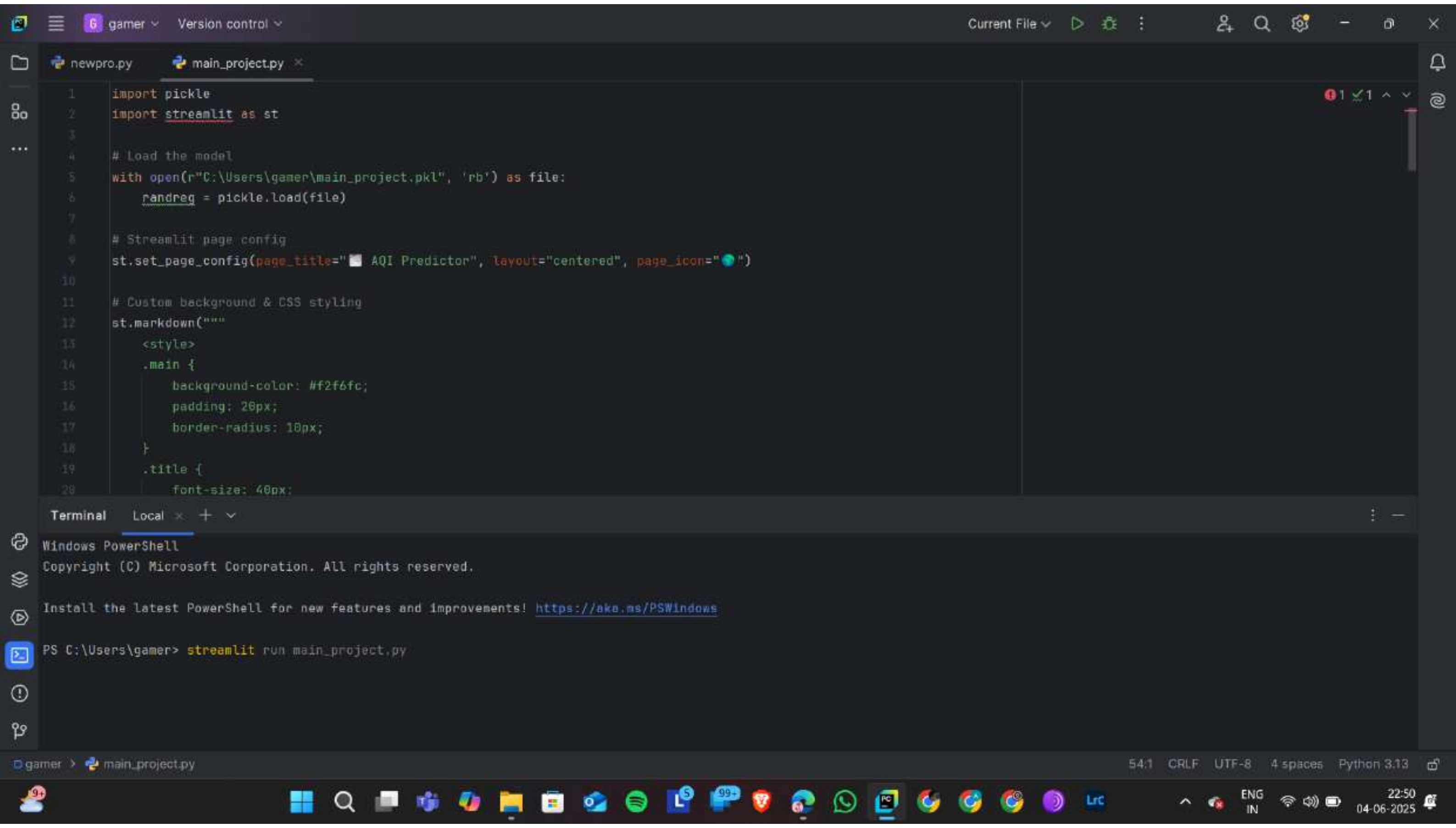
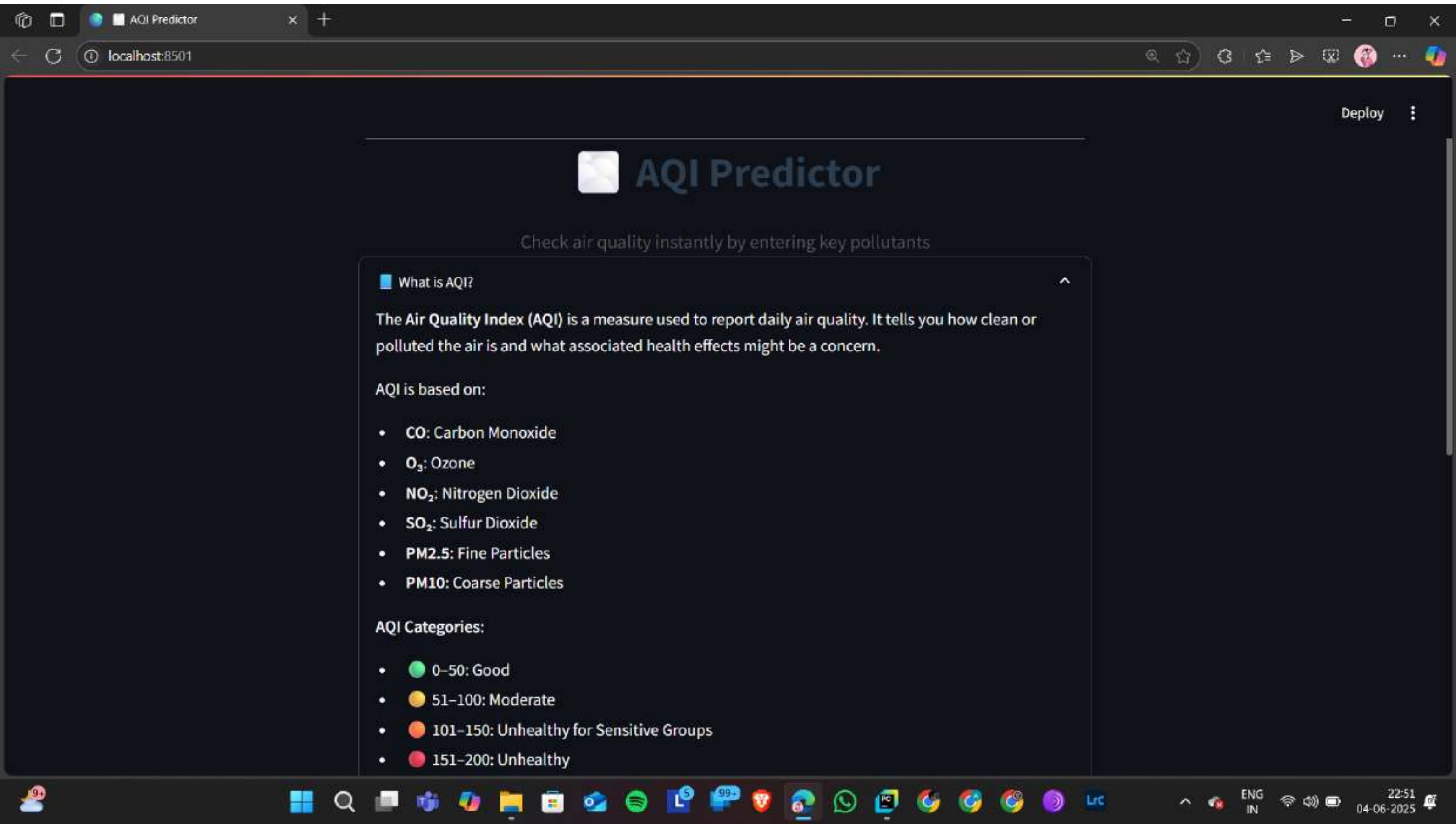


```python
import pickle
import streamlit as st

# Load the model
with open(r"C:\Users\gamer\main_project.pkl", 'rb') as file:
    randreg = pickle.load(file)

# Streamlit page config
st.set_page_config(page_title="🌫️ AQI Predictor", layout="centered", page_icon="🌍")

# Custom background & CSS styling
st.markdown("""
    <style>
    .main {
        background-color: #f2f6fc;
        padding: 20px;
        border-radius: 10px;
    }
    .title {
        font-size: 40px;
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\gamer> streamlit run main_project.py
```



AQI Predictor

Check air quality instantly by entering key pollutants

📘 What is AQI?

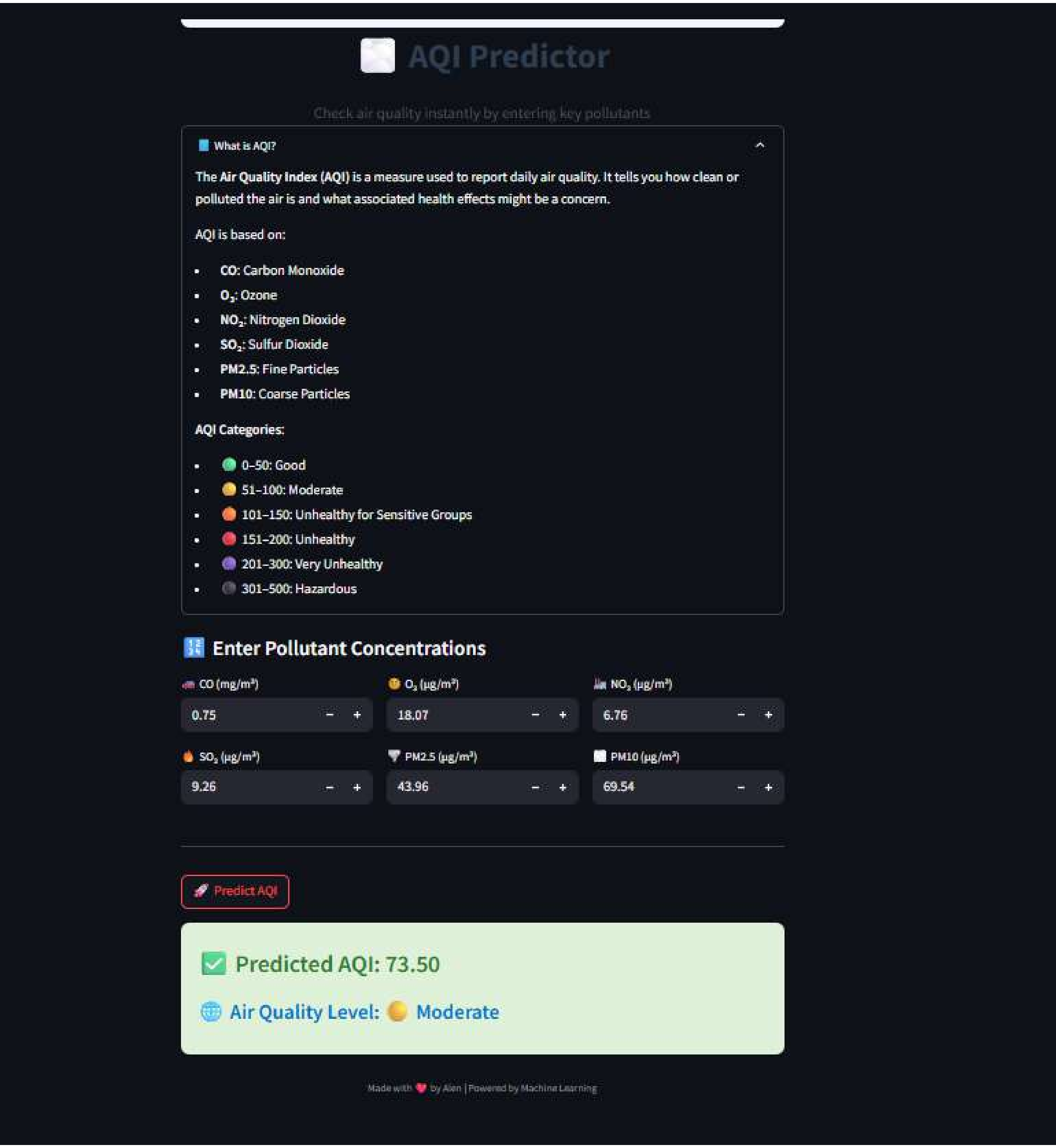
The **Air Quality Index (AQI)** is a measure used to report daily air quality. It tells you how clean or polluted the air is and what associated health effects might be a concern.

AQI is based on:

- **CO**: Carbon Monoxide
- **O$_3$**: Ozone
- **NO$_2$**: Nitrogen Dioxide
- **SO$_2$**: Sulfur Dioxide
- **PM2.5**: Fine Particles
- **PM10**: Coarse Particles

AQI Categories:

- 🟢 0–50: Good
- 🟡 51–100: Moderate
- 🟠 101–150: Unhealthy for Sensitive Groups
- 🔴 151–200: Unhealthy



AQI Predictor

Check air quality instantly by entering key pollutants

What is AQI?

The **Air Quality Index (AQI)** is a measure used to report daily air quality. It tells you how clean or polluted the air is and what associated health effects might be a concern.

AQI is based on:

- **CO**: Carbon Monoxide
- **O_3**: Ozone
- **NO_2**: Nitrogen Dioxide
- **SO_2**: Sulfur Dioxide
- **PM2.5**: Fine Particles
- **PM10**: Coarse Particles

AQI Categories:

- 🟢 0–50: Good
- 🟡 51–100: Moderate
- 🟠 101–150: Unhealthy for Sensitive Groups
- 🔴 151–200: Unhealthy
- 🟣 201–300: Very Unhealthy
- ⚫ 301–500: Hazardous

Enter Pollutant Concentrations

CO (mg/m³): 0.75

O_3 (µg/m³): 18.07

NO_2 (µg/m³): 6.76

SO_2 (µg/m³): 9.26

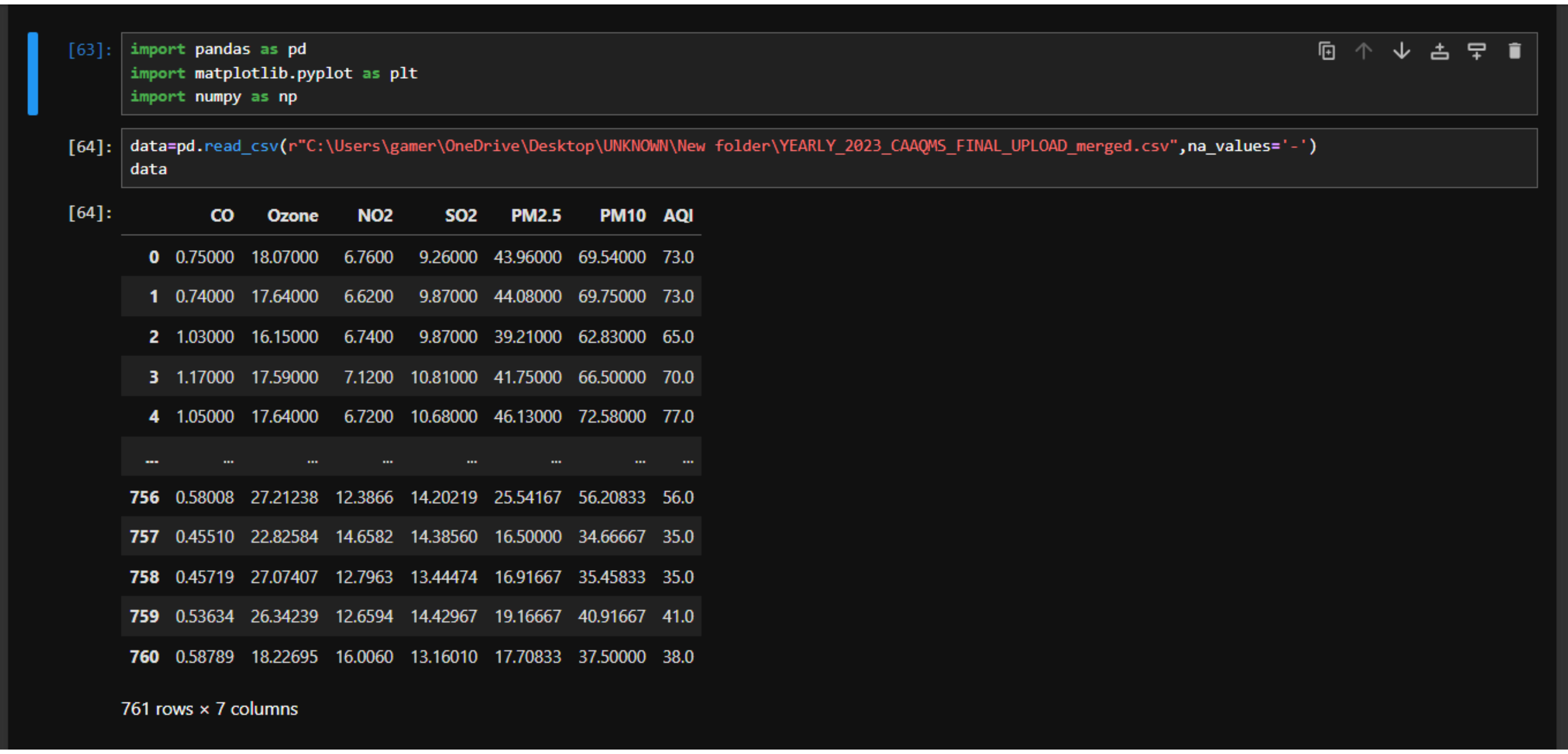
PM2.5 (µg/m³): 43.96

PM10 (µg/m³): 69.54

✅ Predicted AQI: 73.50

Air Quality Level: 🟡 Moderate

Made with ❤ by Alen | Powered by Machine Learning



[63]:

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

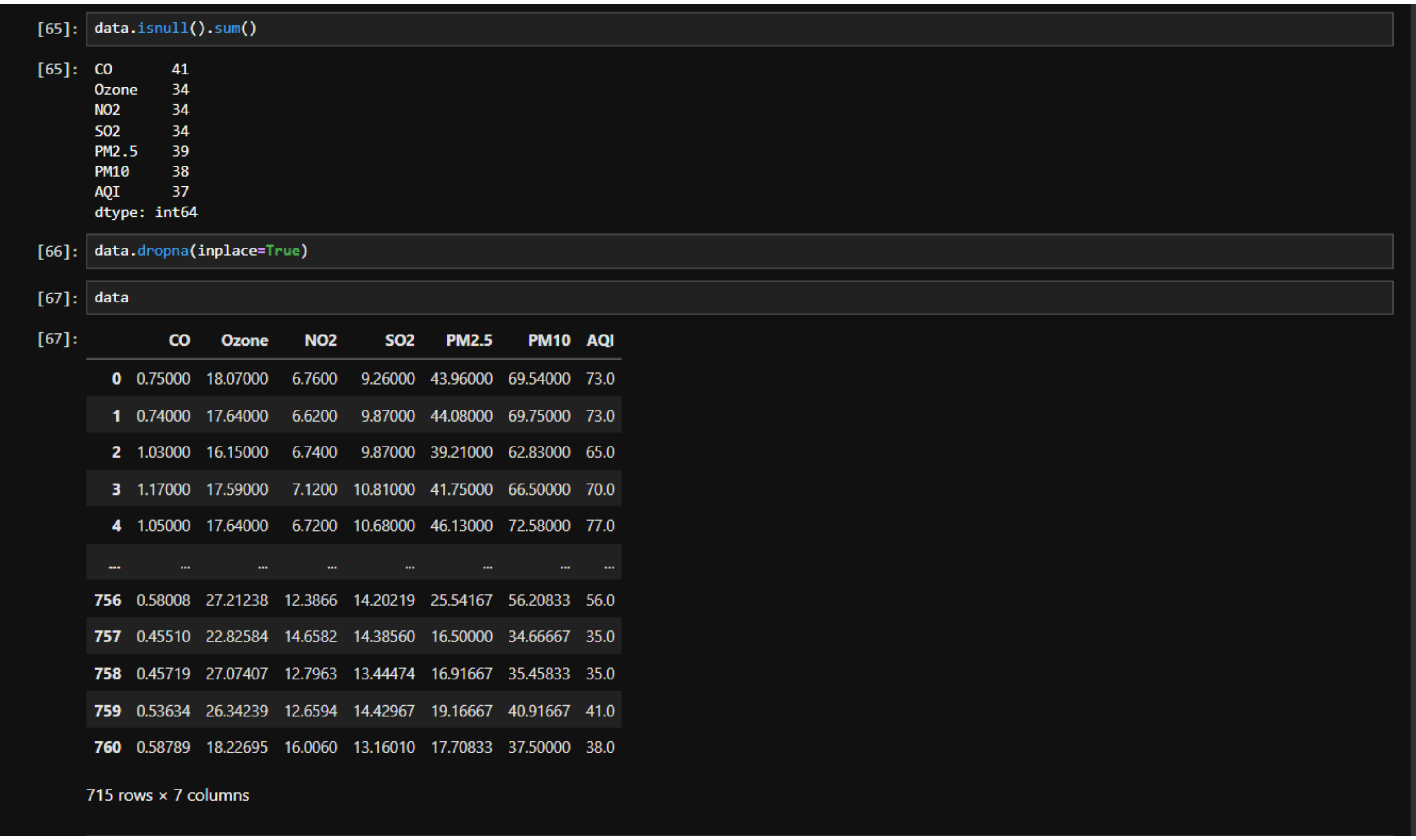
[64]:

```
data=pd.read_csv(r"C:\Users\gamer\OneDrive\Desktop\UNKNOWN\New folder\YEARLY_2023_CAAQMS_FINAL_UPLOAD_merged.csv",na_values='-')
data
```

[64]:

	CO	Ozone	NO2	SO2	PM2.5	PM10	AQI
0	0.75000	18.07000	6.7600	9.26000	43.96000	69.54000	73.0
1	0.74000	17.64000	6.6200	9.87000	44.08000	69.75000	73.0
2	1.03000	16.15000	6.7400	9.87000	39.21000	62.83000	65.0
3	1.17000	17.59000	7.1200	10.81000	41.75000	66.50000	70.0
4	1.05000	17.64000	6.7200	10.68000	46.13000	72.58000	77.0
...	...	...	...	...	...	...	...
756	0.58008	27.21238	12.3866	14.20219	25.54167	56.20833	56.0
757	0.45510	22.82584	14.6582	14.38560	16.50000	34.66667	35.0
758	0.45719	27.07407	12.7963	13.44474	16.91667	35.45833	35.0
759	0.53634	26.34239	12.6594	14.42967	19.16667	40.91667	41.0
760	0.58789	18.22695	16.0060	13.16010	17.70833	37.50000	38.0

761 rows × 7 columns



```
[65]: data.isnull().sum()
```

```
[65]: CO       41
      Ozone    34
      NO2      34
      SO2      34
      PM2.5    39
      PM10     38
      AQI      37
      dtype: int64
```

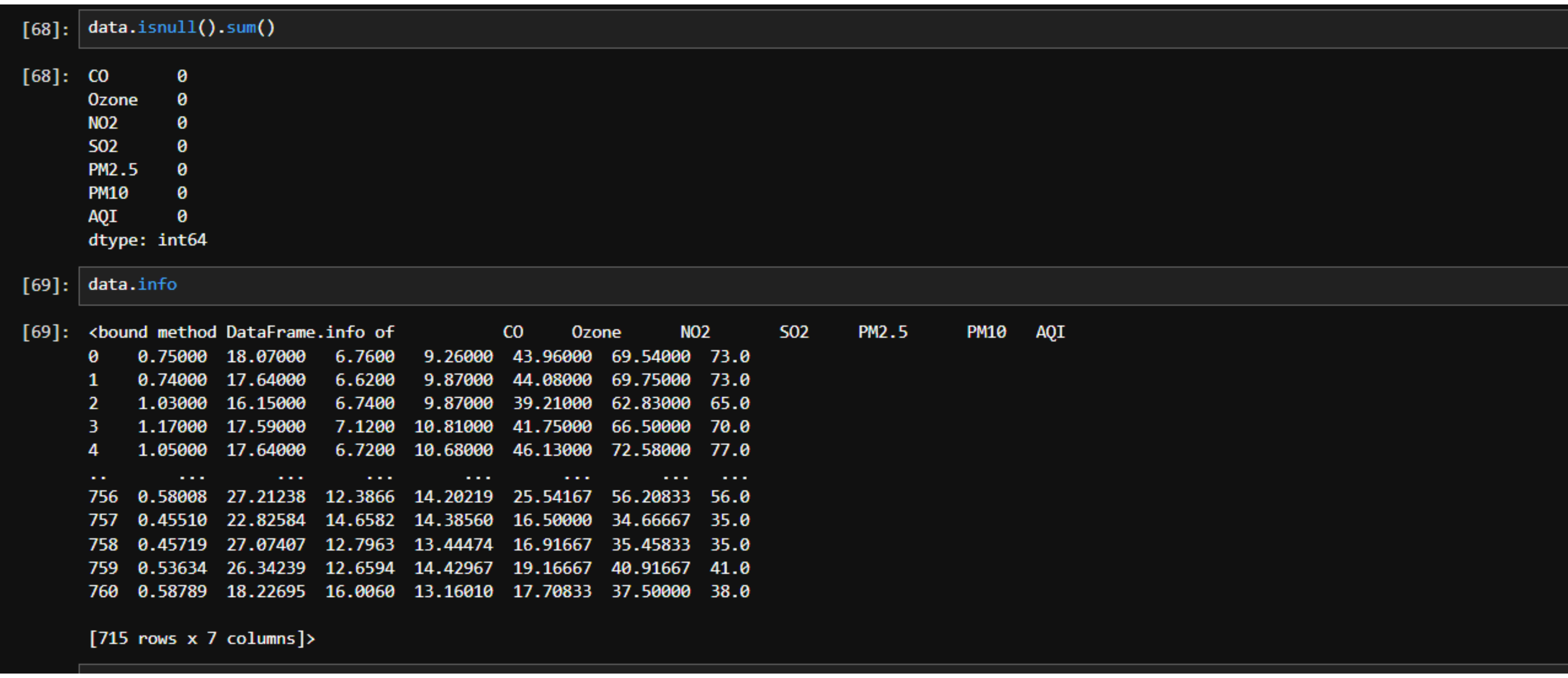
```
[66]: data.dropna(inplace=True)
```

```
[67]: data
```

[67]:

	CO	Ozone	NO2	SO2	PM2.5	PM10	AQI
0	0.75000	18.07000	6.7600	9.26000	43.96000	69.54000	73.0
1	0.74000	17.64000	6.6200	9.87000	44.08000	69.75000	73.0
2	1.03000	16.15000	6.7400	9.87000	39.21000	62.83000	65.0
3	1.17000	17.59000	7.1200	10.81000	41.75000	66.50000	70.0
4	1.05000	17.64000	6.7200	10.68000	46.13000	72.58000	77.0
...	...	...	...	...	...	...	...
756	0.58008	27.21238	12.3866	14.20219	25.54167	56.20833	56.0
757	0.45510	22.82584	14.6582	14.38560	16.50000	34.66667	35.0
758	0.45719	27.07407	12.7963	13.44474	16.91667	35.45833	35.0
759	0.53634	26.34239	12.6594	14.42967	19.16667	40.91667	41.0
760	0.58789	18.22695	16.0060	13.16010	17.70833	37.50000	38.0

715 rows × 7 columns

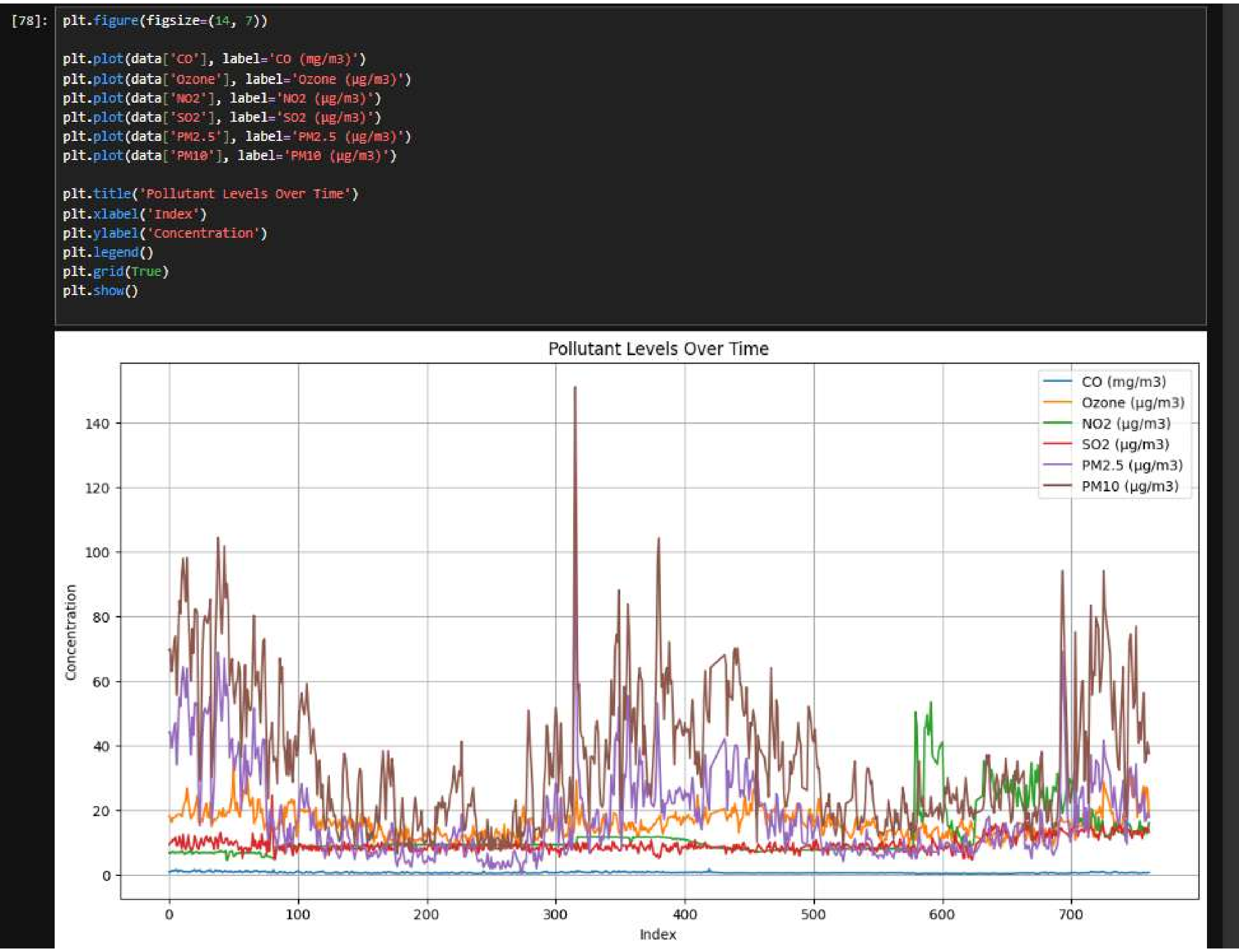


```
[68]: data.isnull().sum()
```

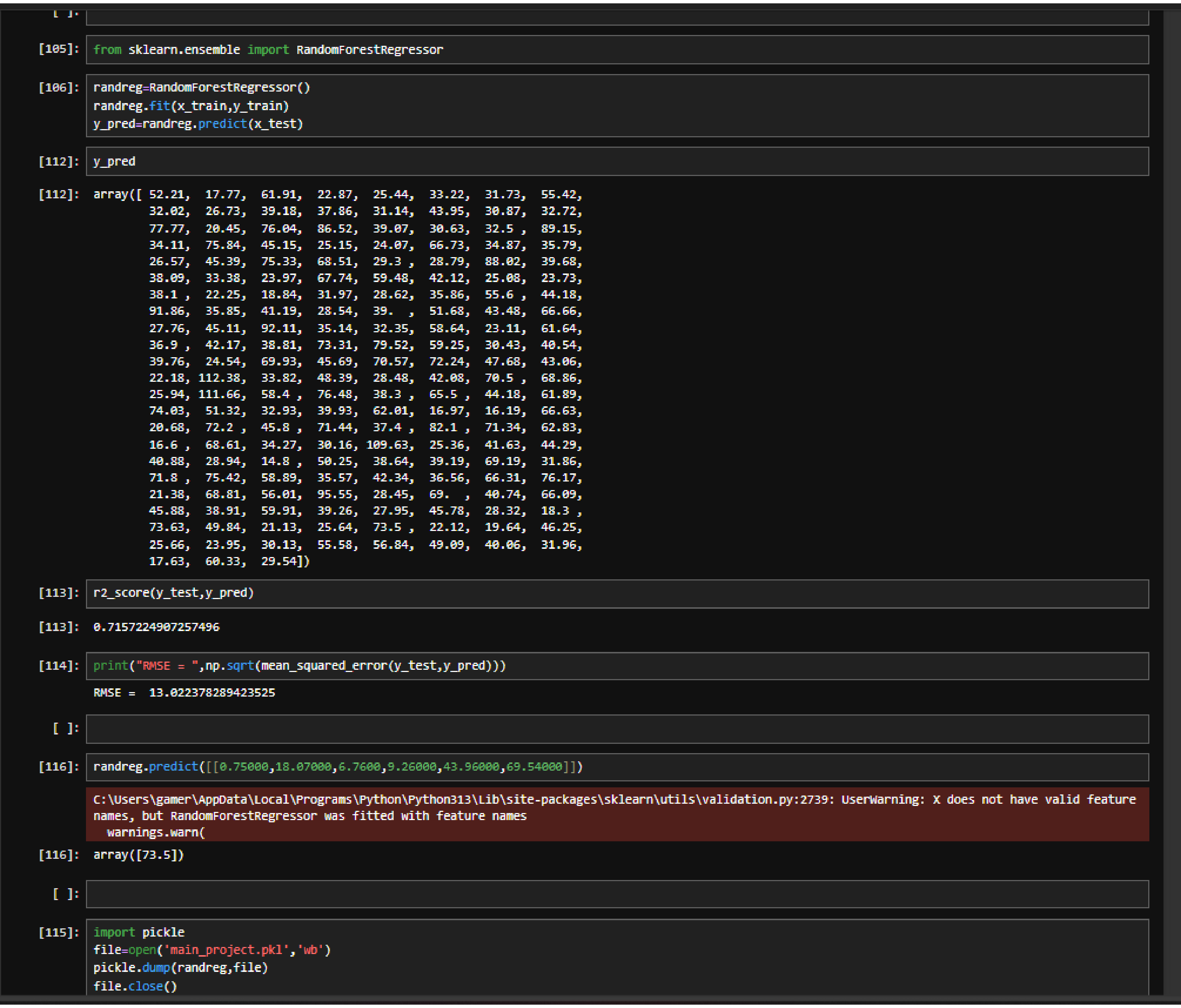
```
[68]: CO       0
      Ozone    0
      NO2      0
      SO2      0
      PM2.5    0
      PM10     0
      AQI      0
      dtype: int64
```

```
[69]: data.info
```

```
[69]: <bound method DataFrame.info of           CO     Ozone      NO2       SO2     PM2.5      PM10   AQI
      0    0.75000  18.07000   6.7600   9.26000  43.96000  69.54000  73.0
      1    0.74000  17.64000   6.6200   9.87000  44.08000  69.75000  73.0
      2    1.03000  16.15000   6.7400   9.87000  39.21000  62.83000  65.0
      3    1.17000  17.59000   7.1200  10.81000  41.75000  66.50000  70.0
      4    1.05000  17.64000   6.7200  10.68000  46.13000  72.58000  77.0
      ..       ...       ...      ...       ...       ...       ...   ...
      756  0.58008  27.21238  12.3866  14.20219  25.54167  56.20833  56.0
      757  0.45510  22.82584  14.6582  14.38560  16.50000  34.66667  35.0
      758  0.45719  27.07407  12.7963  13.44474  16.91667  35.45833  35.0
      759  0.53634  26.34239  12.6594  14.42967  19.16667  40.91667  41.0
      760  0.58789  18.22695  16.0060  13.16010  17.70833  37.50000  38.0

      [715 rows x 7 columns]>
```



```
[78]: plt.figure(figsize=(14, 7))

      plt.plot(data['CO'], label='CO (mg/m3)')
      plt.plot(data['Ozone'], label='Ozone (µg/m3)')
      plt.plot(data['NO2'], label='NO2 (µg/m3)')
      plt.plot(data['SO2'], label='SO2 (µg/m3)')
      plt.plot(data['PM2.5'], label='PM2.5 (µg/m3)')
      plt.plot(data['PM10'], label='PM10 (µg/m3)')

      plt.title('Pollutant Levels Over Time')
      plt.xlabel('Index')
      plt.ylabel('Concentration')
      plt.legend()
      plt.grid(True)
      plt.show()
```



```
[ ]:
```

```
[105]: from sklearn.ensemble import RandomForestRegressor
```

```
[106]: randreg=RandomForestRegressor()
       randreg.fit(x_train,y_train)
       y_pred=randreg.predict(x_test)
```

```
[112]: y_pred

[112]: array([ 52.21,  17.77,  61.91,  22.87,  25.44,  33.22,  31.73,  55.42,
                32.02,  26.73,  39.18,  37.86,  31.14,  43.95,  30.87,  32.72,
                77.77,  20.45,  76.04,  86.52,  39.07,  30.63,  32.5 ,  89.15,
                34.11,  75.84,  45.15,  25.15,  24.07,  66.73,  34.87,  35.79,
                26.57,  45.39,  75.33,  68.51,  29.3 ,  28.79,  88.02,  39.68,
                38.09,  33.38,  23.97,  67.74,  59.48,  42.12,  25.08,  23.73,
                38.1 ,  22.25,  18.84,  31.97,  28.62,  35.86,  55.6 ,  44.18,
                91.86,  35.85,  41.19,  28.54,  39.  ,  51.68,  43.48,  66.66,
                27.76,  45.11,  92.11,  35.14,  32.35,  58.64,  23.11,  61.64,
                36.9 ,  42.17,  38.81,  73.31,  79.52,  59.25,  30.43,  40.54,
                39.76,  24.54,  69.93,  45.69,  70.57,  72.24,  47.68,  43.06,
                22.18, 112.38,  33.82,  48.39,  28.48,  42.08,  70.5 ,  68.86,
                25.94, 111.66,  58.4 ,  76.48,  38.3 ,  65.5 ,  44.18,  61.89,
                74.03,  51.32,  32.93,  39.93,  62.01,  16.97,  16.19,  66.63,
                20.68,  72.2 ,  45.8 ,  71.44,  37.4 ,  82.1 ,  71.34,  62.83,
                16.6 ,  68.61,  34.27,  30.16, 109.63,  25.36,  41.63,  44.29,
                40.88,  28.94,  14.8 ,  50.25,  38.64,  39.19,  69.19,  31.86,
                71.8 ,  75.42,  58.89,  35.57,  42.34,  36.56,  66.31,  76.17,
                21.38,  68.81,  56.01,  95.55,  28.45,  69.  ,  40.74,  66.09,
                45.88,  38.91,  59.91,  39.26,  27.95,  45.78,  28.32,  18.3 ,
                73.63,  49.84,  21.13,  25.64,  73.5 ,  22.12,  19.64,  46.25,
                25.66,  23.95,  30.13,  55.58,  56.84,  49.09,  40.06,  31.96,
                17.63,  60.33,  29.54])
```

```
[113]: r2_score(y_test,y_pred)

[113]: 0.7157224907257496
```

```
[114]: print("RMSE = ",np.sqrt(mean_squared_error(y_test,y_pred)))

       RMSE =  13.022378289423525
```

```
[ ]:
```

```
[116]: randreg.predict([[0.75000,18.07000,6.7600,9.26000,43.96000,69.54000]])

       C:\Users\gamer\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
       names, but RandomForestRegressor was fitted with feature names
         warnings.warn(

[116]: array([73.5])
```

```
[ ]:
```

```
[115]: import pickle
       file=open('main_project.pkl','wb')
       pickle.dump(randreg,file)
       file.close()
```