Udacity Starbucks Capstone Project

George Helsby

Project Overview:

The project involves analysing a large dataset from Starbucks that simulates how people make purchasing decisions and how those decisions are influenced by promotional offers.

The goal is to **find the hidden traits that influence the purchasing patterns** of certain individuals in the dataset. People produce various events, including receiving offers, opening offers, and making purchases.

There are no explicit products to track. Only the amounts of each transaction or offer are recorded.

There are three types of offers that can be sent:

1. BOGO, buy-one-get-one: user needs to spend a certain amount to get a reward equal to that threshold amount.
2. Discount: a user gains a reward equal to a fraction of the amount spent.
3. Informational: there is no reward.

The basic task is to use the data to **identify which groups of people are most responsive to each type of offer, and how best to present each type of offer.**

Explanation of Data:

profile.json

Rewards program users (17,000 users x 5 fields)

- gender: (categorical) M, F, O, or null
- age: (numeric) missing value encoded as 118
- id: (string/hash)
- became_member_on: (date) format YYYYMMDD
- income: (numeric)

portfolio.json

Offers sent during 30-day test period (10 offers x 6 fields)

- reward: (numeric) money awarded for the amount spent
- channels: (list) web, email, mobile, social
- difficulty: (numeric) money required to be spent to receive reward
- duration: (numeric) time for offer to be open, in days
- offer_type: (string) bogo, discount, informational
- id: (string/hash)

transcript.json

Event log (306,648 events x 4 fields)

- person: (string/hash)
- event: (string) offer received, offer viewed, transaction, offer completed
- value: (dictionary) different values depending on event type
  - offer id: (string/hash) not associated with any "transaction"
  - amount: (numeric) money spent in "transaction"
  - reward: (numeric) money gained from "offer completed"
- time: (numeric) hours after start of test

Problem Statement:

Evaluation Metric:

Data Assessing and Cleaning

Portfolio Assessing:

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

The portfolio dataframe is simply the 10 different types of offer that were sent to users during this simulated 30-day trial.

Portfolio Cleaning

- Renamed id column to offer_id.
- Converted channels and offer_type columns to one-hot encoded dummy columns.
- Changed duration column title to duration_days.
- Changed difficulty column title to min_spend.

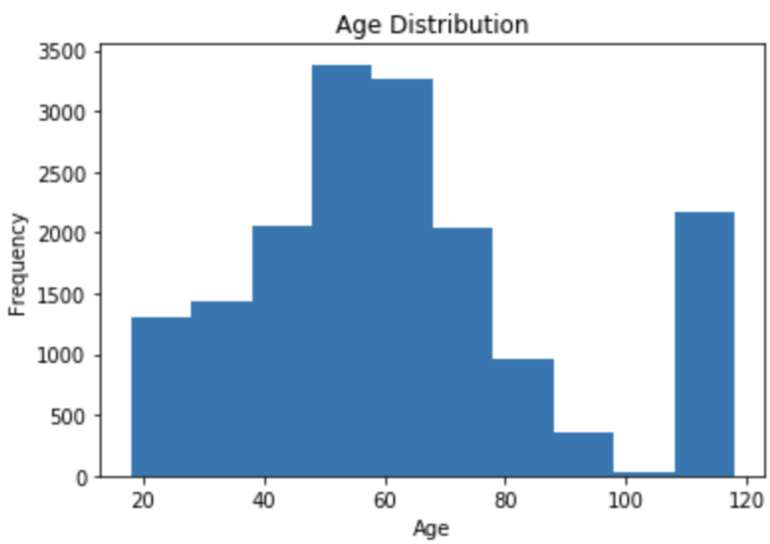
	reward	min_spend	duration_days	offer_type	offer_id	email	mobile	social	web	bogo	discount	informational
0	10	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd	1	1	1	0	1	0	0
1	10	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0	1	1	1	1	1	0	0
2	0	0	4	informational	3f207df678b143eea3cee63160fa8bed	1	1	0	1	0	0	1
3	5	5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

Profile Assessing:

	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

The profile data frame contains all of the information from the users in the trial.

Missing Data & Age 118



There were an unusual number of users with the age 118.

After further investigation **these same users also had null gender and income values**.

Therefore they were removed as they make up a negligible proportion of the set and offer no predictive ability.

Profile Cleaning

- Removed all null rows as null incomes, null genders and age being 118 are found in the same rows.
- Changed became_member_on column from string to datetime.
- Changed id to user_id.
- Create age group bins and dummy variables on those bins for later classification analysis.
- Created dummy gender categorical columns.
- Created year_joined column and year joined dummy categorical columns.

	gender	age	user_id	became_member_on	income	20s	30s	40s	50s	60s	...	female	male	other	year_joined	2013
1	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	0	2017	0
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	2017-05-09	100000.0	0	0	0	0	0	...	1	0	0	2017	0
5	M	68	e2127556f4f64592b11af22de27a7932	2018-04-26	70000.0	0	0	0	0	1	...	0	1	0	2018	0
8	M	65	389bc3fa690240e798340f5a15918d5c	2018-02-09	53000.0	0	0	0	0	1	...	0	1	0	2018	0
12	M	58	2eeac8d8feae4a8cad5a6af0499a211d	2017-11-11	51000.0	0	0	0	1	0	...	0	1	0	2017	0
13	F	61	aa4862eba776480b8bb9c68455b8c2e1	2017-09-11	57000.0	0	0	0	0	1	...	1	0	0	2017	0
14	M	26	e12aeaf2d47d42479ea1c4ac3d8286c6	2014-02-13	46000.0	1	0	0	0	0	...	0	1	0	2014	0
15	F	62	31dda685af34476cad5bc968bdb01c53	2016-02-11	71000.0	0	0	0	0	1	...	1	0	0	2016	0
16	M	49	62cf5e10845442329191fc246e7bcea3	2014-11-13	52000.0	0	0	1	0	0	...	0	1	0	2014	0
18	M	57	6445de3b47274c759400cd68131d91b4	2017-12-31	42000.0	0	0	0	1	0	...	0	1	0	2017	0

Transcripts Assessing:

	person	event	value	time
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0
1	a03223e636434f42ac4c3df47e8bac43	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
2	e2127556f4f64592b11af22de27a7932	offer received	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0
4	68617ca6246f4fbc85e91a2a49552598	offer received	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0
5	389bc3fa690240e798340f5a15918d5c	offer received	{'offer id': 'f19421c1d4aa40978ebb69ca19b0e20d'}	0
6	c4863c7985cf408faee930f111475da3	offer received	{'offer id': '2298d6c36e964ae4a3e7e9706d1fb8c2'}	0
7	2eeac8d8feae4a8cad5a6af0499a211d	offer received	{'offer id': '3f207df678b143eea3cee63160fa8bed'}	0
8	aa4862eba776480b8bb9c68455b8c2e1	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0
9	31dda685af34476cad5bc968bdb01c53	offer received	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0

The transcripts dataframe contains all of the individual transactions made by all users through the 30 day trial. The transaction event can be either:

- Offer received
- Offer viewed
- Offer completed
- Transaction

Transcripts Cleaning:

- Changed person column to user_id to match with profile dataframe.
- Removed whitespace from the event column.

- Converted time column name to time_hours
- Converted the event column with one-hot encoding.
- Created offer_id, amount and reward column with a for loop from the value column.
- Then removed the value column.

	user_id	event	time_hours	offer_completed	offer_received	offer_viewed	transaction	offer_ic
0	78afa995795e4d85b5d9ceeca43f5fef	offer_received	0	0	1	0	0	9b98b8c7a33c4b65b9aebfe6a799e6d9
1	a03223e636434f42ac4c3df47e8bac43	offer_received	0	0	1	0	0	0b1e1539f2cc45b7b9fa7c272da2e1d7
2	e2127556f4f64592b11af22de27a7932	offer_received	0	0	1	0	0	2906b810c7d4411798c6938adc9daaa5
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer_received	0	0	1	0	0	fafdcd668e3743c1bb461111dcafc2a4
4	68617ca6246f4fbc85e91a2a49552598	offer_received	0	0	1	0	0	4d5c57ea9a6940dd891ad53e9dbe8da0
5	389bc3fa690240e798340f5a15918d5c	offer_received	0	0	1	0	0	f19421c1d4aa40978ebb69ca19b0e20c
6	c4863c7985cf408faee930f111475da3	offer_received	0	0	1	0	0	2298d6c36e964ae4a3e7e9706d1fb8c2
7	2eeac8d8feae4a8cad5a6af0499a211d	offer_received	0	0	1	0	0	3f207df678b143eea3cee63160fa8bec
8	aa4862eba776480b8bb9c68455b8c2e1	offer_received	0	0	1	0	0	0b1e1539f2cc45b7b9fa7c272da2e1d7
9	31dda685af34476cad5bc968bdb01c53	offer_received	0	0	1	0	0	0b1e1539f2cc45b7b9fa7c272da2e1d7

amount	reward
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None
NaN	None

Data Cleaning - Merging

All 3 dataframes can now be merged together.

Firstly the user profile dataframe with the transactional dataframe.

	gender	age	user_id	became_member_on	income	20s	30s	40s	50s	60s	...	2017	2018	event	time_hours	offer_
0	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	transaction	18	
1	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	transaction	144	
2	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	offer_received	408	
3	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	offer_received	504	
4	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	1	0	transaction	528	

5 rows × 33 columns

Then the 'portfolio offer' dataframe with the above new merged user-transaction set.

	gender	age	user_id	became_member_on	income	20s	30s	40s	50s	60s	...	min_spend	duration_days	offer_type	emai
0	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	NaN	NaN	NaN	Na
1	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	NaN	NaN	NaN	Na
2	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	5.0	7.0	bogo	1.
3	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	0.0	4.0	informational	1.
4	F	55	0610b486422d4921ae7d2bf64640c50b	2017-07-15	112000.0	0	0	0	1	0	...	NaN	NaN	NaN	Na

