DSBDA Assignment 2b

Perform the following operations using Python on the Air Quality and Heart Diesases data sets

1. Data Cleaning
2. Data Integration
3. Error Correcting
4. Data Model Building

Import Python Libraries

In [1]:

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

Reading the air_quality dataset -

In [2]:

```
df = pd.read_csv("air_quality.csv")
df
```

Out[2]:

	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	NaN
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

In [3]:

```
df.isnull().sum()
```

Out[3]:

```
Unnamed: 0     0
Ozone         37
Solar.R        7
Wind           2
Temp           0
Month          0
Day            0
Humidity      15
dtype: int64
```

Data Cleaning

Fill the null values with the mean and the mode accordingly -

Removing unwanted column from the datset

In [8]:

```
df = df.drop(['Unnamed: 0'], axis = 1)
df
```

Out[8]:

	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	NaN
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

In [9]:

```
df.columns
```

Out[9]:

```
Index(['Ozone', 'Solar.R', 'Wind', 'Temp', 'Month', 'Day', 'Humidity'], dtype='object')
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
Ozone        37
Solar.R       7
Wind          2
Temp          0
Month         0
Day           0
Humidity     15
dtype: int64
```

In [11]:

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

Now we have to use mode in the case of Humidity because it is categorical dataset and we need to convert it in numerical form.

In [15]:

```
df['Humidity'] = df['Humidity'].fillna(df['Humidity'].mode()[0])
```

In [16]:

```
df
```

Out[16]:

	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	High
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	High
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

Error Correcting

Look for missing values

In [17]:

```
df.isnull().sum()
```

Out[17]:

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

In [18]:
```
df.describe()
```

Out[18]:

	Ozone	Solar.R	Wind	Temp	Month	Day
count	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000
mean	42.129310	185.931507	9.945033	77.882353	6.993464	15.803922
std	28.693372	87.960267	3.520648	9.465270	1.416522	8.864520
min	1.000000	7.000000	1.700000	56.000000	5.000000	1.000000
25%	21.000000	120.000000	7.400000	72.000000	6.000000	8.000000
50%	42.129310	194.000000	9.700000	79.000000	7.000000	16.000000
75%	46.000000	256.000000	11.500000	85.000000	8.000000	23.000000
max	168.000000	334.000000	20.700000	97.000000	9.000000	31.000000

In [21]:
```
df.dtypes
```

Out[21]:
```
Ozone        float64
Solar.R      float64
Wind         float64
Temp           int64
Month          int64
Day            int64
Humidity      object
dtype: object
```

Data Transformation

In [22]:
```
from sklearn.preprocessing import LabelEncoder
lb = LabelEncoder()
```

In [23]:

```
df["Humidity"] = lb.fit_transform(df["Humidity"])
df
```

Out[23]:

	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	0
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	0
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

Data Model Building

Splitting x and y

In [24]:

```
x = df.iloc[:, [0, 3]].values
y = df['Humidity']
```

Spliting the training and testing data

In [25]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

Importing logistic regression

In [26]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
```

Out[26]:

```
LogisticRegression()
```

In [27]:

```
pred = model.predict(x_test);
pred
```

Out[27]:

```
array([0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0])
```

In [29]:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.38      0.81      0.52        16
           1       0.00      0.00      0.00        13
           2       0.20      0.10      0.13        10

    accuracy                           0.36        39
   macro avg       0.19      0.30      0.22        39
weighted avg       0.21      0.36      0.25        39


C:\Users\Rishabh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisio
n and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rishabh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisio
n and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rishabh\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precisio
n and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: