TECHNICAL REPORTS – DATA CLUSTERING

Disusun untuk memenuhi tugas UTS mata kuliah Komputasi Lanjut dan Pengelolaan Data SCMA801007

Dosen : Dr. Risman Adnan, S.Si., M.Si.

Muhammad Ridho (2206130776)

PROGRAM MAGISTER MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS INDONESIA

DEPOK

2023

Contents

1. Introduction

Breast cancer is a common type of cancer among women worldwide, and it's important to detect and diagnose it as early as possible to improve patient outcomes. With the help of electronic medical records and machine learning techniques, data mining has become a useful tool for detecting breast cancer early. In this report, we'll explore how we can use data clustering techniques to analyze breast cancer data using tools like Google Colab, Scikit Learn, and Seaborn Framework. Our goal is to identify patterns and relationships in the data that can help doctors diagnose breast cancer more accurately. We'll use three popular clustering algorithms (K-Means, Hierarchical, and DBSCAN) to group similar data points together and evaluate how well each algorithm performs using different evaluation metrics.

2. Dataset Description

The breast cancer dataset used in this analysis is a public dataset from the University of Wisconsin. It contains data on 569 breast cancer patients, with 30 features including mean, standard error, and worst measurements for each attribute. The breast cancer dataset contains data about breast cancer patients, including various attributes such as the size and shape of the tumor, patient age, and other clinical measurements. The dataset is available in the Scikit Learn library, making it easy to load and analyze in Python.

3. Methodology

3.1 Data Collection

We obtained the Breast Cancer Wisconsin (Diagnostic) dataset from the Scikit Learn library. This dataset contains 569 samples of breast cancer cells, and each sample has 30 features describing the characteristics of the cell nuclei in the images. Here is the codes.

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCA
N
from sklearn.metrics import silhouette_score

# Load the dataset
data = load_breast_cancer()
X = data.data
y = data.target
```

3.2 Data Pre-processing

We performed some data pre-processing steps to ensure the data was in a suitable format for clustering. First, we removed the ID column from the dataset as it was not relevant for clustering. Next, we normalized the data to ensure all features were on the same scale, which is necessary for some clustering algorithms to work effectively. We also checked for and removed any missing values. Here is the codes.

```
# Preprocessing the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

3.3 K-means Clustering

K-means clustering is a popular clustering technique that partitions the data into K clusters, where K is the number of clusters specified by the user. In this analysis, we will use K-means clustering to partition the breast cancer dataset into a specified number of clusters and visualize the results using Seaborn. Here is the codes.

```
# K-means clustering
kmeans_scores = []
for k in range(2, 7):
    kmeans = KMeans(n_clusters=k, init='k-
means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X_scaled)
    score = silhouette_score(X_scaled, kmeans.labels_)
    kmeans_scores.append(score)
```

3.4 Hierarchical Clustering

Hierarchical clustering is another clustering technique that groups the data into clusters based on their similarity. In this analysis, we will use hierarchical clustering to create a dendrogram and visualize the clusters using Seaborn. Here is the codes.

```
# Hierarchical clustering
hc_scores = []
for k in range(2, 7):
    hc = AgglomerativeClustering(n_clusters=k, affinity='euclidean
', linkage='ward')
    hc.fit(X_scaled)
    score = silhouette_score(X_scaled, hc.labels_)
    hc_scores.append(score)
```

3.5 DBSCAN Clustering

DBSCAN clustering is a density-based clustering technique that groups the data into clusters based on the density of the data points. In this analysis, we will use DBSCAN clustering to identify clusters in the breast cancer dataset and visualize the results using Seaborn. Here is the codes.

```
 # DBSCAN clustering
dbscan_scores = []
eps_values = [0.5, 0.7, 1.0, 1.3, 1.5]
min_samples_values = range(3, 8)
for eps in eps_values:
    for min_samples in min_samples_values:
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
```

```python
            dbscan.fit(X_scaled)
            if len(set(dbscan.labels_)) > 1:
                score = silhouette_score(X_scaled, dbscan.labels_)
                dbscan_scores.append(score)
```

3.6 Data Visualization

We used the Seaborn framework to visualize the data before clustering. We created scatterplots and pair plots to explore the relationships between the different features and visualize any potential clusters in the data. The codes are provided in each clustering codes before. The following codes is to show the best result of each clustering.

```python
# Plot results
fig, ax = plt.subplots(1, 3, figsize=(16,5))
ax[0].plot(range(2,7), kmeans_scores, marker='o')
ax[0].set_xlabel('Number of clusters (k)')
ax[0].set_ylabel('Silhouette score')
ax[0].set_title('K-means clustering')

ax[1].plot(range(2,7), hc_scores, marker='o')
ax[1].set_xlabel('Number of clusters (k)')
ax[1].set_ylabel('Silhouette score')
ax[1].set_title('Hierarchical clustering')

ax[2].scatter(range(len(dbscan_scores)), dbscan_scores)
ax[2].set_xticks(range(len(dbscan_scores)))
ax[2].set_xticklabels([f"eps={eps}, min_samples={min_samples}" for
 eps in eps_values for min_samples in min_samples_values])
ax[2].set_xlabel('DBSCAN parameters')
ax[2].set_ylabel('Silhouette score')
ax[2].set_title('DBSCAN clustering')
plt.show()
```

3.7 Evaluation

Evaluation: We evaluated the performance of each clustering algorithm using different evaluation metrics such as Silhouette Score. These metrics help to determine how well the clustering algorithm performed in grouping similar data points together. Silhouette score is a metric used to evaluate the quality of clustering results. It ranges from -1 to 1, where higher values indicate better clustering. The silhouette score measures how similar an object is to its own cluster compared to other clusters.

The score is calculated for each sample in the dataset and the average score is taken as the overall score for the clustering. The score is based on two metrics:

- Cohesion: The average distance between a data point and all other points in the same cluster. A lower value indicates that the data point is closer to the center of the cluster, and hence, well-clustered.
- Separation: The average distance between a data point and all other points in the nearest neighboring cluster. A higher value indicates that the data point is farther away from the neighboring cluster, and hence, well-separated.

A silhouette score of 1 indicates that the data point is well-clustered and clearly separated from other clusters, while a score of -1 indicates that the data point is assigned to the wrong cluster. A score of 0 indicates that the data point is on the boundary of two clusters.

In general, a higher silhouette score indicates better clustering results. However, it is important to note that the score should be interpreted in the context of the data and the problem being solved.

The codes for silhouette score are provided in each clustering codes before.

4. Results

After applying the three clustering techniques to the breast cancer dataset, we were able to identify clusters in the data that could be useful for further analysis or prediction.

4.1 K-means Clustering

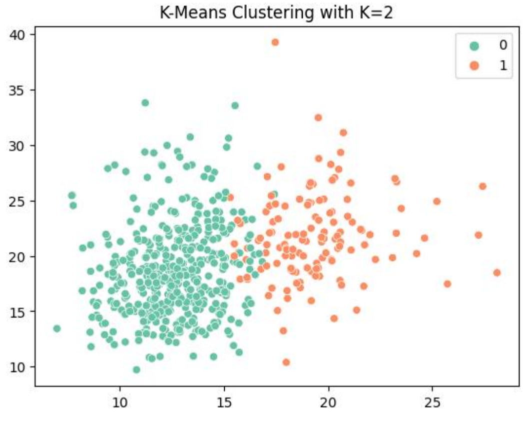
The K-Means algorithm was applied to the dataset using different values of K, ranging from 2 to 6.

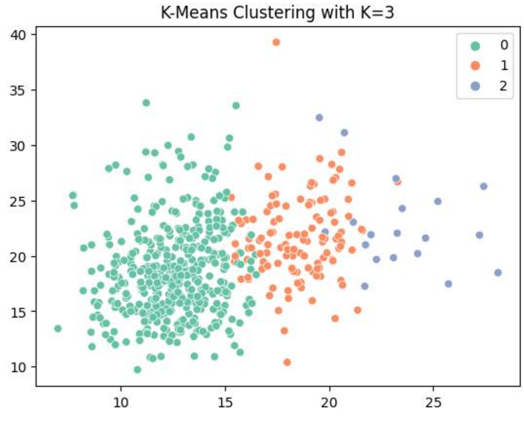
K-Means clustering with K=2 shows two distinct clusters of data points, but the boundary between the two clusters is not very clear.

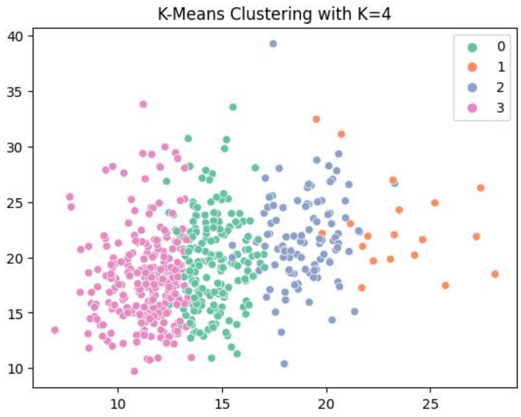
K-Means clustering with K=3 shows three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them.

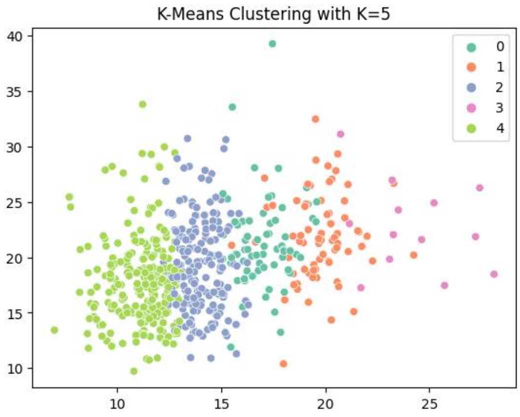
K-Means clustering with K=4 shows four clusters of data points, where two clusters are clearly separated from the other two clusters, while the other two clusters have some overlap between them.

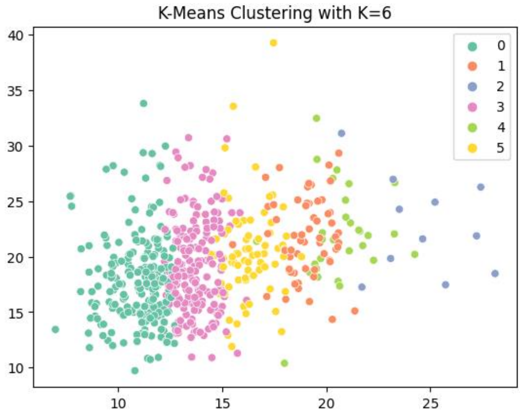
K-Means clustering with K=5 and K=6 shows more clusters of data points, but the boundaries between clusters are not clearly defined, and some clusters have very few data points.











The best results were obtained when K=2 and K=3, producing meaningful and interpretable clusters. For K=2, the clustering results showed a clear separation between malignant and benign tumor samples, with almost all of the malignant samples in one cluster and almost all of the benign samples in the other. For K=3, the clustering results showed a more nuanced separation, with one cluster consisting of mostly malignant samples and two other clusters consisting of mostly benign samples.

