Classification and Regression, Using Pollutant Data From The EPA

Before we begin doing any coding, its important to know that we are analyzing 4 different types of pollutant data, in 48 states across the United States. The 4 different pollutants come from different sources, and the 4 that we are looking to analyze specifically are O3, CO, SO2 and NO2.

O3 is known as ground level ozone pollutant, and is not directly emitted into the air, but comes from chemical reactions between Volatile Organic Compounds, Nitrous Oxides in the air and sunlight. O3 pollutant usually comes from industrial facilities, car exhaust, gas vapors, chemical solvents and some major sources of nitrogen oxides.

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

Examining the Data

In this section, we are trying to get an understanding of our data from the csv file. Attempting to do our best at visualizing it, and getting as much information possible out of our dataset, and picking and choosing which features of the dataset might be best to use for our Machine Learning Model

```
df = pd.read_csv('pollution_2000_2021.csv')   #Reading the original
data file
df.head()    #Returns the first 5 rows of the data
```

```
         Date  Year  Month  ...  NO2 1st Max Value NO2 1st Max Hour
NO2 AQI
0  2000-01-01  2000      1  ...               49.0               19
46
1  2000-01-02  2000      1  ...               36.0               19
34
2  2000-01-03  2000      1  ...               51.0                8
48
3  2000-01-04  2000      1  ...               74.0                8
72
4  2000-01-05  2000      1  ...               61.0               22
58

[5 rows x 24 columns]
```

```
df.tail(5)   #Returns the last 5 rows of the data
```

```
             Date  Year  Month  ...  NO2 1st Max Value NO2 1st Max
Hour NO2 AQI
```

```
608694  2021-06-26  2021      6  ...                2.1
0       2
608695  2021-06-27  2021      6  ...                2.3
20       2
608696  2021-06-28  2021      6  ...                5.8
23       5
608697  2021-06-29  2021      6  ...                7.4
0       7
608698  2021-06-30  2021      6  ...                7.5
5       7

[5 rows x 24 columns]
```

```
df.shape # (Rows, Columns), lets us know how much data we are dealing with
```

```
(608699, 24)
```

```
df.columns # These are all the titles of the columns in the data file, good to know which data we are dealing with
```

```
Index(['Date', 'Year', 'Month', 'Day', 'Address', 'State', 'County',
'City',
       'O3 Mean', 'O3 1st Max Value', 'O3 1st Max Hour', 'O3 AQI', 'CO
Mean',
       'CO 1st Max Value', 'CO 1st Max Hour', 'CO AQI', 'SO2 Mean',
       'SO2 1st Max Value', 'SO2 1st Max Hour', 'SO2 AQI', 'NO2 Mean',
       'NO2 1st Max Value', 'NO2 1st Max Hour', 'NO2 AQI'],
      dtype='object')
```

```
df.describe()   #Quick rundown of all the data in the file
```

```
                Year          Month  ...  NO2 1st Max Hour        NO2
AQI
count  608699.000000  608699.000000  ...     608699.000000
608699.000000
mean     2011.040529       6.508670  ...         11.606845
22.124240
std         6.122558       3.310633  ...          7.881014
14.610125
min      2000.000000       1.000000  ...          0.000000
0.000000
25%      2006.000000       4.000000  ...          5.000000
10.000000
50%      2012.000000       7.000000  ...          9.000000
20.000000
75%      2016.000000       9.000000  ...         20.000000
31.000000
max      2021.000000      12.000000  ...         23.000000
133.000000
```

```
[8 rows x 19 columns]
```

In this part of the notebook we are cleaning up data from the original file, too much data to work with, so we choose which specific type of data we want to work with in order to create the best machine learning model

```
pollution_data = df.drop(['Year', 'Month', 'Day', 'Address', 'County',
'City', 'O3 Mean', 'O3 1st Max Value', 'O3 1st Max Hour', 'CO Mean',
       'CO 1st Max Value', 'CO 1st Max Hour','SO2 Mean', 'SO2 1st Max
Value', 'SO2 1st Max Hour','NO2 Mean',
        'NO2 1st Max Value', 'NO2 1st Max Hour',], axis=1)

pollution_data2 = df.drop(['Month', 'Day', 'Address', 'County',
'City', 'O3 Mean', 'O3 1st Max Value', 'O3 1st Max Hour', 'CO Mean',
       'CO 1st Max Value', 'CO 1st Max Hour','SO2 Mean', 'SO2 1st Max
Value', 'SO2 1st Max Hour','NO2 Mean',
        'NO2 1st Max Value', 'NO2 1st Max Hour',], axis=1)

pollution_data
```

	Date	State	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
0	2000-01-01	Arizona	37	25.0	13.0	46
1	2000-01-02	Arizona	30	26.0	4.0	34
2	2000-01-03	Arizona	15	28.0	16.0	48
3	2000-01-04	Arizona	31	34.0	23.0	72
4	2000-01-05	Arizona	11	42.0	21.0	58
...	...	...	...	...	...	...
608694	2021-06-26	Wyoming	45	1.0	0.0	2
608695	2021-06-27	Wyoming	39	1.0	0.0	2
608696	2021-06-28	Wyoming	46	1.0	0.0	5
608697	2021-06-29	Wyoming	61	1.0	0.0	7
608698	2021-06-30	Wyoming	50	1.0	0.0	7

```
[608699 rows x 6 columns]

pollution_data2
```

	Date	Year	State	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
0	2000-01-01	2000	Arizona	37	25.0	13.0	46
1	2000-01-02	2000	Arizona	30	26.0	4.0	34
2	2000-01-03	2000	Arizona	15	28.0	16.0	48
3	2000-01-04	2000	Arizona	31	34.0	23.0	72
4	2000-01-05	2000	Arizona	11	42.0	21.0	58
...	...	...	...	...	...	...	...
608694	2021-06-26	2021	Wyoming	45	1.0	0.0	2
608695	2021-06-27	2021	Wyoming	39	1.0	0.0	2
608696	2021-06-28	2021	Wyoming	46	1.0	0.0	5
608697	2021-06-29	2021	Wyoming	61	1.0	0.0	7
608698	2021-06-30	2021	Wyoming	50	1.0	0.0	7

```
[608699 rows x 7 columns]
```

The difference between pollution_data and pollution_data2, is that pollution_data2 includes the year

```
#pollution_data.to_csv("pollution_data_cleaned.csv", index=False)
```

We just created a new file with the new data that we cleaned up, and saved it as a new file, commented out so it isn't ran everytime

```
pollution_data.sample(10)   #random sample of the data from the file
```

	Date	State	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
17419	2000-05-09	Pennsylvania	80	2.0	39.0	47
410015	2015-04-28	California	77	2.0	1.0	46
52899	2002-07-05	California	133	3.0	4.0	12
488900	2017-02-12	Ohio	36	8.0	0.0	29
308232	2012-10-04	California	31	5.0	0.0	6
366002	2013-06-10	Texas	31	5.0	3.0	17
523830	2018-10-04	Utah	32	2.0	0.0	17
398575	2014-04-14	Texas	37	1.0	0.0	16
6676	2000-12-14	California	25	22.0	13.0	47
277695	2011-05-02	California	50	8.0	1.0	46

```
by_state = pollution_data.groupby(['State']).mean()
by_state
```

State	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
Alabama	38.137431	4.198401	5.612194	20.126437
Alaska	19.052111	5.789578	13.736148	19.298153
Arizona	44.281801	8.255187	3.461847	33.490963
Arkansas	36.923070	5.658980	2.651785	20.636196
California	39.746546	6.840152	2.943838	22.704071
Colorado	39.038586	6.770029	7.386067	34.126421
Connecticut	39.171476	3.576753	2.530753	19.596879
Delaware	36.355980	3.727421	1.869406	20.399512
District Of Columbia	37.570771	9.415644	9.457055	27.004930
Florida	38.851235	5.860236	2.366987	14.799416
Georgia	36.061751	5.039127	0.984562	21.307692
Hawaii	27.456891	3.261642	1.654533	9.166624
Idaho	36.869880	2.897992	0.426506	22.314056
Illinois	35.584846	6.105019	12.316014	27.565077
Indiana	42.324185	5.071660	9.842005	21.708024
Iowa	34.914837	2.968498	0.986088	13.750969
Kansas	37.299635	6.049474	8.131573	21.169564
Kentucky	45.378688	3.247510	13.426424	23.141891
Louisiana	36.479598	5.485700	8.976865	24.650566

```
Maine                29.799066  2.863908   2.194635  10.939937
Maryland             38.888899  3.482966   4.067775  15.782022
Massachusetts        28.451862  4.665777   5.742094  26.996486
Michigan             44.702885  6.397205  16.619928  30.755185
Minnesota            34.282933  3.401823   0.661558  14.759735
Mississippi          34.002037  3.219280   0.820095  14.799050
Missouri             42.659572  5.828356  11.410344  26.047259
Nevada               44.312478  4.523005   0.979435  25.082433
New Hampshire        36.884165  3.829430   4.714104  10.883147
New Jersey           36.188149  5.573693   7.219758  28.980657
New Mexico           44.522416  3.459329   0.868188  22.655839
New York             34.966001  5.357387  11.744751  29.546135
North Carolina       46.539024  5.069889   6.199738  19.606527
North Dakota         31.817211  2.070370   0.998911  11.645098
Ohio                 34.068484  3.749636   9.445016  20.764721
Oklahoma             43.812477  2.814584   2.994412  14.548461
Oregon               27.669785  4.701109   1.424407  16.314771
Pennsylvania         43.592473  3.798927  14.494253  22.646773
Rhode Island         38.865380  2.929215   0.761390  13.109910
South Carolina       41.746873  1.590232   4.603335   5.063133
South Dakota         36.449813  2.385305   0.566874  10.810710
Tennessee            44.580006  3.160886   1.024774   2.174876
Texas                38.639054  4.241435   4.205502  22.480511
Utah                 43.623262  4.377543   0.890702  25.295828
Vermont              33.105291  2.662963   0.207407  13.588360
Virginia             41.553280  5.401340  10.033667  20.049956
Washington           26.302728  3.772186   4.569343  24.028045
Wisconsin            34.846535  5.608204  11.115983  28.708628
Wyoming              43.967644  1.543984   1.405865   9.948837
```

```
len(pollution_data.State.unique())
```

```
48
```

The output of 48 implies that there is data for 48 states, Montana and Nebraska are the two states that are not included in the dataset

```
by_state['O3 AQI'].mean(), by_state['CO AQI'].mean(), by_state['SO2 AQI'].mean(), by_state['NO2 AQI'].mean()
```

```
(37.54873126004455, 4.431228543244901, 5.2238519688155165, 19.8952180849919)
```

Returned the average for each AQI value over the entire United States

```
ordered_O3 = by_state.sort_values(by='O3 AQI', ascending = False)
top_10_O3, bottom_10_O3  = ordered_O3.head(10)['O3 AQI'], ordered_O3.tail(10)['O3 AQI']
top_10_O3, bottom_10_O3
```

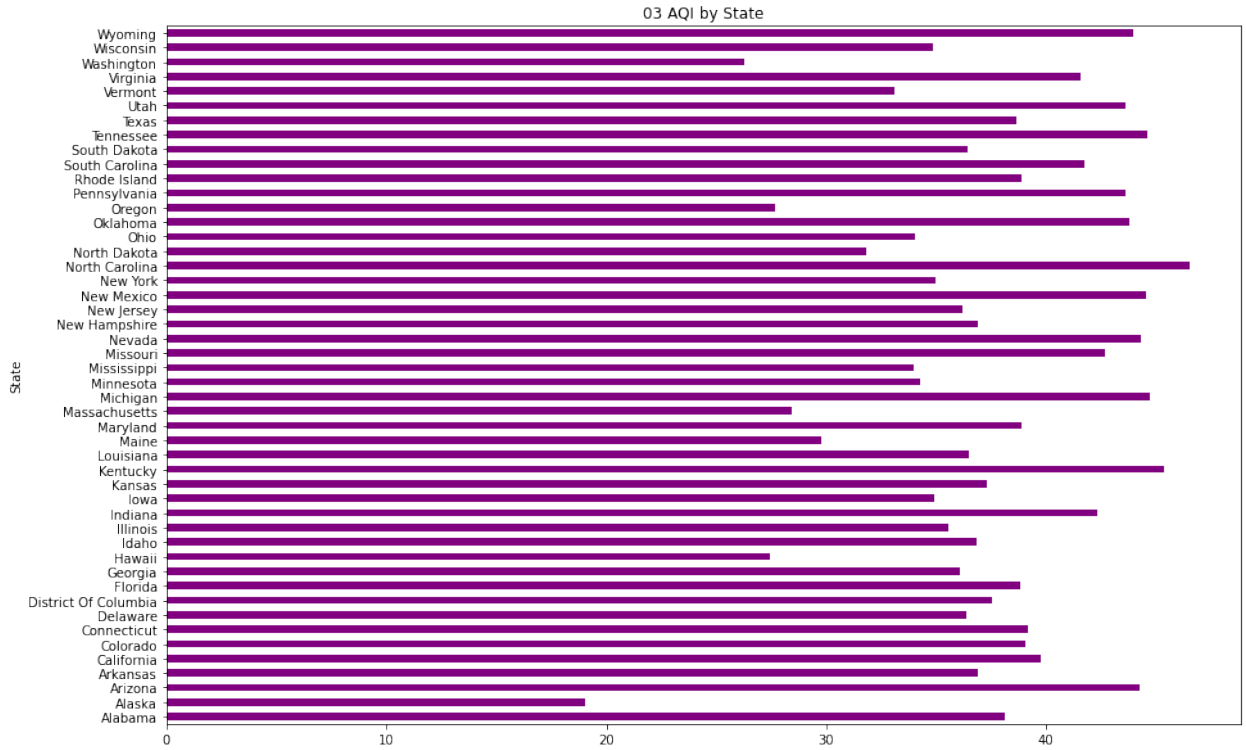
```
(State
 North Carolina    46.539024
 Kentucky          45.378688
 Michigan          44.702885
 Tennessee         44.580006
 New Mexico        44.522416
 Nevada            44.312478
 Arizona           44.281801
 Wyoming           43.967644
 Oklahoma          43.812477
 Utah              43.623262
 Name: O3 AQI, dtype: float64,
 State
 Ohio             34.068484
 Mississippi      34.002037
 Vermont          33.105291
 North Dakota     31.817211
 Maine            29.799066
 Massachusetts    28.451862
 Oregon           27.669785
 Hawaii           27.456891
 Washington       26.302728
 Alaska           19.052111
 Name: O3 AQI, dtype: float64)
```

Returned values for the top 10 O3 AQI states, and also the bottom 10

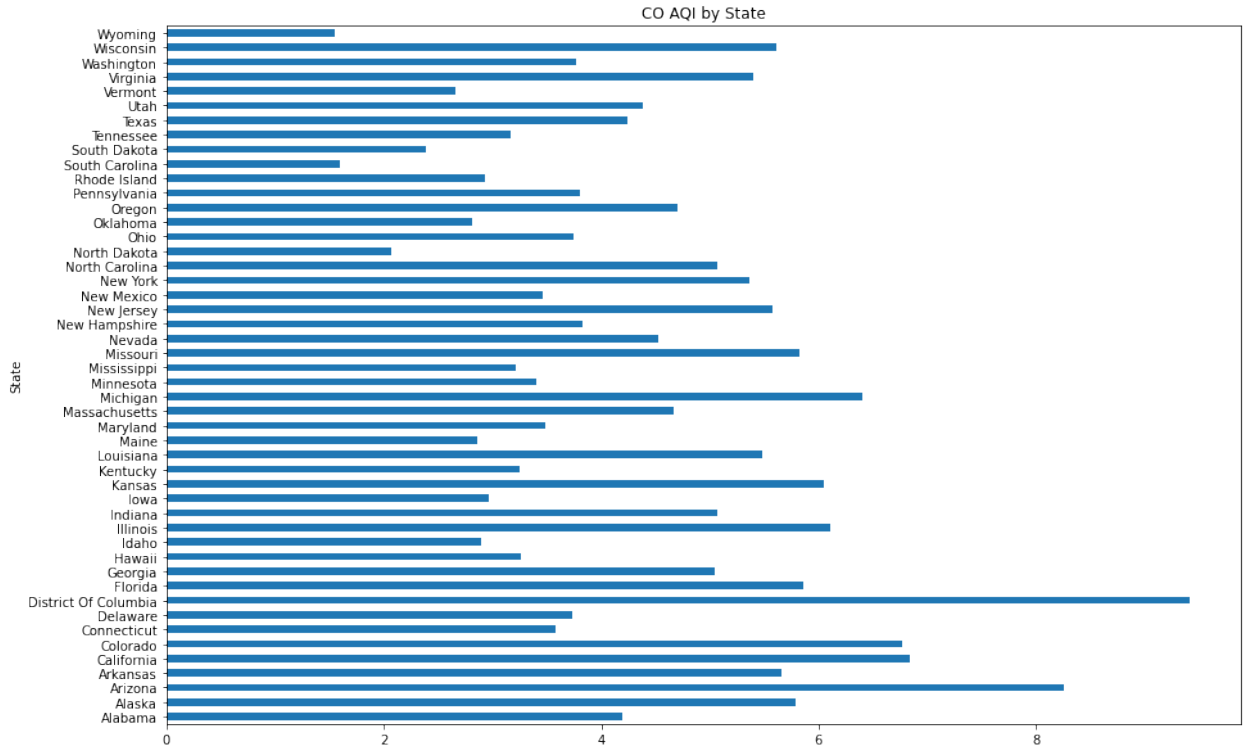
Visualizing the Data

In this section we are plotting our data from the dataset to get a visual understanding of our data

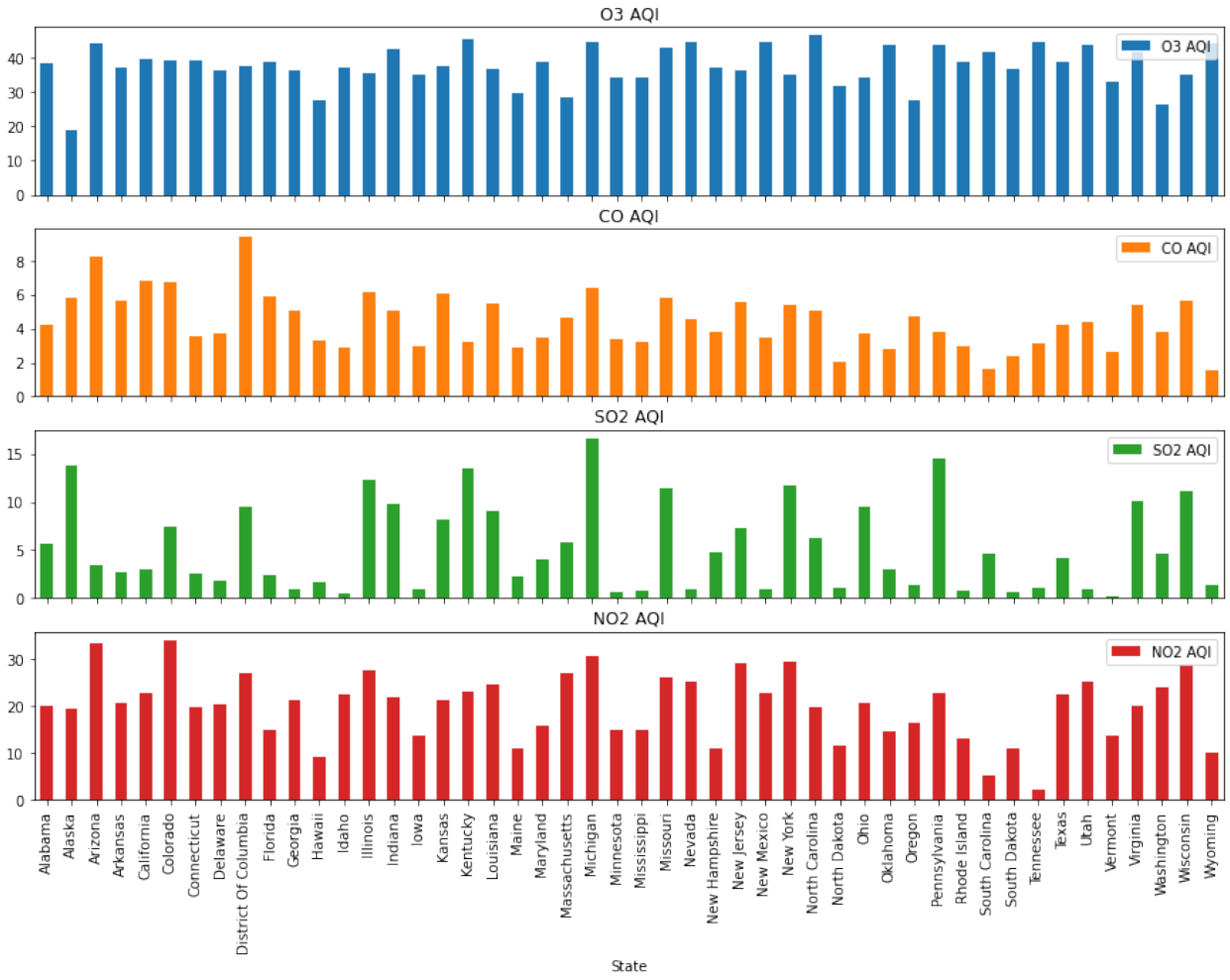
```
total_O3 = pollution_data.groupby(['State'])['O3 AQI'].mean()
total_O3.plot(kind='barh', title='03 AQI by State', ylabel='O3 AQI
Value',
         xlabel='State', figsize=(15,10), color = 'purple');
```




```
total_CO = pollution_data.groupby(['State'])['CO AQI'].mean()
total_CO.plot(kind='barh', title='CO AQI by State', ylabel='CO AQI
Value',
         xlabel='State', figsize=(15,10));
```




```
by_state.plot(subplots = True, kind='bar', figsize=(15,10));
```



This is the mean AQI values for O3, CO, SO2 and NO2, organized by state

```
grouped_by_state = pollution_data2.groupby(['State']).count()
grouped_by_state = grouped_by_state.sort_values(by=['Date'])
grouped_by_state
```

State	Date	Year	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
Idaho	1245	1245	1245	1245	1245	1245
Wisconsin	1414	1414	1414	1414	1414	1414
Mississippi	1473	1473	1473	1473	1473	1473
Alaska	1516	1516	1516	1516	1516	1516
South Carolina	1679	1679	1679	1679	1679	1679
Vermont	1890	1890	1890	1890	1890	1890
Alabama	2001	2001	2001	2001	2001	2001
Michigan	2218	2218	2218	2218	2218	2218
Minnesota	2414	2414	2414	2414	2414	2414
Delaware	2458	2458	2458	2458	2458	2458
Washington	2603	2603	2603	2603	2603	2603
Tennessee	3431	3431	3431	3431	3431	3431

New Mexico	3725	3725	3725	3725	3725	3725
Georgia	3757	3757	3757	3757	3757	3757
Rhode Island	3885	3885	3885	3885	3885	3885
New Hampshire	3928	3928	3928	3928	3928	3928
South Dakota	4015	4015	4015	4015	4015	4015
North Dakota	4590	4590	4590	4590	4590	4590
Oregon	4597	4597	4597	4597	4597	4597
Wyoming	4945	4945	4945	4945	4945	4945
Indiana	5247	5247	5247	5247	5247	5247
Kentucky	5321	5321	5321	5321	5321	5321
Nevada	5738	5738	5738	5738	5738	5738
Massachusetts	7115	7115	7115	7115	7115	7115
Louisiana	7867	7867	7867	7867	7867	7867
Florida	8221	8221	8221	8221	8221	8221
Utah	8701	8701	8701	8701	8701	8701
Missouri	8739	8739	8739	8739	8739	8739
District Of Columbia	9128	9128	9128	9128	9128	9128
Maine	9207	9207	9207	9207	9207	9207
Kansas	9318	9318	9318	9318	9318	9318
Connecticut	9739	9739	9739	9739	9739	9739
Iowa	10063	10063	10063	10063	10063	10063
New Jersey	10598	10598	10598	10598	10598	10598
Arkansas	10841	10841	10841	10841	10841	10841
Oklahoma	10916	10916	10916	10916	10916	10916
Maryland	11125	11125	11125	11125	11125	11125
Ohio	11667	11667	11667	11667	11667	11667
Hawaii	11703	11703	11703	11703	11703	11703
North Carolina	12992	12992	12992	12992	12992	12992
Illinois	13607	13607	13607	13607	13607	13607
Colorado	14254	14254	14254	14254	14254	14254
Virginia	18356	18356	18356	18356	18356	18356
New York	21383	21383	21383	21383	21383	21383
Arizona	23183	23183	23183	23183	23183	23183
Texas	36457	36457	36457	36457	36457	36457
Pennsylvania	52026	52026	52026	52026	52026	52026
California	187403	187403	187403	187403	187403	187403

The returned values above, for grouped_by_state is the amount of datapoints that were in the dataset for each state, in this case, Idaho has the fewest data points and California has the most data points.

```
idaho = pollution_data2[pollution_data2.State == 'Idaho']
idaho = idaho.sort_values(by='Date')
idaho
```

	Date	Year	State	O3 AQI	CO AQI	SO2 AQI	NO2 AQI
233137	2009-04-30	2009	Idaho	7	2.0	0.0	24
233138	2009-05-01	2009	Idaho	43	2.0	1.0	19
233139	2009-05-02	2009	Idaho	41	2.0	0.0	23

```
233140  2009-05-03  2009  Idaho      40     2.0      1.0       16
233141  2009-05-04  2009  Idaho      42     2.0      0.0       17
...            ...   ...    ...     ...     ...      ...      ...
597510  2021-06-26  2021  Idaho      44     2.0      0.0       29
597511  2021-06-27  2021  Idaho      48     2.0      0.0       22
597512  2021-06-28  2021  Idaho      50     2.0      0.0       18
597513  2021-06-29  2021  Idaho      64     2.0      0.0       16
597514  2021-06-30  2021  Idaho      58     2.0      0.0       20

[1245 rows x 7 columns]
```

Since Idaho has the fewest amount of values, we are going to use Idaho as a state to create our ML algorithm with because, its only 1245 data points, and it is easier for a computer to handle training 1245 data points as opposed to Californias 187403 Datapoints

```
idaho.groupby(['Year']).count()
```

```
      Date  State  O3 AQI  CO AQI  SO2 AQI  NO2 AQI
Year
2009    74     74      74      74       74       74
2010   361    361     361     361      361      361
2011    23     23      23      23       23       23
2019   254    254     254     254      254      254
2020   357    357     357     357      357      357
2021   176    176     176     176      176      176
```

Now we also find out how many datapoints there are for each year, so there is only 6 years of data for Idaho, 2009-2011 and 2019-2021, where 2009 and 2011 do not even have much datapoints, nevertheless it will still be incorporated into our algorithm

```
#idaho.to_csv('idaho_Data.csv', index=False)
```

The above is creating a new csv file of just Idaho data sorted by Date, and now we can just manipulate our data from the Idaho file, rather than the entire dataset

```
idaho_data = pd.read_csv('idaho_Data.csv')
idaho_data['year_label'] = idaho_data.groupby('Year',
sort=False).ngroup() + 1
idaho_data
```

```
            Date  Year  State  O3 AQI  CO AQI  SO2 AQI  NO2 AQI
year_label
0     2009-04-30  2009  Idaho       7     2.0      0.0       24
1
1     2009-05-01  2009  Idaho      43     2.0      1.0       19
1
2     2009-05-02  2009  Idaho      41     2.0      0.0       23
1
3     2009-05-03  2009  Idaho      40     2.0      1.0       16
```

```
1
4     2009-05-04  2009  Idaho      42     2.0      0.0        17
1
...          ...   ...    ...     ...     ...      ...       ...
...
1240  2021-06-26  2021  Idaho      44     2.0      0.0        29
6
1241  2021-06-27  2021  Idaho      48     2.0      0.0        22
6
1242  2021-06-28  2021  Idaho      50     2.0      0.0        18
6
1243  2021-06-29  2021  Idaho      64     2.0      0.0        16
6
1244  2021-06-30  2021  Idaho      58     2.0      0.0        20
6

[1245 rows x 8 columns]
```

So the new datafile with only Idaho data was read, and a new row was introduced in the last cell, where we are adding a tag that represents the year to make it easier to incorporate the Machine Learning algorithm. The year label is going to be 1-6, where 1 will represent 2009 and 2 will represent 2010 and so on.

Modelling our Data

So now we are starting to build machine learning models in this section to model and make predictions about our data using features that we believe are valid.

k-NN Classification

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler


feature_names = ['O3 AQI','CO AQI', 'SO2 AQI', 'NO2 AQI']

X_idaho = idaho_data[feature_names]
y_idaho = idaho_data['year_label']
target_years = ['2009', '2010', '2011', '2019', '2020', '2021']

X_train, X_test, y_train, y_test = train_test_split(X_idaho, y_idaho,
test_size=0.2, random_state=1)
scaler = MinMaxScaler()

X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.fit_transform(X_test)

knn = KNeighborsClassifier(n_neighbors = 1)
knn.fit(X_train_scaled, y_train)
```

```
print(knn.score(X_train_scaled, y_train))
print(knn.score(X_test_scaled, y_test))

example_aqi = [[25, 1.0, 1.0, 10]]
example_aqi_scaled = scaler.transform(example_aqi)

print(target_years[knn.predict(example_aqi_scaled)[0]-1])
```

```
0.9267068273092369
0.3453815261044177
2010
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

X_idaho2d = idaho_data[feature_names]
y_idaho2d = idaho_data['year_label']

y_idaho2011 = y_idaho2d == 3 #Here we are making the problem the aqi
values in 2011 vs every other year

x_train2, x_test2, y_train2, y_test2 =
train_test_split(X_idaho2d.values, y_idaho2011.values)

clf = LogisticRegression(C=100).fit(x_train2, y_train2)
print(clf.score(x_train2, y_train2))
print(clf.score(x_test2, y_test2))


print(clf.predict([[25, 2.0, 3.0, 45]])[0])
print(clf.predict([[45, 1.0, 0.0, 20]])[0])
```

```
0.9914255091103966
0.9615384615384616
False
False
```

Random Forest

```
from plot_decision_boundaries import
plot_class_regions_for_classifier_subplot
from plot_decision_boundaries import plot_class_regions_for_classifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

X_train4, X_test4, y_train4, y_test4 =
```

```
train_test_split(X_idaho.values, y_idaho.values, random_state=0)

X = X_train4
y = y_train4

RFC = RandomForestClassifier(max_features= 4,
random_state=0).fit(X_train4, y_train4)

print('Accuracy of RF classifier on training set: {:.2f}'
     .format(RFC.score(X_train4, y_train4)))
print('Accuracy of RF classifier on test set: {:.2f}'
     .format(RFC.score(X_test4, y_test4)))

RandomFC = RandomForestClassifier(max_features=4,
random_state=0).fit(X_train_scaled, y_train)
RandomFC2 = RandomForestClassifier(max_features=4,
random_state=0).fit(X_test_scaled, y_test)

print('Accuracy of RF classifier on scaled training set: {:.2f}'
     .format(RandomFC.score(X_train_scaled, y_train)))
print('Accuracy of RF classifier on scaled test set: {:.2f}'
     .format(RandomFC2.score(X_test_scaled, y_test)))
```

```
Accuracy of RF classifier on training set: 0.93
Accuracy of RF classifier on test set: 0.32
Accuracy of RF classifier on scaled training set: 0.93
Accuracy of RF classifier on scaled test set: 0.97
```