Capstone Project
Machine Learning Nanodegree

Deasy Indrawati
February 9, 2021

Starbucks Capstone Challenge

Definition

Project Overview

Simulated datasets from Starbucks rewards mobile app contain data that mimics customer behavior. From the datasets, my task is determining demographic groups respond best to each offer type.

Problem Statement

Starbucks sends out offer to the users once every few days through rewards mobile app. Offer type is discount, BOGO (buy one get one free) and information. Not all users get the same offer. Some users might not receive any offer during certain weeks. Every offer has validity period, informational offer has 7 days validity and BOGO has 5 days validity. The other important information is user might make purchase through app without having received or seen any offer.
From the datasets given, my goals are combining transaction, demographic and offer to determine demographic groups respond best to each offer type.

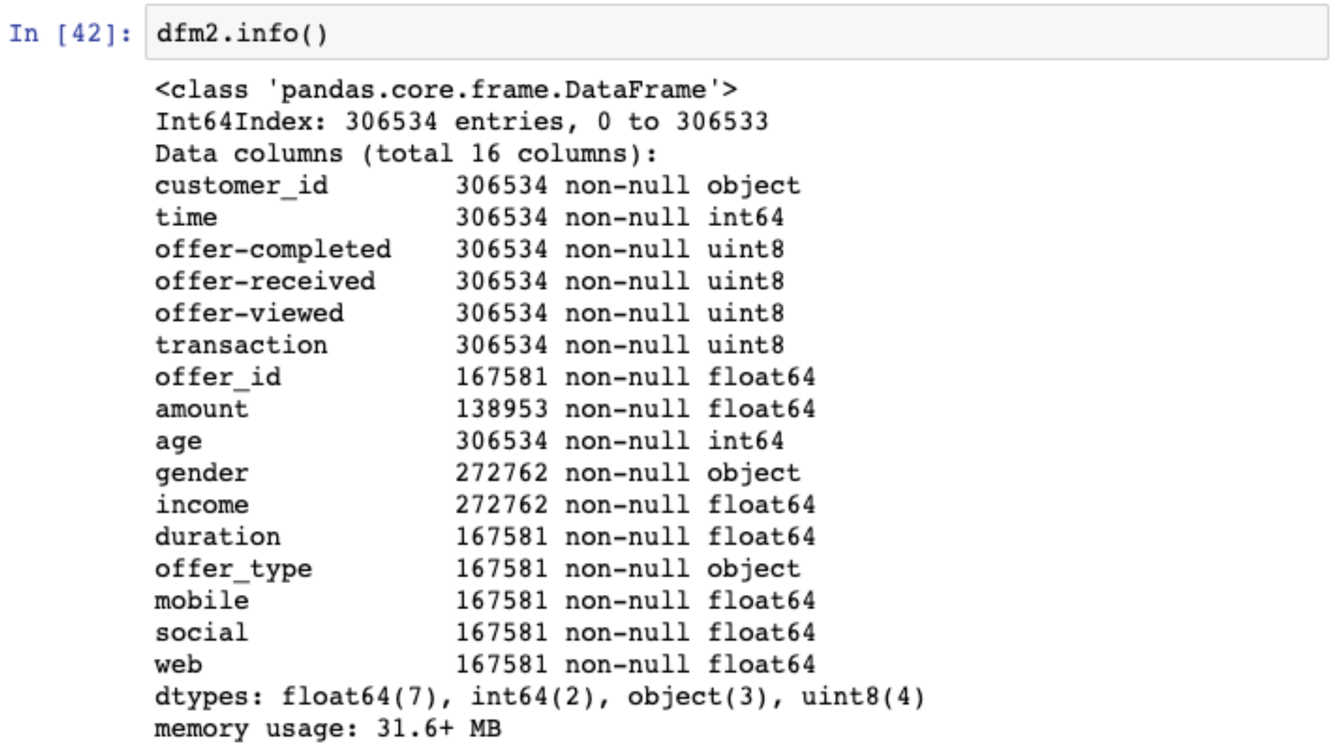
Metrics

To find demographic groups that respond best to each offer type I need to used clustering method. The most suitable algorithm for this project is KMeans algorithm. By using this technique, we try to minimize the distance between the points within a cluster.

Analysis

Data Exploration

For this project, we are provided with 3 datasets: portfolio, profile and transcript. Portfolio files contain offer id and meta data about each offer (duration, type, etc). Profile files contain demographic data of each customer. Transcript files contain records for transactions, offers received, offers viewed, and offers completed. From portfolio files we can get detailed information about difficulty to get reward from the offer.



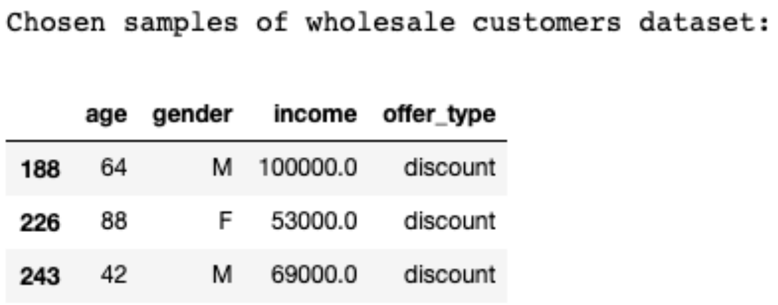
```
In [42]: dfm2.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 306534 entries, 0 to 306533
Data columns (total 16 columns):
customer_id        306534 non-null object
time               306534 non-null int64
offer-completed    306534 non-null uint8
offer-received     306534 non-null uint8
offer-viewed       306534 non-null uint8
transaction        306534 non-null uint8
offer_id           167581 non-null float64
amount             138953 non-null float64
age                306534 non-null int64
gender             272762 non-null object
income             272762 non-null float64
duration           167581 non-null float64
offer_type         167581 non-null object
mobile             167581 non-null float64
social             167581 non-null float64
web                167581 non-null float64
dtypes: float64(7), int64(2), object(3), uint8(4)
memory usage: 31.6+ MB
```

From the profile files, there are 17000 data and 5 columns. There are 2175 data with status N/A. Amounts of data with "null" income also 2175 data. The data with N/A value will be cleaned. Beside that outlier data inside the profile files also be removed. Data from profile files will be clustered the customer. After that, clustered data is useful for the Starbucks to give the right offer to customers who have big chance to complete the offer.

