Alexander Vaillant 6/10/2021

D212 Performance Assessment Task 1:
Exploratory Data Analysis using KMeans Clustering

Part I: Research Question

Research Question:

To what extent can you accurately group customers into clusters based on customer cost, usage, internet service, contract type, and income and how many clusters are best?

Goal:

Determine how many clusters are best with our given dataset and parameters.

Part II: Technique Justification

Kmeans explanation:

The Kmeans algorithm takes data points as the input and groups them into clusters with centroids based on their Euclidean distance from each other. The centroids are the center of each cluster. The number of clusters that the data is grouped into must be designated with the value k. In our data analysis, I have tested multiple values of k from 2-7 with a For Loop and found the distortion, or the sum of the squared Euclidean distances between the observations and their cluster's centroid, of each number of clusters. As the number of clusters increases, the distortion decreases. However, to avoid having too many clusters, it is best to use an elbow plot to decide the healthy balance between number of clusters and distortion. This health balance would be at the instance of where the plot elbows. In our elbow plot, there was a very slight elbow at 4 clusters. A slight elbow is typical with datasets that are extremely equal much like our fictitious dataset.

One Assumption of Kmeans:

All variables must have the same variance. If the variables do not have the same variance, this will allow one feature more importance over other features. This can be avoided by normalizing or standardizing your data depending on your situation. I have decided to normalize my dataset by using the MinMaxScaler() from sklearn.preprocessing.MinMaxScaler.

Libraries/Packages:

1. **Pandas** – this was imported to read the csv dataset into a dataframe, create dataframes, transform categorical variables into binary dummy variables, and count the values of each cluster.
2. **Scipy.cluster.vq.kmeans** – this was imported to use the kmeans algorithm, get the distortion for each number of clusters, and test various numbers of clusters with our dataset.
3. **Scipy.cluster.vq.kmeans2** – this was imported to use the kmeans2 algorithm with 3 clusters, assign each observation their cluster label (0,1,2), and extract those labels to a new dataframe.
4. **Seaborn** – this was imported to plot an elbow plot of the numbers of clusters by distortion for each of number of clusters. After visualizing representing the data, I could decide which number of clusters was best for our analysis.

5. **Sklearn.preprocessing.MinMaxScaler** – this was imported to normalize our dataset. If we did not normalize our dataset, some features, mainly the numeric features like monthly charge and bandwidth_gb_year, would have more importance than the categorical dummy variables.
6. **Numpy.random** – this was imported to assign our analysis a random seed of 1234 that would reproduce our results the exact same way every time.

Part III: Data Preparation

Data Preprocessing Goals:

1. The variables must have the same variance. If the variables do not have the same variance, this will allow one feature more importance over other features. This can be avoided by normalizing or standardizing your data depending on your situation. I have decided to normalize my dataset. I used sklearn.preprocessing.MinMaxScaler to normalize my dataset for kmeans algorithm optimization.
2. Transforming Categorical variables to binary dummy variables. This was done by using the pandas library function .get_dummies(). The function converts categorical variables into binary dummy variables.
3. Indexing my dataset to only the 6 needed features (5 for analysis, 1 for further analysis beyond this project). I indexed my dataframe by calling only the needed columns.

Initial Dataset Variables:

Variable Name	Variable Type
Churn	Categorical
MonthlyCharge	Continuous
Bandwidth_GB_Year	Continuous
Contract	Categorical
InternetService	Categorical
Income	Continuous

For data preparation code, labeling, and step-by-step explanations, please see "D212 Performance Assessment Task 1.ipynb", "Task 1 – Data Preparation Steps and Explanations in SS.png", and "D212 Performance Assessment Task 1.html".

For a copy of my prepared dataset, please see "prepped_data.csv".

Part IV: Analysis

Description of my Kmeans Data Analysis:

The Kmeans algorithm takes data points as the input and groups them into clusters with centroids based on their Euclidean distance from each other. The centroids are the center of each cluster. The number of clusters that the data is grouped into must be designated with the value k. In our data analysis, I have tested multiple values of k from 2-7 with a For Loop and found the distortion, or the sum of the squared Euclidean distances between the observations and their cluster's centroid, of each number of clusters. As the number of clusters increases, the distortion decreases. However, to avoid having too many clusters, it is best to use an elbow plot to decide the healthy balance between number of clusters and distortion. This health balance would be at the instance of where the plot elbows. In our elbow plot, there was a very slight elbow at 4 clusters. A slight elbow is typical with datasets that are extremely equal much like our fictitious dataset. The Kmeans2 algorithm was then used to group the data into 4 clusters and export the labels for each observation.

For screenshots of my intermediate calculations as an elbow plot in seaborn, please see "Task 1 – Kmeans Algorithm Analysis.jpg".

For the code I used to perform the cluster analysis technique, please see "D212 Performance Assessment Task 1.ipynb" and "D212 Performance Assessment Task 1.html".

Part V: Data Summary and Implications

Explanation of Accuracy of my KMeans Clustering Analysis

In our data analysis, I have tested multiple values of k from 2-7 with a For Loop and found the distortion, or the sum of the squared Euclidean distances between the observations and their cluster's centroid, of each number of clusters. As the number of clusters increases, the distortion decreases. There needs to be a healthy balance between the number of clusters we group the data into and how high the distortion is. This is typically found by using an elbow plot like I did for this analysis. At the point of the elbow, or bend, this is where the optimal number of clusters is decided because the distortion starts to lessen at slower rates closer to 0. If you use too many clusters, then you run the risk of overfitting the algorithm and not allowing the opportunity to gleam more insights from the data. The optimal number of clusters for our kmeans algorithm was 4 clusters. The distortion was 0.7 which means the average sum of squared Euclidean distance between our observations and their assigned centroid was 0.7. This was the best number of clusters for our dataset. With 2 and 3 clusters, the distortion was near 1 and 0.85, respectively.

Results and Implications

The result of our kmeans cluster analysis were 4 clusters with a split of 3316 (1), 3037 (2), 2442 (3), and 1205 (0) observations. Each cluster is paired with similar observations with an average squared Euclidean distance of 0.7 normalized units. These four clusters can be further analyzed by splitting each into training and test data sets and using logistic regression to gleam more insights from the data. Since our ultimate goal is a reduction in customer churn, these results provide a unique opportunity to reduce customer churn when we focus on similar customers in each group.

Limitations:

1. This is a fictitious dataset and the distribution of the variables are equally balanced across the observations. This leads to a slight elbow plot of the number of clusters vs distortion. In a dataset with real values, a stronger elbow motion may be better seen from this analysis.

2. Kmeans is an exploratory data analysis clustering technique. This means that further analysis is required to gain better insights from the data. The 4 clusters our dataset is split into do not represent any specific variable. For example, just because the data is split into 4 clusters does not mean each group belongs to four different payment types.
3. Using kmeans with more than 2-3 features makes it very difficult to plot centroids and each of the observations on a graph. For this analysis, I would have needed to plot the features as a 5-dimension plot.

Recommendation:

Based on this exploratory data analysis using Kmeans, I would take the final dataset with the added kmeans 4-cluster labels and churn variable and complete further analysis using logistic regression on each of the 4 groups. This would include splitting each of the groups into train and test sets, fitting a regression model based on the train set, and testing the model on our test set. I would use the insights gleamed from this next analysis to target customers like those in each of these clusters and provide incentives to avoid customer churn.