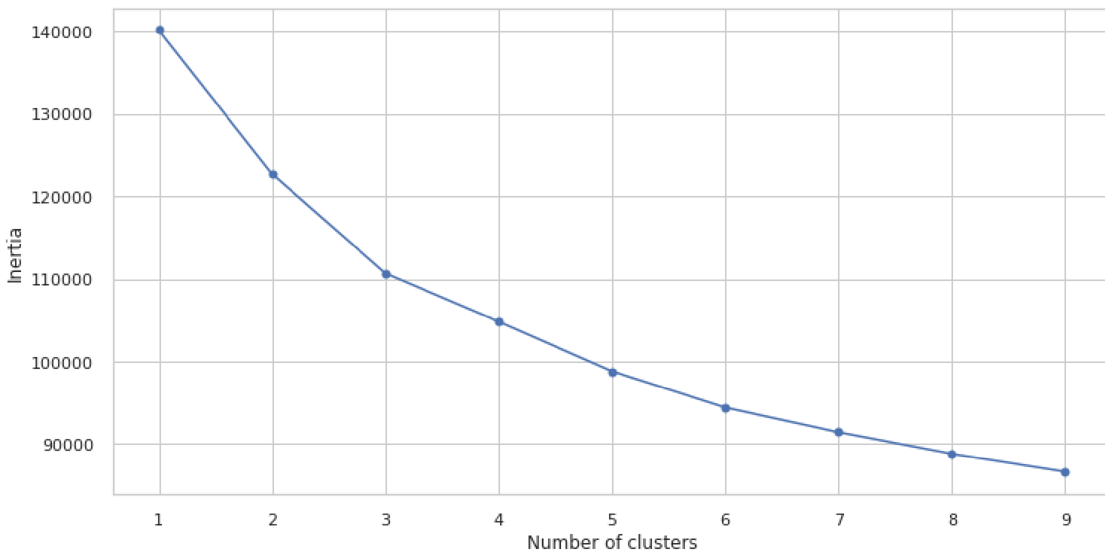
The bulk of those that completed offers were low to mid and mid income earners

Customer clusters

Dimensionality reduction

[209]:

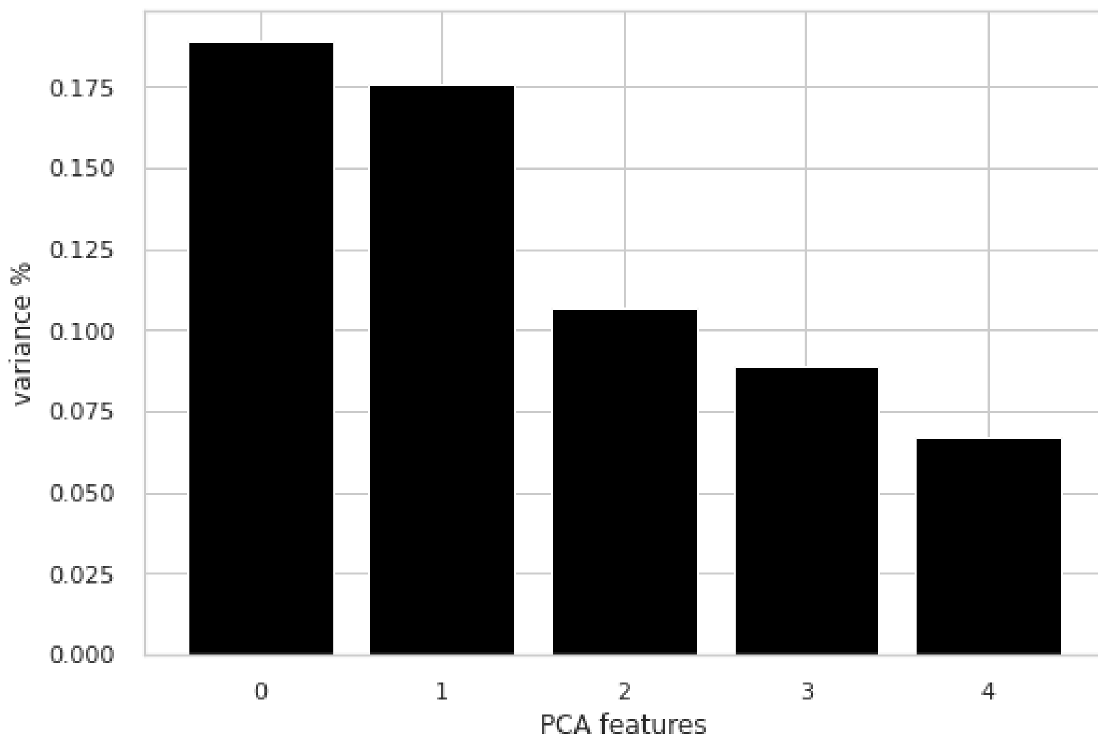
[209]: Text(0, 0.5, 'Inertia')



After 5 clusters the the models begins to deteriorate, so we will pick 5 as optimal

Explained variance of the PCA

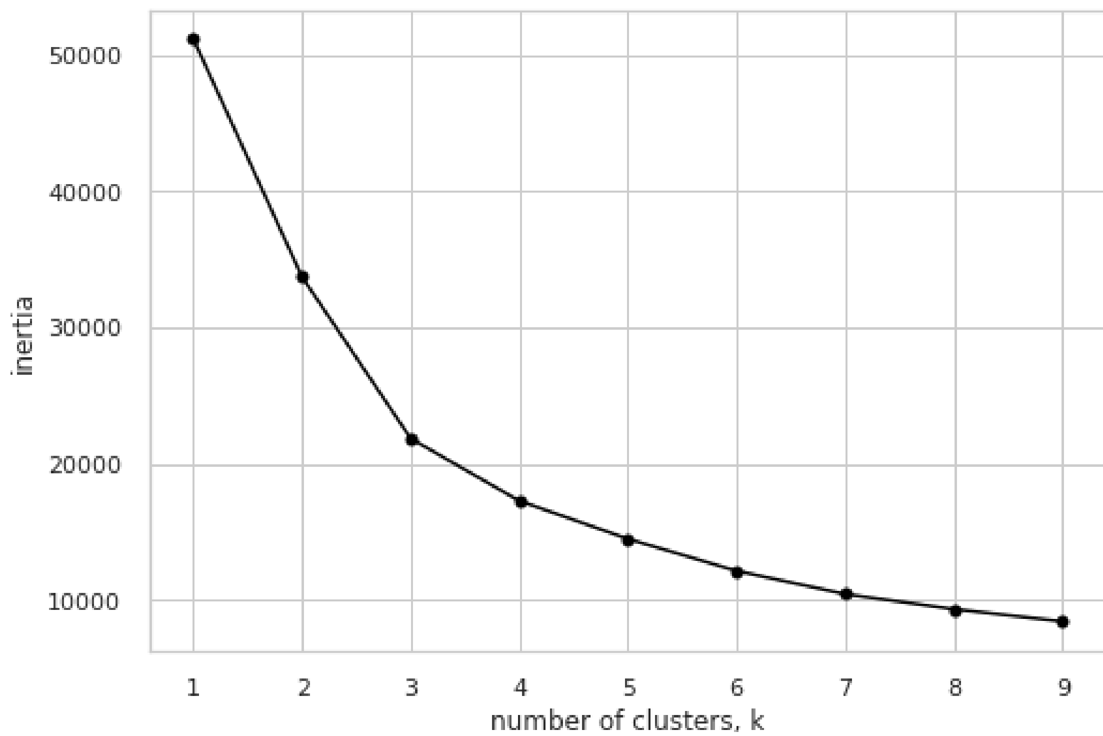
[210]:



