Nada Gaber	20217014
Shehab Sayed	20217019
Amr Gamal	20217007
Meral Mostafa	20217012
Doha Sami	20216054
Malak Gamal	20217009

Assignment 2

Clustering Analysis Report

Dataset:

The Social Media Sentiments Analysis Dataset shows a mix of emotions and trends from various platforms, including user posts, timestamps, hashtags, countries, likes, and retweets. We'll use clustering to look at how often users post (temporal activity) and how much they interact with others (engagement level). This helps us see patterns in when people are active and how much attention they get, giving us insights for better social media strategies.

Structure:

- Unnamed: 0.1
- Unnamed: 0
- Text
- Sentiment
- Timestamp
- User
- Platform
- Hashtags
- Retweets
- Likes
- Country
- Year
- Month

- Day
- Hour

(732 rows, 15 columns)

Preprocessing:

- We dropped *Timestamp, User, Platform, Hashtags, Country, Unnamed: 0.1, Unnamed: 0, Text* because we don’t need them in the clustering.
- Using the Hopkins statistic, we tested the dataset's tendency for clustering and got a result of 0.997 which means high clustering tendency and meaningful clusters can likely be found in the dataset.
- We use Label Encoding to convert categorical columns into numerical representations.
- Numerical columns ('Retweets', 'Likes', 'Year', 'Month', 'Day', 'Hour') are standardized using StandardScaler. Standardization scales the features to have a mean of 0 and a standard deviation of 1, which is important for PCA since it's sensitive to the scale of the features.
- We used Likes, Retweets, Sentiment to represent Engagement Level and Year, Month, Day, Hour to represent Temporal Activity.
- We perform PCA separately on two sets of columns: 'Engagement_Level' and 'Temporal_Activity'. We specify the number of principal components to be 1 for each set. We combine the results of PCA for each aspect into a single DataFrame ('X_pca').

DBSCAN:

The main idea behind DBSCAN is to group together data points that are closely packed together, defining clusters as areas of high density separated by areas of low density. Unlike centroid-based clustering algorithms like k-means, DBSCAN doesn't require specifying the number of clusters beforehand, making it particularly useful when the number of clusters is not known a priori or when dealing with datasets where the clusters have irregular shapes.

DBSCANmodel = DBSCAN(eps= 3.5, min_samples= 11,metric='euclidean').fit(X_pca):-

epsilon (eps)defines the radius within which to search for neighboring points.
min_samples specifies the minimum number of points within ε for a point to be considered a core point.

n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

n_noise_ = list(labels).count(-1):-

Here, the number of clusters is calculated. It counts the unique cluster labels assigned by DBSCAN, excluding noise points (labeled as -1).
counting the number of noise points in the dataset, which are labeled as -1 by DBSCAN.

Cluster Visualization:

The clusters are visualized using a scatter plot, with each cluster represented by a different color and noise points shown in gray.

The plot provides a visual representation of how the data points are grouped into clusters based on their engagement level and temporal activity.

Evaluation:

The Calinski-Harabasz score is calculated to evaluate the quality of the clustering results.

This score measures the ratio of between-cluster dispersion to within-cluster dispersion, where a higher score indicates better-defined clusters.

The Calinski-Harabasz (DBSCAN) measure = 1003.3 which is reasonable.

KMedoids:

Hopkins Statistic Analysis:

- The Hopkins Statistic is used to evaluate the clustering tendency to cluster . A higher Hopkins Statistic value (closer to 1) indicates a higher tendency to cluster.
- Hopkin statistic = 0.997581 so it is high tendency to cluster.

Data Preprocessing and PCA:

- Numerical columns were standardized using StandardScaler to bring them to a common scale.
- PCA was applied to reduce the dimensionality of the data while retaining important information.

Elbow Method and Emprical Method for Optimal K in KMedoids:

- The Elbow Method was used to determine the optimal number of clusters (K) for the KMedoids algorithm we used k = 4
- By plotting the sum of squared distances against different values of K, an "elbow" point was identified where the rate of decrease in the sum of squared distances slows down. This point often represents a good choice for K.
- Also we used Emprical Method and it gave us k = 9

Clustering with KMedoids:

We fitted KMedoids with different values of K (4 and 9) to explore different clustering configurations.

The cluster labels obtained from KMedoids were added to the DataFrame for further analysis and visualization.

Cluster Visualization with PCA:

The clusters formed by KMedoids were visualized using PCA, where each point represents a data instance projected onto the reduced-dimensional space.

The clusters were visualized providing insights into how the data points are grouped.

Intra-cluster Distances and Cluster Centroids:

- Intra-cluster distances were calculated to measure the compactness of clusters formed by KMedoids.

- Cluster centroids and pairwise distances between them were computed, providing information about the separation between clusters.

Calinski-Harabasz Score for KMedoids:

- K = 9 is better than k = 4
- The Calinski-Harabasz score was calculated to evaluate the clustering quality of KMedoids.
- Calinski-Harabasz Score = 5004.57, which indicates that the clustering algorithm has produced clusters that are well-separated and distinct, making the clustering solution statistically significant.
- The Calinski-Harabasz score is preferred over the Davies-Bouldin index because it considers both intra-cluster compactness and inter-cluster separation, providing a more comprehensive evaluation of clustering quality.
- The silhouette score assesses the compactness and separation of clusters at the individual sample level, while the Calinski-Harabasz score evaluates the overall separation and compactness of clusters based on the entire dataset.
- So we finally used the Calinski-Harabasz score.

Model Selection:

Finally, the code compares the Calinski-Harabasz scores of the DBSCAN clustering with another clustering algorithm (presumably K-medoids) and selects the model with the higher score as the best model.

The Calinski-Harabasz (DBSCAN) measure = 1003.3,

The Calinski-Harabasz (K-medoids) measure = 5004.5

K-medoids is performing significantly better than DBSCAN according to the Calinski-Harabasz measure, and the clusters identified by K-medoids are more distinct and well-separated in the feature space compared to those identified by DBSCAN.