

**COLLEGE CODE: 5113**

# APPLIED DATA SCIENCE

**AIR QUALITY ANALYSIS AND**

**PREDICTION IN TAMILNADU-project9**

***TEAM MEMBERS:***

* R.AFREEN TAJ-au511321104003
* B.DEEPIKA-au511321104017
* J.DELSI JULIE-au511321104019
* J.DHANU PRIYA-au511321104020

***INTRODUCTION***.

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
* Kaggle is a popular platform for sharing datasets and hosting data science competition 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.

**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.

***BEGINNING WITH THE PROJECT***

To begin building a project for air quality analysis and prediction, we first need to load the dataset.

We have a dataset file in a common format like CSV, here are the steps to load the dataset:

**1.Importing the required Libraries(data.csv):**

In this step, we import the necessary Python libraries and modules to work with our data and perform various data processing and machine learning tasks.

## import pandas as pd

## import numpy as np

## from sklearn.preprocessing import Imputer

## from sklearn.model\_selection import train\_test\_split

## from sklearn.preprocessing import StandardScaler

## from sklearn.preprocessing import OneHotEncoder

**2.Importing the data set(read data set; create matrix ):**

This step involves loading our dataset into memory. We use libraries like pandas to read data from a CSV file or other formats. After loading, we create a feature matrix (often denoted as X) and a target vector (often denoted as Y).

## dataset = pd.read\_csv('data.csv')

## X = dataset.iloc[:, :-1].values

## Y = dataset.iloc[:, -1].values

**3.Handling the Missing Data.(sklearn.preprocessing library contains class called imputer, helps in missing data):**

Datasets often have missing values. The sklearn.preprocessing.Imputer class is used to address this issue. You can specify a strategy for imputing missing values, such as replacing them with the mean, median, or mode of the column.

## imputer = Imputer(missing\_values='NaN', strategy='mean', axis=0)

## imputer = imputer.fit(X[:, columns\_with\_missing\_data])

## X[:, columns\_with\_missing\_data] = imputer.transform(X[:, columns\_with\_missing\_data])

**4.Encoding Categorical Data.(one-hot encoding):**

One-hot encoding is a technique used to convert categorical data into a numerical format. Each category becomes a binary feature (0 or 1) in a new column, making it suitable for machine learning algorithms.

## Encode=OneHotEncode(categoricalfeatures=categoricalcolumn)

## X = encode.fit\_transform(X).toarray()

**5.Splitting the data set into test set and training set.( import train train\_test\_split)(X\_train,X\_test, Y\_train,Y\_test):**

Before building a machine learning model, it's essential to divide our dataset into two sets: a training set and a test set. The training set is used to train the model, while the test set is used to evaluate its performance.

## X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=0)

**6.Feature Scaling.(import StandardScaler):**

Feature scaling ensures that all features have the same scale, typically with a mean of 0 and a standard deviation of 1.

## scaler = StandardScaler()

## X\_train = scaler.fit\_transform(X\_train)

## X\_test = scaler.transform(X\_test)

**PREPROCESSING THE DATASET:**

Preprocessing of data in a dataset refers to the various techniques and operations applied to the data before using it for analysis, modeling, Here's a more detailed explanation of data preprocessing within the context of a dataset:

**1. Data Cleaning:**

Handling Missing Values: Identify and deal with missing data, which may involve filling in missing values, removing rows with missing data, or using imputation techniques.

Dealing with Duplicates: Detect and remove duplicate records to ensure data integrity.

**2. Data Transformation:**

Feature Scaling: Normalize or standardize numerical features to bring them to a similar scale. This is important for algorithms sensitive to feature scales.

Feature Encoding: Convert categorical variables into a numerical format using techniques like one-hot encoding or label encoding.

Feature Engineering: Create new features or modify existing ones to capture relevant information and patterns in the data.

Binning: Group continuous data into bins or categories to simplify analysis.

Log Transformation: Apply logarithmic transformations to features when necessary to make their distribution more normal.

**3. Data Reduction:**

Dimensionality Reduction: Reduce the number of features, often using techniques like Principal Component Analysis (PCA) or feature selection to select the most relevant variables.

Outlier Detection and Handling: Identify and deal with outliers, which can distort analysis and modeling results.

**4. Data Integration:**

Merge data from multiple sources or datasets to create a consolidated dataset for analysis.

**5. Data Formatting:**

Ensure data is in the appropriate format and structure for the specific analysis or modeling task.

**6. Data Splitting:**

Split the dataset into subsets, such as training, validation, and test sets, for model development, evaluation, and testing.

**7. Data Validation:**

Verify data consistency, correctness, and integrity by checking for data entry errors and anomalies.

**PERFORMING DIFFERENT ANALYSIS**:

Performing different types of analysis on a dataset depends on the goals of your analysis and the nature of the data. Here are some common types of analysis that you might perform on a dataset:

**Descriptive Analysis:**

Summarize and describe the main characteristics of the dataset, including measures of central tendency, dispersion, and visualizations such as histograms, box plots, and bar charts.

**Exploratory Data Analysis (EDA):**

Explore the dataset to uncover patterns, relationships, and anomalies.

Visualize data using scatter plots, heatmaps, and correlation matrices.

Identify potential outliers and trends.

**Statistical Analysis:**

Conduct hypothesis testing and statistical inference to make inferences about the data.

Perform t-tests, ANOVA, chi-squared tests, and other statistical tests as appropriate.

**Regression Analysis:**

Build regression models to predict a continuous target variable based on one or more predictor variables.

Evaluate model performance using metrics like R-squared, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

**Classification Analysis:**

Develop classification models to predict categorical outcomes or classes.

Evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and ROC curves.

**Clustering Analysis:**

Apply clustering algorithms to group similar data points together.

Use techniques like K-means, hierarchical clustering, or DBSCAN.

**Anomaly Detection:**

Identify outliers or anomalies in the dataset using methods like isolation forests or one-class SVM.

**CODE:**

The code provides a basic example of linear regression, a simple machine learning technique used for predicting a continuous target variable (PM10 levels) based on one or more predictor variables

## import pandas as pd import numpy as np from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression import matplotlib.pyplot as plt np.random.seed(0) dates = pd.date\_range(start="2014-01-01", end="2014-12-31", freq="D") pm10\_levels = np.random.randint(10, 100, len(dates)) temperature = np.random.randint(25, 35, len(dates)) humidity = np.random.randint(40, 70, len(dates))

## data = pd.DataFrame({"Date": dates, "PM10": pm10\_levels, "Temperature": temperature, "Humidity": humidity})

## X = data[["Temperature", "Humidity"]] y = data["PM10"] X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

## model = LinearRegression() model.fit(X\_train, y\_train) y\_pred = model.predict(X\_test) from sklearn.metrics import mean\_squared\_error, r2\_score mse = mean\_squared\_error(y\_test, y\_pred) r2 = r2\_score(y\_test, y\_pred) print("Mean Squared Error:", mse) print("R-squared:", r2) plt.scatter(y\_test, y\_pred)

## plt.xlabel("Actual PM10 Levels") plt.ylabel("Predicted PM10 Levels") plt.title("Actual vs. Predicted PM10 Levels") plt.show()

**CONCLUSION:**

In this air quality analysis, we have conducted a comprehensive examination of the dataset to gain insights into the quality of the air and the factors affecting it. Our analysis started with a descriptive examination of the data. We calculated statistics such as mean, median, and standard deviation for air quality parameters like PM2.5, PM10, CO, SO2, and NO2. This provided a general understanding of the data distribution.

## 