The first 5 components explain 80% of the variance

[211]:

```

```

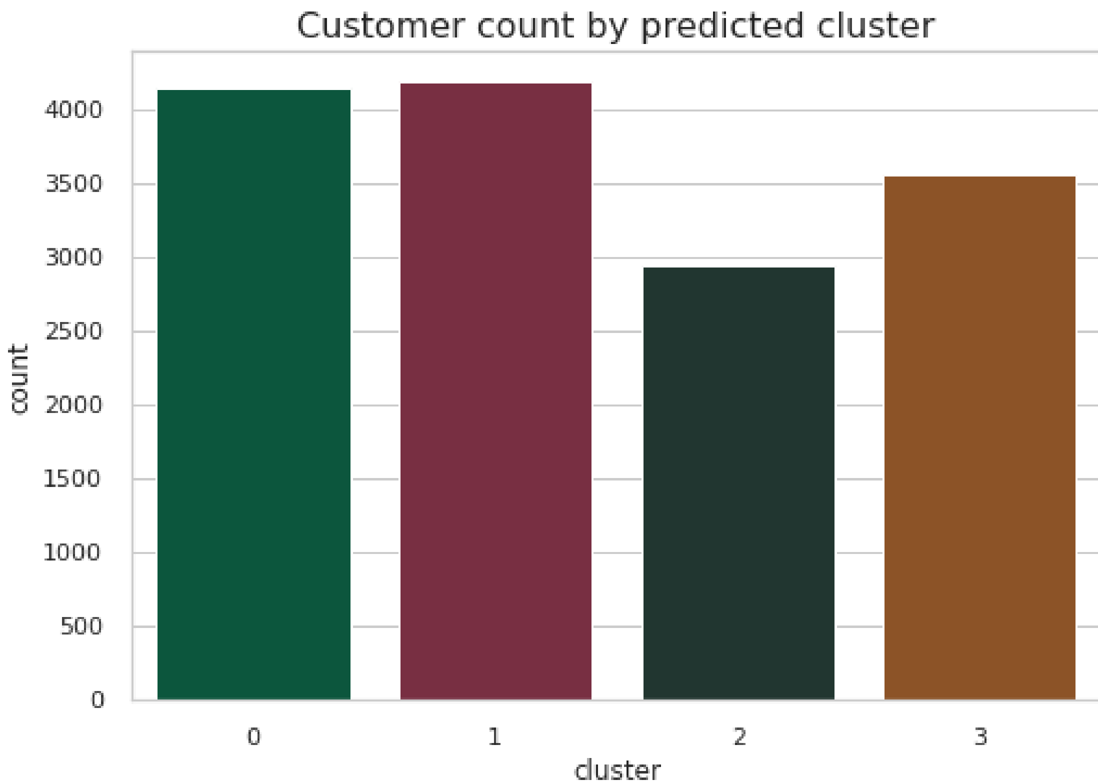


[220]:
```python
customer_df.sample(3)
```

[220]:
```
       Unnamed: 0  BOGO  Discount  Informational       time  \
3421         3421     4         4              4  15.107143
11645       11645     2         8              2  14.777778
7701         7701     3         4              1  12.516667

       offer_duration_days  money_spent  money_gained  offer completed  \
3421                 6.000     3.321429      0.238095                2
11645                6.500    25.943333      0.777778                5
7701                 8.125     2.910000      0.333333                1

       offer ignored  casual purchase  offer auto completed  cluster  gender  \
3421               6               11                     0        0  Female
11645              6               20                     0        2    Male
7701               2                8                     0        1    Male

                 event      income member_type    age_group     offer
3421   casual purchase  low_to_mid         new        adult      BOGO
11645  casual purchase         mid     regular  young-adult  Discount
7701   casual purchase         mid       loyal      elderly  Discount
```

Customer distribution per predicted cluster

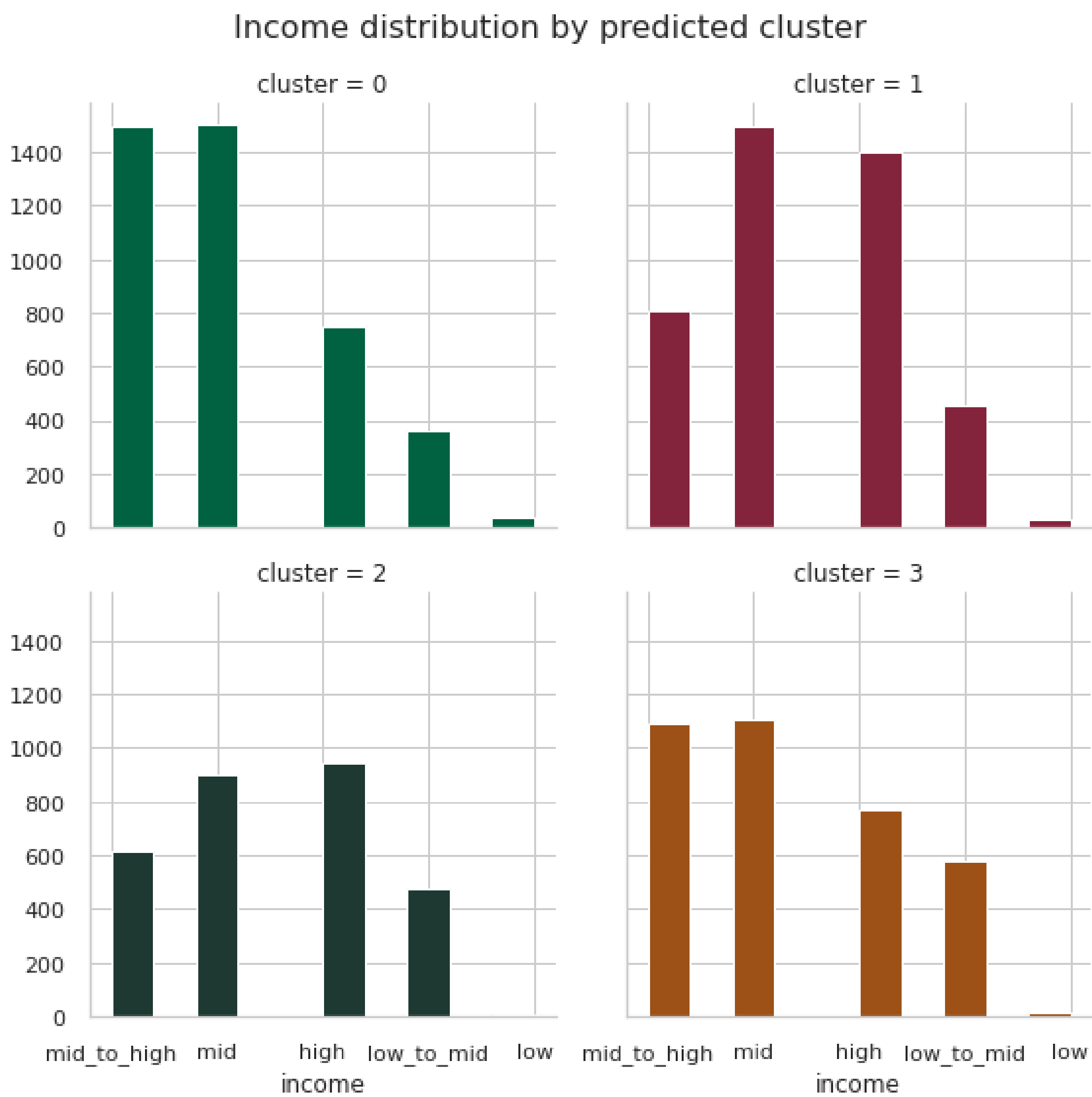
[221]:



There's almost an even distribution of customers among the four predicted clusters

Income Distribution among the predicted clusters

[222]:



Clusters 0 has mostly mid to high and high earners, cluster 1 has mid and high income earners.

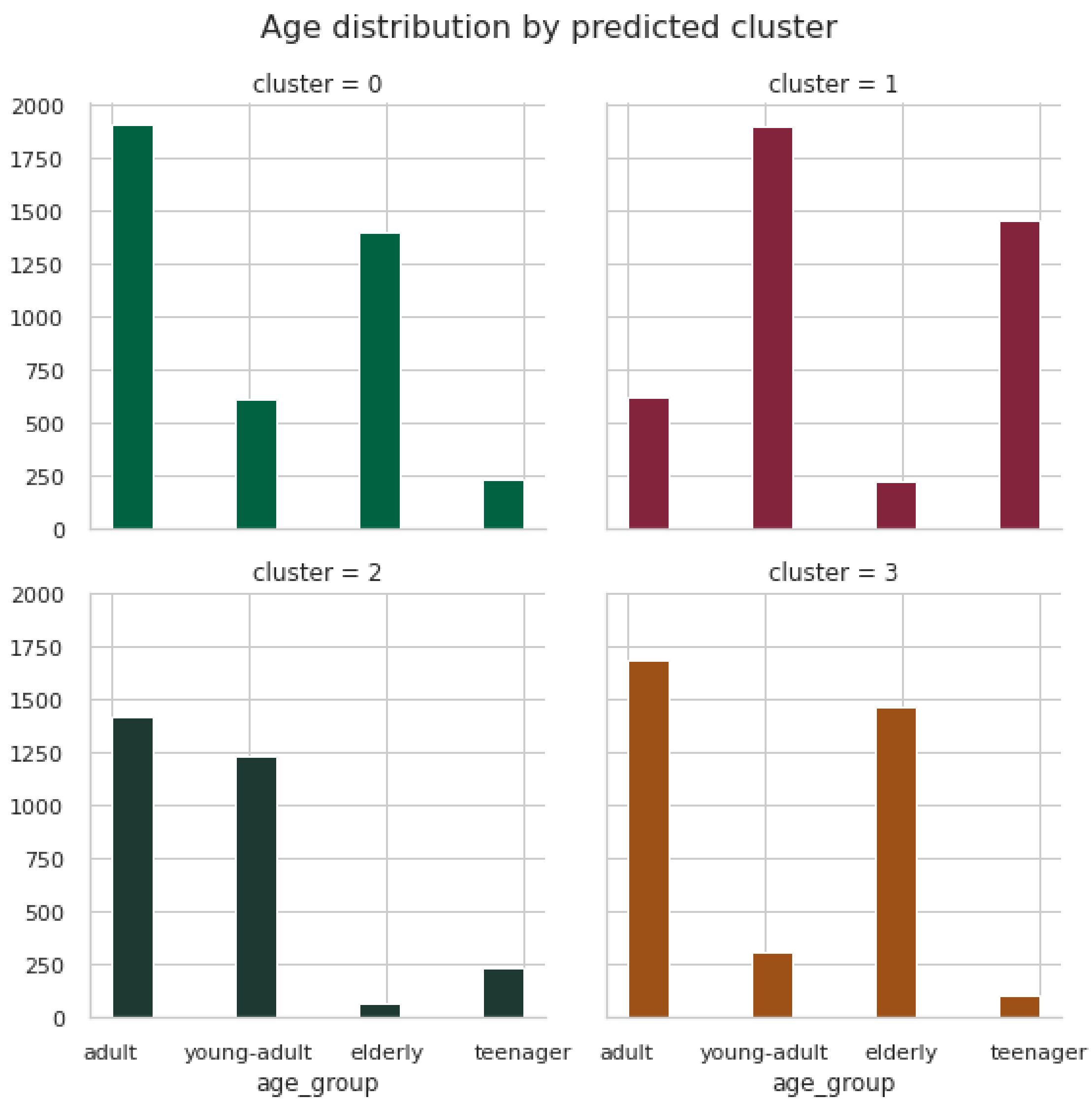
Cluster 2 has a normal distribution from between low to mid to high income earners with mid and high income earners being the most.

Cluster 3 is mostly skewed towards high income earners

Age distribution per predicted cluster

[223] :

```
```

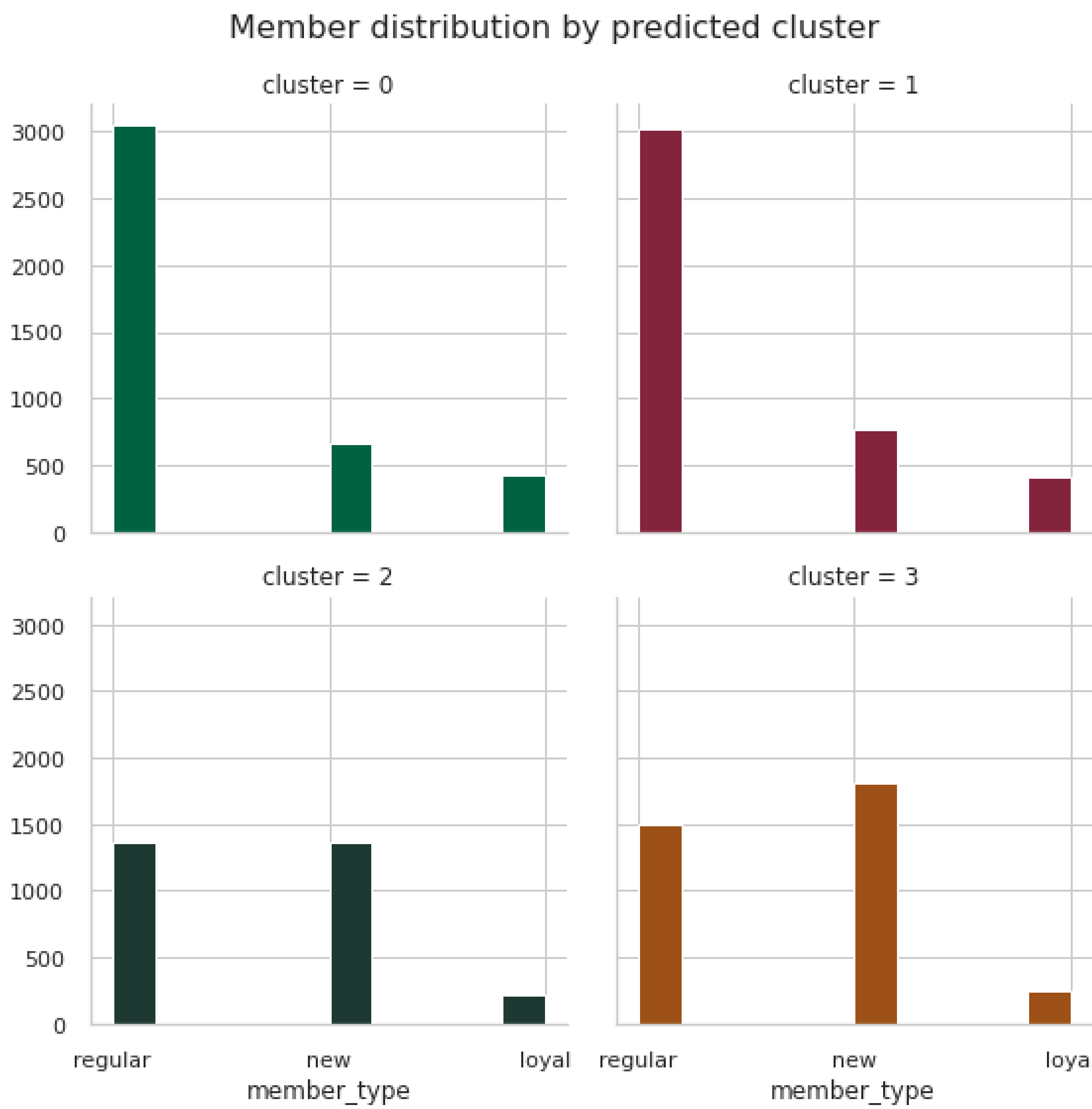


Cluster 0 has a lot of elderly and young adults mostly. Cluster 2 has adults and teenagers. Cluster 3 has adults and the elderly

Member distribution per predicted cluster

[224]:

```

```

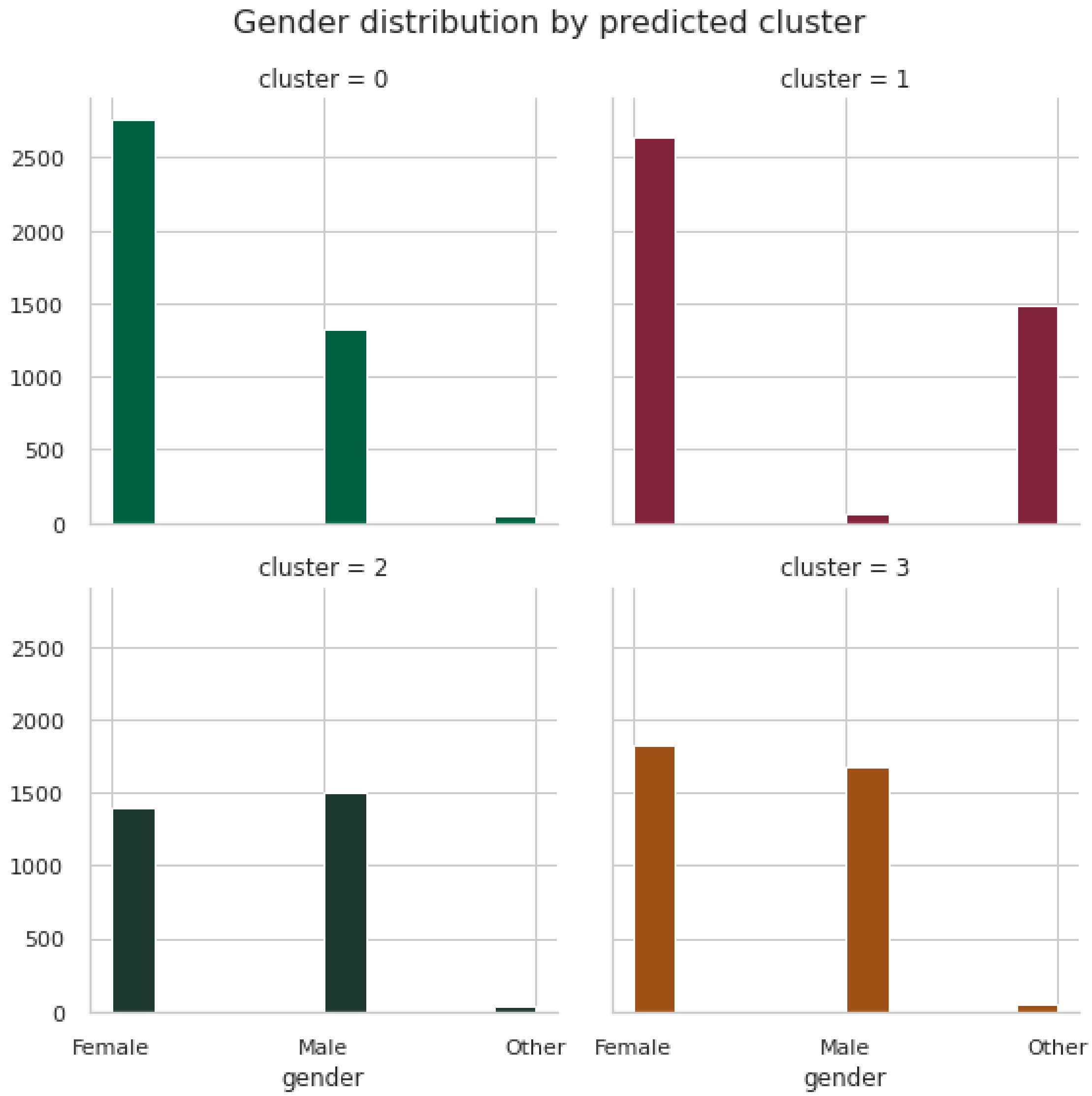


Cluster 0 and 1 have mostly regular members. Cluster 2 and 3 have an almost equal distribution of regular and new members

Gender distribution per predicted cluster

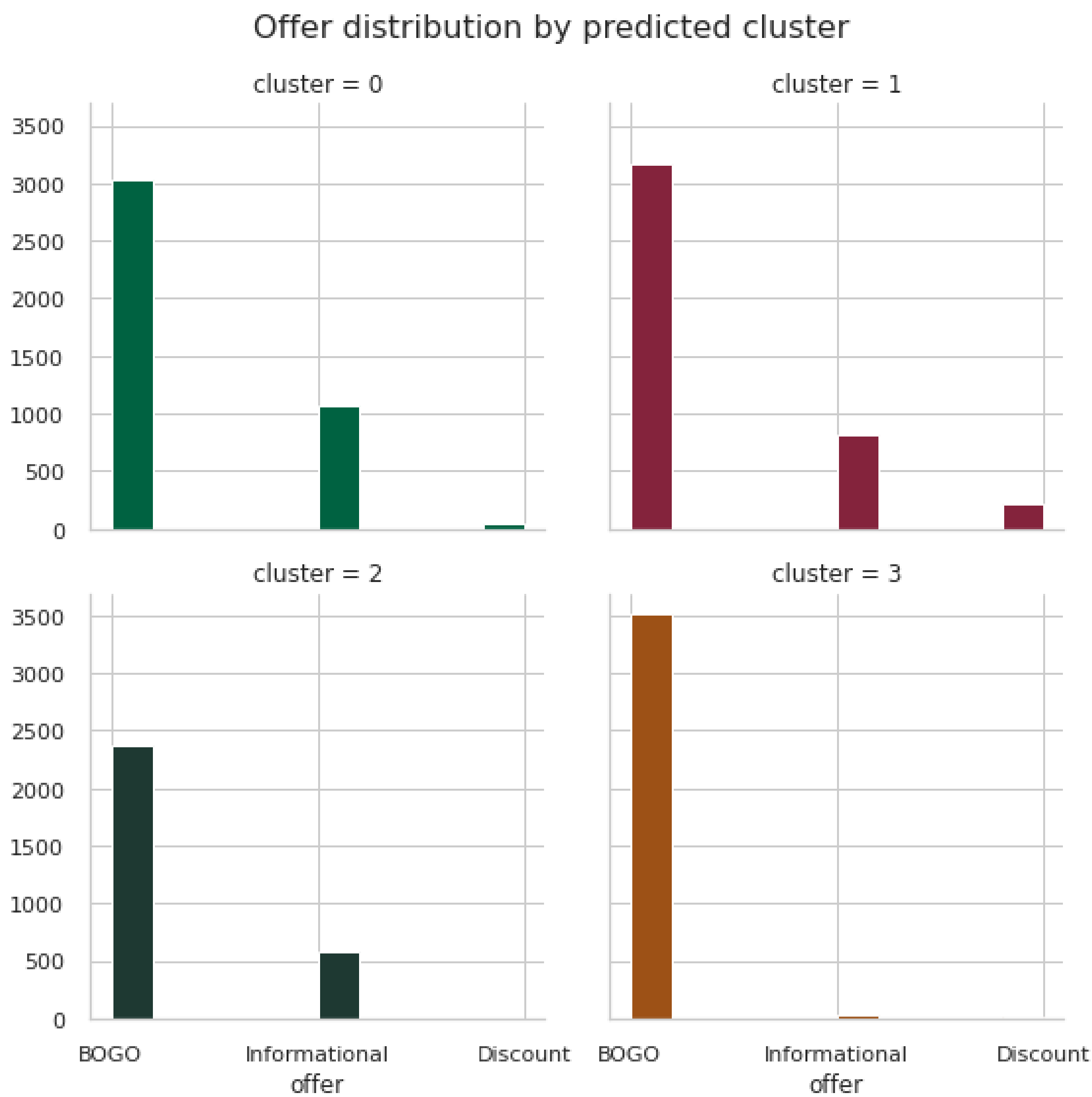
[225]:

```

```



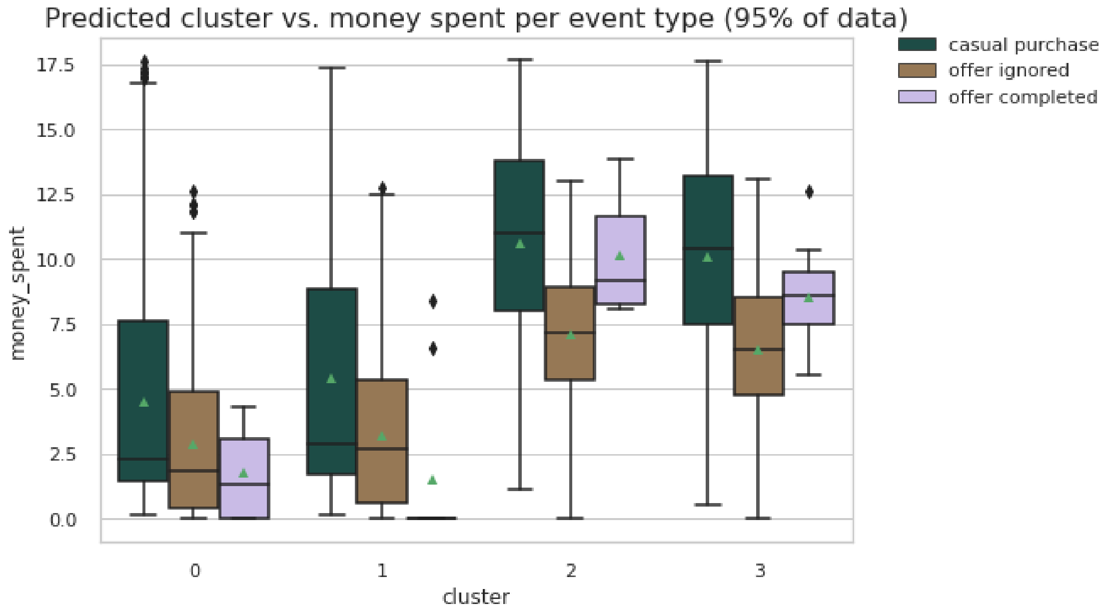
Offer distribution per predicted cluster

[226]:



Money spent by each predicted cluster

[227]:



Cluster 2 and 3 made the most transactions. Cluster 3 has the highest average money spent for both transactions and completed offers. Followed closely by cluster 2. Cluster 0 had a few outliers but most customers in that cluster did not complete offers, a large number of them made casual purchases and or ignored the offers.

Clusters 2 and 3 are the groups to target with offers.

- Cluster 2 has a lot of customers in the mid and high earning bracket, has mostly adults and teenagers who are new and regular members and evenly distributed among males and females
- Cluster 3, the highest spenders, has a customers normally distributed between low to mid and high earning bracket, with mid and mid to high earners being the highest spenders in the group. In terms of member type it is evenly distributed between new and regular members with very few loyal members, same even distribution among males and females and very very few other genders. This cluster is filled with adults and the elderly which is consistent with what we have been finding

[228]:

Cluster 3 customers spent a lot on Informational offers, more than 50% of them spend above average (10USD) on it. They also spend the highest on average on Discount offers, most of them spending between 7.5 to 12.5USD. They also spent a lot on BOGO offers though not the most popular offer among this cluster. They spent a minimum of 5USD which is higher than the minimum spend for all other clusters. Most of them spent an average of 9USD on BOGO offers.

