UDACITY | STARBUCKS CAPSTONE

Asif Alam

1. Problem Definition

1.1 Project Overview

More companies are embracing making decisions based on data and with the rise of supervised machine learning techniques, this is one of the most popular topics among data scientists. The success of a company is largely attributed to how well they know and serve their customers and with a large pool of data in today's world, knowing your customers has become vital.

Starbucks has collected data from customers and aims to explore it in detail and extract meaning from it. 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. It is important to target product offerings to specific customers for retention and making sure the offers are maximized.

1.2 Problem Statement

With the help of the datasets we look to answer two important questions:

- What makes an offer irresistible to the customer? Can we focus on certain features?
- Is it possible to predict if a customer will take a certain offer?

This is a very interesting problem to solve, as most companies are fighting to retain customers and to earn new ones. Having a targeted campaign will enable them to achieve this. Also, this is great for customers as they will receive promotions and offers that they find useful.

1.3 Metrics

The two metrics we will be looking at are the accuracy score and the F1-score.

"F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution"

In this project we will aim for an accuracy of anywhere between 75 to 95 percent.

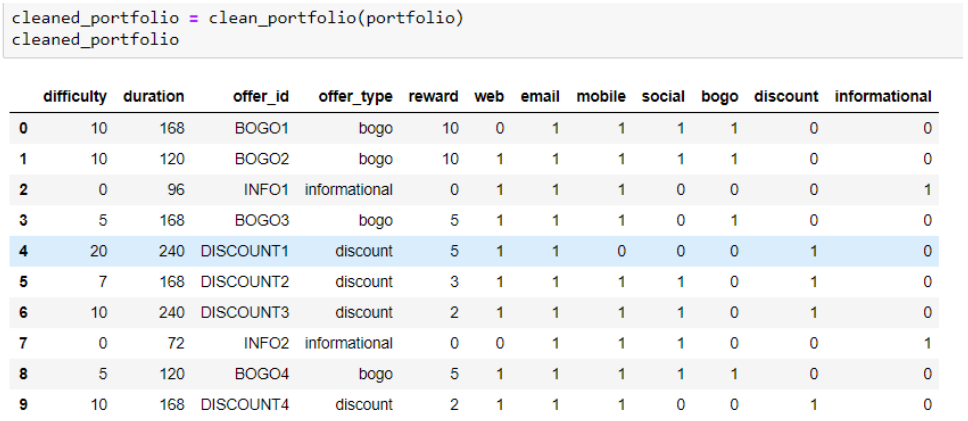
2. Analysis

2.1 Data Exploration

There are 3 datasets present as json files in this project. Each file will be explained in greater detail below.

Filename	File contents	Data size
portfolio.json	• 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)	10 x 5 fields
profile.json	• 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)	17000 x 5 fields
transcript.json	• 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"	306648 x 4 fields

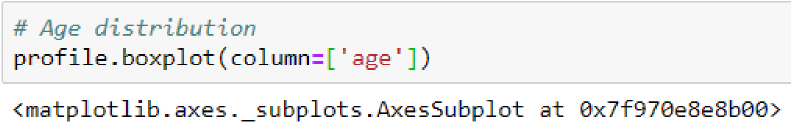
1. **portfolio.json**
   This was converted into a data frame. The channel and offer_type was one-hot encoded and the id column name was changed to offer_id to make it easier to identify. Also the offer id hash was mapped into an id which was easier to read. After the transformation the data frame looked like below:



```
cleaned_portfolio = clean_portfolio(portfolio)
cleaned_portfolio
```

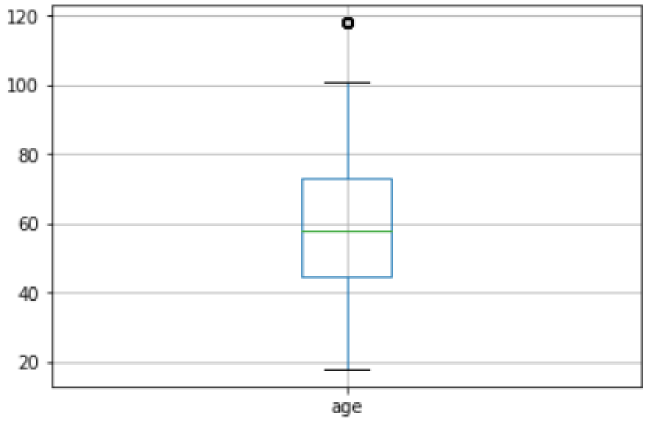
	difficulty	duration	offer_id	offer_type	reward	web	email	mobile	social	bogo	discount	informational
0	10	168	BOGO1	bogo	10	0	1	1	1	1	0	0
1	10	120	BOGO2	bogo	10	1	1	1	1	1	0	0
2	0	96	INFO1	informational	0	1	1	1	0	0	0	1
3	5	168	BOGO3	bogo	5	1	1	1	0	1	0	0
4	20	240	DISCOUNT1	discount	5	1	1	0	0	0	1	0
5	7	168	DISCOUNT2	discount	3	1	1	1	1	0	1	0
6	10	240	DISCOUNT3	discount	2	1	1	1	1	0	1	0
7	0	72	INFO2	informational	0	0	1	1	1	0	0	1
8	5	120	BOGO4	bogo	5	1	1	1	1	1	0	0
9	10	168	DISCOUNT4	discount	2	1	1	1	0	0	1	0

2. **profile.json**
This was converted to a data frame. The date was converted from string to datetime. Also, this column was split into another three columns: *became_member_on_year_month*, *became_member_on_year*, *became_member_on_month*. The gender column was one-hot encoded. Upon studying the age range of the customers, an outlier was found:



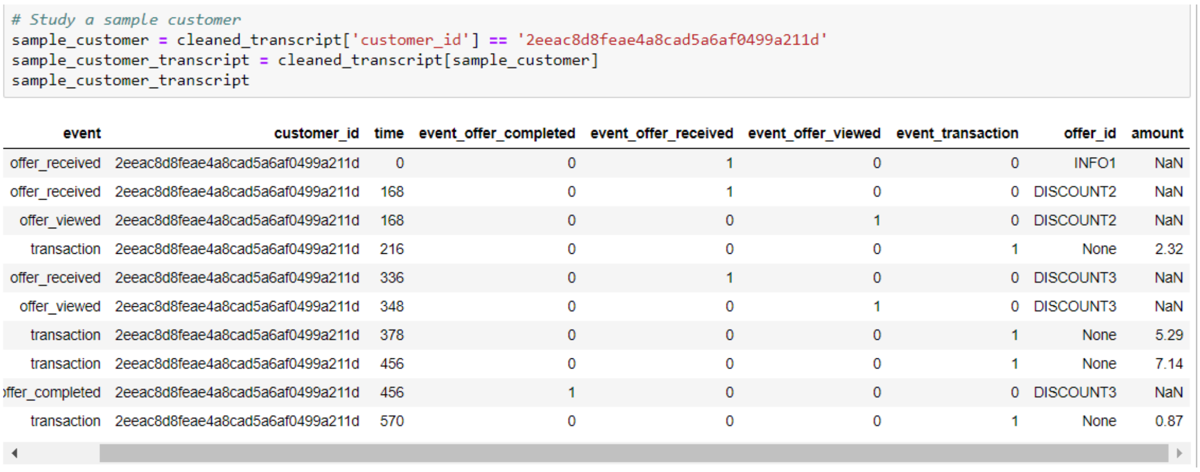
```
# Age distribution
profile.boxplot(column=['age'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f970e8e8b00>
```



An outlier flag was introduced to the rows with ages of 118. Further investigation showed these rows had null incomes.

3. **transcript.json**
This dataset contains the events in chronological order. In the data frame the amount spent was extracted when the event was a transaction. It is important to understand this data set from a customer point of view:

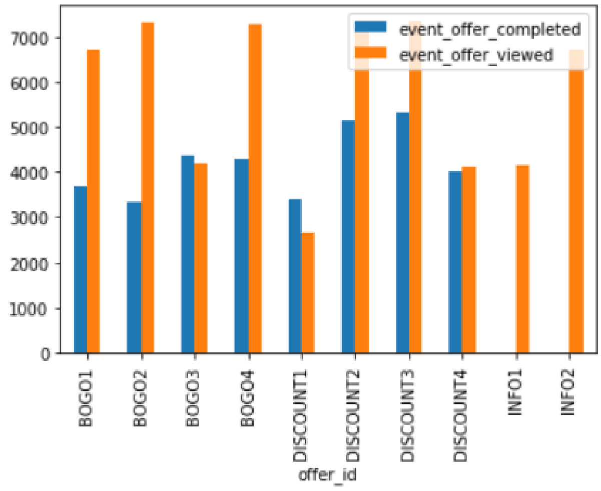


```
# Study a sample customer
sample_customer = cleaned_transcript['customer_id'] == '2eeac8d8feae4a8cad5a6af0499a211d'
sample_customer_transcript = cleaned_transcript[sample_customer]
sample_customer_transcript
```

event	customer_id	time	event_offer_completed	event_offer_received	event_offer_viewed	event_transaction	offer_id	amount
offer_received	2eeac8d8feae4a8cad5a6af0499a211d	0	0	1	0	0	INFO1	NaN
offer_received	2eeac8d8feae4a8cad5a6af0499a211d	168	0	1	0	0	DISCOUNT2	NaN
offer_viewed	2eeac8d8feae4a8cad5a6af0499a211d	168	0	0	1	0	DISCOUNT2	NaN
transaction	2eeac8d8feae4a8cad5a6af0499a211d	216	0	0	0	1	None	2.32
offer_received	2eeac8d8feae4a8cad5a6af0499a211d	336	0	1	0	0	DISCOUNT3	NaN
offer_viewed	2eeac8d8feae4a8cad5a6af0499a211d	348	0	0	1	0	DISCOUNT3	NaN
transaction	2eeac8d8feae4a8cad5a6af0499a211d	378	0	0	0	1	None	5.29
transaction	2eeac8d8feae4a8cad5a6af0499a211d	456	0	0	0	1	None	7.14
ffer_completed	2eeac8d8feae4a8cad5a6af0499a211d	456	1	0	0	0	DISCOUNT3	NaN
transaction	2eeac8d8feae4a8cad5a6af0499a211d	570	0	0	0	1	None	0.87

An offer is deemed successfully complete if the flow of the event is:
offer received -> offer viewed -> transaction
As the above customer received a DISCOUNT3 offer, views the offer and proceeds have a couple of transaction and thus completing the offer.

From this dataset, the most popular offers were examined:



From this bar chart alone, it can be seen that, discount offers have a higher degree of completion especially when communicated over all channels - [web, email, mobile, social]. Also, advertising over socials proves to have better customer views as suggested by data on BOGO3 and DISCOUNT1. These were not viewed by many customers.

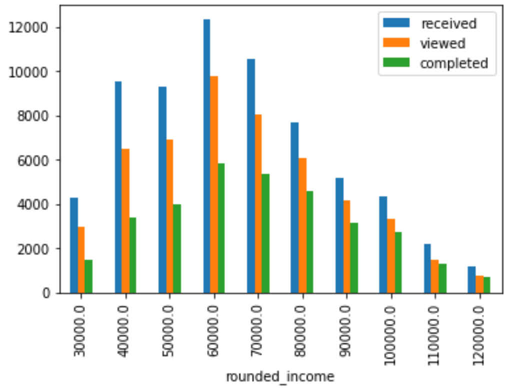
2.2 Data Visualization

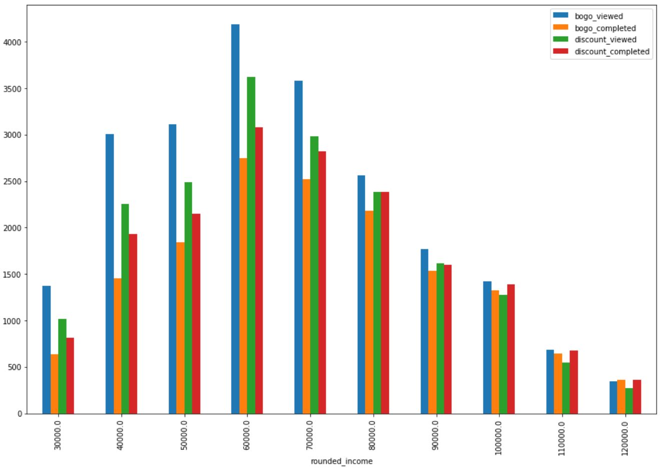
The three datasets discussed above was combined into two sets, one grouped by customer for visualization and another event based for modelling.
The customer_data contained all the columns discussed above with the addition of event_offer_successful and completion_ratio. The first indicates how many times a customer **successfully** availed an offer. The second is just the comparison between offer viewed and completed.
The aim of this dataset is study four features, income, gender, age and membership date, of the customer data which might be useful for targeted offers.

2.2.1 How does customer income effect offer completion?

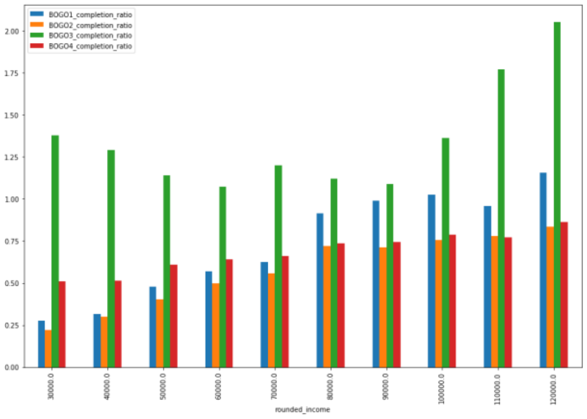
In this segment we will take a look at the income column of the customer dataset. We will try to find a relationship between offer completed (both successfully/unsuccessfully) and how much a customer earns.

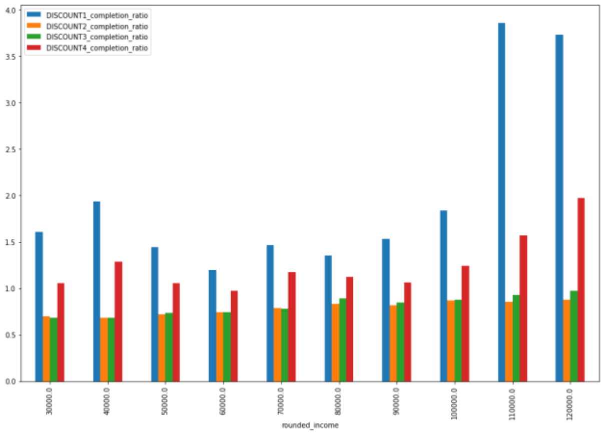
The first overview is a pattern of how income effects offers viewed and completed. As we can see customers with income ranges from 50,000 to 60,000 are receiving, viewing and completing offers more. A deeper dive into the data by offer type shows discount offers get completed more often, even though BOGO offers are viewed more. In some income ranges, discounts are completed without having viewed the offer



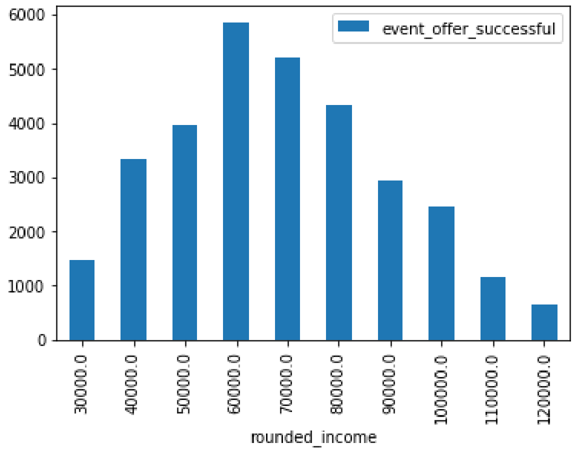


The completion ratio field was introduced to see which offers are popular among these income groups. This shows people with higher income range is completing offers more frequently.





Lastly, we look at people who are completing offers successfully. From the data, people with an income between 60,000 to 70,000 are completing most offers.



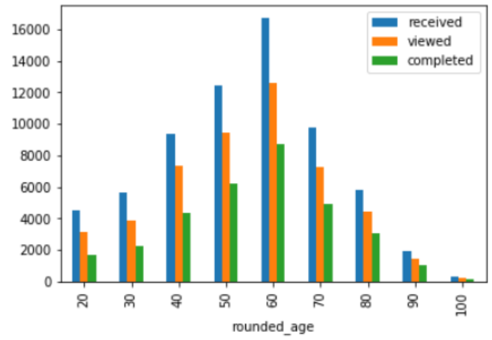
Findings from income study:

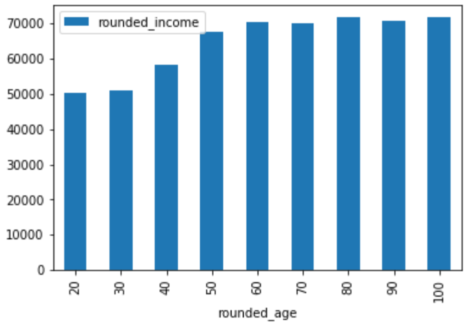
- Customers with 60,000 - 70,000 range are completing offers more successfully. They also view the most number of offers.
- Customers with higher income ranges are not given equal number of offers.
- Discount offers get completed more often, even though BOGO offers are viewed more. In some income ranges, discounts are completed without having viewed the offer.
- DISCOUNT3 was viewed and completed the most, but DISCOUNT1 has a much higher completion ratio.
- DISCOUNT2 and DISCOUNT3 has the same level of completion ratio throughout the income groups. People in the higher income group seems to complete DISCOUNT1 without viewing the offer.
- BOGO3 has the highest completion ratio out of all the BOGO offers.

2.2.2 How does age impact offer completion?

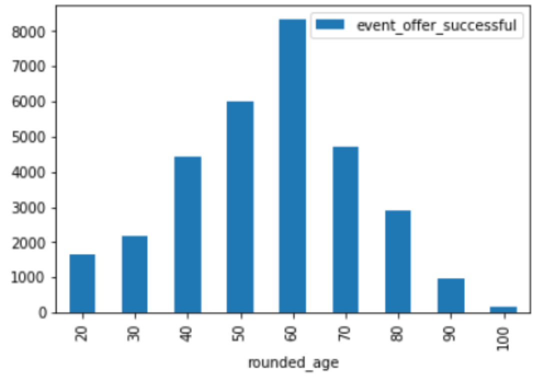
In this section we will see how customer age groups impact offer completion. To make things simpler, the age column is rounded to nearest 10s.

Customers aging 50-60 are receiving, viewing and completing the most offers. However, the income distribution is quite uniform between 60 to 100.





The number of successfully completed offers are also the highest among the age group of 60.



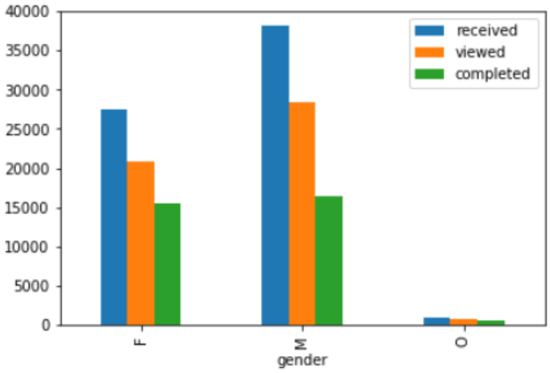
Findings from age study:

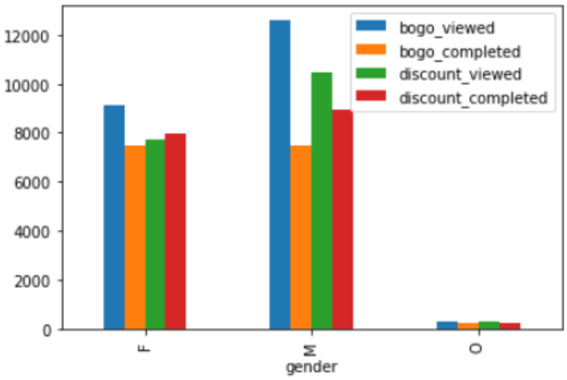
- Age group between 60 to 100 has the same income distribution
- The age group between 50 to 60 has viewed and completed most offers
- The most successful completion comes from the age group of 60, followed by 50.

2.2.3 How does Gender impact offer completion?

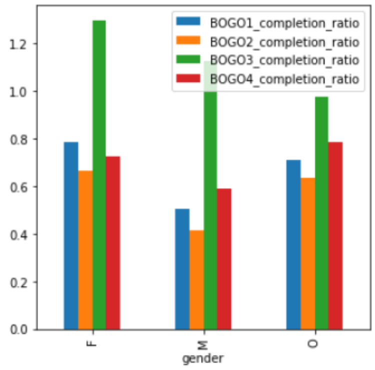
The gender groups are divided into M and F and a relationship between the data and gender is studied. There is an imbalance between the number of males and females in this dataset. So the distribution will be slightly biased towards males.

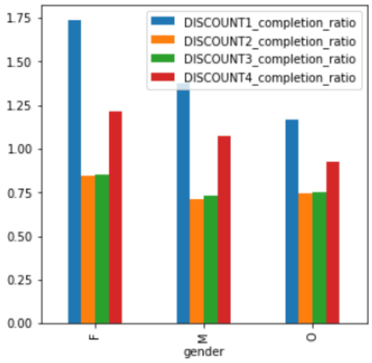
Even though males tend to receive and view more offers, the offer completion between female and male tend to be the same. Also, Both BOGO and DISCOUNT offers are availed almost equally by females. The data suggests males go more for DISCOUNT offers.

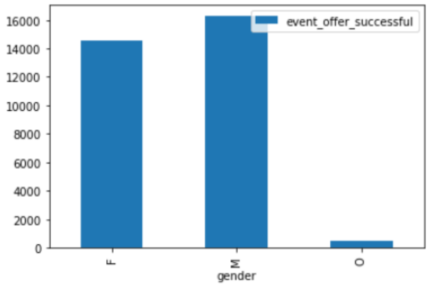




We looked at some of the offer ids in greater detail and compared the completion ratio between different offers. Also, the successful offer event was collated.







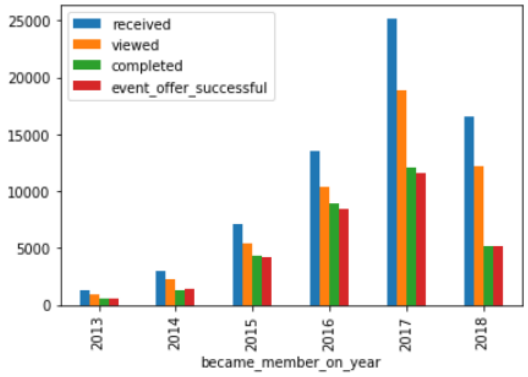
Findings from gender study:

- The dataset contains an uneven number of female, males and others. The others are too little compared to females/males, so it can be dropped.
- Even though males tend to receive and view more offers, the offer completion between female and male tend to be the same
- Both BOGO and DISCOUNT offers are availed almost equally by females. The data suggests males go more for DISCOUNT offers.
- BOGO offer completion rate is higher in females compared to males especially when it comes to BOGO3.
- DISCOUNT1 is completed most of the time, sometimes without even viewing by Females.

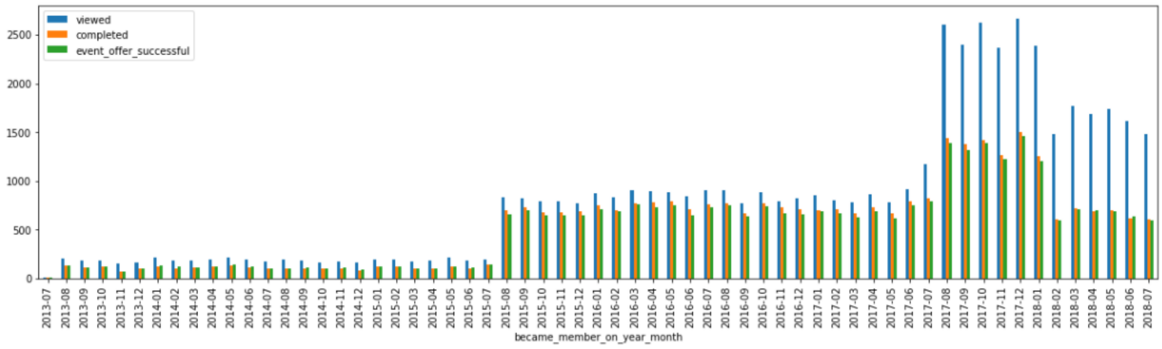
2.2.4 How does membership duration impact offer completion?

In this section, the membership start year and month is studied and any correlation is visualized.

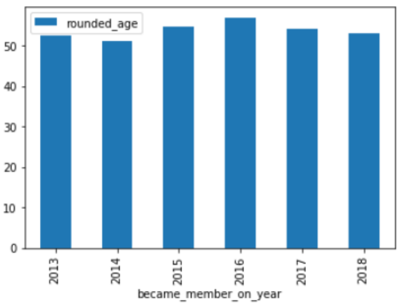
Looking at the membership year, it can be seen members who joined in 2017 are receiving, viewing and completing more offers than others.

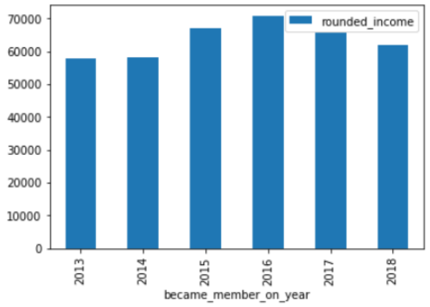


A deeper look at the dates indicate a surge of views and completions around late 2017 and early 2018.



The characteristics (age, income) of the members who joined at each year was also studied. The average age of the customer tends to be the same for members who joined at different years. Also, the average income remains somewhat consistent, with members joining in 2016 earning slightly higher on average.



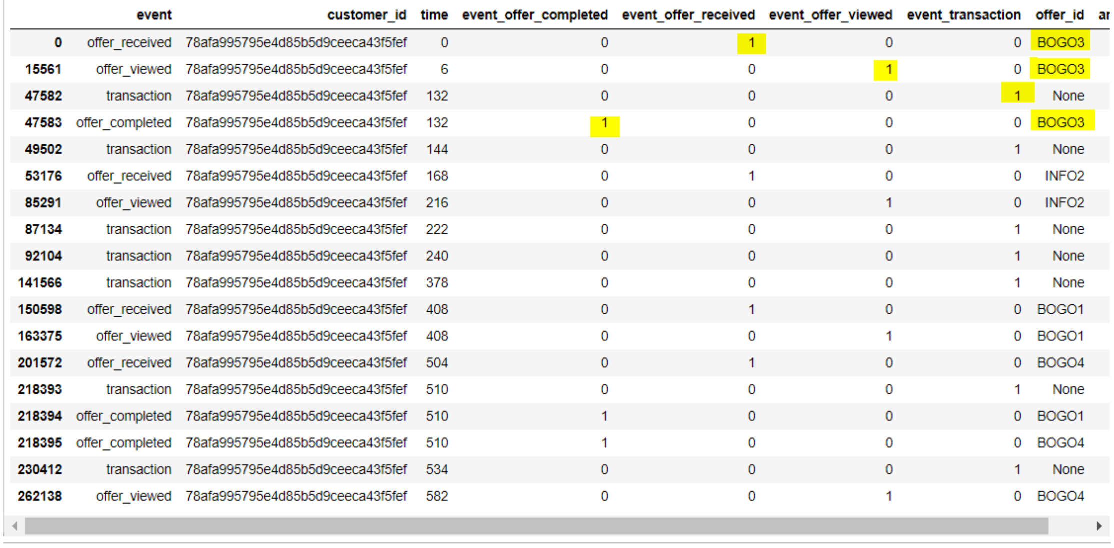


Findings:

- Members who joined in 2017 appears to have completed more offers. This is because they received the most offers
- There is a spike in offer views and offers completed in members who joined between late 2017 and early 2018

2.3 Algorithms & Techniques

To prepare our modelling data set, the first thing to calculate is if an offer was successfully completed. The transcript data was split into two – transactions and offers. If the transaction happened after the offer was received and viewed within the time of the experiment, the **event_offer_successful** flag was set. This is the most important part of the project, as this would become our target feature.
The problem then calls for **a supervised learning classification** algorithm to be implemented and extract important features.



	event	customer_id	time	event_offer_completed	event_offer_received	event_offer_viewed	event_transaction	offer_id	ar
0	offer_received	78afa995795e4d85b5d9ceeca43f5fef	0	0	1	0	0	BOGO3	
15561	offer_viewed	78afa995795e4d85b5d9ceeca43f5fef	6	0	0	1	0	BOGO3	
47582	transaction	78afa995795e4d85b5d9ceeca43f5fef	132	0	0	0	1	None	
47583	offer_completed	78afa995795e4d85b5d9ceeca43f5fef	132	1	0	0	0	BOGO3	
49502	transaction	78afa995795e4d85b5d9ceeca43f5fef	144	0	0	0	1	None	
53176	offer_received	78afa995795e4d85b5d9ceeca43f5fef	168	0	1	0	0	INFO2	
85291	offer_viewed	78afa995795e4d85b5d9ceeca43f5fef	216	0	0	1	0	INFO2	
87134	transaction	78afa995795e4d85b5d9ceeca43f5fef	222	0	0	0	1	None	
92104	transaction	78afa995795e4d85b5d9ceeca43f5fef	240	0	0	0	1	None	
141566	transaction	78afa995795e4d85b5d9ceeca43f5fef	378	0	0	0	1	None	
150598	offer_received	78afa995795e4d85b5d9ceeca43f5fef	408	0	1	0	0	BOGO1	
163375	offer_viewed	78afa995795e4d85b5d9ceeca43f5fef	408	0	0	1	0	BOGO1	
201572	offer_received	78afa995795e4d85b5d9ceeca43f5fef	504	0	1	0	0	BOGO4	
218393	transaction	78afa995795e4d85b5d9ceeca43f5fef	510	0	0	0	1	None	
218394	offer_completed	78afa995795e4d85b5d9ceeca43f5fef	510	1	0	0	0	BOGO1	
218395	offer_completed	78afa995795e4d85b5d9ceeca43f5fef	510	1	0	0	0	BOGO4	
230412	transaction	78afa995795e4d85b5d9ceeca43f5fef	534	0	0	0	1	None	
262138	offer_viewed	78afa995795e4d85b5d9ceeca43f5fef	582	0	0	1	0	BOGO4	

2.4 Benchmark

The worst-case benchmark was using naïve a predictor accuracy and F1-score. If the model predicted positives for all cases:

- Naive predictor accuracy: 0.468492016816
- Naive predictor f1-score: 0.638058649895

A logistic regression classifier was used to produce the benchmark:

- LogisticRegression model accuracy: 0.71
- LogisticRegression model f1-score: 0.701

The aim is to produce a model with an accuracy and f1 score higher than **71**%.

3. Methodology

This section is walkthrough for creating the final ML models.

3.1 Data Pre-processing

The initial step in pre-processing the data is to select rows without the outlier flag which was set in the first data exploration steps.

Secondly, the string values are represented as integers. Gender and offer types are converted to numerical representation. Also, the membership start date was converted to number of days a customer has been a member with Starbucks. Then the data rows are shuffled to ensure random distribution.

In the end we get 65585 rows with 16 features and 1 target.

The final dataset is passed through a MinMaxScaler which scales all the features down to a range of 0 to 1 for faster processing.

The training data is 70% of the entire data, and the rest are used for testing.

3.2 Implementation & Refinement

For the ML model implementation, I chose to use classifiers from sk learn. With every classifiers, I have used RandomizedSearchCV, which implements a randomized search over parameters, where each setting is sampled from a distribution over possible parameter values. This was chosen as it is faster than GridSearch.

3.2.1 Logistic Regression

A logistic regression classifier was used to create a benchmark. This is a widely used classifier and one of the most common ones available. The reason behind choosing this for our benchmark:

- Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.
- Easy to implement, interpret, and very efficient to train.

3.2.2 SVC

A support vector classifier was one of the models considered for this problem. This was chosen as svm has less risk of over-fitting our data and having a kernel choice allows us to solve complex problems.

This model was used with RandomizedSearchCV to get the best hyperparameter values. A range of values was tried out for gamma and C and the best value that was produced was:

'kernel': 'poly',
'gamma': 0.24657938454623807,
'degree': 4,
'C': 0.93151531873191462

3.2.3 Decision Tree

A decision tree was used as it is one of the simplest models to understand and interpret. It also works well with data that have non-linear relationships between features. Also, exploratory analysis is good with decision trees.
The two hyperparameter that was tuned to get the most out of this model were: max_features and min_samples_leaf using RandomizedSearchCV. After a few iterations the best parameters were seen to be:
'min_samples_leaf': 10,
'max_features': 12,
'max_depth': None,
'criterion': 'entropy'

3.2.4 Random Forest

Random forest has some properties which makes it suitable for this use case:

- It provides higher accuracy through cross validation
- If there are more trees, it won't allow over-fitting trees in the model
- It has the power to handle a large data set with higher dimensionality

The two hyperparameters that were tuned to get optimal results were: min_samples_split, min_samples_leaf and n_estimators using RandomizedSearchCV. After a few iterations the best parameters were extracted:
'n_estimators': 200,
'min_samples_split': 7,
'min_samples_leaf': 5,
'max_features': 'sqrt',
'max_depth': 10

3.2.5 AdaBoost

Adaboost was used for refining and improving accuracy on the decision tree model. It involves using very short (one-level) decision trees as weak learners that are added sequentially to the ensemble. Each subsequent model attempts to correct the predictions made by the model before it in the sequence.
The two hyperparameter that was tweaked to avoid over-fitting were: learning_rate and n_estimators.
The best set of values that worked on this dataset were:
'n_estimators': 75,
'learning_rate': 2.1

4. Results & Conclusion

The machine learning algorithms implemented with their performance stats and feature importance are:

Classifier	Accuracy score	F1-score	Important Feature 1	Important Feature 2	Important Feature 3
Logistic Regression	0.71	0.70	N/A	N/A	N/A
SVC	0.74	0.73	N/A	N/A	N/A
Decision Trees	0.81	0.79	member_for_days	income	age
Random Forest	0.77	0.75	member_for_days	income	age
AdaBoost	0.90	0.89	member_for_days	age	income

To ensure consistency, these models were trained with varying hyperparameters using RandomizedSearchCV as stated above.

It is evident that the features highlighted above plays a key role in determining whether a customer will take up an offer or not. From this we can recommend:

- Sending more offers to customers who have been a member longer
- Focusing and sending more offers to customers with a higher income, from the visualization it is evident they have higher completion ratio, but are not getting enough offers in the first place
- Sending more offers to female customers, as they are likely to complete more offers
- Send more offers to older age groups as they tend to have higher income.