In [ ]:

```
//Sourabh Raut
//Roll no- 60
//Assignment - Group B-(2)
//Perform the following operations using Python on the Air quality and Heart
Diseases data sets
a. Data cleaning
b. Data integration
c. Data transformation
d. Error correcting
e. Data model building
```

In [3]:

```
import pandas as pd
import numpy as np
df = pd.read_csv('AirQuality.csv',sep=";")
df
```

Out[3]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0
...	...	...	...	...	...	...	...	...
9466	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9467	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9468	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9469	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9470	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

9471 rows × 17 columns

In [4]:

```
df.isnull()
```

Out[4]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
9466	True	True	True	True	True	True	True	True	

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3
9467	True	True	True	True	True	True	True	True	
9468	True	True	True	True	True	True	True	True	
9469	True	True	True	True	True	True	True	True	
9470	True	True	True	True	True	True	True	True	

9471 rows × 17 columns

In [5]:
```
df.loc[1:4]
```

Out[5]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	

In [6]:
```
df1 = df.loc[0:4]
df1
```

Out[6]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	

In [7]:
```
df1.isnull()
```

Out[7]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NO
0	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	Fa

In [8]:
```
df1.isna().any()
```

Out[8]:
```
Date             False
Time             False
CO(GT)           False
PT08.S1(CO)      False
NMHC(GT)         False
C6H6(GT)         False
PT08.S2(NMHC)    False
NOx(GT)          False
PT08.S3(NOx)     False
NO2(GT)          False
PT08.S4(NO2)     False
PT08.S5(O3)      False
T                False
RH               False
AH               False
Unnamed: 15       True
Unnamed: 16       True
dtype: bool
```

In [9]:
```
df1.drop_duplicates(subset=['Unnamed: 15','Unnamed: 16'])
```

Out[9]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	

In [10]:
```
df.duplicated().sum()
```

Out[10]:
```
113
```

In [11]:
```
df1.duplicated().sum()
```

Out[11]:
```
0
```

Data integration

In [12]:
```
df1=df.loc[1:4,['C6H6(GT)','PT08.S2(NMHC)']]
df1
```

Out[12]:

	C6H6(GT)	PT08.S2(NMHC)
1	9,4	955.0
2	9,0	939.0
3	9,2	948.0
4	6,5	836.0

In [13]:
```
df2=df.loc[9466:9470,['C6H6(GT)','PT08.S2(NMHC)']]
```

```
df2
```

Out[13]:

	C6H6(GT)	PT08.S2(NMHC)
9466	NaN	NaN
9467	NaN	NaN
9468	NaN	NaN
9469	NaN	NaN
9470	NaN	NaN

In [14]:

```
frames=[df1,df2]
merge = pd.concat(frames)
merge
```

Out[14]:

	C6H6(GT)	PT08.S2(NMHC)
1	9,4	955.0
2	9,0	939.0
3	9,2	948.0
4	6,5	836.0
9466	NaN	NaN
9467	NaN	NaN
9468	NaN	NaN
9469	NaN	NaN
9470	NaN	NaN

Data transformation

In [15]:

```
df1 = df.loc[0:4]
df1
```

Out[15]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	P1
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	

In [21]:

```
df1.melt()
```

Out[21]:

	variable	value
0	Date	10/03/2004
1	Date	10/03/2004
2	Date	10/03/2004
3	Date	10/03/2004
4	Date	10/03/2004
...	...	...
80	Unnamed: 16	NaN
81	Unnamed: 16	NaN
82	Unnamed: 16	NaN
83	Unnamed: 16	NaN
84	Unnamed: 16	NaN

85 rows × 2 columns

In [22]:
```
df2.melt()
```

Out[22]:

	variable	value
0	C6H6(GT)	NaN
1	C6H6(GT)	NaN
2	C6H6(GT)	NaN
3	C6H6(GT)	NaN
4	C6H6(GT)	NaN
5	PT08.S2(NMHC)	NaN
6	PT08.S2(NMHC)	NaN
7	PT08.S2(NMHC)	NaN
8	PT08.S2(NMHC)	NaN
9	PT08.S2(NMHC)	NaN

error correcting

In [59]:
```
df1["Unnamed: 15"] = df1["Unnamed: 15"].fillna("mean")
df1
```

Out[59]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	

In [63]:
```
df1["Unnamed: 16"].fillna(df1["Unnamed: 16"].mean() , inplace= True)
df1
```