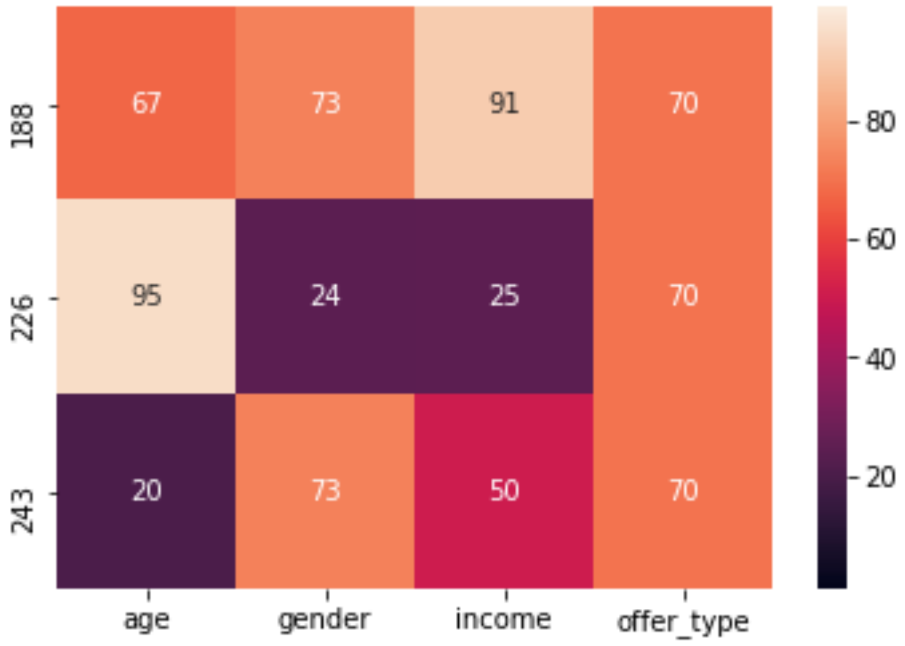
Exploratory Visualization

In this section we will summaries relevant characteristic about the data. To find out about this, we will use samples of wholesale customers datasets number 188, 226 and 243. For data visualization we will import seaborn library to create heat map. The step is calculate the percentile ranks of the dataset, round it up and multiply by 100. Heat map use value of percentiles on samples of customer dataset before.



```
Chosen samples of wholesale customers dataset:
```

	age	gender	income	offer_type
188	64	M	100000.0	discount
226	88	F	53000.0	discount
243	42	M	69000.0	discount



Algorithms and Techniques

To get higher percentage of customer who complete the offer, we will cluster the data. First thing that need to be done is cleaning the data. We need to remove null data and outliers from the lists. After the data clean, we will merge the files profile and transcript with the key customer id. Data ready to be clustered. The clustered data will contain group of customers based on their demographic who give higher percentage to complete the offer.

This data will be helpful for the Starbucks to give offer to the right customer that had higher chance to complete the offer.

Starbucks is big company and sell famous coffee in the world. They also have a lot of offer for their customers. But they need to find out the most effective rewards for their customers based on their demographic. Cleaning data is must have procedure before processing it. Outliers and null value will be removed.

Profile and portfolio files will be merged with the key customer id. After data ready, we will cluster them. There will be around 15000 data clustered. Clustered data must have special demographic criteria. They will be sorted based the question, "will they have higher chance to complete the offer?"

K-Means algorithm will be used to clustering customer. This method is the most compatible to identify customer who might complete the offer. First thing to do is specify the number of clusters. Second step is initializing centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement. Last, keep iterating until there is no change to the centroids.

There are needs to evaluation for this algorithm, too. Elbow method will be used to find the good K numbers of the clustered data. To get the number we need to sum the squared distance (SSE) between data points and their assigned cluster's centroids. A good number of K found at the spot where SSE start to flatten out and forming an elbow.

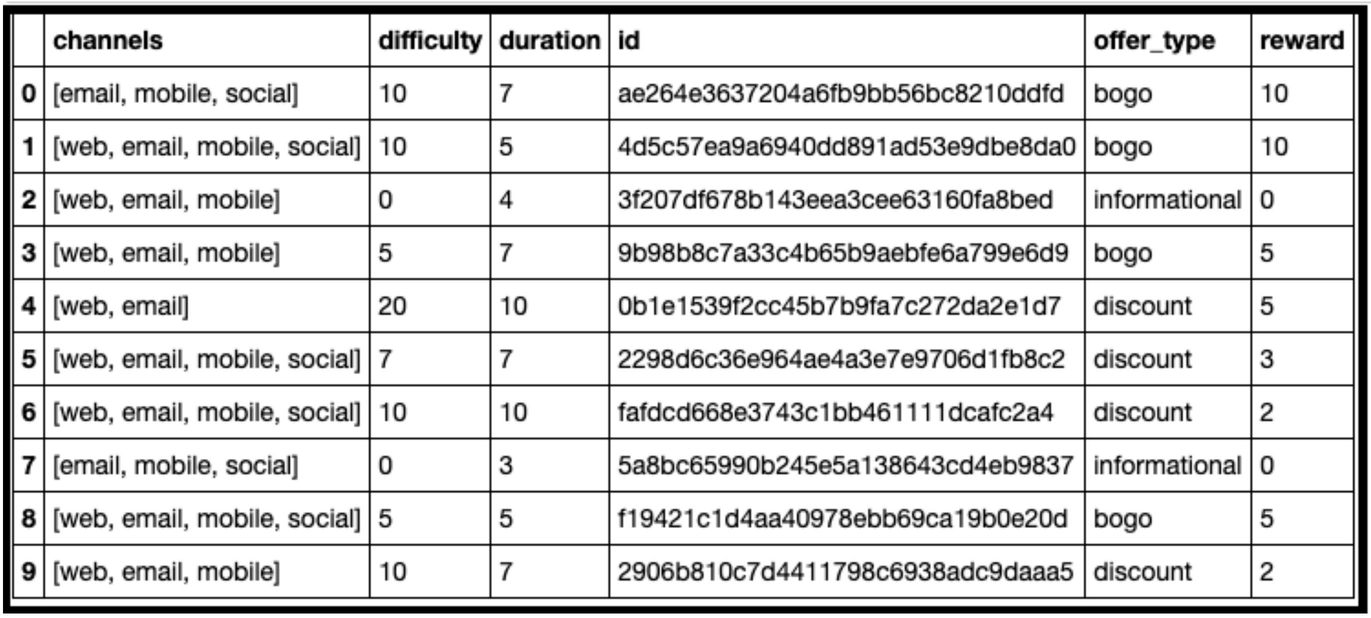
Benchmark

There are other clustering algorithms that could be used to do customer segmentation. One possible to use is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. But K-Means is the most compatible for clustering because DBSCAN is used to clustering different shapes and sizes from large amount of data. Beside that DBSCAN is good for data that is containing noise and outlier.

Methodology

Data Pre-processing

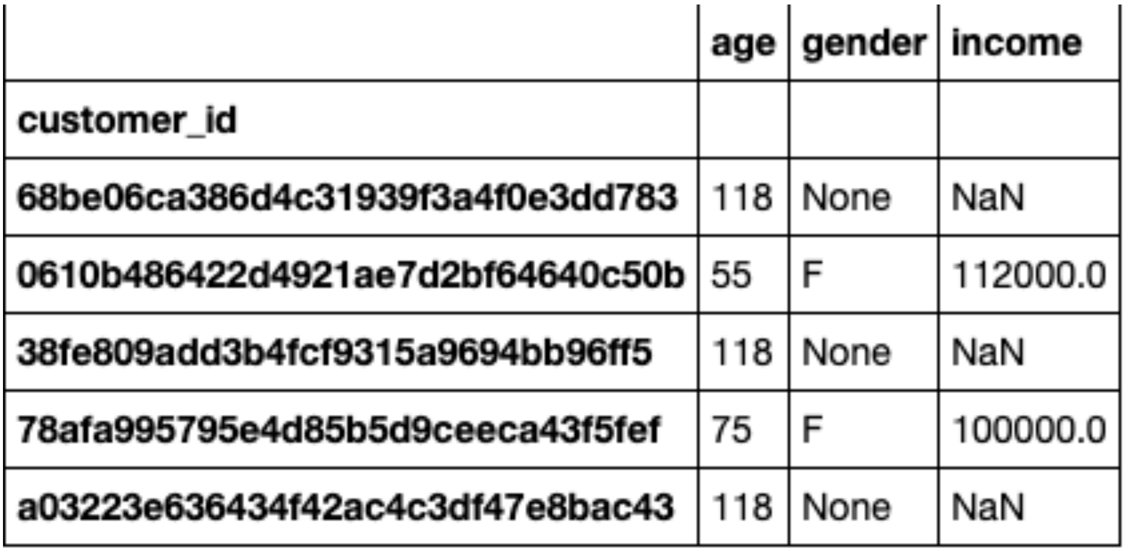
From this project we got 3 datasets; portfolio, transcript and profile. The three datasets is json file. Portfolio file contain 6 columns; offer id, offer type, difficulty, reward, duration and channels.



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

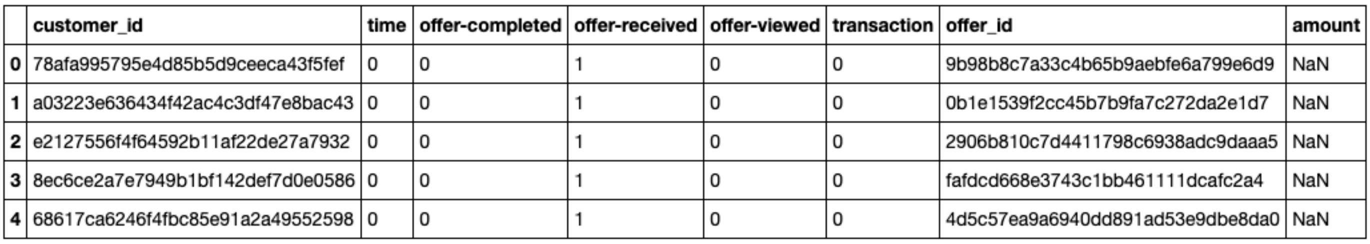
The files is sorted with offer id as key. Channels is filter out into four columns; email, mobile, social and web. After that the channels is dropped. Column difficulty is used to filter out the specific condition later. After processed; portfolio file contains 6 columns; offer id, duration, offer type, mobile, social and web. Email columns is dropped because all offer id use email to give the offer.

The second files names transcript. Profile files contain 5 columns; age, date of becoming member, gender, id and income. Id is replaced to customer id. Customer id then become the key from this dataset followed by age, gender and income. Column date becoming member is dropped because will not be used for data analysis.



	age	gender	income
customer_id			
68be06ca386d4c31939f3a4f0e3dd783	118	None	NaN
0610b486422d4921ae7d2bf64640c50b	55	F	112000.0
38fe809add3b4fcf9315a9694bb96ff5	118	None	NaN
78afa995795e4d85b5d9ceeca43f5fef	75	F	100000.0
a03223e636434f42ac4c3df47e8bac43	118	None	NaN

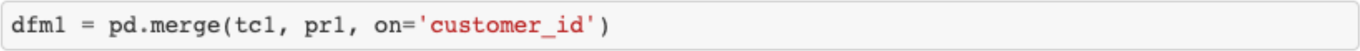
The third file, transcript, contain 4 columns; event, person, time and value. The event column contain value offer received, offer viewed, transaction, offer completed. The column value replace into offer id. The person is replaced into customer id. The event column then filter out into different column and give value 1 on each offer. After that column value and event is dropped. The column customer id then use to be the key. Transcript datasets after processed contain 8 columns; customer id, time, offer completed, received, viewed, transaction, offer id and amount.



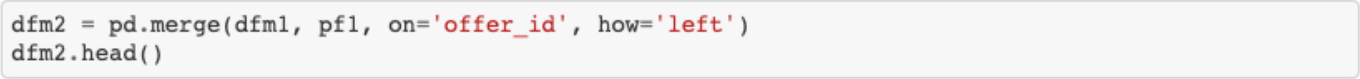
	customer_id	time	offer-completed	offer-received	offer-viewed	transaction	offer_id	amount
0	78afa995795e4d85b5d9ceeca43f5fef	0	0	1	0	0	9b98b8c7a33c4b65b9aebfe6a799e6d9	NaN
1	a03223e636434f42ac4c3df47e8bac43	0	0	1	0	0	0b1e1539f2cc45b7b9fa7c272da2e1d7	NaN
2	e2127556f4f64592b11af22de27a7932	0	0	1	0	0	2906b810c7d4411798c6938adc9daaa5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

Implementation

All the processed then merged. First, merged transcript and profile by the customer id then merged with profile by offer id.



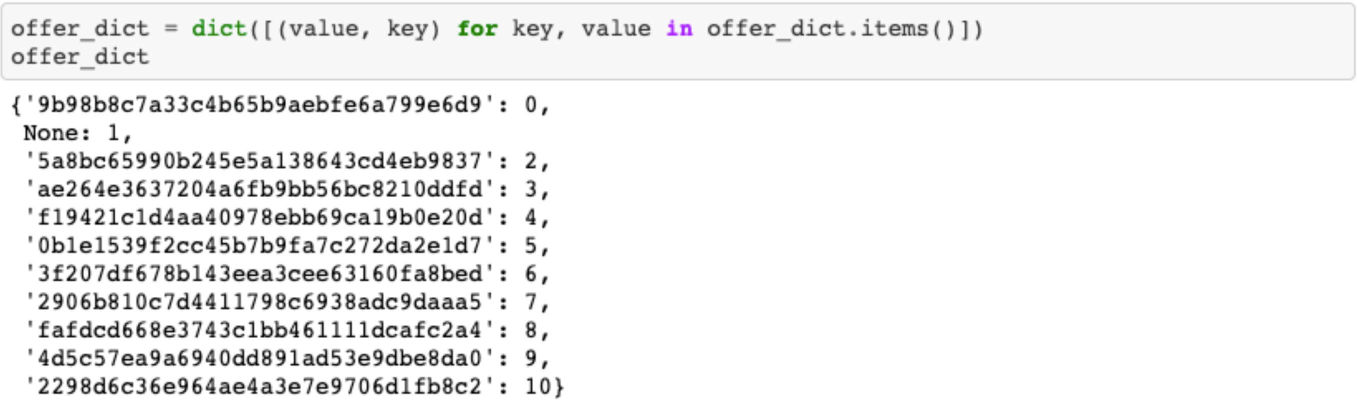
```
dfm1 = pd.merge(tc1, pr1, on='customer_id')
```



```
dfm2 = pd.merge(dfm1, pf1, on='offer_id', how='left')
dfm2.head()
```

Refinement

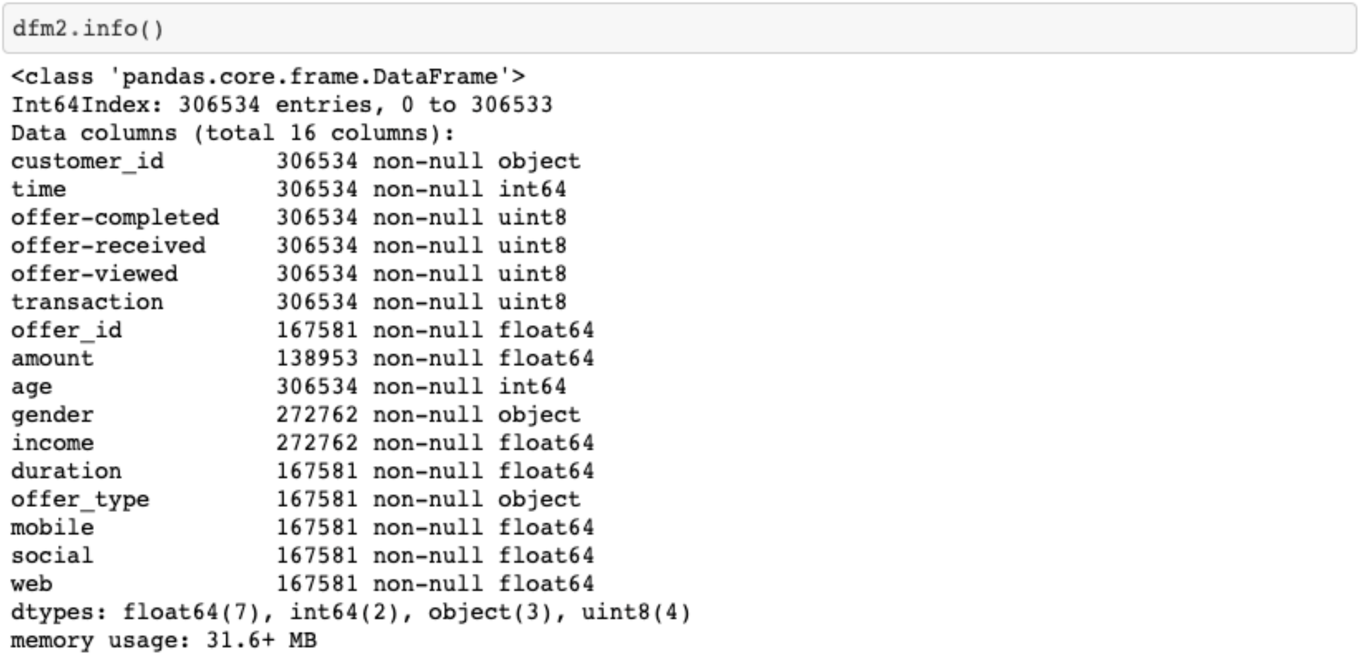
The next step is replace the offer id using unique and to_dict. From that formula we can get 11 items of offer id. Each offer id then replaced by number.



```
offer_dict = dict([(value, key) for key, value in offer_dict.items()])
offer_dict
```

```
{'9b98b8c7a33c4b65b9aebfe6a799e6d9': 0,
 None: 1,
 '5a8bc65990b245e5a138643cd4eb9837': 2,
 'ae264e3637204a6fb9bb56bc8210ddfd': 3,
 'f19421c1d4aa40978ebb69ca19b0e20d': 4,
 '0b1e1539f2cc45b7b9fa7c272da2e1d7': 5,
 '3f207df678b143eea3cee63160fa8bed': 6,
 '2906b810c7d4411798c6938adc9daaa5': 7,
 'fafdcd668e3743c1bb461111dcafc2a4': 8,
 '4d5c57ea9a6940dd891ad53e9dbe8da0': 9,
 '2298d6c36e964ae4a3e7e9706d1fb8c2': 10}
```

The processed file contain 306,534 data and 16 columns; customer id, time, offer completed, offer received, offer viewd, transaction, offer id, amount, age, gender, income, duration, offer type, mobile, social, and web. From each column there are information:



```
dfm2.info()
```

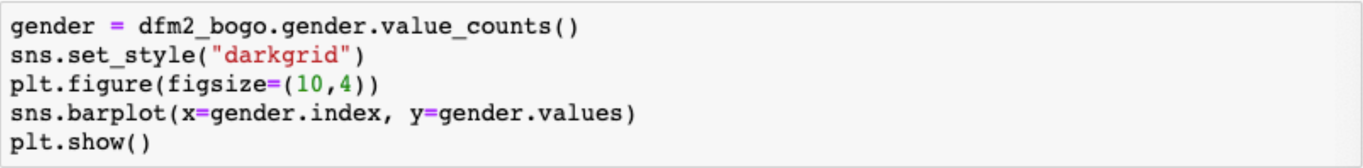
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 306534 entries, 0 to 306533
Data columns (total 16 columns):
customer_id        306534 non-null object
time               306534 non-null int64
offer-completed    306534 non-null uint8
offer-received     306534 non-null uint8
offer-viewed       306534 non-null uint8
transaction        306534 non-null uint8
offer_id           167581 non-null float64
amount             138953 non-null float64
age                306534 non-null int64
gender             272762 non-null object
income             272762 non-null float64
duration           167581 non-null float64
offer_type         167581 non-null object
mobile             167581 non-null float64
social             167581 non-null float64
web                167581 non-null float64
dtypes: float64(7), int64(2), object(3), uint8(4)
memory usage: 31.6+ MB
```

In the column offer type we can calculate the number of each value; bogo is 71617 data, discount 69898 data and informational 26066 data. The next step is drop the value in the column offer complete that contain number 0. From this step we can get 33,579 complete offer data. From this data we need to find the completed offer with condition the time completing the offer is less than duration given. On each value then we can get 3532 data for discount and 2349 data for bogo.

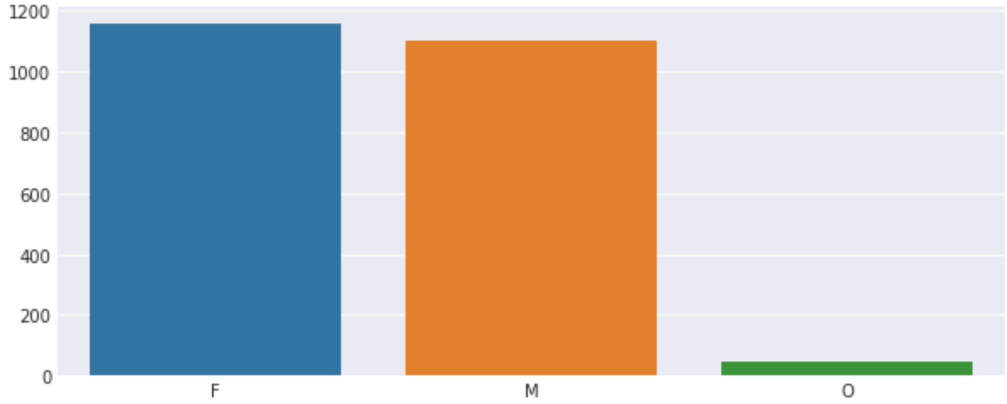
Results

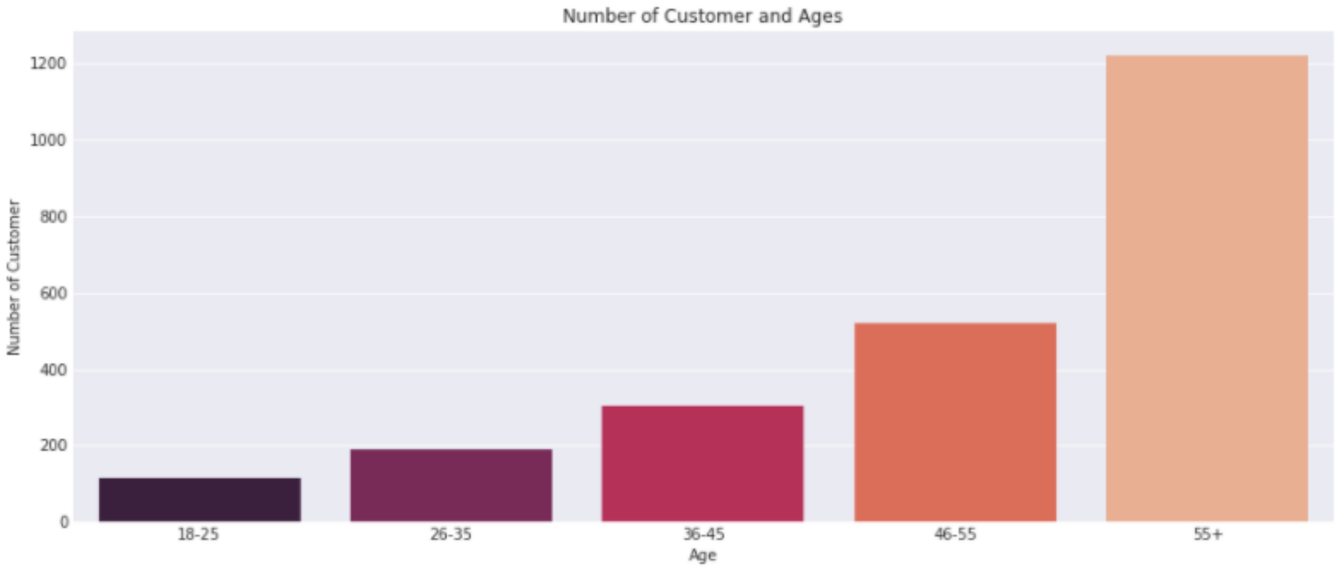
Model Evaluation and Validation

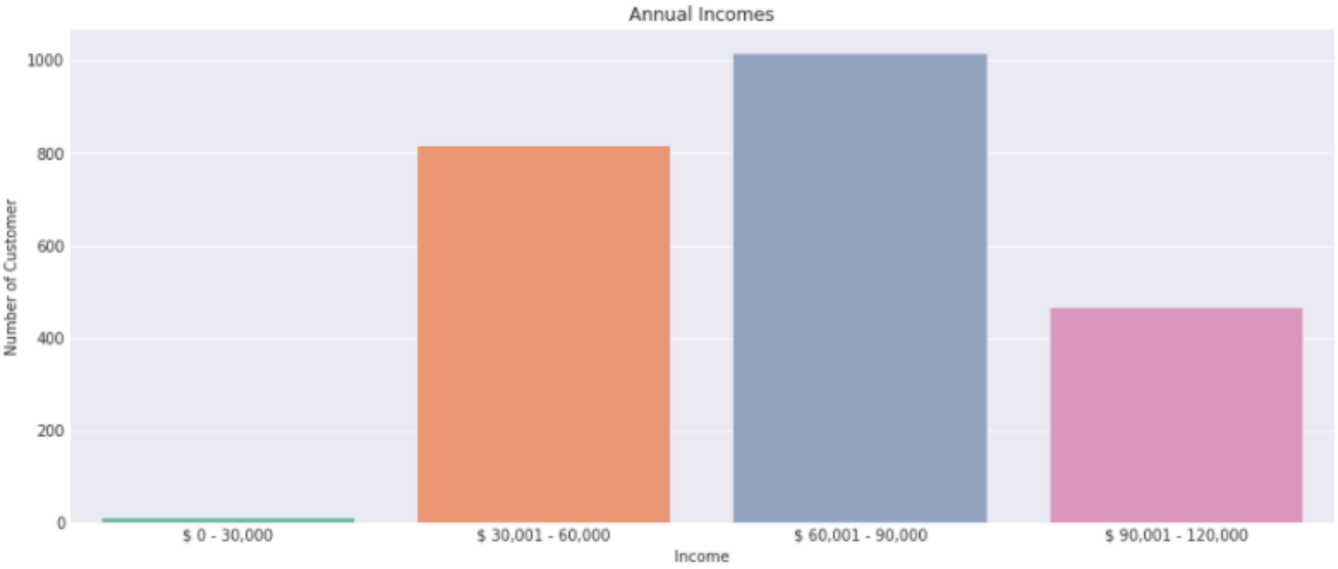
The next step is evaluate data on each offer type with the first type to evaluate is bogo. We only need column age, gender, income and offer type for this step. Then, offer type except the value bogo is dropped. After that the column offer type also dropped so we can figure out the information using bars diagram.
From the offer type bogo there are this information:



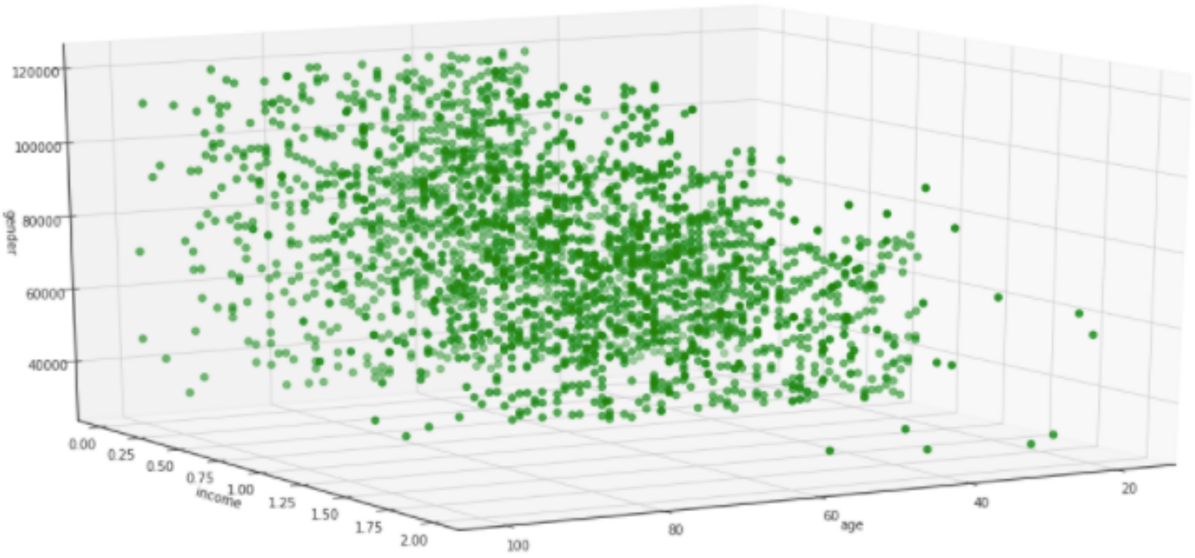
```
gender = dfm2_bogo.gender.value_counts()
sns.set_style("darkgrid")
plt.figure(figsize=(10,4))
sns.barplot(x=gender.index, y=gender.values)
plt.show()
```



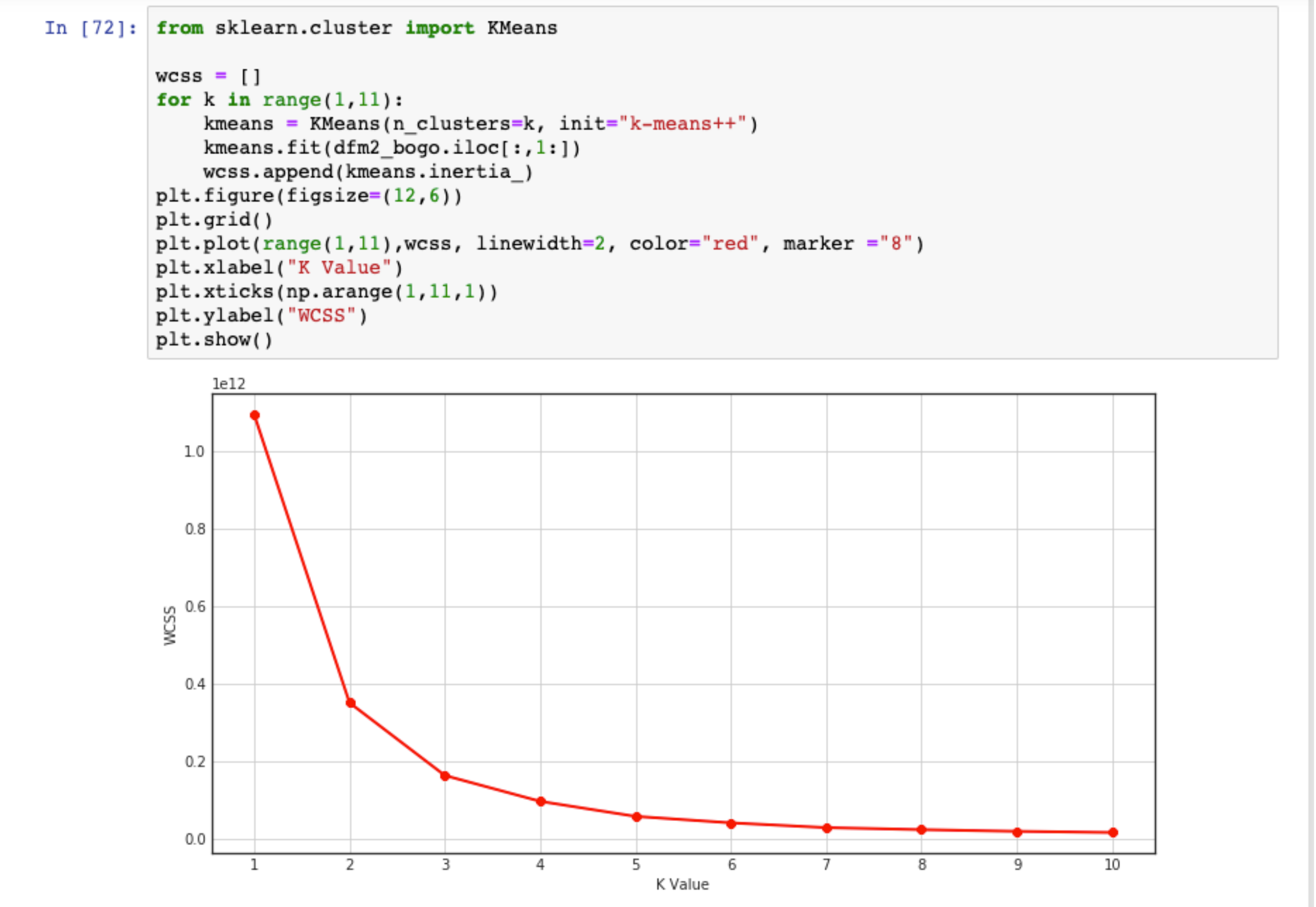




To create 3 dimentional graphing, we need to replace the value in gender with index; 0 for female, 1 for male, 2 for othe and 3 for Nan value. Then here is the graph showing the data in offer type bogo.



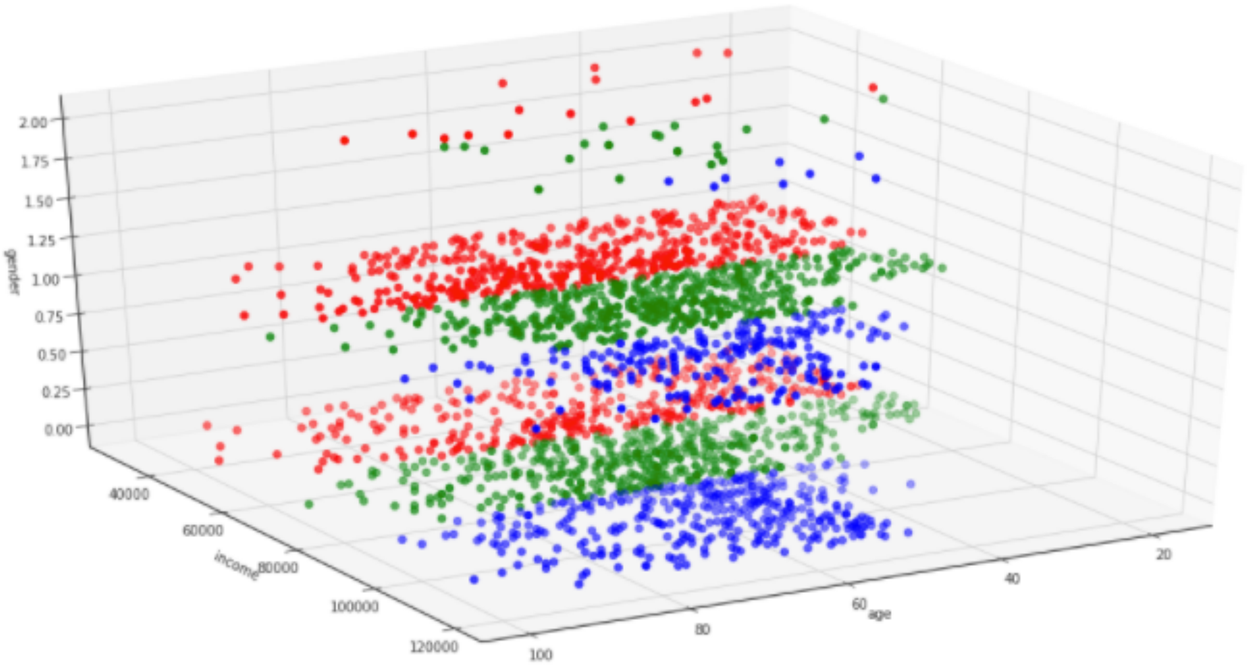
The next step is find out the number of cluster (k) using elbow method.



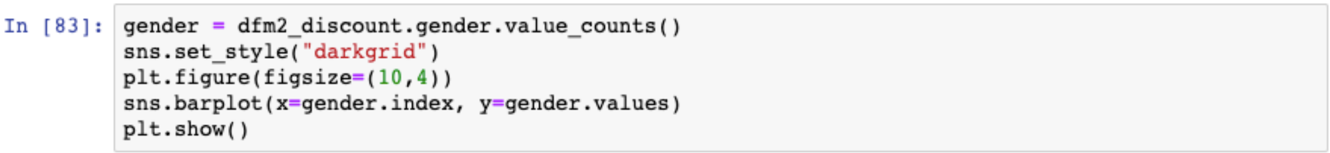
```
In [72]: from sklearn.cluster import KMeans

         wcss = []
         for k in range(1,11):
             kmeans = KMeans(n_clusters=k, init="k-means++")
             kmeans.fit(dfm2_bogo.iloc[:,1:])
             wcss.append(kmeans.inertia_)
         plt.figure(figsize=(12,6))
         plt.grid()
         plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")
         plt.xlabel("K Value")
         plt.xticks(np.arange(1,11,1))
         plt.ylabel("WCSS")
         plt.show()
```

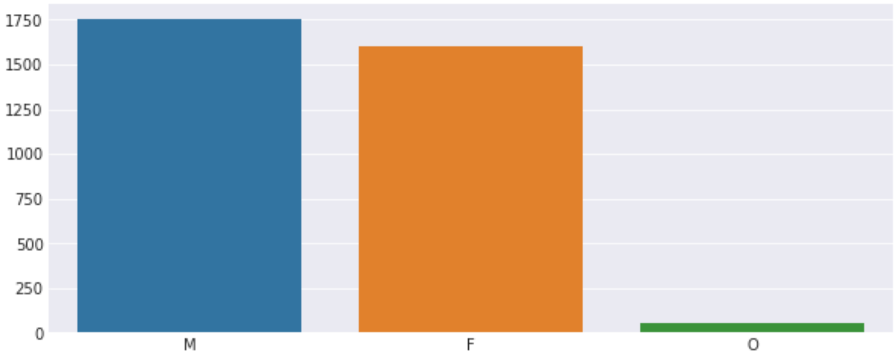
From the elbow method we can see the elbow is on k value 3. This is the maximum number of k to cluster bogo offer type.

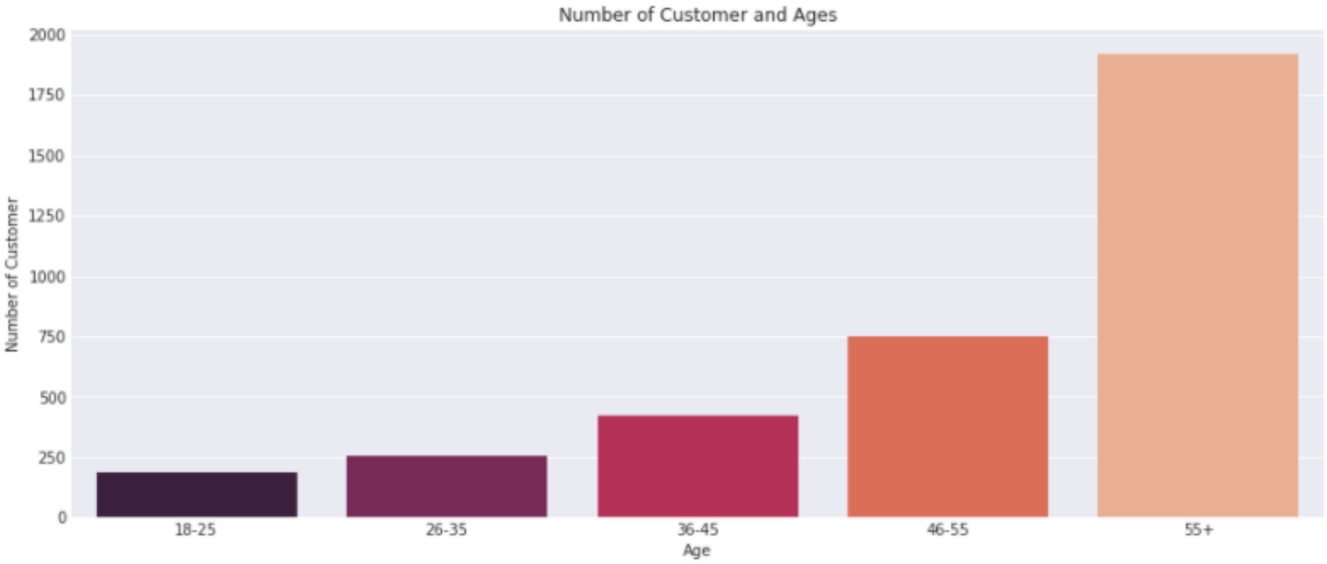


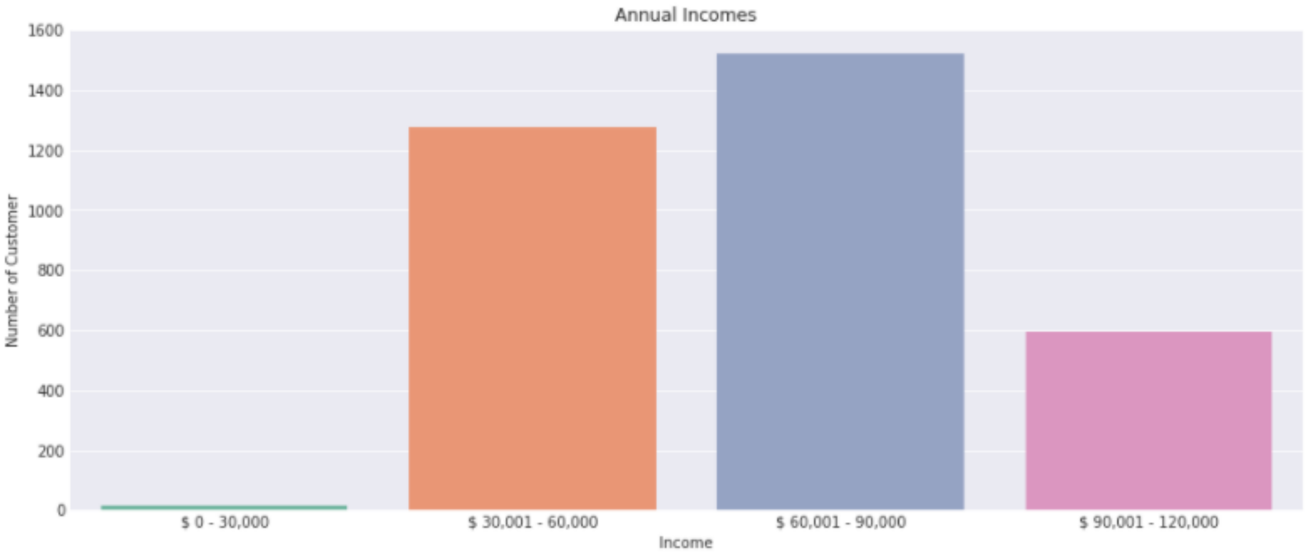
There are the same step too for the offer type discount:

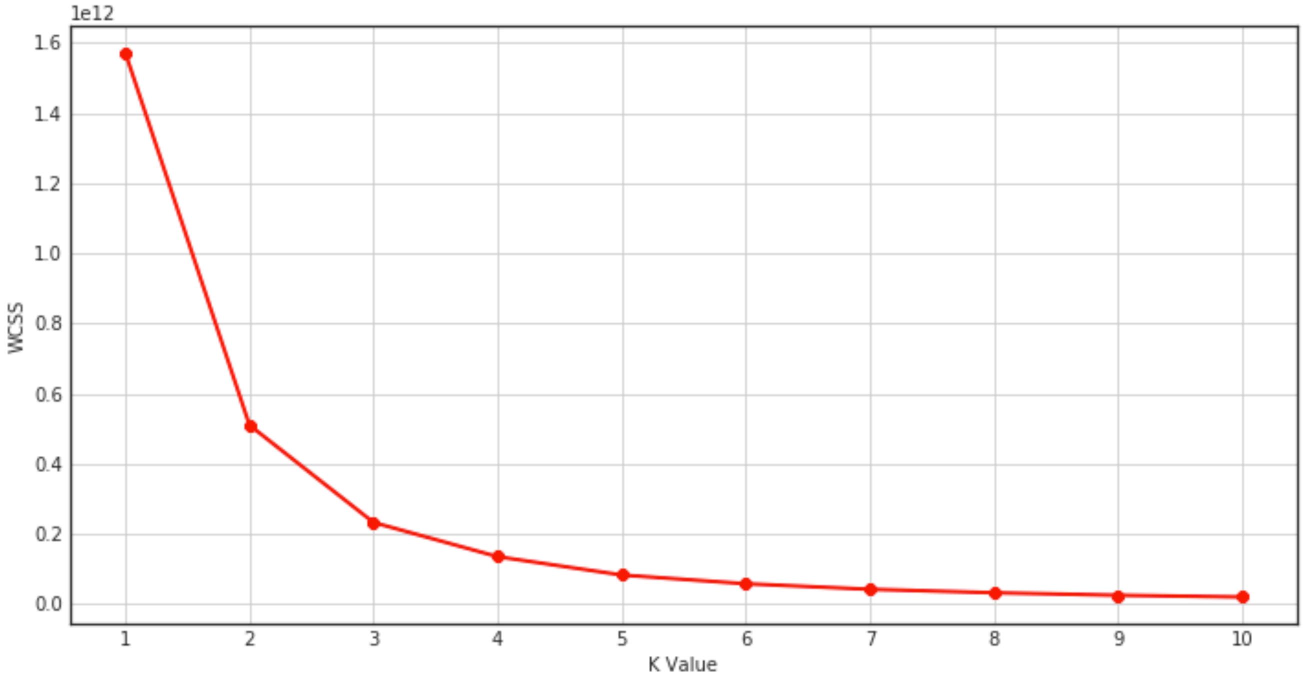


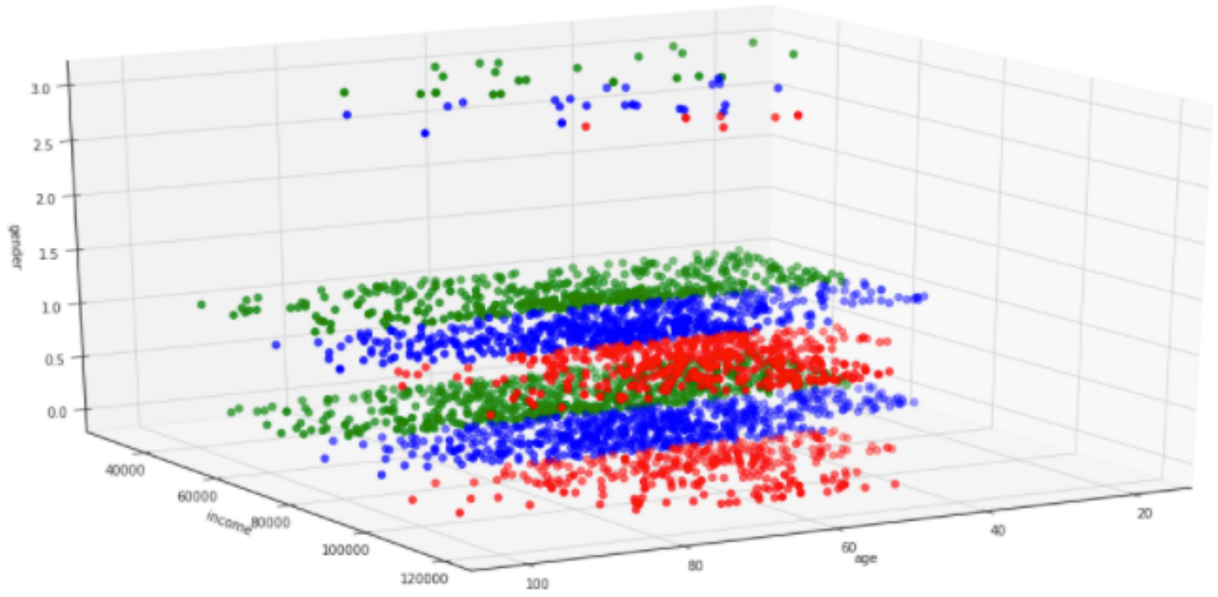
```
In [83]: gender = dfm2_discount.gender.value_counts()
         sns.set_style("darkgrid")
         plt.figure(figsize=(10,4))
         sns.barplot(x=gender.index, y=gender.values)
         plt.show()
```











Justification

KMeans algorithms is the most suitable for clustering data for this project because there are lots of data to processed. On each offer type there are more than thousand list of data. If we used DBScan, there will be overemphasize on gender because that variables are densely clustered.

Conclusion

Free-Form Visualization

From the graph above we can see that there are 3 cluster for each of offer type, bogo and discount. For the bogo and discount each cluster is:

a. Customer with income between 40.000 and 60.000 USD;

b. Customer with income between 60.000 and 80.000 USD;

c. Customer with income between 80.000 and 100.000 USD.

Reflection

This project is challenging and to cleaning the data itself is the most struggling part. Finding the column that correlated to do clustering. The next thing to do is clean the irrelated value from each column by specific condition. Last step to do is dividing datasets into each offer type. From that specific offer type dataset then we started to use the algorithm, KMeans to do clustering and elbow methods to find the fittest number of k.

Improvement

Further improvement that could be make might be visualize the result data into diagram and filter out more data that might not related but still caught in the formula.