"AIR QUALITY PREDICTION AND DETECTION USING MACHINE LEARNING"

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering
Department of Computer Application

Fall Semester 2022-2023

SET CONFERENCE

First Review (7th December 2022)

22MCA0209 - HRIDAM CHAKRABORTY

22MCA0265 - BHUBESH SR

22MCA0273 - RITESH GUPTA

Guide Name: Dr. RAMALINGAM M

ABSTRACT :

Air quality Plays a vital role in human Health. Pollution in Air leads to a vast number of health issues. It is a vast problem in major metropolitan cities where the population density is high. Air pollution caused by various parameters taken by people, such as transportation, power, and fuel use, are affecting air quality. Monitoring, predicting, and detecting air quality have become essentially important in this era, mainly in developing countries like India. Air Quality Index (AQI), is used to measure the quality of air. Machine learning (ML) is a field in which an artificial intelligence device collects sensor data and learns to act. Machine Learning (ML) is the better approach to predict air quality as compared to other techniques such as probability and statistical methods. Predict air relative humidity by considering various parameters such as CO, Benzene, Titanium, NO, Sulfur dioxide, etc.

Through this project, we are aiming to predict air quality using machine learning algorithms. In this paper, we use a Support Vector Regression (SVR) model to forecast the levels of various pollutants and the air quality index. Linear Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest Method (RF) to predict the Relative humidity of air and uses Root Mean Square Error to predict the accuracy.

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DESCRIPTION

Because of new inventions there is a rapid increase in the development, serious population growth and increased number of vehicles will give rise to so many critical problems related to the environment such as acid rain, deforestation, air pollution, water pollution, emission of toxic materials and so on. To fulfil the needs of the growing population there is the drastic increase in industrialization that may lead to the emission of harmful gases in the atmosphere from various industries that will cause the serious air pollution problem in urban areas throughout the world. This means that the air we or people breathe is not clean air but it is polluted as so many harmful gases and particles are present in the air that adversely affect the human health.

In most of the urban areas the air pollution becomes a serious concern. The people should know about the air they breathe. So the Central Pollution Control Board (CPCB) develops the national Air Quality Index (AQI) for the cities in India . AQI gives the idea about quality of air or at what extent the air in the particular location is polluted. However, the selection of pollutants depends on the AQI objectives, averaging μperiod, Data availability, Monitoring frequency and measurement methods. AQI can be defined as it is a numerical value that the governmental agencies used to measure the levels of air pollution in the atmosphere and communicate it with population. If AQI increases then large percentage of population is affected because it adversely affects the human health. As we know that AQI can be calculated by using the concentration of different air pollutants and finally we get the single numerical value as AQI.

CHAPTER 2

LITERATURE SURVEY

The following chapters give an overview of the various methodologies used by various authors for air pollution detection using machine learning methodologies. We can observe that there is fine comparison made between 5 major machine learning algorithms whether they are able to predict the presence of the disease with greater accuracy, achieving optimal performance.

Dr. Ravindra P Rajput and Deepu B P stated in their paper that Machine Learning algorithms can provide the best prediction for Air Quality Index. They used several sensors and Arduino Uno Microcontroller to collect the dataset for predicting the Air Quality Index. To calculate AQI they specifically used the KNN algorithm.

The authors Shreyas Simu, Varsha Tukar and Rohit Martires compared multiple algorithms to measure Air Quality Index. They took help of Logistic regression (LR), Artificial Neural Network (ANN), Auto regression (AR) to identify the best algorithm for the prediction. According to them, ANN has the best result for the prediction.

The authors proposed a model where they used Logistic regression to detect the air quality by using data samples. Later Auto Regression came out with the prediction values of PM 2.5 based on the previous dataset values of PM 2.5. Logistic Regression, Auto Regression in combined gave the most efficient prediction of the air quality.

The authors proposed a model which gave prediction results based on the current dataset of air. Big Data Analysis is employed to process the current data set. Later they found out that the Decision Tree Algorithm gave the most efficient result over all the algorithms they used.

The authors used Deep Learning in their proposed model to predict the PM 2.5 . They specifically used BiLSTM (Bi Directional Long Short Time Memory) network for their model. More over Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) used in the model for efficient solution.

CHAPTER 3

PROPOSED WORK

The information of Air pollution is collected using some sensors and some others from satellites that get processed and stored in a dataset. After collection of the dataset , it gets preprocessed using some methods such as Normalization , Discretization and Attribute selection. If the Dataset is ready , the dataset is split into training and testing dataset. Supervised Machine Learning Algorithms are applied on the training dataset to get the appropriate results.The results are matched with the testing dataset and results are analyzed.The Air pollution prediction uses several supervised machine learning algorithms considers Autoregression , Linear Regression , Decision tree , Support Vector Machine.

3.1 AutoRegression

An AutoRegression Model examine observations from previous time steps as a input to a regression equation to predict the value at the next time step.Autoregression is used for forecasting on a range of time series problems and also when there is a correlation between values and these values precede and succeed them. A regression model, such as linear regression, models an output value based on a linear combination of input values. Basically, the process is a linear regression of the data in the series against one or more past values in the same series.

In an AutoRegression model, the value of the outcome variable (Y) at some point in time _t‘ is like “regular” linear regression directly related to the predictor variable (X). Where simple linear regression and AR models differ, is when Y is dependent on X and previous values for Y.

3.2 Linear Regression (LR)

The supervised learning algorithm is used to find linear relationships between target (**independent variable**) and predictors(**dependent variable**). Linear Regression gives a sloped line of straight which describes the relationships within the variables. In case the Value of x i.e independent variable increases , the y axis i.e dependent variable also increases , directly proportional to each other.

3.3 Random Forest

The implementation of the RF model is using the Scikit-learn random forest classifier class. This meta estimator fits several decision tree classifiers on various sub-samples of the dataset.

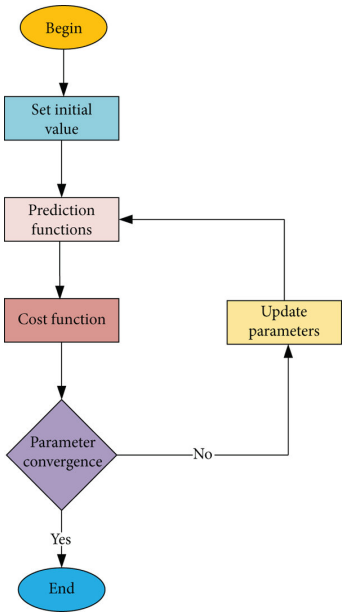

Fig : 3.3 Flowchart of Random Forest Model

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

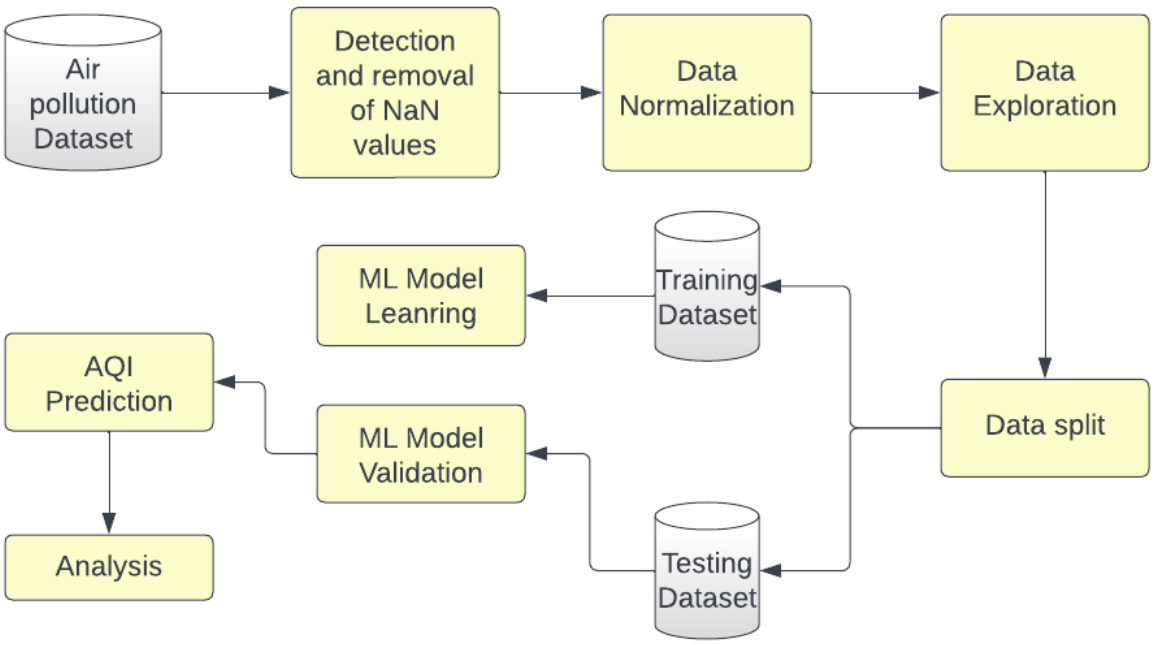

Fig : 4.1 Architecture of Air Quality Prediction model

The entire project is divided into several parts. First, we choose data to understand business and environmental needs. Next, we select air quality data, which is important for both the environment and human life. Predicting air quality is important to industry and pollution management. The second stage of data pre-processing Whatever data we have is raw, and if we apply it to a model straight, the output could be incorrect. Preprocessing is an important stage in data forecasting. When the values are too large for the model to understand, a transformation step is required. Following data preparation, we must use a variety of models. The next stage is

evaluation, which is critical since whatever model we use, we must compare the results on an error basis in order to justify the better model.

4.2 DATA FLOW

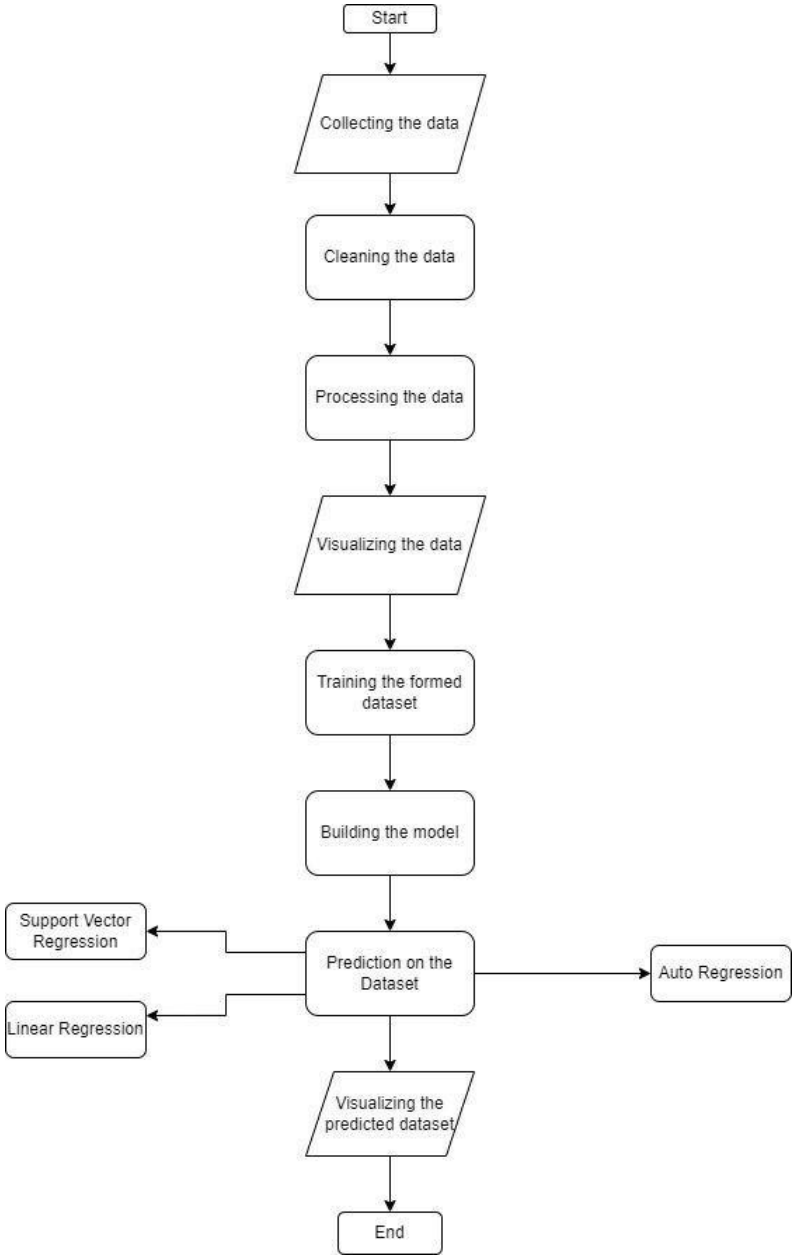

Fig : 4.2 Data Flow Diagram

CHAPTER 5

METHODOLOGY AND MODULE DESCRIPTION

1. Data Collection
2. Data Cleaning
3. Data Pre-processing
4. Data Visualization
5. Training Dataset Formation
6. Model Building
7. Prediction on testing dataset

5.1 Data Collection

Meteorological parameters like Maximum temperature (°C), Minimum temperature (°C), Minimum temperature (°C), Average relative humidity (%), Total rainfall and / or snowmelt (mm), Average visibility (Km), Average wind speed (Km/h), Maximum sustained wind speed (Km/h) were taken from TuTiempo.net. the Concentration of various Air Pollutants like PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3 Collected from kaggle.com.

5.2 Data Cleaning

Data cleaning identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.Meteorological parameters were web-scraped from WEBSITE and the air pollutants concentrations were taken from Kaggle dataset.From the data, various features which are not useful in AQI prediction like ‘NH3’, ‘NO’, ‘Benzene’, ‘Toluene’ are removed. Finally, the independent features used are PM2.5, PM10, NO2, NOx, CO, SO2, O3, average temp, maximum temperature, minimum temperature, humidity, visibility, and wind speed. Rows with missing AQI values were dropped whereas the missing values in independent features were interpolated.Feature Scaling was performed.

5.3 DATA PREPROCESSING :

The first and most important criterion to ensure the effective construction of forecasting models is data quality and representativity. Missing data imputation, eliminating or changing outlier observations, data transformation (typically normalization and standardization), and feature engineering are all examples of data preparation. While the first two phases are important for obtaining more precise and full data sets, the third step is often used to obtain data that is more consistently distributed and to reduce data variability. Finally, the fourth phase is utilized to generate a new dataset that is often smaller and more informative. Feature extraction and feature selection are usually included in the final stage.We perform these steps on our data as follows.

5.3.1 Imputation of missing data:

We found that more than 25 - 35% data was missing, so we removed the Unrequired data from the dataset .For the other fields in the US embassy data and the Central Pollution Control Board data, we substituted missing data by second order polynomial estimation using nearest available data points. It gave better results than using series mean or linear interpolation.

5.3.2 Removing or modifying outliers:

An irregular pattern was observed in pollutant data between August and October 2020 in both US embassy data and the Central Pollution Control Board data. Thus, these data were removed. For data modification, we used the power transformation method. This provides a nonlinear transformation that is more robust to noise and hence produces better data.

5.3.3 Feature extraction:

The date component in the Central Pollution Control Board data and the date-time component in the US embassy data were used to produce new features. The date component was used to obtain a field called seasons. Four seasons were used (Summer, Fall, Winter, Spring). The cyclic nature of the time component was exploited to obtain two fields $\{\sin(2\pi hour/24), \cos(2\pi hour/24)\}$. The date component was also used to obtained fields for day, month and year

5.3.4 Feature selection:

From the features obtained in the previous step using feature engineering, a few variables were selected to reduce dimensionality of the dataset and remove collinearity. Correlation-based feature selection was used , to check for collinearity among features.We also used Principal Component Analysis (PCA) for reducing the dimensionality of the dataset. It enabled us to reduce the number of variables for each pollutant by about 76%. We compare the results with and without PCA in the next section.

5.4 Data Visualization

5.4.1 Linear Regression

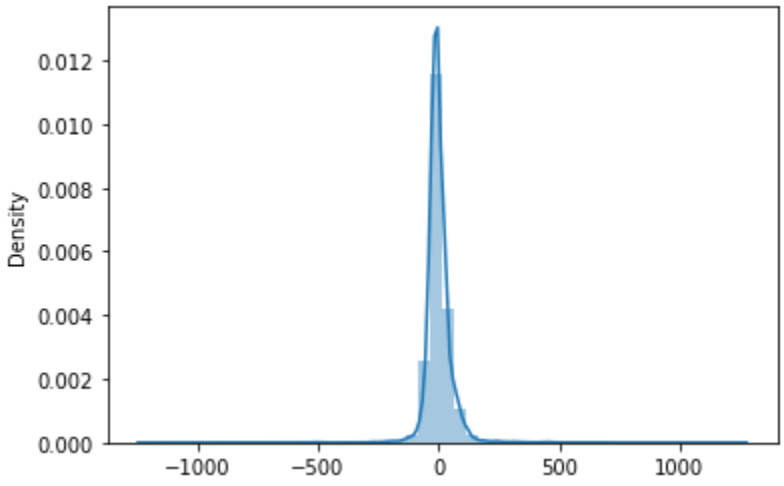

Fig : 5.4.1 Model Evaluation

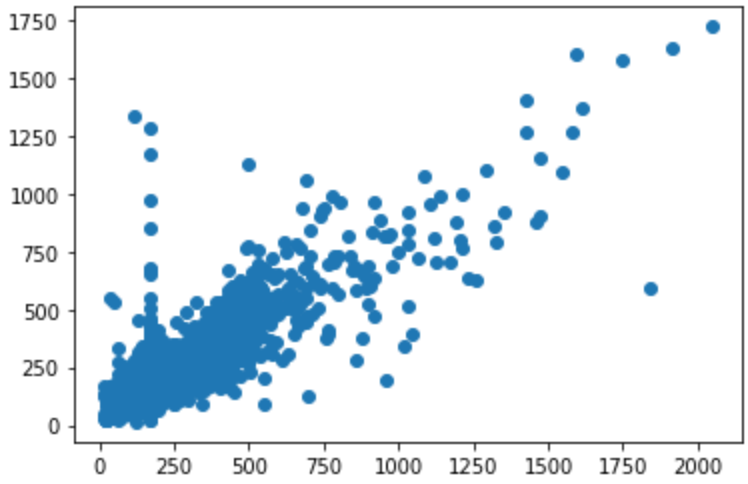

Fig : 5.4.2 Scatter Diagram

5.4.2 RANDOM FOREST

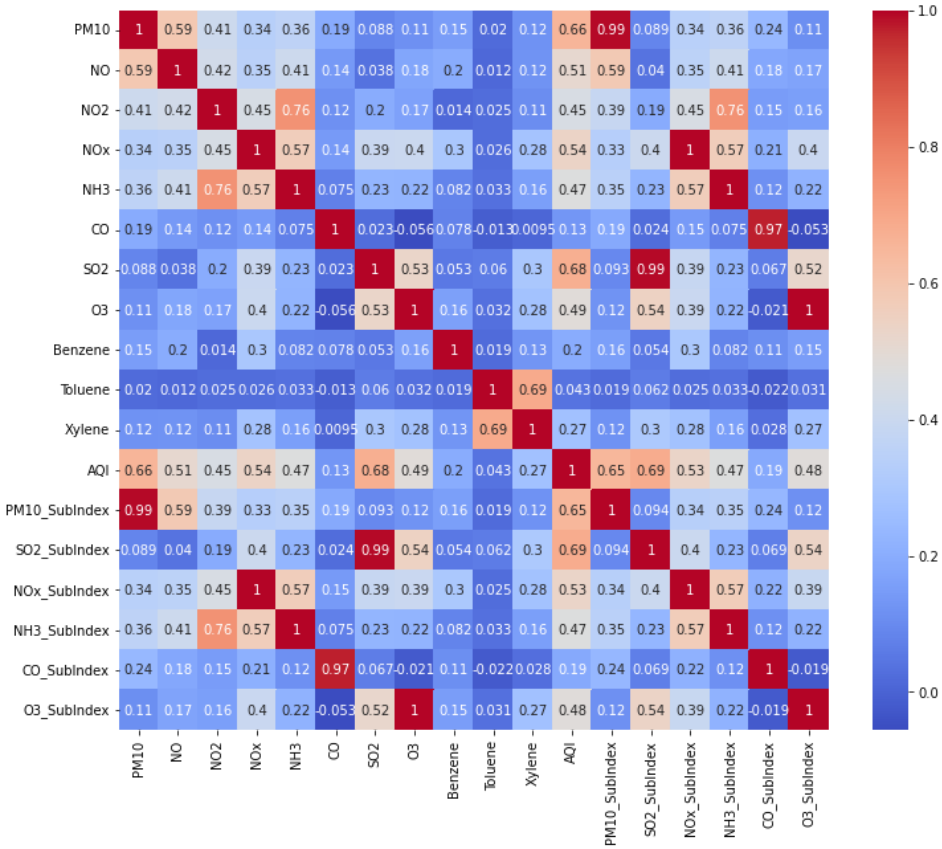

Fig:5.4.2 Correlation

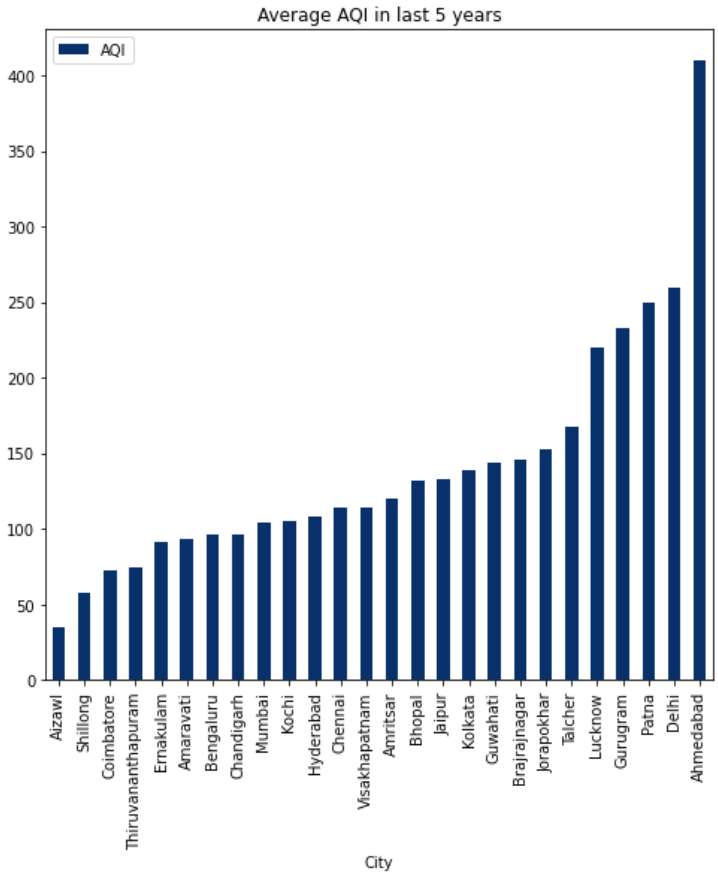

Fig:5.4.2 Average AQI

CHAPTER 6

IMPLEMENTATION

SOURCE CODE

6.1 Random Forest

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv('city_day.csv',parse_dates = ["Date"])
df
X = final_df[['AQI']]
y = final_df[['AQI_Bucket']]
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import  train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0)
clf = RandomForestClassifier(random_state = 0).fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Enter the value of AQI:")
AQI = float(input("AQI :  "))
output = clf.predict([[AQI]])
output
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

6.2 Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

df=pd.read_csv("city_day.csv")
df.head()
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest= train_test_split(x,y,test_size=0.4,random_state=0)
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(xtrain,ytrain)
ypred=linreg.predict(xtest)
x=df.iloc[:,2:13]
y=df.iloc[:,-1]
linreg.intercept_
linreg.coef_
coef_df=pd.DataFrame(linreg.coef_,x.columns,columns=["Coefficient"])
coef_df
plt.scatter(ytest,ypred)
sns.distplot((ytest-ypred),bins=50)
from sklearn.metrics import mean_absolute_error as mae, mean_squared_error as mse,r2_score
print(f"MSE:-{mse(ytest,ypred)}")
print(f"R-squared:-{r2_score(ytest,ypred)}")
```

SCREENSHOTS

RANDOM FOREST

DATASET DESCRIPTION:

```
df=pd.read_csv('city_day.csv',parse_dates = ["Date"])
df
```

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29526	Visakhapatnam	2020-06-27	15.02	50.94	7.68	25.06	19.54	12.47	0.47	8.55	23.30	2.24	12.07	0.73	41.0	Good
29527	Visakhapatnam	2020-06-28	24.38	74.09	3.42	26.06	16.53	11.99	0.52	12.72	30.14	0.74	2.21	0.38	70.0	Satisfactory
29528	Visakhapatnam	2020-06-29	22.91	65.73	3.45	29.53	18.33	10.71	0.48	8.42	30.96	0.01	0.01	0.00	68.0	Satisfactory
29529	Visakhapatnam	2020-06-30	16.64	49.97	4.05	29.26	18.80	10.03	0.52	9.84	28.30	0.00	0.00	0.00	54.0	Satisfactory
29530	Visakhapatnam	2020-07-01	15.00	66.00	0.40	26.85	14.05	5.20	0.59	2.10	17.05	NaN	NaN	NaN	50.0	Good

29531 rows × 16 columns

PREDICTION

In [33]:

```
print("Enter the value of AQI:")
AQI = float(input("AQI : "))
output = clf.predict([[AQI]])
output
#0-->Good
#1-->Satisfactory
#2-->moderate
#3-->poor
#4-->Very poor
#5-->Severe
```

```
Enter the value of AQI:
AQI : 110
```

Out[33]: array([2])

In [34]:

```
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

```
1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       388
           1       1.00      1.00      1.00      2397
           2       1.00      1.00      1.00      2608
           3       1.00      1.00      1.00       805
           4       1.00      1.00      1.00       861
           5       1.00      1.00      1.00       324

    accuracy                           1.00      7383
   macro avg       1.00      1.00      1.00      7383
weighted avg       1.00      1.00      1.00      7383

[[ 388    0    0    0    0    0]
 [   0 2397    0    0    0    0]
 [   0    0 2608    0    0    0]
 [   0    0    0  805    0    0]
 [   0    0    0    0  861    0]
 [   0    0    0    0    0  324]]
```

LINEAR REGRESSION

```
coef_df=pd.DataFrame(linreg.coef_,x.columns,columns=["Coefficient"])
coef_df
```

	Coefficient
PM2.5	0.952038
PM10	0.274162
NO	-0.031770
NO2	0.408310
NOx	0.133462
NH3	-0.053039
CO	9.941018
SO2	0.625933
O3	0.186043
Benzene	-0.313022
Toluene	0.241787

Dividing the Data into X and Y

```
[23] x=df.iloc[:,2:13].values
     y=df.iloc[:,-1].values
```

```
[24] x
```

```
array([[6.74505779e+01, 1.18127103e+02, 9.20000000e-01, ...,
        1.33360000e+02, 0.00000000e+00, 2.00000000e-02],
       [6.74505779e+01, 1.18127103e+02, 9.70000000e-01, ...,
        3.40600000e+01, 3.68000000e+00, 5.50000000e+00],
       [6.74505779e+01, 1.18127103e+02, 1.74000000e+01, ...,
        3.07000000e+01, 6.80000000e+00, 1.64000000e+01],
       ...,
       [2.29100000e+01, 6.57300000e+01, 3.45000000e+00, ...,
        3.09600000e+01, 1.00000000e-02, 1.00000000e-02],
       [1.66400000e+01, 4.99700000e+01, 4.05000000e+00, ...,
        2.83000000e+01, 0.00000000e+00, 0.00000000e+00],
       [1.50000000e+01, 6.60000000e+01, 4.00000000e-01, ...,
        1.70500000e+01, 3.28084030e+00, 8.70097208e+00]])
```

```
[36] from sklearn.metrics import mean_absolute_error as mae, mean_squared_error as mse,r2_score
```

```
[37] print(f"MSE:-{mse(ytest,ypred)}")
```

```
MSE:-3313.3544843736368
```

```
[38] print(f"R-squared:-{r2_score(ytest,ypred)}")
```

```
R-squared:-0.7982536798114572
```

REFERENCES :

1.Bose, R., D. R. R. S. and Sarddar (2020). Time series forecasting using double exponential smoothing for predicting the major ambient air pollutants, In Information and communication technology for sustainable development,Springer,Singapore pp. 603–613.

2.da Rocha, B. and de Sousa Junior, R. (2010). Identifying bank frauds using crisp-dm and decision trees., International Journal of Computer Science and Information Technology pp. 162–169.

3. Deepu B P, Dr. Ravindra P Rajput"Air Pollution Prediction using Machine Learning" IRJET Volume : 09 Issue -2022

4. Shreyas Simu,Varsha Turkar, Rohit Martires, "Air Pollution Prediction using Machine Learning", IEEE - 2020

5.D˚aderman, A. and Rosander, S. (2018). Evaluating frameworks for implementing machine learning in signal processing: A comparative study of crisp-dm, semma and kdd.

6.Erskine, J.R., P. G. M. B. and Grimaila, M. (2010). Developing cyberspace data understanding: Using crisp-dm for host-based ids feature mining., In Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research pp. 1–4.

7.Gocheva-Ilieva, S.G., I. A. V. D. and Boyadzhiev, D. (2014). Time series analysis and forecasting for air pollution in small urban area: an sarima and factor analysis approach., Stochastic environmental research and risk assessment 28(4): 1045–1060.

8. Aditya C R, Chandana R Deshmukh, Nayana D K,Praveen Gandhi Vidyavastu "Detection and Prediction of Air Pollution using Machine Learning Models" IJETT– volume 59 Issue 4 – 2018

9. JSeya, Dr. L. Sankari "Air Pollution Prediction by Deep Learning Model", IEEE – 2020

10. Venkat Rao Pasupuleti, Uhasri , Pavan Kalyan "Air Quality Prediction Of Data Log By Machine Learning" IEEE - 2020

1.

2.

3

Signature (Students) **Signature (Guide)**