In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
```

In [2]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\csvs_per_year\csvs_per_year\madrid_2016
a
```

Out[2]:

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2016-11-01 01:00:00 | NaN | 0.7 | NaN | NaN | 153.0 | 77.0 | NaN | NaN | NaN | 7.0 | NaN | NaN | 28079004 |
| **1** | 2016-11-01 01:00:00 | 3.1 | 1.1 | 2.0 | 0.53 | 260.0 | 144.0 | 4.0 | 46.0 | 24.0 | 18.0 | 2.44 | 14.4 | 28079008 |
| **2** | 2016-11-01 01:00:00 | 5.9 | NaN | 7.5 | NaN | 297.0 | 139.0 | NaN | NaN | NaN | NaN | NaN | 26.0 | 28079011 |
| **3** | 2016-11-01 01:00:00 | NaN | 1.0 | NaN | NaN | 154.0 | 113.0 | 2.0 | NaN | NaN | NaN | NaN | NaN | 28079016 |
| **4** | 2016-11-01 01:00:00 | NaN | NaN | NaN | NaN | 275.0 | 127.0 | 2.0 | NaN | NaN | 18.0 | NaN | NaN | 28079017 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **209491** | 2016-07-01 00:00:00 | NaN | 0.2 | NaN | NaN | 2.0 | 29.0 | 73.0 | NaN | NaN | NaN | NaN | NaN | 28079056 |
| **209492** | 2016-07-01 00:00:00 | NaN | 0.3 | NaN | NaN | 1.0 | 29.0 | NaN | 36.0 | NaN | 5.0 | NaN | NaN | 28079057 |
| **209493** | 2016-07-01 00:00:00 | NaN | NaN | NaN | NaN | 1.0 | 19.0 | 71.0 | NaN | NaN | NaN | NaN | NaN | 28079058 |
| **209494** | 2016-07-01 00:00:00 | NaN | NaN | NaN | NaN | 6.0 | 17.0 | 85.0 | NaN | NaN | NaN | NaN | NaN | 28079059 |
| **209495** | 2016-07-01 00:00:00 | NaN | NaN | NaN | NaN | 2.0 | 46.0 | 61.0 | 34.0 | NaN | NaN | NaN | NaN | 28079060 |

209496 rows × 14 columns

In [3]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209496 entries, 0 to 209495
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209496 non-null  object 
 1   BEN      50755 non-null   float64
 2   CO       85999 non-null   float64
 3   EBE      50335 non-null   float64
 4   NMHC     25970 non-null   float64
 5   NO       208614 non-null  float64
 6   NO_2     208614 non-null  float64
 7   O_3      121197 non-null  float64
 8   PM10     102892 non-null  float64
 9   PM25     52165 non-null   float64
 10  SO_2     86023 non-null   float64
 11  TCH      25970 non-null   float64
 12  TOL      50662 non-null   float64
 13  station  209496 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [4]:
```
b=a.fillna(value=86)
b
```

Out[4]:

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2016-11-01 01:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 153.0 | 77.0 | 86.0 | 86.0 | 86.0 | 7.0 | 86.00 | 86.0 | 28079004 |
| **1** | 2016-11-01 01:00:00 | 3.1 | 1.1 | 2.0 | 0.53 | 260.0 | 144.0 | 4.0 | 46.0 | 24.0 | 18.0 | 2.44 | 14.4 | 28079008 |
| **2** | 2016-11-01 01:00:00 | 5.9 | 86.0 | 7.5 | 86.00 | 297.0 | 139.0 | 86.0 | 86.0 | 86.0 | 86.0 | 86.00 | 26.0 | 28079011 |
| **3** | 2016-11-01 01:00:00 | 86.0 | 1.0 | 86.0 | 86.00 | 154.0 | 113.0 | 2.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079016 |
| **4** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 275.0 | 127.0 | 2.0 | 86.0 | 86.0 | 18.0 | 86.00 | 86.0 | 28079017 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **209491** | 2016-07-01 00:00:00 | 86.0 | 0.2 | 86.0 | 86.00 | 2.0 | 29.0 | 73.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079056 |
| **209492** | 2016-07-01 00:00:00 | 86.0 | 0.3 | 86.0 | 86.00 | 1.0 | 29.0 | 86.0 | 36.0 | 86.0 | 5.0 | 86.00 | 86.0 | 28079057 |
| **209493** | 2016-07-01 00:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 1.0 | 19.0 | 71.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079058 |
| **209494** | 2016-07-01 00:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 6.0 | 17.0 | 85.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079059 |
| **209495** | 2016-07-01 00:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 2.0 | 46.0 | 61.0 | 34.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079060 |

209496 rows × 14 columns

In [5]:
```
b.columns
```

Out[5]:
```
Index(['date', 'BEN', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'O_3', 'PM10', 'PM25',
       'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [6]:
```
c=b.head(30)
c
```

Out[6]:

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2016-11-01 01:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 153.0 | 77.0 | 86.0 | 86.0 | 86.0 | 7.0 | 86.00 | 86.000000 | 28079004 |
| **1** | 2016-11-01 01:00:00 | 3.1 | 1.1 | 2.0 | 0.53 | 260.0 | 144.0 | 4.0 | 46.0 | 24.0 | 18.0 | 2.44 | 14.400000 | 28079008 |
| **2** | 2016-11-01 01:00:00 | 5.9 | 86.0 | 7.5 | 86.00 | 297.0 | 139.0 | 86.0 | 86.0 | 86.0 | 86.0 | 86.00 | 26.000000 | 28079011 |
| **3** | 2016-11-01 01:00:00 | 86.0 | 1.0 | 86.0 | 86.00 | 154.0 | 113.0 | 2.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079016 |
| **4** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 275.0 | 127.0 | 2.0 | 86.0 | 86.0 | 18.0 | 86.00 | 86.000000 | 28079017 |
| **5** | 2016-11-01 01:00:00 | 0.9 | 0.5 | 0.5 | 86.00 | 66.0 | 82.0 | 1.0 | 27.0 | 86.0 | 8.0 | 86.00 | 6.000000 | 28079018 |
| **6** | 2016-11-01 01:00:00 | 0.7 | 0.8 | 0.4 | 0.13 | 57.0 | 66.0 | 3.0 | 23.0 | 15.0 | 4.0 | 1.35 | 5.000000 | 28079024 |
| **7** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 52.0 | 78.0 | 1.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079027 |
| **8** | 2016-11-01 01:00:00 | 86.0 | 1.2 | 86.0 | 86.00 | 205.0 | 85.0 | 6.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079035 |
| **9** | 2016-11-01 01:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 114.0 | 91.0 | 86.0 | 37.0 | 86.0 | 6.0 | 86.00 | 86.000000 | 28079036 |
| **10** | 2016-11-01 01:00:00 | 2.5 | 86.0 | 3.3 | 86.00 | 166.0 | 114.0 | 86.0 | 45.0 | 27.0 | 8.0 | 86.00 | 16.299999 | 28079038 |
| **11** | 2016-11-01 01:00:00 | 86.0 | 2.4 | 86.0 | 86.00 | 475.0 | 165.0 | 5.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079039 |
| **12** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 74.0 | 109.0 | 86.0 | 38.0 | 86.0 | 8.0 | 86.00 | 86.000000 | 28079040 |
| **13** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 168.0 | 93.0 | 86.0 | 41.0 | 26.0 | 86.0 | 86.00 | 86.000000 | 28079047 |
| **14** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 150.0 | 85.0 | 86.0 | 31.0 | 21.0 | 86.0 | 86.00 | 86.000000 | 28079048 |
| **15** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 81.0 | 97.0 | 1.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079049 |
| **16** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 307.0 | 134.0 | 86.0 | 55.0 | 35.0 | 86.0 | 86.00 | 86.000000 | 28079050 |
| **17** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 160.0 | 113.0 | 1.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079054 |

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **18** | 2016-11-01 01:00:00 | 1.4 | 86.0 | 1.3 | 0.20 | 72.0 | 84.0 | 86.0 | 21.0 | 86.0 | 86.0 | 1.50 | 6.900000 | 28079055 |
| **19** | 2016-11-01 01:00:00 | 86.0 | 0.5 | 86.0 | 86.00 | 67.0 | 83.0 | 6.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079056 |
| **20** | 2016-11-01 01:00:00 | 86.0 | 1.0 | 86.0 | 86.00 | 181.0 | 109.0 | 86.0 | 42.0 | 86.0 | 13.0 | 86.00 | 86.000000 | 28079057 |
| **21** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 4.0 | 36.0 | 11.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079058 |
| **22** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 99.0 | 66.0 | 7.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079059 |
| **23** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 79.0 | 103.0 | 4.0 | 29.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079060 |
| **24** | 2016-11-01 02:00:00 | 86.0 | 0.6 | 86.0 | 86.00 | 116.0 | 65.0 | 86.0 | 86.0 | 86.0 | 6.0 | 86.00 | 86.000000 | 28079004 |
| **25** | 2016-11-01 02:00:00 | 2.7 | 1.0 | 2.1 | 0.40 | 139.0 | 114.0 | 4.0 | 37.0 | 21.0 | 14.0 | 2.30 | 15.000000 | 28079008 |
| **26** | 2016-11-01 02:00:00 | 4.7 | 86.0 | 5.6 | 86.00 | 111.0 | 97.0 | 86.0 | 86.0 | 86.0 | 86.0 | 86.00 | 16.700001 | 28079011 |
| **27** | 2016-11-01 02:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 67.0 | 90.0 | 2.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079016 |
| **28** | 2016-11-01 02:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 99.0 | 84.0 | 2.0 | 86.0 | 86.0 | 13.0 | 86.00 | 86.000000 | 28079017 |
| **29** | 2016-11-01 02:00:00 | 0.5 | 0.5 | 0.2 | 86.00 | 61.0 | 73.0 | 1.0 | 17.0 | 86.0 | 7.0 | 86.00 | 3.300000 | 28079018 |

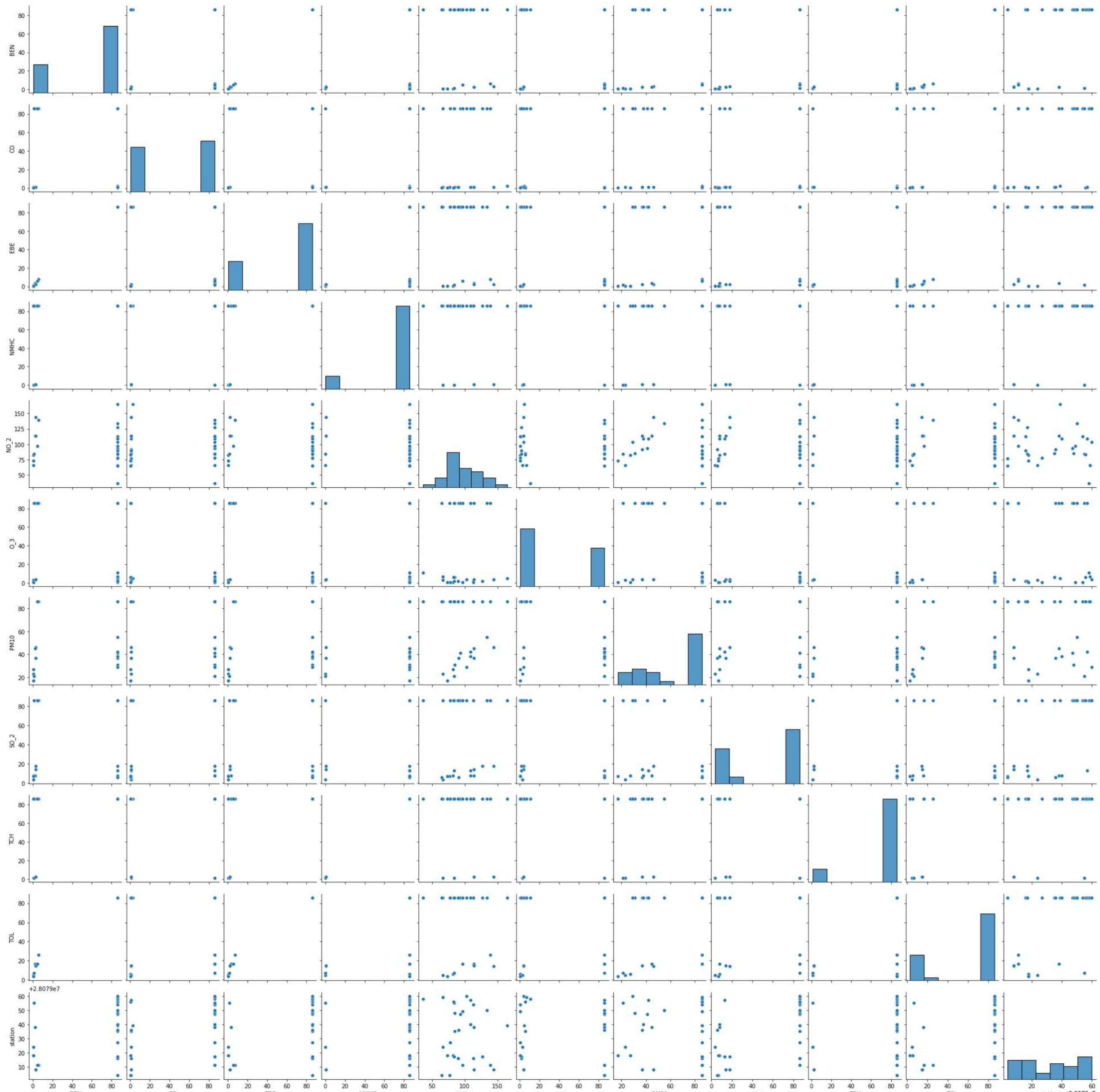
In [7]:
```
d=c[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2', 'O_3',
 'PM10', 'SO_2', 'TCH', 'TOL', 'station']]
d
```

Out[7]:

| | BEN | CO | EBE | NMHC | NO_2 | O_3 | PM10 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 86.0 | 0.7 | 86.0 | 86.00 | 77.0 | 86.0 | 86.0 | 7.0 | 86.00 | 86.000000 | 28079004 |
| **1** | 3.1 | 1.1 | 2.0 | 0.53 | 144.0 | 4.0 | 46.0 | 18.0 | 2.44 | 14.400000 | 28079008 |
| **2** | 5.9 | 86.0 | 7.5 | 86.00 | 139.0 | 86.0 | 86.0 | 86.0 | 86.00 | 26.000000 | 28079011 |
| **3** | 86.0 | 1.0 | 86.0 | 86.00 | 113.0 | 2.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079016 |
| **4** | 86.0 | 86.0 | 86.0 | 86.00 | 127.0 | 2.0 | 86.0 | 18.0 | 86.00 | 86.000000 | 28079017 |
| **5** | 0.9 | 0.5 | 0.5 | 86.00 | 82.0 | 1.0 | 27.0 | 8.0 | 86.00 | 6.000000 | 28079018 |
| **6** | 0.7 | 0.8 | 0.4 | 0.13 | 66.0 | 3.0 | 23.0 | 4.0 | 1.35 | 5.000000 | 28079024 |
| **7** | 86.0 | 86.0 | 86.0 | 86.00 | 78.0 | 1.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079027 |
| **8** | 86.0 | 1.2 | 86.0 | 86.00 | 85.0 | 6.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079035 |
| **9** | 86.0 | 0.7 | 86.0 | 86.00 | 91.0 | 86.0 | 37.0 | 6.0 | 86.00 | 86.000000 | 28079036 |
| **10** | 2.5 | 86.0 | 3.3 | 86.00 | 114.0 | 86.0 | 45.0 | 8.0 | 86.00 | 16.299999 | 28079038 |
| **11** | 86.0 | 2.4 | 86.0 | 86.00 | 165.0 | 5.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079039 |
| **12** | 86.0 | 86.0 | 86.0 | 86.00 | 109.0 | 86.0 | 38.0 | 8.0 | 86.00 | 86.000000 | 28079040 |
| **13** | 86.0 | 86.0 | 86.0 | 86.00 | 93.0 | 86.0 | 41.0 | 86.0 | 86.00 | 86.000000 | 28079047 |
| **14** | 86.0 | 86.0 | 86.0 | 86.00 | 85.0 | 86.0 | 31.0 | 86.0 | 86.00 | 86.000000 | 28079048 |
| **15** | 86.0 | 86.0 | 86.0 | 86.00 | 97.0 | 1.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079049 |
| **16** | 86.0 | 86.0 | 86.0 | 86.00 | 134.0 | 86.0 | 55.0 | 86.0 | 86.00 | 86.000000 | 28079050 |
