D212 Clustering Techniques Performance Assessment, Task 1

Alexa R. Fisher

Western Governors University

Degree: M.S. Data Analytics

Table of Contents

Part I: Research Question

A1. Proposal of Question

The research question for this thesis was, “Are there any meaningful groupings provided by an analysis of the customer survey responses?”. The thesis analysis was evaluated by utilizing the hierarchical clustering method on the telecommunication churn dataset. The variables used in this were the responses to the eight questions within the survey questionnaire. These questions included the following: “Timely Responses”, “Timely Fixes”, “Timely Replacements”, “Reliability”, “Options”, “Respectful Responses”, “Courteous Exchange”, and “Evidence of Active Listening”. The responses ranged from “1” to “8” with “1” being the most important and “8” being the least important. Evaluating these responses could provide insight into whether there was a pattern within and a good key area in client retention.

A2. Defined Goal

This analysis aimed to determine if any meaningful clusters within the questionnaire data could be identified that would otherwise be unclear by reviewing the complete distribution. In the general view of the questionnaire data, the survey responses had very similar mean and standard deviation. This was shown using the .describe() method within Python. Each survey variable had a mean of about 5.50 and a standard deviation of about 1.03 after the variables were remapped in the data preprocessing stage. The use of hierarchical clustering could help identify what would be the main survey variables between each grouping. The use of this would help the telecommunication company in putting focused actions toward client retention and satisfaction.

Part II: Technique Justification

B1. Explanation of Clustering Technique

The clustering technique used for this analysis was hierarchical clustering. This clustering method groups each data point based on similarity (Yıldırım, 2020). The groups were visually represented in a tree-like representation called a dendrogram. Hierarchical clustering allowed clusters to be formed without needing to specify a certain number of clusters. It was automatically completed within the creation of the dendrogram. It allowed the data to be easily interpreted.

Various functions from the SciPy package were applied in the data analysis. The linkage() function was utilized to perform the hierarchical clustering on the telecommunication survey responses for each of the 10,000 customers. The function created a matrix of the data by including the questionnaire variables, the Ward method, and the Euclidean metric. The matrix was used to populate the dendrogram. The .dendrogram() function provided the visualization of the clustering. It allowed the data to easily show the cluster groupings and the distance between them. Lastly, the .fcluster() function was used to label each observation based on the cluster it belonged to. The labeling allowed the dataset to be analyzed and each distribution reviewed.

B2. Summary of Technique Assumption

There were several assumptions of the hierarchical clustering technique. Hierarchical clustering was based on the distance between each data observation. One of the assumptions for this technique was the standardization of the variables. Usually, this was completed with the use of SciKitLearn’s function StandardScaler. It processed the variables under the same scale. The variables having different scales and distributions could indicate inaccurate and skewed results.

As the questionnaire variables were on the same standardized scale, the use of a standardization method was not needed. The data range for each of the variables was from “1” to “8”.

B3. Packages of Libraries List

Please see below for a complete listing of the libraries and packages used in the analysis of the telecommunication dataset.

Package/ Library	**Usage**
Pandas	The Pandas library was used to analyze data. It allowed the data set to be explored by importing the telecommunication churn CSV file. It provided a way for the dataframe to be manipulated during the data cleaning and exploration phases.
NumPy	The NumPy library was utilized to work with arrays. This allowed data to be used in calculations and manipulations.
scipy.cluster.hierarchy import linkage	The linkage function from the SciPy was used to compute the distances within hierarchical clustering.
scipy.cluster.hierarchy import dendrogram	The dendrogram function from the SciPy library allowed for a graphical tree-like representation of the hierarchical clusters that were generated by the linkage function.
scipy.cluster.hierarchy import fcluster	The fcluster function from SciPy allowed for the labeling of data points within the hierarchical clustering analysis.
Matplotlib.pyplot	The Pyplot module within the Matplotlib library was used to provide visualizations.

Seaborn	The Seaborn library was applied to visualize and explore just like Pyplot. It was used to provide a visual of the distributions for the average(mean) of each survey question via a line plot and count plots.
sklearn.metrics import silhouette_score	The Silhouette Score from the SciKitLearn library allowed for the evaluation of the clustering method accuracy.

Please see below for the code to import packages and libraries:

```
#install libraries and packages to use with environment for analysis


import pandas as pd
import numpy as np
from scipy.cluster.hierarchy import linkage, dendrogram
from scipy.cluster.hierarchy import fcluster
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import silhouette_score
```

Part III: Data Preparation

C1. Data Preprocessing

There were various preprocessing goals that were relevant to the hierarchical clustering technique. The main goal addressed with the telecommunication data set was the preparation of the questionnaire responses. The questionnaire responses were integer values from “1” to “8”. Based on the provided data dictionary, which notated the survey response of “1” had the highest importance for a customer. The opposite was true for the survey response of “8”, the lowest importance. These values provided a skewed view when blindly reviewing the data as “8” was usually greater than “1”. The variable values were remapped to make the value of “8” as most important with “1” being the least important. The remapped variables were converted from integer to float data types. The data type needed to be float to utilize the linkage function and create the distance matrix.

C2. Dataset Variables

The variables utilized for the data analysis of the telecommunication data set included the eight survey questionnaire variables of Item1 through Item8. Each of these variables was qualitative or categorical. The questionnaire responses were the customers’ rating of importance from “1” to “8”. The rating of “1” would be the most important and “8” being the least important. This was remapped during the data processing stage to have “1” notate the least important rating and “8” being the most important.

Variables	**Data Type**
Item1: Timely Response (renamed to Timely_Respd)	Categorical
Item2: Timely Fixes (renamed to Timely_Fixes)	Categorical
Item3: Timely Replacements (renamed to Timely_Rplc)	Categorical
Item4: Reliability (renamed to Reliability)	Categorical
Item5: Options (renamed to Options)	Categorical
Item6: Respectful response (renamed to Respect_Resp)	Categorical
Item7: Courteous Exchange (renamed to Courteous_Exch)	Categorical
Item8: Evidence of active listening (renamed to Evidence_ActListen)	Categorical

C3. Steps for Analysis

There were several steps to prepare the data for analysis. The survey questions were notated as Item1 through Item8 within the original data set. The .rename() function was used to change the questionnaire response variables from “Item1”, “Item2”, “Item3”, “Item4”, “Item5”, “Item6”, “Item7”, and “Item8” to “Timely_Respd”, “Timely_Fixes”, “Timely_Replc”, “Reliability”, “Options”, “Respect_Resp”, “Courteous_Exch”, and “Evidence_ActListen”. This was completed with the following code.

```
#renaming unclear variables.
data = data.rename(columns = { "Item1": "Timely_Respd", "Item2":
"Timely_Fixes",
                        "Item3": "Timely_Replc", "Item4": "Reliability", "Item5":
"Options",
                        "Item6": "Respect_Resp", "Item7": "Courteous_Exch",
                        "Item8": "Evidence_ActListen"})
```

The next step was to review the remaining variables for shape, duplications, and null values. The data set had a total of fifty variables with 10,000 entries. There were no null values or duplicates found.

```
data.shape
data.info()
# checking for duplicates
data.duplicated()
#checking for null values
data.isnull().sum()
```

The unnecessary variables were removed from the data set by using the .drop() function. Once the questionnaire variables were left, the variables were converted from integers to float and remapped.

```
#dropping unneeded variables
data.drop(['CaseOrder','Customer_id', 'Interaction', 'UID', 'City', 'State',
'Children', 'Age',  'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone',
'Job',
    'Income', 'Marital', 'Gender', 'Churn',
    'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equip_failure',
    'Techie', 'Contract', 'Port_modem', 'Tablet', 'InternetService',
    'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup',
    'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
    'PaperlessBilling', 'PaymentMethod', 'Tenure', 'MonthlyCharge',
    'Bandwidth_GB_Year'], axis=1, inplace=True)
#convert int to float
data['Timely_Respd'] = data['Timely_Respd'].astype('float64')
data['Timely_Fixes'] = data['Timely_Fixes'].astype('float64')
data['Timely_Replc'] = data['Timely_Replc'].astype('float64')
data['Reliability'] = data['Reliability'].astype('float64')
data['Options'] = data['Options'].astype('float64')
data['Respect_Resp'] = data['Respect_Resp'].astype('float64')
data['Courteous_Exch'] = data['Courteous_Exch'].astype('float64')
data['Evidence_ActListen'] = data['Evidence_ActListen'].astype('float64')
data.info()
#remapping survey values from 1-8 to be 8-1
remap={1: 8, 2: 7, 3 : 6, 4: 5, 5: 4, 6: 3, 7 : 2, 8 : 1}
data['Timely_Respd'] = data['Timely_Respd'].map(remap)
data['Timely_Fixes'] = data['Timely_Fixes'].map(remap)
data['Timely_Replc'] = data['Timely_Replc'].map(remap)
data['Reliability'] = data['Reliability'].map(remap)
data['Options'] = data['Options'].map(remap)
```

```
data['Respect_Resp'] = data['Respect_Resp'].map(remap)
data['Courteous_Exch'] = data['Courteous_Exch'].map(remap)
data['Evidence_ActListen'] = data['Evidence_ActListen'].map(remap)
data.head()
```

Lastly, the .describe() function was used to review the mean and standard deviation of the survey responses. Each of the variables had a similar mean and standard deviation. The mean was around 5.50 for each variable, the range was from 5.4905 to 5.5130. The standard deviation was around 1.03, the actual range was from 1.0248 to 1.0377.

```
#showing mean and standard deviation of survey responses
data.describe()
```

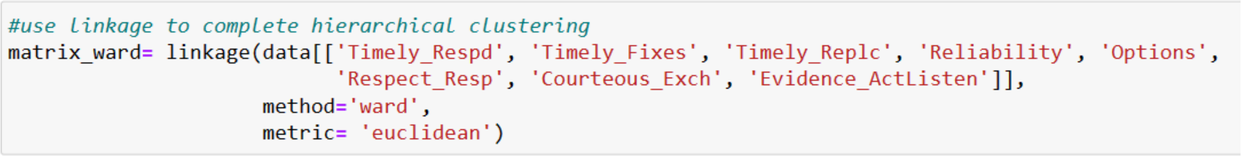
C4. Cleaned Dataset

Please see the attached CSV file called AFCodeD212Tk1_clean.csv to view the results of the cleaned dataset.

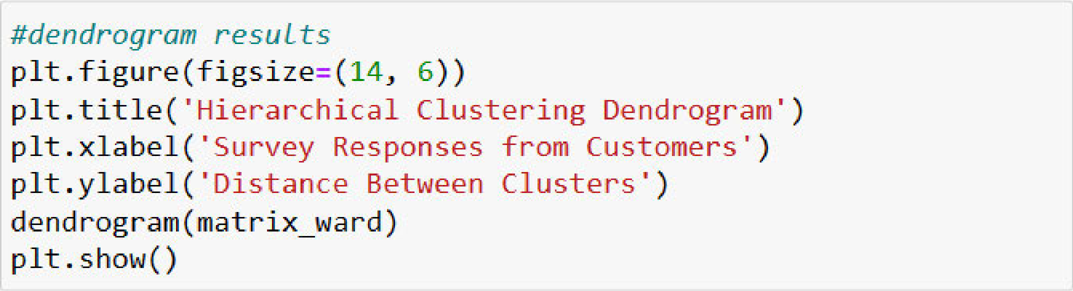
Part IV: Analysis

D1. Output and Intermediate Calculations

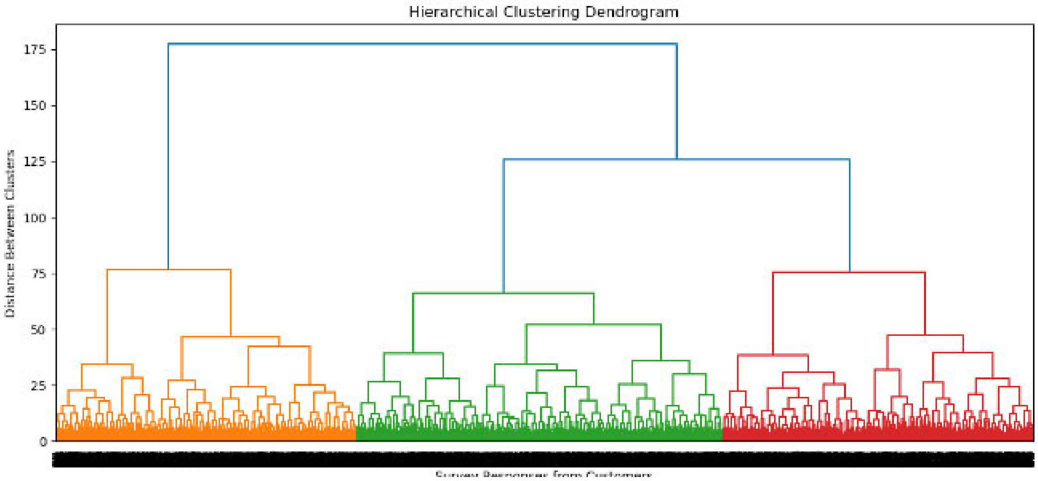
The analysis technique used to analyze the telecommunication data set was hierarchical clustering. As noted previously, hierarchical clustering took each data point and grouped the points based on how similar the point was to other points (Yıldırım, 2020). There were various method types of hierarchical clustering such as ward, single, and complete. The ward method was based on the sum of the squares (Daityari, n.d.). The single method was based on the two closest objects (Daityari, n.d.). The complete method was based on the two furthest objects (Daityari, n.d.). Each method was attempted during the exploration stage, but the ward method was selected for the analysis. The ward method was deemed the best methodology within the linkage function as other methods either crashed the kernel or had a longer processing time in comparison. The ward method allowed for an obvious distinction between the distances of the clusters especially at a top level. It showed two clear clusters.



```
#use linkage to complete hierarchical clustering
matrix_ward= linkage(data[['Timely_Respd', 'Timely_Fixes', 'Timely_Replc', 'Reliability', 'Options',
                           'Respect_Resp', 'Courteous_Exch', 'Evidence_ActListen']],
                     method='ward',
                     metric= 'euclidean')
```

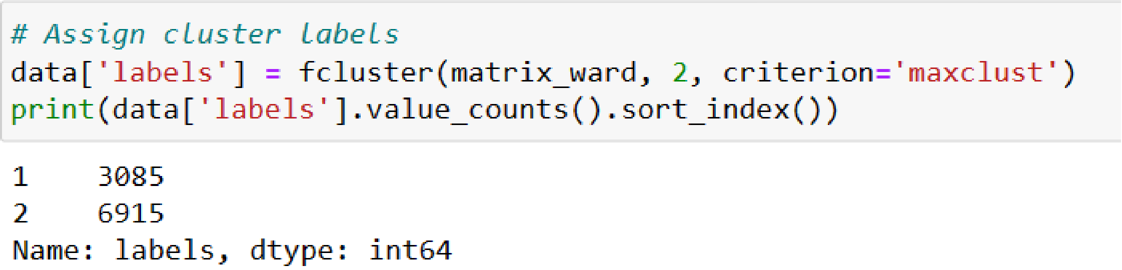


```
#dendrogram results
plt.figure(figsize=(14, 6))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Survey Responses from Customers')
plt.ylabel('Distance Between Clusters')
dendrogram(matrix_ward)
plt.show()
```



In the dendrogram computed, the x-axis was the survey responses from the 10,000 customers within the telecommunication data set. The y-axis was the distance between the clusters. It was shown that the top-level clusters consisted of two main clusters with a distance of 175 between them. Although there were three distinct clusters notated by the orange, green, and red when the distance threshold was at ~90, there was a closer proximity between the green and red clusters versus the orange. It was deemed two clusters would be appropriate by utilizing a distance threshold of greater than ~125. These clusters would be a grouping of the green and red clusters together and then the separate orange cluster. The quantity of two was applied within the .fcluster() function.

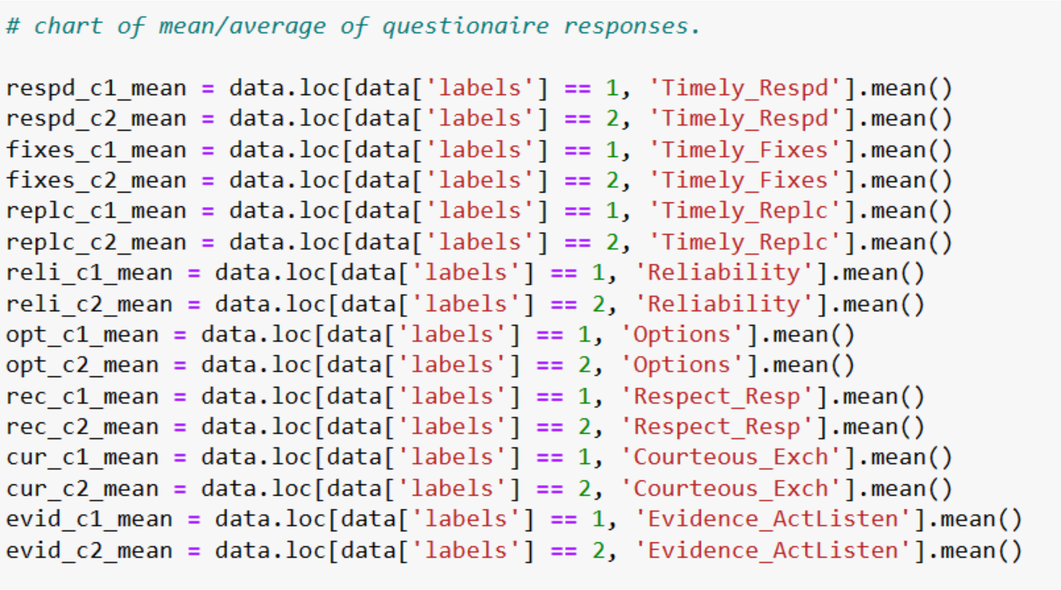
The .fcluster() function labeled each of the data points and added a label to each row to indicate if the row would be in the first cluster or the second cluster. The .value_counts() function in conjunction with the .sort_index() function was used to count each data point and sort them based on which cluster was assigned. This provided the following metrics for the 10,000 customers: 3085 or 30.85% of them were in the first cluster and 6915 or 69.15% were in the second cluster. This matched the data which was represented in the dendrogram.



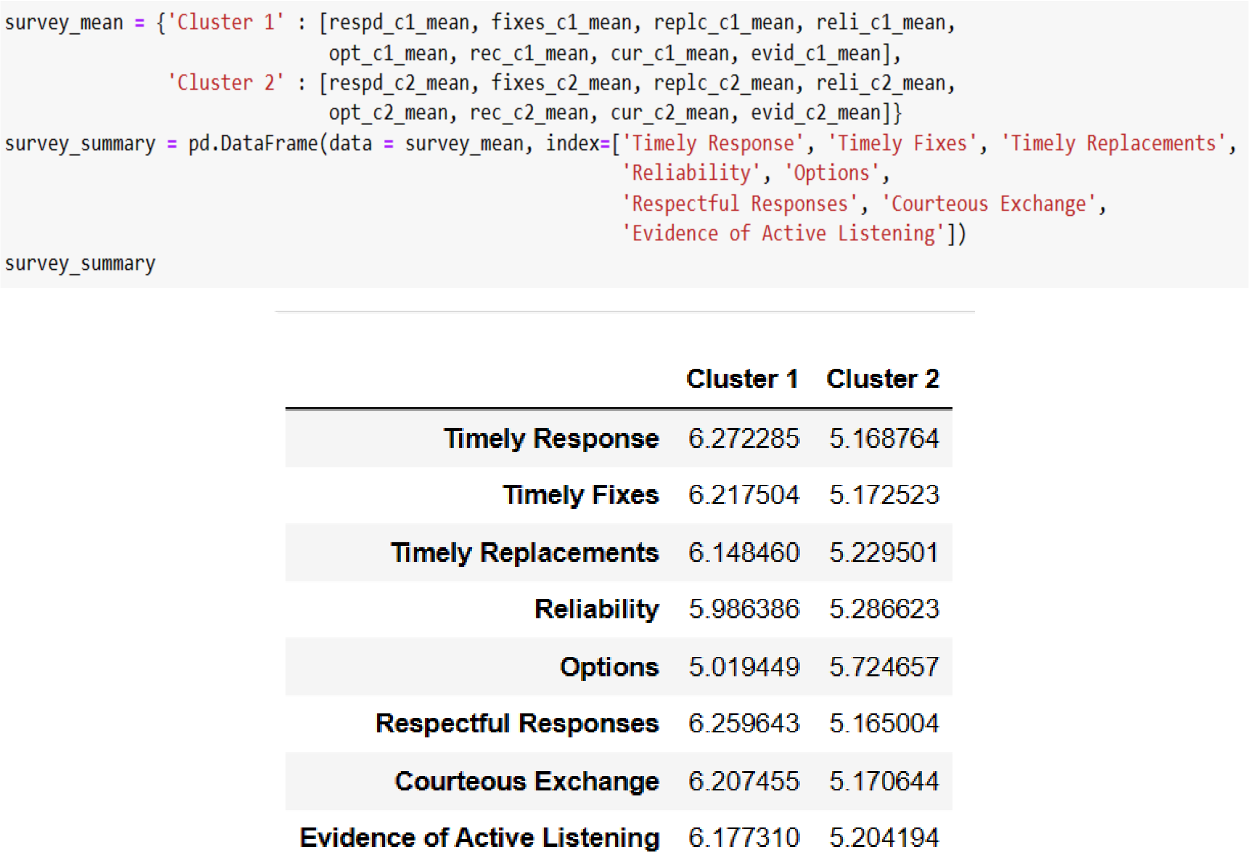
```
# Assign cluster labels
data['labels'] = fcluster(matrix_ward, 2, criterion='maxclust')
print(data['labels'].value_counts().sort_index())
```

```
1    3085
2    6915
Name: labels, dtype: int64
```

The separation of the clusters allowed for the creation of additional visualizations. A summary of the means for each cluster based on their questionnaire response was created. It showed the mean for each questionnaire variable in the second cluster was lower, excluding the variable, “Options”. It was further plotted on a line plot.

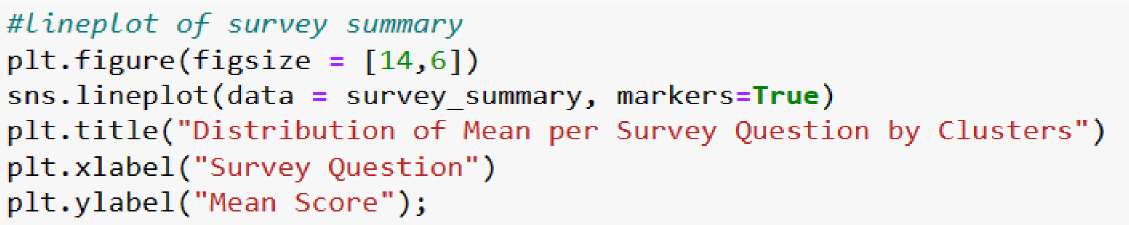


```
# chart of mean/average of questionaire responses.

respd_c1_mean = data.loc[data['labels'] == 1, 'Timely_Respd'].mean()
respd_c2_mean = data.loc[data['labels'] == 2, 'Timely_Respd'].mean()
fixes_c1_mean = data.loc[data['labels'] == 1, 'Timely_Fixes'].mean()
fixes_c2_mean = data.loc[data['labels'] == 2, 'Timely_Fixes'].mean()
replc_c1_mean = data.loc[data['labels'] == 1, 'Timely_Replc'].mean()
replc_c2_mean = data.loc[data['labels'] == 2, 'Timely_Replc'].mean()
reli_c1_mean = data.loc[data['labels'] == 1, 'Reliability'].mean()
reli_c2_mean = data.loc[data['labels'] == 2, 'Reliability'].mean()
opt_c1_mean = data.loc[data['labels'] == 1, 'Options'].mean()
opt_c2_mean = data.loc[data['labels'] == 2, 'Options'].mean()
rec_c1_mean = data.loc[data['labels'] == 1, 'Respect_Resp'].mean()
rec_c2_mean = data.loc[data['labels'] == 2, 'Respect_Resp'].mean()
cur_c1_mean = data.loc[data['labels'] == 1, 'Courteous_Exch'].mean()
cur_c2_mean = data.loc[data['labels'] == 2, 'Courteous_Exch'].mean()
evid_c1_mean = data.loc[data['labels'] == 1, 'Evidence_ActListen'].mean()
evid_c2_mean = data.loc[data['labels'] == 2, 'Evidence_ActListen'].mean()
```

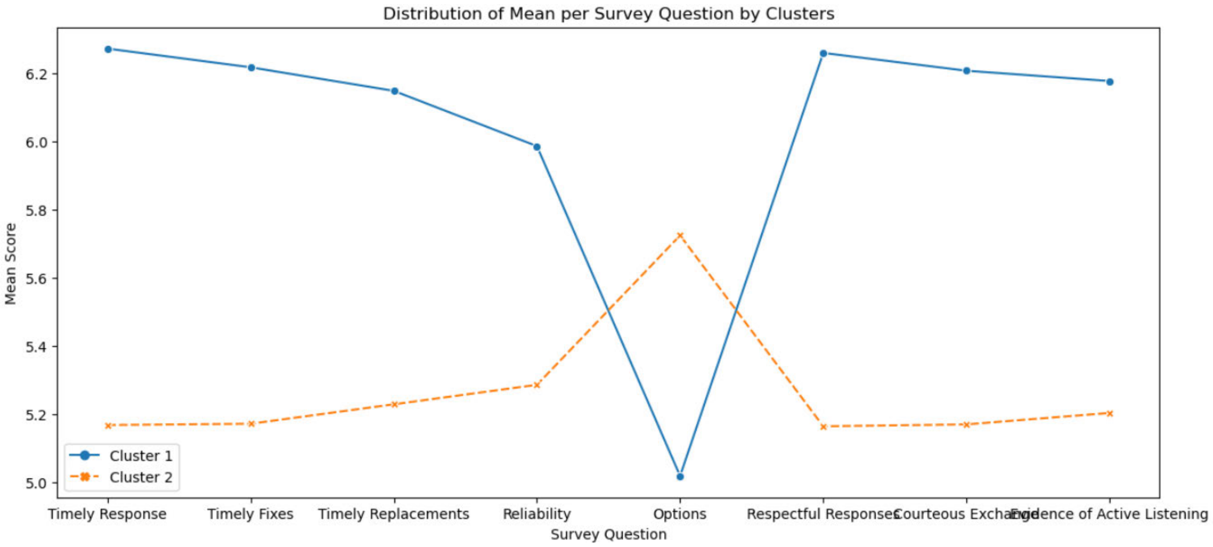


```
survey_mean = {'Cluster 1' : [respd_c1_mean, fixes_c1_mean, replc_c1_mean, reli_c1_mean,
                              opt_c1_mean, rec_c1_mean, cur_c1_mean, evid_c1_mean],
               'Cluster 2' : [respd_c2_mean, fixes_c2_mean, replc_c2_mean, reli_c2_mean,
                              opt_c2_mean, rec_c2_mean, cur_c2_mean, evid_c2_mean]}
survey_summary = pd.DataFrame(data = survey_mean, index=['Timely Response', 'Timely Fixes', 'Timely Replacements',
                                                         'Reliability', 'Options',
                                                         'Respectful Responses', 'Courteous Exchange',
                                                         'Evidence of Active Listening'])
survey_summary
```

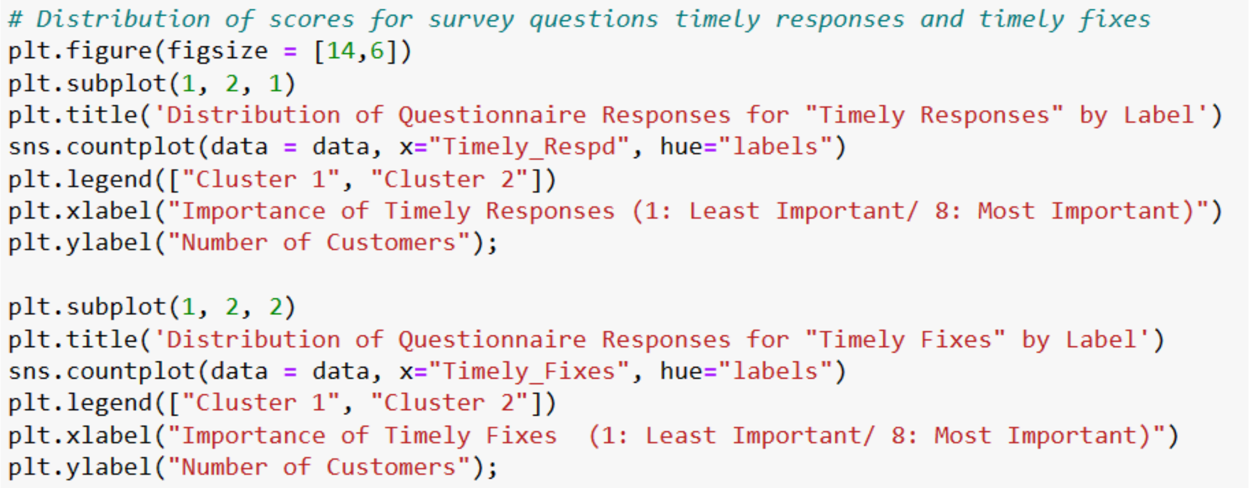
	Cluster 1	Cluster 2
Timely Response	6.272285	5.168764
Timely Fixes	6.217504	5.172523
Timely Replacements	6.148460	5.229501
Reliability	5.986386	5.286623
Options	5.019449	5.724657
Respectful Responses	6.259643	5.165004
Courteous Exchange	6.207455	5.170644
Evidence of Active Listening	6.177310	5.204194



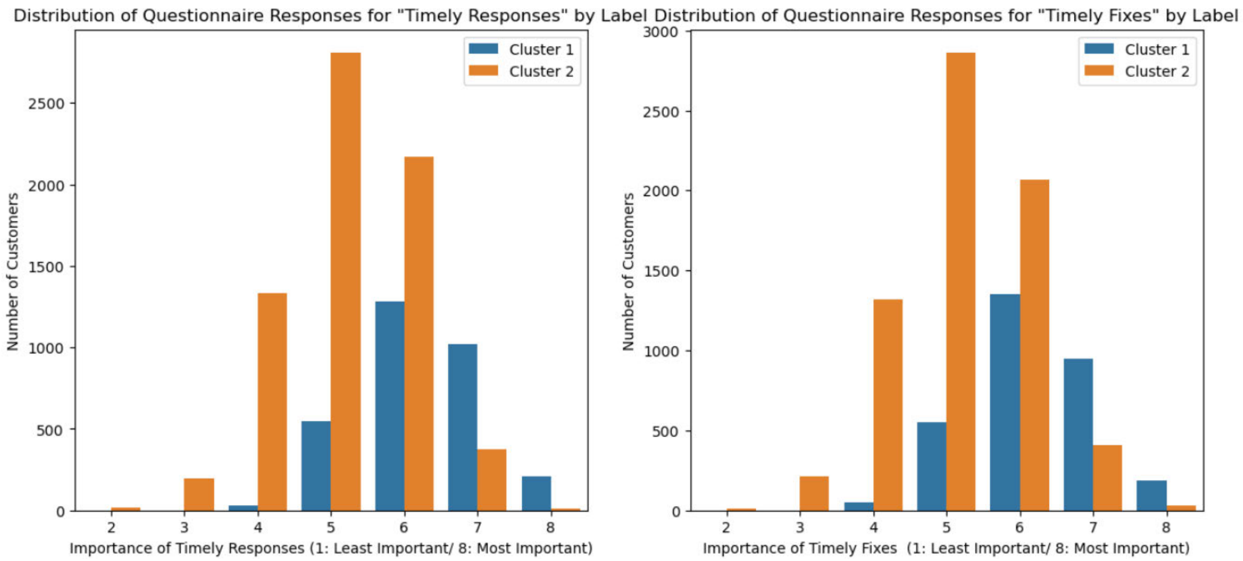
```
#lineplot of survey summary
plt.figure(figsize = [14,6])
sns.lineplot(data = survey_summary, markers=True)
plt.title("Distribution of Mean per Survey Question by Clusters")
plt.xlabel("Survey Question")
plt.ylabel("Mean Score");
```

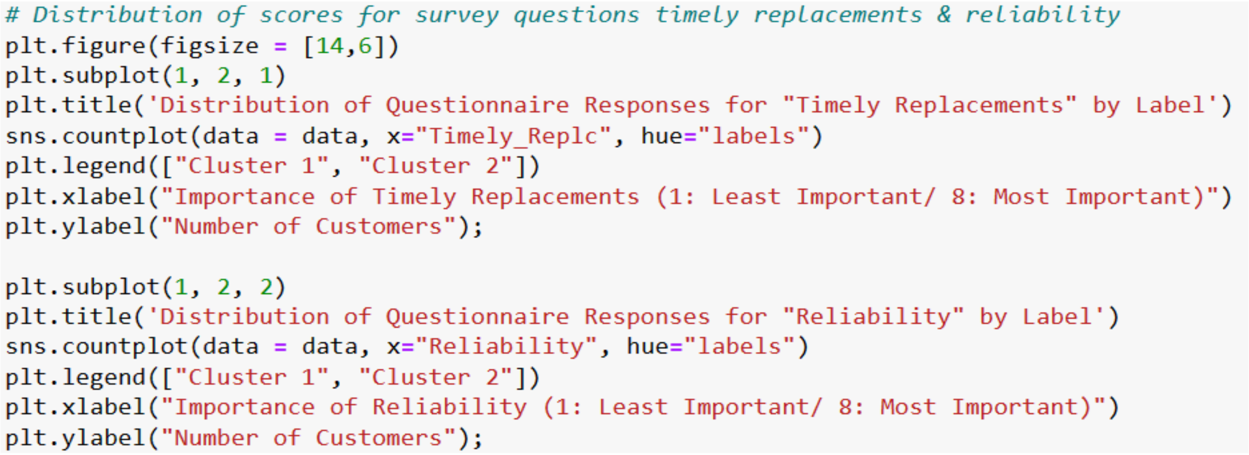


The second visualization was the distribution comparison of each cluster per survey question. The distributions varied across each cluster.

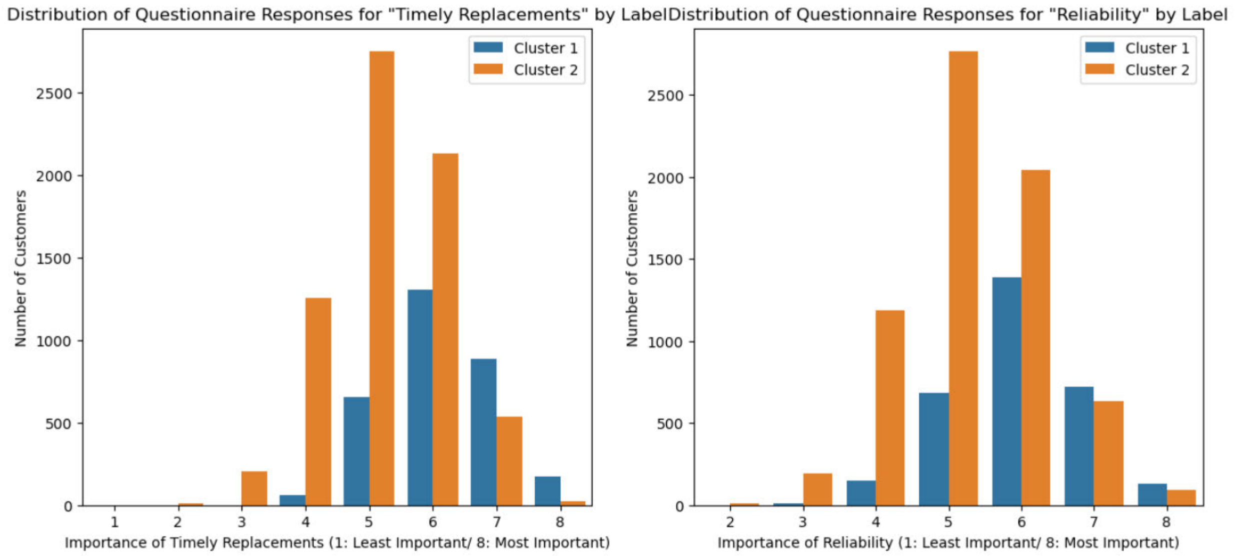


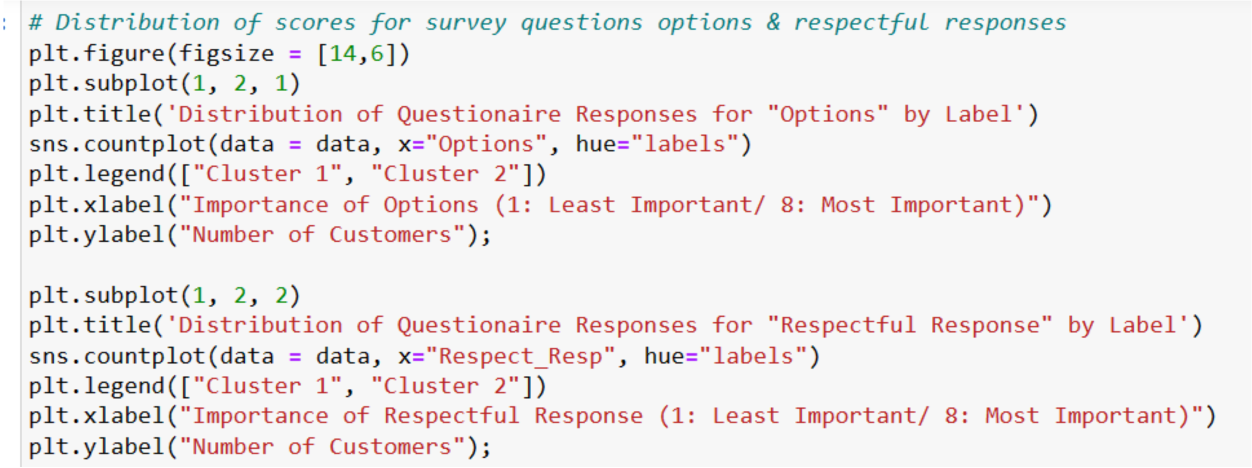
```
# Distribution of scores for survey questions timely responses and timely fixes
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Timely Responses" by Label')
sns.countplot(data = data, x="Timely_Respd", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Responses (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Timely Fixes" by Label')
sns.countplot(data = data, x="Timely_Fixes", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Fixes  (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
```



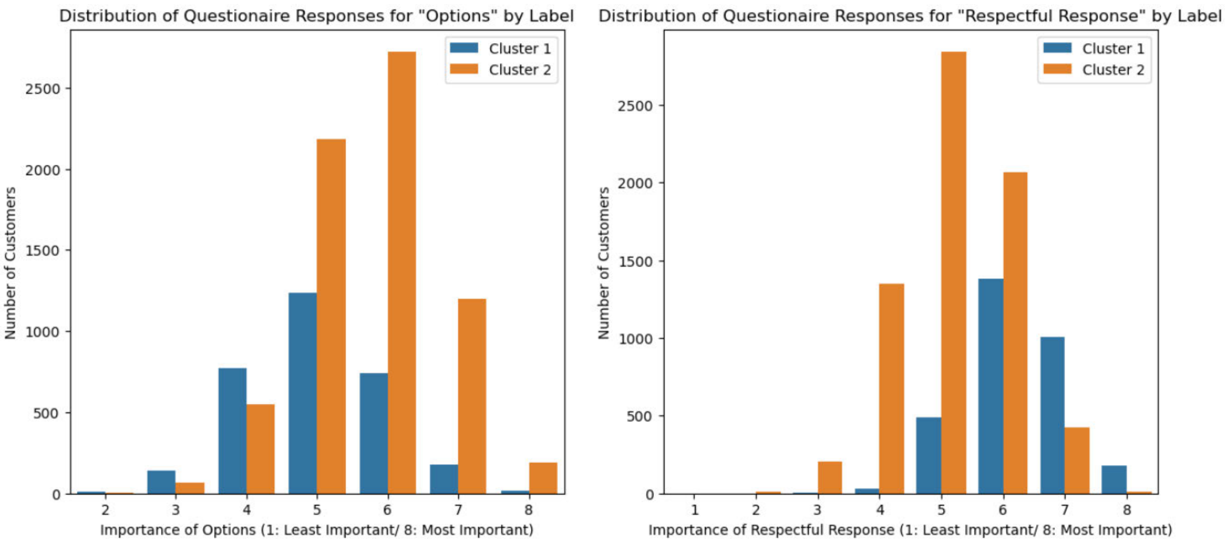


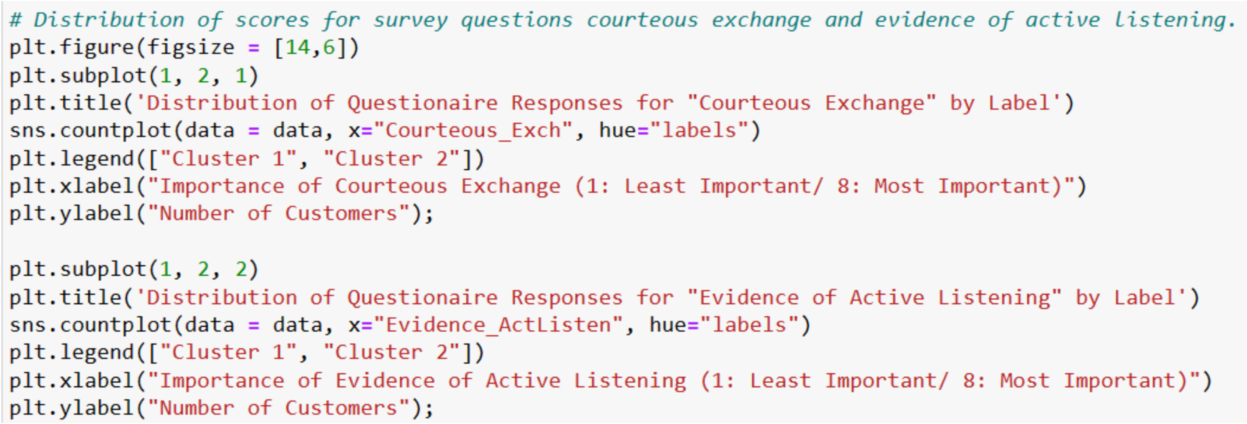
```
# Distribution of scores for survey questions timely replacements & reliability
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Timely Replacements" by Label')
sns.countplot(data = data, x="Timely_Replc", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Replacements (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Reliability" by Label')
sns.countplot(data = data, x="Reliability", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Reliability (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
```



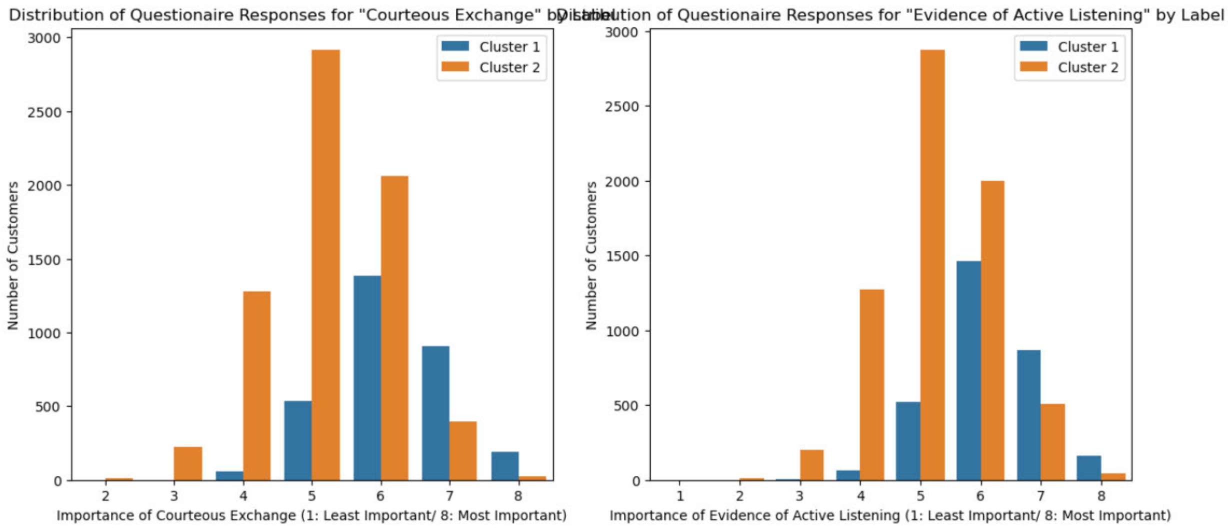


```
# Distribution of scores for survey questions options & respectful responses
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionaire Responses for "Options" by Label')
sns.countplot(data = data, x="Options", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Options (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionaire Responses for "Respectful Response" by Label')
sns.countplot(data = data, x="Respect_Resp", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Respectful Response (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
```





```
# Distribution of scores for survey questions courteous exchange and evidence of active listening.
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionaire Responses for "Courteous Exchange" by Label')
sns.countplot(data = data, x="Courteous_Exch", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Courteous Exchange (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionaire Responses for "Evidence of Active Listening" by Label')
sns.countplot(data = data, x="Evidence_ActListen", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Evidence of Active Listening (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
```



D2. Code Execution

Please see the below Python code for the hierarchical clustering analysis technique.

```
#use linkage to complete hierarchical clustering

matrix_ward= linkage(data[['Timely_Respd', 'Timely_Fixes', 'Timely_Replc',
'Reliability', 'Options', 'Respect_Resp', 'Courteous_Exch',
'Evidence_ActListen']],

                    method='ward',

                    metric= 'euclidean')

#dendrogram results

plt.figure(figsize=(14, 6))

plt.title('Hierarchical Clustering Dendrogram')

plt.xlabel('Survey Responses from Customers')

plt.ylabel('Distance Between Clusters')

dendrogram(matrix_ward)

plt.show()

# Assign cluster labels

data['labels'] = fcluster(matrix_ward, 2, criterion='maxclust')

print(data['labels'].value_counts().sort_index())

#chart of mean/average of questionaire responses.


respd_c1_mean = data.loc[data['labels'] == 1, 'Timely_Respd'].mean()
```

```
respd_c2_mean = data.loc[data['labels'] == 2, 'Timely_Respd'].mean()
fixes_c1_mean = data.loc[data['labels'] == 1, 'Timely_Fixes'].mean()
fixes_c2_mean = data.loc[data['labels'] == 2, 'Timely_Fixes'].mean()
replc_c1_mean = data.loc[data['labels'] == 1, 'Timely_Replc'].mean()
replc_c2_mean = data.loc[data['labels'] == 2, 'Timely_Replc'].mean()
reli_c1_mean = data.loc[data['labels'] == 1, 'Reliability'].mean()
reli_c2_mean = data.loc[data['labels'] == 2, 'Reliability'].mean()
opt_c1_mean = data.loc[data['labels'] == 1, 'Options'].mean()
opt_c2_mean = data.loc[data['labels'] == 2, 'Options'].mean()
rec_c1_mean = data.loc[data['labels'] == 1, 'Respect_Resp'].mean()
rec_c2_mean = data.loc[data['labels'] == 2, 'Respect_Resp'].mean()
cur_c1_mean = data.loc[data['labels'] == 1, 'Courteous_Exch'].mean()
cur_c2_mean = data.loc[data['labels'] == 2, 'Courteous_Exch'].mean()
evid_c1_mean = data.loc[data['labels'] == 1, 'Evidence_ActListen'].mean()
evid_c2_mean = data.loc[data['labels'] == 2, 'Evidence_ActListen'].mean()


survey_mean = {'Cluster 1' : [respd_c1_mean, fixes_c1_mean,
replc_c1_mean, reli_c1_mean, opt_c1_mean, rec_c1_mean, cur_c1_mean,
evid_c1_mean], 'Cluster 2' : [respd_c2_mean, fixes_c2_mean,
replc_c2_mean, reli_c2_mean, opt_c2_mean, rec_c2_mean, cur_c2_mean,
evid_c2_mean]}
survey_summary = pd.DataFrame(data = survey_mean, index=['Timely
Response', 'Timely Fixes', 'Timely Replacements', 'Reliability', 'Options',
'Respectful Responses', 'Courteous Exchange', 'Evidence of Active
Listening'])
survey_summary
#lineplot of survey summary
plt.figure(figsize = [14,6])
sns.lineplot(data = survey_summary, markers=True)
plt.title("Distribution of Mean per Survey Question by Clusters")
plt.xlabel("Survey Question")
plt.ylabel("Mean Score");
# Distribution of scores for survey questions timely responses and timely
fixes
```

```
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Timely Responses" by Label')
sns.countplot(data = data, x="Timely_Respd", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Responses (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Timely Fixes" by Label')
sns.countplot(data = data, x="Timely_Fixes", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Fixes  (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
# Distribution of scores for survey questions timely replacements & reliability
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Timely Replacements" by Label')
sns.countplot(data = data, x="Timely_Replc", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Timely Replacements (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");

plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Reliability" by Label')
sns.countplot(data = data, x="Reliability", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Reliability (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
# Distribution of scores for survey questions options & respectful responses
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Options" by Label')
sns.countplot(data = data, x="Options", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Options (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");


plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Respectful Response" by Label')
sns.countplot(data = data, x="Respect_Resp", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Respectful Response (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");
# Distribution of scores for survey questions courteous exchange and evidence of active listening.
plt.figure(figsize = [14,6])
plt.subplot(1, 2, 1)
plt.title('Distribution of Questionnaire Responses for "Courteous Exchange" by Label')
sns.countplot(data = data, x="Courteous_Exch", hue="labels")
plt.legend(["Cluster 1", "Cluster 2"])
plt.xlabel("Importance of Courteous Exchange (1: Least Important/ 8: Most Important)")
plt.ylabel("Number of Customers");


plt.subplot(1, 2, 2)
plt.title('Distribution of Questionnaire Responses for "Evidence of Active Listening" by Label')

sns.countplot(data = data, x="Evidence_ActListen", hue="labels")

plt.legend(["Cluster 1", "Cluster 2"])

plt.xlabel("Importance of Evidence of Active Listening (1: Least Important/ 8: Most Important)")

plt.ylabel("Number of Customers");
```

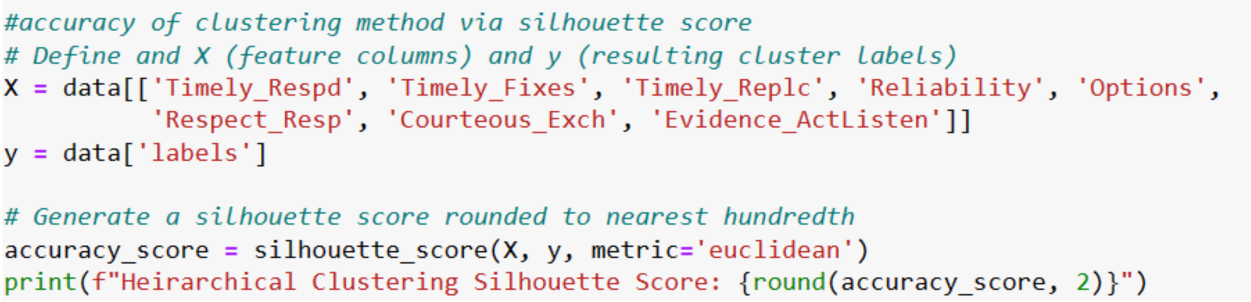
Part V: Data Summary and Implications

E1. Accuracy of Clustering Technique

The accuracy of the clustering technique was evaluated by utilizing the Scikit-Learn silhouette score to compute the silhouette coefficient. The silhouette coefficient or score was a metric used to calculate the goodness of a clustering technique (Bhardwaj, 2020). The silhouette coefficient ranges from -1 to 1. See the below chart to show the difference between the ranges.

Silhouette Score/Coefficient	Meaning
1	Clusters were further apart from each other and easily identified/classified (Bhardwaj, 2020)
0	The clusters were indifferent. The distance between the clusters was not significant and harder to distinguish (Bhardwaj, 2020)
-1	Clusters were assigned the incorrect way (Bhardwaj, 2020).

For the data analysis of the telecommunication data set, the silhouette score was 0.17. The score of 0.17 notated the clusters were close to being indifferent. This could identify that there may be overlapping clusters, but overall, the clustering method was not bad as it was positive and led toward one. The coefficient confirmed there was a difference in the clusters, but not a significant difference. Please see below for the code used to complete the accuracy of the method.



```
#accuracy of clustering method via silhouette score
# Define and X (feature columns) and y (resulting cluster labels)
X = data[['Timely_Respd', 'Timely_Fixes', 'Timely_Replc', 'Reliability', 'Options',
          'Respect_Resp', 'Courteous_Exch', 'Evidence_ActListen']]
y = data['labels']

# Generate a silhouette score rounded to nearest hundredth
accuracy_score = silhouette_score(X, y, metric='euclidean')
print(f"Heirarchical Clustering Silhouette Score: {round(accuracy_score, 2)}")
```

E2. Results and Implications

The results of the hierarchical clustering analysis were summarized by examining the clusters of the questionnaire data. The grouped data was shown in two clusters. The two clusters seemed to have similar comparisons in the same questionnaire responses. The only differences were the level of importance in general. The first cluster put a higher importance on the majority of the questions, whereas the second cluster deemed only two questions more important relative to the other survey questions.

The first cluster, which represented about 31% of the customers, scored the survey nearly the same for the following questions: “Timely Responses”, “Timely Fixes”, “Timely Replacements”, “Respectful Responses”, “Courteous Exchange”, and “Evidence of Active Listening”. The range of the means for those variables was from 6.14 to 6.27. The only noticeably different scores were noted within the questions of “Options” and “Reliability”, which respectively have means of 5.01 and 5.98.

The second cluster, which represented about 69% of the customers, scored the survey similarly for the same questions of “Timely Responses”, “Timely Fixes”, “Timely Replacements”, “Respectful Responses”, “Courteous Exchange”, and “Evidence of Active

Listening". The means for the second cluster ranged from 5.16 to 5.22. There were noted higher scores in the questions of "Options" and "Reliability" which had means of 5.28 and 5.72.

There were some implications to this analysis. The first cluster data provided a strong distinction of the importance of the variables excluding "Options" and "Reliability". The survey score for these variables was about -0.16 to -1.13 difference from the other survey questions. Even though the cluster only represented a small percentage of the customers as a total, the importance of those questions was significantly lower in their consideration. This would imply that the telecommunication company's clientele, would not concern themselves necessarily with these variables. When viewing the data with the second cluster, which represented the majority of the clientele, these same variables were deemed slightly more important than the other variables. Yet the difference between their importance was not as strong as the first cluster. The importance range for these was only 0.06 to 0.50 higher in comparison to the other survey scores in the cluster. The first cluster had a clear distinction for these two questions' importance with their interaction with the telecommunication company., while the second cluster failed to provide a strong distinction.

E3. Limitation

One main limitation of the hierarchical clustering technique is the processing time. Hierarchical clustering has a significant increase in runtime the larger the amount of data points being used. This clustering technique does not work well on vast amounts of data points as it would be inefficient due to the computation processing time needed. Processing time can be computed using the timeit package within Python. This would give an estimation of the execution time. For this analysis, the hierarchical clustering using the ward method was 8.81 s ±

1.83 s per loop. This could be deemed not sufficient if the analysis required a faster execution time.

E4. Course of Action

The recommended course of action for the telecommunication company would be the following suggestion, a revision of the survey response range. The reduction of the survey response range would be beneficial in the analysis. Most response scores were within the “3” to “6” range of importance. The use of “1” to “8” did not allow for an ideal picture of the customers’ satisfaction. It was shown in the difficulty in distinguishing how important a “5” score was compared in relation to a “4” or even a “6” score. Reducing the range to possibly “1” to “5” would allow for the customer’s opinion on the importance level to be clearly identified. It could note “1” as least important and “5” as highly important. The completion of this would provide the company with a key focused area for improvement.

Part VI: Demostration

F. Panopto Recording

Please see attached Panopto video link. This is a video providing an overview of the Python code used for the data analysis. The recording will demonstrate the code’s functionality as well as provide an overview of the Jupyter Notebook programming environment.

Link Found here:

[redacted]

G. Third-Party Web Sources

There were no third party web sources used to acquire data nor segments of third-party code to support the analysis.

H. References

Bhardwaj, A. (2020, May 26). *Silhouette Coefficient*. Retrieved from Toward DataScience: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

Daityari, S. (n.d.). *Cluster Analysis in Python*. Retrieved from DataCamp: https://app.datacamp.com/learn/courses/cluster-analysis-in-python

Yıldırım, S. (2020, April 3). *Hierarchical Clustering — Explained*. Retrieved from Toward DataScience: https://towardsdatascience.com/hierarchical-clustering-explained-e58d2f936323