Assignment 2 - Group B

Problem Statement

Perform the following operations using Python on the Air quality data sets

a. Data cleaning

b. Data integration

c. Data transformaton

d. Error correcting

e. Data model building

In [175…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Reading data from CSV file

In [176…
```
Tej = pd.read_csv("C:\\Users\\Shree\\Desktop\\dsbdl_lab\\airquality.csv")
```

In [177…
```
Tej
```

Out[177]:

	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	Low
2	3	12.0	149.0	12.6	74	5	3	High
3	4	18.0	313.0	11.5	62	5	4	Medium
4	5	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...	...
148	149	30.0	193.0	6.9	70	9	26	High
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	High

153 rows × 8 columns

In [178…
```
Tej.head
```

Out[178]:
```
<bound method NDFrame.head of      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      Low
2             3   12.0    149.0  12.6    74      5    3     High
3             4   18.0    313.0  11.5    62      5    4   Medium
4             5    NaN      NaN  14.3    56      5    5     High
..          ...    ...      ...   ...   ...    ...  ...      ...
148         149   30.0    193.0   6.9    70      9   26     High
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     High

[153 rows x 8 columns]>
```

In [179…
```
Tej.shape
```

Out[179]:
```
(153, 8)
```

In [180…
```
Tej.isnull().sum()
```

Out[180]:
```
Unnamed: 0     0
Ozone         37
Solar.R        7
Wind           0
Temp           0
Month          0
Day            0
Humidity       4
dtype: int64
```

Data Cleaning

Removing unwanted columns

In [181…
```
Tej.drop(Tej.iloc[: , [0]], axis=1 , inplace=True)
```

In [182…
```
Tej
```

Out[182]:

	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	Low
2	12.0	149.0	12.6	74	5	3	High
3	18.0	313.0	11.5	62	5	4	Medium
4	NaN	NaN	14.3	56	5	5	High
...	...	...	...	...	...	...	...
148	30.0	193.0	6.9	70	9	26	High
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	High

153 rows × 7 columns

Replacing Numerical Null values

In [183...

```
Tej['Ozone']=Tej['Ozone'].fillna(Tej['Ozone'].mean())
Tej['Solar.R']=Tej['Solar.R'].fillna(Tej['Solar.R'].mean())
Tej["Wind"] = Tej["Wind"].fillna(Tej["Wind"].mean())
```

Replacing Categorical Null values

In [184...

```
Tej['Humidity']=Tej['Humidity'].fillna(Tej['Humidity'].mode()[0])
Tej.isnull().sum()
```

Out[184]:

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

In [185...

```
Tej.dtypes
```

Out[185]:

```
Ozone       float64
Solar.R     float64
Wind        float64
Temp          int64
Month         int64
Day           int64
Humidity     object
dtype: object
```

Data Transformation

Using Label Encoding on Humidity column

In [186…
```
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
Tej['Humidity']=label.fit_transform(Tej['Humidity'])
Tej['Humidity'].unique()
```

Out[186]:
```
array([0, 1, 2])
```

In [187…
```
Tej.dtypes
```

Out[187]:
```
Ozone       float64
Solar.R     float64
Wind        float64
Temp          int64
Month         int64
Day           int64
Humidity      int32
dtype: object
```

Data Integration

Subset Creation (Row-wise)

In [188…
```
#Subset-1
s1 = Tej.iloc[[1,2,3,6,12,28],:]
s1
```

Out[188]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
1	36.0	118.0	8.0	72	5	2	1
2	12.0	149.0	12.6	74	5	3	0
3	18.0	313.0	11.5	62	5	4	2
6	23.0	299.0	8.6	65	5	7	0
12	11.0	290.0	9.2	66	5	13	0
28	45.0	252.0	14.9	81	5	29	0

In [189…
```
#Subset-2
s2 = Tej.iloc[[70,81,95,105,123,137,149],:]
s2
```

Out[189]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
70	85.00000	175.000000	7.4	89	7	10	0
81	16.00000	7.000000	6.9	74	7	21	1
95	78.00000	185.931507	6.9	86	8	4	2
105	65.00000	157.000000	9.7	80	8	14	1
123	96.00000	167.000000	6.9	91	9	1	2
137	13.00000	112.000000	11.5	71	9	15	1
149	42.12931	145.000000	13.2	77	9	27	1

Merging Subsets

In [190…
```
merge = pd.concat([s1,s2])
```

In [191…
```
merge
```

Out[191]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
1	36.00000	118.000000	8.0	72	5	2	1
2	12.00000	149.000000	12.6	74	5	3	0
3	18.00000	313.000000	11.5	62	5	4	2
6	23.00000	299.000000	8.6	65	5	7	0
12	11.00000	290.000000	9.2	66	5	13	0
28	45.00000	252.000000	14.9	81	5	29	0
70	85.00000	175.000000	7.4	89	7	10	0
81	16.00000	7.000000	6.9	74	7	21	1
95	78.00000	185.931507	6.9	86	8	4	2
105	65.00000	157.000000	9.7	80	8	14	1
123	96.00000	167.000000	6.9	91	9	1	2
137	13.00000	112.000000	11.5	71	9	15	1
149	42.12931	145.000000	13.2	77	9	27	1

Deriving correlation between Columns

In [192…
```
corr = Tej.corr()
```

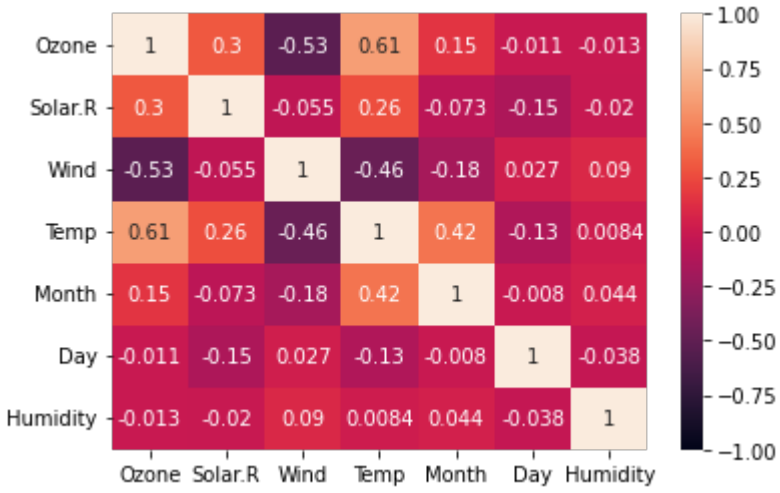
In [193…
```
corr
```

Out[193]:

	Ozone	Solar.R	Wind	Temp	Month	Day	Humidity
Ozone	1.000000	0.302970	-0.530936	0.608742	0.149081	-0.011355	-0.012681
Solar.R	0.302970	1.000000	-0.055245	0.262569	-0.072904	-0.145621	-0.020428
Wind	-0.530936	-0.055245	1.000000	-0.457988	-0.178293	0.027181	0.090264
Temp	0.608742	0.262569	-0.457988	1.000000	0.420947	-0.130593	0.008397
Month	0.149081	-0.072904	-0.178293	0.420947	1.000000	-0.007962	0.043569
Day	-0.011355	-0.145621	0.027181	-0.130593	-0.007962	1.000000	-0.038271
Humidity	-0.012681	-0.020428	0.090264	0.008397	0.043569	-0.038271	1.000000

In [194…
```
import seaborn as sns
sns.heatmap(corr, vmin = -1, vmax = 1, annot=True)
```

Out[194]:
```
<AxesSubplot:>
```



Building Data Model

Using Linear Regression

In [195…
```
x = Tej[["Ozone"]]
y = Tej[["Temp"]]
```

In [196…
```
from sklearn.model_selection import train_test_split
```

In [197…
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

In [198…
```
from sklearn.linear_model import LinearRegression
```

In [199…
```
lr = LinearRegression()
```

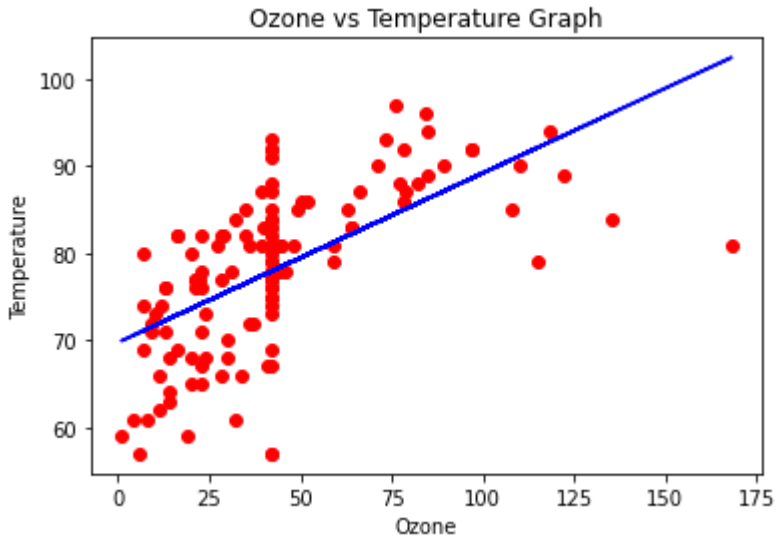
In [200…
```
model = lr.fit(x_train, y_train)
```

In [201…
```
y_predict = model.predict(x_test)
```

Plotting Graph

In [202…
```
import matplotlib.pyplot as plt
```

In [203…
```
plt.scatter(x_train, y_train, color="red")
plt.plot(x_train, lr.predict(x_train), color="blue")
plt.xlabel("Ozone")
plt.ylabel("Temperature")
plt.title("Ozone vs Temperature Graph")
plt.show()
```



Calculating Metrics

In [213…
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
import numpy as np
```

In [214…
```
MSE = mean_squared_error(y_test,y_predict)
MAE = mean_absolute_error(y_test,y_predict)
r2_score = r2_score(y_test,y_predict)
RMSE = np.sqrt(MSE)
```

In [215…
```
print("MSE : {} \nRMSE : {} \nMAE : {} \nR2 Score : {}".format(MSE, RMSE, MAE, r2_s
```

```
MSE : 58.75683748803407
RMSE : 7.665300873940571
MAE : 5.3610432690122485
R2 Score : 0.33559267613488397
```

In [ ]:

In [ ]: