Customer Segmentation Project Process:

My intent here isn't to document everything that I did in my analysis (this can be seen in the .ipynb files attached to this project), but to give a guide to the process of segmenting a customer database so that it can be reviewed and improved in the future whenever this project is repeated.

Objective
The objective of the Customer Segmentation is to create the most distinct customer segments possible. These segments should accurately reveal the attributes and buying patterns of current and potential customers.

It's important to understand that the process of discovering the most usable clusters is an iterative one. One will not always know the optimal treatment or model for the data before he/she begins the process. This project requires you to map out the process from start to finish and iteratively make changes to that process until the optimal results are obtained. Mapping out the process is the only way to make small improvements and monitor the results of each change. Having the process laid out from beginning to end allows failure or success (based on consistent and quantitative measures) to guide the modifications made to the project instead of intuition or top-down knowledge alone. Making small tweaks and changes mimics the evolutionary process that created the beautiful world that we live in and allows the user to make changes/grow with his/her data. My process can be iterated upon with only a few small changes mostly thanks to adaptable functions I created to handle tasks that need to be completed regardless of the iteration performed such as profiling and clustering.

Data Exploration/Cleaning
The first step is always to explore and clean the data. Exploring the data simply means getting to know it and formulating ideas on how I'll treat it in the future. While it's a wise idea to get an idea of the variable distributions and a sense of the unique values, this should not substitute the analysis phase. Don't get too caught up in analyzing the data because after cleaning and feature analysis, patterns in the data might appear or disappear.

Cleaning will involve treating the outliers, null values, and other inconsistencies. It can also include removing impossible results, rectifying data entry mistakes, and ensuring that each variable is matched to the correct data type. Again, it's important not to get too far ahead and analyze untreated data.

Since I have mapped out my method ahead of time, I know that I will be using Principal Component Analysis (PCA) as a dimensionality reduction technique, which requires me to treat my outliers. PCA is sensitive to outliers and leaving outliers in your dataset will result in skew. This analysis defines outliers as a data point that is more or less than 1.5 times the interquartile range of a variable. The best results were achieved when imputing the maximum or minimum value a feature could contain without being considered an outlier for each identified outlier. I hypothesize that this was the best method because the dataset contained many outliers and removing each observation containing an outlier resulted in the loss of too much important information.

Take note of inconsistencies, data entry mistakes, null values, and variable data types. Record practical, useful questions you want to answer about the data in the analysis phase. Knowing that I sometimes have the tendency to "get lost in the weeds" or overfocus on a question, record all questions so that I can review them later to determine which are practical and which are not. Other questions that I can use to guide my curiosity are:

- Why is answering this question important?
- What insights will be revealed?
- Is it worth the time/effort to answer this question?

Feature Engineering

Once the data had been cleaned, I created new features. These features were created based on common sense. In another project, mathematical or domain knowledge will guide the creation of new features.

Feature creation is another example of why the process should be completely mapped out before iterations are performed. An engineered feature may be detrimental to the analysis by creating additional noise or for some other reason. An iterative approach allows you to create new features, and take note of the impact that new feature has on your results.

Data Analysis

Now that the data is clean and has a few additional features, it's important to make sure that the dataset is well understood before moving on. I find that visualizing and becoming familiar with variable distribution and answering relevant questions are best practices in this step. This highlights the importance of making note of questions you want to answer while exploring and cleaning your data. It's important to have a good sense of your data **after** you've cleaned and created new features. This step is not necessarily distinct from Cleaning and Feature Engineering. Often cleaning and creating new features requires you to already have a level of understanding of your data, which is the purpose of your analysis.

Processing

For our Dimensionality Reduction tool and segmentation models to work best, the data must all be numeric type. This wasn't difficult as most were ordinal, binary, or already numeric to begin with. The rest of the categorial data was simply one-hot encoded. Meaning that a new binary column was created for each unique value of the variable.

It is very important through all steps to make sure to preserve the index or unique customer ID so that as cluster labels are generated for each model, they are always applied to the correct customer. This allows for accurate profiling to be carried out. I made sure to make the Customer ID the index so that the numeric customer ID would never be manipulated or transformed leading to inaccurate labeling and bad customer profiling.

Dimensionality Reduction

After general processing, the features of the data will be divided into 2 different groups: behavioral features (total purchases, preferred channels, etc) and attribute features (age, education, home size). The behavioral dataset was used to cluster the data, and the attribute data was used to profile the data. This returns us to our objective: to create segments that reveal the buying habits and attributes

of our current and potential customers. We group them by their habits and then create profiles that allow us to target potential customers by shared attributes.

The more variables there are, the more complex the model becomes. Overly complex models tend to be unduly specialized on your current data or "overfit". Overfit models won't hold up as new data from the same source becomes available. To avoid overly complex or overfit models, we use Dimensionality Reduction techniques to make our data more "digestible". While there are many DR techniques available, I chose to use PCA because it does a great job preserving information and it can self-report on the explained variation from the original dataset. One downside is that there is some difficulty in interpreting the results.

I iterated a few times on this step. I tried excluding and including different attribute features in the PCA process using silhouette scoring as my guide. The best results were achieved after excluding the campaign acceptance variables. I believe that removing these variables reduced noise since they were sparse features to begin with.

The result of Dimensionality reduction through PCA is a reduced dataset with, in my case, only 3 variables that retain ~80% of the variation in the original dataset as opposed to 20-something features and a lot more noise.

Segmentation:
As I stated above, one cannot always know the optimal treatment or model for the dataset as they begin a project. For this reason, I tested 3 different clustering models: K-Means, K-Medoids, and Gaussian Mixture Modeling.
The quality of each cluster, or Cluster distinction, in this analysis, is based on the silhouette score, which measures how distinct and different one grouping of customers is from another group in the same model, and the profile of each customer grouping.

Each of the models I used requires you to input the desired number of clusters beforehand. To determine the optimal number of clusters for each model, I used a for-loop to model the data several times with a different number of clusters each time. I chose the optimal number of clusters based on the Silhouette Score and Davies-Bouldin Scoring. The ideal number of clusters was placed into a dictionary where the keys were the model and the values were the silhouette score. This was done so that I could later compare the model scores in a visualization.

Once I found the ideal number of clusters for that model, I modeled it one last time and added the cluster labels to the behavioral dataset that contained the original attribute variables. I was then able to generate visualizations that profiled the cluster group.

Profiling:
To profile each cluster group, the mean of each attribute feature was used as a representative value. If the cluster means for an attribute differed from each other, that attribute was interpreted as a defining characteristic. For example, many of the cluster groups had similar average ages and so age was determined not to be a defining characteristic of any of the clusters for that model.

The quality of profiling was based on attribute distinction. The profiles for K-Medoids segmentation were not dissimilar enough to extract any useful information, which failed our purpose, to find identify groups within our current customers and better target potential customers.

In another iteration of this project, I might use boxplots or histograms to compare cluster profiles so that I can view the distribution. While the mean is important, it does not always tell the entire story. It's important to utilize a measure of center AND spread. A boxplot with the mean or a density curve might be best.

There was some challenge in determining how to create the large number of visualizations needed for each attribute in every cluster. In the end, for comparison's sake, I found that the easiest way to visualize this was to create one graph for each attribute where each graph has one bar representing the mean for each cluster. This allowed me to view each attribute across clusters. I could easily see how the clusters differed from one another one attribute at a time.

Selection:
My selection criteria consisted of silhouette scoring and usability of profiles. The final results showed that K-Means performed best. It had the highest silhouette score and the most useable profiles. These profiles allowed me to make actionable recommendations for the business.

This process can and should be improved upon. The highest Silhouette Score was 0.42. A score over 0.5 is considered a high-quality cluster and a score under 0.5 is considered to be low-quality cluster. In further iterations, a different dimensionality reduction technique, different outlier treatment, feature selection, or feature engineering might allow for improved clustering.

Presentation:
Here are a few notes from presenting this analysis to my friend Jon George:

- Focus on the process in the presentation. Peers and stakeholders need to know that you're competent and you know what you're doing.
- Return to purpose and key focuses often during the presentation. Start with your purpose and key focuses and explain how the information you're presenting is relevant to the purpose and key focuses as relevant.
- More focus on analytical visualization.