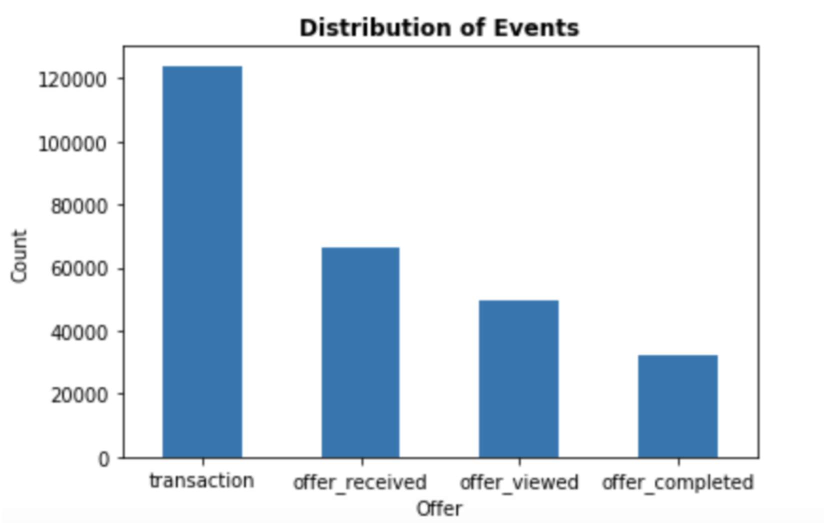
5 rows × 44 columns

I then also created new offer_name and event_id columns by mapping dictionaries to the existing respective offer_id and event columns.

The new clean merged dataset was hence ready to be saved to_csv in order to store and save time in later stages.

Exploratory Data Analysis

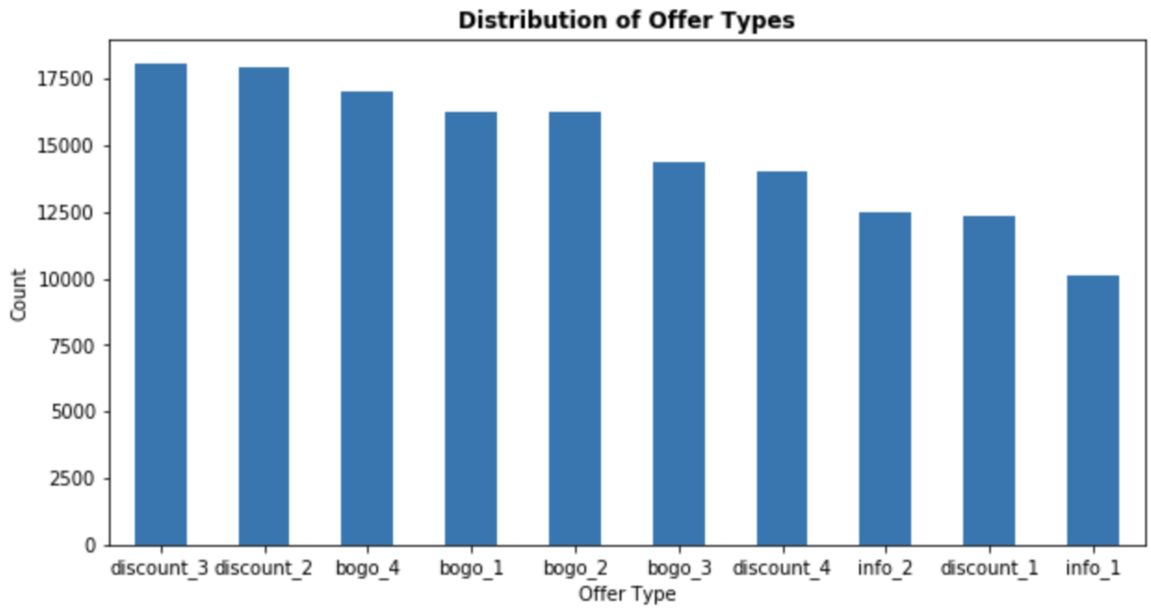
Firstly I observed the distribution of events in the simulation.



Clearly transactions were the most common event.

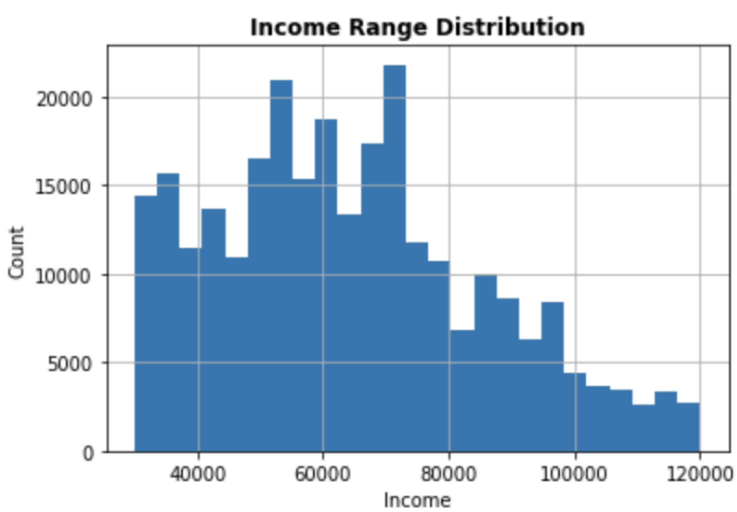
An expected dropoff from users receiving to viewing to completing an offer also observed.

Secondly I looked into the distribution of offer types sent out to users.



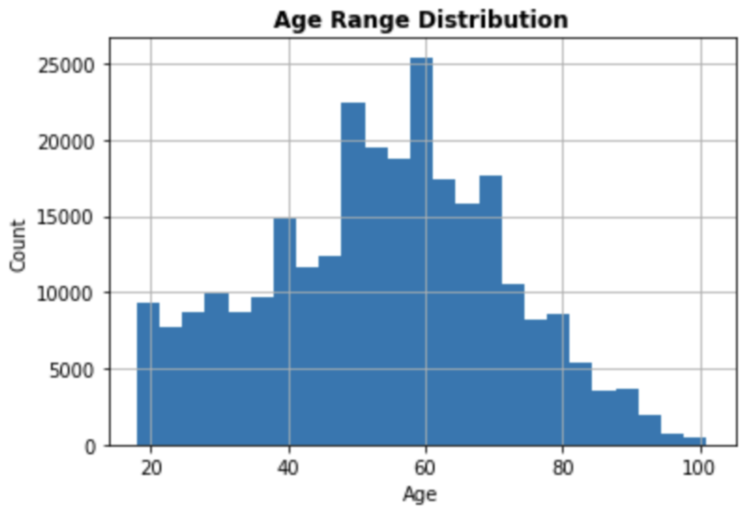
While not completely random, the distribution of offers is regular enough to enable actionable analysis.

User Characteristics
I then observed the distributions of several user characteristics.



Income
Income distribution in the data follows an expected right skew with the mean being higher than the median income.

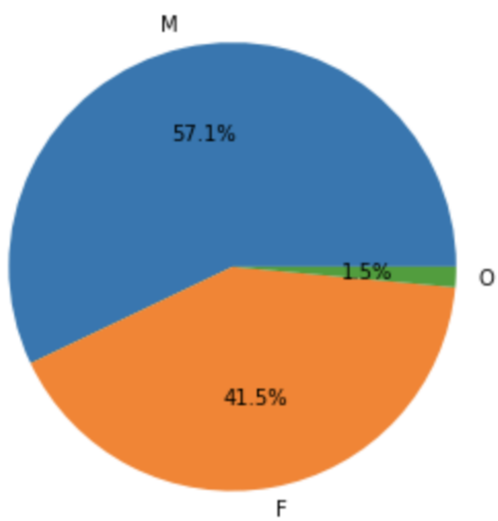
```
mean     64337.000755
std      21243.762941
min      30000.000000
25%      48000.000000
50%      62000.000000
75%      78000.000000
max     120000.000000
```



Age
Age follows a bell curve distribution. There is a slight left skew however.

This is likely due to younger people being more likely to sign up for rewards programs on the mobile app.

mean	53.840696
std	17.551337
min	18.000000
25%	41.000000
50%	55.000000
75%	66.000000
max	101.000000



Gender
The breakdown of gender in the trial is skewed towards more men at 57%.

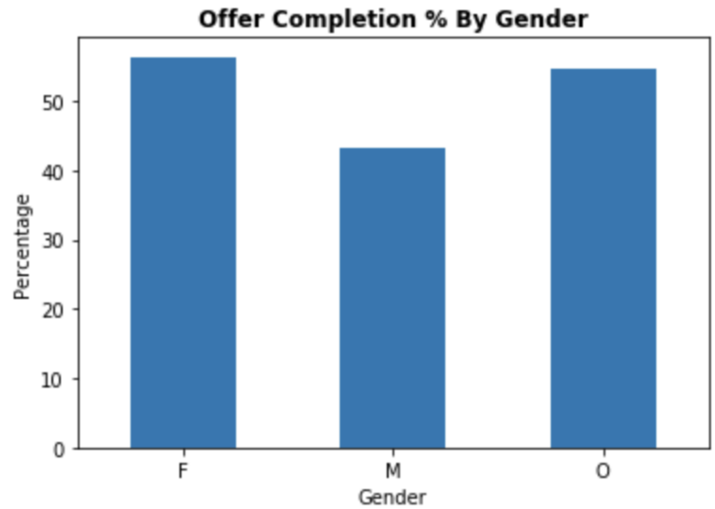
Hopefully this is representative of Starbucks' customer base as a whole.

If not, we would like to see this be adjusted accordingly in future trials.

Rewards Member Sign Up Year
This indicates that the vast majority of members have signed up in the last 3 years.

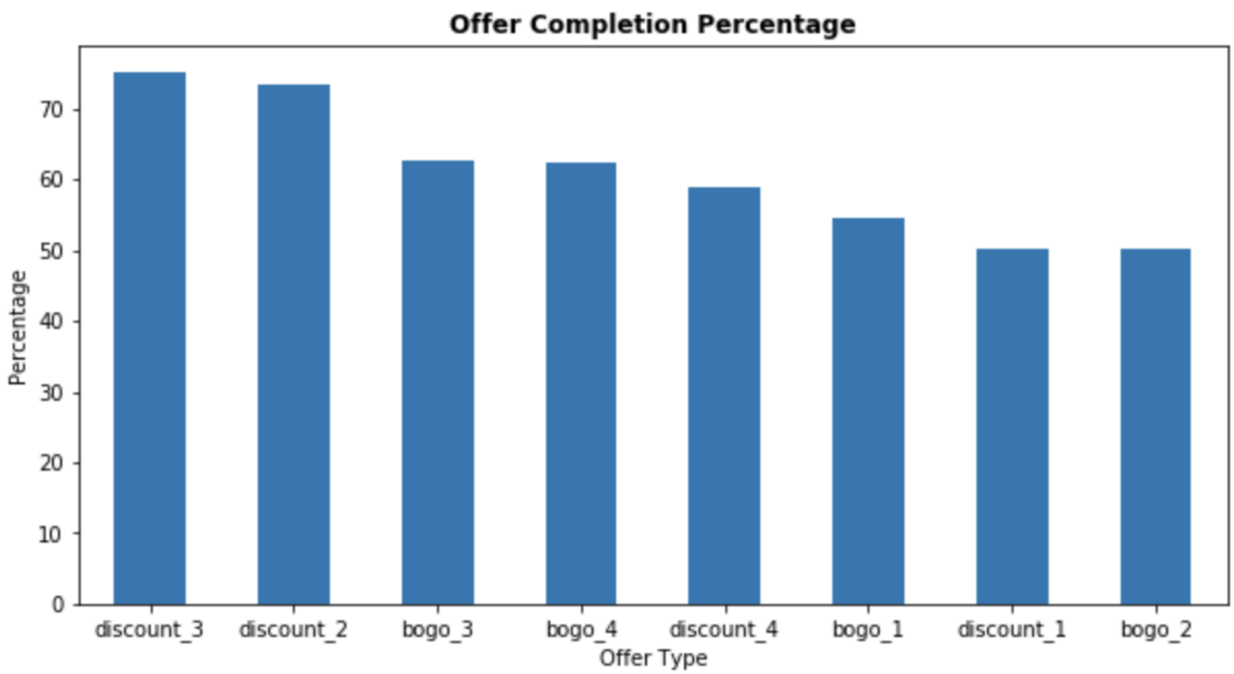
Would be interesting to observe the difference in outcomes between long-time members and brand new members.

Multi-Factor Relationship Exploration



Gender - Offer Completion
Firstly I looked at the relationship between offer completion percentage and gender.

Females are 13% more likely than men to complete offers in general. Other categories are similar to female %.



Offer Type - Promotion Effectiveness
Then I observed the effectiveness of each sub group of offers.

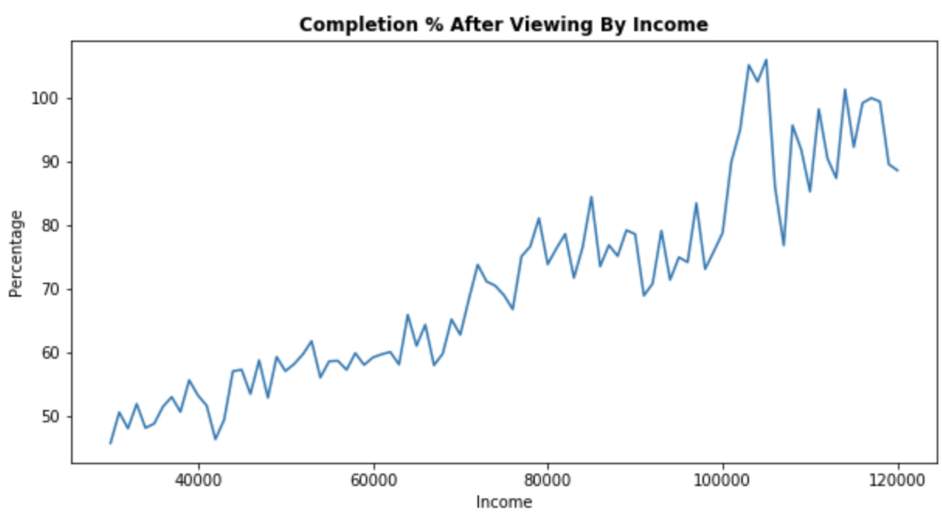
Clearly informational offers do not have the chance to be completed.

The differential between viewing and completing is much smaller for discounts.

Income - Promotion Effectiveness

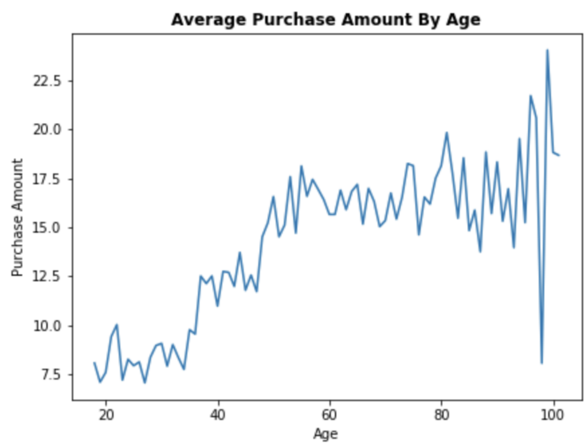
This is a clear positive relationship between income and the effectiveness of promotions.

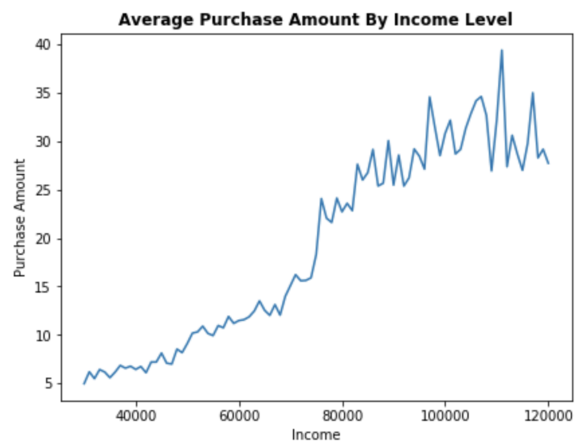
Paradoxically, the highest income earners in the sample are also the most likely to complete offers even after having viewed them.



Some of the highest income earners in the trial are actually completing more offers than they are viewing. This means that they have completed offers unknowingly without seeing them.

Age/Income - Purchase Amount





We can clearly observe here that there is a positive relationship between age/income & average purchase amount.

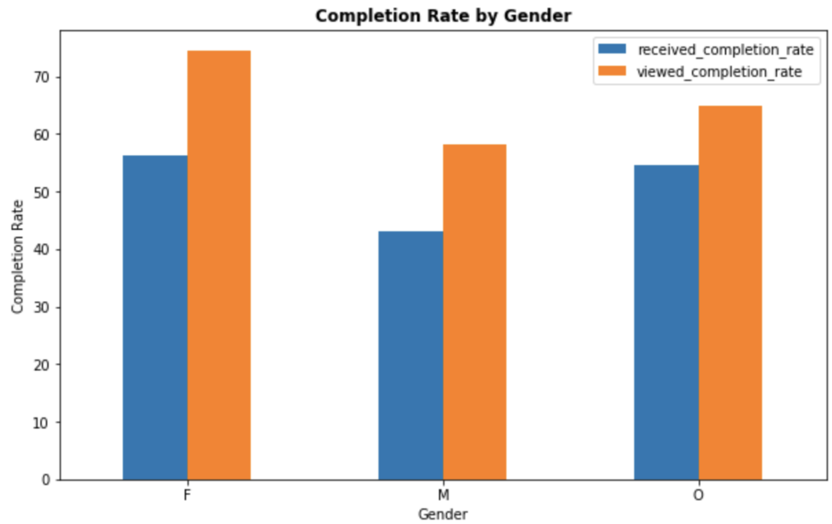
In particular there is a clear spike around an income of $75,000 where average purchase amount jumps significantly.

Membership Sign Up Year - Promotional Effectiveness

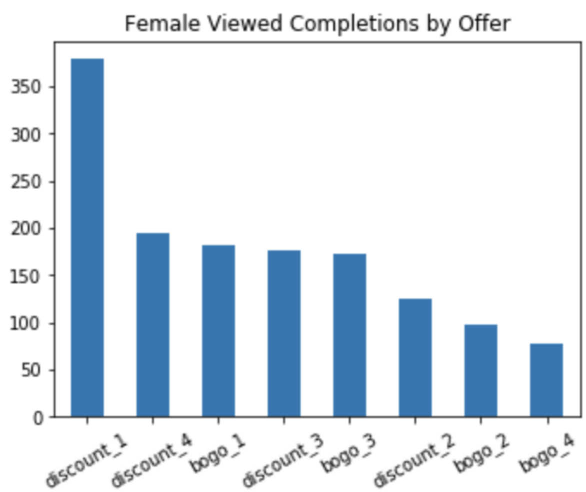
Strangely, members who signed up in 2016 are roughly 35% more likely to complete offers than those who signed up in 2018.

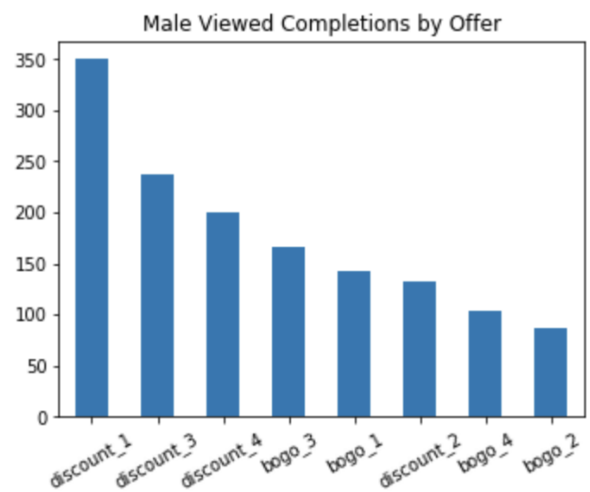
Completion Rates - Gender

Men, women and others tend to all increase the chance of completing an offer after viewing it by similar proportions.

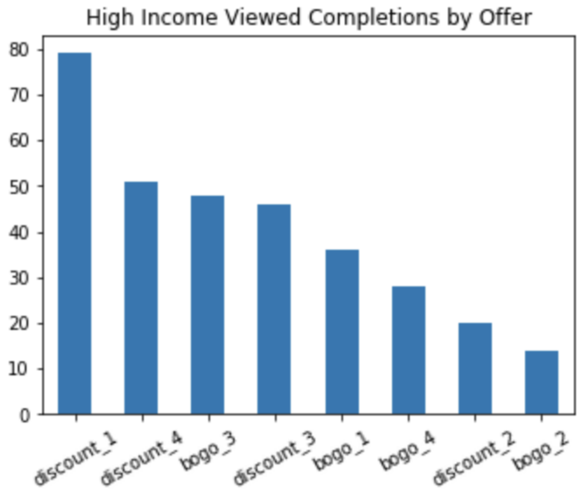


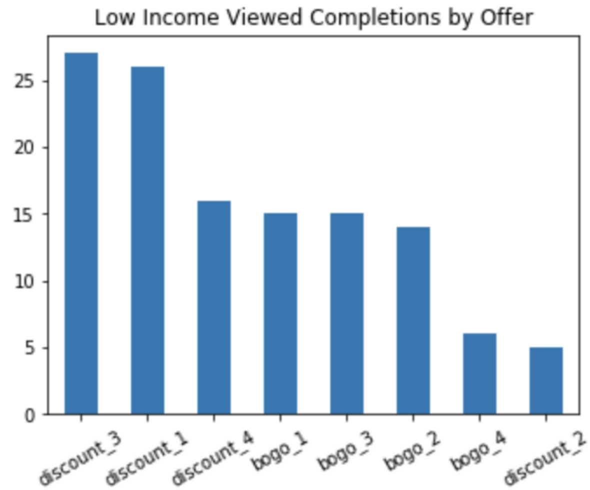
Gender and Offer Type



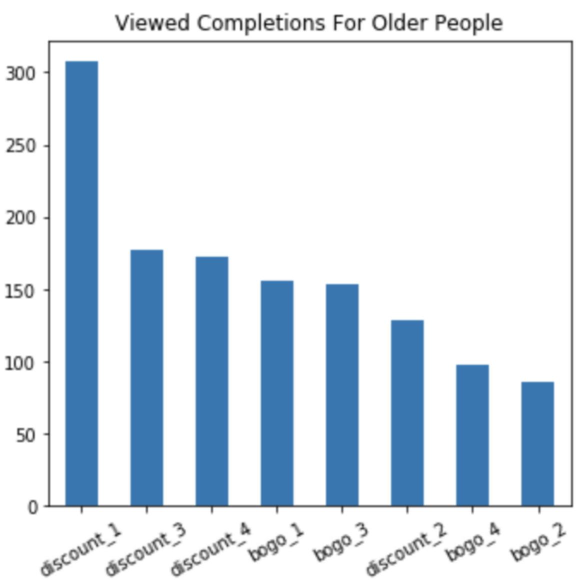


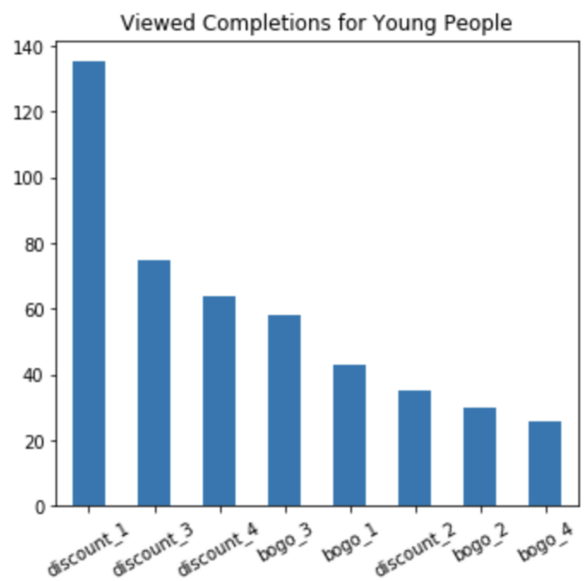
Income and Offer Type





Age and Offer Type





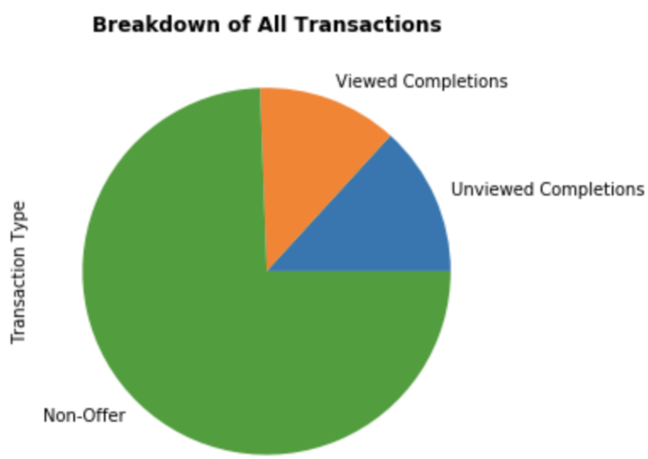
General Heuristics from Exploration

1. Discount offers are more effective than bogo offers
2. Females are slightly more receptive to offers than males & others
3. Higher income earners are more receptive to offers
4. Bogo 4 offer is poor for low income groups
5. Discount 1 is the best for anyone, especially mid-to-high-income groups
6. High income, older users more likely to complete unviewed offers

7. Users who signed up in 2013, 2014 and 2018 are less likely to complete offers.

Distinct Transaction Categories

1. Non-offer related transactions
2. Offer completion transactions
3. Unviewed offer completion transactions

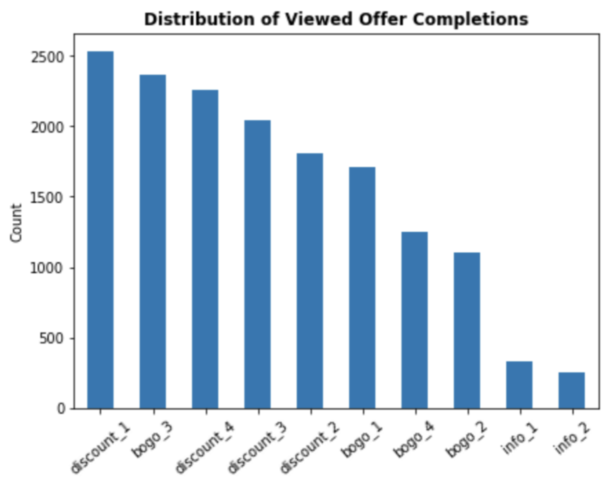


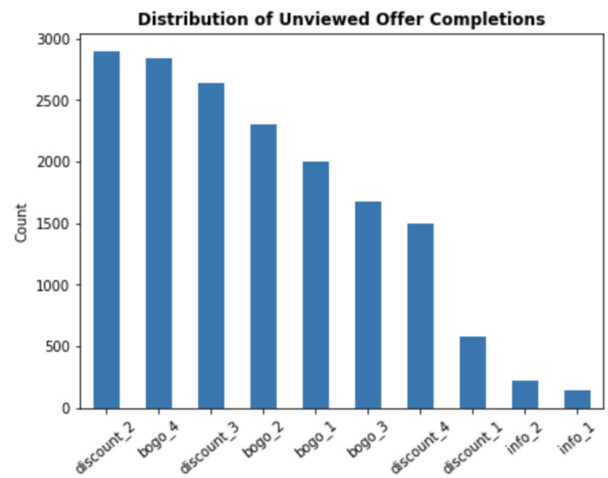
	non_offer	viewed_offer	unviewed
age	51.778755	54.404806	57.151098
income	59543.131126	66001.789480	72625.052093
female	0.374114	0.453314	0.499137
male	0.611889	0.534479	0.482408
other	0.013997	0.012207	0.018456
amount	11.958272	18.341809	22.139314

Key Transaction Group Takeaways

1. There are roughly the same amount of offer completion transactions as there are unviewed offer completions.
2. Unviewed completions are more likely to be from higher earners.
3. 40% of unviewed completions are coming from the top 25% of income earners.
4. Average transaction amount is significantly higher for unviewed completions.
5. Unviewed completions are equivalent across genders.

Viewed vs Unviewed Offer Completions





Discount 2, bogo 4 and discount 3 are the most commonly completed offers without users viewing them.
On the other hand, discount 1, 4, 3 and bogo 3 are the most completed after being viewed.

Model and Recommendation Engine

- Ran 3 different types of model and the decision tree classifier was the best.
- This is to be expected as a decision tree is a perfect algorithm for the question we are asking in this trial.

	Model	train F1 score	test F1 score
0	KNeighborsClassifier (Benchmark)	71.718168	58.473026
1	RandomForestClassifier	96.854612	67.244430
2	DecisionTreeClassifier	96.854612	79.345468

Final function of recommendation engine:

```
def user_user_recs(user_id, m=3, user_item=user_item):
    #recommendations = movie_names(recs)

    #return recommendations

    most_similar_users = find_similar_users(user_id, user_item = user_item)
    user_offer_ids, user_offer_names = get_user_offers(user_id)

    recs = np.array([])

    for similar_user_id in most_similar_users:
        if len(recs)<m:
            similar_offer_ids, similar_offer_names = get_user_offers(similar_user_id)

            new_recs = np.setdiff1d(similar_offer_ids, user_offer_ids, assume_unique=True)

            recs = np.unique(np.concatenate([new_recs, recs], axis=0))
            recs = list(recs)

            #recs = [item for sublist in recs for item in sublist]
        else:
            break

    recs = recs[:m]

    return recs # return your recommendations for this user_id
```

Example of recommendation:

- 55 year old male with an income of 83,000 who joined in 2015.
- Is recommended bogo 4, discount 1 & 3 which is to be expected

```
user_user_recs('00ae03011f9f49b8a4b3e6d416678b0b')
```

```
['bogo_4', 'discount_1', 'discount_3']
```

```
df_pred[df_pred['user_id']=='00ae03011f9f49b8a4b3e6d416678b0b'].head(1)
```

	gender	age	user_id	became_member_on	income	20s	30s	40s	50s
249035	M	55	00ae03011f9f49b8a4b3e6d416678b0b	2015-11-15	83000.0	0	0	0	1

Final Recommendations

Firstly I would implement a feature to the promotions **where a link must be activated in order to receive the reward for completing it**. It is a glaring fact that there are just as many unviewed promotion completions as there are viewed ones. This would eliminate the unviewed completions while not affecting the viewed completions at all.

Secondly I would employ several heuristics when deciding on whom to and what types of promotions I would send out. There are as follows:

- Do not bother with promotions bogo 2 or 4, they are not popular and when they are completed it is very often without the users having viewed the offer.
- Discount 1 is by far the most effective offer. It has the least unviewed completions and the most viewed completions. This despite it being the most difficult offer to complete. When confronted with a new user, send discount 1.
- Scale offer difficulty to income levels. Higher income earners need harder offers.

Use a decision tree style method.

1. Income the 1st branch: discount 1 for high incomes, discount 1 & 3 for low incomes
2. Gender the 2nd: discount 1 for females, discount 1 & 3 for males
3. Year they joined as the final: