NAME : M.BHARATHI

COLLEGE CODE : 4101

DEPARTMENT : CSE

DATA ANALYTICS WITH COGNOS

PROJECT TITLE : AIR QUALITY ANALYSIS.

PHASE 4 : Development part 2

AIM: To perform air quality analysis and create visualizations and also to calculate the SO2 and NO2 levels.

VISUALIZATION PART:

import warnings

warnings.filterwarnings('ignore')

import pandas as pd

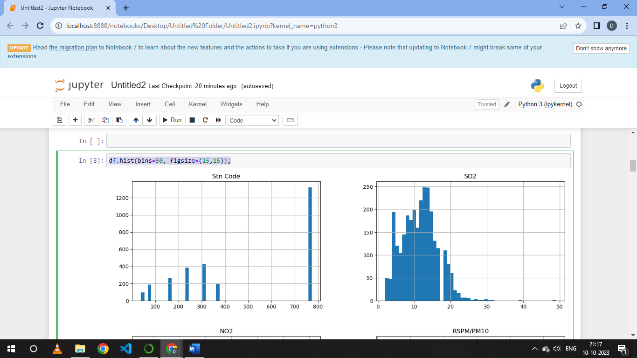
import matplotlib.pyplot as plt

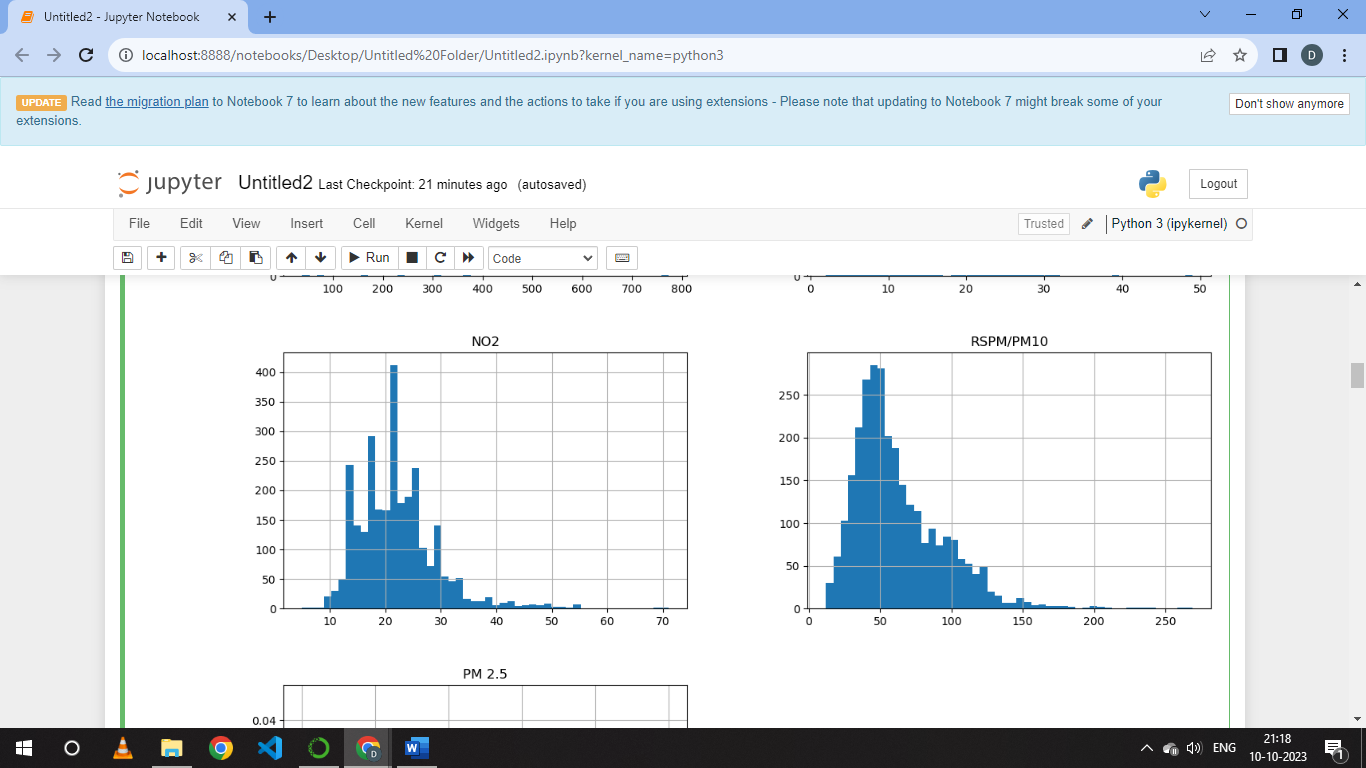
import seaborn as sns

import numpy as np

df=pd.read\_csv('weather.csv')

df.hist(bins=50, figsize=(15,15))





import plotly.express as px

fig1 = px.scatter(df,x='City/Town/Village/Area',y='SO2')

fig1.show()

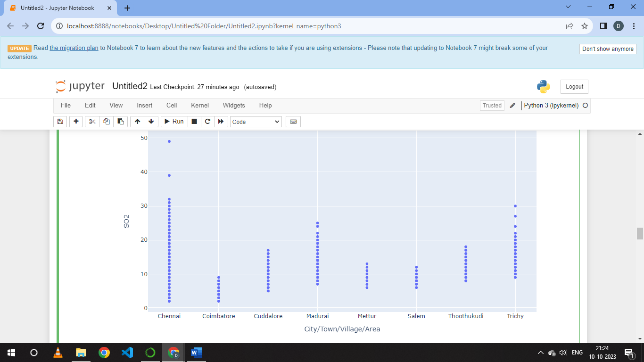
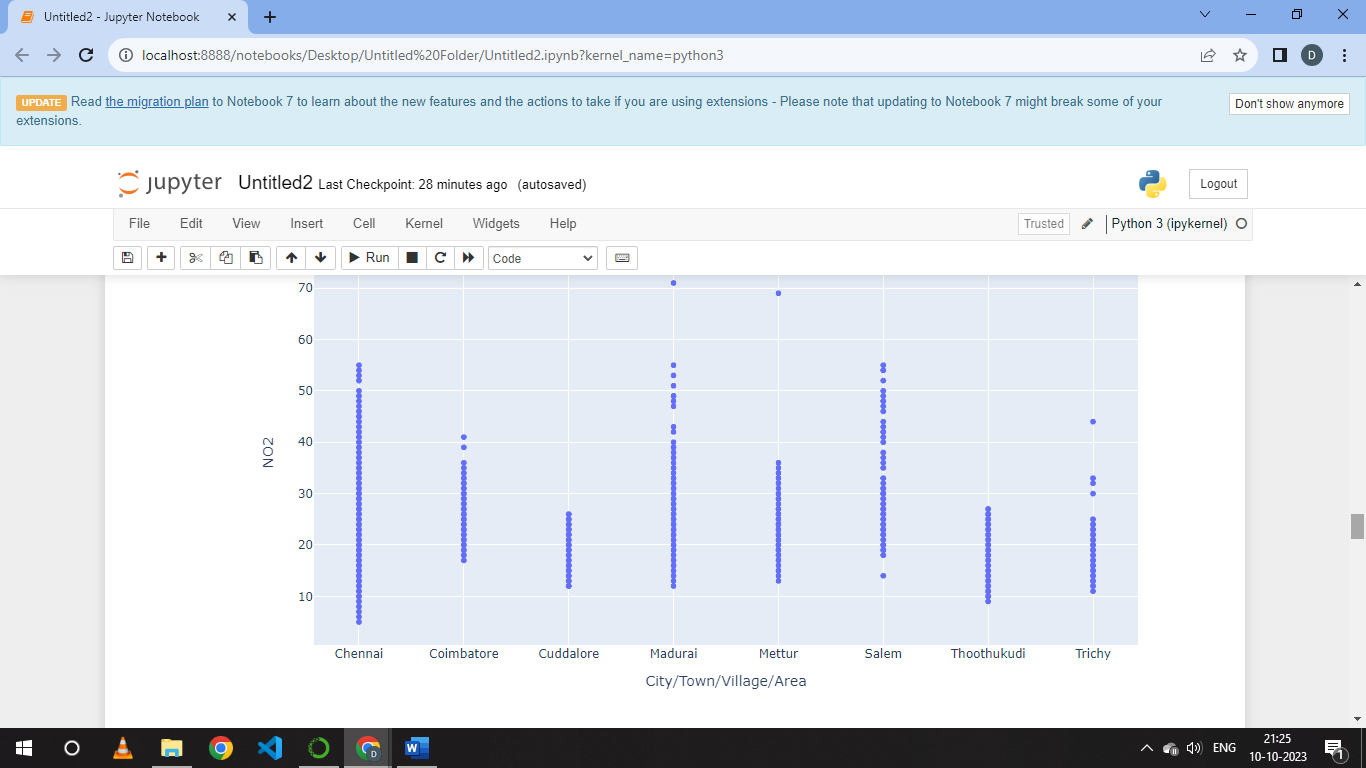


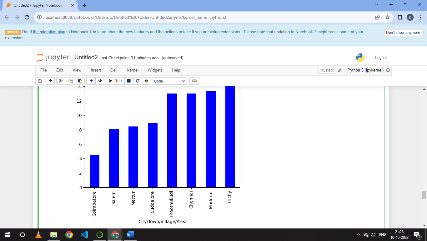
fig2 = px.scatter(df,x='City/Town/Village/Area',y='NO2')

fig2.show()



df[['SO2','City/Town/Village/Area']].groupby(["City/Town/Village/Area"]).mean().sort\_values(by='SO2').head(20).plot.bar(color='b')

plt.show()



CALCULATION PART :

1.Calculating SO2 :

def cal\_SOi(SO2):

si=0

if (SO2<=40):

si= SO2\*(50/40)

elif (SO2>40 and SO2<=80):

si= 50+(SO2-40)\*(50/40)

elif (SO2>80 and SO2<=380):

si= 100+(SO2-80)\*(100/300)

elif (SO2>380 and SO2<=800):

si= 200+(SO2-380)\*(100/420)

elif (SO2>800 and SO2<=1600):

si= 300+(SO2-800)\*(100/800)

elif (SO2>1600):

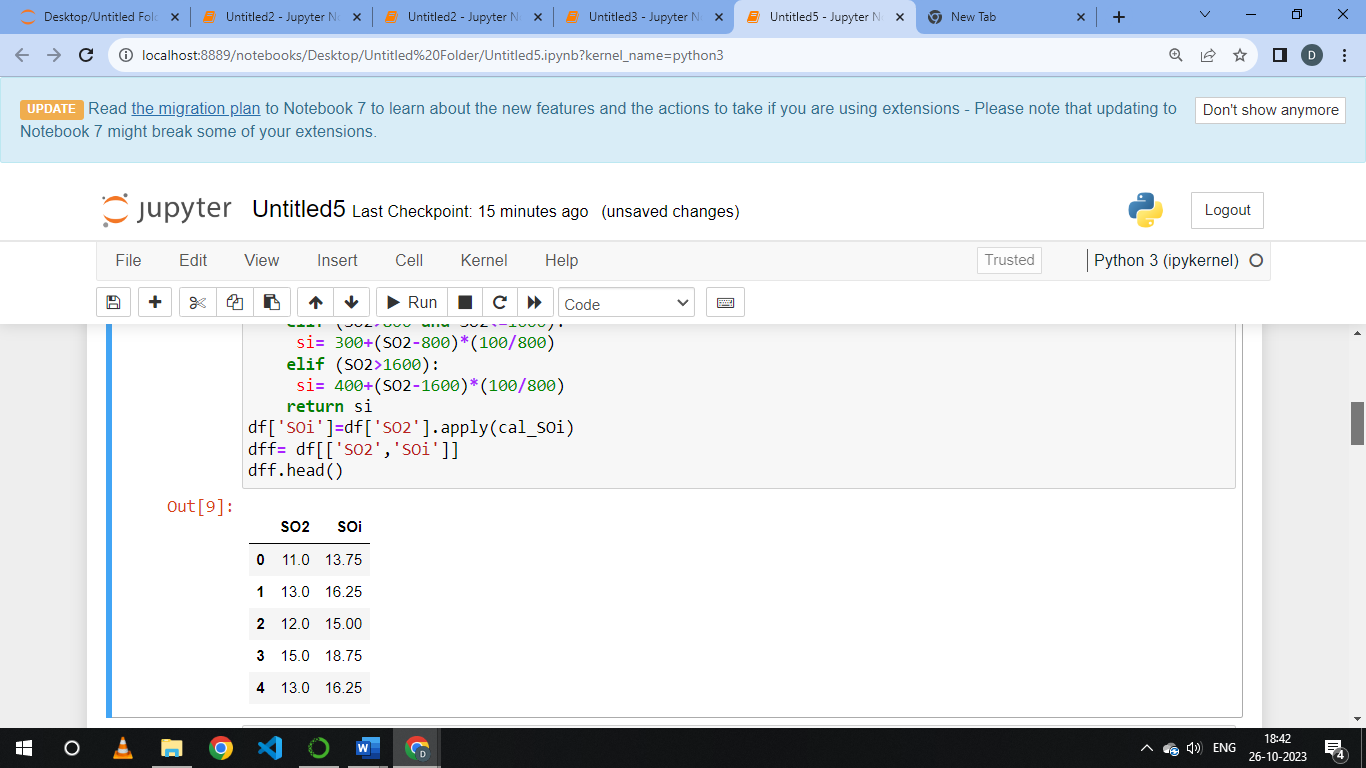
si= 400+(SO2-1600)\*(100/800)

return si

df['SOi']=df['SO2'].apply(cal\_SOi)

dff= df[['SO2','SOi']]

dff.head()



2. Calculating NO2 :

def cal\_NOi(NO2):

ni=0

if (NO2<=40):

ni= NO2\*(50/40)

elif (NO2>40 and NO2<=80):

ni= 50+(NO2-40)\*(50/40)

elif (NO2>80 and NO2<=380):

ni= 100+(NO2-80)\*(100/300)

elif (NO2>380 and NO2<=800):

ni= 200+(NO2-380)\*(100/420)

elif (NO2>800 and NO2<=1600):

ni= 300+(NO2-800)\*(100/800)

elif (NO2>1600):

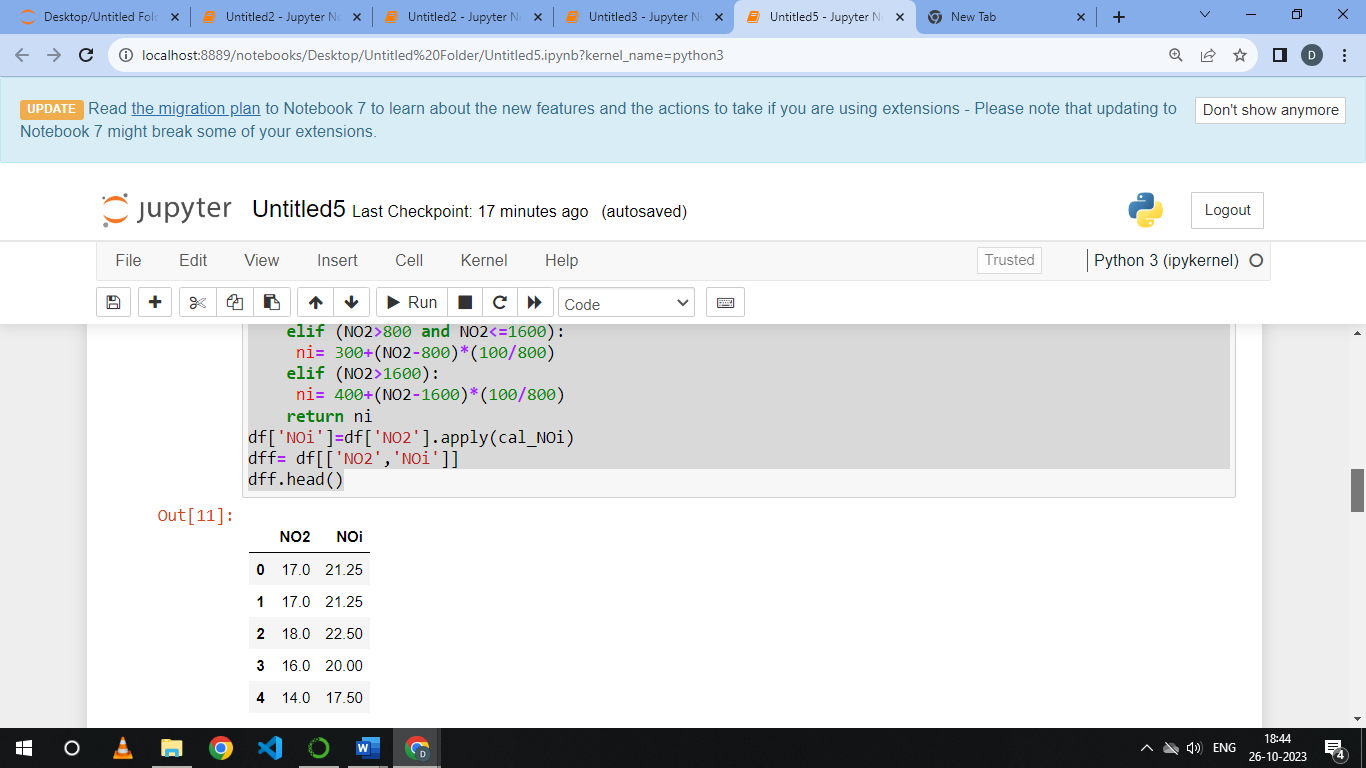
ni= 400+(NO2-1600)\*(100/800)

return ni

df['NOi']=df['NO2'].apply(cal\_NOi)

dff= df[['NO2','NOi']]

dff.head()



TREND MOST AFFECTED CITY :

city = df.groupby('City/Town/Village/Area')[‘RSPM/PM10’].sum().reset\_index()

most\_affected\_city = city.loc[city[‘RSPM/PM10'].idxmax()]

print("The most affected city is:", most\_affected\_city['City/Town/Village/Area']) 