| **17** | 86.0 | 86.0 | 86.0 | 86.00 | 113.0 | 1.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079054 |
| **18** | 1.4 | 86.0 | 1.3 | 0.20 | 84.0 | 86.0 | 21.0 | 86.0 | 1.50 | 6.900000 | 28079055 |
| **19** | 86.0 | 0.5 | 86.0 | 86.00 | 83.0 | 6.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079056 |
| **20** | 86.0 | 1.0 | 86.0 | 86.00 | 109.0 | 86.0 | 42.0 | 13.0 | 86.00 | 86.000000 | 28079057 |
| **21** | 86.0 | 86.0 | 86.0 | 86.00 | 36.0 | 11.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079058 |
| **22** | 86.0 | 86.0 | 86.0 | 86.00 | 66.0 | 7.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079059 |
| **23** | 86.0 | 86.0 | 86.0 | 86.00 | 103.0 | 4.0 | 29.0 | 86.0 | 86.00 | 86.000000 | 28079060 |
| **24** | 86.0 | 0.6 | 86.0 | 86.00 | 65.0 | 86.0 | 86.0 | 6.0 | 86.00 | 86.000000 | 28079004 |
| **25** | 2.7 | 1.0 | 2.1 | 0.40 | 114.0 | 4.0 | 37.0 | 14.0 | 2.30 | 15.000000 | 28079008 |
| **26** | 4.7 | 86.0 | 5.6 | 86.00 | 97.0 | 86.0 | 86.0 | 86.0 | 86.00 | 16.700001 | 28079011 |
| **27** | 86.0 | 0.7 | 86.0 | 86.00 | 90.0 | 2.0 | 86.0 | 86.0 | 86.00 | 86.000000 | 28079016 |
| **28** | 86.0 | 86.0 | 86.0 | 86.00 | 84.0 | 2.0 | 86.0 | 13.0 | 86.00 | 86.000000 | 28079017 |
| **29** | 0.5 | 0.5 | 0.2 | 86.00 | 73.0 | 1.0 | 17.0 | 7.0 | 86.00 | 3.300000 | 28079018 |

In [8]:
```
sns.pairplot(d)
```

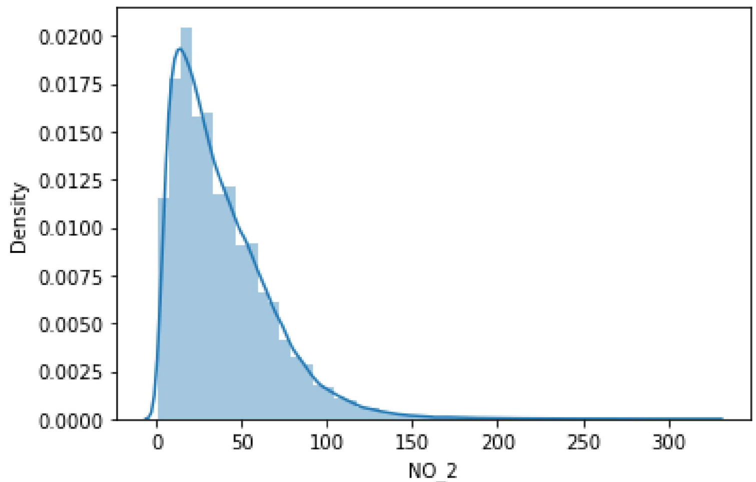
Out[8]: <seaborn.axisgrid.PairGrid at 0x1d524e13b50>



In [9]:
```
sns.distplot(a['NO_2'])
```

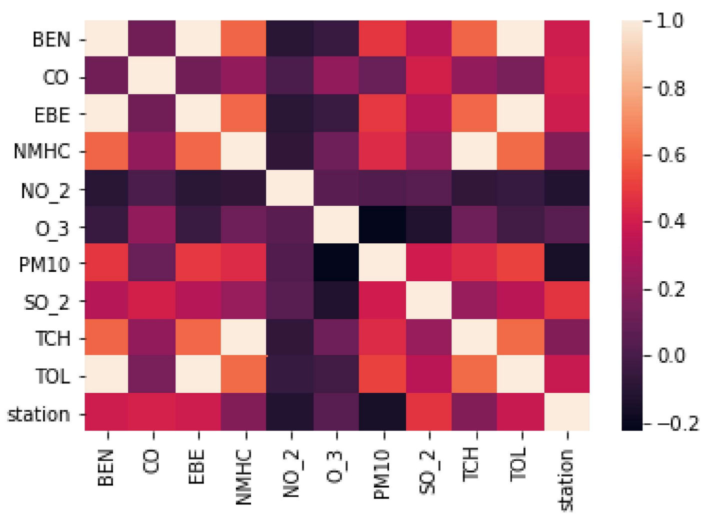
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarni
ng: `distplot` is a deprecated function and will be removed in a future version. Plea
se adapt your code to use either `displot` (a figure-level function with similar flex
ibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]: <AxesSubplot:xlabel='NO_2', ylabel='Density'>




In [10]:
```
sns.heatmap(d.corr())
```

Out[10]: <AxesSubplot:>




In [11]:
```
x=d[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2']]
y=d['TCH']
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [13]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[13]: LinearRegression()

In [14]:
```
print(lr.intercept_)
```

```
1.4748792800566548
```

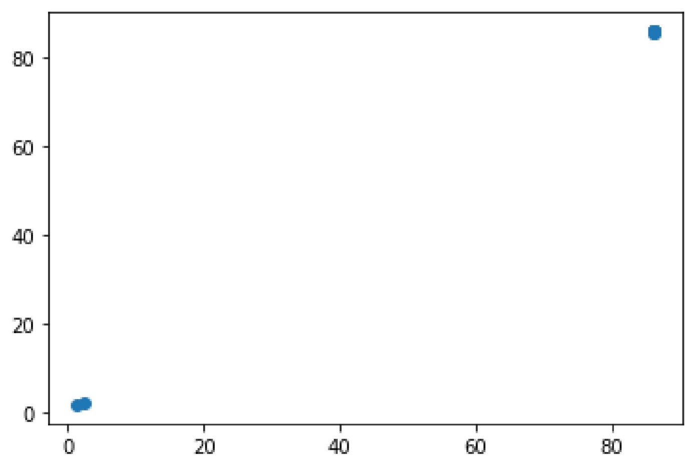
In [15]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[15]:

| | Co-efficient |
|---|---|
| BEN | 0.134493 |
| CO | -0.000405 |
| EBE | -0.134567 |
| NMHC | 0.981957 |
| NO_2 | 0.001019 |

In [16]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[16]: <matplotlib.collections.PathCollection at 0x1d52d895700>




In [17]:
```
print(lr.score(x_test,y_test))
```

```
0.9999806673343318
```

In [18]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [19]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[19]: Ridge(alpha=10)

In [20]:
```
rr.score(x_test,y_test)
```

Out[20]: 0.9999768962712836

In [21]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```
Out[21]: Lasso(alpha=10)

In [22]:
```
la.score(x_test,y_test)
```
Out[22]: 0.999678913074025

In [23]:
```
a1=b.head(7000)
a1
```
Out[23]:

| | date | BEN | CO | EBE | NMHC | NO | NO_2 | O_3 | PM10 | PM25 | SO_2 | TCH | TOL | station |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2016-11-01 01:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 153.0 | 77.0 | 86.0 | 86.0 | 86.0 | 7.0 | 86.00 | 86.0 | 28079004 |
| **1** | 2016-11-01 01:00:00 | 3.1 | 1.1 | 2.0 | 0.53 | 260.0 | 144.0 | 4.0 | 46.0 | 24.0 | 18.0 | 2.44 | 14.4 | 28079008 |
| **2** | 2016-11-01 01:00:00 | 5.9 | 86.0 | 7.5 | 86.00 | 297.0 | 139.0 | 86.0 | 86.0 | 86.0 | 86.0 | 86.00 | 26.0 | 28079011 |
| **3** | 2016-11-01 01:00:00 | 86.0 | 1.0 | 86.0 | 86.00 | 154.0 | 113.0 | 2.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079016 |
| **4** | 2016-11-01 01:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 275.0 | 127.0 | 2.0 | 86.0 | 86.0 | 18.0 | 86.00 | 86.0 | 28079017 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **6995** | 2016-11-13 04:00:00 | 86.0 | 0.7 | 86.0 | 86.00 | 96.0 | 71.0 | 5.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079039 |
| **6996** | 2016-11-13 04:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 45.0 | 70.0 | 86.0 | 26.0 | 86.0 | 9.0 | 86.00 | 86.0 | 28079040 |
| **6997** | 2016-11-13 04:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 87.0 | 70.0 | 86.0 | 28.0 | 23.0 | 86.0 | 86.00 | 86.0 | 28079047 |
| **6998** | 2016-11-13 04:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 66.0 | 59.0 | 86.0 | 33.0 | 26.0 | 86.0 | 86.00 | 86.0 | 28079048 |
| **6999** | 2016-11-13 04:00:00 | 86.0 | 86.0 | 86.0 | 86.00 | 98.0 | 53.0 | 1.0 | 86.0 | 86.0 | 86.0 | 86.00 | 86.0 | 28079049 |

7000 rows × 14 columns

In [24]:
```
e=a1[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2', 'O_3',
 'PM10', 'SO_2', 'TCH', 'TOL', 'station']]
```

In [25]:
```
f=e.iloc[:,0:14]
g=e.iloc[:,-1]
```

In [26]:
```
h=StandardScaler().fit_transform(f)
```

In [27]:
```
logr=LogisticRegression(max_iter=10000)
logr.fit(h,g)
```

Out[27]:
```
LogisticRegression(max_iter=10000)
```

In [28]:
```
from sklearn.model_selection import train_test_split
h_train,h_test,g_train,g_test=train_test_split(h,g,test_size=0.3)
```

In [29]:
```
i=[[10,20,30,40,50,60,15,26,37,47,58]]
```

In [30]:
```
prediction=logr.predict(i)
print(prediction)
```

```
[28079059]
```

In [31]:
```
logr.classes_
```

Out[31]:
```
array([28079004, 28079008, 28079011, 28079016, 28079017, 28079018,
       28079024, 28079027, 28079035, 28079036, 28079038, 28079039,
       28079040, 28079047, 28079048, 28079049, 28079050, 28079054,
       28079055, 28079056, 28079057, 28079058, 28079059, 28079060],
      dtype=int64)
```

In [32]:
```
logr.predict_proba(i)[0][0]
```

Out[32]:
```
0.0
```

In [33]:
```
logr.predict_proba(i)[0][1]
```

Out[33]:
```
0.0
```

In [34]:
```
logr.score(h_test,g_test)
```

Out[34]:
```
0.9452380952380952
```

In [35]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[35]:
```
ElasticNet()
```

In [36]:
```
print(en.coef_)
```

```
[ 0.00000000e+00 -4.53645120e-04  5.21275022e-04  9.79350801e-01
  0.00000000e+00]
```

In [37]:
```
print(en.intercept_)
```

```
1.7459678449370415
```

In [38]:
```
prediction=en.predict(x_test)
print(en.score(x_test,y_test))
```

```
0.9999713903612383
```

In [39]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(h_train,g_train)
```

Out[39]:
```
RandomForestClassifier()
```

In [40]:
```
parameters={'max_depth':[1,2,3,4,5],
 'min_samples_leaf':[5,10,15,20,25],
 'n_estimators':[10,20,30,40,50]
 }
```

In [41]:
```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(h_train,g_train)
```

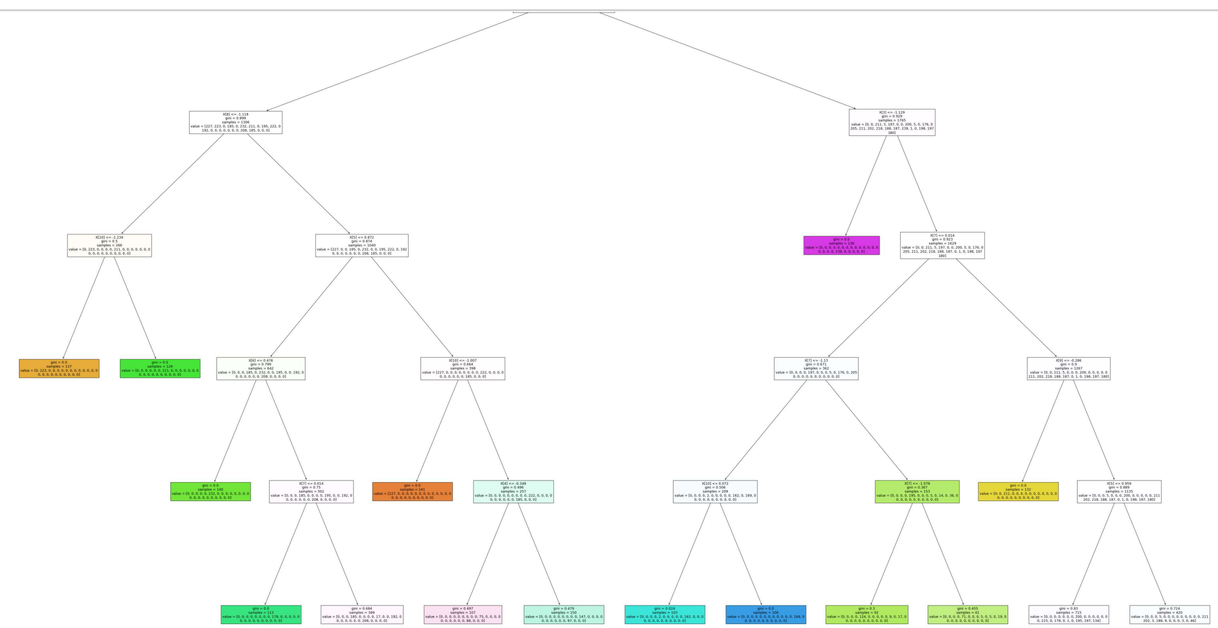
Out[41]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [42]:
```
grid_search.best_score_
```

Out[42]:
```
0.9936734693877551
```

In [43]:
```
rfc_best=grid_search.best_estimator_
```

In [44]:
```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,50))
plot_tree(rfc_best.estimators_[2],filled=True)
```



# Conclusion: from this data set i observed that the ridge has the highest accuracy of 0.9999768962712836

In [ ]: