# Download Air Quality Dataset from Kaggle Predict Air Quality Index using Linear regression and classify it into five categories using SVM (i.e. Very good, good, moderate, poor, worst)

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import LabelEncoder

from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

pd.set_option('display.max_columns', None)
pd.set_option('display.float_format', lambda x: '%.3f' % x
              )
import warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/city_day.csv')
```

```
df
```

| | City | Date | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | SO2 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Ahmedabad | 2015-01-01 | NaN | NaN | 0.920 | 18.220 | 17.150 | NaN | 0.920 | 27.640 |
| **1** | Ahmedabad | 2015-01-02 | NaN | NaN | 0.970 | 15.690 | 16.460 | NaN | 0.970 | 24.550 |
| **2** | Ahmedabad | 2015-01-03 | NaN | NaN | 17.400 | 19.300 | 29.700 | NaN | 17.400 | 29.070 |
| **3** | Ahmedabad | 2015-01-04 | NaN | NaN | 1.700 | 18.480 | 17.970 | NaN | 1.700 | 18.590 |
| **4** | Ahmedabad | 2015-01-05 | NaN | NaN | 22.100 | 21.420 | 37.760 | NaN | 22.100 | 39.330 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **29526** | Visakhapatnam | 2020-06-27 | 15.020 | 50.940 | 7.680 | 25.060 | 19.540 | 12.470 | 0.470 | 8.550 |
| **29527** | Visakhapatnam | 2020-06-28 | 24.380 | 74.090 | 3.420 | 26.060 | 16.530 | 11.990 | 0.520 | 12.720 |

```
df.head()
```

| | City | Date | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | SO2 | O3 | Benzene | Toluene | Xylene | AQI | AQI_Bucket |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Ahmedabad | 2015-01-01 | NaN | NaN | 0.920 | 18.220 | 17.150 | NaN | 0.920 | 27.640 | 133.360 | 0.000 | 0.020 | 0.000 | NaN | NaN |
| **1** | Ahmedabad | 2015-01-02 | NaN | NaN | 0.970 | 15.690 | 16.460 | NaN | 0.970 | 24.550 | 34.060 | 3.680 | 5.500 | 3.770 | NaN | NaN |
| **2** | Ahmedabad | 2015-01-03 | NaN | NaN | 17.400 | 19.300 | 29.700 | NaN | 17.400 | 29.070 | 30.700 | 6.800 | 16.400 | 2.250 | NaN | NaN |
| **3** | Ahmedabad | 2015-01-04 | NaN | NaN | 1.700 | 18.480 | 17.970 | NaN | 1.700 | 18.590 | 36.080 | 4.430 | 10.140 | 1.000 | NaN | NaN |
| **4** | Ahmedabad | 2015-01-05 | NaN | NaN | 22.100 | 21.420 | 37.760 | NaN | 22.100 | 39.330 | 39.310 | 7.010 | 18.890 | 2.780 | NaN | NaN |

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   City        29531 non-null  object 
 1   Date        29531 non-null  object 
 2   PM2.5       24933 non-null  float64
 3   PM10        18391 non-null  float64
 4   NO          25949 non-null  float64
 5   NO2         25946 non-null  float64
 6   NOx         25346 non-null  float64
 7   NH3         19203 non-null  float64
 8   CO          27472 non-null  float64
 9   SO2         25677 non-null  float64
 10  O3          25509 non-null  float64
 11  Benzene     23908 non-null  float64
 12  Toluene     21490 non-null  float64
 13  Xylene      11422 non-null  float64
 14  AQI         24850 non-null  float64
 15  AQI_Bucket  24850 non-null  object 
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
```

```
df.describe()
```

| | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | S |
|---|---|---|---|---|---|---|---|---|
| **count** | 24933.000 | 18391.000 | 25949.000 | 25946.000 | 25346.000 | 19203.000 | 27472.000 | 25677.0 |
| **mean** | 67.451 | 118.127 | 17.575 | 28.561 | 32.309 | 23.483 | 2.249 | 14.5 |
| **std** | 64.661 | 90.605 | 22.786 | 24.475 | 31.646 | 25.684 | 6.963 | 18.1 |
| **min** | 0.040 | 0.010 | 0.020 | 0.010 | 0.000 | 0.010 | 0.000 | 0.0 |
| **25%** | 28.820 | 56.255 | 5.630 | 11.750 | 12.820 | 8.580 | 0.510 | 5.6 |
| **50%** | 48.570 | 95.680 | 9.890 | 21.690 | 23.520 | 15.850 | 0.890 | 9.1 |
| **75%** | 80.590 | 149.745 | 19.950 | 37.620 | 40.127 | 30.020 | 1.450 | 15.2 |
| **max** | 949.990 | 1000.000 | 390.680 | 362.210 | 467.630 | 352.890 | 175.810 | 193.8 |

```
df.isnull().sum()    ### checking null values
```

```
City              0
Date              0
PM2.5          4598
PM10          11140
NO             3582
NO2            3585
NOx            4185
NH3           10328
CO             2059
SO2            3854
O3             4022
Benzene        5623
Toluene        8041
Xylene        18109
AQI            4681
AQI_Bucket     4681
dtype: int64
```

```
df.isnull().count()
```

```
City          29531
Date          29531
PM2.5         29531
PM10          29531
NO            29531
NO2           29531
NOx           29531
NH3           29531
CO            29531
SO2           29531
O3            29531
```

```
Benzene       29531
Toluene       29531
Xylene        29531
AQI           29531
AQI_Bucket    29531
dtype: int64
```

```
df.shape
```

```
(29531, 16)
```

```
data = df[['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene', 'AQI']]

# Drop rows with missing values
df = df.dropna()
```

```
## label encoder

label_encoder = LabelEncoder()
df['AQI_Bucket'] = label_encoder.fit_transform(df['AQI_Bucket'])
```

```
X = df.drop(['AQI', 'AQI_Bucket', 'City', 'Date'], axis=1)
y = df['AQI_Bucket']
```

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
X_train
```

| | PM2.5 | PM10 | NO | NO2 | NOx | NH3 | CO | SO2 | O3 | Benzene | Toluene | Xylene |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **4004** | 33.950 | 61.350 | 18.800 | 17.210 | 28.340 | 10.340 | 0.470 | 12.850 | 24.200 | 1.920 | 1.080 | 1.960 |
| **15405** | 48.680 | 119.710 | 9.410 | 32.910 | 10.060 | 14.760 | 0.140 | 9.660 | 48.790 | 0.050 | 0.770 | 0.240 |
| **25715** | 27.710 | 68.290 | 16.140 | 26.480 | 28.900 | 2.760 | 0.950 | 2.980 | 14.750 | 1.950 | 4.680 | 1.180 |
| **16396** | 62.380 | 109.070 | 8.670 | 34.510 | 24.290 | 14.800 | 0.630 | 5.240 | 32.490 | 0.640 | 3.590 | 0.070 |
| **16471** | 33.940 | 68.490 | 9.050 | 22.050 | 17.670 | 16.430 | 0.460 | 4.690 | 23.970 | 1.650 | 6.780 | 0.750 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **15839** | 27.830 | 86.250 | 2.800 | 20.050 | 12.920 | 22.110 | 0.570 | 9.660 | 20.790 | 0.370 | 3.670 | 0.480 |
| **28166** | 51.420 | 94.820 | 28.900 | 41.100 | 44.030 | 16.590 | 0.970 | 17.900 | 31.390 | 7.790 | 15.090 | 10.140 |
| **28206** | 68.700 | 106.530 | 3.720 | 26.070 | 16.540 | 12.740 | 1.130 | 8.590 | 70.650 | 5.250 | 6.770 | 4.070 |
| **28583** | 62.290 | 125.910 | 8.620 | 41.420 | 28.900 | 10.870 | 1.070 | 7.340 | 95.860 | 4.460 | 8.250 | 1.980 |
| **3728** | 43.930 | 100.620 | 25.340 | 30.820 | 44.510 | 7.940 | 0.020 | 6.340 | 16.430 | 6.220 | 0.970 | 5.190 |

4988 rows × 12 columns

```
y_train
```

```
4004     3
15405    1
25715    3
16396    1
16471    3
        ..
15839    3
28166    1
28206    1
28583    1
3728     3
Name: AQI_Bucket, Length: 4988, dtype: int64
```

```
# Train the linear regression model
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred_regression = regressor.predict(X_test)
```

```
## predict SVM
svm_classifier = SVC()
svm_classifier.fit(X_train, y_train)
y_pred_svm = svm_classifier.predict(X_test)
```

```
svm_accuracy = accuracy_score(y_test, y_pred_svm)
print("SVM Accuracy:", svm_accuracy)
```

```
SVM Accuracy: 0.8028846153846154
```