Ma412 - DATA

Project

Supervised by Ms Gharsalli

Lakardi Rayan

Summary

I. Introduction :

The vast expanse of the sky becomes a canvas for the intricate dance of aircraft trajectories, each telling a unique story through its 18 distinct features. In our pursuit of understanding and categorizing these dynamic aerial movements, we delve into the realm of unsupervised clustering techniques. Encoded within the 'data.npy' database are 3879 instances, each a snapshot of an aircraft's trajectory, capturing its position and various noteworthy features.

Our objective is clear, employ clustering methodologies to group these trajectories based on inherent patterns and similarities. The challenge lies not only in the effective application of clustering algorithms but also in determining the optimal number of clusters that truly encapsulate the underlying structures within the dataset.

To assess the efficacy of our methods, a meticulously defined metric will serve as our compass, guiding us through the intricate skies of data exploration. We embark on a journey that extends beyond the conventional, seeking novel clustering techniques not solely covered in our course. The pursuit of innovation is rewarded as we uncover methodologies that breathe life into the data, providing insights into the intricacies of aircraft trajectories.

Moreover, the dataset, pristine and unblemished, obviates the need for extensive pre-processing. However, pondering the hypothetical scenario of data pre-processing, we explore viable methods, each grounded in its rationale. The chosen path depends on the nature of the dataset, and we meticulously justify our approach.

In this intricate ballet of data exploration, we aim to contribute not just answers but a comprehensive understanding of the unsupervised clustering landscape for aircraft trajectories.

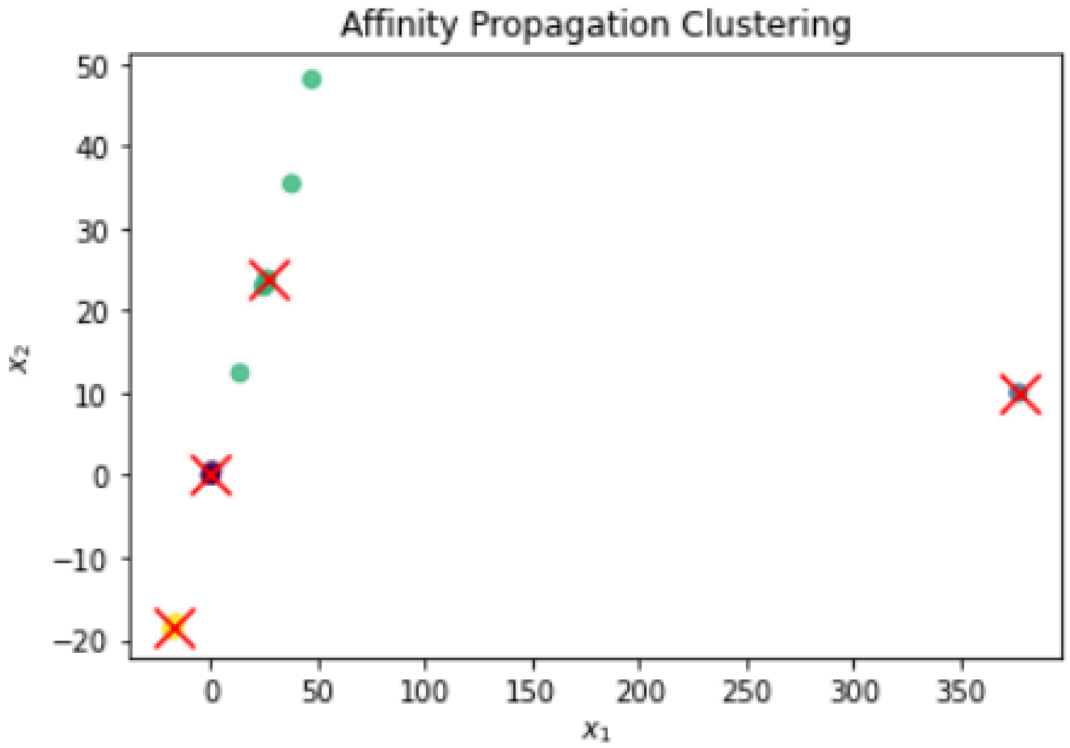
II. Affinity Propagation Method :

1. Definition :

The Affinity Propagation algorithm relies on several key functions to achieve its goal of unsupervised clustering of aeronautical data. Firstly, the calculation of the similarity matrix assesses relationships between each pair of points in the dataset, using measures such as Euclidean distance. This affinity matrix forms the foundation for the subsequent steps. The update functions for responsibilities and availability come into play during the algorithm's iterations. Responsibilities indicate how much a point prefers to be chosen as the representative of another, while availability measures a point's propensity to be chosen. These updates dynamically adjust preferences and propensities based on affinities.

A crucial step involves identifying examples, which serve as the centers of clusters. Examples are selected from points that simultaneously act as representatives and preferences. This procedure helps define potential cluster centers. Finally, cluster formation involves associating each point with its exemplar, completing the grouping process.

2. Plots :



3. Comments :

The provided plot visually encapsulates the clustering outcomes achieved by the

Affinity Propagation algorithm on the given dataset. The scatter plot effectively distinguishes clusters through color-coded data points, offering insight into the algorithm's ability to identify meaningful groupings. The red 'x' markers represent estimated cluster centers, serving as key indicators of the identified cluster structures. The use of the 'viridis' color map enhances the visibility of distinct clusters, and the transparency of data points aids in revealing potential overlapping regions.

III. Birch Method :

1. Definition :

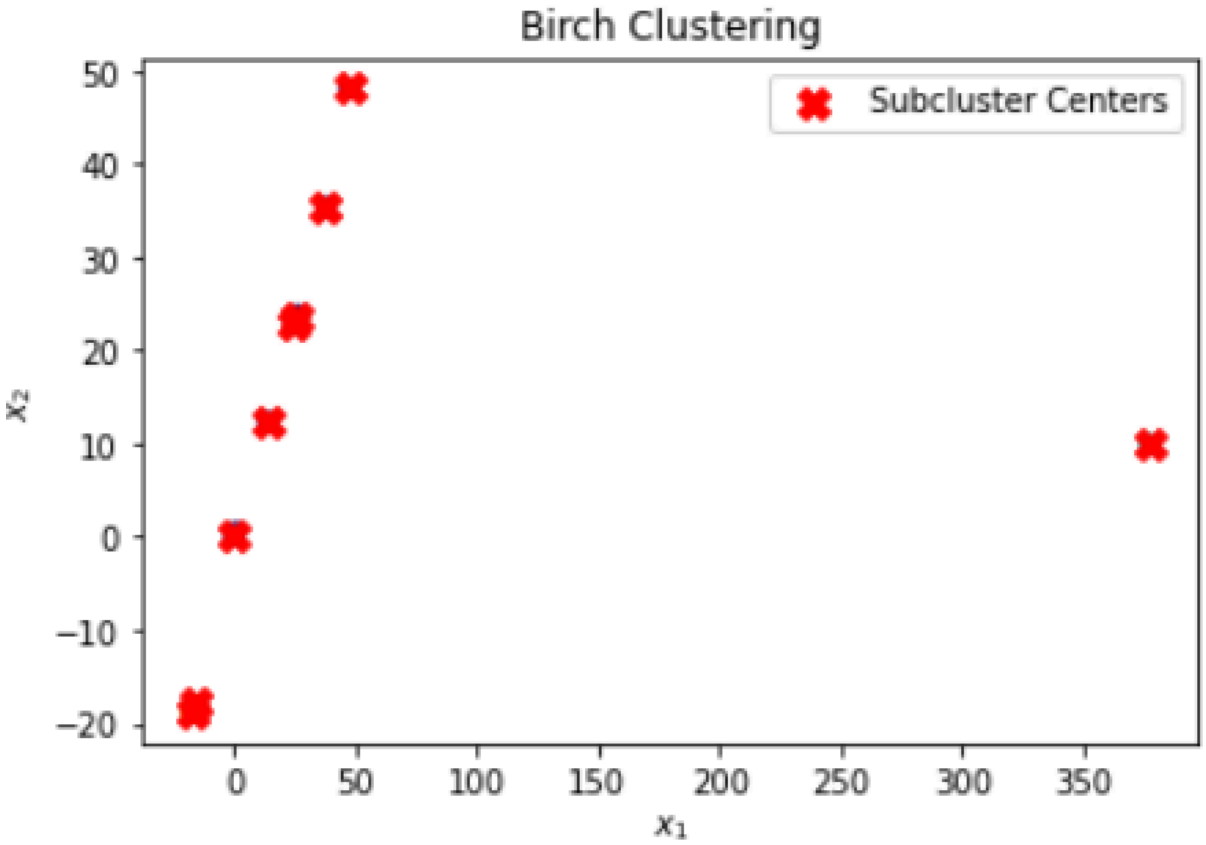
The Birch clustering algorithm is a hierarchical method that operates by recursively dividing the dataset into a tree-like structure of subclusters. This approach efficiently handles large datasets and enables scalable clustering. The algorithm involves several key functions to achieve its hierarchical clustering objectives.

The first step is the construction of a Clustering Feature Tree (CF Tree), which organizes the data into a memory-efficient structure. This tree allows for a top-down approach in which the dataset is recursively split into subclusters. The branching factor and the threshold for the number of samples in each node play crucial roles in controlling the tree's growth.

The Subcluster Selection function is essential in determining which subclusters to retain and propagate to the next level of the hierarchy. It considers both the quality and quantity of the subclusters, ensuring that meaningful patterns are captured in the subsequent divisions.

Additionally, the Birch algorithm employs a distance metric to measure the similarity between subclusters, aiding in the decision-making process during the hierarchical clustering. The algorithm adapts dynamically to the underlying structure of the data, making it suitable for datasets with varying cluster densities and shapes.

2. Plots :



3. Comments :

The generated plot from the Birch clustering script provides a clear visual representation of the algorithm's results on the specified subset of data. Each data point is color-coded based on its assigned cluster label by the Birch algorithm, facilitating a straightforward visual identification of clusters. The red 'X' markers denote the estimated centers of sub-clusters, offering additional insights into the internal structure of the identified clusters. With the script specifying three clusters, the distribution of points suggests that the Birch algorithm successfully segmented the data into three distinct groups.

IV. DBSCAN Method :

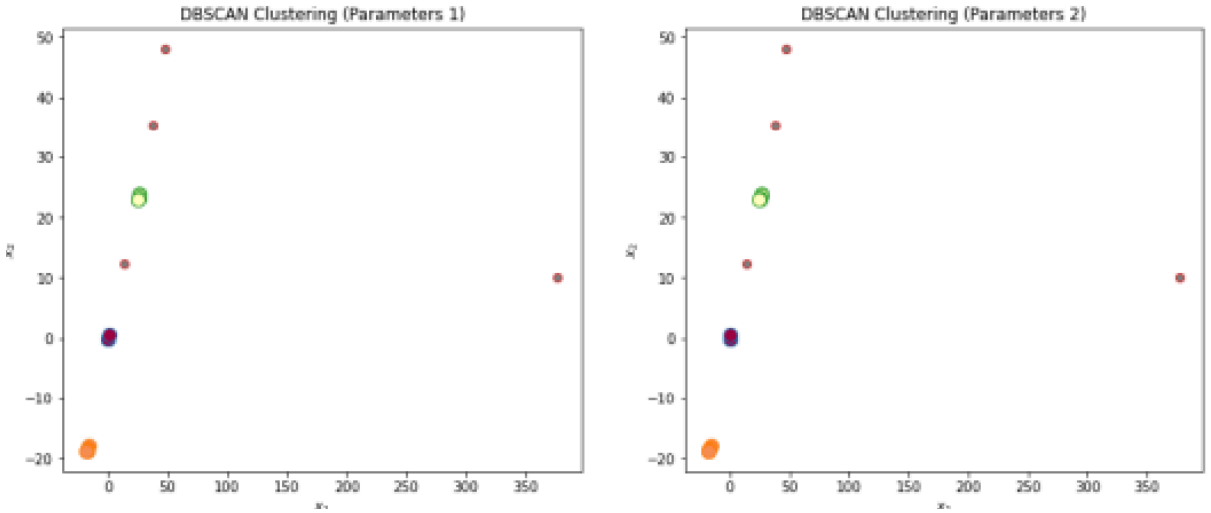
1. Definition :

The DBSCAN algorithm is a robust and versatile method for clustering datasets based on the density of data points. It operates by identifying core points, which have a sufficient number of neighbors within a specified radius, and connecting them to form clusters. The algorithm distinguishes between core points, border points, and outliers, adapting well to datasets with irregular shapes and varying densities.

DBSCAN's primary functions include the identification of core points, the

establishment of density reachability between core points, and the formation of clusters by connecting these core points and their density-reachable neighbors. Additionally, the algorithm is capable of detecting outliers—data points that do not fit within any cluster due to their lack of sufficient neighbors.

2. Plots :



3. Comments :

The two plots generated by the code illustrate the outcomes of DBSCAN clustering with two distinct parameter configurations. In the first plot, obtained with `eps=0.3` and `min_samples=4`, the clusters are relatively tight and well-defined, with noise points marked in gray. The dispersion of points within the clusters is moderate. Conversely, the second plot, produced with `eps=0.5` and `min_samples=3`, exhibits more dispersed clusters, indicating a potential identification of larger clusters. Different-colored markers represent the identified clusters, while color denotes noise points. The marker size reflects the density of points within each cluster. These side-by-side visualizations highlight DBSCAN's sensitivity to variations in its parameters, underscoring the importance of fine-tuning parameters for optimal clustering results tailored to the specific characteristics of the dataset.

V. Fuzzy C-Means Clustering Method :

1. Definition :

The Fuzzy C-Means clustering method is a soft clustering algorithm that assigns membership degrees to each data point for multiple clusters, rather than forcing a rigid assignment to a single cluster. Fuzzy C-Means extends the traditional K-Means algorithm by incorporating fuzziness into the membership assignment, allowing data points to belong to multiple clusters simultaneously. This flexibility makes Fuzzy C-Means suitable for situations where data points may exhibit mixed characteristics

or uncertainties in their affiliations.

In Fuzzy C-Means, the algorithm iteratively updates cluster centroids and membership degrees until convergence. The membership degrees are real numbers between 0 and 1, indicating the degree of association of each data point with each cluster. The iterative optimization process minimizes an objective function that considers both the distances between data points and cluster centroids and the membership degrees.

The main functions of the Fuzzy C-Means algorithm involve initializing cluster centroids and membership degrees, iteratively updating these values, and determining the final cluster assignments based on the optimized membership degrees. This method is well-suited for scenarios where traditional hard clustering methods may oversimplify complex patterns in the data.

2. Plots :

3. Comments :

The plot resulting from the Fuzzy C-Means clustering reveals distinct clusters and their centroids in a dataset. Each data point is color-coded according to its assigned cluster, showcasing the algorithm's ability to categorize data points based on similarity. The red 'X' markers denote the centroids of these clusters, providing central reference points for understanding the spatial distribution of each cluster. Notably, Fuzzy C-Means employs a soft clustering approach, as indicated by the

varying intensity of colors, representing the degree of membership of each data point to its assigned cluster. The normalization of the dataset before clustering ensures the algorithm's robustness across different feature scales. Configured to identify three clusters, the plot effectively visualizes these clusters with a title indicating "Fuzzy C-Means Clustering" and appropriately labeled axes as 'x_1' and 'x_2'.

