Question 2 Report: Clustering Analysis of Weather Data with K-means Algorithm

---

- **First and Last Name**: Fahrettin Solak
- **School Number**: 201401053
- **The Subject of the Problem**: Grouping of Weather Data with K-means Clustering
- **The Data Set and Link Used**:
    - **Data Set Source**: Kaggle
    - **Link**: Weather Forecast Data on Kaggle

Methods Used:

- K-means Clustering
- Data Pre-Processing
- Hyperparameter Optimization
- Evaluation Metrics:
    - Silhouette Score
    - Davies-Bouldin Score
    - Calinski-Harabasz Score

---

1. Introduction and Data Set

Project Objective and Data Set Selection:

The aim of this project is to divide similar weather conditions into clusters based on weather-related data and to make inferences about these clusters. The K-means clustering algorithm has been used to divide this data into a certain number of clusters. Our goal is to make this data more meaningful by analyzing weather data and observing which clusters form under certain conditions.

Our data set is the "**Weather Forecast Data**" data set available on the **Kaggle** platform. This data set includes various parameters related to the weather and makes it possible to cluster these parameters.

Characteristics of the Data Set:

- **Temperature**: The temperature value of the air (in °C)
- **Humidity**: Humidity in the air (in%)
- **Wind Speed**: Indicates the wind speed
- **Cloud Cover**: Indicates the percentage of clouds in the sky
- **Rain**: It refers to whether it is raining (rain or no rain)

There are a total of **5000 data lines** and **5 properties** in the data set. Since it was observed that there was no missing data at first glance, there was no need to perform operations such as filling in missing values during the data preprocessing process.

First Look and Statistical Analysis of the Data Set:

To get an overview of the data set first, we examined the first 5 lines of the data set and the statistical summary. In this way, we had the chance to better understand the structure and distribution of the data set.

- **The First 5 Lines of the Data Set**: To observe the overall structure of the data set.

	Temperature	Humidity	Wind_Speed	Cloud_Cover	Pressure	Rain
0	23.720338	89.592641	7.335604	50.501694	1032.378759	rain
1	27.879734	46.489704	5.952484	4.990053	992.614190	no rain
2	25.069084	83.072843	1.371992	14.855784	1007.231620	no rain
3	23.622080	74.367758	7.050551	67.255282	982.632013	rain
4	20.591370	96.858822	4.643921	47.676444	980.825142	no rain

---

2. Data Pre-Processing

Digitization of Categorical Variables:

The Rain column in the data set is a categorical variable, and this variable needs to be digitized in order to be included in the model. The rain and no rain expressions in this column have been translated to the values **1** and **0**, respectively. In this way, the model has been put in a position to analyze this variable.

Scaling of Data:

Since the scales of the numerical values in the dataset are different, scaling the data using **StandardScaler** allowed each property to be evaluated with equal weight. With **StandardScaler**, the data was scaled by subtracting from the average value and dividing it by the standard deviation, and thus we ensured that the characteristics were in similar.

	Temperature	Humidity	Wind_Speed	Cloud_Cover	Pressure	Rain
0	0.155431	1.265393	-0.444814	0.028972	0.894714	2.638519
1	0.723225	-0.895074	-0.684143	-1.534074	-1.074570	-0.379000
2	0.339547	0.938599	-1.476731	-1.195246	-0.350663	-0.379000
3	0.142018	0.502270	-0.494138	0.604355	-1.568924	2.638519
4	-0.271701	1.629599	-0.910571	-0.068058	-1.658406	-0.379000

---

3. Model Training and Evaluation

Determining the Number of Clusters using the Elbow Method:

The **Elbow Method** is used to determine the optimal number of clusters (k) in the K-means clustering algorithm. This method aims to find the appropriate number of clusters by looking at the WCSS (Within Cluster Sum of Squares) values. By looking at the elbow point formed on the Elbow graph, it was determined as k=4.

- **Elbow Graph**: A fracture was observed at the point **k=4** and this value was chosen as the most appropriate number of clusters.

```
k = 4
kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=300, n_init=10, random_state=42)
kmeans.fit(scaled_data)
```

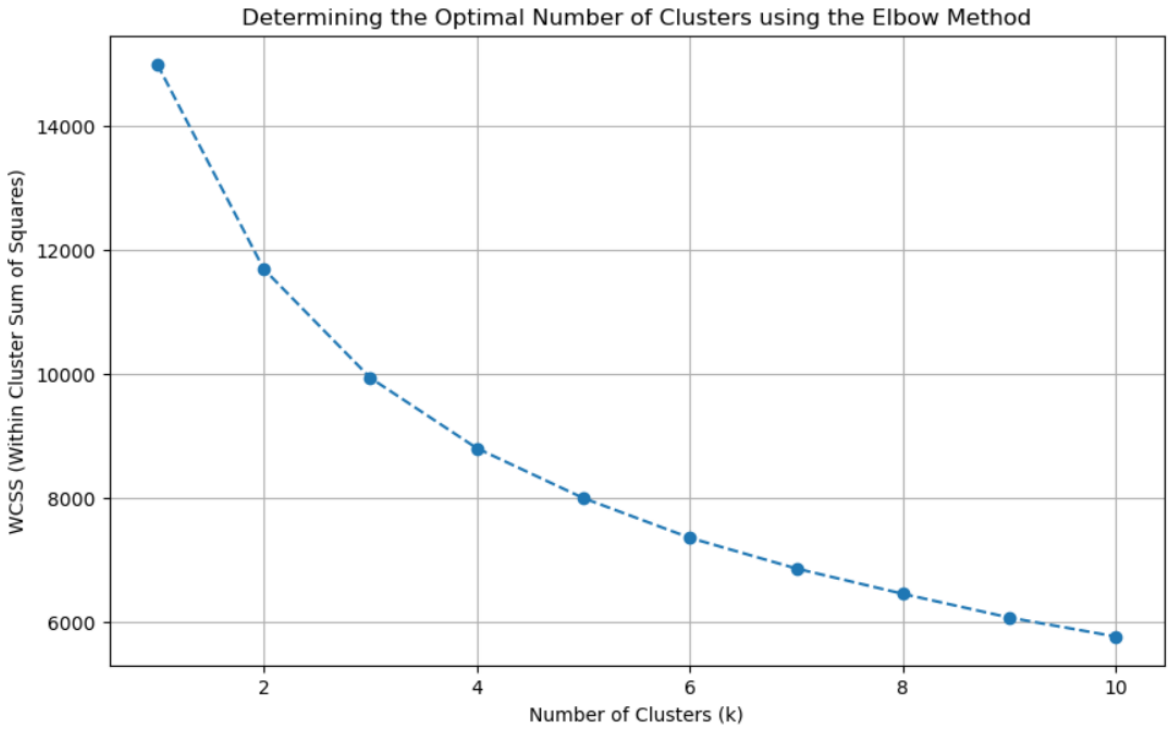
K-means Clustering Application:

The K-means algorithm was applied according to the determined number of clusters **k=4**. The model divided the data into 4 clusters, assigning each data point to the nearest average cluster center. These clustering results were added to the dataset as a **Cluster** column and included in the dataset.

```
Cluster
0    780
1    711
2    695
3    314
Name: count, dtype: int64
```

Visualization of Clustering Results with PCA:

In order to better analyze and visualize the clusters, the data was reduced to two dimensions using **PCA (Principal Component Analysis)**. Using these data reduced by PCA, the distribution of clusters was visually analyzed.



---

4. Hyperparameter Adjustment and Optimization

Hyperparameter Optimization:

Hyperparameter optimization has been performed to make the model performance better. For this purpose, various parameter combinations were tried using the **Randomized Search** method and the performance of the model was tried to be improved.

Hyperparameter Settings:

- **n_clusters**: Number of clusters
- **max_iter**: Maximum number of iterations
- **init**: How to determine the cluster centers (k-means++ was used)

```
param_dist = {
    'n_clusters': [3, 4, 5],
    'init': ['k-means++'],
    'max_iter': [200, 300],
    'n_init': [10]
}
```

The model was retrained with the most appropriate parameters obtained as a result of **Randomized Search**. The **Silhouette score** increased to **0.197** after optimization, which indicates that the model separates clusters more clearly.

```
The Best Parameters: {'n_init': 10, 'n_clusters': 3, 'max_iter': 200, 'init': 'k-means++'}
The Highest Silhouette Score: 0.20
```

---

5. Evaluation of Clustering Quality

Evaluation Metrics:

- **Silhouette Score**: It is a criterion used to measure clustering quality. The value was obtained as **0.197**, which means that the clusters show a reasonable decomposition
- **Davies-Bouldin Score**: The cluster measures the similarity between them. The score is **1.82**, which means that the intra-cluster similarities are low, and the clusters are more different from each other.
- **Calinski-Harabasz Score**: It is a metric that is required to be high, and it was obtained as **636.20**. Which shows that the clusters have a distinct decomposition.

---

6. Analysis of Cluster Properties

Distribution of Clusters According to Their Properties: In order to understand cluster analysis, the distribution of each cluster in certain characteristics was visualized using **boxplot**. Thanks to this, it became clear which weather conditions the clusters represent.

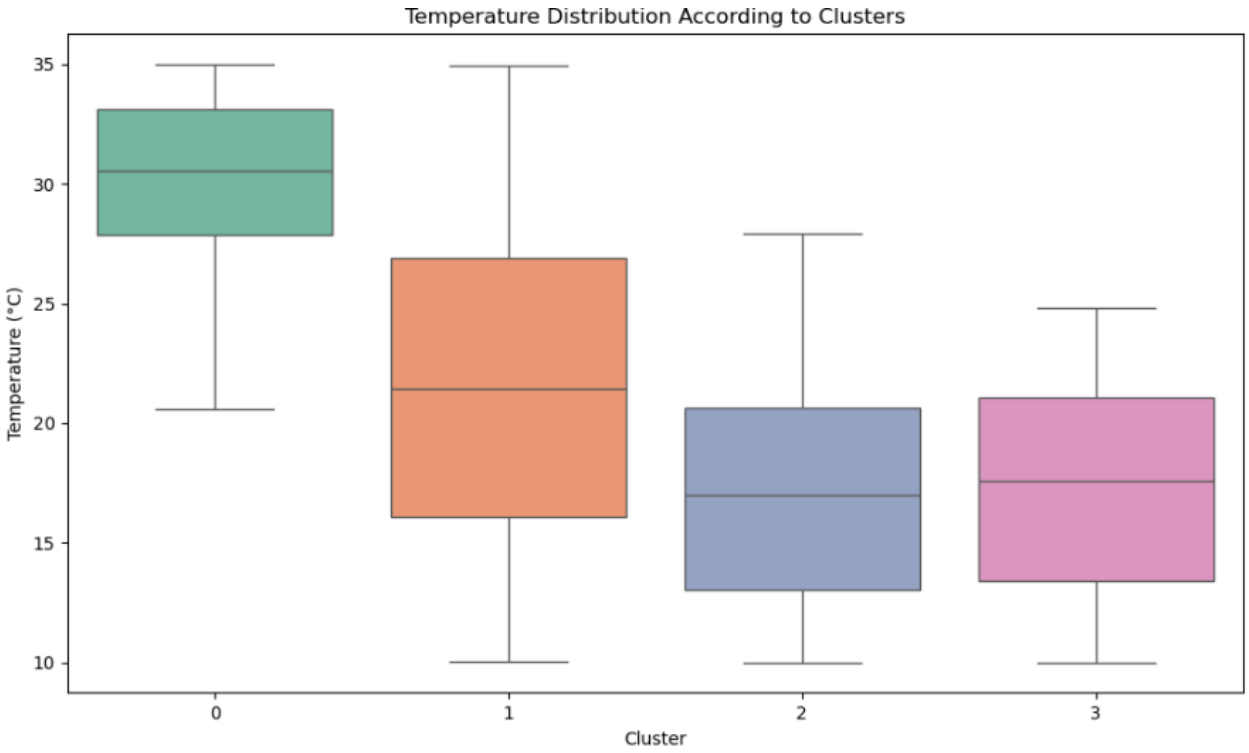
- **Temperature, Humidity, Wind Speed and Cloud Cover Distribution**: Boxplot graphics were created for each feature. These graphs show how each cluster differs in terms of these properties.

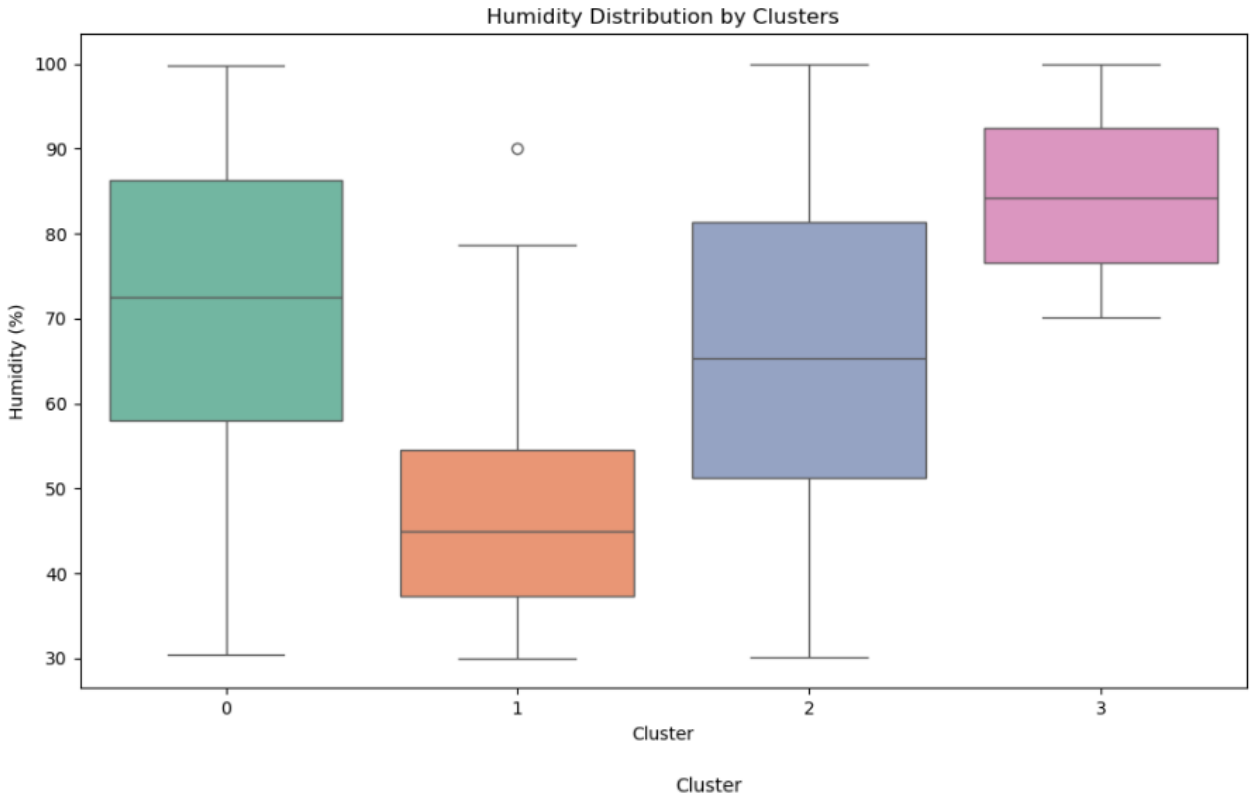
---

7. Comparison Before and After the Hyperparameter Setting

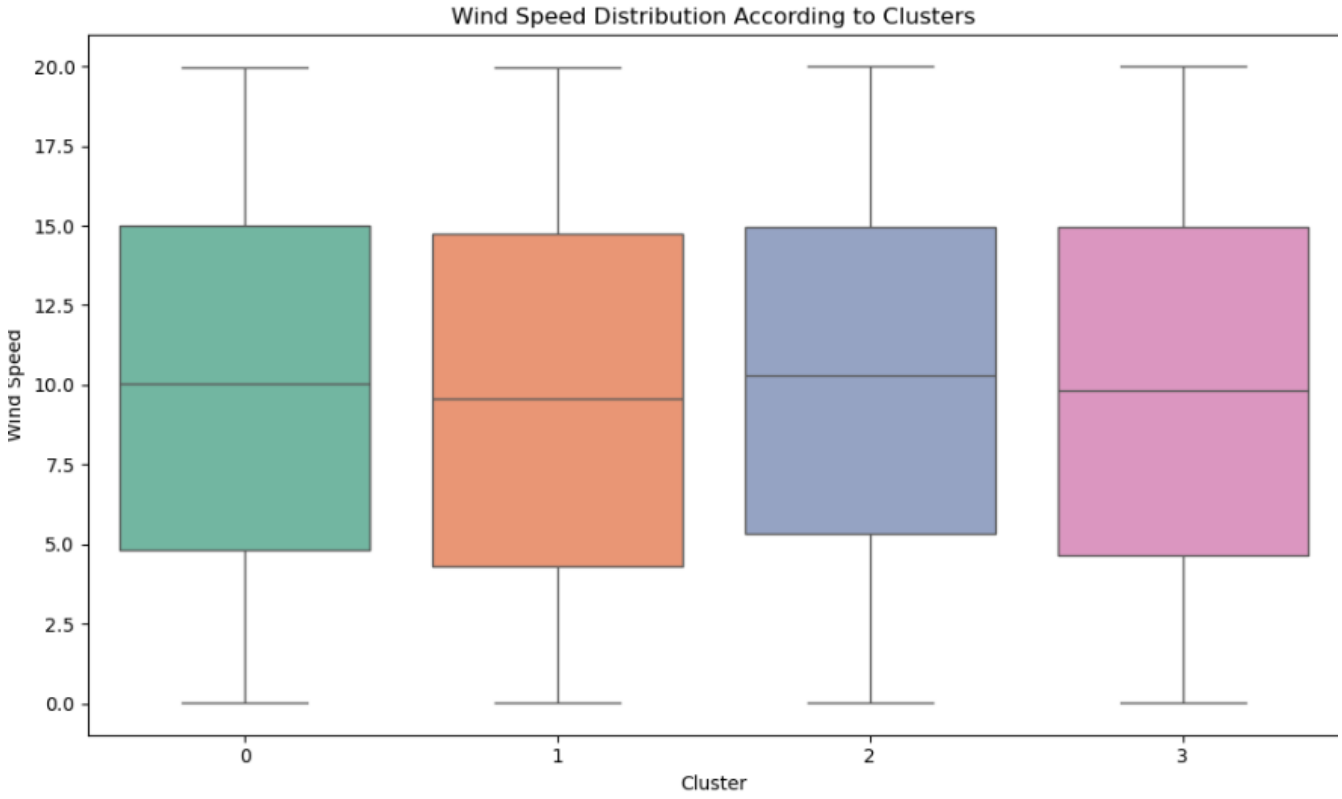
The performance metrics before the hyperparameter adjustment and after the optimization were compared. This comparison is important for understanding how hyperparameter adjustment affects model performance.

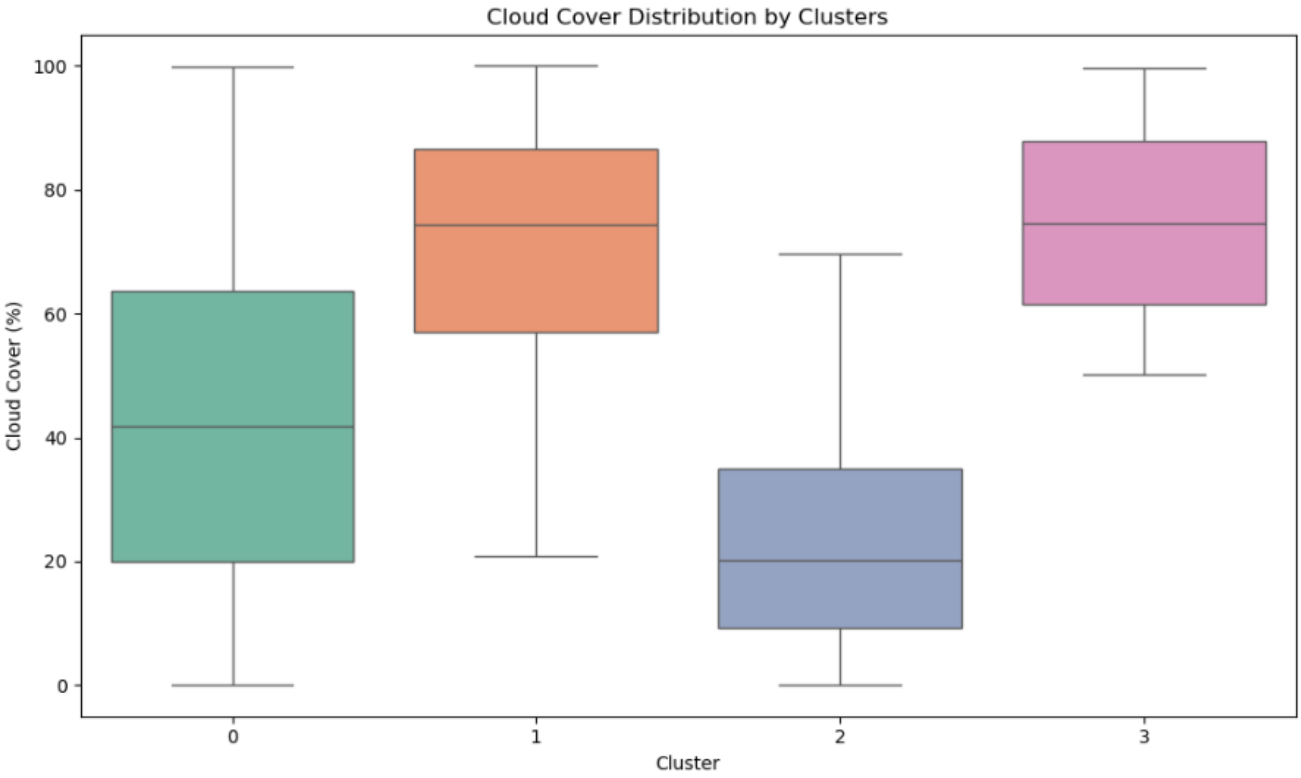
- **Silhouette Score Before and After**: While the initial score was **0.15**, it increased to **0.197** after optimization.
- **Davies-Bouldin Score Before and After**: Decreased from **2.10** to **1.82**.
- **Before and After the Calinski-Harabasz Score**: Increased from **570** to **636.20**.




Humidity Distribution by Clusters
Humidity (%)
100
90
80
70
60
50
40
30
0
1
2
3
Cluster

Cluster


Wind Speed Distribution According to Clusters
Wind Speed
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
1
2
3
Cluster



---

8. Results and Learnings

General Evaluation of the Project: In this project, the data were divided into groups using the **K-means** clustering algorithm using weather data and these clusters were analyzed. During the project:

- We have seen the importance of accurate data preprocessing (digitization and scaling of categorical data and December of features to appropriate ranges) on model performance. Correct data preprocessing is an essential step for the model to give meaningful results and for the data to be grouped appropriately.

The Use and Strengths of the K-means Clustering Algorithm: K-means clustering is an effective method for dividing data into groups with similar characteristics. In this study, we divided the weather data into four groups and examined how the weather conditions are grouped according to certain characteristics. Thanks to the K-means algorithm, the data were analyzed by assigning them to clusters that have obvious similarities.

The Importance of Hyperparameter Settings: We have experienced that model performance can be improved by seeing the effects of hyperparameter settings. After finding the optimal number of clusters and other parameters using Randomized Search, we observed that the model separated the data better. Such adjustments are very important for machine learning models to better adapt to the data and be able to generalize.

Inferences: At the end of the project, we observed that different weather conditions collected in clusters with similar properties, and these clusters decomposed in a meaningful way. For example, some clusters

represented days with higher temperatures and lower humidity, while others represented colder and wetter days. By performing hyperparameter optimization, we obtained better Decoupling clusters and were able to see the differences between these clusters more clearly.

In this project, I used the K-means clustering algorithm to group weather data into different clusters. The dataset included information about temperature, humidity, wind speed, cloud cover and rain. First, I processed the data by converting the "rain" column to numerical values and scaled the data. Then I applied the K-means algorithm to create 4 clusters. I used the " elbow method" to determine the best number of clusters.

After applying the K-means algorithm, I visualized the results using PCA. I also optimized the hyperparameters, which improved the model performance. After adjusting the parameters, I saw that the silhouette score increased and showed that my model did a better job of separating clusters.

I learned how to prepare data, how k-means clusters work, and how important it is to adjust hyperparameters for better results.