Group B Assignment - 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

Importing libraries

In [97]:

```
import pandas as pd
import numpy as np
```

Reading the csv file

In [98]:

```
data = pd.read_csv("airquality (1).csv")
data
```

Out[98]:

	Unnamed: 0	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	1	41.0	190.0	7.4	67	5	1	High
1	2	36.0	118.0	8.0	72	5	2	High
2	3	12.0	149.0	12.6	74	5	3	Low
3	4	18.0	313.0	11.5	62	5	4	NaN
4	5	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...	...
148	149	30.0	193.0	6.9	70	9	26	Low
149	150	NaN	145.0	13.2	77	9	27	Low
150	151	14.0	191.0	14.3	75	9	28	High
151	152	18.0	131.0	8.0	76	9	29	Medium
152	153	20.0	223.0	11.5	68	9	30	Low

153 rows × 8 columns

Removing unnecessary columns

In [99]:

```
data.drop(data.iloc[:,[0]], axis=1, inplace=True)
data
```

Out[99]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.0	190.0	7.4	67	5	1	High
1	36.0	118.0	8.0	72	5	2	High
2	12.0	149.0	12.6	74	5	3	Low
3	18.0	313.0	11.5	62	5	4	NaN
4	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.0	193.0	6.9	70	9	26	Low
149	NaN	145.0	13.2	77	9	27	Low
150	14.0	191.0	14.3	75	9	28	High
151	18.0	131.0	8.0	76	9	29	Medium
152	20.0	223.0	11.5	68	9	30	Low

153 rows × 7 columns

Replacing null values by mean values

In [100]:

```
data.isnull().sum()
```

Out[100]:

```
Ozone       37
Solar.R      7
Wind         2
Temp         0
Month        0
Day          0
Humidity     8
dtype: int64
```

In [101]:

```
data.shape
```

Out[101]:

```
(153, 7)
```

In [102]:

```
data["Ozone"].fillna(data["Ozone"].mean(), inplace=True)
data["Solar.R"].fillna(data["Solar.R"].mean(), inplace=True)
data["Wind"].fillna(data["Wind"].mean(), inplace=True)
data
```

Out[102]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.00000	190.000000	7.4	67	5	1	High
1	36.00000	118.000000	8.0	72	5	2	High
2	12.00000	149.000000	12.6	74	5	3	Low
3	18.00000	313.000000	11.5	62	5	4	NaN
4	42.12931	185.931507	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	Low
149	42.12931	145.000000	13.2	77	9	27	Low
150	14.00000	191.000000	14.3	75	9	28	High
151	18.00000	131.000000	8.0	76	9	29	Medium
152	20.00000	223.000000	11.5	68	9	30	Low

153 rows × 7 columns

In [103]:

```
data["Humidity"].fillna("Medium", inplace=True)
```

In [104]:

```
data.isnull().sum()
```

Out[104]:

```
Ozone       0
Solar.R     0
Wind        0
Temp        0
Month       0
Day         0
Humidity    0
dtype: int64
```

Performing label encoding on Humidity column to convert categorical data to continuous data

In [105]:

```
from sklearn.preprocessing import LabelEncoder
```

In [106]:

```
le = LabelEncoder()
```

In [107]:

```
data["Humidity"] = le.fit_transform(data["Humidity"])
```

In [108]:

```
data
```

Out[108]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
0	41.00000	190.000000	7.4	67	5	1	0
1	36.00000	118.000000	8.0	72	5	2	0
2	12.00000	149.000000	12.6	74	5	3	1
3	18.00000	313.000000	11.5	62	5	4	2
4	42.12931	185.931507	14.3	56	5	5	0
...	...	...	...	...	...	...	...
148	30.00000	193.000000	6.9	70	9	26	1
149	42.12931	145.000000	13.2	77	9	27	1
150	14.00000	191.000000	14.3	75	9	28	0
151	18.00000	131.000000	8.0	76	9	29	2
152	20.00000	223.000000	11.5	68	9	30	1

153 rows × 7 columns

Declaring dependent and independent variables

In [109]:

```
# x = data.iloc[:, [0,1,2,3]]
# y = data["Humidity"]
x=data.iloc[:,:4]
y=data.iloc[:,4]
```

