**DHAANISH AHMED COLLEGE OF ENGINEERING, CHENNAI**

APPLIED DATASCIENCE-PHASE 2

**AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU-project9**

**BY**

**A.M.MOHAMMED**

**NOOORISHA BILALY**

***INTRODUCTION:***

Air quality analysis and prediction are crucial components of environmental science and public health management. These fields focus on monitoring, assessing, and forecasting the quality of the air we breathe, with the ultimate goal of improving air quality and minimizing its adverse effects on human health and the environment.

Air quality analysis involves the collection and examination of data related to various pollutants present in the Earth's atmosphere, such as particulate matter (PM2.5 and PM10), ground-level ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2), carbon monoxide (CO), and volatile organic compounds (VOCs). Monitoring stations, satellite technology, and specialized instruments continuously measure these pollutants to generate real-time data.

***ABOUT DATASET:***

Where did we get the dataset?

**Kaggle:**

* we got our ‘Air quality analysis and

prediction in Tamil Nadu’ dataset

from Kaggle. [www.kaggle.com/data](http://www.kaggle.com/data).

* Kaggle is a popular platform for sharing

datasets and hosting data science competitions.

**Details:**

* The dataset contains all details about the

quality of air in Tamil Nadu which is for

predicting the quality of Air.

* Where the air contains S02,NO2,etc., which helps us to analyze the Air quality in various cities of Tamil Nadu.

The dataset contains the following details,

* Sample: Unique identifiers for data measurements.
* State: Denotes Tamil Nadu as the data's geographical origin.
* City: Names of cities in Tamil Nadu where monitoring occurred.
* Location of Monitoring Station: Specific addresses or locations of monitoring stations.
* Agency: Organizations responsible for data collection and analysis.
* SO2 (Sulfur Dioxide): Concentrations of sulfur dioxide, a harmful gas from combustion.
* NO2 (Nitrogen Dioxide): Levels of nitrogen dioxide, an air pollutant from various sources.
* RSPM10 (PM10): Measurements of particulate matter with a diameter of 10 micrometers.
* PM 2.5 (Particulate Matter 2.5): Data on fine particulate matter with a diameter of 2.5 micrometers, posing health risks.

***COLUMNS THAT WE USED:***

**Sample:**

This column likely contains identifiers or labels for individual data samples or measurements. Each row in your dataset represents a specific data sample, and the "Sample" column helps distinguish one sample from another. It may consist of unique numerical or alphanumeric identifiers.

**State:**

This column indicates the state within India where the air quality measurements were taken. In your case, it is Tamil Nadu, which is a state in the southern part of India.

**City:**

This column specifies the name of the city or urban area within Tamil Nadu where the air quality monitoring was conducted. Different cities within Tamil Nadu may have varying levels of air pollution due to factors like industrial activity, traffic, and geography.

**Location of Monitoring Station:**

This column provides information about the specific location or address of the monitoring station within the city. Monitoring stations are strategically placed to capture air quality data from different parts of the city, including industrial, residential, and commercial areas.

**Agency:**

This column likely mentions the organization or agency responsible for conducting the air quality monitoring and data collection. Various government agencies and environmental organizations are typically involved in managing and reporting air quality data.

**SO2 (Sulfur Dioxide):**

This column contains measurements of sulfur dioxide (SO2) concentrations in the air. SO2 is a gaseous air pollutant primarily produced by the burning of fossil fuels, such as coal and oil. It can have adverse health effects and contribute to acid rain and smog formation.

**NO2 (Nitrogen Dioxide):**

This column contains measurements of nitrogen dioxide (NO2) concentrations in the air. NO2 is a common air pollutant produced by combustion processes, including vehicle emissions and industrial activities. High levels of NO2 can irritate the respiratory system and contribute to the formation of ground-level ozone.

**RSPM10 (Respirable Suspended Particulate Matter - PM10):**

This column includes measurements of particulate matter with a diameter of 10 micrometers or less (PM10). RSPM10 consists of tiny solid particles or liquid droplets suspended in the air. These particles can come from various sources, including dust, construction, and industrial emissions, and can have adverse health effects when inhaled.

**PM 2.5 (Particulate Matter 2.5):**

This column contains measurements of fine particulate matter with a diameter of 2.5 micrometers or less (PM2.5). PM2.5 is even smaller and more harmful than PM10, as it can penetrate deeper into the lungs and pose significant health risks. It originates from similar sources as PM10 but has a more significant impact on respiratory and cardiovascular health.

***DETAILS OF LIBRARIES USED AND WAY TO DOWNLOAD:***

**Pandas (Data Manipulation):**

Pandas is a fundamental library for data manipulation and analysis, offering data structures like Data Frames and Series.

**Installation:**

Use the following command to install Pandas: pip install pandas

**NumPy (Numerical Computing):**

NumPy is essential for numerical operations, providing support for arrays and matrices.

**Installation:**

Install NumPy with the following command: pip install numpy

**Matplotlib (Data Visualization):**

Matplotlib is a widely used library for creating static and interactive data visualizations.

**Installation:**

Use this command to install Matplotlib:

pip install matplotlib

**Seaborn (Statistical Data Visualization):**

Seaborn is built on top of Matplotlib and simplifies the creation of informative statistical graphics.

**Installation:**

Install Seaborn with pip: pip install seaborn.

**Scikit-learn (Machine Learning):**

Scikit-learn is a powerful library for machine learning and data mining tasks.

**Installation:**

Use this command to install Scikit-learn: pip install scikit-learn

***HOW TO TRAIN AND TEST:***

Training and testing a dataset is a fundamental step in building and evaluating machine learning models. The process involves dividing your dataset into two parts: one for training the model and the other for testing the model's performance. Here are the general steps to train and test a dataset:

**Data Splitting:**

* Start by splitting your dataset into two subsets: a training set and a testing set.
* The common practice is to allocate a larger portion of the data for training, often around 70-80% of the dataset, and the remaining portion for testing.

**Data Preprocessing:**

* Before training or testing, preprocess the data. This may involve tasks such as data cleaning, feature selection, handling missing values, and scaling or normalizing features to ensure the data is in a suitable format for the machine learning algorithm.
* Select a Machine Learning Algorithm.
* Choose an appropriate machine learning algorithm based on your problem's characteristics. Common algorithms include regression, classification, clustering, and more specialized models for specific tasks.

**Training the Model:**

* Use the training data to fit the chosen machine learning model. This involves feeding the algorithm with features (input variables) and their corresponding labels (output or target variable).
* The model will learn patterns and relationships in the training data.

**Model Evaluation:**

* Once the model is trained, use the testing set to evaluate its performance. The model has not seen this data during training, making it a suitable benchmark for assessing generalization.
* Common evaluation metrics include accuracy, precision, recall, F1-score, mean squared error (MSE), etc., depending on the type of problem (classification, regression, etc.).

**Tuning and Validation (Optional):**

* If the model's performance is unsatisfactory, you can fine-tune hyperparameters or explore different algorithms. Cross-validation techniques can help optimize the model further.

**Make Predictions (Inference):**

* After achieving a satisfactory model, you can use it to make predictions on new, unseen data.

**Assess Model on Real Data (Deployment):**

* if the model performs well during testing, you can deploy it for real-world applications, where it can make predictions based on incoming data.

**Monitoring and Maintenance:**

* Continuously monitor your model's performance in the real world and update it as necessary to adapt to changing data distributions or requirements.

***METRICS USED FOR ACCURACY CHECK:***

Metrics used for the analysis and prediction of a dataset depend on the type of problem you are addressing—whether it's a classification problem, regression problem, clustering problem, or any other specific task. Here are some commonly used metrics for different types of data analysis and prediction tasks:

Classification Metrics (for binary or multi-class classification problems):

* **Accuracy**: Measures the proportion of correctly classified instances. It's suitable when classes are balanced.
* **Precision:** Measures the ratio of true positives to the total predicted positive instances. Useful when minimizing false positives is essential.
* **Recall (Sensitivity or True Positive Rate):** Measures the ratio of true positives to the total actual positive instances. It's useful when you want to capture all relevant positive instances.
* **F1-Score:** The harmonic mean of precision and recall. It provides a balanced measure of a model's performance, especially when classes are imbalanced.
* **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values.
* **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual values. It gives more weight to larger errors.
* **Root Mean Squared Error (RMSE):** The square root of MSE, providing a more interpretable measure.
* **R-squared (R²) Score:** Measures the proportion of variance in the dependent variable explained by the independent variables. A higher R² indicates a better fit.

**THANK YOU**