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

Code removed but header comments left in.

April 23, 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.

[ ]:
```
########
# Subtask 1 - Load the dataset ("Dataset.csv") and verify its integrity. #
# Purpose: To load the required data from Dataset.csv into a pandas dataframe␣ ↪called 'df'
  and then to verify its
# integrity. Integrity is understood to mean that the correct number␣ ↪of rows and columns
  have been
# loaded that are in agreement with the manual inspection of these␣ ↪counts.
# If the row or column count is not in agreement with the manual␣ ↪count then the
  assertion will fail.
# Takeaway: The datasets integrity has been validation otherwise execution will␣ ↪be
  stopped.
########
```

Dataset shape: 900 rows, 8 columns
Dataset integrity verified

1.0.2 Subtask 2: Confirm that there are no missing values.

Count the number of missing values in each column and throw an error if any are found.

[3]:
```
########
```

```
# Subtask 2 - Confirm that there are no missing values.
#
# Purpose: To ensure that there are no missing values anywhere in the dataset. # If there
are missing values then the assertion will fail. # Takeaway: The dataset has no missing
values otherwise execution will be␣ ↪stopped.
########
```

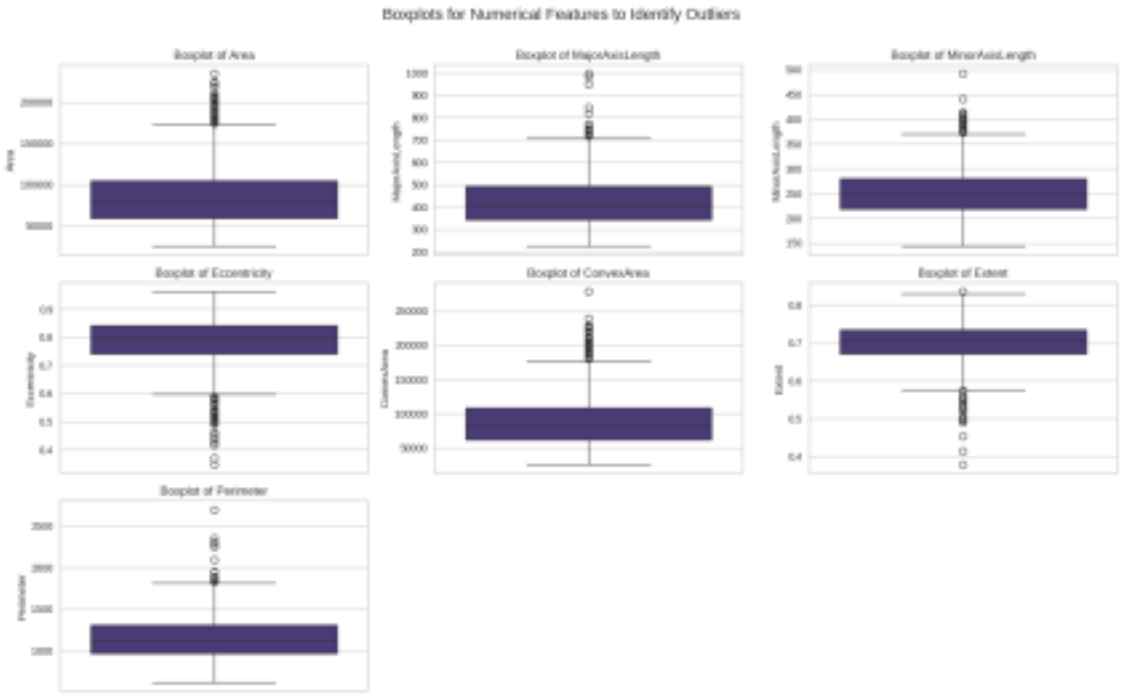
Good, No missing values

2

1.0.3 Subtask 3: Identify and analyze outliers using visualizations such as boxplots.

Boxplots for each numerical feature to identify and analyze outliers. Calculate and display statistics about potential outliers. This can be done by calculating the IQR and then using that to identify the lower and upper bounds of the outliers. The label is categorical so not included in outlier detection.

[4]:
```
########
# Subtask 3 - Identify and analyze outliers using visualizations such as␣ ↪boxplots.
#
# Purpose: Outliers may occur in numerical features. This subtask is to␣ ↪identify
  and analyse these outliers.
# Boxplots are used to identify and analyse outliers. The label is␣ ↪categorical so not
  included in outlier detection.
# Takeaway: A boxplot per numerical feature is created and displayed. This␣ ↪allows for
  the manual inspection of the outliers.
# Outliers in the plots are identified as points that are outside of␣ ↪the whiskers of the
  boxplot. It can be seen from the plots
# that there are outliers in ever feature, but either above or below␣ ↪the upper or lower
  whiskers respectively.
########
```

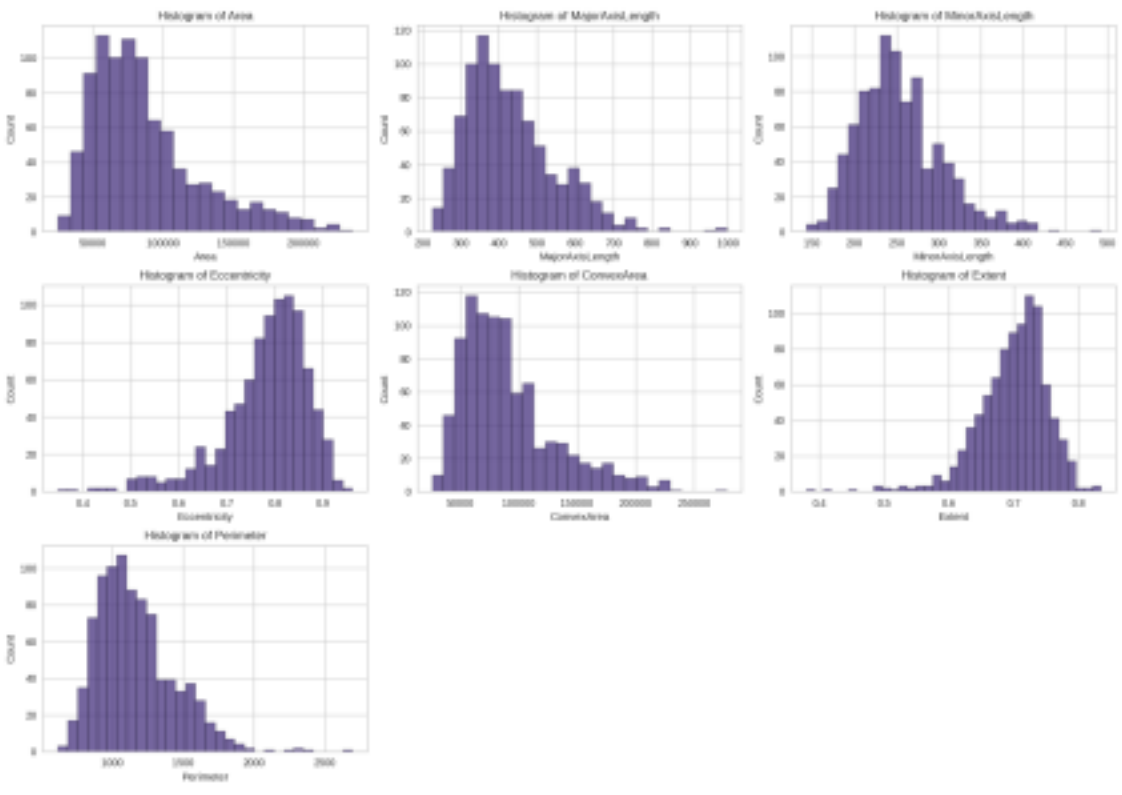


1.0.4 Subtask 4: Visualise feature distributions with histograms and KDE plots to understand the overall distribution of each feature.

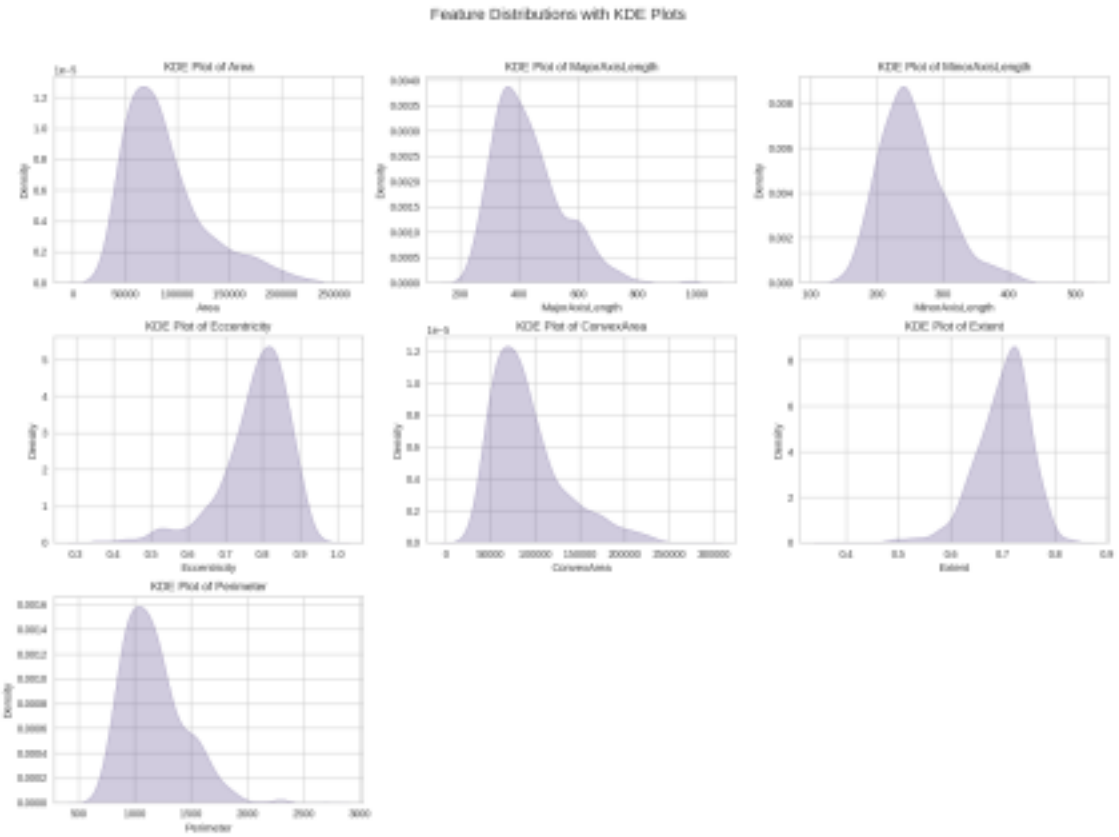
Seaborn has differing functions for histograms and KDE plots. Use these.

[5]:
```
########
# Subtask 4: Visualise feature distributions with histograms and KDE plots to␣ ↪understand the overall distribution of each feature.
#
# Purpose: To visualise the distribution of each feature in the dataset.␣ ↪Histograms and KDE plots are used to visualise the distribution of each␣ ↪feature.
# Takeaway: A histogram and KDE plot per numerical feature was created and␣ ↪displayed. It showed that ever numerical feature is skewed. The skew␣ ↪direction for each feature is:
# - Area: Right
# - MajorAxisLength: Right
# - MinorAxisLength: Right
# - Eccentricity: Left
# - ConvexArea: Right
# - Extent: Left
# - Perimeter: Right
#########
```

5


Feature Distributions with Histograms
Histogram of Area
Count
Area
Histogram of MajorAxisLength
Count
MajorAxisLength
Histogram of MinorAxisLength
Count
MinorAxisLength
Histogram of Eccentricity
Count
Eccentricity
Histogram of ConvexArea
Count
ConvexArea
Histogram of Extent
Count
Extent
Histogram of Perimeter
Count
Perimeter

6



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.

[6]:
```
#########
# Subtask 5 - Review feature statistics (e.g., mean, standard deviation) to get␣ ↪insights into
  the data.
#
# Purpose: To review the statistics of the numerical features in the dataset. # Takeaway:
From the descriptive statistics it can be observed that: # - Area: Right skewed as mean is
greater than the median (50%␣ ↪percentile)
# - MajorAxisLength: Right skewed as mean is greater than the median␣ ↪(50% percentile)
# - MinorAxisLength: Right skewed as mean is greater than the median␣ ↪(50% percentile)
# - Eccentricity: Left skewed as mean is less than the median (50%␣ ↪percentile)
# - ConvexArea: Right skewed as mean is greater than the median (50%␣ ↪percentile)
```



```
# - Extent: Left skewed as mean is less than the median (50%␣ ↪percentile)
# - Perimeter: Right skewed as mean is greater than the median (50%␣ ↪percentile)
# Standard deviations are quite large for Area, ConvexArea,␣ ↪MajorAxisLength,
  MinorAxisLength, Perimeter. This indicates there are a␣ ↪large range of objects in the
  dataset
#########
```

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:

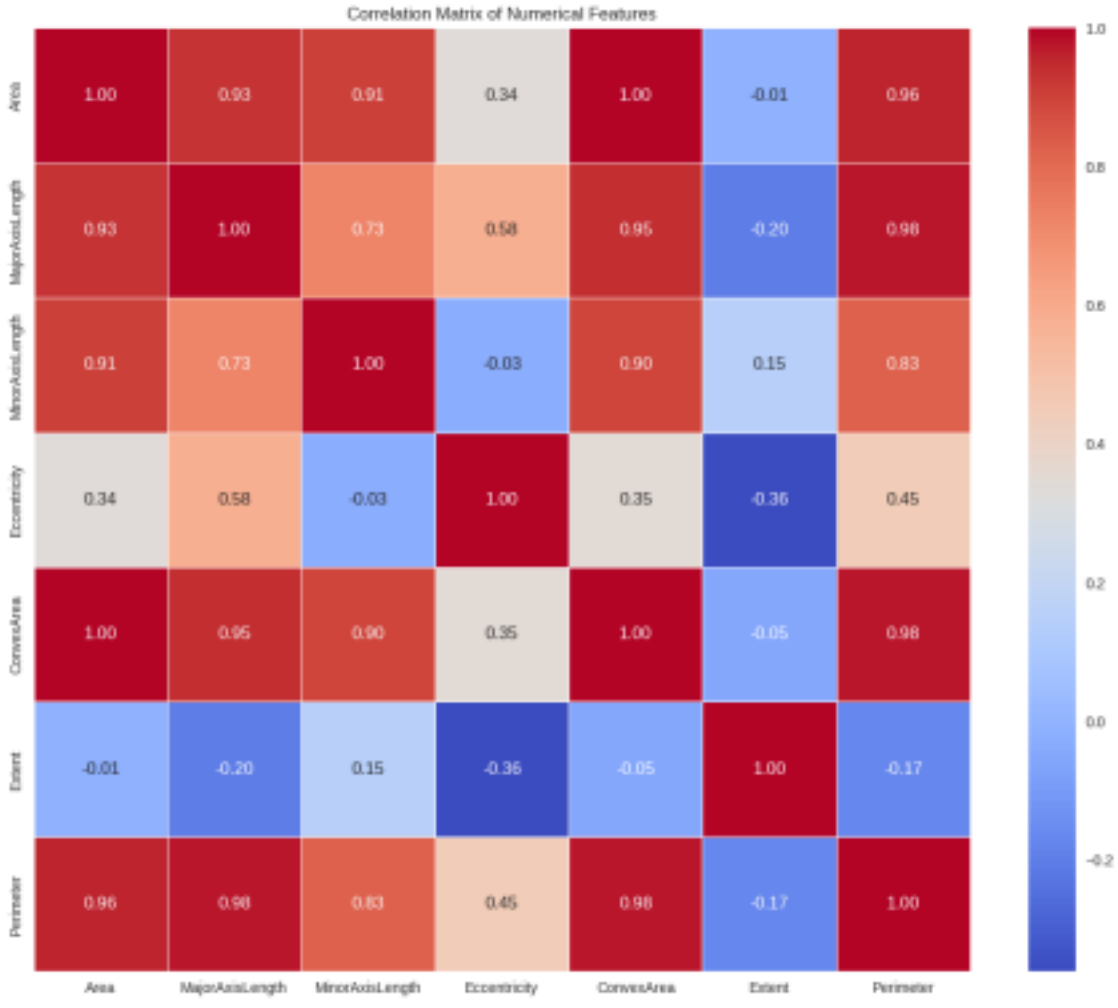
Median Skewness Kurtosis IQR Range
Area 78902.000000 1.175237 1.074073 45680.250000 209660.000000 MajorAxisLength 407.803951 0.989544 1.326808 148.744116 771.662400 MinorAxisLength 247.848409 0.800049 0.953915 60.777448 348.564407 Eccentricity 0.798846 -1.327503 2.492121 0.100805 0.613395 ConvexArea 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]:
```
#########
# Subtask 6 - Normalize or standardize the dataset so that all features␣
↪contribute equally in distance calculations, which is crucial for clustering. #
# Purpose: Normalise the dataset so that all features contribute equally in␣
↪distance
```

```
    calculations.
# Takeaway: The dataset was normalized to a range of 0 to 1 for every numeric␣
↪feature.
```