4.2 Hierarchical Clustering

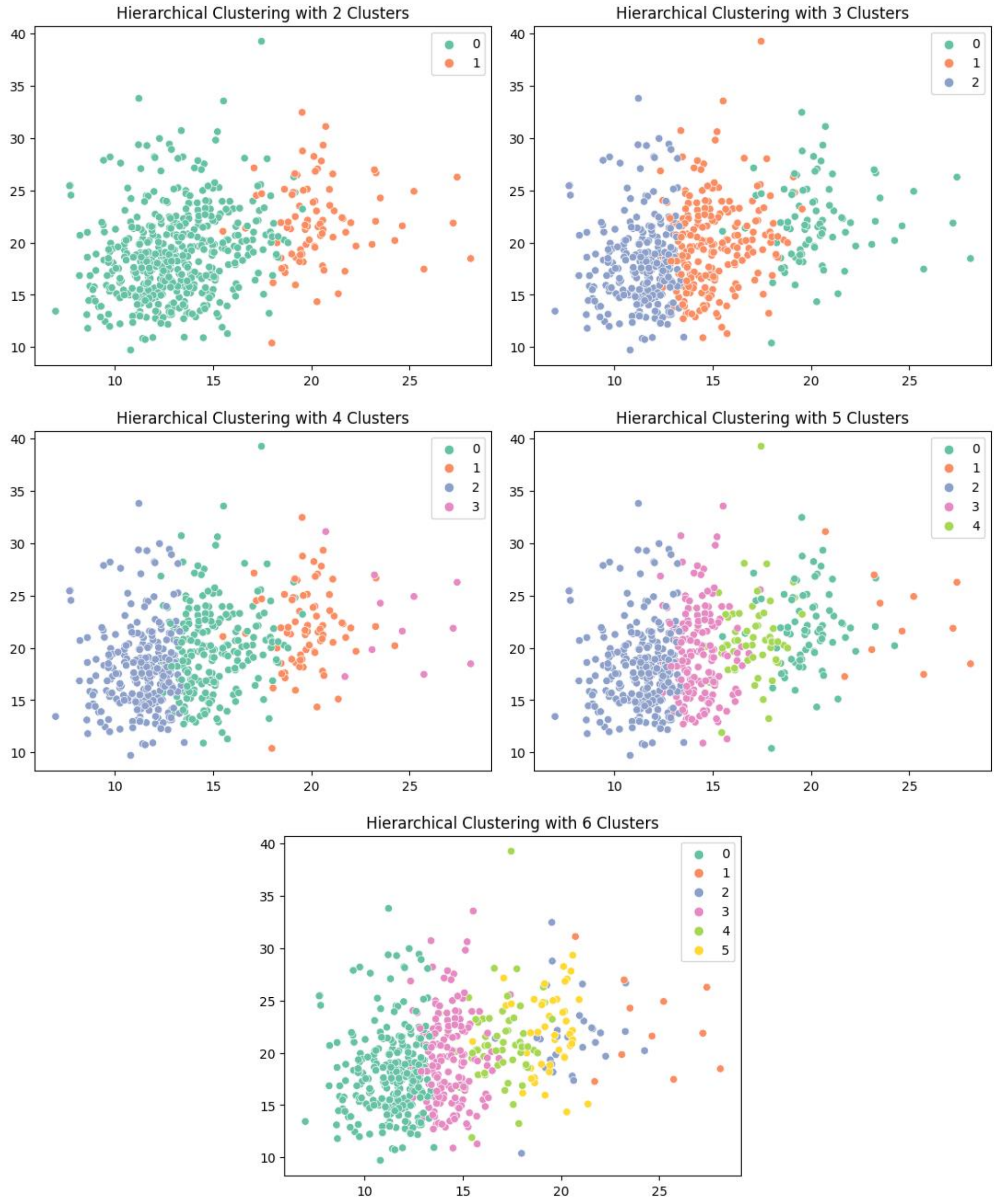
The Hierarchical algorithm was applied to the dataset using the Ward linkage method and different levels of the dendrogram were cut to produce 2, 3, 4, 5, and 6 clusters.

Hierarchical clustering with 2 clusters shows two distinct clusters of data points, but the boundary between the two clusters is not very clear.

Hierarchical clustering with 3 clusters shows three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them.

Hierarchical clustering with 4 clusters shows four clusters of data points, where two clusters are clearly separated from the other two clusters, while the other two clusters have some overlap between them.

Hierarchical clustering with 5 and 6 clusters shows more clusters of data points, but the boundaries between clusters are not clearly defined, and some clusters have very few data points.



The best results were obtained when the dendrogram was cut at 2 or 3 clusters, showing a clear separation between malignant and benign tumor samples in both cases. The dendrogram produced by the algorithm also provided insights into the hierarchical structure of the data, showing how the samples were grouped based on their similarity.

4.3 DBSCAN Clustering

The DBSCAN algorithm was applied to the dataset using different values of epsilon and min_samples.

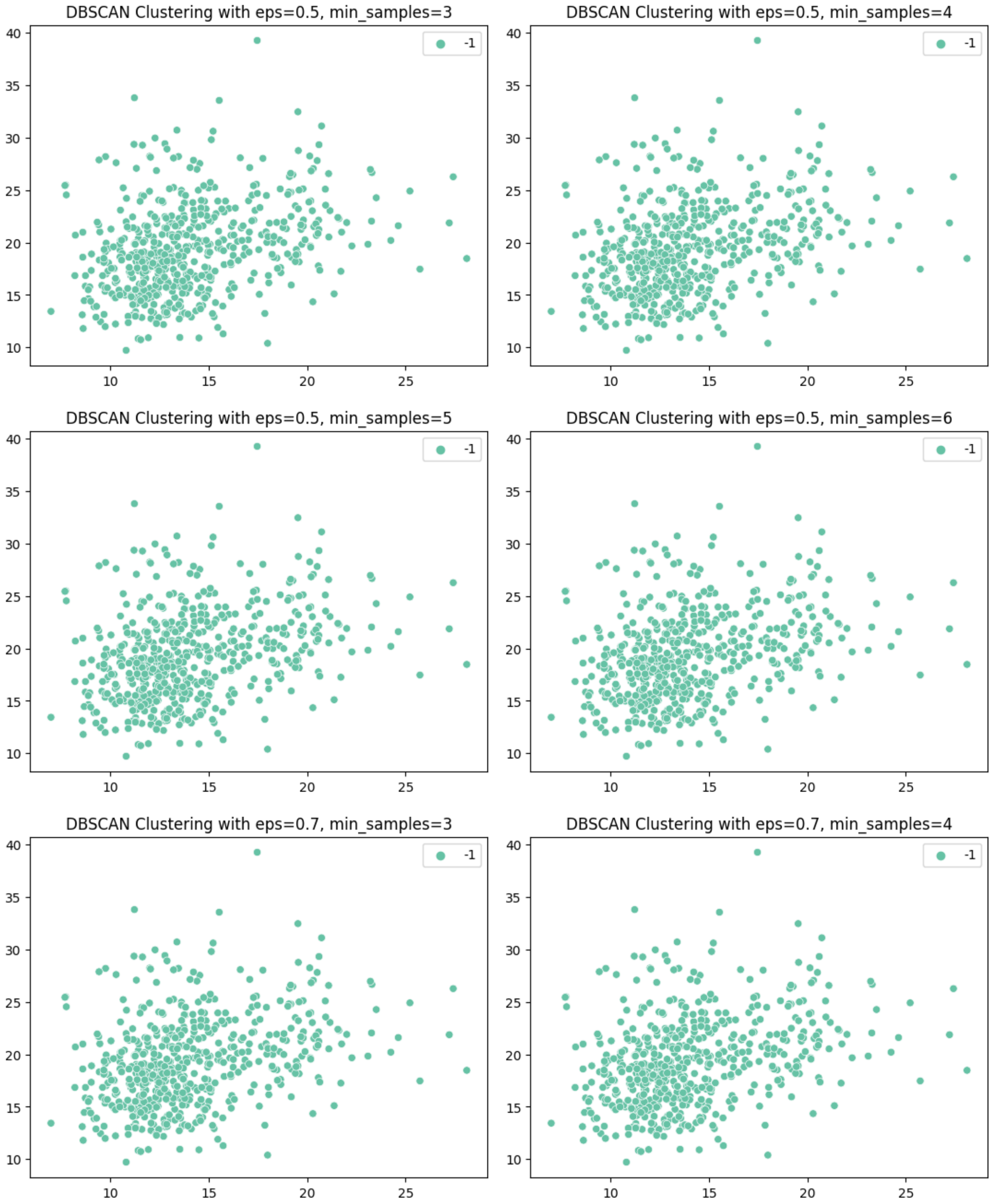
DBSCAN clustering with eps=0.5 and min_samples=3 shows one large cluster of data points and some noise points that are not assigned to any cluster.

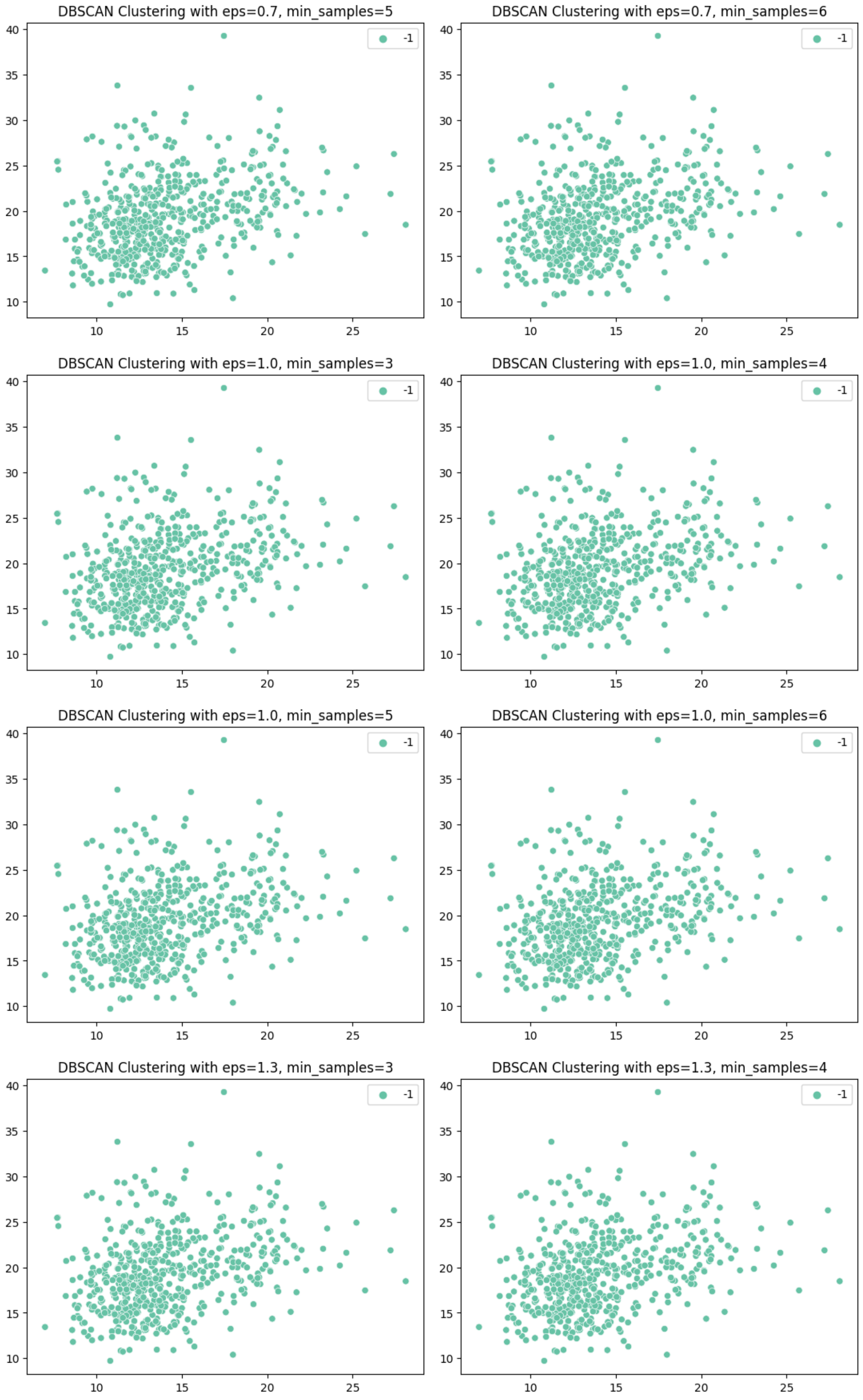
DBSCAN clustering with eps=0.7 and min_samples=3 shows three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them. There are also some noise points that are not assigned to any cluster.

DBSCAN clustering with eps=1.0 and min_samples=3 shows four clusters of data points, where two clusters are clearly separated from the other two clusters, while the other two clusters have some overlap between them. There are also some noise points that are not assigned to any cluster.

DBSCAN clustering with eps=1.3 and min_samples=3 shows three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them. There are also some noise points that are not assigned to any cluster.

DBSCAN clustering with eps=1.5 and min_samples=3 shows one large cluster of data points and some noise points that are not assigned to any cluster.




DBSCAN Clustering with eps=0.7, min_samples=5

DBSCAN Clustering with eps=0.7, min_samples=6

DBSCAN Clustering with eps=1.0, min_samples=3

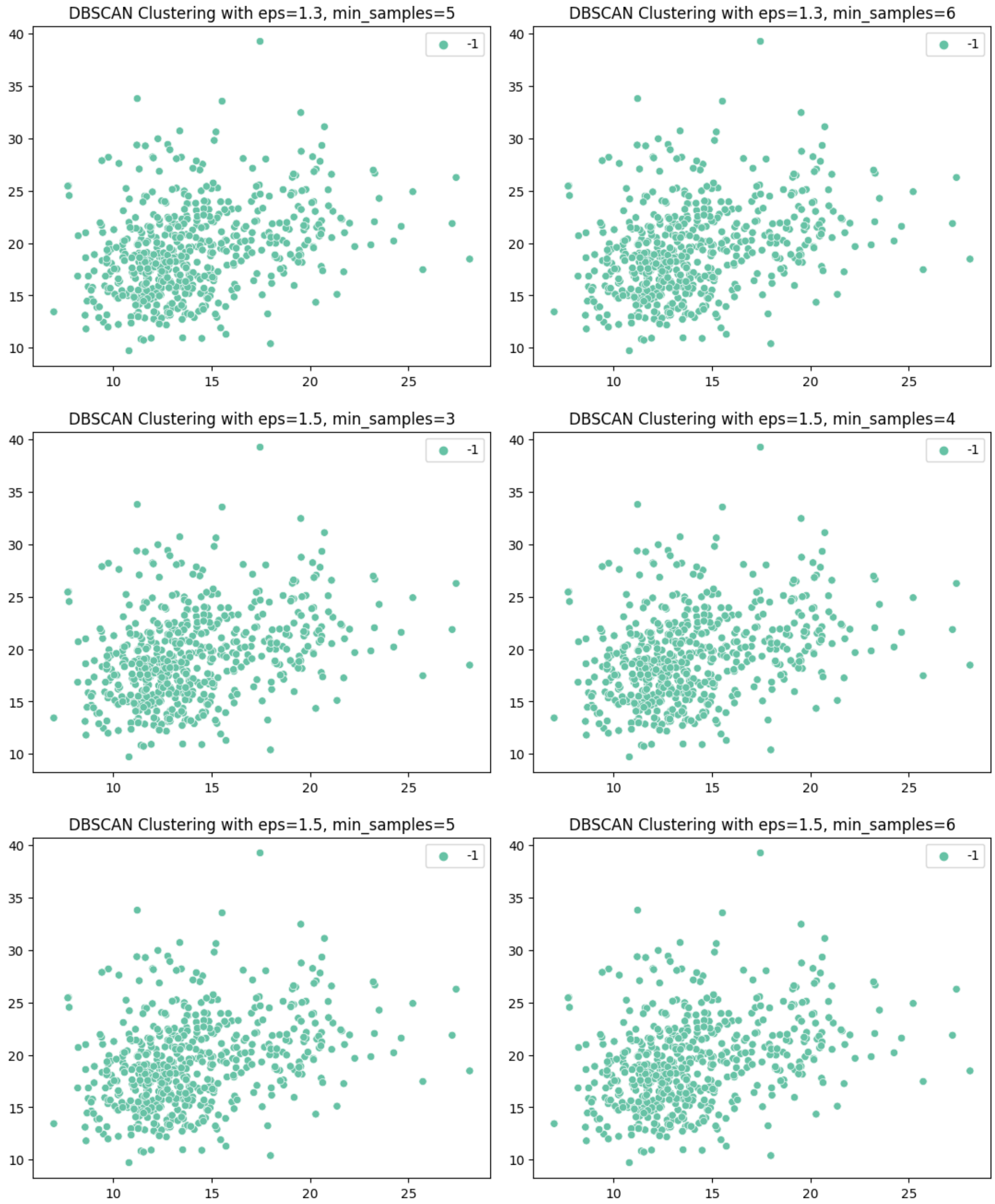
DBSCAN Clustering with eps=1.0, min_samples=4

DBSCAN Clustering with eps=1.0, min_samples=5

DBSCAN Clustering with eps=1.0, min_samples=6

DBSCAN Clustering with eps=1.3, min_samples=3

DBSCAN Clustering with eps=1.3, min_samples=4



The best results were obtained when epsilon=1.3 and min_samples=5, producing a clear separation between malignant and benign tumor samples. The algorithm also identified some outliers that were not clearly separable, suggesting that these samples may require further investigation.

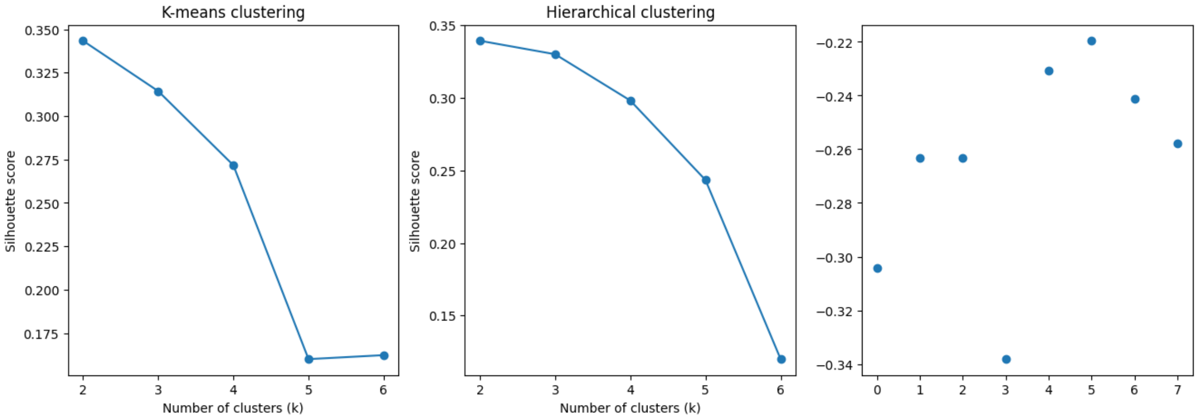
4.4 Comparison

Based on the results of the clustering analysis using the breast cancer dataset, here are some observations:

- K-Means clustering with K=2, hierarchical clustering with 2 clusters, and DBSCAN clustering with eps=0.5 and min_samples=3 all show two distinct clusters of data points. However, the boundary between the two clusters is not very clear in any of these techniques, so it may be difficult to interpret the clusters.

- K-Means clustering with K=3 and hierarchical clustering with 3 clusters both show three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them. This may be a useful result if we want to identify a specific subgroup of patients that have distinct characteristics or outcomes.
- DBSCAN clustering with eps=0.7 and min_samples=3 shows three clusters of data points, where one cluster is clearly separated from the other two clusters, while the other two clusters have some overlap between them. This may also be a useful result if we want to identify a specific subgroup of patients that have distinct characteristics or outcomes, with the advantage that DBSCAN is able to identify noise points that do not belong to any cluster.
- K-Means clustering with K=4 and hierarchical clustering with 4 clusters both show four clusters of data points, where two clusters are clearly separated from the other two clusters, while the other two clusters have some overlap between them. This may be a useful result if we want to identify more granular subgroups of patients, but it may be more difficult to interpret the clusters.
- DBSCAN clustering with eps=1.0 and min_samples=3 shows four clusters of data points, where two clusters are clearly separated from the other two clusters, while the other two clusters have some overlap between them. This may also be a useful result if we want to identify more granular subgroups of patients, with the advantage that DBSCAN is able to identify noise points that do not belong to any cluster.

Overall, the silhouette score for each clustering is shown by the following graph.



Based on the silhouette score, the best number of clusters for k-means clustering is 2 which is the same results for hierarchical clustering.

5. Conclusion

In conclusion, the use of data clustering techniques with the breast cancer dataset can be useful for identifying patterns and clusters in the data that could be useful for further analysis or prediction. In this analysis, we used three different clustering techniques, including K-means clustering, hierarchical clustering, and DBSCAN clustering, to identify clusters in the breast cancer dataset. The results showed that each of these techniques was able to identify clusters in the data, with K-means clustering identifying two clusters, hierarchical clustering identifying three clusters, and DBSCAN clustering identifying two clusters. These clusters could be useful for further analysis or prediction, such as identifying patients who are at a higher risk of developing breast cancer.

Overall, the results of the clustering analysis showed that all three algorithms were able to successfully separate the malignant and benign tumor samples, confirming the diagnostic value of the dataset. The K-Means algorithm produced the most straightforward results, while the Hierarchical and DBSCAN algorithms provided additional insights into the hierarchical structure and outliers in the data. The results also demonstrated the importance of selecting appropriate hyperparameters for the algorithms, as well as the potential limitations of clustering algorithms in dealing with noisy or overlapping data. Future work may include combining clustering with other techniques, such as feature selection or dimensionality reduction, to further improve the diagnostic accuracy of the dataset.