Capstone Project Andrew Ferrell

Machine Learning Engineer Nanodegree January 2020

I. Definition

Overview

All businesses have faced issues keeping customers engaged with their products. Over time this problem has been addressed in a variety of ways, from advertising to discounts. It's usually easy for small businesses to notice how well these methods work, but much harder for larger businesses with a diverse customer demographic.

This is where machine learning can help address making sense of and analyzing the large amounts of data generated by businesses. Specifically, how some customers might respond to certain special offers.

This project demonstrates some ways to find useful insights into modern business data collected and generated each day. While the dataset is simulated data, the format is realistically presented and the methods remain valid.

Problem statement

The goal is to make sense of the large amount of disorganized data. Specifically, to determine which demographic groups respond best to which offer type(s). This can help create more successful future targeted campaigns.

This sort of problem has traditionally been addressed with normal statistical analysis. While that is still a major aspect of the problem, the goal is to show how certain machine learning algorithms can help automate this process. Especially when the number of data points to consider become significantly larger.

After all, it doesn't take an unsupervised machine learning algorithm to determine that women are more likely to buy dresses than men. (In the vast majority of cultural norms at the time of writing this…) It might make it easier and faster to determine; however, which women prefer which aspects of dresses on a larger scale based on aspects simpler statistical analysis might miss.

Metrics

Since this project will limit the machine learning to unsupervised algorithms to help make sense and group the large amount of data, there are not any clearly defined metrics to be had.

The success or failure of the project can be seen if the algorithms find any groupings of customer demographics to app transactions and offer completions that are not apparent within the initial data exploration phase.

II. Analysis

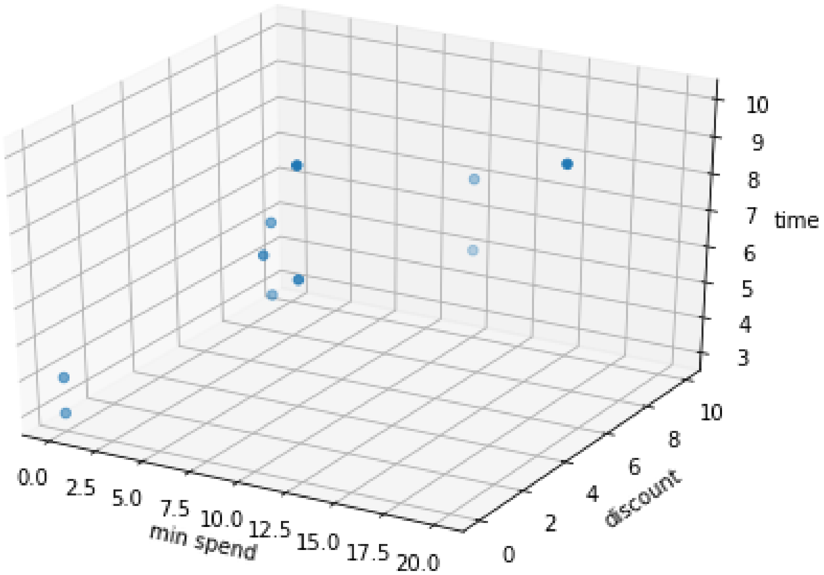
Data Exploration

The dataset comes in the form of three data files. One file contains information about certain promotional offers. Another contains information about customer demographics, while withholding personally identifying information. The third contains information about the offers: when they were received, viewed, and completed.

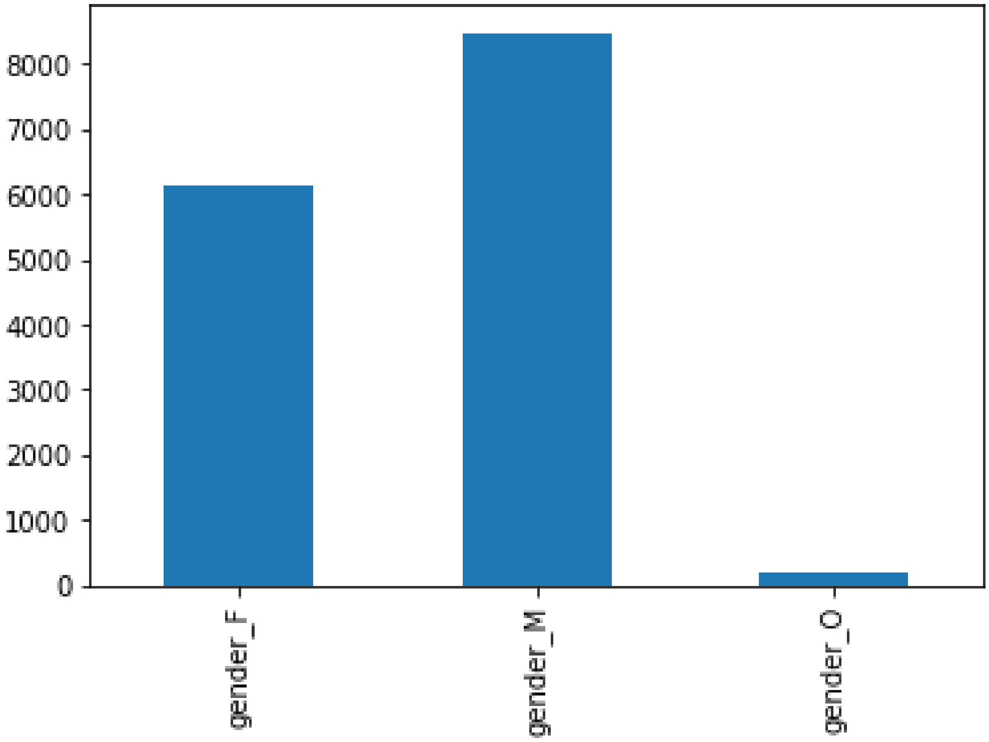
- Profile – demographic information about the customers:
  - Age - in years (111 if didn't answer)
  - Date joined/became a member on – YYYYMMDD format
  - Gender of the customer – male, female, other, or didn't answer
  - Customer id – non personally identifiable, but a way to link to transactions
  - Income – in USD
- Portfolio – information on the promotional offers given to the customers
  - Offer id – way to link offer information to other data tables
  - Offer types – the type of promotion (informational, BOGO, discount)
  - Difficulty – the minimum amount of money required to spend to receive the award
  - Reward – the amount of money user "gets" by completing the offer
  - Duration – length the offer is good for, in days
  - Channels – the way the offer was sent to customers (email, mobile, social, web)
- Transcripts – information about the offers and money spent
  - Customer id – matches id from profile
  - Time – hours since promotion started that the event happened
  - Value – more information depending on event type
    - Offer viewed/received – offer id to link to portfolio data
    - Offer completed – offer id and reward
    - Transaction – money spent by the user at that time

Visualizations

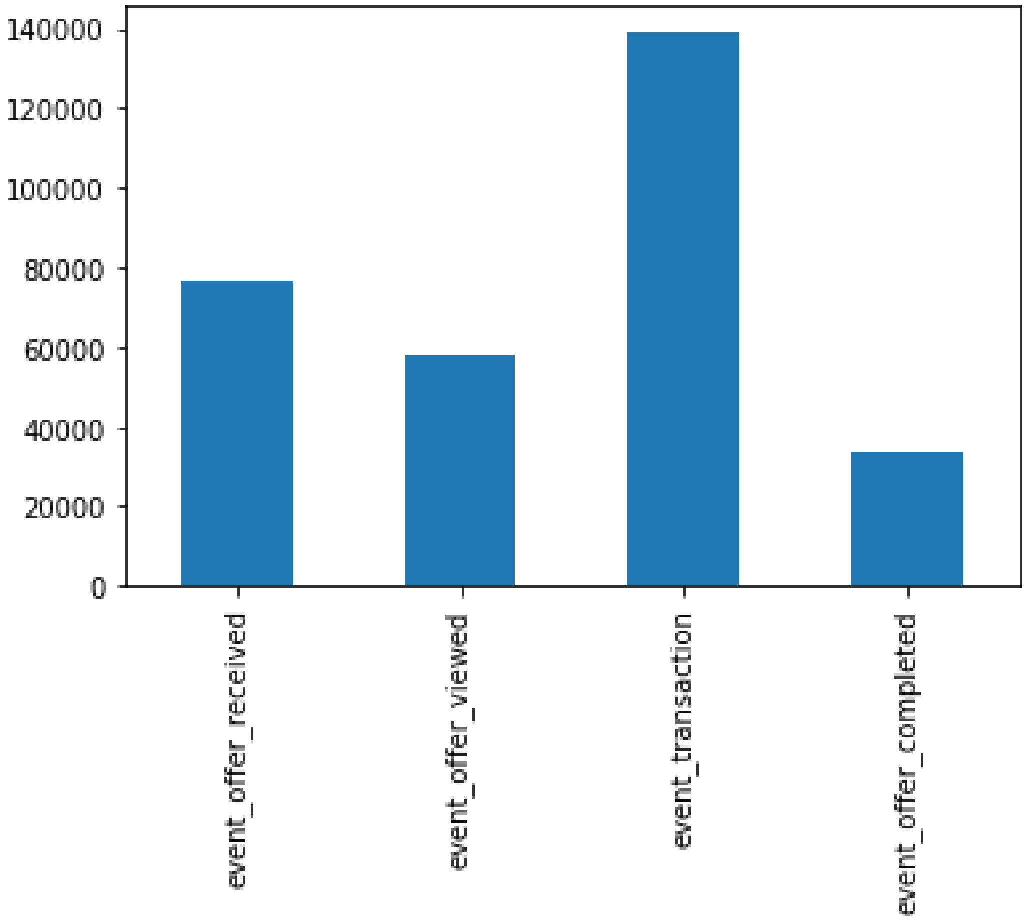
From initial graphs of raw data, we can see some things. For example, a 3d scatter plot of the minimum amount of money required to spend, the discount received for the offer, and the duration (in days) the offer was available for here:



From the initial demographics, we can tell the number of customers (members of the promotional program) in certain demographic groups. For instance, the number of customers who chose a preferred gender type:



Looking at the number of types of each type of customer interaction, we can see a lot more transactions than other events in the following graph:



This is probably due to all transactions being recorded independently of promotional offer events. While this section of data can be very useful to understand spending habits of different demographic groups, we want to determine factors of successful promotions as defined in this project's scope.

Algorithms and Techniques

Principal Component Analysis (PCA) and K-means are standard algorithms to begin with most unsupervised learning projects.

PCA together with explained variance can help determine which features best explain the correlation of all the data points within a collection.

K-means helps determine the best groupings of the data based on the number of components chosen for the algorithm. It does so by randomly assigning different cluster centers or centroids and slowly changing these centroid values until the average distance of all the points within its cluster is minimized. The "best" number of centroids (k) is the smallest number of components that still greatly minimizes the average distance to each centroid.

Benchmark

Determining the top components from PCA can be achieved by measuring the explained variance of these components with the following equation:

$$\frac{\sum_n s_n^2}{\sum s^2}$$

In this equation n are the number of principal components chosen, while s is the singular values of the components. By running this equation, we can determine the number of components that explain most of the variance among the data. The goal is to reduce the number of components while still maintaining a high percentage value. 80% or above is usually considered a good benchmark.

In order to determine the best k value in k-means, the algorithm will have to be run multiple times, each recording the average distance to the centroids and then plotting that information in a line graph to determine the best value for k.

III. Methodology

Data Preprocessing

The data is split into three datasets. Each one contains some information that will need to be converted, separated, or dropped as all machine learning algorithms can only run on numerical data.

The first step will be extracting and then one-hot encoding the categorical data into separate feature columns. This includes the offer types, the channels they were sent on, the gender of individuals, and the event type from the transcripts. This will also have to be done on the ‘value’ part of the transcripts as each contains a different value depending on the event type.

The next will be combining the data into one dataset to link offers and customers with the information within the transcripts. This can be achieved by joining the sets based on matching the customer and offer ids from each dataset.

Finally, the data will have to be split into separate sets. This is necessary for analysis because while the transactions include customer ids for demographic data it doesn’t include offer ids to link the purchases to a specific offer. Offers received and viewed don’t include a reward (discount) the customer received.

After this lengthy data exploration and preprocessing, I concluded that gaining demographic insight into offer completions was the best dataset to run calculations on. This is due to the problem statement wanting to know which users respond best to which types of offers.

Furthermore, I decided to drop all data where the users decided not to include their demographic data due to the problem statement's scope. (*Although, it could be useful to determine which offers users went for who chose not to include their information.*)

Implementation

Most of the work within this project involved navigating the data and processing it. After that there remained two main tasks to run:

1. Principal Component Analysis
2. K-means Clustering

With PCA the data had to first be normalized. This was accomplished with SciKit Learn's `MinMaxScaler()` method.

This project used Amazon's SageMaker platform which comes with a predefined PCA method. This was used to create a model which was fed 17 component principals as input. One less than the number of features in the final training dataset.

Finally, before fitting the model the data had to be converted to float-32 values. This was a parameter restriction for SageMaker's PCA algorithm.

After training the PCA model, it was deployed in order to send and generate principal components for each data point. This generated data was necessary for the next step.

K-means was run after PCA with the generated data. The optimal value for k was chosen by training the algorithm with different values for k (5-13) and averaging the distances to the nearest centroid. The best configuration (least average distance and smallest k value) was determined to be 10.

Refinement

After noticing a few things about the data, I decided it was good to rerun the algorithms with slightly different dataset features.

Firstly, due to my failure to notice initially that all the values for informational offers were 0, meaning that no informational offers were recorded as completed. Thus, every single data point would belong in that cluster and is not a necessary (and possibly causing extra bias) feature.

I noticed the same with the email type feature as well, only all the data points were 1 because all the offers in the portfolio were sent via email.

Then I repeated the PCA and K-means steps mentioned in the previous section.

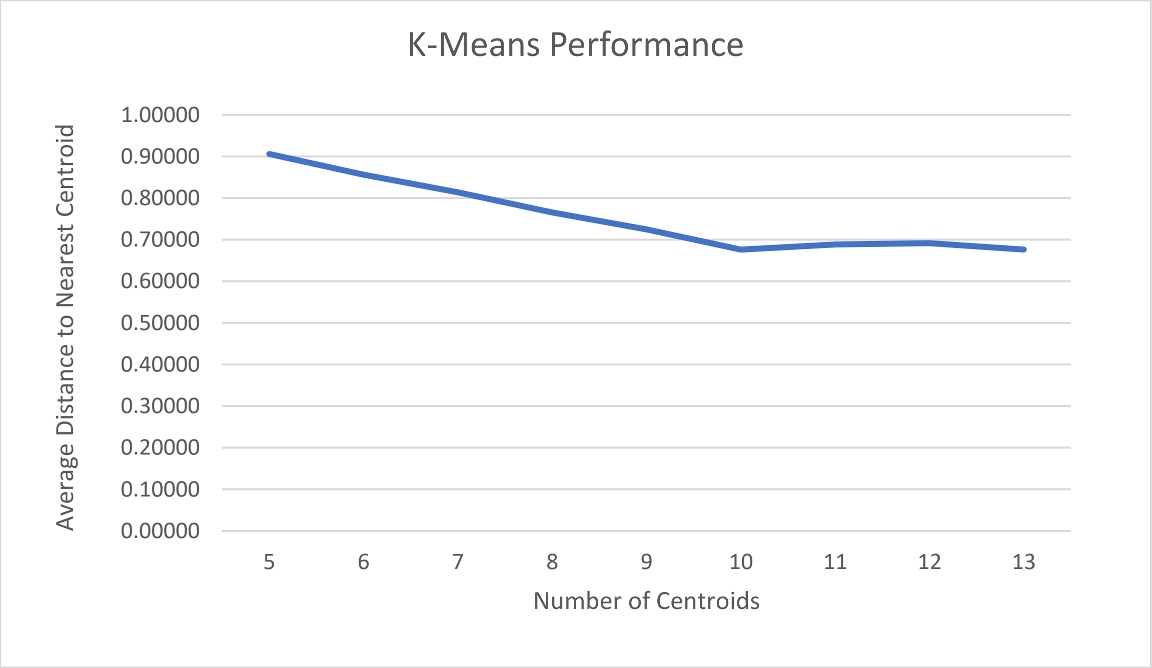
IV. Results

Model Evaluation and Validation

As mentioned in the metrics and benchmark sections, it's hard to validate the final results of unsupervised learning if we are trying to gain new insights into the data; however, I did individually evaluate PCA and K-means.

For PCA I used the explained variance equation to determine which features to use when transforming the dataset prior to training and fitting the K-means model. This allowed me to reduce the number of features (components) that k-means would use while still retaining near 100% of the relevance.

With K-means I merely ran the algorithm with differing values of k, starting with 5 increasing to 13. (As 13 are the top components from PCA that still explain all the variance.) With each iteration, I calculated the average distance of each point from the nearest centroid and plotted a graph:



From this graph I was able to choose the optimal value for k to be 10. This number is right at the "elbow", meaning that it is a good approximation to have because performance stops increasing significantly. I then used that model (k=10) to generate a heat/density map of clusters and centroid points. This allows a way to see which factors (components) affected the groupings (clusters) the most.

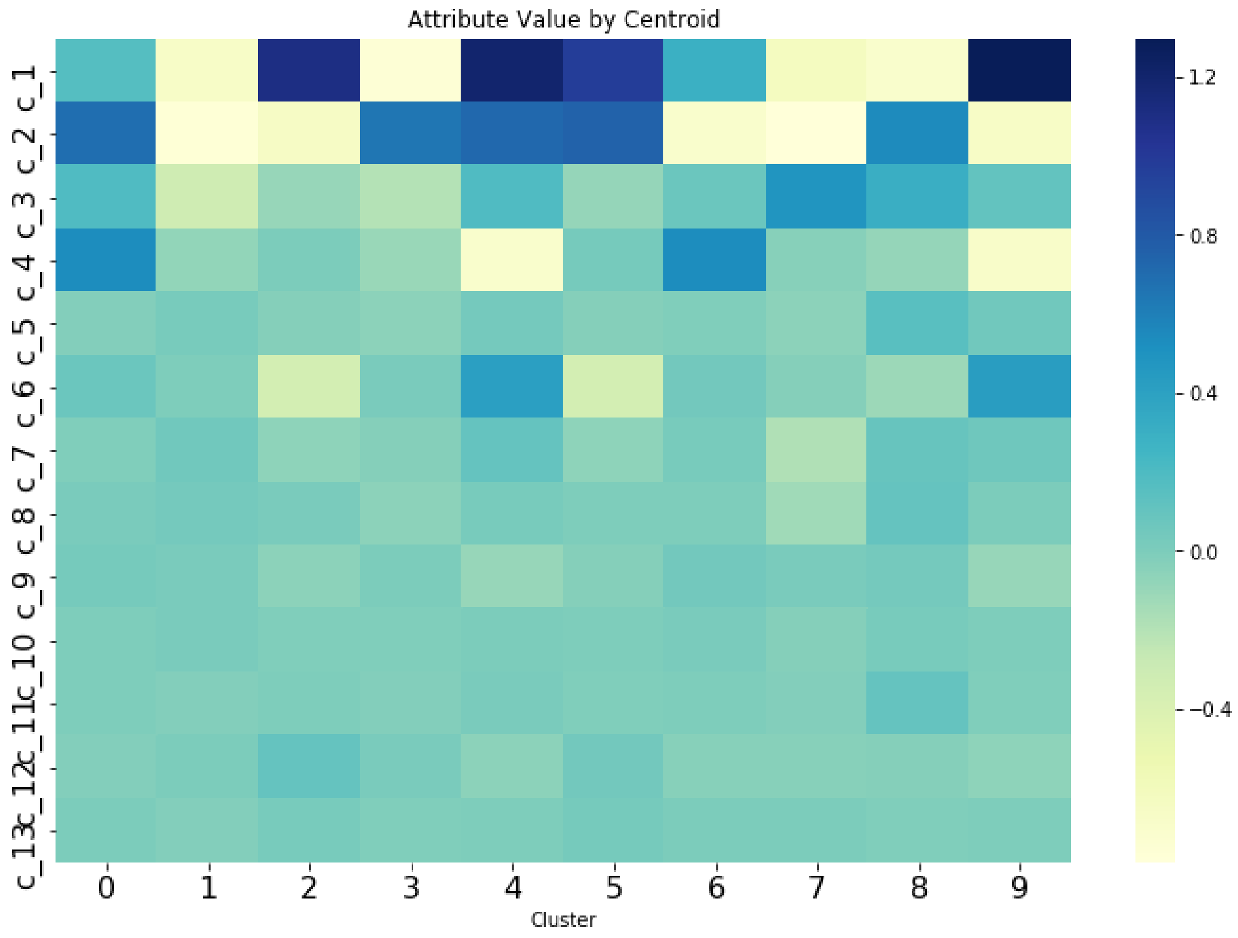
Justification

These methods allow us to gain a deeper insight than merely mapping 2 or 3 dimensional graphs of the data. This combined analysis shows the more complex real world grouping of data that exists. Successful marketing campaigns need to target the appropriate audience. Understanding the factors that make up most of that success is crucial. PCA and K-means allows us to visualize this in the final density map.

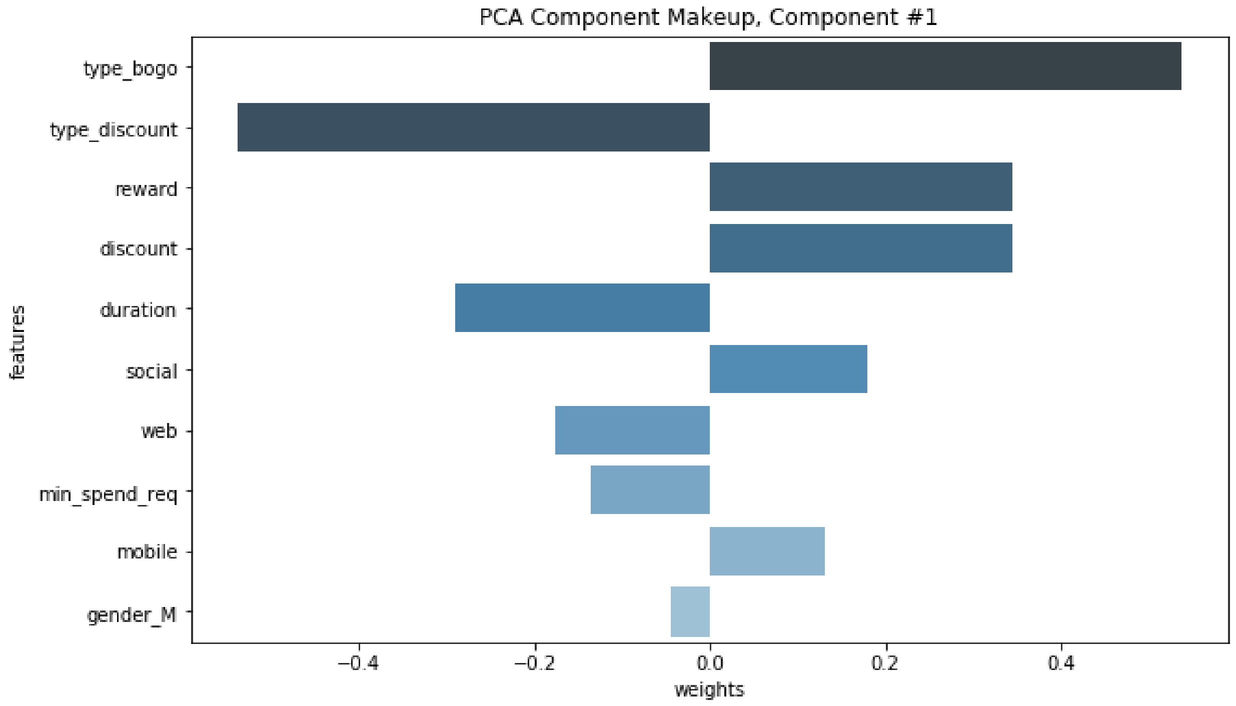
V. Conclusion

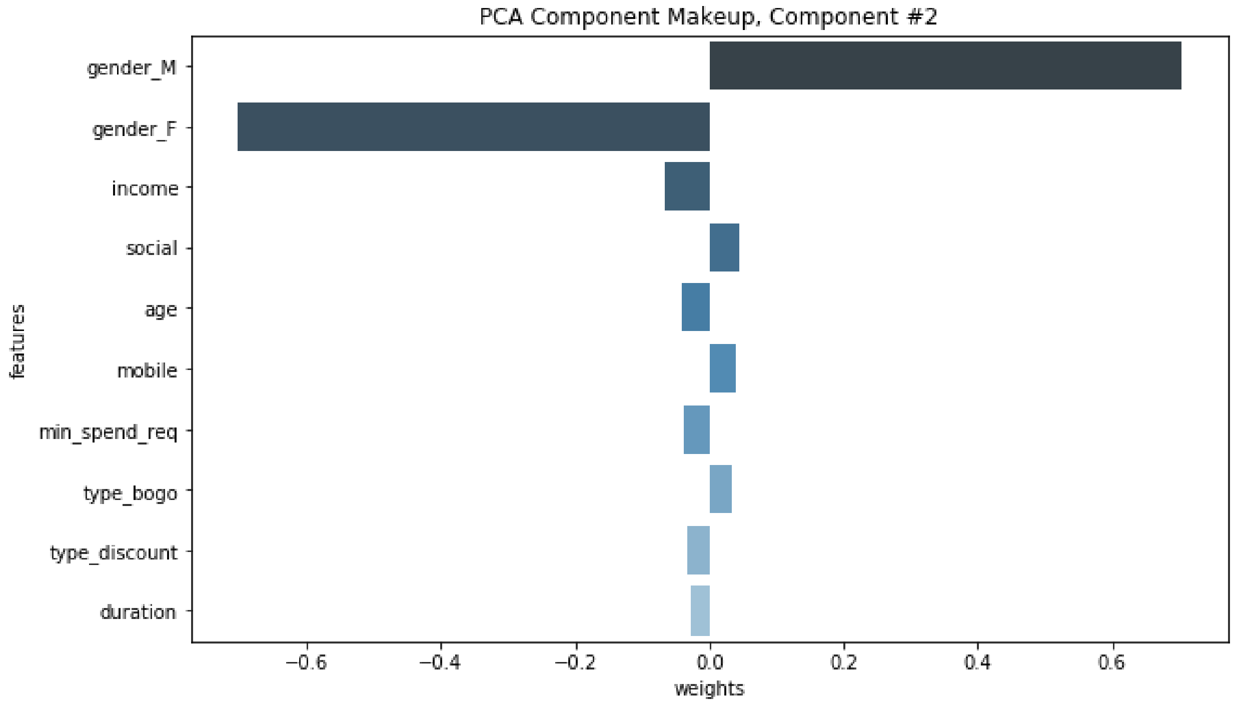
Free-Form Visualization

Heat Map of datapoints in cluster by components from PCA:



Components one and two seem to account for a lot of the groupings. These components can be seen by evaluating their composition from PCA:


PCA Component Makeup, Component #1


PCA Component Makeup, Component #2

So we can see that some of the major components that affected the groupings of successful offer completions were they types of BOGO and discount offers along with male and female identifying customers.

Reflection

To summarize the project steps:

1. Define the problem, in this case determining factors that make up successful offer completions.
2. Data pre-processing and transformation.
3. Choosing unsupervised learning techniques. (PCA and K-means)
4. Running K-means multiple times to find a good value for k.
5. Evaluating data with optimal K-means model.

Improvement

I think this project can gain deeper understanding and even make some prediction tools on possible success rates of future campaigns if the offers received and viewed were analyzed the same way and then the datasets were compared to see which offers were more likely to be completed after being sent out.

I also now understand the importance of the data collection phase of any machine learning project. For instance, if the transactions associated the offer ids it would have led to a lot more insights to be gained about offers. Specifically, the amount of money spent by each demographic group on a certain offer. This could lead to better future earnings predictions or more easily associate a monetary value with an offer.

Looking back, I could have tried playing around with combining the transactions data with offer completion data and combined the rows matching customer id and time. Most likely that would indicate a particular transaction with a specific offer. Time and uncertainty on accuracy without the exact offer id or even a transaction id persuaded me not to pursue that option.