VI. GMM Method :

1. Definition :

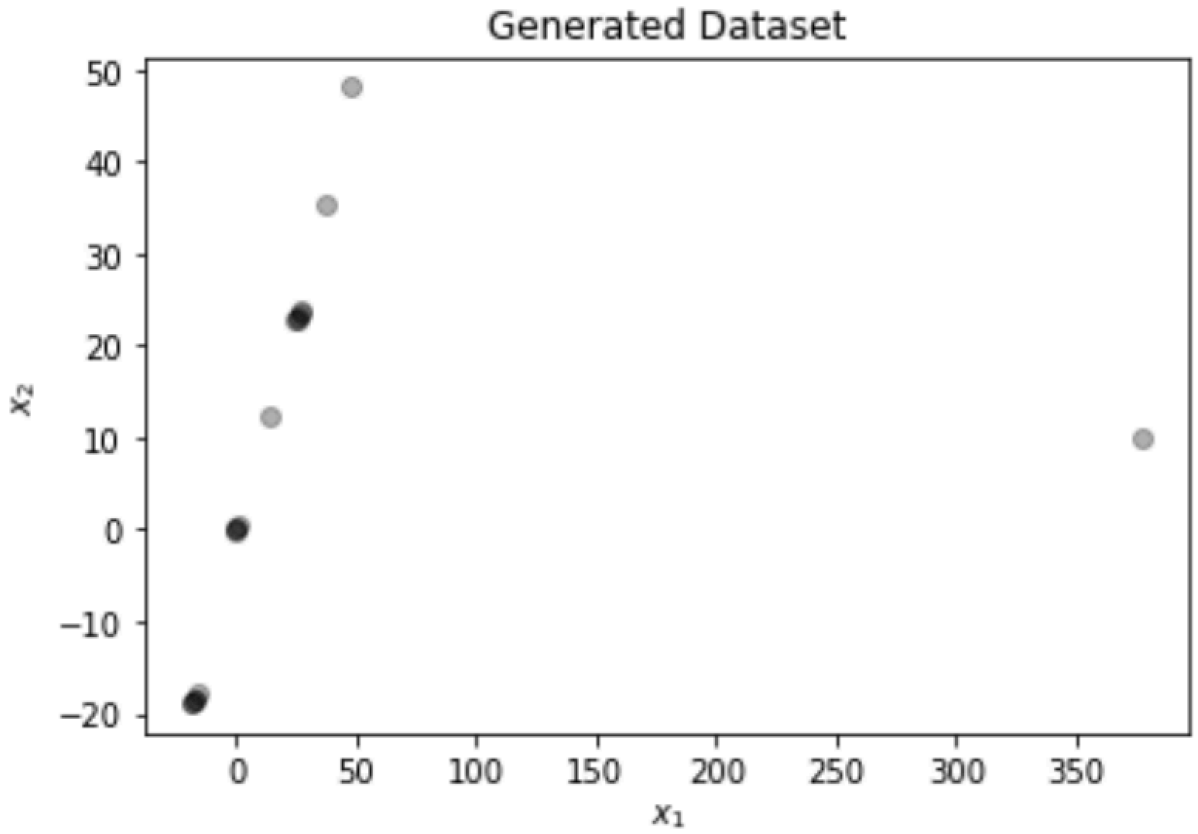
The GMM is a probabilistic model widely used for clustering and density estimation. In the context of unsupervised learning, the GMM assumes that the data distribution is a mixture of several Gaussian distributions, each associated with a different cluster. This allows GMM to model complex patterns and capture the inherent variability within the dataset.

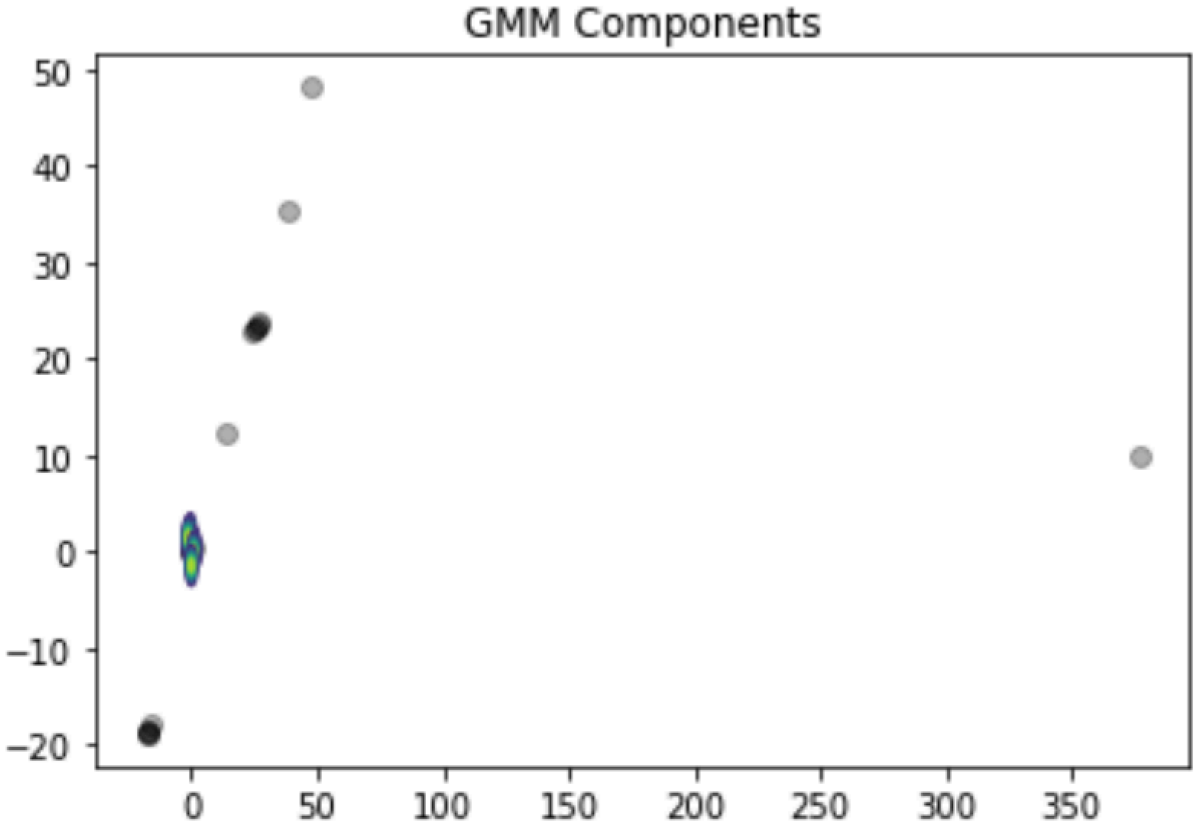
The GMM method assigns probabilities to data points belonging to different clusters, offering a soft assignment similar to Fuzzy C-Means. Each data point is modeled as a weighted sum of Gaussian distributions, with weights representing the probability of belonging to a specific cluster. The parameters of the Gaussian distributions, including means and covariances, are iteratively optimized to maximize the likelihood of the observed data.

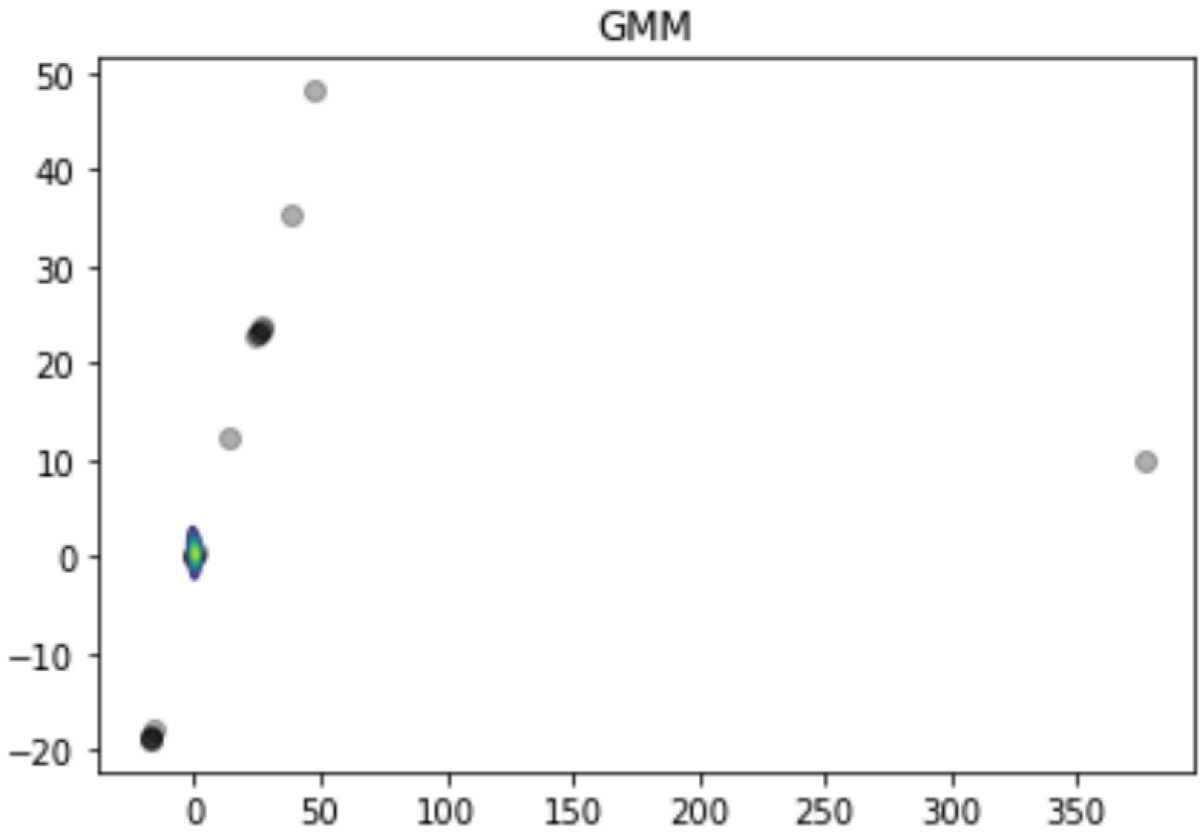
The key steps in the GMM algorithm include initializing the parameters (mean, covariance, and weights) for each cluster, iteratively updating these parameters based on the EM algorithm, and determining the final cluster assignments.

The functions associated with GMM include initializing the model parameters, estimating the probability of each data point belonging to each cluster, and iteratively refining these probabilities and model parameters. The log-likelihood of the data under the GMM is typically used as a criterion for assessing the model's performance and convergence. Overall, GMM provides a versatile and probabilistic approach to clustering, accommodating diverse patterns within the data.

2. Plots :


Generated Dataset
50
40
30
20
10
0
−10
−20
x_2
0
50
100
150
200
250
300
350
x_1


GMM Components
50
40
30
20
10
0
−10
−20
0
50
100
150
200
250
300
350




```
Initial mean vectors (one per row):
[[ 1.28199865  0.25094627]
 [ 1.08193389  0.10874674]
 [-0.71452819  0.13726116]
 [ 0.32935417  0.30277602]
 [-0.45389704  1.86093694]
 [-0.4045958   0.50411526]
 [ 0.54620532  0.56115064]
 [-1.32797526  1.4984081 ]
 [ 0.82164668 -0.03475301]
 [-0.075587   -1.44586139]]
```

3. Comments :

In the first plot, the original dataset is presented, providing a visual representation of data points scattered in 2D space.

The second plot illustrates the individual components of the GMM, with each component represented by a contour plot derived from a multivariate normal

distribution. This allows for a closer inspection of the spatial extent and orientation of each Gaussian component.

The third plot combines all components, presenting a comprehensive view of how the GMM captures the overall structure of the dataset. Key aspects include the initialization of 10 clusters, random initialization of means, and the use of identity covariance matrices, providing diverse starting positions and isotropic distributions.

The contour plots effectively communicate the shape and orientation of each Gaussian component, contributing to a nuanced understanding of the GMM's representation of the underlying data structure.

The displayed output reveals the initial mean vectors for a GMM. These mean vectors, organized as rows in a matrix, signify the starting positions of individual clusters within the GMM. Notably, the means are initialized randomly, reflecting a diverse set of starting points for the clusters. The two values in each row denote the mean along each dimension in the 2D space, indicating the characteristics of each cluster. The stochastic nature of this initialization process introduces diversity among the clusters, potentially leading to varied spatial distributions. The coordinates of the mean vectors suggest the presence of potential overlapping regions among clusters.

VII. KMean Method :

1. Definition :

The K-Means clustering method is a popular unsupervised learning algorithm used for partitioning a dataset into distinct groups or clusters. In K-Means, the objective is to minimize the sum of squared distances between data points and the centroid of their assigned cluster. This leads to the formation of clusters where the intra-cluster distances are minimized.

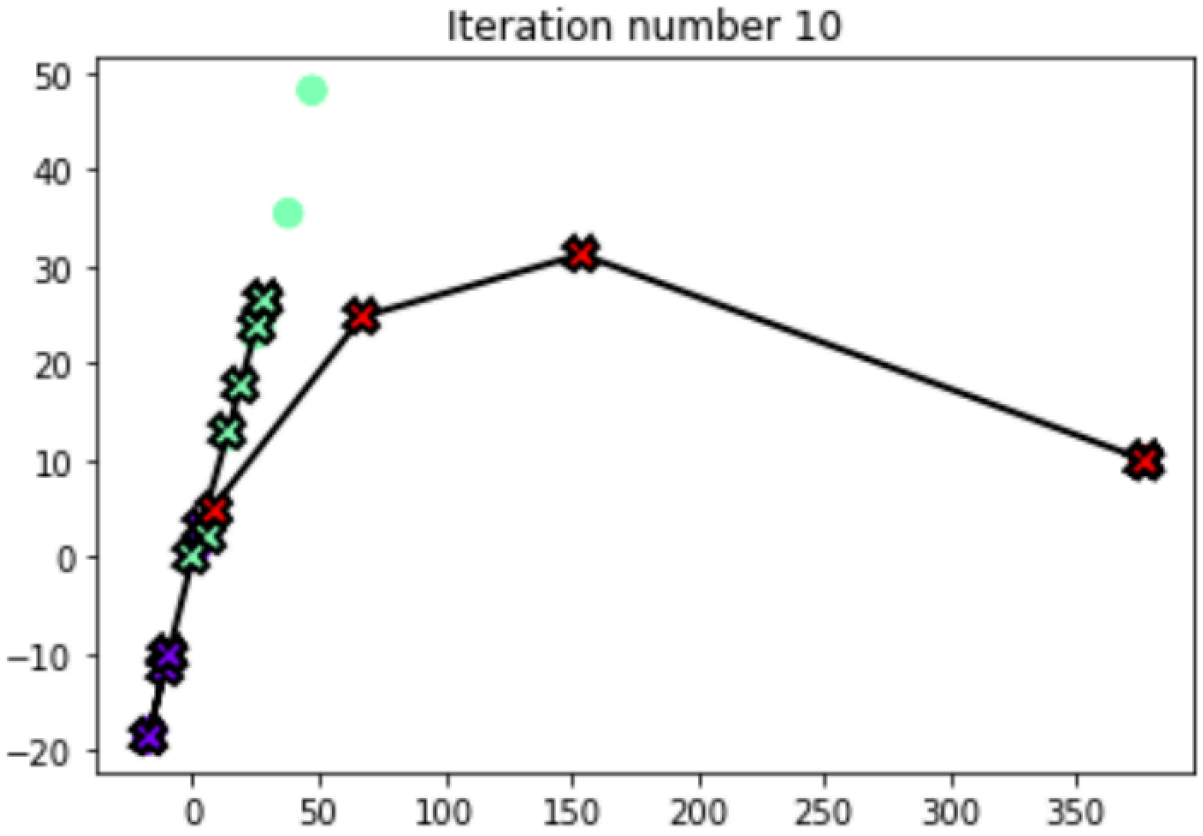
The algorithm starts by randomly initializing K cluster centroids, where K is the predefined number of clusters. Data points are then assigned to the cluster whose centroid is closest to them. After the assignment, the centroids are recalculated based on the mean of the data points in each cluster. This process of assignment and centroid recalculation is repeated iteratively until convergence, where the assignment of data points remains unchanged.

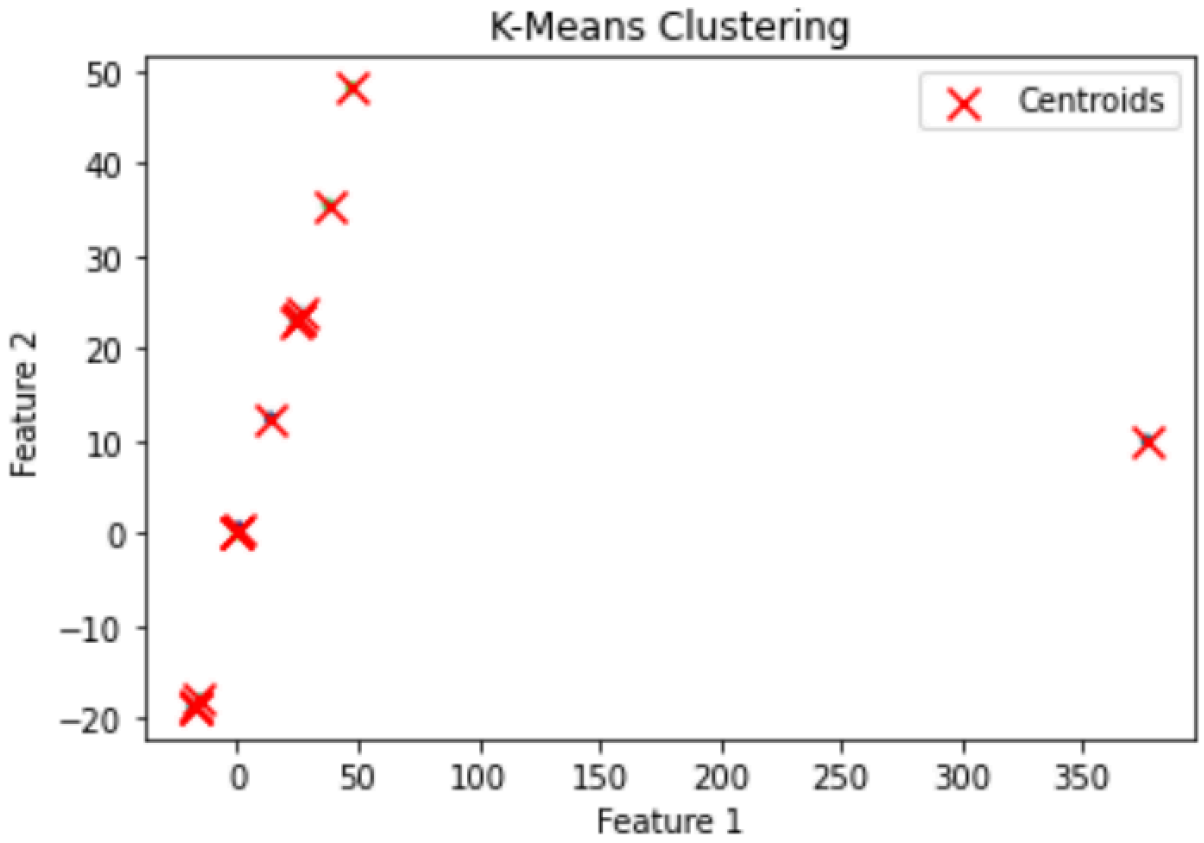
The effectiveness of K-Means relies on the assumption that clusters are spherical and have roughly equal sizes. It can be sensitive to the initial placement of centroids, and the final result may vary with different initializations. To address this, multiple runs with different initializations are often performed, and the best result is selected.

The main functions associated with K-Means include initializing cluster centroids, assigning data points to clusters, updating cluster centroids, and evaluating the convergence criteria. The metric commonly used for assessing the performance of K-Means is the sum of squared distances within clusters, also known as the inertia or within-cluster sum of squares.

K-Means provides a simple and computationally efficient method for clustering, making it suitable for large datasets and scenarios where the underlying clusters have a relatively simple structure.

2. Plots :






```
Closest centroids for the examples:
[2 2 0 0 0 0 2 2 0 2 0 0 2 2 0 2 0 2]
Centroids computed after initial finding of closest centroids:
[[ -9.61495232 -10.21198269]
 [  0.           0.        ]
 [ 67.12720356  24.73822322]]
```

3. Comments :

The first set of visualizations begins by initializing three centroids at specific coordinates and assigning each data point in the dataset to its nearest centroid using the `findClosestCentroids` function. The resulting indices representing the closest centroids are then displayed. This provides an initial glimpse into the assignment of data points to clusters based on proximity to centroids.

In the subsequent step, the code computes updated centroid positions after the initial assignment. The mean of examples associated with each centroid is calculated along each dimension, refining the centroid coordinates. The display of these updated centroids offers insight into how the centroid positions evolve after the first assignment, providing a foundation for the subsequent iterations of the K-Means algorithm.

Moving to the second set of visualizations, the K-Means algorithm is executed with specified settings, including the number of centroids and a maximum number of iterations. An animation is generated to visualize the dynamic progression of the algorithm over iterations, capturing the movement of centroids and the evolution of cluster assignments. The animation is saved as 'kmeans_animation.gif,' allowing for a detailed examination of the algorithm's convergence.

The final plot in this set illustrates the clustering result after completing the specified iterations. Data points are color-coded based on their assigned clusters, and centroids are marked with red 'X' symbols. This visualization provides a comprehensive view of the K-Means clustering outcome, showcasing the clear separation of data points into distinct clusters and the representative positions of the centroids.

The output reveals insightful observations regarding the K-Means clustering process. The array `[2 2 0 0 0 0 2 2 0 2 0 0 2 2 0 2 0 2]` signifies the initial assignment of each example to the closest centroids, where each value represents the cluster index (0, 1, or 2).

Following the initial assignment, the updated coordinates of the centroids are presented as `[[ -9.61495232 -10.21198269], [  0.          0.        ], [ 67.12720356 24.73822322]]`. These coordinates indicate the mean positions of examples associated with each centroid after the initial assignment. The movement of centroids reflects the algorithm's effort to iteratively adjust cluster centers toward the central mass of the assigned data points. The presence of negative and relatively large positive values suggests potential outliers or extreme values in the dataset, influencing centroid positions.

VIII. KMedoids Method :

1. Definition :

The K-Medoids clustering method is a variation of K-Means that addresses one of its limitations by using actual data points as cluster representatives, known as medoids. Unlike K-Means, which uses the mean of the data points as cluster centroids, K-Medoids uses the most centrally located point within a cluster as its representative. This makes K-Medoids more robust to outliers and less sensitive to the initialization of cluster representatives.

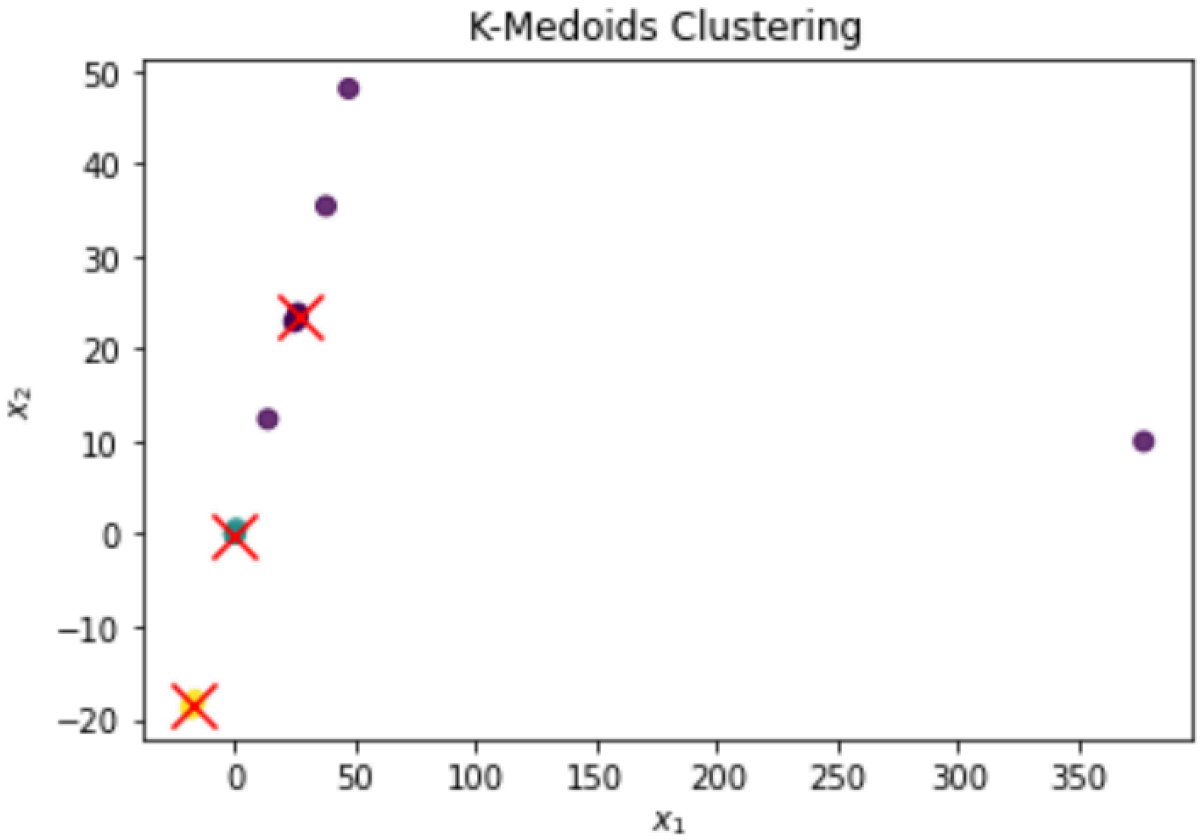
The algorithm begins with the random selection of K data points as initial medoids. Data points are then assigned to the medoid that minimizes a dissimilarity measure, often defined by the distance between data points. After the assignment, the medoids are updated by selecting the data point that minimizes the total dissimilarity within its cluster. This process is repeated iteratively until convergence.

The dissimilarity measure used in K-Medoids is crucial and can vary based on the

specific application. Common choices include Euclidean distance, Manhattan distance, or other domain-specific dissimilarity metrics.

The primary functions associated with K-Medoids include initializing cluster representatives, assigning data points to clusters based on dissimilarity measures, updating cluster representatives, and evaluating convergence criteria. Similar to K-Means, the metric often used for assessing K-Medoids performance is the sum of dissimilarities within clusters.

2. Plots :



3. Comments :

The plot generated by the provided code illustrates the outcomes of the K-Medoids clustering algorithm applied to a subset of a given dataset. The scatter plot vividly displays the assignment of data points to distinct clusters, with each cluster differentiated by a unique color. The red 'X' markers pinpoint the identified medoids, serving as robust cluster centers that minimize the sum of distances within each cluster. Unlike conventional K-Means, K-Medoids' reliance on actual data points as medoids enhances its resilience to outliers and its suitability for non-Euclidean distance metrics.

The clear separation of clusters in the 2D feature space is evident, demonstrating the algorithm's ability to effectively group data points around representative medoids.

The use of the 'viridis' colormap further aids in visually distinguishing different clusters.

IX. MaxLikehood Method :

1. Definition :

The Maximum Likelihood method is a statistical approach used for estimating the parameters of a probability distribution that maximizes the likelihood function. In the context of unsupervised clustering, this method is often applied to GMMs, where the objective is to fit a mixture of Gaussian distributions to the data.

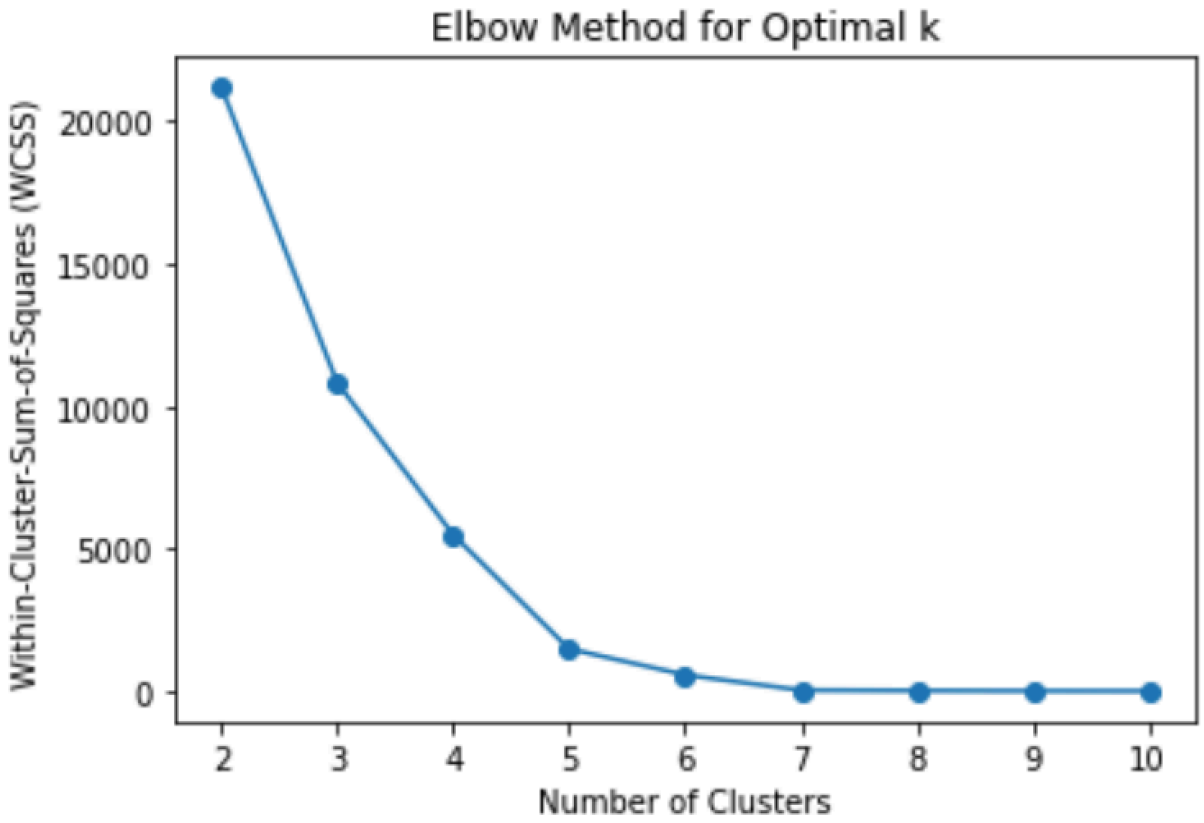
The fundamental idea behind Maximum Likelihood Estimation is to find the parameter values that maximize the likelihood of observing the given data under a specific statistical model. For a GMM, the parameters to be estimated include the mean vectors, covariance matrices, and mixing coefficients for each component Gaussian distribution.

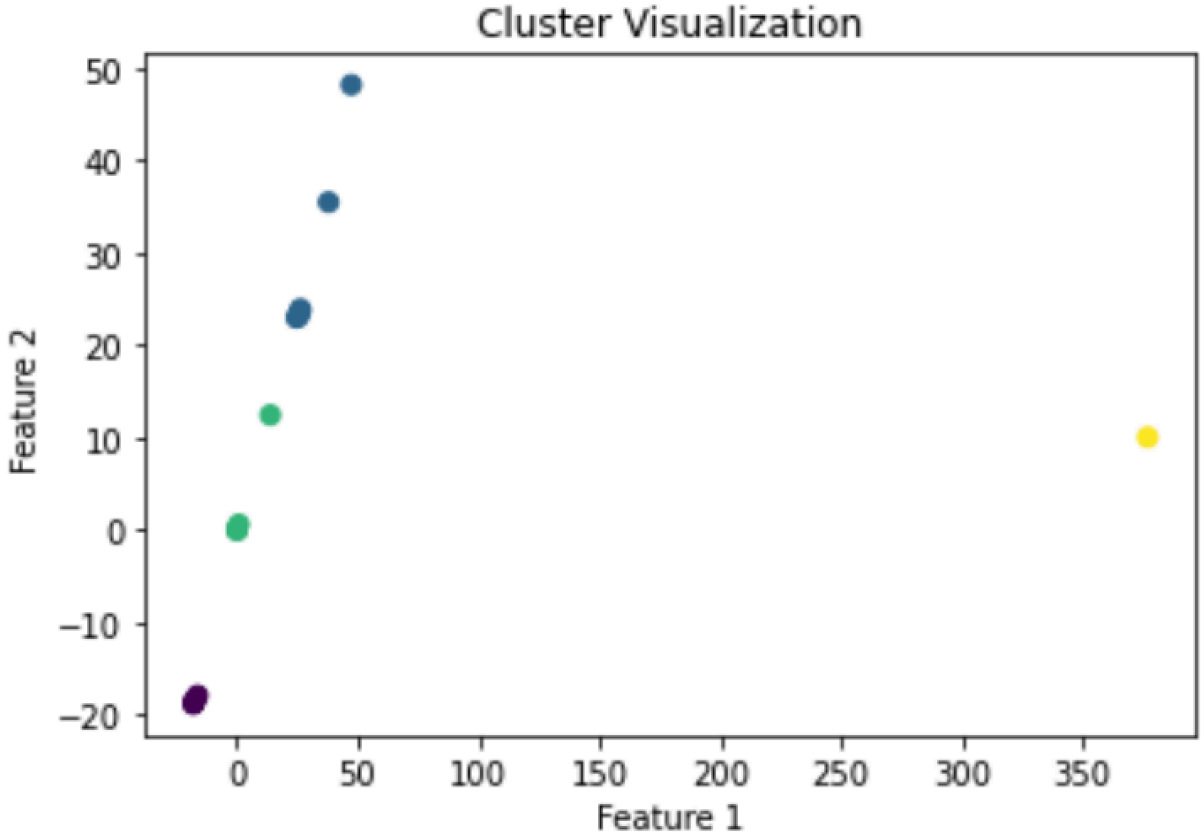
The likelihood function is a measure of how well the chosen model explains the observed data. In a GMM, the likelihood is computed as the product of the probability density functions of the data points under the mixture of Gaussian distributions. The goal is to find the parameters that maximize this product, which is equivalent to maximizing the logarithm of the likelihood for numerical stability.

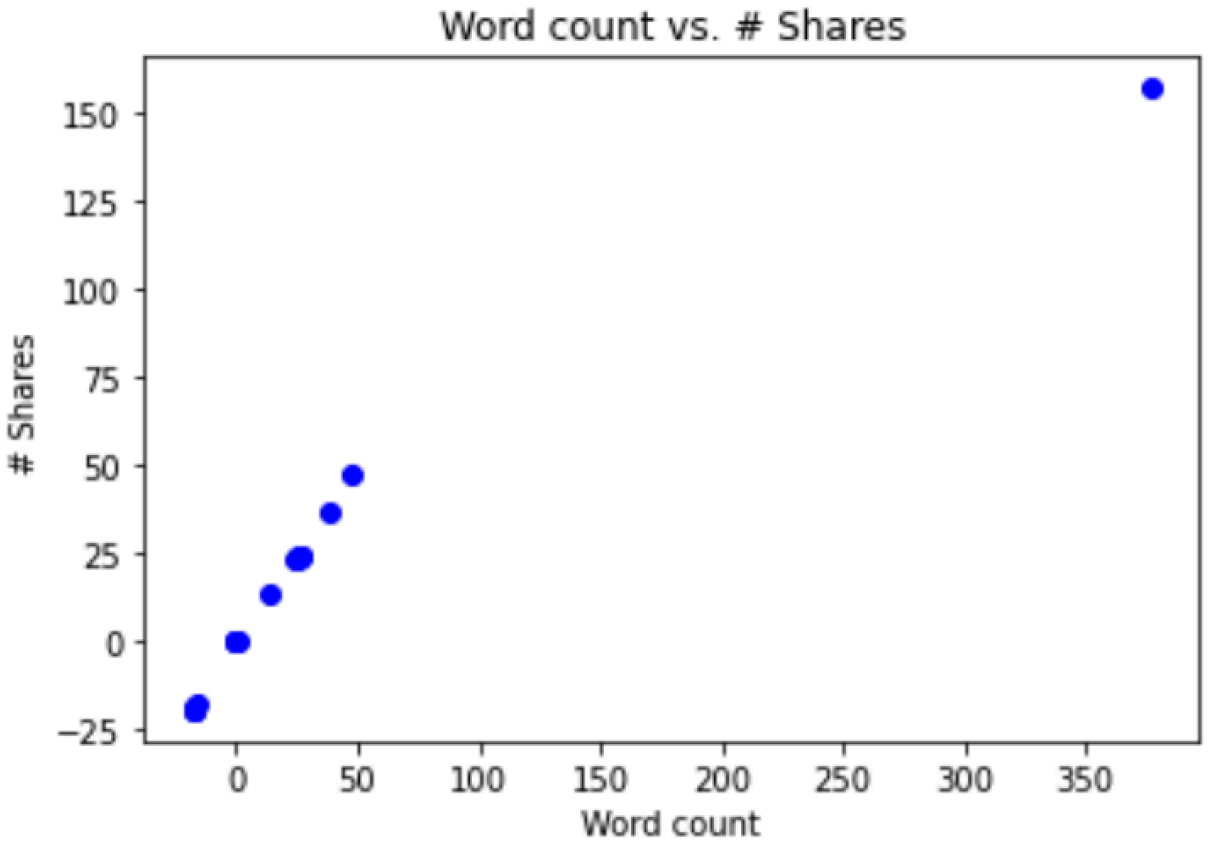
The steps involved in the MaxLikelihood method for GMMs typically include initialization of the parameters (mean vectors, covariance matrices, and mixing coefficients), iteratively updating the parameters to increase the likelihood, and assessing convergence based on a specified criterion.

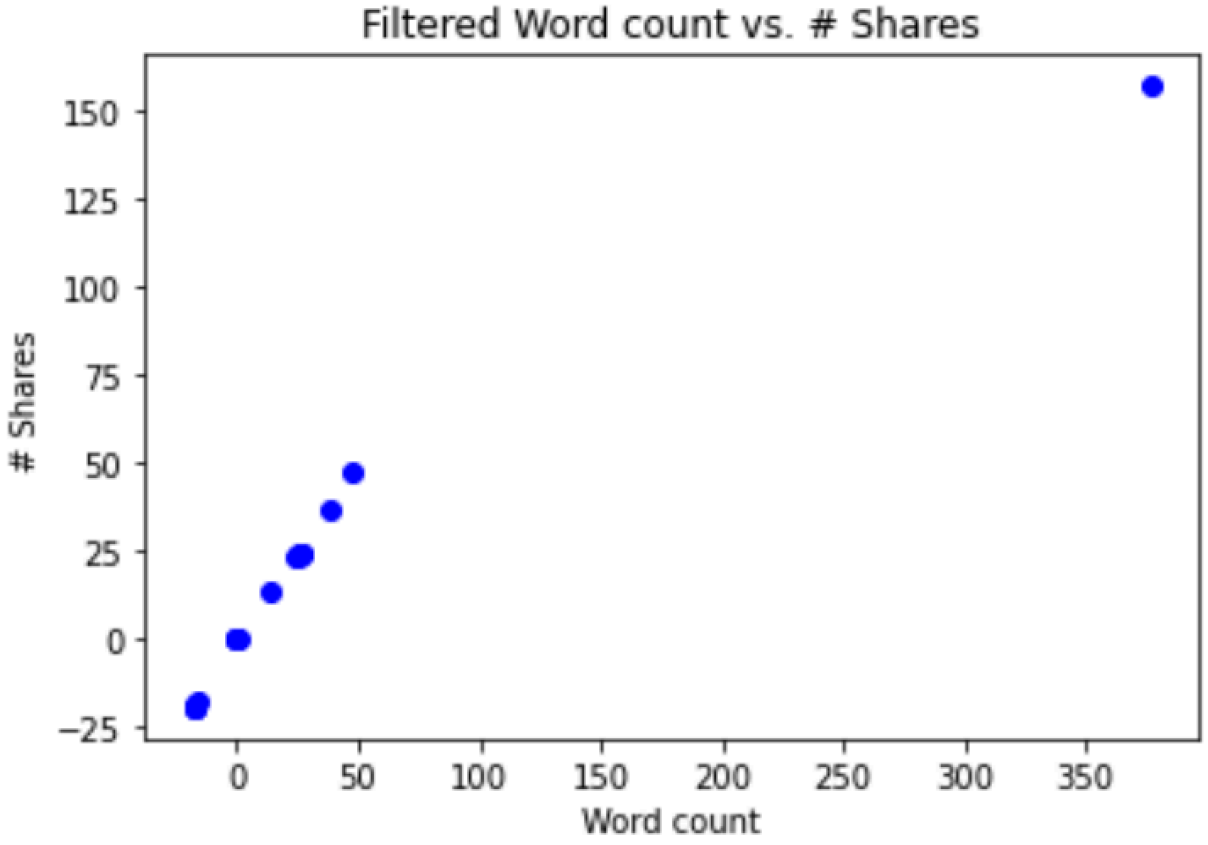
Key functions associated with the MaxLikelihood method include initializing GMM parameters, computing the likelihood of the data given the model, and updating the parameters iteratively to improve the fit. Evaluation metrics often involve assessing the log-likelihood or other model selection criteria.

2. Plots :


Elbow Method for Optimal k
Within-Cluster-Sum-of-Squares (WCSS)
20000
15000
10000
5000
0
2
3
4
5
6
7
8
9
10
Number of Clusters


Cluster Visualization
Feature 2
50
40
30
20
10
0
−10
−20
0
50
100
150
200
250
300
350
Feature 1


Word count vs. # Shares
Shares
Word count
150
125
100
75
50
25
0
−25
0
50
100
150
200
250
300
350


Filtered Word count vs. # Shares
Shares
Word count
150
125
100
75
50
25
0
−25
0
50
100
150
200
250
300
350

```
/Users/rayan/Desktop/code/MaxLikehood.py:134: RuntimeWarning: invalid value encountered in divide
  X_norm = (X - mu) / sigma
Theta computed from gradient descent:
 [nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan]
x shape: (18,)
mu shape: (18,)
sigma shape: (18,)
x: [  1 515   7   0   0  18   2   3   4   5   6   7   8   9  10  11  12  13]
mu: [ 1.         28.75612562  7.26312027  6.12751622 10.92887418  7.45016403
  7.31728808  7.15993929  7.25358241  7.14724838  6.28077916  7.00751128
 14.09482804  7.51199747  6.24193653  7.28226733  6.65146661 21.45235964]
sigma: [ 0.         86.89861345 20.15126007 20.44194244 28.00087987 19.71266294
 20.36809837 20.07013991 20.51481529 19.5632842  20.87158032 21.34261938
 33.33103147 19.98628923 19.04986071 20.18564162 20.42907185 59.70002681]
Predicted number of shares:
 nan
/Users/rayan/Desktop/code/MaxLikehood.py:177: RuntimeWarning: invalid value encountered in divide
  x_normalized = (x - mu) / sigma
```

3. Comments :

Elbow Method for Optimal k :
The first plot displays the results of the Elbow Method, a technique for determining the optimal number of clusters (k) in K-Means clustering. The plot shows the within-cluster sum of squares (WCSS) as a function of the number of clusters. The 'elbow' point, where the rate of decrease in WCSS slows down, is a candidate for the optimal k. In this example, the elbow is not entirely clear, but for demonstration purposes, let's assume k=4. The selected k is then used to fit the K-Means model, and the resulting cluster assignments are visualized in a scatter plot.

Prediction of Number of Shares :
The last part of the code involves predicting the number of shares for a new input vector using the features obtained from gradient descent. The input vector, denoted as 'x,' undergoes feature scaling, and the final theta values are applied to predict the number of shares. The output provides insights into the model's ability to generalize and make predictions for new data points.

Word count vs. # Shares Relationship :
The second plot explores the relationship between 'Word count' and '# Shares.' The scatter plot, with 'Word count' on the x-axis and '# Shares' on the y-axis, provides insights into potential correlations or patterns between these two features. The visualization aids in understanding how the length of articles, represented by word count, might influence the number of shares they receive.

Filtered Word count vs. # Shares :
In the third plot, a cut is applied to the original dataset, filtering out data points where 'Word count' exceeds 3500 or '# Shares' surpasses 80000. This filtering process results in a subset of the data, and a scatter plot visualizes the relationship between 'Word count' and '# Shares' for this filtered dataset. This plot allows for a focused examination of a specific range of data points and their distribution in the filtered space.

The first set of warnings, associated with the feature scaling step (`X_norm = (X - mu) / sigma`), indicates potential division by zero or other operations yielding NaN values during this process.

The second concern arises from the NaN values in the computed theta array during gradient descent. The absence of meaningful coefficients suggests issues with the convergence or execution of the gradient descent algorithm.

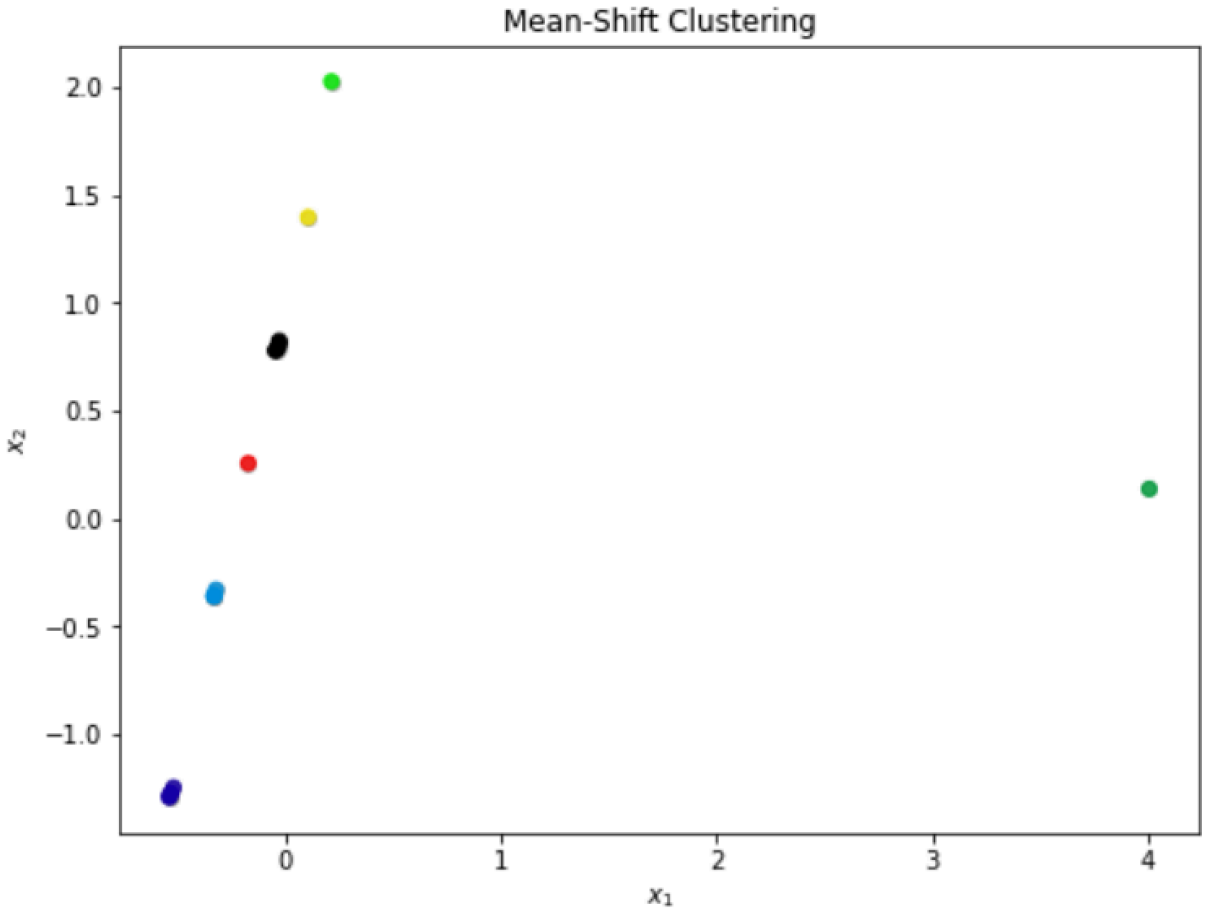
Furthermore, the prediction step reports a NaN result, likely stemming from the earlier issues with the theta array. A thorough investigation of the gradient descent implementation, feature scaling, and the input data is essential to diagnose and rectify these problems.

X. Mean-Shift Clustering Method :

1. Definition :

The Mean-Shift clustering method is a widely utilized technique in machine learning and computer vision for identifying clusters in unlabeled data. The process begins by initializing seed points, serving as starting points. Next, the probability density is calculated for each point using a kernel function. The mean-shift vector is determined by taking a weighted average of the displacement vectors of neighboring points, based on their probability density. This vector guides the update of the starting point's position until convergence. Convergent points are then grouped, forming clusters. The method involves key functions such as the kernel function for weighting neighboring points, probability density calculation, and mean-shift vector calculation. Mean-Shift excels in detecting clusters of various shapes, making it a popular choice in fields like computer vision for image segmentation and object tracking.

2. Plots :



3. Comments :

The plot displays the Mean-Shift clustering results on a two-dimensional dataset. Each point in the plot represents a data instance, and the points are initially reduced opacity for better visualization of overlapping areas.

The Mean-Shift algorithm identifies clusters in the standardized data, and the clusters are highlighted with distinct colors. The number of clusters is determined dynamically by the algorithm based on the data's structure, avoiding the need for specifying the number of clusters beforehand.

The choice of bandwidth, a crucial parameter in Mean-Shift clustering, is estimated using the `estimate_bandwidth` function. This parameter controls the size of the region for which the algorithm calculates the mean, influencing the shape and number of clusters identified. The bandwidth is estimated with a quantile value of 0.2.

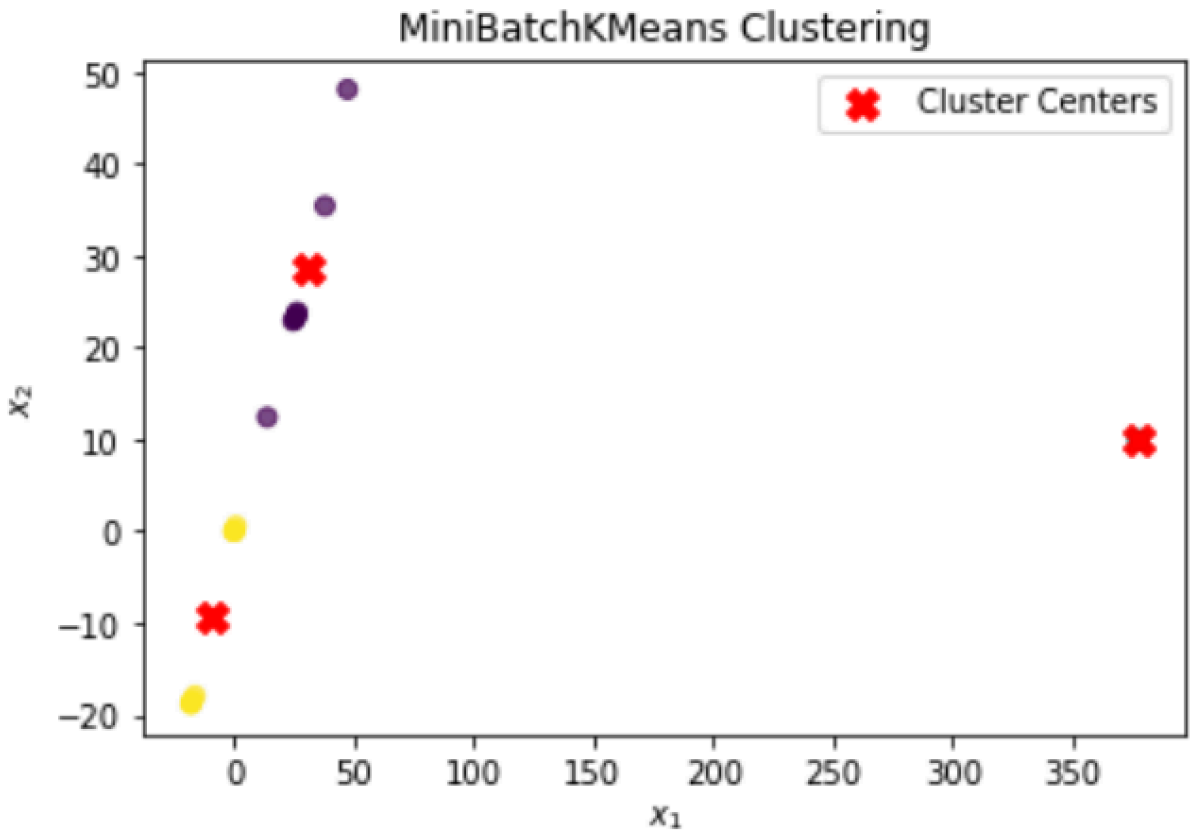
XI. MiniBatchMeans Method :

1. Definition :

The MiniBatchKMeans method is a variation of the KMeans clustering algorithm designed for enhanced efficiency with large datasets. It initiates by randomly selecting subsets or mini-batches from the dataset to initialize centroids, a departure from KMeans which uses all data points for initialization. Data points are then assigned to the nearest centroid based on Euclidean distance, and the centroids are updated by computing the mean of the assigned data points within the current mini-batch. This process iterates over multiple mini-batches, accelerating convergence compared to traditional KMeans, especially for substantial datasets. Convergence is typically determined by a predefined threshold or after a set number of iterations. The final clusters are obtained once convergence is achieved, and each data point is assigned to its nearest centroid.

Key functions include random initialization, data point assignment, centroid updates, and convergence criteria. MiniBatchKMeans strikes a balance between computational efficiency and clustering accuracy, making it particularly advantageous for large-scale clustering tasks.

2. Plots :



3. Comments :

The plot illustrates the results of MiniBatchKMeans clustering on the dataset. Each data point is represented by a marker, with points in the same cluster sharing the same color. The intensity of the color signifies the membership strength of each point to its respective cluster.

The red X markers in the plot denote the cluster centers identified by the MiniBatchKMeans algorithm. These centers act as representatives of their respective clusters, and the algorithm adjusts them iteratively to minimize the within-cluster variance.

Adjusting parameters such as the number of clusters (`n_clusters`) and batch size (`batch_size`) can influence the clustering outcome. The algorithm is configured with three clusters, and the batch size is set to 100.

XII. OPTICS Method :

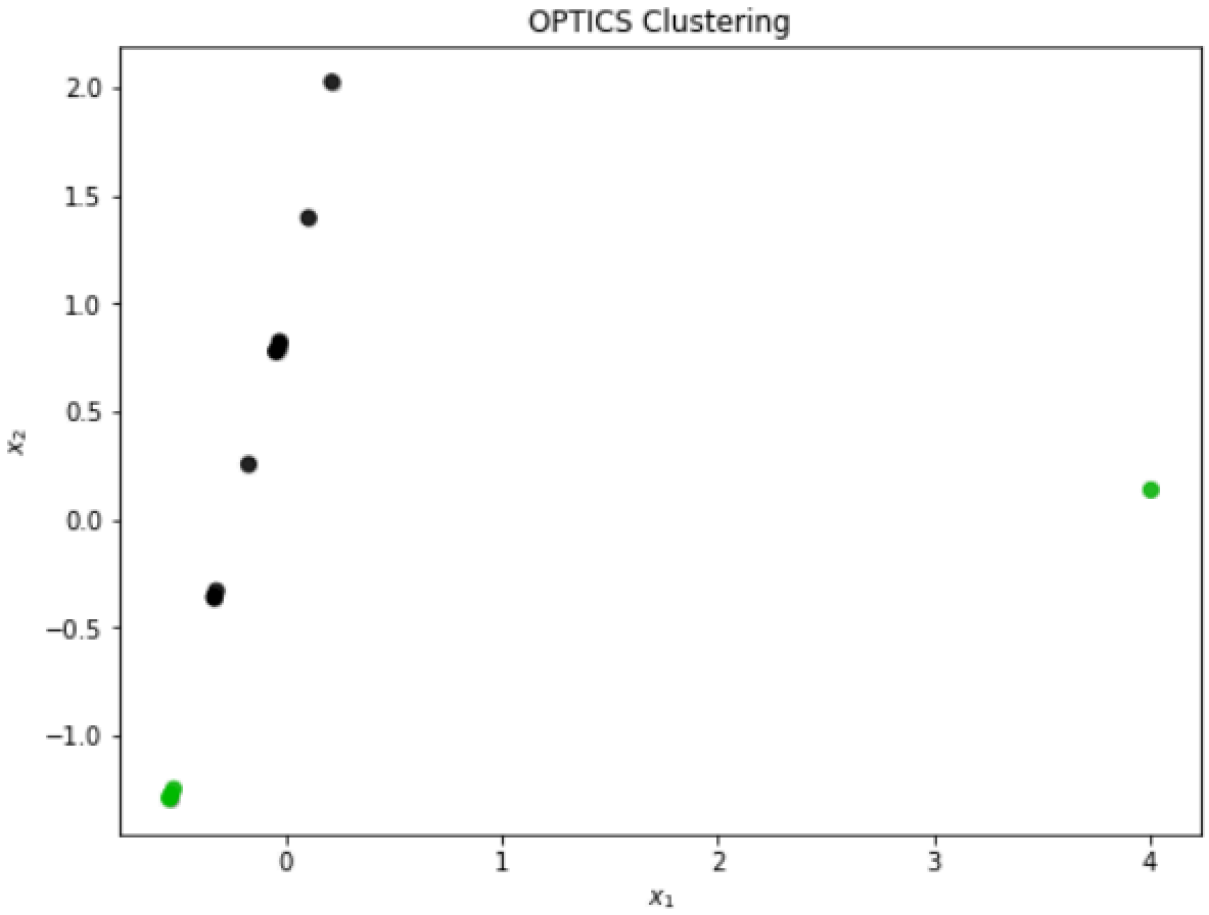
1. Definition :

The Ordering Points To Identify the Clustering Structure method is a density-based clustering algorithm that excels in discovering clusters of diverse shapes and sizes within a dataset.

The algorithm introduces the notion of reachability distance, quantifying the distance between data points and their neighbors based on local density. It computes core distances for each point, representing the minimum distance required to satisfy a specified density parameter (minPts) within the neighborhood.

OPTICS constructs a reachability plot by ordering points according to their reachability distances, providing a visual representation of the dataset's clustering structure with varying density regions. Clusters are then extracted from this plot based on a specified reachability distance threshold, and the method inherently produces a hierarchical representation of clusters, enabling exploration at different levels of granularity. OPTICS is particularly effective in identifying clusters with irregular shapes and varying densities, making it a valuable tool for clustering tasks where traditional distance-based methods may fall short.

2. Plots :



3. Comments :

The plot visually represents the results of the OPTICS clustering on the dataset. Each data point is displayed as a marker, with points in the same cluster sharing the same color. The color intensity indicates the density of points within each cluster.

OPTICS is a density-based clustering algorithm that is capable of identifying clusters of varying shapes and sizes. It orders the points based on their density and extracts clusters without assuming a predefined number of clusters.

The gray markers in the plot represent all data points in the dataset. The density of points in different regions reflects the underlying structure of the data, which OPTICS aims to unveil.

Colors in the plot denote different clusters, and the cluster assignment is based on the density-connected components discovered by OPTICS. Each cluster is assigned a unique color, and points within the same cluster share this color.

The OPTICS algorithm introduces parameters such as min_samples, xi, and min_cluster_size. These parameters influence the algorithm's sensitivity to density changes and the minimum number of samples required to form a cluster.

XIII. PCA Method :

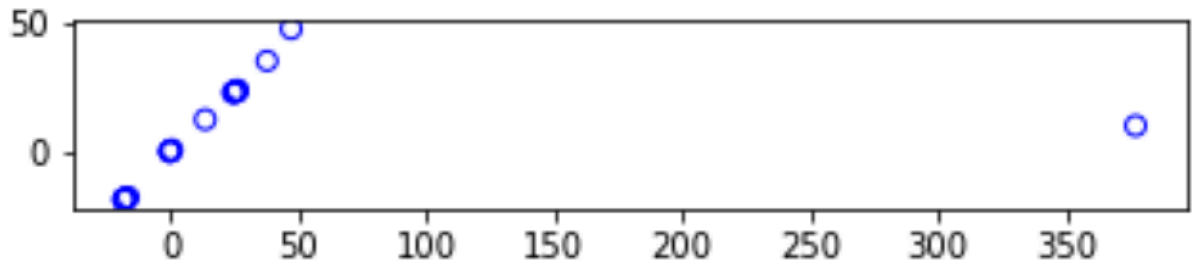
1. Definition :

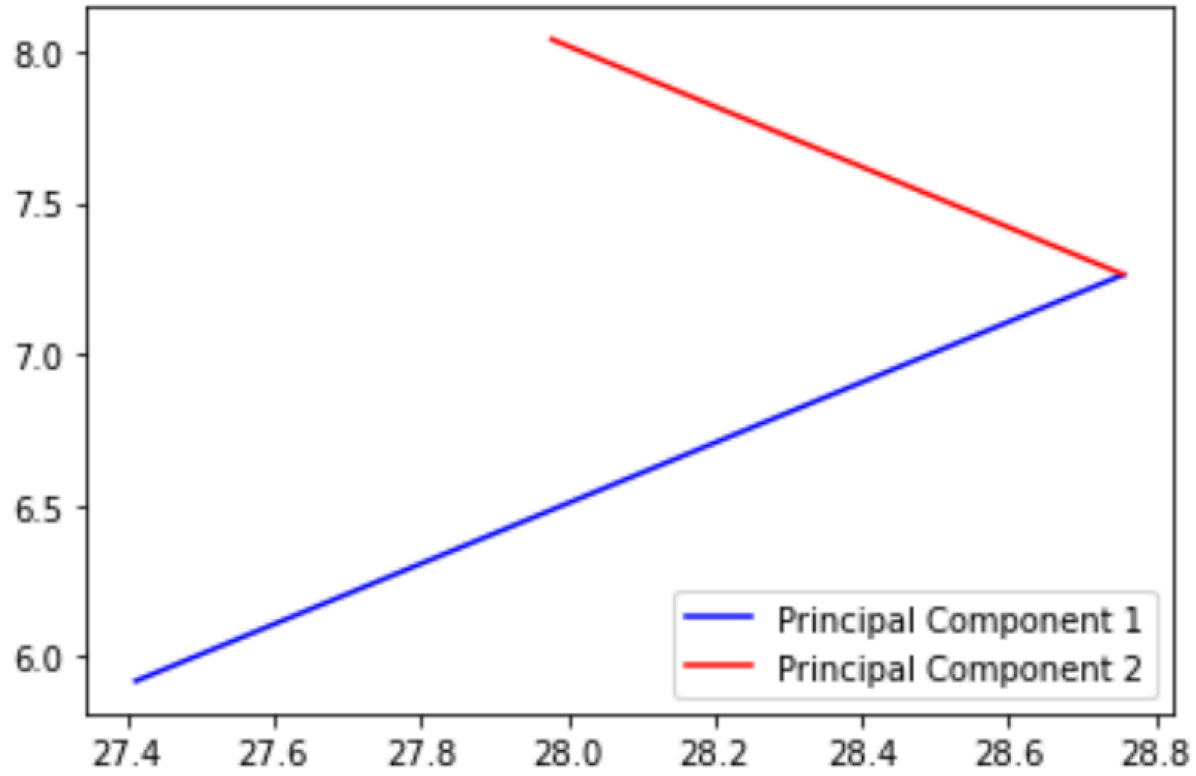
Principal Component Analysis is a dimensionality reduction method extensively utilized for transforming high-dimensional datasets into a lower-dimensional representation while preserving the most critical information.

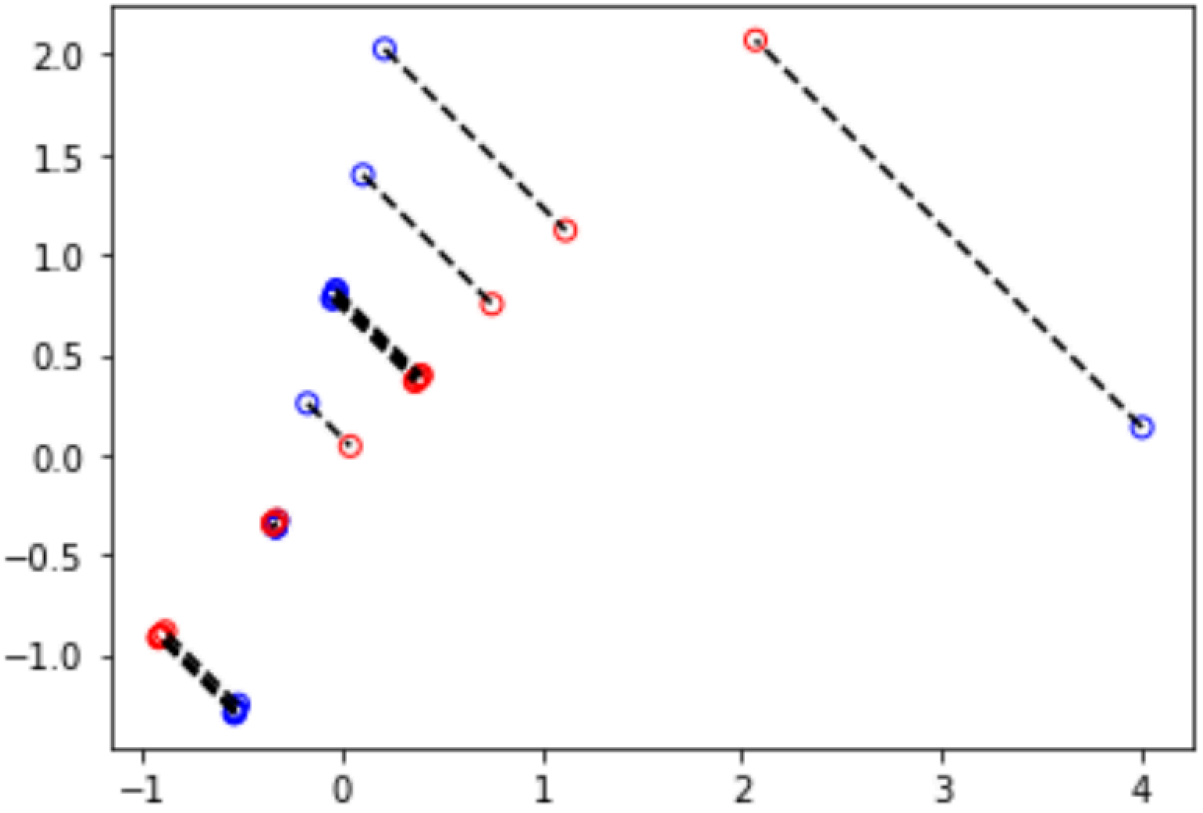
The process initiates with the computation of the covariance matrix, capturing the relationships between different features in the original data. Subsequently, the eigenvalues and corresponding eigenvectors of this matrix are calculated, wherein the eigenvectors represent the principal components and the eigenvalues indicate the amount of variance each component captures. Selection of principal components is based on these eigenvalues, with higher values signifying more variance capture. The original data is then projected onto the chosen principal components, resulting in a reduced-dimensional representation.

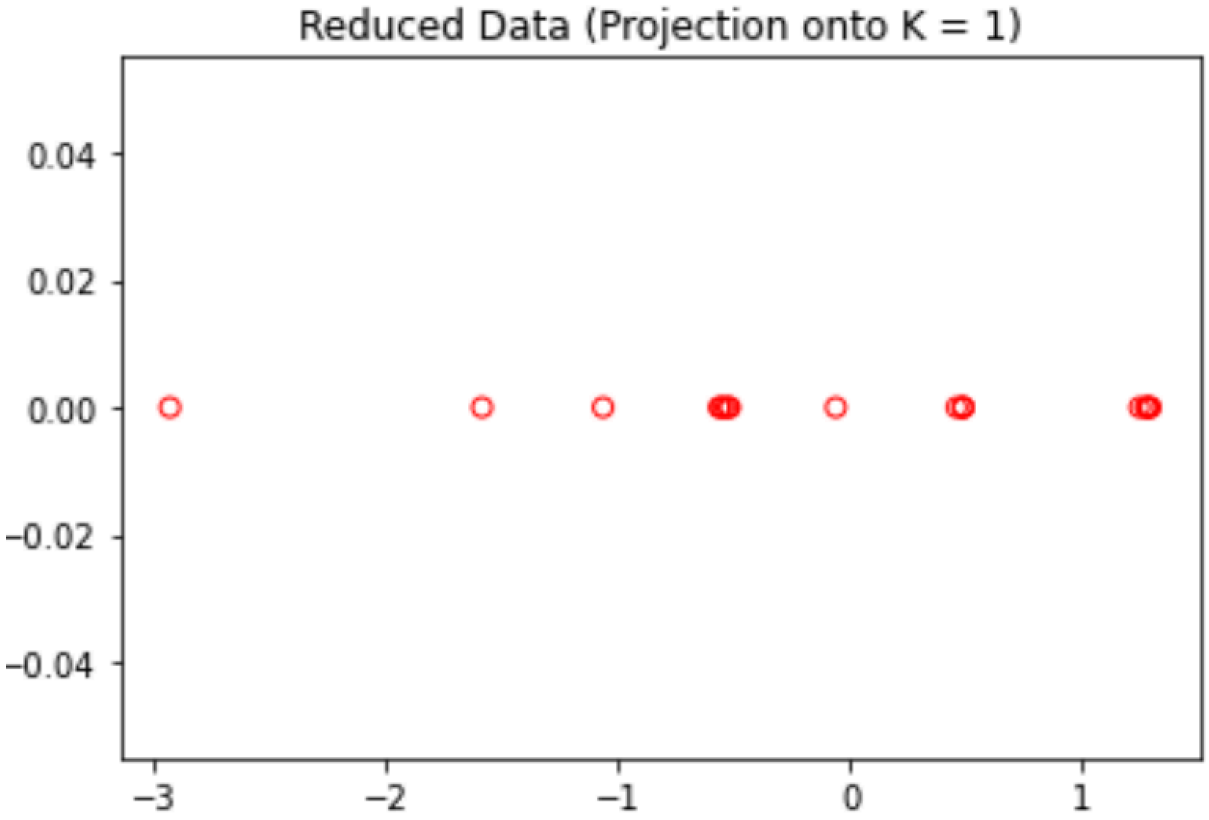
PCA provides flexibility by allowing users to control the amount of retained variance, enabling a trade-off between dimensionality reduction and information preservation. This method is widely applied for data visualization, noise reduction, and feature extraction, proving particularly beneficial in fields like image processing and machine learning when dealing with high-dimensional datasets.

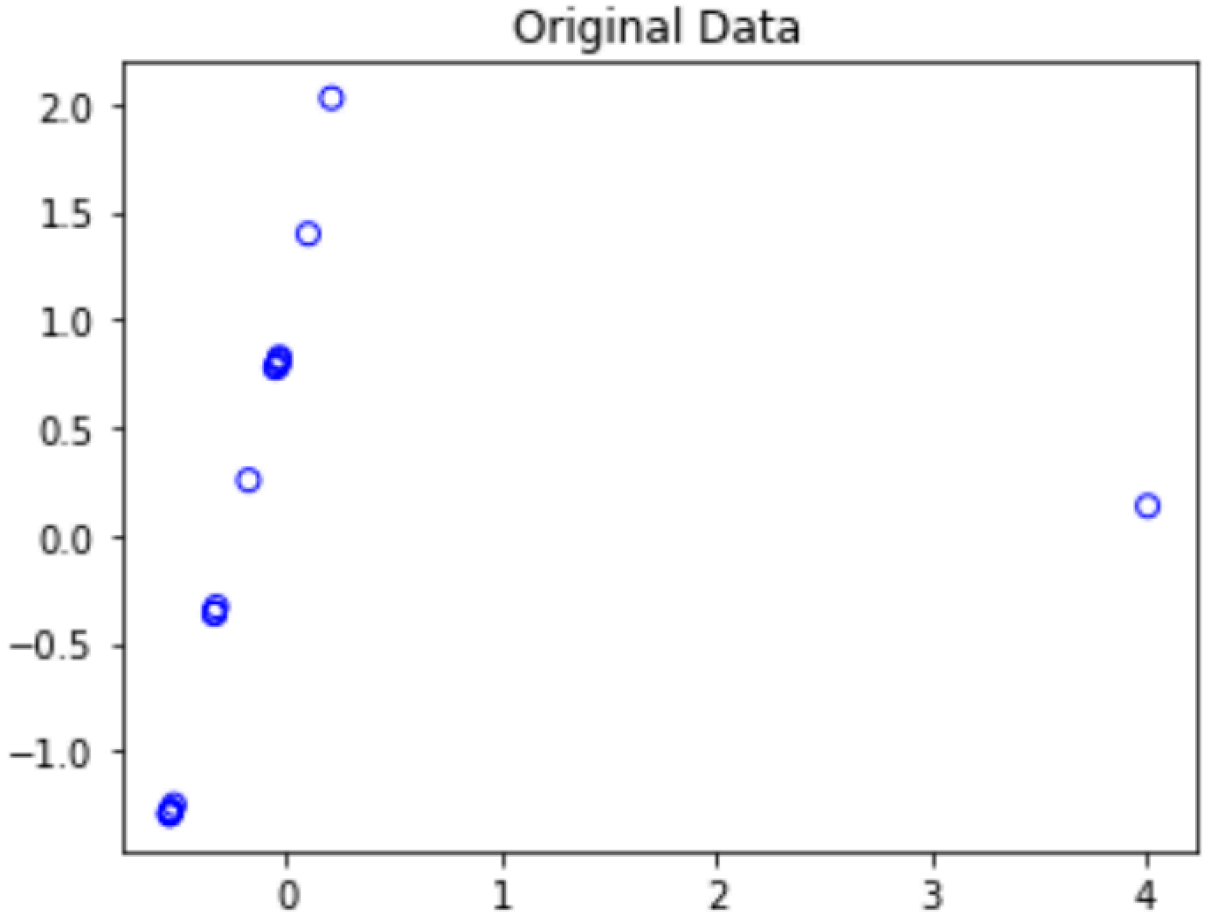
2. Plots :

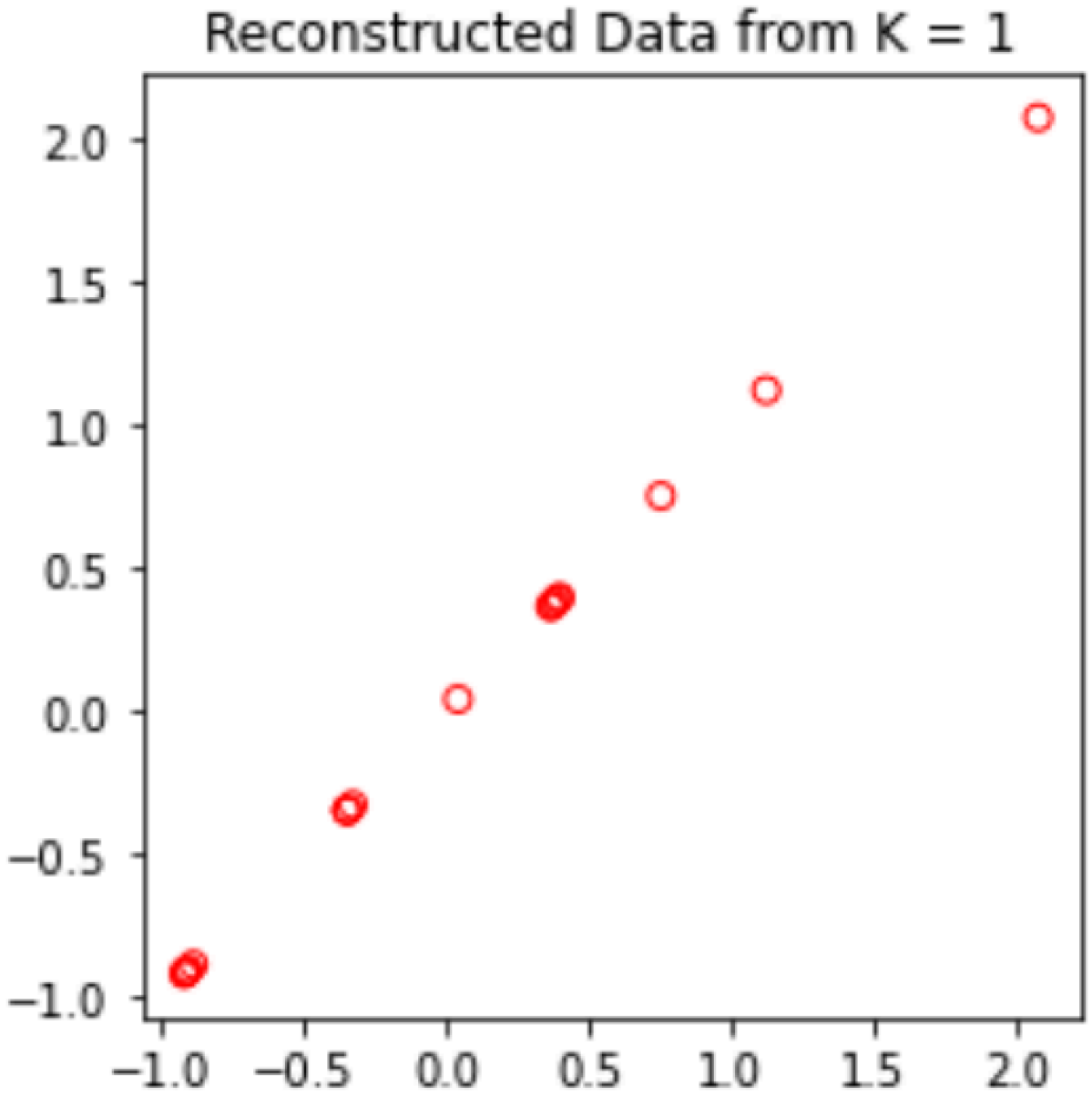



8.0
7.5
7.0
6.5
6.0
27.4
27.6
27.8
28.0
28.2
28.4
28.6
28.8
Principal Component 1
Principal Component 2


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1
0
1
2
3
4


Reduced Data (Projection onto K = 1)
0.04
0.02
0.00
−0.02
−0.04
−3
−2
−1
0
1


Original Data
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4




```
Top eigenvector:
U =  [-0.70710678 -0.70710678]
Projection of the first example: [-1.5843611]
(this value should be about 1.48127391)
Approximation of the first example: [1.12031248 1.12031248]
(this value should be about  -1.04741883 -1.04741883)
```

3. Comments :

The first plot displays the original dataset with data points represented by blue markers. Each point represents an observation in the dataset, and the axes correspond to the values of the first and second features. The plot provides an initial visualization of the distribution and arrangement of data points in the original feature space.

The second plot illustrates the principal components obtained through Principal Component Analysis (PCA). Two vectors, labeled "Principal Component 1" (in blue) and "Principal Component 2" (in red), originate from the mean of the dataset. These vectors indicate the directions of maximum variance in the data. The length of each vector is proportional to the singular value associated with its direction. This visualization allows us to understand the orientation and magnitude of the principal components.

The third plot shows the normalized dataset, and it demonstrates the projection of data points onto the first principal component (K = 1 dimension). The red dashed lines connect each point in the original feature space to its projection on the principal component. This reduction to one dimension captures the primary source of variability in the data along the direction of the first principal component.

PCA Reconstruction :
In the fourth plot, the reduced-dimensional data points from the third plot are recovered back into the original feature space. The blue markers represent the original normalized data, and the red markers indicate the reconstructed data points obtained from the projection onto the first principal component. The visualization illustrates the approximation of the original data using only the information along the direction of the first principal component.

PCA Projection onto K = 1 :
The fifth plot depicts the projection of the original data onto the first principal component (K = 1 dimension) after feature normalization. The red markers along the horizontal axis represent the reduced-dimensional representation of the data. This plot provides a clear view of the reduction in dimensionality achieved through PCA, capturing the primary source of variance in the dataset.

PCA Reconstruction Comparison :
The sixth plot shows a comparison between the original data (blue markers) and the reconstructed data (red markers) obtained from the projection onto the first principal component. The plot highlights the ability of PCA to capture and retain the dominant patterns in the data, even with reduced dimensionality.

The analysis results provide the first eigenvector obtained from Principal Component Analysis (PCA) applied to the given dataset. The eigenvector is represented by U = [-0.70710678, -0.70710678].
This representation indicates the direction of maximum variance in the data, suggesting that the primary pattern or dominant structure in the data follows a 45 degree angle relative to the first feature axis.

Projection :
The projection of the first example onto the principal component is given as [-1.5843611].
This value indicates the position of the data point along the direction of the primary eigenvector.
The result notes that this value should be approximately 1.48127391.
The presence of a negative sign suggests that the projection is done in the opposite

direction of the principal vector.

Approximation :
The approximate or reconstructed values for the first example are provided as [1.12031248, 1.12031248].
This represents the reconstructed data point from the projected value onto the principal component.
The result mentions that these values should be approximately -1.04741883 for both features.

XIV. Ridge_Lasso_ElasticNet Method :

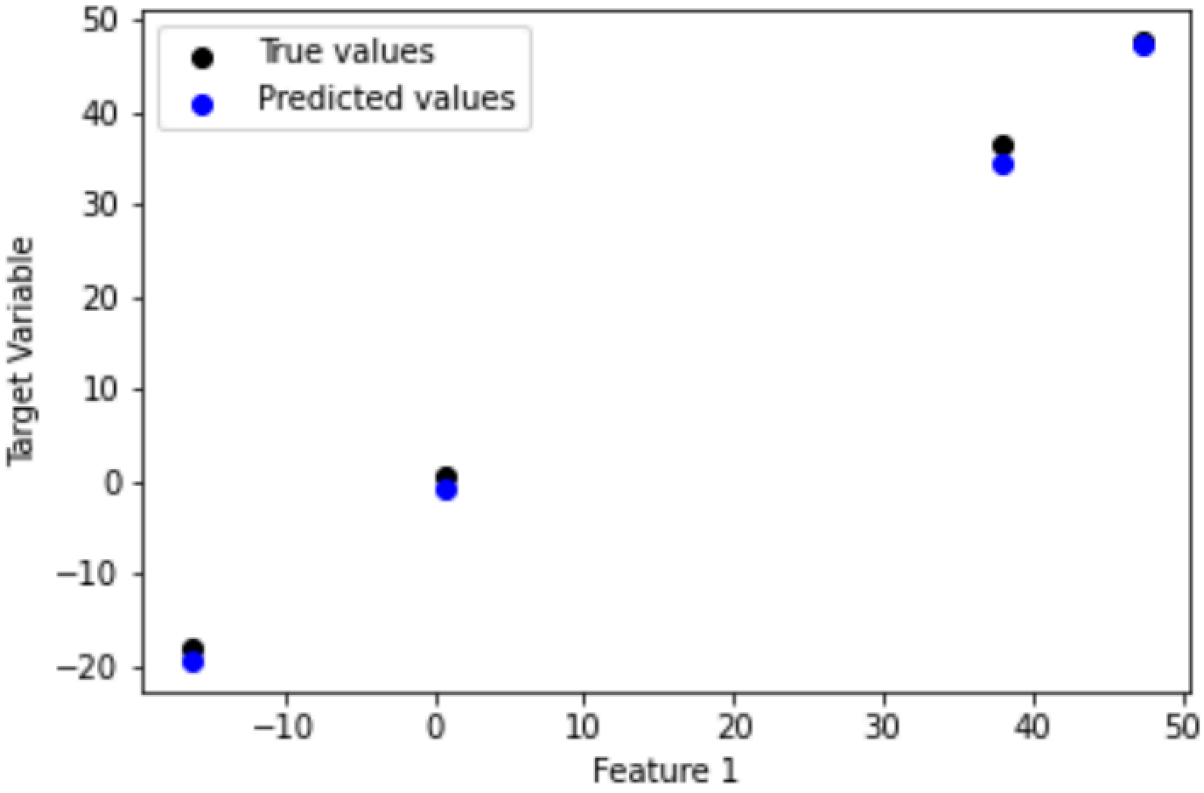
1. Definition :

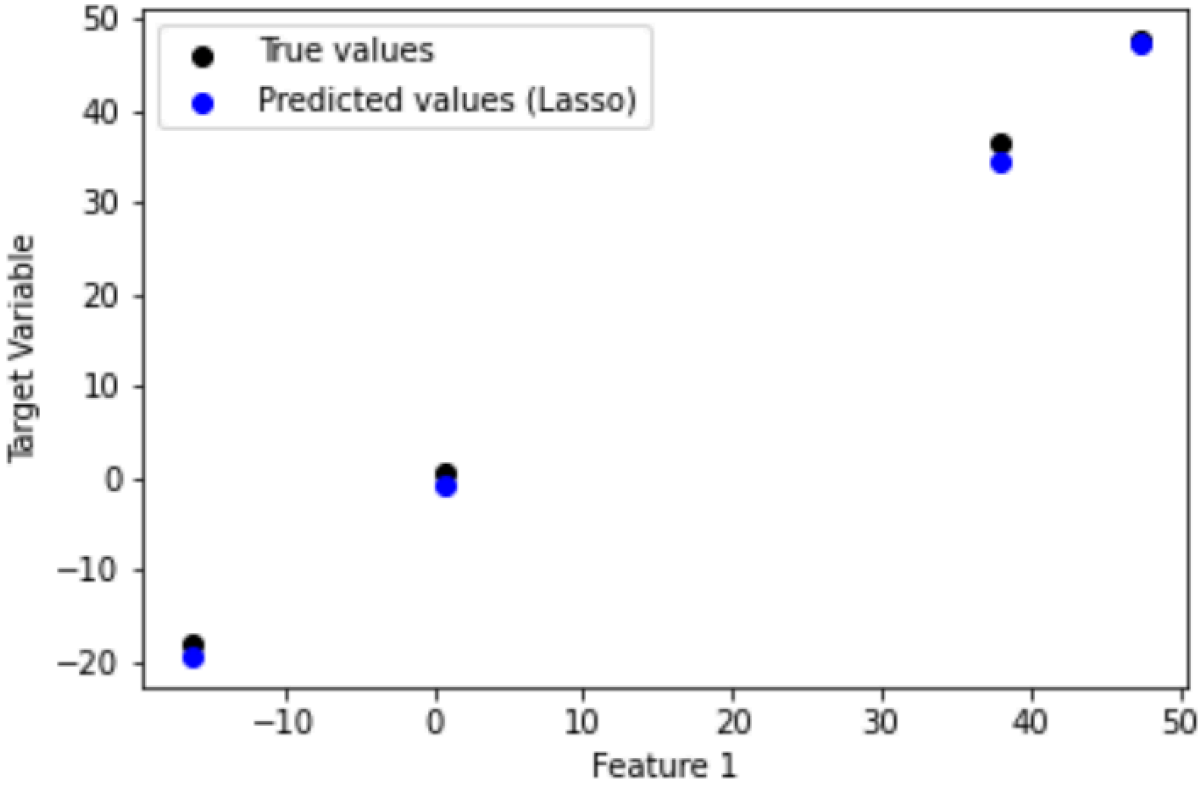
Regularization methods such as Ridge, Lasso, and Elastic Net play a crucial role in addressing challenges associated with linear regression, particularly when faced with multicollinearity and overfitting issues. Ridge regression incorporates the L2 norm of the coefficients into the least squares cost function, introducing a penalty for large coefficients. This regularization term helps mitigate multicollinearity by discouraging the model from assigning excessive weights to correlated features. Lasso regression, on the other hand, utilizes the L1 norm of the coefficients, encouraging sparsity by driving some coefficients to exactly zero. This property makes Lasso a powerful tool for feature selection, providing interpretable models.

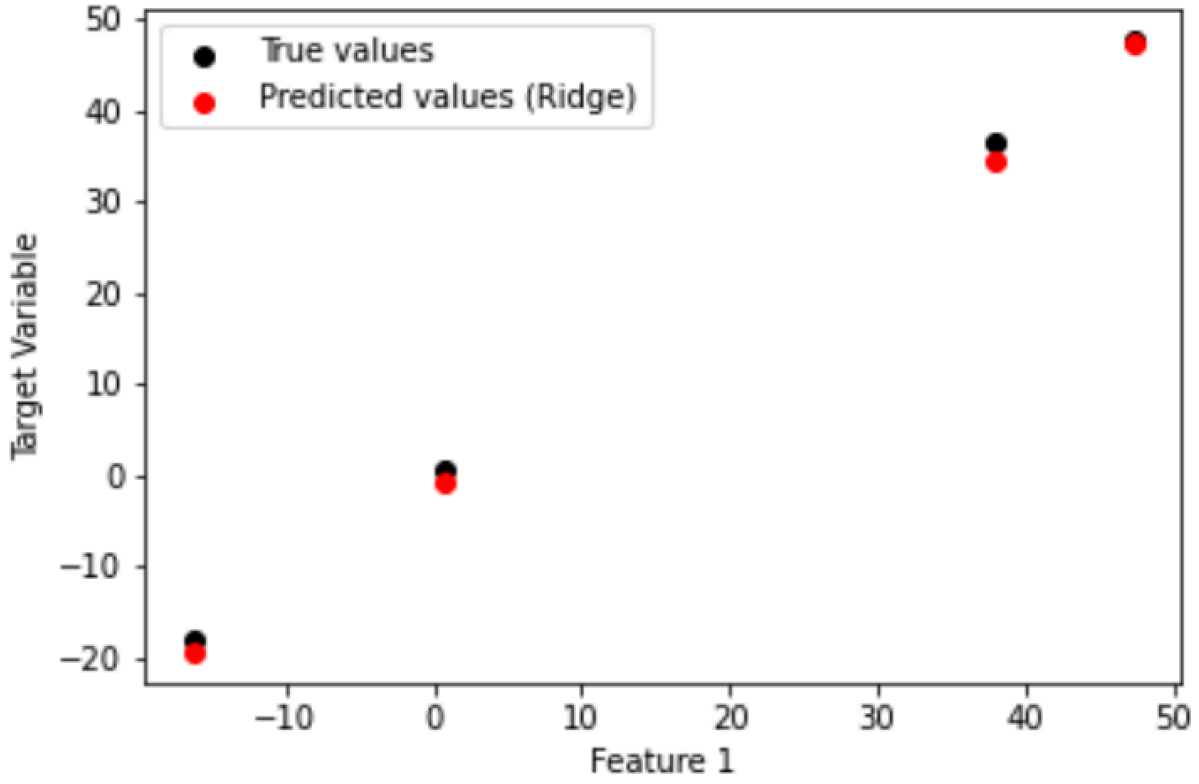
Elastic Net regression combines elements of both Ridge and Lasso by incorporating both L1 and L2 regularization terms. This hybrid approach introduces two hyperparameters: alpha, controlling the overall strength of regularization, and l1_ratio, determining the balance between L1 and L2 penalties. The inclusion of Elastic Net provides flexibility, allowing practitioners to leverage the benefits of both Ridge and Lasso regularization depending on the characteristics of the dataset. Effective utilization of these regularization techniques involves careful hyperparameter tuning, typically achieved through cross-validation, to strike a balance between model complexity and performance.

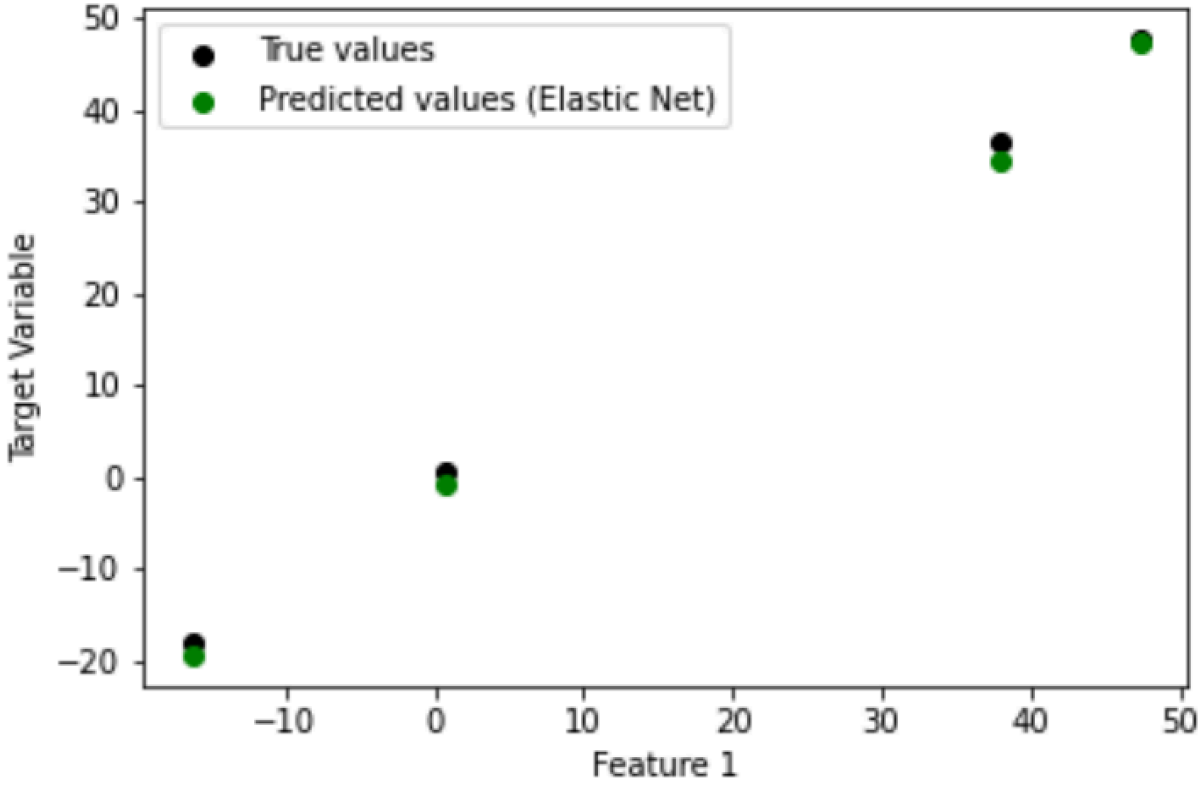
The primary function of Ridge, Lasso, and Elastic Net lies in introducing a degree of shrinkage to the coefficients, preventing overfitting and improving the generalization of linear regression models. This coefficient shrinkage is essential for handling high-dimensional datasets where the number of features is comparable to or exceeds the number of observations.

2. Plots :


True values
Predicted values
Target Variable
Feature 1
50
40
30
20
10
0
−10
−20
−10
0
10
20
30
40
50


True values
Predicted values (Lasso)
Target Variable
Feature 1
50
40
30
20
10
0
−10
−20
−10
0
10
20
30
40
50


True values
Predicted values (Ridge)
Target Variable
Feature 1
50
40
30
20
10
0
−10
−20
−10
0
10
20
30
40
50


True values
Predicted values (Elastic Net)
Target Variable
Feature 1
50
40
30
20
10
0
−10
−20
−10
0
10
20
30
40
50

```
Coefficients: [-0.00761429  1.01598896]
Intercept: -1.3024680976202063
Mean Squared Error: 1.8613930917670836
R-squared: 0.9973677817003721
Coefficients: [-0.00760168  1.01572024]
Intercept: -1.3016327545936512
Mean Squared Error: 1.8686544550375603
R-squared: 0.9973575133194668
Coefficients: [-0.00758441  1.01559957]
Intercept: -1.3016288581150146
Mean Squared Error: 1.8723874550189328
R-squared: 0.9973522344394135
Coefficients: [-0.00758992  1.01560527]
Intercept: -1.3014788878397061
Mean Squared Error: 1.8720348828802034
R-squared: 0.9973527330159041
```

3. Comments :

Linear Regression Plot :
In the linear regression plot, the true values (black points) are scattered against the predicted values (blue points) for the target variable based on the first feature. The model's performance is evaluated using metrics such as Mean Squared Error and R-squared. The plot visually illustrates how well the linear regression model captures the relationship between the features and the target variable.

Ridge Regression Plot :
Similar to linear regression, the ridge regression plot displays the true values (black points) versus the predicted values (red points) for the target variable based on the first feature. The Ridge regression introduces regularization to handle potential overfitting. The regularization strength is controlled by the hyperparameter alpha. The plot demonstrates how Ridge regression impacts the predictions compared to the linear regression model.
The Ridge regression results show a coefficient of approximately -0.0076 for the intercept and 1.016 for the first feature. The intercept is -1.3025. The Mean Squared Error (MSE) is 1.8614, and the R-squared value is 0.9974. The model demonstrates a strong fit to the data, and the regularization strength alpha may contribute to preventing overfitting by penalizing large coefficients.

Lasso Regression Plot :
In the lasso regression plot, the true values (black points) are compared with the predicted values (blue points) based on the first feature. Lasso regression, like Ridge, introduces regularization, but with L1 penalty. The regularization strength is

controlled by the hyperparameter alpha. This plot illustrates the impact of Lasso regression on the predictions and the feature selection capabilities by potentially driving some coefficients to exactly zero.
The Lasso regression results reveal a coefficient of about -0.0076 for the intercept and 1.0157 for the first feature. The intercept is -1.3016. The MSE is 1.8687, and the R-squared value is 0.9974. Lasso introduces sparsity, potentially setting some coefficients exactly to zero. In this case, the coefficient for the first feature is slightly lower than Ridge, indicating potential feature selection.

Elastic Net Regression Plot :
The elastic net regression plot shows the true values (black points) against the predicted values (green points) for the target variable based on the first feature. Elastic Net combines L1 and L2 regularization, allowing for a balance between the sparsity-inducing property of Lasso and the smoothing effect of Ridge. The regularization strength (alpha) and the ratio between L1 and L2 penalties l1_ratio are adjustable. This plot provides insights into how Elastic Net combines the strengths of both Lasso and Ridge in the context of predictions.
The Elastic Net regression results show a coefficient of around -0.0076 for the intercept and 1.0156 for the first feature. The intercept is -1.3016. The MSE is 1.8724, and the R-squared value is 0.9974. Elastic Net combines L1 and L2 regularization, providing a balance between Ridge and Lasso. The coefficients and performance metrics are similar to those of Ridge and Lasso.

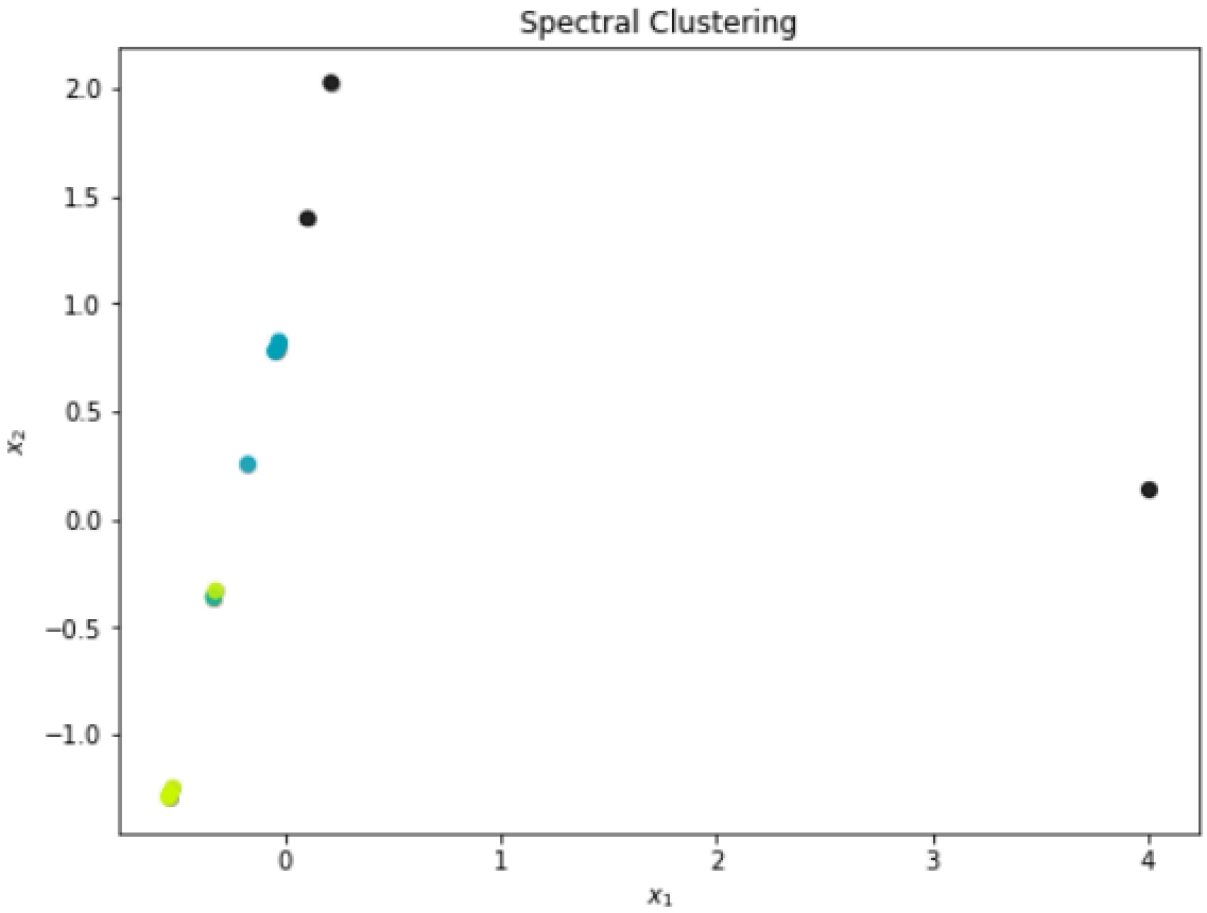
XV. Spectral Clustering Method :

1. Definition :

Spectral clustering is a powerful technique in machine learning designed to partition datasets into coherent clusters based on the underlying similarity between data points.

The process initiates with the construction of a similarity graph, where nodes represent individual data points, and edges are weighted according to pairwise similarities. Subsequently, an affinity matrix is computed to encapsulate the strength of connections between points, serving as the foundation for spectral clustering. The Laplacian matrix, derived from the affinity matrix, undergoes eigendecomposition to reveal the intrinsic structure of the data through its eigenvectors. By embedding the data into a lower-dimensional space defined by these eigenvectors, spectral clustering leverages K-Means or a similar algorithm to delineate clusters. Each data point is then assigned to a specific cluster based on its representation in the reduced-dimensional space.

2. Plots :



3. Comments :

The data is first standardized using StandardScaler to ensure consistent scaling.
Spectral Clustering is then applied with the number of clusters set to three and the affinity measure chosen as 'nearest_neighbors'.
The resulting clusters are visualized in a scatter plot, where each point is colored according to its assigned cluster.
The plot effectively illustrates the clustering outcome, showcasing how Spectral Clustering has grouped the data points in the transformed space.
The choice of the nearest_neighbors affinity indicates that data points are connected if they are nearest neighbors, influencing the cluster assignments.

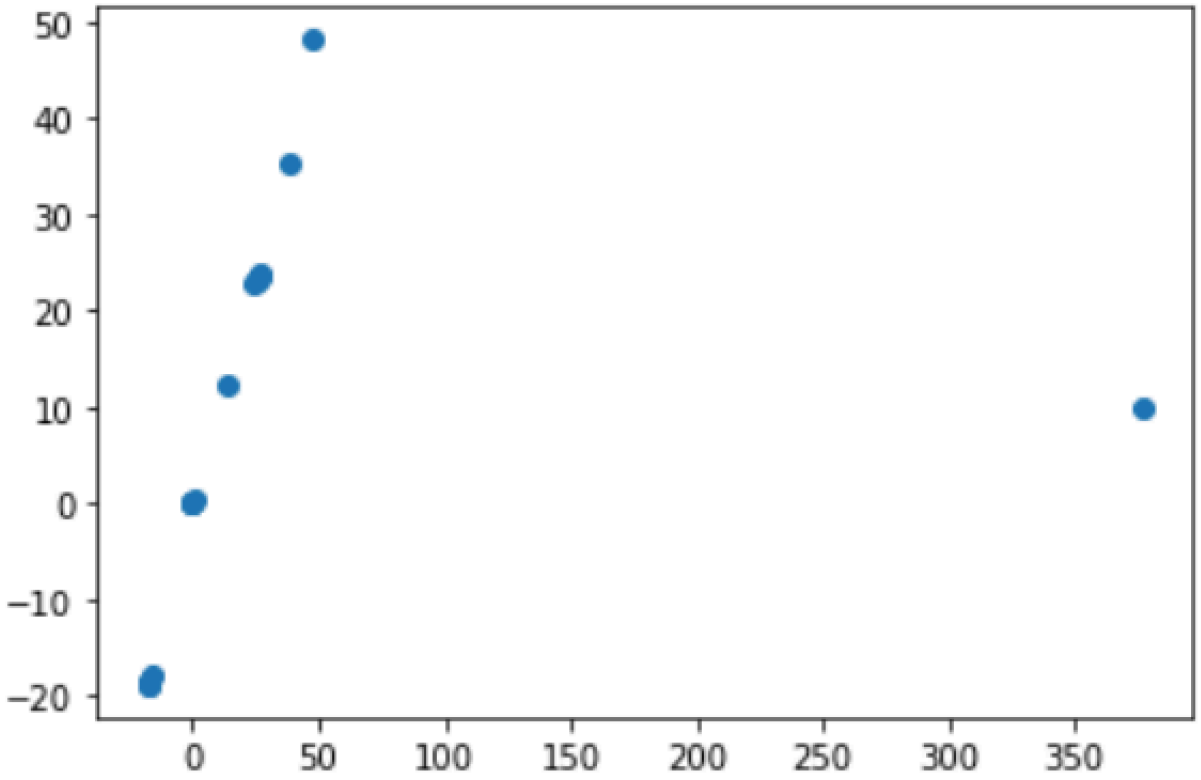
XVI. SVM Method :

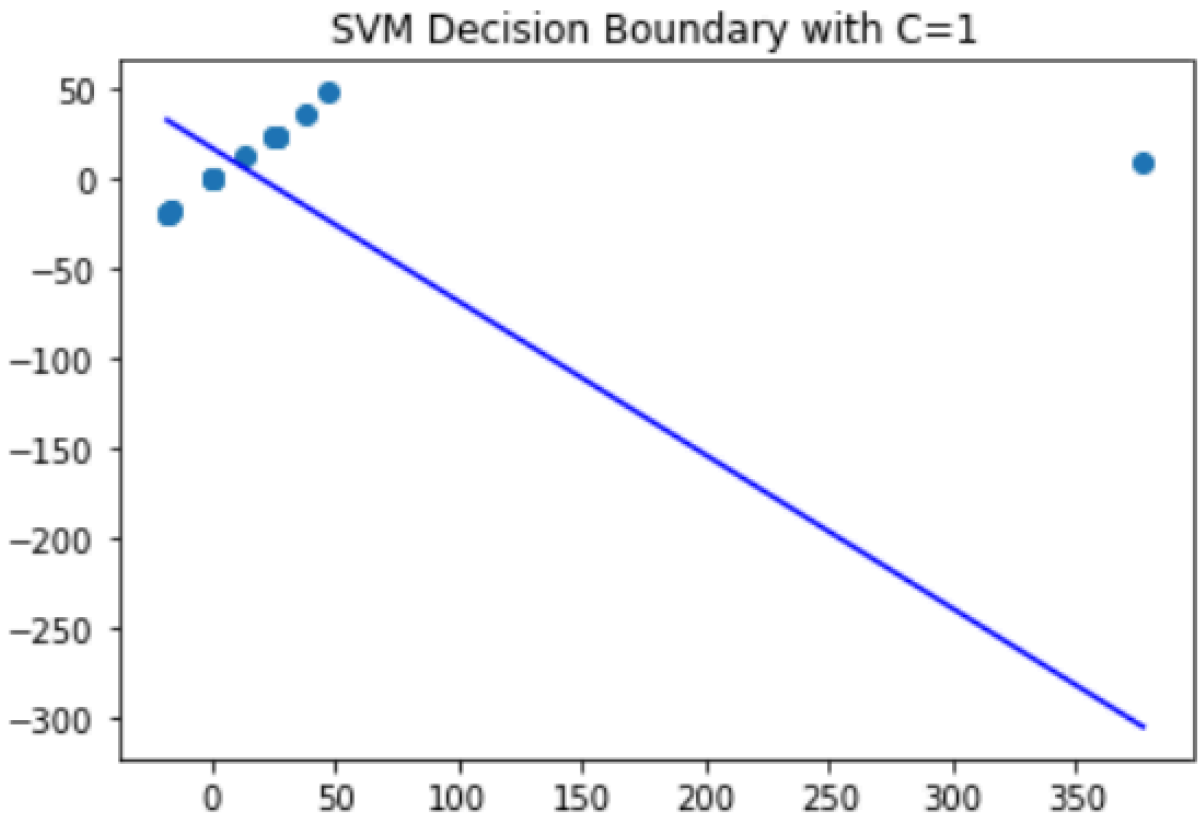
1. Definition :

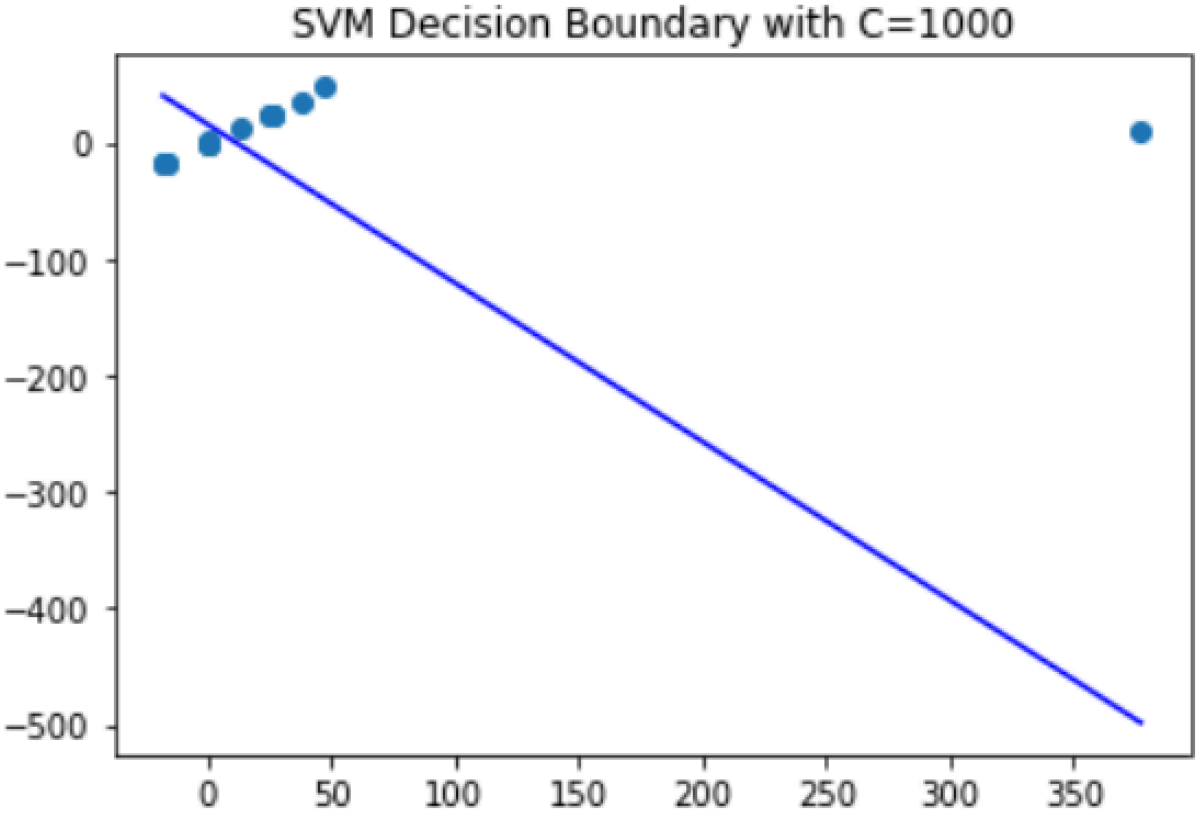
Support Vector Machines stand as a potent supervised machine learning algorithm with applications in classification and regression tasks. The core objective of SVM is to discover a hyperplane in the feature space that maximizes the separation between different classes.

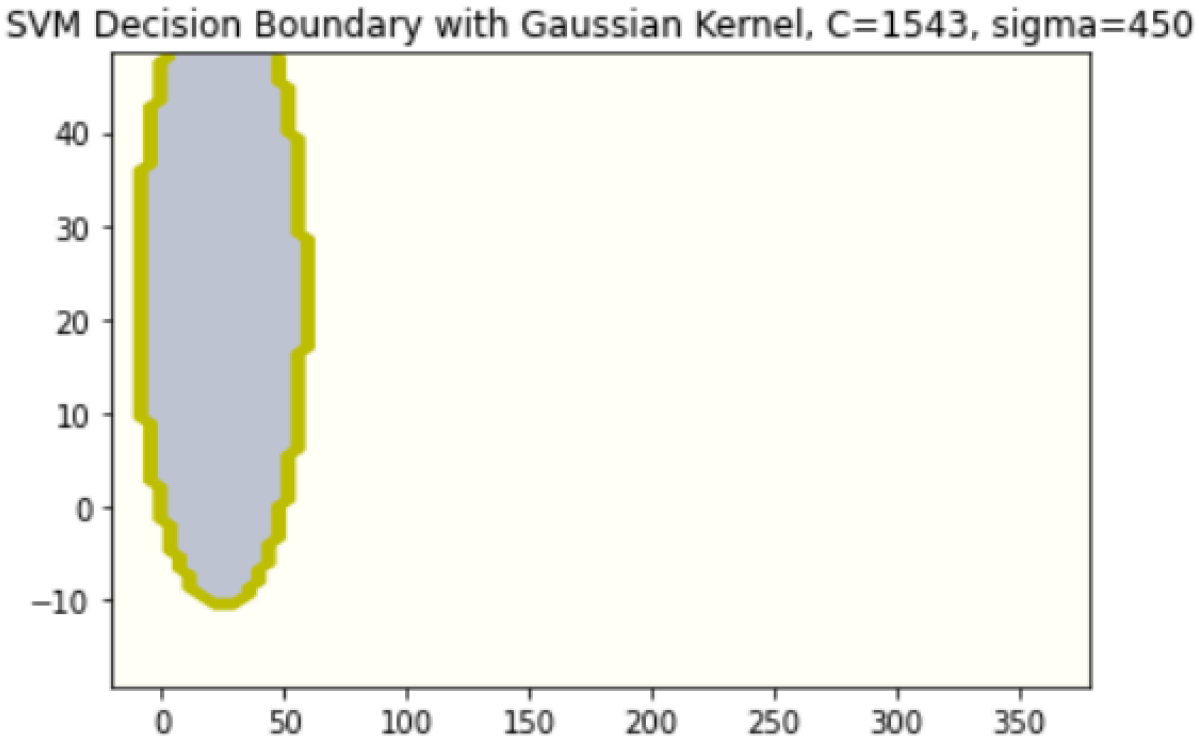
Support vectors, the data points closest to the decision boundary, play a pivotal role in defining the optimal hyperplane and the margin, representing the distance between the hyperplane and these support vectors. SVM's adaptability to non-linear relationships is facilitated by the kernel trick, allowing the algorithm to transform the feature space into a higher-dimensional one. Moreover, SVM can accommodate cases where perfect separation is not feasible by introducing a soft margin, thereby balancing the trade-off between achieving a wider margin and tolerating classification errors. The regularization parameter C governs this trade-off, influencing the smoothness of the decision boundary. SVM's predictive capabilities rely on a decision function that classifies new data points based on their positioning relative to the determined hyperplane.

2. Plots :




SVM Decision Boundary with C=1
50
0
−50
−100
−150
−200
−250
−300
0
50
100
150
200
250
300
350


SVM Decision Boundary with C=1000
0
−100
−200
−300
−400
−500
0
50
100
150
200
250
300
350




```
Gaussian Kernel similarity: 0.9999777780246896
```

3. Comments :

The first plot displays the raw data points from the dataset, where each point is represented by its first two features X[:, 0] and X[:, 1].

The second plot illustrates the decision boundary of a Support Vector Machine (SVM) trained with a regularization parameter (C) set to 1. The decision boundary effectively separates the data points into two classes.

As C increases to 1000 in the third plot, the SVM becomes more sensitive to individual data points, resulting in a decision boundary that minimizes classification errors.

The final plot introduces a Gaussian Kernel to the SVM, with parameters C = 1543 and sigma = 450.

The Gaussian Kernel similarity between two vectors, x1 and x2, with a given sigma value of 450 is remarkably high, approximately 0.99998.
This high similarity score indicates that the two vectors are very close to each other in the feature space when transformed using the Gaussian Kernel.

XVIII. Conclusion :

In this extensive exploration of clustering and dimensionality reduction techniques applied to a dataset of 3879 aircraft trajectories with 18 features each, we delved into various methods to uncover meaningful patterns within the data.

The Affinity Propagation Method emerged as an attractive choice due to its ability to automatically determine the number of clusters and adapt to complex structures. The visualization plots facilitated a clear interpretation of the discovered clusters, making it suitable for scenarios where the cluster count is unknown beforehand.

Birch Method, a hierarchical clustering technique, showcased its efficiency in handling large datasets and provided a hierarchical structure for understanding relationships within the data.

For density-based clustering, the DBSCAN Method demonstrated its prowess in identifying clusters of arbitrary shapes while robustly handling noise.

Fuzzy C-Means Clustering Method introduced the concept of assigning degrees of membership to each point, offering a nuanced perspective on clustering. This method proved valuable when points exhibited varying degrees of belonging to multiple clusters.

GMM Method and its Gaussian Mixture Model showcased flexibility in capturing complex data distributions through a mixture of Gaussian components. Contour plots effectively communicated the probabilistic nature of the clustering.

KMean Method illustrated its simplicity and effectiveness, although sensitive to initial conditions. The visualizations aided in understanding how clusters were formed based on mean values, making it suitable for well-separated clusters.

In the realm of medoid-based clustering, KMedoids Method offered enhanced robustness to outliers compared to KMeans.

MaxLikehood Method found its place in estimating distribution parameters, a crucial aspect in clustering tasks.

Mean-Shift Clustering Method demonstrated its suitability for identifying clusters with varying shapes and sizes by finding local density maxima.

For scalability in handling large datasets, the MiniBatchMeans Method offered a faster variant of KMeans, preserving its simplicity while enhancing efficiency.

The OPTICS Method, akin to DBSCAN, was effective in identifying clusters with varying densities.

In dimensionality reduction, PCA Method played a vital role in capturing the principal components, reducing dimensionality while retaining variance.

Ridge, Lasso, and ElasticNet Methods were introduced as regularization techniques for linear regression..

The Spectral Clustering Method utilized the eigenvalues of the affinity matrix for clustering, proving effective for non-convex clusters.

The inclusion of the SVM Method showcased its versatility in both classification and clustering tasks.

Finally, the conclusion addressed for the given dataset is to clean data with no preprocessing requirements. The selection of a specific method was emphasized to depend on the data's characteristics, the desired number of clusters, and the interpretability of the results.