SIT720 Machine Learning

Task 4

Michael Rideout
Student Id: 225065259

Item 1.1 Main Points Summary

Week 3 Main Points Summary - Clustering Concepts

Clustering is the method of grouping data points based on a similarity or distance metric / measure.

Distance Metrics

A distance metric measures the similarity or distance between data points. Some distance metrics include:

Manhattan distance: $$d_{Cityblock}(x_i, x_j) = \sum_{k=1}^{D} |x_{i,k} - x_{j,k}|$$

Euclidean distance: $$d_{Euclidean}(x_i, x_j) = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2}$$

Chebyshev distance: $$d_{\text{Chebyshev}}(x_i, x_j) = \max(|x_{i,1} - x_{j,1}|, |x_{i,2} - x_{j,2}|, \ldots, |x_{i,D} - x_{j,D}|)]$$

Minkowski distance: $$d(x, y) = (\sum_{i=0}^{n-1} |x_i - y_i|^p)^{1/p}$$

Cosine distance: $$d_{Cosine}(x_i, x_j) = 1 - \frac{x_i^T x_j}{\|x_i\|_2 \|x_j\|_2}$$

Mahalanobis distance: $$d_{Mahalanobis}(x_i, x_j) = \sqrt{(x_i - x_j)^T M^{-1} (x_i - x_j)}$$

Jaccard distance: $$d_{Jaccard}(x_i, x_j) = 1 - \frac{|x_i \cap x_j|_1}{|x_i \cup x_j|_1}$$

Clustering Algorithms

Clustering algorithms use distance metrics to group together similar datapoints into potentially interesting clusters.
Some clustering algorithms include:

K-means:
Is a clustering algorithm which performs the following steps:

1. Initialise k centroids
2. Assign each datapoint to the closest centroid
3. Recalculate the centroids to be the mean of all data points in each cluster
4. Repeat 2 and 3 until cluster assignments do not change substantially

Cluster Evaluation

- Two categories exist to evaluate clusters:

External Assessment - Compares the assigned cluster to a known ground truth cluster.
Internal Assessment - Evaluates the quality of the clustering based on intrinsic properties of the clusters themselves.

Rand Index - measures how similar two clusters are.
Purity - is a measure of how well a cluster algorithm's output matches some ground truth
Mutual Information - measures the consensus of two clustering assignments
Silhouette Coefficient - Measure the degree of similarity an instance is to its cluster as opposed to other clusters

Limitation of K-means

K-means has several limitations:

1. Results can vary due to random initialisation
2. Number of clusters needs to be specified
3. Has difficulty with arbitrary shapes
4. Sensitive to noisy data

Week 4 Main Points Summary

Eigenvalues and Eigenvectors

Are tools that aid in the investigation of linear transforms. Attributes of them are:

- Defined as pairs (λ, u) satisfying $Au = \lambda u$ for a square matrix A
- A d x d matrix has d eigenvalue/eigenvector pairs
- The number of non-zero eigenvalues equals the matrix rank
- Eigenvectors form an orthogonal matrix U.
- Finding eigenvalues involves solving the characteristic polynomial $\det(A - \lambda I) = 0$
- Eigenvectors are found by solving $(A - \lambda I)u = 0$ for each eigenvalue.

Singular Value Decomposition

A way to break down a matrix into three matrices
Attributes of the method are:

- Decomposes a matrix X into $X = USV^T$, where U and V are orthogonal matrices and S is a diagonal matrix of singular values.
- Singular values (σ_i) are the square roots of eigenvalues from XX^T or X^TX.
- Eigenvectors of XX^T form U, and eigenvectors of X^TX form V.
- SVD represents data in a coordinate system where the covariance matrix is diagonal

Curse of Dimensionality

Highly dimensional data is common in areas like text, image and genomic data. Increased dimensionality can cause exponential increase in the size of data. In high dimensions, more data points reside near the surface of a hyperplane. Distances between points become less distinct making clustering less effective. Dimensionality reduction aims to mitigate these effects whilst preserving information

Principal Component Analysis (PCA)

The goal of PCA is to summarise correlated high-dimensional data using a smaller set of uncorrelated variables called principal components. These components are linear combinations of the original dimensions, sorted by the amount of variance they capture.

Formulation (Maximising Error)

- Find the direction (eigenvector) that maximises the variance of the project data
- Leads to an eigenvalue $Cu_i = \lambda_i u_i$,
- Subsequent principal components are found similar to the above, maximising variance while being orthogonal to the previous principal components

Formulation (Minimising Error)

An alternate method which minimises the reconstruction error when projecting data onto a lower k-dimensional subspace.

1.2 Summary of Reading List Items

Week 3 Readings

k-means++: The Advantages of Careful Seeding by Arther and Vassilviskii.

Article - k-means++: The Advantages of Careful Seeding

This article discussed improvements to the k-means algorithm. It proposed an enhanced seeding technique that involved choosing initial cluster centers sequentially with specific probabilities related to the points' distances from existing centers.

Video - How Does The DBSCAN Algorithm Work

Video is no longer publicly accessible

Video - Spectral Clustering and How It Works
An introduction to spectral clustering. It is a clustering technique that doesn't assume specific cluster shapes, handles intertwined data well and avoids the iterative process and sensitivity to initialisation.

Week 4 Readings

Video - Eigenvectors and eigenvalues
Video explaining how to find principal components in PCA using linear algebra. This is achieved by finding eigenvalues and eigenvectors of the covariance matrix.

Video - Lecture: The Singular Value Decomposition (SVD)
A lecture on Singular Value Decomposition. It explains the matrix multiplications that fundamentally involve the rotation and stretching of vectors. The method to compute the SVD using eigenvalue decomposition is explained.

Video - StatQuest: Principal Component Analysis (PCA)
A step by step guide to principal component analysis using singular value decomposition. Demonstrates how PCA can reduce the dimensionality of data while retaining important information.

Video - PCA 3: direction of greatest variance
This video explains the significations of the components in PCA, that being the first component is the direction of the greatest variance, the second orthogonal to the first and has the next greatest variance and so on.

Video - PCA 4 : principal components = eigenvectors
Video about PCA explaining that principal components are eigenvectors of the covariance matrix.

Video - finding eigenvalues and eigenvectors
A video about eigenvectors and eigenvalues. The core idea is that eigenvectors are special vectors which, when transformed by a matrix, only scale and don't rotate. It explains how to compute eigenvectors and eigenvalues.

Python Libraries

Python libraries utilised in this task include:

- **Pandas:** Data manipulation and analysis library
- **Numpy:** Library for scientific and numerical computing
- **Mathplotlib:** Visualisation library
- **Seaborn:** Visualisation library
- **Sklearn:** Machine learning toolkit
- **Mplot3d:** Generates 3D plots

- **Yellowbrick:** ML visualisation library
- **SciPy:** Scientific library
- **Kneed:** Library to detect elbow points in a curve

1.3 Learning Reflection

Weeks 3 and 4 have provided a foundational basis for the key machine learning concepts of clustering and dimensionality reduction.

I learnt that clustering the process of grouping similar data points together. From that the focus was on what is ‘similarity’ and that was described by the concept of distance metrics. The k-means clustering algorithm was explained in detail as it is currently the most popular clustering technique. Cluster evaluation was also described in detail as it is important to not only generate clusters, but to know how to compare them to ground truths or determine inherent characteristics of clusters.

The lecture on dimensionality reduction introduced the eigenvalues and eigenvectors and the mathematical underpinnings for these, that being linear algebra. Singular Value Decomposition was also described as a powerful matrix factorisation technique that is useful in uncovering underlying structures in data. Principal Component Analysis was presented as a means to reduce dimensionality by finding new, uncorrelated variables that capture the maximum variance in the data.

1.4 Quiz Results

Week 3 Quiz:

Week 3 quiz SIT 720

Your work has been saved and submitted

Written 18 April, 2025 8:36 AM - 18 April, 2025 8:39 AM • Attempt 2 of unlimited

Your quiz has been submitted successfully, the answer(s) for the following question(s) are incorrect.

Attempt Score	**9 / 10 - 90 %**
Overall Grade (Highest Attempt)	**9 / 10 - 90 %**

Week 4 Quiz:

Week 4 Quiz for SIT720

Your work has been saved and submitted

Written 18 April, 2025 2:47 PM - 18 April, 2025 2:58 PM • Attempt 1 of unlimited

Your quiz has been submitted successfully, the answer(s) for the following question(s) are incorrect.

Attempt Score	**9 / 10 - 90 %**
Overall Grade (Highest Attempt)	**9 / 10 - 90 %**

Task 4 Jupyter Notebook Output

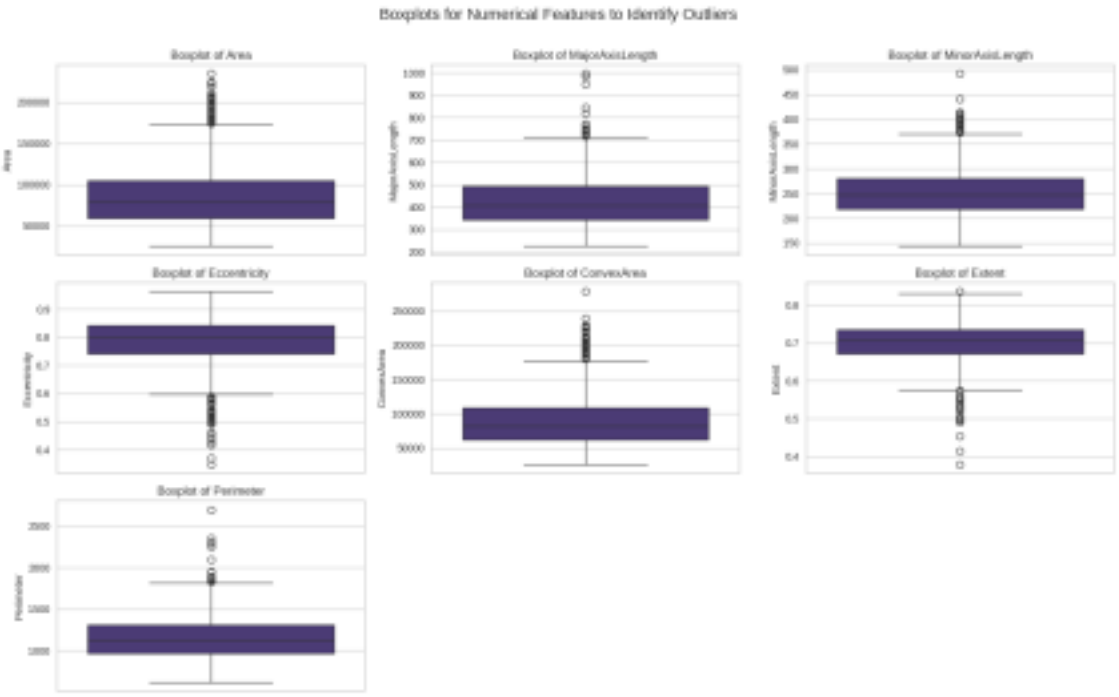
April 18, 2025

1 Task 1 Data Preprocessing and Exploratory Data Analysis

We perform the following steps: 1. Load the dataset (“Dataset.csv”) and verify its integrity. 2. Confirm that there are no missing values. 3. Identify and analyze outliers using visualizations such as boxplots. 4. Visualize feature distributions with histograms and KDE plots to understand the overall distribution of each feature. 5. Review feature statistics (e.g., mean, standard deviation) to get insights into the data. 6. Normalize or standardize the dataset so that all features contribute equally in distance calculations, which is crucial for clustering.

1.0.1 Subtask 1: Load the dataset (“Dataset.csv”) and verify its integrity.

Manual inspection of the dataset determined that there are 900 rows (excluding the header row) and 8 columns. There to satisfy the integrity requirement we take that to mean the row and column counts are equal after the dataframe is loaded.



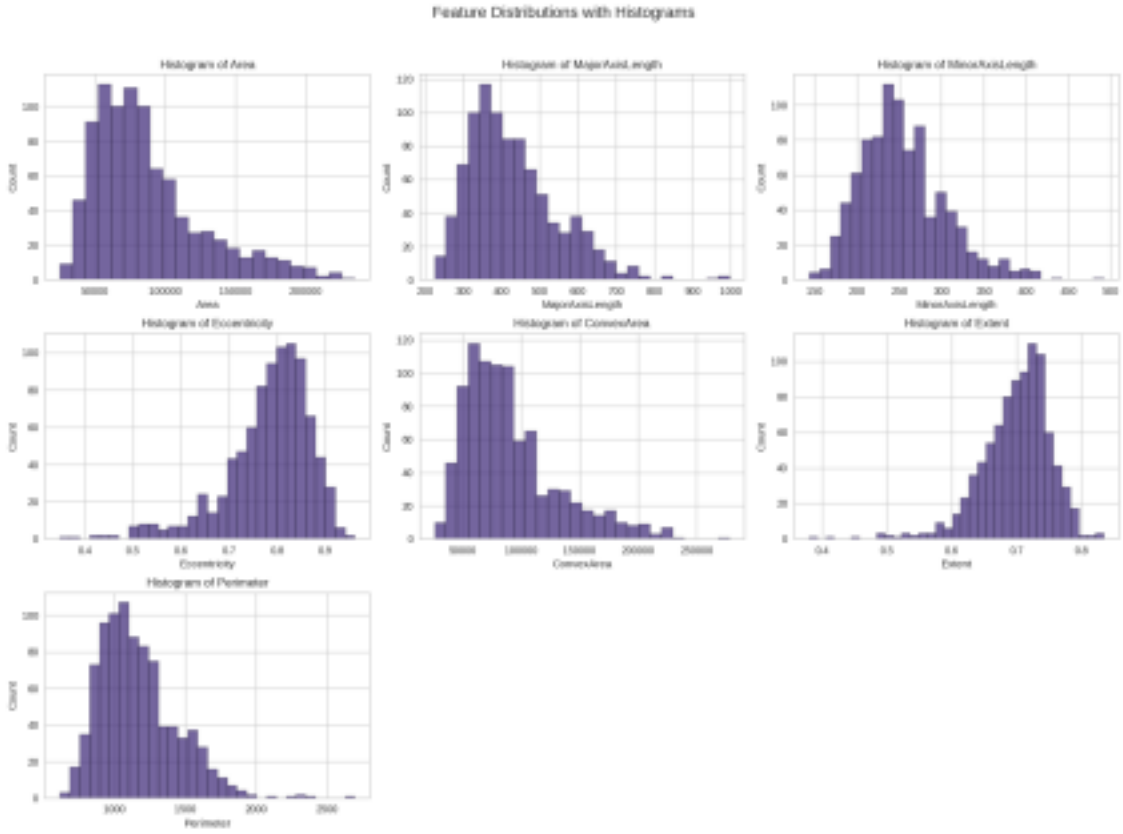
Potential outliers analysis:
Area: 41 outliers detected
- Min Boundary: 175247.00, Max Boundary 235047.00]

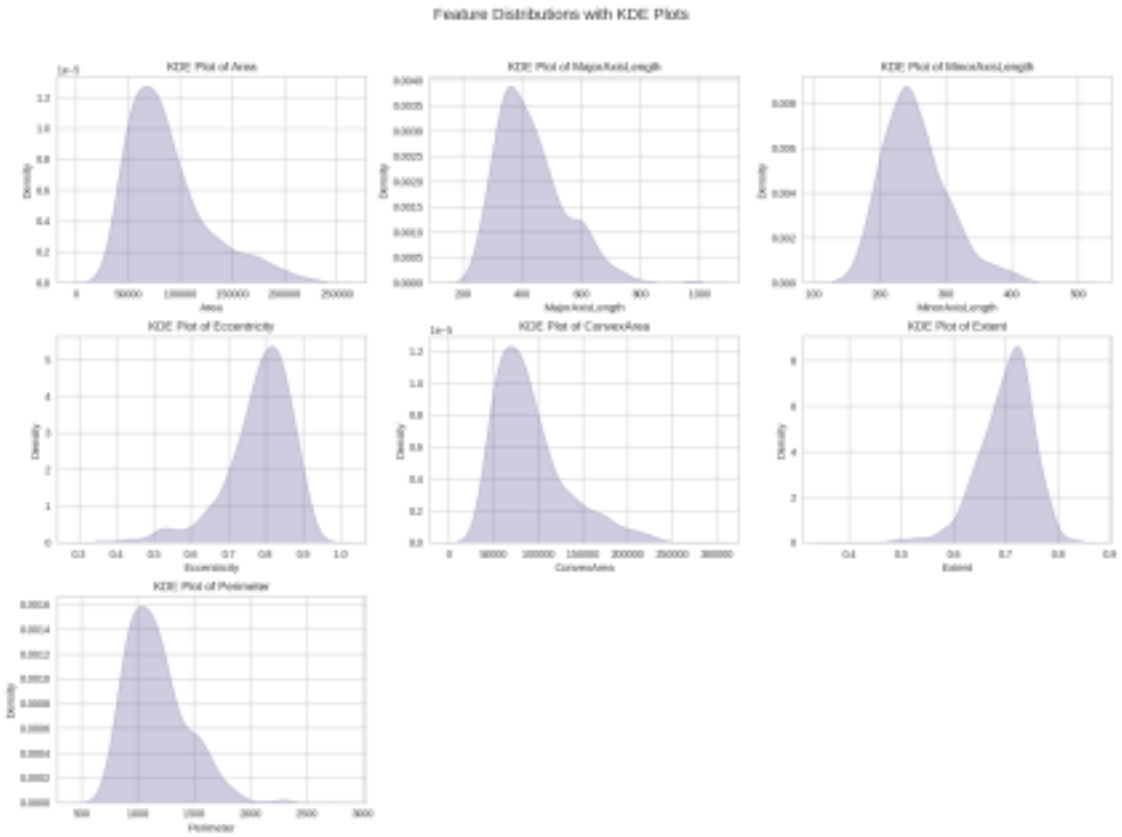
3

MajorAxisLength: 17 outliers detected
- Min Boundary: 719.51, Max Boundary 997.29]
MinorAxisLength: 26 outliers detected
- Min Boundary: 372.85, Max Boundary 492.28]
Eccentricity: 43 outliers detected
- Min Boundary: 0.35, Max Boundary 0.59]
ConvexArea: 42 outliers detected
- Min Boundary: 179961.00, Max Boundary 278217.00]
Extent: 21 outliers detected
- Min Boundary: 0.38, Max Boundary 0.84]
Perimeter: 17 outliers detected
- Min Boundary: 1827.90, Max Boundary 2697.75]

1.0.4 Subtask 4: Visualise feature distributions with histograms and KDE plots to understand the overall distribution of each feature.

Seaborn has differing functions for histograms and KDE plots. Use these.





All features are skewed to either the left or right

1.0.5 Subtask 5 - Review feature statistics (e.g., mean, standard deviation) to get insights into the data.

Basic Statistics for Numerical Features via Pandas Dataframe describe:

	Area	MajorAxisLength	MinorAxisLength	Eccentricity
count	900.000000	900.000000	900.000000	900.000000
mean	87804.127778	430.929950	254.488133	0.781542
std	39002.111390	116.035121	49.988902	0.090318
min	25387.000000	225.629541	143.710872	0.348730
25%	59348.000000	345.442898	219.111126	0.741766
50%	78902.000000	407.803951	247.848409	0.798846
75%	105028.250000	494.187014	279.888575	0.842571
max	235047.000000	997.291941	492.275279	0.962124

	ConvexArea	Extent	Perimeter
count	900.000000	900.000000	900.000000
mean	91186.090000	0.699508	1165.906636
std	40769.290132	0.053468	273.764315
min	26139.000000	0.379856	619.074000
25%	61513.250000	0.670869	966.410750
50%	81651.000000	0.707367	1119.509000
75%	108375.750000	0.734991	1308.389750
max	278217.000000	0.835455	2697.753000

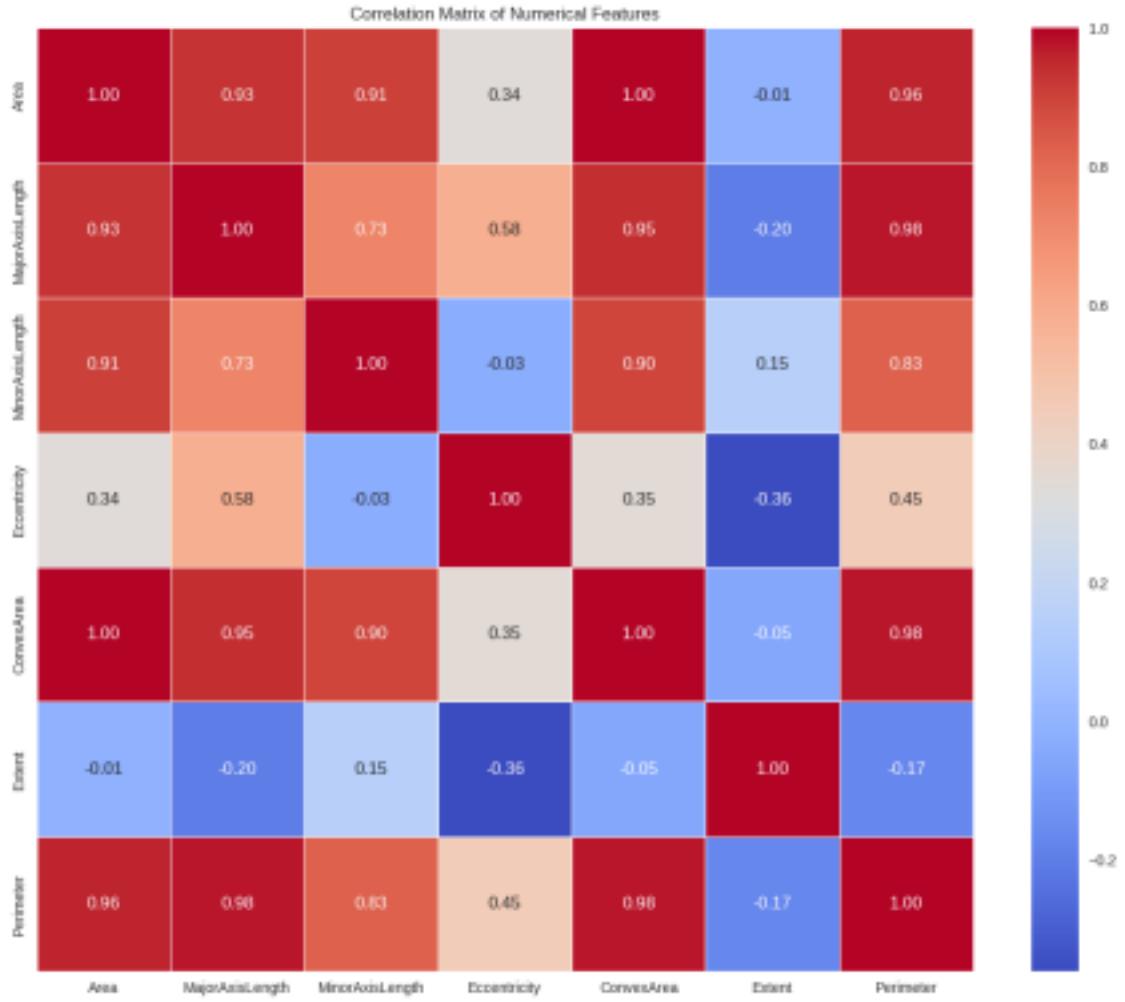
Additional Statistics:

Median Skewness Kurtosis IQR Range
Area 78902.000000 1.175237 1.074073 45680.250000 209660.000000 MajorAxisLength 407.803951 0.989544 1.326808 148.744116 771.662400 MinorAxisLength 247.848409 0.800049 0.953915 60.777448 348.564407 Eccentricity 0.798846 -1.327503 2.492121 0.100805 0.613395 ConvexArea 81651.000000 1.242904 1.427258 46862.500000 252078.000000 Extent 0.707367 -1.151505 3.341384 0.064122 0.455598 Perimeter 1119.509000 1.017761 1.744706 341.979000 2078.679000

Correlation Matrix:

Area MajorAxisLength MinorAxisLength Eccentricity \
Area 1.000000 0.932774 0.906650 0.336107 MajorAxisLength 0.932774 1.000000 0.728030 0.583608 MinorAxisLength 0.906650 0.728030 1.000000 -0.027683 Eccentricity 0.336107 0.583608 -0.027683 1.000000 ConvexArea 0.995920 0.945031 0.895651 0.348210 Extent -0.013499 -0.203866 0.145322 -0.361061 Perimeter 0.961352 0.977978 0.827417 0.447845

ConvexArea Extent Perimeter
Area 0.995920 -0.013499 0.961352
MajorAxisLength 0.945031 -0.203866 0.977978
MinorAxisLength 0.895651 0.145322 0.827417
Eccentricity 0.348210 -0.361061 0.447845
ConvexArea 1.000000 -0.054802 0.976612
Extent -0.054802 1.000000 -0.173449
Perimeter 0.976612 -0.173449 1.000000



It can be seen that lengths and areas are highly correlated, which is expected as area is a function of length.

1.0.6 Subtask 6 - Normalize or standardize the dataset so that all features contribute equally in distance calculations, which is crucial for clustering.

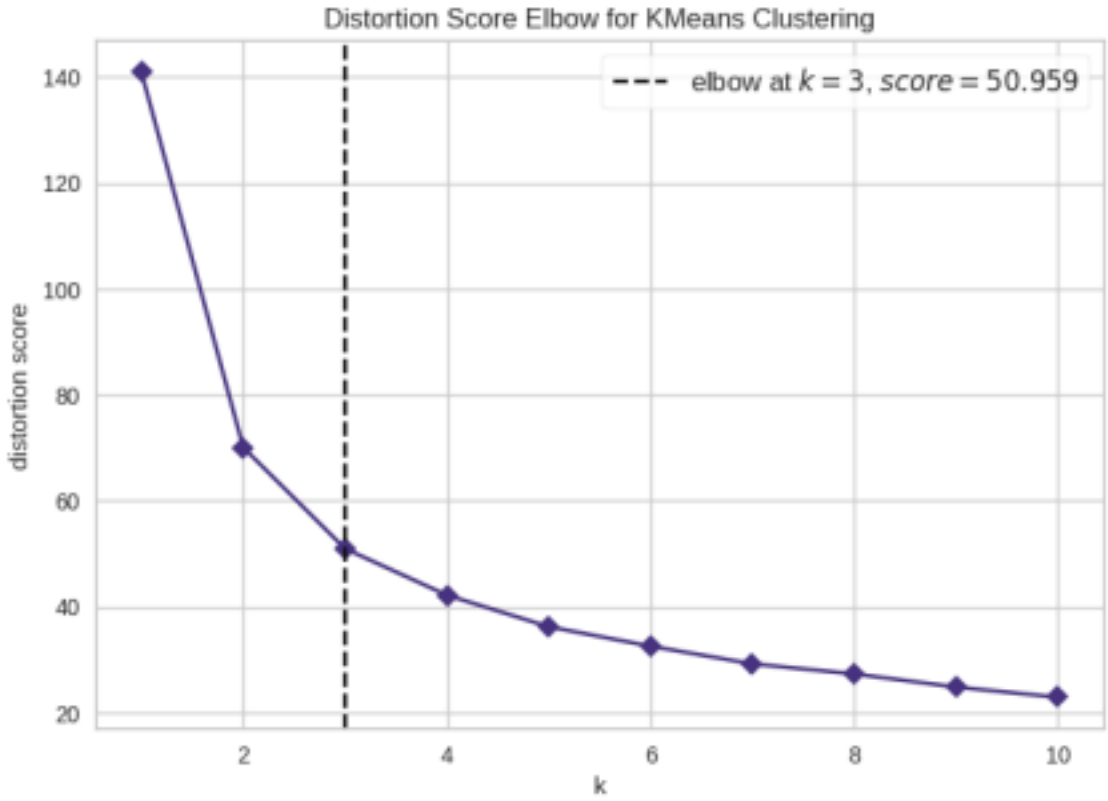
For every numeric feature, we will normalize it to a range of 0 to 1.

[7]: Area MajorAxisLength MinorAxisLength Eccentricity ConvexArea \ 0 0.296370 0.280714 0.314376 0.767872 0.255504 1 0.237427 0.234638 0.284945 0.738636 0.208864 2 0.312263 0.280741 0.351778 0.733009 0.268084 3 0.097973 0.078935 0.186620 0.548194 0.084089 4 0.257660 0.164011 0.422064 0.350968 0.219472

	Extent	Perimeter	label
0	0.831422	0.271791	Kecimen
1	0.667854	0.241842	Kecimen
2	0.565754	0.283594	Kecimen
3	0.701809	0.108284	Kecimen
4	0.906315	0.218493	Kecimen

2 Task 2 - Impact of the Number of Clusters on KMeans Cluster ing with Euclidean Distance

The subtask for this are: 1. Apply KMeans clustering (using Euclidean distance) on the standard ized dataset. 2. For a range of cluster numbers (e.g., from 1 to 10), compute the inertia (SSE) and plot these values to identify the “elbow” point.



[8]: <Axes: title={'center': 'Distortion Score Elbow for KMeans Clustering'}, xlabel='k', ylabel='distortion score'>

From the above plot, the elbow appears to be when the cluster number is 5 as after that point the inertia decreases at a slower rate than for lower cluster numbers.

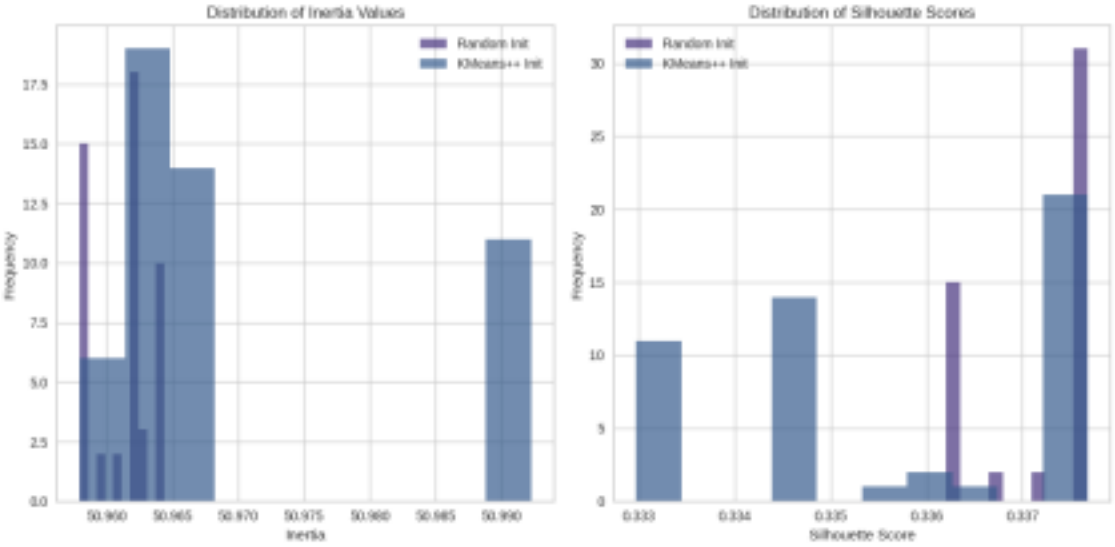
3 Task 3 - Evaluating the Stability of KMeans and KMeans++ Initialization

Subtasks are: 1. Run KMeans clustering 50 times using two initialization methods: - Standard random initialization. - KMeans++ initialization. 2. Compute and compare the average inertia

(SSE) and the Silhouette Score for each method over these iterations.

Standard Random Initialisation:
Average Inertia: 50.96 (±0.00)
Average Silhouette Score: 0.3372 (±0.0007)

KMeans++ Initialisation:
Average Inertia: 50.97 (±0.01)
Average Silhouette Score: 0.3356 (±0.0019)



As can be seen from the above, kmeans++ is slightly more susceptible to differences in initialisation values.

4 Task 4 - Clustering Evaluation Using Purity and Mutual Information

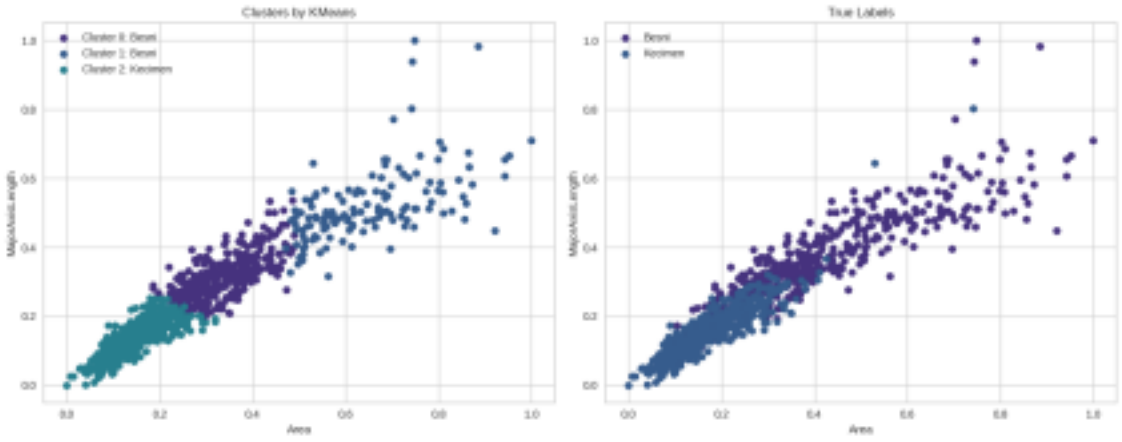
Subtasks are:

1. Use KMeans (with the optimal k from Question 2) to cluster the data. Assume the dataset contains a ground-truth label column (e.g., “label”). For each cluster, assign a label based on the majority class.
2. Evaluation Metrics: Compute and report the following:
    1. Purity Score: Measures how homogeneous each cluster is relative to the true labels.
    2. Mutual Information Score: Quantifies the mutual dependence between the clustering results and the true labels.
    3. Silhouette Score: Evaluates the clustering quality without reference to the ground truth by comparing intra-cluster cohesion versus inter-cluster separation.

Cluster to Label Mapping:
Cluster 0 has label: Besni
Cluster 1 has label: Besni

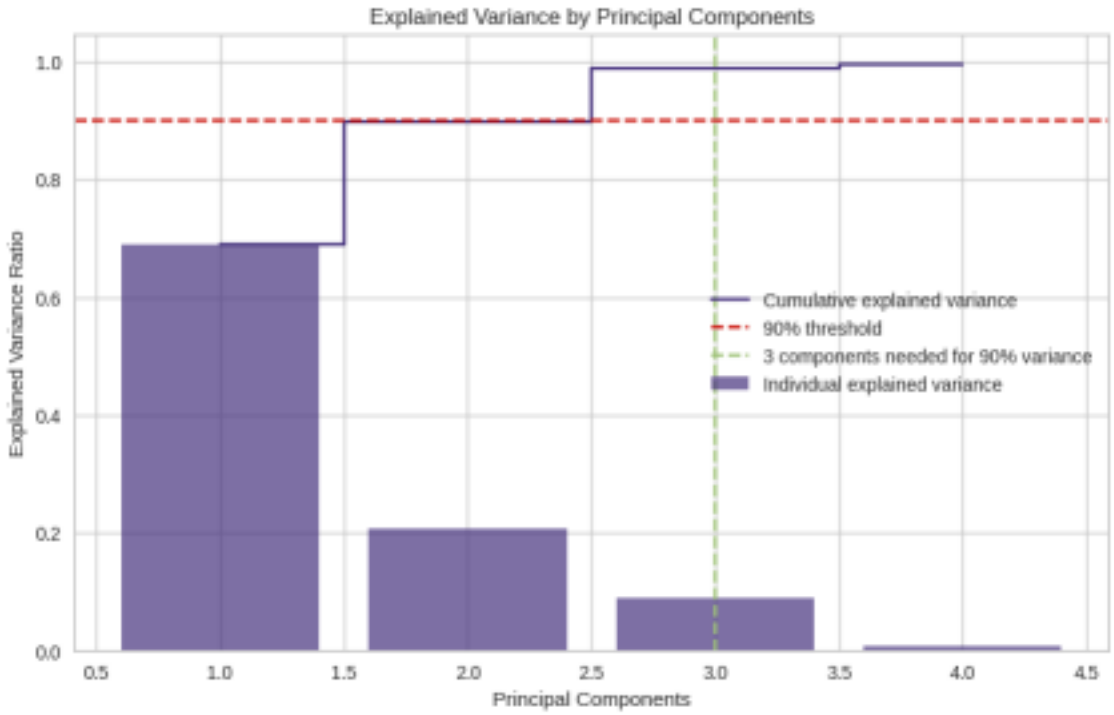
Cluster 2 has label: Kecimen

Evaluation Metrics:
Purity Score: 0.8400
Normalized Mutual Information Score: 0.3343
Silhouette Score: 0.3372



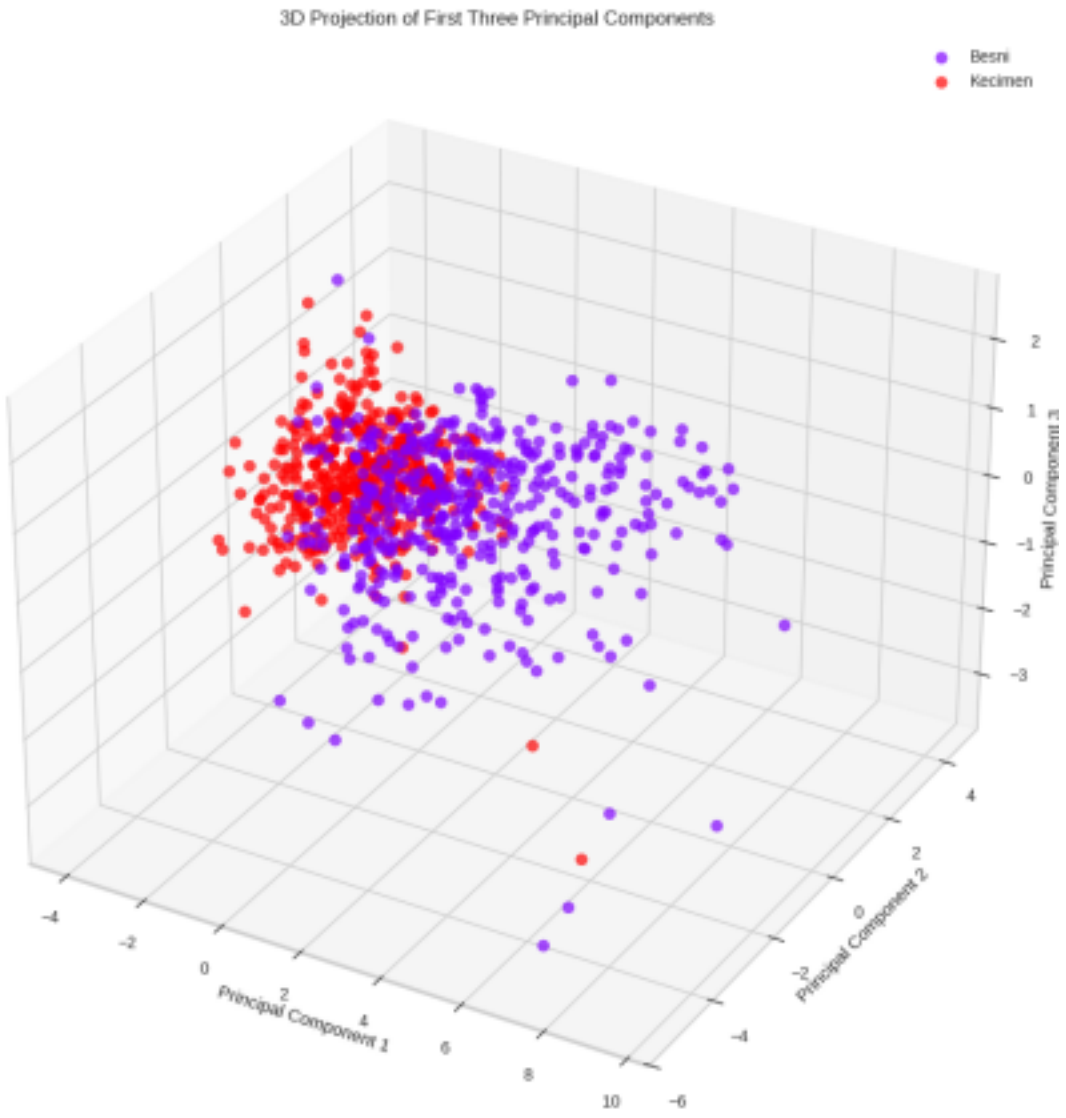
5 Task 5 Principal Component Analysis (PCA) for Dimensionality Reduction

Subtasks are: 1. Apply PCA to reduce the dataset to 4 principal components. 2. Plot the cumulative variance explained by the principal components and determine how many components are needed to retain 90% of the total variance. 3. Create a 3D scatter plot of the first three principal components.



Explained variance ratio by component:
PC1: 0.6903 (0.6903 cumulative)
PC2: 0.2076 (0.8979 cumulative)
PC3: 0.0898 (0.9877 cumulative)
PC4: 0.0081 (0.9958 cumulative)

Number of components needed to retain 90% variance: 3



6 Task 6 - Density Based Clustering Using DBSCAN with Differ ent Distance Metrics

Subtasks are:

1. Apply DBSCAN to the dataset twice:
    1. Once using Euclidean distance.
    2. Once using Mahalanobis distance.

2. Determine the optimal values for eps (ff) and min_samples for each distance metric. 3. Compare the clustering results from both distance metrics.

Will test min_samples in range: [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]

--- Param search for Euclidean Distance ---
Testing min_samples = 14… Auto eps = 0.3855. Clusters=1, Noise=876. ARI = 0.001
Testing min_samples = 15… Auto eps = 0.4000. Clusters=1, Noise=865. ARI = 0.002
Testing min_samples = 16… Auto eps = 0.4169. Clusters=2, Noise=857. ARI = 0.001
Testing min_samples = 17… Auto eps = 0.4252. Clusters=2, Noise=861. ARI = 0.003
Testing min_samples = 18… Couldnt find elbow for min_samples=18. Skipping. Testing min_samples = 19… Auto eps = 0.4442. Clusters=1, Noise=868. ARI = 0.002
Testing min_samples = 20… Auto eps = 0.4628. Clusters=1, Noise=844. ARI = 0.007
Testing min_samples = 21… Couldnt find elbow for min_samples=21. Skipping. Testing min_samples = 22… Couldnt find elbow for min_samples=22. Skipping. Testing min_samples = 23… Auto eps = 0.4759. Clusters=1, Noise=867. ARI = 0.002
Testing min_samples = 24… Couldnt find elbow for min_samples=24. Skipping. Testing min_samples = 25… Couldnt find elbow for min_samples=25. Skipping. Testing min_samples = 26… Couldnt find elbow for min_samples=26. Skipping. Testing min_samples = 27… Couldnt find elbow for min_samples=27. Skipping. Testing min_samples = 28… Auto eps = 0.5086. Clusters=1, Noise=872. ARI = 0.002
Testing min_samples = 29… Couldnt find elbow for min_samples=29. Skipping. Testing min_samples = 30… Auto eps = 0.5262. Clusters=1, Noise=851. ARI = 0.004
--- search done for Euclidean ---
best score (ARI): 0.0069
best parameters: min_samples = 20, eps = 0.4628

Using inverse of covariance matrix for Mahalanobis.

--- Param search for Mahalanobis Distance ---
Testing min_samples = 14… Couldnt find elbow for min_samples=14. Skipping. 25

Testing min_samples = 15… Couldnt find elbow for min_samples=15. Skipping. Testing min_samples = 16… Couldnt find elbow for min_samples=16. Skipping. Testing min_samples = 17… Couldnt find elbow for min_samples=17. Skipping. Testing min_samples = 18… Couldnt find elbow for min_samples=18. Skipping. Testing min_samples = 19… Couldnt find elbow for min_samples=19. Skipping. Testing min_samples = 20… Couldnt find elbow for min_samples=20. Skipping. Testing min_samples = 21… Couldnt find elbow for min_samples=21. Skipping. Testing min_samples = 22… Couldnt find elbow for min_samples=22. Skipping. Testing min_samples = 23… Couldnt find elbow for min_samples=23. Skipping. Testing min_samples = 24… Auto eps = 0.6915. Clusters=1, Noise=839. ARI = 0.001
Testing min_samples = 25… Couldnt find elbow for min_samples=25. Skipping. Testing min_samples = 26… Couldnt find elbow for min_samples=26. Skipping. Testing

min_samples = 27… Couldnt find elbow for min_samples=27. Skipping. Testing min_samples = 28… Couldnt find elbow for min_samples=28. Skipping. Testing min_samples = 29… Couldnt find elbow for min_samples=29. Skipping. Testing min_samples = 30… Couldnt find elbow for min_samples=30. Skipping. --- search done for Mahalanobis ---
best score (ARI): 0.0008
best parameters: min_samples = 24, eps = 0.6915

Applying DBSCAN with best found parameters…
best and final Euclidean DBSCAN with min_samples=20, eps=0.4628 best and final Mahalanobis DBSCAN with min_samples=24, eps=0.6915 final DBSCAN fitting complete.
---
comparing best found params…
--- Euclidean DBSCAN (Best Params) ---
Estimated number of clusters: 1
Estimated number of noise points: 844 (93.78%)
Adjusted Rand Index (ARI): 0.007
V-Measure: 0.036
Silhouette Score: Requires >= 2 clusters.

--- Mahalanobis DBSCAN (Best Params) ---
Estimated number of clusters: 1
Estimated number of noise points: 839 (93.22%)
Adjusted Rand Index (ARI): 0.001
V-Measure: 0.005
Silhouette Score: Requires >= 2 clusters.
---

7 Task 7 - Clustering Performance on PCA-Reduced v Full Dataset

1. Apply KMeans clustering to:
    1. The original standardized dataset.

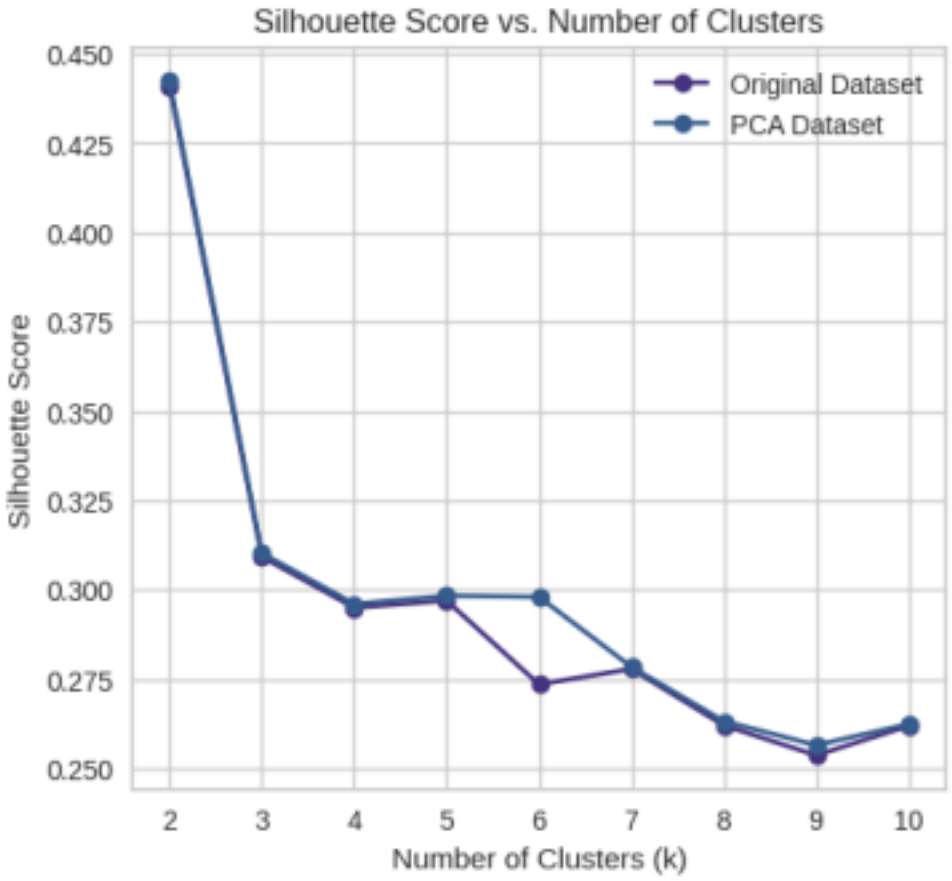
    2. The PCA-transformed dataset (using the principal components from Question5).
2. Evaluate the clustering quality using the Silhouette Score.
3. Compare whether the PCA-transformed dataset results in better-separated and more compact clusters relative to the full dataset.

Original Dataset with 2 clusters - Silhouette Score: 0.441, Inertia: 3397.84 Original Dataset with 3 clusters - Silhouette Score: 0.309, Inertia: 2591.65 Original Dataset with 4 clusters - Silhouette Score: 0.295, Inertia: 2168.41 Original Dataset with 5 clusters - Silhouette Score: 0.297, Inertia: 1895.93 Original Dataset with 6 clusters - Silhouette Score: 0.273,

Inertia: 1684.06 Original Dataset with 7 clusters - Silhouette Score: 0.278, Inertia: 1496.56 Original Dataset with 8 clusters - Silhouette Score: 0.262, Inertia: 1354.20 Original Dataset with 9 clusters - Silhouette Score: 0.254, Inertia: 1250.89 Original Dataset with 10 clusters - Silhouette Score: 0.262, Inertia: 1148.24
PCA Dataset with 2 clusters - Silhouette Score: 0.442, Inertia: 3371.77 PCA Dataset with 3 clusters - Silhouette Score: 0.310, Inertia: 2570.65 PCA Dataset with 4 clusters - Silhouette Score: 0.296, Inertia: 2147.90 PCA Dataset with 5 clusters - Silhouette Score: 0.298, Inertia: 1876.73 PCA Dataset with 6 clusters - Silhouette Score: 0.298, Inertia: 1667.99 PCA Dataset with 7 clusters - Silhouette Score: 0.278, Inertia: 1476.91 PCA Dataset with 8 clusters - Silhouette Score: 0.263, Inertia: 1339.36 PCA Dataset with 9 clusters - Silhouette Score: 0.256, Inertia: 1237.68 PCA Dataset with 10 clusters - Silhouette Score: 0.262, Inertia: 1135.95

Best results:
Original Dataset: k=2, Silhouette Score=0.441
PCA Dataset: k=2, Silhouette Score=0.442

PCA dataset is better.

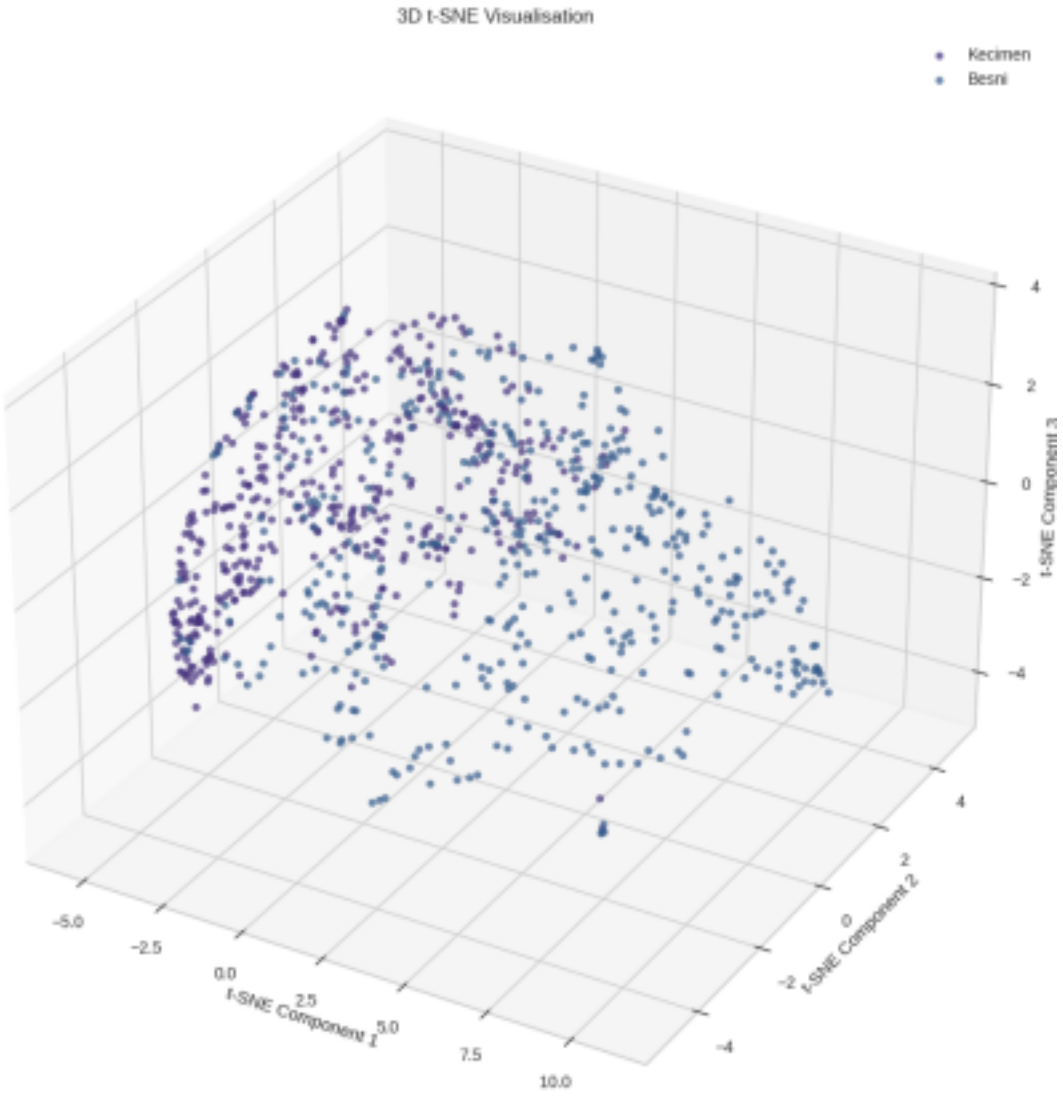


The original dataset's best silhouette score is 0.441. The PCA dataset's best silhoutte score is 0.442. Therefore the PCA dataset results in a better separated and more compact clusters,

however the difference between the results is very minor.

8 Task 8 - Clustering Using t-SNE

1. Apply t-SNE (using the exact method) to reduce the dataset to 4 components.
2. Create a 3D scatter plot of the first three t-SNE components.
3. Apply KMeans clustering on the t-SNE–reduced data using an appropriate number of clusters (e.g., based on prior optimal k or an elbow method on the t-SNE output).
4. Evaluate the clustering performance on the t-SNE–reduced data using metrics such as the Silhouette Score and compare these results to clustering on the original and PCA–transformed dataset.
5. Discuss whether the clusters formed on the t-SNE–reduced data are more distinct and how well they correspond to the known data structure.



Results:
Original Dataset: k=2, Silhouette Score=0.441
PCA Dataset: k=2, Silhouette Score=0.442
t-SNE Dataset: k=2, Silhouette Score=0.400

The PCA dataset provides better clustering quality with a silhouette score of 0.442.

The t-SNE dataset has the lowest silhouette score, therefore it has the least well defined cluster. t-SNE (t-Distributed Stochastic Neighbour Embedding). Even though t-SNE is a dimensionality reduction technique which theoretically could improve cluster cohesion, the original dataset has only 8 features and therefore may not possess enough dimensions to benefit from this technique.