Starbucks Capstone Challenge

Table of Contents

Project Overview

This is a simulation for the 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.

Data Sets

The data is contained in three files:

Portfolio file that contains offer ids and meta data about each offer

1. id - offer id
2. offer_type - type of offer ie BOGO, discount, informational
3. difficulty - minimum required spend to complete an offer
4. reward - reward given for completing an offer
5. duration - time for offer to be open, in days
6. channels

Profile file that contains demographic data for each customer

1. age - customer's age
2. became_member_on - account creation date
3. gender - customers's gender
4. id - customer's id
5. income - customer's income

Transcript file that contains records for transactions, offers received, offers viewed, and offers completed

1. event - record description (ie transaction, offer received, offer viewed, etc.)
2. person - customer id
3. time - time in hours since start of offer. The data begins at time t=0
4. value - either an offer id or transaction amount depending on the record

Problem Statement / Metrics

As long as not all users receive the same offer, I need to build a recommendation system that provides the most suitable offers for each user.

This new recommendation system can be tested using the A/B testing concept to compare it to existing systems and evaluate the added value of this new system.

Data Exploration and Visualization

in this part we'll explore the datasets to understand more the data and the relation between different variables.

1-Portfolio Data Set

- 10 different offers of types bogo, discount and informational offers
- The offers' channels are concatenated in one column and needs splitting

All the Data Set:

	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5

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

2- Profile Data Set

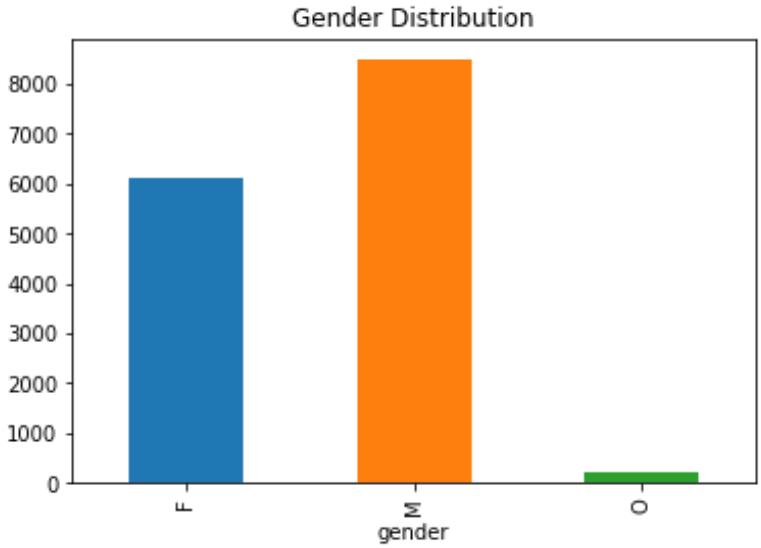
- 17000 distinct customers.
- Some NaN values in the columns.
- Age column has some big values – like 118, this needs to be removed.
- Males have higher distribution than Females
- Most of the Customers are member since 2017/2018

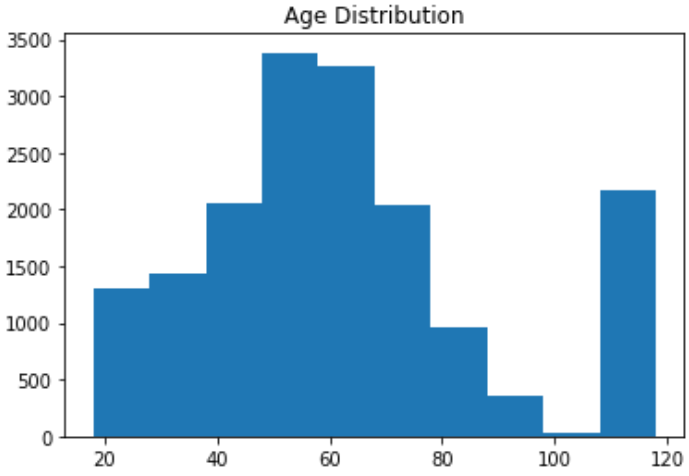
Sample of the Dataset

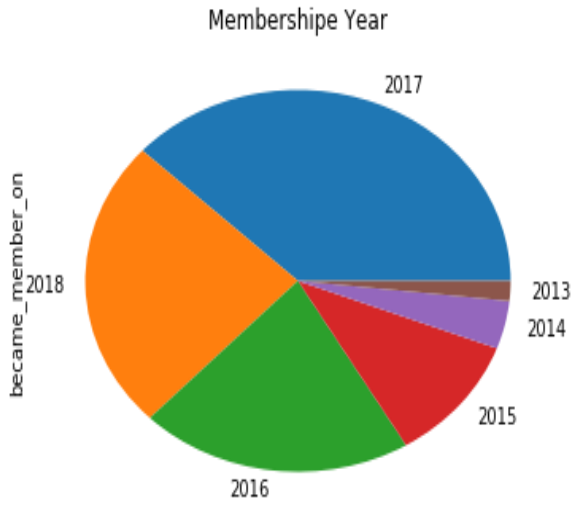
	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0

Some Statistics about the numerical columns:

	count	mean	std	min	25%	50%	75%	max
age	17000.0	6.253141e+01	26.738580	18.0	45.0	58.0	73.0	118.0
became_member_on	17000.0	2.016703e+07	11677.499961	20130729.0	20160526.0	20170802.0	20171230.0	20180726.0
income	14825.0	6.540499e+04	21598.299410	30000.0	49000.0	64000.0	80000.0	120000.0



