In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("Heart.csv")
df
```

Out[2]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	299	45	1	typical	110	264	0	0	132	0	1.2	2	0.0
299	300	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2.0
300	301	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1.0
301	302	57	0	nontypical	130	236	0	2	174	0	0.0	2	1.0
302	303	38	1	nonanginal	138	175	0	0	173	0	0.0	1	NaN

303 rows × 15 columns

In [8]:
```
df.drop('Unnamed: 0',axis=1)
```

Out[8]:

	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca	Thal
0	63	1	typical	145	233	1	2	150	0	2.3	3	0.0	fixed
1	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0	normal
2	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0	reversable
3	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0	normal
4	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0	normal
...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	45	1	typical	110	264	0	0	132	0	1.2	2	0.0	reversable
299	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2.0	reversable
300	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1.0	reversable
301	57	0	nontypical	130	236	0	2	174	0	0.0	2	1.0	normal
302	38	1	nonanginal	138	175	0	0	173	0	0.0	1	NaN	normal

303 rows × 14 columns

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
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
```

In [11]:
```
med = df['Ca'].mean()
med
```

Out[11]:
```
0.6722408026755853
```

In [13]:
```
df['Ca'] = df['Ca'].fillna(med,inplace=False)
```

In [14]:
```
df
```

Out[14]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.00
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.00
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.00
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.00
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	299	45	1	typical	110	264	0	0	132	0	1.2	2	0.00
299	300	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2.00
300	301	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1.00
301	302	57	0	nontypical	130	236	0	2	174	0	0.0	2	1.00
302	303	38	1	nonanginal	138	175	0	0	173	0	0.0	1	0.67

303 rows × 15 columns

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
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          2
AHD           0
dtype: int64
```

In [17]:
```
df = df.dropna(subset=['Thal'])
df
```

Out[17]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.00
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.00
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.00
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.00
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	299	45	1	typical	110	264	0	0	132	0	1.2	2	0.00
299	300	68	1	asymptomatic	144	193	1	0	141	0	3.4	2	2.00
300	301	57	1	asymptomatic	130	131	0	0	115	1	1.2	2	1.00
301	302	57	0	nontypical	130	236	0	2	174	0	0.0	2	1.00
302	303	38	1	nonanginal	138	175	0	0	173	0	0.0	1	0.67

301 rows × 15 columns

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
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
```

In [19]:
```
from sklearn.linear_model import LogisticRegression
```

In [20]:
```
model = LogisticRegression()
```

In [21]:
```
df.head()
```

Out[21]:

	Unnamed: 0	Age	Sex	ChestPain	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng	Oldpeak	Slope	Ca
0	1	63	1	typical	145	233	1	2	150	0	2.3	3	0.0
1	2	67	1	asymptomatic	160	286	0	2	108	1	1.5	2	3.0
2	3	67	1	asymptomatic	120	229	0	2	129	1	2.6	2	2.0
3	4	37	1	nonanginal	130	250	0	0	187	0	3.5	3	0.0
4	5	41	0	nontypical	130	204	0	2	172	0	1.4	1	0.0

In [25]:
```
x = df[['Age']]
y = df[['Fbs']]
```

In [26]:
```
model.fit(x,y)
```

```
C:\Users\pd277\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\utils\validation.py:1107: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[26]:

▾ LogisticRegression

LogisticRegression()

In [27]:
```
pre=model.predict([[12]])
pre
```

```
C:\Users\pd277\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

Out[27]:
```
array([0], dtype=int64)
```

In [28]:
```
from sklearn.model_selection import train_test_split
```

In [29]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=42)
```

In [30]:
```
newmodel=LogisticRegression()
```

In [31]:
```
newmodel.fit(x_train,y_train)
```

```
C:\Users\pd277\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\utils\validation.py:1107: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[31]:

▾ LogisticRegression

LogisticRegression()

In [32]:
```
x_test
```

Out[32]:

	Age
178	43
291	55
229	66
199	59
60	51
...	...
235	54
298	45
283	35
287	58
183	59

61 rows × 1 columns

In [33]:
```
y_pred=newmodel.predict(x_test)
y_pred
```

Out[33]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

In [34]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[34]:
```
array([[57,  0],
       [ 4,  0]], dtype=int64)
```

In [ ]: