1.Perform the following operations using Python on the Facebook metrics

datasets

a.Create data subsets

b.Merge Data

c. Transposing Data

```
#subset
df1=df[['Page total likes','Type','Category']].loc[0:15]
df1
#subset2
df2=df[['Page total likes','Type']].loc[18:40]
df2
#merge
merg=pd.concat([df1,df2])
merg
#transpose
transpose=df.transpose()
transpose
```

2.PerformthefollowingoperationsusingPythonontheFacebookmetrics

datasets

a.Createdatasubsets

b.SortData

c.ShapeandreshapeData

```
#sorting
sort=df.sort_values('like',ascending=True)
sort
df3=df[['like','Type']].loc[1:15]
df3
sort1=df3.sort_values('like',ascending=True)
sort1
pd.pivot_table(df, index=['Post Month','Paid'],values=['share','Post Weekday'])
#shaping
stack_df=df.stack()
stack_df
df.isna()
```

3.PerformthefollowingoperationsusingPythonontheAirqualityandHeart

Diseasesdatasets

a.Datacleaning

b.Dataintegration

c.Datatransformation

#a.data cleaning

```
df1.isnull().sum() #checking null values
droped_df1=df1.dropna() #droping null values
droped_df1.isnull().sum()
df2=pd.read_csv("heart.csv")
df2
cleaned_df2 = df2.isnull().sum() #checking null values
cleaned_df2
#b.DATA Integration

df3=df2[['age','sex','trestbps','chol']].loc[0:20]
df3
df4=df2[['age','sex','trestbps','chol']].loc[21:40]
df4
integrated_df=pd.concat([df3,df4])
integrated_df
#c. Data Transformation

df2.loc[df2['sex']==1, 'sex']='M' #replacing 1 with M

df2.loc[df2['sex']==0, 'sex']='F' #replacing 0 with F
from sklearn.preprocessing  import LabelEncoder
```

```
labelencoder=LabelEncoder()
df2['sex']=labelencoder.fit_transform(df2['sex'])
df2      #used to encode categorical variables into numerical labels
#Replacing the Object dtype of Date to Date dtype
df1['Date'] = pd.to_datetime(df1['Date'])

#To Replace the Comma's with Dot
df1.replace(to_replace=',',value='.', regex=True, inplace=True)
df1
df1.head()
```

4.PerformthefollowingoperationsusingPythonontheAirqualityandHeart

Diseasesdatasets

a.Datatransformation

b.Errorcorrecting

c.Datamodelbuilding

```
import pandas as pd #Python library used for working with data sets

import numpy as np #Python library used for working with arrays

import seaborn as sn # library for making statistical graphics in Python

import random as rn

import matplotlib.pyplot as mat #used to create 2D graphs and plots by using python scripts

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import GaussianNB,MultinomialNB

from sklearn.metrics import accuracy_score

DataFrame1=pd.read_csv('heart.csv') #Read a comma-separated values (csv) file into DataFrame

DataFrame1

DataFrame2=pd.read_csv('AirQuality.csv',sep=';') #Read a comma-separated values (csv) file into
DataFrame

DataFrame2

DataFrame1.isna().sum() #Detect missing values for an array-like object

#.sum() sums up the numbers in the list

DataFrame1.loc[DataFrame1['sex']==1,'sex']='M' #Replacing 1 with M

DataFrame1.loc[DataFrame1['sex']==0,'sex']='F' #Replacing 0 with F

DataFrame1.head()

from sklearn.preprocessing import LabelEncoder

labelencoder=LabelEncoder()

DataFrame1["sex"]=labelencoder.fit_transform(DataFrame1["sex"])

DataFrame1 #used to encode categorical variables into numerical labels that is transform

DataFrame1[DataFrame1['ca']==4]
```

DataFrame1.loc[DataFrame1['ca']==4,'ca']=np.NaN #It locates the rows where the value in the 'ca' column is equal to 4 and replaces those values with NaN (Not a Number)

DataFrame1 = DataFrame1.fillna(DataFrame1.median())

#The statement DataFrame1 = DataFrame1.fillna(DataFrame1.median()) fills the missing values in DataFrame1 with the median value of each column.

DataFrame1.isnull().sum()

Model Building

X_train, X_test, y_train, y_test = train_test_split(DataFrame1.iloc[:,:-1], DataFrame1.iloc[:,-1], test_size = 0.3, random_state = 0)

X_train.shape, X_test.shape,y_train.shape

gnb = GaussianNB()

gnb.fit(X_train, y_train)

#fitting a Gaussian Naive Bayes (GNB) model on the training data

y_pred = gnb.predict(X_test)

y_pred

print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))

5.VisualizethedatausingPythonlibrariesmatplotlib,seabornbyplottingthe

graphsfortheAirqualityandHeartDiseasesdatasets

a.Datacleaning

b.Dataintegration

```
import pandas as pd #Python library used for working with data sets

import numpy as np #Python library used for working with arrays

import seaborn as sn # library for making statistical graphics in Python

import matplotlib.pyplot as mat #used to create 2D graphs and plots by using python scripts

df1=pd.read_csv('D:/dsbda basic/DSBDA-Final-Problem-Statements-main/PS5/heart.csv')

df1.head()

df2=pd.read_csv('D:/dsbda basic/DSBDA-Final-Problem-Statements-main/PS5/AirQuality.csv',
sep=';')

df2.head()
```

1.Data Cleaning

```
df3=df2.iloc[:,:15] #iloc stands for “integer location”.

#It is used to select rows and columns from a Pandas DataFrame or a Series using integer-based
indexing

df3

df3.isna().sum().sum()

df4=df3.dropna()

df4

#Replacing the Object dtype of Date to Date dtype

df4['Date']=pd.to_datetime(df4['Date'])

#To Replace the Comma's with Dot

df4.replace(to_replace=',',value='.',regex=True,inplace=True)

df4

df4.drop_duplicates(inplace=True)

df4 #Drop Duplicates
```

```
# b. Data Integration
DataSet1=df4[['Date','Time','T','RH','AH']].loc[0:50]
DataSet1.head()
DataSet2=df4[['Date','Time','T','RH','AH']].loc[51:100]
DataSet2.head()
integrated=pd.concat([DataSet1,DataSet2])
integrated
DataSet1=df4[['Date','Time','T','RH','AH']].loc[0:50]
DataSet1.head()
DataSet2=df4[['Date','Time','T','RH','AH']].loc[51:100]
DataSet2.head()
integrated=pd.concat([DataSet1,DataSet2])
integrated
# Data Visaulization
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.subplot(2, 2, 1)  # Create a subplot (2 rows, 2 columns, subplot 1)
plt.hist(df1['age'], bins=10, edgecolor='k')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
# Bar plot of sex
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
sex_counts = df1['sex'].value_counts()
plt.bar(sex_counts.index, sex_counts.values)
plt.xlabel('Sex (0 = female, 1 = male)')
```

```
plt.ylabel('Count')
plt.title('Gender Distribution')
# Scatter plot of age and cholesterol
plt.subplot(2, 2, 3)  # Create a subplot (2 rows, 2 columns, subplot 3)
plt.scatter(df1['age'], df1['chol'], alpha=0.5)
plt.xlabel('Age')
plt.ylabel('Cholesterol')
plt.title('Age vs. Cholesterol')
plt.subplot(2,2,4)
plt.boxplot([df1[df1['target']==0]['trestbps'],
        df1[df1['target']==1]['trestbps']],
        labels=['No Disease','Disease'])
plt.xlabel('Disease')
plt.ylabel('Restng Blood Pressure')
plt.title('Resting Blood Pressure by Target')

plt.tight_layout()
#Visaulization for AirQuality.csv
# Scatter plot of NOx(GT) vs. NO2(GT)
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
plt.scatter(df2['NOx(GT)'], df2['NO2(GT)'], alpha=0.5)
plt.xlabel('NOx(GT)')
plt.ylabel('NO2(GT)')
plt.title('NOx(GT) vs. NO2(GT)')
# Scatter plot of NO2(GT) vs. PT08.S4(NO2)
plt.subplot(2, 2, 4)  # Create a subplot (2 rows, 2 columns, subplot 4)
plt.scatter(df2['NO2(GT)'], df2['PT08.S4(NO2)'], alpha=0.5)
```

```
plt.xlabel('NO2(GT)')
plt.ylabel('PT08.S4(NO2)')
plt.title('NO2(GT) vs. PT08.S4(NO2)')

plt.tight_layout()  # Adjust the spacing between subplots
plt.show()  # Display the plots
# Scatter plot of PT08.S1(CO) vs. PT08.S2(NMHC)
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
plt.scatter(df2['PT08.S1(CO)'], df2['PT08.S2(NMHC)'], alpha=0.5)
plt.xlabel('PT08.S1(CO)')
plt.ylabel('PT08.S2(NMHC)')
plt.title('PT08.S1(CO) vs. PT08.S2(NMHC)')
# Box plot of NOx(GT)
plt.subplot(2, 2, 3)  # Create a subplot (2 rows, 2 columns, subplot 3)
plt.boxplot(df2['NOx(GT)'])
plt.xlabel('NOx(GT)')
plt.ylabel('Value')
plt.title('NOx(GT)')
```

8.VisualizethedatausingPythonlibrariesmatplotlib,seabornbyplottingthe

graphsfortheAirqualityandHeartDiseasesdatasets

a.Datatransformation

b.Errorcorrecting

c.Datamodelbuilding

```
import pandas as pd #Python library used for working with data sets

import numpy as np #Python library used for working with arrays

import seaborn as sn # library for making statistical graphics in Python

import random as rn

import matplotlib.pyplot as mat #used to create 2D graphs and plots by using python scripts

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import GaussianNB,MultinomialNB

from sklearn.metrics import accuracy_score

DataFrame1=pd.read_csv('D:/dsbda basic/DSBDA-Final-Problem-Statements-
main/PS8/heart.csv') #Read a comma-separated values (csv) file into DataFrame

DataFrame1

DataFrame2=pd.read_csv('D:/dsbda basic/DSBDA-Final-Problem-Statements-
main/PS8/AirQuality.csv',sep=';') #Read a comma-separated values (csv) file into DataFrame

DataFrame2

DataFrame1.isna().sum().sum() #Detect missing values for an array-like object

#.sum() sums up the numbers in the list


# a. Data Transformation

DataFrame1.loc[DataFrame1['sex']==1,'sex']='M' #Replacing 1 with M

DataFrame1.loc[DataFrame1['sex']==0,'sex']='F' #Replacing 0 with F

DataFrame1.head()

from sklearn.preprocessing import LabelEncoder

labelencoder=LabelEncoder()
```

```
DataFrame1["sex"]=labelencoder.fit_transform(DataFrame1["sex"])

DataFrame1 #used to encode categorical variables into numerical labels that is transform


# b. Error Correction

DataFrame1[DataFrame1['ca']==4]

DataFrame1.loc[DataFrame1['ca']==4,'ca']=np.NaN  #It locates the rows where the value in the 'ca' column is equal to 4 and replaces those values with NaN (Not a Number)

DataFrame1 = DataFrame1.fillna(DataFrame1.median())

#The statement DataFrame1 = DataFrame1.fillna(DataFrame1.median()) fills the missing values in DataFrame1 with the median value of each column.

DataFrame1.isnull().sum()

DataFrame1


# c. Model Building

X_train, X_test, y_train, y_test = train_test_split(DataFrame1.iloc[:,:-1], DataFrame1.iloc[:,-1], test_size = 0.3, random_state = 0)

X_train.shape, X_test.shape,y_train.shape

gnb = GaussianNB()

gnb.fit(X_train, y_train)

#fitting a Gaussian Naive Bayes (GNB) model on the training data


y_pred = gnb.predict(X_test)

y_pred

print('Model accuracy score: {0:0.4f}'. format(accuracy_score(y_test, y_pred)))


# data visualization

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
```

```
plt.subplot(2, 2, 1)  # Create a subplot (2 rows, 2 columns, subplot 1)
plt.hist(DataFrame1['age'], bins=10, edgecolor='k')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
# Bar plot of sex
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
sex_counts = DataFrame1['sex'].value_counts()
plt.bar(sex_counts.index, sex_counts.values)
plt.xlabel('Sex (0 = female, 1 = male)')
plt.ylabel('Count')
plt.title('Gender Distribution')
# Scatter plot of age and cholesterol
plt.subplot(2, 2, 3)  # Create a subplot (2 rows, 2 columns, subplot 3)
plt.scatter(DataFrame1['age'], DataFrame1['chol'], alpha=0.5)
plt.xlabel('Age')
plt.ylabel('Cholesterol')
plt.title('Age vs. Cholesterol')
plt.subplot(2,2,4)
plt.boxplot([DataFrame1[DataFrame1['target']==0]['trestbps'],
        DataFrame1[DataFrame1['target']==1]['trestbps']],
        labels=['No Disease','Disease'])
plt.xlabel('Disease')
plt.ylabel('Restng Blood Pressure')
plt.title('Resting Blood Pressure by Target')
```

```
plt.tight_layout()
#Visaulization for AirQuality.csv
# Scatter plot of NOx(GT) vs. NO2(GT)
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
plt.scatter(DataFrame2['NOx(GT)'], DataFrame2['NO2(GT)'], alpha=0.5)
plt.xlabel('NOx(GT)')
plt.ylabel('NO2(GT)')
plt.title('NOx(GT) vs. NO2(GT)')
# Scatter plot of NO2(GT) vs. PT08.S4(NO2)
plt.subplot(2, 2, 4)  # Create a subplot (2 rows, 2 columns, subplot 4)
plt.scatter(DataFrame2['NO2(GT)'], DataFrame2['PT08.S4(NO2)'], alpha=0.5)
plt.xlabel('NO2(GT)')
plt.ylabel('PT08.S4(NO2)')
plt.title('NO2(GT) vs. PT08.S4(NO2)')

plt.tight_layout()  # Adjust the spacing between subplots
plt.show()  # Display the plots
# Scatter plot of PT08.S1(CO) vs. PT08.S2(NMHC)
plt.subplot(2, 2, 2)  # Create a subplot (2 rows, 2 columns, subplot 2)
plt.scatter(DataFrame2['PT08.S1(CO)'], DataFrame2['PT08.S2(NMHC)'], alpha=0.5)
plt.xlabel('PT08.S1(CO)')
plt.ylabel('PT08.S2(NMHC)')
plt.title('PT08.S1(CO) vs. PT08.S2(NMHC)')
# Box plot of NOx(GT)
plt.subplot(2, 2, 3)  # Create a subplot (2 rows, 2 columns, subplot 3)
plt.boxplot(DataFrame2['NOx(GT)'])
plt.xlabel('NOx(GT)')
```

```
plt.ylabel('Value')
plt.title('NOx(GT)')
```