Out[63]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	166.0	
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	103.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	131.0	
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	172.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	131.0	

In [67]:
```
df["PT08.S4(NO2)"].fillna(df["PT08.S4(NO2)"].mean() , inplace= True)
df
```

Out[67]:

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(G
0	10/03/2004	18.00.00	2,6	1360.000000	150.000000	11,9	1046.000000	166.0000
1	10/03/2004	19.00.00	2	1292.000000	112.000000	9,4	955.000000	103.0000
2	10/03/2004	20.00.00	2,2	1402.000000	88.000000	9,0	939.000000	131.0000
3	10/03/2004	21.00.00	2,2	1376.000000	80.000000	9,2	948.000000	172.0000
4	10/03/2004	22.00.00	1,6	1272.000000	51.000000	6,5	836.000000	131.0000
...	...	...	...	...	...	...	...	
9466	NaN	NaN	NaN	1048.990061	-159.090093	NaN	894.595276	168.6169
9467	NaN	NaN	NaN	1048.990061	-159.090093	NaN	894.595276	168.6169
9468	NaN	NaN	NaN	1048.990061	-159.090093	NaN	894.595276	168.6169
9469	NaN	NaN	NaN	1048.990061	-159.090093	NaN	894.595276	168.6169
9470	NaN	NaN	NaN	1048.990061	-159.090093	NaN	894.595276	168.6169

9471 rows × 17 columns

model building

In [70]:
```
# Import Python libraries for data manipuation and visualization
```

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

# Import the Python machine learning libraries we need
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Import some convenience functions.  This can be found on the course github
#from functions import *
```

In [87]:

```
# Split into input and output features
y = df1["Date"]
X = df1[["NO2(GT)","PT08.S5(O3)"]]
X.head(7)
```

Out[87]:

	NO2(GT)	PT08.S5(O3)
0	113.0	1268.0
1	92.0	972.0
2	114.0	1074.0
3	122.0	1203.0
4	116.0	1110.0

In [88]:

```
y.head()
```

Out[88]:

```
0    10/03/2004
1    10/03/2004
2    10/03/2004
3    10/03/2004
4    10/03/2004
Name: Date, dtype: object
```

In [89]:

```
# Split into test and training sets
test_size = 0.33
seed = 7
X_train, X_test, y_train, y_test =  train_test_split(X, y, test_size=test_size, rand
```

In [90]:

```
X_train
```

Out[90]:

	NO2(GT)	PT08.S5(O3)
2	114.0	1074.0
1	92.0	972.0
4	116.0	1110.0

In [91]:

```
X_test
```

Out[91]:

	NO2(GT)	PT08.S5(O3)
0	113.0	1268.0

	NO2(GT)	PT08.S5(O3)
3	122.0	1203.0

In [92]:
```
y_train
```

Out[92]:
```
2    10/03/2004
1    10/03/2004
4    10/03/2004
Name: Date, dtype: object
```

In [93]:
```
y_test
```

Out[93]:
```
0    10/03/2004
3    10/03/2004
Name: Date, dtype: object
```

In [94]:
```
# Select algorithm
model = DecisionTreeClassifier()
```

In [95]:
```
# Fit model to the data
model.fit(X_train, y_train)
```

Out[95]:
```
DecisionTreeClassifier()
```

In [107…
```
predictions
```

Out[107…
```
array(['10/03/2004', '10/03/2004'], dtype=object)
```

In [108…
```
print(accuracy_score(y_test, predictions))
```

```
1.0
```

In [111…
```
df1 = X_test.copy()
df1['Actual'] = y_test
df1['Prediction'] = predictions
df1
```

Out[111…

	NO2(GT)	PT08.S5(O3)	Actual	Prediction
0	113.0	1268.0	10/03/2004	10/03/2004
3	122.0	1203.0	10/03/2004	10/03/2004

In [ ]:
```

```