Splitting the data for training and testing

In [110]:

```
from sklearn.model_selection import train_test_split
```

In [111]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

Creating the model

In [112]:

```
from sklearn.linear_model import LinearRegression
```

In [113]:

```
model = LinearRegression()
```

Training the model

In [114]:

```
model.fit(x_train, y_train)
```

Out[114]:

```
▾ LinearRegression
LinearRegression()
```

Predicting the values

In [115]:

```
predictions = model.predict(x_test)
```

In [116]:

```
predictions
```

Out[116]:

```
array([6.90092094, 6.8263338 , 6.6633515 , 7.15291569, 7.97613425,
       7.71428955, 6.90833602, 6.94156525, 7.7366941 , 7.84569234,
       6.71208764, 6.99398299, 7.00130619, 7.43888818, 6.97260696,
       6.08581802, 7.04225697, 7.53206299, 6.61113791, 6.16899371,
       5.08065398, 7.3168772 , 6.70695618, 6.95913271, 7.06821777,
       7.50578449, 7.16076928, 6.7155374 , 5.8006455 , 7.49240097,
       6.83360675])
```

Calculating the performance metrics

In [118]:

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(predictions, y_test)
rmse = np.sqrt(mse)
```

In [123]:

```
print("MSE : " ,mse)
```

```
MSE :  2.0500714627933028
```

In [124]:

```
print("RMSE : " ,rmse)
```

```
RMSE :  1.4318070620000807
```

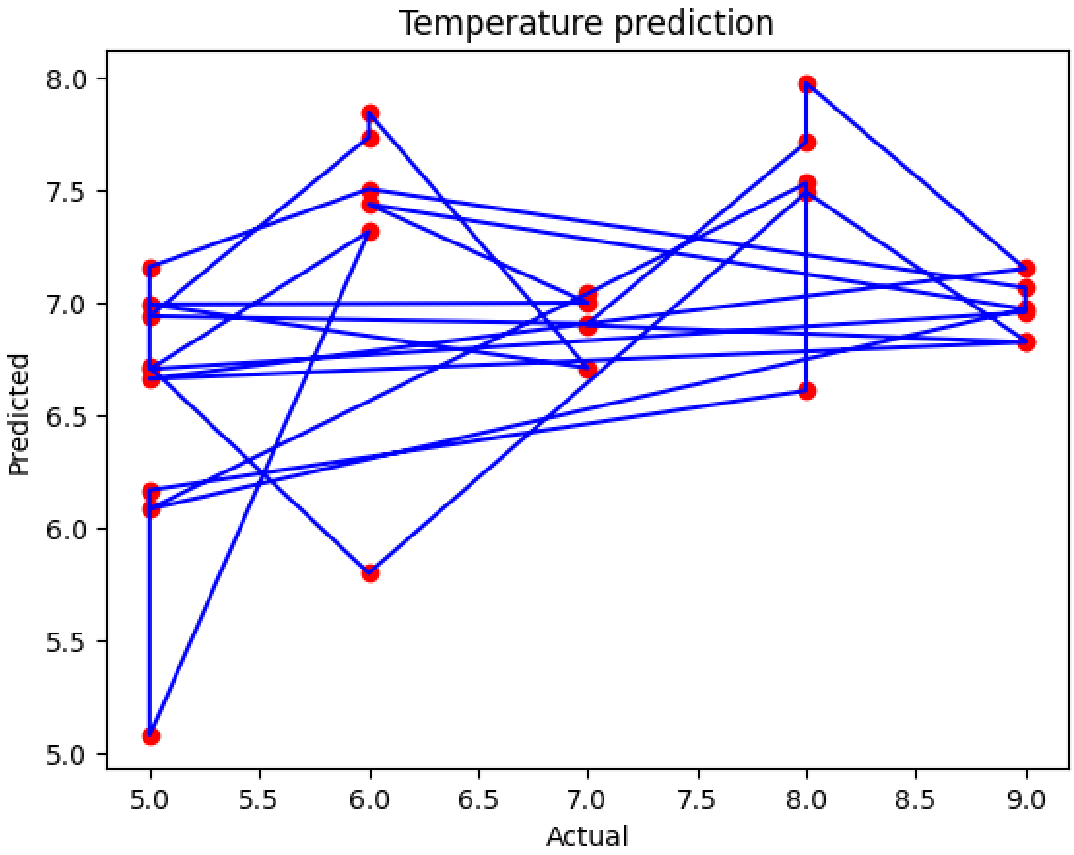
Model Visualization

In [125]:

```
import matplotlib.pyplot as plt
plt.title("Temperature prediction")
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.scatter(y_test, predictions, color='red')
plt.plot(y_test, model.predict(x_test), color='blue')
```

Out[125]:

```
[<matplotlib.lines.Line2D at 0x192e566d790>]
```


Temperature prediction
Predicted
Actual