Informational offers were not popular among cluster 2 who did not spent on it. Only a few outliers completed this offer. They spent quite a lot on discount and BOGO offers spending an average of 10USD on both which was the second highest spend on offers among the clusters.

Cluster 1 liked and spent the most on informational offers spending between 2.5 and 10USD on them. They spent about the same on BOGO and Discount spending on average between 1 to 7.5USD.

Cluster O also like all 3 offers with Informational being the most popular,followed by discount and last BOGO offers. Cluster O and 1 spent the least on all offers

1.4 Machine Learning Methodology

1.4.1 Data Preprocessing Steps

Common steps for both models:

- Clean the data

- Drop the "unknown" customer data
- Drop the columns that will not be used for clustering

Algorithm specific steps:

K-MEANS ALGORITHM:

- use K means clustering algorithm to create customer clusters
- use PCA for dimensionality reduction
- Use features that explain at least 80% of the variance
- Encode "gender" using a one-hot encoding scheme
- Fill in missing values
- Data standardization to cater for class imbalance
- PCA will also be used in the second step, for modelling response of customers to offers
- Create formatted, k-means training data

Random Forest ALGORITHM :

- Prepare data labels - "no view", "no order", "possible order", this is a multi-classification problem
- Split the data and use 30% for cros validation
- Use random forest and gridsearch to find the best hyperparameters for the best model
- Convert labels to numeric
- Check for missing values
- Encode "gender" and "offer_type" using a one-hot encoding scheme
- Create new customers and deploy the model to see how well it performs

[236]:
```
columns_to_keep = ['customer_id','age','gender',
␣
 ↪'income','channel_social','channel_email','channel_mobile','channel_web','offer_duration_da
'money_gained','spent_required','days_of_membership',
 'offer_type',
 'offer_received',
 'offer_viewed',
 'offer_completed']
```

[238]:
```
%%time
improved_model_df=model_df.
 ↪groupby(['customer_id','gender','income','offer_type','spent_required','money_gained'],␣
 ↪as_index=False).sum()
```

```
CPU times: user 296 ms, sys: 28 ms, total: 324 ms
Wall time: 326 ms
```

[243]:
```
improved_model_df['label'].value_counts()
```

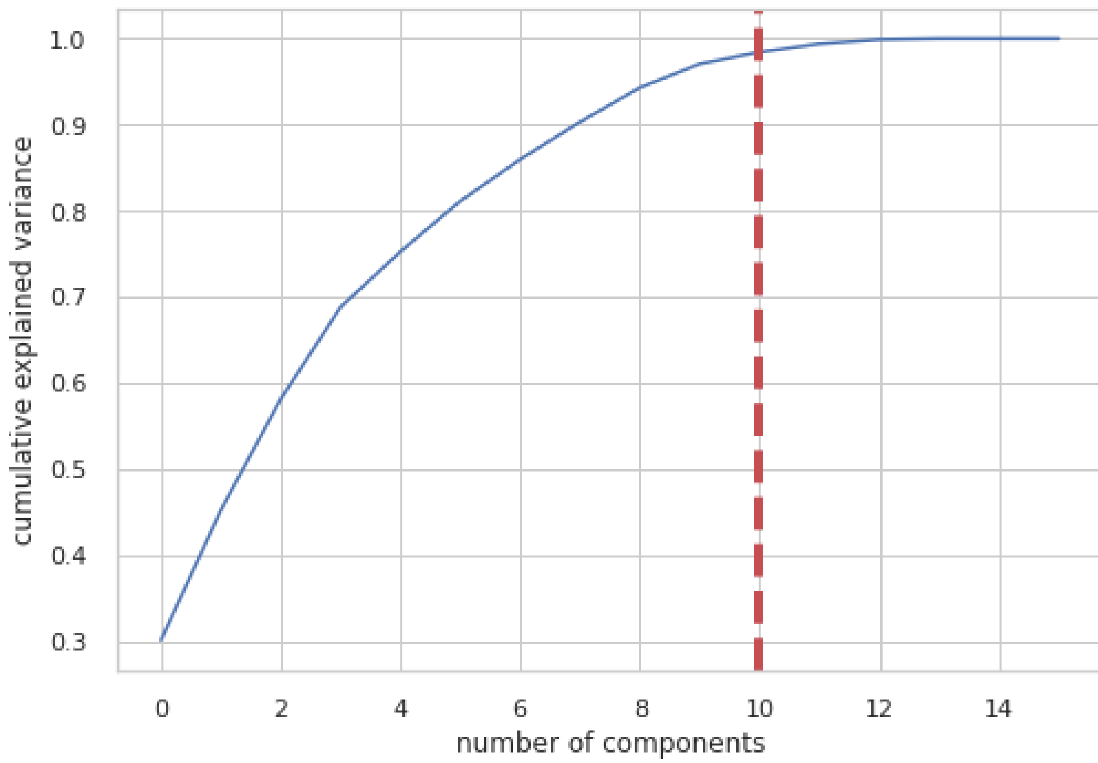
[243]:
```
no view           31804
possible order    20080
no order          19516
Name: label, dtype: int64
```

Cater for class imbalance by normalizing the data

[258]:
```
ss = StandardScaler()
X_train_scaled = ss.fit_transform(X_train)
X_test_scaled = ss.transform(X_test)
y_train = np.array(y_train)
```

Dimensionality reduction

[259]:
```

```




```
None

   Cumulative Variance Ratio  Explained Variance Ratio
0                   0.301023                  0.301023
1                   0.452145                  0.151122
2                   0.581108                  0.128963
3                   0.687933                  0.106825
4                   0.752388                  0.064455
5                   0.811185                  0.058798
6                   0.859530                  0.048345
7                   0.903105                  0.043575
8                   0.943013                  0.039908
9                   0.970594                  0.027581
```

The first 10 components explain more than 80% of the variance. So we reduce the features from

16 to 10

Show the 10 PCA components that will be used

[262]:

[262]:

	PCA Component 0	PCA Component 1	PCA Component 2 \
income	-0.028562	-0.036625	0.275378
spent_required	-0.169380	0.446360	0.073624
money_gained	-0.128817	-0.174937	0.072448
age	0.392124	0.040493	0.097037
channel_social	0.329502	-0.080087	0.005546
channel_email	0.441519	0.052701	0.026723
channel_mobile	0.432665	-0.088407	0.007308
channel_web	0.286502	0.273583	0.052277
offer_duration_days	0.331012	0.354808	0.065766
days_of_membership	0.275249	0.017564	-0.000360
gender_F	-0.034878	-0.062536	0.665758
gender_M	0.035851	0.062421	-0.667367
gender_O	-0.004605	-0.000464	0.016833
bogo	0.034535	-0.382530	0.020408
discount	-0.131470	0.563949	0.031534
informational	0.144968	-0.277401	-0.076936

	PCA Component 3	PCA Component 4	PCA Component 5 \
income	0.006455	-0.158450	0.891863
spent_required	0.296529	-0.015432	0.029398
money_gained	0.398101	-0.013212	0.193610
age	-0.001928	-0.018094	0.157566
channel_social	0.137463	-0.003726	-0.062453
channel_email	0.010143	0.005786	-0.001670
channel_mobile	-0.003196	0.009227	-0.025921
channel_web	0.102929	0.021088	-0.033636
offer_duration_days	0.176817	0.006436	-0.026223
days_of_membership	0.001778	-0.033827	0.094164
gender_F	-0.090922	-0.094660	-0.220687
gender_M	0.089813	-0.137836	0.181333
gender_O	0.003264	0.971754	0.161382
bogo	0.564491	0.008544	-0.096284
discount	-0.172596	-0.004933	0.034859
informational	-0.572614	-0.005212	0.089601

	PCA Component 6	PCA Component 7	PCA Component 8 \
income	-0.226312	0.089534	0.115347
spent_required	0.156203	0.057561	0.182167
money_gained	0.608934	-0.104287	-0.602865
age	-0.002695	0.043434	-0.110044
channel_social	0.227764	0.652029	0.168288

```
channel_email              0.032748     0.038297    -0.083064
channel_mobile            -0.010107     0.114983    -0.100969
channel_web               -0.367388    -0.445522    -0.353391
offer_duration_days       -0.019763     0.034665    -0.066331
days_of_membership         0.488847    -0.549413     0.578696
gender_F                   0.028768    -0.015380     0.009565
gender_M                  -0.029240     0.013446    -0.018437
gender_O                   0.002410     0.007860     0.037284
bogo                      -0.270397    -0.087753     0.181347
discount                   0.157231     0.140846    -0.063367
informational              0.163296    -0.080720    -0.172167

                     PCA Component 9
income                    -0.097069
spent_required             0.566722
money_gained              -0.072534
age                        0.380196
channel_social            -0.262367
channel_email              0.182826
channel_mobile            -0.203393
channel_web               -0.257306
offer_duration_days        0.162309
days_of_membership        -0.185649
gender_F                  -0.006354
gender_M                   0.007022
gender_O                  -0.002891
bogo                       0.082079
discount                  -0.327625
informational              0.367116
```

Prediction modelling

Train the baseline model for Random Forest

[263]:
```
%%time
rfc = RandomForestClassifier()
rfc.fit(X_train_scaled_pca, y_train)
display(rfc.score(X_train_scaled_pca, y_train))
```

```
1.0

CPU times: user 19 s, sys: 35 ms, total: 19 s
Wall time: 19.4 s
```

Hyperparameter Tuning

Hyperparameter Tuning Round 1: RandomSearchCV We will be tuning these hyperparameters: * n_estimators: the number of "trees" in our Random Forest. * max_features: the number of features at each split. * max_depth: the max number of "splits"each tree can have. * min_samples_split: the minimum number of observations required before a node of a tree can split itself. * min_samples_leaf: the minimum number of observations required at each leaf at the ends of each tree. * bootstrap: whether to use bootstrapping or not to provide data to each tree in the Random Forest. (Bootstrapping is a random sampling from the dataset with replacement.)

[437]:
```
%%time
rs.fit(X_train_scaled_pca, y_train)
rs.best_params_
```

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
CPU times: user 1min 53s, sys: 380 ms, total: 1min 53s
Wall time: 1h 53min 42s
```

[437]:
```
{'n_estimators': 500,
 'min_samples_split': 12,
 'min_samples_leaf': 18,
 'max_features': 'sqrt',
 'max_depth': 13,
 'bootstrap': False}
```

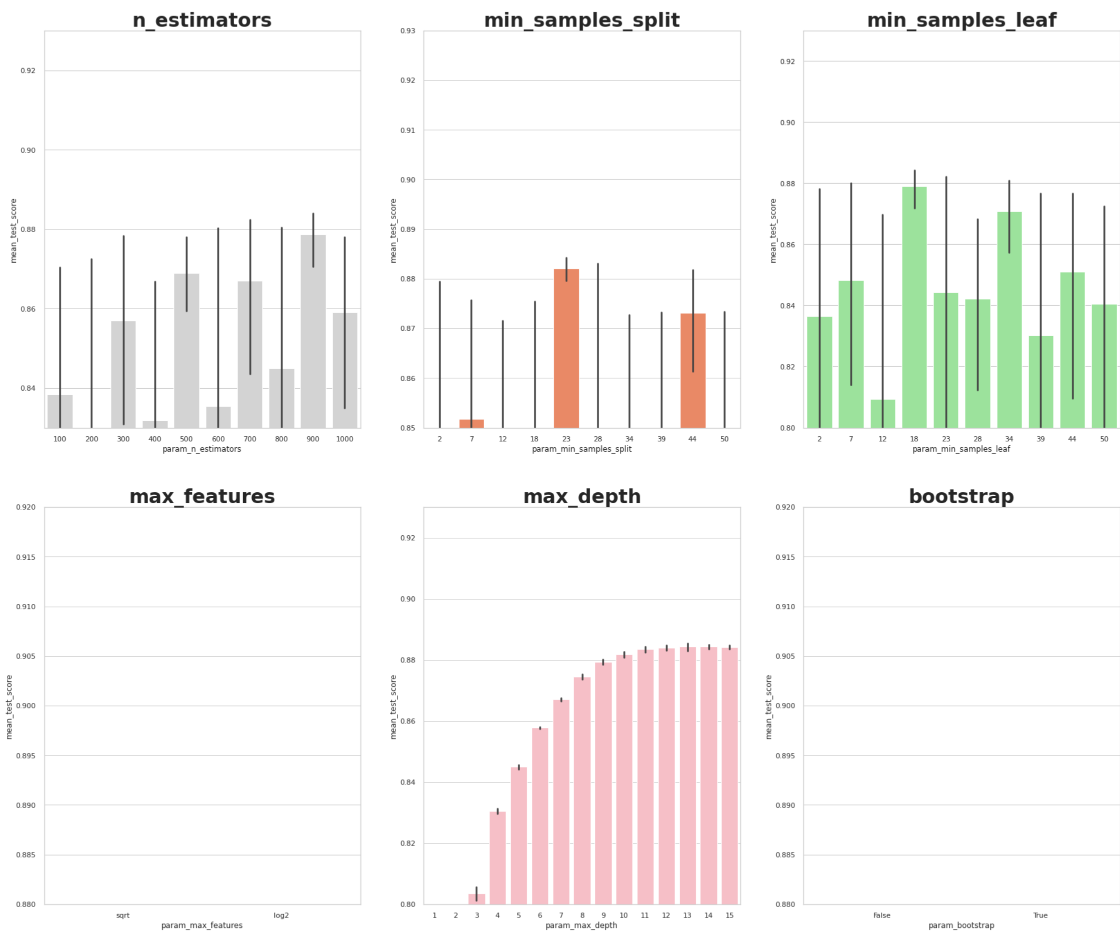
With n_iter = 100 and cv = 3, we created 300 Random Forest models, randomly sampling combinations of the hyperparameters input above. We can call "best_params_" to get the best performing model's parameters. However, "best_params_" at this stage may not give us the best insight to get a range of parameters to try for the next round of hyperparameter tuning. To get a good range of values to try next, we can easily get a dataframe of our RandomSearchCV results

[438]:
```
```

[438]:
```
  param_n_estimators param_min_samples_split param_min_samples_leaf  \
0                500                      12                     18
1                700                      12                      2
2                400                      23                      2
3                300                      28                      7
4                700                       2                      2
5                600                      50                     23
6                800                      28                     12
7                600                      23                      2
8                300                       2                     23
9                300                      18                     18

  param_max_features param_max_depth param_bootstrap  mean_test_score  \
0               sqrt              13           False         0.886116
1               sqrt              13           False         0.886053
2               sqrt              14           False         0.885593
3               log2              13            True         0.885551
4               log2              11            True         0.885530
5               sqrt              14           False         0.885321
6               sqrt              14            True         0.885154
7               sqrt              15           False         0.885091
8               log2              14           False         0.885070
```

```
9              sqrt              13              True           0.885050

   rank_test_score
0                1
1                2
2                3
3                4
4                5
5                6
6                7
7                8
8                9
9               10
```

[440]:



Looking at the plots above, we can extract insights about how well each value for each hyperparameter performed on average.

n_estimators: 300, 500, 700, 900 seem to have the highest average scores.

min_samples_split: There are high scores at 23and 44

min_samples_leaf: We can try values between 2,7,18,34.

max_features: “sqrt” has the highest average score.

max_depth: values of 8-15 seem to do well.

bootstrap: “False” has the highest average score.

So now we can take these insights and move into the second round of hyperparameter tuning to further narrow our selections.

Hyperparameter Tuning Round 2: GridSearchCV

[441]:

```
Fitting 3 folds for each of 144 candidates, totalling 432 fits
CPU times: user 2min 10s, sys: 587 ms, total: 2min 10s
Wall time: 3h 38min 32s
```

[441]:
```
{'max_depth': 13,
 'min_samples_leaf': 2,
 'min_samples_split': 23,
 'n_estimators': 900}
```

1.5 Evaluate Performance Of Models On Test Data

Now, we can evaluate each of the models that we have made on our test data. Remember that we are testing 3 models:

```
1. Baseline Random Forest
2. Baseline Random Forest With PCA Reduced Dimensionality
3. Baseline Random Forest With PCA Reduced Dimensionality & Hyperparameter Tuning
```

[469]:
```
y_pred = rfc.predict(x_test_scaled_df)
y_pred_pca = rfc.predict(X_test_scaled_pca)
y_pred_gs = gs.best_estimator_.predict(X_test_scaled_pca)
```

[486]:

	predicted 0	predicted 1	predicted 2
actual 0	8226	1303	1007
actual 1	1966	610	3840
actual 2	5044	855	711

```
'Baseline Random Forest recall score'
```

```
0.4051863169510228
```

	predicted 0	predicted 1	predicted 2
actual 0	8724	1040	772

```
actual 1          643         5773            0
actual 2          395           12         6203
```

```
'Baseline Random Forest With PCA recall score'
```

```
0.878533231474408
```

```
          predicted 0  predicted 1  predicted 2
actual 0         8644         1000          892
actual 1          541         5874            1
actual 2          204           29         6377
```

```
'Hyperparameter Tuned Random Forest With PCA Reduced Dimensionality recall score'
```

```
0.8868092691622104
```

From this our result, our best perfoming model is our hyper parameter tund Random Forest model

[490]:
```
evaluate_model_performance(gs.best_estimator_, X_train_scaled_pca, y_train)
```

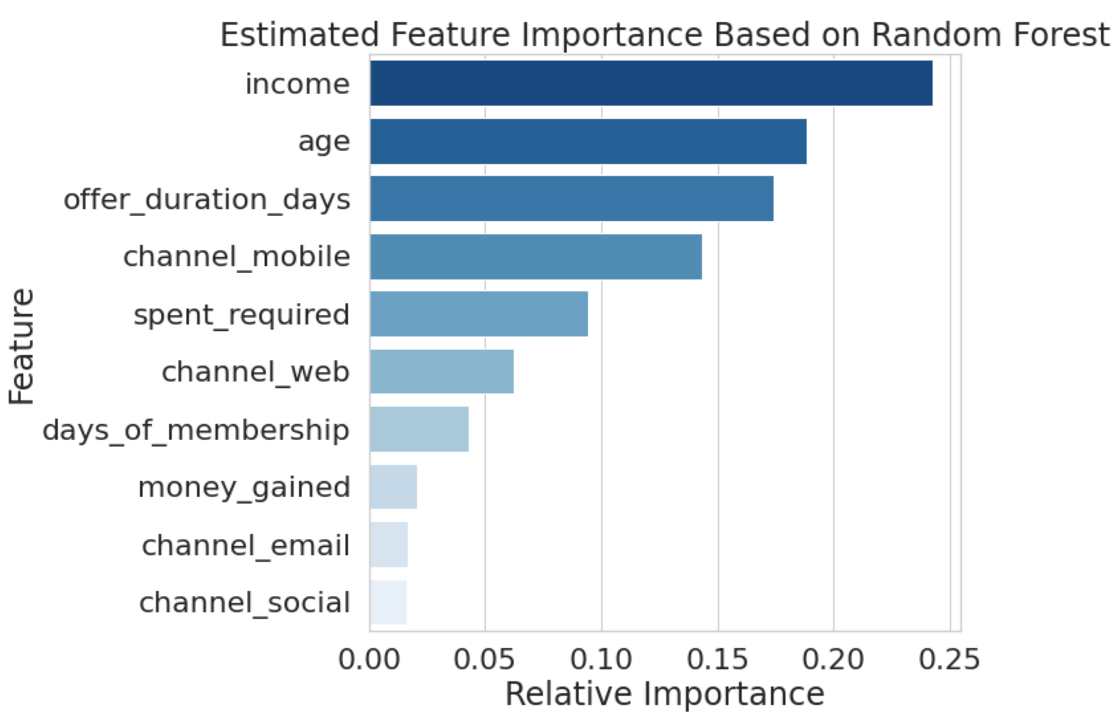
```
RandomForestClassifier model accuracy: 0.916
RandomForestClassifier model f1-score: 0.915
```

[490]:
```
(0.9159454826706802, 0.9153190471092886)
```

1.5.1 Feature importance

[491]:
```

```

[491]:
```
Text(0.5, 1.0, 'Estimated Feature Importance Based on Random Forest')
```



[456]:
```python
feature_importance.head(n=10)
```

[456]:

	feature	relativeimportance
0	income	0.242512
1	age	0.188434
2	offer_duration_days	0.174120
3	channel_mobile	0.143021
4	spent_required	0.093982
5	channel_web	0.062092
6	days_of_membership	0.042789
7	money_gained	0.020546
8	channel_email	0.016501
9	channel_social	0.016003

According to our model, the 10 most important features that determine whether a customer will ignore, make an order or not make an order are: * Income which contributes the most- 24% * Age which contributes-18% * Duration of offer-17% * Offer sent on mobile-14% * The spend required to complete the offer-9% * If offer was sent on web-6% * How long a customer has been a member-4% * The reward to be gained-2% * Offer sent via email-1.6% * Offer sent via social media-1.6%

> Which means demographics contribute 42% of the decision. The channel used to communicate the offer contributes 23.2%. The details of the offer contribute 11%. How long the customer has been a member contributes 4%.

Print the best model's hyperparameters

[492]:
```python
print(gs.best_estimator_)
```

```
RandomForestClassifier(max_depth=13, min_samples_leaf=2, min_samples_split=23,
                       n_estimators=900)
```

1.6 Results

1.6.1 Test on new data

[297]:
```python
frames['Prediction'] = frames['Prediction'].map({0: 'no view', 1: 'no order', 2:
 ↪ 'possible order'})
```

[308]:
```python
frames.Prediction.value_counts()
```

[308]:
```
no view           12565
possible order     6387
no order           4610
Name: Prediction, dtype: int64
```

[324]:
```python
frames.head()
```

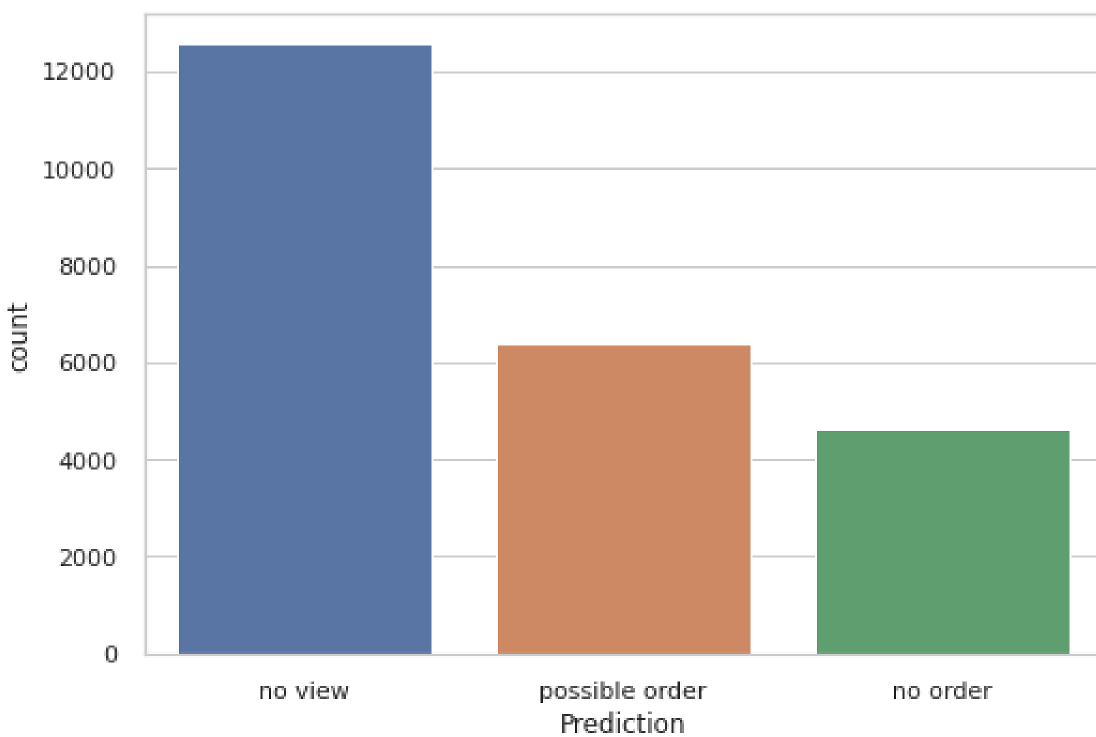
[324]:

```
      income  spent_required  money_gained       age  channel_social  \
0  0.480289        0.308519     -0.570975  0.836877        0.872210
1 -0.303513       -1.582487     -0.570975 -0.785519       -1.026961
2 -0.026877       -0.258783     -0.570975 -0.576696        0.872210
3  0.618607        0.308519      2.948957 -0.062669       -0.077375
4 -0.441831        2.199524      1.188991 -0.383936       -1.026961

   channel_email  channel_mobile  channel_web  offer_duration_days  \
0       0.326346        0.421475    -1.453699             0.498934
1      -0.700134       -0.493049    -0.360391            -1.112714
2       0.326346        0.421475     0.732917             0.498934
3      -0.700134       -0.493049    -1.453699            -0.629219
4      -0.700134       -1.407573    -0.360391            -0.145725

   days_of_membership Prediction
0           -0.250415    no view
1           -0.571356    no view
2            1.569288    no view
3           -0.218217    no view
4           -0.563047    no view
```

[325]:

```python
sns.countplot(frames['Prediction'])
```

[325]: <AxesSubplot:xlabel='Prediction', ylabel='count'>



1.7 Summary of Key Findings

Starbucks Customer Demographics:

58.6 % of customers are male, 39.8% are female and 1.4% prefer not to specify gender. While men are the largest population, females are the highest earners and spenders. Most men are in the low_to_mid and mid income bracket.

There's an almost equal distribution of male and females among the mid to high income bracket. Middle income and low_to_mid income earners occupy a huge proportion of the population, with mid income earners being the dorminant. Low earners are fewer, they are the least.

Starbucks Customer spending behavior:

Females are the highest spenders, they spend on average 0.23USD per day followed by gender O who spend 0.21USD per day and least spenders are males who spent on average 0.16USD.

The elderly, that is those above 60 are the highest age group spending an average of 0.21USD per day. Adults, those aged 35-60 are the second highest spenders on average, spending 0.19USD per day. Ages 17-34 are the least spenders, spending roughly 0.14USD per day

As expected, high income earners have the highest average spend. Spending on average, 0.28USD per day, mid to high earners also spend highly with 0.23USD per day. Low and low to mid income earners spend right around the same amount. They spend about 0.11US per day

New members have the highest average daily spend followed by regular members. Loyal members, those who have been members for over 3 years are no longer daily spenders

Summary of average daily spending has revealed that: * High earning elderly females who are new members have the highest daily average spend i.e Female members who are over 60 years earning over 90K and have been members for at least 1200 days

Starbucks Offers: * 41 395 new customers received offers. Of those that received offers, only 41.5% viewed the offer and did not complete it and 37.5% completed the offer.

- 40 539 new customers made transactions on the app, new members made the highest number of transactions of all the members. Regular members also made a sizeable amount of transactions on the app, they made 30 036 transactions. Loyal members had the least number of transactions, making 12 299 transactions
- Regular members were the second highest to receive offers, receiving 19 223 offers but only 24% viewing the offers and 60% completing the offers. Most regular members completed the offers without viewing them.
- 5785 loyal customers received offers, of this number, 38.2% viewed the offers and 41% completed the offers.

Perfomance of Offers on different members: * On new customers:

```
* 39.88% of new customers received BOGO offer, 41.4% of those that received the BOGO viewed it

* 40% of new customers received the discount offer, 27.5% of those that received this offer vi
```

```
* 20% of new customers received informational offers, 69.5% of them viewed it, which accounts
```

- On regular customers:
  - 39.77% of regular customers received the BOGO offer, of these only 18% viewed it. This means that 29.4% of all views of BOGO offer were done by regular customers. 74% of regular customers who received the BOGO offer completed it. This accounts for 48.6% of all offers completed by regular customers
  - 40.3% of regular customers received the discount offer, of these only 7% of them viewed it. This means that 12.1% of offer views by regular customers were viewing discount offer. 77.1% of regular customers who received discount offer completed it. This means 51.4% of all offers completed by regular customers was on discount offers.
  - 19.91% of regular customers received the informational offer, of these 71.6% of them viewed it. 23.5% of all views made by regular customers were on informational offers.
- On loyal customers:
  - 40.3% of loyal customers received the BOGO offer, of these 46% of them viewed it. This means that, 48.85% of all loyal customer views were on BOGO. 40.5% of loyal customers who received BOGO completed it. This accounts for 39.86% of all offers completed by loyal customers.
  - 39.6% of loyal customers received discount offer, 13.2% viewed it. This means that 13.7% of all views by loyal customers was on discount offer.62.28% of loyal customers who received the discount offer completed it. This accounts for 60% of all offers completed by loyal customers.
  - The bogo_3 is the most succesful offer with 29% success. The discount offers perfomed pretty well averaging 27%.

Pefomance of Offers on different Genders: * Female BOGO offer:

```
* 10 975 females received the BOGO offers, 1368 viewed it but did not complete it. 7566 receiv
```

- Female Discount offer:
  - 10 943 females received the Discount offers, 1267 viewed but did not complete. 6369 received and completed it.1360 of them auto completed it.
- Female Informational offer:
  - 5538 females received the informational offer, 1242 viewed but did not complete the offer. 2668 received and completed the offer.
- Male BOGO offer:
  - 15 208 males received the BOGO offer. Of these 2534 viewed the offer but did not complete it. 9785 received and completed the offer. 963 completed the offer automatically.
- Male Discount offer:
  - 15 354 males received the discount offer. Of these 2201 males only viewed the offer. 8049 received and completed the offer. 1271 auto completed the offer
- Male Informational offer:

  - 7567 males received the informational offer. 1480 males viewed the offer without completing it. 3809 received and completed it

- Gender unspecified BOGO offer:
  - 354 of these customers received the offer. 56 only viewed the offer. 253 received and completed the offer while 20 auto completed it.
  - Mid income earners received and completed the most offers and made the most transactions.
  - Low to mid income earners were the second highest in terms of offers received and completed.

Customer Clusters There are 4 types clusters/Segments of customers. * **Cluster 0**: * Clusters 0 has mostly mid to high and high earners and has mostly elderly and young adults. This cluster liked and responded to all three offers though they spent the least of all 4 clusters. Cluster 0 had a few outliers but most customers in that cluster did not complete offers, a large number of them made casual purchases and or ignored the offers. Cluster O also like all 3 offers with Informational being the most popular,followed by discount and last BOGO offers

- **Cluster 1**:
  - This cluster has mostly mid to high income earners. Cluster 1 liked and spent the most on informational offers spending between 2.5 and 10 USD on them. They spent about the same on BOGO and Discount spending on average between 1 to 7.5 USD. In general, this cluster spent the least amount on all offers.
- **Cluster 2**:
  - Cluster 2 made the highest transactions. Cluster 2 has a normal distribution from between low to mid to high income earners with most members in the higher income bracket. Cluster 2 has mostly new and regular members. The prevalent ages being adults and teenagers with an even distribution among males and females. Informational offers were not popular among cluster 2, they did not spent on it. Only a few outliers completed this offer. They spent quite a lot on discount and BOGO offers spending an average of 10USD on both which was the second highest spend on offers among the clusters.
- **Cluster 3**:
  - Cluster 3 is mostly skewed towards high income earners. Cluster 3 has the highest average money spent for both transactions and completed offers. Followed closely by cluster 2. In terms of member type it is evenly distributed between new and regular members with very few loyal members, same even distribution among males and females and very very few other genders. This cluster is filled with adults and the elderly which is consistent with what we have been finding. Cluster 3 customers spent a lot on Informational offers, more than 50% of them spend above average (10USD) on it. They also spend the highest on average on Discount offers, most of them spending between 7.5 to 12.5USD. They also spent a lot on BOGO offers though not the most popular offer among this cluster. They spent a minimum of 5USD which is higher than the minimum spend for all other clusters. Most of them spent an average of 9USD on BOGO offers.

1.7.1 Findings from the model

According to our model, the 10 most important features that determine whether a customer will ignore, make an order or not make an order are: * Income which contributes the most- 24% * Age which contributes-18% * Duration of offer-17% * Offer sent on mobile-14% * The spend required to complete the offer-9% * If offer was sent on web-6% * How long a customer has been a member-4% * The reward to be gained-2% * Offer sent via email-1.6% * Offer sent via social media-1.6%

> Which means demographics contribute 42% of the decision. The channel used to communicate the offer contributes 23.2%. The details of the offer contribute 11%. How long the customer has been a member contributes 4%.

1.8 Recommendations

Offers: BOGO and Discount offers perfomed very well accross all demographics and had a lot of money spent on them. Starbucks should keep rolling these out. Informational offers did not perform very well and are also difficult to monitor.There is also no way to track whether informational offers are redeemed (can only see if user received or opened informational offer). We have to assume they were influenced by the informational offer if they viewed it. Offer duration is also very significant factor in whether customers respond to offers, it contributes 17% to the decision of completing and not completing an offer. More money was spent for offers that lasted longer, on average 2USD more was spent on offers that lasted more than 5 days and more. The spent required is also a good measure, generally customers will spend at most under 7 dollars for spent required above 7.

Target Groups: Females on average spend the most on all offers. Adult and elderly women who are mid to high earners are the most profitable group. They respond to offers and spend a lot on them.

In general, the higher the income the more customers spend on offers.

Adults and the elderly on average spend more on offers than all other groups.

42% of the decision to complete an offer or not is influenced by demographics i.e age and income.

Clusters 2 and 3 should be targeted more because they are high spenders, they are newer members, adults and have good income.

Channel of sending offers: Channels sent via mobile received the highest spend and were the most completed. Most people who received offer via mobile completed it. It seems that mobile is the most important channel. According to the model, receiving the offer via mobile contributes 14% to the response on the offer. Probably because the rewards app is a mobile app. Other media do contribute like the web (6%), other channels contribute 1.6% each.

Model: Starbucks should use the random forest classifier model, it can handle large amount of training data and is good for the multiclassification problem. By tuning it using random search and grid search, its best parameters can be found.