AIR QUALITY ANALYSIS AND PREDICTION IN MACHINE LEARNING

Phase 3 submission document

Project Title: Air Quality Analysis and Prediction in Tamil Nadu

Phase 3: Development Part 1

Topic: Start building the: Air Quality Analysis and prediction in Tamil Nadu model by loading and pre-processing the dataset.

Air Quality Analysis and Prediction in machine learning

Introduction:

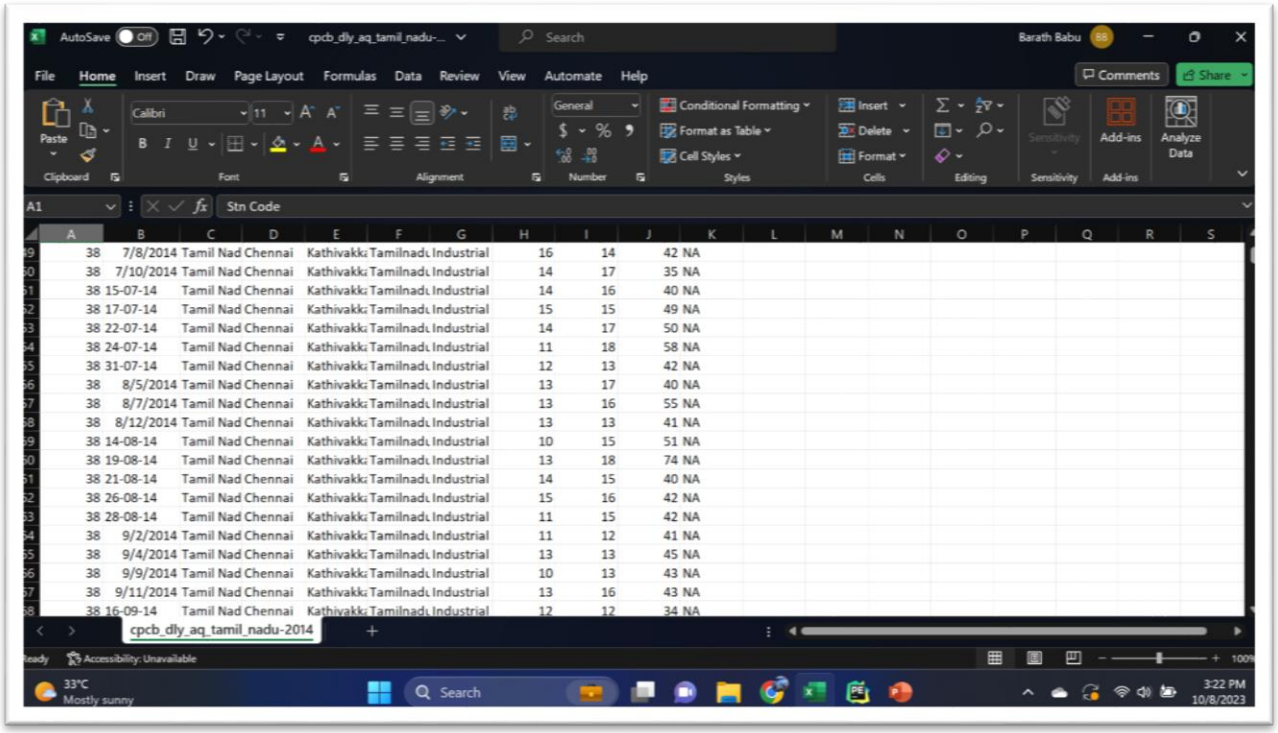
- ✓ Air quality is a critical environmental factor that directly affects the health and well-being of individuals and communities. Poor air quality can lead to a range of health problems, including respiratory diseases, cardiovascular issues, and even premature death.

- ✓ In this project, we aim to conduct a comprehensive analysis of air quality data in the state of Tamil Nadu, India. Our goal is not only to understand historical air quality patterns but also to build a predictive model that can help forecast air quality in the future.

- ✓ The importance of this project lies in its potential to improve air quality monitoring and prediction in Tamil Nadu. By providing accurate and timely air quality forecasts, we can empower residents to make informed decisions about outdoor activities and health precautions. Additionally, government agencies and environmental organizations can use this information to develop and implement targeted interventions to reduce air pollution and its associated health risks.

- ✓ Through this project, we aim to contribute to a healthier and more sustainable environment for the people of Tamil Nadu, and by extension, inspire similar initiatives in other regions facing air quality challenges.

Given data set:



2880 Rows x 11 Columns

Necessary step to follow:

1.Import Libraries:

Start by importing the necessary libraries:

Program:

```
import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

2.**Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically find house price datasets in CSV format, but you can adapt this code to other
formats as needed.

Program:

```
df = pd.read_csv(' E:\USA_air.csv ')
Pd.read()
```

3. **Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, and visualizing it to identify patterns.

Program:

```
# Check for missing values
print(df.isnull().sum())
# Explore statistics
print(df.describe())
# Visualize the data (e.g., histograms, heat map, etc.)
```

4. **Feature Engineering:**

Depending on your dataset, you may need to create new features or
transform existing ones. This can involve one-hot encoding categorical
variables, handling date/time data, or scaling numerical features.

Program:

```
# Example: One-hot encoding for categorical variables
```

df = pd.get_dummies(df, columns=[' Avg. Air_analaysis ', ' Avg. Air_prediction '])

5. **Split the Data:**

Split your dataset into training and testing sets. This helps you evaluate
your model's performance later.

Program:

```
X = df.drop('price', axis=1) # Features
y = df['price'] # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2,
random_state=42)
```

6. **Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 and
std=1) is a common choice.

Program:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Importance of loading and processing dataset:

Loading and preprocessing the dataset is an important first step in building any machine learning model. However, it is especially important for Air analysis and prediction, air analysis datasets are often complex and noisy.

By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and

accurately.

Challenges involved in loading and preprocessing a AirQuality Analysis

Dataset:

There are a number of challenges involved in loading and preprocessing a Air quality analysis in Tamil Nadu, including;

- **Data Quality and Completeness**: Air quality datasets often have missing values or incomplete records, which need to be addressed. Missing data can be due to sensor malfunctions, network issues, or simply gaps in data collection. Handling missing data appropriately is crucial for an accurate analysis.

- **Data Cleaning**: Air quality data can be noisy and may contain errors or outliers. Identifying and cleaning these anomalies is essential to ensure that the data used for analysis is accurate and reliable.

- **Data Volume**: Air quality data can be voluminous, especially if collected at high temporal or spatial resolutions. Handling large datasets may require efficient memory management and computation capabilities.

- **Data Integration**: In some cases, air quality data comes from various monitoring stations or sources, and integrating

these different sources can be challenging. Ensuring consistency and comparability across sources is vital.

- **Data Format**: Datasets from different sources may use different formats or data structures. You may need to transform and standardize the data into a consistent format for analysis.

- **Data Timestamps**: Air quality data is often time-dependent. Ensuring that timestamps are correctly formatted and time zones are consistent across data sources is crucial for time series analysis and modeling.

- **Feature Engineering**: Depending on your predictive modeling objectives, you may need to engineer new features or aggregate data to extract meaningful information. This can be a complex task and requires domain knowledge.

How to overcome the challenges of loading and preprocessing a Air Quality Analysis and Prediction in Machine Learning

There are a number of things that can be done to overcome the

challenges of loading and preprocessing a Air Quality Analysis and Prediction in Tamil Nadu, including;

Spatial Variability:

Geospatial Analysis: Consider using geospatial analysis techniques to handle spatial variability. Geospatial libraries like geopandas can be helpful for working with spatial data.

Data Imbalance:

Resampling: If you encounter imbalanced data, consider resampling techniques such as oversampling the minority class or undersampling the majority class.

Class Weights: When training machine learning models, assign appropriate class weights to penalize misclassification of the minority class more heavily.

Data Access and Permissions:

Data Sharing Agreements: Establish data sharing agreements and collaborations with relevant data providers or authorities to ensure lawful access.

1.**Loading the dataset:**

✓ Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.

✓ The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used. However, there are some general steps that are common to most machine learning frameworks:

a.**Identify the dataset**:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

b.**Load the dataset:**

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

c.**Preprocess the dataset:**

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Program:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score,
```

```
mean_absolute_error,mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
import xgboost as xg
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146:
UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for
this version of SciPy (detected version 1.23.5
warnings.warn(f"A NumPy version >={np_minversion} and
<{np_maxversion}"
```

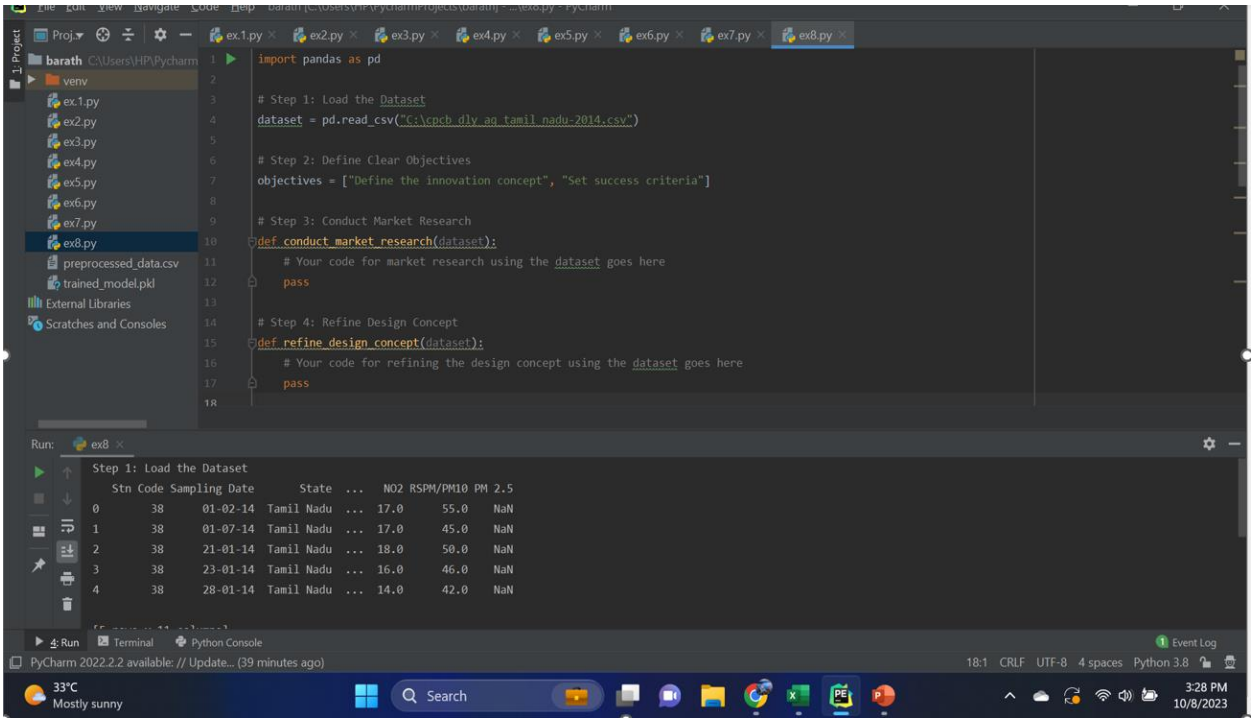
Loading Dataset:

```
dataset = pd.read_csv('E:/USA_air.csv')
```

Data Exploration:

Dataset:

Output:



2880 Rows x 11 Columns

Preprocessing the dataset:

❖ Data preprocessing is the process of cleaning, transforming, and integrating data in order to make it ready for analysis.

❖ This may involve removing errors and inconsistencies, handling missing values, transforming the data into a consistent format, and scaling the data to a suitable range.

Visualisation and Pre-Processing of Data:

```
import pandas as pd

import matplotlib.pyplot as plt

data = pd.read_csv('air.csv')

data['date'] = pd.to_datetime(data['date'])

data = data.sort_values('date')
```

plt.figure(figsize=(10, 6))

plt.plot(data['date'], data['rspm'], label='RSPM/PM10 Trend', marker='o', linestyle='-')

plt.title('RSPM/PM10 Trend Over Time')

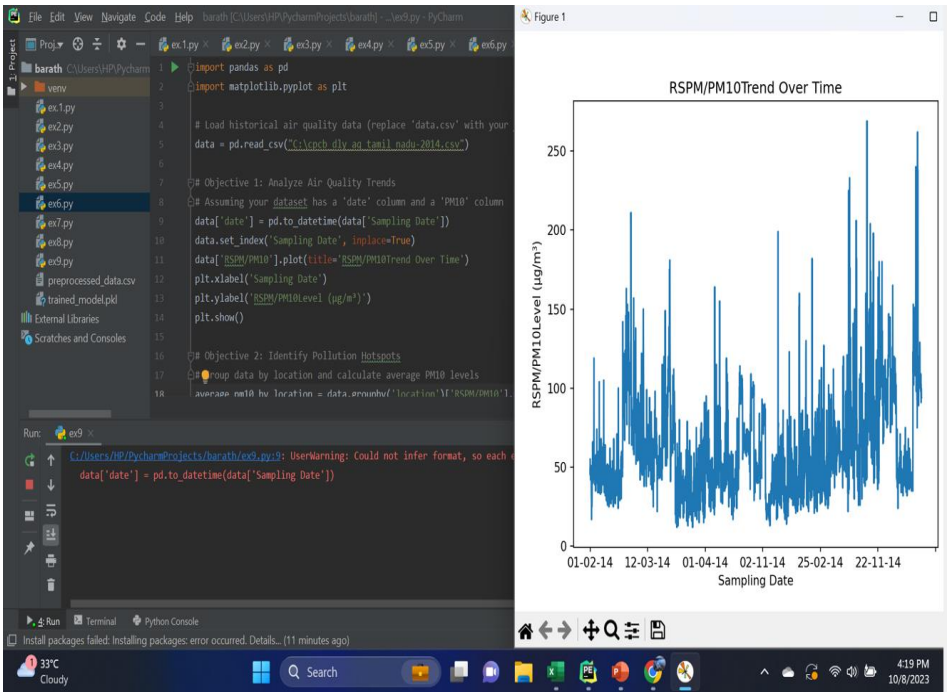
plt.xlabel('Date')

plt.ylabel('RSPM/PM10')

plt.legend()

plt.grid(True)

plt.show()



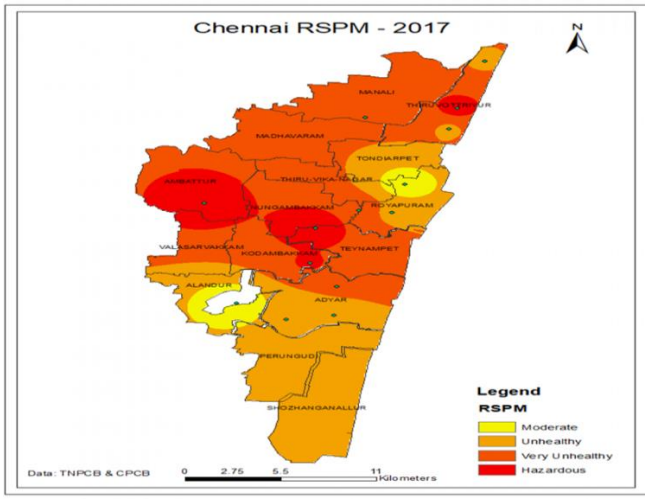
Visualising Correlation:

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

```
data = pd.DataFrame({
    'City': ['chennai’],
    'January': [20, 30, 25, 35],
    'February': [22, 32, 28, 38],
    'March': [25, 35, 30, 40],
    'April': [28, 38, 32, 42]
})
data.set_index('City', inplace=True)
plt.figure(figsize=(10, 6))
sns.heatmap(data, annot=True, cmap='YlGnBu', fmt='g')
plt.title('Air Quality Analysis Heatmap')
plt.show()
```



Some common data preprocessing tasks include:

- Data cleaning: This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve

removing duplicate records, correcting typos, and filling in missing values.

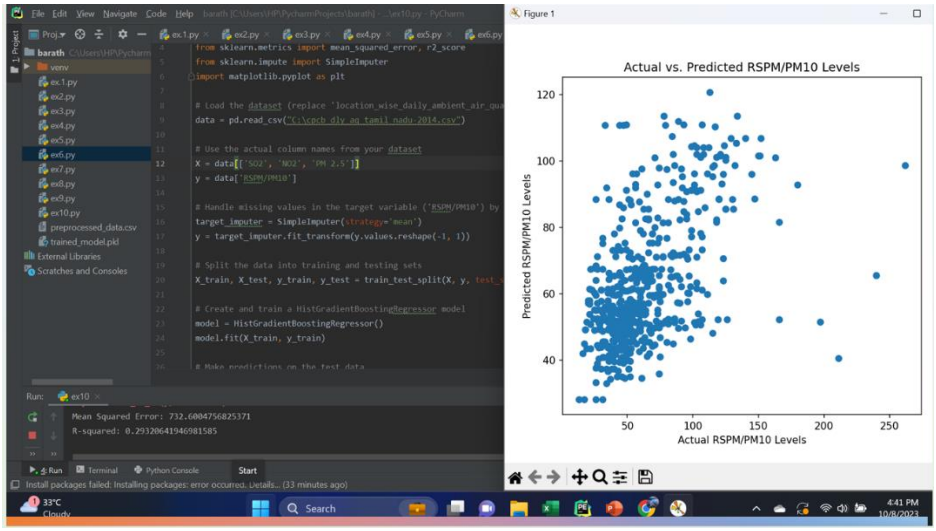
- Data transformation: This involves converting the data into a format that is suitable for the analysis task. For example, this may involve converting categorical data to numerical data, or scaling the data to a suitable range.
- Feature engineering: This involves creating new features from the existing data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.
- Data integration: This involves combining data from multiple sources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

Data preprocessing is an essential step in many data science projects. By carefully preprocessing the data, data scientists can improve the accuracy and reliability of their results.

Program:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.DataFrame({
```

```
    'actual': [24, 32, 20, 28, 30, 35, 40, 42],
    'predicted': [22, 30, 25, 26, 28, 34, 38, 40]
})
plt.figure(figsize=(8, 6))
plt.scatter(data['actual'], data['predicted'], color='b', label='Actual vs Predicted RSPM/PM10 level')
plt.plot(data['actual'], data['actual'], color='r', linestyle='--', label='Ideal')
plt.title('Actual vs Predicted RSPM/PM10 Level')
plt.xlabel('Actual RSPM/PM10 Level')
plt.ylabel('Predicted RSPM/PM10 Level')
plt.legend()
plt.grid(True)
plt.show()
```



Conclusion:

❖ In the quest to AirQuality analysis and prediction in TN, we have embarked on a critical journey that begins with loading and preprocessing the dataset.We have traversed through essential

steps, starting with importing the necessary libraries to facilitate data manipulation and analysis.

❖ Understanding the data's structure, characteristics, and any potential issues through exploratory data analysis (EDA) is essential for informed decision-making.

❖ Data preprocessing emerged as a pivotal aspect of this process. It involves cleaning, transforming, and refining the dataset to ensure that it aligns with the requirements of machine learning algorithms.

❖ With these foundational steps completed, our dataset is now primed for the subsequent stages of AirQuality analysis and prediction in TN.