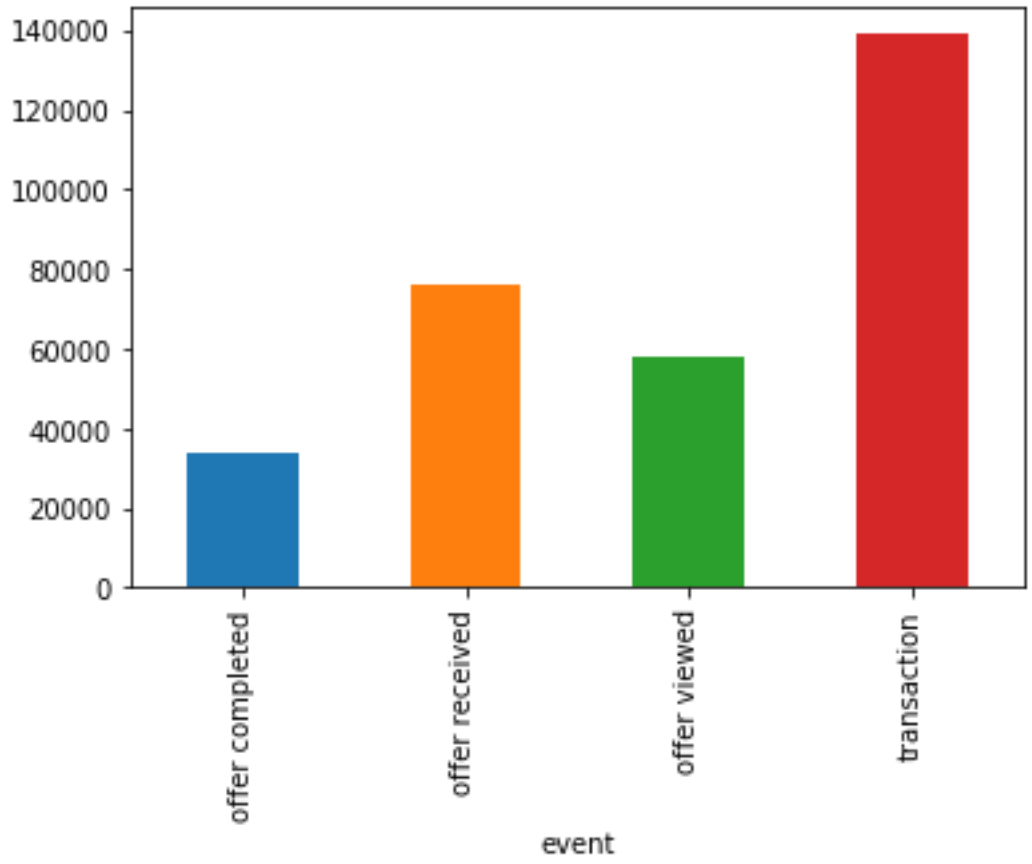




3-Transcript Data Set

- For different event status: 'offer received', 'offer viewed', 'transaction', 'offer completed'
- The transaction event has no offer id associated with it.
- The Value column contains offer_ids, amount spent

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



Data Preprocessing

Cleaning Portfolio dataset

1- Rename the id column to 'offer_id'

2- Encoding the channels column and offer_type column

3- Removing the old columns (channels and offer_type)

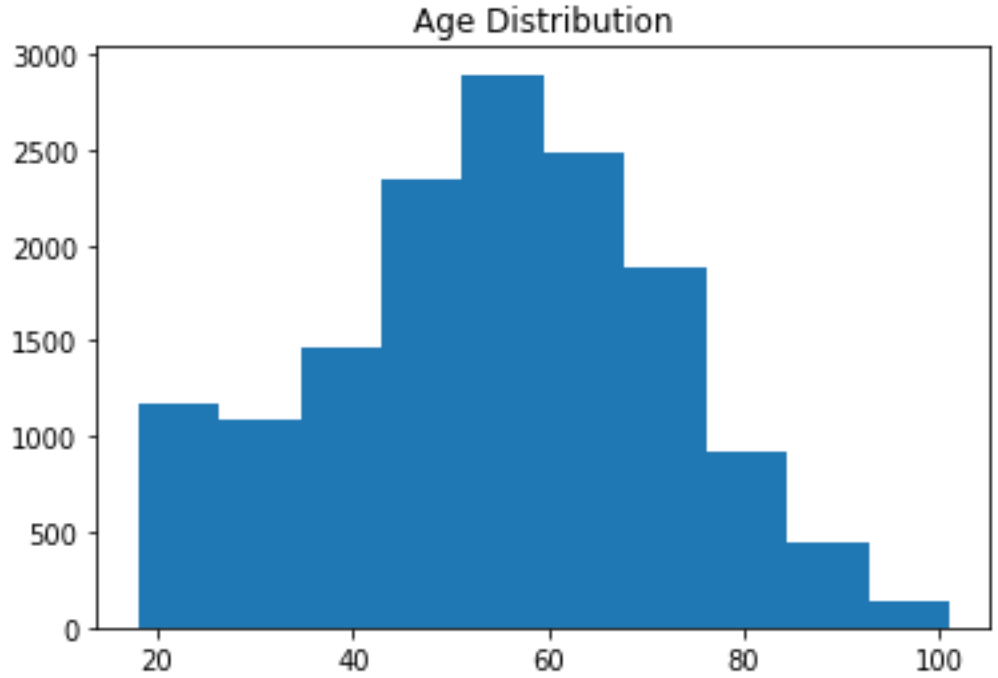
	difficulty	duration	offer_id	reward	web	email	mobile	social	bogo	discount	informational
0	10	7	ae264e3637204a6fb9bb56bc8210ddfd	10	0	1	1	1	1	0	0
1	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	10	1	1	1	1	1	0	0

Cleaning Profile Dataset

1. removing null values
2. removing the outliers in age column
3. mapping the id of the customer to a simpler form of 'user_id'

After Removing the Null values the max age became 101 instead of 118

	count	mean	std	min	25%	50%	75%	max
age	14825.0	5.439352e+01	17.383705	18.0	42.0	55.0	66.0	101.0
became_member_on	14825.0	2.016689e+07	11885.653317	20130729.0	20160520.0	20170802.0	20171230.0	20180726.0
income	14825.0	6.540499e+04	21598.299410	30000.0	49000.0	64000.0	80000.0	120000.0



Adding a simpler mapping user id for the customers

	age	became_member_on	gender	id	income	user_id
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0	1

	age	became_member_on	gender	id	income	user_id
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0	2

Cleaning Transcript Dataset

1. extract the offer_id and the amount of money from value column
2. encode the 'event' column
3. drop the old columns 'value', 'event'

Data after parsing and extracting offer id and amount columns from value column:

	amount	offer_id	reward	offer completed	offer received	offer viewed	transaction	user_id	time_in_days
0	NaN	9b98b8c7a33c4b65b9aebfe6a799e6d9	NaN	0	1	0	0	78afa995795e4d85b5d9ceeca43f5fef	0.0
1	NaN	0b1e1539f2cc45b7b9fa7c272da2e1d7	NaN	0	1	0	0	a03223e636434f42ac4c3df47e8bac43	0.0
2	NaN	2906b810c7d4411798c6938adc9daaa5	NaN	0	1	0	0	e2127556f4f64592b11af22de27a7932	0.0
3	NaN	fafdcd668e3743c1bb461111dcafc2a4	NaN	0	1	0	0	8ec6ce2a7e7949b1bf142def7d0e0586	0.0
4	NaN	4d5c57ea9a6940dd891ad53e9dbe8da0	NaN	0	1	0	0	68617ca6246f4fbc85e91a2a49552598	0.0

Merging All Data Frames Together:

Merging the 3 data frames together transcript, portfolio, profile using User_id and Offer_id to g et a wider data frame that contains the user id with his profile info., the offer id with the offer's portfolio, the status of the offer per user.

Implementation

Building a Recommendation System

User-User Based Collaborative Filtering:

This is to be used for the Current users for Starbucks offers based on similarity and closeness to the target us er.

This is done on 4 steps:

1. Creating user_item matrix of users as rows and offers as columns
2. Find the similar users to the target user sorted by similarity and on a tie it sorts by the number of use d offer by each user.
3. Get the Offers used by the above extracted users.
4. Final ranking for the above extracted offers by the total number that these offers have been used an d completed.

Sample of the similar users data frame:

	user_id	Similarity	completed_cnt
3624	3806	5	2
7148	7520	5	2
12580	13238	5	2
8401	8830	5	1
9756	10250	5	1

Sample of the top 5 recommended offers for a specific user:
['2298d6c36e964ae4a3e7e9706d1fb8c2',
'f19421c1d4aa40978ebb69ca19b0e20d',
'ae264e3637204a6fb9bb56bc8210ddfd',
'9b98b8c7a33c4b65b9aebfe6a799e6d9',
'4d5c57ea9a6940dd891ad53e9dbe8da0']

Refinement

As new user will have no similarity with any of the existing users as they haven't used any offers yet, so we can use rank based recommendation for those customers based on top offers used.

Rank Based Recommendations:

This is to be used for any new user as it ranks the offers based on the usage.
Sample of the output:

	offer_id	usage_cnt
9	fafdcd668e3743c1bb461111dcafc2a4	4436
4	4d5c57ea9a6940dd891ad53e9dbe8da0	4423
1	2298d6c36e964ae4a3e7e9706d1fb8c2	4370

Evaluation and Validation

As we didn't use a machine learning model in this problem, the evaluation of this recommendation system shall be through A/B testing concept where the users are divided randomly into two paths (one for the existing recommendation system and the other for our new recommendation system).

This would show the performance of the new system over the old one, and based on the testing we can decide the way forward.

Justification

In this project we covered different types of users (existing and new) through different types of recommendation systems (Rank based and collaborative filtering)

Conclusion - Reflection

From the above we can see that we can use a combination of 2 techniques for Offers Recommendations

1. for New users that have never used the Starbucks offers, we can use Rank Based Recommendations.
2. for Current users for Starbucks offers, we can use User-User Based Collaborative Filtering.

Improvement

The recommendation system can be further improved by combining different types of recommendation systems like matrix factorization and other techniques; which I will consider in phase 2 of this project.