Cheat number :5

5. Perform the following operations using Python on the Air quality data set

a. Data cleaning

b. Data integration

c. Data transformation

d. Error correcting

1] import pandas as pd

import numpy as np

2] df = pd.read\_csv('airquality\_data.csv', encoding='cp1252')

3] df.head()

4]df.info()

5]df.columns

**a. Data cleaning**

6]

# Change data type from float64 to float32 for Space Complexity

df['so2'] = df['so2'].astype('float32')

df['no2'] = df['no2'].astype('float32')

df['rspm'] = df['rspm'].astype('float32')

df['spm'] = df['spm'].astype('float32')

df['date'] = df['date'].astype('string') ……..here if not run then use astype(‘str’)

df.info()

7] df=df.drop\_duplicates()

8] df.isna().sum()

9] percent\_missing = df.isnull().sum() \* 100 / len(df)

10] percent\_missing.sort\_values(ascending=False)

11] df=df.drop(['stn\_code', 'agency','sampling\_date','location\_monitoring\_station','pm2\_5'], axis = 1)

12] df.head()

13] df.columns

14] col\_var = ['state', 'location', 'type','date']

col\_num = ['so2','no2','rspm','spm']

15]for col in df.columns:

if df[col].dtype == 'object' or df[col].dtype == 'string': ……here also not accepted then str

df[col] = df[col].fillna(df[col].mode()[0])

else:

df[col] = df[col].fillna(df[col].mean())

16]df.isna().sum()

17] df

**Data integration:**

19] subSet1 = df[['state', 'type']]

subSet2 = df[['state','location']]

20] subSet1.head()

21] subSet2.head()

22] concatenated\_df = pd.concat([subSet1, subSet2], axis=1)

23] concatenated\_df

**d. Error correcting:**

24] def remove\_outliers(column):

Q1 = column.quantile(0.25)

Q3 = column.quantile(0.75)

IQR = Q3 - Q1

threshold = 1.5 \* IQR

outlier\_mask = (column < Q1 - threshold) | (column > Q3 + threshold)

return column[~outlier\_mask]

25]df.columns

26] # Remove outliers for each column using a loop

col\_name = ['so2', 'no2', 'rspm', 'spm']

for col in col\_name:

df[col] = remove\_outliers(df[col])

27] import seaborn as sns

import matplotlib.pyplot as plt

28] plt.figure(figsize=(10, 6)) # Adjust the figure size if needed

for col in col\_name:

sns.boxplot(data=df[col])

plt.title(col)

plt.show()

**c. Data transformation**

29] from sklearn.preprocessing import LabelEncoder

col\_label= ['state','location','type']

# Initialize LabelEncoder

encoder = LabelEncoder()

# Iterate over columns

for col in df.columns:

# Fit and transform the column

df[col] = encoder.fit\_transform(df[col])

30] df