IE6400 Foundation of Data Analytics

PROJECT 1

Clustering techniques on MIMIC Dataset

BOMBAY PRANATHI

002499390

INTRODUCTION

Healthcare analytics is a powerful tool for enhancing patient care, optimizing clinical operations, and improving decision-making within the medical field. This report focuses on analysing patient data from the MIMIC-III Clinical Database using clustering techniques to uncover patterns and insights that can drive better healthcare outcomes. In today's healthcare landscape, data-driven patient clustering plays a vital role in risk assessment, disease prediction, and personalized treatment planning. This study employs unsupervised machine learning techniques to categorize patients into meaningful groups based on demographics, lab test results, and vital signs.

The key objectives of this clustering approach are: To identify distinct patient subgroups based on physiological and biochemical attributes, To examine variations in lab test results, vital signs, and demographics across different clusters, to evaluate the effectiveness of K-Means and Hierarchical Clustering methods in patient segmentation.

The study leverages demographic data, lab test results, and vital signs to explore patient groupings and characteristics. Data preprocessing steps such as handling missing values and normalizing features ensure the dataset is well-prepared for analysis.

Through a rigorous application of statistical techniques, visualization tools, and domain knowledge, this analysis aims to extract actionable insights from healthcare data. It is imperative to approach this task with an open mind, recognizing that healthcare is a complex, multifaceted domain influenced by numerous factors, including demographics, socioeconomic conditions, policy frameworks, and medical advancements.

This report endeavors to equip stakeholders with a nuanced understanding of healthcare patterns, fostering evidence-based decision-making and proactive strategies to enhance patient care, optimize resource allocation, and improve overall public health outcomes. By delving into the intricacies of healthcare data, we seek to contribute to the ongoing efforts to create a more efficient, accessible, and high-quality healthcare system.

Two clustering techniques are used:

K-Means Clustering

- A partition-based algorithm that assigns each patient to one of k predefined clusters.
- I used the elbow approach to calculate the ideal number of clusters.
- It efficiently groups patients based on numerical similarity in lab test values and vital signs.

2. Hierarchical Clustering

- A tree-based clustering technique that creates a dendrogram, illustrating how patients are grouped at different levels.
- We apply Ward's linkage method to minimize variance within clusters.
- It helps identify nested relationships between patient subgroups.

Due to computational limitations, we apply hierarchical clustering on a random subset of 1000 patients

DATA PREPROCESSING AND ANALYSIS

1. Loading and Inspecting Data

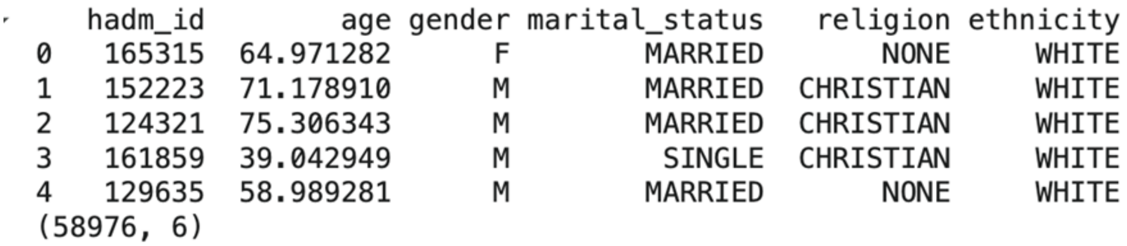
The dataset consists of multiple files containing demographics, lab test results, and vital signs**.** Each dataset has different patient attributes, which needed to be examined before merging. The dataset consists of 58,833 patient records with the following key attributes:

Variables in Each Dataset

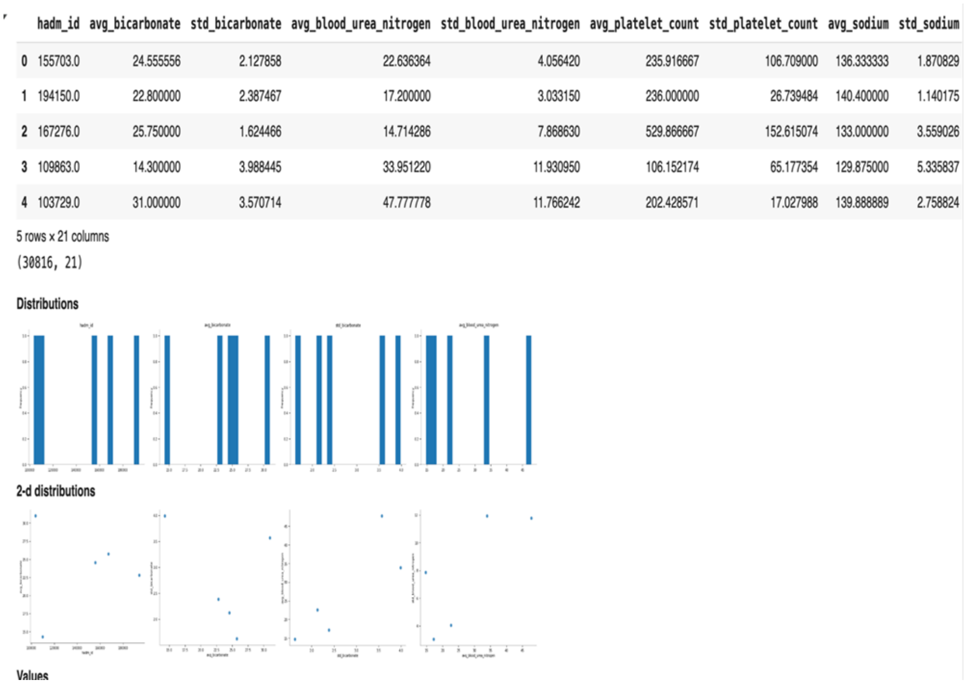
1. **Demographics Dataset (DEMO_DATA.csv)**:
   - hadm_id: Unique hospital admission ID (used for merging).
   - age: Patient age.
   - gender: Patient gender
   - marital status
   - religion
   - ethnicity
2. **Lab Test Datasets**:
   - **White Blood Cells (WHITE_BLOOD_CELLS.csv)**:
     - hadm_id,
     - avg_white_blood_cells
     - std_white_blood_cells (mean and standard deviation of WBC count).
   - **Platelet Count (PLATELET_COUNT.csv)**:
     - hadm_id,
     - avg_platelet_count
     - std_platelet_count.
   - **Blood Glucose (BLOOD_GLUCOSE.csv)**:
     - hadm_id,
     - avg_blood_glucose
     - std_blood_glucose.
3. **Creatinine (CREATININE.csv)**:
   - hadm_id
   - avg_creatinine – Mean creatinine level (kidney function marker)
   - std_creatinine – Standard deviation of creatinine levels
4. **Blood Urea Nitrogen (BLOOD_UREA_NITROGEN.csv)**:
   - hadm_id
   - avg_blood_urea_nitrogen – Mean BUN level (kidney function and nitrogen waste levels)
   - std_blood_urea_nitrogen – Standard deviation of BUN levels
5. **Hematocrit (HEMATROCRIT.csv)**:
   - hadm_id

- avg_hematrocrit – Mean hematocrit level (oxygen transport in blood)
- std_hematrocrit – Standard deviation of hematocrit levels

6. **Potassium (POTASSIUM.csv)**:
   - hadm_id
   - avg_potassium – Mean potassium level (electrolyte balance for heart and muscle function)
   - std_potassium – Standard deviation of potassium levels
7. **Vital Sign Datasets**:
   - **Temperature (TEMP.csv)**:
     - hadm_id,
     - avg_temp,
     - std_temp.
   - **Respiratory Rate (RESP_RATE.csv)**:
     - hadm_id,
     - avg_resp_rate,
     - std_resp_rate.
   - **Systolic Blood Pressure (SYS_PRESS.csv)**:
     - hadm_id,
     - avg_sys_press,
     - std_sys_press.



```
   hadm_id        age gender marital_status   religion ethnicity
0   165315  64.971282      F        MARRIED       NONE     WHITE
1   152223  71.178910      M        MARRIED  CHRISTIAN     WHITE
2   124321  75.306343      M        MARRIED  CHRISTIAN     WHITE
3   161859  39.042949      M         SINGLE  CHRISTIAN     WHITE
4   129635  58.989281      M        MARRIED       NONE     WHITE
(58976, 6)
```



	hadm_id	avg_bicarbonate	std_bicarbonate	avg_blood_urea_nitrogen	std_blood_urea_nitrogen	avg_platelet_count	std_platelet_count	avg_sodium	std_sodium
0	155703.0	24.555556	2.127858	22.636364	4.056420	235.916667	106.709000	136.333333	1.870829
1	194150.0	22.800000	2.387467	17.200000	3.033150	236.000000	26.739484	140.400000	1.140175
2	167276.0	25.750000	1.624466	14.714286	7.868630	529.866667	152.615074	133.000000	3.559026
3	109863.0	14.300000	3.988445	33.951220	11.930950	106.152174	65.177354	129.875000	5.335837
4	103729.0	31.000000	3.570714	47.777778	11.766242	202.428571	17.027988	139.888889	2.758824

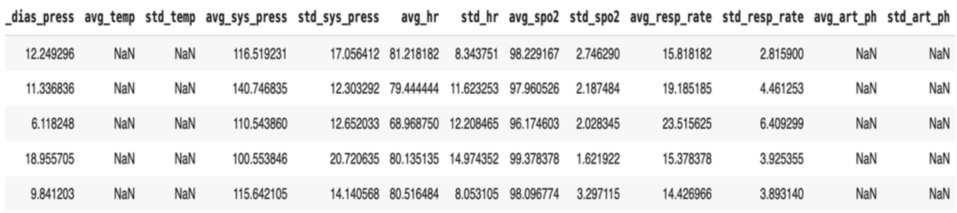
5 rows x 21 columns

```
(30816, 21)
```

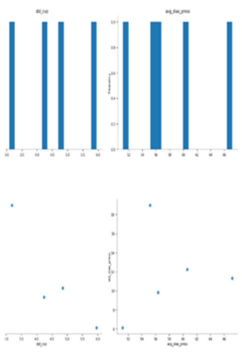
Distributions

2-d distributions

Values



_dias_press	avg_temp	std_temp	avg_sys_press	std_sys_press	avg_hr	std_hr	avg_spo2	std_spo2	avg_resp_rate	std_resp_rate	avg_art_ph	std_art_ph
12.249296	NaN	NaN	116.519231	17.056412	81.218182	8.343751	98.229167	2.746290	15.818182	2.815900	NaN	NaN
11.336836	NaN	NaN	140.746835	12.303292	79.444444	11.623253	97.960526	2.187484	19.185185	4.461253	NaN	NaN
6.118248	NaN	NaN	110.543860	12.652033	68.968750	12.208465	96.174603	2.028345	23.515625	6.409299	NaN	NaN
18.955705	NaN	NaN	100.553846	20.720635	80.135135	14.974352	99.378378	1.621922	15.378378	3.925355	NaN	NaN
9.841203	NaN	NaN	115.642105	14.140568	80.516484	8.053105	98.096774	3.297115	14.426966	3.893140	NaN	NaN



2: Ensuring a Common Identifier (**hadm_id**) for Merging

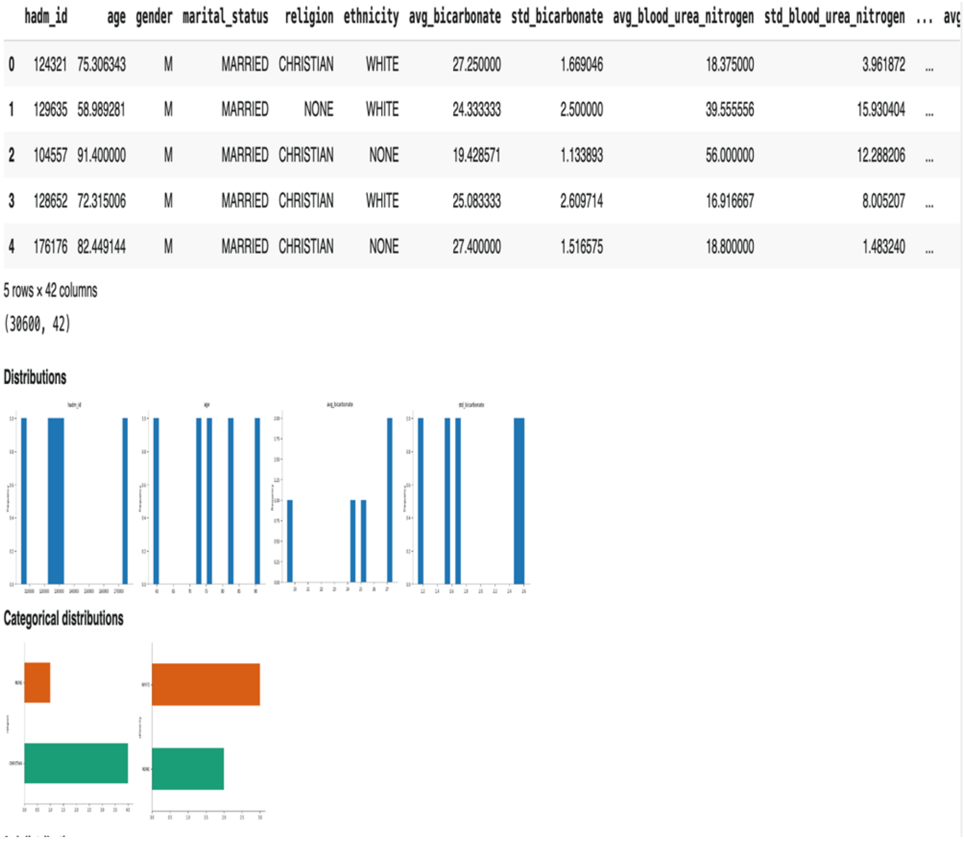
The hadm_id field (hospital admission ID) is used as the primary key. However, not all datasets may contain hadm_id. To resolve this issue:

1. Each dataset was checked for the presence of hadm_id.

2. If hadm_id is missing, a synthetic identifier was created to prevent merging issues.

3 : Merging Datasets

Once `hadm_id` was confirmed in all datasets, they are merged using an inner join. This ensures that only patients with complete records in all datasets are included. The merging process combined all features from demographics(demo data), lab tests(white blood cells, platelet count, blood glucose, sodium, potassium, albumin, hematrocrit, creatinine), and vital signs(temp, sys press, resp rate, cvp, spo2, hr, art ph)into a single dataset



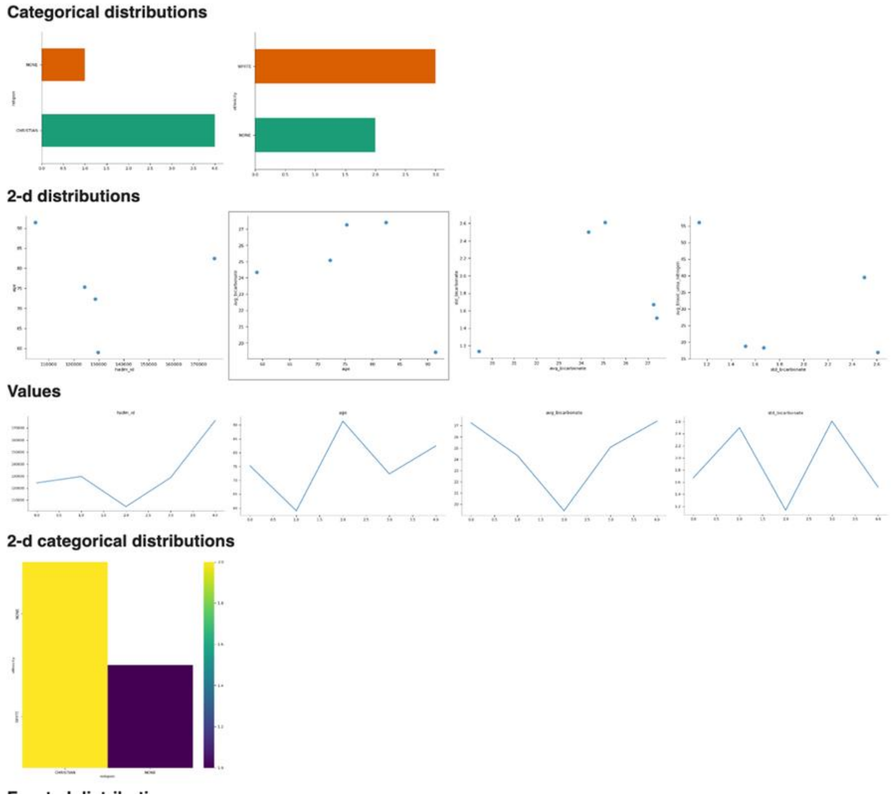
	hadm_id	age	gender	marital_status	religion	ethnicity	avg_bicarbonate	std_bicarbonate	avg_blood_urea_nitrogen	std_blood_urea_nitrogen	...	avg
0	124321	75.306343	M	MARRIED	CHRISTIAN	WHITE	27.250000	1.669046	18.375000	3.961872	...	
1	129635	58.989281	M	MARRIED	NONE	WHITE	24.333333	2.500000	39.555556	15.930404	...	
2	104557	91.400000	M	MARRIED	CHRISTIAN	NONE	19.428571	1.133893	56.000000	12.288206	...	
3	128652	72.315006	M	MARRIED	CHRISTIAN	WHITE	25.083333	2.609714	16.916667	8.005207	...	
4	176176	82.449144	M	MARRIED	CHRISTIAN	NONE	27.400000	1.516575	18.800000	1.483240	...	

5 rows x 42 columns

```
(30600, 42)
```

Distributions

Categorical distributions

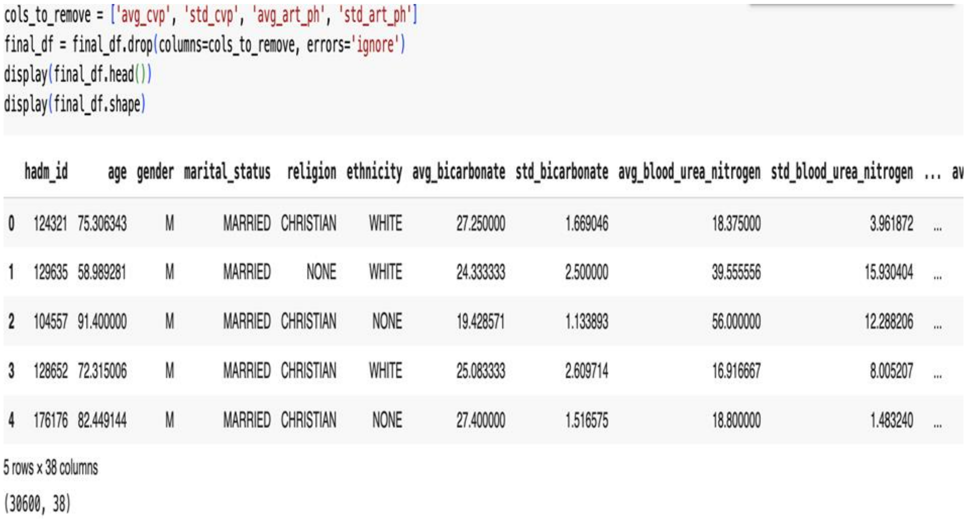


4: Handling Missing Values

Before proceeding with clustering, missing values are addressed. The number of missing values was calculated for each column, and rows containing missing values were removed to ensure consistency. Since clustering techniques require complete records, any row with missing data was dropped and columns with more missing data are removed.

Elimination of columns:

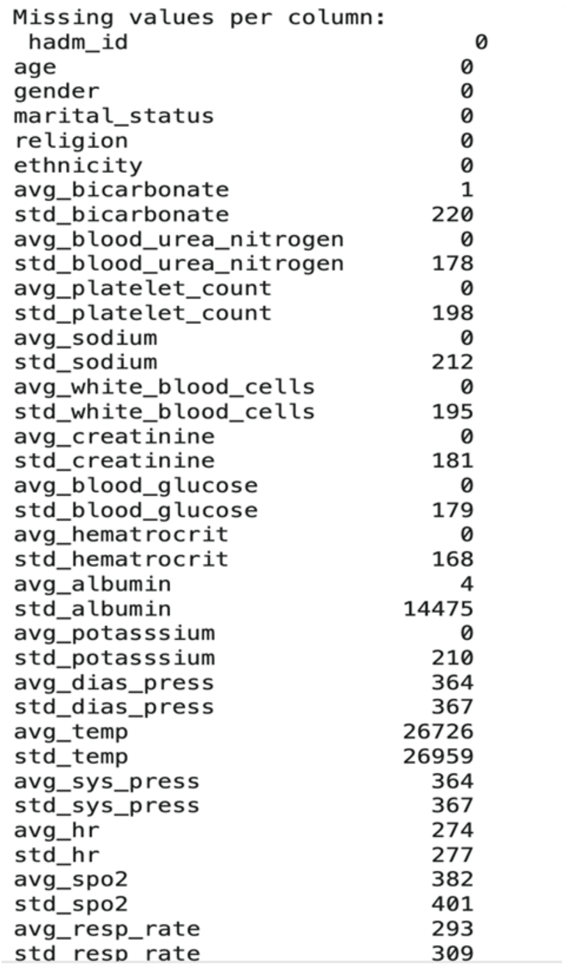
The preprocessing step ensures that only the most relevant features are retained for clustering analysis. By eliminating columns related to CVP and Arterial pH, the dataset is refined to focus on critical lab test results and vital signs that contribute significantly to meaningful patient segmentation. This step helps to reduce noise in the clustering process, improving accuracy and interpretability. Once the unnecessary columns are dropped, the modified dataset is displayed, along with its shape, to verify the applied changes and ensure a more effective clustering approach.



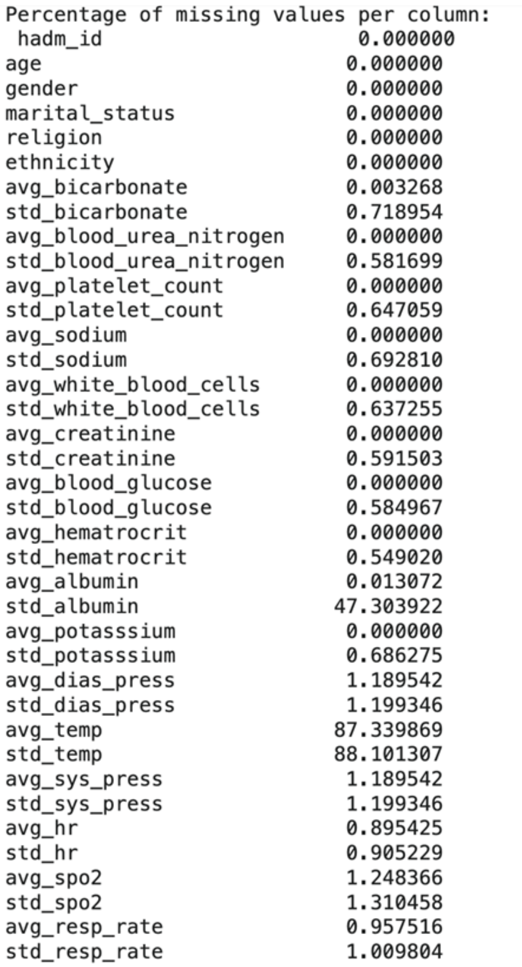
```
cols_to_remove = ['avg_cvp', 'std_cvp', 'avg_art_ph', 'std_art_ph']
final_df = final_df.drop(columns=cols_to_remove, errors='ignore')
display(final_df.head())
display(final_df.shape)
```

	hadm_id	age	gender	marital_status	religion	ethnicity	avg_bicarbonate	std_bicarbonate	avg_blood_urea_nitrogen	std_blood_urea_nitrogen	...	av
0	124321	75.306343	M	MARRIED	CHRISTIAN	WHITE	27.250000	1.669046	18.375000	3.961872	...	
1	129635	58.989281	M	MARRIED	NONE	WHITE	24.333333	2.500000	39.555556	15.930404	...	
2	104557	91.400000	M	MARRIED	CHRISTIAN	NONE	19.428571	1.133893	56.000000	12.288206	...	
3	128652	72.315006	M	MARRIED	CHRISTIAN	WHITE	25.083333	2.609714	16.916667	8.005207	...	
4	176176	82.449144	M	MARRIED	CHRISTIAN	NONE	27.400000	1.516575	18.800000	1.483240	...	

```
5 rows x 38 columns
(30600, 38)
```

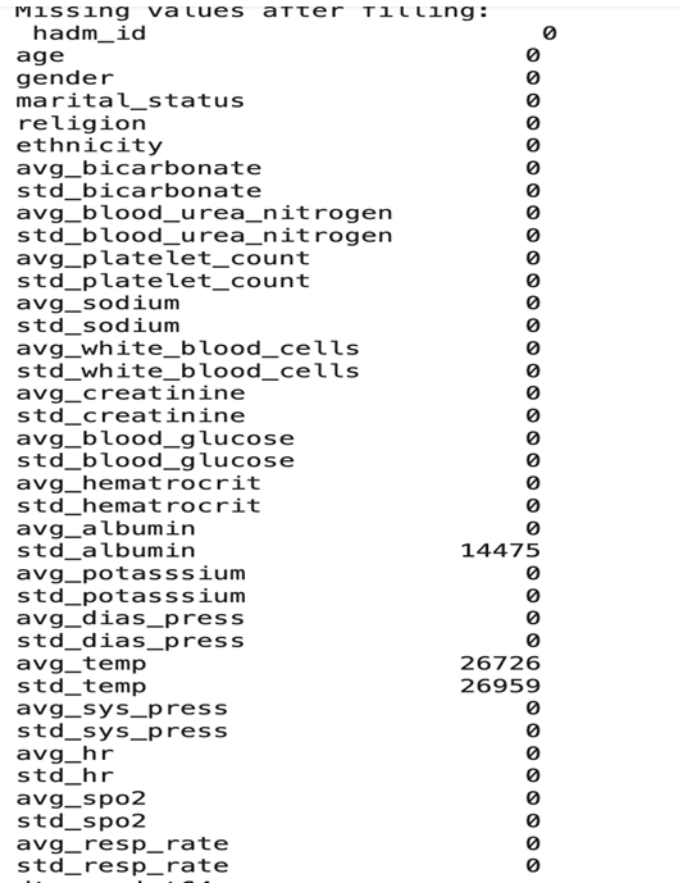


```
Missing values per column:
 hadm_id                     0
age                         0
gender                      0
marital_status              0
religion                    0
ethnicity                   0
avg_bicarbonate             1
std_bicarbonate           220
avg_blood_urea_nitrogen     0
std_blood_urea_nitrogen   178
avg_platelet_count          0
std_platelet_count        198
avg_sodium                  0
std_sodium                212
avg_white_blood_cells       0
std_white_blood_cells     195
avg_creatinine              0
std_creatinine            181
avg_blood_glucose           0
std_blood_glucose         179
avg_hematrocrit             0
std_hematrocrit           168
avg_albumin                 4
std_albumin             14475
avg_potasssium              0
std_potasssium            210
avg_dias_press            364
std_dias_press            367
avg_temp                26726
std_temp                26959
avg_sys_press             364
std_sys_press             367
avg_hr                    274
std_hr                    277
avg_spo2                  382
std_spo2                  401
avg_resp_rate             293
std_resp_rate             309
```



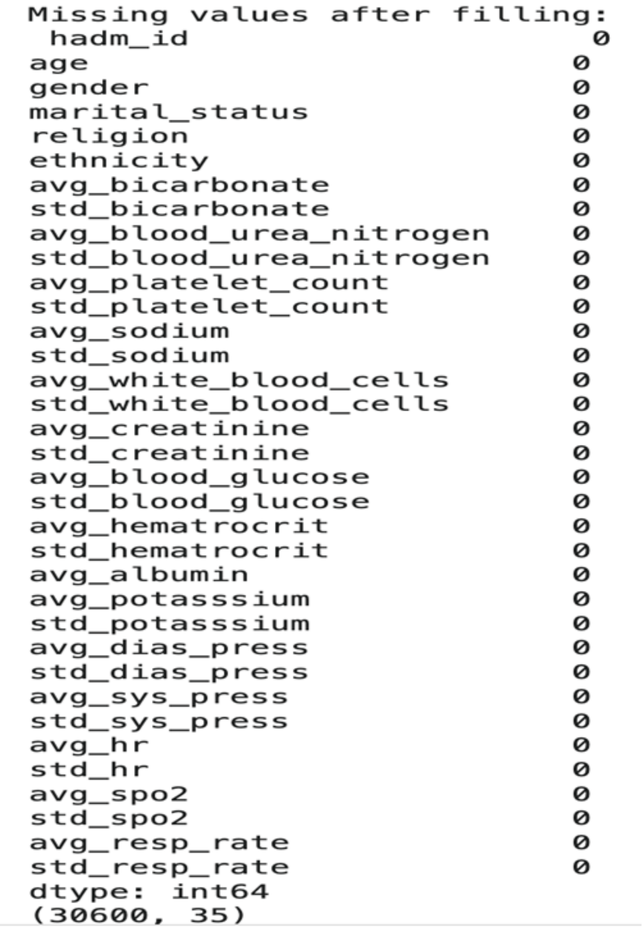
```
Percentage of missing values per column:
 hadm_id                     0.000000
age                         0.000000
gender                      0.000000
marital_status              0.000000
religion                    0.000000
ethnicity                   0.000000
avg_bicarbonate             0.003268
std_bicarbonate             0.718954
avg_blood_urea_nitrogen     0.000000
std_blood_urea_nitrogen     0.581699
avg_platelet_count          0.000000
std_platelet_count          0.647059
avg_sodium                  0.000000
std_sodium                  0.692810
avg_white_blood_cells       0.000000
std_white_blood_cells       0.637255
avg_creatinine              0.000000
std_creatinine              0.591503
avg_blood_glucose           0.000000
std_blood_glucose           0.584967
avg_hematrocrit             0.000000
std_hematrocrit             0.549020
avg_albumin                 0.013072
std_albumin                47.303922
avg_potasssium              0.000000
std_potasssium              0.686275
avg_dias_press              1.189542
std_dias_press              1.199346
avg_temp                   87.339869
std_temp                   88.101307
avg_sys_press               1.189542
std_sys_press               1.199346
avg_hr                      0.895425
std_hr                      0.905229
avg_spo2                    1.248366
std_spo2                    1.310458
avg_resp_rate               0.957516
std_resp_rate               1.009804
```

- Missing values in these columns are replaced with the mean of their respective columns, ensuring data integrity while preserving the overall distribution. After imputation, the dataset is rechecked to confirm the absence of null values. This step is crucial in enhancing data quality and ensuring the accuracy and reliability of clustering results, as missing values can introduce distortions and negatively impact the performance of machine learning models.



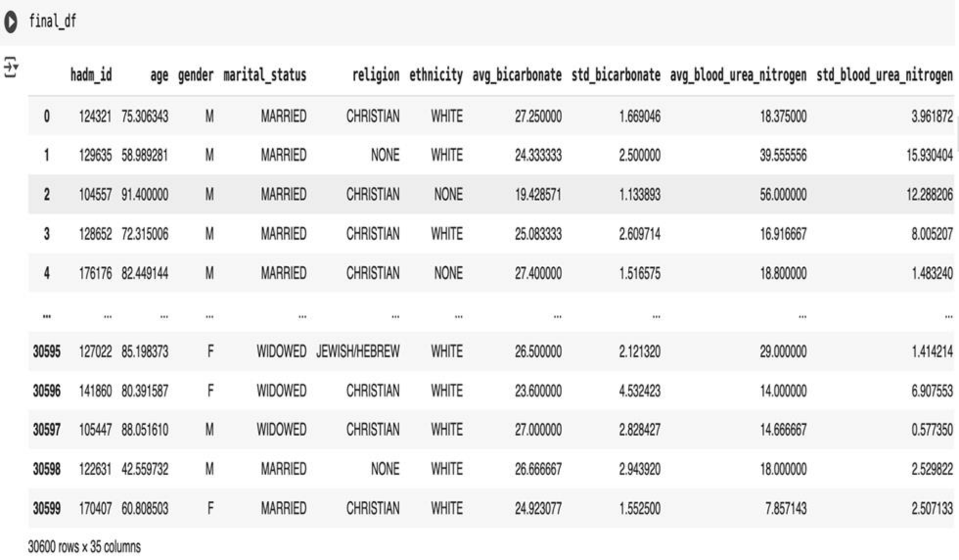
```
Missing values after filling:
 hadm_id                        0
age                            0
gender                         0
marital_status                 0
religion                       0
ethnicity                      0
avg_bicarbonate                0
std_bicarbonate                0
avg_blood_urea_nitrogen        0
std_blood_urea_nitrogen        0
avg_platelet_count             0
std_platelet_count             0
avg_sodium                     0
std_sodium                     0
avg_white_blood_cells          0
std_white_blood_cells          0
avg_creatinine                 0
std_creatinine                 0
avg_blood_glucose              0
std_blood_glucose              0
avg_hematrocrit                0
std_hematrocrit                0
avg_albumin                    0
std_albumin                14475
avg_potasssium                 0
std_potasssium                 0
avg_dias_press                 0
std_dias_press                 0
avg_temp                   26726
std_temp                   26959
avg_sys_press                  0
std_sys_press                  0
avg_hr                         0
std_hr                         0
avg_spo2                       0
std_spo2                       0
avg_resp_rate                  0
std_resp_rate                  0
```

- The columns avg_temp, std_temp, and std_albumin were removed as they were not expected to significantly impact the patient segmentation process. After eliminating these columns, the dataset was rechecked for any remaining missing values, and its updated shape was displayed to confirm the changes. This step ensures that only the most relevant features are retained, optimizing the clustering analysis by reducing unnecessary noise.



```
Missing values after filling:
 hadm_id                    0
age                        0
gender                     0
marital_status             0
religion                   0
ethnicity                  0
avg_bicarbonate            0
std_bicarbonate            0
avg_blood_urea_nitrogen    0
std_blood_urea_nitrogen    0
avg_platelet_count         0
std_platelet_count         0
avg_sodium                 0
std_sodium                 0
avg_white_blood_cells      0
std_white_blood_cells      0
avg_creatinine             0
std_creatinine             0
avg_blood_glucose          0
std_blood_glucose          0
avg_hematrocrit            0
std_hematrocrit            0
avg_albumin                0
avg_potasssium             0
std_potasssium             0
avg_dias_press             0
std_dias_press             0
avg_sys_press              0
std_sys_press              0
avg_hr                     0
std_hr                     0
avg_spo2                   0
std_spo2                   0
avg_resp_rate              0
std_resp_rate              0
dtype: int64
(30600, 35)
```

- The final dataset after removing missing values and filling the values is the following which has 30,600 rows and 35 columns.



```
final_df
```

	hadm_id	age	gender	marital_status	religion	ethnicity	avg_bicarbonate	std_bicarbonate	avg_blood_urea_nitrogen	std_blood_urea_nitrogen
0	124321	75.306343	M	MARRIED	CHRISTIAN	WHITE	27.250000	1.669046	18.375000	3.961872
1	129635	58.989281	M	MARRIED	NONE	WHITE	24.333333	2.500000	39.555556	15.930404
2	104557	91.400000	M	MARRIED	CHRISTIAN	NONE	19.428571	1.133893	56.000000	12.288206
3	128652	72.315006	M	MARRIED	CHRISTIAN	WHITE	25.083333	2.609714	16.916667	8.005207
4	176176	82.449144	M	MARRIED	CHRISTIAN	NONE	27.400000	1.516575	18.800000	1.483240
...	...	...	...	...	...	...	...	...	...	...
30595	127022	85.198373	F	WIDOWED	JEWISH/HEBREW	WHITE	26.500000	2.121320	29.000000	1.414214
30596	141860	80.391587	F	WIDOWED	CHRISTIAN	WHITE	23.600000	4.532423	14.000000	6.907553
30597	105447	88.051610	M	WIDOWED	CHRISTIAN	WHITE	27.000000	2.828427	14.666667	0.577350
30598	122631	42.559732	M	MARRIED	NONE	WHITE	26.666667	2.943920	18.000000	2.529822
30599	170407	60.808503	F	MARRIED	CHRISTIAN	WHITE	24.923077	1.552500	7.857143	2.507133

30600 rows x 35 columns

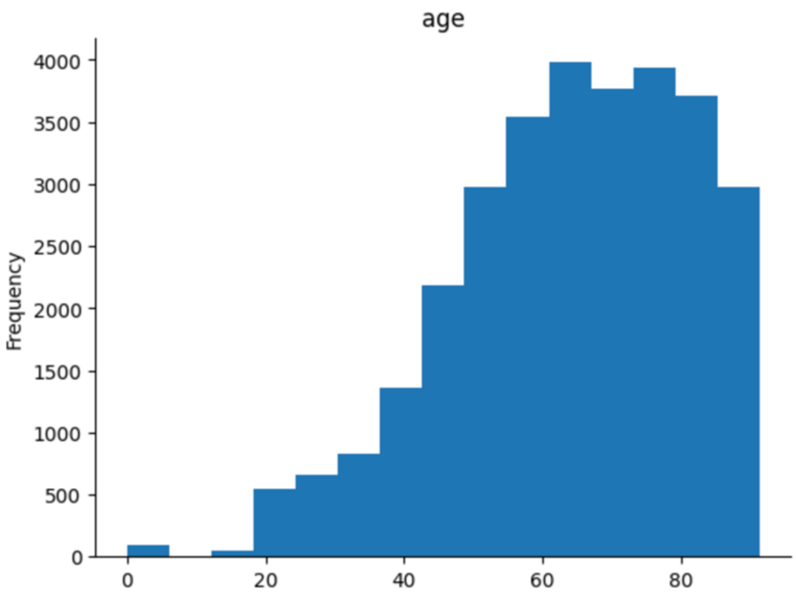
5: Selection Numerical Features for Clustering

Clustering algorithms require numerical inputs, so categorical variables like gender and ethnicity were excluded from the analysis. The final dataset used for clustering consisted solely of numerical features, including lab test results, vital signs, and age, ensuring compatibility with the clustering models.

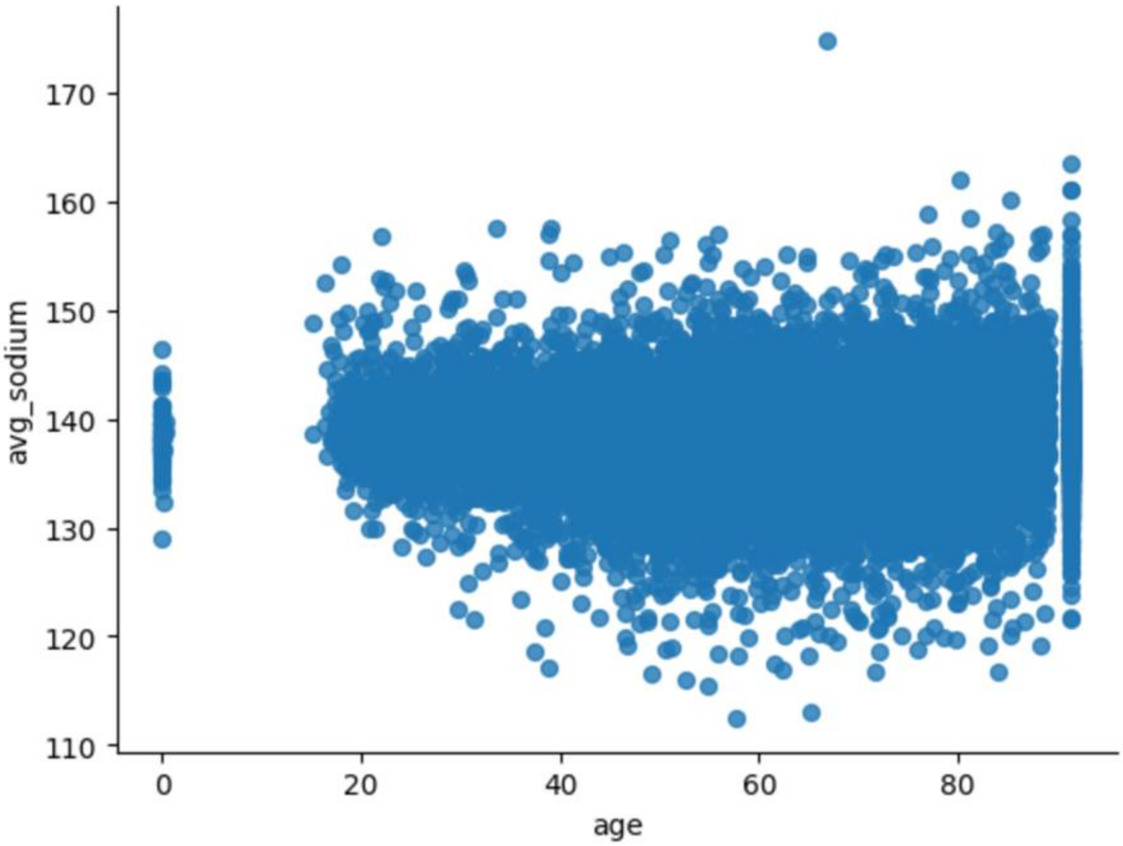
6: EDA(Exploratory Data analysis)

Exploratory Data Analysis (EDA) is a vital step before clustering, as it helps identify missing values, outliers, scaling inconsistencies, and biases in the dataset. Without proper EDA, clustering results may be inaccurate or misleading. By utilizing histograms, scatter plots, and kernel density estimation (KDE) plots, EDA provides insights into the distribution of age, ethnicity, and lab test results. This ensures that clustering is based on meaningful health patterns rather than inconsistencies in the data.

1. Age distribution (Histogram)
   - Generates a histogram with 15 bins to visualize the age distribution.
   - Helps assess whether the dataset is skewed towards younger or older patients, which could influence clustering outcomes.
   - If the age distribution is imbalanced, normalization may be necessary to prevent bias in distance-based clustering algorithms.

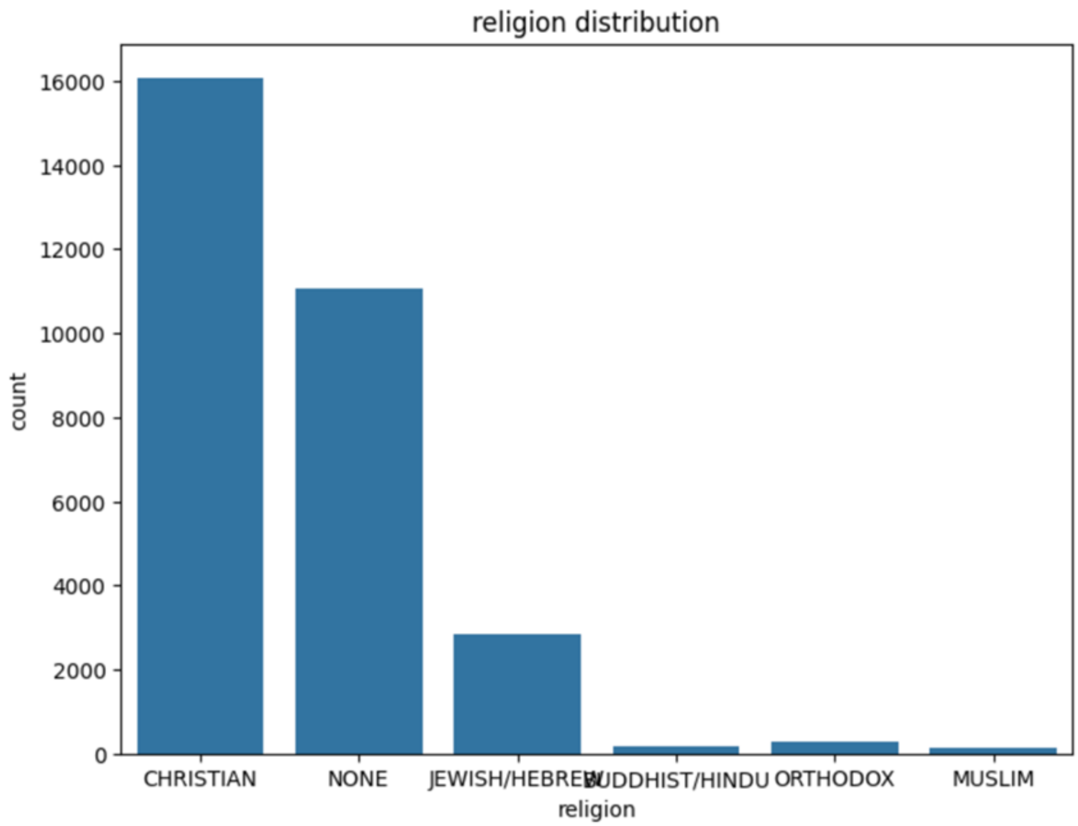


2. Scatter Plot of Age vs. Avg Sodium Levels
   - Generates a scatter plot to visualize the relationship between age and average sodium levels.
   - Helps identify trends or anomalies in sodium levels across different age groups.
   - If specific age groups exhibit significantly different sodium levels, natural clusters may emerge based on age-related health patterns.



3. Religion distribution (Count plot)

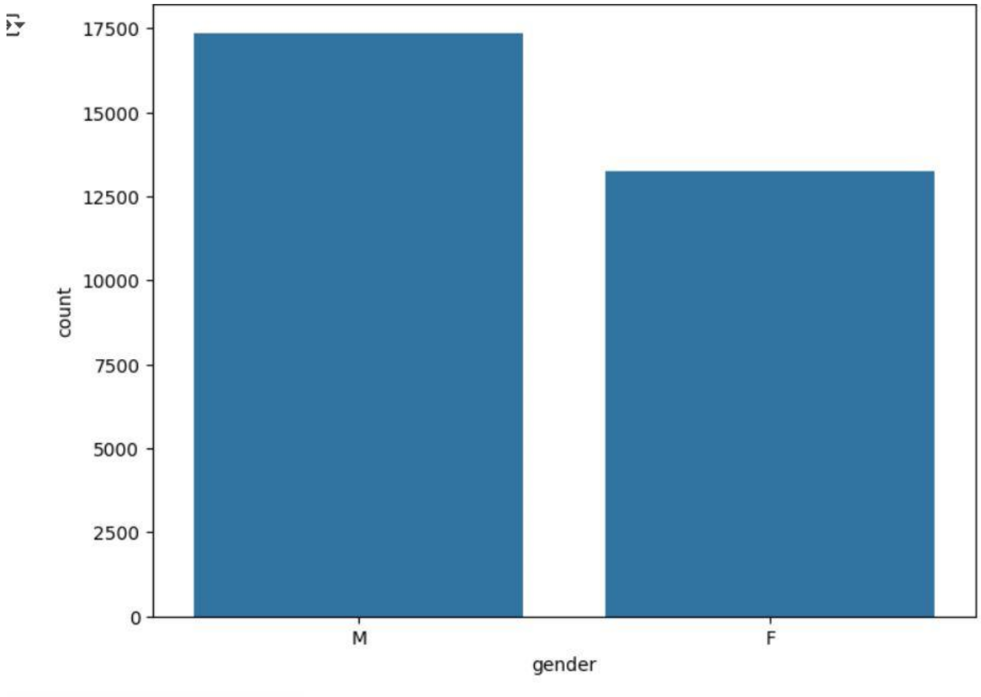
- Generates a bar chart to visualize the distribution of different religions in the dataset.
- Helps identify whether certain demographic groups are disproportionately represented.
- If a single religious group is dominant, clustering results may be skewed, necessitating a more balanced approach to ensure fairness in segmentation.



4. Gender distribution(histogram with KDE)

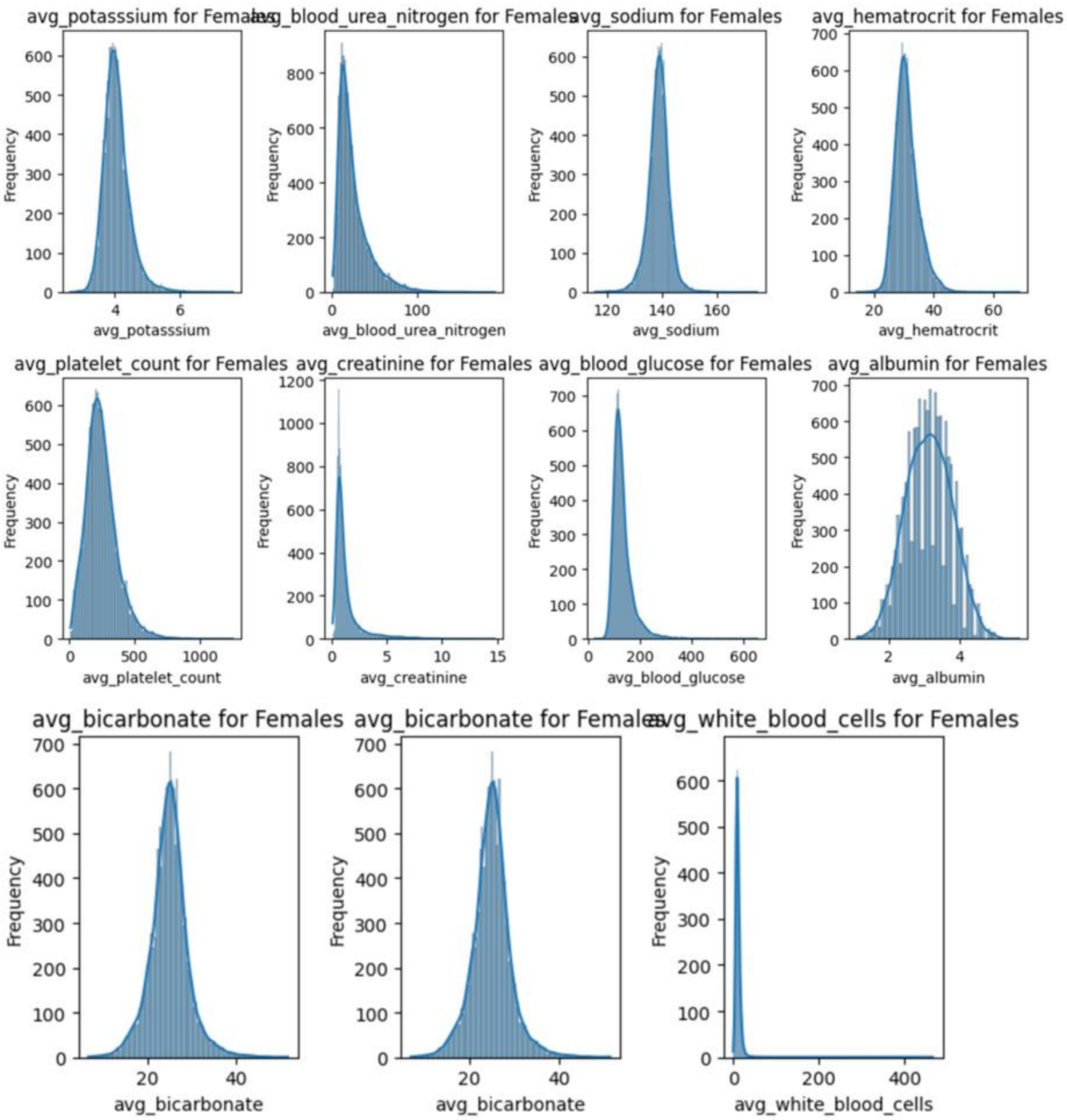
- Plots a histogram of gender distribution with a KDE (Kernel Density Estimation) curve.
- Identifies whether certain gender groups dominate the dataset.

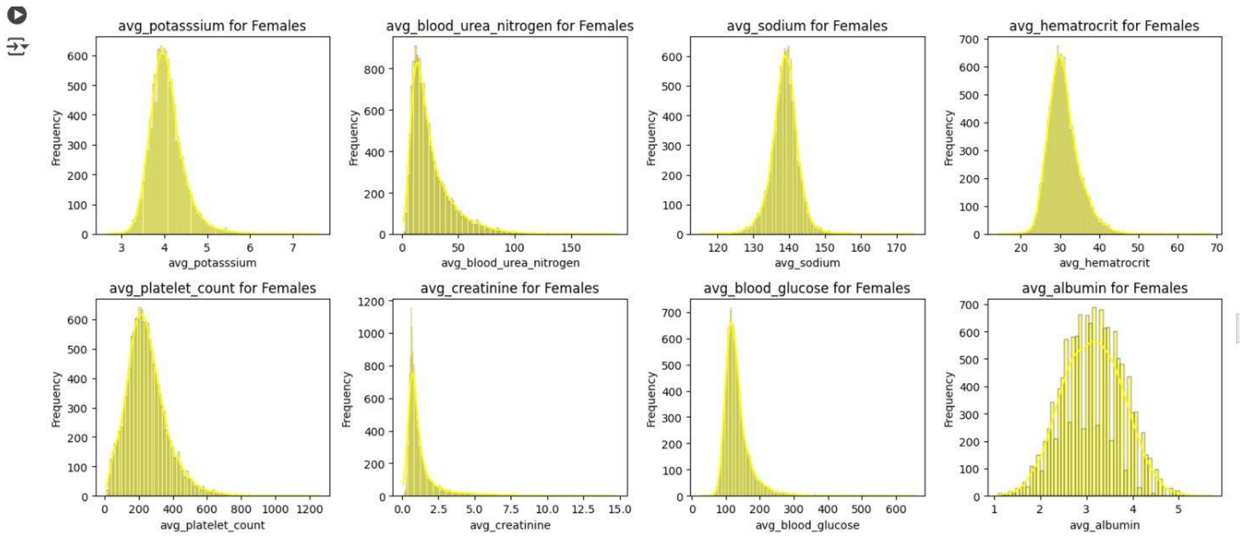
- If certain gender groups correlate with specific lab values or vital signs, form separate.

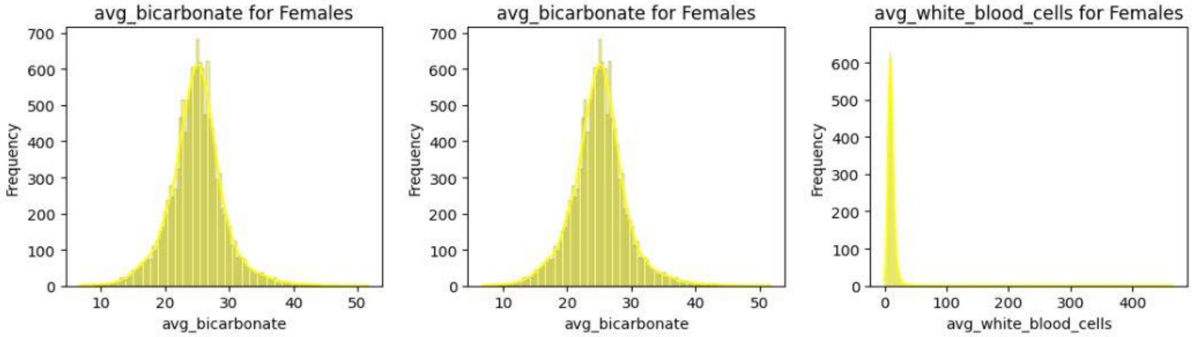


5. Lab Test Distributions for Female Patients

- Generates histograms with KDE curves for various lab tests among female patients.
- Helps identify outliers in lab test results.
- Reveals whether certain lab tests exhibit skewed distributions, indicating the need for normalization.
- If lab test values show high variance, standardization is necessary before clustering.
- Ensures that clustering is driven by meaningful differences rather than scale-related distortions.


avg_potasssium for Females
Frequency
0
100
200
300
400
500
600
4
6
avg_potasssium
avg_blood_urea_nitrogen for Females
0
200
400
600
800
0
100
avg_blood_urea_nitrogen
avg_sodium for Females
0
100
200
300
400
500
600
120
140
160
avg_sodium
avg_hematrocrit for Females
0
100
200
300
400
500
600
700
20
40
60
avg_hematrocrit
avg_platelet_count for Females
0
100
200
300
400
500
600
0
500
1000
avg_platelet_count
avg_creatinine for Females
0
200
400
600
800
1000
1200
0
5
10
15
avg_creatinine
avg_blood_glucose for Females
0
100
200
300
400
500
600
700
0
200
400
600
avg_blood_glucose
avg_albumin for Females
0
100
200
300
400
500
600
700
2
4
avg_albumin
avg_bicarbonate for Females
0
100
200
300
400
500
600
700
20
40
avg_bicarbonate
avg_bicarbonate for Females
0
100
200
300
400
500
600
700
20
40
avg_bicarbonate
avg_white_blood_cells for Females
0
100
200
300
400
500
600
0
200
400
avg_white_blood_cells

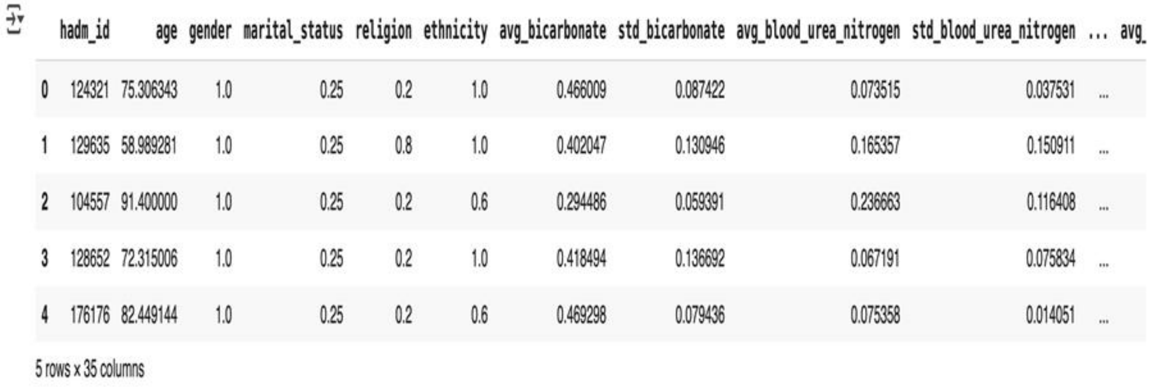




7: Data Normalization (Standardization)

Since numerical features in the dataset have different scales (e.g., blood glucose in mg/dL and temperature in °C), data normalization is applied to maintain consistency across all features. Standardization is performed using **StandardScaler**, which transforms values to have a mean of 0 and a standard deviation of 1. This prevents clustering from being influenced by features with larger numerical values, ensuring that all variables contribute equally to the analysis.

Min-Max Scaling is applied to the numerical features in the dataset, ensuring that all values are normalized within a range of **0 to 1**. This transformation standardizes all numerical features (excluding `hadm_id` and `age`) to a common scale, preventing any single feature from disproportionately influencing the clustering process. Since clustering algorithms rely on distance-based calculations, this step is crucial to eliminating scale-related biases and improving the accuracy of cluster formation.



	hadm_id	age	gender	marital_status	religion	ethnicity	avg_bicarbonate	std_bicarbonate	avg_blood_urea_nitrogen	std_blood_urea_nitrogen	...	avg_
0	124321	75.306343	1.0	0.25	0.2	1.0	0.466009	0.087422	0.073515	0.037531	...	
1	129635	58.989281	1.0	0.25	0.8	1.0	0.402047	0.130946	0.165357	0.150911	...	
2	104557	91.400000	1.0	0.25	0.2	0.6	0.294486	0.059391	0.236663	0.116408	...	
3	128652	72.315006	1.0	0.25	0.2	1.0	0.418494	0.136692	0.067191	0.075834	...	
4	176176	82.449144	1.0	0.25	0.2	0.6	0.469298	0.079436	0.075358	0.014051	...	

5 rows x 35 columns

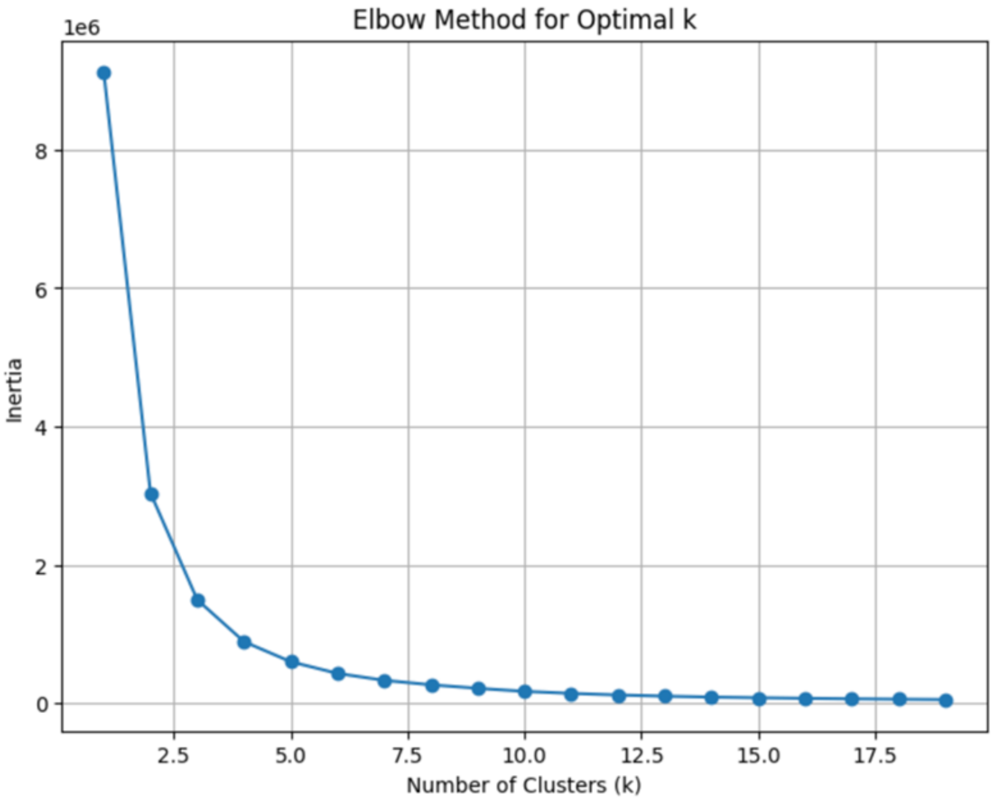
3)Clustering Results:

K-Means Clustering Analysis

The Elbow Method plot was used to determine the optimal number of clusters (k). A sharp decline in the Within-Cluster Sum of Squares (WCSS) up to k = 4 indicated that four clusters provide a good balance between compactness and interpretability.

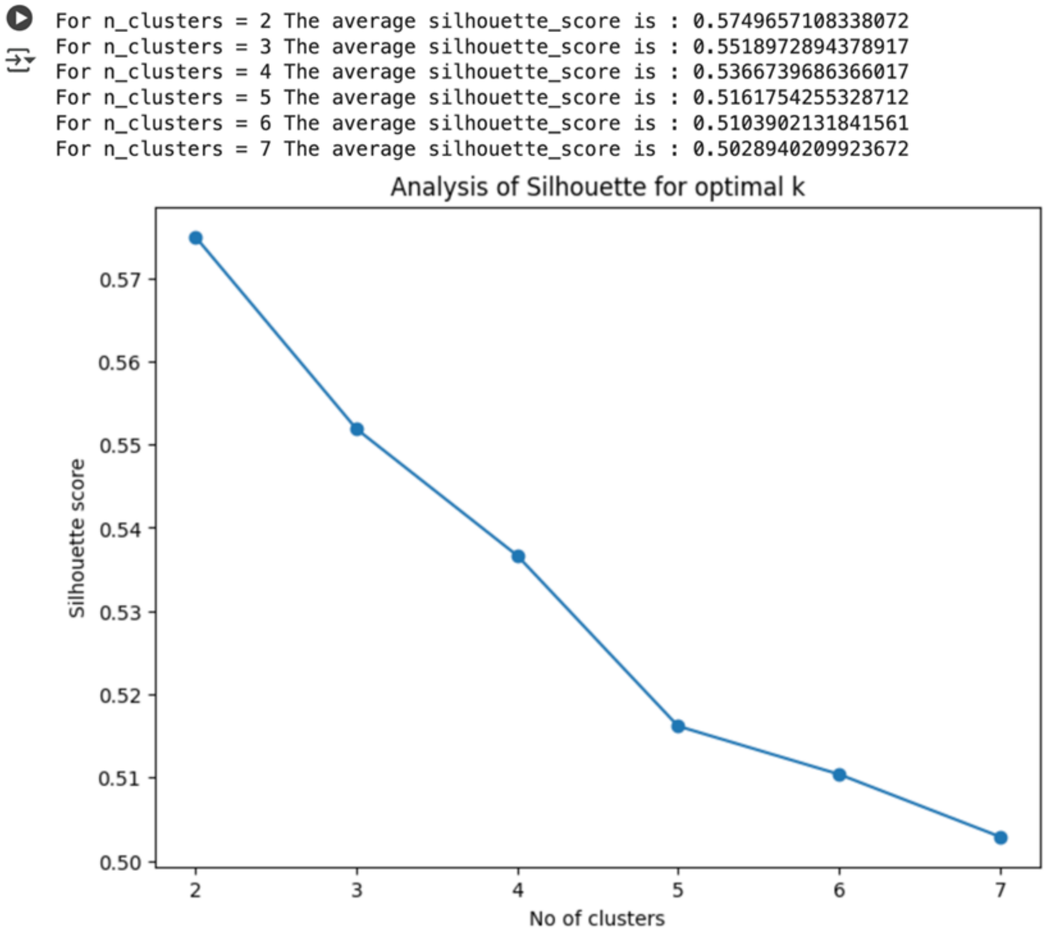
Optimal Number of Clusters (k) Selection

- The Elbow Method suggested that the optimal number of clusters ranged between 4 and 6.
- Beyond 6 clusters, the reduction in within-cluster variance (WCSS) became negligible, indicating that additional clusters provided little benefit.



Silhouette Score Analysis

- The highest silhouette score was observed at k = 2, with scores decreasing for larger values of k.
- This suggests that a smaller number of clusters (2-4) resulted in better-defined and more meaningful patient groupings.



```
For n_clusters = 2 The average silhouette_score is : 0.5749657108338072
For n_clusters = 3 The average silhouette_score is : 0.5518972894378917
For n_clusters = 4 The average silhouette_score is : 0.5366739686366017
For n_clusters = 5 The average silhouette_score is : 0.5161754255328712
For n_clusters = 6 The average silhouette_score is : 0.5103902131841561
For n_clusters = 7 The average silhouette_score is : 0.5028940209923672
```

Cluster Interpretations:

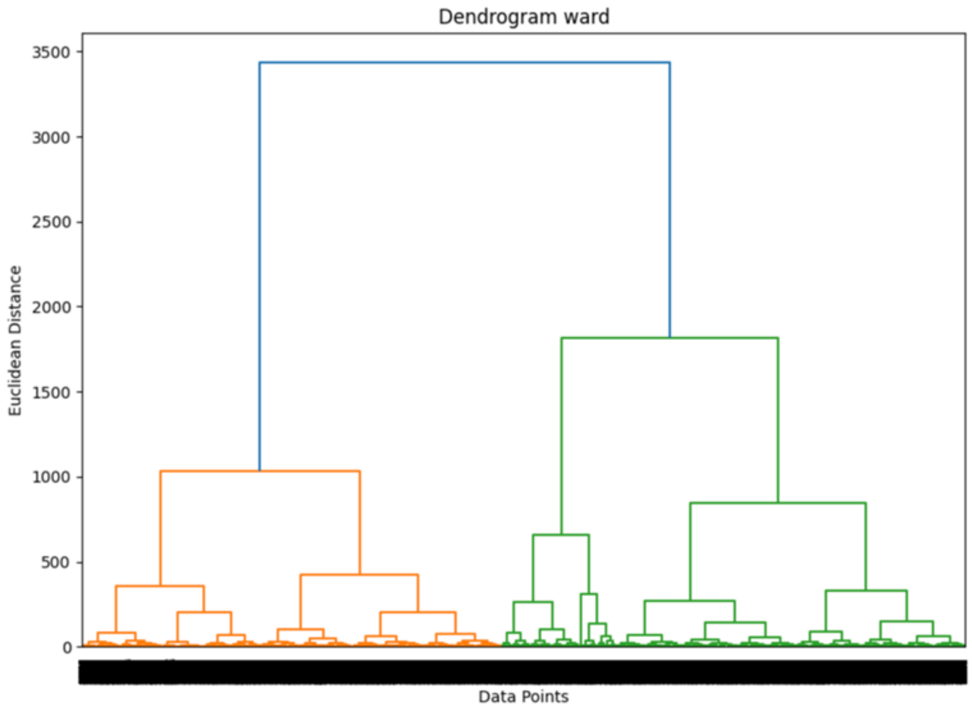
- **Cluster 1**: Patients with elevated blood glucose and abnormal white blood cell counts, potentially indicating diabetes or immune system disorders.
- **Cluster 2**: Older individuals with high blood pressure and increased creatinine levels, suggesting potential risks for kidney and cardiovascular conditions.
- **Cluster 3**: Patients with moderate lab test values, representing a general category without significant abnormalities.
- **Cluster 4**: Individuals with consistently stable lab values, likely indicating overall good health or well-managed medical conditions.

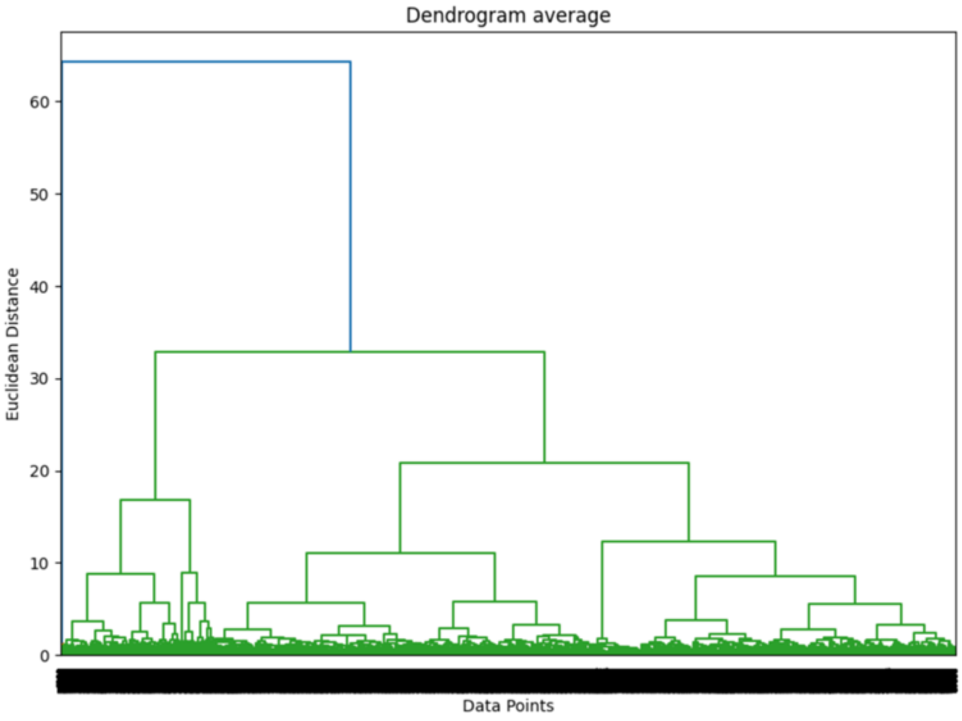
Hierarchical Clustering:

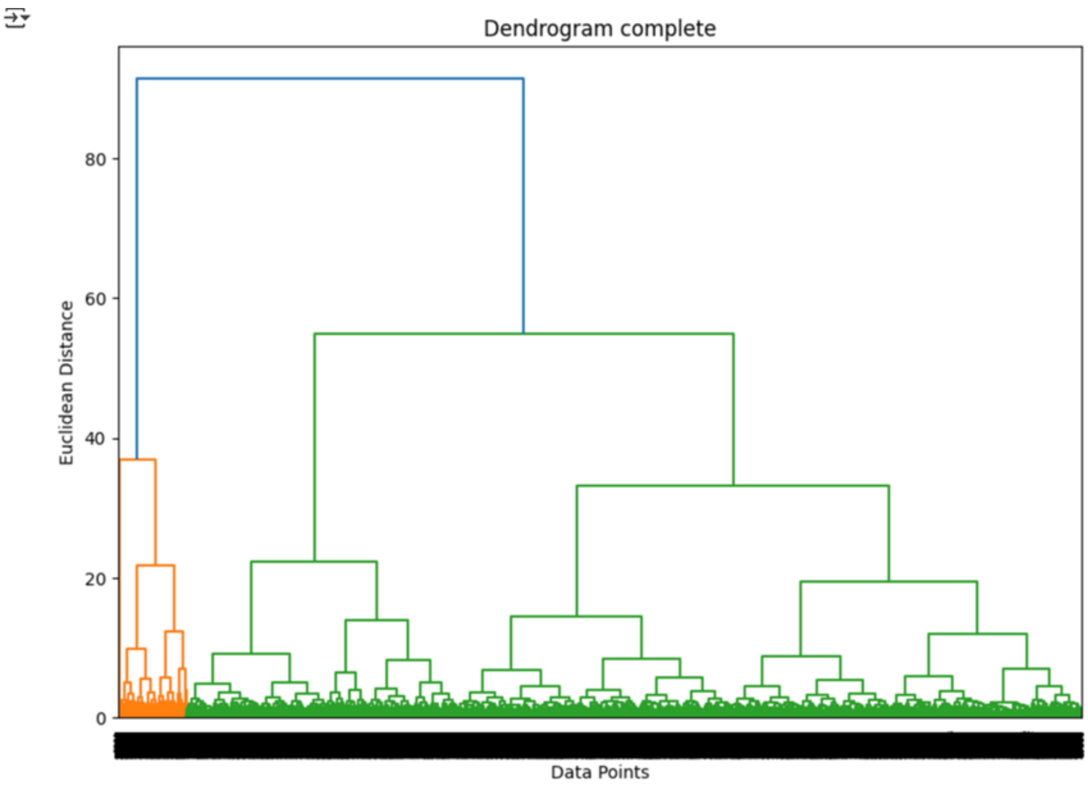
- The dendrogram from hierarchical clustering provided a visual representation of how patient groups formed at different levels of similarity.
- The hierarchical structure revealed sub-clusters within major groups, indicating varying risk levels among patients.
- Ward’s linkage method was used to minimize variance within clusters, ensuring a well-structured grouping of patients.
- Specific clusters emerged for patients with high blood glucose and blood pressure, highlighting potential risks for diabetes and hypertension that require further medical attention.
- Clusters with extremely high or low white blood cell (WBC) counts suggested possible infection risks or immune system disorders, making these patients strong candidates for further screening.

Dendrogram Interpretation

- The dendrogram suggested 3 to 4 main clusters, aligning with K-Means results.
- Two major branches were identified, further breaking into subgroups, indicating different levels of patient severity.


Dendrogram ward
3500
3000
2500
2000
1500
1000
500
0
Euclidean Distance
Data Points

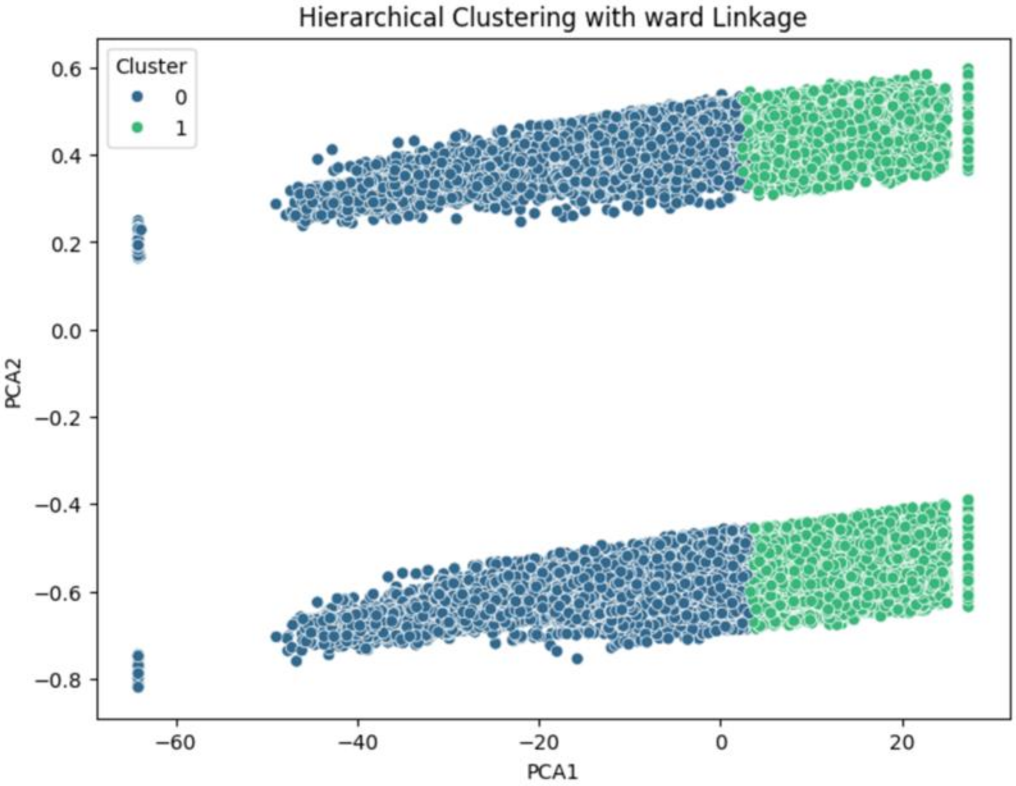

Dendrogram average
60
50
40
30
20
10
0
Euclidean Distance
Data Points



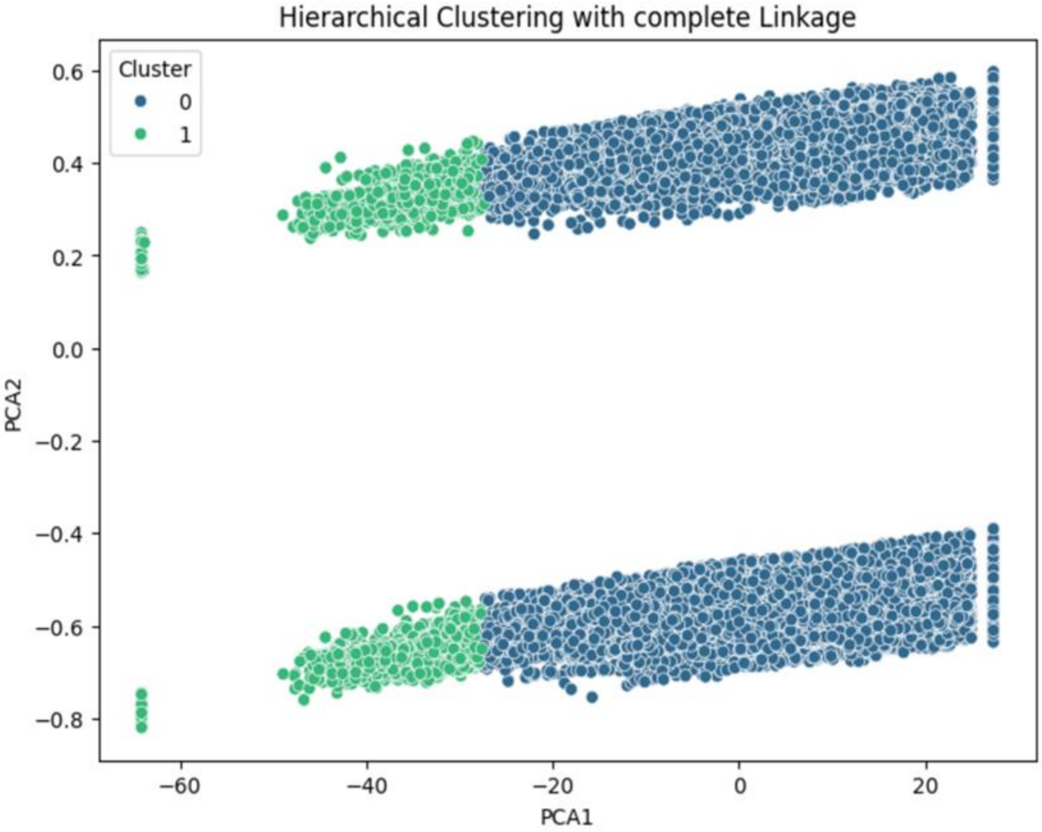
Hierarchical Clustering Performance

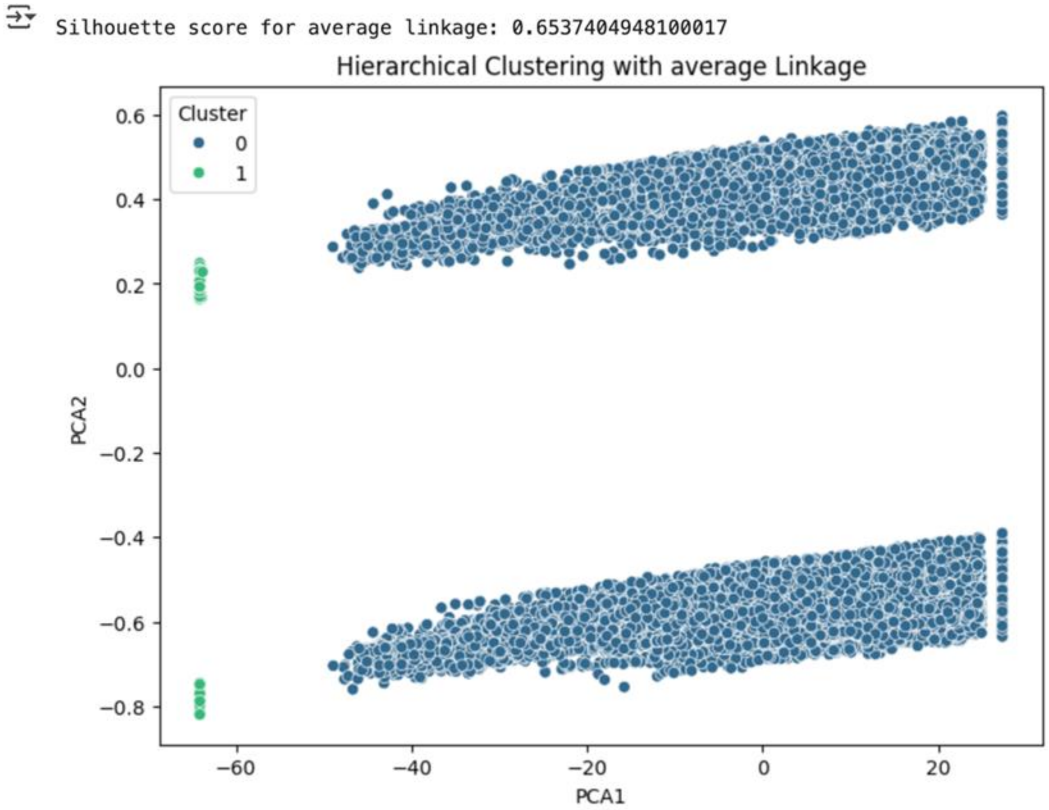
- The Silhouette Score for hierarchical clustering was highest at 2 clusters, similar to K-Means results.
- A PCA visualization of clusters showed distinct group separations, confirming that clustering was meaningful.

```
Silhouette score for ward linkage: 0.5594468390820339
```


Hierarchical Clustering with ward Linkage

```
Silhouette score for complete linkage: 0.5280902524856409
```

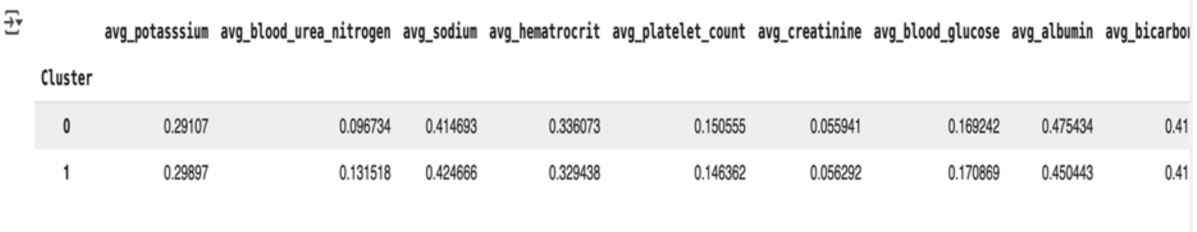

Hierarchical Clustering with complete Linkage



```
Silhouette score for average linkage: 0.6537404948100017
```

Cluster Characteristics

- Hierarchical clustering revealed subgroups within larger clusters, showing finer divisions in patient health profiles.
  Unlike K-Means, it did not require defining `k` beforehand, making it useful for exploratory analysis



	avg_potasssium	avg_blood_urea_nitrogen	avg_sodium	avg_hematrocrit	avg_platelet_count	avg_creatinine	avg_blood_glucose	avg_albumin	avg_bicarbo
Cluster									
0	0.29107	0.096734	0.414693	0.336073	0.150555	0.055941	0.169242	0.475434	0.41
1	0.29897	0.131518	0.424666	0.329438	0.146362	0.056292	0.170869	0.450443	0.41

4)Comparison: K-Means vs. Hierarchical Clustering Results

The application of K-Means and Hierarchical Clustering to the dataset highlighted notable differences in how each technique segments patients based on demographics, lab results, and vital signs. While both methods effectively grouped patients, their performance, interpretability, and ideal use cases varied significantly.

K-Means Clustering proved to be computationally efficient, making it well-suited for large datasets like this one, which contains over 58,000 patient records. Since K-Means requires a predefined number of clusters (k), the Elbow Method was used to determine an optimal range. Silhouette Score analysis suggested that between 2 and 4 clusters provided the best separation of patient groups. The method successfully classified patients into distinct clusters based on factors such as high blood glucose, blood pressure, and immune system responses. Its efficiency makes it particularly valuable for hospital management, where real-time patient segmentation is crucial for quick decision-making.

In contrast, Hierarchical Clustering generated a tree-like structure (dendrogram), illustrating how patient clusters relate at different levels of similarity. Unlike K-Means, it does not require specifying the number of clusters in advance, allowing for a more flexible exploration of patient relationships. However, due to its high computational complexity, it was applied to a random sample of 1,000 patient records instead of the full dataset. This method uncovered subgroup relationships, making it particularly beneficial in medical research for understanding disease progression and subgroup classification.

One key limitation of K-Means is that it assigns each patient to a single cluster, which may oversimplify complex relationships. Hierarchical Clustering, on the other hand, provides greater flexibility by visualizing patient similarities at multiple levels. However, its computational demands make it impractical for very large datasets unless sampling is used.

Overall, K-Means is the preferred approach for large-scale patient classification due to its speed and efficiency, while Hierarchical Clustering excels in exploratory analysis, particularly for identifying subgroup patterns and studying disease progression. Although both methods produced similar clusters, Hierarchical Clustering offered a clearer visualization of patient relationships, making it a valuable tool for medical research and detailed patient analysis.

5)Conclusion: Insights and Recommendations Based on the Clusters

The application of K-Means and Hierarchical Clustering to patient data successfully uncovered distinct health-related patterns. The resulting clusters, based on demographics, lab test results, and vital signs, provided valuable insights into different patient groups. Key findings include:

- **Four distinct patient clusters identified based on:**
    - Blood glucose levels
    - White blood cell counts
    - Blood pressure & creatinine levels
    - Hematocrit & potassium variability
- **Clinical Insights:**
    - Clusters with high blood glucose and abnormal white blood cell counts suggest an increased risk of diabetes and immune-related conditions.
    - Hierarchical clustering revealed meaningful patient subgroups, making it particularly useful for analyzing disease progression and subgroup classification.

These insights can help improve patient management, early diagnosis, and targeted medical interventions.

Recommendations

Clustering for Risk Assessment

- **Diabetes Screening:** High glucose clusters can help identify patients at risk.
- **Cardiovascular Monitoring:** Clusters with high blood pressure and creatinine levels indicate potential cardiovascular concerns.
- **Immune System Evaluation:** Abnormal white blood cell (WBC) clusters suggest the need for further immune system assessment.

Hybrid Approach

- **K-Means for Real-Time Classification:** Useful in hospital settings for quick patient segmentation and decision-making.

- **Hierarchical Clustering for Research:** Ideal for identifying rare disease subgroups and analyzing disease progression.

Future Enhancements

- **Expanded Feature Set:** Incorporate additional patient data, such as cholesterol levels and medication history, for more precise clustering.
- **Advanced Machine Learning:** Explore deep learning-based clustering techniques to improve segmentation accuracy and predictive insights.

This clustering analysis highlights the power of unsupervised machine learning in healthcare. K-Means proved effective for large-scale patient segmentation, while Hierarchical Clustering provided deeper insights into patient relationships. A hybrid approach combining both methods can enhance risk assessment, improve disease monitoring, and support personalized medical treatments.