Question 1 Report: Air Quality Prediction and Model Comparison

- **First and Last Name**: Fahrettin Solak
- **School Number**: 201401053
- **The Subject of the Problem**: Air Quality Prediction with Decision Tree and Comparison with Different Models
- **The Data Set and Link Used**:
  - **Data Set Source**: Kaggle
  - **Link**: Air Quality and Pollution Assessment
- **Methods Used**:
    - Decision Tree,
    - Random Forest,
    - Support Vector Machine (SVM),
    - Logistic Regression,
    - Data Preprocessing,
    - Hyperparameter Adjustment,
    - Ensemble Learning (Voting Classifier)

1. Introduction

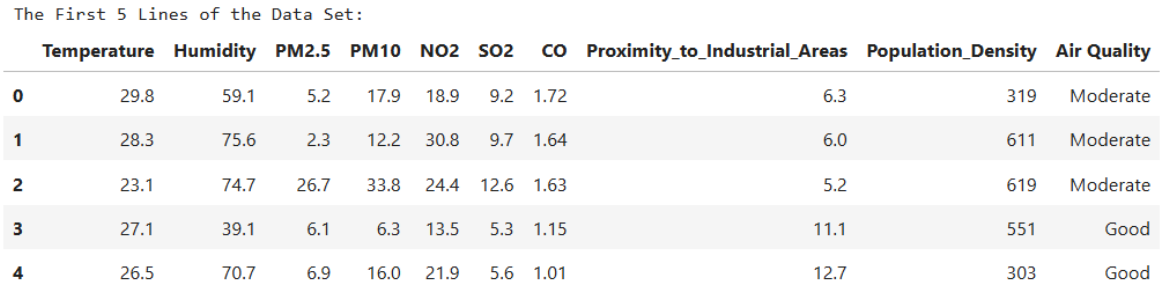
Project Objective: The aim of this project is to classify air quality using various environmental factors. Different models of machine learning have been applied and compared to improve the accuracy of the classification process. Predicting air quality plays an important role in protecting environmental health, reducing pollution and improving people's quality of life. In this study, a decision tree model for air quality classification was first used. Its performance was then compared with other popular classifiers such as Random Forest, Logistic Regression and Support Vector Machine. In addition, a community learning method was used by creating a voting classifier model that combines the strengths of different models to achieve better results

Data Mining and its Importance: Data Mining extracts valuable information from large data sets by identifying patterns, relationships, and trends. It helps to transform complex data into meaningful information and allows for accurate forecasts and data-based decisions. In environmental research, data mining plays a crucial role in analyzing environmental factors to address critical issues such as air pollution. In this project, data mining techniques were applied to study air quality. By studying various environmental factors, we have studied how these elements affect air quality and how they can be classified to support better environmental management and pollution control.

---

2. Data Set and Selection

The "Air Quality and Pollution Assessment" dataset published on the Kaggle platform was used in the project. This data set is designed to assess air quality and pollution level based on various environmental parameters. The dataset has 5000 samples and 10 features and contains important parameters related to air pollution. The first 5 lines of the dataset are as follows:

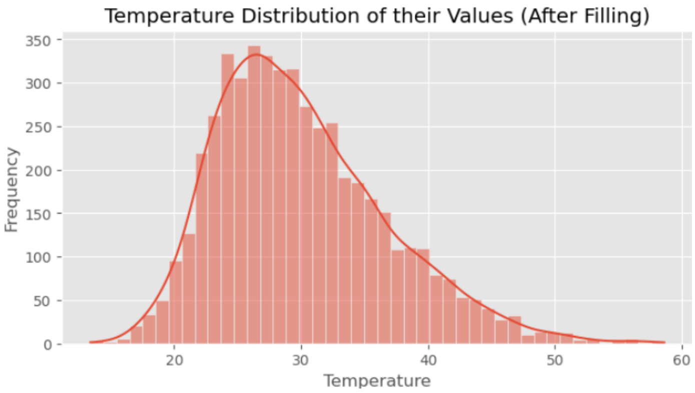


```
The First 5 Lines of the Data Set:
```

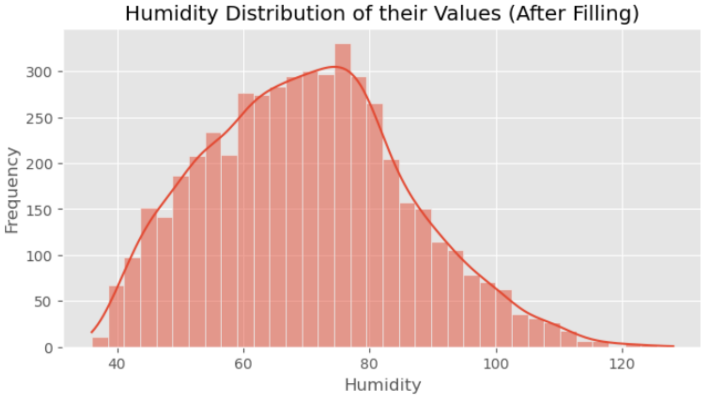
	Temperature	Humidity	PM2.5	PM10	NO2	SO2	CO	Proximity_to_Industrial_Areas	Population_Density	Air Quality
0	29.8	59.1	5.2	17.9	18.9	9.2	1.72	6.3	319	Moderate
1	28.3	75.6	2.3	12.2	30.8	9.7	1.64	6.0	611	Moderate
2	23.1	74.7	26.7	33.8	24.4	12.6	1.63	5.2	619	Moderate
3	27.1	39.1	6.1	6.3	13.5	5.3	1.15	11.1	551	Good
4	26.5	70.7	6.9	16.0	21.9	5.6	1.01	12.7	303	Good

Features Found in the Data Set:

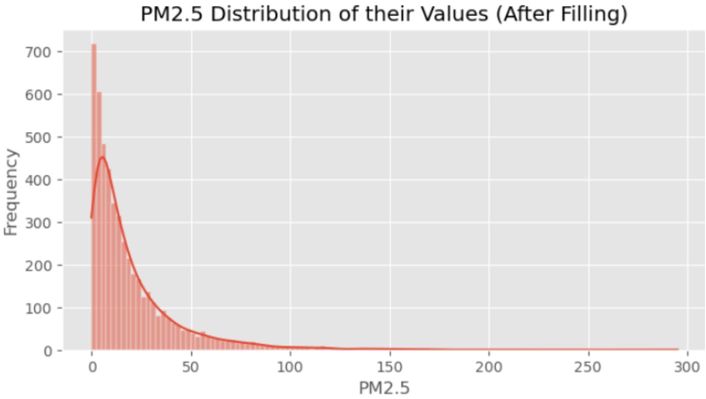
1. **Temperature**: The value of the temperature in the air (in Celsius).



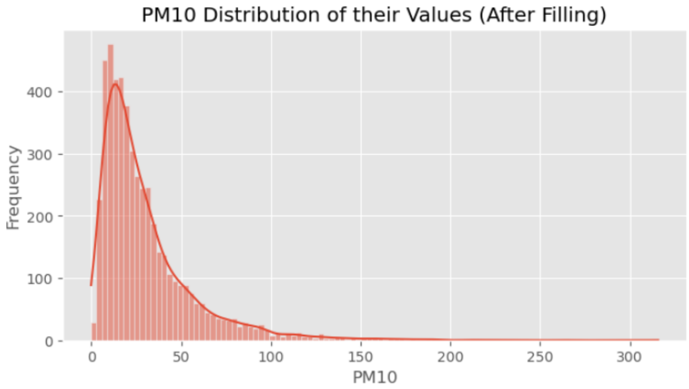
2. **Humidity**: The humidity in the air (%).



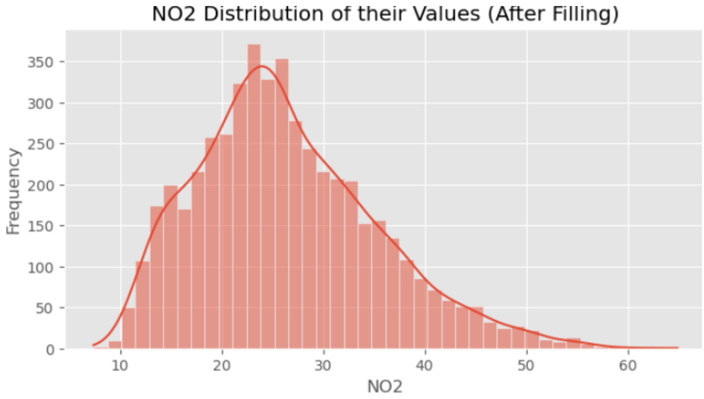
3. **PM2.5**: The amount of particulate matter smaller than 2.5 micrometers.



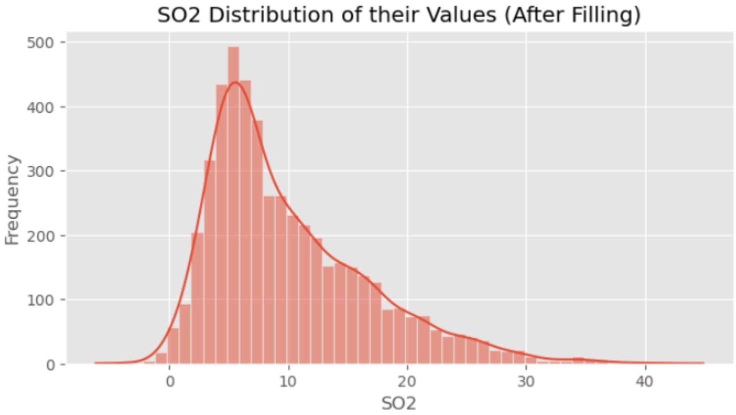
4. **PM10**: The amount of particulate matter smaller than 10 micrometers.



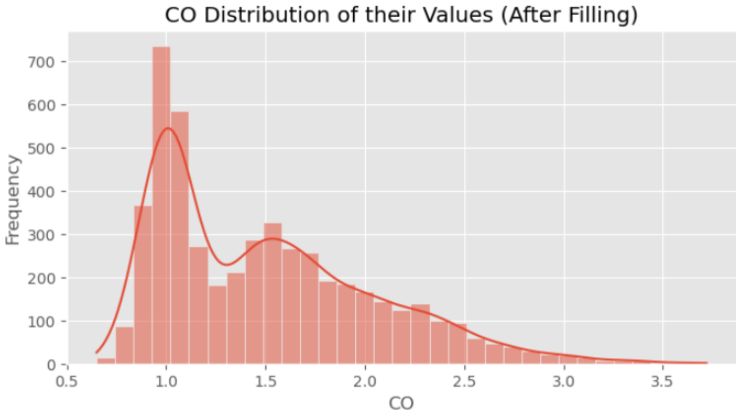
5. **NO2 (Nitrogen Dioxide)**: The concentration of nitrogen dioxide gas in the air.



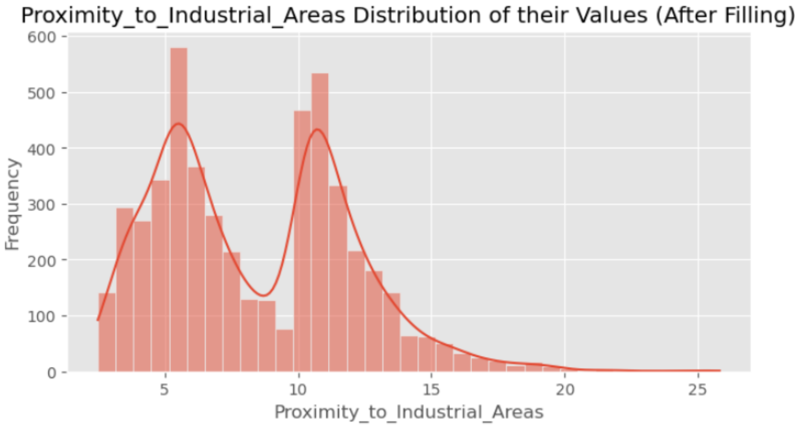
6. **SO2 (Sulfur Dioxide)**: The concentration of sulfur dioxide gas in the air.



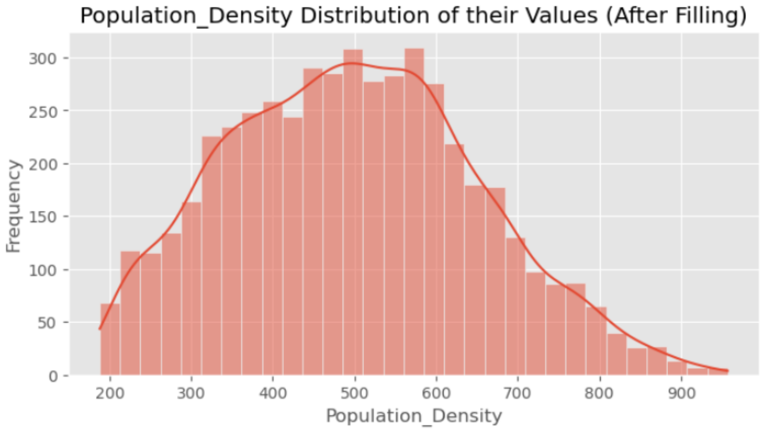
7. **CO (Carbon Monoxide)**: The concentration of carbon monoxide gas in the air.



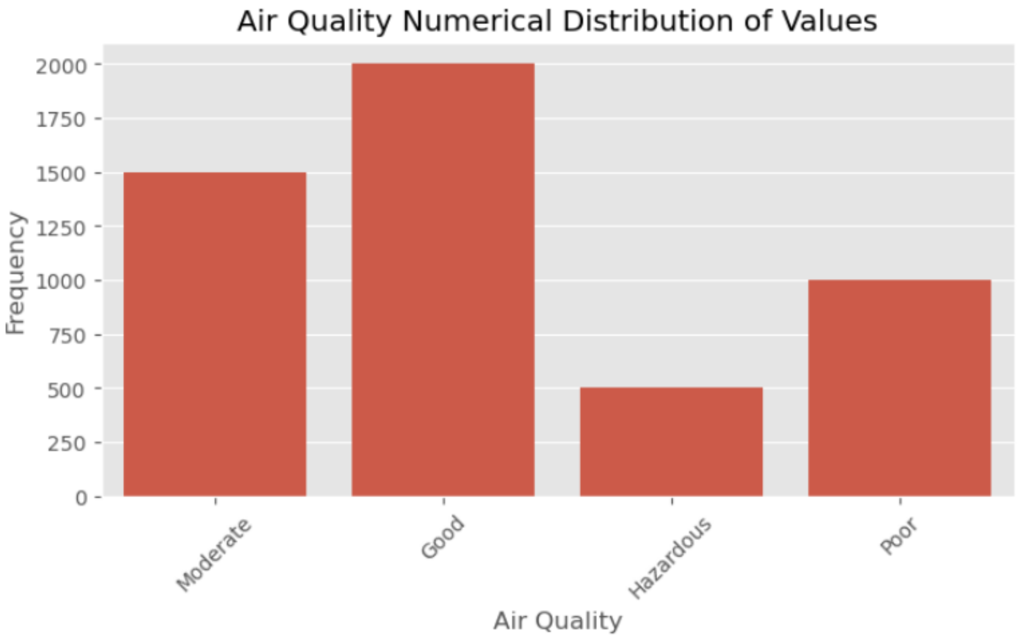
8. **Proximity_to_Industrial_Areas:** The distance to the industrial areas of the region (in km).



9. **Population_Density**: The density of people in the area.

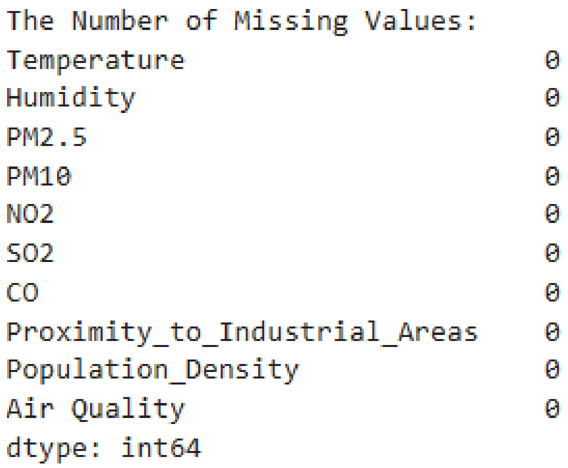


10. **Air Quality**: The target variable indicating air quality. This variable is divided into four classes: "Good", "Moderate", "Poor" and "Hazardous".



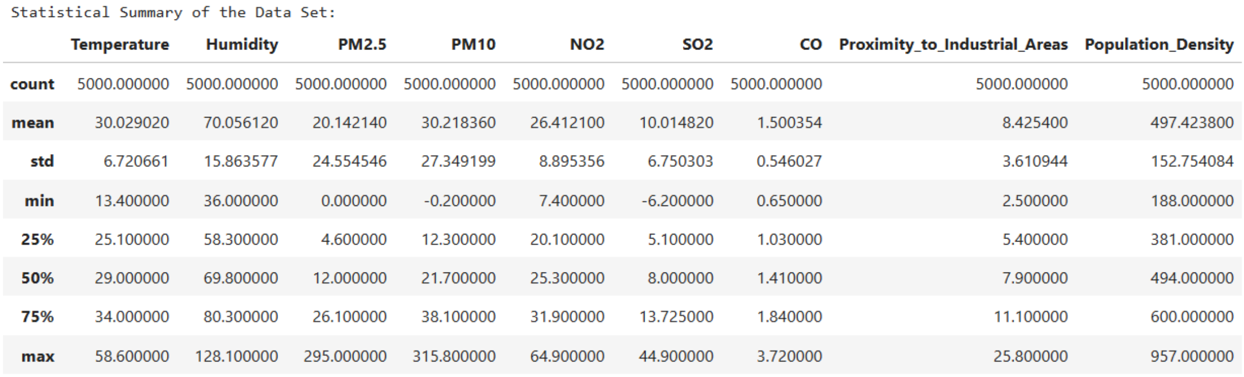
General Review of the Data Set:

- **Missing Value Analysis:** It was checked whether there were missing values in the data set, and it was found that there were no missing values. This means that the model can be trained without the need for additional steps such as data cleaning or filling in missing data. Oct.



```
The Number of Missing Values:
Temperature                      0
Humidity                         0
PM2.5                            0
PM10                             0
NO2                              0
SO2                              0
CO                               0
Proximity_to_Industrial_Areas    0
Population_Density               0
Air Quality                      0
dtype: int64
```

- **Statistical Information:** Statistical information such as minimum, maximum, average and standard deviation have been extracted for each feature in the data set. This information helped us to understand the overall structure of the data and revealed that some variables may have excessive values. It means that it can be trained.

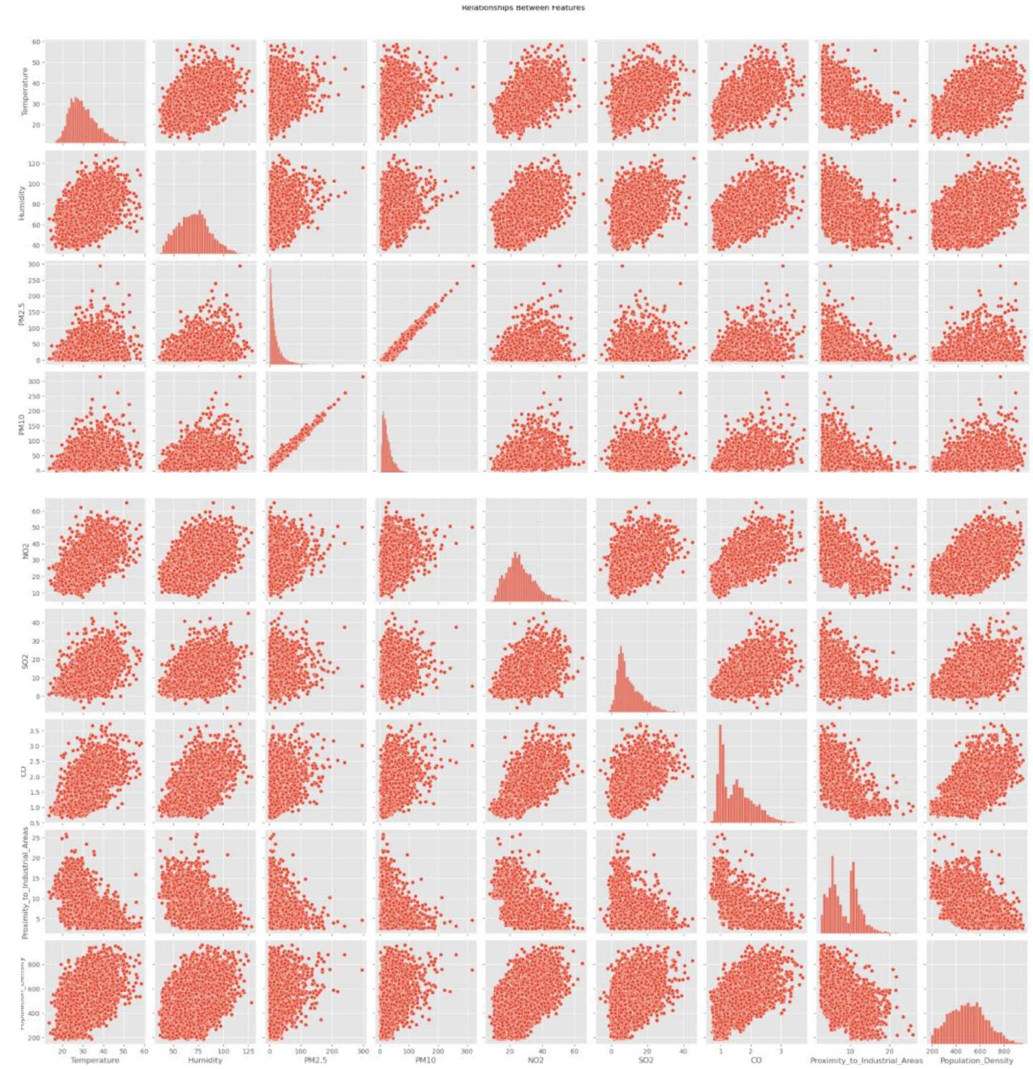


```
Statistical Summary of the Data Set:
```

