INDEX

S.NO.	LIST OF REPORTS	PAGE NO.
1.	**REPORT - 1** QUESTION - Perform partitioning, hierarchical, and density-based clustering algorithms on a downloaded dataset and evaluate the cluster quality by changing the algorithm's parameters.	3-9
2.	**REPORT - 2** QUESTION - Perform the following text mining preprocessing steps on a text document: a. Stop Word Removal b. Stemming c. Removal of punctuation marks d. Compute the inverse document frequency of the words in the document	10-14
3.	**REPORT - 3** QUESTION - Use the Decision Tree classification algorithm to construct a classifier on two datasets. Evaluate the classifier’s performance by dividing the dataset into a training set (75%) and a test set (25%). Compare the performance with that of: a. Bagging ensemble consisting of 3,5,7,9 Decision tree classifiers b. Adaboost ensemble consisting of 3,5,7,9 Decision tree classifiers	15-23
4.	**REPORT - 4** QUESTION - Download a dataset and check whether outliers are present in the dataset. Use different methods of outlier detection and compare their performance.	24-29
5.	**REPORT - 5** QUESTION - Perform CluStream algorithm on any time series data from Kaggle and compare its output with that of K-means clustering. Evaluate the cluster quality by changing the algorithm’s parameters.	30-35

REPORT – 1

QUESTION - Perform partitioning, hierarchical, and density-based clustering algorithms on a downloaded dataset and evaluate the cluster quality by changing the algorithm's parameters.

➢ ABOUT DATASETS –

- **Name: Clustering Penguins Species**
- **Source:** https://www.kaggle.com/datasets/youssefaboelwafa/clustering-penguins-species/data
- **Description:** The dataset is designed for clustering and other machine learning tasks related to penguins. It provides information about various physical characteristics of penguins, along with their sex, which can help identify species groupings. The dataset is suitable for exploratory data analysis, clustering algorithms, and species classification studies.
- **Columns:**
    - **culmen_length_mm** – float64
    - **culmen_depth_mm** – float64
    - **flipper_length_mm** – float64
    - **body_mass_g** – float64
    - **sex** – object

- **Name: Weather Type Classification**
- **Source:** https://www.kaggle.com/datasets/nikhil7280/weather-type-classification/data
- **Description:** The dataset, though primarily designed for classification tasks, has been utilized for clustering in this context. It provides a comprehensive set of meteorological and environmental features that can be analyzed to uncover patterns and group similar weather conditions without predefined labels.
- **Columns:**
    - **Temperature** – float64
    - **Humidity** – int64
    - **Wind Speed** – float64
    - **Precipitation (%)** – float64

- **Cloud Cover** – object
- **Atmospheric Pressure** – float64
- **UV Index** – int64
- **Season** – object
- **Visibility (km)** – float64
- **Location** – object
- **Weather Type** – object

➢ ABOUT ALGORITHMS –

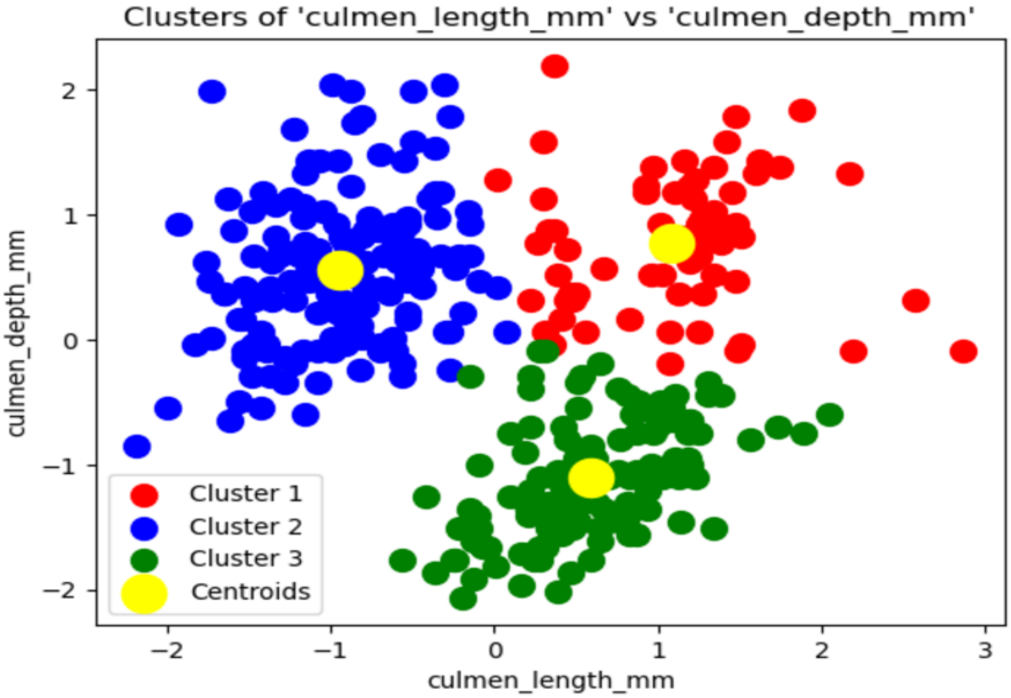
- **K-Means Clustering Algorithm:** K-means clustering is an unsupervised machine learning algorithm used to group data into k clusters based on feature similarity. It iteratively assigns data points to the nearest cluster centroid and updates centroids until convergence, minimizing intra-cluster variance.
- **Hierarchical Clustering Algorithm (Agglomerative):** Hierarchical clustering is an unsupervised algorithm that builds a hierarchy of clusters by either merging smaller clusters (agglomerative) or splitting larger ones (divisive). It produces a dendrogram, allowing selection of the optimal number of clusters.
- **Density-Based Clustering Algorithm (DBSCAN):** DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups data points based on density, identifying regions of high density as clusters and treating sparse regions as noise. It is effective for detecting clusters of arbitrary shapes and handling outliers.

➢ **PROBLEM STATEMENT –** The objective is to analyze the given datasets by applying three clustering techniques: partitioning-based (e.g., k-means), hierarchical, and density-based (e.g., DBSCAN).

➢ ANALYSIS –

- **K-Means Clustering Algorithm:**

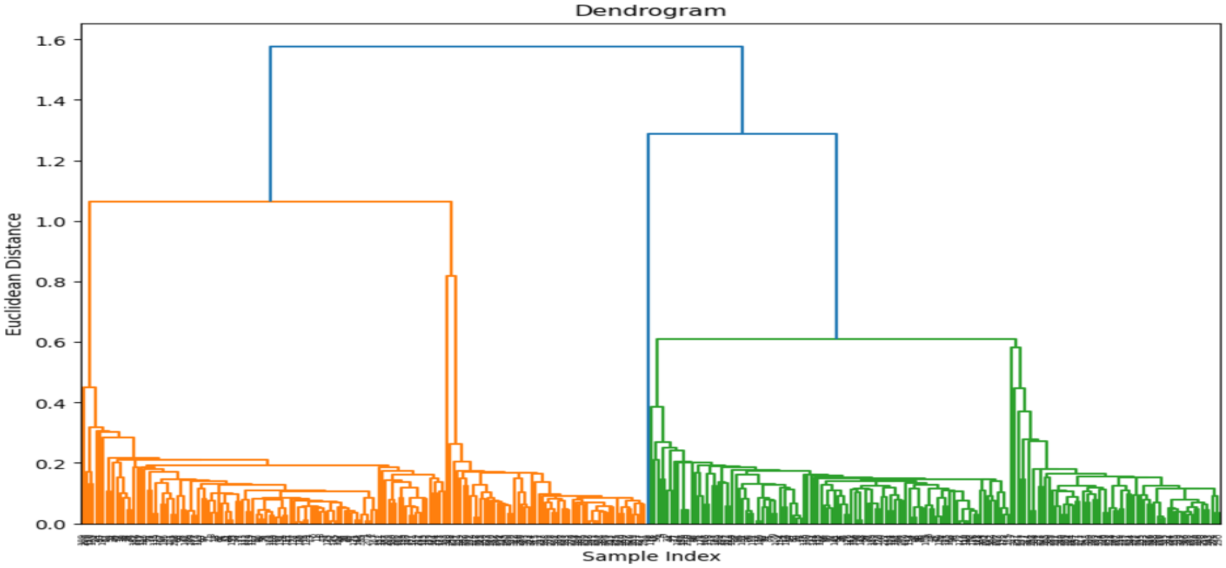
DATASET – 1



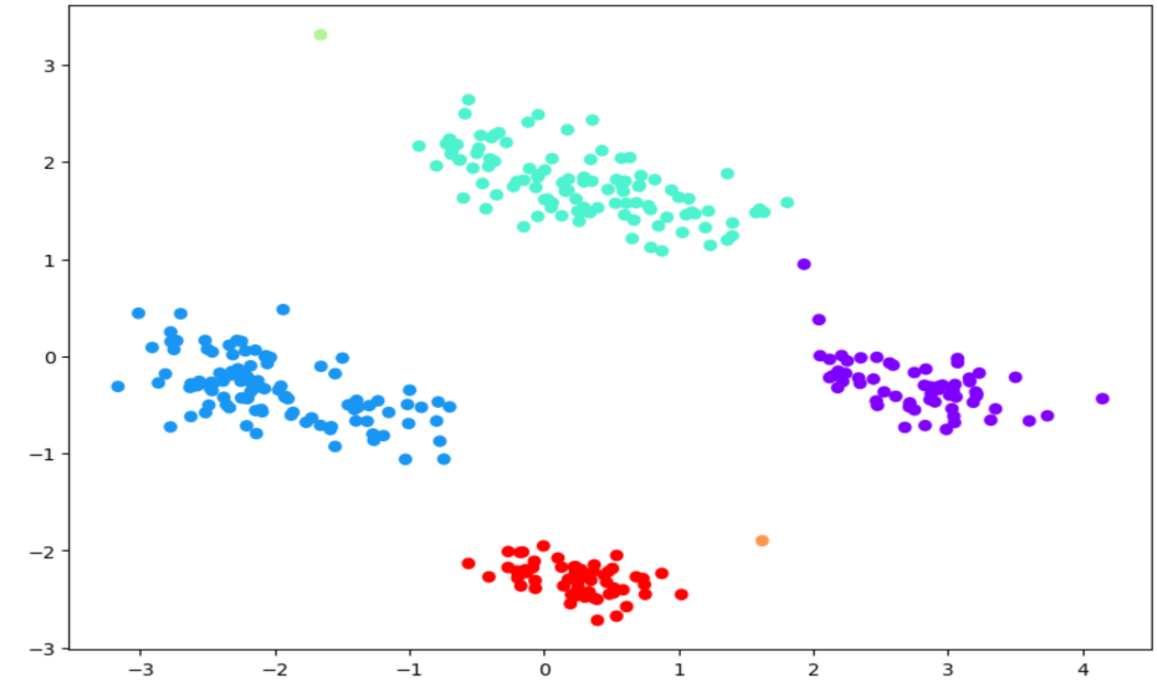
Scatter Plot of Clusters of ‘culmen_length_mm’ vs ‘culmen_depth_mm’

- **Hierarchical Clustering Algorithm (Agglomerative):**

DATASET – 1

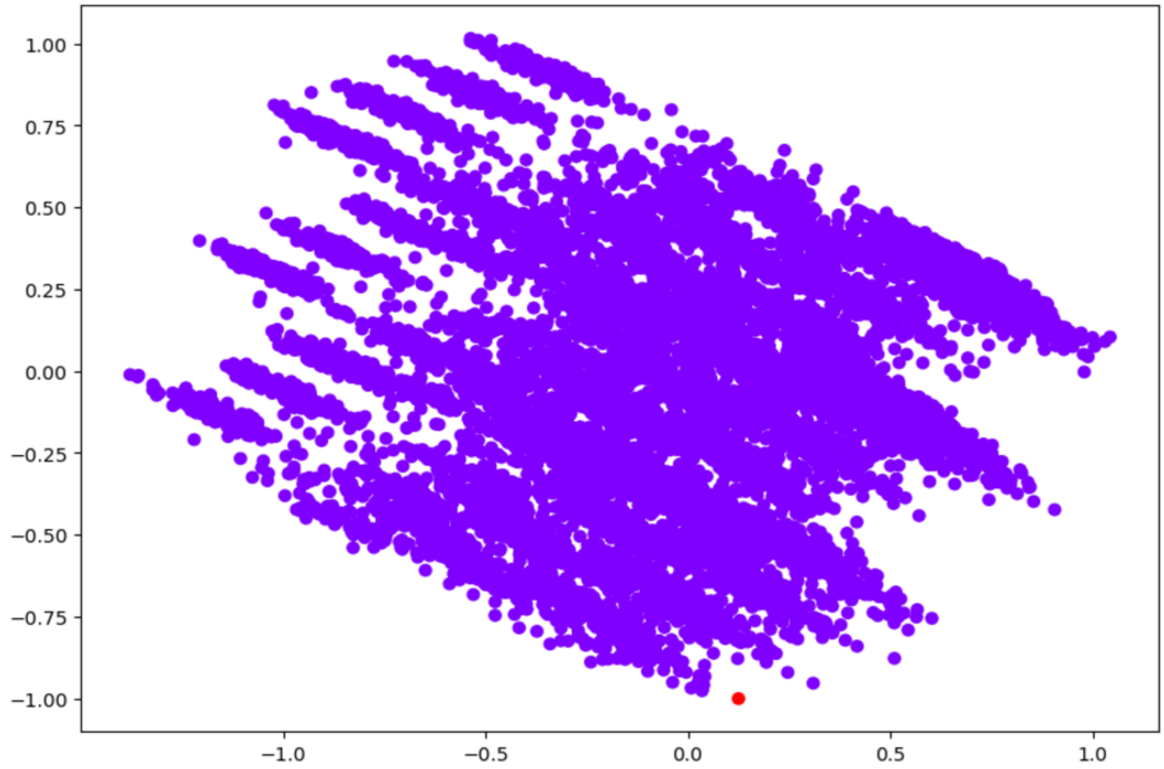


Dendrogram of Scaled Principal Component Analysis (PCA) Data (linkage=’single’)

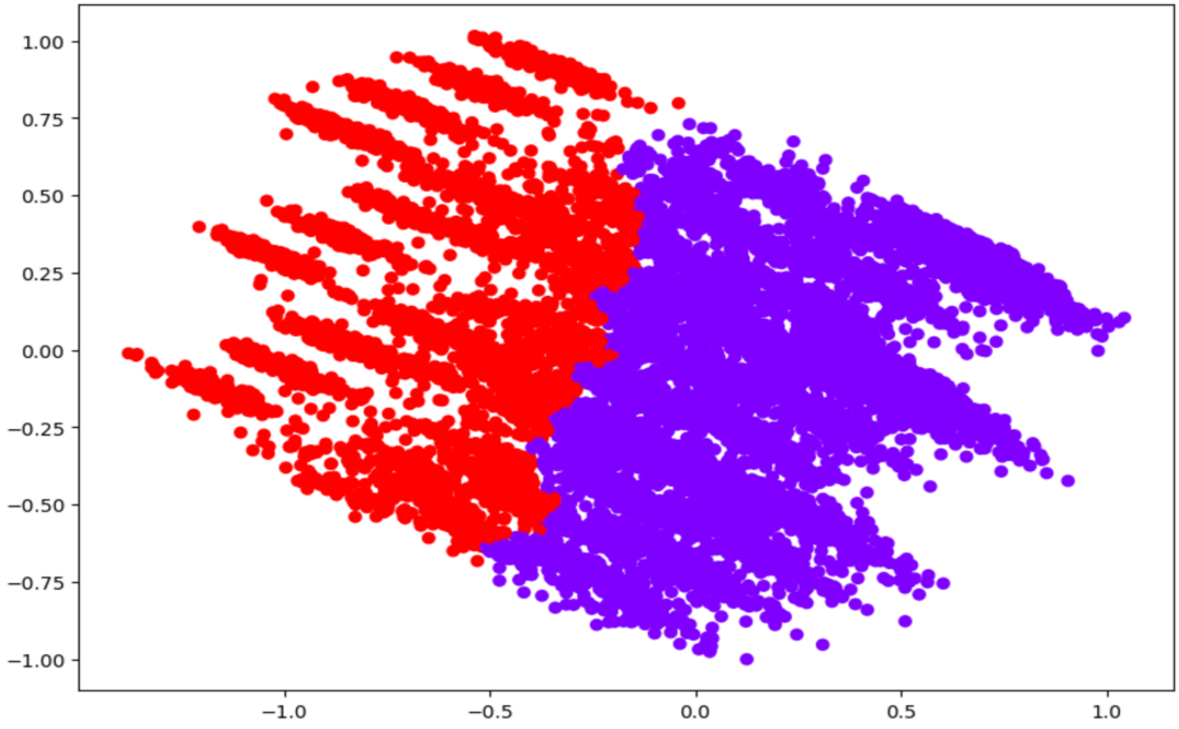


Scatter Plot of PCA Data (n_clusters=6, linkage='single')

DATASET – 2



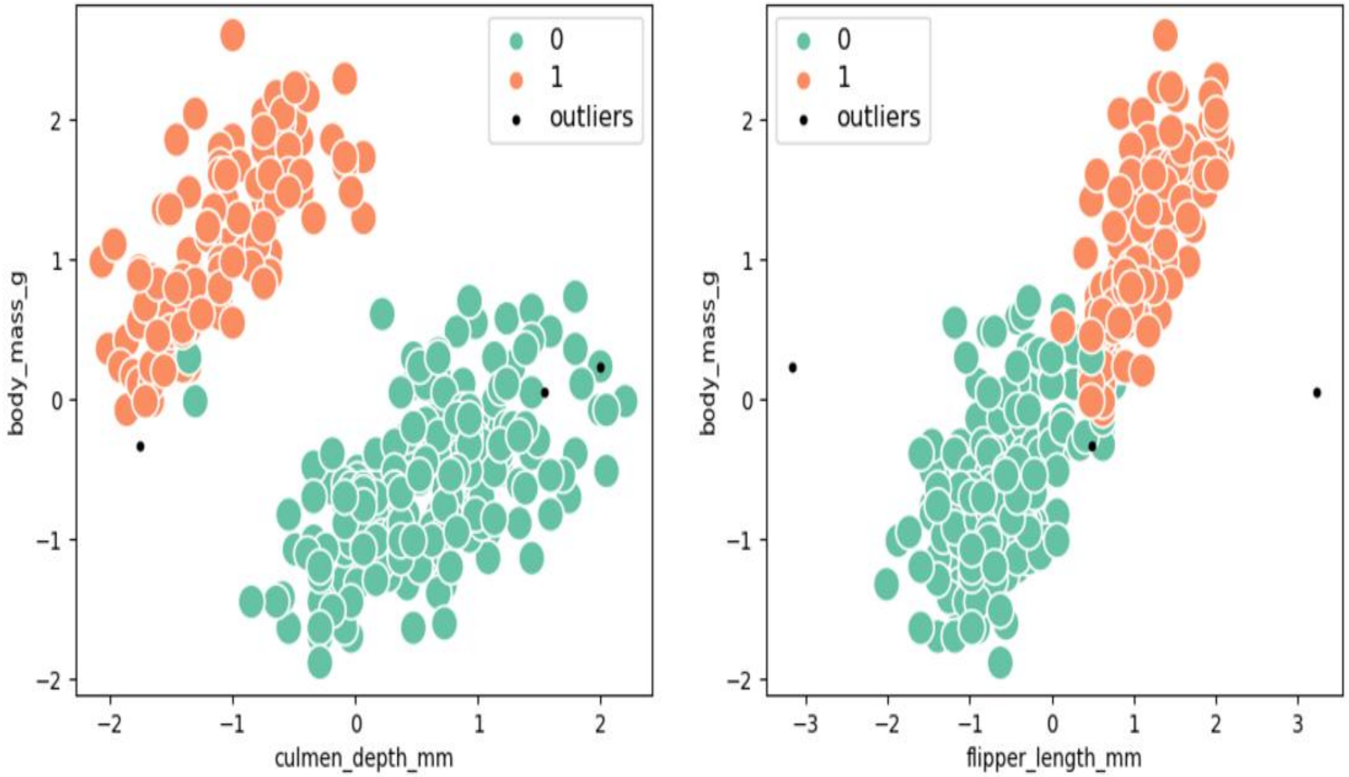
Scatter Plot of PCA Data (n_clusters=2, linkage='single')



Scatter Plot of PCA Data (n_clusters=2, linkage='complete')

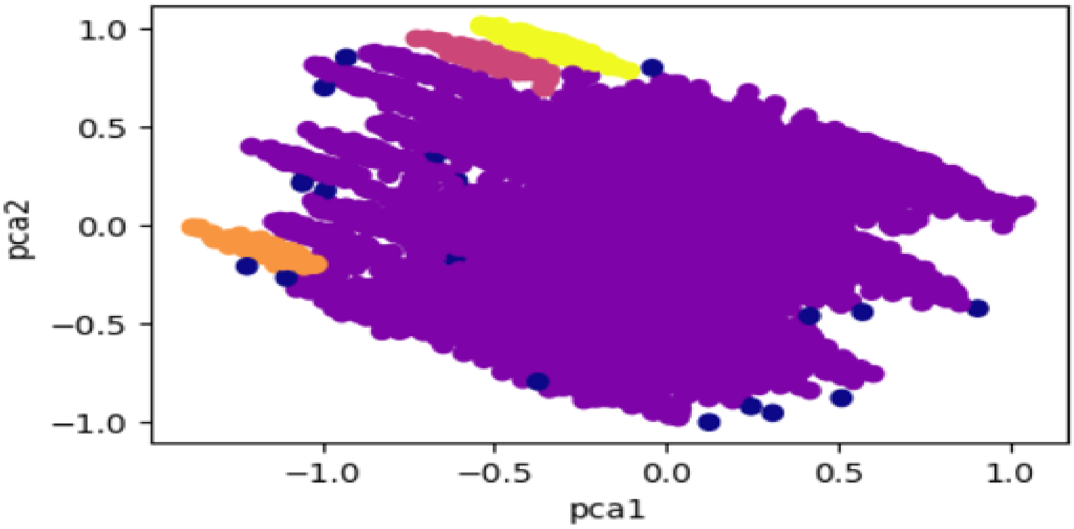
- **<u>Density-Based Clustering Algorithm (DBSCAN)</u>:**

<u>DATASET – 1</u>



Scatter Plots for 'culmen_depth_mm' vs 'body_mass_g' and 'flipper_length_mm' vs 'body_mass_g'

DATASET – 2



Scatter Plot of PCA Data (eps=0.05, min_samples=4)

- **Evaluating Performance (Purity Scores of All Clustering Algorithms on DATASET – 2):**

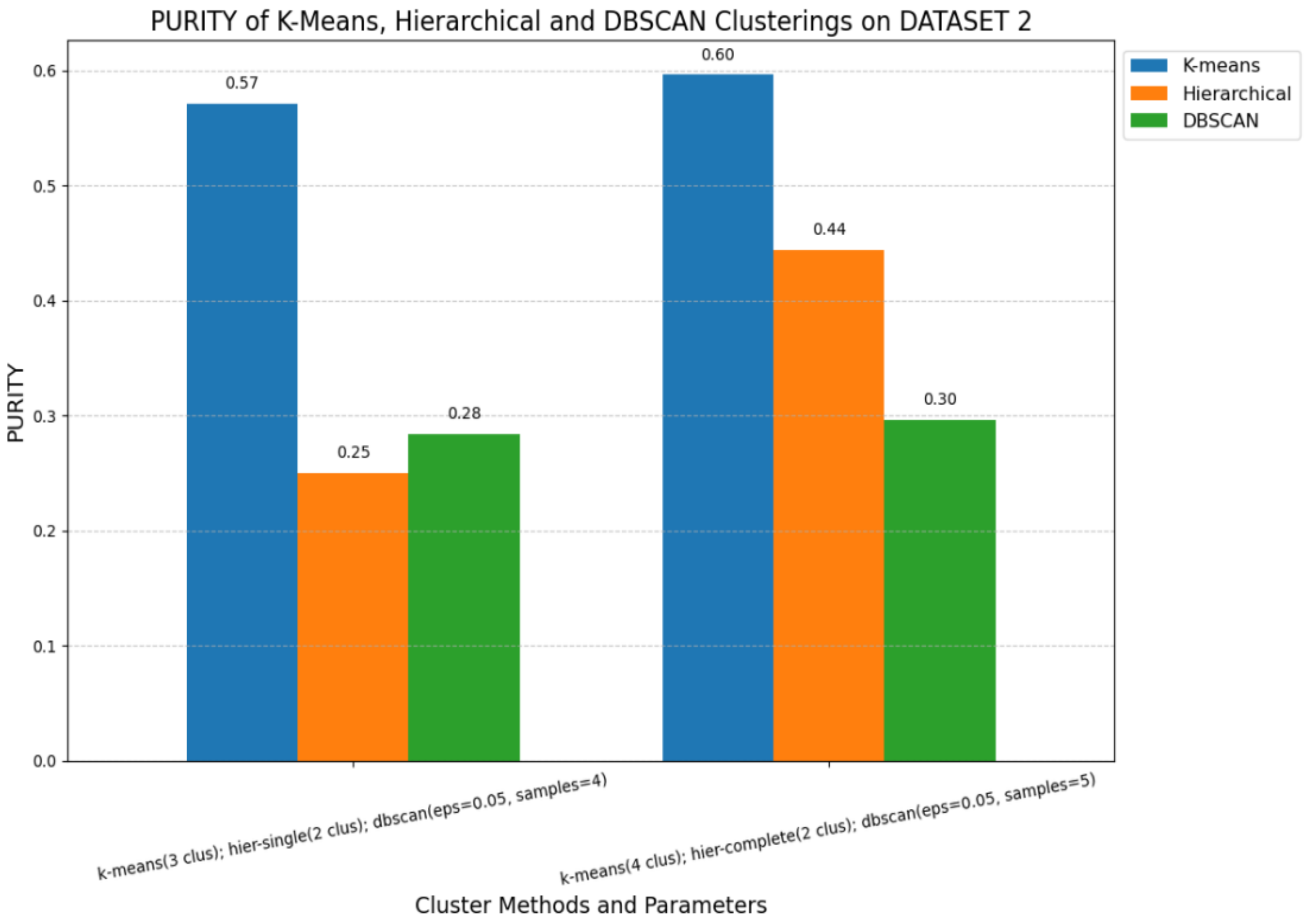
Purity is the extent to which a cluster contains objects of a single class.

Purity of cluster i ->

purity(i)= $\max_j P_{ij}$ **where** $P_{ij} = \frac{m_{ij}}{m_i}$ **(**m_i **is the no. of objects in cluster i and** m_{ij} **is the no. of objects of class j in cluster i)**

ALGORITHM	PARAMETERS	SCORE
K-Means	3 clusters	57.11%
	4 clusters	59.62%
Hierarchical	n_clusters=2, linkage='single'	25.01%
	n_clusters=2, linkage='complete'	44.37%
DBSCAN	eps=0.05, min_samples=4	28.42%
	eps=0.05, min_samples=5	29.58%

BAR PLOT OF PURITY SCORES FOR ALL CLUSTERING ALGORITHMS ON DATASET – 2



➢ CONCLUSION –

From the purity scores obtained for **DATASET - 2**, the **K-Means algorithm** demonstrates the **highest clustering performance**, achieving a **purity score of 59.62% with 4 clusters**, making it the most effective approach for this dataset. Hierarchical clustering with complete linkage shows moderate performance (44.37%) but is less effective compared to K-Means. DBSCAN performs poorly, with a maximum purity score of only (29.58%) using eps=0.05 and min_samples=5, indicating it may not be suitable for this dataset's characteristics. Adjusting parameters in all methods shows that K-Means benefits most from parameter tuning.

REPORT – 2

QUESTION - Perform the following text mining preprocessing steps on a text document:

a. Stop Word Removal

b. Stemming

c. Removal of punctuation marks

d. Compute the inverse document frequency of the words in the document

➢ ABOUT DATASET –

- **Name: India Spam SMS Classification**
- **Source:** https://www.kaggle.com/datasets/junioralive/india-spam-sms-classification
- **Description:** This dataset is a collection of real-world SMS messages labeled as spam or ham, specifically curated to represent communication patterns in the Indian telecom sector. It provides a snapshot of typical promotional and personal communications, reflecting the everyday challenges faced by spam filters.
- **Columns**:
    - **Msg** – object
    - **Label** – object

➢ ABOUT TEXT MINING PREPROCESSING STEPS –

- **Stop Word Removal:** Eliminates common words (e.g., is, the, and) that do not contribute to the text's semantic meaning to reduce noise and focus on important terms.
- **Stemming:** Reduces words to their root or base form (e.g., running → run), helping to standardize and group similar terms.
- **Removal of Punctuation Marks:** Strips punctuation (e.g., commas, periods) to clean the text and ensure focus remains on meaningful words.

- **Inverse Document Frequency:** Calculates how unique a word is across documents, helping to down weight commonly occurring terms while highlighting distinctive ones.

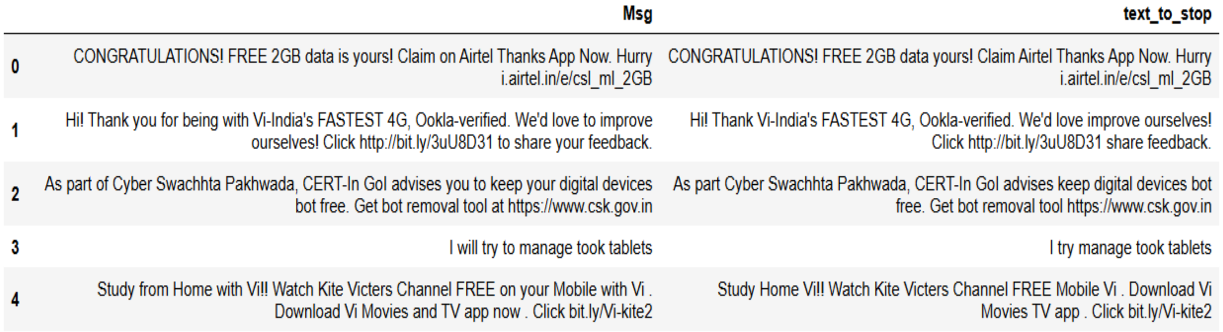
➢ PROBLEM STATEMENT – Develop a text-mining preprocessing pipeline to clean and analyze a given text document by performing the following steps:

- Remove common stop words to focus on meaningful terms.
- Apply stemming to reduce words to their base forms.
- Remove punctuation marks to simplify the text structure.
- Calculate the inverse document frequency (IDF) for each word to identify unique and significant terms within the document.

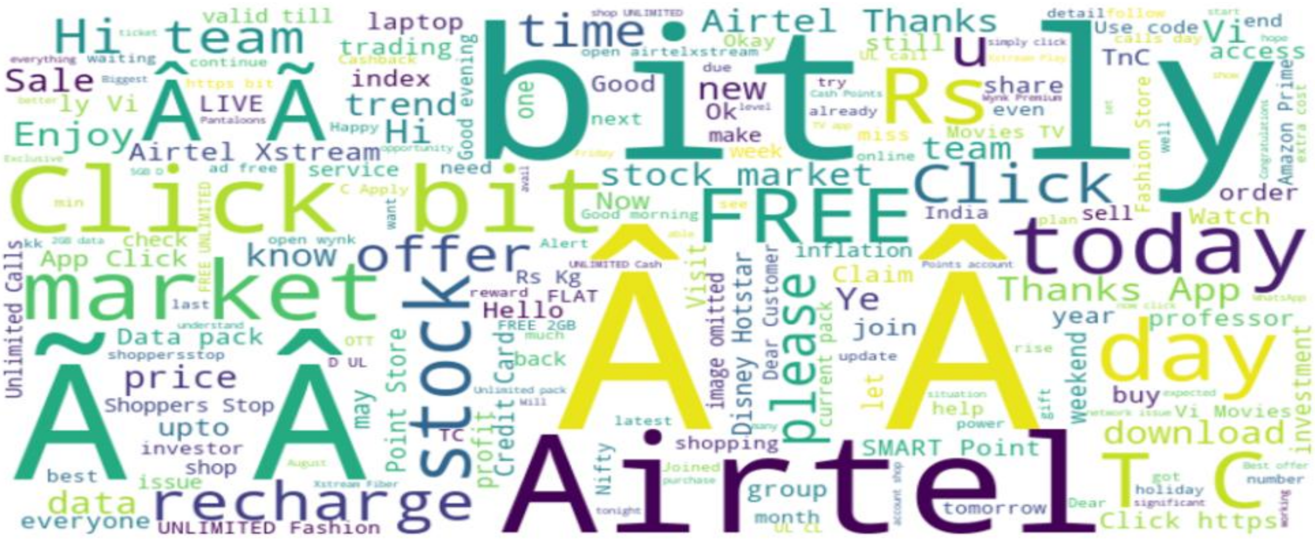
➢ ANALYSIS –

- **Stop Word Removal:**

OUTPUT



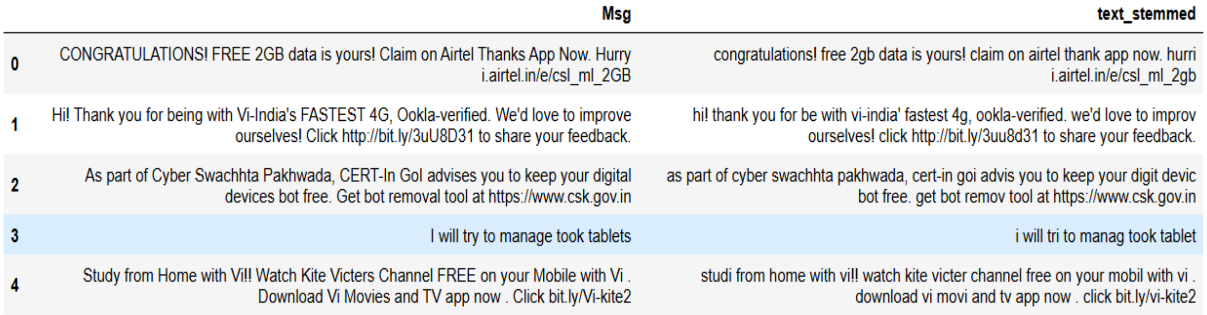
	Msg	text_to_stop
0	CONGRATULATIONS! FREE 2GB data is yours! Claim on Airtel Thanks App Now. Hurry i.airtel.in/e/csl_ml_2GB	CONGRATULATIONS! FREE 2GB data yours! Claim Airtel Thanks App Now. Hurry i.airtel.in/e/csl_ml_2GB
1	Hi! Thank you for being with Vi-India's FASTEST 4G, Ookla-verified. We'd love to improve ourselves! Click http://bit.ly/3uU8D31 to share your feedback.	Hi! Thank Vi-India's FASTEST 4G, Ookla-verified. We'd love improve ourselves! Click http://bit.ly/3uU8D31 share feedback.
2	As part of Cyber Swachhta Pakhwada, CERT-In GoI advises you to keep your digital devices bot free. Get bot removal tool at https://www.csk.gov.in	As part Cyber Swachhta Pakhwada, CERT-In GoI advises keep digital devices bot free. Get bot removal tool https://www.csk.gov.in
3	I will try to manage took tablets	I try manage took tablets
4	Study from Home with Vi!! Watch Kite Victers Channel FREE on your Mobile with Vi . Download Vi Movies and TV app now . Click bit.ly/Vi-kite2	Study Home Vi!! Watch Kite Victers Channel FREE Mobile Vi . Download Vi Movies TV app . Click bit.ly/Vi-kite2



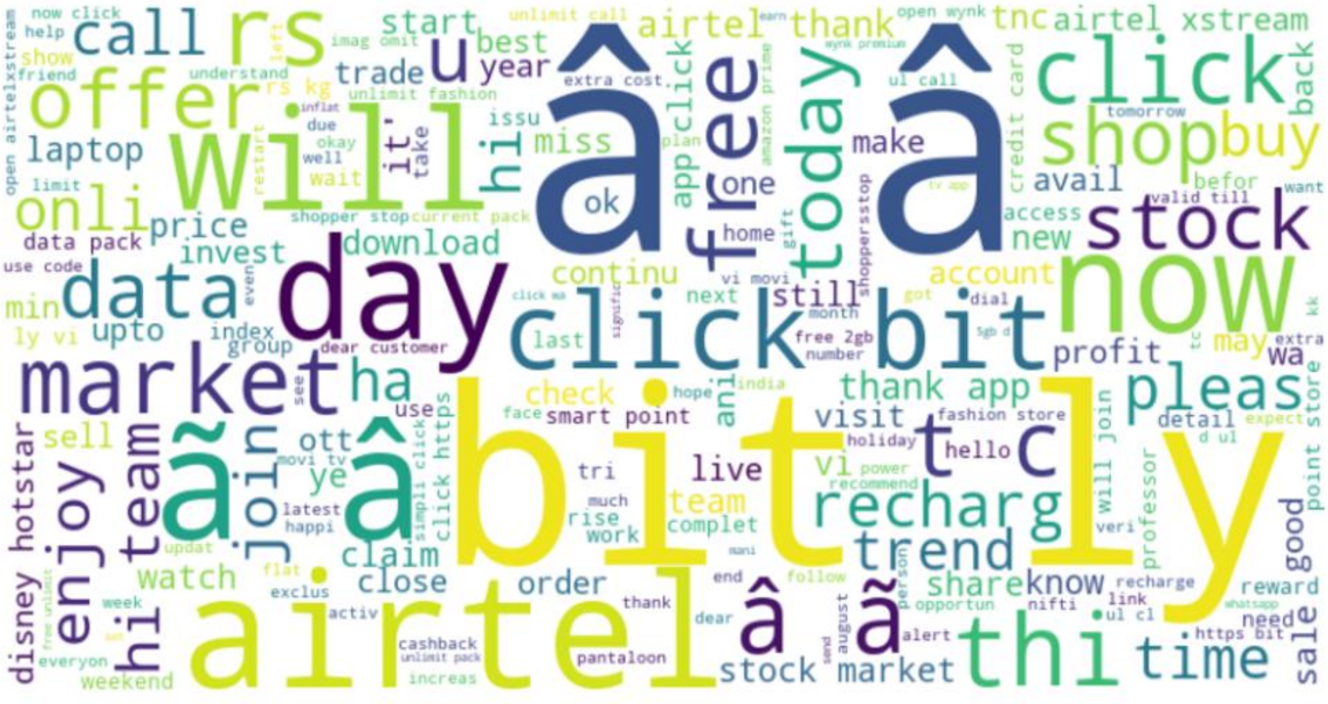
Word Cloud After Stop Word Removal

- **Stemming**:

OUTPUT



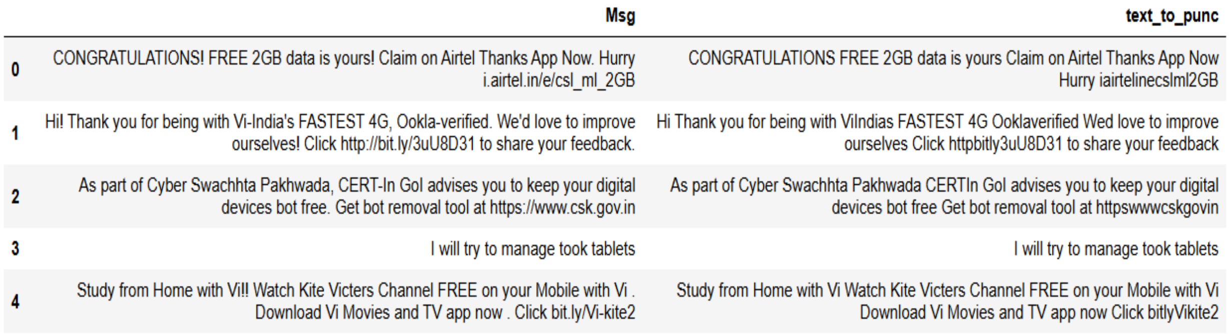
	Msg	text_stemmed
0	CONGRATULATIONS! FREE 2GB data is yours! Claim on Airtel Thanks App Now. Hurry i.airtel.in/e/csl_ml_2GB	congratulations! free 2gb data is yours! claim on airtel thank app now. hurri i.airtel.in/e/csl_ml_2gb
1	Hi! Thank you for being with Vi-India's FASTEST 4G, Ookla-verified. We'd love to improve ourselves! Click http://bit.ly/3uU8D31 to share your feedback.	hi! thank you for be with vi-india' fastest 4g, ookla-verified. we'd love to improv ourselves! click http://bit.ly/3uu8d31 to share your feedback.
2	As part of Cyber Swachhta Pakhwada, CERT-In GoI advises you to keep your digital devices bot free. Get bot removal tool at https://www.csk.gov.in	as part of cyber swachhta pakhwada, cert-in goi advis you to keep your digit devic bot free. get bot remov tool at https://www.csk.gov.in
3	I will try to manage took tablets	i will tri to manag took tablet
4	Study from Home with Vi!! Watch Kite Victers Channel FREE on your Mobile with Vi . Download Vi Movies and TV app now . Click bit.ly/Vi-kite2	studi from home with vi!! watch kite victer channel free on your mobil with vi . download vi movi and tv app now . click bit.ly/vi-kite2



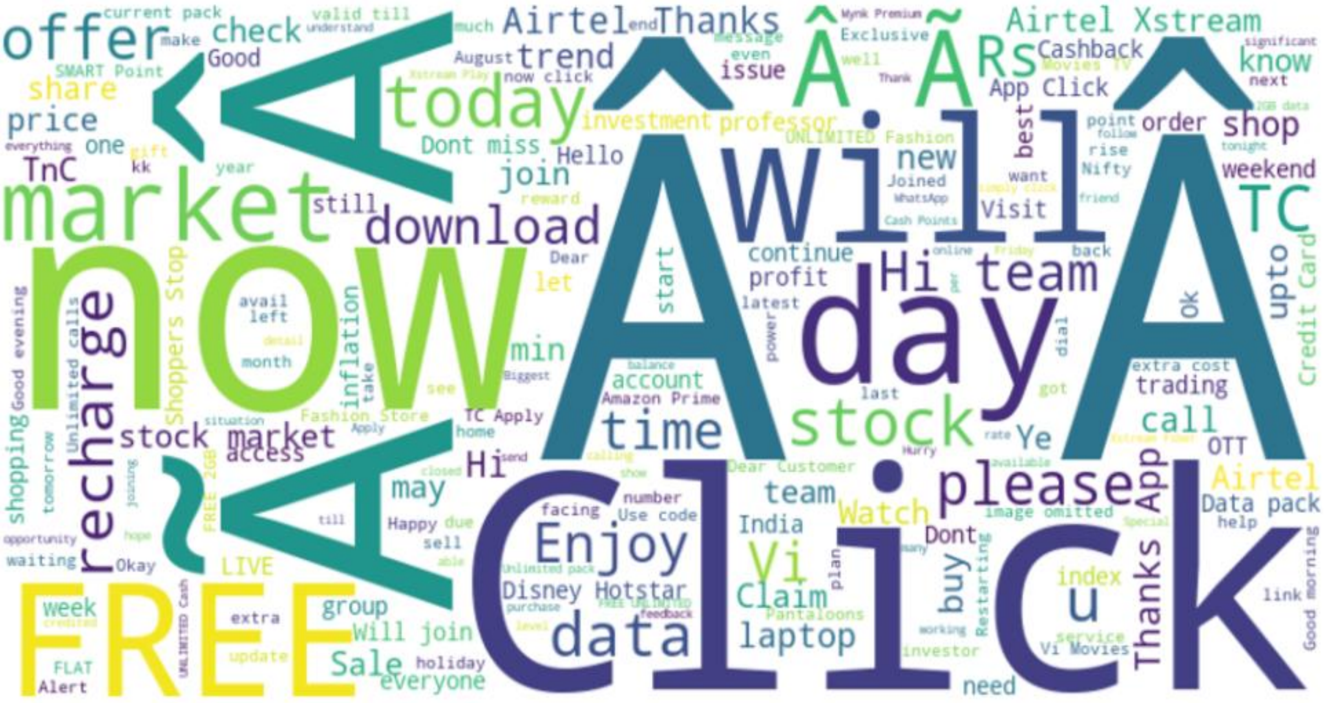
Word Cloud After Stemming

- **Removal of Punctuation Marks**:

OUTPUT



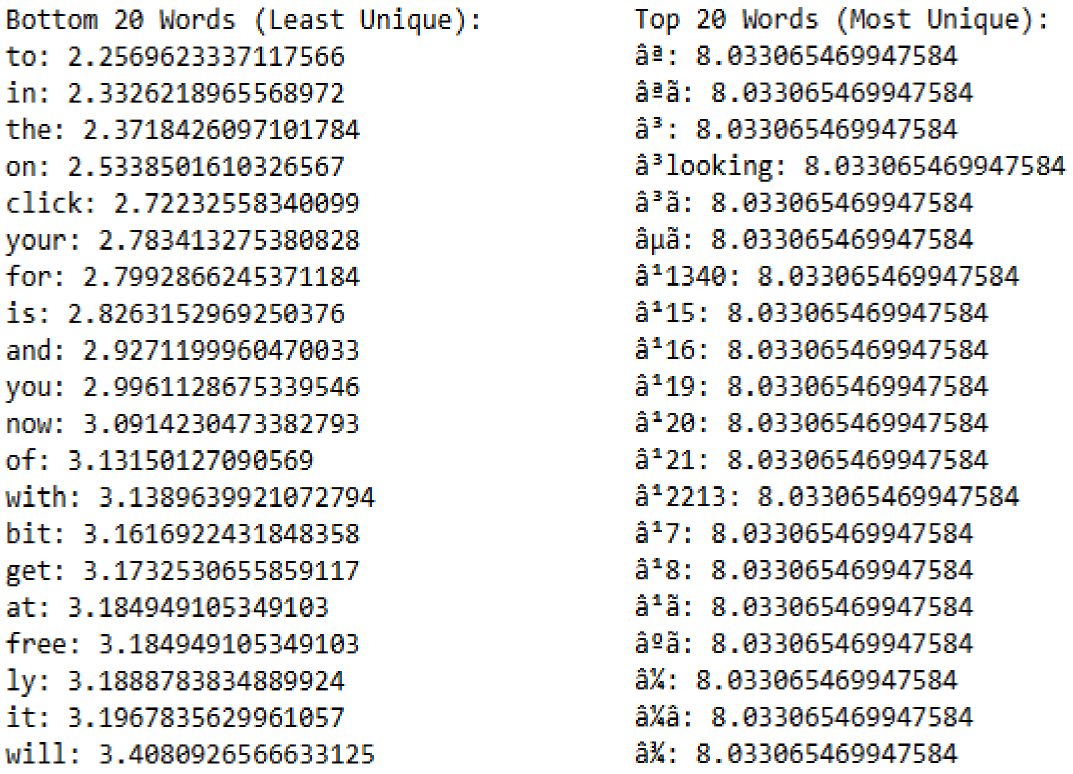
	Msg	text_to_punc
0	CONGRATULATIONS! FREE 2GB data is yours! Claim on Airtel Thanks App Now. Hurry i.airtel.in/e/csl_ml_2GB	CONGRATULATIONS FREE 2GB data is yours Claim on Airtel Thanks App Now Hurry iairtelinecslml2GB
1	Hi! Thank you for being with Vi-India's FASTEST 4G, Ookla-verified. We'd love to improve ourselves! Click http://bit.ly/3uU8D31 to share your feedback.	Hi Thank you for being with ViIndias FASTEST 4G Ooklaverified Wed love to improve ourselves Click httpbitly3uU8D31 to share your feedback
2	As part of Cyber Swachhta Pakhwada, CERT-In GoI advises you to keep your digital devices bot free. Get bot removal tool at https://www.csk.gov.in	As part of Cyber Swachhta Pakhwada CERTIn GoI advises you to keep your digital devices bot free Get bot removal tool at httpswwwcskgovin
3	I will try to manage took tablets	I will try to manage took tablets
4	Study from Home with Vi!! Watch Kite Victers Channel FREE on your Mobile with Vi . Download Vi Movies and TV app now . Click bit.ly/Vi-kite2	Study from Home with Vi Watch Kite Victers Channel FREE on your Mobile with Vi Download Vi Movies and TV app now Click bitlyVikite2



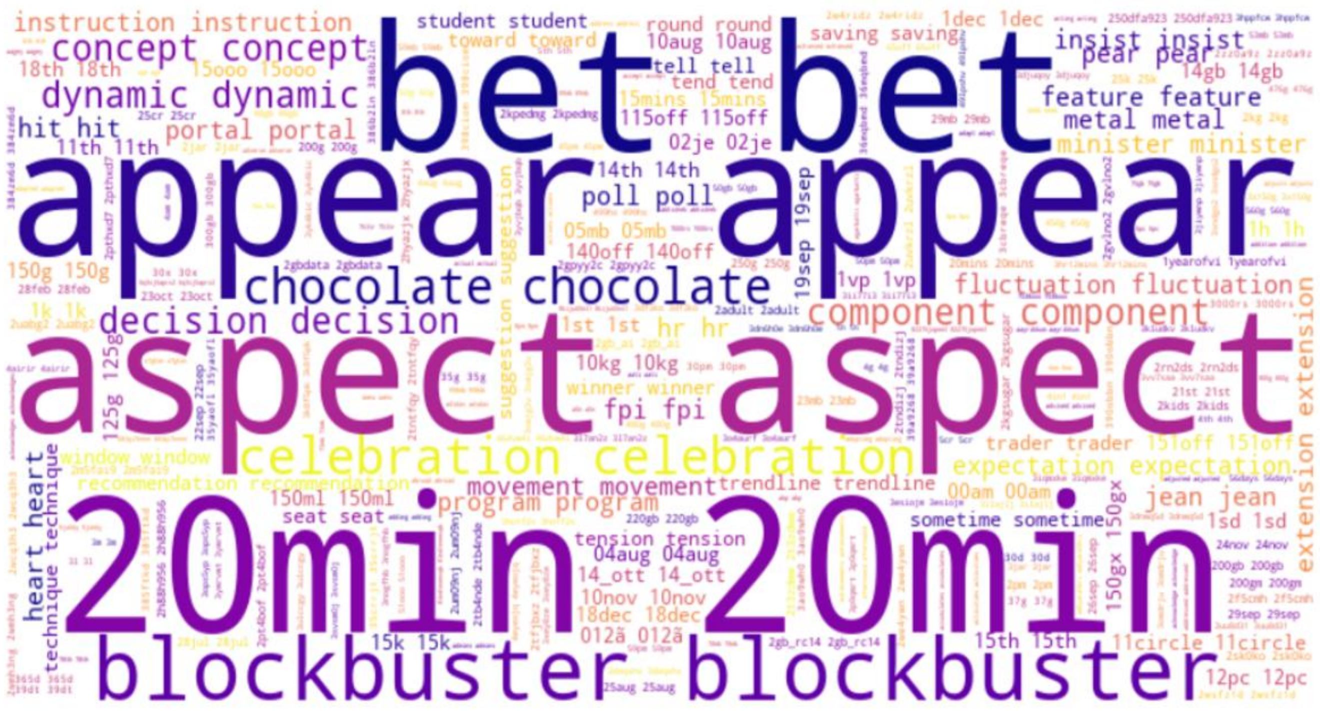
Word Cloud After Removal of Punctuation Marks

- **Inverse Document Frequency**:

OUTPUT



```
Bottom 20 Words (Least Unique):         Top 20 Words (Most Unique):
to: 2.2569623337117566                  âª: 8.033065469947584
in: 2.3326218965568972                  âªã: 8.033065469947584
the: 2.3718426097101784                 â³: 8.033065469947584
on: 2.5338501610326567                  â³looking: 8.033065469947584
click: 2.72232558340099                 â³ã: 8.033065469947584
your: 2.783413275380828                 âµã: 8.033065469947584
for: 2.7992866245371184                 â¹1340: 8.033065469947584
is: 2.826315296925O376                  â¹15: 8.033065469947584
and: 2.9271199960470033                 â¹16: 8.033065469947584
you: 2.9961128675339546                 â¹19: 8.033065469947584
now: 3.0914230473382793                 â¹20: 8.033065469947584
of: 3.13150127090569                    â¹21: 8.033065469947584
with: 3.1389639921072794                â¹2213: 8.033065469947584
bit: 3.1616922431848358                 â¹7: 8.033065469947584
get: 3.1732530655859117                 â¹8: 8.033065469947584
at: 3.184949105349103                   â¹ã: 8.033065469947584
free: 3.184949105349103                 âºã: 8.033065469947584
ly: 3.1888783834889924                  â¼: 8.033065469947584
it: 3.196783562996105 7                 â¼â: 8.033065469947584
will: 3.4080926566633125                â¾: 8.033065469947584
```



Word Cloud After Computing IDF Values of All the Words

➢ CONCLUSION –

The text document was successfully preprocessed using various steps. Stop word removal reduced noise by eliminating common words, stemming standardized terms to their root forms, and punctuation removal simplified the text structure. Finally, calculating the inverse document frequency (IDF) highlighted unique and significant words, making the document well-prepared for further text-mining analysis.

REPORT – 3

QUESTION - Use the Decision Tree classification algorithm to construct a classifier on two datasets. Evaluate the classifier's performance by dividing the dataset into a training set (75%) and a test set (25%). Compare the performance with that of:

a. Bagging ensemble consisting of 3,5,7,9 Decision tree classifiers

b. AdaBoost ensemble consisting of 3,5,7,9 Decision tree classifiers

➢ ABOUT DATASETS –

- **Name: Rice (Cammeo and Osmancik)**
- **Source:** https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik
- **Description:** The dataset is designed for analyzing and classifying two varieties of rice grains: Cammeo and Osmancik. It includes shape-based features extracted from rice grain images, making it suitable for classification and clustering tasks.
- **Columns:**
    - **Area** – int64
    - **Perimeter** – float64
    - **Major_Axis_Length** – float64
    - **Minor_Axis_Length** – float64
    - **Eccentricity** – float64
    - **Convex_Area** – int64
    - **Extent** – float64
    - **Class** – object

- **Name: Dry Bean**
- **Source:** https://archive.ics.uci.edu/dataset/602/dry+bean+dataset
- **Description:** The dataset is designed for analyzing and classifying different varieties of dry beans based on shape and morphological features extracted from bean images. It is suitable for machine learning tasks such as classification and clustering.

- **Columns:**

  - **Area** – int64
  - **Perimeter** – float64
  - **MajorAxisLength** – float64
  - **MinorAxisLength** – float64
  - **AspectRatio** – float64
  - **Eccentricity** – float64
  - **ConvexArea** – int64
  - **EquivDiameter** – float64
  - **Extent** – float64
  - **Solidity** – float64
  - **Roundness** – float64
  - **Compactness** – float64
  - **ShapeFactor1** – float64
  - **ShapeFactor2** – float64
  - **ShapeFactor3** – float64
  - **ShapeFactor4** – float64
  - **Class** – object

➢ ABOUT ALGORITHM / ENSEMBLE METHODS –

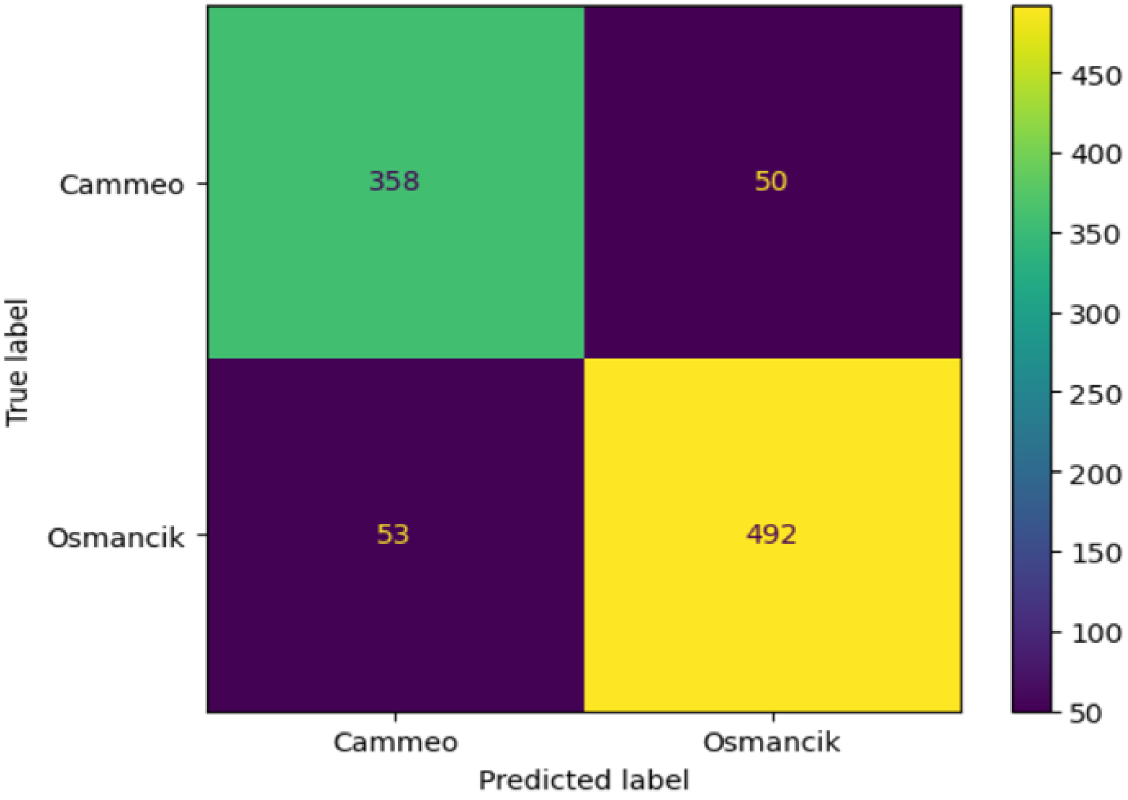
- **Decision Tree Classification Algorithm:** A decision tree is a supervised machine learning algorithm that splits data into subsets based on feature values, creating a tree-like structure. It makes predictions by following the branches from the root to a leaf node, where each leaf represents a class label.
- **Bagging (Bootstrap Aggregating):** Bagging is an ensemble method that trains multiple models (typically decision trees) on different random subsets of the data (with replacement). The final prediction is made by aggregating the predictions of all individual models, usually by voting (for classification) or averaging (for regression).
- **AdaBoost (Adaptive Boosting):** AdaBoost is an ensemble technique that combines weak learners (usually decision trees) by adjusting their weights based on their performance. It gives more weight to misclassified instances in each iteration, aiming to improve the overall model's accuracy through a weighted combination of models.

➢ **PROBLEM STATEMENT –** Construct a classifier using the Decision Tree algorithm on two datasets and evaluate its performance by splitting the data into a 75% training set and a 25% test set. Compare the performance of the Decision Tree classifier with Bagging and AdaBoost ensembles, where Bagging and AdaBoost are implemented with 3, 5, 7, and 9 Decision Tree classifiers respectively.

➢ **ANALYSIS** –

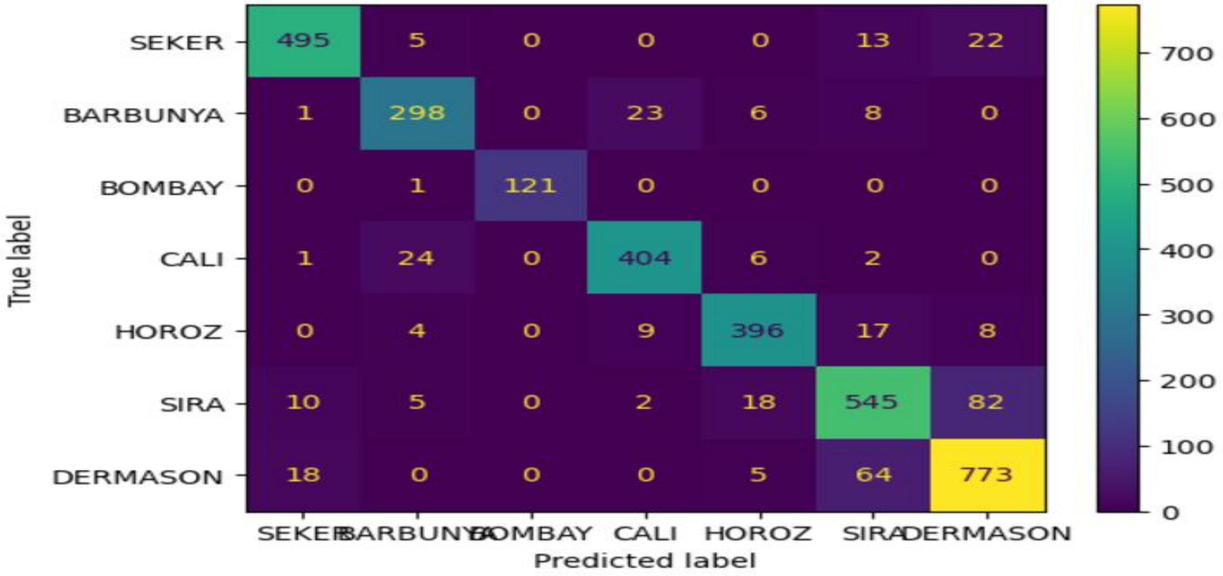
- **Decision Tree Classification Algorithm:**

DATASET – 1



Confusion Matrix (criterion='gini', random_state=0)

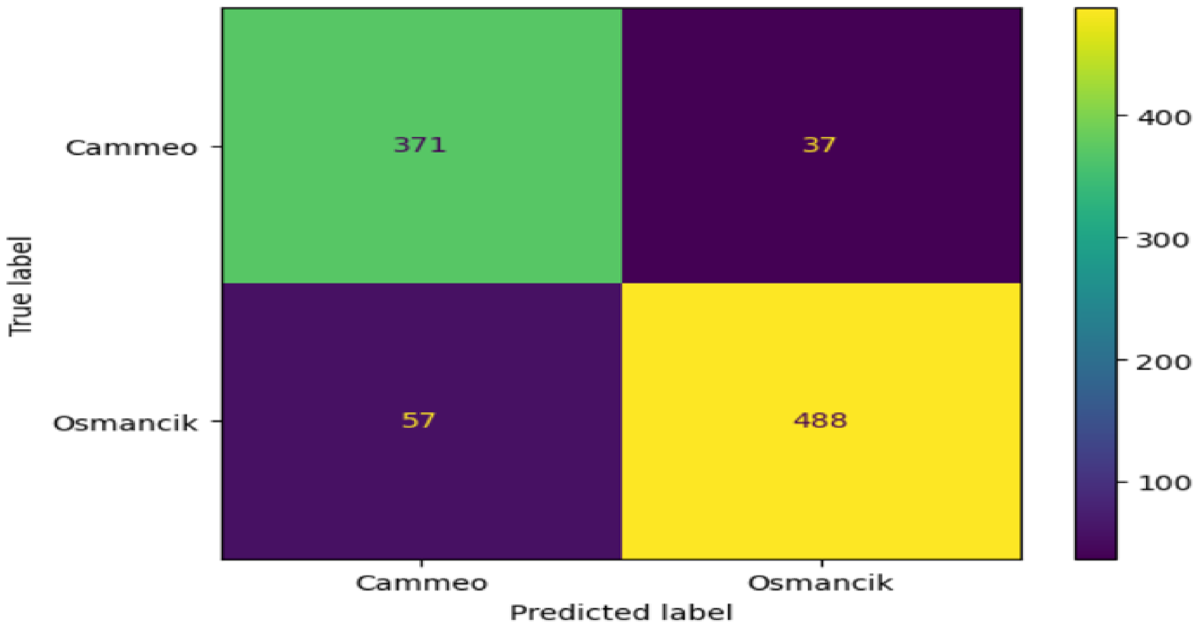
DATASET – 2



Confusion Matrix (criterion='gini', random_state=0)

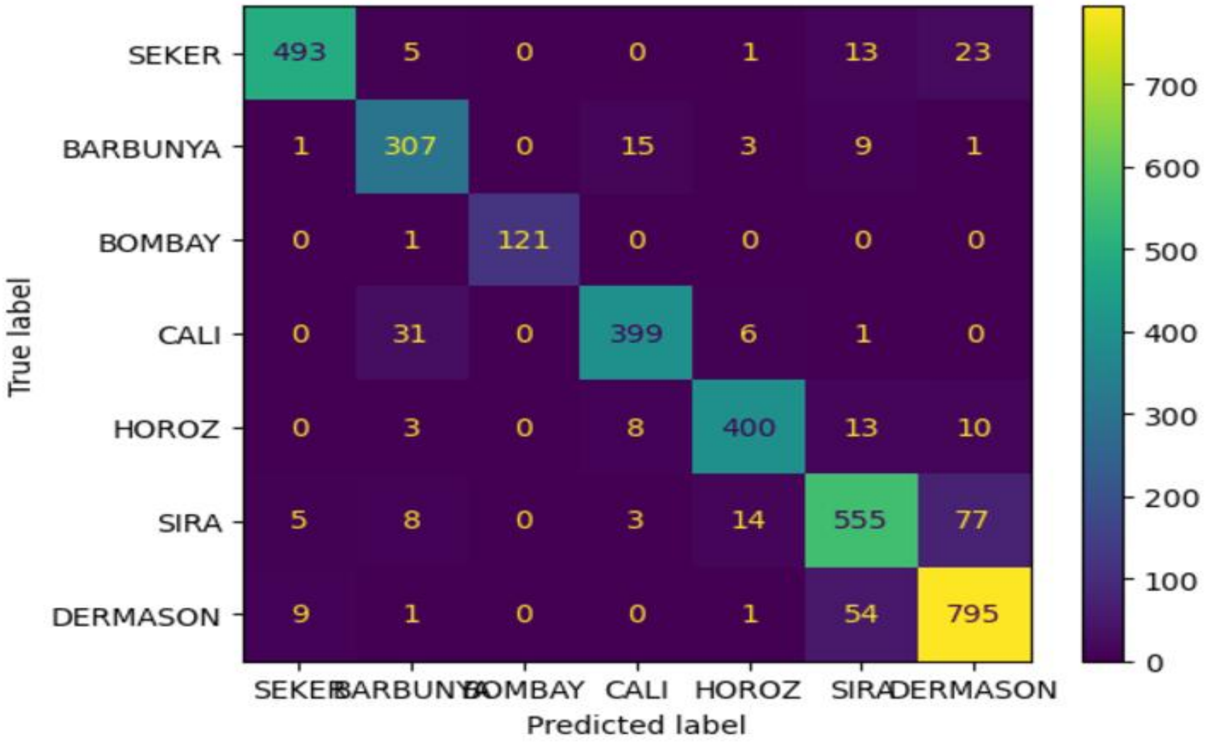
- **Bagging Ensemble Method**:

DATASET – 1



Confusion Matrix (n_estimators=3, random_state=0)

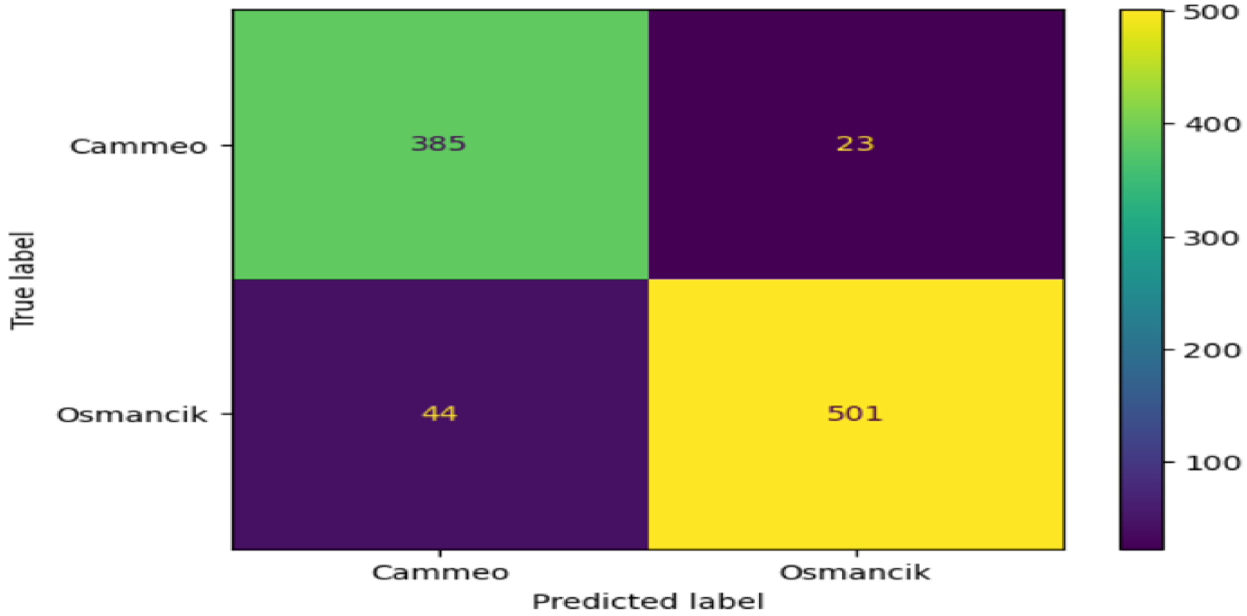
DATASET – 2



Confusion Matrix (n_estimators=3, random_state=100)

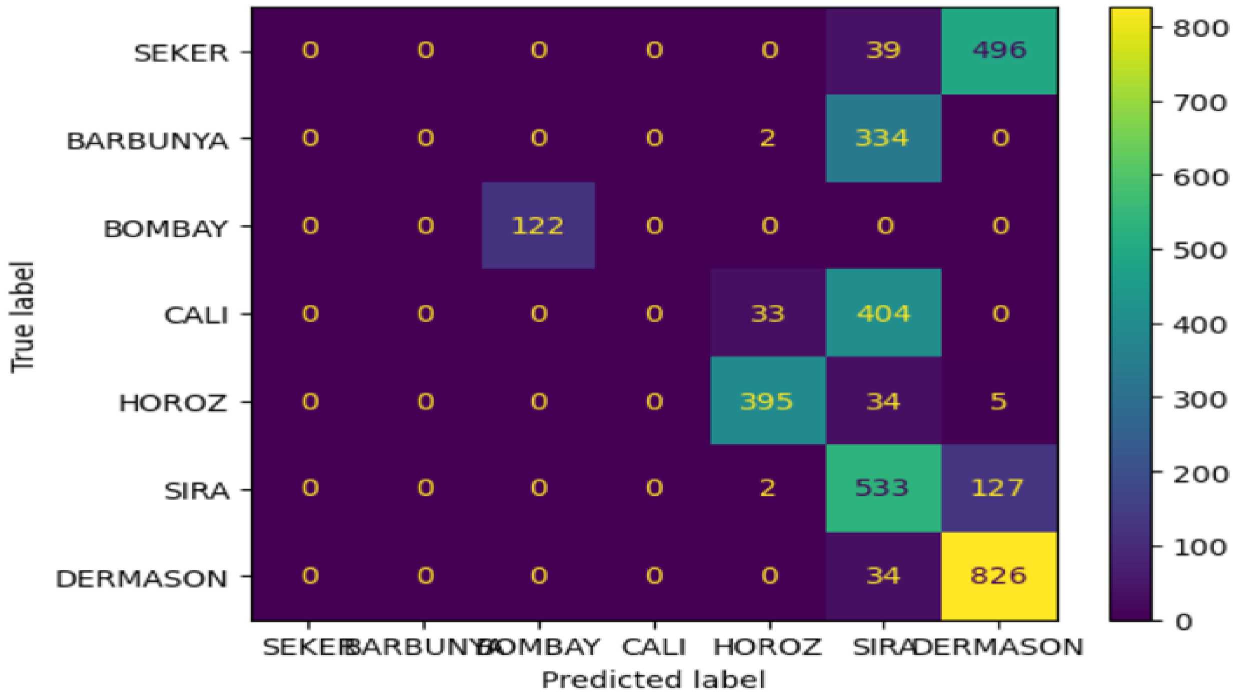
- **<u>AdaBoost Ensemble Method</u>:**

<u>DATASET – 1</u>



Confusion Matrix (n_estimators=3, learning_rate=0.5, random_state=0)

<u>DATASET – 2</u>



Confusion Matrix (n_estimators=3, learning_rate=0.5, random_state=100)

- **Evaluating Performance (Accuracy and Precision Scores of Decision Tree, Bagging and AdaBoost on DATASET – 1):**

A **confusion matrix** is a table used to evaluate the performance of a classification model by comparing predicted and actual outcomes. It has four key components:

True Positives (TP): Correctly predicted positive instances.
True Negatives (TN): Correctly predicted negative instances.
False Positives (FP): Incorrectly predicted as positive.
False Negatives (FN): Incorrectly predicted as negative.

Accuracy = (TN+TP) / (TN+FP+TP+FN)
Precision = (TP) / (TP+FP)

Accuracy using Decision Tree Classification: **89.19202518363065**
Precision using Decision Tree Classification: **90.7749077490775**

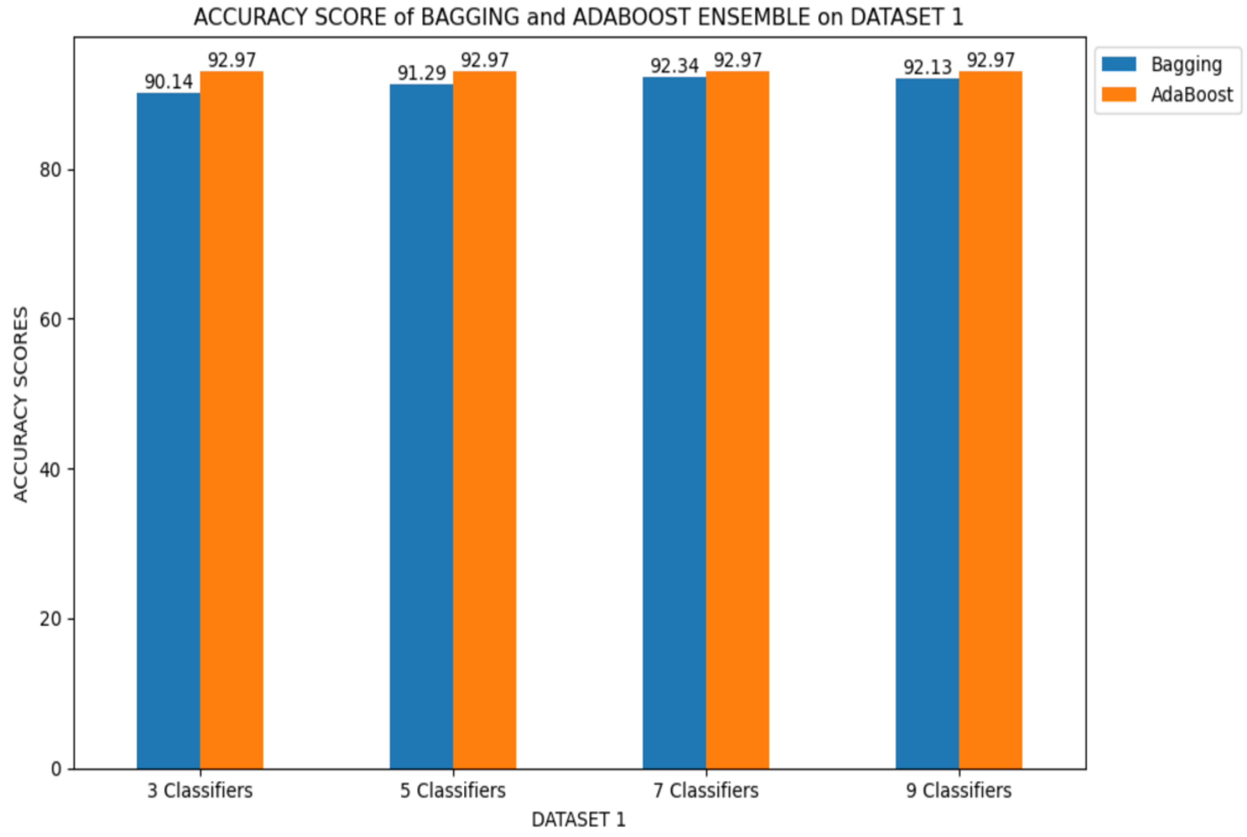
ACCURACY SCORES

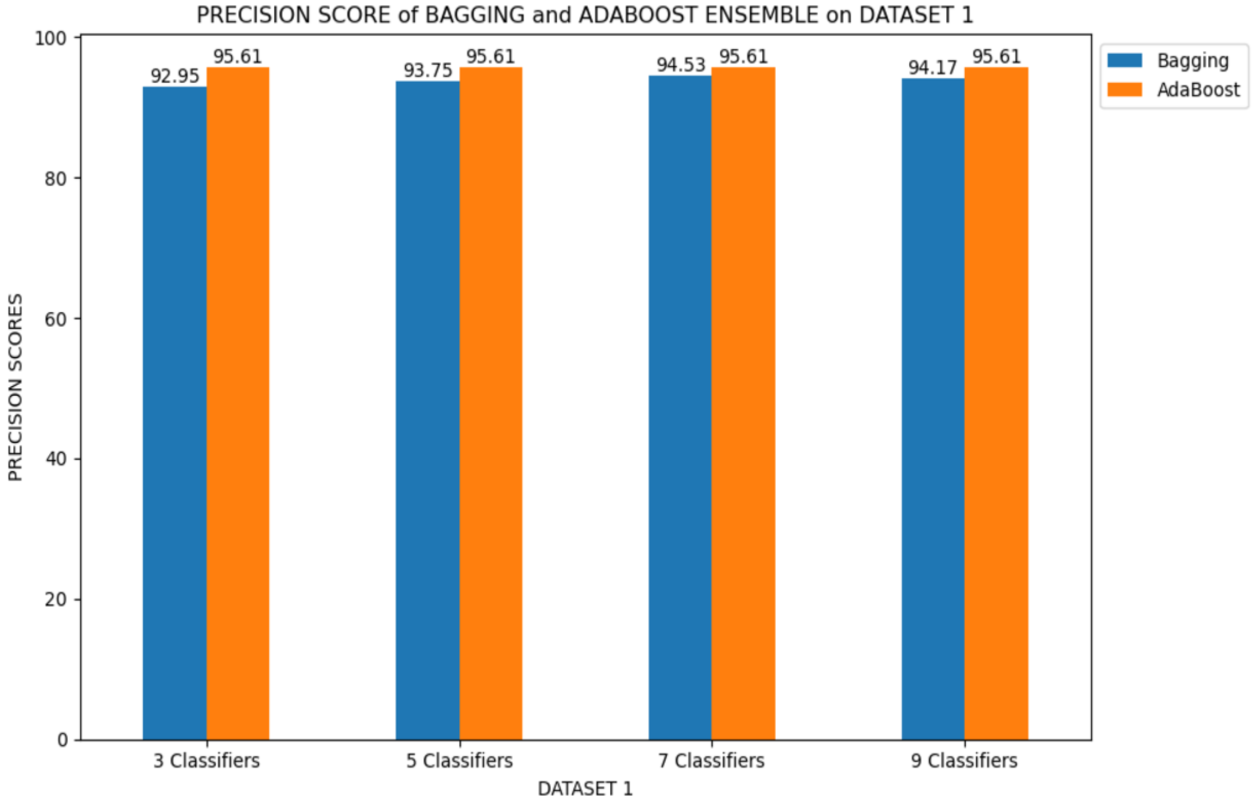
CLASSIFIERS	**BAGGING (%)**	**ADABOOST (%)**
3	90.1364113326338	92.96956977964324
5	91.2906610703043	92.96956977964324
7	92.33997901364114	92.96956977964324
9	92.13011542497377	92.96956977964324

PRECISION SCORES

CLASSIFIERS	**BAGGING (%)**	**ADABOOST (%)**
3	92.95238095238095	95.61068702290076
5	93.75	95.61068702290076
7	94.52830188679245	95.61068702290076
9	94.17293233082707	95.61068702290076

BAR PLOTS OF ACCURACY AND PRECISION SCORES FOR BAGGING AND ADABOOST ON DATASET – 1





- **Evaluating Performance (Accuracy and Precision Scores of Decision Tree, Bagging and AdaBoost on DATASET – 2):**

  Accuracy using Decision Tree Classification: **89.54518606024808**
  Precision using Decision Tree Classification: **91.16425524499333**

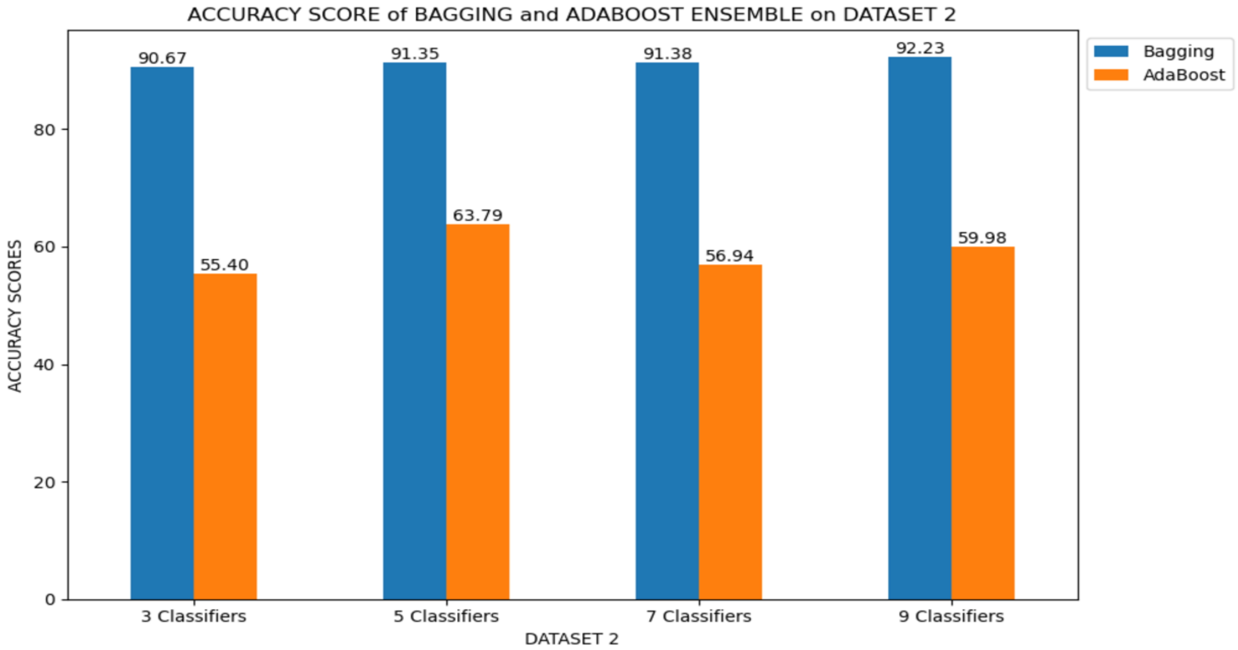
ACCURACY SCORES

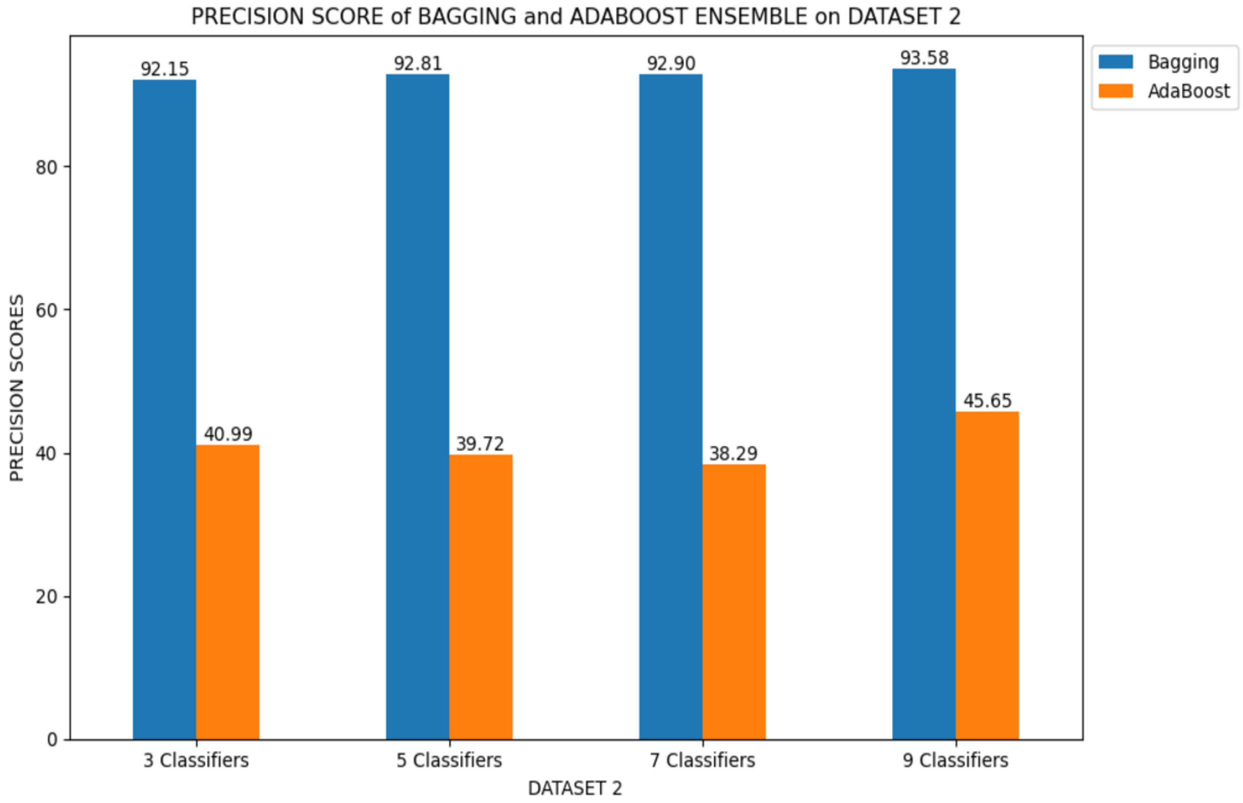
CLASSIFIERS	**BAGGING (%)**	**ADABOOST (%)**
3	90.66745422327229	55.40460720614294
5	91.34672179562907	63.792085056113415
7	91.37625516834022	56.940342587123446
9	92.23272297696397	59.9822799763733

PRECISION SCORES

CLASSIFIERS	**BAGGING (%)**	**ADABOOST (%)**
3	92.15400787849083	40.98903339563457
5	92.81485320449676	39.71771178179341
7	92.90163698267031	38.29190405672081
9	93.5779875689393	45.64523759390837

BAR PLOTS OF ACCURACY AND PRECISION SCORES FOR BAGGING AND ADABOOST ON DATASET – 2





➢ **CONCLUSION** –

For **DATASET - 1**, both Bagging and AdaBoost ensembles outperformed the Decision Tree classifier in terms of accuracy and precision. **Bagging** achieved the **highest accuracy** of **92.34% with 7 classifiers**, while **AdaBoost** consistently maintained a **high accuracy** of approximately **92.97%** across different classifier counts. **Precision** scores for AdaBoost were also **superior**, with a peak of **95.61%** across all configurations.

For **DATASET - 2**, while Bagging and AdaBoost improved accuracy and precision over the Decision Tree classifier, the performance was more varied. **Bagging** reached its **highest accuracy** of **92.23% with 9 classifiers**, whereas **AdaBoost's** accuracy and precision were notably **lower**, especially with fewer classifiers, reaching a **peak precision** of **45.65% at 9 classifiers**.

Overall, **Bagging and AdaBoost provided better performance than the Decision Tree classifier**, with **Bagging generally yielding more stable results across both datasets**.

<u>REPORT – 4</u>

<u>QUESTION</u> - Download a dataset and check whether outliers are present in the dataset. Use different methods of outlier detection and compare their performance.

➢ <u>ABOUT DATASET</u> –

- **<u>Name</u>: Wine Quality**
- **<u>Source</u>:** https://archive.ics.uci.edu/dataset/186/wine+quality
- **<u>Description</u>:** The dataset contains information about various chemical properties of wines, used to predict their quality. The dataset includes both red and white wine samples and is commonly used for regression and classification tasks.
- **<u>Columns</u>:**
    - **fixed_acidity** – float64
    - **volatile_acidity** – float64
    - **citric_acid** – float64
    - **residual_sugar** – float64
    - **chlorides** – float64
    - **free_sulfur_dioxide** – float64
    - **total_sulfur_dioxide** – float64
    - **density** – float64
    - **pH** – float64
    - **sulphates** – float64
    - **alcohol** – float64
    - **quality** – int64

➢ <u>ABOUT DIFFERENT OUTLIER DETECTION METHODS</u> –

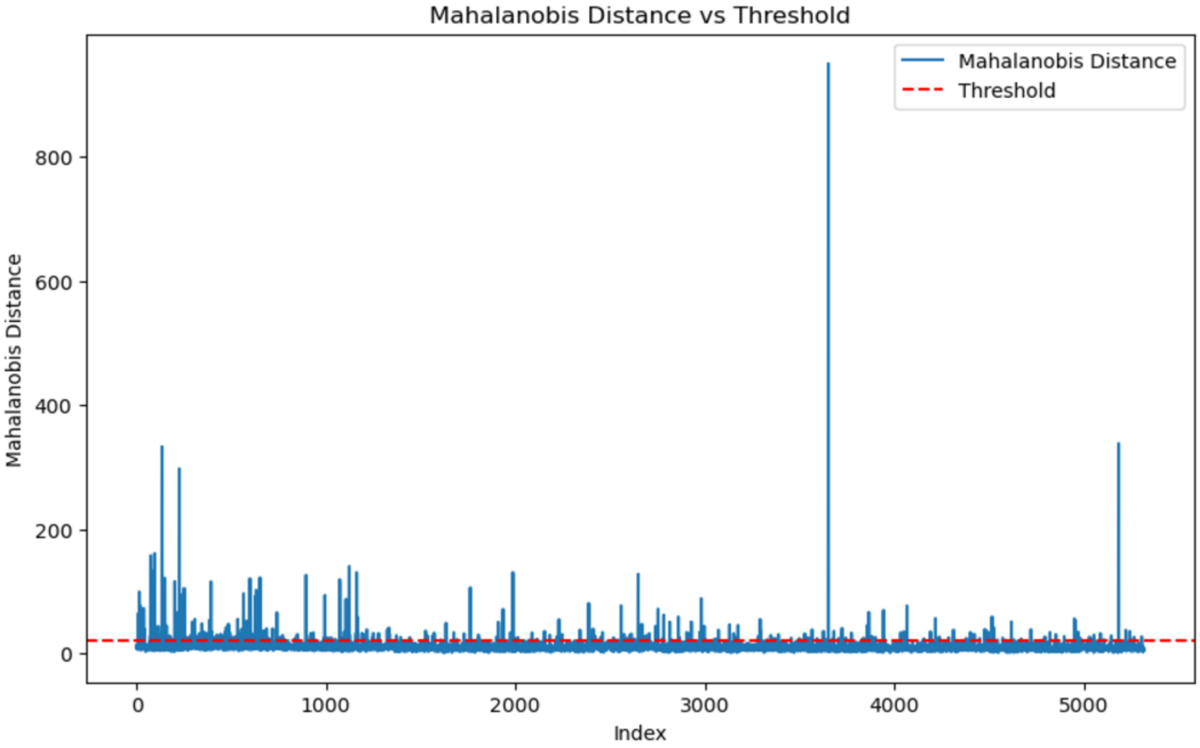
- **<u>Z-Score</u>:** Identifies outliers by measuring how far a data point is from the mean in terms of standard deviations. Points with z-scores beyond a threshold (e.g., ±3) are considered outliers.
- **<u>Mahalanobis Distance</u>:** Measures the distance of a point from the mean of a distribution while accounting for correlations between features. It is effective for multivariate outlier detection.

- **IQR (Interquartile Range):** Detects outliers by identifying points outside the range of Q1−1.5×IQR to Q3+1.5×IQR, where IQR=Q3−Q1.
- **Local Outlier Factor (LOF):** Measures the density of a point relative to its neighbors; points with significantly lower density are considered outliers.
- **k-Nearest Neighbors (k-NN):** Flags points as outliers based on the distance to their k nearest neighbors. Larger distances may indicate outliers.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Identifies outliers as points that do not belong to any dense cluster, based on a minimum number of neighbors within a specified distance.

➢ **PROBLEM STATEMENT –** Analyze a dataset to identify the presence of outliers using various outlier detection methods, such as Z-Score, Mahalanobis Distance, IQR, Local Outlier Factor, k-Nearest Neighbors, and DBSCAN. Compare the effectiveness of these methods in detecting outliers.

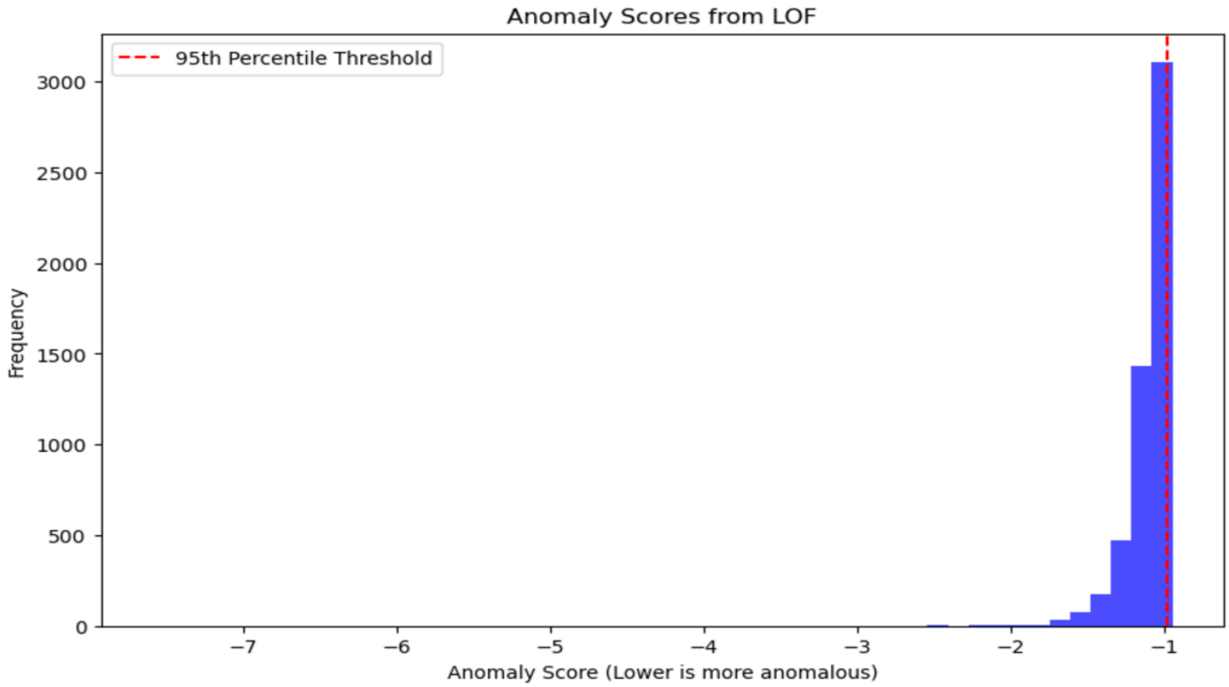
➢ ANALYSIS –

- **Mahalanobis Distance (Statistical Approach – Parametric):**



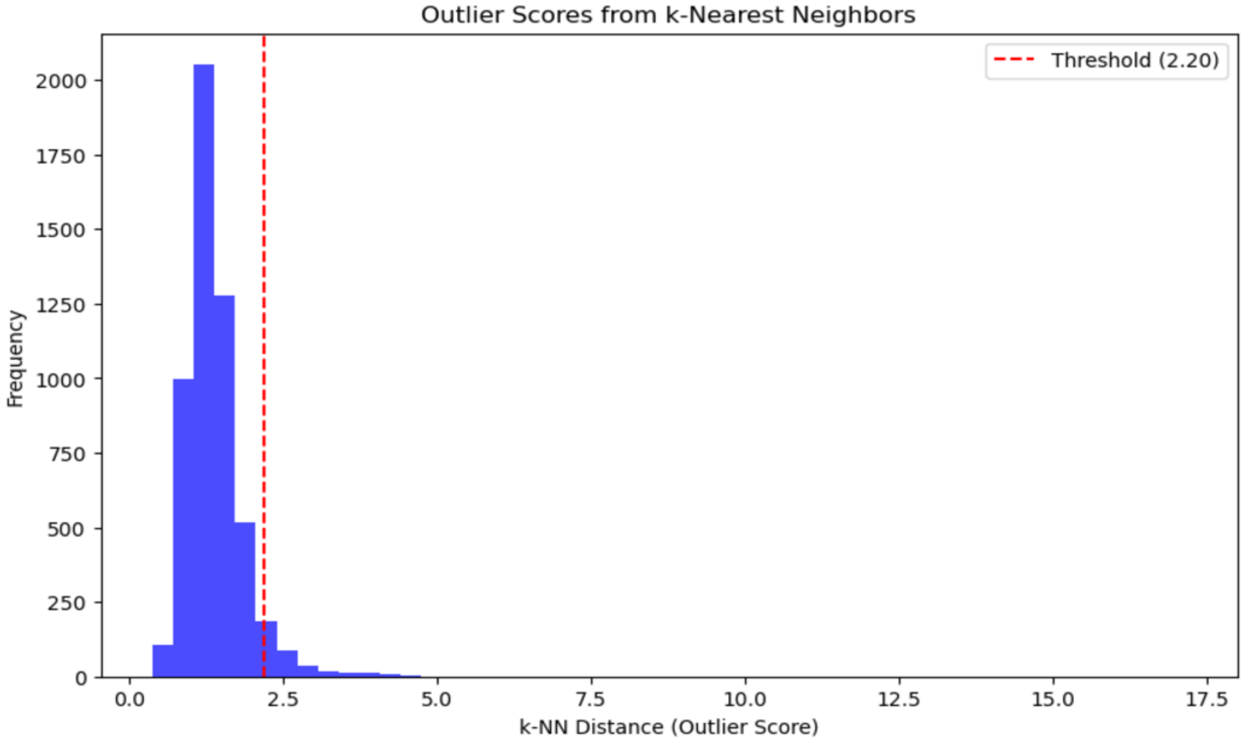
Line Plot for Mahalanobis Distance vs Threshold

- **Local Outlier Factor (LOF -> Proximity Based Approach):**



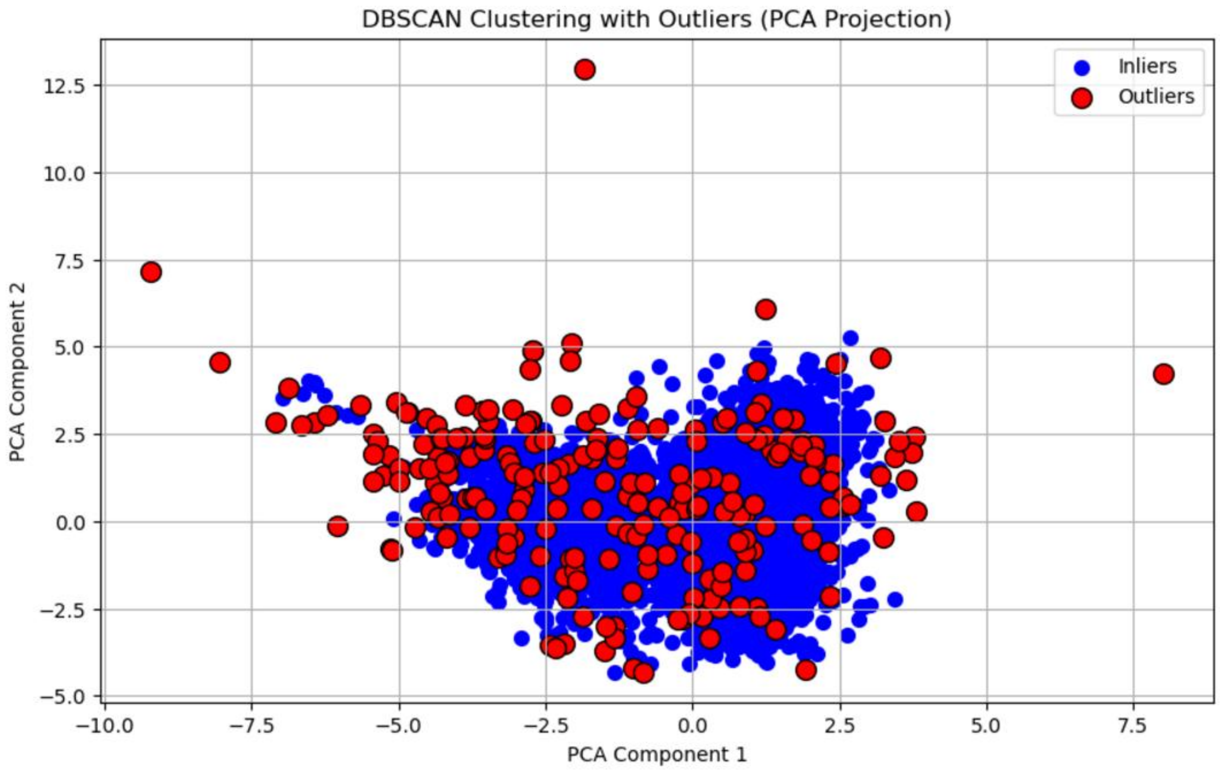
Histogram for Anomaly Scores (n_neighbors=20)

- **k-Nearest Neighbors (k-NN -> Proximity Based Approach):**



Histogram for Outlier Scores (n_neighbors=5)

- **DBSCAN (Clustering Based Approach):**

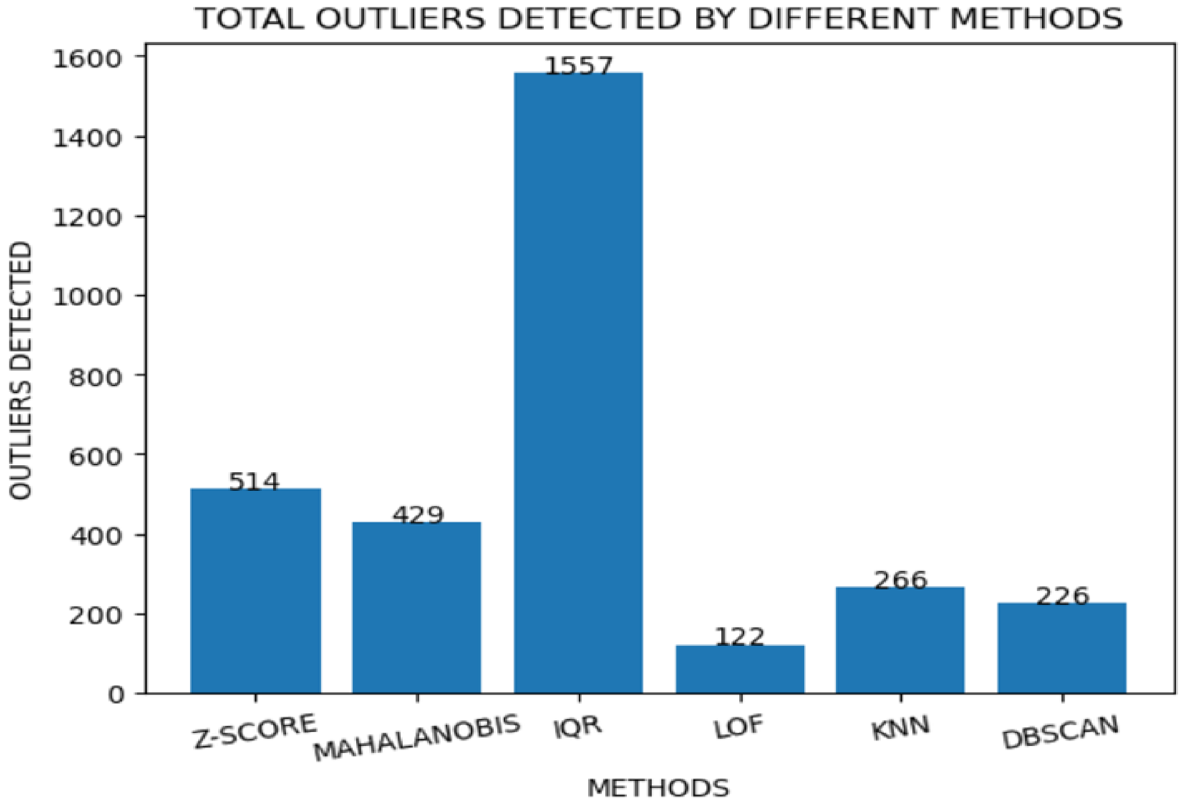


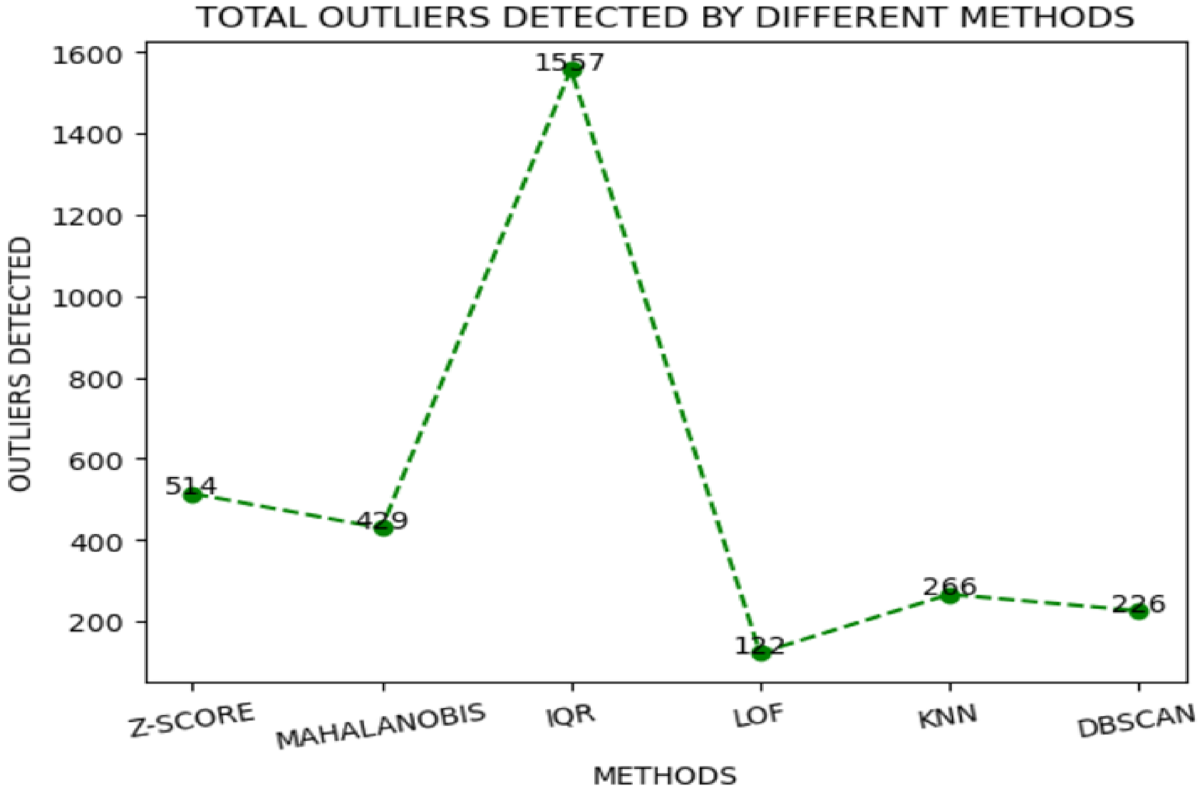
Scatter Plot for DBSCAN Clustering with Outliers (PCA Projection) (eps=2, min_samples=5)

- **Comparing Performance of all Outlier Detection Methods :**

METHOD	**NO. OF OUTLIERS DETECTED**
Z-Score	514
Mahalanobis Distance	429
IQR (Interquartile Range)	1557
Local Outlier Factor (LOF)	122
k-Nearest Neighbors (k-NN)	266
DBSCAN	226

BAR AND LINE PLOTS FOR COMPARING THE PERFORMANCE OF DIFFERENT METHODS (TOTAL DATASET ENTRIES – 5318)





➢ CONCLUSION –

Different outlier detection methods identified varying numbers of outliers, reflecting their unique approaches and sensitivities. The **IQR method** detected the highest number of outliers (**1557**), suggesting it is sensitive to extreme values. In contrast, the **Local Outlier Factor (LOF)** identified the fewest (**122**), focusing on local density variations. Methods like **Z-Score** (**514**), **Mahalanobis Distance** (**429**), **k-NN** (**266**), and **DBSCAN** (**226**) fell in between, each offering distinct insights into data anomalies based on their criteria. The results highlight the importance of method selection depending on the dataset's characteristics and the specific goals of the analysis.

REPORT – 5

QUESTION - Perform CluStream algorithm on any time series data from Kaggle and compare its output with that of K-means clustering. Evaluate the cluster quality by changing the algorithm's parameters.

➢ ABOUT DATASET –

- **Name: Crude Oil Stock Dataset 2000-2024**
- **Source:** https://www.kaggle.com/datasets/mhassansaboor/crude-oil-stock-dataset-2000-2024
- **Description:** This dataset contains historical stock price data for Crude Oil from 2000 to 2024. This data is extracted by using Python's yfinance library and it provides detailed insights into Crude Oil's stock performance over the years. It includes daily values for the stock's opening and closing prices, adjusted close price, high and low prices, and trading volume. This dataset is ideal for time series analysis, stock trend analysis, and financial machine learning projects such as price prediction models and volatility analysis. The dataset is extracted from Yahoo Finance.
- **Columns:**
  - **Date** – object
  - **Adj_Close** – float64
  - **Close** – float64
  - **High** – float64
  - **Low** – float64
  - **Open** – float64
  - **Volume** – int64

➢ ABOUT ALGORITHMS –

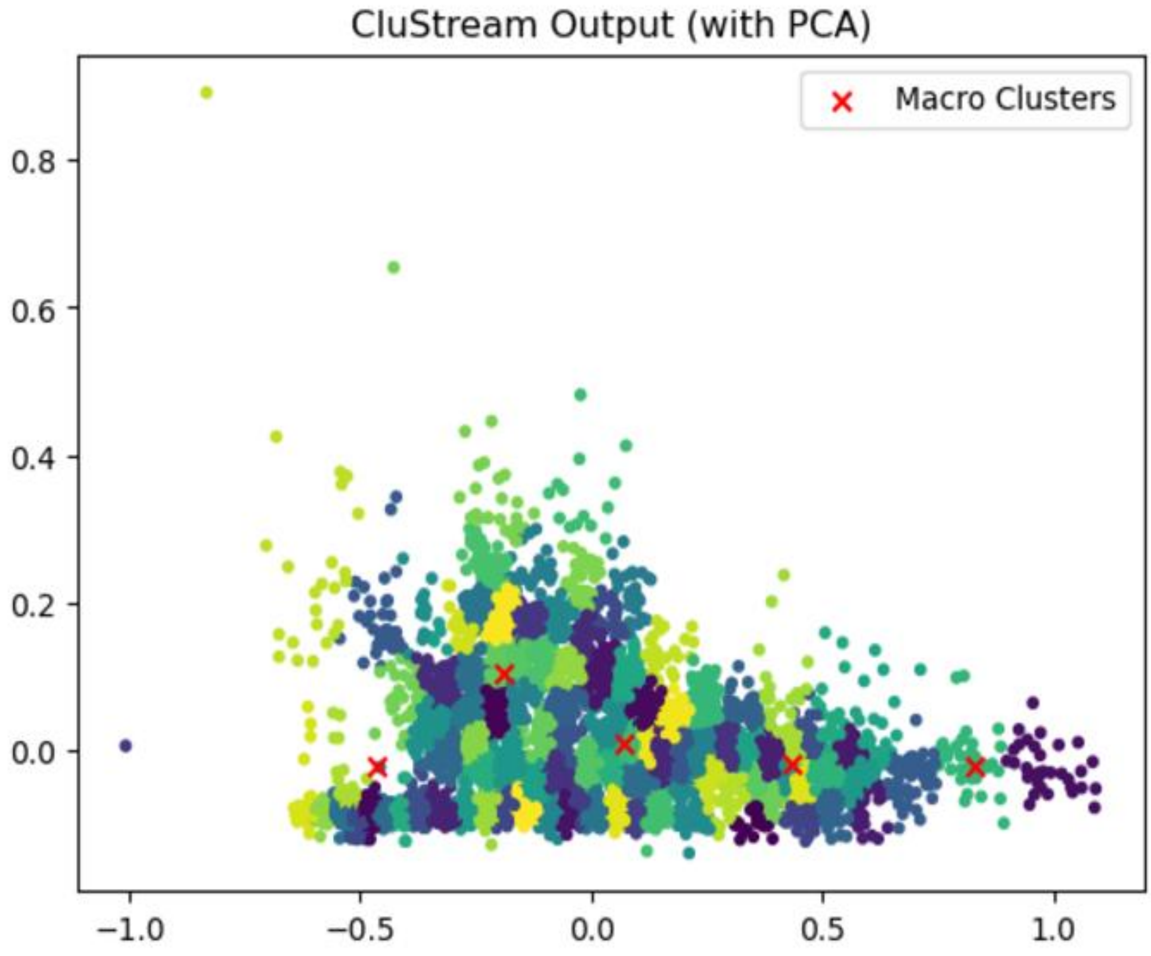
- **CluStream Algorithm:** CluStream is an online-offline clustering algorithm designed for data streams. It performs incremental clustering in real-time (online phase) and periodically summarizes the clusters for more detailed analysis in the offline phase, making it suitable for dynamic, high-speed data.
- **K-Means Clustering Algorithm:** K-Means is a centroid-based clustering algorithm that partitions data into k clusters by iteratively assigning points to the nearest cluster centroid and updating the centroids until

convergence. It is efficient for large datasets but sensitive to the initial choice of centroids and outliers.

- **PROBLEM STATEMENT –** Implement the CluStream algorithm on a time series dataset from Kaggle and compare its clustering results with K-Means. Analyze and evaluate the quality of clusters by varying the parameters of both algorithms to observe their impact on clustering performance.

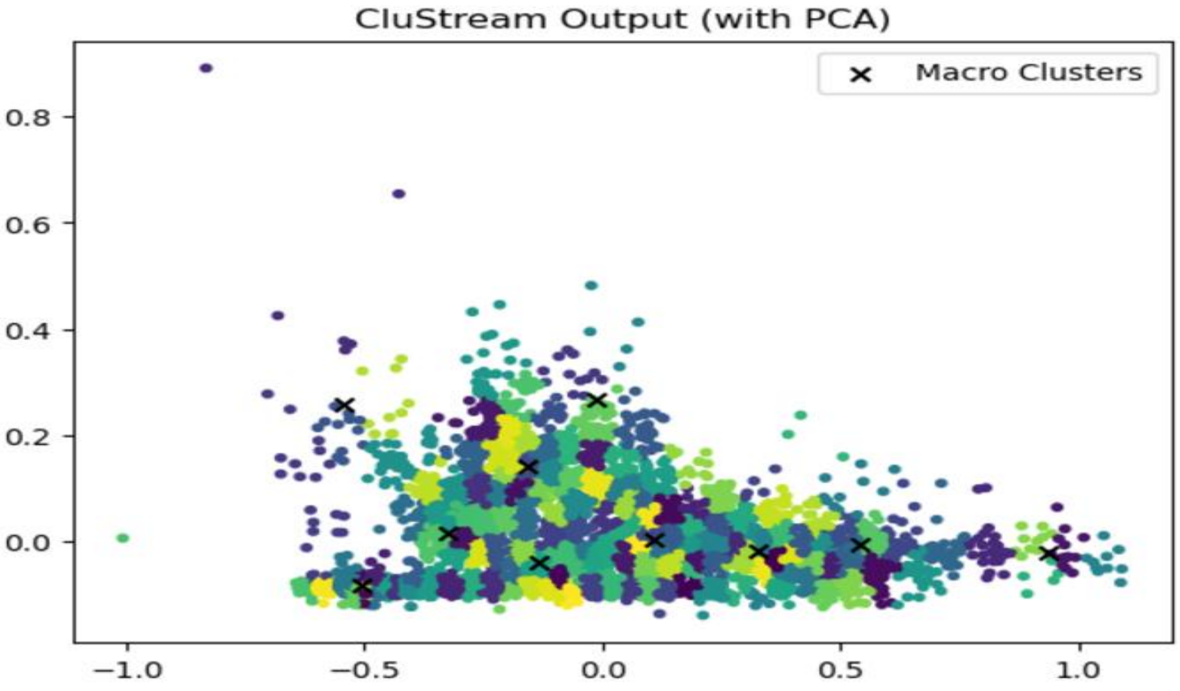
- **ANALYSIS –**
  - **CluStream Algorithm**:

PARAMETER SET - 1



Scatter Plot for CluStream with PCA (n_micro_clusters=100, batch_size=400, n_macro_clusters=5)

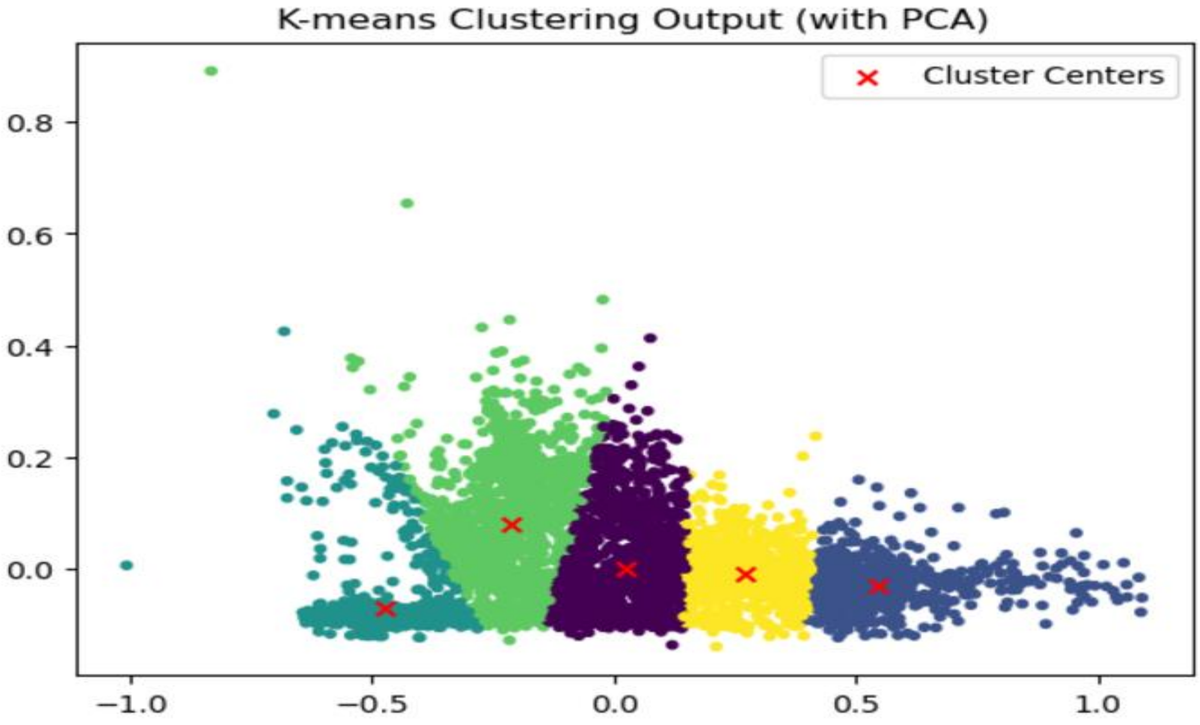
PARAMETER SET - 2



Scatter Plot for CluStream with PCA (n_micro_clusters=150, batch_size=500, n_macro_clusters=10)

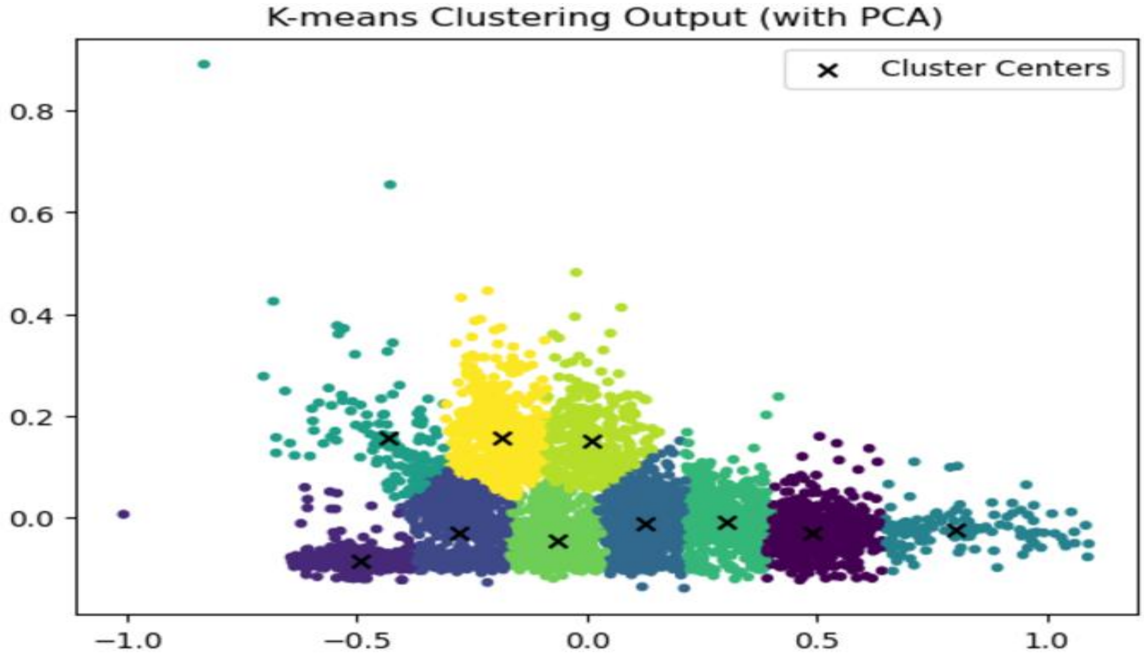
- **K-Means Clustering Algorithm**:

PARAMETER SET - 1



Scatter Plot for K-Means with PCA (n_clusters=5)

PARAMETER SET - 2



Scatter Plot for K-Means with PCA (n_clusters=10)

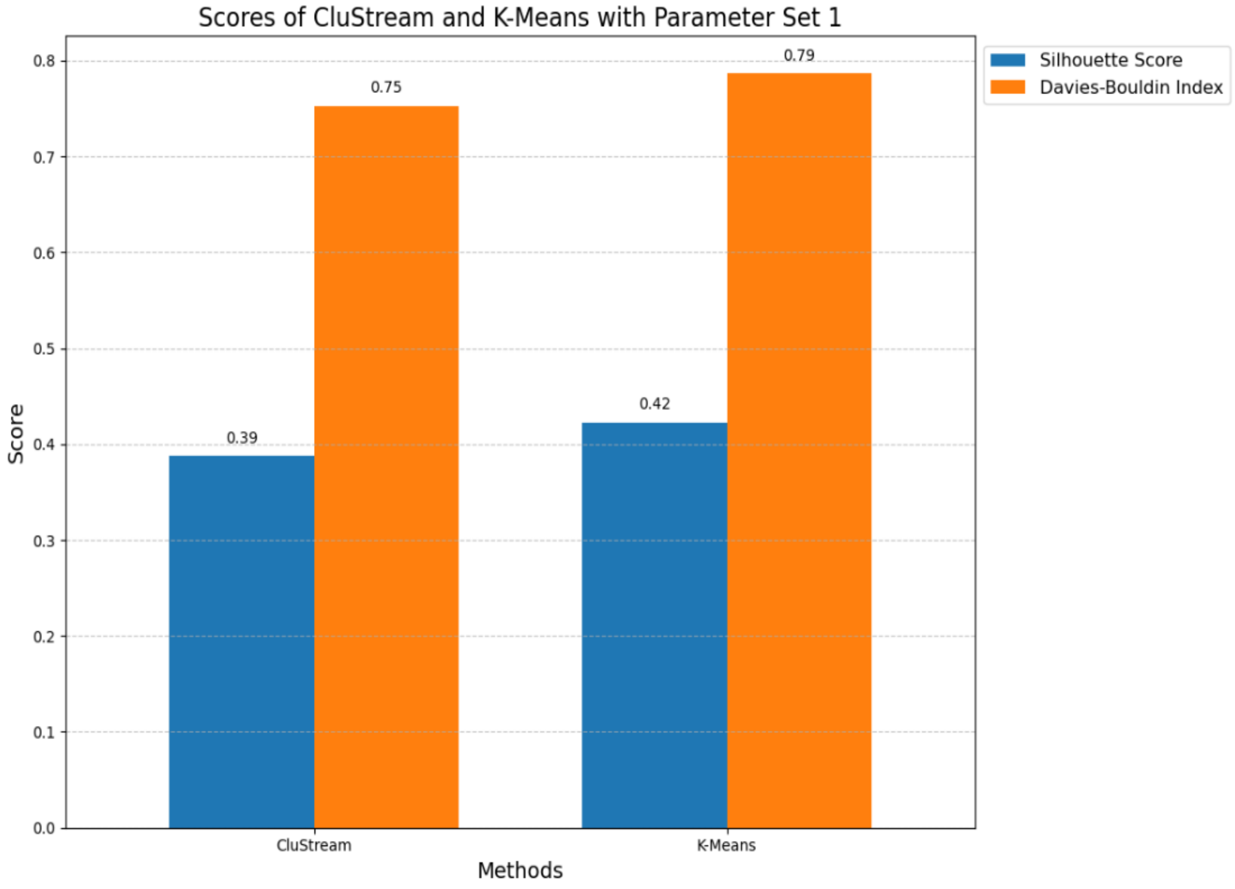
- **Evaluating Cluster Quality:**

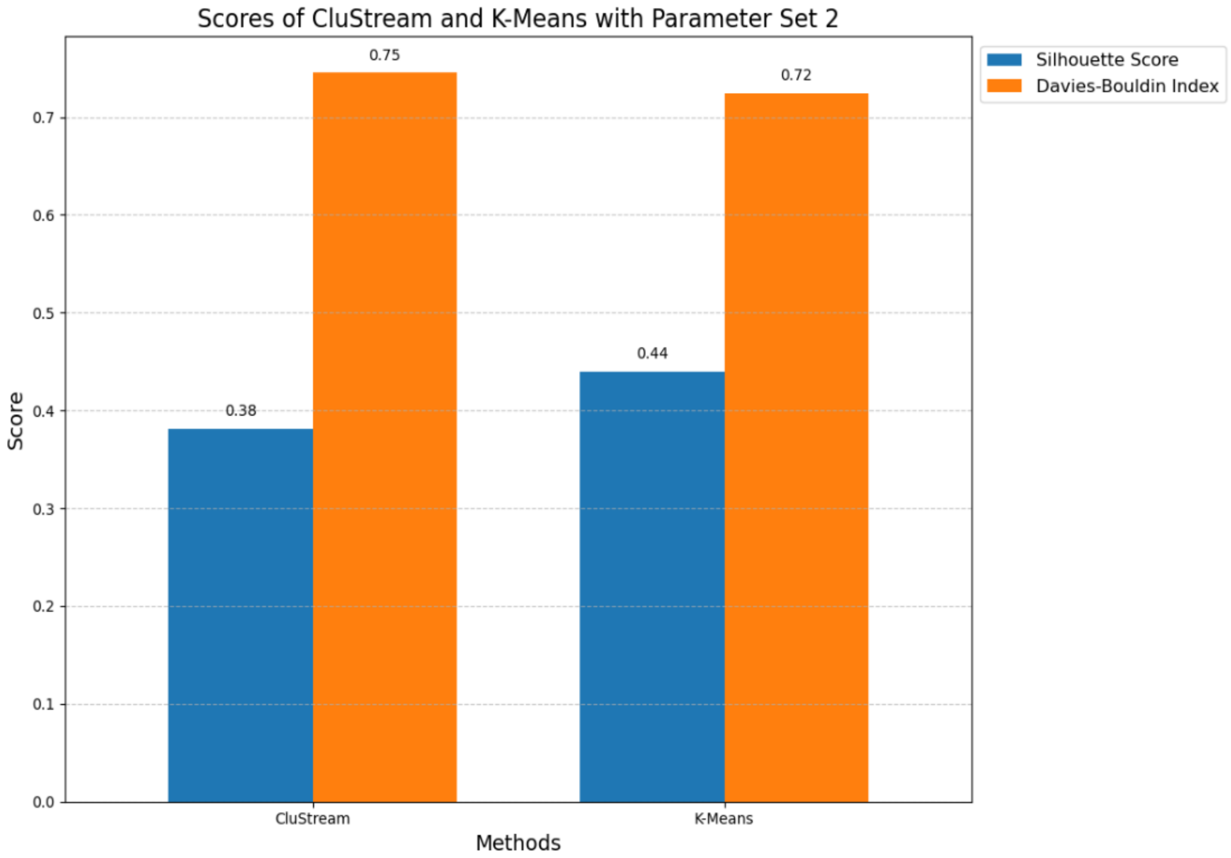
  Silhouette Score: The Silhouette Score measures the quality of clustering by evaluating how similar data points are within a cluster (cohesion) compared to points in other clusters (separation). Its value ranges from −1 to 1, where higher values indicate better-defined clusters.

  Davies-Bouldin Index: The Davies-Bouldin Index evaluates clustering performance based on the ratio of intra-cluster dispersion to inter-cluster separation. A lower index indicates better clustering quality, with more compact and well-separated clusters.

	CLUSTREAM		K-MEANS	
PARAMETER SETS	SILHOUETTE SCORE	DAVIES-BOULDIN INDEX	SILHOUETTE SCORE	DAVIES-BOULDIN INDEX
n_micro_clusters=100, batch_size=400, n_macro_clusters=5, n_clusters=5	0.3877340889578921	0.7525412873490852	0.42284040685832375	0.786174610378013
n_micro_clusters=150, batch_size=500, n_macro_clusters=10, n_clusters=10	0.38127444704479757	0.745077890870974	0.43953771821244747	0.7240647694610353

BAR PLOTS OF SILHOUETTE SCORE AND DAVIES-BOULDIN INDEX FOR CLUSTREAM ALGORITHM AND K-MEANS CLUSTERING ALGORITHM FOR BOTH PARAMETER SETS





➢ <u>**CONCLUSION**</u> **–**

The comparison between **CluStream** and **K-Means** based on the table indicates that **K-Means** achieves **slightly higher Silhouette Scores**, suggesting better cluster cohesion and separation in both parameter sets. For the **Davies-Bouldin Index**, **K-Means performs better in the second parameter set**, showing improved compactness and separation of clusters compared to **CluStream**.

Overall, **K-Means demonstrates marginally better clustering quality** in these scenarios.

<u>**THANK YOU**</u>