



جامعة الأمير سطام بن عبدالعزيز
PRINCE SATTAM BIN ABDULAZIZ UNIVERSITY

Data Mining Project

K means Clustering Algorithm

PRESENT BY:

ID	Name

Supervisor:

Table of Contents

Introduction:	2
Supervised learning:	2
Unsupervised Learning:	2
K-Means clustering algorithm:	2
Advantages of k-means Clustering:	3
Disadvantages of K-Means Clustering:	4
Application of K-means Clustering:	5
About Database:	6
Database Source:	7
Project code:	7
Output:	10
Elbow Method:	10
K-means Clustering of Breast Cancer Dataset:	11
Feature Distribution for Each Cluster:	12
Show Mean values of each feature for each cluster:	13
Conclusion:	13
References:	14

Introduction:

Data mining is the method of uncovering valuable and concealed information or data from extensive datasets. The obtained data has the potential to enhance unsupervised clustering, a process that categorizes similar objects into clusters with minimal distance between them by excluding unsuitable data objects. Data mining encompasses six fundamental tasks: Anomaly detection, Association rule learning, Clustering, Classification, Regression, and Report generation. Clustering, in particular, stands out as a crucial aspect of data mining, involving the unsupervised classification of data objects into distinct groups or clusters.

Supervised learning:

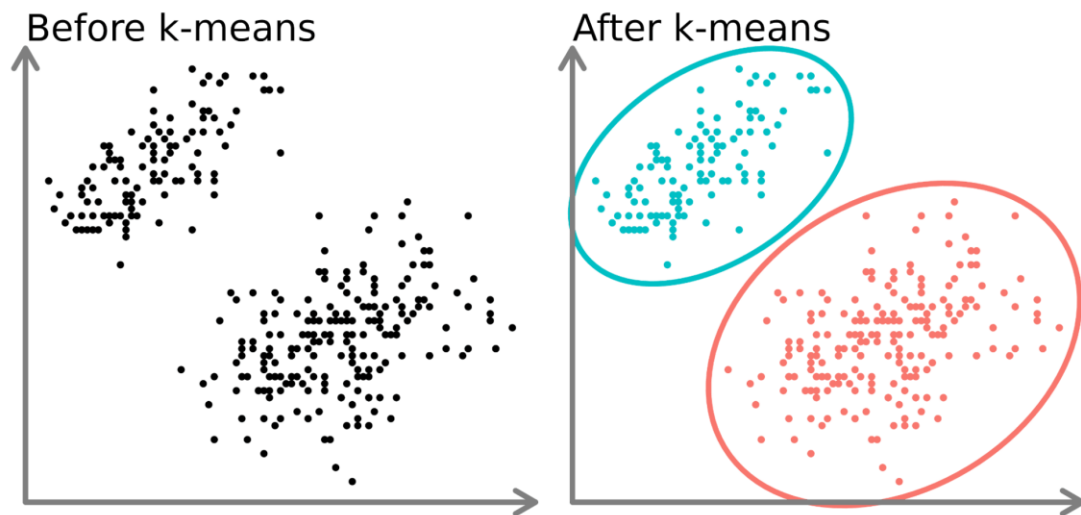
In supervised learning, both input data and corresponding desired outcomes are utilized for training. This approach is efficient and accurate, as it involves providing known target results as inputs to the model during the learning process. Examples of supervised models include neural networks, multilayer perceptrons, and decision trees.

Unsupervised Learning:

The model is not guided by correct results during the entire training process. Instead, it can be employed to categorize input data into groups based solely on their inherent probabilistic properties. Unsupervised models encompass various types such as clustering, amplitude and normalization, k-means, and self-organizing maps.

K-Means clustering algorithm:

K-means is an unsupervised learning method characterized by an iterative process wherein the dataset is partitioned into k predefined, non-overlapping clusters or subgroups. The objective is to enhance the similarity of data points within each cluster while maintaining distinct separation between clusters. This is achieved by allocating data points to clusters so that the sum of squared distances between the cluster centroid and data points is minimized. The cluster centroid, in this context, represents the arithmetic mean of the data points belonging to the cluster.



Advantages of k-means Clustering:

The k-means clustering algorithm offers several advantages:

1. **Simplicity and Ease of Implementation:** K-means is straightforward and relatively easy to implement. The algorithm's simplicity makes it accessible for a wide range of users.
2. **Efficiency:** K-means is computationally efficient, especially with large datasets. It converges quickly, making it suitable for scenarios where efficiency is crucial.
3. **Scalability:** The algorithm can handle large datasets and scales well as the size of the dataset increases.
4. **Versatility:** K-means can be applied to various types of data, making it a versatile clustering algorithm. It is effective in identifying clusters in different domains.
5. **Quantifiable Results:** The algorithm produces quantifiable and easy-to-interpret results. Each data point is assigned to a specific cluster, providing a clear categorization.
6. **Empirical Success:** K-means is widely used and has demonstrated success in various applications, including image segmentation, customer segmentation, and document clustering.
7. **Initial Centroid Flexibility:** Users have the flexibility to choose the initial centroids or allow the algorithm to randomly initialize them. This flexibility can impact the convergence speed and final results.

8. **Applicability to Different Distance Metrics:** K-means can accommodate different distance metrics, allowing users to tailor the algorithm to the specific characteristics of their data.
9. **Linear Time Complexity:** The time complexity of k-means is generally linear with respect to the number of data points, making it efficient for large datasets.
10. **Well-Suited for Spherical Clusters:** K-means performs well when clusters are spherical and equally sized. It tends to struggle with clusters of irregular shapes or varying sizes.

Disadvantages of K-Means Clustering:

Despite its advantages, k-means clustering comes with several disadvantages:

1. **Sensitivity to Initial Centroids:** K-means is sensitive to the initial placement of centroids. Different initializations can lead to different final cluster assignments, impacting the algorithm's reliability.
2. **Dependence on the Number of Clusters (k):** The user must specify the number of clusters (k) in advance. Choosing an inappropriate value for k can result in suboptimal clustering. Various techniques exist to estimate an optimal k, but the problem remains challenging.
3. **Assumption of Spherical Clusters:** K-means assumes that clusters are spherical and equally sized. It struggles with clusters of different shapes, densities, or orientations, leading to suboptimal results in such cases.
4. **Difficulty with Outliers:** Outliers can significantly impact the centroids' positions, affecting the entire clustering structure. K-means is not robust to outliers, and their presence can distort cluster boundaries.
5. **Unsuitability for Non-Numeric Data:** K-means relies on distance measures, making it less suitable for categorical or non-numeric data. Preprocessing steps, such as converting categorical variables, may be necessary.
6. **Equal Variances Across Clusters:** The algorithm assumes that clusters have roughly equal variances. If variances differ significantly, especially in high-dimensional spaces, it can lead to biased cluster assignments.
7. **Inability to Handle Noisy Data and Missing Values:** K-means is sensitive to noisy data and may produce suboptimal results when dealing with datasets containing noise. It also struggles when faced with missing values.

8. **Fixed and Circular Decision Boundaries:** The decision boundaries created by k-means are fixed and circular, limiting the algorithm's ability to capture complex, non-linear relationships in the data.
9. **Global Optimum Not Guaranteed:** K-means is prone to converging to local optima. The algorithm's final result depends on the initial centroids, and it may not always find the global optimum.
10. **Scalability Issues with Large Datasets:** While k-means is generally efficient, it may face scalability issues with extremely large datasets. Other clustering algorithms, like mini-batch k-means, may be more suitable for such cases.

Application of K-means Clustering:

K-means clustering finds applications across various domains due to its simplicity and efficiency in grouping similar data points. Some common applications include:

1. **Image Segmentation:** K-means is used for segmenting images into distinct regions based on pixel similarity. This is valuable in computer vision applications, object recognition, and medical image analysis.
2. **Customer Segmentation:** Businesses use k-means to segment customers into groups based on purchasing behavior, demographics, or other relevant features. This helps tailor marketing strategies and improve customer satisfaction.
3. **Anomaly Detection:** K-means can be employed to identify anomalous patterns in data by clustering normal data points and identifying outliers or data points that deviate from the established clusters.
4. **Document Clustering:** In natural language processing, k-means can cluster documents based on their content. This is useful in organizing large document collections, topic modeling, and information retrieval.
5. **Genetic Data Analysis:** K-means is applied in genetic research to cluster genes or genetic markers based on similarities in expression patterns. This aids in understanding genetic relationships and identifying potential biomarkers.
6. **Network Security:** K-means can be used to detect unusual patterns or behaviors in network traffic. By clustering normal network behavior, deviations from established clusters can indicate potential security threats or anomalies.
7. **Recommendation Systems:** E-commerce and streaming platforms leverage k-means to cluster users with similar preferences. This helps in building

recommendation systems that suggest products or content based on the behavior of similar users.

8. **Climate Pattern Analysis:** K-means is utilized in climate science to analyze patterns in climate data. It can cluster regions with similar climate characteristics, aiding in the study of climate variability and trends.
9. **Speech Recognition:** K-means clustering is applied in speech processing to categorize phonemes or acoustic features. This is crucial in developing accurate speech recognition systems.
10. **Manufacturing Quality Control:** In manufacturing, k-means can be used to group products based on quality attributes. This facilitates quality control by identifying clusters of products with similar characteristics.
11. **Financial Fraud Detection:** K-means helps detect unusual patterns in financial transactions that may indicate fraudulent activity. By clustering normal transaction behavior, deviations can be flagged for further investigation.
12. **Healthcare Data Analysis:** In healthcare, k-means is applied to cluster patients based on health-related features. This assists in personalized medicine, disease diagnosis, and treatment planning.
13. **Astronomical Data Analysis:** K-means clustering is used in astronomy to group celestial objects based on their observable characteristics. This aids in categorizing stars, galaxies, or other astronomical entities.


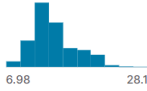
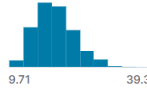
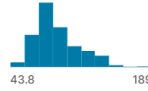
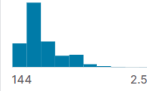
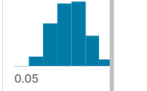
These applications highlight the versatility of k-means clustering across different domains, making it a widely adopted algorithm for various data analysis tasks.

About Database:

The dataset encapsulates features related to breast cancer characteristics, with the target variable being the diagnosis categorized into Malignant or Benign cases. Sourced from [Kaggle](<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>), this dataset provides a comprehensive foundation for our analytical endeavors.

data.csv (125.2 kB) Download Fullscreen Close

Detail Compact Column 10 of 32 columns

# id	diagnosis	# radius_mean	# texture_mean	# perimeter_mean	# area_mean	# smoothness_r
ID number	The diagnosis of breast tissues (M = malignant, B = benign)	mean of distances from center to points on the perimeter	standard deviation of gray-scale values	mean size of the core tumor		mean of local vari radius lengths
	B 63% M 37%					
8670 911m		6.98 28.1	9.71 39.3	43.8 189	144 2.5k	0.05
842382	M	17.99	18.38	122.8	1001	0.1184
842517	M	20.57	17.77	132.9	1326	0.08474
84300903	M	19.69	21.25	130	1203	0.1096
84348301	M	11.42	20.38	77.58	386.1	0.1425
84358402	M	20.29	14.34	135.1	1297	0.1003
843786	M	12.45	15.7	82.57	477.1	0.1278

Database Source:

<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data/data>

Project code:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.metrics import silhouette_score,
classification_report, confusion_matrix
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Load Breast Cancer dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/breast-cancer-wisconsin/wdbc.data'
column_names = ['ID', 'Diagnosis', 'Mean Radius', 'Mean Texture',
'Mean Perimeter', 'Mean Area', 'Mean Smoothness',
'Mean Compactness', 'Mean Concavity', 'Mean
Concave Points', 'Mean Symmetry', 'Mean Fractal Dimension',
'SE Radius', 'SE Texture', 'SE Perimeter', 'SE
Area', 'SE Smoothness', 'SE Compactness', 'SE Concavity',
'SE Concave Points', 'SE Symmetry', 'SE Fractal
Dimension', 'Worst Radius', 'Worst Texture',
'Worst Perimeter', 'Worst Area', 'Worst
Smoothness', 'Worst Compactness', 'Worst Concavity',
```



```

        'Worst Concave Points', 'Worst Symmetry', 'Worst
Fractal Dimension']
breast_cancer_data = pd.read_csv(url, names=column_names,
header=None)

# Extract features (excluding the 'Diagnosis' column)
features = breast_cancer_data.drop(['Diagnosis', 'ID'], axis=1)

# Impute missing values (replace NaN with mean)
imputer = SimpleImputer(strategy='mean')
features_imputed = imputer.fit_transform(features)

# Standardize the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features_imputed)

# Find the optimal number of clusters using the Elbow method
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300,
n_init=10, random_state=0)
    kmeans.fit(features_scaled)
    wcss.append(kmeans.inertia_)

# Plot the Elbow method
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Choose the optimal number of clusters (elbow point)
optimal_clusters = 2

# Apply KMeans with the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++',
max_iter=300, n_init=10, random_state=0)
kmeans.fit(features_scaled)

# Add the cluster labels to the dataset
breast_cancer_data['Cluster'] = kmeans.labels_

# Split the data into training and testing sets
X_train, X_test, y_train, y_test =
train_test_split(features_scaled,
breast_cancer_data['Diagnosis'], test_size=0.2, random_state=0)

```

```

# Train a classification model (Logistic Regression as an
example)
model = LogisticRegression()
model.fit(X_train, y_train)

# Visualize the clusters using PCA for dimensionality reduction
pca = PCA(n_components=2)
features_pca = pca.fit_transform(features_scaled)

# Plot the clusters in 2D
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.scatter(features_pca[:, 0], features_pca[:, 1],
            c=kmeans.labels_, cmap='viridis')
plt.title('K-means Clustering of Breast Cancer Dataset')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')

# Plot the distribution of each feature for each cluster
plt.subplot(1, 2, 2)
sns.boxplot(x='Cluster', y='Mean Radius',
            data=breast_cancer_data)
sns.boxplot(x='Cluster', y='Mean Texture',
            data=breast_cancer_data)
sns.boxplot(x='Cluster', y='Mean Perimeter',
            data=breast_cancer_data)
sns.boxplot(x='Cluster', y='Mean Area', data=breast_cancer_data)
plt.title('Feature Distribution for Each Cluster')
plt.show()

# Check silhouette score for cluster quality evaluation
silhouette_avg = silhouette_score(features_scaled,
                                   kmeans.labels_)
print(f"\nSilhouette Score: {silhouette_avg}")

# Display the counts of each cluster
cluster_counts =
breast_cancer_data['Cluster'].value_counts().sort_index()
print("\nCounts of each cluster:")
print(cluster_counts)

# Display the mean values of each feature for each cluster
features_mean = breast_cancer_data.groupby('Cluster').mean()
print("\nMean values of each feature for each cluster:")
print(features_mean)

# Convert the 'Cluster' column to string

```

```

breast_cancer_data['Cluster'] =
breast_cancer_data['Cluster'].astype(str)

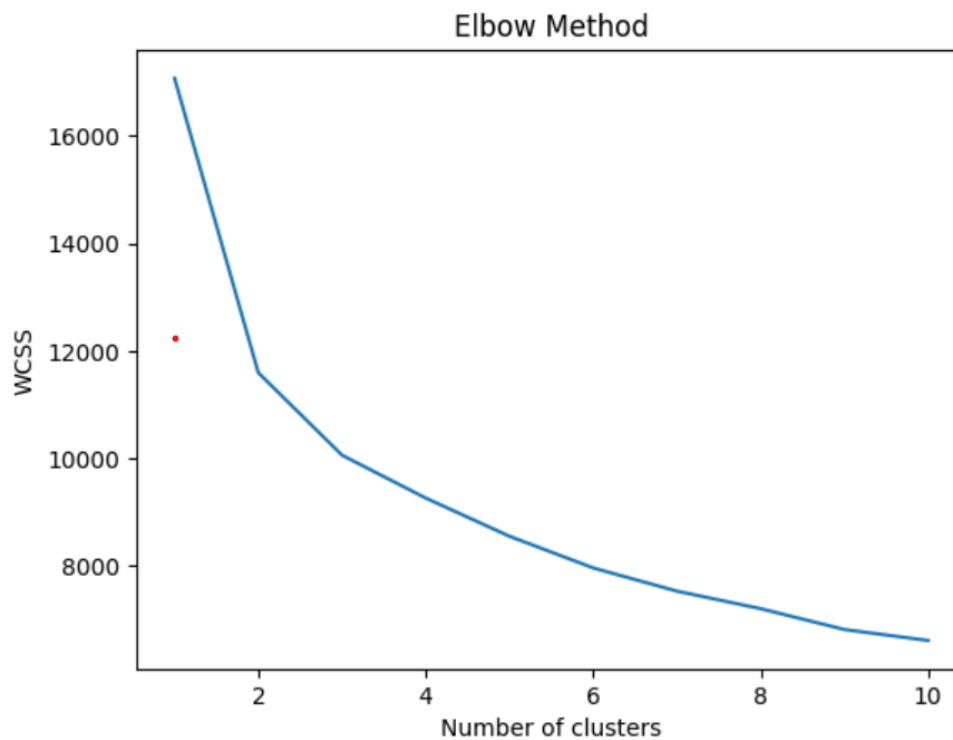
# Classification Report
try:
    classification_rep = classification_report(y_test,
model.predict(X_test))
    print("\nClassification Report:\n", classification_rep)
except KeyError as e:
    print(f"Error: {e}. Make sure the column names are correct.")

# Confusion Matrix
try:
    conf_matrix = confusion_matrix(y_test, model.predict(X_test))
    print("\nConfusion Matrix:\n", conf_matrix)
except KeyError as e:
    print(f"Error: {e}. Make sure the column names are correct.")

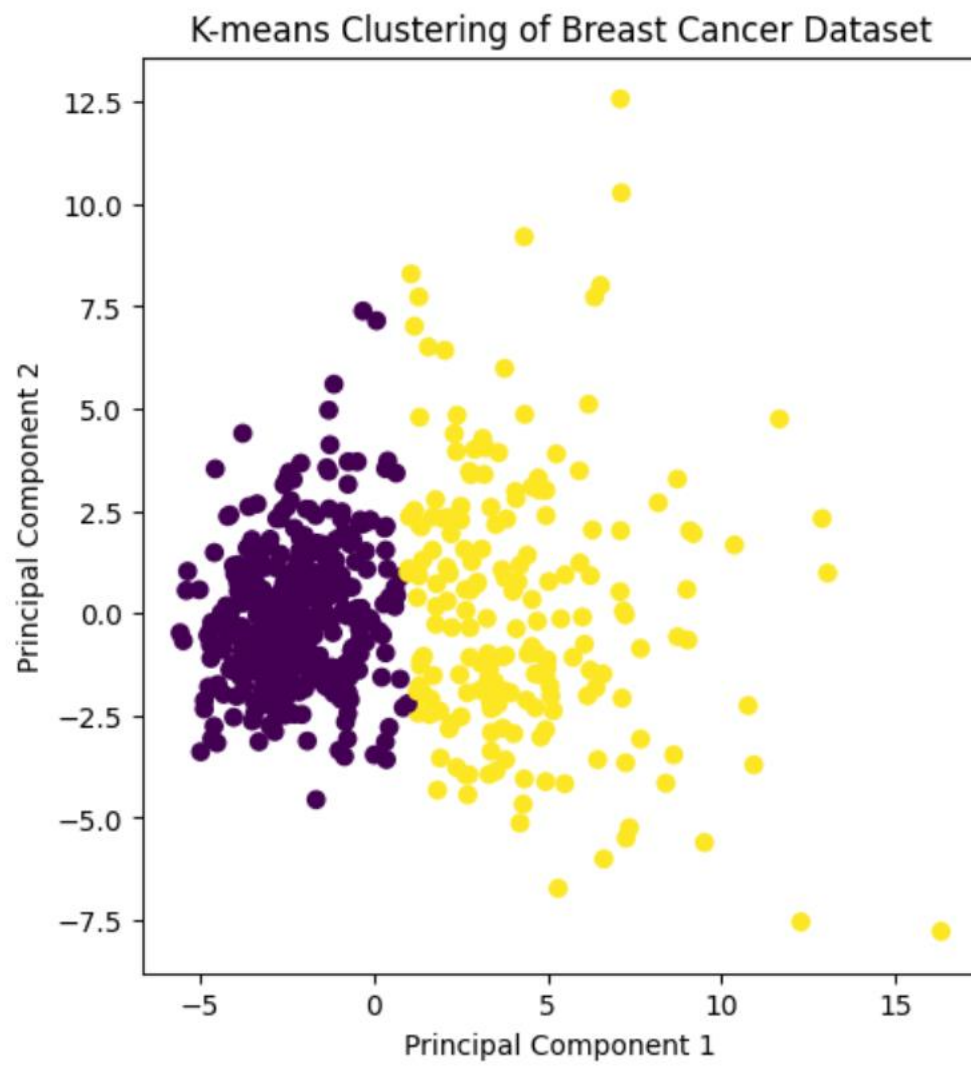
```

Output:

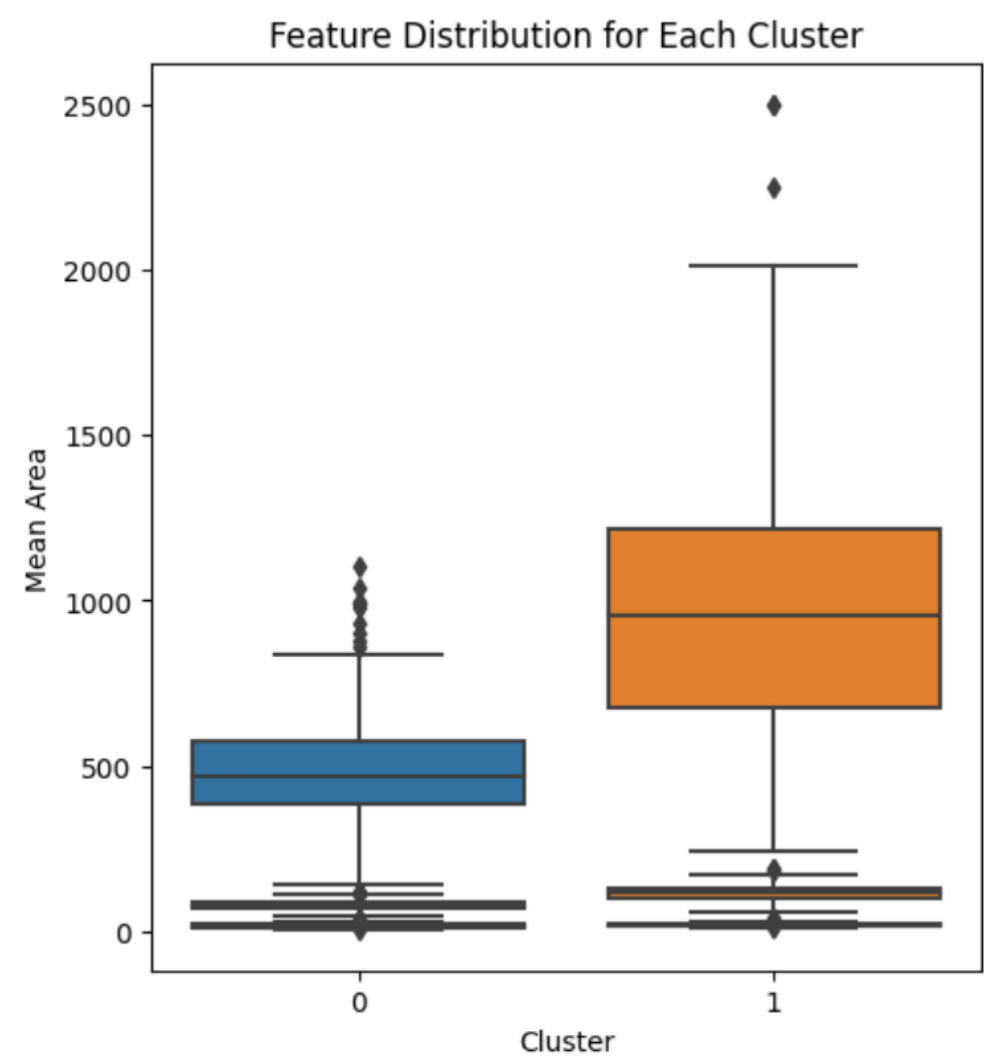
Elbow Method:



K-means Clustering of Breast Cancer Dataset:



Feature Distribution for Each Cluster:



Show Mean values of each feature for each cluster:

```
Silhouette Score: 0.34338224069077805

Counts of each cluster:
0    375
1    194
Name: Cluster, dtype: int64

Mean values of each feature for each cluster:
      ID  Mean Radius  Mean Texture  Mean Perimeter  Mean Area \
Cluster
0      2.601300e+07   12.426691    18.262427     79.820053  486.772267
1      3.879741e+07   17.414536    21.275258    115.452887  979.857216

      Mean Smoothness  Mean Compactness  Mean Concavity \
Cluster
0           0.091990           0.076585           0.042679
1           0.104808           0.157993           0.177949

      Mean Concave Points  Mean Symmetry  ...  Worst Radius  Worst Texture \
Cluster
0           0.026155           0.172696  ...    13.768477     24.132933
1           0.092921           0.197526  ...    21.103041     28.662320

      Worst Perimeter  Worst Area  Worst Smoothness  Worst Compactness \
Cluster
0           89.389920   596.520267           0.125158           0.176886
1          141.806237  1429.673711           0.146306           0.403839

      Worst Concavity  Worst Concave Points  Worst Symmetry \
Cluster
0           0.160288           0.076508           0.271324
1           0.488490           0.188249           0.326322

      Worst Fractal Dimension
Cluster
0           0.077869
1           0.095691

[2 rows x 31 columns]
```

Conclusion:

In conclusion, this thorough analysis resides at the crossroads of sophisticated data analytics and pressing healthcare challenges. The integration of clustering and classification techniques enhances our grasp of inherent patterns within breast cancer data. The insights derived have the capacity to make substantial contributions to early diagnosis and well-informed decision-making in clinical practice.

References:

- [1] Bennett, K. P., & Mangasarian, O. L. (1992) Robust Linear Programming Discrimination of Two Linearly Inseparable Sets. Optimization Methods and Software, 1, 23-34.
<http://dx.doi.org/10.1080/10556789208805504>.
- [2] Pedamkar, P. (2023). K- Means Clustering Algorithm. EDUCBA.
<https://www.educba.com/kmeans-clustering-algorithm/>.
- [3] Banoula, M. (2023). K-means Clustering Algorithm: Applications, Types, & How Does It Work? Simplilearn.com. <https://www.simplilearn.com/tutorials/machine-learning-tutorial/kmeans-clustering-algorithm>.
- [4] Breast Cancer Wisconsin (Diagnostic) Data Set. Kaggle.
<https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data/data>.
- [5] K-means Clustering and its applications:
<https://www.linkedin.com/pulse/k-means-clustering-its-applications-ritvik-ranjan/>