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

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("/content/India Air Quality Data (2).csv")
d2=pd.read_csv("/content/heart (1).csv")
```

```
df
```

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	locat
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN	
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	
49000	NaN	23-03-05	Chandigarh	Chandigarh	NaN	Residential and others	6.0	15.0	47.0	125.0	
49001	NaN	25-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	12.0	54.0	161.0	
49002	NaN	28-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	10.0	116.0	196.0	
49003	NaN	30-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	9.0	38.0	154.0	
49004	NaN	4/1/2005	Chandigarh	Chandigarh	NaN	Residential and others	10.0	27.0	43.0	152.0	

49005 rows × 13 columns

```
d2
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

a) Data Cleaning

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

```
stn_code                       15764
sampling_date                      0
state                              0
location                           0
agency                         16355
type                             994
so2                             1312
no2                              858
rspm                            2696
spm                            28659
location_monitoring_station     2537
pm2_5                          49005
date                               1
dtype: int64
```

```
df.dropna(thresh=0.3*len(df),axis=1,inplace=True)
df
```

	stn_code	sampling_date	state	location	agency	type	so2	no2	rspm	spm	locat
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4	NaN	NaN	
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0	NaN	NaN	
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5	NaN	NaN	
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7	NaN	NaN	
4	151.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	4.7	7.5	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	
49000	NaN	23-03-05	Chandigarh	Chandigarh	NaN	Residential and others	6.0	15.0	47.0	125.0	
49001	NaN	25-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	12.0	54.0	161.0	
49002	NaN	28-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	10.0	116.0	196.0	
49003	NaN	30-03-05	Chandigarh	Chandigarh	NaN	Residential and others	NaN	9.0	38.0	154.0	
49004	NaN	4/1/2005	Chandigarh	Chandigarh	NaN	Residential and others	10.0	27.0	43.0	152.0	

49005 rows × 12 columns

```
d2.duplicated().sum()
```

```
723
```

```
d2.drop_duplicates()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
723	68	0	2	120	211	0	0	115	0	1.5	1	0	2	1
733	44	0	2	108	141	0	1	175	0	0.6	1	0	2	1
739	52	1	0	128	255	0	1	161	1	0.0	2	1	3	0
843	59	1	3	160	273	0	0	125	0	0.0	2	0	2	0
878	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

302 rows × 14 columns

```
d2
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

```
d2.duplicated().sum()
```

```
723
```

```
df1=d2[['age','sex','cp','thal']].loc[0:15]
df1
```

	age	sex	cp	thal
0	52	1	0	3
1	53	1	0	3
2	70	1	0	3
3	61	1	0	3
4	62	0	0	2
5	58	0	0	2
6	58	1	0	1
7	55	1	0	3
8	46	1	0	3
9	54	1	0	2
10	71	0	0	2
11	43	0	0	3
12	34	0	1	2
13	51	1	0	3
14	52	1	0	0
15	34	0	1	2

```
df2=d2[['age','sex','cp','thal']].loc[16:30]
df2
```

	age	sex	cp	thal
16	51	0	2	2
17	54	1	0	3
18	50	0	1	2
19	58	1	2	2
20	60	1	2	2

b) Data Integration

22 45 1 0 2

```
merge = pd.merge(df1,df2,on='age',how='inner')
merge
```

	age	sex_x	cp_x	thal_x	sex_y	cp_y	thal_y
0	61	1	0	3	0	0	3
1	58	0	0	2	1	2	2
2	58	0	0	2	0	1	2
3	58	1	0	1	1	2	2
4	58	1	0	1	0	1	2
5	55	1	0	3	0	0	2
6	54	1	0	2	1	0	3
7	51	1	0	3	0	2	2

```
d2['target']=d2['target'].apply(lambda x :1 if x>0 else 0)
```

```
d2
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

```
del df['rspm']
df
```

	stn_code	sampling_date	state	location	agency	type	so2	no2
0	150.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	4.8	17.4
1	151.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Industrial Area	3.1	7.0
2	152.0	February - M021990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.2	28.5
3	150.0	March - M031990	Andhra Pradesh	Hyderabad	NaN	Residential, Rural and other Areas	6.3	14.7

Data Model Building

```
from sklearn.model_selection import train_test_split
x=merge.drop(['age'],axis=1)
x
```

	sex_x	cp_x	thal_x	sex_y	cp_y	thal_y
0	1	0	3	0	0	3
1	0	0	2	1	2	2
2	0	0	2	0	1	2
3	1	0	1	1	2	2
4	1	0	1	0	1	2
5	1	0	3	0	0	2
6	1	0	2	1	0	3
7	1	0	3	0	2	2

```
y=merge['thal_y']
y
```

```
0    3
1    2
2    2
3    2
4    2
5    2
6    3
7    2
Name: thal_y, dtype: int64
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=40)
```

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
```

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

```
▾ LogisticRegression
LogisticRegression()
```

```
from sklearn.metrics import classification_report,confusion_matrix
y_pred=logreg.predict(x_test)
```

```
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[3]]
              precision    recall  f1-score   support

           2       1.00      1.00      1.00         3

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3
```

✓ 0s completed at 12:51 PM ● ✕