Capstone Project Oluwanifesimi Ademoye

Machine Learning Engineer Nanodegree June 13th, 2021

Starbucks Project (Customer Segmentation)

Definition

Introduction

Customer Segmentation involves dividing a customer base into similar groups based on certain attributes the customers have. Companies and organisations perform customer segmentation to analyse, draw conclusions and make business decisions concerning their customer base.

Algorithms such as K-means clustering algorithm is very popular with customer segmentation and t fare well with clustering customers according to their RFM (Recency, Frequency and Monetary) attributes. Another well used algorithm is the Agglomerative Hierarchical clustering algorithm which can be applied to transactional data of customers.

In this project I focused on performing customer segmentation on the customer base of the American beverage company, Starbucks.

Customers were grouped based on demographic traits such as age, income and gender and their behaviour towards types of offers sent from the company. To perform the segmentation, 3 datasets containing the demographic information of customers, offer types and customer transactions were used with the Gaussian Mixture Models (GMM) clustering algorithm. Each cluster was analysed to view similarities

Problem Statement

Starbucks would like to how certain customers react to offers sent to them. This project aims at identifying the groups of individuals that are responsive to these offers by performing the following tasks:
1. Download and Explore the data
2. Data Cleaning and Exploration
3. Feature Engineering and Dimensionality Reduction
4. Clustering the data using GMM
5. Selecting the best clustering algorithm
6. Extracting the trained model attributes and visualizing clusters
7. Stating the observations of each cluster

Every customer is assigned to a cluster. These clusters have attributes that is used to analyse and make decisions on the customer-offer relationship.

Analysis

Data Exploration

Three (3) datasets were used in this project.

1. profile.json: This contains demographic data about the rewards program users. There are 17000 users and 5 fields which include:

- gender: (categorical) M, F, O, or null

- age: (numeric) missing value encoded as 118

- id: (string/hash) The customer id. Same values as those in the 'person' feature of the transcript.json dataset.

- became_member_on: (date) format YYYYMMDD

- income: (numeric)

This dataset contained 2175 missing values in the gender and income features. This was taken care of using imputation.

Also since customer segmentation is the task at hand, profile.json dataset formed a base for the final dataset to be used.

2. portfolio.json: This contains the offers sent during the 30-days simulation period. There are 10 offers and each has 6 fields which are:

- 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)

There were no missing values in this dataset.

3. transcript.json: This is the transactional data that shows the events such as when a user views, receives or completes an offer. It contains 306648 events and 4 fields which are:

- 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 the start of test

There were also no missing values in this data.

To form a more detailed dataset, the transcript.json dataset is merged with the profile.json dataset on the 'id' and 'person' features. Then more relevant features are created and a group by operation is performed on the dataset. This is explained in more detail in the Data preprocessing section.

These datasets were provided by Udacity.

Solution Statement

We were not given any labels or ground truth that would enable us to use supervised learning models, we will be exploring unsupervised learning methods amongst our data to determine potential strategies for adjusting the Starbucks Rewards program given our customer insights.

We'll be leveraging hierarchical modeling to cluster our data into a few respective customer segments for analysis.

Evaluation Metrics

The average Silhouette score is used as a measure of accuracy of clustering. The silhouette score indicates how close a sample is to its own cluster compared to other clusters. The best value for a silhouette score is 1 and the worst is -1.

$$\text{silhouette score} = \frac{(b - a)}{\max(a, b)}$$

Where a is the mean intra-cluster distance and b is the mean inter-cluster distance.

The silhouette score is used to measure the performance of each of the clustering algorithm by calculating the average silhouette score of the samples in clusters formed by the algorithm. In other words, in this project it is used to compare the performance of the model used in the project and the benchmark model.

Also, we will be leveraging the elbow method of determining k-means clusters through a simple function that will iterate through a number of K-Means clusters and displaying the Sum of Squared Errors (SSE) in visual form. This in conjunction with the silhouette coefficient will idealize the number of clusters for our final algorithm.

Algorithms & Methodologies

It's important to explain some of the algorithms I used to throughout the project and why I used them. We will be deeply examining the data and its attributes to tell how why it has been clustered that way. Due to the reason that we do not have a "ground truth" of what is necessarily right or wrong, we will have to leverage unsupervised learning algorithms. I specifically talked about three of the algorithm even though I only made use of two of them in the project.

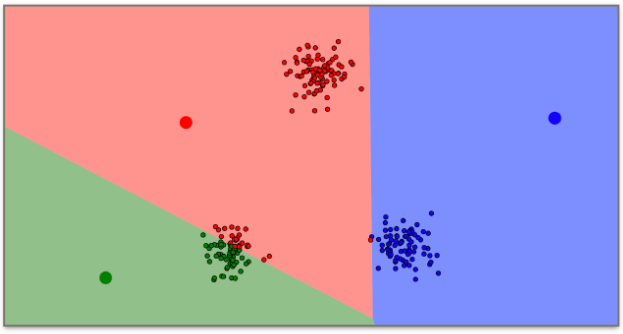
(You can check out this blog for helping me create the explanatory visuals below across K-Means

and DBSCAN: https://www.naftaliharris.com/blog/visualizing-k-means-clustering/)

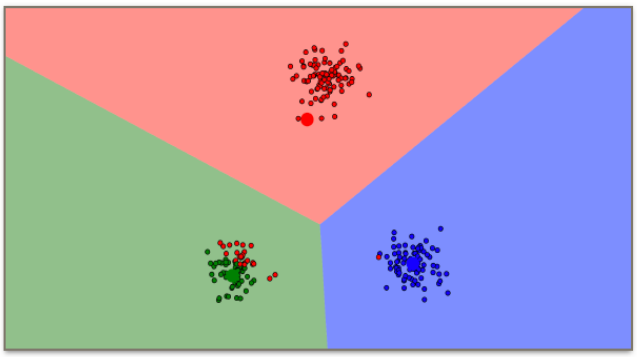
K-Means Clustering

K-Means works by initializing a number of centroids that serve as sort of magnets to attracting nearby clusters of data around them. These centroids are determined by us as the user. For example, in scikit-learn's "KMeans" algorithm, we initialize how many centroids there will be by passing in an "n_clusters" parameter.

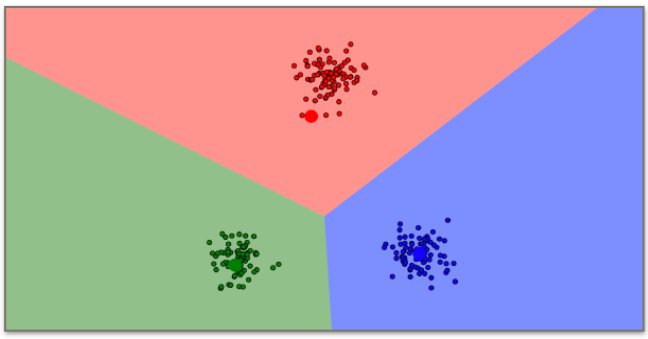
Each time the algorithm steps through an iteration, each respective centroid moves closer and closer toward the "center" of the clustered data. I've visualized this with some very basic data in the three screenshots below. Notice that the initialized centroids do a pretty poor job at clustering the data in the first pass, then do a slightly better job in the second pass, and then finally do a great job in the third pass.



First Pass



Second Pass

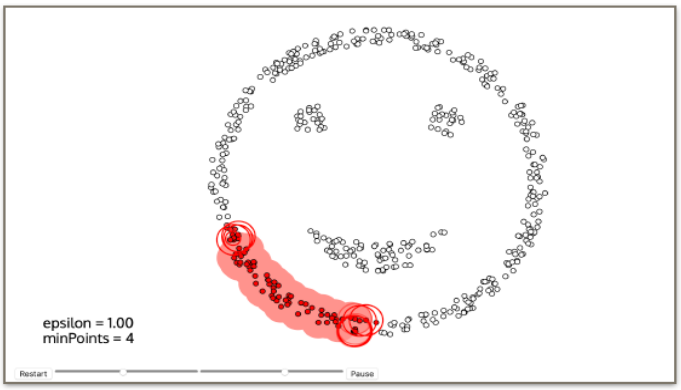


Third Pass

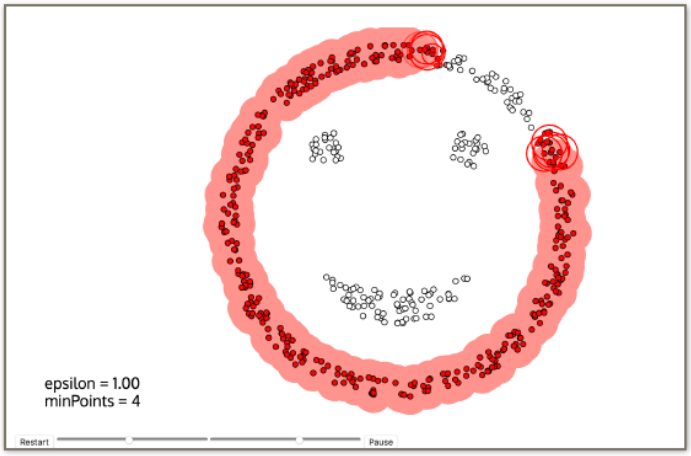
This example here is a radical oversimplification, and it happens to work out nicely that the clusters are so neat and tidy but this definitely won't be the case with most datasets. In fact, look at what happens when we try to apply K-Means to something like this smiley face set of data, even after several iterations of the algorithm.

As you can see ,K-means does not seem to be enough therefore we will look at other alternatives

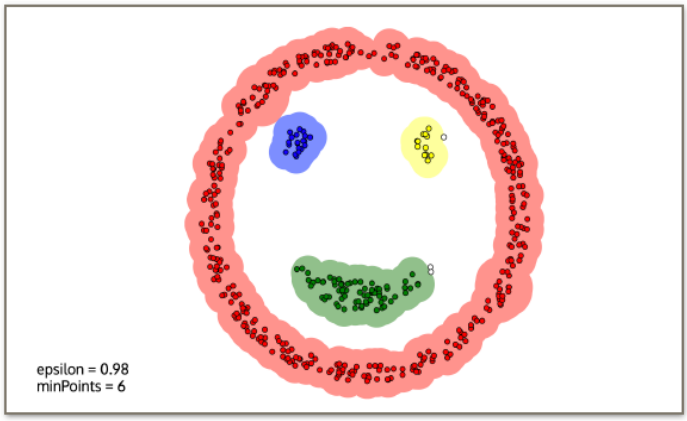
DBSCAN
DBSCAN combs (or scans) through our data by randomly selecting a starting point in our dataset and then branching out to other nearby data points. It'll continue along its current path of clustering so long as the epsilon and minimum point parameters are satisfied. The epsilon value is best thought as the "radius" of how big you want the previous data point to be away from the next one, and the minimum points describe at what minimum number of points should be captured along the way. If these parameters are not satisfied, DBSCAN simply jumps to another random place in the dataset and begins again. This is best described visually, again using our smiley face friend. (Try not to have nightmares with the last one.)



DBSCAN after ~5 seconds



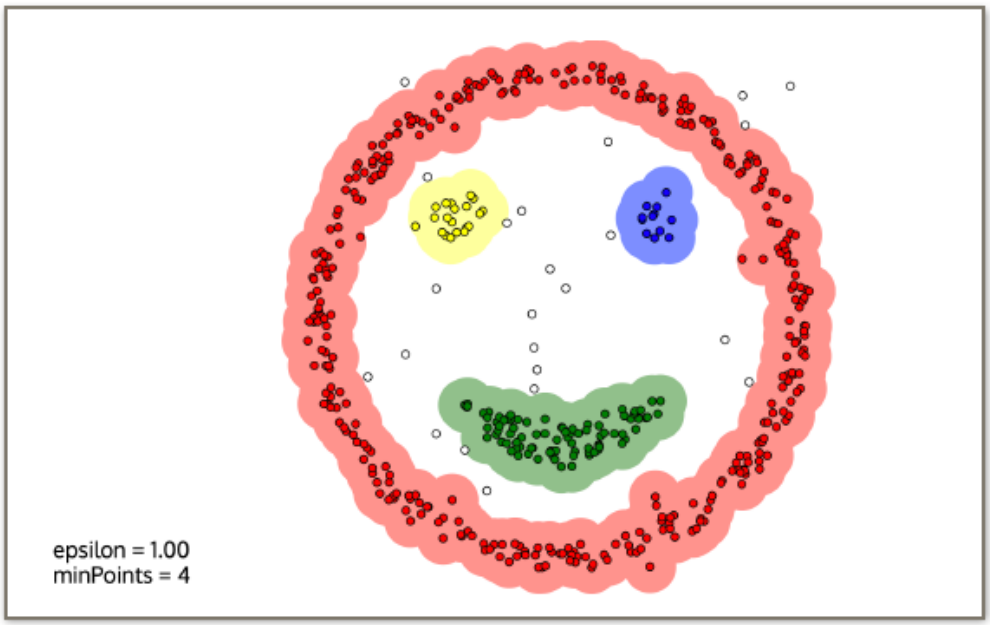
DBSCAN after ~20 seconds



DBSCAN in the end

As can be seen , DBSCAN seems to be a better choice as an algorithm with the visualization above. Now you might wonder why if dbscan did so well, why we aren't using it for out project. Well you have to keep in mind what DBSCAN stands for: Density-Based Spatial Clustering of Applications with Noise, which means that DBSCAN works really well on datasets that you know will have noise, but given how concise we're going to get our our data in this project, I'm not sure will be any noise at all.

Perhaps one final DBSCAN visual will help to illustrate this best.



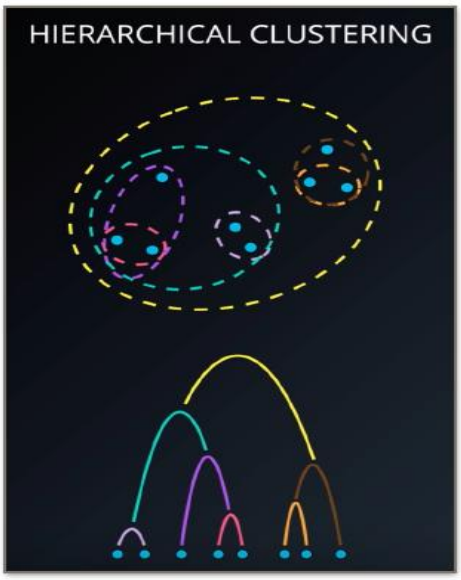
DBSCAN on pimpled smiley face

Here, we've again applied DBSCAN on a "pimpled" smiley face. Notice that these "pimples" were left out of all clusters entirely. Now, if we were to perform a DBSCAN on our dataset, we might be losing out on valuable insights that were incorrectly classified as noise. That said, let's round out this section with a brief discussion on hierarchical algorithms.

Hierarchical Algorithms

Hierarchical clustering seems similar to DBSCAN in practice but is radically different under the hood. Where DBSCAN clustered elements by sweeping through with epsilon values, hierarchical clustering automatically begins with the underlying assumption that every single data point is already its own cluster. (That said, no data point gets left behind in a hierarchical

method.) From there, the hierarchical clustering then leverages different forms of linkage to determine how to best cluster the data. Here's a very simple visual of how this might look in general.



You'll see in the visual a sort of "tiering" amongst the clustering. At the very bottom, you see that, as noted above, every data point is clustered as its own cluster. From there our hierarchical cluster then moves up the hierarchy to determine the next best clustering of our data. Eventually, we keep moving up and up the hierarchy until eventually everything is clustered together in one massive cluster, as indicated by the yellow circle.

Of course, we don't want either end of the spectrum. We don't want a million little clusters, nor do we want one giant, massive cluster. Instead, we want just the right amount of clusters, and that is defined by us within the hyperparameters of our model. We'll see how this shakes out for our own project further on down.

Initial Cleansing

The data in its initial form is decent, but we will need to clean it up some in order to best leverage it for our unsupervised model later on in the project. Specifically, I cleaned up the initial datasets and then later combined them to form a master dataset that we'll be regularly working from in the remainder of the project. We'll discuss about that initial cleansing here and talk more about that latter preprocessing down in another section.

Portfolio Clean Up

• Changing the column name from ‘id’ to the more descriptive ‘offer_id’ since the id column is present in our other datasets

• One hot encoding the ‘offer_type’ column to work well with our algorithms later

• Separating and one hot encoding the ‘channels’ column to also work with our algorithms later

• Dropping the ‘offer_type’ and ‘channels’ columns now that they are one hot encoded in other columns

Profile Clean Up

• Dropping rows with null information

• Changing ‘id’ column to ‘customer_id’ name

• Changing the ‘became_member_on’ column to a date object type

• Calculating number of days that a person has been a member as a new ‘days_as_member’ column (as of August 1, 2018)

• Creating new ‘age_range’ column based off ‘age’ column

Transcript Clean Up

• Changing the name of the ‘person’ column to ‘customer_id’

• Removing the customers that are not reflected in the ‘profile’ dataset

• One hot encoding the ‘event’ values

• Changing the ‘time’ column to ‘days’ along with appropriate values

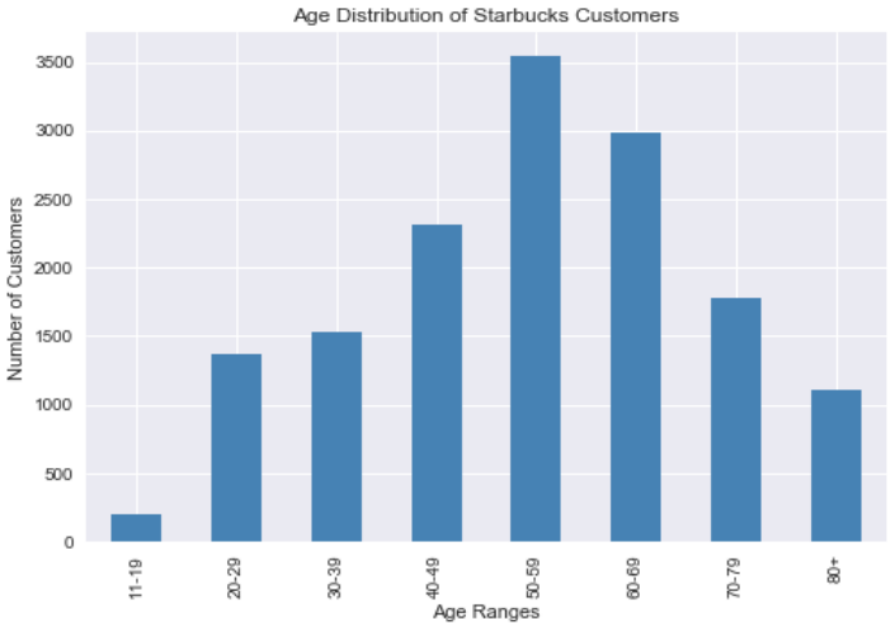
• Separating value from key in ‘value’ dictionary in order to form two wholly separate datasets: transcript_offer and transcript_amount.

Exploratory Data Analysis

Since we have done an initial analysis and cleaning. We are going to do a more formal analysis to see how we might need top pre-process our data even further down in formal data pre-processing. We will break down what the data actually refelcts

Q1: What are the general age ranges of our customers?

Given that young people anedoctally are known to consume more coffee and are enamoured by the starbucks brand, i have a suspicion the trend will be skewed in their directions.



A1: The actual age distributions of Starbucks customers.

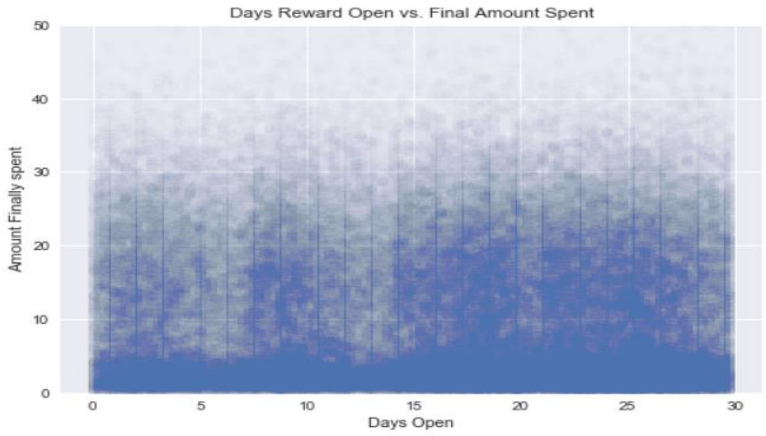
It would seem like that from the data shows that the largest purchaser seems to be people in their 40's , 50's and 60's range which probably means that the starbucks have become an hangout spot for those close to retirement or already retired. This speaks to the fact that the older one gets , the more likey they are able to afford starbucks high steeped prices of coffee,This might be another reason.

Q3: What does the correlation between number of days an offer has been open vs. final

Transaction amount have to tell us?

The correlation might be that since an offer has been open for an extended period of time, The customer might not feel particularly that excited about it. A More enthuastic customer would have claimed the reward earlier, for fear of missing out on the reward. Lets move on and see

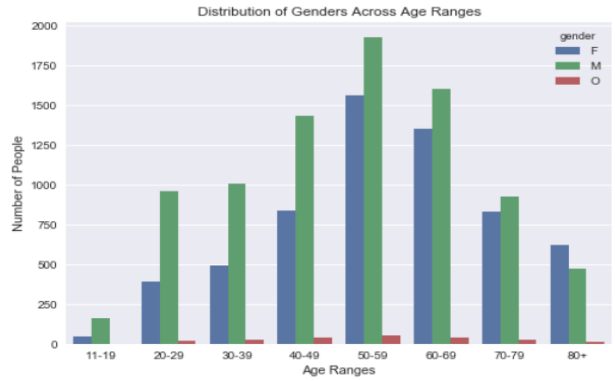
what the data has to say about this concept

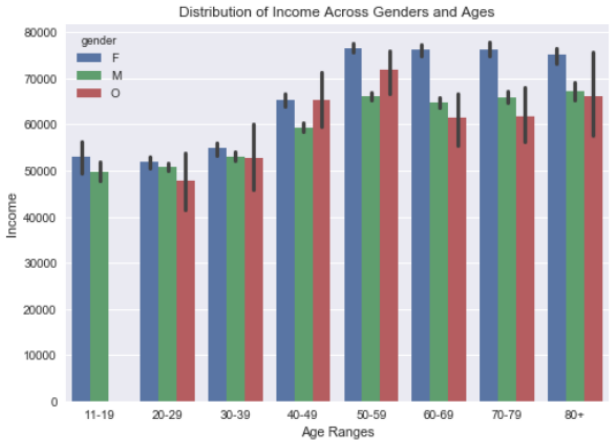


A3: Analysis of Amount Spent vs. Days Open

There seems to be no correlation between the amount spent and the days the offer was left open. This is clarified but the two scatter plots which shows there were no correlation between the two

Q4: Do gender distributions have any major effect on our data here Lets take a dive into how gender might have an effect on the videos. Our dataset have three indistinct genders: Male, Female and other



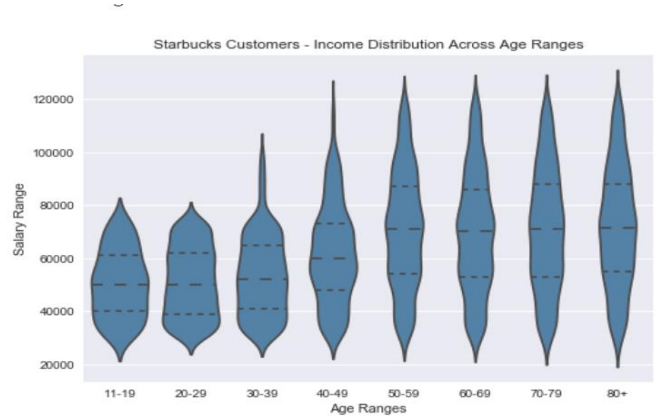


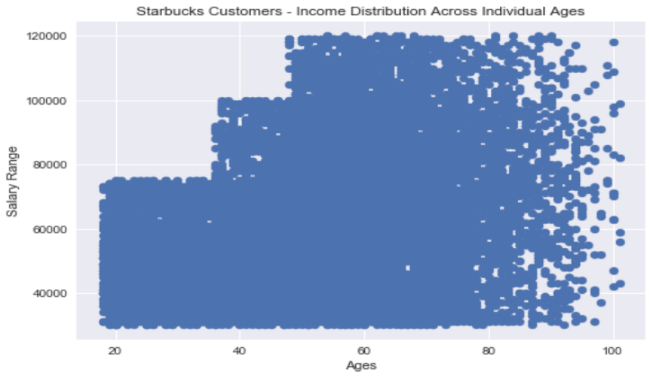
A4: Analysis of gender across our customer data

The first plot did not surprise me at all, i had a feeling that since it's the general sentiment that men made more money, then its only normal the had more income, However the second plot nullifies that point and shows that women had higher income on average across the age ranges, even though there was not much disparity between the income ranges. I was really surprised by this plot prediction

Q4: What are the salary ranges of people across different age groups

I wanted to see how the salary ranges of these various age groups might affect how often a person visits Starbucks and use their rewards program. My hypothesis is that the older you are the higher you earn and are more likely to buy more.





A4: Analysis of Income Distribution

When visualizing the data with the violin plots, My hypothesis was confirmed that older customers definitely tend to make more money; There seemed to be an hard cap at 80,000 on the salary range for young people while for old people the hard cap its 120,000. I think the problem is that i know people in their 20's and 30's don't have that salary cap so that might be as a result of the data gathered.

Data Preprocessing

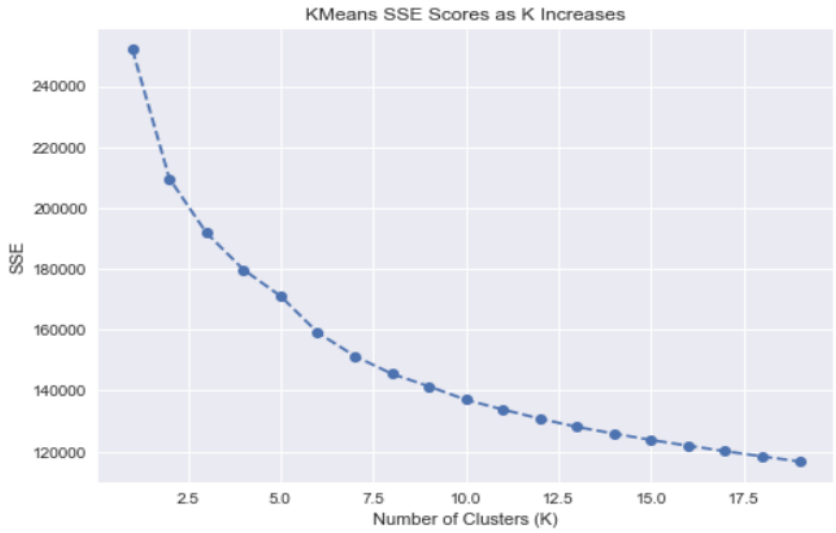
With initial analysis / clean up and EDA under our belt, let's move on into formalizing the master dataset I will work from in the remainder of the project. We are going to take several things in mind to engineer some new features that I feel will be helpful when we actually move toward running our unsupervised algorithms. Here are the features we will leverage as part of this master dataset:

• customer_id: The unique customer identifier

• age: The age of the customer
• age_range: The age range the customer falls into
• gender: The gender of the customer, either male (M), female (F), or other (O)
• income: How much money the customer makes each year
• became_member_on: The date that the customer because a Starbucks Rewards member
• days_as_member: How many days that the customer has been a Starbucks Rewards member
• total_completed: The total number of offers actually completed by the customer
• total_received: The total number of offers that Starbucks sent to the customer
• total_viewed: The total number of offers that the customer viewed
• percent_completed: The ratio of offers that the customer completed as compared to how many offers Starbucks sent to the customer
• total_spent: The total amount of money spent by the customer across all transactions
• avg_spent: The mean average amount of money spent by the customer across all transactions
• num_transactions: The total amount of individual monetary transactions performed by the customer
• completed_bogo: The number of completed BOGO offers by the customer
• num_bogos: The total number of BOGO offers sent to the customer by Starbucks
• bogo_percent_completed: The ratio of how many BOGO offers were actually completed by the customer as compared to how many Starbucks sent them
• completed_discount: The number of completed discount offers by the customer
• num_discounts: The number of discount offers sent to the customer by Starbucks
• discount_percent_completed: The ratio of how many discount offers were actually completed by the customer as compared to how many Starbucks sent them

Number of Clusters

Finally, determining the number of clusters we'll leverage is super in the final analysis of our project. In order to fine tune this parameter, we will perform the elbow method and run a series of silhouette scores. When running these, here is the outcome we get:




```
For n_clusters =  2 The avg silhouette_score is : 0.34563481440633215
For n_clusters =  3 The avg silhouette_score is : 0.36225910479663387
For n_clusters =  4 The avg silhouette_score is : 0.25964869765424275
For n_clusters =  5 The avg silhouette_score is : 0.1690494000201967
For n_clusters =  6 The avg silhouette_score is : 0.16426608035740384
For n_clusters =  7 The avg silhouette_score is : 0.15331092053578937
For n_clusters =  8 The avg silhouette_score is : 0.1468250309154469
For n_clusters =  9 The avg silhouette_score is : 0.14906760038942352
For n_clusters =  10 The avg silhouette_score is : 0.14660499296752108
For n_clusters =  11 The avg silhouette_score is : 0.14673761337162686
For n_clusters =  12 The avg silhouette_score is : 0.1385323001091669
For n_clusters =  13 The avg silhouette_score is : 0.13767789144230955
For n_clusters =  14 The avg silhouette_score is : 0.11906444664195084
For n_clusters =  15 The avg silhouette_score is : 0.11712040302488348
For n_clusters =  16 The avg silhouette_score is : 0.11346082071144224
For n_clusters =  17 The avg silhouette_score is : 0.11590318664862236
For n_clusters =  18 The avg silhouette_score is : 0.1145853371831212
For n_clusters =  19 The avg silhouette_score is : 0.11383930385524677
```

```
Number of Clusters We Will Use: 4
Given both pieces here, we'll go ahead and leverage 4 clusters. Both the
sihouette score and elbow method started showing diminishing returns follo
wing 4 clusters
```

Given this information, I've settled on leveraging 4 clusters for our final algorithm. Transparently, I could have gone as high as 9 given the diminishing returns, but I felt 9 clusters would be too many for the purposes of this project. Four clusters is a reasonable amount to comb through in the final evaluation of this project.

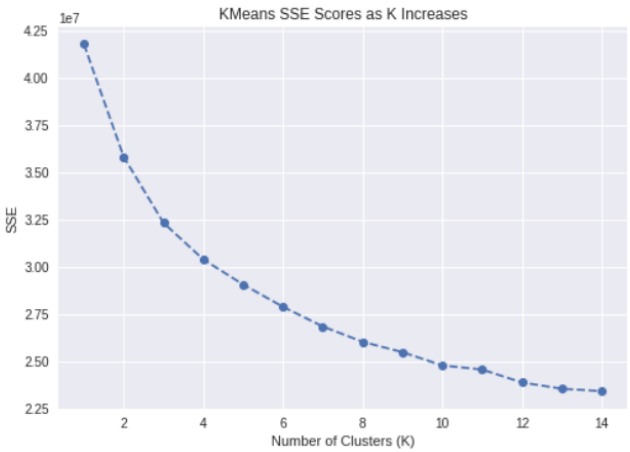
Machine Learning Modeling

Now that we've spent some time refining our model, let's go ahead and utilize our insights to run our model once again for more optimal results. We'll review these results in a following section.

Benchmark Definition & Comparison

Before we move on, I want to discuss the benchmarks and metric evaluation since these came Heavily into play in the prior section. Given that there isn't necessarily a labelled right or wrong to the provided dataset, we can't really objectively evaluate how well our unsupervised dataset performed after it has already processed through the data. What we can do, however, is leverage our benchmark and metrics to determine the ideal number of clusters for the final algorithm.

We already explored leveraging the elbow method and silhouette score , so we also explored u utilizing a very similar elbow method in this project, here is what the results were from there:



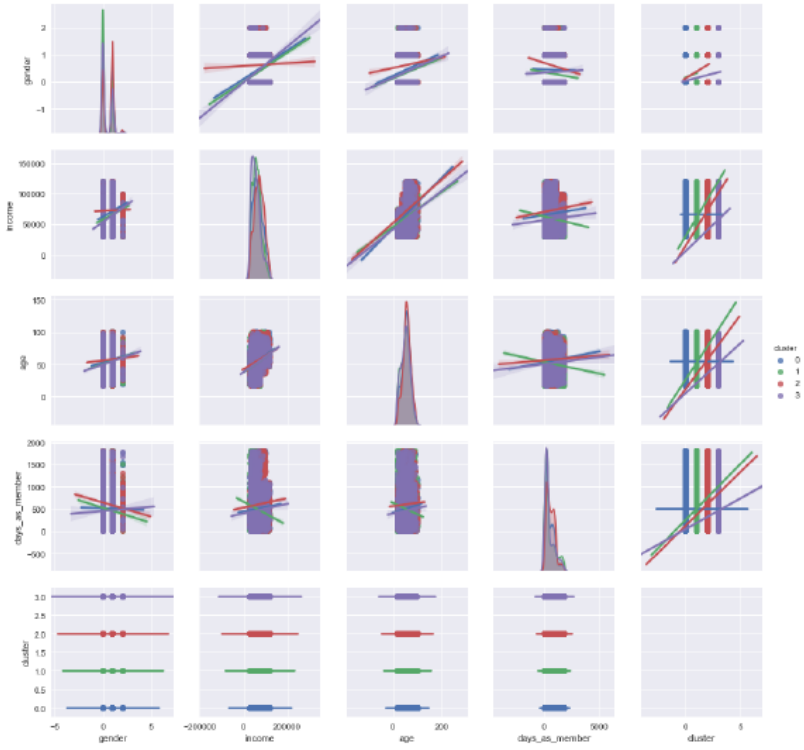
Final Analysis & Evaluation

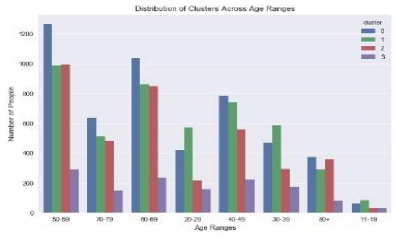
Now that we've gathered our various clusters from our hierarchical algorithm, let's go ahead

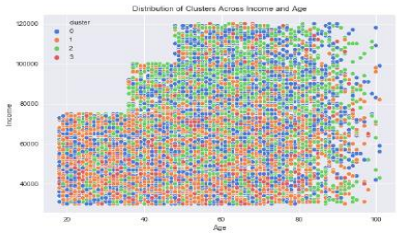
and visualize the results! Like we did in the Exploratory Data Analysis section, we're going to explore two more high level questions and how these might be utilized by Starbucks to adjust their rewards program.

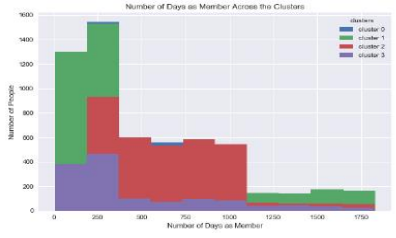
Q5: What personal attributes of our customers are defined throughout each of our clusters?

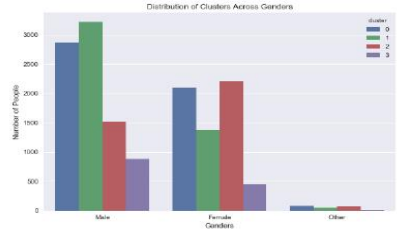
Firstly, we will dive into the personal attributes of our customers as clustered by our algorithm. We'll visualise this in a PairPlot and make the diagrams very easy to understand.











Analysis of Clustered Personal Attributes

There are lot of insights to glean from this visualisation. Let’s cover each of this clusters with clusters below

Cluster 0

This is the largest cluster, cluster 0 seems to consist of older people with higher incomes. As shown by the countplot with the age ranges, the ages of these people tend to falls into that 50 to 80 year old range. It's actually very common to see the same sets of people have some of the higher income ranges. Gender was relatively split with men accounting for slightly a little more, This might have to do with the fact that the original dataset collected more data with men actually. Although this cluster was the biggest,it did have a lot of discrepancies especially with income and that makes it unreliable for future inferences.

Cluster 1

This cluster would seem like the cluster filled with young people. Looking at the age range distribution, we see the strongest distribution here amongst the 20-40 year old. There seems to be a significant gap between males and females in this cluster. The people in this cluster also seem to fall in the low income range. And as far as number of days as member goes, this cluster's distribution is very similar to that of cluster 0.

Cluster 2

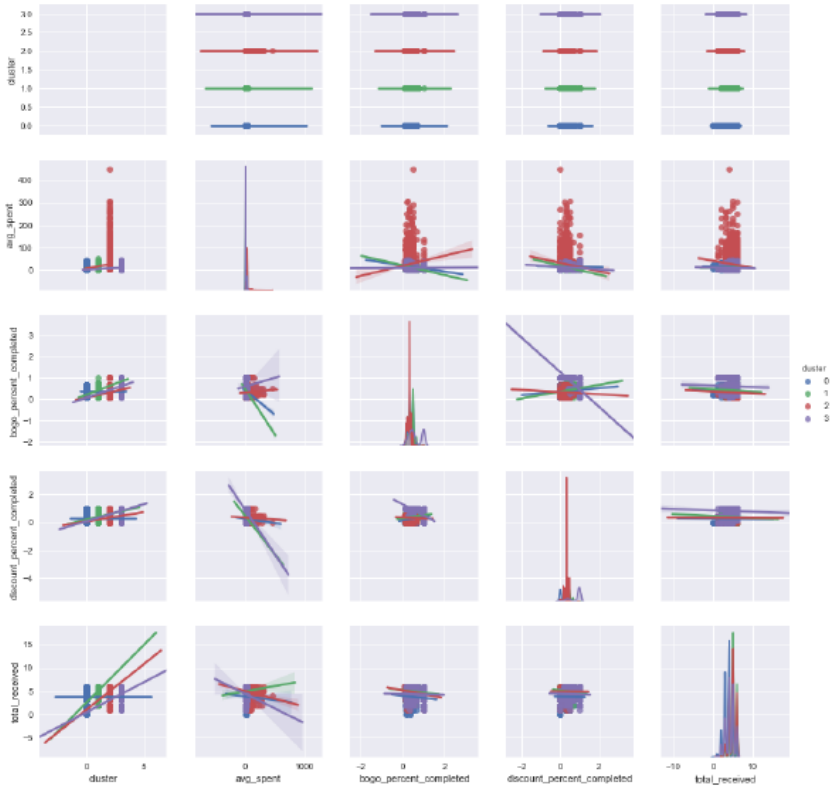
There seem to be more females than males in this cluster and the cluster showed the largest distribution for Starbucks reward . it was not surprising that income tended to be higher for this group given from our initial analysis that men seemed to make more than women.

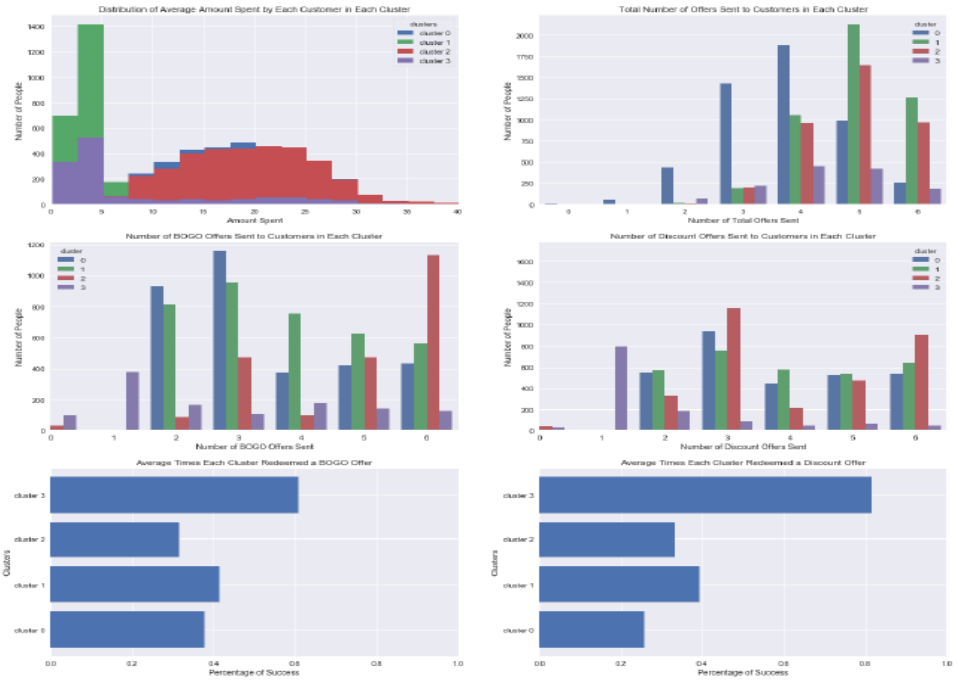
Cluster 3

This cluster is the smallest of our four clusters, yet it is very similar to cluster 0 in a few ways. Basically the distribution of days as member and gender are fairly similar. Perhaps the exception that separates it from cluster 0 is that there tended to be more people in the younger age range, and I suppose that would make sense given that younger people consisted of a smaller subset of the original data provided.

Analysis of Clustered Behavioral Attributes

We dive into a deeper perspective of the behavioral attributes of these customers





A6: Analysis of Clustered Behavioral Attributes

We dive into a deeper perspective of the behavioral attributes of these customers

Cluster 0

Firstly, This cluster on average seems to be the biggest spenders, with an average transaction total peaking at ~$19. This amount is really high, This is actually in tune with our previous observation due to the fact that this particular cluster contains our older customers, which we have established that have an higher income. Also more likely than not this cluster probably have families and these customers are buying coffee for multiple people. Considering all these it makes sense the average number makes more sense.

This cluster has the lowest yield of using discount offers and pretty close to last for BOGO offers. This might make more sense because these clusters might be made of older customers who might not take full advantage of the BOGO offers being sent out electronically. Also this might be that Starbucks does not see any interest in trying to appeal to groups that historically don't take advantage of their offers .

Cluster 1

We might recall that this cluster is more of a younger nature and therefore we see the average amount spend towards the lower end. However We see that this particular group utilises the offers and take advantage of the offers for Bogo more that the discount offers . The success levels for both are 40% and this might because the difficulty level to claim these rewards are high

Cluster 2

This cluster is predominantly female as established earlier and have an high spend amount, much the same as cluster 0. This cluster makes use of the offers especially the Bogo offers. I am not sure particularly why this is but personally i think there's a lot of are of improvement for this cluster given the demographic information of this group

Cluster 3

This cluster was the hardest to read of all clusters because even though the group's demographic aligned to that of Cluster 0's it tended to have a lot of younger age range. This group has a high of a success rate with offers, especially with discount based offers. Even though this is the smallest of the clusters, I would personally recommend that Starbucks leans more into this cluster to discover what is causing the level of success.

Other Approaches

Before wrapping up our project here, I want to touch quickly on the fact that we could have easily chosen to go other routes when completing this project. Here are two very quick

Summarizations of what we could have done.

Supervised learning methods: While the dataset itself did not necessarily pose any right or wrong answers, we could have engineered some features that labelled our dataset as such based on things like if an offer was completed or not. This seemed to be way too messy and disingenuous to be considered a viable route. Messy because it's hard to classify success versus non-success, and disingenuous because it is really relying upon the modeler's best judgment to determine what that success versus non-success even begins to look like. For those reasons, I steered away from supervised learning methods.

Conclusion

A lot of improvements can be done to boost the clustering of this data. This could be done by better collection of Data because the initial data seems flawed and skewed thereby making the clusters meaningless