NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC4020 Data Analytics and Mining

Project 1

Data Decomposition using Clustering Analysis

Name	Matriculation Number
Allen Lu Zhao Quan	U2320787A
Gao Xin Yue	U2322378A
Tio Hilda	U2321338F
Tio Sher Min	U2320324H

Abstract:

This report provides an in-depth analysis of clustering algorithms used to reveal hidden patterns and natural groupings in complex datasets. It evaluates the performance of several techniques, including K-Means, K-Means++, Agglomerative Hierarchical Clustering, and Gaussian Mixture Models. The study examines how different hyperparameter choices influence the effectiveness of each method, identifies optimal configurations, and presents a comparative assessment of the performance of each algorithm. The report aims to improve the precision and scalability of clustering approaches, delivering valuable insights applicable to a wide range of industries, including but not limited to the social sciences, biological sciences and health sciences examined in this report.

Table of Contents

1. Introduction

1.1. Definition of Clustering

Clustering is a type of unsupervised machine learning that organises unlabeled data points into groups called clusters, based on how similar they are. The similarity between points is typically measured using distance metrics like Euclidean distance or cosine similarity. This technique helps identify natural groupings or patterns within the data.

1.2. Use Cases

Clustering is a valuable technique used across various industries to tackle different challenges by helping users make sense of complex data. Common uses include segmenting markets, grouping search results, and detecting anomalies. Additionally, clustering serves as an effective tool to improve other machine learning models, especially when handling large volumes of unlabeled data. By organizing similar data points into groups, clustering uncovers hidden patterns and structures within the data, enabling users to gain deeper insights and a more thorough understanding of the information.

1.3. Datasets

In this project, we analysed three datasets from Kaggle, each containing data across different fields. To showcase different clustering algorithms, we selected a diabetes dataset for examining diabetes status; a body fat dataset for Body Fat percentage segmentation; and a world bank dataset for grouping, which could serve as a preprocessing step for determining the quality of life.

Our analysis utilised charts and multiple metrics to evaluate the datasets. Additionally, these datasets were used to perform comparative analysis of different algorithms applied to the same data, dissecting their strengths and shortcomings, to derive insightful results on which algorithm is best suited for specific data characteristics.

Table 1: Diabetes Prediction Dataset (Diabetes Examination, n.d.)

Variable	Description
Gender	The biological sex of the individual.
Age	The chronological number of years a person has lived thus far.
Hypertension	Hypertension is a medical condition in which the blood pressure

	in the arteries is persistently elevated.
Heart_Disease	A medical condition that refers to any disorder of the heart or blood vessels.
Smoking_History	Whether a person has smoked before.
BMI	A measure of body fat based on weight and height.
HbA1c_Level	A measure of a person's average blood sugar level over the past 2-3 months.
Blood_Glucose_Level	The amount of glucose in the bloodstream at a given time.

Table 2: Body Fat Prediction Dataset (BMI Classification, n.d.)

Variable	Description
Density	The body density of a person.
Age	The chronological number of years a person has lived thus far.
Weight	The weight of a person measured in the air.
Height	The height of a person.
Neck_Circumference	The neck circumference of a person.
Chest_Circumference	The chest circumference of a person.
Abdomen_Circumference	The abdomen circumference of a person.
Hip_Circumference	The hip circumference of a person.
Thigh_Circumference	The thigh circumference of a person.
Knee_Circumference	The knee circumference of a person.
Ankle_Circumference	The ankle circumference of a person.
Biceps_Circumference	The bicep circumference of a person.
Forearm_Circumference	The forearm circumference of a person.
Wrist_Circumference	The wrist circumference of a person.

Table 3: Word Bank Dataset (Quality of life measurement, n.d.)

Variable	Description
Country	Name of the country for which the data is recorded.
Year	The year in which the data is observed.
Gross_Domestic_Product	The Gross Domestic Product of the country in billions of US dollars, indicating the economic output.
Population	The total population of the country in millions.
Life_Expectancy	The average number of years a newborn is expected to live, assuming that current mortality rates remain constant throughout their life.
Unemployment_Rate	The percentage of the total labour force that is unemployed but actively seeking employment.
CO2_Emissions	The amount of carbon dioxide emissions per person in the country, measured in metric tons.
Access_to_Electricity	The percentage of the population with access to electricity.

2. Clustering Methods

2.1. K-Means

2.1.1. Mechanism

K-Means clustering is an unsupervised learning method used to categorise unlabeled data into distinct clusters. It is among the most popular clustering techniques in data science, often applied to tasks like market segmentation, document clustering, image segmentation, and fraud detection. The algorithm aims to reduce the variance within clusters by repeatedly recalculating the centroids and reassigning data points to their nearest centroid until a stable clustering solution is reached.

While Euclidean distance is conventionally used to measure the similarity between points, it can also use cosine distance in cases where the direction of the data vectors is more important, such as high-dimensional sparse datasets.

2.1.2. Methodology

1. **Initialisation:** Specify and select K random centroids from the dataset. These centroids serve as a starting point for the clusters.
2. **Assignment:**
    a. **Using Euclidean Distance**: Each point is assigned to the nearest cluster based on Euclidean Distance, calculated based on the shortest straight-line distance in 2-dimension between 2 points.
    b. **Using Cosine Distance**: Data points are assigned to the nearest centroid based on the smallest cosine distance, which measures the angular difference between vectors.
3. **Centroid Update**: Compute the new centroid for each cluster based on the mean of the data points assigned to the cluster.
    a. **Using Euclidean Distance**: The new centroid is calculated based on the geometric mean of the data points for each cluster.
    b. **Using Cosine Distance**: The new centroid is calculated based on the vector that minimises the cosine distance from the points within the cluster
4. **Repeat:** The algorithm repeats between the assignment and update step until convergence is reached, i.e., the centroids no longer change significantly or a maximum number of iterations is reached.

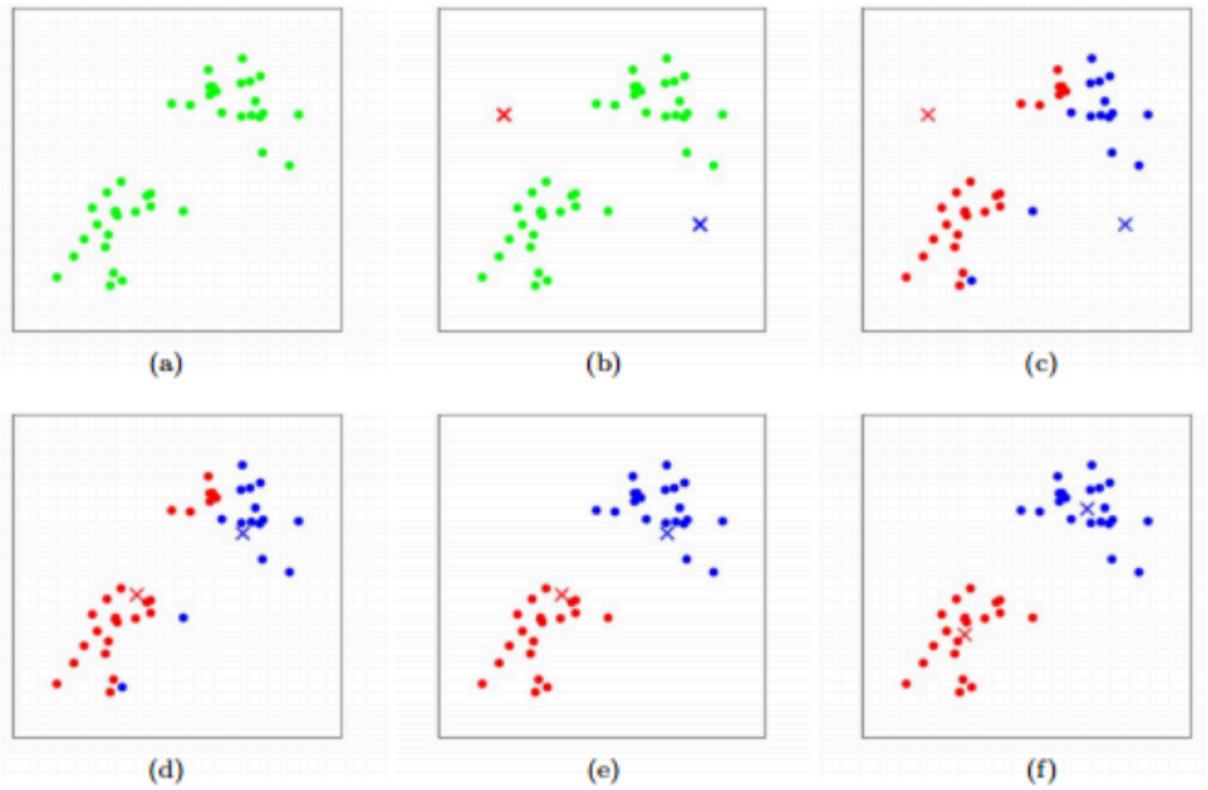


Figure 1: Illustration of the K-Means algorithm. Training data points are shown as dots while centroids are crosses. (a) Original dataset, (b) Randomly initialised centroids (c) - (f) Illustration of running two rounds of K-Means clustering (Piech, 2012)

2.1.3. Training

2.1.3.1. K-Means Initiation in Diabetes Prediction Dataset

We first applied K-Means clustering to the Diabetes Prediction Dataset. The goal was to determine underlying relationships among different health metrics that may be associated with diabetes risk.

To assess how dimensionality reduction and distance metrics influence clustering performance, four different scenarios were implemented and compared:

1. **PCA Before Clustering (Euclidean Distance):**
   Principal Component Analysis (PCA) was applied to reduce dimensionality before running the K-Means algorithm. This preprocessing step reduces noise, minimises feature redundancy and retains the most significant variance in the dataset, leading to potentially clearer cluster separation. Clustering was performed using the Euclidean distance metric to partition patients into clusters based on the principal components.

2. **PCA After Clustering (Euclidean Distance):**
   K-Means was first applied directly on the original high-dimensional data using the Euclidean distance metric. PCA was then performed for post-clustering visualisation for easier interpretation and analysis of patient groupings.

3. **PCA Before Clustering (Cosine Distance):**
   PCA was conducted on the normalised dataset prior to clustering to reduce dimensionality and compress the data. The normalised principal components were clustered using the cosine distance metric.

4. **PCA After Clustering (Cosine Distance):**
   Normalisation was first applied to the dataset, followed by K-Means clustering using cosine distance to form clusters. After clustering, PCA was conducted for visualisation and to identify which features contributed most strongly to each cluster's separation.

Since cosine and Euclidean distances are linearly related for normalised vectors, applying normalisation allows Euclidean distance to behave similarly to cosine distance. This ensures a more meaningful comparison between clustering results across the four scenarios.

Before applying K-Means, the optimal number of clusters K, for each scenario is determined using the **Elbow Method** and **Silhouette Score**.

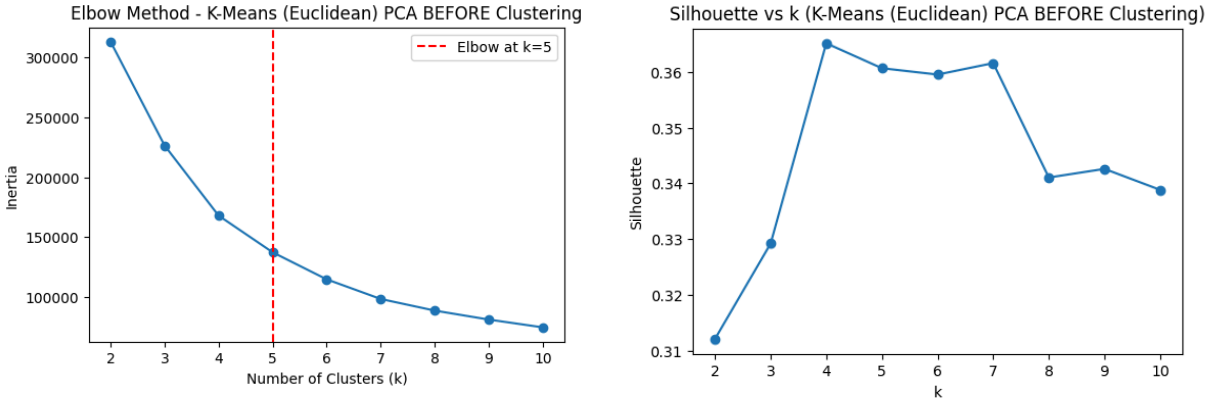


Figure 2: Illustration of Elbow Method and Silhouette Score for PCA Before Clustering (Euclidean Distance)

From Figure 2, we observe that the optimal number of clusters differ between the two methods for PCA Before Clustering (Euclidean Distance). For this analysis, we choose the optimal number of clusters based on the Silhouette Score method as it provides a better measure of how well-defined the clusters are by accessing the cohesion and separation of data points.

For the remaining scenarios, the optimal number of clusters are also chosen in a similar manner, i.e. we use the optimal number of clusters based on the Silhouette Score.

The K-Means algorithm was then performed for the above mentioned scenarios.

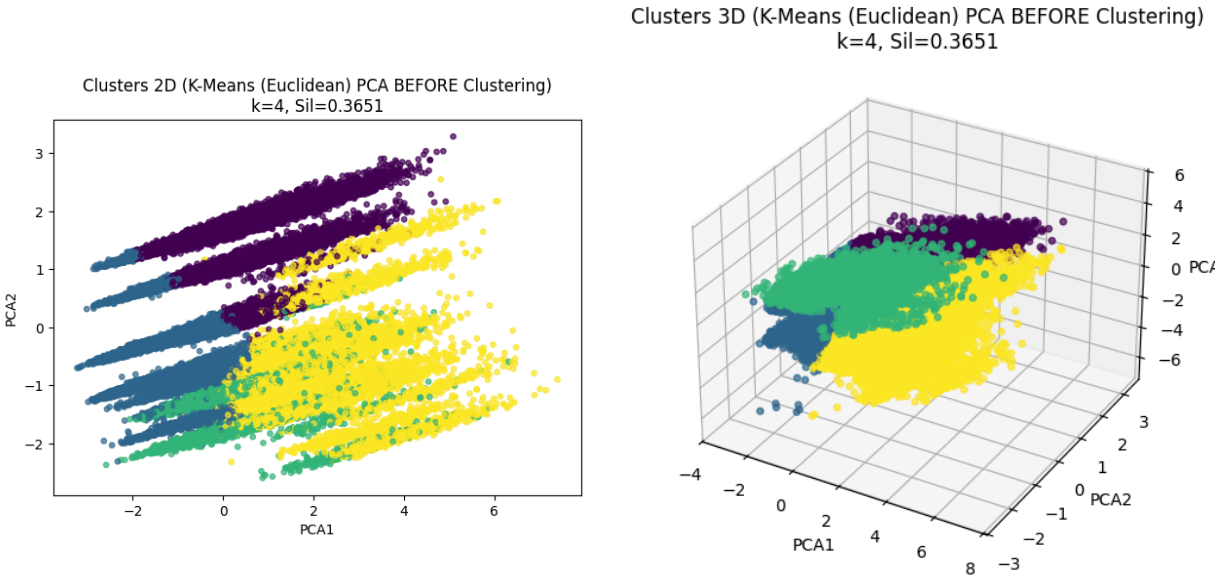


Figure 3: 2D and 3D Scatter Plot for PCA Before Clustering (Euclidean Distance)

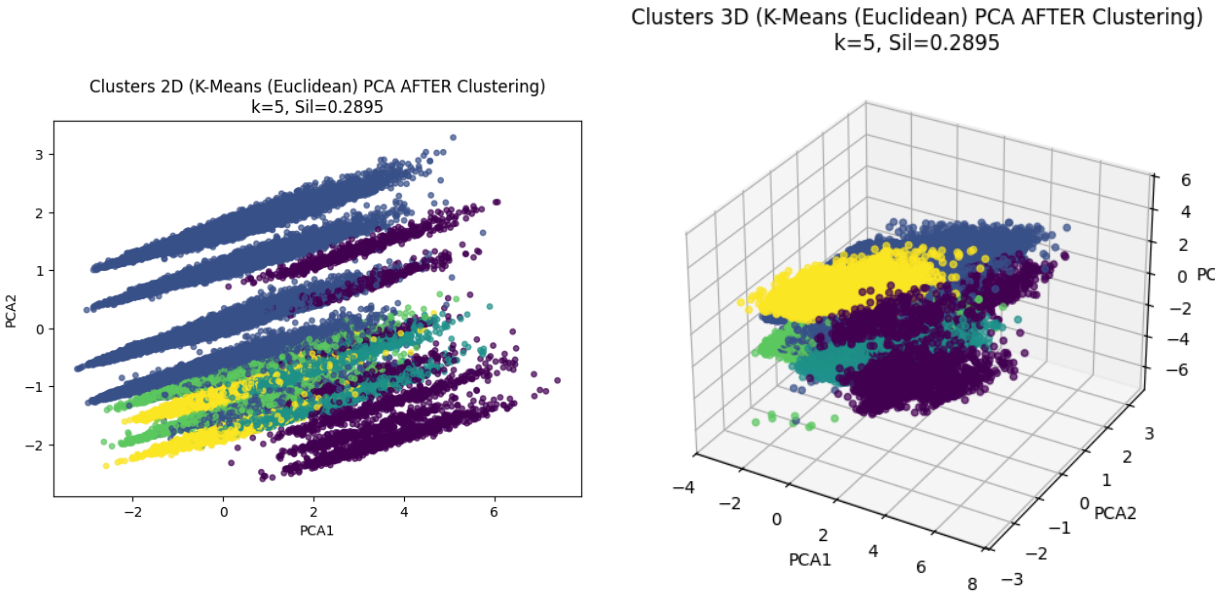


Figure 4: 2D and 3D Scatter Plot for PCA After Clustering (Euclidean Distance)

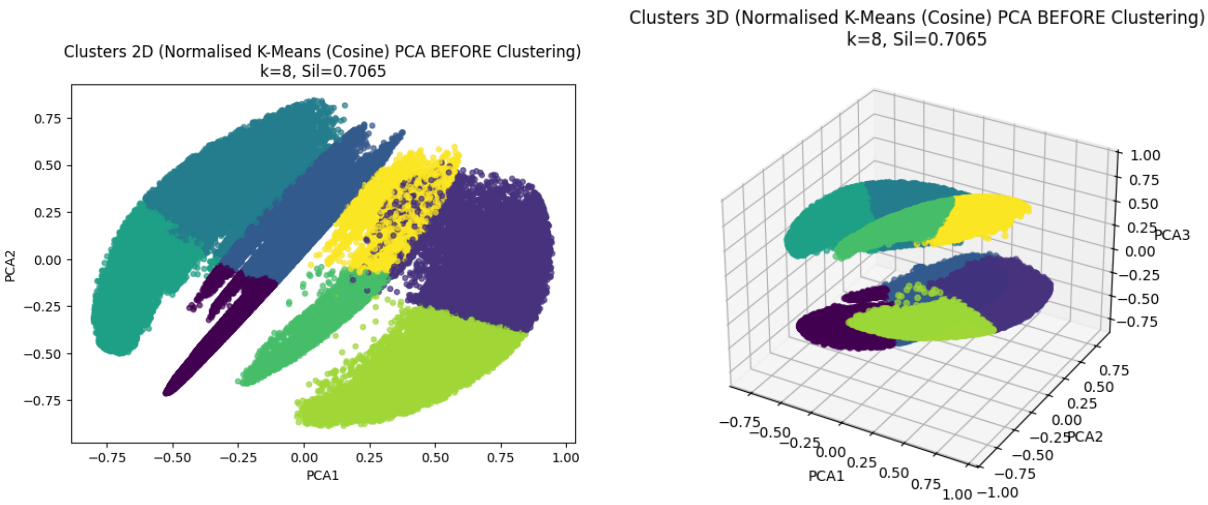


Figure 5: 2D and 3D Scatter Plot for PCA Before Clustering (Cosine Distance)

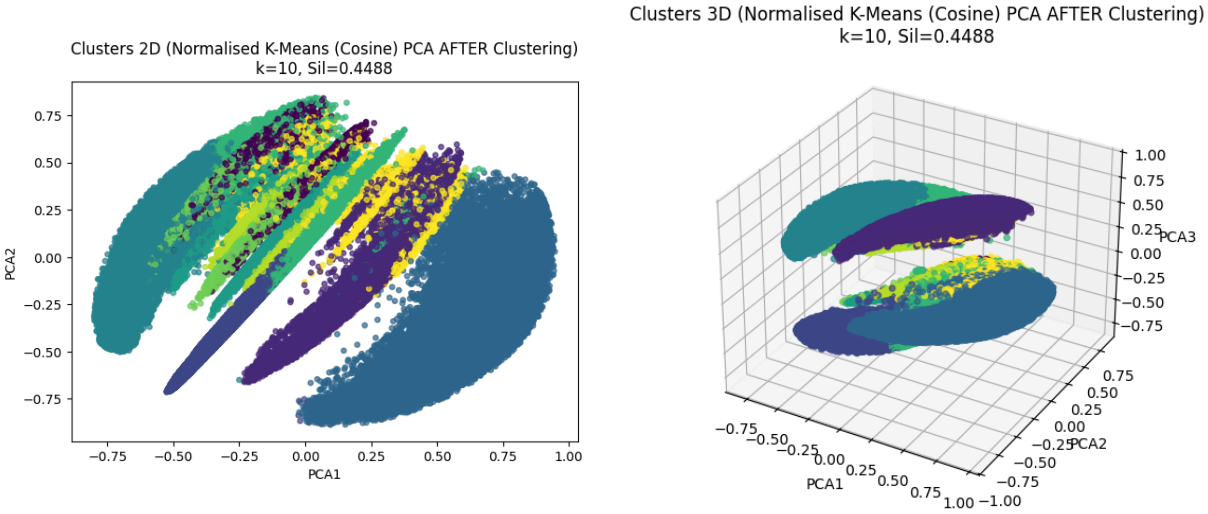


Figure 6: 2D and 3D Scatter Plot for PCA After Clustering (Cosine Distance)

2.1.3.2. K-Means Initiation in Body Fat Prediction Dataset

K-Means is applied to the Body Fat Prediction Dataset to explore relationships between body composition measurements and overall body fat percentage, helping to identify key physical factors that influence body fat levels.

For this dataset, we use the same four scenarios and Silhouette Score method for determining the optimal number of clusters for each scenario, following the same workflow as the Diabetes Prediction Dataset.

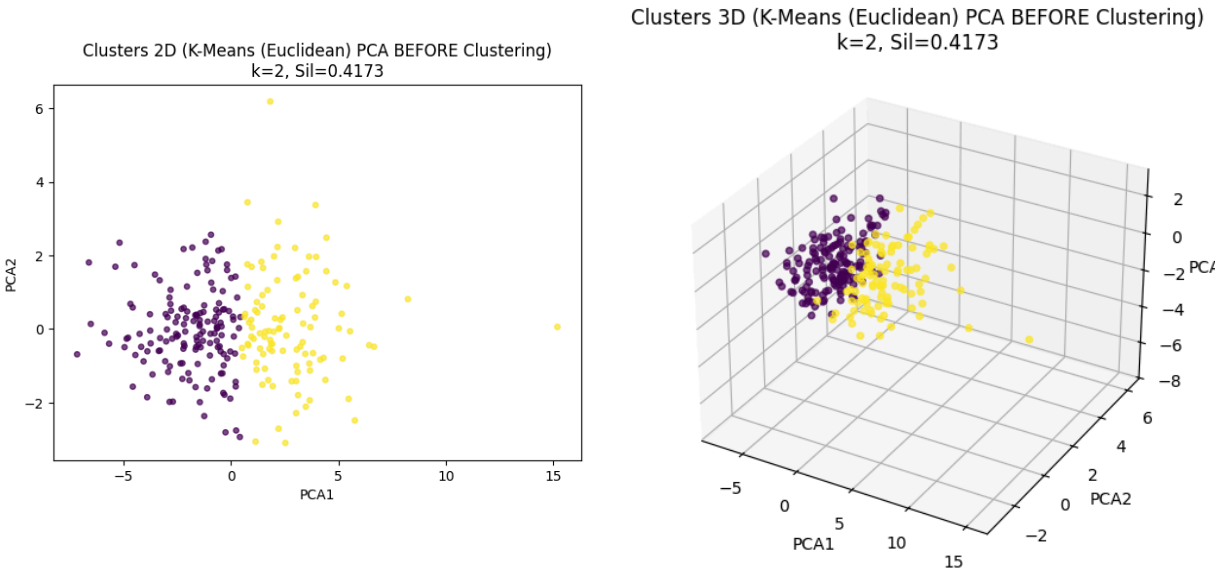


Figure 7: 2D and 3D Scatter Plot for PCA Before Clustering (Euclidean Distance)

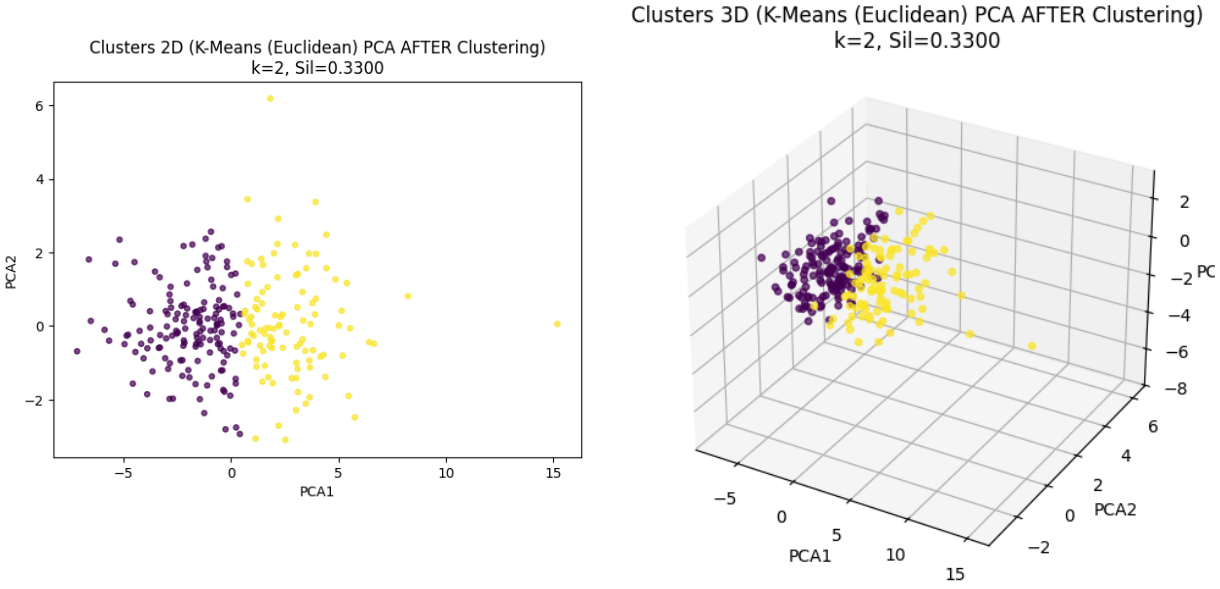


Figure 8: 2D and 3D Scatter Plot for PCA After Clustering (Euclidean Distance)

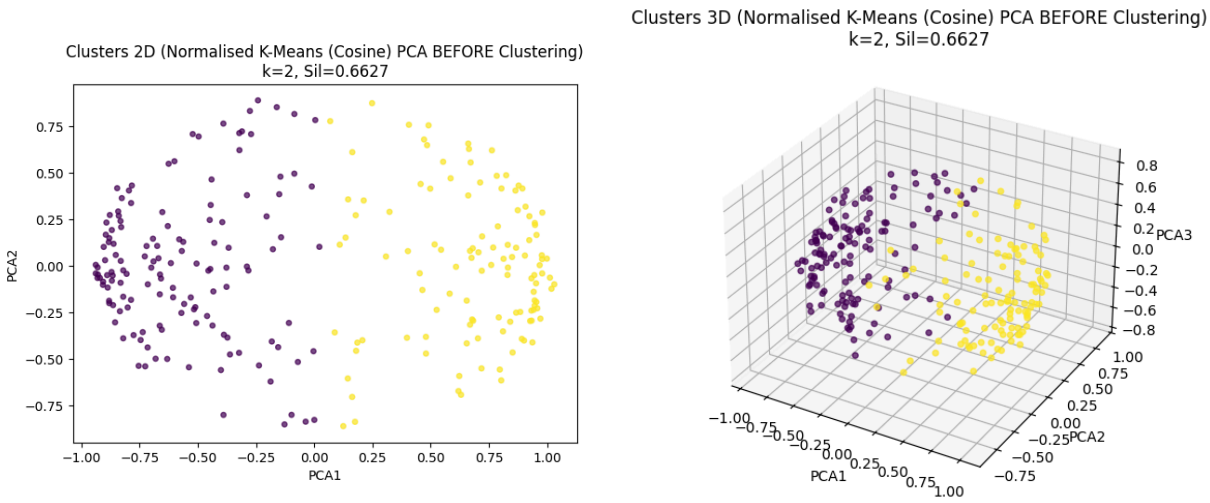


Figure 9: 2D and 3D Scatter Plot for PCA Before Clustering (Cosine Distance)

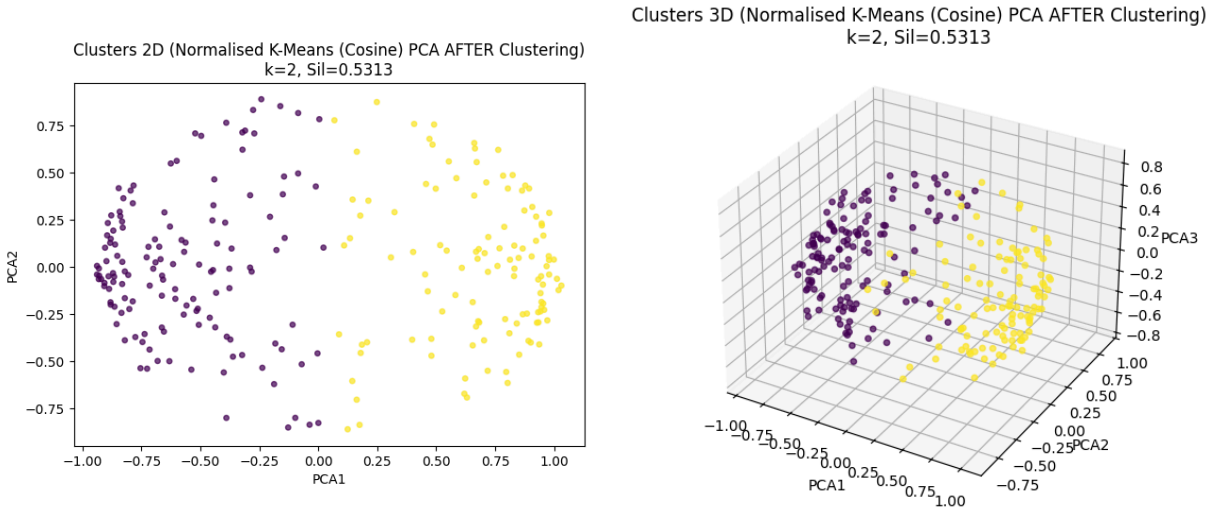


Figure 10: 2D and 3D Scatter Plot for PCA After Clustering (Cosine Distance)

2.1.3.3. K-Means Initiation in World Bank Dataset

This analysis applies K-Means to the World Bank Dataset to identify global patterns across economic, social, health, and environmental indicators. By grouping countries with similar development profiles, it provides a basis for assessing quality of life and overall development status.

The same four scenarios, clustering procedures and workflow as those used for the Diabetes Prediction and Body Fat Prediction Datasets are applied.

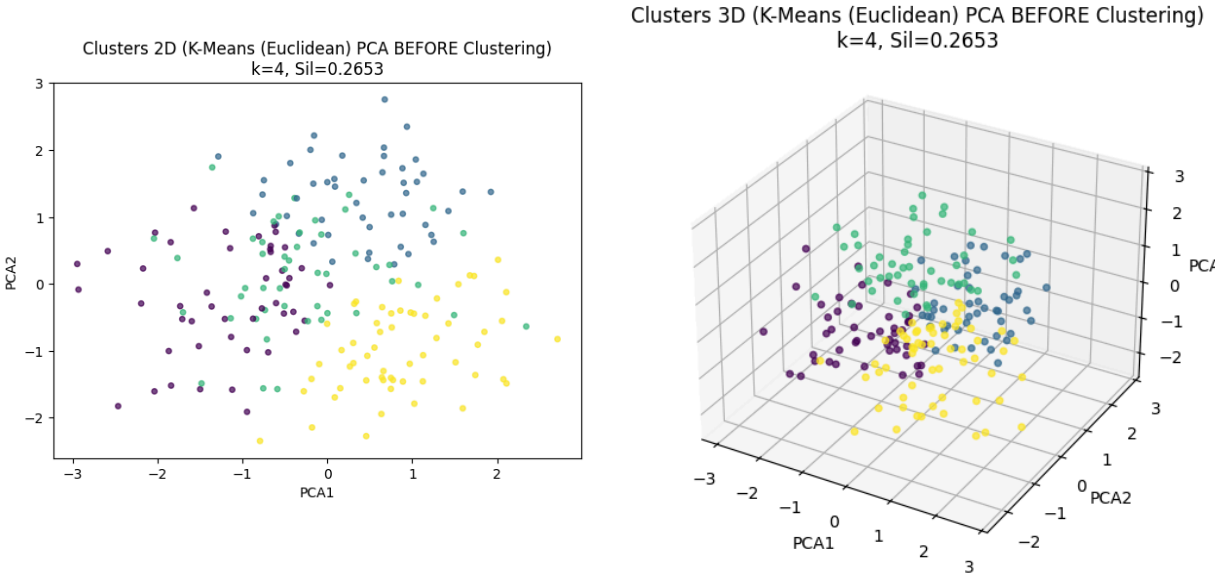


Figure 11: 2D and 3D Scatter Plot for PCA Before Clustering (Euclidean Distance)

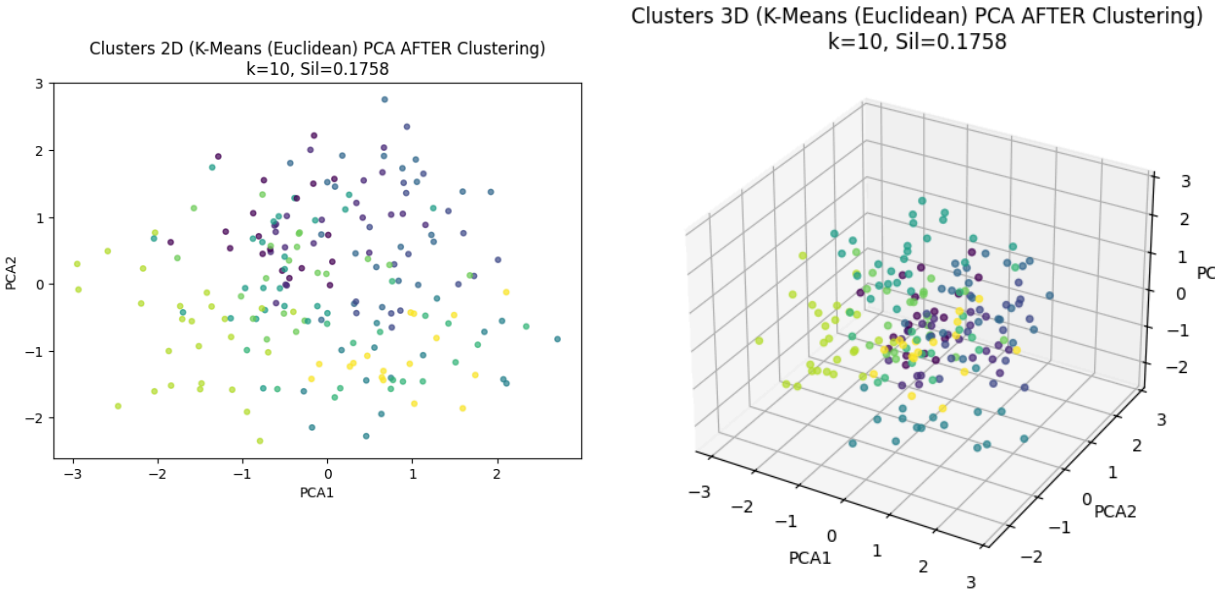


Figure 12: 2D and 3D Scatter Plot for PCA After Clustering (Euclidean Distance)

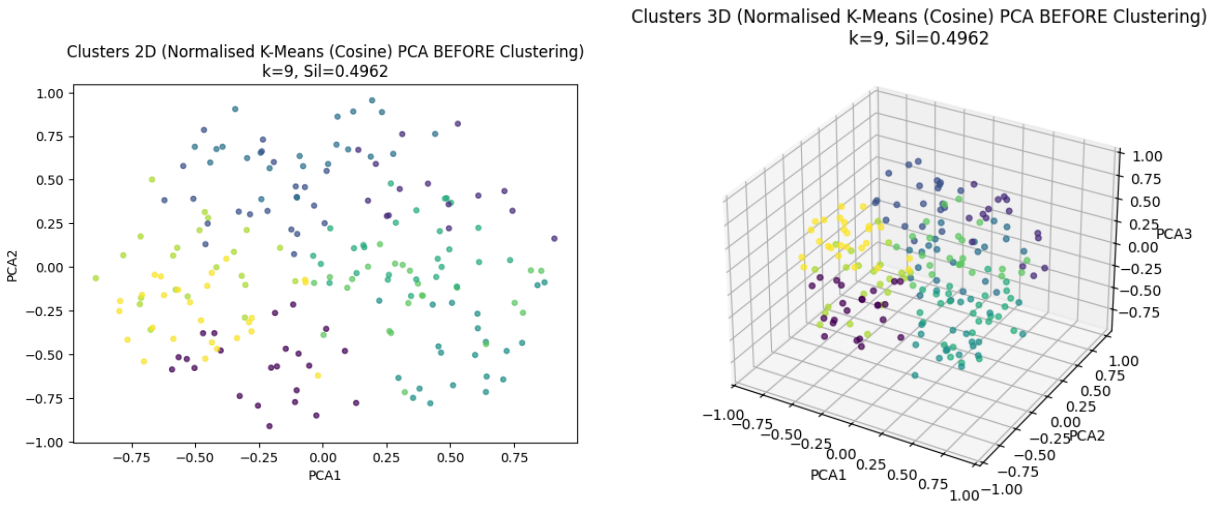


Figure 13: 2D and 3D Scatter Plot for PCA Before Clustering (Cosine Distance)

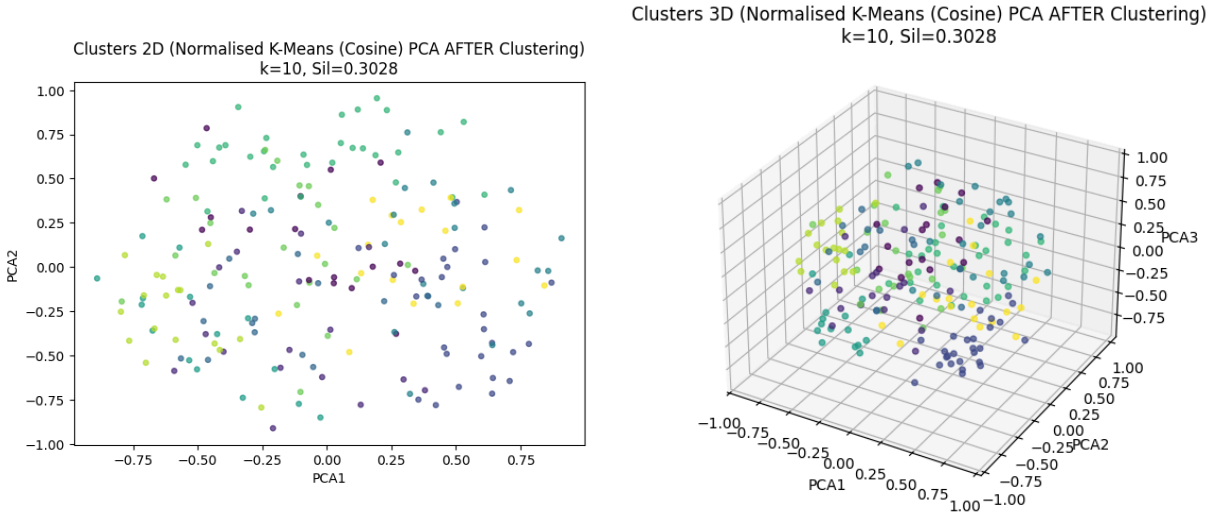


Figure 14: 2D and 3D Scatter Plot for PCA After Clustering (Cosine Distance)

2.2. K-Means++

2.2.1. Mechanism

K-Means++ is an enhanced version of the standard K-Means algorithm that uses a more strategic method for initialising centroids. Instead of choosing centroids randomly, it employs a probabilistic approach to pick centroids that are spread farther apart. This reduces the likelihood of poor clustering caused by unfavorable initial centroid placement. Aside from this improved initialisation process, the rest of the algorithm operates the same way as standard K-Means.

2.2.2. Methodology

1. **Initialisation**: Select the first centroid from the dataset randomly. For subsequent K-1 centroids, each centroid is selected with probability proportional to the squared euclidean distance from the nearest existing centroid. The data point with the highest squared distance will be chosen as the next centroid.
2. **Assignment**:
    a. **Using Euclidean Distance:** Each point is assigned to the nearest cluster based on Euclidean distance.
    b. **Using Cosine Distance:** Data points are assigned to the nearest centroid based on the smallest cosine distance, which measures the angular difference between vectors.
3. **Centroid Update**: Compute the new centroid for each cluster based on the mean of the data points assigned to the cluster.
    a. **Using Euclidean Distance**: The new centroid is calculated based on the geometric mean of the data points for each cluster.
    b. **Using Cosine Distance**: The new centroid is calculated based on the vector that minimises the cosine distance from the points within the cluster.
4. **Repeat**: The algorithm repeats between the assignment and update step until convergence is reached, i.e. the centroids no longer change significantly or a maximum number of iterations is reached.

2.2.3. Training

2.2.3.1. K-Means++ Initiation in Diabetes Prediction Dataset

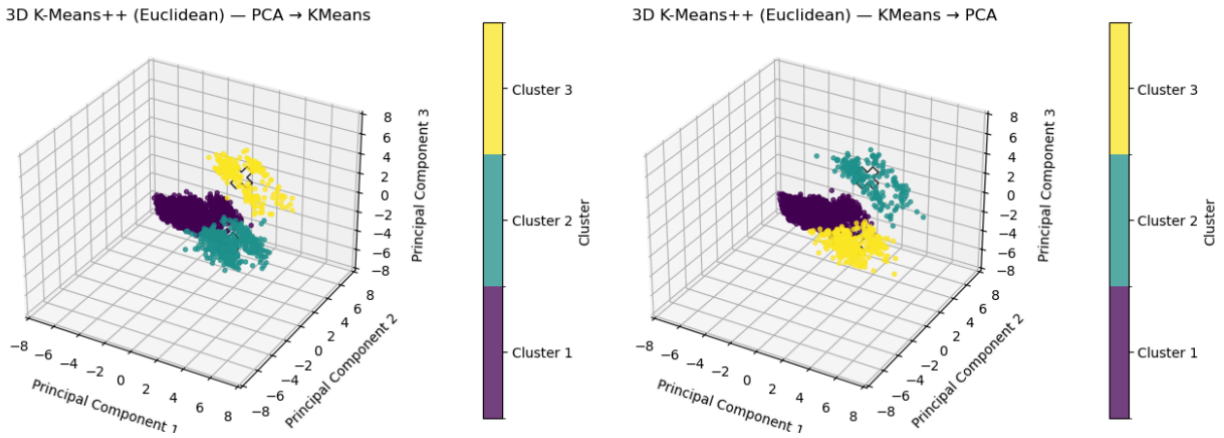
The dataset described in Section 2.1.3.1 includes several physical attributes such as neck circumference, height, weight. To perform clustering, the K-Means++ algorithm was used to run the simulation. Unlike the standard K-Means, K-Means++ enhances clustering performance by selecting initial centroids more strategically, thereby increasing the chances of achieving optimal clusters.

To assess how dimensionality reduction affects clustering outcomes, four experimental setups were considered:

- Reduce Dimension Before Clustering: Dimensionality reduction using PCA was performed before applying the K-Means++ algorithm. The clustering was performed using Euclidean distance and cosine distance.
- Reduce Dimension After Clustering: Apply K-Means++ directly on the original high-dimensional dataset. The clustering was performed using both Euclidean distance and cosine distance. After clustering, PCA will then be performed to visualise and interpret the clusters more effectively.

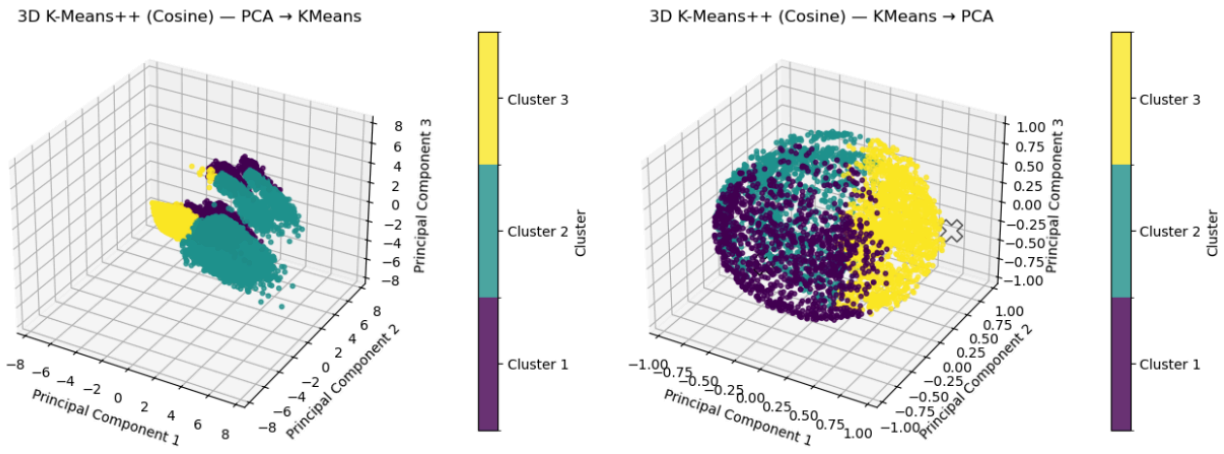
Before implementing the K-Means++ algorithm, the optimal number of clusters (K) was identified using both the Elbow Method and the Silhouette Score. In this study, the Silhouette Score was selected to determine K, as it offers a more accurate assessment of cluster quality by evaluating the cohesion within clusters and the separation between them.

Subsequently, the K-Means++ algorithm from the scikit-learn library was applied to perform clustering across the previously described scenarios.



(a) (b)

Figure 15: Illustration of Clusters on Diabetes Prediction dataset After K-Means++ Clustering Using Euclidean Distance (a) Reduce Dimension Before Clustering (b) Reduce Dimension After Clustering



(a) (b)

Figure 16: Illustration of Clusters on Diabetes Prediction dataset After K-Means++ Clustering Using Cosine Distance (a) Reduce Dimension Before Clustering (b) Reduce Dimension After Clustering

2.2.3.2. K-Means++ Initiation in Body Fat Prediction Dataset

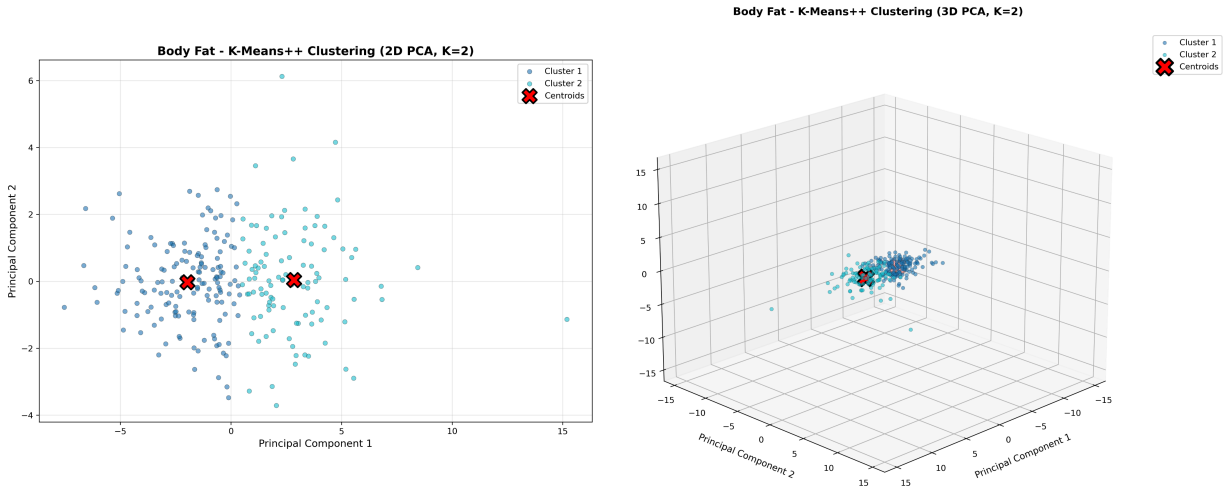


Figure 17: Illustration of K-Means++ Varying Principal Component Analysis for Body Fat Prediction

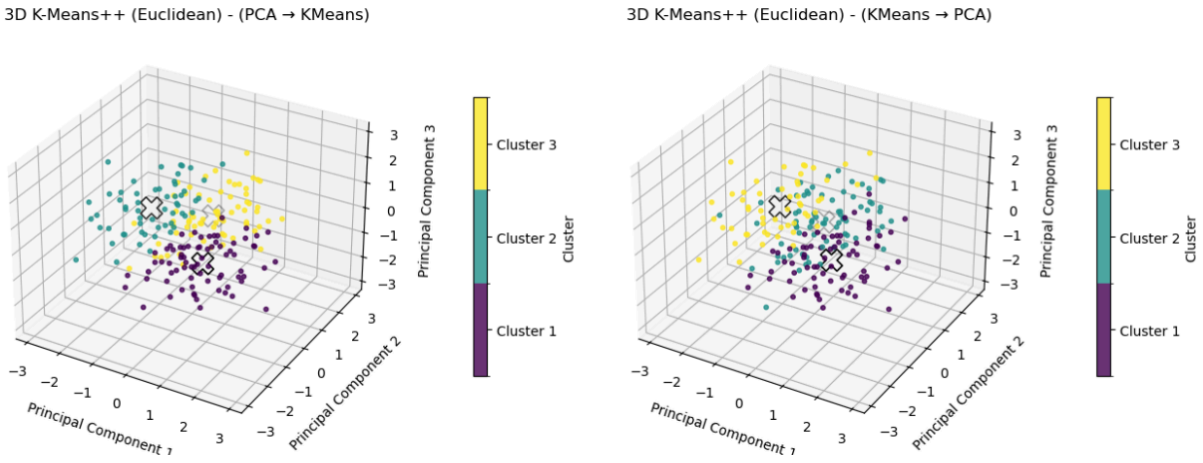
K-Means++ can be used to divide patients into various body fat percentages based on localised physical attributes (neck circumference, height, weight etc). Such segmentation has practical applications in:

- Tailoring appropriate diet measures to patients of certain body fat percentage groups

- Determining the need for surgical intervention such as Gastric Bypass to reduce stomach size or Insulin reduction to prevent the body from storing too much fats.

2.2.3.3. K-Means++ Initiation in World Bank Dataset

Similarly, K-Means++ algorithm is run on the World Bank dataset using Euclidean distance and cosine distance, for when dimension on the dataset is reduced before clustering and one for clustering is done and then PCA is applied.



(a) (b)

Figure 18: Illustration of Clusters on World Bank Dataset After K-Means++ Clustering Using Euclidean Distance (a) Reduce Dimension Before Clustering (b) Reduce Dimension After Clustering

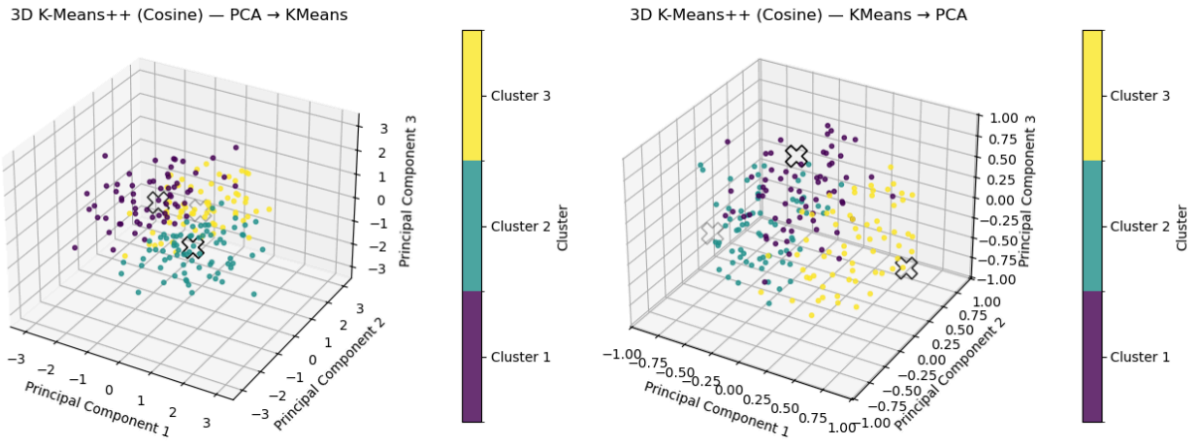


Figure 19: Illustration of Clusters on World Bank dataset After K-Means++ Clustering Using Cosine Distance (a) Reduce Dimension Before Clustering (b) Reduce Dimension After Clustering

2.3. Agglomerative Hierarchical Clustering

2.3.1. Mechanism

Agglomerative Hierarchical Clustering (AHC) is a widely adopted unsupervised learning technique that builds a cluster hierarchy using a bottom-up approach. It is especially useful when the underlying structure of the data is unknown, as it allows for clear visualisation of relationships between data points. The process begins by considering each data point as its own cluster, then progressively merges clusters according to a chosen distance metric until only one cluster remains or a predefined stopping condition is satisfied.

2.3.2. Methodology

1. **Initialisation**: Each data point is considered a cluster of its own. Thus, for n data points, there are n clusters.
2. **Distance Calculation**: A distance metric is used to compute the pairwise distances between clusters. The most commonly used distance metrics include:
    a. **Euclidean Distance**: Measures the straight-line geometric distance between two points.
    b. **Cosine Similarity**: Measures the cosine of the angle between two vectors.
    c. **Manhattan Distance**: Measures the distance between two points along axes that are perpendicular to each other.
3. **Linkage Condition**: Once the distances are calculated, the clustering process involves determining which clusters to merge based on a linkage criterion. The main linkage methods include:
    a. **Single Linkage**: The minimum distance between the closest points in the clusters.
    b. **Complete Linkage**: The maximum distance between the farthest points of two clusters.
    c. **Average Linkage**: The average distance between all pairs of points in the two clusters.
    d. **Centroid Method**: Combining clusters based on the minimum distance of centroids of the two clusters
    e. **Ward's Method**: A criterion that minimises the increase in variance when two clusters are merged.

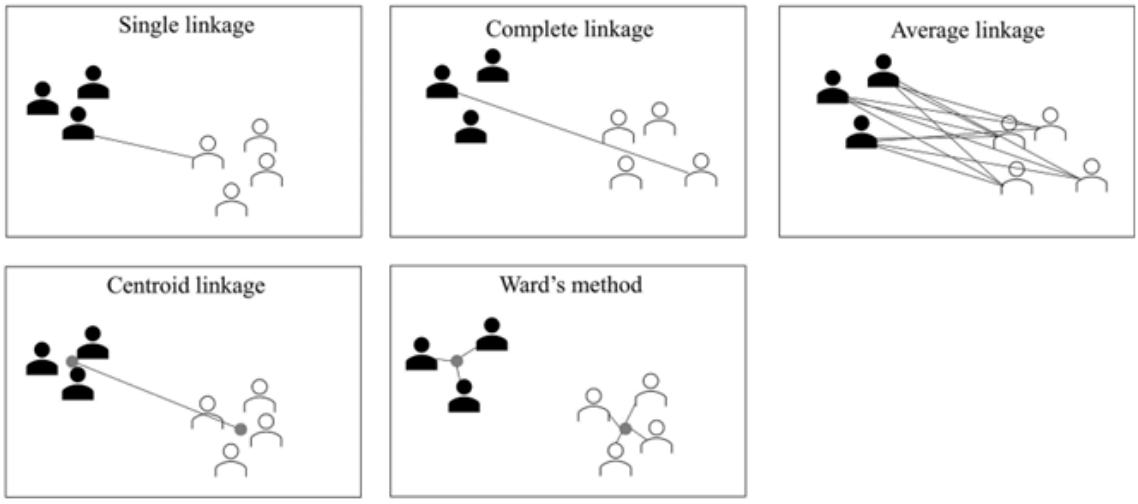


Figure 20: Illustration of Linkage Criteria (Gere, 2023)

4. **Merging**: The two clusters that are closest to each other based on the selected linkage criterion are merged to form a single cluster. After each merge, the distance matrix is updated to reflect the new set of clusters.
5. **Repeat**: Iterate the process of calculating distances and merging clusters until all data points are grouped into a single cluster or a predefined number of clusters is reached. The entire process can be visualised using a dendrogram to visualise the clustering process, a tree-like diagram to visualise the clustering process.
6. **Dendrogram Analysis**: The outcome of AHC is a hierarchical tree structure that displays clusters at various levels of detail. By slicing the dendrogram at a specific height, users can decide on the most suitable number of clusters. This flexibility makes AHC useful for instances where the ideal number of clusters is not predetermined.

2.3.3. Training

In this section, we trained different AHC models to find the optimal number of clusters that would give us the best clustering result to conduct experimental analysis on the different datasets. The following procedure is used to identify the optimal number of clusters for each dataset:

1. Visualisation of data points before AHC to get a rough idea about the data points
2. Evaluate AHC performance against number of clusters using the score metrics: Silhouette score, Davies-Bouldin score, and Calinski-Harabasz score
    a. A range of 2 to 50 has been used to determine the optimal number of clusters. We kept 50 as a maximum as a very large number of clusters can result in inconclusive results.

b. High Silhouette score (Angelou, 2024), low Davies-Bouldin score (Geekforgeeks, 2025), and high Calinski-Harabasz score (Geekforgeeks, 2025) suggests that the clusters are well-defined and compact.

3. Plot 2D PCA Scatter plot to visualise clustering results (if needed).

2.3.3.1. AHC Initiation in Diabetes Prediction Dataset

AHC is used for the Diabetes Prediction Dataset to investigate if certain health metrics (e.g. Gender, Age, Hypertension, BMI, etc.) of different patients can be associated with one another to determine whether they have diabetes or not. This is beneficial in helping other patients with similar health metrics determine if they are at risk of diabetes.

In the training of the AHC model here, a sample size of 10000 was taken as AHC does not scale well with this dataset due to its sheer size (100000 rows) (Frees, 2023). This sample dataset of 10000 rows will also be used for the experimental analysis later on.

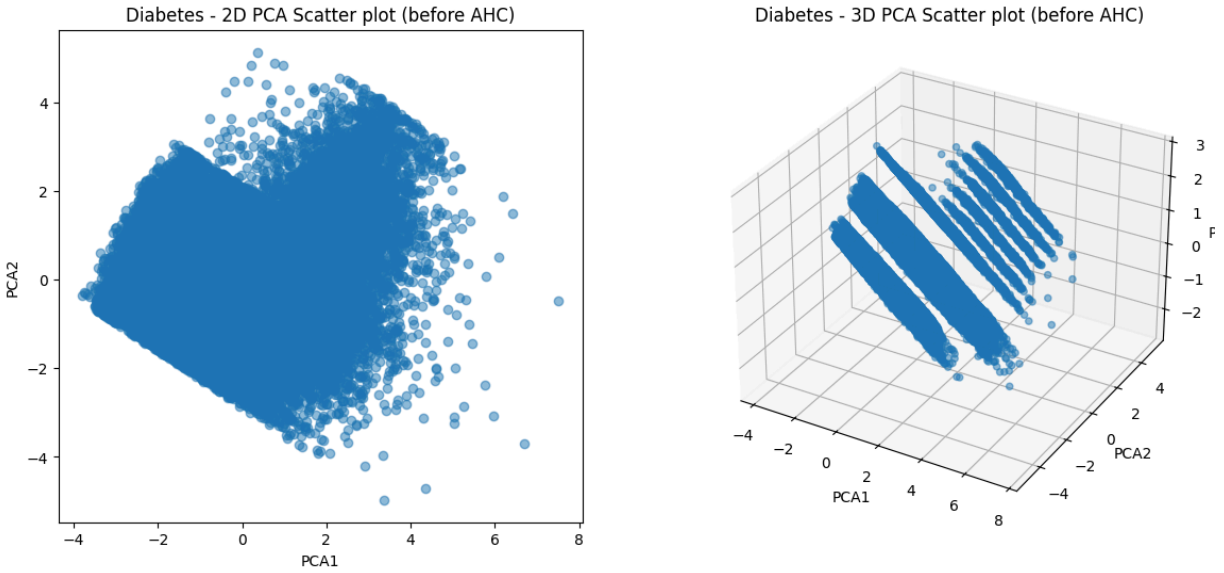


Figure 21: 2D and 3D PCA Scatter Plot to Visualise Data Points before AHC

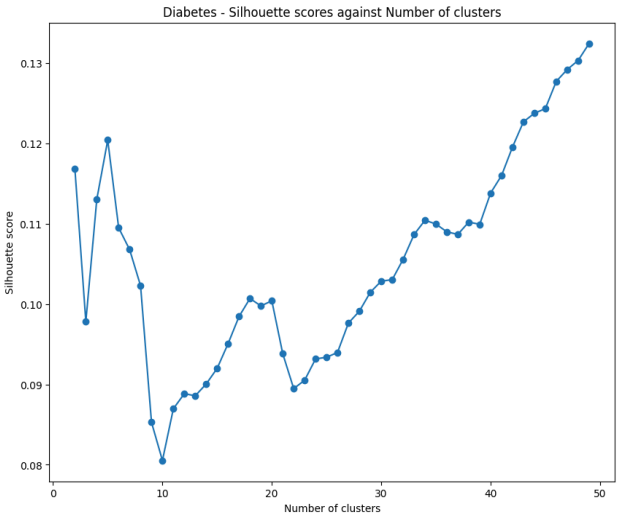


Figure 22(a): Line Graph to Visualise Silhouette Score against Number of clusters

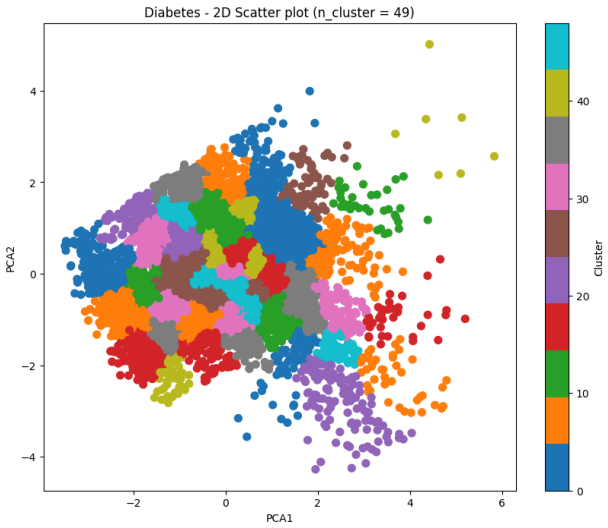


Figure 22(b): 2D Scatter Plot to Visualise the Data Points after AHC using Optimal Number of Clusters based on Silhouette Score

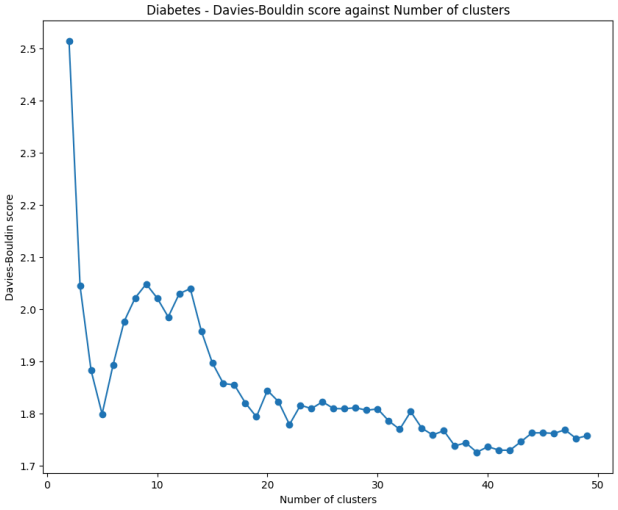


Figure 23(a): Line Graph to Visualise Davies-Bouldin score against Number of Clusters

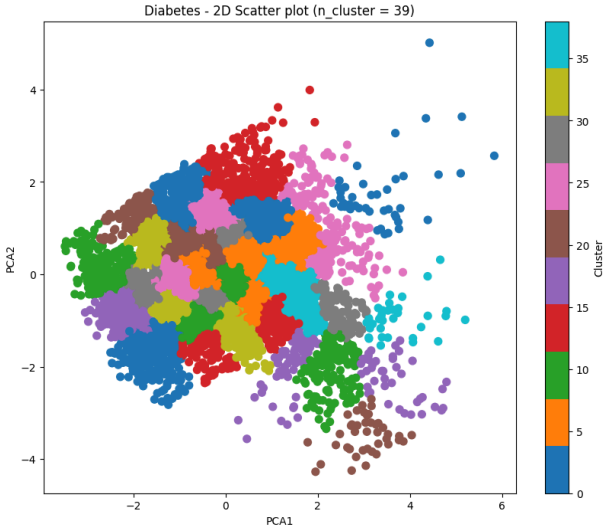


Figure 23(b): 2D Scatter Plot to Visualise Data Points after AHC using the Optimal Number of Clusters based on Davies-Boudin Score

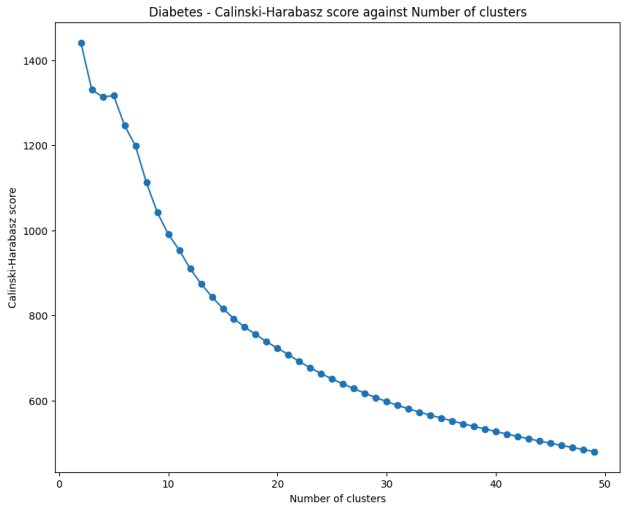


Figure 24(a): Line Graph to Visualise Calinski-Harabasz Score against Number of Clusters

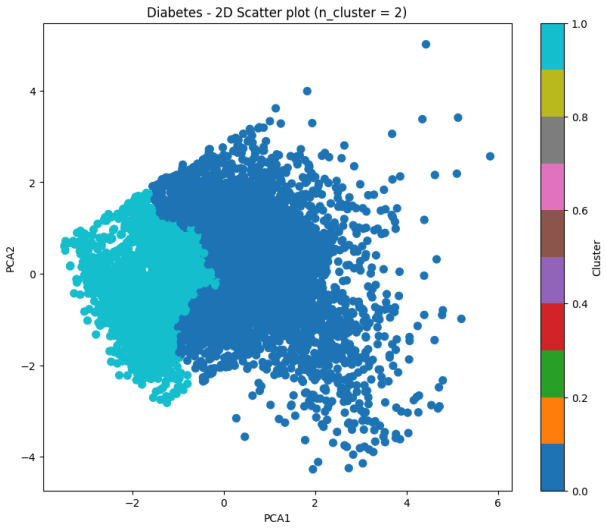


Figure 24(b): 2D Scatter Plot to Visualise Data Points after AHC with Optimal Number of Clusters based on Calinski-Harabasz Score

```
Highest Silhouette score: 0.132, Optimal number of clusters: 49
Lowest Davies-Bouldin score: 1.725, Optimal number of clusters: 39
Highest Calinski-Harbasz score: 1441.207, Optimal number of clusters: 2
```

Figure 25: Summary of Optimal Scores and their Corresponding Number of Clusters

As seen from Figure 25, the optimal scores of the different score metrics result in different optimal numbers of clusters. Hence, further investigation into cluster separation is conducted using 2D Scatter plots to visualise the data points after AHC.

Figure 22(b) and 23(b) shows data points being separated into numerous clusters. This large number of clusters may be a result of clustering the data points according to very specific characteristics, however, this may not be meaningful for the experimental analysis in our context (diabetic vs non-diabetic). Therefore, we will proceed with the optimal number of clusters being 2, as seen in Figure 24(b). Based on intuition and examining the original Diabetes Prediction CSV file, this is logical as the patient data points are separated into 2 main groups – diabetic and non-diabetic.

The presence of outliers, as seen in Figure 24(b), may be due to biological variations in the patients.

2.3.3.2. AHC Initiation in Body Fat Prediction Dataset

AHC is used for the Body Fat Prediction Dataset to explore whether specific ranges of body measurements (e.g. Chest, Abdomen, Hip, etc.) are associated with different body fat percentages, and to determine if individuals with similar body measurements can be grouped together based on their overall body composition. This can be beneficial to discover common patterns of body fat percentage in relation to certain body compositions, which can be beneficial in preventing chronic health conditions and increasing overall wellness (Borst & Gregory, 2025).

Figure 26: 2D and 3D Scatter Plot to Visualise Data Points before AHC

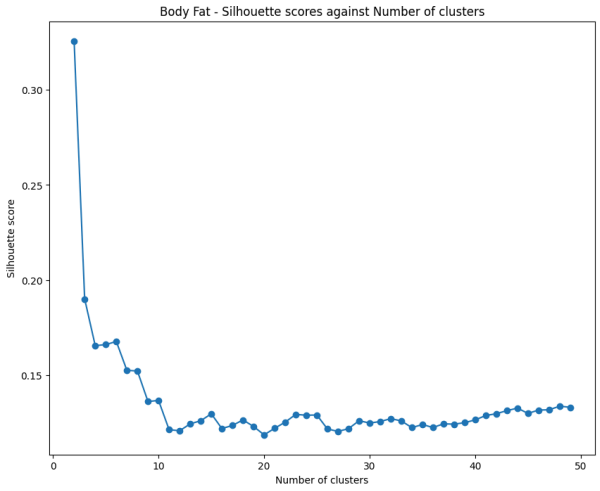


Figure 27: Line Graph to Visualise Silhouette Score against Number of Clusters

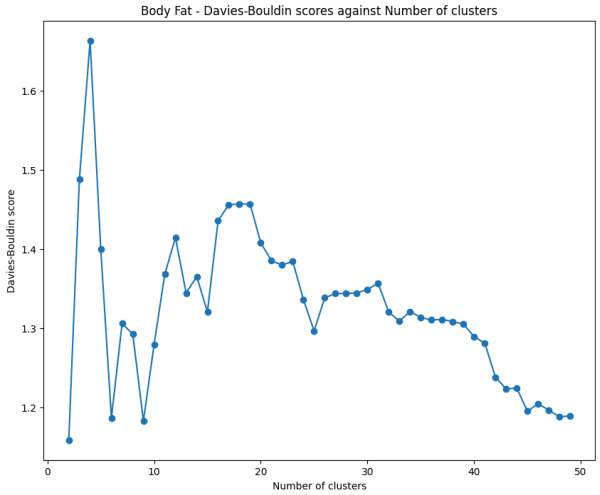


Figure 28: Line Graph to Visualise the Davies-Bouldin Score against Number of Clusters

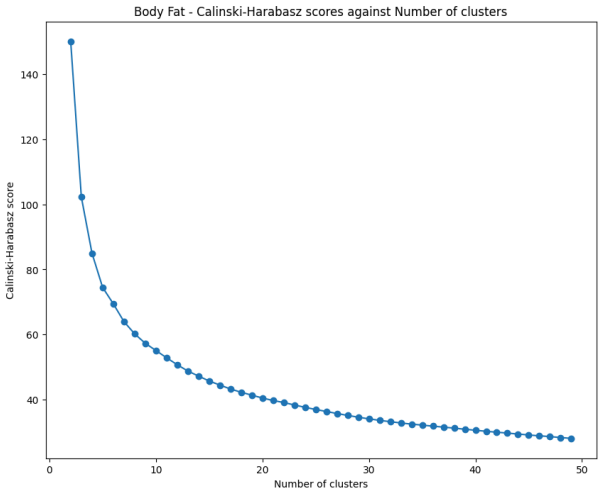


Figure 29: Line Graph to Visualise Calinski-Harabasz Score against Number of Clusters

```
Highest silhouette score: 0.326, Optimal number of clusters: 2
Lowest Davies-Bouldin score: 1.158, Optimal number of clusters: 2
Highest Calinski-Harabasz: 150.019, Optimal number of clusters: 2
```

Figure 30: Summary of Optimal Scores and their Corresponding Number of Clusters

As seen in Figure 30, the optimal score for each score metric results in the same optimal number of clusters. Hence, we will take the optimal number of clusters for this dataset to be 2. This suggests that the individuals in this dataset are grouped by general body composition profiles, possibly high body fat and larger measurements, and low body fat and smaller measurements.

2.3.3.3. AHC Initiation in World Bank Dataset

In this section, AHC is used for the World Bank Dataset to understand global patterns in regards to economic, social, health and environmental factors (e.g. GDP, Life Expectancy, CO2 emission, etc.). Insights drawn from grouping countries based on similarities of these factors can guide policy making and development strategies.

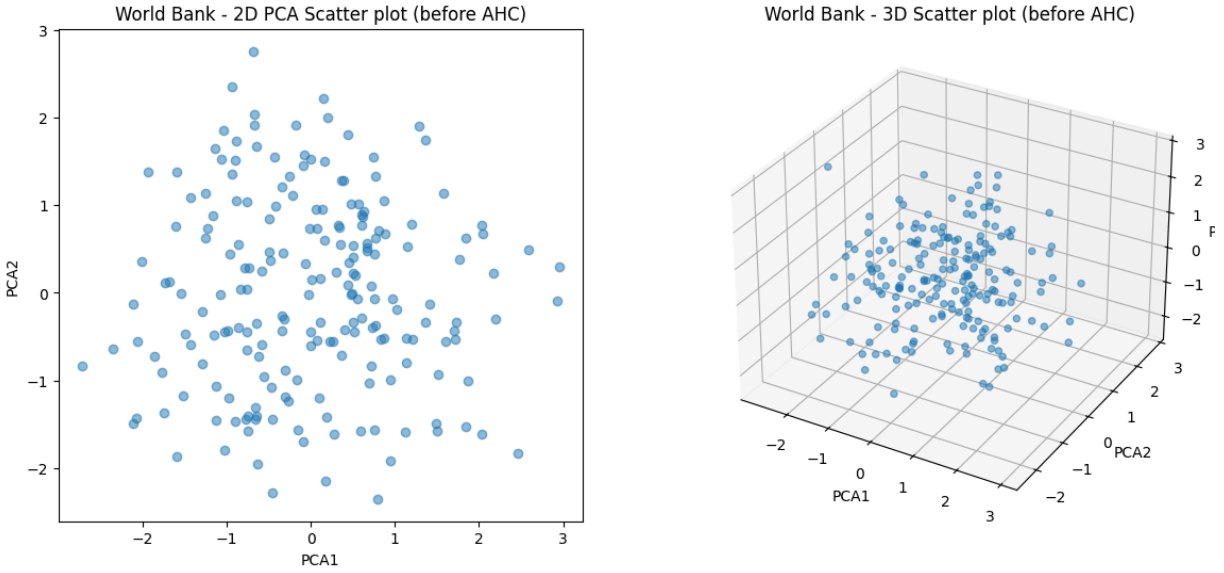


Figure 31: 2D and 3D Scatter Plot to Visualise Data Point before AHC

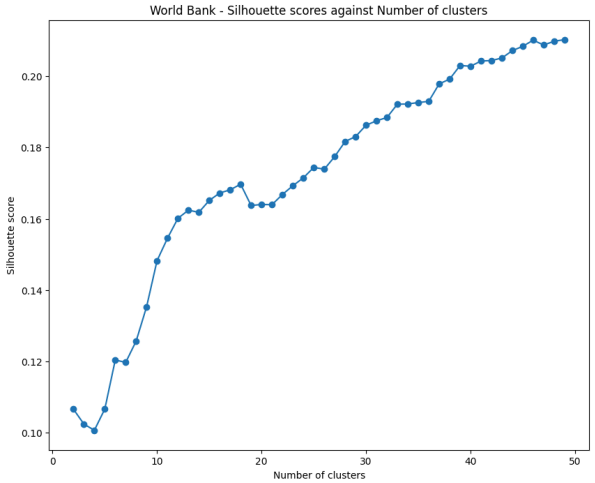


Figure 32(a): Line Graph to Visualise Silhouette Score against Number of Clusters

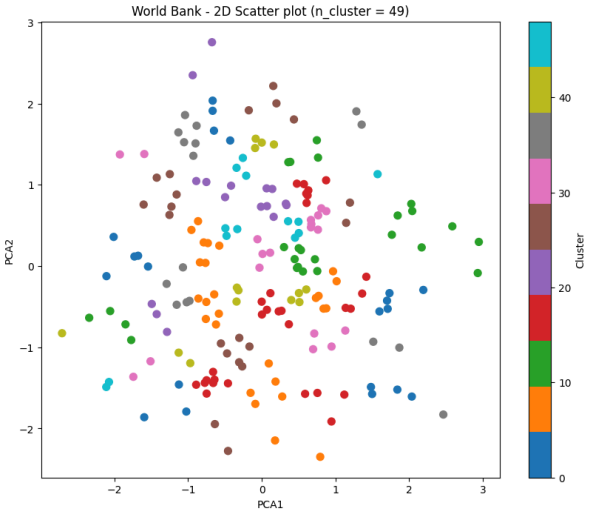


Figure 32(b): 2D Scatter Plot to Visualise Data Points after AHC with Optimal Number of Clusters based on Silhouette Score

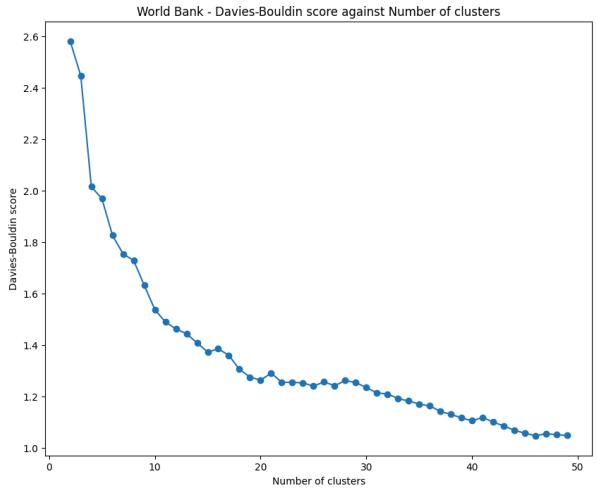


Figure 33(a): Line Graph to Visualise David-Bouldin Score against Number of Clusters

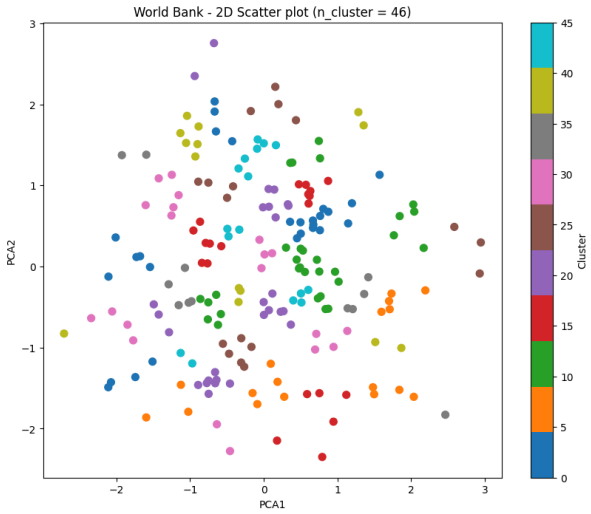


Figure 33(b): 2D Scatter Plot to Visualise Data Points after AHC using Optimal Number of Clusters based on Davies-Bouldin Score

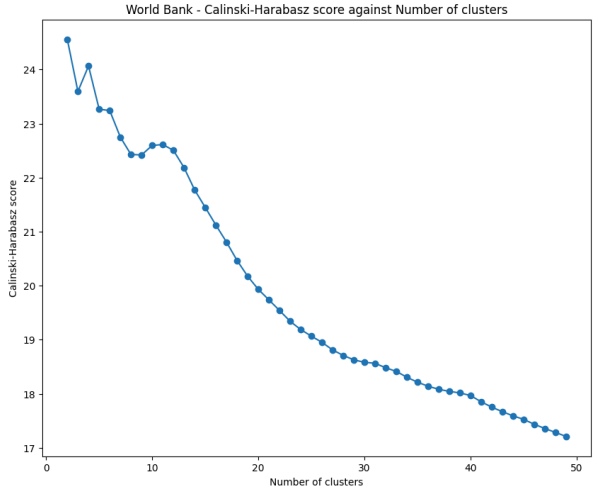


Figure 34(a): Line Graph to Visualise Calinski-Harabasz Score against Number of Clusters

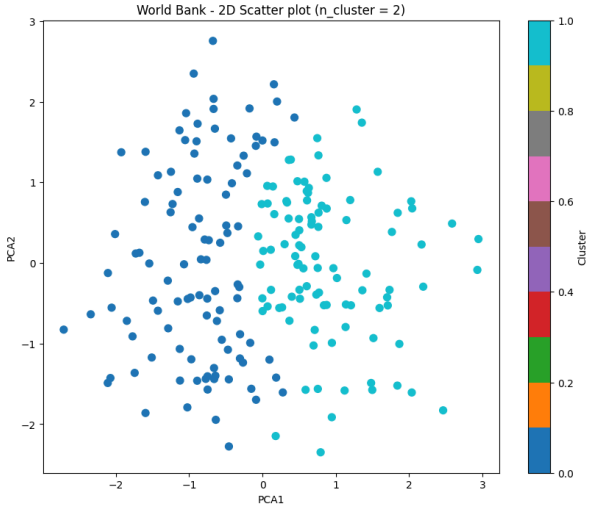


Figure 34(b): 2D Scatter Plot to Visualise Data Points after AHC using Optimal Number of Clusters based on Calinksi-Harabasz Score

```
Highest Silhouette score: 0.210, Optimal number of clusters: 49
Lowest Davies-Bouldin score: 1.048, Optimal number of clusters: 46
Highest Calinski-Harbasz score: 24.559, Optimal number of clusters: 2
```

Figure 35: Summary of Optimal Scores and their Corresponding Optimal Number of Clusters

The score metrics result in different numbers of optimal clusters, as seen in Figure 35, hence we visualised these clusters using 2D scatter plots to further evaluate the optimal number of clusters for this World Bank Dataset.

Figure 32(b) and 33(b) shows a large number of clusters where data points from different clusters overlap across the clusters. This is not beneficial for our experimental analysis later on and so, we will proceed with the optimal number of clusters being 2, as seen in Figure 34(b).

2.4. Gaussian Mixed Model

2.4.1. Mechanism

A Gaussian Mixture Model (GMM) is a probabilistic approach that models data as being produced by a combination of several Gaussian distributions with unknown parameters. It is widely applied in clustering to identify hidden subgroups within data. Each Gaussian component represents a cluster defined by its mean (center of Gaussian), covariance (spread), and mixing coefficient (weight or importance). Unlike K-Means, which strictly assigns each point to one cluster, GMM uses a probabilistic framework where each data point has a likelihood of belonging to multiple clusters simultaneously.

2.4.2. Methodology

1. **Initialise the parameters**: Randomly initialise the parameters of the Gaussian distributions. Let μ be a set containing the mean of each Gaussian distribution, where u_k is the k^{th} component of the Gaussian Distribution. $\sum$ be a set containing the covariance matrix of each Gaussian Distribution, where $\sum_k$ is the k^{th} covariance matrix component of the Gaussian Distribution. τ be a set containing the weight of each Gaussian Distribution, where τ_k is the k^{th} weight component of the Gaussian Distribution, also called the mixing parameter.
2. **Expectation-Maximisation**:
    a. Use the Expectation-Maximisation (EM) algorithm to iteratively update the parameters and cluster assignments.
    b. In the Expectation-step, estimate the likelihood of each data point belonging to each Gaussian cluster.
    c. In the Maximisation-step, amend the parameters of the Gaussian distributions based on the cluster likelihoods.

Expectation Maximisation Algorithm

Goal: $\max_{\theta} \log(\theta; X, Z) = \sum_{i=1}^{N} \sum_{k=1}^{K} \prod(z_i = k)[\log \tau_k + \log p_k(x_i; u_k, \sum_k)]$,

where $\theta := \{\tau, u, \sum\}$; N is the N^{th} component of Z, a latent variable which indicates which Gaussian component is 'used', X is a n dimensional random variable observed.

1. E-step: Compute the expectation of log of $P(X|Z)$

$$
\begin{aligned}
Q(\theta; X) &:= E_{Z|X}[\log L(\theta; X, Z)] \\
&= E_{Z|X}\left[\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{I}(z_i = k)\left[\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)\right]\right] \\
&= \sum_{i=1}^{N} E_{z_i|x_i}\left[\sum_{k=1}^{K} \mathbb{I}(z_i = k)\left[\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)\right]\right] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} P(z_i = k|x_i)\left[\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)\right]
\end{aligned}
$$

2. M-step: Solve by finding the optimal model parameters that maximise the expected log-likelihood function Q:

$$
\theta^* = \arg\max_{\theta} Q(\theta; X) = \sum_{i=1}^{N}\sum_{k=1}^{K} T_{k,i}\left[\log \tau_k + \log p_k(x_i; \mu_k, \Sigma_k)\right]
$$

Figure 36: Illustration of EM Algorithm used in GMM (Feng, 2020)

3. **Convergence**: Iterate the EM loop until the change in log-likelihood is small or reaches a predefined threshold.

2.4.3. Training

2.4.3.1. GMM Initiation in Diabetes Prediction Dataset

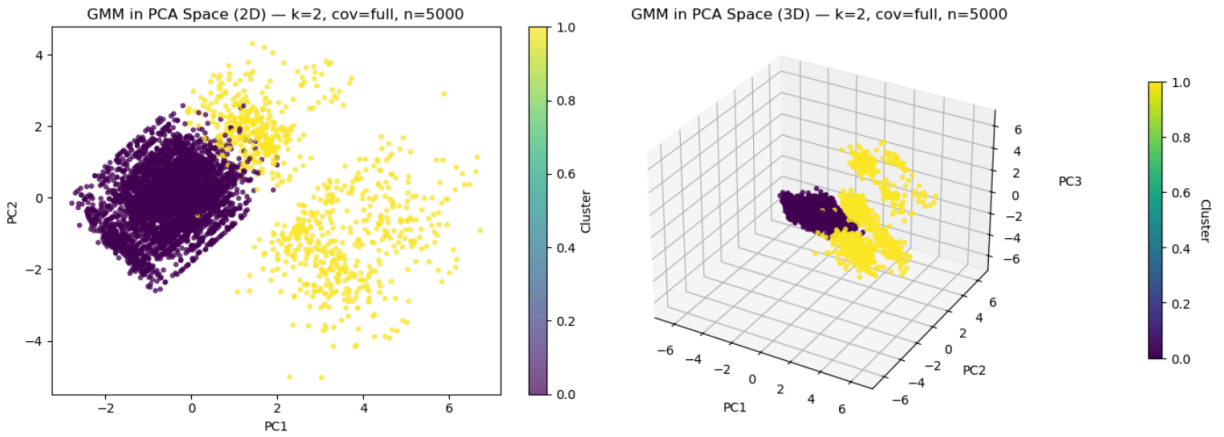


Figure 37: Illustration of GMM varying Principal Component Analysis for Diabetes Prediction Clustering

Using GMM on the Diabetes Prediction Dataset can help achieve valuable insights and benefits, particularly for comprehending the factors that may lead to increased risk of diabetes, as well as understanding the overlapping cluster boundaries. In this context, a person can be simultaneously at moderate and high risk of getting diabetes, at respective probabilities. GMM provides probabilities of data points belonging to each cluster, which gives us a better understanding of patients that might overlap between clusters. Moreover, GMM can help detect anomalies or outliers in diabetic outcomes by identifying lack of fit in these patients. These outliers may be due to:

- Mixed phenotypes: Some individuals show contradictory signals (e.g., low BMI but high glucose), which no single Gaussian captures well.
- Rare comorbidities: Patients with kidney disease, gestational diabetes, or steroid-induced glucose spikes differ from the typical Type 2 diabetes clusters.
- Socioeconomic or genetic subgroups: Different ethnic or regional risk patterns not explicitly modeled can create cross-cluster blends.

2.4.3.2. GMM Initiation in World Bank Dataset

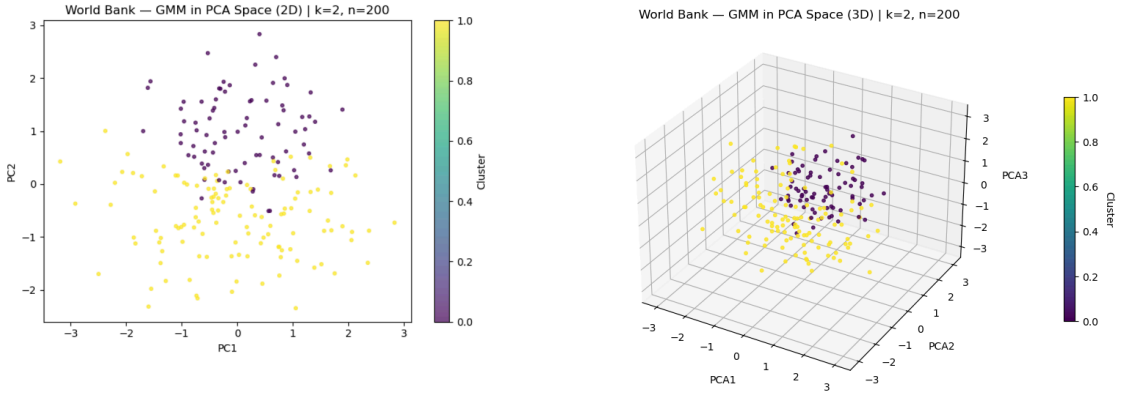


Figure 38: Illustration of GMM varying Principal Component Analysis for World Bank Prediction Clustering

In addition, GMM is carried out on the World Bank Dataset. Doing so can dissect the factors that lead to a longer life expectancy and the country's Gross Domestic Product (GDP), as well as understanding the overlapping cluster boundaries. In this context, a country can be considered both rich and poor, at respective probabilities. Outliers as expected, could be present in this case and in this case may be due to:

- Economic-of-scale: Some countries may have a high GDP per capita, but due to its small population, a country may have a low GDP but high life expectancy which the Gaussian Models would fail to capture
- Level of productivity: While a low unemployment rate suggests a buoyant economy, it also depends on what is the driving sectors of the country and potential prevalence of underemployment
- Intangible factors: Factors such as level of happiness, satisfaction are arbitrary figures that do contribute to a country’s life expectancy.

3. Experimental Analysis

3.1. K-Means

3.1.1. Diabetes Prediction Dataset

Clustering Scenario	Silhouette Score
K-Means (Euclidean) PCA Before Clustering	0.3651
K-Means (Euclidean) PCA After Clustering	0.2895
K-Means (Cosine) PCA Before Clustering	0.7065
K-Means (Cosine) PCA After Clustering	0.4488

Figure 39: Silhouette Scores for Different Scenarios with K-Means (Cosine) PCA Before Clustering having the highest Silhouette Score

Based on the Silhouette Score, we observed that the K-Means (Cosine) PCA Before Clustering achieved the highest Silhouette Score of 0.7065 among the 4 different scenarios. This indicates that applying PCA before clustering enhances the separability of data points by reducing noise and retaining the most significant variance in the dataset.

When comparing the effect of dimensionality reduction timing (before/after clustering), we observe that both Euclidean and cosine distance metrics produced higher Silhouette Scores when PCA was applied before clustering, as opposed to after. This suggests that PCA simplifies the clustering task when high-dimensional data makes distance measurements less meaningful, thus weakening cluster boundaries.

Between the two distance metrics, cosine distance significantly outperformed Euclidean distance in both PCA-before and PCA-after setups. This highlights cosine distance's ability in capturing angular relationships between normalised feature vectors, allowing it to recognise patterns in direction instead of magnitude. Such an approach is particularly effective in datasets where attributes differ in scale or range.

By combining PCA before clustering and cosine distance, it produces the most distinct and interpretable clusters. This combination effectively balances dimensionality reduction with a robust similarity measure, resulting in well-separated and cohesive cluster structures.

3.1.2. Body Fat Prediction Dataset

Clustering Scenario	Silhouette Score
K-Means (Euclidean) PCA Before Clustering	0.4173
K-Means (Euclidean) PCA After Clustering	0.3300
K-Means (Cosine) PCA Before Clustering	0.6627
K-Means (Cosine) PCA After Clustering	0.5313

Figure 40: Silhouette Scores for Different Scenarios with K-Means (Cosine) PCA Before Clustering having the highest Silhouette Score

Based on the Silhouette Score, K-Means (Cosine) PCA Before Clustering achieved the highest score of 0.6627. The results are consistent with those of the Diabetes Prediction Dataset and the same reasoning applies, indicating that this combination provides clearer and more well-defined clusters.

3.1.3. World Bank Dataset

Clustering Scenario	Silhouette Score
K-Means (Euclidean) PCA Before Clustering	0.2653
K-Means (Euclidean) PCA After Clustering	0.1758
K-Means (Cosine) PCA Before Clustering	0.4962
K-Means (Cosine) PCA After Clustering	0.3028

Figure 41: Silhouette Scores for Different Scenarios with K-Means (Cosine) PCA Before Clustering having the highest Silhouette Score

Based on the Silhouette Score, K-Means (Cosine) PCA Before Clustering achieved the highest score of 0.4962. The results are similar to the previous findings and the reasoning remains the same as before.

3.2. K-Means++

3.1.1 Diabetes Prediction Dataset

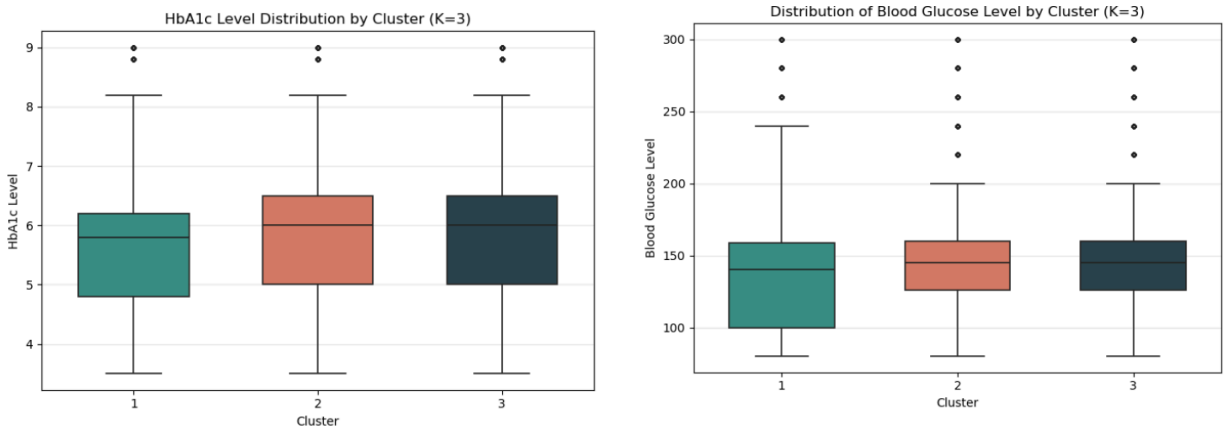
Approach	Score
K-Means++ (Reduce Dimension before Clustering)	0.599
K-Means++ (Reduce Dimension after Clustering)	0.601
Normalised K-Means++ (Reduce Dimension before Clustering)	0.574
Normalised K-Means++ (Reduce Dimension after Clustering)	0.353

Based on the Silhouette score, we observed that K-Means++ with Euclidean distance performed better when PCA was applied after clustering. This suggests that the clusters are spherical in shape, with all features contributing equally to distance and dimension reduction before clustering blurred the boundaries.

In the following section, we will examine the clusters formed by K-Means++ with Euclidean distance.

Figure 42: Distribution of the clusters on Diabetes Prediction Dataset

We observe that many data points belong to Cluster 1, this suggests the possibility of a large number of patients having similar features and K-Means++ assigned the majority of the datapoints to one cluster, leaving a small amount of anomalies into Clusters 2 and 3.



(a) (b)

Figure 43: Boxplot on (a) HbA1c and (b) Blood Glucose Level

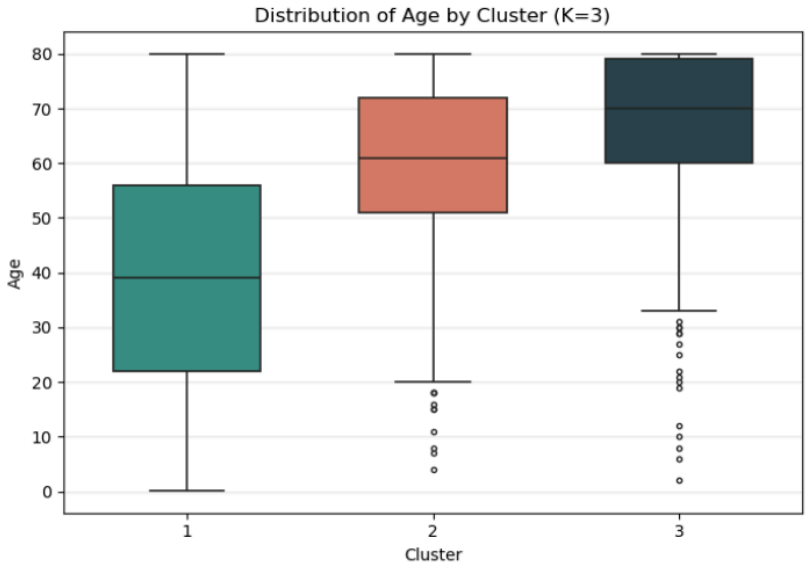


Figure 44: Box Plot on Patient's Age

From the above observations, the main observations of the clusters are as follows:

Cluster 1	• Lower HbA1c level compared to clusters 2 and 3 • Blood glucose levels have fewer outliers compared to clusters 2 and 3 • Patients are generally younger compared to both clusters 2 and 3

Cluster 2	● Higher HbA1c level compared to clusters 1 and similar levels compared to cluster 3 ● Blood glucose levels have more outliers compared to clusters 1 and similar compared to cluster 3 ● Patients are generally older compared to cluster 1
Cluster 3	● Higher HbA1c level compared to clusters 1 and similar levels compared to cluster 3 ● Blood glucose levels have more outliers compared to clusters 1 and similar compared to cluster 3 ● Patients are generally older compared to both clusters 1 and 2

3.2.2 Body Fat Prediction Dataset

```
Inertia (sum of squared distances), K=2: 2362.855933

Cluster Centers (original units):
```

	Density	BodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist
Cluster 1	1.064806	14.972483	44.295302	160.480872	69.672819	36.664430	95.696644	85.869128	95.765772	56.484564	37.252349	22.382550	30.558389	27.694631	17.774497
Cluster 2	1.042218	25.195146	45.737864	205.604854	70.837379	39.912621	108.241748	102.229126	105.892233	63.632039	40.526214	24.143689	34.754369	30.066019	18.888350

Figure 45: Output of Cluster Centers and Inertia

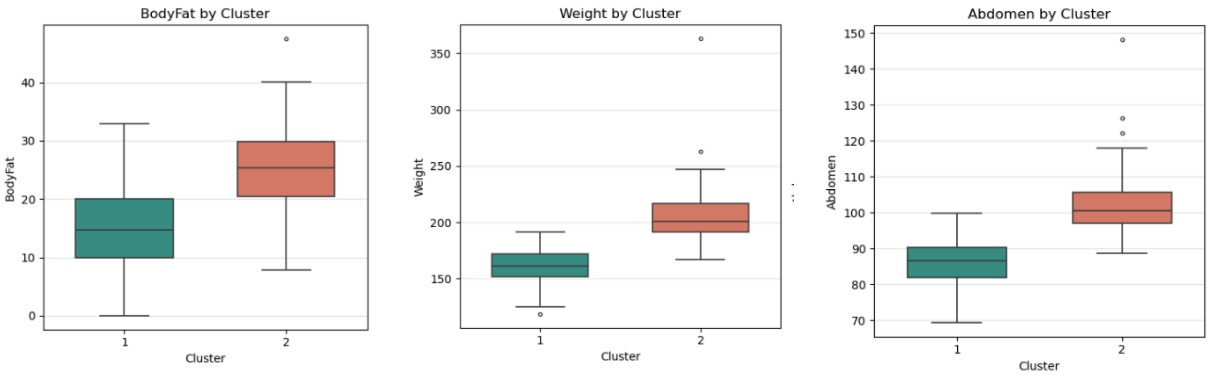
The computed inertia value, representing the compactness of the clusters, is 2362.856. A lower inertia typically implies that data points are tightly grouped around their respective centroids. However, it is essential to interpret this value in relation to the dataset's scale and the selected number of clusters (K). It is also worth noting that an effective clustering model achieves low inertia with a relatively small K, although inertia naturally decreases as K increases.

	Cluster 1	Cluster 2	abs_diff	pct_diff
Weight	160.48	205.60	45.12	28.12
Abdomen	85.87	102.23	16.36	19.05
Chest	95.70	108.24	12.55	13.11
BodyFat	14.97	25.20	10.22	68.28
Hip	95.77	105.89	10.13	10.57
Thigh	56.48	63.63	7.15	12.65
Biceps	30.56	34.75	4.20	13.73
Knee	37.25	40.53	3.27	8.79
Neck	36.66	39.91	3.25	8.86
Forearm	27.69	30.07	2.37	8.56
Ankle	22.38	24.14	1.76	7.87
Age	44.30	45.74	1.44	3.26

Figure 46: In-depth Analysis of Difference in Cluster Centres for Factors that Contribute to Body Fat Percentage in terms of Absolute and Percentage Difference

Factors that contribute to body fat percentage include: Weight, Abdomen, Chest, BodyFat, Hip, Thigh, Biceps, Knee, Neck, Forearm, Ankle and Age.

It has been identified that the top 3 differences in features include: Body Fat, Weight and Abdomen. This is intuitive as absolute body fat determines body fat percentage, and usually the fat is deposited in the abdomen area. We shall select these three factors and perform a box plot for better visualisation.



(a) (b) (c)

Figure 47: Visualisation of Features of Interest (a) Bodyfat (b) Weight (c) Abdomen

Cluster 1 **(Low Body Fat Percentage)**	• Lower body fat • Lower weight • Shorter abdomen circumference
Cluster 2 **(High Body Fat Percentage)**	• Higher body fat • Higher weight • Longer abdomen circumference

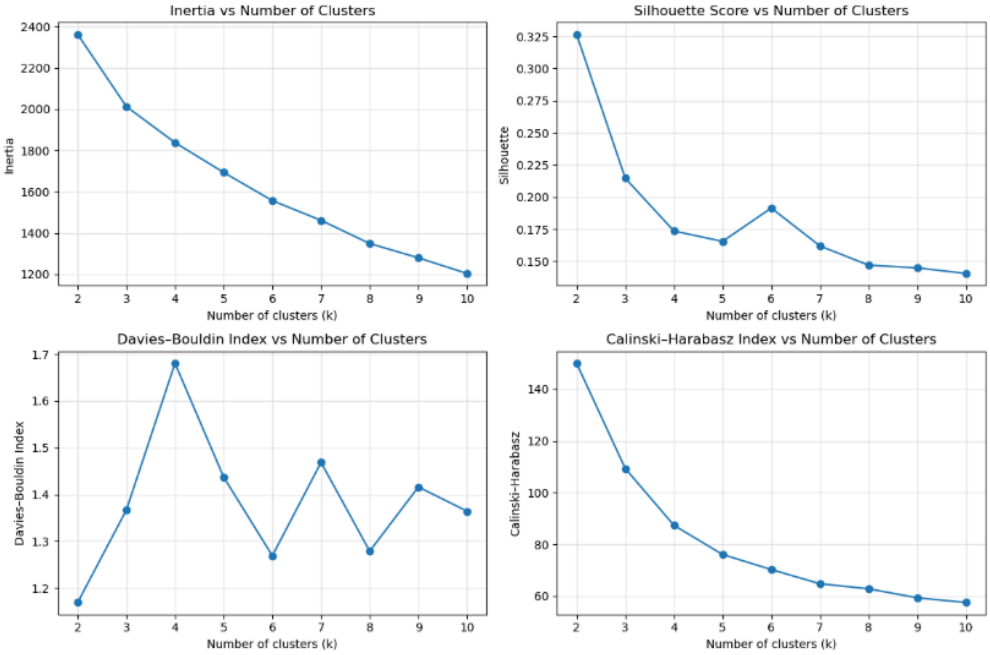


Figure 48: Analysis using Varying Metrics for K-Parameter Settings

Inertia vs K-parameter	Inertia decreases at a decreasing rate as K increases
Silhouette Score vs K-parameter	Silhouette Score shows downward trend when K increases
Davies-Bouldin Index vs K-parameter	Davies-Bouldin index fluctuates as K increases, but converging to ~1.4
Calinski-Harabasz Index vs K-Parameter	Calinski-Harabasz Index decreases at a decreasing rate when K increases
<u>Conclusion</u> We can conclude that for Body Fat classification, K-Parameter should be set at 2 given the	

trajectory, subjective for the various scenarios.

3.2.3 World Bank Dataset

Approach	Score
K-Means++ (Reduce Dimension before Clustering)	0.262
K-Means++ (Reduce Dimension after Clustering)	0.242
Normalised K-Means++ (Reduce Dimension before Clustering)	0.468
Normalised K-Means++ (Reduce Dimension after Clustering)	0.209

Based on the Silhouette score, we observed that K-Means++ with normalised distance performed better when PCA was applied before clustering. This might be due to the drawbacks of dimensionality mentioned in Section 3.1.1.

In the following section, we will examine the clusters formed by K-Means++ with normalised distance.

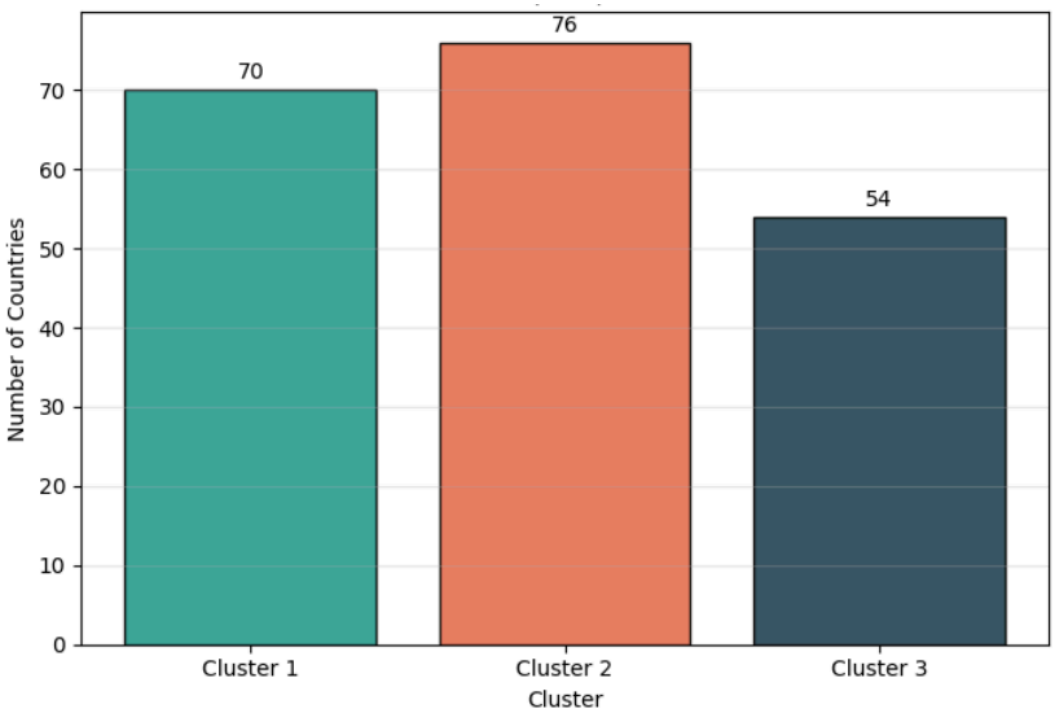


Figure 49: Distribution of the Clusters on World Bank Dataset

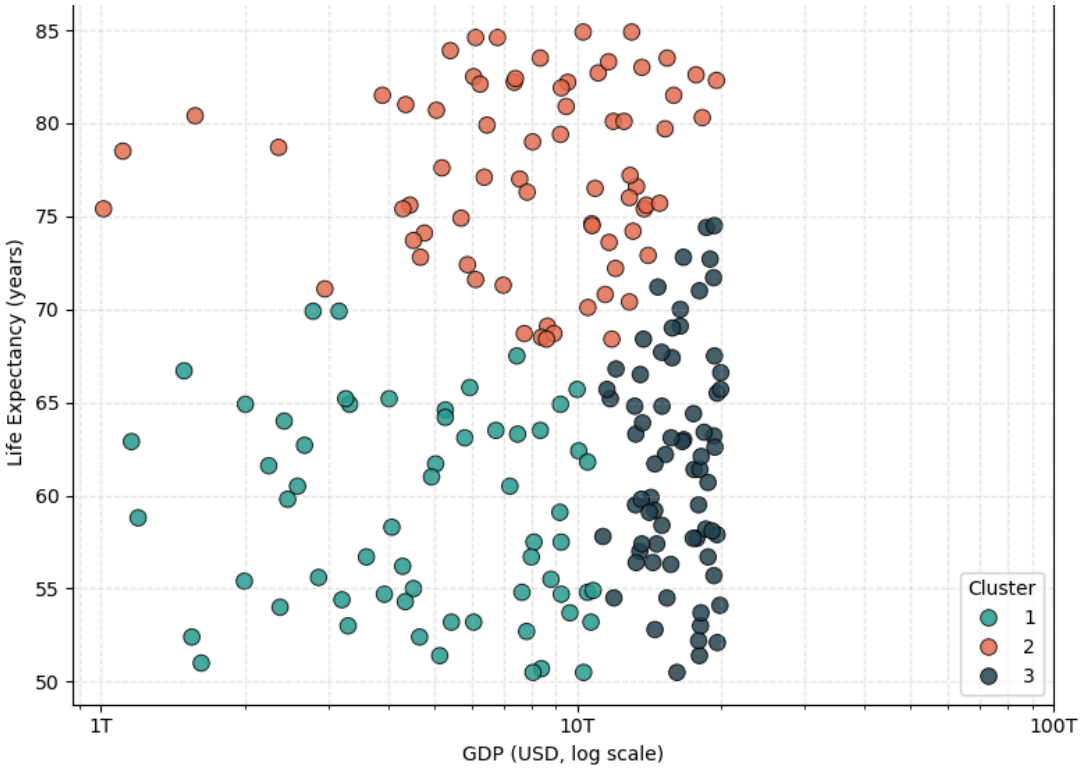
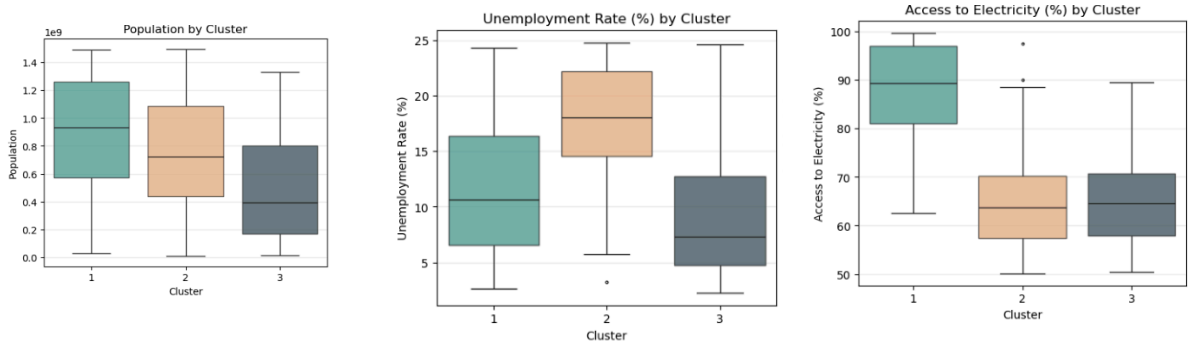


Figure 50: Scatterplot of Clusters based on Life Expectancy and GDP



(a) (b) (c)

Figure 51: Boxplot on (a) Population of Country, (b) Employment Rate and (c) Access Rate to Electricity

From the above observations, the main observations of the clusters are as follows:

Cluster 1	● Generally has lower life expectancy and GDP ● Median population is the highest among the 3 clusters ● Lower median unemployment rate lower than cluster 2, but higher than 3 ● Has the highest access rate to electricity
Cluster 2	● Generally has the highest life expectancy among the 3 clusters and GDP is generally higher than cluster 1 but lower than cluster 3 ● Median population is higher than cluster 3, and lower than cluster 1 ● Highest median unemployment rate among the 3 clusters ● Has the lower access rate to electricity compared to cluster 1, and similar access rate compared to cluster 3
Cluster 3	● Generally has similar life expectancy as cluster 1 and GDP is generally higher than both clusters 1 and 2 ● Median population is the lowest among all 3 clusters ● Lowest median unemployment rate among the 3 clusters ● Has the lower access rate to electricity compared to cluster 1, and similar access rate compared to cluster 2

3.3. Agglomerative Hierarchical Clustering

3.3.1. Diabetes Prediction Dataset

3.3.1.1. Comparing Linkage Criteria

In this section, we compare how different linkage methods affect the performance of AHC. This allows us to find the optimal linkage method to obtain the best AHC model.

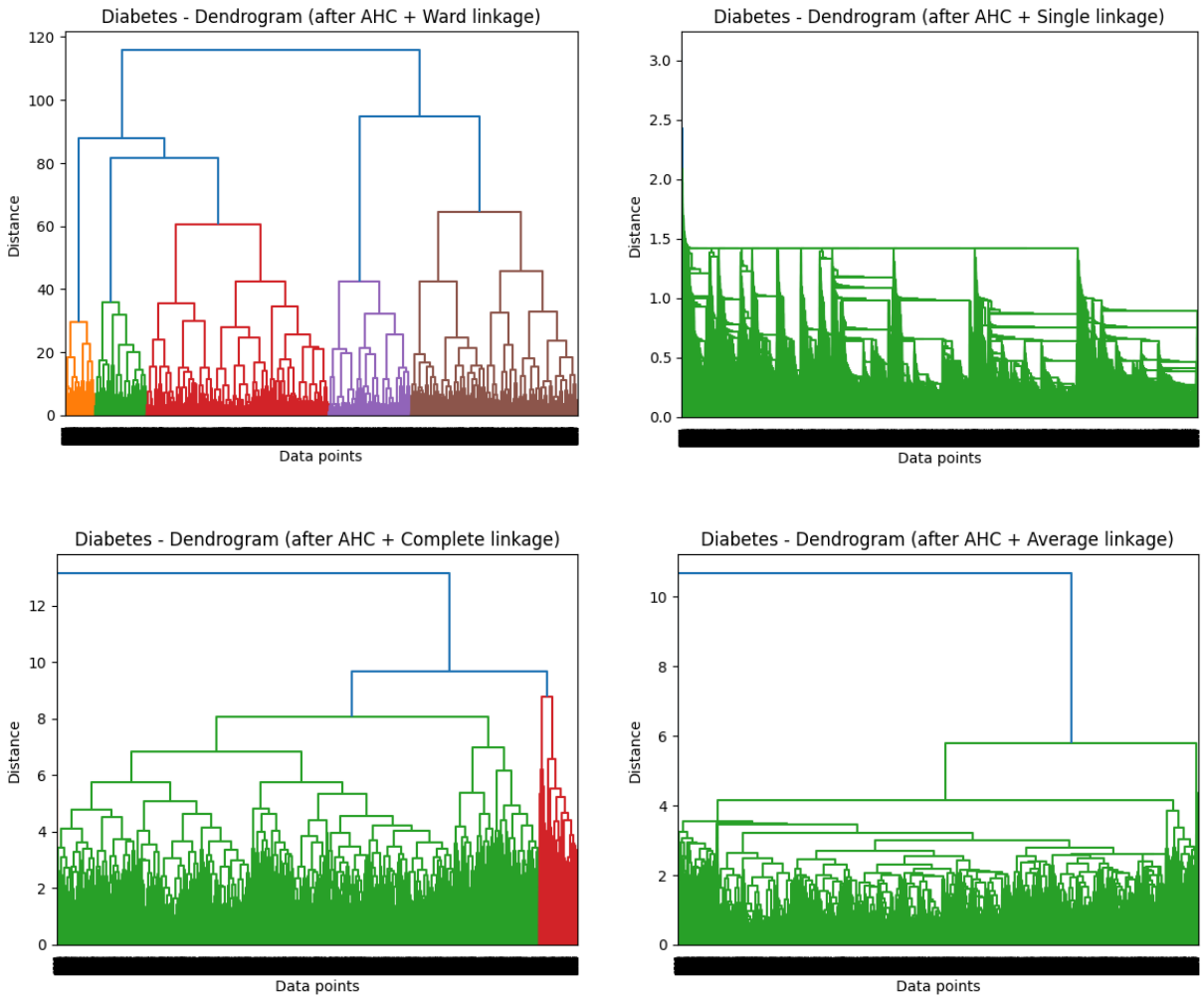


Figure 52: Dendrograms of AHC using Different Linkage Criteria

By comparing the dendrograms in Figure 52, we can visualise the hierarchy of clustering using the different linkage methods.

Ward's linkage clusters the data points using the total spread or variance increase when the clusters are combined. This results in compact and well-separated clusters (Geekforgeeks, 2025). Hence, the corresponding dendrogram in Figure 52 shows a more evenly-separated hierarchy clustering.

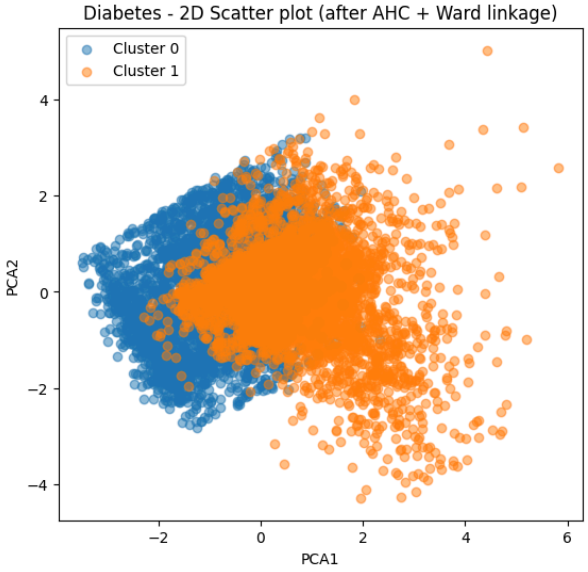
Single linkage clusters the data points using the minimum distance between 2 points. As this method is sensitive to outliers, it results in long, chain-like clusters (Geekforgeeks, 2025).

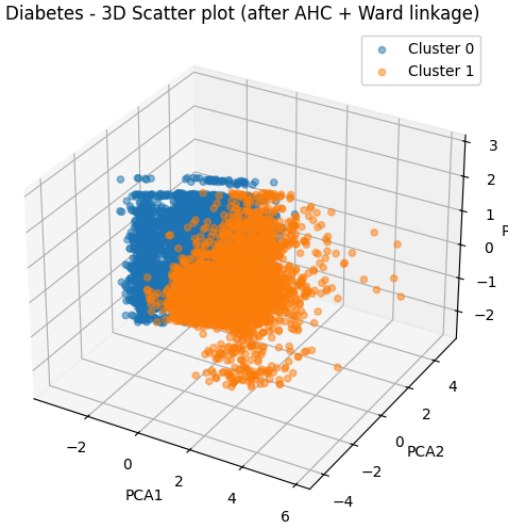
Complete linkage clusters using maximum distance between 2 points. This method is also sensitive to outliers and tries to make the clusters not too far apart, resulting in compact and spherical clusters (Geekforgeeks, 2025).

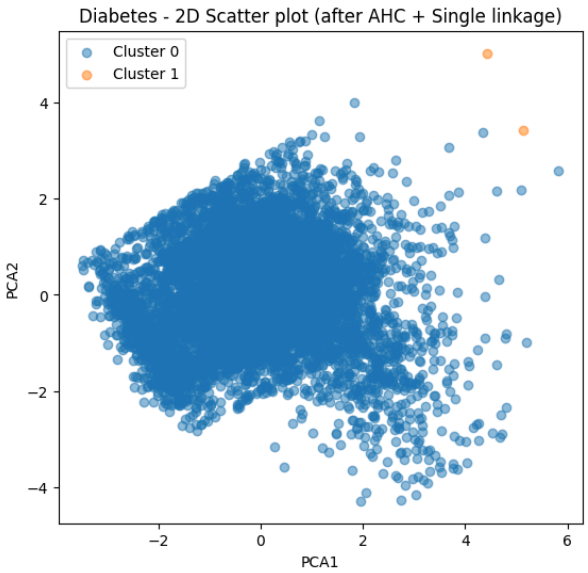
Average linkage clusters using the average distance between all pairs of points from 2 clusters. This method strikes a balance between single and complete linkage as it does not take the extreme distances (Geekforgeeks, 2025). Hence, clusters are moderately compact (Geekforgeeks, 2025).

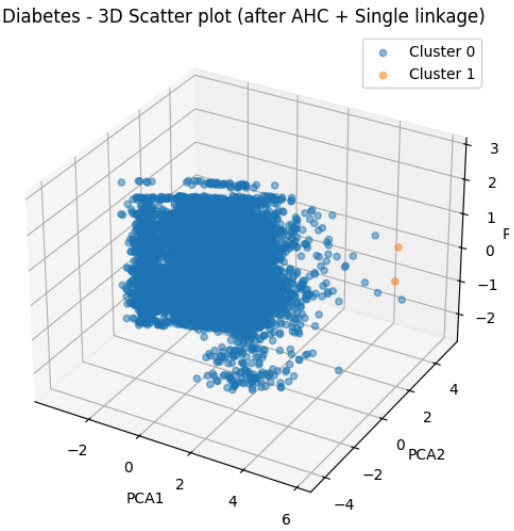
Single, complete, and average linkage results in skewed dendrograms as seen in Figure 52.

Comparing the 4 dendrograms, the optimal linkage method is Ward's linkage. The above observations are further corroborated by the following scatter plots:









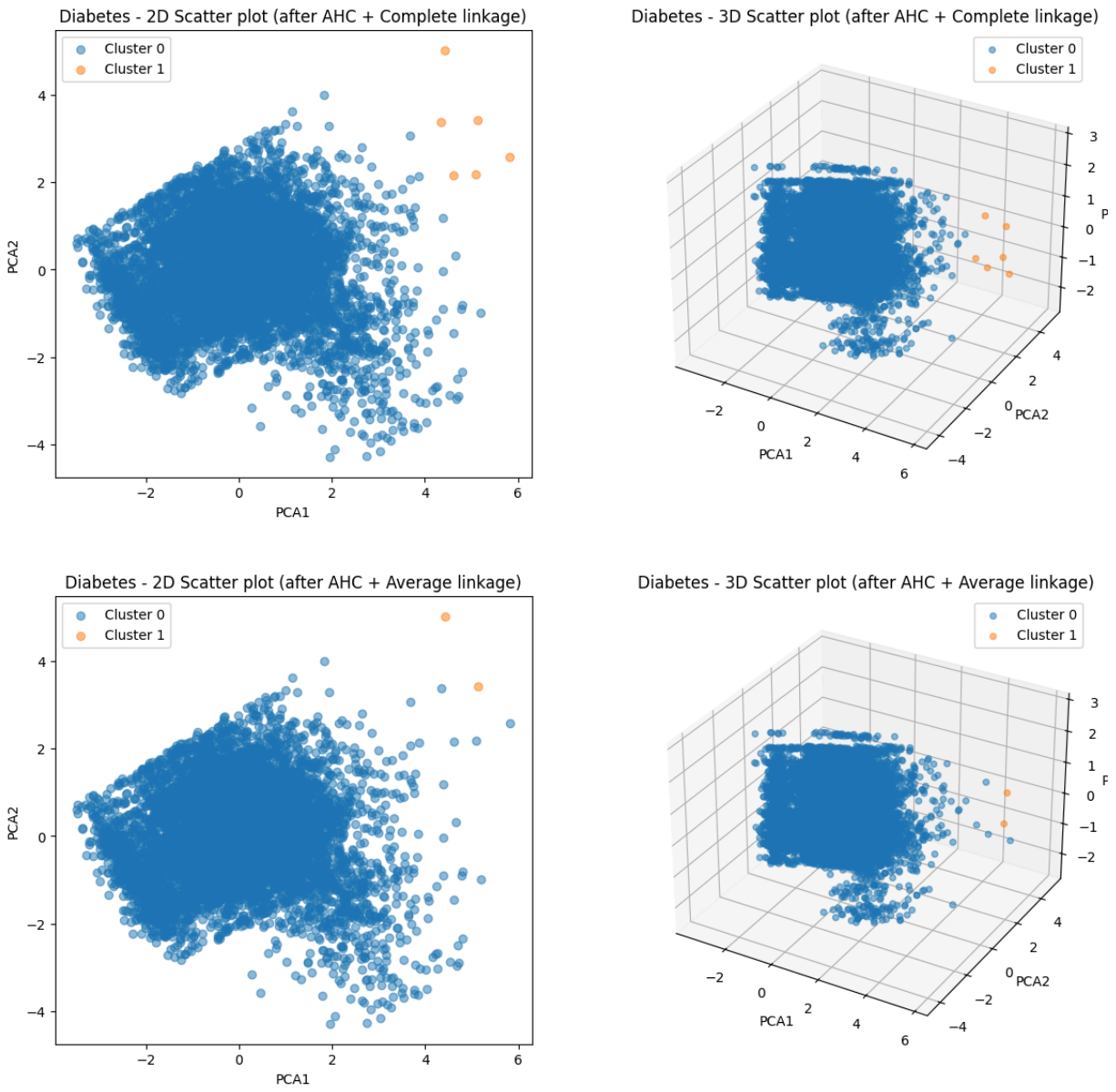


Figure 53: 2D and 3D Scatter Plots of Data Points using Different Linkage Criteria

As seen in Figure 53, Ward's linkage results in evenly-clustered data points. This is unlike that seen in the scatter plots using single, complete, and average linkage, where it seems like the 2 main clusters are the outliers and the majority of data points, which is undesirable in our analysis.

3.3.1.2. Comparing PCA Performance

In this section, we compare different numbers of principal components to determine the optimal dimensionality for reducing the Diabetes Prediction Dataset.

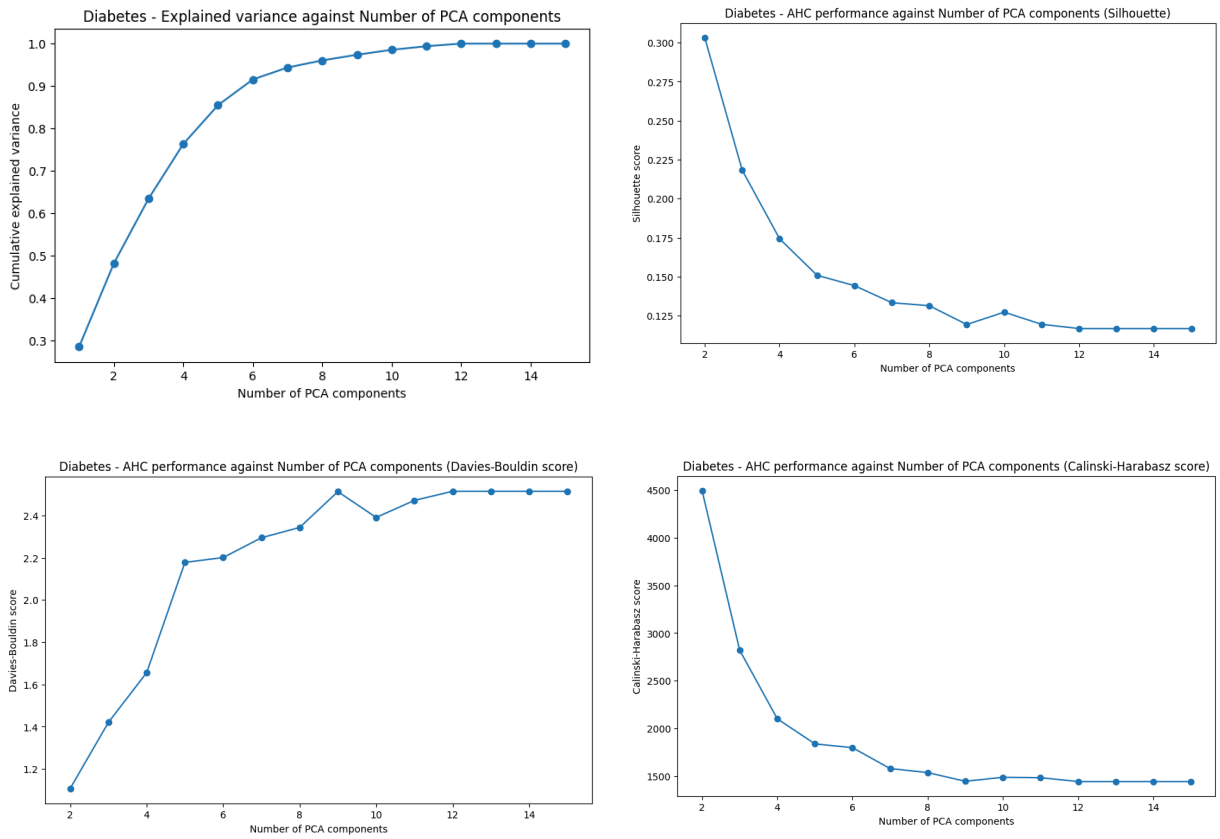


Figure 54: Line graphs to Visualise Cumulative Explained Variance, Silhouette sScore, Davies-Bouldin Score and Calinski-Harabasz Score against Number of Principal Components

```
Highest Silhouette score: 0.303, Optimal number of components: 2
Lowest Davies-Bouldin score: 1.108, Optimal number of components: 2
Highest Calinski-Harabasz score: 4497.314, Optimal number of components: 2
```

Figure 55: Summary of Score Metrics and their Corresponding Optimal Number of Principal Components

In Figure 54, the cumulative explained variance graph suggests that the optimal number of principal components is around 6, where about 90% of variance of the data can be explained. However, based on the silhouette score, Davies-Bouldin score, and Calinski-Harabasz score in Figure 55 , they indicate that the best clustering performance is achieved with the optimal number of principal components being 2. Therefore, the addition of more principal components may be capturing noise instead of meaningful separation of the data points. Hence we will proceed with the optimal number of principal components being 2.

3.3.1.3. Cluster Analysis Using Optimal Configuration

This section provides the final AHC on the data points using the optimal configuration of hyperparameters (i.e. optimal number of clusters, optimal linkage method, optimal number of principal components) found in previous sections.

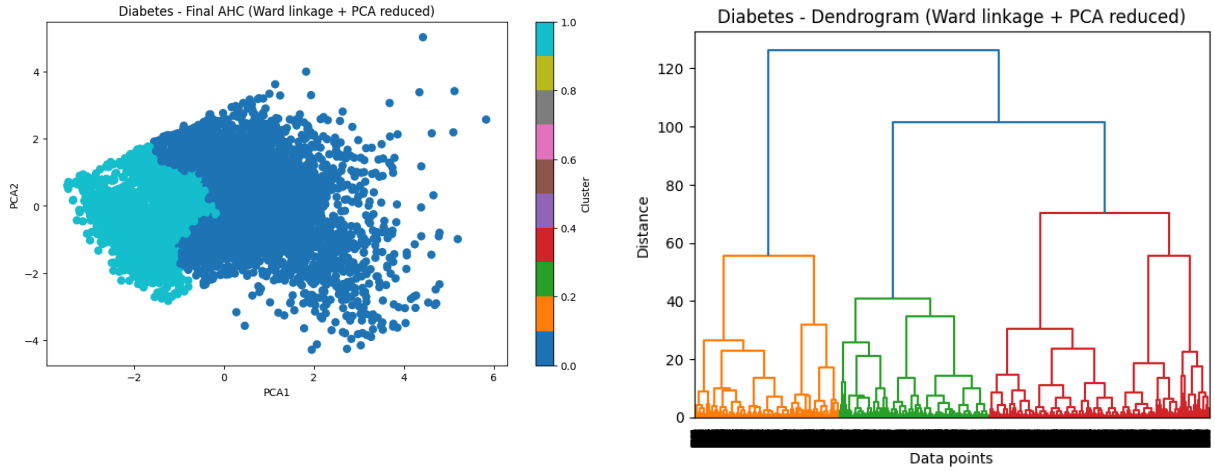


Figure 56: 2D Scatter Plot and Dendrogram to Visualise Final AHC on Diabetes Prediction Dataset

Figure 56 shows the final AHC on the data points in the Diabetes Prediction Dataset using optimal configuration – number of clusters: 2, linkage method: ward, number of principal components: 2.

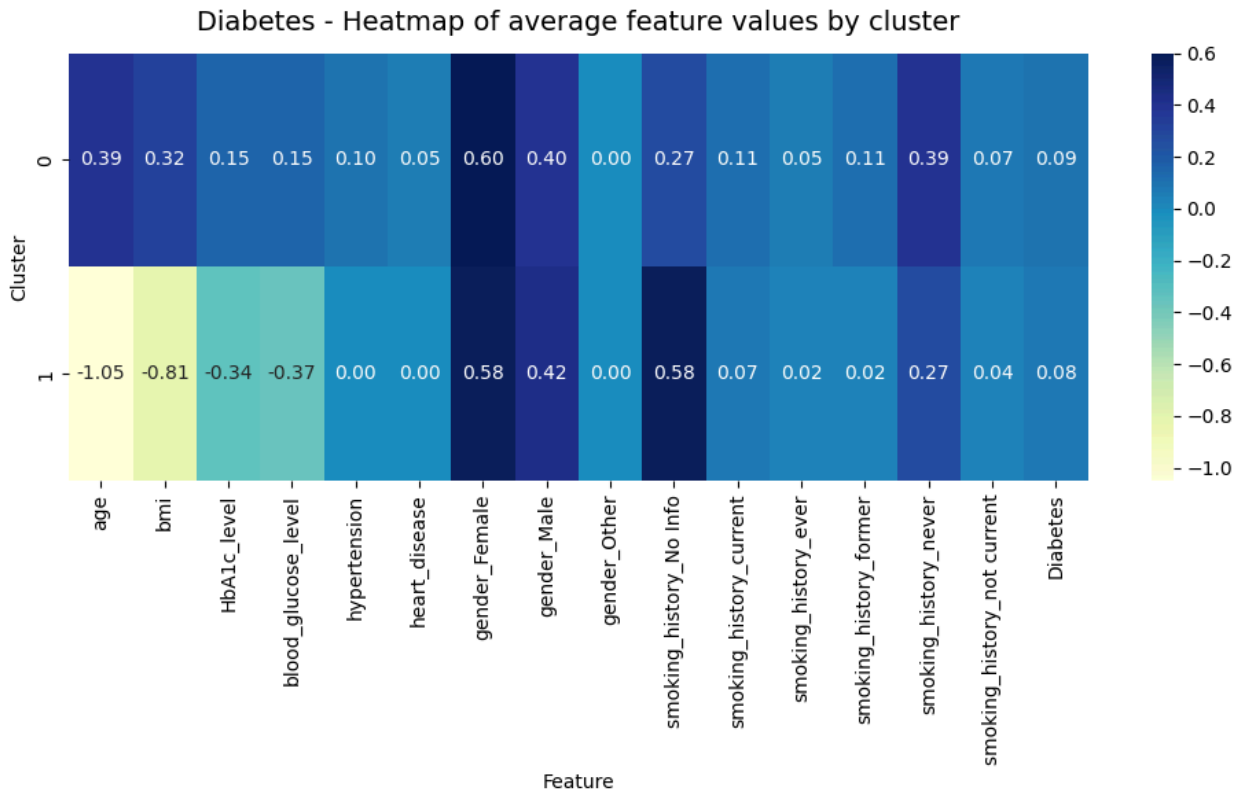


Figure 57: Heatmap to Visualise the Clusters after AHC with Optimal Configuration

In Figure 57, cluster 0 shows older individuals with higher BMI, higher blood glucose levels, and higher prevalence of hypertension and heart disease, among other health metrics. On the other hand, cluster 1 shows younger individuals with lower BMI, and lower blood glucose levels, among other health metrics.

Although the proportion of diabetic individuals are similar across both clusters (see Diabetes feature), the health metrics still suggest that individuals in cluster 0 still have a higher risk of diabetes.

3.3.2. Body Fat Prediction Dataset

3.3.2.1. Comparing Linkage Criteria

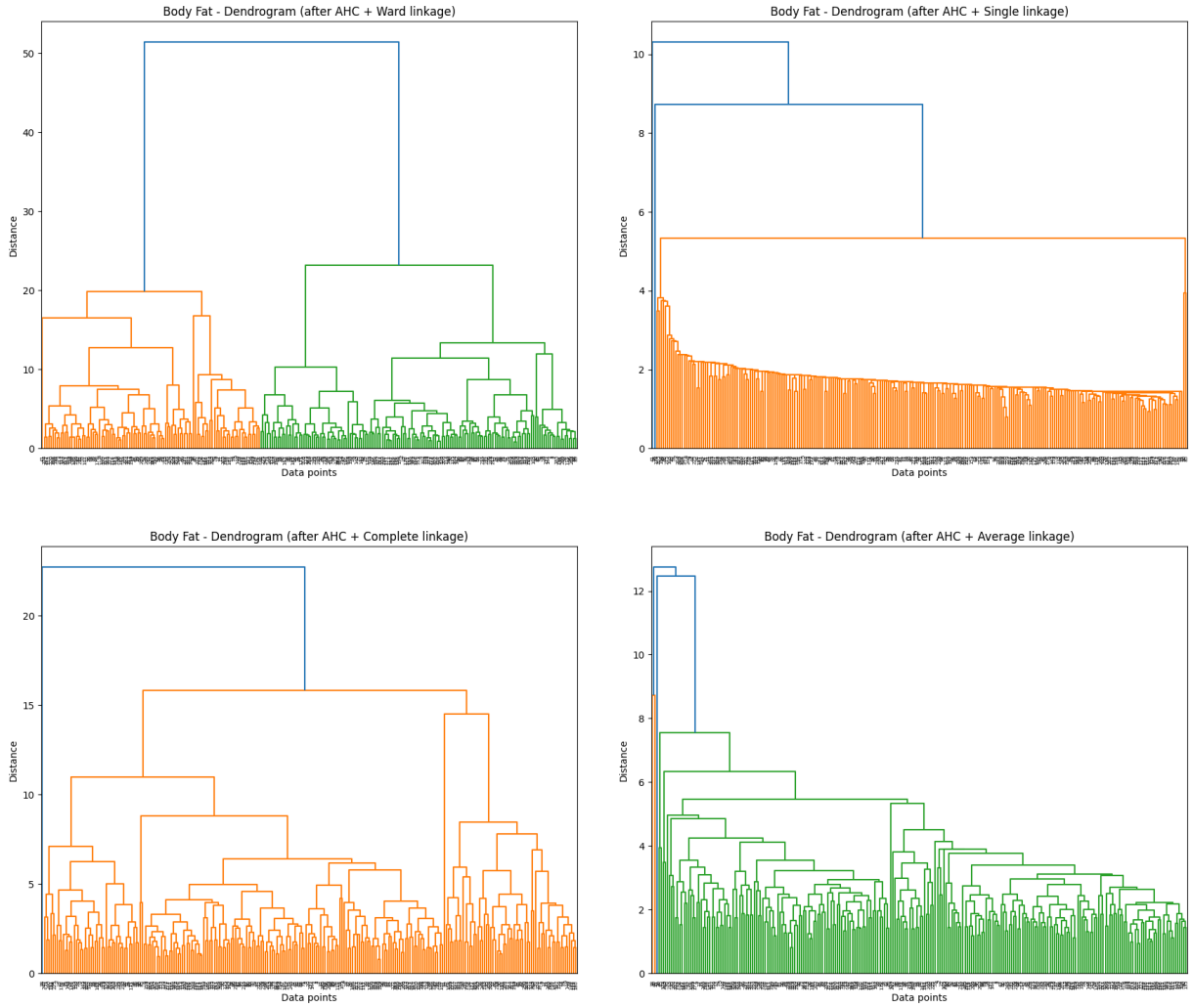
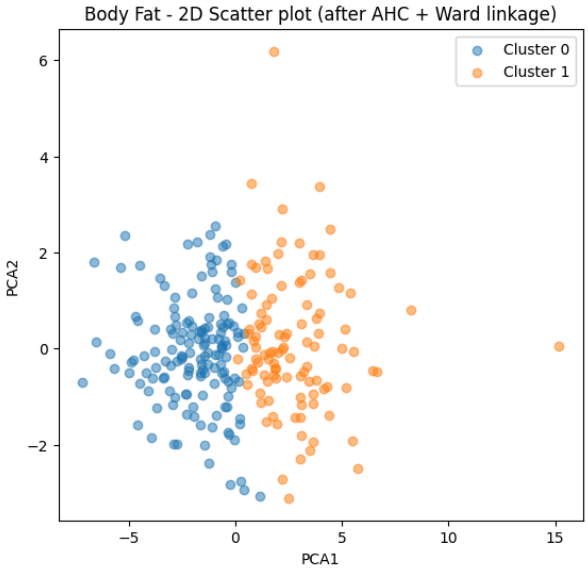
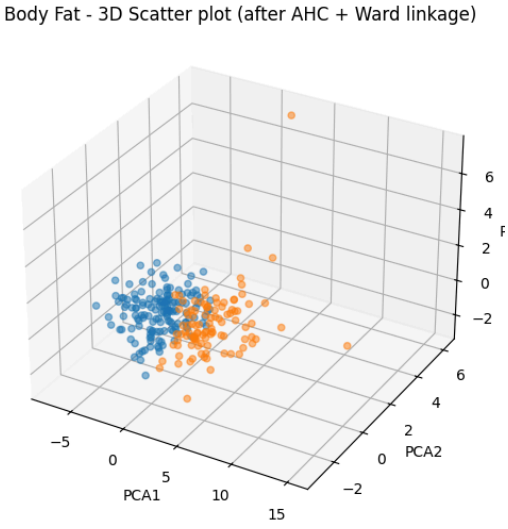
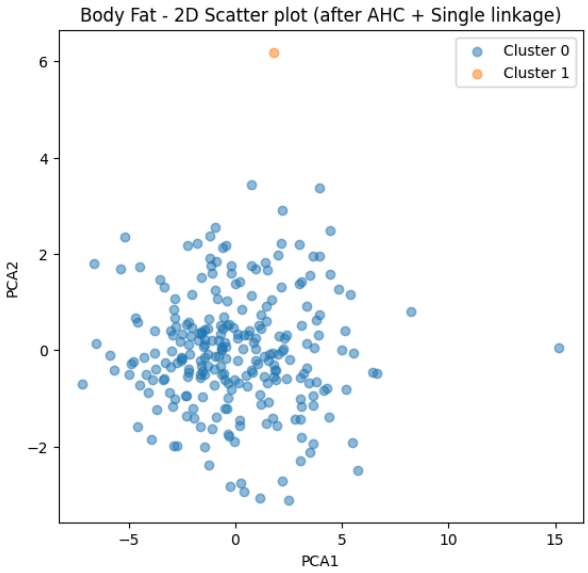


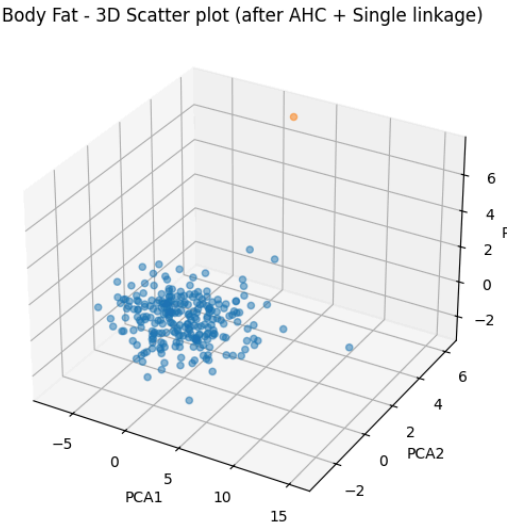
Figure 58: Dendrograms of AHC using Different Linkage Criteria

Figure 58 shows that Ward's linkage method gives the most evenly-clustered results of AHC as compared to single, complete, and average linkage methods. This is further corroborated by the following scatter plots:


Body Fat - 2D Scatter plot (after AHC + Ward linkage)
Cluster 0
Cluster 1
PCA1
PCA2


Body Fat - 3D Scatter plot (after AHC + Ward linkage)
PCA1
PCA2


Body Fat - 2D Scatter plot (after AHC + Single linkage)
Cluster 0
Cluster 1
PCA1
PCA2


Body Fat - 3D Scatter plot (after AHC + Single linkage)
PCA1
PCA2

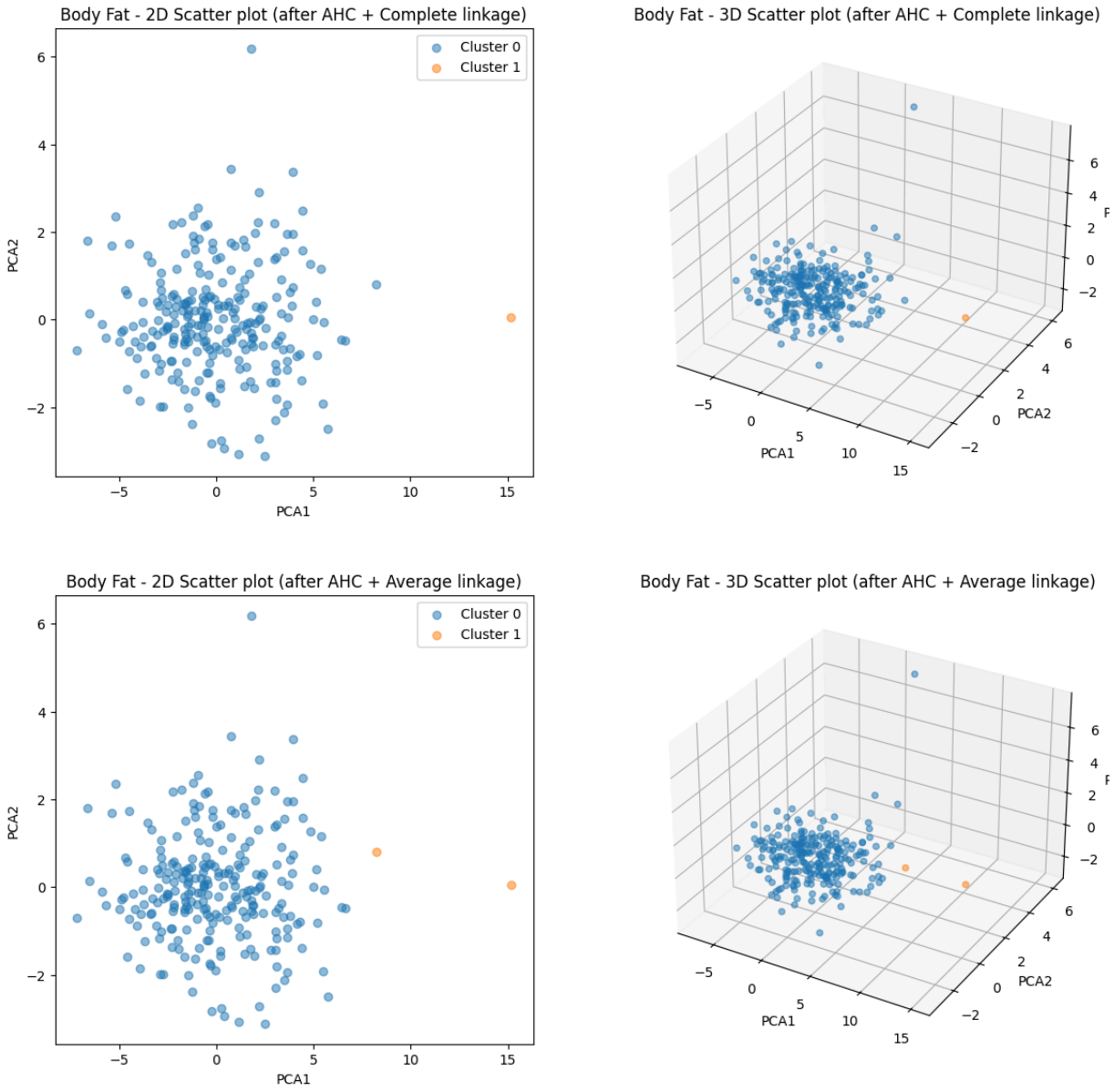


Figure 59: 2D and 3D Scatter Plots using Different Linkage Criteria

Figure 59 illustrates that Ward's linkage achieves the most desirable clustering results.

3.3.2.2. Comparing PCA Performance

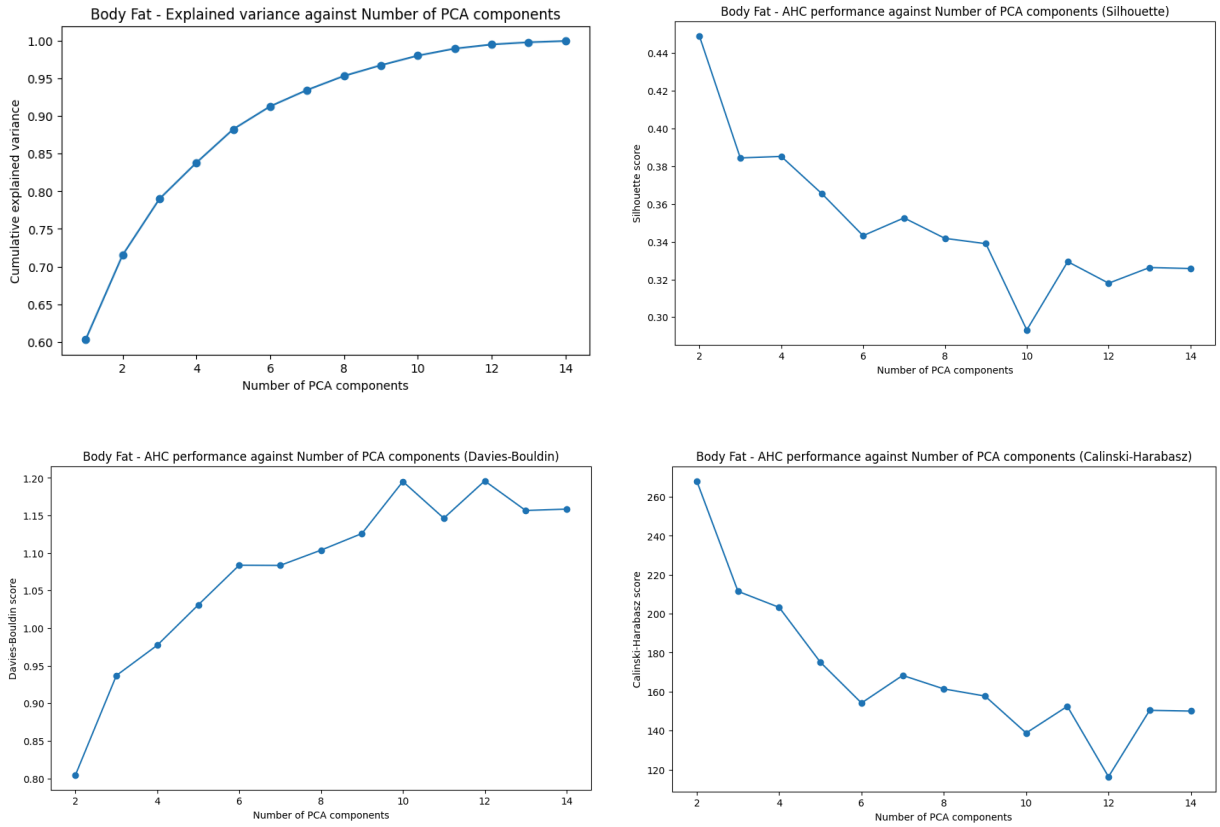


Figure 60: Line Graphs to Visualise Cumulative Explained Variance, Silhouette Score, Davies-Bouldin Score and Calinski-Harabasz Score against Number of Principal Components

```
Highest Silhouette score: 0.449, Optimal number of components: 2
Lowest Davies-Bouldin score: 0.805, Optimal number of components: 2
Highest Calinski-Harabasz score: 268.084, Optimal number of components: 2
```

Figure 61: Summary of Score Metrics and their Corresponding Optimal Number of Principal Components

The cumulative explained variance graph in Figure 60 suggests that the optimal number of principal components is around 6 or 7 where about 90% to 95% of variance in data can be explained. However, based on cluster performance analysis using Silhouette score, Davies-Bouldin score, and Calinski-Harabasz score in Figure 61, the optimal number of principal components is 2. The higher number of principal components from the explained variance graph may be due to fitting of noisy data. Hence, we determine the optimal number of principal components to be 2.

3.3.2.3. Cluster Analysis Using Optimal Configuration

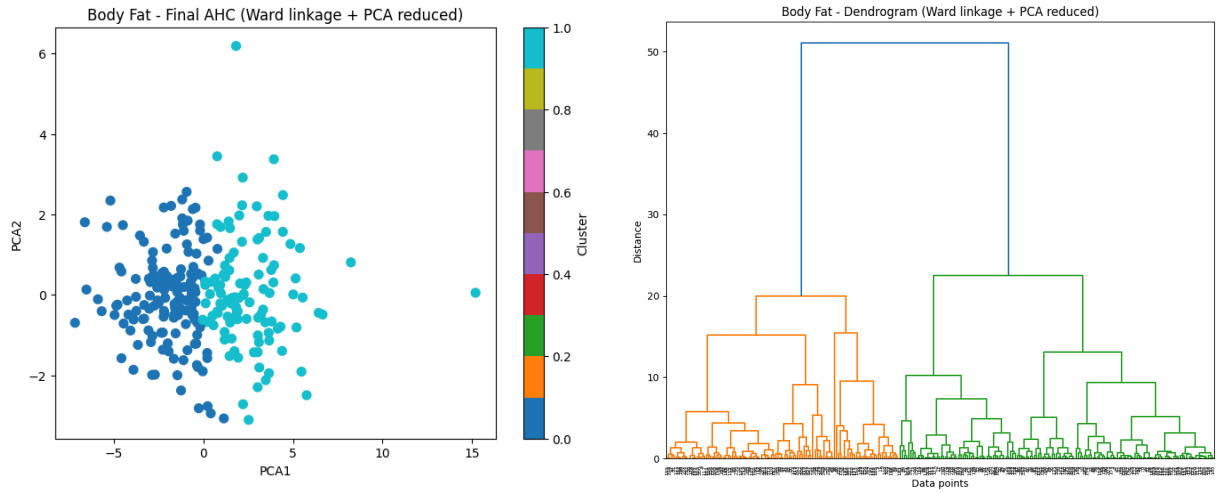


Figure 62: 2D Scatter Plot and Dendrogram to Visualise Final AHC on Body Fat Dataset

Figure 62 illustrates the final AHC on the data points using optimal configuration – number of clusters: 2, linkage method: ward, number of principal components: 2.

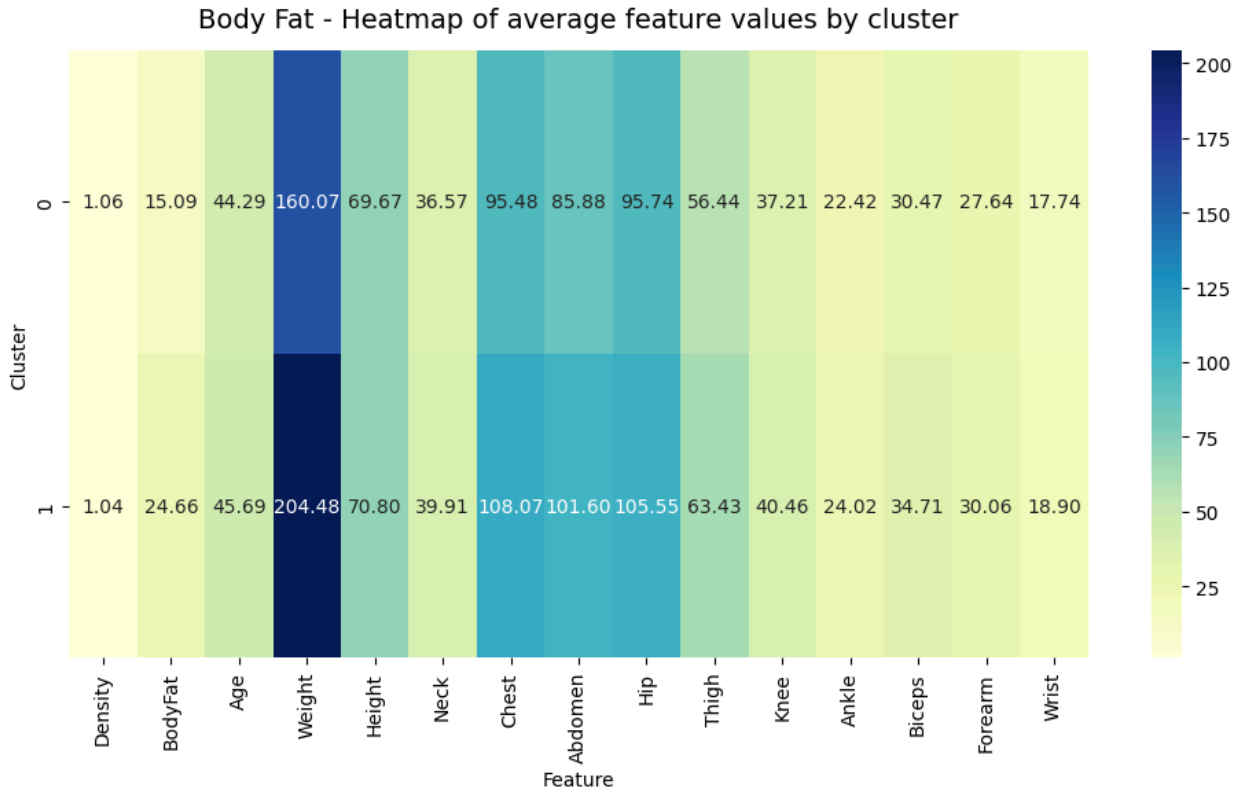


Figure 63: Heatmap to Visualise the Clusters after AHC using Optimal Configuration

In Figure 63, cluster 0 shows younger individuals with lower weight, shorter in height, and smaller body measurements (e.g. smaller neck, chest, abdomen, etc. measurements). On the other hand, cluster 1 shows older individuals with higher weight, taller in height, and larger body measurements. This corresponds with the BodyFat feature where individuals in cluster 0 have lower body fat percentage as compared to individuals in cluster 1.

3.3.3. World Bank Dataset

3.3.3.1. Comparing Linkage Criteria

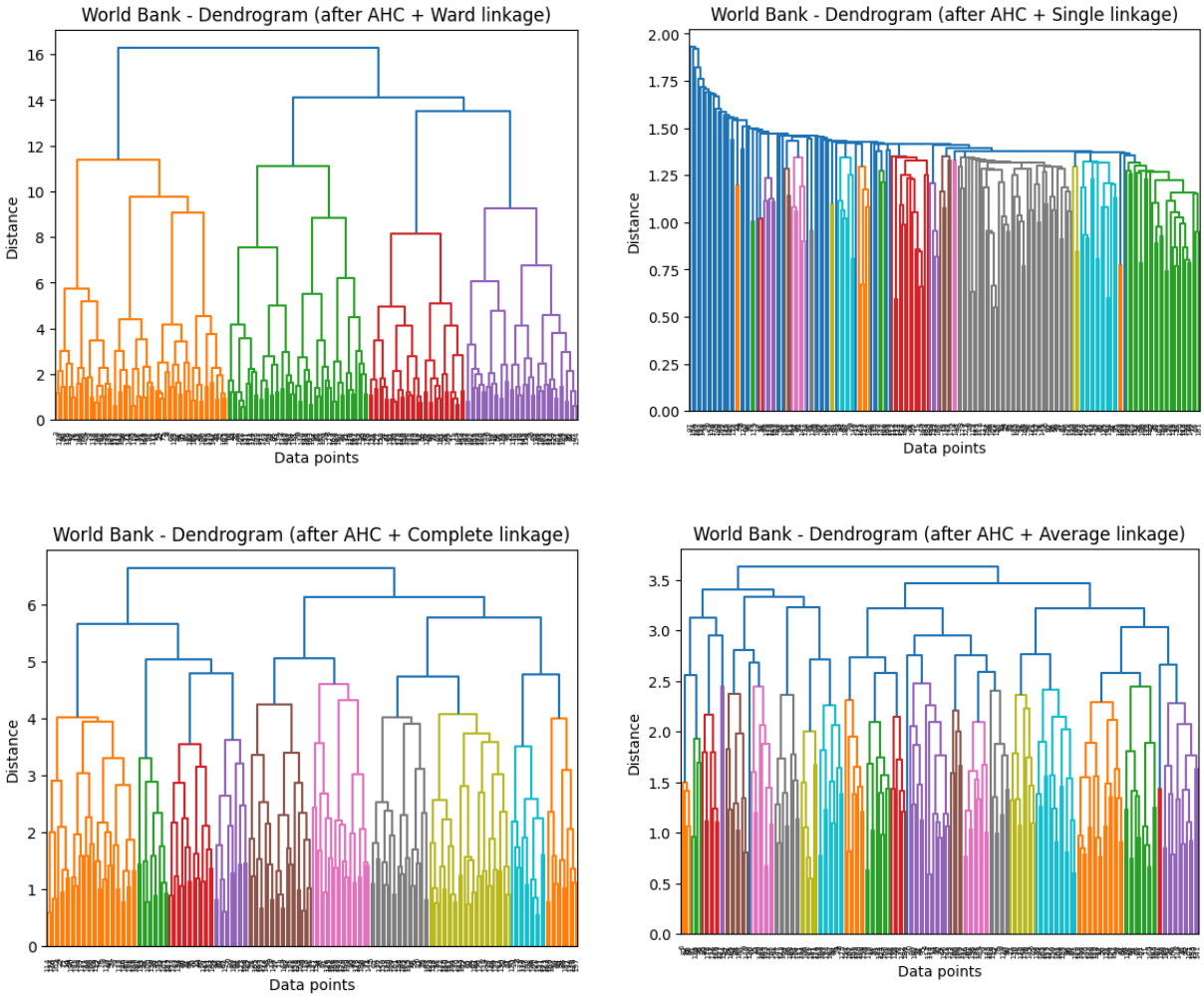


Figure 64: Dendrograms of AHC using Different Linkage Criteria

Figure 64 shows that Ward's linkage provides the most compact and evenly-separated clusters as compared to the clustering using single, complete, and average linkage. Hence Ward's linkage will be used for further analysis.

3.3.3.2. Comparing PCA Performance

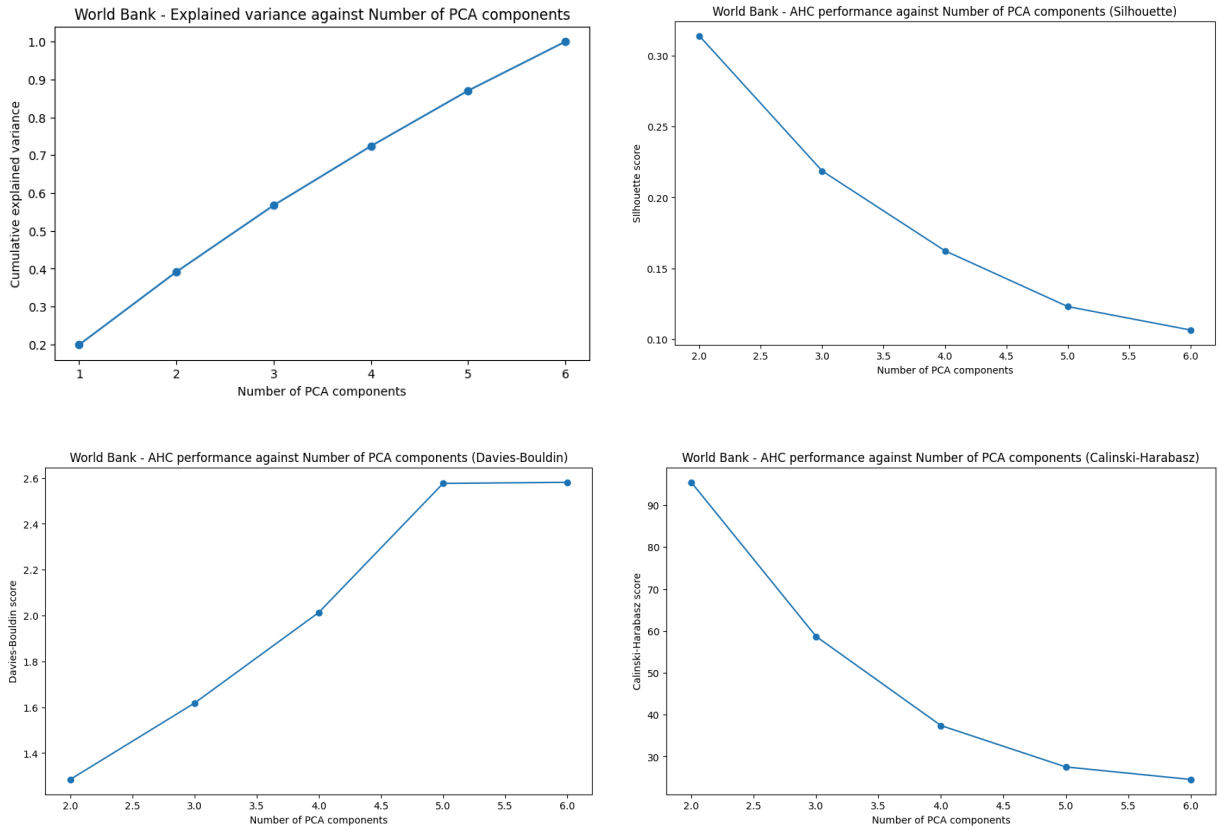


Figure 65: Line Graphs to Visualise the Cumulative Explained Variance, Silhouette Score, Davies-Bouldin Score and Calinski-Harabasz Score against Number of Principal Components

```
Highest Silhouette score: 0.314, Optimal number of components: 2
Lowest Davies-Bouldin score: 1.285, Optimal number of components: 2
Highest Calinski-Harabasz score: 95.473, Optimal number of components: 2
```

Figure 66: Summary of Score Metrics and their Corresponding Optimal Number of Principal Components

The cumulative explained variance graph in Figure 65 shows a higher optimal number of principal components as compared to that in Figure 66 which may be due to fitting of noisy data. Figure 66 shows the AHC performance based on the different score metrics, hence, we will use the optimal number of principal components found here, which is 2.

3.3.3.3. Cluster Analysis Using Optimal Configuration

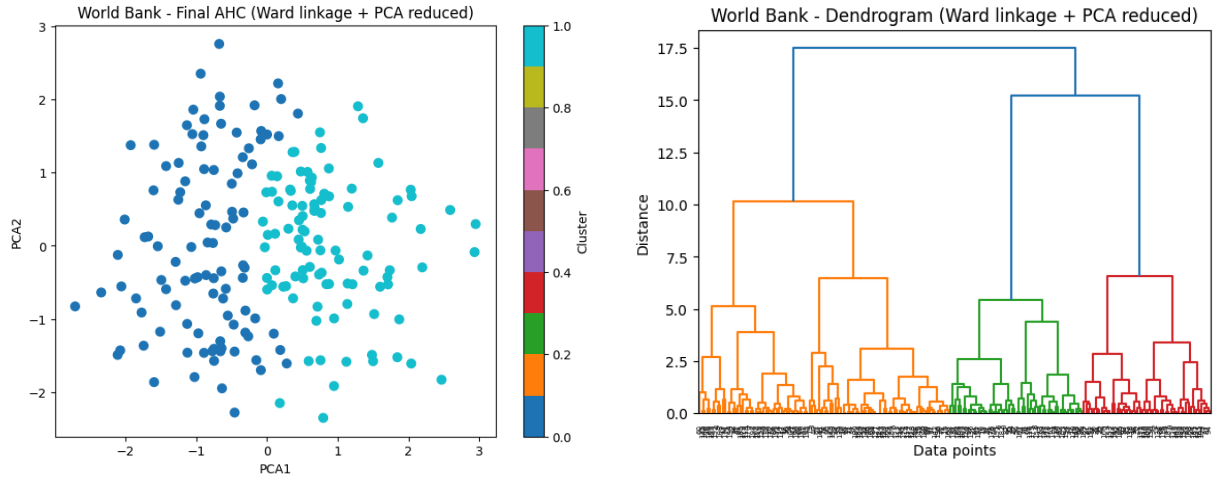


Figure 67: 2D Scatter Plot and Dendrogram to Visualise Final AHC on World Bank Dataset

Figure 67 illustrates the final AHC on the data points using the optimal configuration found above – number of clusters: 2, linkage method: ward, number of principal components: 2.

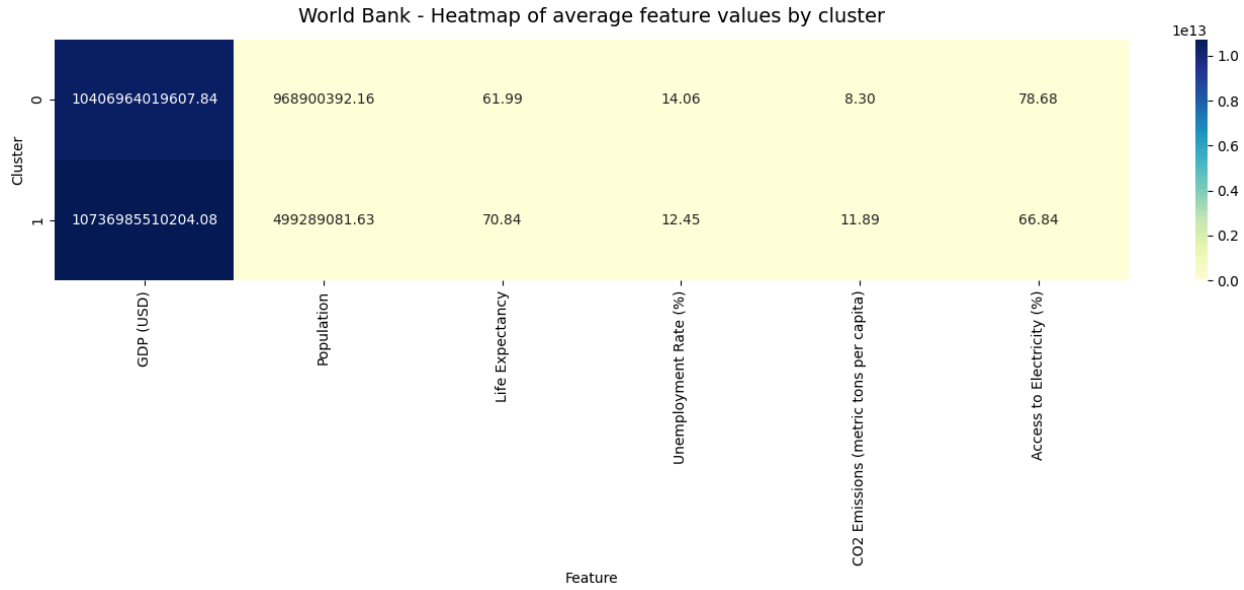


Figure 68: Heatmap to Visualise the Clusters after AHC using Optimal Configuration

In Figure 68, both clusters show similar total GDP. Cluster 0 contains countries with larger populations, lower life expectancy, and slightly higher unemployment rate, among other features as compared to the countries in cluster 1. This suggests that countries in cluster 0 are likely to be developing countries with dense populations, poorer overall healthcare, and weaker labour

market, while countries in cluster 1 are likely to be developed countries with better overall healthcare and a stronger labour market.

3.4. Gaussian Mixed Model Analysis

3.4.1. Diabetes Prediction Dataset

n_components	covariance_type	Silhouette	Calinski-Harabasz	Davies-Bouldin	AIC	BIC
2	full	0.503241	1661.348351	1.565676	2722.762972	2989.967892
	tied	0.515629	1282.462707	1.487119	66920.625243	67090.072266
	diag	0.285981	429.551745	3.227321	60689.944410	60826.805467
	spherical	0.483633	1539.234780	1.665787	65420.724863	65505.448375
3	full	0.513461	1502.554174	1.384635	1816.726895	2220.792873
	tied	0.204940	1445.196329	1.414941	70648.545921	70857.096103
	diag	0.183371	1006.952367	1.751244	54766.683287	54975.233469
	spherical	0.222556	1216.782386	1.581558	62730.181728	62860.525591
4	full	0.495937	1523.706112	0.906981	-2548.319222	-2007.392187
	tied	0.223922	1460.072223	1.326000	55513.501815	55761.155157
	diag	0.208613	1010.163186	1.541433	53268.878204	53549.117511
	spherical	0.210074	1014.046374	1.542039	60389.078716	60565.042932
5	full	0.253834	1786.075449	1.121387	-7988.221937	-7310.433846
	tied	0.252941	1789.108149	1.120422	50727.919567	51014.676068
	diag	0.136323	1019.117557	1.748182	49893.338353	50245.266785
	spherical	0.239367	1154.382511	1.309782	59044.609755	59266.194324

Figure 69: Ablation Study of Gaussian Mixed Model on Diabetes Prediction Dataset

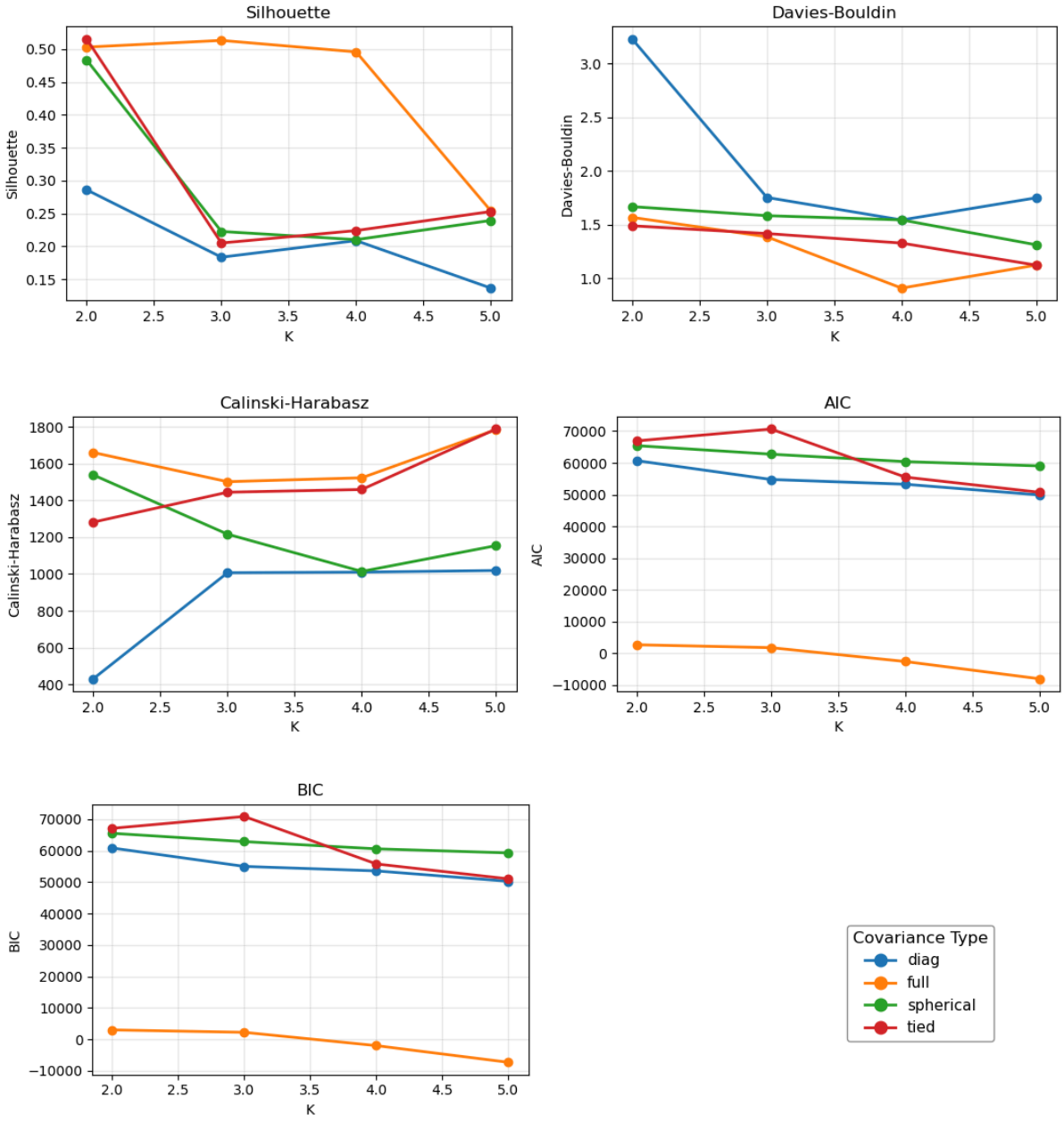


Figure 70: Analysis of Ablation for Varying Number of Components in Diabetes Prediction Dataset

Silhouette Score	n_components = 2 provides the highest Silhouette Scores across most covariance types, this implies that 2 components gives the most well-separated clusters. • Covariance Type: ○ Full, Spherical and Tied covariance

	types yield the best scores at 2 components. ○ Diag covariance produces the worst Silhouette Score overall. **Takeaway: For well-separated clusters, 2 components with any covariance type besides Diag perform the best.**
Davies-Bouldin Index	A lower index indicates small overlap in clusters. ● n_components = 4 with Tied, Full and Spherical covariance generally perform pretty well, with low index values, indicating high separation of clusters and low variance within clusters. ● The Diag covariance type generally has the poorest index, suggesting this configuration has a lot of overlaps between its clusters. **Takeaway: The best model fit is achieved with 2 components and Tied, Full and Spherical covariance type.**
Calinski-Harabasz Index	Contrary to the Davies-Bouldin Index, a higher Calinski-Harabaz Index shows small overlap in clusters with the centroids of each cluster being far away from each other. ● n_components = 5 with Tied and Full covariance generally perform pretty well. ● Spherical Covariance type with smaller components performed alright, but generally declines when more components are added. ● The Diag covariance type generally has the poorest index, suggesting this configuration has a lot of overlaps between its clusters. **Takeaway: The best model fit is achieved with 5 components and Tied and Full covariance type. Spherical Covariance performs decently with smaller components, achieving its optimality at 2.**

Akaike Information Criterion (AIC)	Lower AIC values indicate better model fit. ● n_components = 5 with Full covariance type shows the best (most negative) AIC, showing this configuration gives the best fit to the data. ● Tied, Diag and Spherical covariance generally perform worse, with higher AIC values, suggesting a lack of fit. **Takeaway: The best model fit is achieved with 5 components leveraging on full covariance type.**
Bayesian Information Criterion (BIC)	Lower BIC values indicate better models. ● Similar to AIC, n_components = 5 with full covariance type has the most negative BIC values, indicating optimality. ● Tied, Diag and Spherical covariance results in higher BIC values, suggesting that this configuration is suboptimal compared to the full covariance types. **Takeaway: The best BIC is achieved with 5 components and full covariance type.**

Conclusion

- 2 components provide the best Silhouette Score and Davies-Bouldin Index indicating well-separated clusters, especially for Full, Diag & Spherical covariance types.
- 5 components gave the best Calinski-Harabasz Index, indicating separation of clusters as well, and works best for Tied and Full covariance types.
- 5 components with Full covariance offer the best fit to the data, as reflected by both AIC and BIC metrics.
- Full covariance shows adequate performance across various metrics, making it a viable option in many instances.

3.4.2. World Bank Dataset

	n_components	covariance_type	Silhouette	Calinski-Harabasz	Davies-Bouldin	AIC	BIC
0	2	full	0.128489	29.753444	2.496823	2952.679030	3087.910042
1	2	tied	0.143847	34.959061	2.235381	2942.852692	3028.608944
2	2	diag	0.111914	25.045509	2.722390	2939.551949	3008.816614
3	2	spherical	0.137205	32.864787	2.345620	2936.381601	2979.259727
4	3	full	0.135653	31.854047	2.102624	2941.152532	3145.648209
5	3	tied	0.160171	36.913007	1.832835	2937.007376	3042.553531
6	3	diag	0.151009	34.817391	1.895756	2955.748007	3061.294163
7	3	spherical	0.159097	37.039484	1.835579	2942.839954	3008.806301
8	4	full	0.132640	29.568020	1.826936	2948.930212	3222.690554
9	4	tied	0.126658	28.505916	1.927303	2914.005752	3039.341812
10	4	diag	0.141950	30.732514	1.828882	2958.639299	3100.466946
11	4	spherical	0.151776	34.404619	1.763090	2947.611693	3036.666262
12	5	full	0.137571	29.950658	1.689560	2968.873639	3311.898645
13	5	tied	0.156442	32.279963	1.723088	2944.187947	3089.313911
14	5	diag	0.148004	32.581810	1.628374	2959.710017	3137.819155
15	5	spherical	0.137896	32.076715	1.584172	2937.364245	3049.507035
16	6	full	0.153707	30.403895	1.500915	2941.861201	3354.150872
17	6	tied	0.107615	27.373599	1.707349	2921.975716	3086.891584
18	6	diag	0.129577	29.144092	1.553191	2961.343736	3175.734365
19	6	spherical	0.154163	32.421133	1.466234	2947.086646	3082.317658

Figure 71: Ablation Study of Gaussian Mixed Model on World Bank Dataset

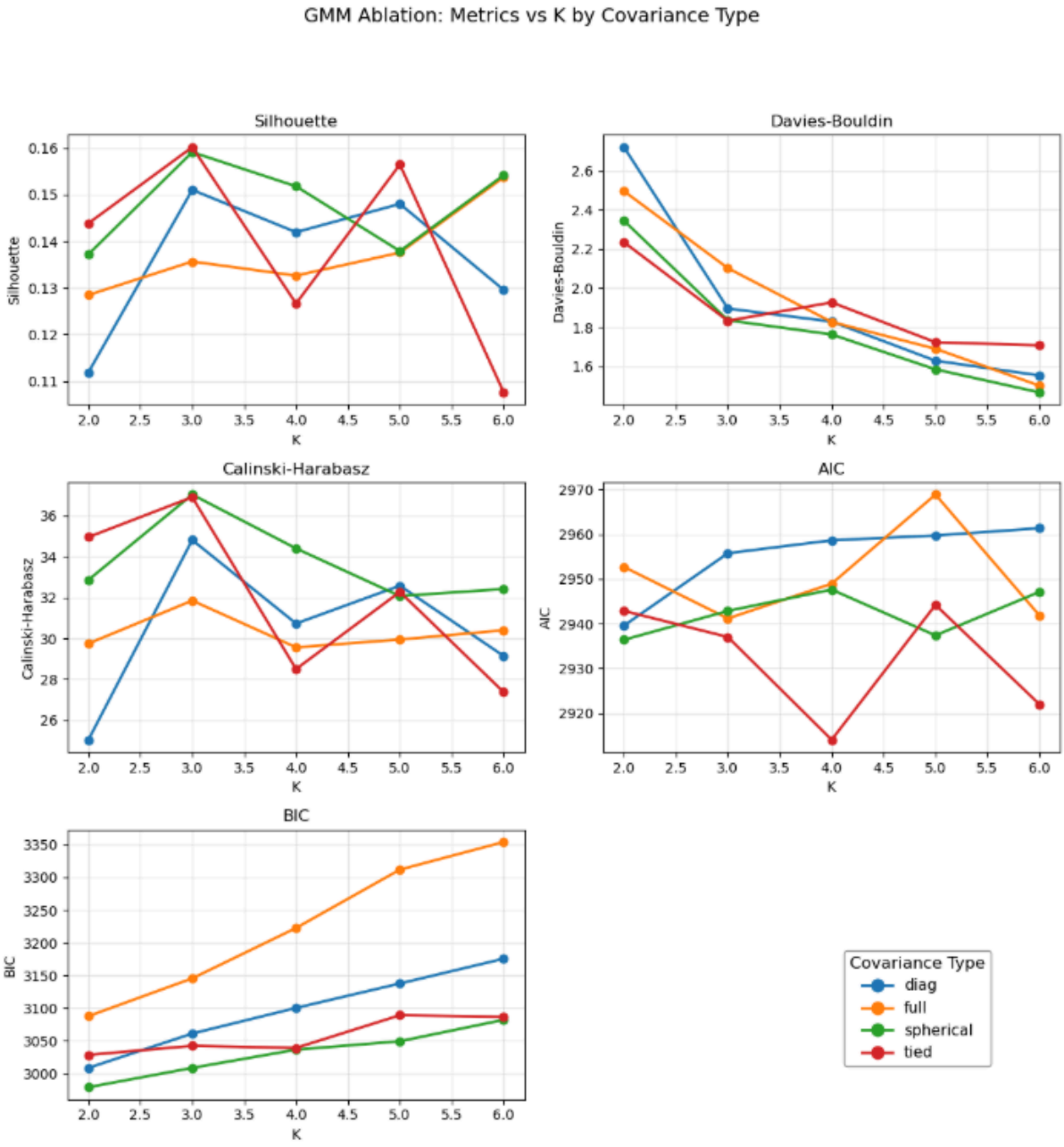


Figure 72: Analysis of Ablation for Varying Number of Components in World Bank Dataset

Silhouette Score	n_components = 3 provides the highest Silhouette Scores across most covariance types, this implies that 3 components gives the most well-separated clusters. ● Covariance Type: ○ Spherical and Tied covariance types yield the best scores at 3 components. ○ Diag covariance type performs relatively well, peaking at 2 components. ○ Full covariance generally produces the worst Silhouette Score overall. **Takeaway: For well-separated clusters, 3 components with Spherical and Tied covariance type perform the best.**
Davies-Bouldin Index	A lower index indicates small overlap in clusters. ● n_components = 6 for all four types of covariance generally perform pretty well, with low index values, indicating high separation of clusters and low variance within clusters. ● Diag covariance type performed the best. **Takeaway: The best model fit is achieved with 6 components across all 4 covariance types.**
Calinski-Harabasz Index	Contrary to the Davies-Bouldin Index, a higher Calinski-Harabaz Index shows small overlap in clusters with the centroids of each cluster being far away from each other. ● n_components = 3 with Tied and Spherical covariance generally perform pretty well. ● The Diag covariance type generally has the poorest index, suggesting this configuration has a lot of overlaps between its clusters. **Takeaway: The best model fit is achieved with 3 components and Tied and Spherical covariance type.**

Akaike Information Criterion (AIC)	Lower AIC values indicate better model fit. ● n_components = 4 with Tied covariance type shows the best (lowest) AIC, showing this configuration gives the best fit to the data. ● Full and Diag covariance types generally perform worse, with higher AIC values, suggesting a lack of fit. **Takeaway: The best model fit is achieved with 4 components leveraging on Tied covariance type.**
Bayesian Information Criterion (BIC)	Lower BIC values indicate better models. ● BIC performs better with fewer components, achieving the best results when n_components = 2 with Spherical covariance type has the lowest BIC values, indicating optimality. ● Tied, Diag and Full covariance results in higher BIC values, suggesting that this configuration is suboptimal compared to the Spherical covariance types. **Takeaway: The best BIC is achieved with 2 components and Spherical covariance type.**

Conclusion

- 3 components provide the best Silhouette Score and Calinski-Harabasz Index, indicating well-separated clusters, especially for Tied & Spherical covariance types.
- 6 components gave the best Davies-Bouldin Index, indicating separation of clusters as well, and works best for all 4 covariance types.
- 4 components with Tied covariance type, as well as 2 components with Spherical covariance type offer the best fit to the data, as reflected by both AIC and BIC metrics respectively.
- Tied and Spherical covariance shows adequate performance across various metrics, making it a viable option in many instances.

4. Comparative Analysis

4.1. K-Means vs K-Means++

4.1.1. Diabetes Prediction Dataset

The Diabetes Prediction dataset contains 100,000 samples with 7 numeric features. Both algorithms were evaluated across K=2 to K=6, with 10 trials per configuration.

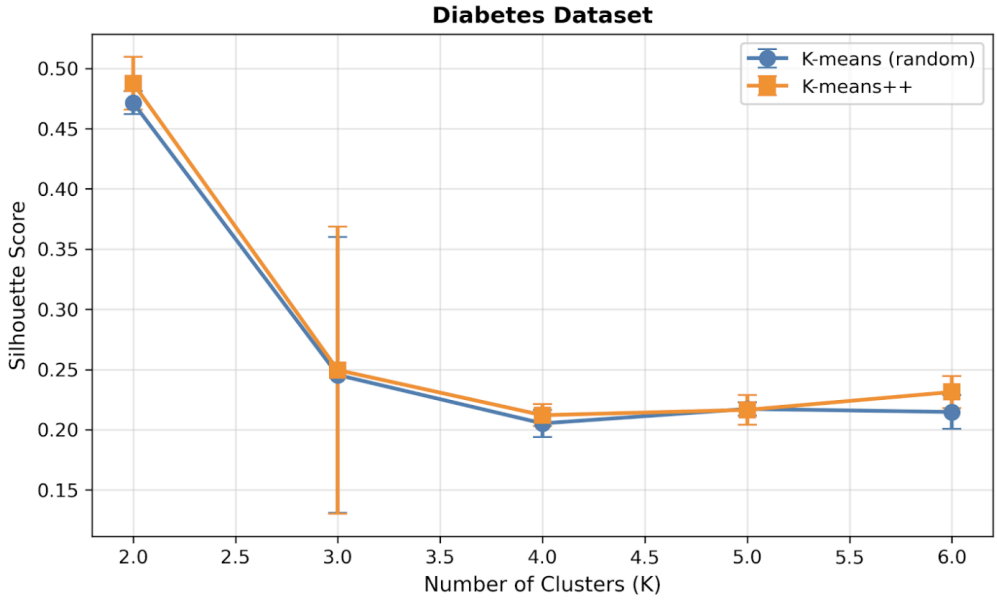


Figure 73: Diabetes Silhouette Scores Comparison

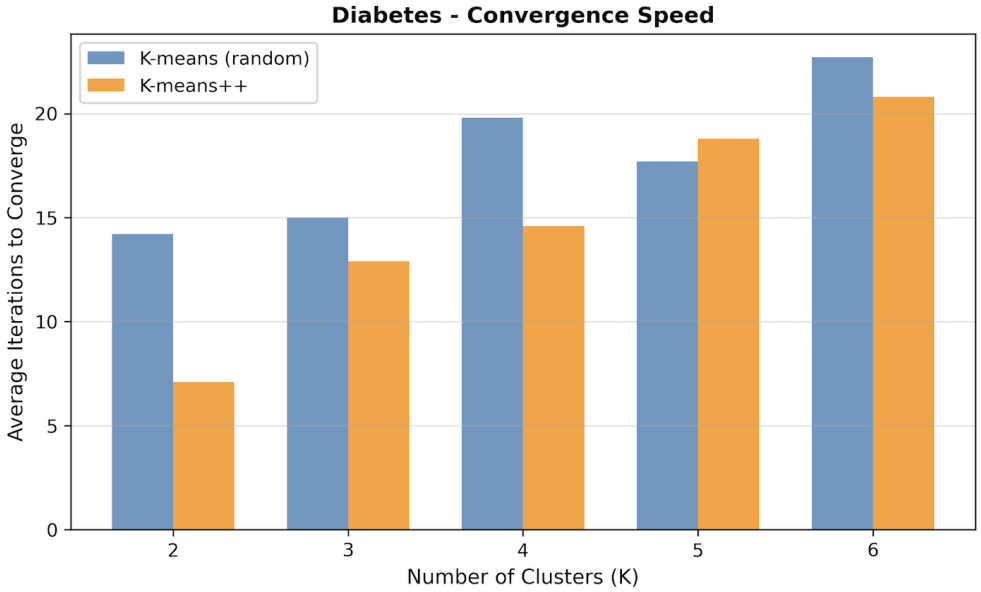


Figure 74: Diabetes Convergence Speed Comparison

Performance at Optimal K=2			
Metric	**K-Means**	**K-Means++**	**Difference (K-Means++ as compared to K-Means)**
Silhouette Score	0.4715	0.4876	+3.4%
Convergence (iterations)	14.2	7.1	50% faster
Inertia	538,635	536,403	Similar
Stability (std)	0.0096	0.0219	Higher variance

Silhouette Score	
K-Means	**K-Means++**
Lower silhouette scores due to poor initialization leading to clusters that overlap or are not well-defined	Generally yields higher silhouette scores, indicating more distinct and well-separated clusters. This improvement arises from the strategic centroid initialization that captures the data distribution more effectively

Convergence Speed	
K-Means	**K-Means++**
Requires more iterations to converge (14.2 avg) as random initialization often starts far from optimal centroids, necessitating more refinement steps	Converges significantly faster (7.1 avg iterations) due to superior initial centroid placement, reducing computational time by approximately 50%

Conclusion
K-Means++ demonstrated superior performance on the Diabetes dataset, achieving 3.4% higher clustering quality (silhouette: 0.4876 vs 0.4715) and converging 50% faster than standard K-Means (7.1 vs 14.2 iterations). The optimal K=2 configuration reveals a natural binary separation in the data. The strategic initialization of K-Means++ proves highly effective for large-scale datasets, providing both better cluster quality and significant computational efficiency gains.

4.1.2. World Bank Dataset

The World Bank dataset contains 200 samples with 7 numeric features. Both algorithms were evaluated across K=2 to K=6, with 10 trials per configuration.

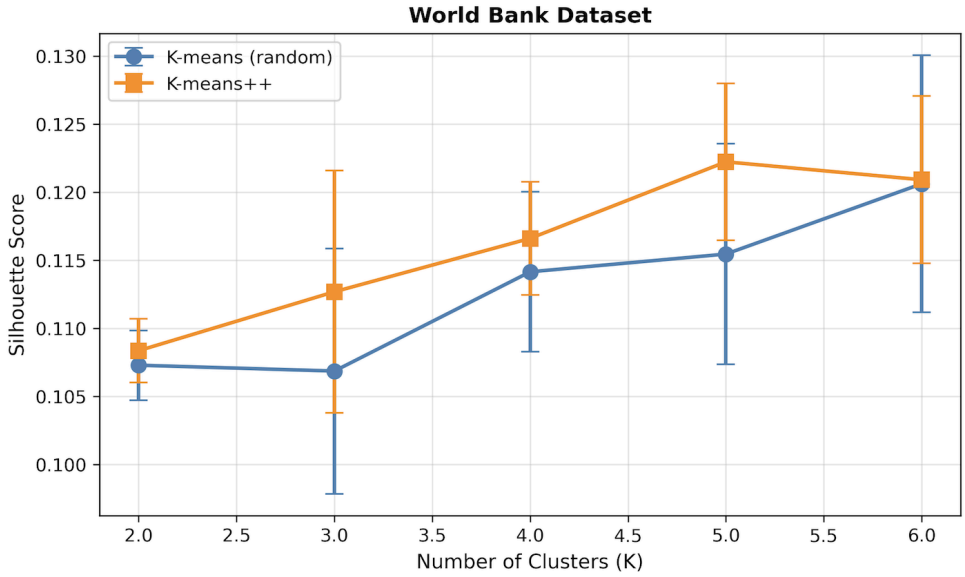


Figure 75: World Bank Silhouette Comparison

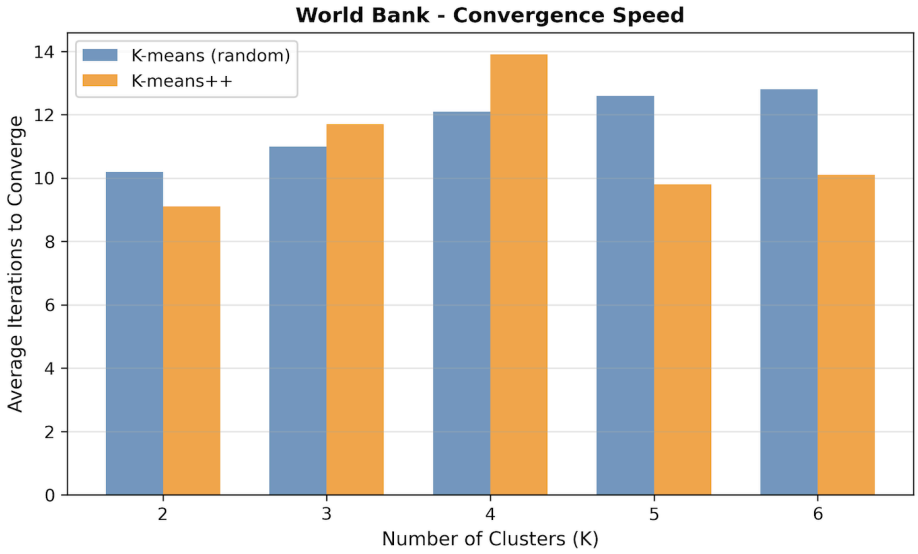


Figure 76: World Bank Convergence Speed Comparison

Silhouette Score	
K-Means	**K-Means++**
Achieves marginally lower silhouette scores with higher variance across runs (std: 0.0094 at K=6), indicating less consistent clustering quality	Consistently achieves higher silhouette scores with lower variance (std: 0.0058 at K=5), demonstrating 38% better stability and more reproducible results

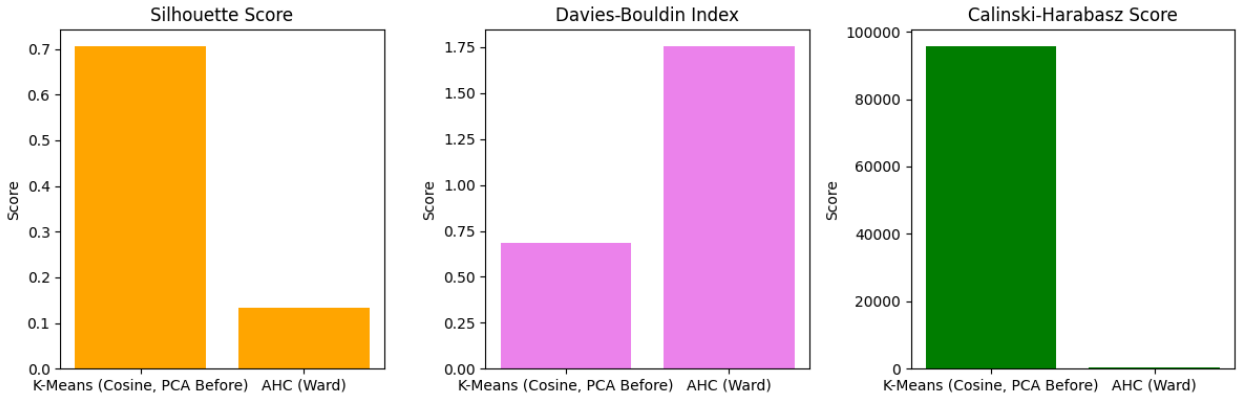
Convergence Speed	
K-Means	**K-Means++**
Averages 12.8 iterations at optimal K=6, with variable convergence across different K values due to initialization sensitivity	Requires 23% fewer iterations (9.8 at K=5), demonstrating computational efficiency even on smaller datasets

Conclusion
K-Means++ outperformed standard K-Means on the World Bank dataset, achieving marginally better clustering quality (optimal silhouette: 0.1222 vs 0.1206) with significantly improved stability with 38% lower variance across runs (0.0058 vs 0.0094). While the absolute performance improvements were smaller than those observed in larger datasets, K-Means++ demonstrated 23% faster convergence and more reproducible results. The consistent initialization strategy proves particularly valuable when result reliability and reproducibility are critical requirements.

4.2 K-Means vs Agglomerative Hierarchical Clustering

4.2.1. Diabetes Prediction Dataset

As previously compared across the 4 different scenarios, K-Means (Cosine) PCA Before Clustering achieved the best performance based on Silhouette Score. This optimal scenario for K-Means is now used to compare against Agglomerative Hierarchical Clustering on the Diabetes Prediction Dataset.



	Method	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Score
0	K-Means (Cosine, PCA Before)	0.7065	0.6838	95770.1166
1	AHC (Ward)	0.1325	1.7568	480.0601

Figure 77: Comparison of Clustering Performance Metrics between K-Means vs AHC

Silhouette Score	
K-Means	**Agglomerative Hierarchical Clustering**
Higher Silhouette Score, indicating well-separated and cohesive clusters	Lower Silhouette Score, suggesting weaker cluster structures and more overlap between data points

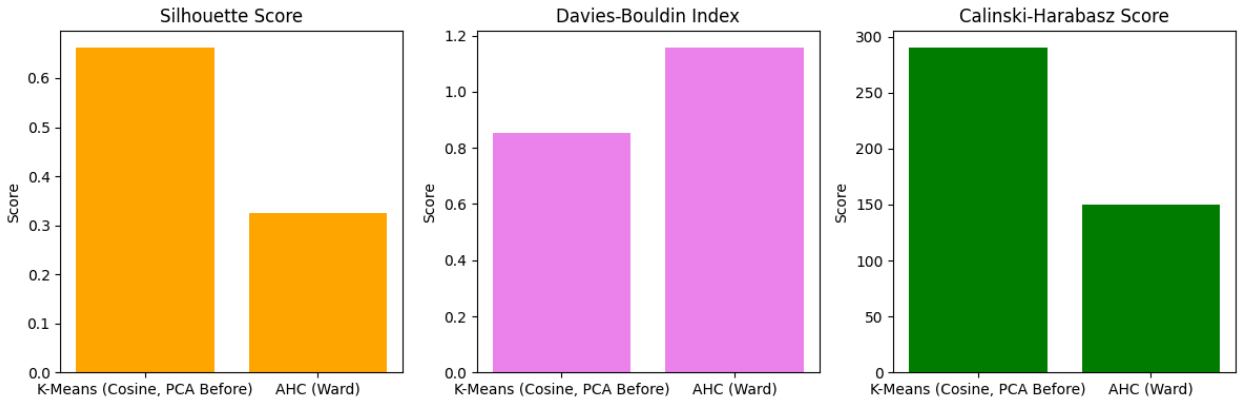
Davies-Bouldin Index	
K-Means	**Agglomerative Hierarchical Clustering**
Lower Davies-Bouldin Index, implying better cluster compactness and clear separation	Higher Davies-Bouldin Index, indicating poorer cluster definition and more similarity between clusters

Calinski-Harabasz Index	
K-Means	**Agglomerative Hierarchical Clustering**
Much higher Calinski-Harabasz Index, showing that clusters are distinct and densely packed	Very low Calinski-Harabasz Index, reflecting less distinct and loosely formed clusters

Conclusion
All three metrics show that K-Means outperforms Agglomerative Hierarchical Clustering for this dataset, producing more cohesive and well-separated clusters.

4.2.2. Body Fat Prediction Dataset

As previously compared across the 4 different scenarios, K-Means (Cosine) PCA Before Clustering achieved the best performance based on Silhouette Score. This optimal scenario for K-Means is now used to compare against Agglomerative Hierarchical Clustering on the Body Fat Prediction Dataset.



	Method	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Score
0	K-Means (Cosine, PCA Before)	0.6627	0.8549	290.5301
1	AHC (Ward)	0.3258	1.1585	150.0195

Figure 78: Comparison of Clustering Performance Metrics between K-Means vs AHC

Silhouette Score	
K-Means	**Agglomerative Hierarchical Clustering**
Higher Silhouette Score, indicating clearer and more distinct cluster boundaries	Lower Silhouette Score, suggesting weaker separation between clusters and possible overlap

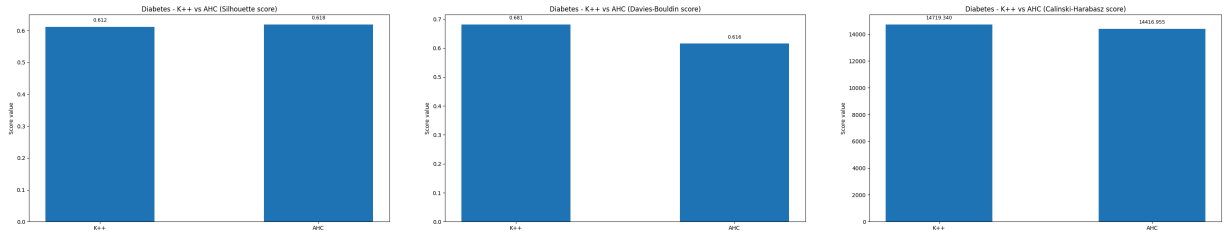
Davies-Bouldin Index	
K-Means	**Agglomerative Hierarchical Clustering**
Lower Davies-Bouldin Index, showing more compact and well-separated clusters	Higher Davies-Bouldin Index, indicating poorer cluster compactness and greater similarity between clusters

Calinski-Harabasz Index	
K-Means	**Agglomerative Hierarchical Clustering**
Higher Calinski-Harabasz Index, suggesting better-defined and more distinct clusters	Lower Calinski-Harabasz Index, suggesting less well-defined clusters

Conclusion
All three metrics indicate that K-Means performs better than Agglomerative Hierachical Clustering for this dataset, producing more compact and well-separated clusters.

4.3. K-Means++ vs Agglomerative Hierarchical Clustering

4.3.1. Diabetes Prediction Dataset




```
K++ | Silhouette score: 0.612, Davies-Bouldin score: 0.681, Calinski-Harabasz score: 14719.340
AHC | Silhouette score: 0.618, Davies-Bouldin score: 0.616, Calinksi-Harabasz score: 14416.955
```

Figure 79: Comparison of Clustering Performance Metrics between K-Means++ vs AHC

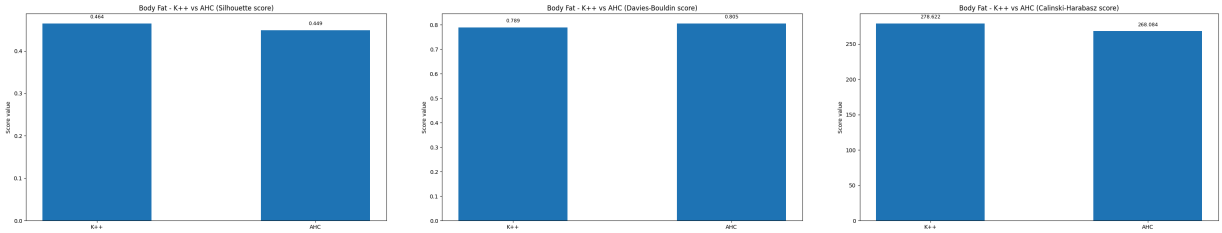
Silhouette Score	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows a very slightly smaller score compared to AHC, suggesting poorer clustering performance	Shows a slightly higher score compared to K-Means++, suggesting better performance as higher Silhouette score indicates that data points are fitted well in their respective clusters and clusters are compact (Geekforgeeks, 2025)

Davies-Bouldin Index	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows a higher score as compared to AHC, suggesting poorer clustering	Shows a lower score as compared to K-Means++, suggesting that AHC performed better as lower Davies-Bouldin index indicates that the clusters are well-separated and compact (Geekforgeeks, 2025)

Calinski-Harabasz Index	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows a higher score compared to AHC, suggesting better performance, as higher Calinski-Harabasz index indicates that clusters are dense and well-separated (Geekforgeeks, 2025)	Shows a lower score compared to K-Means++, suggesting poorer clustering performance

Conclusion
Overall, despite scoring lower on the Calinski-Harabas index, AHC is able to achieve a higher Silhouette score and lower Davies-Bouldin index, as compared to K-Means++. This suggests that AHC is able to group the data points in the Diabetes Prediction Dataset into compact and well-separated clusters.

4.3.2. Body Fat Prediction Dataset




```
K++ | Silhouette score: 0.464, Davies-Bouldin score: 0.789, Calinski-Harabasz score: 278.622
AHC | Silhouette score: 0.449, Davies-Bouldin score: 0.805, Calinksi-Harabasz score: 268.084
```

Figure 80: Comparison of Clustering Performance Metrics between K-Means++ vs AHC

Silhouette Score	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows a slightly higher Silhouette score compared to AHC, suggesting better performance as higher Silhouette score indicates better clustering of data points	Shows a slightly lower score compared to K-means++, suggesting poorer clustering performance

Davies-Bouldin Index	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows slightly lower score, suggesting better performance as lower Davies-Bouldin index indicates well-separated and compact clusters	Shows a slightly higher score compared to K-means++, suggesting poorer performance

Calinski-Harabasz Index	
K-Means++	**Agglomerative Hierarchical Clustering**
Shows a higher score, suggesting better clustering performance as higher Calinksi-Harabasz index indicates well-separated clusters	Shows a lower score, suggesting poorer performance

Conclusion
Overall, the score metrics for both K-means++ and AHC on the Body Fat Dataset are

relatively similar, which can suggest similar clustering qualities. However, K-means++ is still able to perform slightly better as indicated by its higher Silhouette score, lower Davies-Bouldin index, and higher Calinski-Harabasz index as compared to AHC. Hence, K-means++ is marginally better at clustering data points in the Body Fat Dataset.

4.4. K-Means vs Gaussian Mixed Model

4.4.1. Diabetes Prediction Dataset

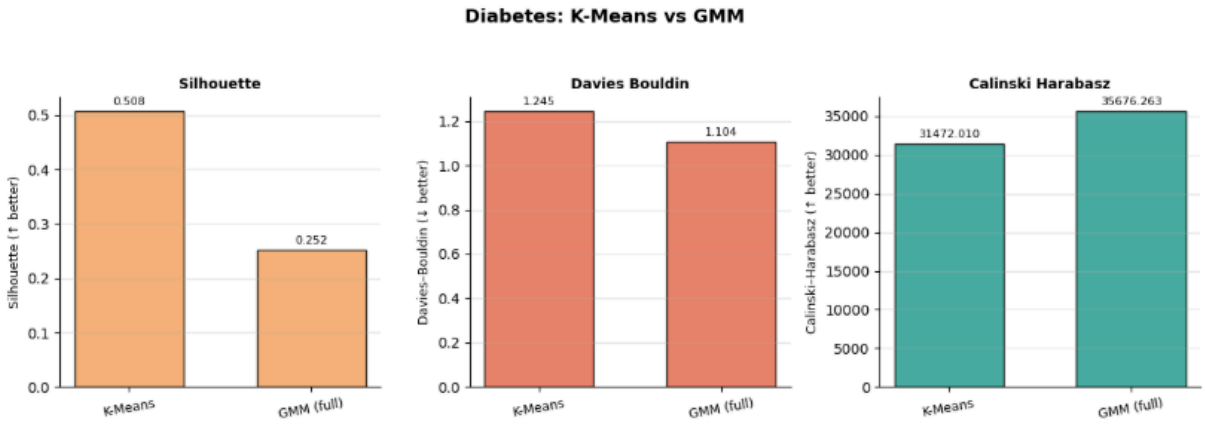


Figure 81: Comparison of Clustering Performance Metrics between K-Means and GMM

Silhouette Score	
K-Means	**Gaussian Mixed Model**
Performs better with a higher Silhouette score compared to Gaussian Mixed Model A higher Silhouette Score indicates that clusters are more tightly grouped and well-separated	Shows a significantly lower score, meaning the clusters formed have more overlaps

Davies-Bouldin Index	
K-Means	**Gaussian Mixed Model**
Both methods yield similar indexes. This suggests that the clusters have similar compression and separation between one another.	

Calinski-Harabasz Index	
K-Means	**Gaussian Mixed Model**
Both methods have similar scores for the Calinski-Harabasz Index. This indicates that both methods perform similarly in terms of maximising the distance between clusters while minimising the variance within each cluster.	

Conclusion
K-Means outperformed the Gaussian Mixed Model in terms of Silhouette score, while achieving similar results for both Davies-Bouldin and Calinksi-Harabasz Indexes. Thus, we can infer that K-Means forms more compact and well-defined clusters on the Diabetes Prediction Dataset.

4.4.2. World Bank Dataset

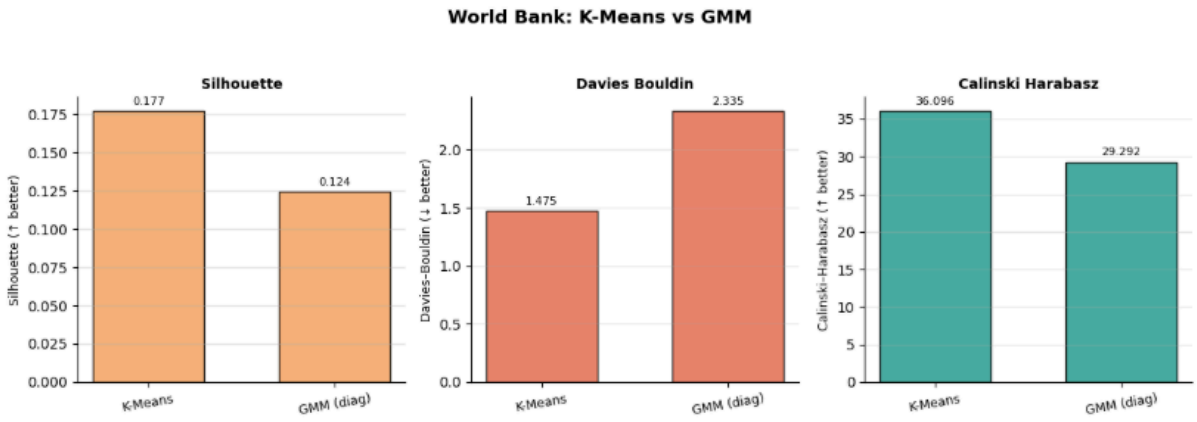


Figure 82: Comparison of Clustering Performance Metrics between K-Means and GMM

Silhouette Score	
K-Means	**Gaussian Mixed Model**
Performs better with a higher Silhouette score compared to Gaussian Mixed Model A higher Silhouette Score indicates that clusters are more tightly grouped and well-separated	Shows a significantly lower score, meaning the clusters formed have more overlaps

Davies-Bouldin Index	
K-Means	**Gaussian Mixed Model**
Performs better with a lower Davies-Bouldin Index compared to Gaussian Mixed Model, indicating that the clusters are more compressed with better separation between one another	Has a higher Davies-Bouldin Index, suggesting poorer cluster separation and larger cluster overlap

Calinski-Harabasz Index	
K-Means	**Gaussian Mixed Model**
Performs better with a higher index compared to Gaussian Mixed Model A higher Silhouette Score suggests clusters have less variance and each centroid is far away from the rest	Shows a noticeably lower index, meaning the clusters formed have high variance and large overlaps

Conclusion
K-Means outperformed the Gaussian Mixed Model in terms of Silhouette score, Davies-Bouldin Index and Calinksi-Harabasz Index. Thus, we can infer that K-Means forms more compact and well-defined clusters, with greater distances between the centroids on the World Bank Dataset.

5. Conclusion

5.1. Pros and Cons

5.1.1. K-Means

Pros: K-Means is simple to implement and scales well to large datasets. It can be customised to different distance metrics and tends to perform well with clusters that are roughly spherical.

Cons: K-Means is sensitive to outliers and is dependent on the initial centroid positions. It also assumes spherical cluster shapes, which can lead to poor results when clusters are non-spherical.

5.1.2. K-Means++

Pros: K-Means++ tends to choose initial centroids that improve clustering quality over standard K-Means when initialisation is non-random, often achieving faster convergence.

Cons: Similar to K-Means, it is vulnerable to outliers and assumes spherical clusters, which can degrade performance on non-spherical data. In addition, it demands more resources for selecting the initial centroids.

5.1.3. Agglomerative Hierarchical Clustering (AHC)

Pros: AHC does not require predefining the number of clusters. A dendrogram can visualise the clustering process and clearly show how clusters relate to one another.

Cons: AHC is computationally demanding, which limits its use with large datasets. As the number of clusters grows, interpreting the dendrogram can become more challenging.

5.1.4. Gaussian Mixture Model (GMM)

Pros: GMM can represent elliptical cluster shapes, allowing it to capture more complex structures. Data points may belong to multiple clusters with varying probabilities, offering flexibility and a closer match to real-world data. Outliers tend to be handled more elegantly, assigned with lower probabilities

Cons: GMM assumes that the data follows a Gaussian distribution, which may not necessarily hold true. It can converge to local minima if the initialisation is poor. It is also computationally intensive, especially with high-dimensional data.

5.2. Use Cases

K-Means proved to perform better than Agglomerative Hierarchical Clustering on the Diabetes Prediction and Body Fat Prediction datasets. It also rivaled Gaussian Mixture Model on the Diabetes Prediction and World Bank datasets. Our experiments indicated that applying PCA helps alleviate the curse of dimensionality in K-Means, as reducing input dimensions prior to clustering enhanced model performance in each case. Different distance metrics were also explored, with cosine distance demonstrating superior performance Euclidean distance across all datasets, resulting in more distinct and compact cluster formations.

K-Means++ outperformed K-Means on the Diabetes Prediction and World Bank datasets and Agglomerative Hierarchical Clustering on the Body Fat Prediction Dataset. This indicates that optimised and more informed centroid initialization can greatly improve clustering performance. Similar to K-Means, data scaling and the application of PCA contributed to achieving superior results.

Agglomerative Hierarchical Clustering (AHC) produced coherent clusters across the Diabetes Prediction and Body Fat Prediction datasets. Notably, it surpassed K-Means++ on the Diabetes Prediction dataset, suggesting that the data contained an underlying hierarchical structure effectively captured by AHC. Moreover, the dendrograms facilitated the visualisation of data groupings by similarity, leading to distinct and easily interpretable clusters.

Gaussian Mixture Model (GMM) performed decently on the Diabetes Prediction dataset, particularly in identifying patients with varying factors that lead to diabetes. Its strength lies in its ability to handle complex, overlapping clusters as it does not enforce strict cluster assignments and is able to detect subtle fluctuations in its predictor variables such as Blood Glucose level. This makes GMM a decent model for the Diabetes Prediction dataset as the risk of getting diabetes is often probabilistic rather than distinct in nature.

6. Datasets

Fedesoriano. (n.d.). Body Fat Prediction Dataset. Kaggle.
https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset

Mohit, B. (n.d.). World Bank Dataset. Kaggle.
https://www.kaggle.com/datasets/bhadramohit/world-bank-dataset

Mustafa, M. (n.d.). Diabetes prediction dataset. Kaggle.
https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset

7. References

Angelou, R. (2024, November 26). *Understanding Silhouette Score in Clustering | by FARSHAD K | Medium*. FARSHAD K. Retrieved October 18, 2025, from https://farshadabdulazeez.medium.com/understanding-silhouette-score-in-clustering-8aedc06ce9c4

Borst, H., & Gregory, N. (2025, August 26). *Your Guide To Body Fat Percentage – Forbes Health*. Forbes. Retrieved October 17, 2025, from https://www.forbes.com/health/wellness/body-fat-percentage/

Frees, D. (2023, August 30). *Scaling Agglomerative Clustering for Big Data*. towards data science. https://towardsdatascience.com/scaling-agglomerative-clustering-for-big-data-an-introduction-to-rac-fb26a6b326ad/#:~:text=Agglomerative%20clustering%20has%20a%20terrible,implemented%20with%20a%20min%2Dheap

Geekforgeeks. (2025, June 23). *What is Silhouette Score?* GeeksforGeeks. Retrieved October 18, 2025, from https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/

Geekforgeeks. (2025, July 12). *Types of Linkages in Hierarchical Clustering*. GeeksforGeeks. Retrieved October 17, 2025, from https://www.geeksforgeeks.org/machine-learning/ml-types-of-linkages-in-clustering/

Geekforgeeks. (2025, July 23). *Calinski-Harabasz Index – Cluster Validity indices | Set 3*. GeeksforGeeks. Retrieved October 18, 2025, from https://www.geeksforgeeks.org/machine-learning/calinski-harabasz-index-cluster-validity-indices-set-3/

Geekforgeeks. (2025, July 23). *Davies-Bouldin Index*. GeeksforGeeks. Retrieved October 18, 2025, from https://www.geeksforgeeks.org/machine-learning/davies-bouldin-index/

Gere, A. (2023). Recommendations for validating hierarchical clustering in consumer sensory projects. *ScienceDirect*, *6*.

Piech, C. (2012). *K Means*. Standford CS221. https://stanford.edu/~cpiech/cs221/handouts/kmeans.html