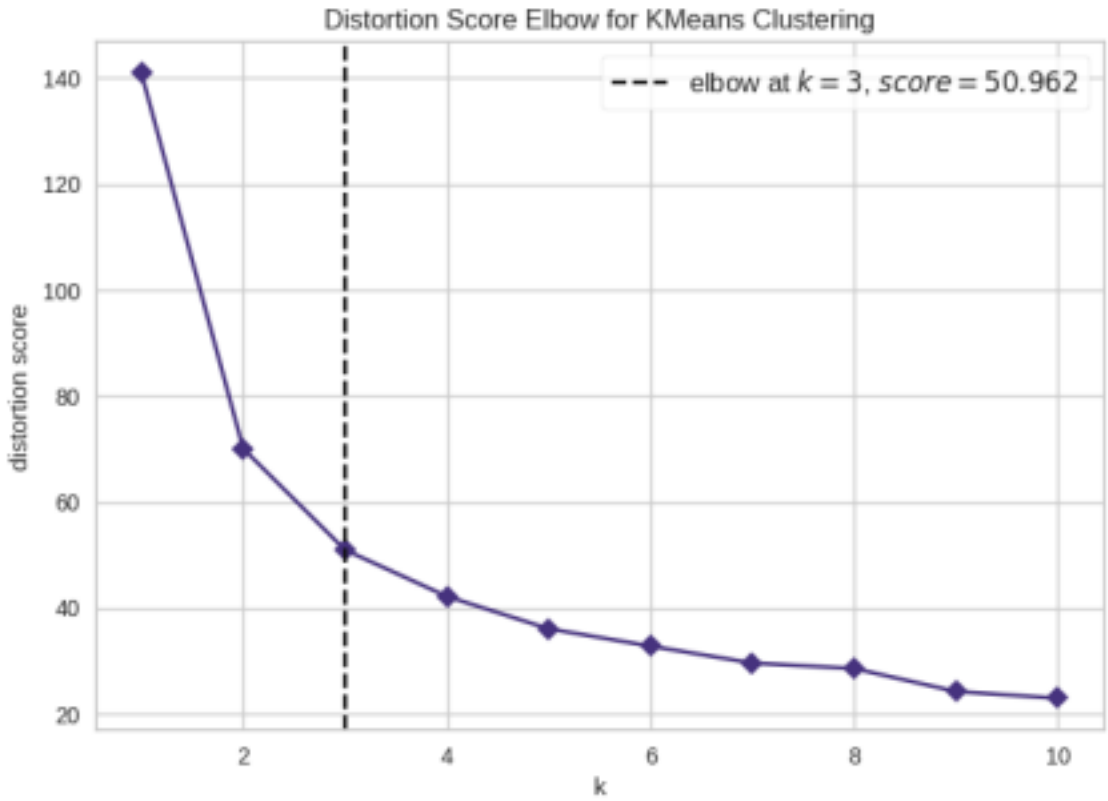
```
########
```

[7]:
	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	label
0	0.296370	0.280714	0.314376	0.767872	0.255504	0.831422	0.271791	Kecimen
1	0.237427	0.234638	0.284945	0.738636	0.208864	0.667854	0.241842	Kecimen
2	0.312263	0.280741	0.351778	0.733009	0.268084	0.565754	0.283594	Kecimen
3	0.097973	0.078935	0.186620	0.548194	0.084089	0.701809	0.108284	Kecimen
4	0.257660	0.164011	0.422064	0.350968	0.219472	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]:
```
########
# Task 2 - Impact of the Number of Clusters on KMeans Clustering with Euclidean␣
↪Distance
#
# Purpose: Apply KMeans clustering on the normalised dataset and to determine␣
↪the
  optimal number of cluster using the elbow method.
# Takeaway: The elbow appears to be when the cluster number is 3. The␣
↪KElbowVisualizer is a handy library that displays 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.

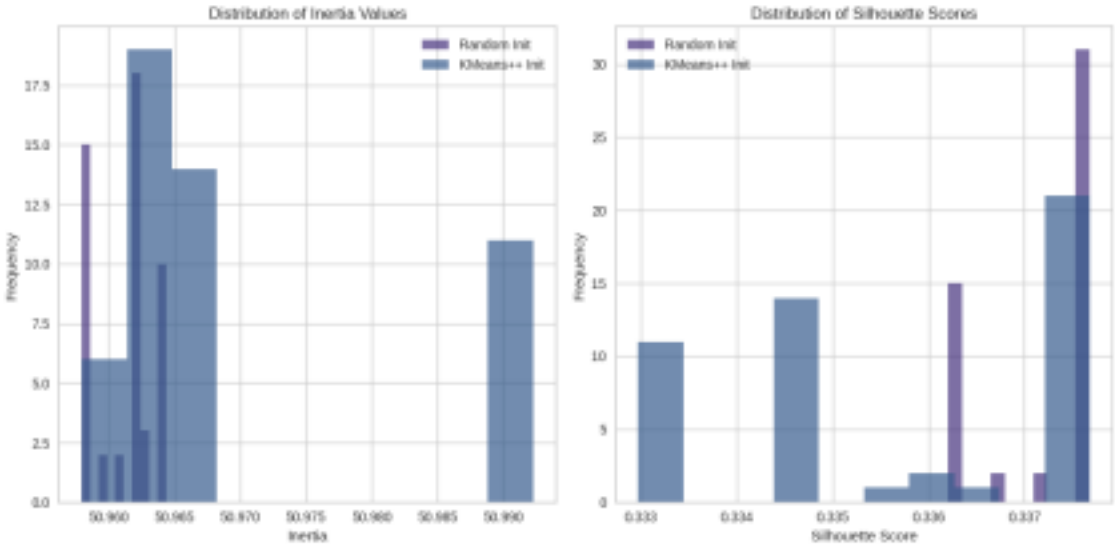
3 Task 3 - Evaluating the Stability of KMeans and KMeans++ Initialization

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.

[9]:
```
##########
# Task 3 - Evaluating the Stability of KMeans and KMeans++ Initialization #
# Purpose: To compare the KMeans and Kmeans++ with regard to their stability␣ ↪when initialised randomly in 50 different runs.
```



```
# Takeaway: KMeans++ is slightly more susceptible to differences in␣ ↪initialisation values.
# This can be observed from the distribution of the inertia and␣ ↪silhouette scores with KMeans++ distribution being more dispersed. ##########
```



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.

[10]: ##########

```
# Task 4 - Clustering Evaluation Using Purity and Mutual Information #
```

```
# Purpose: To evaluate the clustering results using purity, mutual information␣ ↪and
  silhouette score.
# Takeaway: The purity score of 0.84 is quite high, indicating that the members␣ ↪of each
cluster are quite homogeneous and in agreement with the true labels. # The mutual
information score of 0.3343 is not very high indicating␣ ↪that the structure of the clusters is not
very similar to the true labels. # The silhouette score of 0.3372 is not very high indicating that
the␣ ↪clusters are not very well separated.
```

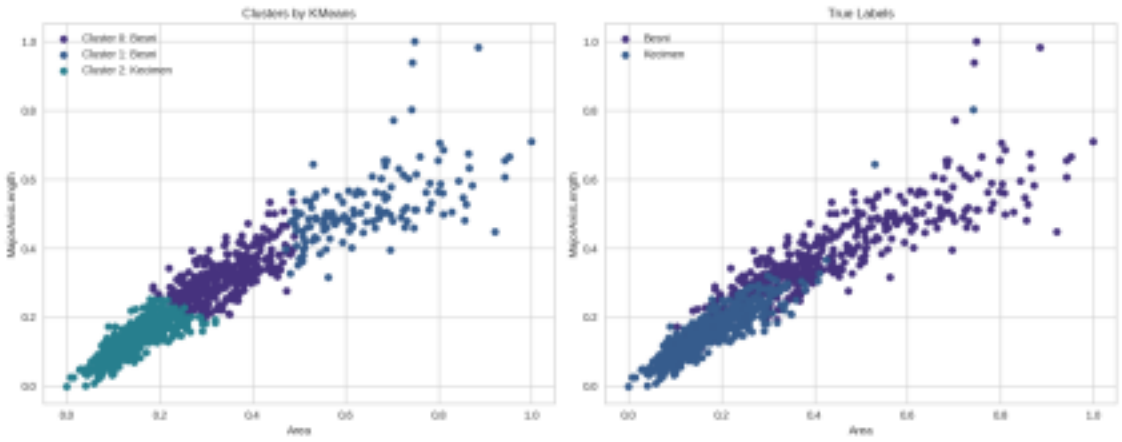
15

```
# Although the cluster purity was high, the clustering results were␣ ↪not very good most
  likely due to splitting of the Besni class into two␣ ↪clusters.
##########
```

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

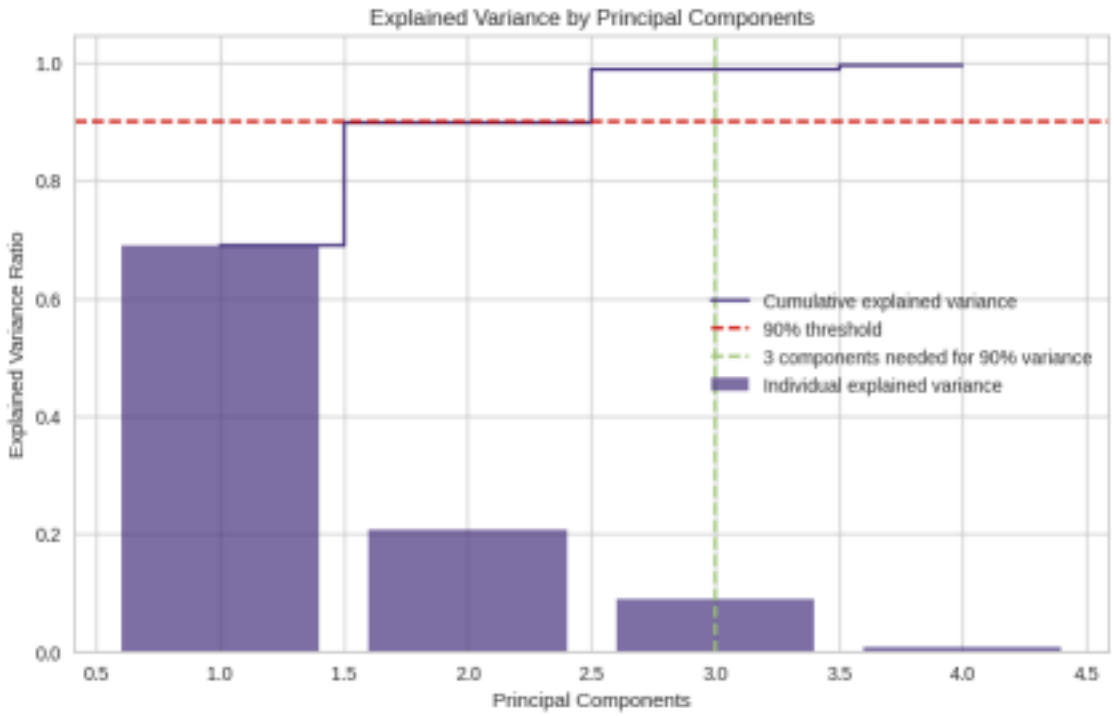
17



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.

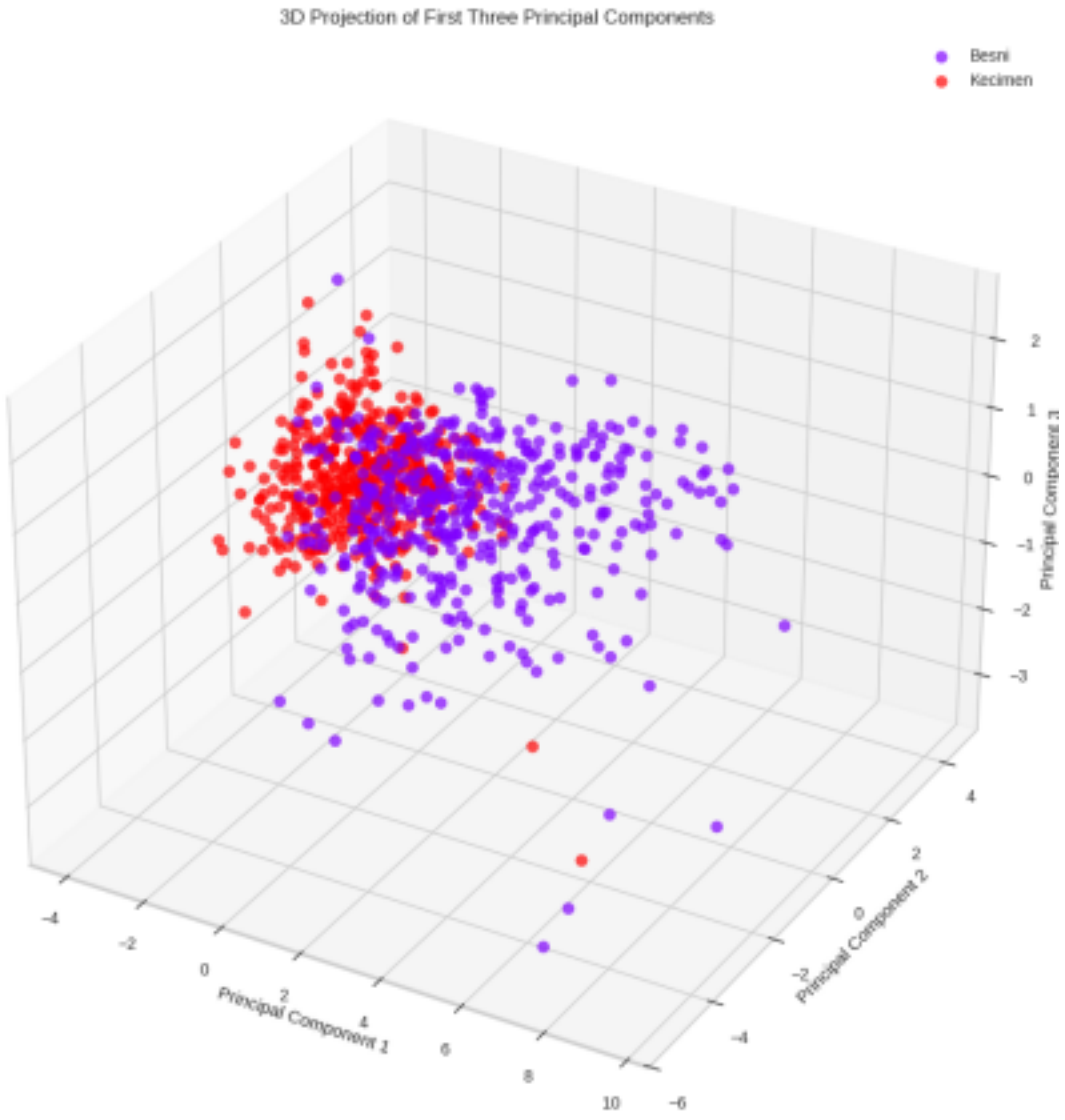
[11]:
```
#######
# Task 5 - Principal Component Analysis (PCA) for Dimensionality Reduction #
# Purpose: To reduce the dataset to 4 principal components, to plot the␣ ↪cumulative
  variance explained by the principal components and to determine␣ ↪the number of
  components needed to retain 90% of the total variance.
# Takeaway: PCA was used to reduce the dataset to 4 principal components. From␣ ↪this it
  was determined that PC1, PC2 and PC3 were needed to retain 90% of␣ ↪the total
  variance. (3 components were needed). The amount of variance␣ ↪explained by each
  component is as follows:
# PC1: 0.6903 (0.6903 cumulative)
# PC2: 0.2076 (0.8979 cumulative)
# PC3: 0.0898 (0.9877 cumulative)
# PC4: 0.0081 (0.9958 cumulative)
#######
```



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.

[12]:
```
########
# Task 6 - Density Based Clustering Using DBSCAN with Different Distance Metrics #
# Purpose: To apply DBSCAN to the dataset twice: once using Euclidean distance␣ ↪and once using Mahalanobis distance.
# Takeaway: Care had to be taken when using DBSCAN as the number of clusters to␣ ↪be used cannot be set beforehand. As we know there are two label ideally we␣ ↪would have two clusters.
# DBSCAN also assigns a -1 label to noise points which had to be␣ ↪filtered out when calculating the silhouette score.
# The best euclidean parameters were eps=0.65, min_samples=5. The␣ ↪best mahalanobis parameters were eps=7.00, min_samples=5.
# The cluster evaluation metrics were as follows:
# Euclidean - Purity: 0.5522, Mutual Information: 0.0148, Silhouette:␣ ↪0.2529876685755327, Cluster Sizes: 3
# Mahalanobis - Purity: 0.5022, Mutual Information: 0.0009,␣ ↪Silhouette: 0.6096798484416615, Cluster Sizes: 3
# The Euclidean distance had a better clustering result as it had a␣ ↪higher purity and mutual information score. The mahalanobis distance had a␣ ↪higher silhouette score which indicated that the clusters were more␣ ↪separated.
########
```



```
Euclidean - min_samples=5, eps=0.65, clusters=2, noise points=116
Euclidean - min_samples=10, eps=0.40, clusters=2, noise points=647
Euclidean - min_samples=15, eps=0.55, clusters=2, noise points=433
Euclidean - min_samples=20, eps=0.50, clusters=2, noise points=695

Best Euclidean parameters: eps=0.65, min_samples=5
Mahalanobis - min_samples=5, eps=7.00, clusters=2, noise points=1
Mahalanobis - min_samples=10, eps=1.00, clusters=1, noise points=259
```

Mahalanobis - min_samples=15, eps=1.00, clusters=1, noise points=287
Mahalanobis - min_samples=20, eps=1.00, clusters=1, noise points=324

Best Mahalanobis parameters: eps=7.00, min_samples=5

Metrics results:
Euclidean - Purity: 0.5522, Mutual Information: 0.0148, Silhouette: 0.2529876685755327, Cluster Sizes: 3
Mahalanobis - Purity: 0.5022, Mutual Information: 0.0009, Silhouette: 0.6096798484416615, Cluster Sizes: 3

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

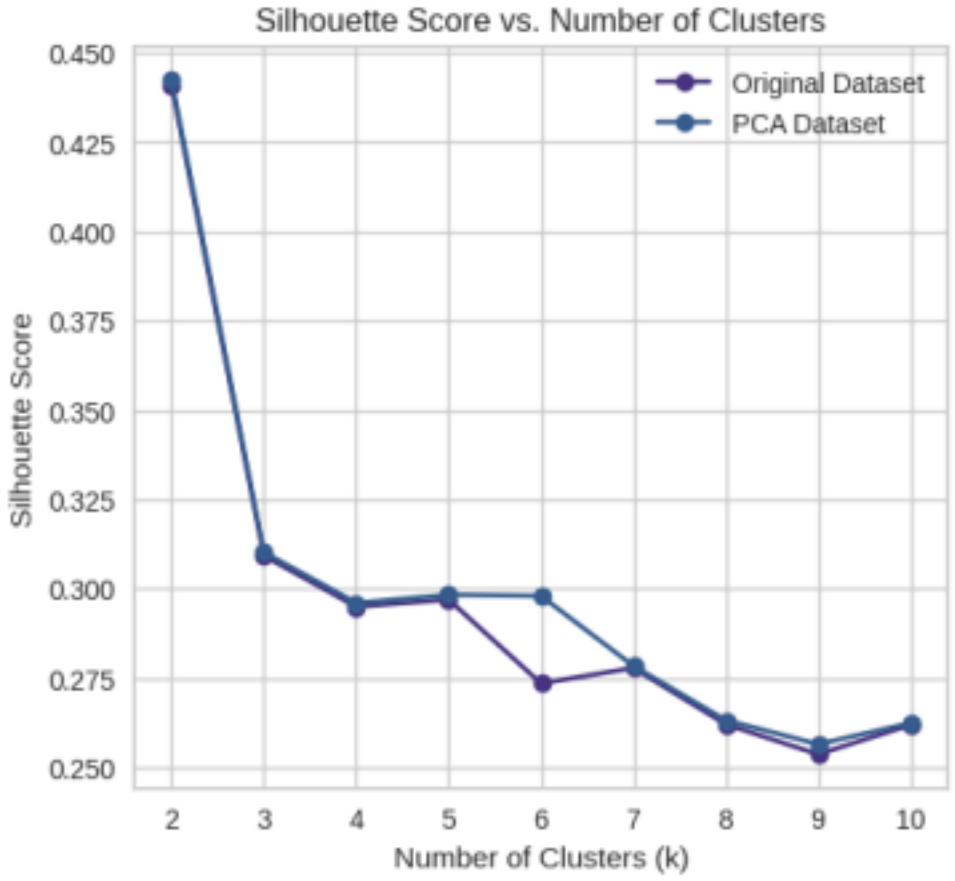
1. Apply KMeans clustering to:
    1. The original standardized dataset.
    2. The PCA-transformed dataset (using the principal components from Question5).
2. Evaluate the clustering quality using the Silhouette Score.
3. Compare whether the PCA-transformed dataset results in better-separated and more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

[13]:
```
##########
# Task 7 - Clustering Performance on PCA-Reduced v Full Dataset
#
# Purpose: To apply KMeans clustering to the original and PCA-transformed datasets
and to evaluate the clustering quality using the Silhouette Score. # Takeaway: The results
for the two datasets were:
# Original Dataset: k=2, Silhouette Score=0.441
#PCA Dataset: k=2, Silhouette Score=0.484
##########
```

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.

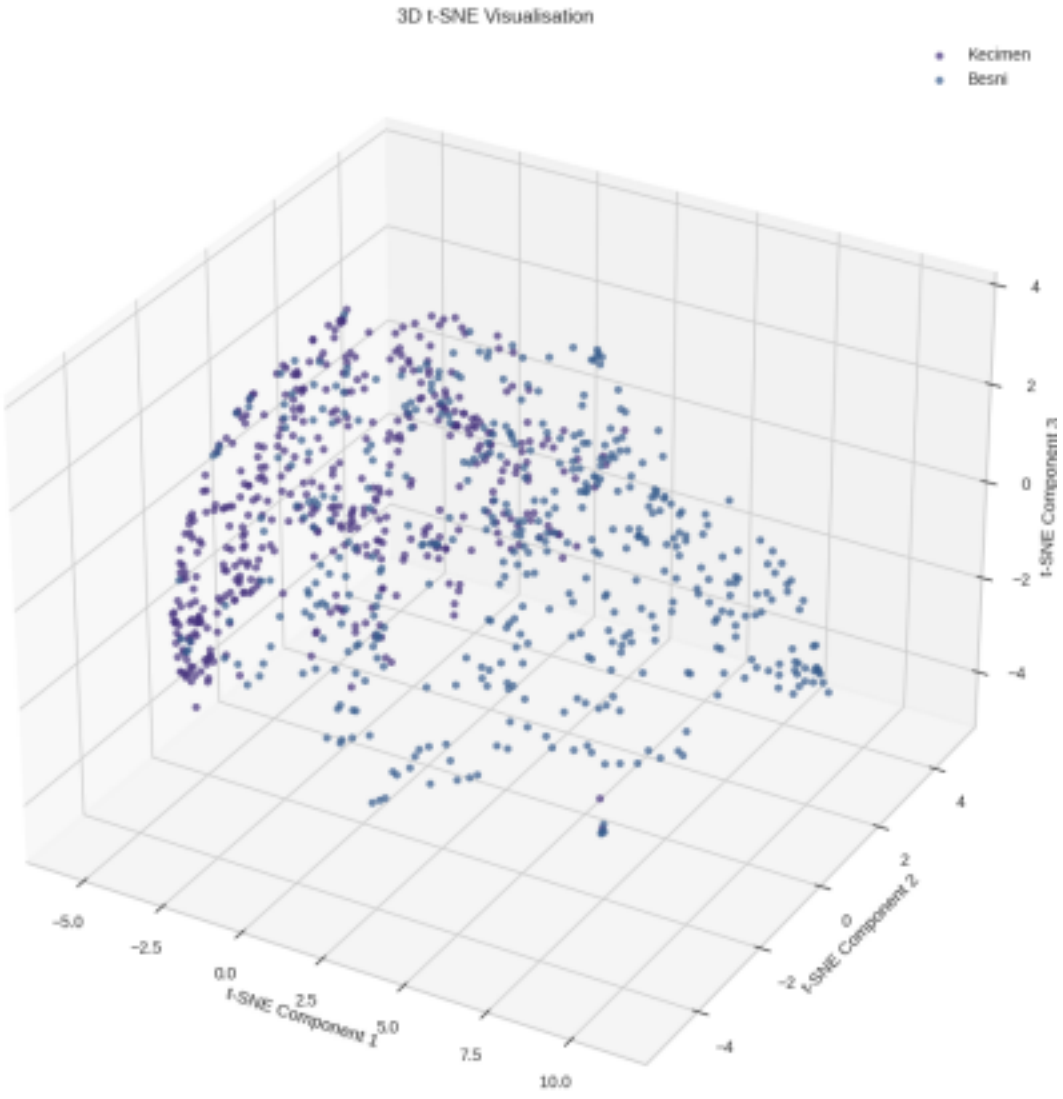
8 Task 8 - Clustering Using t-SNE

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.

[14]:
```
##########
# Task 8 - Clustering Using t-SNE
#
```



```
# Purpose: To apply t-SNE to generate a new dataset with 4 components. Compare␣ ↪the
  clustering results using KMeans on the t-SNE reduced data, the original␣ ↪data and the
  PCA transformed data.
# Takeaway: The results for the three datasets were:
# Original Dataset: k=2, Silhouette Score=0.441
# PCA Dataset: k=2, Silhouette Score=0.484
# t-SNE Dataset: k=2, Silhouette Score=0.400
# The PCA dataset has the highest silhouette score, therefore it has␣ ↪the best defined
  clusters.
# The t-SNE dataset has the lowest silhouette score, therefore it has␣ ↪the least well defined
  cluster.
# 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.
##########
```



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.