	Temperature	Humidity	PM2.5	PM10	NO2	SO2	CO	Proximity_to_Industrial_Areas	Population_Density
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	30.029020	70.056120	20.142140	30.218360	26.412100	10.014820	1.500354	8.425400	497.423800
std	6.720661	15.863577	24.554546	27.349199	8.895356	6.750303	0.546027	3.610944	152.754084
min	13.400000	36.000000	0.000000	-0.200000	7.400000	-6.200000	0.650000	2.500000	188.000000
25%	25.100000	58.300000	4.600000	12.300000	20.100000	5.100000	1.030000	5.400000	381.000000
50%	29.000000	69.800000	12.000000	21.700000	25.300000	8.000000	1.410000	7.900000	494.000000
75%	34.000000	80.300000	26.100000	38.100000	31.900000	13.725000	1.840000	11.100000	600.000000
max	58.600000	128.100000	295.000000	315.800000	64.900000	44.900000	3.720000	25.800000	957.000000

Relationships Between Variables and Visualization:

- **Pair Plot Graph:** A pair plot graph was created in order to better understand the relationship of properties to each other. In particular, the relationships of variables such as PM2.5, NO2 and CO with other factors were observed thanks to this graph.

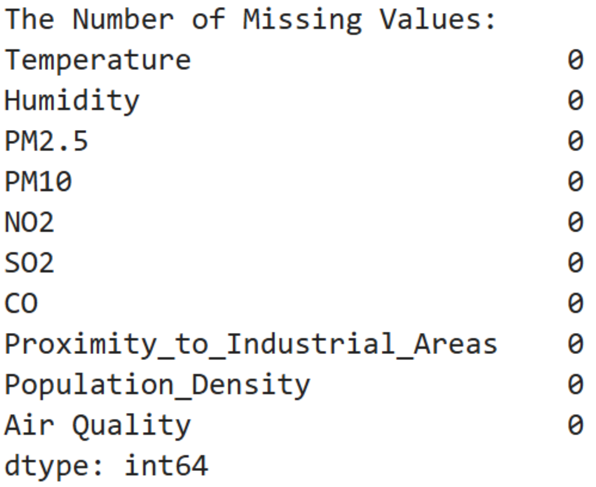


3. Data Pre-Processing

Data preprocessing is a very important stage in the machine learning process and directly affects model performance. During data preprocessing in this project, the following steps were applied:

3.1 Filling in the Missing Values:

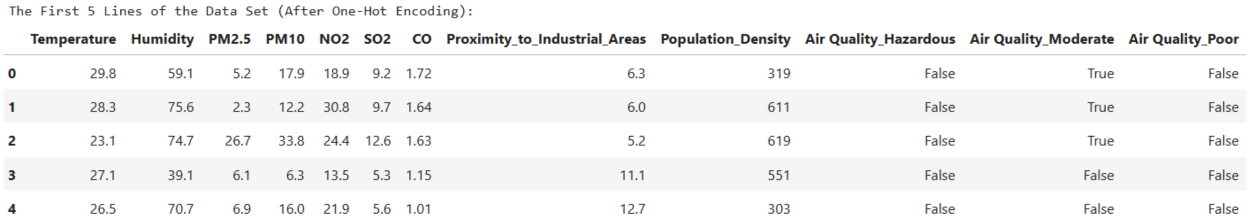
- It was checked whether there are missing values in the data set. The absence of missing values provided a great advantage for the model, because at this stage there was no need to fill in missing values in order to prevent any data loss or misleading data formation.



```
The Number of Missing Values:
Temperature                      0
Humidity                         0
PM2.5                            0
PM10                             0
NO2                              0
SO2                              0
CO                               0
Proximity_to_Industrial_Areas    0
Population_Density               0
Air Quality                      0
dtype: int64
```

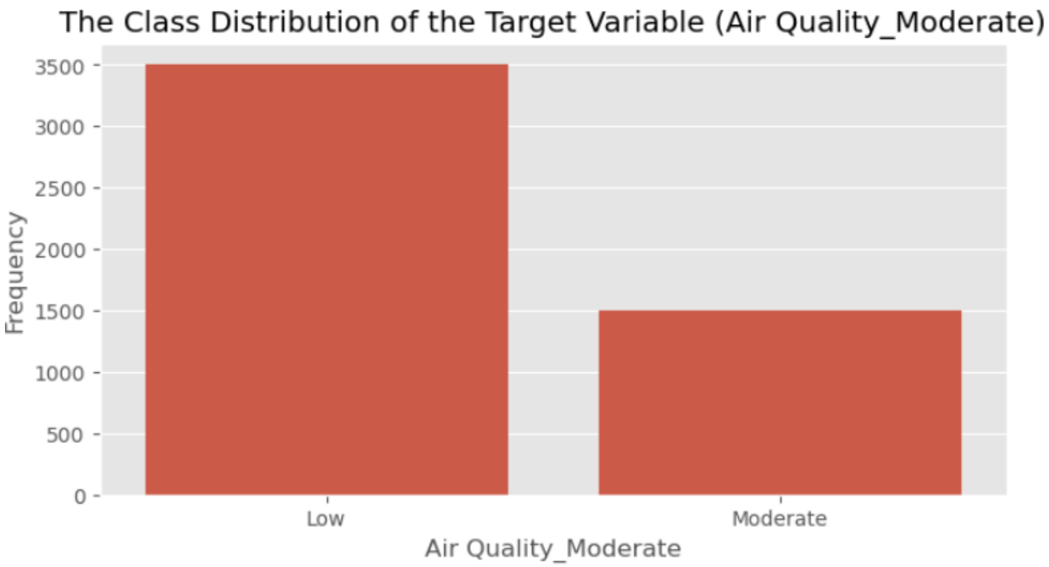
3.2 Digitization of Categorical Variables (One-Hot Encoding):

- Categorical variables such as **Air Quality** had to be numericized in order to be processed by machine learning algorithms. For this reason, using One-Hot Encoding, each category was expressed with a value of 0 or 1.



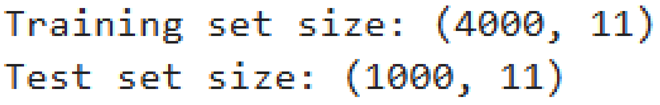
```
The First 5 Lines of the Data Set (After One-Hot Encoding):
```

	Temperature	Humidity	PM2.5	PM10	NO2	SO2	CO	Proximity_to_Industrial_Areas	Population_Density	Air Quality_Hazardous	Air Quality_Moderate	Air Quality_Poor
0	29.8	59.1	5.2	17.9	18.9	9.2	1.72	6.3	319	False	True	False
1	28.3	75.6	2.3	12.2	30.8	9.7	1.64	6.0	611	False	True	False
2	23.1	74.7	26.7	33.8	24.4	12.6	1.63	5.2	619	False	True	False
3	27.1	39.1	6.1	6.3	13.5	5.3	1.15	11.1	551	False	False	False
4	26.5	70.7	6.9	16.0	21.9	5.6	1.01	12.7	303	False	False	False

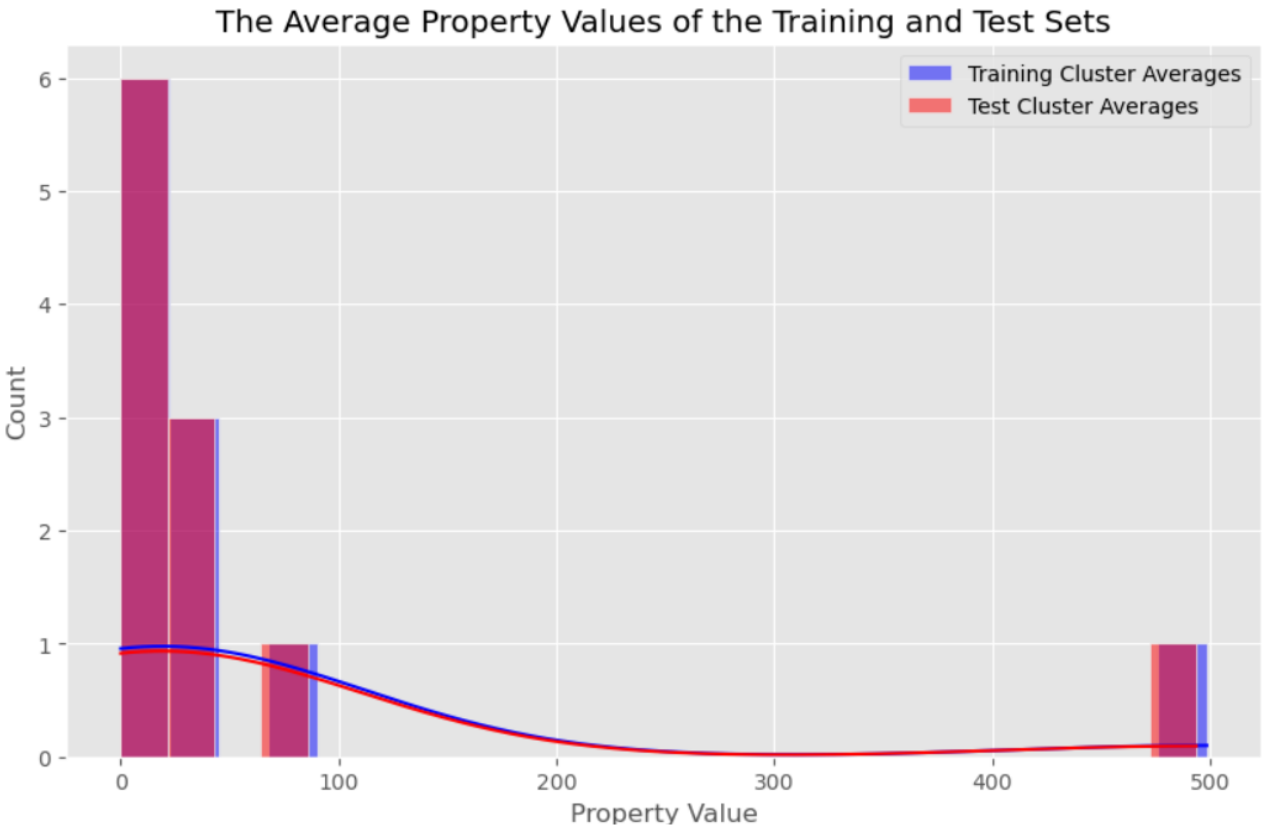


3.3 Dividing the Data Set into Training and Testing:

- The data was divided into 80% training and 20% testing. This ratio allowed us to test the overall performance of the model and its accuracy on the new data. While the training set was used for the model to learn the data, the test set was used to measure the performance of the model.



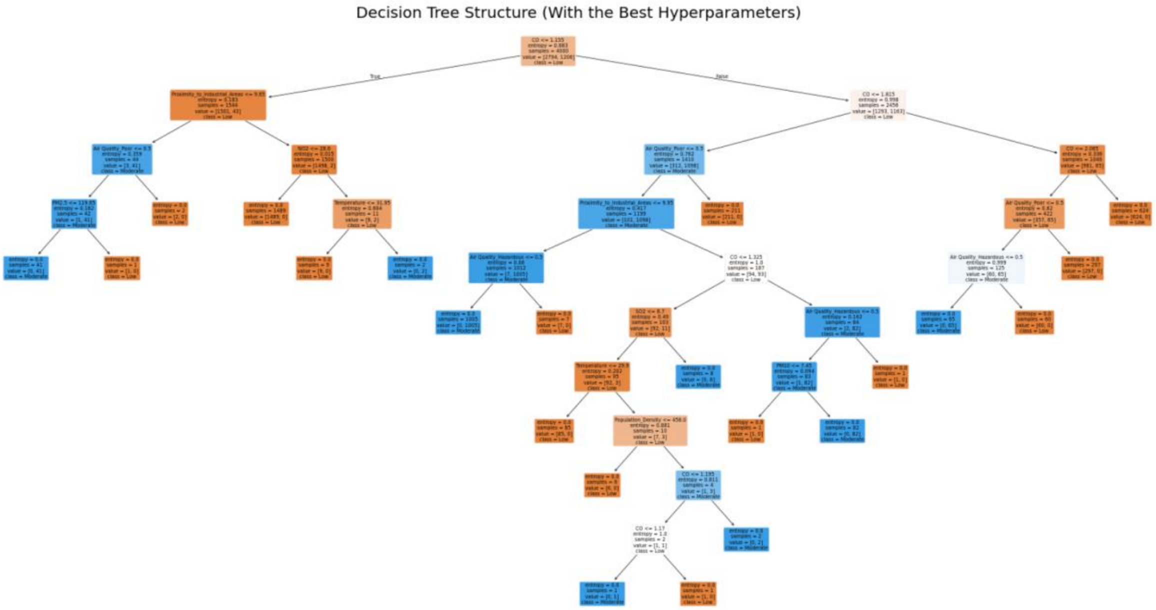
```
Training set size: (4000, 11)
Test set size: (1000, 11)
```



4. Model Installation and Training Stages

4.1 The Decision Tree Model:

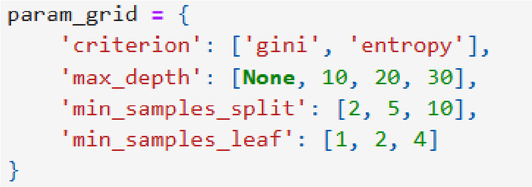
- **Model Setup**: The Decision Tree is a highly explainable model that classifies data by dividing it into branches. The Decision Tree continues the prediction process by making a division according to a certain property at each node.



Decision Tree Structure (With the Best Hyperparameters)

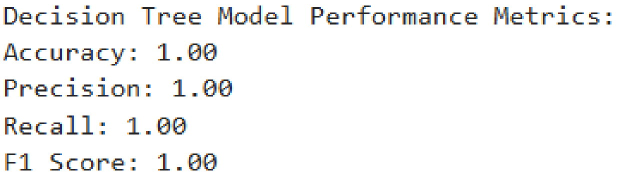
CO <= 1.155 entropy = 0.883 samples = 4000 value = [2794, 1206] class = Low
True
False
Proximity_to_Industrial_Areas <= 9.85 entropy = 0.183 samples = 1544 value = [1501, 43] class = Low
CO <= 1.815 entropy = 0.998 samples = 2456 value = [1293, 1163] class = Low
Air_Quality_Poor <= 0.5 entropy = 0.359 samples = 44 value = [3, 41] class = Moderate
NO2 <= 28.6 entropy = 0.015 samples = 1500 value = [1498, 2] class = Low
Air_Quality_Poor <= 0.5 entropy = 0.762 samples = 1410 value = [312, 1098] class = Moderate
CO <= 2.085 entropy = 0.338 samples = 1046 value = [981, 65] class = Low
PM2.5 <= 119.85 entropy = 0.162 samples = 42 value = [1, 41] class = Moderate
entropy = 0.0 samples = 2 value = [2, 0] class = Low
entropy = 0.0 samples = 1489 value = [1489, 0] class = Low
Temperature <= 31.95 entropy = 0.684 samples = 11 value = [9, 2] class = Low
Proximity_to_Industrial_Areas <= 9.95 entropy = 0.417 samples = 1199 value = [101, 1098] class = Moderate
entropy = 0.0 samples = 211 value = [211, 0] class = Low
Air_Quality_Poor <= 0.5 entropy = 0.82 samples = 422 value = [357, 65] class = Low
entropy = 0.0 samples = 624 value = [624, 0] class = Low
entropy = 0.0 samples = 41 value = [0, 41] class = Moderate
entropy = 0.0 samples = 1 value = [1, 0] class = Low
entropy = 0.0 samples = 9 value = [9, 0] class = Low
entropy = 0.0 samples = 2 value = [0, 2] class = Moderate
Air_Quality_Hazardous <= 0.5 entropy = 0.06 samples = 1012 value = [7, 1005] class = Moderate
CO <= 1.325 entropy = 1.0 samples = 187 value = [94, 93] class = Low
Air_Quality_Hazardous <= 0.5 entropy = 0.999 samples = 125 value = [60, 65] class = Moderate
entropy = 0.0 samples = 297 value = [297, 0] class = Low
entropy = 0.0 samples = 1005 value = [0, 1005] class = Moderate
entropy = 0.0 samples = 7 value = [7, 0] class = Low
SO2 <= 8.7 entropy = 0.49 samples = 103 value = [92, 11] class = Low
Air_Quality_Hazardous <= 0.5 entropy = 0.162 samples = 84 value = [2, 82] class = Moderate
entropy = 0.0 samples = 65 value = [0, 65] class = Moderate
entropy = 0.0 samples = 60 value = [60, 0] class = Low
Temperature <= 29.9 entropy = 0.202 samples = 95 value = [92, 3] class = Low
entropy = 0.0 samples = 8 value = [0, 8] class = Moderate
PM10 <= 7.45 entropy = 0.094 samples = 83 value = [1, 82] class = Moderate
entropy = 0.0 samples = 1 value = [1, 0] class = Low
entropy = 0.0 samples = 85 value = [85, 0] class = Low
Population_Density <= 458.0 entropy = 0.881 samples = 10 value = [7, 3] class = Low
entropy = 0.0 samples = 1 value = [1, 0] class = Low
entropy = 0.0 samples = 82 value = [0, 82] class = Moderate
entropy = 0.0 samples = 6 value = [6, 0] class = Low
CO <= 1.195 entropy = 0.811 samples = 4 value = [1, 3] class = Moderate
CO <= 1.17 entropy = 1.0 samples = 2 value = [1, 1] class = Low
entropy = 0.0 samples = 2 value = [0, 2] class = Moderate
entropy = 0.0 samples = 1 value = [0, 1] class = Moderate
entropy = 0.0 samples = 1 value = [1, 0] class = Low

- **Hyperparameter Setting**: Using GridSearchCV, parameters such as **criterion**, **max_depth**, **min_samples_split** and **min_samples_leaf** are set to find the most appropriate values.
    - **The Best Parameters**: Criterion = 'entropy', Max Depth = None, Min Samples Split = 2, Min Samples Leaf = 1.



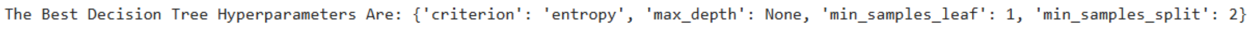
```
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

- **Evaluation of Model Performance**: The performance of the Decision Tree model was evaluated using metrics such as accuracy, precision, recall, F1 score.

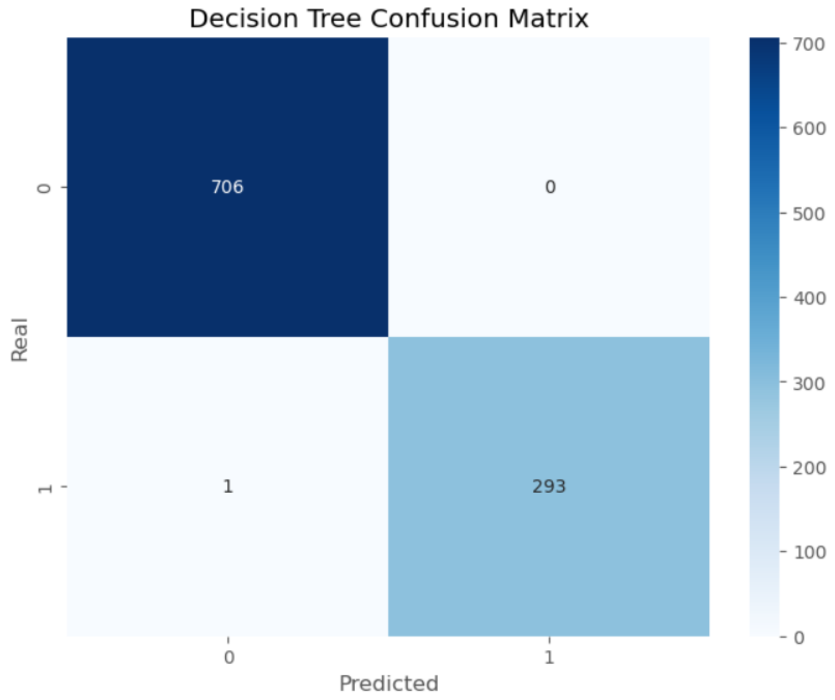


```
Decision Tree Model Performance Metrics:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1 Score: 1.00
```

The best values:

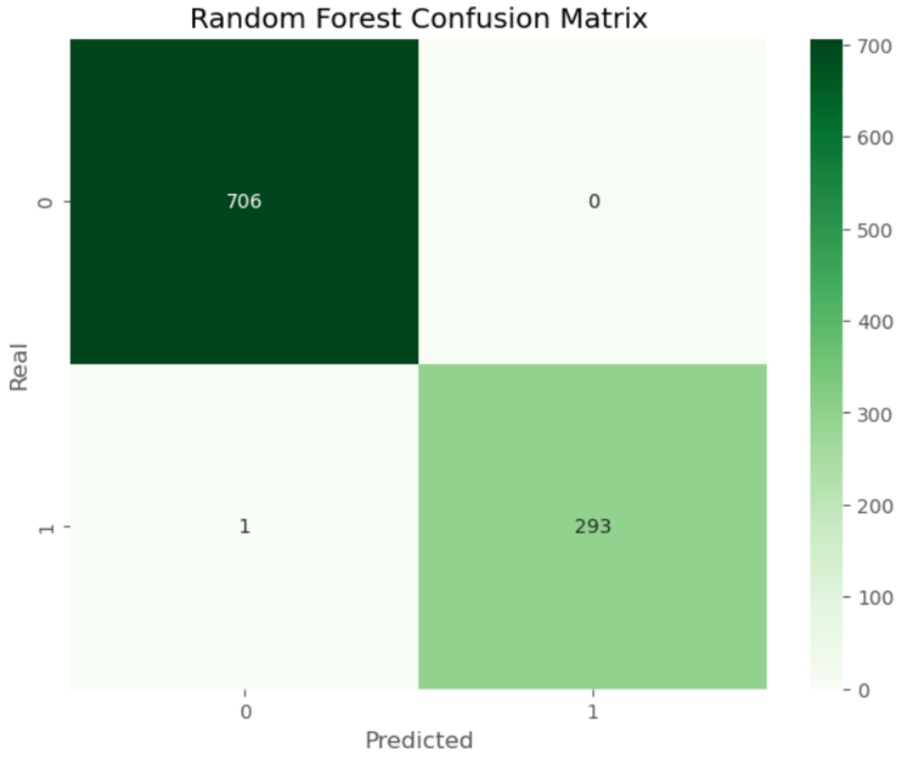


```
The Best Decision Tree Hyperparameters Are: {'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
```

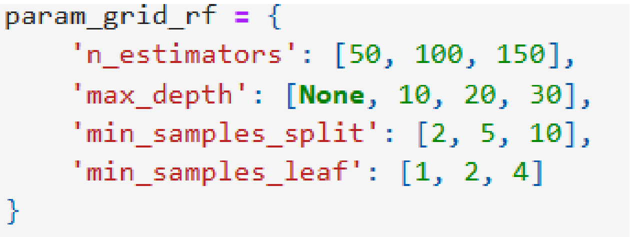


4.2 Random Forest Model:

- **Model Setup**: Random Forest combines many Decision Trees to create a model that is more balanced and has a high generalization ability. Dec. In this way, it increases the overall accuracy by compensating for the weaknesses of different trees.

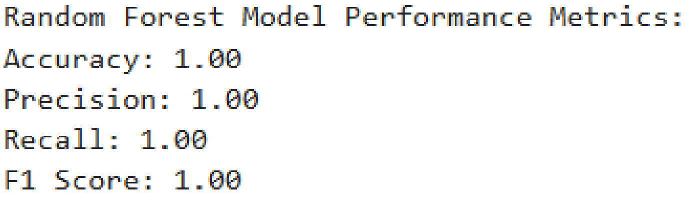


- **Hyperparameter Setting**: Using GridSearchCV, the optimal combination of parameters such as **n_estimators**, **max_depth**, **min_samples_split**, **min_samples_leaf** was determined.
    - **The Best Parameters**: n_estimators = 50, Max Depth = None, Min Samples Split = 10, Min Samples Leaf = 1.



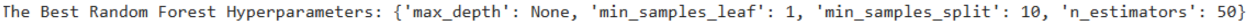
```
param_grid_rf = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20, 30],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

- **Evaluation of Model Performance**: The Random Forest model has shown a more balanced and better generalization performance compared to the Decision Tree.



```
Random Forest Model Performance Metrics:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1 Score: 1.00
```

The best values:



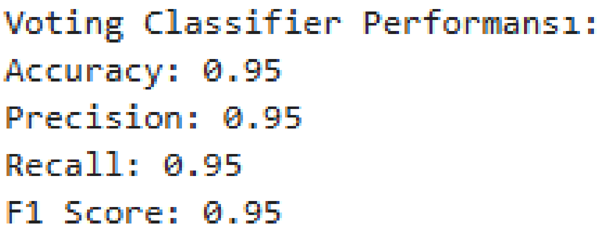
```
The Best Random Forest Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 50}
```

4.3 Other Classifiers:

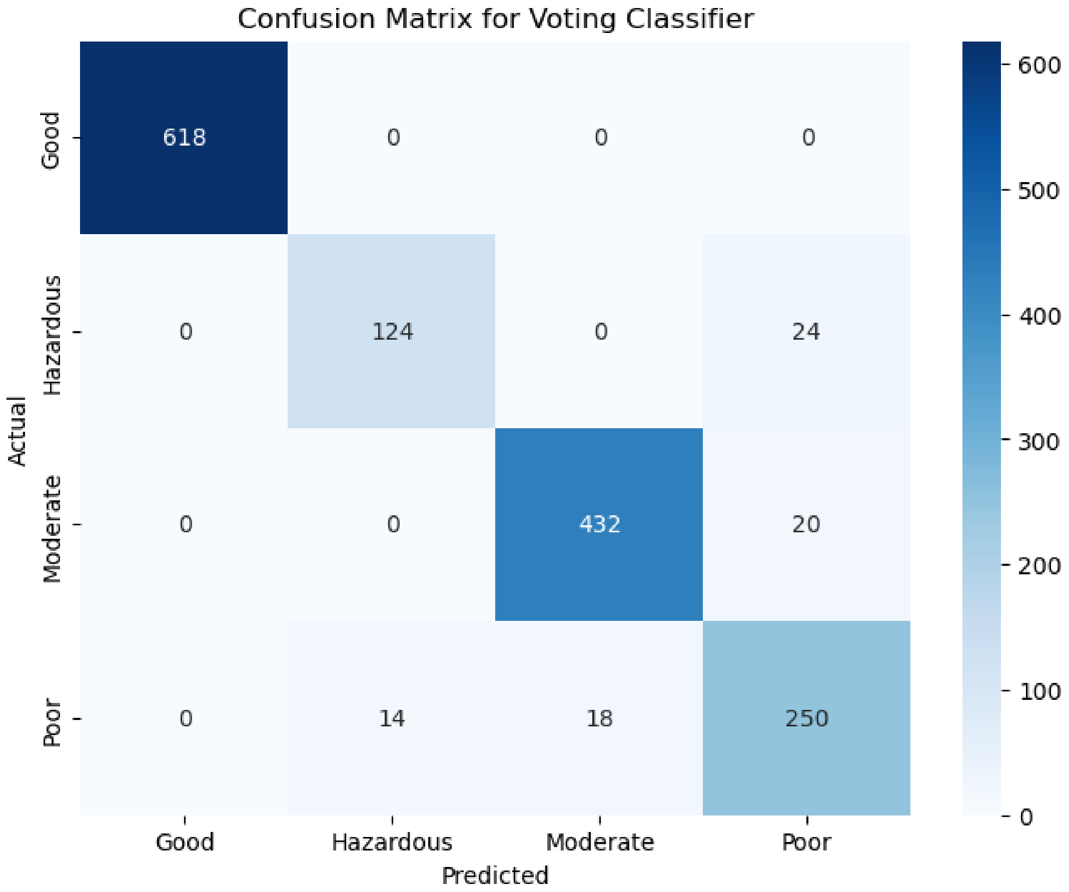
- **Logistic Regression and SVM**: Both of these models were trained on the data set and their performance was evaluated. Logistic Regression was successful in determining linear boundaries between classes, while SVM was effective in learning more complex classroom boundaries.

4.4 Community Learning with Voting Classifier:

- **Ensemble Learning**: Voting Classifier has created a more balanced and powerful prediction model by combining Decisional Tree, Random Forest, Logistic Regression and SVM models. This method provides higher accuracy and balance by combining the strengths of the models.



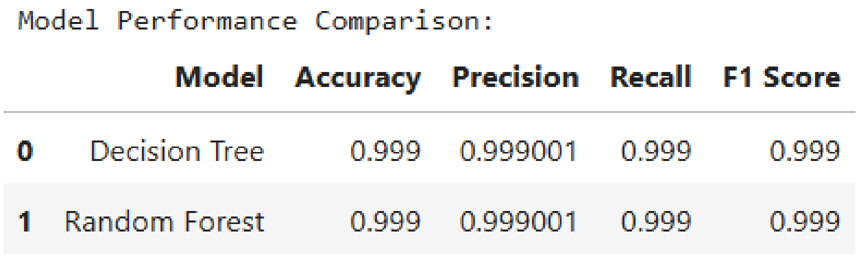
```
Voting Classifier Performansı:
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
F1 Score: 0.95
```



5. Model Performance Evaluation and Comparison

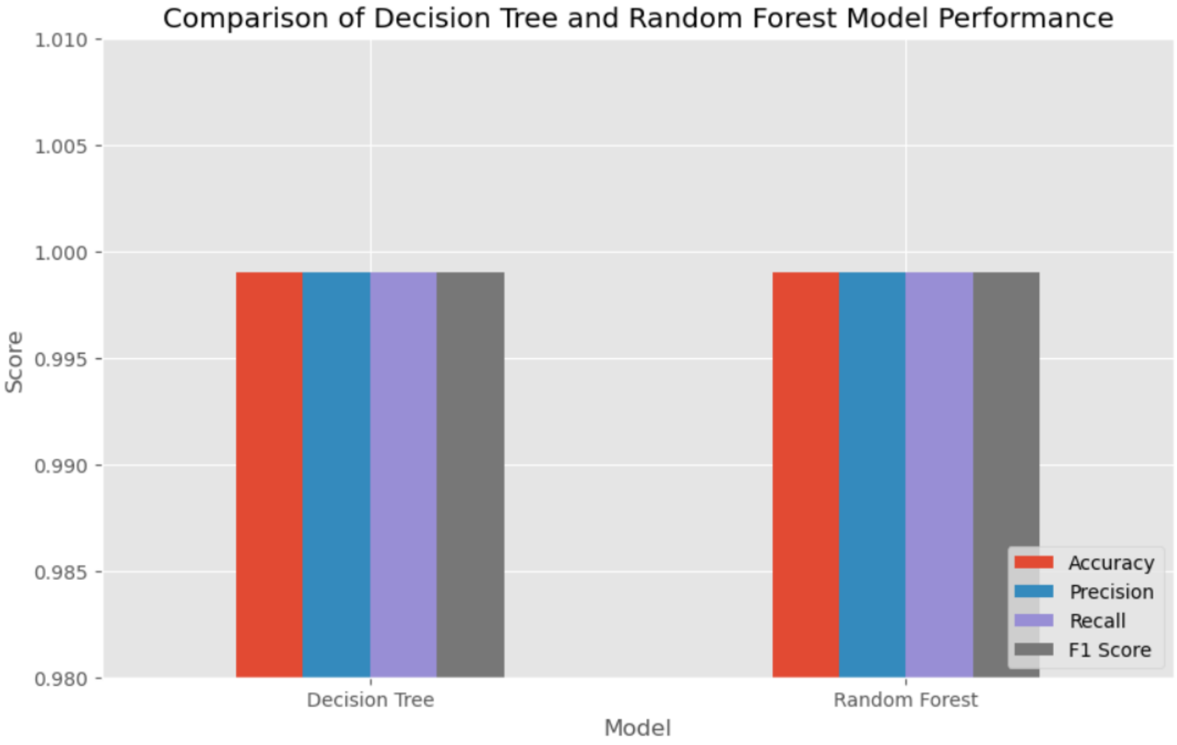
Model Comparison:

- We analyzed the performance of each model with metrics such as accuracy, accuracy, recall and F1 score. As a result of the comparison:
  - **Decision Tree**: It provided high accuracy, but excessive compliance problems were observed in some classes.
  - **Random Forest:** It gave results that were more balanced and had a high generalization ability.
  - **SVM and Logistic Regression**: It was more effective in determining linear boundaries.
  - **Voting Classifier**: It has achieved the highest accuracy by combining the strengths of all models.



```
Model Performance Comparison:
```

	Model	Accuracy	Precision	Recall	F1 Score
0	Decision Tree	0.999	0.999001	0.999	0.999
1	Random Forest	0.999	0.999001	0.999	0.999



6. Hyperparameter Settings and Results

Hyperparameter optimization is a critical step to ensure that machine learning models work in a more balanced and effective way. In this project, performance improvement was achieved by optimizing the hyperparameters of **Decision Tree** and **Random Forest** models in particular.

Stages of Hyperparameter Settings:

6.1 Determination of Hyperparameters for Decision Tree and Random Forest:

- Parameters such as **max_depth**, **min_samples_split**, **n_estimators** have been determined for **Decision Tree** and **Random Forest** models and the most appropriate combinations of these parameters have been determined using GridSearchCV.
- **The Best Parameters For the Decision Tree**: Criterion = 'entropy', Max Depth = None, Min Samples Split = 2, Min Samples Leaf = 1.

- **The Best Parameters For Random Forest**: n_estimators = 50, Max Depth = None, Min Samples Split = 10, Min Samples Leaf = 1.

6.2 Finding the Best Hyperparameters Using RandomizedSearchCV:

- **RandomizedSearchCV**, provided a faster optimization compared to GridSearchCV, as it tried a certain number of random hyperparameter combinations. In this way, the best results were found quickly without trying too many parameter combinations.

6.3 Performance Improvements and Comparisons:

- After the hyperparameter optimization, significant increases were observed in metrics such as accuracy, precision, recall and F1 score of each model.
    - Especially as a result of the hyperparameter optimization performed in the **Random Forest** model, there was a significant improvement in recall and F1 score. This means that the model is able to recognize especially positive classes more accurately.
- As a result of performance comparison, it has been seen that optimization increases the generalization ability of the model and better results are obtained on the test data.

---

7. Results and Learnings

In this project, various classification operations were performed using different machine learning models in order to classify air quality. The training process, performance evaluation and hyperparameter optimization of each model were performed. Below are the important points that we have extracted from this project:

7.1 The Importance of Data Pre-Processing:

- It was seen once again during this project that the data preprocessing step directly affects the success of the model in data mining projects. Filling in the missing values, digitizing the categorical variables and correctly dividing the data as training/testing have greatly improved the performance of the model.

7.2 Strengths and Weaknesses of the Models:

- **Decision Tree** is a very useful model in data classification with its simple and explainable structure; however, it tends to over-adapt in complex data.
- **Random Forest** is a powerful model that increases the ability to generalize using multiple Decision Trees. In this way, he learned the complex relationships observed in the data set better.
- **SVM and Logistic Regression** are the models that are more successful in linear classifications. SVM has been especially effective in data sets with complex boundaries.
- Ensemble Learning performed using **Voting Classifier** has improved the overall performance by taking advantage of the strengths of the models. This has been especially effective in Decoupling the imbalance between classes and improving the accuracy of forecasts.

7.3 The Effect of Hyperparameter Adjustment on Performance:

- The performance of the models has been significantly improved with the hyperparameter settings. In particular, the increase in recall and F1 scores has shown that the optimized model reduces performance differences between classes and gives more balanced results.

7.4 The Learned:

- In this project, I tried to predict air quality levels using different machine learning models. I worked on data preprocessing steps such as filling in missing values, encoding categorical variables, and dividing the data into training and test sets. I used Decision Tree, Random Forest, Logistic Regression and SVM. The Random Forest and the Voting Classifier gave the best accuracy (95%). I also adjusted the hyperparameters to make the Random Forest model better, which helped me improve its performance and stability. This project taught me how different models work and the importance of adjusting hyper parameters to achieve better results. Hyperparameters to get better results.

---

8. Conclusion: General Evaluation

In this project, different machine learning algorithms, data preprocessing process and hyperparameter optimizations used to improve air quality prediction were analyzed in detail. Data mining played a critical role in this project and made an important contribution to the process of classifying and analyzing environmental factors.

In particular, the use of ensemble methods has increased the accuracy and generalization ability of air quality forecasting by taking advantage of the strengths of different models. Hyperparameter optimization has improved the performance of the models, allowing for more balanced and accurate forecasts.

The most important thing I learned during the project was how big a role data preprocessing, model optimization and hyperparameter adjustment play in the success of machine learning models. These processes directly affect not only the accuracy of the model, but also its generalization ability on the test data. These comparisons with different models and optimization steps clearly showed that we need to develop more reliable machine learning models for real-world applications.