**Air Quality**

**Problem Statement:** Air contains a mixture of gases which affect the environment among which gases containing Carbon monoxide (CO) like PT08.S1(CO), NMHC(GT), PT08.S2(NMHC), NOx(GT), PT08.S3(NOx), and PT08.S5(O3) are very dangerous that affects the people health and the symptoms of the diseases caused by Carbon monoxide cannot be visible soon. If the threshold value of CO increases more than the actual value it causes some diseases:

* If the value of CO=35ppm it causes headache and dizziness within six to eight hours of constant exposure.
* If it is 100ppm then it causes slight headache in two to three hours.

|  |  |
| --- | --- |
| **Effects of carbon monoxide in relation to the concentration in parts per million in the air:** | |
| **Concentration** | **Symptoms** |
| 35 ppm (0.0035%) | Headache and dizziness within six to eight hours of constant exposure |
| 100 ppm (0.01%) | Slight headache in two to three hours |
| 200 ppm (0.02%) | Slight headache within two to three hours; loss of judgment |
| 400 ppm (0.04%) | Frontal headache within one to two hours |
| 800 ppm (0.08%) | Dizziness, nausea, and convulsions within 45 min; insensible within 2 hours |
| 1,600 ppm (0.16%) | Headache, [increased heart rate](https://en.wikipedia.org/wiki/Tachycardia), dizziness, and nausea within 20 min; death in less than 2 hours |
| 3,200 ppm (0.32%) | Headache, dizziness and nausea in five to ten minutes. Death within 30 minutes. |
| 6,400 ppm (0.64%) | Headache and dizziness in one to two minutes. Convulsions, respiratory arrest, and death in less than 20 minutes. |
| 12,800 ppm (1.28%) | Unconsciousness after 2–3 breaths. Death in less than three minutes. |

**SOLUTION:** Every gas contains a threshold value in the atmosphere. So, this Air Quality model predicts the value of Carbon monoxide that present in the air. When the predicted value is more than the threshold value (35ppm) then there is a need of taking precautions.

**REQUIREMENTS:**

Here we are going to consider a dataset which contains 9358 instances of hourly averaged responses from an array of 6 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. The device was located on the field in a significantly polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year) representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses which contains mixture of gases data for the past few years. The attributes in the dataset are:

* Date
* Time
* CO(GT) = Carbon Monoxide ground value
* PT08.S1(CO) = Tin Oxide
* NMHC(GT) = Non Methane Hydro Carbons
* C6H6(GT) = Benzene
* PT08.S2(NMHC) = Titania
* NOx(GT) = Mixture of Nitric Oxide(NO) and Nitrogen Dioxide(NO2)
* PT08.S3 (NOx) = Tungsten Oxide.
* NO2(GT) = Nitrogen Dioxide ground value
* PT08.S4(NO2) = Tungsten Oxide with NO2
* PT08.S5(O3) = Indium Oxide
* T = Temperature
* RH = Relative Humidity
* AH = Air Humidity

**Steps for executing this model:**

**1)** Importing the **libraries** which are required for this model is:

* import matplotlib.pyplot as plt
* import numpy as np
* import pandas as pd
* %matplotlib inline

**2)** Read the excel sheet by using **Pandas** with the below command

air\_data = pd.read\_excel ('AirQualityUCI.xlsx')

* AirQualityUCI.xlsx is our dataset which containing mixture of gases.

**3)** After reading the data the scaling is performed by using **StandardScalar** class which is imported from **sklearn.preprocessing** package

**4)** The plotting is done on the features considered below:

C6H6 (GT), RH, AH, PT08.S1 (CO).

**5)** After considering the required features we are going to plot a **pair plot** on those data by using below command

* sns.pairplot (data\_to\_plot, size=2.0)
* plt.tight\_layout ()
* plt.show ()

Where,

* Data\_to\_plot contains the data of features which we have taken
* Size represents the graph size

**6)** We are going to store the complete data in another variable and we are going to drop some columns which are not required.

**7)** The dropped columns are date, time, C6H6, PT08.S4 (NO2).

**8)** Here we are importing **train\_test\_split** class from **sklearn.cross\_validation** package. The scikit-learn version should be 0.19.0, if the scikit-learn version is greater than 0.19.0 then downgrade the version by using the below command

* pip install scikit-learn==0.19.0

**9)** We will split the dataset into four parts by using below command

* X\_train, X\_test, y\_train, y\_test = train\_test\_split (features, labels, test\_size=0.3)
* Here the training contains 70% of data and testing contains 30% of data

**10)** We are using **LinearRegression** model which is imported from **sklearn.linear\_model** package, then we create an object for the Linear Regressionclass and train the dataset which we have taken.

**11)** After training we test the model by using x\_test and y\_test and we got an accuracy of **99.92%.**

**12)** We can change the data in x\_test to predict the data in the future and compare that value with our own data present in the y\_test.