

MLS Clustering ML

June 30, 2024

```
[34]: # Imports
import numpy as np
import pandas as pd
from IPython.display import display
```

```
[35]: # Load
file_path = '/Users/nickblackford/Desktop/Python/als_data.csv'
df = pd.read_csv(file_path)
```

```
[36]: # Preview df
df.head()
```

```
[36]:
```

	ID	Age_mean	Albumin_max	Albumin_median	Albumin_min	Albumin_range	\
0	1	65	57.0	40.5	38.0	0.066202	
1	2	48	45.0	41.0	39.0	0.010453	
2	3	38	50.0	47.0	45.0	0.008929	
3	4	63	47.0	44.0	41.0	0.012111	
4	5	63	47.0	45.5	42.0	0.008292	

	ALSFRS_slope	ALSFRS_Total_max	ALSFRS_Total_median	ALSFRS_Total_min	...	\
0	-0.965608	30	28.0	22	...	
1	-0.921717	37	33.0	21	...	
2	-0.914787	24	14.0	10	...	
3	-0.598361	30	29.0	24	...	
4	-0.444039	32	27.5	20	...	

	Sodium_min	Sodium_range	SubjectID	trunk_max	trunk_median	trunk_min	\
0	143.0	0.017422	533	8	7.0	7	
1	136.0	0.010453	649	8	7.0	5	
2	140.0	0.008929	1234	5	0.0	0	
3	138.0	0.012469	2492	5	5.0	3	
4	138.0	0.008292	2956	6	4.0	1	

	trunk_range	Urine.Ph_max	Urine.Ph_median	Urine.Ph_min
0	0.002646	6.0	6.0	6.0
1	0.005386	7.0	5.0	5.0
2	0.008929	6.0	5.0	5.0

3	0.004988	7.0	6.0	5.0
4	0.008489	6.0	5.0	5.0

[5 rows x 101 columns]

```
[37]: # Temporarily display all columns
pd.set_option('display.max_columns', None)

# Display the DataFrame
print(df)

# Reset to default (optional)
pd.reset_option('display.max_columns')
```

	ID	Age_mean	Albumin_max	Albumin_median	Albumin_min	Albumin_range	\
0	1	65	57.0	40.5	38.0	0.066202	
1	2	48	45.0	41.0	39.0	0.010453	
2	3	38	50.0	47.0	45.0	0.008929	
3	4	63	47.0	44.0	41.0	0.012111	
4	5	63	47.0	45.5	42.0	0.008292	
...	
2218	2419	33	50.0	49.0	45.0	0.008772	
2219	2420	61	47.0	45.0	42.0	0.009074	
2220	2421	47	46.0	44.0	41.0	0.012111	
2221	2422	37	49.0	44.0	39.0	0.017857	
2222	2424	48	48.0	45.0	40.0	0.018476	

	ALSFRS_slope	ALSFRS_Total_max	ALSFRS_Total_median	ALSFRS_Total_min	\
0	-0.965608	30	28.0	22	
1	-0.921717	37	33.0	21	
2	-0.914787	24	14.0	10	
3	-0.598361	30	29.0	24	
4	-0.444039	32	27.5	20	
...	
2218	-0.239501	35	32.5	30	
2219	-0.388711	31	26.0	17	
2220	-0.108631	26	23.0	20	
2221	-0.855880	34	29.5	21	
2222	-2.050562	37	34.0	11	

	ALSFRS_Total_range	ALT.SGPT._max	ALT.SGPT._median	ALT.SGPT._min	\
0	0.021164	24.0	22.0	18.0	
1	0.028725	25.0	13.0	8.0	
2	0.025000	25.0	20.0	14.0	
3	0.014963	62.0	60.0	41.0	
4	0.020374	38.0	26.5	22.0	
...	

2218	0.009107	46.0	27.0	18.0
2219	0.025408	23.0	18.0	15.0
2220	0.010949	129.0	76.5	62.0
2221	0.023214	95.0	51.0	42.0
2222	0.059908	37.0	32.0	13.0

	ALT.SGPT._range	AST.SGOT._max	AST.SGOT._median	AST.SGOT._min	\
0	0.020906	31	27.5	23.0	
1	0.029617	31	17.0	14.0	
2	0.019643	24	19.0	18.0	
3	0.052369	46	40.0	33.0	
4	0.026534	35	26.5	20.0	
...	
2218	0.049123	38	27.0	23.0	
2219	0.014519	27	22.0	18.0	
2220	0.047619	62	46.0	41.0	
2221	0.094643	63	44.0	36.0	
2222	0.055427	38	28.0	11.0	

	AST.SGOT._range	Bicarbonate_max	Bicarbonate_median	Bicarbonate_min	\
0	0.027875	30.0	28.0	25.0	
1	0.029617	32.0	28.0	25.0	
2	0.010714	35.0	29.0	24.0	
3	0.032419	23.0	20.0	20.0	
4	0.024876	32.0	28.0	23.0	
...	
2218	0.026316	31.0	28.0	23.0	
2219	0.016334	31.0	27.8	24.0	
2220	0.035016	31.0	28.0	21.0	
2221	0.048214	32.0	29.0	21.0	
2222	0.062356	31.0	27.0	22.0	

	Bicarbonate_range	Blood.Urea.Nitrogen..BUN._max	\
0	0.017422	8.0322	
1	0.012195	8.3973	
2	0.019643	5.4765	
3	0.007481	8.0322	
4	0.014925	5.1114	
...	
2218	0.014035	5.4765	
2219	0.012704	6.0700	
2220	0.014925	7.8500	
2221	0.019643	6.2067	
2222	0.020785	7.5000	

	Blood.Urea.Nitrogen..BUN._median	Blood.Urea.Nitrogen..BUN._min	\
0	7.11945	6.57180	
1	4.74630	4.01610	

2	4.38120	3.65100
3	8.03220	6.57180
4	4.19865	3.65100
...
2218	3.65100	2.92080
2219	5.00000	3.57000
2220	6.43000	4.53000
2221	4.01610	3.28590
2222	5.36000	1.27785

	Blood.Urea.Nitrogen..BUN._range	bp_diastolic_max	bp_diastolic_median	\
0	0.005089	90	83.0	
1	0.007633	80	78.0	
2	0.003260	86	76.0	
3	0.003642	90	80.0	
4	0.002422	100	80.0	
...	
2218	0.004484	85	78.0	
2219	0.004537	95	90.0	
2220	0.005817	102	86.0	
2221	0.005216	90	77.0	
2222	0.014370	90	80.0	

	bp_diastolic_min	bp_diastolic_range	bp_systolic_max	\
0	69	0.055556	160	
1	64	0.028725	140	
2	58	0.050000	120	
3	70	0.049875	150	
4	68	0.053068	160	
...	
2218	70	0.027322	150	
2219	80	0.027223	155	
2220	76	0.045694	140	
2221	70	0.035714	150	
2222	70	0.046083	150	

	bp_systolic_median	bp_systolic_min	bp_systolic_range	Calcium_max	\
0	139.0	129	0.082011	2.49500	
1	132.5	104	0.064632	2.32035	
2	110.0	90	0.053571	2.47005	
3	130.0	120	0.074813	2.47005	
4	130.0	104	0.092869	2.42015	
...	
2218	115.0	100	0.091075	2.39520	
2219	140.0	130	0.045372	2.50000	
2220	120.0	102	0.066784	2.58000	
2221	122.0	100	0.089286	2.47005	
2222	130.0	110	0.092166	2.65000	

	Calcium_median	Calcium_min	Calcium_range	Chloride_max	\
0	2.220550	2.22055	0.000956	109.0	
1	2.170650	2.02095	0.000522	108.0	
2	2.295400	2.19560	0.000490	108.0	
3	2.345300	2.23000	0.000474	109.0	
4	2.257975	2.17065	0.000414	107.0	
...	
2218	2.320350	2.17065	0.000394	111.0	
2219	2.300000	2.13000	0.000672	111.0	
2220	2.340000	2.30000	0.000474	111.0	
2221	2.320350	2.22055	0.000446	105.0	
2222	2.430000	2.33000	0.000739	108.0	

	Chloride_median	Chloride_min	Chloride_range	Creatinine_max	\
0	108.0	103.0	0.020906	79.560	
1	102.0	100.0	0.013937	61.880	
2	106.0	104.0	0.007143	88.400	
3	107.0	106.0	0.007481	70.720	
4	104.0	100.0	0.011609	61.880	
...	
2218	104.0	102.0	0.015789	88.400	
2219	106.0	101.0	0.018149	72.000	
2220	105.0	99.0	0.015873	82.000	
2221	102.0	98.0	0.012500	61.880	
2222	107.0	99.0	0.020785	93.704	

	Creatinine_median	Creatinine_min	Creatinine_range	Gender_mean	\
0	79.56	70.72	0.030801	1	
1	53.04	44.20	0.030801	1	
2	79.56	70.72	0.031571	2	
3	61.88	53.04	0.044090	2	
4	48.62	26.52	0.058640	1	
...	
2218	70.72	61.88	0.046526	2	
2219	55.00	41.00	0.056261	1	
2220	54.00	45.00	0.048654	2	
2221	44.20	26.52	0.063143	2	
2222	76.00	68.00	0.059363	2	

	Glucose_max	Glucose_median	Glucose_min	Glucose_range	hands_max	\
0	7.4370	4.4955	4.2180	0.011216	8	
1	6.7710	4.9950	4.0515	0.004738	8	
2	5.6610	5.1060	4.2180	0.002577	4	
3	5.1060	4.7730	4.6620	0.001107	6	
4	7.4925	5.7165	5.0505	0.004050	8	
...	
2218	5.4390	4.8285	3.9960	0.002532	6	

2219	11.3000	6.5000	4.8000	0.011797	6
2220	7.4000	5.7000	4.8000	0.004695	5
2221	6.8820	4.8840	4.1070	0.004955	8
2222	6.3000	5.5500	4.9000	0.003333	8

	hands_median	hands_min	hands_range	Hematocrit_max	Hematocrit_median	\
0	7.5	6	0.005291	44.6	43.15	
1	6.0	6	0.003591	41.9	39.60	
2	1.0	0	0.007143	49.1	46.20	
3	5.5	4	0.004988	46.3	43.00	
4	6.5	3	0.008489	44.0	42.85	
...	
2218	4.0	3	0.005464	51.6	48.20	
2219	4.0	1	0.009074	42.0	40.00	
2220	2.0	2	0.005474	46.0	45.00	
2221	6.5	4	0.007143	51.5	48.05	
2222	7.0	0	0.018433	54.0	51.00	

	Hematocrit_min	Hematocrit_range	Hemoglobin_max	Hemoglobin_median	\
0	40.7	0.013589	156.0	146.0	
1	37.7	0.007317	138.0	132.0	
2	44.0	0.009107	161.0	154.0	
3	41.7	0.011471	154.0	145.0	
4	39.5	0.007463	152.0	146.5	
...	
2218	45.6	0.010526	172.0	161.0	
2219	38.0	0.007260	137.0	132.0	
2220	43.0	0.009701	157.0	151.0	
2221	45.2	0.012186	171.0	162.5	
2222	46.6	0.017090	178.0	167.0	

	Hemoglobin_min	Hemoglobin_range	leg_max	leg_median	leg_min	\
0	143.0	0.045296	8	6.5	4	
1	128.0	0.017422	8	7.5	3	
2	151.0	0.017857	4	3.0	2	
3	144.0	0.024938	4	3.5	2	
4	138.0	0.023217	2	2.0	0	
...	
2218	153.0	0.033333	8	8.0	8	
2219	127.0	0.018149	8	8.0	6	
2220	147.0	0.031056	3	2.0	2	
2221	155.0	0.030948	3	3.0	2	
2222	161.0	0.039261	8	8.0	6	

	leg_range	mouth_max	mouth_median	mouth_min	mouth_range	\
0	0.010582	5	3.5	0	0.013228	
1	0.008977	9	8.0	4	0.008977	
2	0.003571	10	7.0	4	0.010714	

3	0.004988	12	12.0	12	0.000000
4	0.003396	12	12.0	12	0.000000
...
2218	0.000000	12	11.0	10	0.003643
2219	0.003630	8	6.0	4	0.007260
2220	0.001825	10	10.0	9	0.001825
2221	0.001786	12	12.0	10	0.003571
2222	0.004608	10	8.0	3	0.016129

	onset_delta_mean	onset_site_mean	Platelets_max	Platelets_median	\
0	-1023	1	172	169.0	
1	-341	1	286	264.0	
2	-1181	1	233	213.0	
3	-365	2	275	233.0	
4	-1768	2	313	283.5	
...	
2218	-817	2	242	202.0	
2219	-527	1	260	217.0	
2220	-1589	2	246	222.0	
2221	-558	2	271	237.0	
2222	-204	1	357	299.0	

	Platelets_min	Potassium_max	Potassium_median	Potassium_min	\
0	152.0	4.5	4.25	4.0	
1	230.0	5.0	4.30	3.9	
2	167.0	4.1	4.00	3.9	
3	204.0	4.3	4.20	4.0	
4	268.0	4.6	3.75	3.5	
...	
2218	176.0	4.4	4.10	3.9	
2219	196.0	4.8	4.25	3.9	
2220	187.0	4.4	3.95	3.7	
2221	187.0	4.8	4.20	3.9	
2222	248.0	5.3	4.60	4.2	

	Potassium_range	pulse_max	pulse_median	pulse_min	pulse_range	\
0	0.001742	79	68.0	61	0.047619	
1	0.001916	90	76.0	64	0.046679	
2	0.000357	82	73.0	60	0.039286	
3	0.000748	84	72.0	68	0.039900	
4	0.001824	101	96.0	74	0.044776	
...	
2218	0.000877	80	67.5	56	0.043716	
2219	0.001633	86	78.0	72	0.025408	
2220	0.001425	102	86.0	62	0.070299	
2221	0.001607	104	90.0	85	0.033929	
2222	0.002949	98	84.0	64	0.078341	

	respiratory_max	respiratory_median	respiratory_min	respiratory_range	\
0	4	3.0	3	0.002646	
1	4	4.0	3	0.001795	
2	4	4.0	4	0.000000	
3	3	3.0	3	0.000000	
4	4	4.0	3	0.001698	
...	
2218	4	4.0	4	0.000000	
2219	4	4.0	3	0.001815	
2220	4	4.0	3	0.001825	
2221	4	4.0	2	0.003571	
2222	4	4.0	1	0.006912	

	Sodium_max	Sodium_median	Sodium_min	Sodium_range	SubjectID	\
0	148.0	145.5	143.0	0.017422	533	
1	142.0	138.0	136.0	0.010453	649	
2	145.0	143.0	140.0	0.008929	1234	
3	143.0	139.0	138.0	0.012469	2492	
4	143.0	140.0	138.0	0.008292	2956	
...	
2218	144.0	141.0	136.0	0.014035	997136	
2219	146.0	143.0	141.0	0.009074	998047	
2220	144.0	141.0	135.0	0.013123	998773	
2221	140.0	139.0	136.0	0.007143	998908	
2222	145.0	141.0	137.0	0.018476	999482	

	trunk_max	trunk_median	trunk_min	trunk_range	Urine.Ph_max	\
0	8	7.0	7	0.002646	6.00	
1	8	7.0	5	0.005386	7.00	
2	5	0.0	0	0.008929	6.00	
3	5	5.0	3	0.004988	7.00	
4	6	4.0	1	0.008489	6.00	
...	
2218	7	5.0	5	0.003643	7.00	
2219	5	4.0	3	0.003630	7.41	
2220	5	4.0	4	0.001825	9.00	
2221	8	4.5	2	0.010714	6.00	
2222	8	8.0	1	0.016129	5.00	

	Urine.Ph_median	Urine.Ph_min
0	6.0	6.0
1	5.0	5.0
2	5.0	5.0
3	6.0	5.0
4	5.0	5.0
...
2218	6.0	5.0
2219	5.5	5.0

2220	6.0	5.0
2221	5.0	5.0
2222	5.0	5.0

[2223 rows x 101 columns]

```
[38]: # Remove any data that is not relevant to the patient's ALS condition
df = df.drop(columns=['ID'])
```

```
[39]: # Apply standard scaler
from sklearn.preprocessing import StandardScaler

# Initialize scaler
scaler = StandardScaler()

# Fit the scaler on the data and transform it
df_scaled = scaler.fit_transform(df)

# Convert the scaled data back to a DataFrame
X_scaled = pd.DataFrame(df_scaled, columns=df.columns)
```

```
[40]: import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

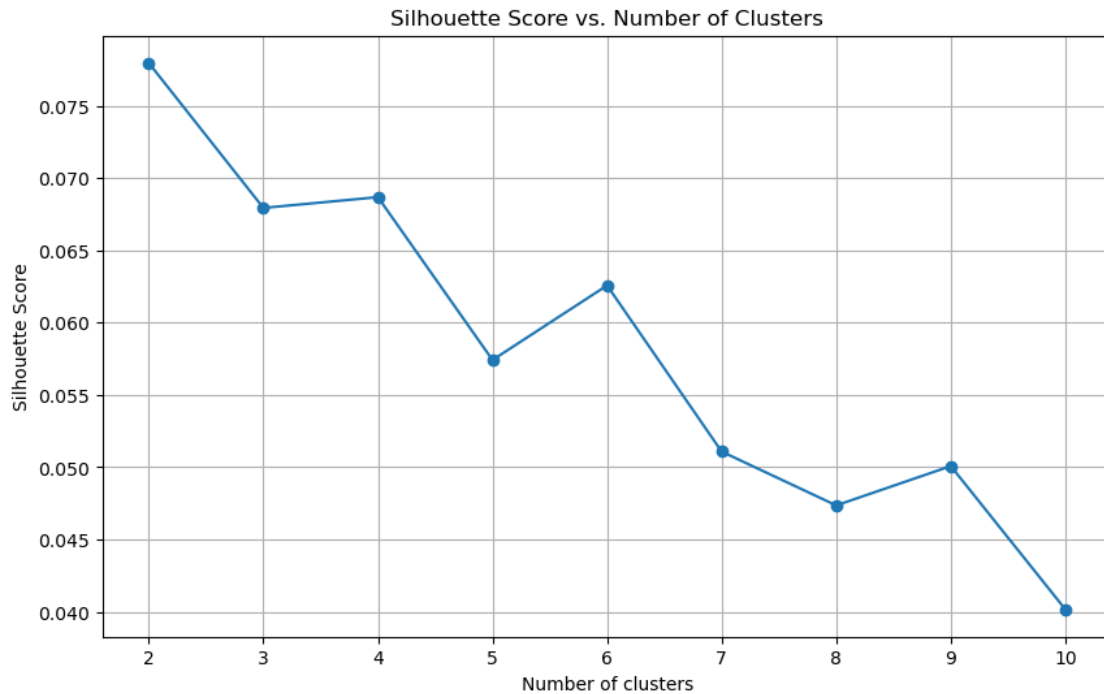
# Calculate silhouette scores for different numbers of clusters
silhouette_scores = []
cluster_range = range(2, 11)

for k in cluster_range:
    kmeans = KMeans(n_clusters=k, n_init=10, random_state=10)
    y_kmeans = kmeans.fit_predict(X_scaled)
    score = silhouette_score(X_scaled, y_kmeans)
    silhouette_scores.append(score)

# Print silhouette scores for debugging
for k, score in zip(cluster_range, silhouette_scores):
    print(f'Number of clusters: {k}, Silhouette Score: {score}')
```

```
Number of clusters: 2, Silhouette Score: 0.0779489418382482
Number of clusters: 3, Silhouette Score: 0.06792675217846486
Number of clusters: 4, Silhouette Score: 0.06867177357095701
Number of clusters: 5, Silhouette Score: 0.05740976392457452
Number of clusters: 6, Silhouette Score: 0.06256617707910656
Number of clusters: 7, Silhouette Score: 0.051075921419424526
Number of clusters: 8, Silhouette Score: 0.04735394077311848
Number of clusters: 9, Silhouette Score: 0.050075176240459866
Number of clusters: 10, Silhouette Score: 0.04016513650091586
```

```
[41]: # Plot silhouette scores vs. number of clusters
plt.figure(figsize=(10, 6))
plt.plot(cluster_range, silhouette_scores, marker='o')
plt.title('Silhouette Score vs. Number of Clusters')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.xticks(cluster_range)
plt.grid(True)
plt.show()
```



Analyzing the Silhouette Score vs. Number of clusters plot, we are going to select 4 as an optimal number of clusters. 4 clusters has the second highest silhouette score only to 2. Given the data we are working with, selecting 4 clusters as opposed to 2 will likely give us more insights when conducting analysis on ALS patients.

```
[48]: # Create and fit the K-means model
kmeans = KMeans(n_clusters=4, n_init=10, random_state=10)
kmeans.fit(X_scaled)

# Predict the clusters for the data points
df['Cluster'] = kmeans.labels_
```

```
[49]: from sklearn.decomposition import PCA

# Fit PCA with 2 components
```

```

pca = PCA(n_components=2)
X_pca = pca.fit_transform(df)

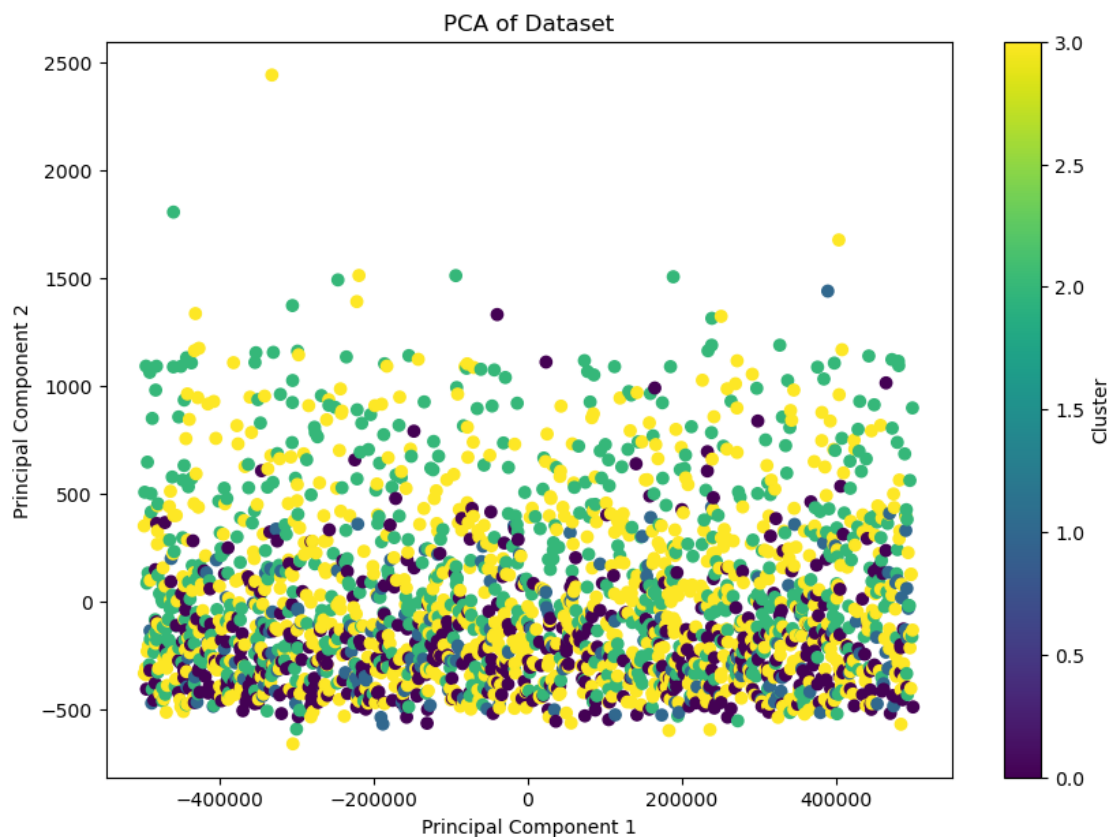
# Create a df with the PCA results
df_pca = pd.DataFrame(data=X_pca, columns=['Principal Component 1', 'Principal_
↪Component 2'])
df_pca['Cluster'] = df['Cluster']

```

```

[50]: # Make a scatterplot of the PCA transformed data
plt.figure(figsize=(10, 7))
scatter = plt.scatter(df_pca['Principal Component 1'], df_pca['Principal_
↪Component 2'], c=df_pca['Cluster'], cmap='viridis')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Dataset')
plt.colorbar(scatter, label='Cluster')
plt.show()

```



0.1 Conclusion

0.1.1 Objective

The goal of this analysis was to cluster ALS (Amyotrophic Lateral Sclerosis) patient health metrics using K-means clustering and evaluate the clustering performance to gain insights into potential patterns or subgroups within the patient population.

0.1.2 Steps Taken

1. Data Preparation:

- Loaded the dataset containing 100 different health metrics for ALS patients.
- Scaled the data to standardize the features, ensuring each feature contributes equally to the clustering process.

2. Optimal Number of Clusters:

- Utilized the Silhouette Score to determine the optimal number of clusters.
- The Silhouette Score indicated that 4 clusters provided the highest silhouette score second to 2 clusters, suggesting a well-defined clustering structure.

3. K-means Clustering:

- Fitted the K-means model with 4 clusters.
- Assigned cluster labels to each data point (patient).

4. PCA Transformation and Visualization:

- Applied PCA to reduce the dimensionality of the data to two principal components.
- Created a scatter plot to visualize the clustering results, coloring each point by its cluster label.

0.1.3 Conclusion

Based on the clustering analysis and performance evaluation, we can draw the following conclusions:

1. Distinct Subgroups:

- The analysis identified four distinct subgroups within the ALS patient population based on the 100 health metrics.
- This suggests that there may be underlying patterns or characteristics that differentiate these four groups, which could be related to disease progression, response to treatment, or other health-related factors.

2. Clinical Implications:

- These subgroups could potentially inform clinical decisions, such as tailoring treatment plans to specific patient profiles or identifying patients who may benefit from more intensive monitoring.
- Further analysis is needed to understand the specific health metrics that contribute most significantly to the clustering, which could highlight critical factors in ALS management.

3. Future Research:

- This preliminary clustering analysis provides a foundation for more detailed studies. Future research could involve investigating the specific characteristics of each cluster, exploring correlations with clinical outcomes, and validating the findings with larger datasets.

- Additionally, integrating other data sources (e.g., genetic data, lifestyle factors) could provide a more comprehensive understanding of the patient subgroups.

4. **Limitations:**

- The current analysis is exploratory and should be interpreted with caution. The clustering results depend on the selected features and the scaling method used.

[]: