Assignment 5

Details

1. Author : Vishwajit Shelke
2. Roll Number : 33172
3. Batch : K-9
4. Class : TE9

Problem Statement

Perform the following operations using Python on the Air quality and Heart Diseases data sets

1. Data cleaning
2. Data integration
3. Data transformation
4. Error correcting
5. Data model building

Implementation details

1. Dataset URL : https://archive.ics.uci.edu/ml/datasets/Heart+Disease (https://archive.ics.uci.edu/ml/datasets/Heart+Disease)
2. Python version : 3.7.4
3. Imports :
   A. pandas
   B. numpy
   C. matplotlib
   D. seaborn

Dataset details

1. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date.
2. The "goal" field refers to the presence of heart disease in the patient.
3. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).
4. The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

Importing libraries

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

reading csv file

In [ ]:

```
dataset = pd.read_csv('heart.csv')
```

In [ ]:

```
dataset.head(10)
```

Out[3]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	S
0	1	63	1	typical	145	233	1	2	150	0	2.3	
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	
5	6	56	1	nontypical	120	236	0	0	178	0	0.8	
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6	
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6	
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4	
9	10	53	1	asymptomatic	140	203	1	2	155	1	3.1	

In [ ]:

```
dataset2 = pd.read_csv('Heart.csv')
```

In [ ]:

```
dataset2.head(10)
```

Out[5]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	S
0	1	63	1	typical	145	233	1	2	150	0	2.3	
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	
5	6	56	1	nontypical	120	236	0	0	178	0	0.8	
6	7	62	0	asymptomatic	140	268	0	2	160	0	3.6	
7	8	57	0	asymptomatic	120	354	0	0	163	1	0.6	
8	9	63	1	asymptomatic	130	254	0	2	147	0	1.4	
9	10	53	1	asymptomatic	140	203	1	2	155	1	3.1	

checking fo rthe null values in the dataset

In [ ]:

```
dataset2.isna().sum()
```

Out[6]:

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            4
Thal          2
AHD           0
dtype: int64
```

dropping the rows with null values

In [ ]:

```
dataset2 = dataset2.dropna(axis=0)
```

rechecking if there are any null values in the dataset

In [ ]:

```
dataset2.isnull().sum()
```

Out[8]:

```
Unnamed: 0    0
Age           0
Sex           0
ChestPain     0
RestBP        0
Chol          0
Fbs           0
RestECG       0
MaxHR         0
ExAng         0
Oldpeak       0
Slope         0
Ca            0
Thal          0
AHD           0
dtype: int64
```

Statistical Analysis on the dataset

In [ ]:

```
dataset2.describe()
```

Out[9]:

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	
count	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	297.000000	2
mean	150.673401	54.542088	0.676768	131.693603	247.350168	0.144781	0.996633	1
std	87.323283	9.049736	0.468500	17.762806	51.997583	0.352474	0.994914	
min	1.000000	29.000000	0.000000	94.000000	126.000000	0.000000	0.000000	
25%	75.000000	48.000000	0.000000	120.000000	211.000000	0.000000	0.000000	1
50%	150.000000	56.000000	1.000000	130.000000	243.000000	0.000000	1.000000	1
75%	226.000000	61.000000	1.000000	140.000000	276.000000	0.000000	2.000000	1
max	302.000000	77.000000	1.000000	200.000000	564.000000	1.000000	2.000000	2

Performing boxplot on the dataset

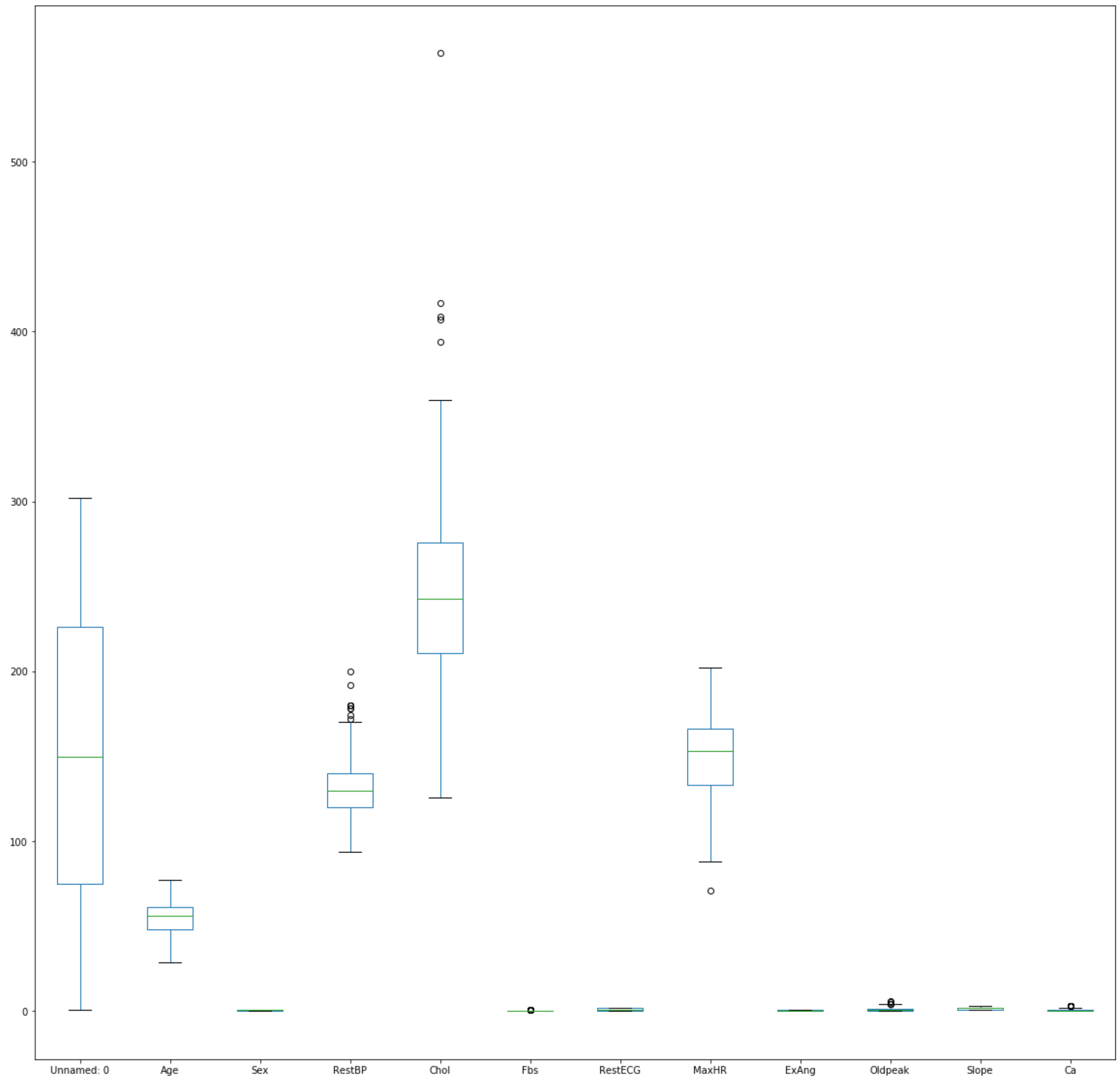
In [ ]:

```
plt.figure(figsize=(9,3))
dataset2.plot(kind='box',figsize=(20,20))
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ead694b608>
```

```
<Figure size 648x216 with 0 Axes>
```



In [ ]:

```
percentile25 = dataset2['Chol'].quantile(0.25)
percentile75 = dataset2['Chol'].quantile(0.75)
```

In [ ]:

```
iqr = percentile75 - percentile25
```

In [ ]:

```
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr
```

In [ ]:

```
dataset2[dataset2['Chol'] > upper_limit]
dataset2[dataset2['Chol'] < lower_limit]
```

Out[15]:

Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope

In [ ]:

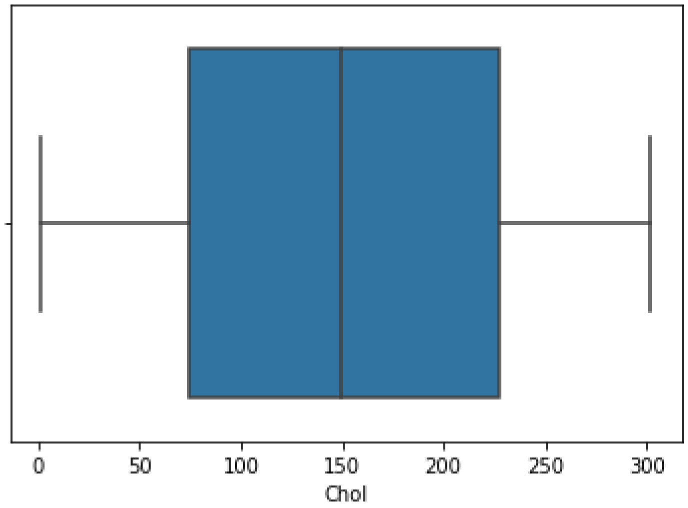
```
dataset2['Chol'] = dataset2[dataset2['Chol'] < upper_limit]
```

In [ ]:

```
sns.boxplot(dataset2['Chol'])
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ead7582b88>
```



In [ ]:

```
percentile25 = dataset2['RestBP'].quantile(0.25)
percentile75 = dataset2['RestBP'].quantile(0.75)
```

In [ ]:

```
iqr = percentile75 - percentile25
```

In [ ]:

```
upper_limit = percentile75 + 1.5 * iqr
lower_limit = percentile25 - 1.5 * iqr
```

In [ ]:

```
dataset2[dataset2['RestBP'] > upper_limit]
dataset2[dataset2['RestBP'] < lower_limit]
```

Out[21]:

Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope

In [ ]:

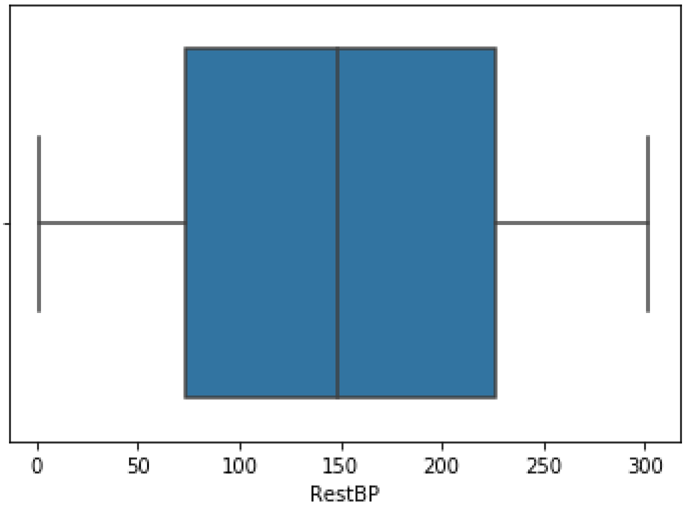
```
dataset2['RestBP'] = dataset2[dataset2['RestBP'] < upper_limit]
```

In [ ]:

```
sns.boxplot(dataset2['RestBP'])
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ead6da4d08>
```



In [ ]:

```

```

In [ ]:

```
print(dataset2.ChestPain.unique())
print(dataset2.Thal.unique())
print(dataset2.AHD.unique())
```

```
['typical' 'asymptomatic' 'nonanginal' 'nontypical']
['fixed' 'normal' 'reversable']
['No' 'Yes']
```

segragating the categorical values into multiple columns

In [ ]:

```
heart_encoding = pd.get_dummies(dataset2[['ChestPain', 'Thal', 'AHD']])
heart_final = pd.concat([dataset2, heart_encoding],1)
heart_final = heart_final.drop(['ChestPain', 'Thal', 'AHD'], axis = 1)
heart_final.head(10)
```

Out[25]:

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	...	Ca	ChestPa
0	1	63	1	1	1	1	2	150	0	2.3	...	0.0	
1	2	67	1	2	2	0	2	108	1	1.5	...	3.0	
2	3	67	1	3	3	0	2	129	1	2.6	...	2.0	
3	4	37	1	4	4	0	0	187	0	3.5	...	0.0	
4	5	41	0	5	5	0	2	172	0	1.4	...	0.0	
5	6	56	1	6	6	0	0	178	0	0.8	...	0.0	
6	7	62	0	7	7	0	2	160	0	3.6	...	2.0	
7	8	57	0	8	8	0	0	163	1	0.6	...	0.0	
8	9	63	1	9	9	0	2	147	0	1.4	...	1.0	
9	10	53	1	10	10	1	2	155	1	3.1	...	0.0	

10 rows × 21 columns

In [ ]:

```
heart_final.Sex.value_counts()
```

Out[26]:

```
1    201
0     96
Name: Sex, dtype: int64
```

In [ ]:

```
pd.crosstab(heart_final.AHD_Yes,heart_final.Sex)
```

Out[27]:

Sex	0	1
AHD_Yes		
0	71	89
1	25	112

In [ ]:

```
heart_final.columns
```

Out[28]:

```
Index(['Unnamed: 0', 'Age', 'Sex', 'RestBP', 'Chol', 'Fbs', 'RestECG', 'Ma
xHR',
       'ExAng', 'Oldpeak', 'Slope', 'Ca', 'ChestPain_asymptomatic',
       'ChestPain_nonanginal', 'ChestPain_nontypical', 'ChestPain_typica
l',
       'Thal_fixed', 'Thal_normal', 'Thal_reversable', 'AHD_No', 'AHD_Ye
s'],
      dtype='object')
```

In [ ]:

```
heart_final.dtypes
```

Out[29]:

```
Unnamed: 0                  int64
Age                         int64
Sex                         int64
RestBP                     object
Chol                       object
Fbs                         int64
RestECG                     int64
MaxHR                       int64
ExAng                       int64
Oldpeak                   float64
Slope                       int64
Ca                        float64
ChestPain_asymptomatic      uint8
ChestPain_nonanginal        uint8
ChestPain_nontypical        uint8
ChestPain_typical           uint8
Thal_fixed                  uint8
Thal_normal                 uint8
Thal_reversable             uint8
AHD_No                      uint8
AHD_Yes                     uint8
dtype: object
```

In [ ]:

```
df = heart_final.drop('AHD_Yes', axis=1)
df_norm = (df-df.min())/(df.max()-df.min())
df_norm = pd.concat((df_norm, heart_final.AHD_Yes), 1)
```

In [ ]:

```
df_norm.head(10)
```

Out[31]:

	Unnamed: 0	Age	Sex	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak
0	0.000000	0.708333	1.0	0	0	1.0	1.0	0.603053	0.0	0.370968
1	0.003322	0.791667	1.0	0.00332226	0.00332226	0.0	1.0	0.282443	1.0	0.241935
2	0.006645	0.791667	1.0	0.00664452	0.00664452	0.0	1.0	0.442748	1.0	0.419355
3	0.009967	0.166667	1.0	0.00996678	0.00996678	0.0	0.0	0.885496	0.0	0.564516
4	0.013289	0.250000	0.0	0.013289	0.013289	0.0	1.0	0.770992	0.0	0.225806
5	0.016611	0.562500	1.0	0.0166113	0.0166113	0.0	0.0	0.816794	0.0	0.129032
6	0.019934	0.687500	0.0	0.0199336	0.0199336	0.0	1.0	0.679389	0.0	0.580645
7	0.023256	0.583333	0.0	0.0232558	0.0232558	0.0	0.0	0.702290	1.0	0.096774
8	0.026578	0.708333	1.0	0.0265781	0.0265781	0.0	1.0	0.580153	0.0	0.225806
9	0.029900	0.500000	1.0	0.0299003	0.0299003	1.0	1.0	0.641221	1.0	0.500000

10 rows × 21 columns

In [ ]:

```
df_norm = df_norm.dropna()
```

In [ ]:

```
X = df_norm.drop(['AHD_Yes', 'Unnamed: 0'], axis=1)
Y = df_norm.AHD_Yes
```

In [ ]:

```
X.isnull().sum()
```

Out[34]:

```
Age                      0
Sex                      0
RestBP                   0
Chol                     0
Fbs                      0
RestECG                  0
MaxHR                    0
ExAng                    0
Oldpeak                  0
Slope                    0
Ca                       0
ChestPain_asymptomatic   0
ChestPain_nonanginal     0
ChestPain_nontypical     0
ChestPain_typical        0
Thal_fixed               0
Thal_normal              0
Thal_reversable          0
AHD_No                   0
dtype: int64
```

In [ ]:

```
Y.isnull().sum()
```

Out[35]:

```
0
```

In [ ]:

```
df_norm.isnull().sum()
```

Out[36]:

```
Unnamed: 0             0
Age                    0
Sex                    0
RestBP                 0
Chol                   0
Fbs                    0
RestECG                0
MaxHR                  0
ExAng                  0
Oldpeak                0
Slope                  0
Ca                     0
ChestPain_asymptomatic    0
ChestPain_nonanginal   0
ChestPain_nontypical   0
ChestPain_typical      0
Thal_fixed             0
Thal_normal            0
Thal_reversable        0
AHD_No                 0
AHD_Yes                0
dtype: int64
```

model training

In [ ]:

```
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 ,random_state=6)
```

In [ ]:

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, metric="minkowski", p=1)
classifier.fit(X_train, y_train)
```

Out[162]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=1,
                     weights='uniform')
```

In [ ]:

```
y_pred = classifier.predict(X_test)
y_pred
```

Out[163]:

```
array([0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 1, 0], dtype=uint8)
```

In [ ]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[164]:

```
array([[44,  0],
       [ 3, 23]], dtype=int64)
```

In [ ]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Out[165]:

```
0.9571428571428572
```

In [ ]:

In [ ]: