**D212 Performance Assessment Task 1**

**CLUSTERING TECHNIQUES FOR MEDICAL DATA**

Fahim A. Akbar Student ID 001434895 Masters Data Analytics (January 1, 2021) Program Mentor: Mandy Rasmuson (385) 428-5957x5957 [fakbar3@wgu.edu](mailto:fakbar3@wgu.edu)

**Part I: Research Question**

**A. Data Mining Report Justification**

**A1. Research Question:**

The research question that this analysis will focus on answering is if any meaningful groupings or clusters exist amongst the survey data. The 8 medical survey questions are given to patients to see how their time at the hospital was during their visit. The survey results can give valuable insight on how to improve the patient experience. Each of the survey questions seem to have similar distributions, means, ranges, and standard deviations. This makes extracting specific insights from the data challenging since all 8 variables appear quite similar in their significance for the patients. While the survey results exhibit a similar distribution overall, underlying patterns within the data could still exist. Perhaps half of the patients considered Questions 1-4 to be of greater importance, while Questions 5-8 were viewed as less significant. Conversely, the remaining half of patients might hold the opposite perspective. While this would still result in a similar overall distribution, any significant distinction could be useful for better understanding the patients’ needs and improving their experiences at the hospital.

**A2. Goal**

Using hierarchical clustering, the goal of this data analysis project is to use to see if any meaningful clusters or groups hidden by the overall distribution scores exist in the survey data. Identifying these groups could help the hospital select primary survey variables of interest to one group of patients or another. These groupings could help the hospital simplify their surveys and remove redundancies in the dataset. Furthermore, identifying patterns in the survey data may uncover key variables that hold significant importance for specific patient groups. The hospital could use this knowledge to focus on improving patient satisfaction during their stays.

**Part II: Method Justification**

**B. Reason for using Hierarchical clustering**

**B1. Classification Method: Hierarchical clustering**

For this analysis, the hierarchical clustering method will be used. The hierarchical clustering technique is a method of data analysis that aims to identify inherent patterns and relationships within a selected dataset. It starts by measuring the similarities or dissimilarities between individual data points and then progressively merges them into clusters based on their proximity. This process continues until all data points are grouped into a hierarchical tree-like structure called a dendrogram (Patlolla 2018). In this analysis, SciPy's linkage function will be used to perform hierarchical clustering of the patients survey responses and their survey data. The expected outcome of hierarchical clustering is a visual representation of the dataset's structure, revealing clusters and groupings from the survey data. These will be visualized via dendrograms that show the creation of the clusters and their distance from each other. The dendrograms will be generated using the SciPy's dendrogram function, and from there the best number of clusters to divide the dataset into. Lastly, SciPy's fcluster function will be used to add assign labels to each observation, with the labeling based on the cluster to which it belongs. Once each patient is placed in a cluster, the distribution of the survey responses can be observed by cluster to see if any patterns can be detected.

**B2. Assumptions of Hierarchical clustering**

Hierarchical clustering assumes that a suitable similarity or distance measure exists to quantify the similarity or dissimilarity between data points (CITE). The data points within each cluster are homogeneous, meaning they share similar characteristics or attributes. Hierarchical clustering requires a fusion or linkage criterion to determine how clusters are merged during the agglomeration process. Different criteria such as complete-linkage, single-linkage, or wards-linkage are used, each with its own assumptions about the structure of clusters. (Boehmke 2023). Outliers can affect the clustering results and distort the identification of meaningful patterns, so they should not be present. Assuming a standardized scale for the variables used in clustering beforehand is important because variables with different ranges can result in those with larger ranges dominating those with smaller ranges (Reusova, 2018). In this dataset, the data is already standardized as the values of the survey fall within range of 1 – 8.

**B3. Benefits of Python, and the Python packages and libraries used**

For this analysis, I will be using Python in a Jupyter Notebook environment.

Listed below are the Python packages and libraries that will be used, and how each item supports Hierarchical clustering analysis:

* Pandas – this standard import provides methods to read and visualize data, as well as providing statistical tools to score and parse data.
* Numpy – this standard import provides methods to read and visualize data. It also offers statistical tools to parse and score data.
* Matplotlib – this package is used for data visualization and provides tools to visualize reports and data points
* Seaborn – this package provides descriptive and visually intuitive plots, graphs, and matrices
* Scikit-learn – the silhouette score allows us to evaluate our clustering metric
* SciPy – this package gives us the linkage and fcluster functions used to conduct the hierarchical clustering of datapoints. It also provides us with the

**Part III: Data Preparation**

**C. Data preparation**

**C1. Relevant data preprocessing goal**

One relevant data preprocessing goal for this analysis is to rename the variables accordingly. Currently, the survey questions are listed as Item1-8. Using the provided data dictionary, we will rename each survey question to make sure that we can identify what they represent throughout the analysis. For example, Item3 will be renamed to Timely\_visits based on the provided data dictionary for the medical dataset.

**C2. Identify the target variables for analysis**

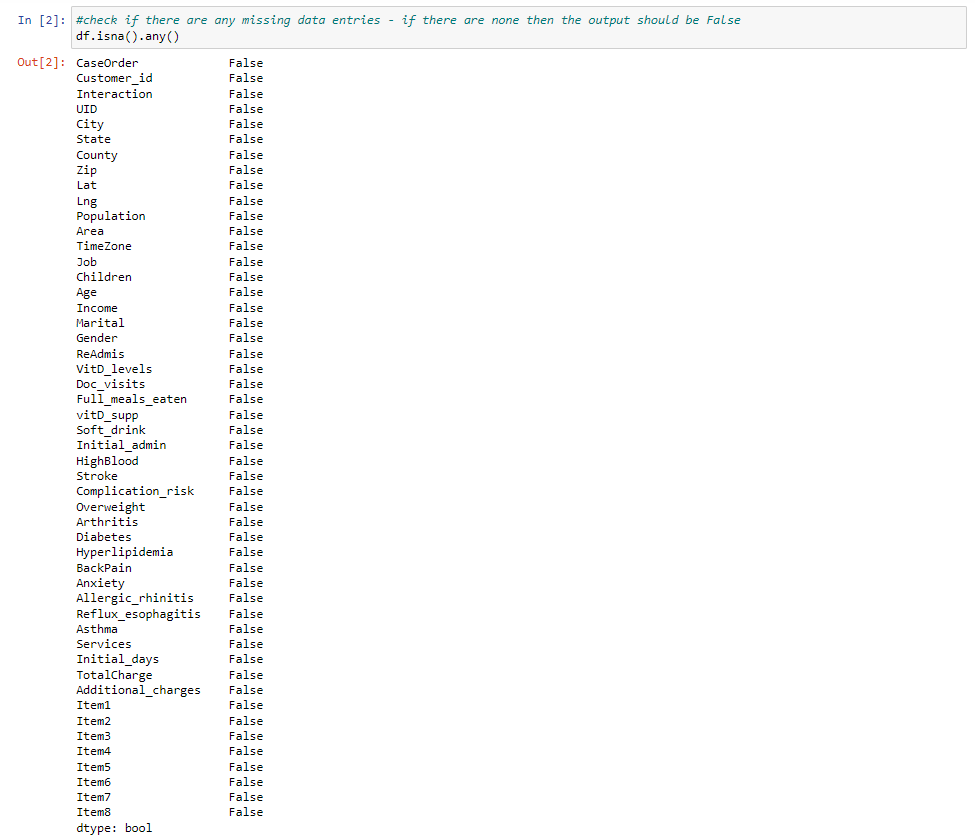
The following categorical variables will be used for our analysis:

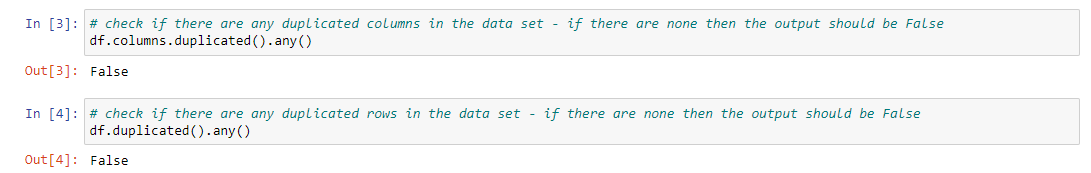
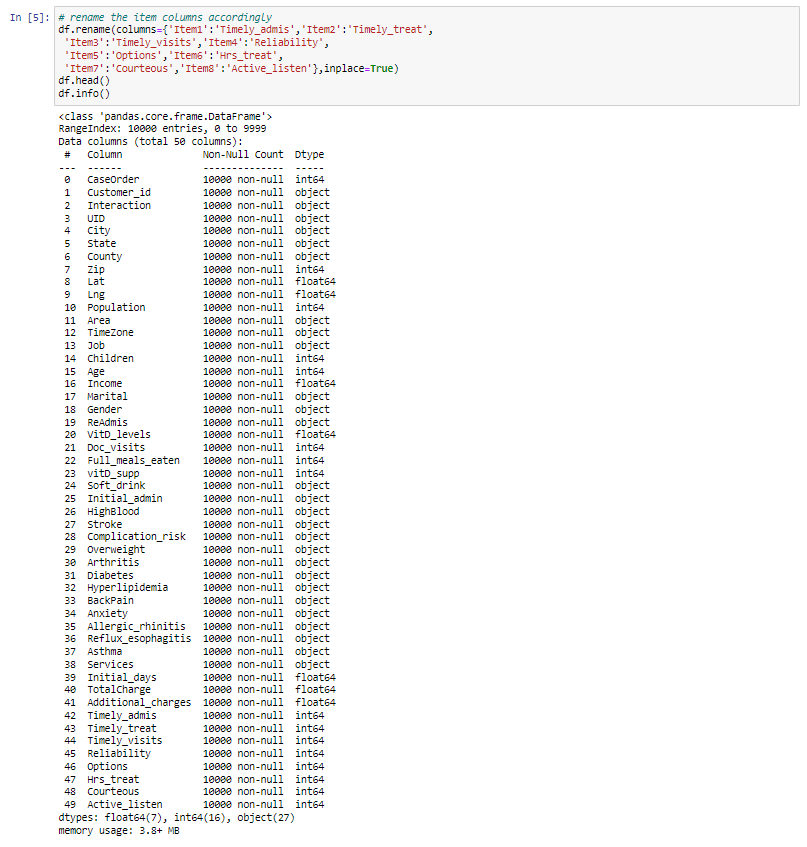
* Item1 will be renamed to: Timely\_admis
* Item2 will be renamed to: Timely\_treat
* Item3 will be renamed to: Timely\_visits
* Item4 will be renamed to: Reliability
* Item5 will be renamed to: Options
* Item6 will be renamed to: Hrs\_treat
* Item7 will be renamed to: Courteous
* Item8 will be renamed to: Active\_listen

Since the scope of this data analysis focuses only on the patient survey responses, other variables from the dataset will not be considered. These variables are all categorical because the numbers for each response represents a relative standard of importance. The survey uses 1 to indicate that the feature is "most important" to the patient, and an 8 indicates that it is "least important" to the patient. Since we’ll be keeping this scale, it’s important to note that final mean scores closer to 1 will indicate that the importance is higher.

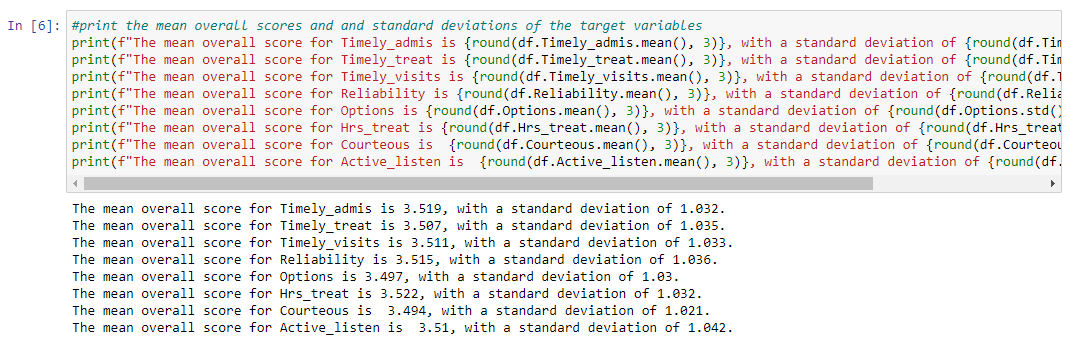
**C3. Data Preparation Steps**

The first step in preparing the data is to make sure no missing data entries are present in any of the columns. The next step is to ensure that there are no duplicated rows or columns to further prevent dealing with repeated entries. Lastly, the columns with our target variables will be renamed to be more descriptive. The preparation of the dataset was performed in a Python using a Jupyter notebook environment. Provided below is screenshots of the data preparation steps.





Before proceeding with the analysis, the homogeneity of the survey data is showcased below.



Both the means and standard deviations of each question are very similar. The means for each question range from 3.494 to 3.522, and the standard deviations range from 1.021 to 1.042.

**C4. Provide a copy of the cleaned dataset.**

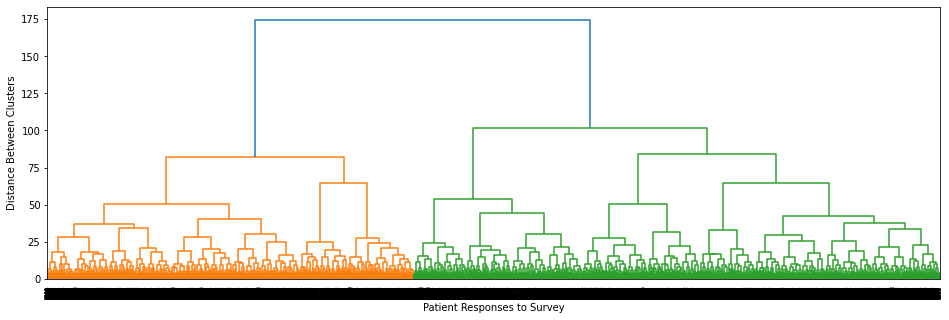
The csv file for the cleaned dataset is provided in the final task submission, and is titled: “1medical\_clean-PREPAREDTASK1d212.csv.”

**Part IV: Analysis**

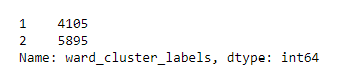
**D. Data Analysis Report**

**D1. Analysis Description**

Hierarchical clustering is used for this data analysis to discover patterns within the survey response data. Three different types of methodology were considered for hierarchical clustering: Single, Complete, and Ward linkage methods. Single linkage calculates all pairwise dissimilarities between all features in clusters 1 and 2, and selects the smallest value from these dissimilarities as the distance between the clusters. This method produces long and less compact clusters. Complete linkage also calculates all pairwise dissimilarities between all features in clusters 1 and 2, but selects the largest value from these dissimilarities as the distance between the clusters. This method produces more compact clusters. Ward’s linkage method works by minimizing the total within-cluster variance. It does so by merging clusters with minimum between-cluster distance (Boehmke 2023). For this analysis, the “ward method” was chosen as the methodology for the linkage function. Using the ward method, the resulting dendrogram showcased an increase of distances among higher-level clusters result in a clear differentiation between the top two clusters. This distinction is made clear by the following dendrogram generated in our analysis:



The y axis showcases the distance between clusters, and the x axis is for the survey results of all 10,000 patients in the dataset. The dendrogram shows multiple clusters in close proximity to one another in regards to distance. The proximity of the clusters to one another diminishes as they grow, which is shown by the longer branches of the dendrogram along the y-axis. Between the final two clusters (shown in orange and green), a distinct threshold emerges as the distance between them becomes significantly larger than in previous steps. This nearly matches the separation of their farthest data points within each cluster, indicating the presence of a distinct and distinguishable relationship from one another. It can therefore be concluded that the ideal model is to divide the data into two clusters, and use the fcluster function to label each of the observations and labels which cluster that observation is from.

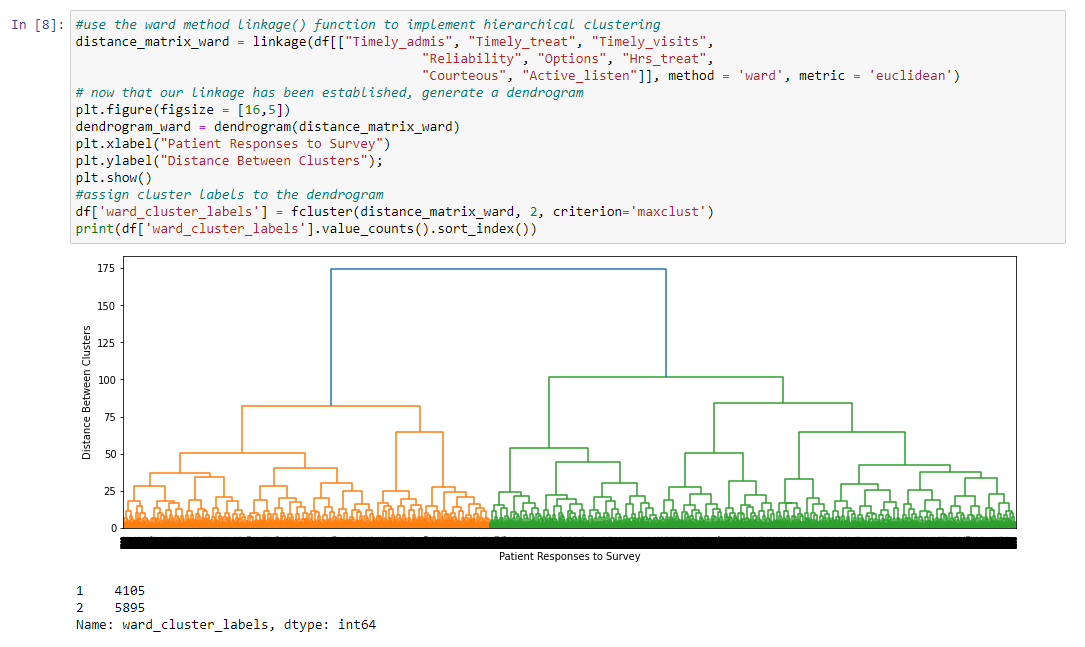


The resulting ward cluster labels show us that 41.05% of the datapoints are in cluster 1, and 58.95% of the datapoints are in cluster 2. Now that two clusters have been identified within the survey data, the analysis can continue to explore differences between these clusters. This will be done by visualizing the distribution of survey responses per question within each cluster and depicting the overall mean of survey responses across all questions per cluster using a line graph. These visualizations are showcased in the Jupyter notebook, as well as section D2 of this report.

**D2. Code used for Hierarchical clustering analysis technique**

The entire code for this project was done in a Jupyter notebook environment. The Jupyter notebook file is attached to the task submission. A pdf copy of the notebook and a txt. file of code used is provided with the task submission as well. Lastly, the entire code used is also provided at the end of the document.Provided below are screenshots of the code used to perform the Hierarchical clustering analysis:

Code used for the ward linkage, dendrogram generation, and fcluster function.



Code used for visualization of the distribution of scores for each survey question by cluster:

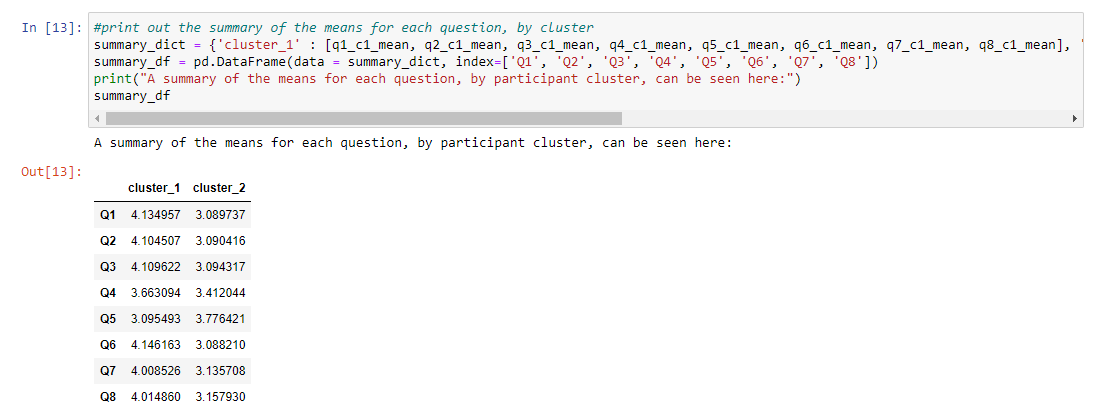


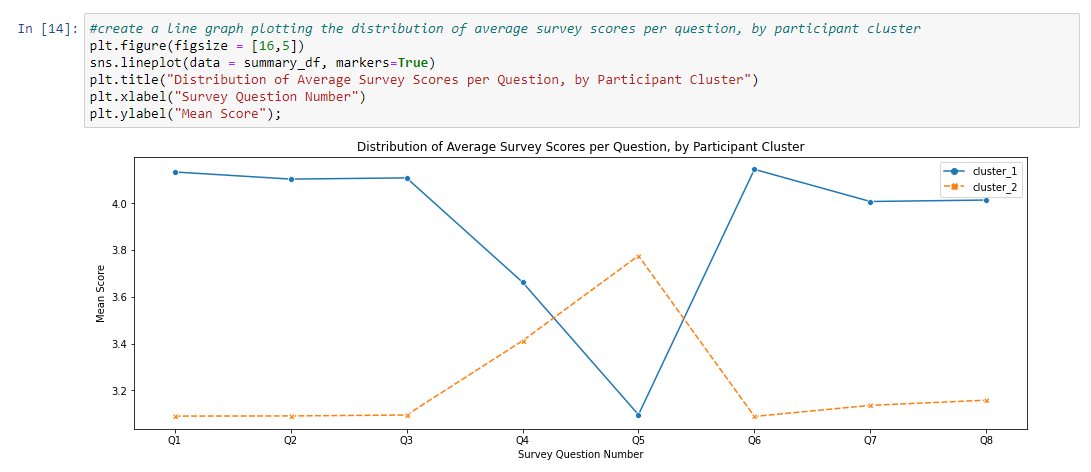






Code used to visualize overall mean of survey responses for all questions by cluster



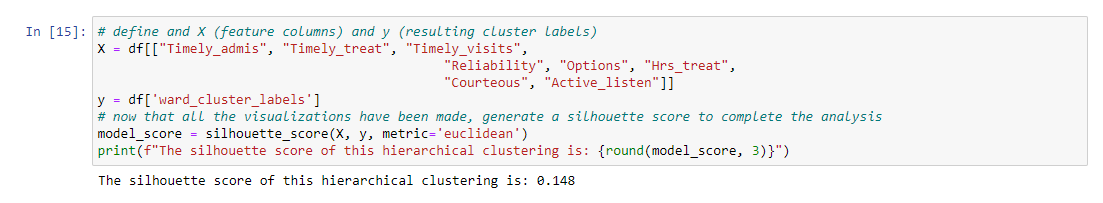


**Part V: Data Summary and Implications**

**E. Data Analysis Summary**

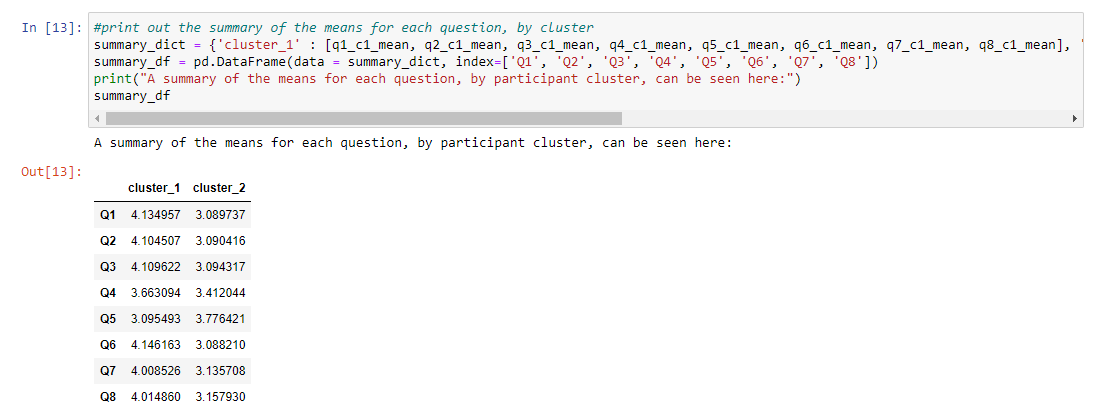
**E1. Accuracy of the clustering technique accuracy**

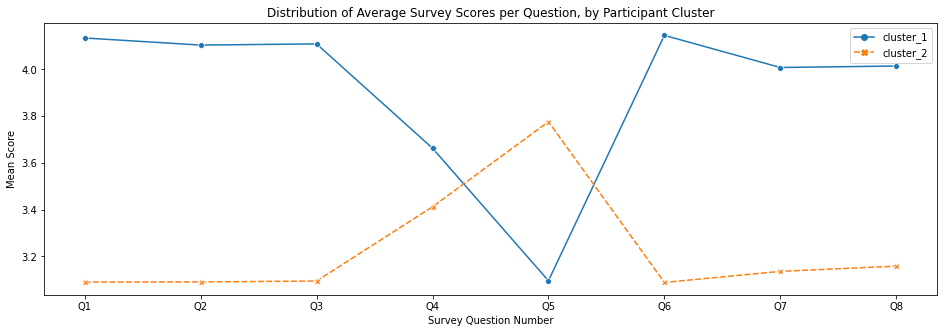
Due to the absence of "true" labels for comparison, a traditional accuracy measurement cannot be computed for this clustering of survey responses. Instead, SciKit Learn's silhouette score can be used to calculate the Silhouette Coefficient, which serves as a measure to evaluate the accuracy of the clustering model.



The resulting Silhouette Coefficient is 0.148, which is a value that falls between -1 and 1, and is close to 0. A silhouette score of -1 would suggest that the model produced poor clustering. A silhouette score 1 suggests that the clusters produced by our model are very well defined and dense. A silhouette score near 0 indicates that the model produced overlapping clusters (Bhardwaj, 2020). While the score of 0.148 isn’t considered bad, the value’s close proximity to 0 suggests that overlap is present between the clusters. This is expected, since the overall data that the clusters were generated from is very homogenous. Furthermore, a dichotomy between these two generated clusters is present, but not immediately pronounced or obvious. This shows that differences are present but not they are very noteworthy. The line graph comparing the means of the survey questions per clusters also supports this conclusion.

**E2. Clustering Analysis Results**



****

The generated hierarchical clustering model divided the survey responses into two clusters. Cluster 2 is the largest cluster. It’s composed of 5895 patients, or roughly 59% of the total observations. The mean scores for Questions 1, 2, 3, 6, 7, and 8 are very close in value, at around 3.08 - 3.15 on the survey score range of 1 - 8. This cluster's survey scores deviate from this average for only Questions 4 and 5, with means of 3.41 and 3.78 respectively. Cluster 1 represents 4105 patients, or approximately 41% of the total observations. Similar to cluster 2, the scores Questions for 1, 2, 3, 6, 7, and 8 very close in value, at around 4.01-4.15. This cluster's survey scores deviate from this average for only Questions 4 and 5, with means of 3.41 and 3.78 respectively. In both clusters, the mean score values for Questions 4 and 5 deviated from the remaining scores. The variation in responses might be attributed to the distinction between Question 4, which assesses the significance of "reliability," and Question 5, which evaluates the importance of "options." These concepts are mentioned without sufficient explanation or contextual information provided, and this apparent vagueness is what may be responsible for the deviation in mean scores. If these concepts were more specific, we could get a more precise response from patients and that could result in mean scores closer to the expected range relative to the other questions. Furthermore, the two clusters show us that patients in cluster 1 considered the survey items as less important than the patients in cluster 2. Even though the mean score values are higher, it’s important to recall that the survey ranks items from a scale of 1-8, and a score 1 represents the most important while a score of 8 represents the least important. Therefore, the mean score values of cluster 1 being higher than the mean score values of cluster 2 represents scores closer to 8, or in other words, less importance. With that in mind, the following conclusions can be drawn. First off, its apparent that the patients of both clusters collectively have a similar level of importance regarding Questions 1, 2, 3, 6, 7, and 8 when compared to each other. Although cluster 1 assigns higher scores to those questions compared to cluster 2, both clusters display consistency by maintaining relatively equal scores for each of the six questions within their respective clusters. The second conclusion that can be made is that cluster 1 patients appear to express a preference to the importance of "reliability" and "options" over the other six features of the survey. In contrast, patients from cluster 2 appear to consider "reliability" and "options" less important than the other six features of the survey

**E3. Data Analysis Limitations**

One limitation to this analysis lies that the nature of selecting a number of optimal clusters to divide the survey into was rather subjective and reliant on using the ward linkage method. Fortunately for this analysis, the two clusters derived showed distinct differences in mean scores across survey features. A separate analysis using smaller clusters could find additional differences amongst the patient observations not found by only dividing the population into two clusters. These additional clusters might not tell us much more since the two clusters derived were not that far apart, but it is worth considering given that the dataset is comprised of a large population of 10,000 patients. Another limitation encountered was how vague the "reliability" and "options" features of the survey were. Significant deviations for the mean scores of these two questions were found in both clusters.

**E4. Recommend course of action**

One recommended course of action is to reduce the range for survey responses. Having a range of 1-8 caused almost all of the questions being bound towards a center gradient of values. Using a smaller scale from 1 (most important) to 5 (least important) would be more informative, and give us a more accurate assessment of the importance of different features to each patient can be obtained. This would align with the prevailing usage pattern of the survey among patients, as the majority of responses predominantly concentrate on the mid-range values of the survey. Another recommendation for the hospital would be to change the wording of the patient survey to be more specific. For the “options” survey question, this feature could be more beneficial to our analysis if we knew whether it was referring to options for treatment, payment, and insurance. Likewise, the same could be said for the “reliability” survey question that was also vague in description. Furthermore, the lack of clarity for some of the survey questions could keep patients from giving the best and most accurate responses possible. Lastly, another recommendation I would make would be to further divide survey responses into groups beforehand based on the reason for admission of each patient. There, we can perform multiple hierarchical clusters and uncover additional patterns specific to causes of admission, such as cancer patients.

**Part VI: Demonstration**

**F. Panopto video recording**

A link to the Panopto recording is provided below, and is also attached in the final submission:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8453ac19-4f49-4ba7-b5ed-b02e0163dcfd>

**G. Third Party Code Used**

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html>

<https://learn.saylor.org/mod/book/view.php?id=55626&chapterid=41483#:~:text=clustering%20results%20are.-,Silhouette,-In%20contrast%20to>

<https://www.kaggle.com/discussions/getting-started/134454>

<https://stackoverflow.com/questions/27875056/how-to-calculate-silhouette-score-of-the-scipys-fcluster-using-scikit-learn-sil>

<https://www.geeksforgeeks.org/python-pandas-dataframe-mean/>

<https://seaborn.pydata.org/generated/seaborn.lineplot.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html>

<https://goodboychan.github.io/python/datacamp/machine_learning/2020/06/06/01-Introduction-to-Clustering.html>

<https://seaborn.pydata.org/generated/seaborn.countplot.html>

**H. Additional Sources**

Bhardwaj, A. (2020, May 26). Silhouette Coefficient: Validating Clustering Techniques. Towards Data Science. Retrieved June 6, 2023, from <https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c>

Boehmke, B. (2023). Hierarchical Clustering. UC Business Analytics R Programming Guide. Retrieved June 6, 2023, from <https://uc-r.github.io/hc_clustering>

Patlolla, C. R. (2018, December 10). Understanding the Concept of Hierarchical Clustering Technique. Towards Data Science. Retrieved June 6, 2023, from <https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec>

Reusova, A. (2018, April 1). Hierarchical Clustering on Categorical Data in R. Towards Data Science. Retrieved June 6, 2023, from <https://towardsdatascience.com/hierarchical-clustering-on-categorical-data-in-r-a27e578f2995>

**Full code used for the project:**

**DATA CLEANING AND PREPARATION CODE**

#import packages and clean data before running the principal component analysis

import numpy as np

import pandas as pd

from sklearn import linear\_model

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

pd.set\_option('display.max\_columns', None)

import pylab

from pylab import rcParams

import statsmodels.api as sm

import statistics

from scipy import stats

from scipy.cluster.hierarchy import linkage, fcluster

from scipy.cluster.hierarchy import dendrogram

from sklearn.metrics import silhouette\_score

import sklearn

from sklearn import preprocessing

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.metrics import classification\_report

from scipy.stats import chisquare

from scipy.stats import chi2\_contingency

from sklearn.decomposition import PCA

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

from sklearn.metrics import accuracy\_score

df = pd.read\_csv (r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean.csv')

df.head()

df.info()

#check if there are any missing data entries - if there are none then the output should be False

df.isna().any()

# check if there are any duplicated columns in the data set - if there are none then the output should be False

df.columns.duplicated().any()

# check if there are any duplicated rows in the data set - if there are none then the output should be False

df.duplicated().any()

# rename the item columns accordingly

df.rename(columns={'Item1':'Timely\_admis','Item2':'Timely\_treat',

'Item3':'Timely\_visits','Item4':'Reliability',

'Item5':'Options','Item6':'Hrs\_treat',

'Item7':'Courteous','Item8':'Active\_listen'},inplace=True)

df.head()

df.info()

**CODE FOR SHOWCASING DATA HOMOGENEITY AND EXPORTING THE FINAL DATAFRAME**

#print the mean overall scores and and standard deviations of the target variables

print(f"The mean overall score for Timely\_admis is {round(df.Timely\_admis.mean(), 3)}, with a standard deviation of {round(df.Timely\_admis.std(), 3)}.")

print(f"The mean overall score for Timely\_treat is {round(df.Timely\_treat.mean(), 3)}, with a standard deviation of {round(df.Timely\_treat.std(), 3)}.")

print(f"The mean overall score for Timely\_visits is {round(df.Timely\_visits.mean(), 3)}, with a standard deviation of {round(df.Timely\_visits.std(), 3)}.")

print(f"The mean overall score for Reliability is {round(df.Reliability.mean(), 3)}, with a standard deviation of {round(df.Reliability.std(), 3)}.")

print(f"The mean overall score for Options is {round(df.Options.mean(), 3)}, with a standard deviation of {round(df.Options.std(), 3)}.")

print(f"The mean overall score for Hrs\_treat is {round(df.Hrs\_treat.mean(), 3)}, with a standard deviation of {round(df.Hrs\_treat.std(), 3)}.")

print(f"The mean overall score for Courteous is {round(df.Courteous.mean(), 3)}, with a standard deviation of {round(df.Courteous.std(), 3)}.")

print(f"The mean overall score for Active\_listen is {round(df.Active\_listen.mean(), 3)}, with a standard deviation o

#now that we have our final dataframe, save and export this dataframe as a CSV file

df.to\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d212\1medical\_clean-PREPAREDTASK1d212.csv', index=False)

**CODE USED FOR THE WARD LINKAGE METHOD AND DENDROGRAM CREATION**

#use the ward method linkage() function to implement hierarchical clustering

distance\_matrix\_ward = linkage(df[["Timely\_admis", "Timely\_treat", "Timely\_visits",

"Reliability", "Options", "Hrs\_treat",

"Courteous", "Active\_listen"]], method = 'ward', metric = 'euclidean')

# now that our linkage has been established, generate a dendrogram

plt.figure(figsize = [16,5])

dendrogram\_ward = dendrogram(distance\_matrix\_ward)

plt.xlabel("Patient Responses to Survey")

plt.ylabel("Distance Between Clusters");

plt.show()

#assign cluster labels to the dendrogram

df['ward\_cluster\_labels'] = fcluster(distance\_matrix\_ward, 2, criterion='maxclust')

print(df['ward\_cluster\_labels'].value\_counts().sort\_index())

**CODE USED FOR PLOTTING THE DISTRIBUTION SCORES OF EACH QUESTION BY CLUSTER**

#plot the distribution scores for survey questions 1 and 2

plt.figure(figsize = [16,5])

# LEFT plot: Distribution of scores for survey question 1, by cluster label

plt.subplot(1, 2, 1)

plt.title('Distribution of Q1 Survey Scores by Cluster Label')

sns.countplot(data = df, x="Timely\_admis", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Timely Admission (1 = most important)")

plt.ylabel("Number of Patients");

# RIGHT plot: Distribution of scores for survey question 2, by cluster label

plt.subplot(1, 2, 2)

plt.title("Distribution of Q2 Survey Scores by Cluster Label")

sns.countplot(data = df, x="Timely\_treat", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Timely Treatment (1 = most important)")

plt.ylabel("Number of Patients");

q1\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Timely\_admis'].mean()

q1\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Timely\_admis'].mean()

print(f"\nFor Importance of Timely Admission respondents from Cluster 1 scored this at {round(q1\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q1\_c2\_mean, 3)}, on average.")

q2\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Timely\_treat'].mean()

q2\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Timely\_treat'].mean()

print(f"\nFor Importance of Timely Treatment, respondents from Cluster 1 scored this at {round(q2\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q2\_c2\_mean, 3)}, on average.")

#plot the distribution scores for survey questions 3 and 4

plt.figure(figsize = [16,5])

# LEFT plot: Distribution of scores for survey question 3, by cluster label

plt.subplot(1, 2, 1)

plt.title('Distribution of Q3 Survey Scores by Cluster Label')

sns.countplot(data = df, x="Timely\_visits", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Timely Visits (1 = most important)")

plt.ylabel("Number of Patients");

# RIGHT plot: Distribution of scores for survey question 4, by cluster label

plt.subplot(1, 2, 2)

plt.title("Distribution of Q4 Survey Scores by Cluster Label")

sns.countplot(data = df, x="Reliability", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Reliability (8 = most important)")

plt.ylabel("Number of Patients");

q3\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Timely\_visits'].mean()

q3\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Timely\_visits'].mean()

print(f"\nFor Importance of Timely Visits, respondents from Cluster 1 scored this at {round(q3\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q3\_c2\_mean, 3)}, on average.")

q4\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Reliability'].mean()

q4\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Reliability'].mean()

print(f"\nFor Importance of Reliability, respondents from Cluster 1 scored this at {round(q4\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q4\_c2\_mean, 3)}, on average.")

#plot the distribution scores for survey questions 5 and 6

plt.figure(figsize = [16,5])

# LEFT plot: Distribution of scores for survey question 5, by cluster label

plt.subplot(1, 2, 1)

plt.title('Distribution of Q5 Survey Scores by Cluster Label')

sns.countplot(data = df, x="Options", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Options (8 = most important)")

plt.ylabel("Number of Patients");

# RIGHT plot: Distribution of scores for survey question 6, by cluster label

plt.subplot(1, 2, 2)

plt.title("Distribution of Q6 Survey Scores by Cluster Label")

sns.countplot(data = df, x="Hrs\_treat", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Hours of Treatment (8 = most important)")

plt.ylabel("Number of Patients");

q5\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Options'].mean()

q5\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Options'].mean()

print(f"\nFor Importance of Options, respondents from Cluster 1 scored this at {round(q5\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q5\_c2\_mean, 3)}, on average.")

q6\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Hrs\_treat'].mean()

q6\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Hrs\_treat'].mean()

print(f"\nFor Importance of Hours of Treatment, respondents from Cluster 1 scored this at {round(q6\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q6\_c2\_mean, 3)}, on average.")

#plot the distribution scores for survey questions 7 and 8

plt.figure(figsize = [16,5])

# LEFT plot: Distribution of scores for survey question 7, by cluster label

plt.subplot(1, 2, 1)

plt.title('Distribution of Q7 Survey Scores by Cluster Label')

sns.countplot(data = df, x="Courteous", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Courteous Staff (8 = most important)")

plt.ylabel("Number of Patients");

# RIGHT plot: Distribution of scores for survey question 8, by cluster label

plt.subplot(1, 2, 2)

plt.title("Distribution of Q8 Survey Scores by Cluster Label")

sns.countplot(data = df, x="Active\_listen", hue="ward\_cluster\_labels")

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

plt.xlabel("Importance of Dr Active Listening (8 = most important)")

plt.ylabel("Number of Patients");

q7\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Courteous'].mean()

q7\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Courteous'].mean()

print(f"\nFor Importance of Courteous Staff, respondents from Cluster 1 scored this at {round(q7\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q7\_c2\_mean, 3)}, on average.")

q8\_c1\_mean = df.loc[df['ward\_cluster\_labels'] == 1, 'Active\_listen'].mean()

q8\_c2\_mean = df.loc[df['ward\_cluster\_labels'] == 2, 'Active\_listen'].mean()

print(f"\nFor Importance of Dr Active Listening, respondents from Cluster 1 scored this at {round(q8\_c1\_mean, 3)}, on average.")

print(f"Respondents from Cluster 2 scored this at {round(q8\_c2\_mean, 3)}, on average.")

**CODE USED FOR PRINTING OUT THE SUMMARY OF MEAN SCORES FOR EACH QUESTION BY CLUSTER**

#print out the summary of the means for each question, by cluster

summary\_dict = {'cluster\_1' : [q1\_c1\_mean, q2\_c1\_mean, q3\_c1\_mean, q4\_c1\_mean, q5\_c1\_mean, q6\_c1\_mean, q7\_c1\_mean, q8\_c1\_mean], 'cluster\_2' : [q1\_c2\_mean, q2\_c2\_mean, q3\_c2\_mean, q4\_c2\_mean, q5\_c2\_mean, q6\_c2\_mean, q7\_c2\_mean, q8\_c2\_mean]}

summary\_df = pd.DataFrame(data = summary\_dict, index=['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8'])

print("A summary of the means for each question, by participant cluster, can be seen here:")

summary\_df

**CODE USED FOR VISUALIZING THE DISTRIBUTION OF MEAN SCORES FOR EACH QUESTION BY CLUSTER**

#create a line graph plotting the distribution of average survey scores per question, by participant cluster

plt.figure(figsize = [16,5])

sns.lineplot(data = summary\_df, markers=True)

plt.title("Distribution of Average Survey Scores per Question, by Participant Cluster")

plt.xlabel("Survey Question Number")

plt.ylabel("Mean Score");

**CODE USED TO FIND THE SILHOUETTE SCORE**

# define and X (feature columns) and y (resulting cluster labels)

X = df[["Timely\_admis", "Timely\_treat", "Timely\_visits",

"Reliability", "Options", "Hrs\_treat",

"Courteous", "Active\_listen"]]

y = df['ward\_cluster\_labels']

# now that all the visualizations have been made, generate a silhouette score to complete the analysis

model\_score = silhouette\_score(X, y, metric='euclidean')

print(f"The silhouette score of this hierarchical clustering is: {round(model\_score, 3)}")