**IBM NAAN MUDHALVAN – PHASE 4**

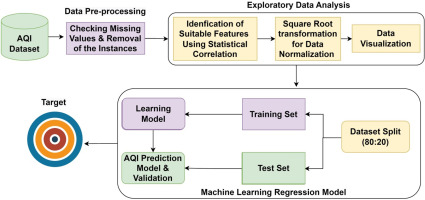
**DOMAIN : DATA SCIENCE**

**TITLE : Air Quality Analysis and Prediction Using Data Science**

**DATASET LINK :**  <https://www.kaggle.com/datasets/india/air-quality>

1.FEATURE ENGINEERING :

Sensors can track all the key pollution markers like particulate matter (PM1, PM2. 5, PM10), NO2, O3, SO2, H2S, NO, and CO gases. They also provide reliable data about key weather parameters such as temperature, humidity, air pressure, and wind from our sensors locations**.**



**PROGRAM :**

**#import**

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

plt.rcParams['figure.figsize'] = (10, 7)

*# Warnings*

import warnings

warnings.filterwarnings('ignore')

*# Input data files are available in the "../input/" directory.*

*# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory*

import os

print(os.listdir("../input"))

*# Any results you write to the current directory are saved as output.*

data=pd.read\_csv('../input/india-air-quality-data/data.csv',encoding="ISO-8859-1")

data.fillna(0, inplace=True)

data.head()

from mpl\_toolkits.basemap import Basemap

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

%config InlineBackend.figure\_format = 'retina'

m = Basemap(projection='mill',llcrnrlat=5,urcrnrlat=40, llcrnrlon=60,urcrnrlon=110,lat\_ts=20,resolution='c')

longitudes = dff["lon"].tolist()

latitudes = dff["lat"].tolist()

*#m = Basemap(width=12000000,height=9000000,projection='lcc',*

*#resolution=None,lat\_1=80.,lat\_2=55,lat\_0=80,lon\_0=-107.)*

x,y = m(longitudes,latitudes)

fig = plt.figure(figsize=(12,10))

plt.title("All affected areas")

m.plot(x, y, "o", markersize = 3, color = 'blue')

m.drawcoastlines()

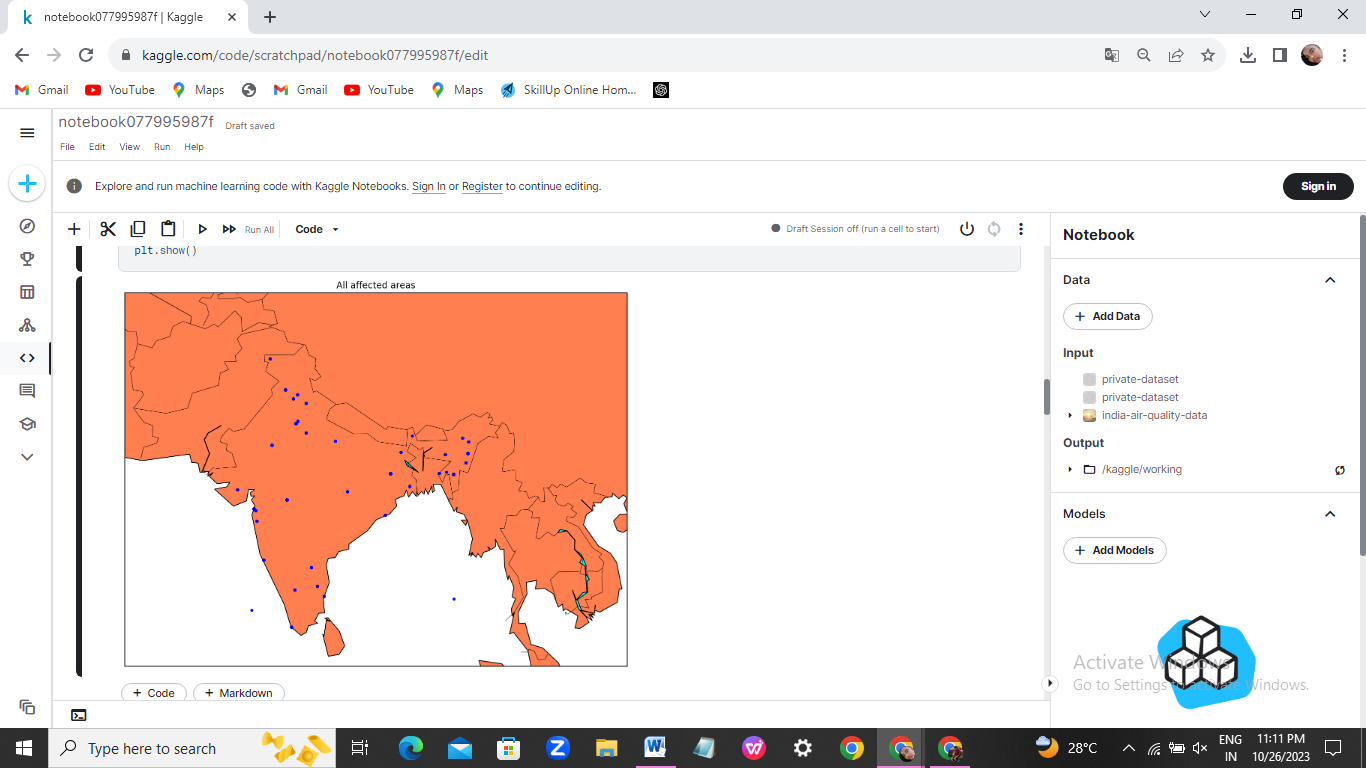
m.fillcontinents(color='coral',lake\_color='aqua')

m.drawmapboundary()

m.drawcountries()

plt.show()

**OUTPUT:**



*#Visualization of AQI across india*

data['date'] = pd.to\_datetime(data['date'],format='%Y-%m-**%d**') *# date parse*

data['year'] = data['date'].dt.year *# year*

data['year'] = data['year'].fillna(0.0).astype(int)

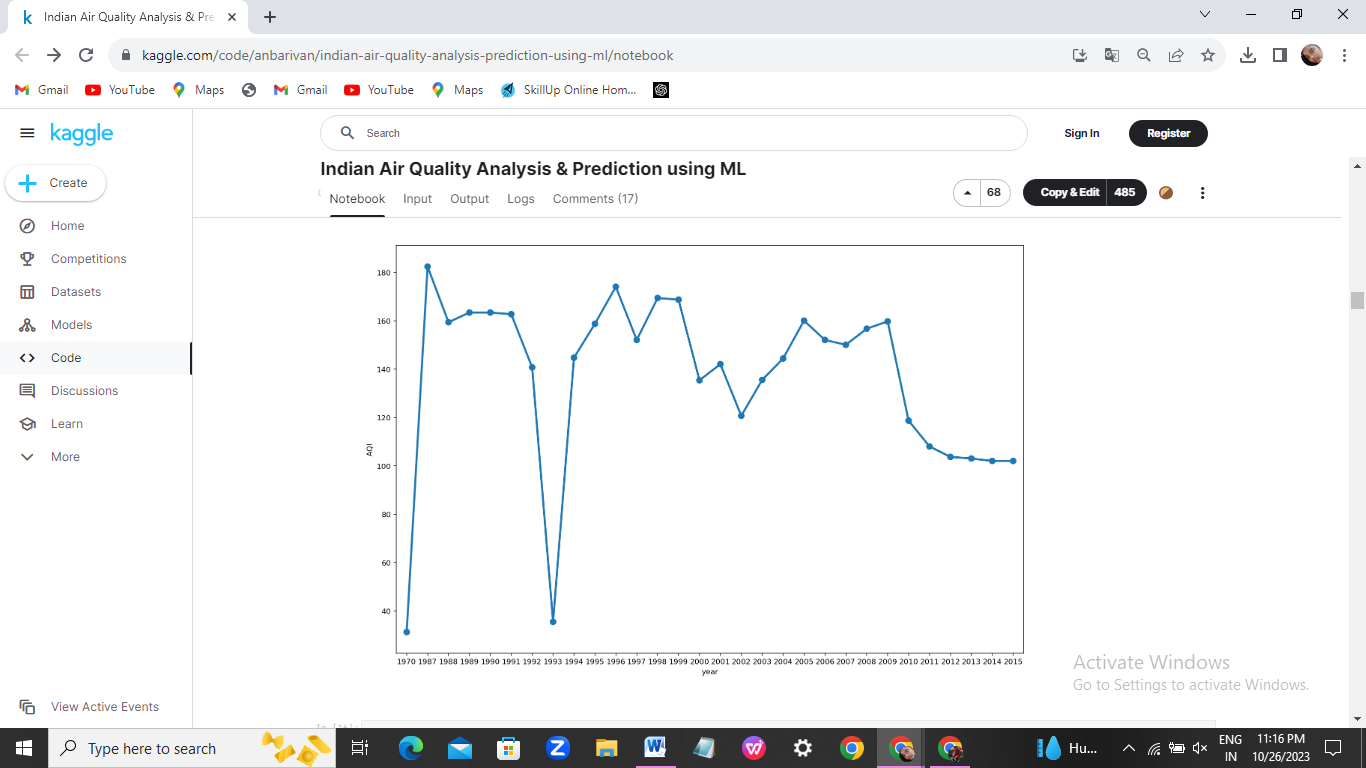
data = data[(data['year']>0)]

df = data[['AQI','year','state']].groupby(["year"]).median().reset\_index().sort\_values(by='year',ascending=False)

f,ax=plt.subplots(figsize=(15,10))

sns.pointplot(x='year', y='AQI', data=df)

**OUTPUT:**



*#exctracting knowledge about data*

*#spliting dataframes into test and train*

n = df.shape[0]

train\_size = 0.65

features\_dataframe = df.sort\_values('date')

train = df.iloc[:int(n \* train\_size)]

test = df.iloc[int(n \* train\_size):]

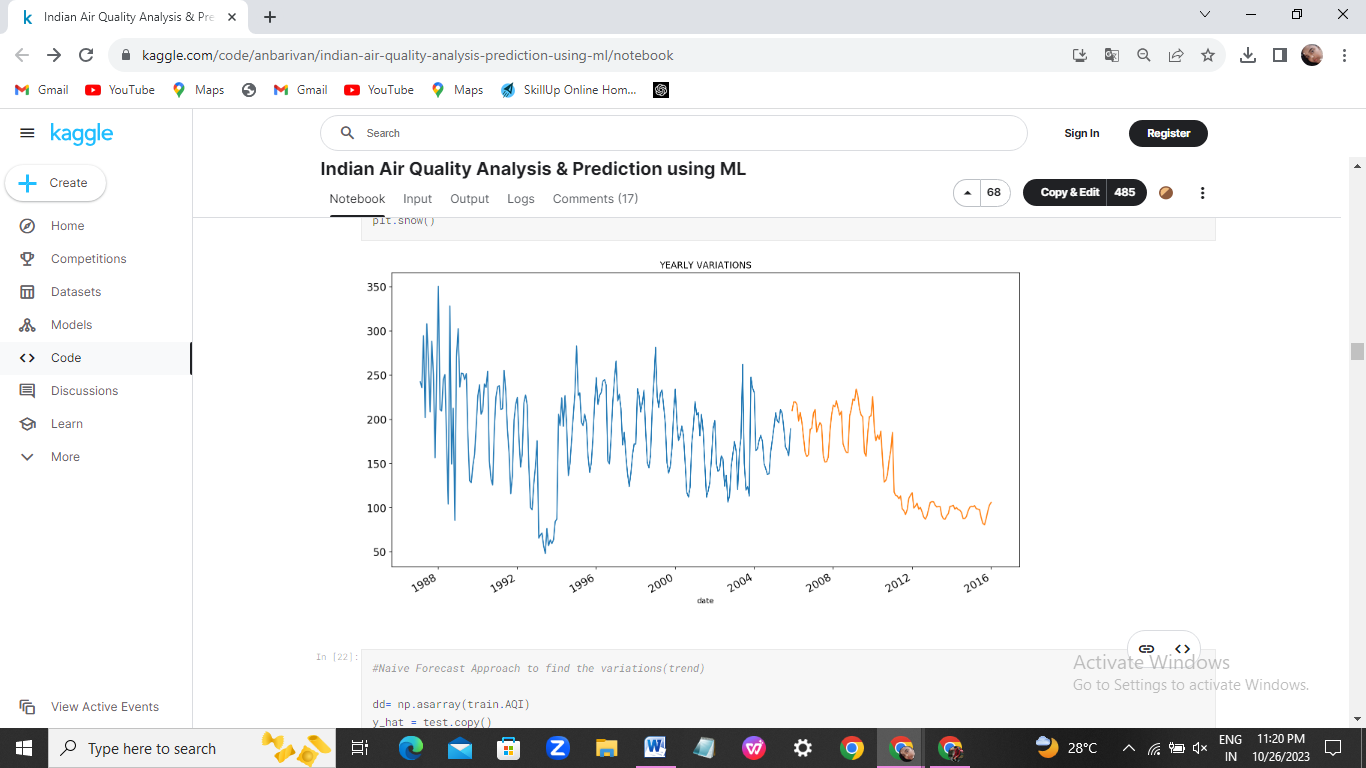
*#plotting the yearly variations of AQI*

train.AQI.plot(figsize=(15,8), title= 'YEARLY VARIATIONS', fontsize=14)

test.AQI.plot(figsize=(15,8), title= 'YEARLY VARIATIONS', fontsize=14)

plt.show()

**OUTPUT:**



*#various statmodel to identity huge variations od data values*

import statsmodels.api as sm

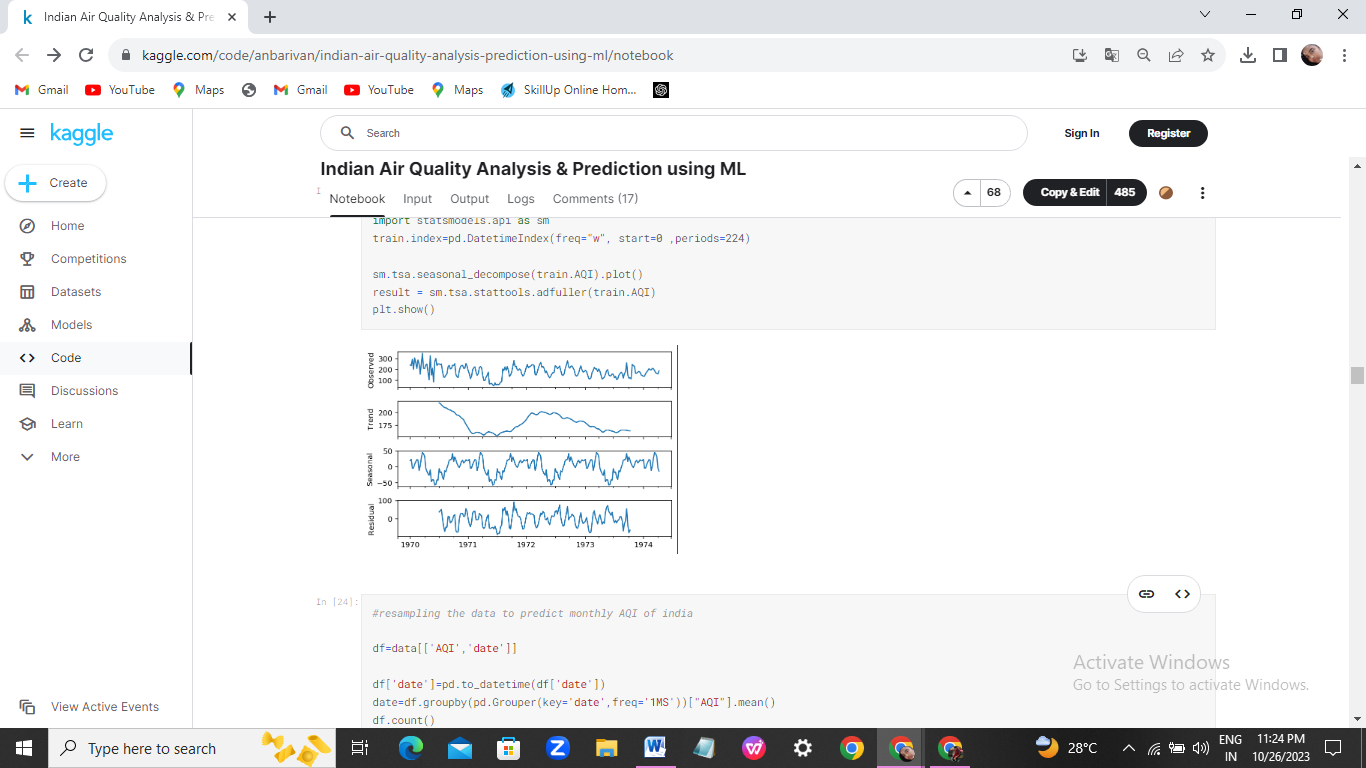
train.index=pd.DatetimeIndex(freq="w", start=0 ,periods=224)

sm.tsa.seasonal\_decompose(train.AQI).plot()

result = sm.tsa.stattools.adfuller(train.AQI)

plt.show()

**OUTPUT:**



*#resampling the data to predict monthly AQI of india*

df=data[['AQI','date']]

df['date']=pd.to\_datetime(df['date'])

date=df.groupby(pd.Grouper(key='date',freq='1MS'))["AQI"].mean()

df.count()

**OUTPUT:**

AQI 346

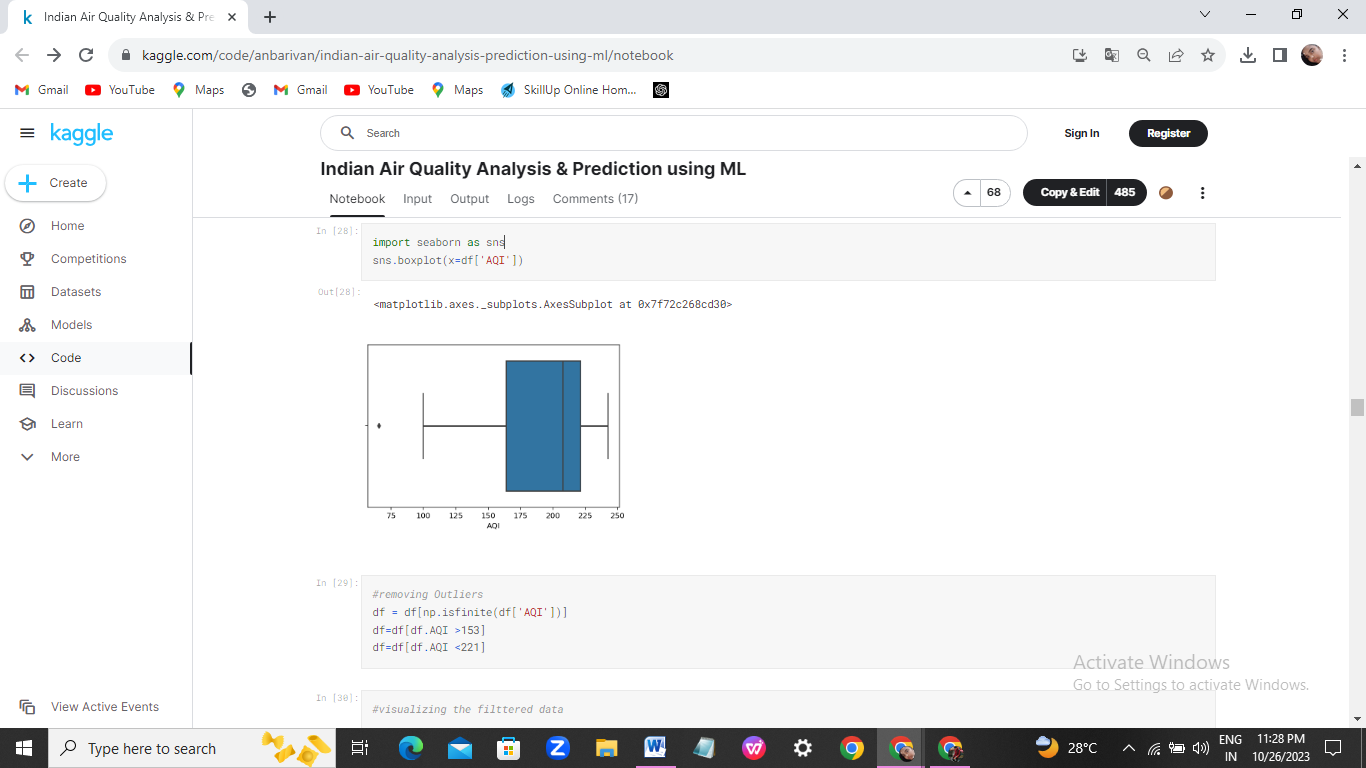
date 346

dtype: int64

import seaborn as sns

sns.boxplot(x=df['AQI'])

**OUTPUT:**



*#removing Outliers*

df = df[np.isfinite(df['AQI'])]

df=df[df.AQI >153]

df=df[df.AQI <221]

*#visualizing the filttered data*

year=df['year'].values

AQI=df['AQI'].values

df['AQI']=pd.to\_numeric(df['AQI'],errors='coerce')

df['year']=pd.to\_numeric(df['year'],errors='coerce')

import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = (20.0, 10.0)

from mpl\_toolkits.mplot3d import Axes3D

fig = plt.figure()

ax = Axes3D(fig)

ax.scatter(year,AQI, color='red')

plt.show()

*#scatter plot of data points*

cols =['year']

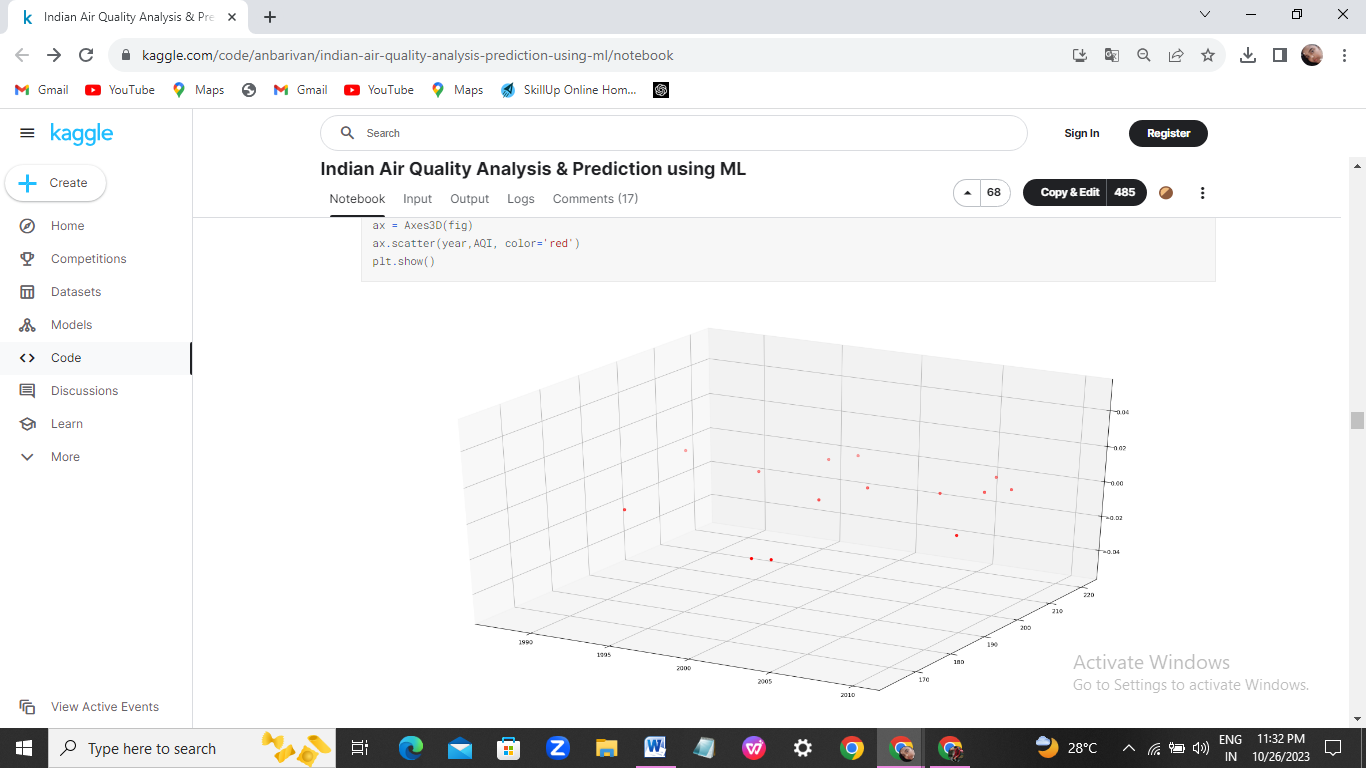
y = df['AQI']

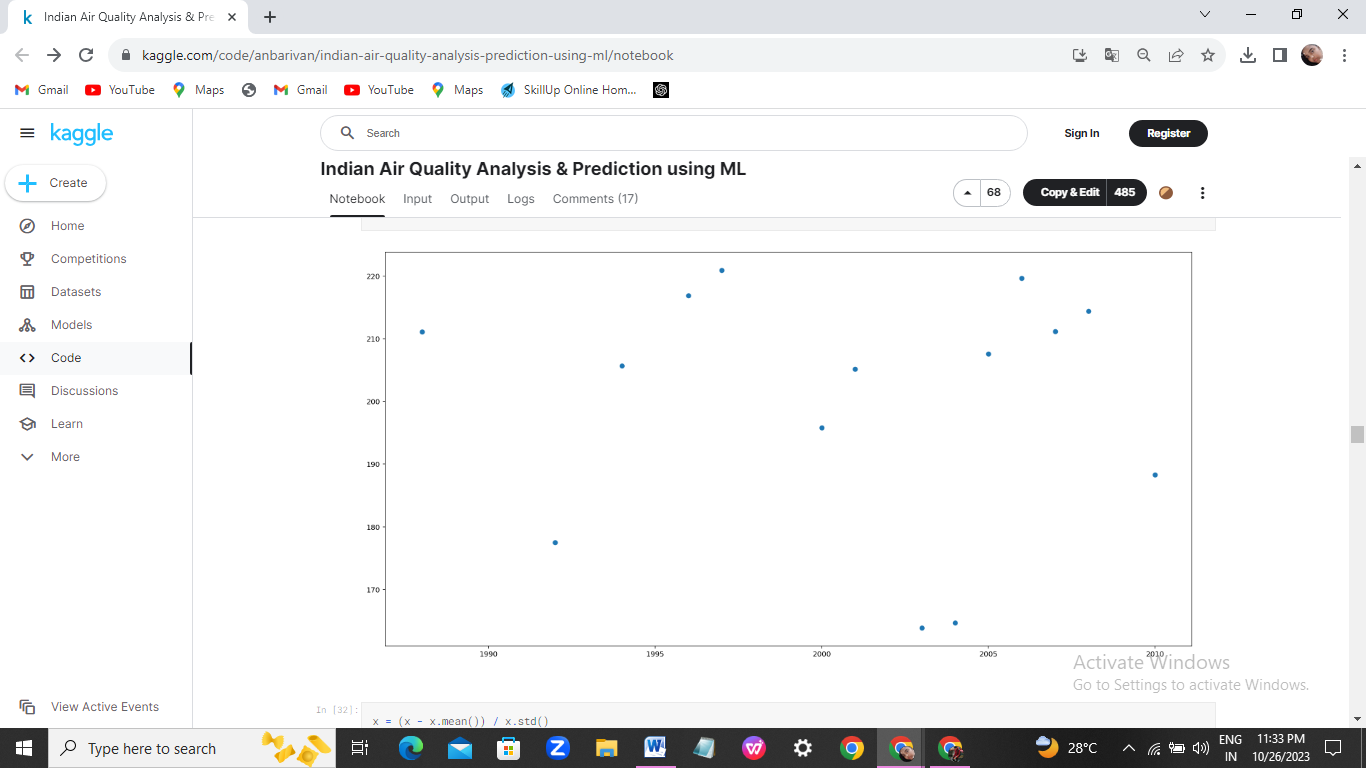
x=df[cols]

plt.scatter(x,y)

plt.show()

**OUTPUT:**





x = (x - x.mean()) / x.std()

x = np.c\_[np.ones(x.shape[0]), x]

x

**OUTPUT:**

array([[ 1. , 1.40346276],

[ 1. , 1.09883519],

[ 1. , 0.9465214 ],

[ 1. , 0.79420761],

[ 1. , 0.64189382],

[ 1. , 0.48958003],

[ 1. , 0.33726625],

[ 1. , 0.03263867],

[ 1. , -0.11967512],

[ 1. , -0.57661648],

[ 1. , -0.72893027],

[ 1. , -1.03355785],

[ 1. , -1.33818543],

[ 1. , -1.94744058]])

*#Plotting the cost function...*

plt.title('Cost Function J')

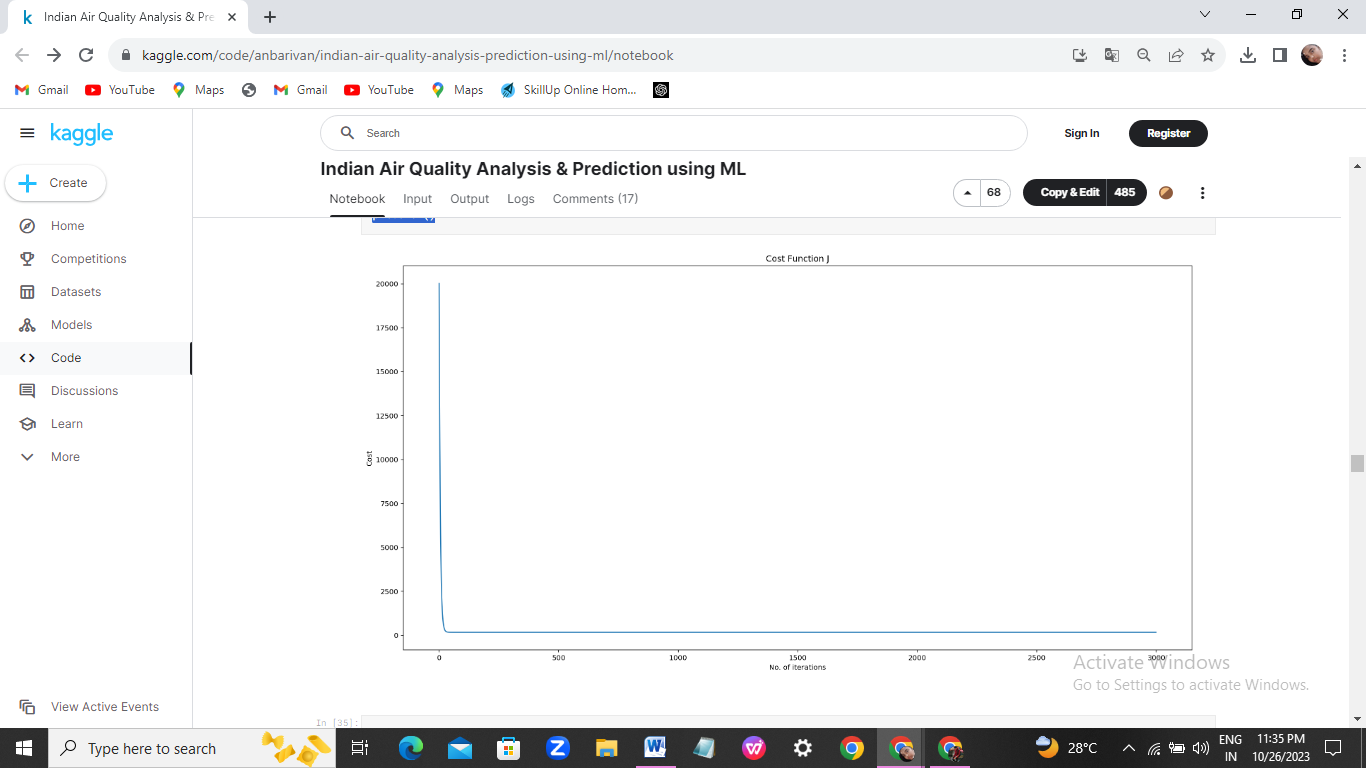
plt.xlabel('No. of iterations')

plt.ylabel('Cost')

plt.plot(past\_costs)

plt.show()

**OUTPUT:**



*#calculating the root mean squared error for the predicted AQi values*

from sklearn import metrics

print(np.sqrt(metrics.mean\_squared\_error(y,y\_pred)))

x\_axis=x.year

y\_axis=x.Actual

y1\_axis=x.Predicted

plt.plot(x\_axis,y\_axis)

plt.plot(x\_axis,y1\_axis)

plt.title("Actual vs Predicted",fontsize=20)

plt.legend(["actual ","predicted"])

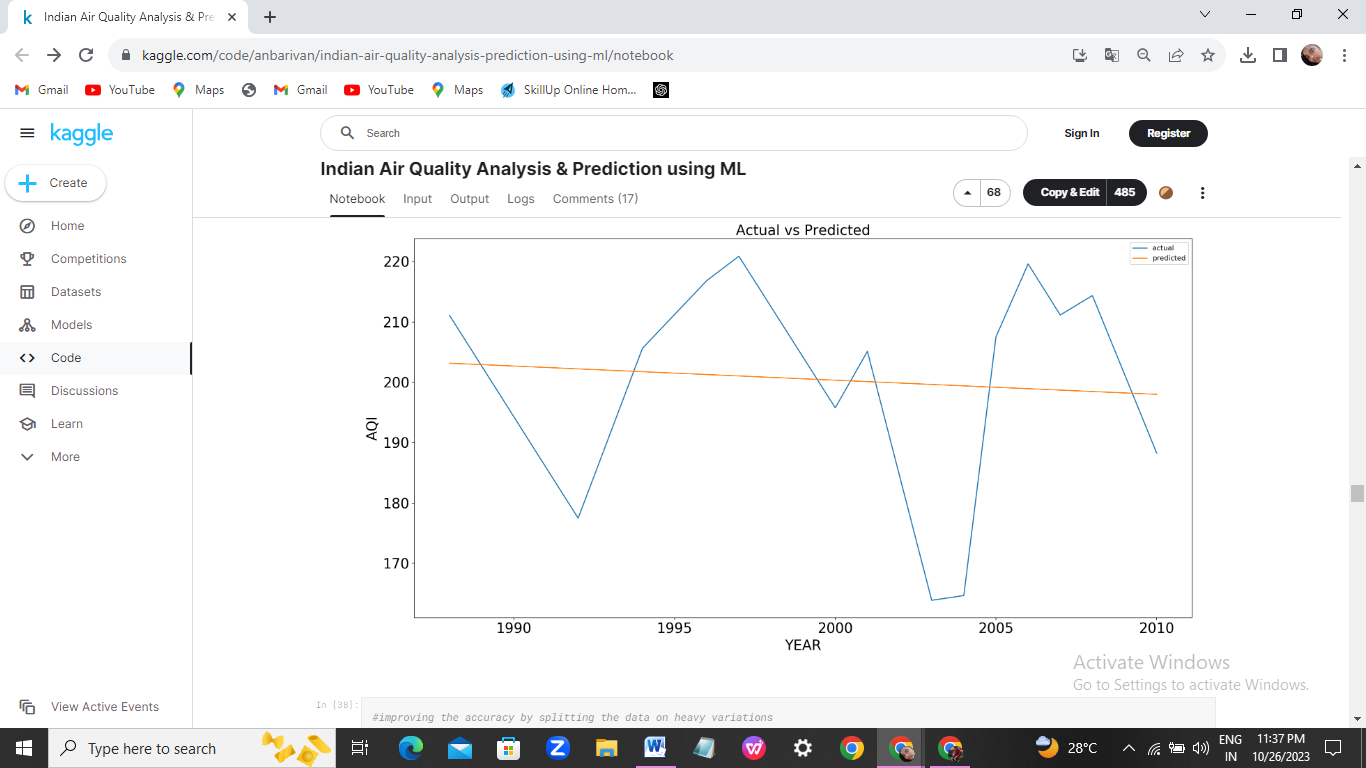
plt.xlabel("YEAR",fontsize=20)

plt.ylabel("AQI",fontsize=20)

plt.tick\_params(labelsize=20)

plt.show()

**OUTPUT:**



*#plotting the actual and predicted results*

x\_axis=x.year

y\_axis=x.Actual

y1\_axis=x.Predicted

plt.plot(x\_axis,y\_axis)

plt.plot(x\_axis,y1\_axis)

plt.title("Actual vs Predicted",fontsize=20)

plt.legend(["actual ","predicted"])

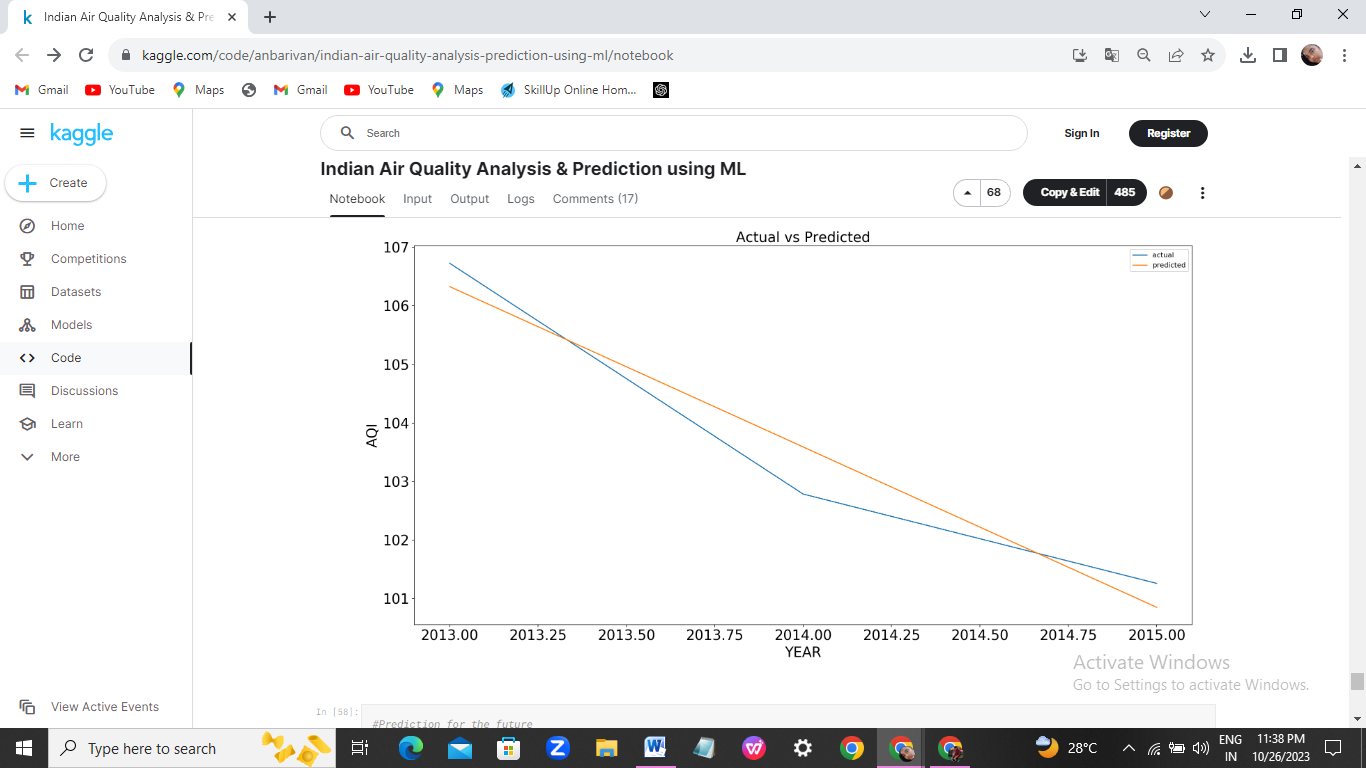
plt.xlabel("YEAR",fontsize=20)

plt.ylabel("AQI",fontsize=20)

plt.tick\_params(labelsize=20)

plt.show()

**OUTPUT:**



*#Prediction for the future*

from sklearn.preprocessing import MinMaxScaler

*#normalization*

scaler=MinMaxScaler(feature\_range=(-1,1))

scaler.fit(data)

x=scaler.transform(data)

**4. INTERPRETATION :**

The higher the AQI value, the greater the level of air pollution and the greater the health concern. For example, an AQI value of 50 or below represents good air quality, while an AQI value over 300 represents hazardous air quality.