

## Assignment No 1

```
[1] import pandas as pd
```

```
import numpy as np
```

```
[3] df=pd.read_csv("Heart.csv")
```

```
[4] df.describe()
```

July 10, 2025

[4]:

Unnamed: 0

Age

Sex

count

303.000000 303.000000

mean

152.000000

54.438944

std

87.612784

9.038662

min

1.000000

29.000000

RestBP

303.000000 303.000000 303.000000

0.679868 131.689769 246.693069

0.467299 17.599748 51.776918

0.000000 94.000000 126.000000

Chol

Fbs

303.000000

0.148515

0.356198

0.000000

25%

76.500000

48.000000

0.000000 120.000000 211.000000

0.000000

50%

152.000000

56.000000

75%

max

227.500000 61.000000

303.000000 77.000000

1.000000 130.000000 241.000000

1.000000 140.000000 275.000000

1.000000 200.000000

0.000000

0.000000

564.000000

1.000000

count

RestECG

303.000000

MaxHR

mean

std

303.000000

0.990099 149.607261

0.994971 22.875003

ExAng

303.000000

Oldpeak

303.000000

Slope

Ca

303.000000

299.000000

0.326733

1.039604

1.600660

0.672241

0.469794

1.161075

0.616226

0.937438

min

0.000000 71.000000

0.000000

0.000000

1.000000

0.000000

25%

0.000000 133.500000

0.000000

0.000000

1.000000

0.000000

50%

1.000000 153.000000

0.000000

0.800000

2.000000

0.000000

75%

2.000000 166.000000

1.000000

1.600000

2.000000

1.000000

max

2.000000 202.000000

1.000000

6.200000

3.000000

3.000000

:

[5] df.shape

[5] (303, 15)

[6] df.isnull().sum()

[6] Unnamed: 0

Age

Sex

Chest Pain

0

0

O O O O

1

RestBP

Chol

Fbs

0

RestECG

MaxHR

0

ExAng

Oldpeak

Slope

Ca

Thal

AHD

0

0

O o O 420

dtype: int64

```
[8]: print("Total Missing Records are:", df.isnull().sum().sum())
```

Total Missing Records are: 6

```
[9] df.dtypes
```

:

:

```
[9] Unnamed: 0
```

int64

Age

Sex

int64

int64

Chest Pain

object

RestBP

int64

Chol

int64

Fbs

int64

RestECG

int64

MaxHR

int64

ExAng

int64

Oldpeak

float64

Slope

int64

Ca

float64

Thal

object

AHD

object

dtype: object

[10] (df==0). sum(axis=0)

[10] Unnamed: 0

:

Age

0

Sex

97

Chest Pain

RestBP

0

Chol

Fbs

258

RestECG

151

MaxHR

0

21

ExAng

204

Oldpeak

99

Slope

0

Ca

176

Thal

0

AHD

dtype: int64

"

```
[11] print("Total Zero's Are: (df==0).sum (axis=0).sum())
```

:

Total Zero's Are: 985

```
[12] mean_age=df['Age'].mean()
```

```
[13]: print("Mean of patients age is: ",mean_age)
```

Mean of patients age is:

54.43894389438944

:



```
[14] df.columns
```

```
[14]: Index(['Unnamed: 0', 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'Fbs',  
'RestECG', 'MaxHR', 'ExAng', 'Oldpeak', 'Slope', 'Ca', 'Thal', 'AHD'],  
dtype='object')
```

```
[15]: df1=df.filter(['Age', 'Sex', 'Chest Pain', 'RestBP', 'Chol'])
```

```
[16] df1.describe()
```

Sex

Chol

RestBP

```
303.000000 303.000000 303.000000  
0.679868 131.689769 246.693069  
0.467299 17.599748 51.776918  
0.000000 94.000000 126.000000  
0.000000 120.000000 211.000000  
1.000000 130.000000 241.000000  
1.000000 140.000000 275.000000  
1.000000 200.000000 564.000000
```

```
[16]:
```

Age

```
count 303.000000
```

```
mean
```

```
54.438944
```

```
std
```

```
9.038662
```

```
min
```

```
25%
```

```
29.000000
```

```
48.000000
```

```
50%
```

```
56.000000
```

```
75%
```

61.000000

max

77.000000

[17]:

df1

[17]:

Age Sex

Chest Pain RestBP

Chol

0

63

1

typical

145 233

1

67

1 asymptomatic

160 286

2

67

1 asymptomatic

120 229

3

37

1

4

41

nonanginal  
0 nontypical

130 250

130

204

:

3

298

45

1

typical

110 264

299

68

1 asymptomatic

144 193

300

57

1

asymptomatic

130

131

301 57

0 nontypical

130 236

302 38

1

nonanginal

138 175

[303 rows x 5 columns]

[18] mean=df['Ca'] .mean ()

:

[19] df ['Ca'].fillna (value=mean, inplace=True)

[23] mode=df['Thal'] .mode() .iloc[0]

:

[24] df ['Thal'].fillna(value=mode, inplace=True)

[25] df.isnull().sum()

:

[25] Unnamed: 0

0

Age

0

Sex

0

Chest Pain

0

RestBP

0

Chol

Fbs

RestECG

MaxHR

ExAng

Oldpeak

Slope

0

Ca

Thal

0

AHD

dtype: int64

```
[28] import matplotlib.pyplot as plt
```

```
!pip install seaborn
```

```
import seaborn as sns
```

Defaulting to user installation because normal site-packages is not writeable  
Collecting seaborn

Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)

294.9/294.9 KB 2.2 MB/s eta 0:00:00 [31m2.9 MB/s

eta 0:00:01

Requirement already satisfied: numpy!=1.24.0,>=1.20 in

/home/administrator/.local/lib/python3.10/site-packages (from seaborn) (1.25.1)

:

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in

/home/administrator/.local/lib/python3.10/site-packages (from seaborn) (3.7.2)

Requirement already satisfied: pandas>=1.2 in

/home/administrator/.local/lib/python3.10/site-packages (from seaborn) (2.0.3)

Requirement already satisfied: cyclopy>=0.10 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)

Requirement already satisfied: kiwisolver>=1.0.1 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)

Requirement already satisfied: packaging>=20.0 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
matplotlib!=3.6.1,>=3.4->seaborn) (23.0)

Requirement already satisfied: pyparsing<3.1,>=2.3.1 in /usr/lib/python3/dist-  
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.4.7)  
Requirement already satisfied: python-dateutil>=2.7 in  
/home/administrator/.local/lib/python3.10/site-packages (from

matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)

Requirement already satisfied: contourpy>=1.0.1 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
matplotlib!=3.6.1,>=3.4->seaborn) (1.1.0)

Requirement already satisfied: fonttools>=4.22.0 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
matplotlib!=3.6.1,>=3.4->seaborn) (4.41.0)

Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages  
(from matplotlib!=3.6.1,>=3.4->seaborn) (9.0.1)

Requirement already satisfied: tzdata>=2022.1 in  
/home/administrator/.local/lib/python3.10/site-packages (from  
pandas>=1.2->seaborn) (2023.3)

Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages  
(from pandas>=1.2->seaborn) (2022.1)

Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from  
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

Installing collected packages: seaborn

Successfully installed seaborn-0.13.2

```
[33] sns.set_style("whitegrid")  
  
sns.countplot(x='AHD',data=df, hue="AHD",palette='pastel')  
  
plt.xlabel("Acute Heart Disease")  
  
plt.ylabel("Total count")  
  
plt.title("Frequency of heart disease")  
  
plt.legend(['No', 'Yes'],loc='upper right')  
  
[33]: <matplotlib.legend.Legend at 0x734bb30a52a0>
```

5



160

140

120

Total count

100

80

60

60

40

40

20

20

Frequency of heart disease

0

No

Yes

Acute Heart Disease

No

Yes

[32] fig, ax

:

=

plt.subplots()

fig.set\_size\_inches (10, 6)

sns.histplot(x="Age", data=df, hue="AHD", palette="bright")  
sns.set\_style("whitegrid")

plt.title("Effect of age on frequency of heart disease")  
plt.xlabel("Age")

plt.ylabel("Frequency")

plt. legend (["Heart Disease", "No Heart Disease"])  
plt.show()

6

30

25

Frequency

20

15

10

10

5

Effect of age on frequency of heart disease

0



30

40

50

60

Age

Heart Disease

No Heart Disease

70

10

```
[34] df ['Chest Pain'] .unique()
```

:

```
[34] array(['typical', 'asymptomatic', 'nonanginal', 'nontypical'],  
dtype=object)
```

```
[35] fig, ax = plt.subplots()
```

```
fig.set_size_inches (10, 6)
```

```
sns. countplot(x="AHD", hue="ChestPain", data=df, palette="dark")  
plt.title("Type of Chest Pain")
```

```
plt.xlabel("Heart Disease")
```

```
plt.ylabel("Frequency")
```

```
plt.legend (["Typical Angina", "Asymptomatic", "Non-Anginal pain",  
"Atypical Angina"])
```

```
plt.show()
```

7

100

Frequency

80

00

40

40

20

Type of Chest Pain

0

No

Yes

Heart Disease

Typical Angina

Asymptomatic

Non-Anginal pain

Atypical Angina

[44]: X

=

```
df[['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol', 'RestECG', 'MaxHR']]
```

```
Y= df['AHD']
```

```
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)
```

[45] X\_train.info()

```
<class 'pandas.core.frame.DataFrame'>
```

Index: 227 entries, 101 to 35

Data columns (total 7 columns):

#

Column

Non-Null Count Dtype

0

Age

227 non-null

int64

1

Sex

227 non-null

int64

2

Chest Pain

227 non-null

object

3

RestBP

227 non-null

int64

4

Chol

227 non-null

int64

5

RestECG

227 non-null

int64

6 MaxHR

227 non-null

int64

dtypes: int64(6), object (1)

memory usage: 14.2+ KB

[46] Y\_train.info()

<class 'pandas.core.series.Series'>

Index: 227 entries, 101 to 35

Series name: AHD

Non-Null Count

227 non-null

Dtype

object

dtypes: object(1)

memory usage: 3.5+ KB

[47] X\_test.info()

<class 'pandas.core.frame.DataFrame'>

Index: 76 entries, 56 to 225

Data columns (total 7 columns):

#

Column

Non-Null Count Dtype

0

Age

76 non-null

int64

1

Sex

76 non-null

int64

2

Chest Pain

76 non-null

object

3

RestBP

76 non-null

int64

4

Chol

76 non-null

int64

5

RestECG

76 non-null

int64

6 MaxHR

76 non-null

int64

dtypes: int64(6), object (1)

memory usage: 4.8+ KB

[48] Y\_test.info()

:

<class 'pandas.core.series. Series'>

Index: 76 entries, 56 to 225

Series name: AHD

Non-Null Count

76 non-null

Dtype

object

dtypes: object (1)

memory usage: 1.2+ KB

```
[49] from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
recall_score, f1_score
```

=

```
y_true
[1] *45+ [1] *5 +
y_pred [1] *45+ [0] *5 +
```

cm =

=

```
[0] *55 + [0] *395
```

```
[1] *55 + [0] *395
```

```
confusion matrix(y_true, y_pred)
print("Confusion Matrix:\n", cm)
```

```
acc = accuracy_score (y_true, y_pred)
print("Accuracy:", acc)
```

```
[ ]:
```

=

```
prec precision_score (y_true, y_pred)
print("Precision:", prec)
```

rec =

```
recall_score (y_true, y_pred)
```

```
print("Recall:", rec)
```

```
f1 = f1_score(y_true, y_pred)  
print("F1 Score:", f1)
```

Confusion Matrix:

```
[[395 55]
```

```
 [ 545]]
```

Accuracy: 0.88

Precision: 0.45

Recall: 0.9

F1 Score: 0.6

10

10