D603 – Machine Learning

Performance Assessment #2 – Clustering Techniques

Bernard Connelly

Master of Science, Data Analytics, Western Governors University

Dr. Sherin Aly

March 16, 2025

Performance Assessment #2 – Clustering Techniques

Part 1: Purpose & Justification

B1. Business Proposal

Does a patient's health metrics and services received post-admission impact re-admission rates to the hospital?

B2. Goal of Analysis

The goal will be to determine if the health conditions, choices, and care received upon admission to the hospital impact whether or not a patient is re-admitted within a month to the hospital. The goal will be to remove demographic and financial factors to focus on the effects a patient's overall health, treatment received, personal choices, and specific details surrounding their initial admission have on re-admission rates.

C1. Explanation of Clustering Technique

The clustering technique selected for this analysis is hierarchical clustering. Specifically, I will be using Agglomerative Clustering to accomplish this task. This method builds a hierarchy of clusters by moving through multiple iterations and merging data points based on their similarity. The final output of the clustering technique is commonly visualized with a dendrogram. More specifically, I utilized Gower's Distance to generate a distance matrix that quantifies how similar each patient is to the others. This sets a baseline for agglomerative clustering to treat each data point as its own cluster. It will iteratively merge the two closest clusters until a predefined number of clusters is reached.

C2. Assumption of Hierarchical Clustering

One of the key assumptions of Hierarchical Clustering is that the data points can be compared using a distance metric, even when the data contains mixed data types. In my selected research question, continuous and categorical data types are utilized, making K-Means clustering a poor option. To maximize my ability to handle mixed variables, I utilized Gower's Distance as the distance metric, as the Euclidean distance typical for K-Means is not viable unless the data is only continuous (Muhammad, 2023).

C3. Packages & Libraries Used

Numpy and Pandas were imported for general calculations in Python and data frames, respectively. IPython.display was imported as an alternative to using the print command to export data or multiple lines of code. Matplotlib was imported to generate cluster visualizations. The gower package was imported to calculate Gower's Distance and create a distance matrix for clustering, and squareform was imported from scipy.spacial.distance to convert Gower's Distance into a condensed format after the clustering would not process without it. The sklearn.cluster was utilized for AgglomerativeClustering, and sklearn.metrics imported silhouette_score to assist in identifying the optimal amount of clusters to select for the analysis. Finally, UMAP was used and imported to reduce dimensionality when visualizing the clusters.

Part 2: Data Preparation

D1. Preprocessing Goal

To prepare the dataset for Hierarchical Clustering, several processing steps were taken. Binary Categorical variables such as “highblood” or “asthma” were converted to numerical values 0 and 1. Ordinal variables such as “complication_risk,” which housed low, medium, and high as options, were converted to numerical values 0, 1, and 2 to preserve their rank and order. Finally, one-hot encoding was utilized for other nominal categorical variables such as “initial_admin” or “services” to ensure no artificial ordinal relationships were formed. This preprocessing was necessary because unprocessed categorical variables cannot be used in distance-based clustering methods without being properly encoded. Specifically, the goal was to ensure the dataset was prepared for meaningful groupings and distance calculations.

D2. Identification & Classification of Variables for Analysis

The breakdown of the datatypes remaining for analysis is as seen below.

Categorical Variables:

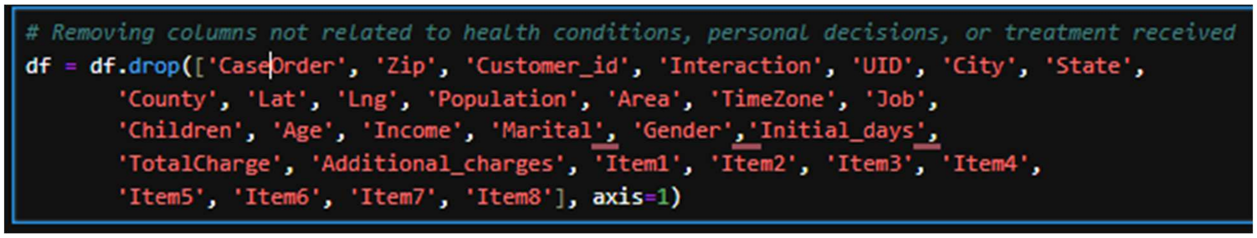
'readmis', 'soft_drink', 'highblood', 'stroke', 'complication_risk', 'overweight', 'arthritis', 'diabetes', 'hyperlipidemia', 'backpain', 'anxiety', 'allergic_rhinitis', 'reflux_esophagitis', 'asthma', 'initial_admin, 'services’

Numeric Variables:

'vitd_levels', 'doc_visits', 'full_meals_eaten', 'vitd_supp'

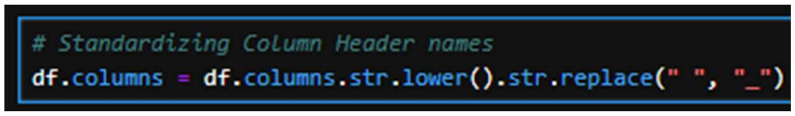
D3 + D4. Explanation of Cleaning & Data File

After importing the medical_clean.csv data file, the first step I took was to drop the columns that were not directly related to the business proposal. All columns related to individual demographics and finances were removed to ensure the analysis targeted the patient's health, the reasons for admission, the services received, and lifestyle choices. In addition, the indexes and survey responses were also dropped as they were irrelevant to the question.



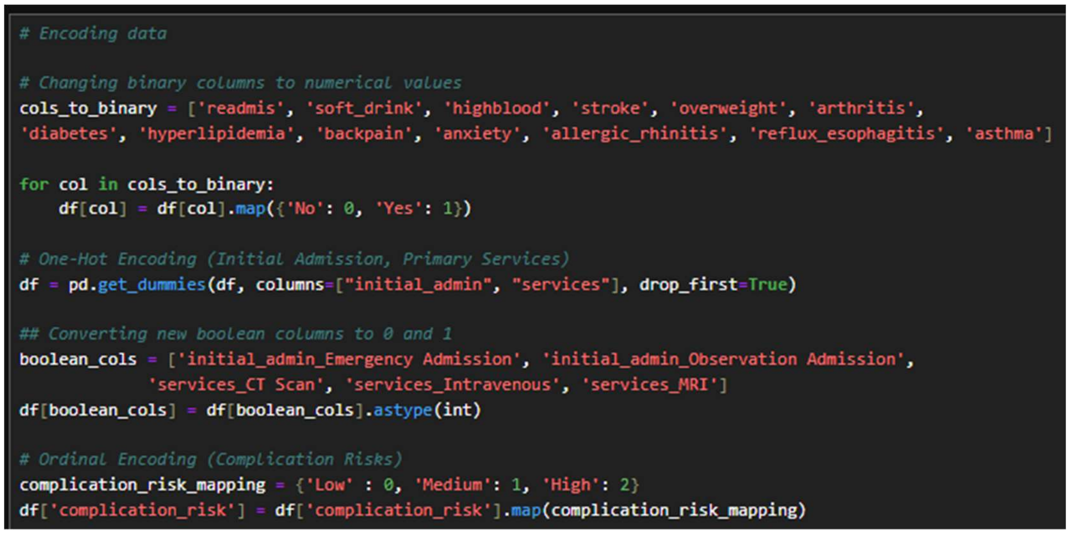
```
# Removing columns not related to health conditions, personal decisions, or treatment received
df = df.drop(['CaseOrder', 'Zip', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
       'County', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job',
       'Children', 'Age', 'Income', 'Marital', 'Gender','Initial_days',
       'TotalCharge', 'Additional_charges', 'Item1', 'Item2', 'Item3', 'Item4',
       'Item5', 'Item6', 'Item7', 'Item8'], axis=1)
```

Subsequently, the column headers were standardized per Python convention by changing spaces to underscores and removing capital letters.



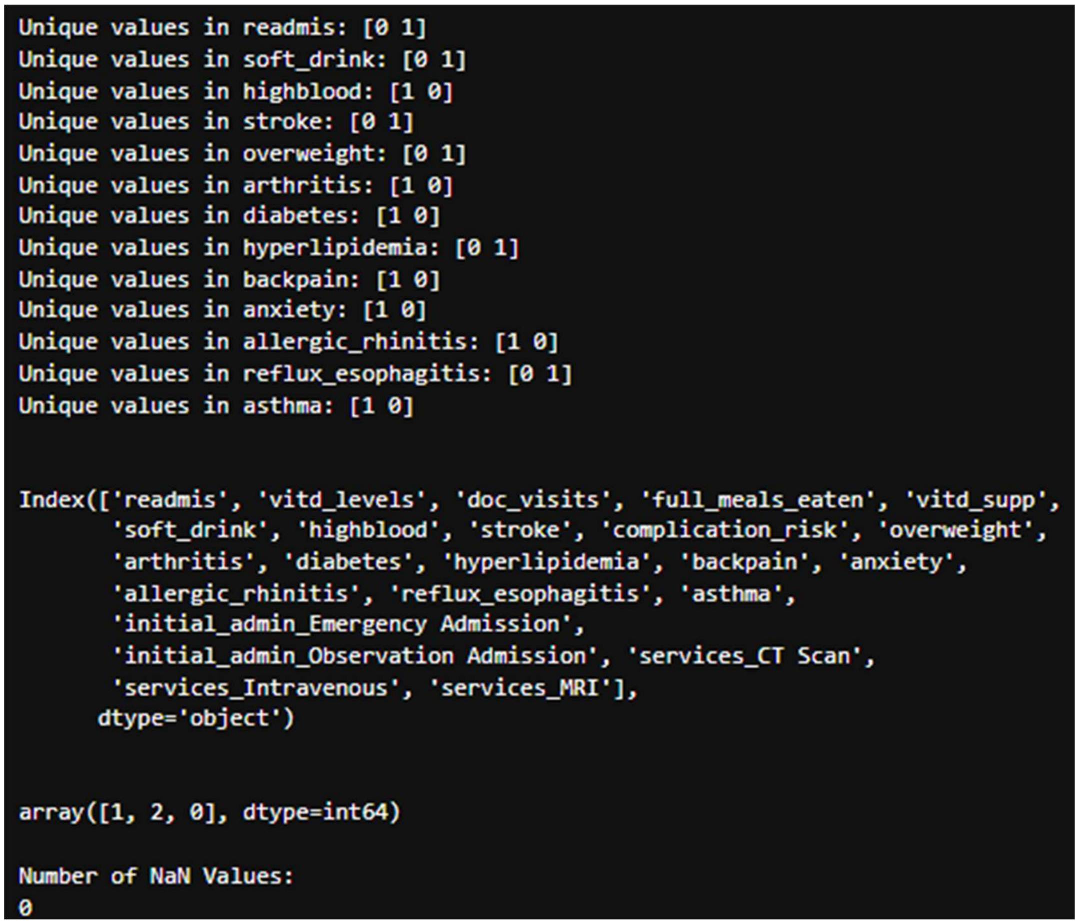
```
# Standardizing Column Header names
df.columns = df.columns.str.lower().str.replace(" ", "_")
```

After confirming that no null values remained in the dataset using an isnull().sum() statement, I moved on to profiling the data as described to verify the best methods to encode the remaining variables properly for Hierarchical Clustering. After review, I identified the binary columns to change from Yes/No to 1/0 and the categorical variables to either one-hot or ordinally encode, depending on their outputs. As a part of the one-hot encoding, I also converted all the new Boolean columns to 0/1.



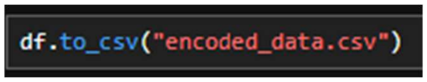
```
# Encoding data

# Changing binary columns to numerical values
cols_to_binary = ['readmis', 'soft_drink', 'highblood', 'stroke', 'overweight', 'arthritis',
'diabetes', 'hyperlipidemia', 'backpain', 'anxiety', 'allergic_rhinitis', 'reflux_esophagitis', 'asthma']

for col in cols_to_binary:
    df[col] = df[col].map({'No': 0, 'Yes': 1})

# One-Hot Encoding (Initial Admission, Primary Services)
df = pd.get_dummies(df, columns=["initial_admin", "services"], drop_first=True)

## Converting new boolean columns to 0 and 1
boolean_cols = ['initial_admin_Emergency Admission', 'initial_admin_Observation Admission',
             'services_CT Scan', 'services_Intravenous', 'services_MRI']
df[boolean_cols] = df[boolean_cols].astype(int)

# Ordinal Encoding (Complication Risks)
complication_risk_mapping = {'Low' : 0, 'Medium': 1, 'High': 2}
df['complication_risk'] = df['complication_risk'].map(complication_risk_mapping)
```

My final step for cleaning before exporting the file was to validate the changes made and prepare the dataset for further analysis. In this step, I also ensured none of the replaced values came back as NaN values, as those would also affect the analysis further down the line.



```
Unique values in readmis: [0 1]
Unique values in soft_drink: [0 1]
Unique values in highblood: [1 0]
Unique values in stroke: [0 1]
Unique values in overweight: [0 1]
Unique values in arthritis: [1 0]
Unique values in diabetes: [1 0]
Unique values in hyperlipidemia: [0 1]
Unique values in backpain: [1 0]
Unique values in anxiety: [1 0]
Unique values in allergic_rhinitis: [1 0]
Unique values in reflux_esophagitis: [0 1]
Unique values in asthma: [1 0]


Index(['readmis', 'vitd_levels', 'doc_visits', 'full_meals_eaten', 'vitd_supp',
       'soft_drink', 'highblood', 'stroke', 'complication_risk', 'overweight',
       'arthritis', 'diabetes', 'hyperlipidemia', 'backpain', 'anxiety',
       'allergic_rhinitis', 'reflux_esophagitis', 'asthma',
       'initial_admin_Emergency Admission',
       'initial_admin_Observation Admission', 'services_CT Scan',
       'services_Intravenous', 'services_MRI'],
      dtype='object')


array([1, 2, 0], dtype=int64)

Number of NaN Values:
0
```

Finally, the dataset was exported.

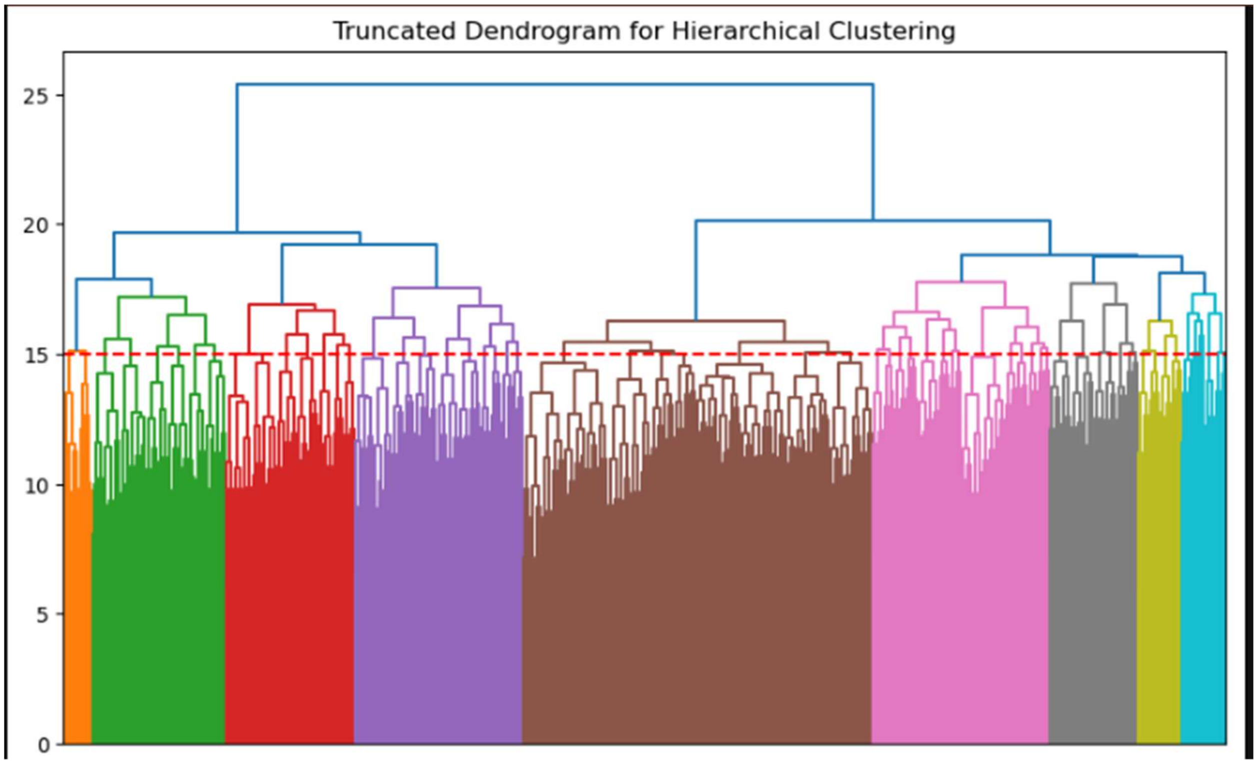


```
df.to_csv("encoded_data.csv")
```

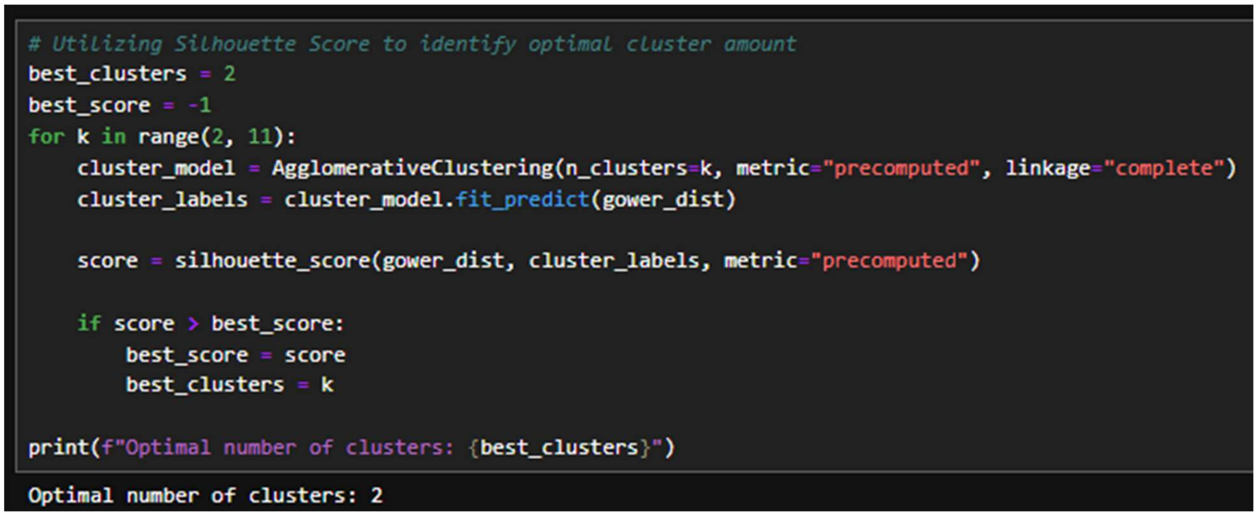
Part 3: Clustering Analysis

E1. Optimizing Clusters

The elbow method could not be utilized for this clustering method as hierarchical clustering does not have sum-of-squared distances like K-Means. As a result, an alternative approach was required. Typically, dendrograms would be helpful to assist with this visualization; however, in this scenario, the size of the Dendrogram in this scenario was too complex to be easily interpreted.`

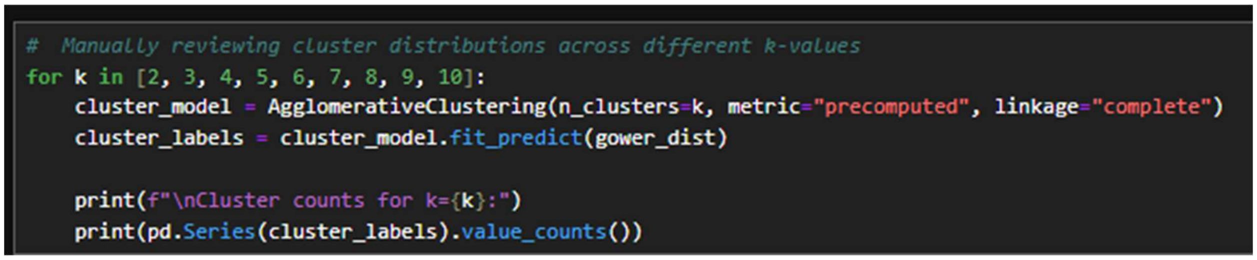


As a result, I opted to use the silhouette score to determine the optimal number of clusters, which suggested k=2 as the best choice.

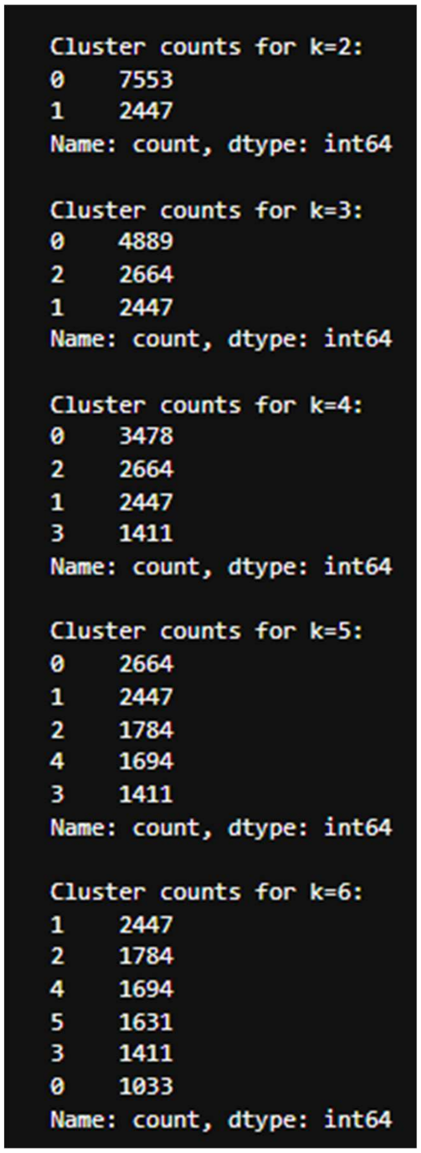


```
# Utilizing Silhouette Score to identify optimal cluster amount
best_clusters = 2
best_score = -1
for k in range(2, 11):
    cluster_model = AgglomerativeClustering(n_clusters=k, metric="precomputed", linkage="complete")
    cluster_labels = cluster_model.fit_predict(gower_dist)

    score = silhouette_score(gower_dist, cluster_labels, metric="precomputed")

    if score > best_score:
        best_score = score
        best_clusters = k

print(f"Optimal number of clusters: {best_clusters}")
Optimal number of clusters: 2
```

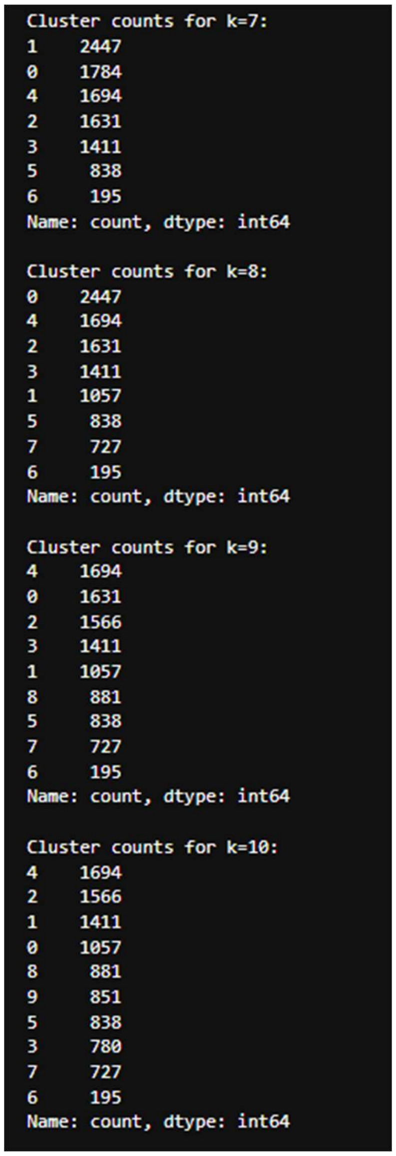
After manually inspecting the cluster distributions at different k (2 to 9) values, I found that k=5 provided a more meaningful segmentation. Per the output below, with k=2, the clusters were too broad, with a 7553 to 2446 split, while k=7 and beyond started producing tiny clusters (838 and 195). Either k=5 or k-6 appeared to be the closest to an even split, and since k=6 had some clusters at or near the three-digit amount, I determined k=5 to be the optimal amount of clusters for this model. Separately, by selecting 5, I also used the information from the Dendrogram, which demonstrated that almost 90% of the dataset was captured when k=5.



```
#  Manually reviewing cluster distributions across different k-values
for k in [2, 3, 4, 5, 6, 7, 8, 9, 10]:
    cluster_model = AgglomerativeClustering(n_clusters=k, metric="precomputed", linkage="complete")
    cluster_labels = cluster_model.fit_predict(gower_dist)

    print(f"\nCluster counts for k={k}:")
    print(pd.Series(cluster_labels).value_counts())
```



```
Cluster counts for k=2:
0    7553
1    2447
Name: count, dtype: int64

Cluster counts for k=3:
0    4889
2    2664
1    2447
Name: count, dtype: int64

Cluster counts for k=4:
0    3478
2    2664
1    2447
3    1411
Name: count, dtype: int64

Cluster counts for k=5:
0    2664
1    2447
2    1784
4    1694
3    1411
Name: count, dtype: int64

Cluster counts for k=6:
1    2447
2    1784
4    1694
5    1631
3    1411
0    1033
Name: count, dtype: int64
```



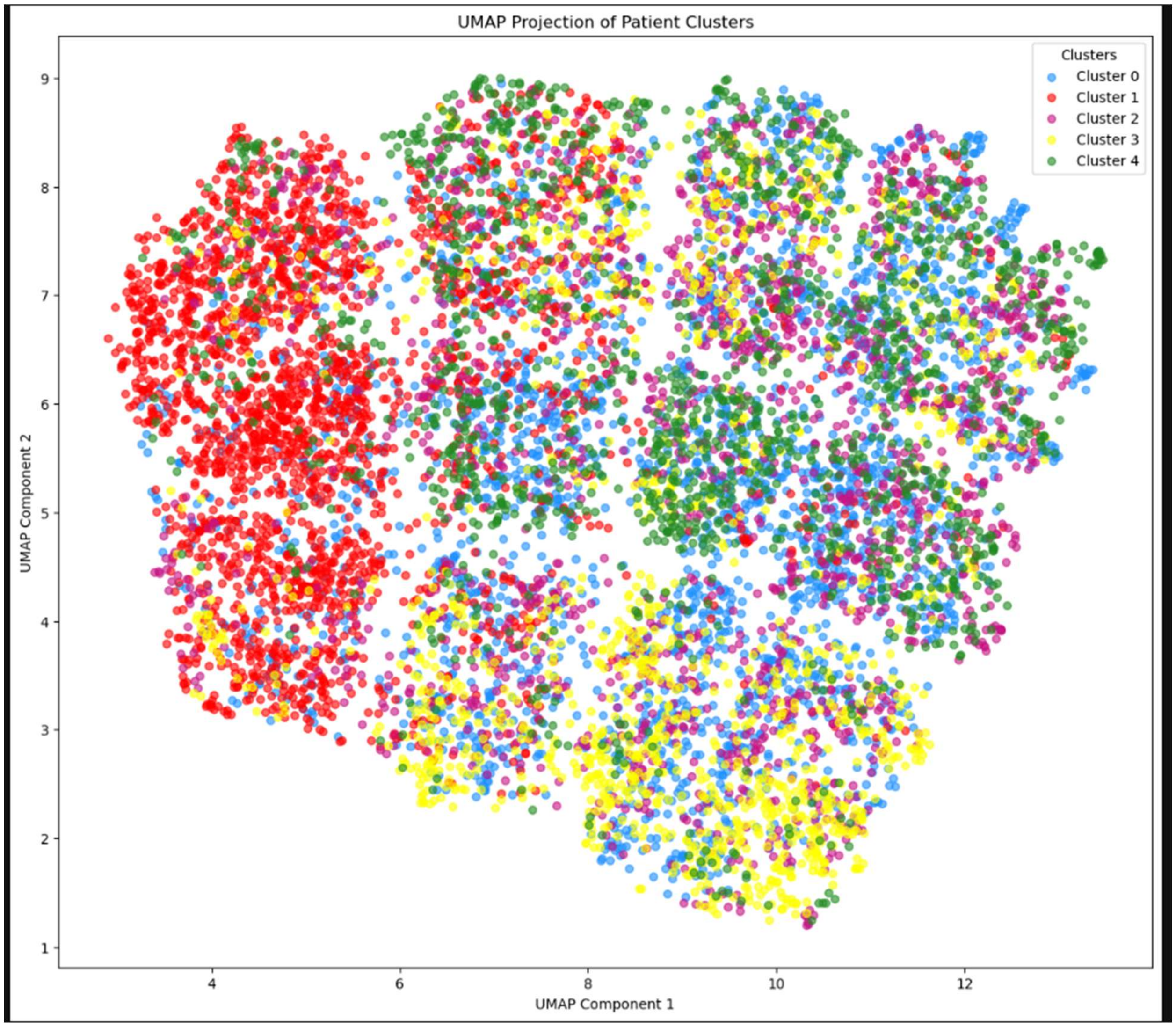
```
Cluster counts for k=7:
1    2447
0    1784
4    1694
2    1631
3    1411
5     838
6     195
Name: count, dtype: int64

Cluster counts for k=8:
0    2447
4    1694
2    1631
3    1411
1    1057
5     838
7     727
6     195
Name: count, dtype: int64

Cluster counts for k=9:
4    1694
0    1631
2    1566
3    1411
1    1057
8     881
5     838
7     727
6     195
Name: count, dtype: int64

Cluster counts for k=10:
4    1694
2    1566
1    1411
0    1057
8     881
9     851
5     838
3     780
7     727
6     195
Name: count, dtype: int64
```

It is also worth noting that there are scenarios where silhouette score can be a poor method for identifying optimal clusters, including when not using Euclidian Distance.

Part 4: Summary and Results

F1. Visualization & Quality of Clusters

After selecting and applying the final cluster value and labels, the next step was to apply dimensionality reduction and plot the clusters in a scatterplot to identify any relevant outcomes in the analysis. Initially, I attempted t-SNE but ran into various issues with loading times, errors, and incorrect outputs. Afterward, I conducted further research and found that regarding computational power and visualizations, UMAP "is generally faster than t-SNE, providing high-quality visualizations with less computational cost. Its algorithmic efficiency makes it suitable for large datasets … and provides clearer and more interpretable visualizations than t-SNE" (Carnot Research Pvt. Ltd., 2024). Overall, the clustering quality in this example was poor, and it isn't easy to glean meaningful insights based on the results—details in the sections below.



F2. Results and Implications

The clustering analysis's overall results were not convincing enough to produce significant details for the research question. Generally, the scatterplot identified patients in cluster one largely being centralized on the left hand of the scatterplot – indicating these patients share similar characteristics that differentiate them from the rest of the dataset. Cluster 3 is primarily visible towards the bottom half of the plot. However, this cluster is not as separated from others, suggesting the defining characteristics may overlap. There are central tendencies for cluster 4 and some drifting towards the top right. Still, this subset has no clear structure, indicating they do not strongly align with defining characteristics. Both clusters 2 and 0 seemed to be spread throughout all portions of the scatterplot, suggesting that those patients may not align closely to any specific features.

The findings imply that further investigation is required and that it is possible that overfitting or arbitrary clustering is occurring. It is also possible that the dataset may not have natural clusters, which indicates

F3. Limitations

Several possible limitations can be highlighted in the clustering analysis. First, the selection of k=5 may have been the root cause of arbitrary clustering in the results, and by selecting fewer clusters, more meaningful analysis may be prevalent. More critically, the features selected for this dataset may not be sufficient to generate distinct, well-separated clusters. Since both categorical and continuous variables were selected, Gower's Distance was necessary, which may have affected the clustering outcome. As a part of the business recommendations below, I would suggest rectifying these items before enacting any significant changes to the business.

F4. Business Recommendations

The business has several different options it can pursue, depending on whether it wants to stick with the research question as it is posed or scrap it and ask a new question. First, if the choice is made to keep the question as is, I recommend adjusting the number of clusters to a lower value to identify if a more meaningful analysis can be identified. While a lower k value may result in more broad categories, it may create clusters that are easier to read and can glean better details about actions. Secondly, a different clustering and dimensional reduction methodology is worth pursuing, as Hierarchical Clustering will have difficulty identifying relevant clusters when the data is so varied and mixed between types. Additionally, UMAP can be very effective for Hierarchical Clustering, but in a dataset that needs a precomputed distance matrix such as Gower's Distance for mixed data, this technique can create arbitrary or overlapping clusters. Engaging in PCA early in the process may be more effective to update the dimensionality and then move into the clustering on the back end to identify meaningful patient groupings.

The second choice, and likely the better option, for the business is to scrap the business question as it is posed, and opt to identify actionable insights with different features in the dataset. This would allow the company to select more cohesive and less mixed data and open up additional possibilities for clustering analysis. A different clustering method, such as K-Means Clustering, may produce more effective and actionable results. By folding in more data points early on, it will be easier to identify features that can more directly yield actionable results.

References

I referred to my work from **D603 - Task 1 (Classification)** as a guide, as it shared multiple similarities with this project.

Carnot Research Pvt. Ltd. (2024, July 2). *Understanding dimensionality reduction: PCA vs t-SNE vs UMAP vs FIt-SNE vs LargeVis vs Laplacian eigenmaps*. Medium. https://carnotresearch.medium.com/understanding-dimensionality-reduction-pca-vs-t-sne-vs-umap-vs-fit-sne-vs-largevis-vs-laplacian-13d0be9ef7f4

Muhammad, U. S. (2023, June 9). *Hierarchical clustering for categorical data.* Medium. https://medium.com/@umarsmuhammed/hierarchical-clustering-for-categorical-data-168fe8fc0e2b