In [3]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("/home/placement/Downloads/Titanic Dataset.csv")
```

In [4]: df.head

Out[4]: <bound method NDFrame.head of   PassengerId Survived Pclass \
0   1   0   3
1   2   1   1
2   3   1   3
3   4   1   1
4   5   0   3
..   ..
886   887   0   2
887   888   1   1
888   889   0   3
889   890   1   1
890   891   0   3

                                                  Name   Sex   Age SibSp \
0   Braund, Mr. Owen Harris   male 22.0   1
1   Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0   1
2   Heikkinen, Miss. Laina female 26.0   0
3   Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0   1
4   Allen, Mr. William Henry   male 35.0   0
..   ..
886   Montvila, Rev. Juozas   male 27.0   0
887   Graham, Miss. Margaret Edith female 19.0   0
888   Johnston, Miss. Catherine Helen "Carrie" female   NaN   1
889   Behr, Mr. Karl Howell   male 26.0   0
890   Dooley, Mr. Patrick   male 32.0   0

     Parch   Ticket   Fare Cabin Embarked
0   0   A/5 21171   7.2500   NaN   S
1   0   PC 17599 71.2833   C85   C
2   0 STON/O2. 3101282   7.9250   NaN   S
3   0   113803 53.1000 C123   S
4   0   373450   8.0500   NaN   S
..   ..
886   0   211536 13.0000   NaN   S
887   0   112053 30.0000   B42   S
888   2   W./C. 6607 23.4500   NaN   S
889   0   111369 30.0000 C148   C
890   0   370376   7.7500   NaN   Q

```
[891 rows x 12 columns]>
```

In [7]: `df.columns`

Out[7]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
               'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
              dtype='object')

In [8]: `df.info`

Out[8]:
```
<bound method DataFrame.info of      PassengerId  Survived  Pclass  \
0                1         0       3
1                2         1       1
2                3         1       3
3                4         1       1
4                5         0       3
..             ...       ...     ...
886            887         0       2
887            888         1       1
888            889         0       3
889            890         1       1
890            891         0       3

                                                  Name     Sex   Age  SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0      1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                               Heikkinen, Miss. Laina  female  26.0      0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                             Allen, Mr. William Henry    male  35.0      0
..                                                 ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q
```

```
[891 rows x 12 columns]>
```

In [9]: `df.describe()`

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [10]: `df.head(10)`

Out[10]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

In [11]: `df.shape`

Out[11]: `(891, 12)`

In [13]:
```python
df.isna().sum()
```

Out[13]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [16]:
```python
df['Pclass'].unique()
```

Out[16]: `array([3, 1, 2])`

In [18]:
```python
df['Survived'].unique()
```

Out[18]: `array([0, 1])`

In [20]:
```python
df['SibSp'].unique()
```

Out[20]: `array([1, 0, 3, 4, 2, 5, 8])`

In [21]:
```python
df['Age'].unique()
```

Out[21]:
```
array([22.  , 38.  , 26.  , 35.  ,   nan, 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
        8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
       49.  , 29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  ,
       16.  , 25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  ,
       71.  , 37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 ,
       51.  , 55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  ,
       45.5 , 20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  ,
       60.  , 10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  ,
       70.  , 24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [47]:
```python
df1=df.drop(columns=['PassengerId','Ticket','Cabin','Fare','SibSp','Name'])
```

In [48]: `df1`

Out[48]:

	Survived	Pclass	Sex	Age	Parch	Embarked
0	0	3	male	22.0	0	S
1	1	1	female	38.0	0	C
2	1	3	female	26.0	0	S
3	1	1	female	35.0	0	S
4	0	3	male	35.0	0	S
...	...	...	...	...	...	...
886	0	2	male	27.0	0	S
887	1	1	female	19.0	0	S
888	0	3	female	NaN	2	S
889	1	1	male	26.0	0	C
890	0	3	male	32.0	0	Q

891 rows × 6 columns

In [49]: `df1.isna().sum()`

Out[49]:
```
Survived      0
Pclass        0
Sex           0
Age         177
Parch         0
Embarked      2
dtype: int64
```

In [50]: `df2=pd.get_dummies(df1)`

In [51]: `df2`

Out[51]:

	Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	22.0	0	0	1	0	0	1
1	1	1	38.0	0	1	0	1	0	0
2	1	3	26.0	0	1	0	0	0	1
3	1	1	35.0	0	1	0	0	0	1
4	0	3	35.0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	1	0	0	1
887	1	1	19.0	0	1	0	0	0	1
888	0	3	NaN	2	1	0	0	0	1
889	1	1	26.0	0	0	1	1	0	0
890	0	3	32.0	0	0	1	0	1	0

891 rows × 9 columns

In [52]: `df2=df2.fillna(df.median())`

```
/snap/jupyter/6/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance co
lumns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise
TypeError.  Select only valid columns before calling the reduction.
  """Entry point for launching an IPython kernel.
```

In [53]: df2

Out[53]:

	Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	22.0	0	0	1	0	0	1
1	1	1	38.0	0	1	0	1	0	0
2	1	3	26.0	0	1	0	0	0	1
3	1	1	35.0	0	1	0	0	0	1
4	0	3	35.0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	1	0	0	1
887	1	1	19.0	0	1	0	0	0	1
888	0	3	28.0	2	1	0	0	0	1
889	1	1	26.0	0	0	1	1	0	0
890	0	3	32.0	0	0	1	0	1	0

891 rows × 9 columns

In [54]: df2.isna().sum()

Out[54]:
```
Survived      0
Pclass        0
Age           0
Parch         0
Sex_female    0
Sex_male      0
Embarked_C    0
Embarked_Q    0
Embarked_S    0
dtype: int64
```
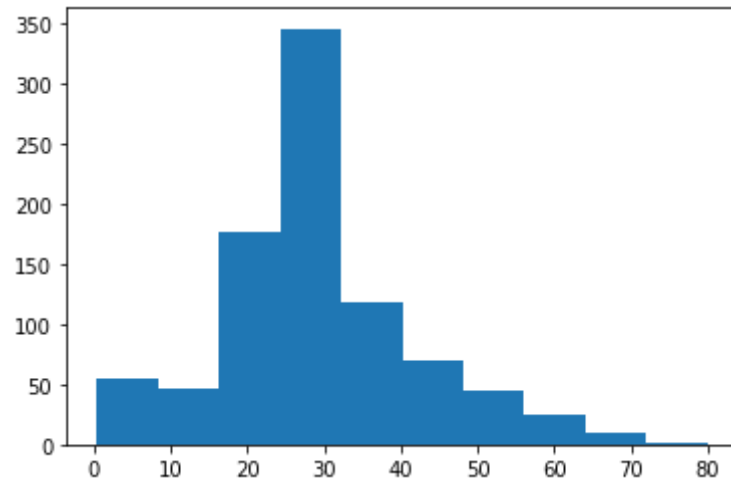
In [55]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(df2.Age)
```

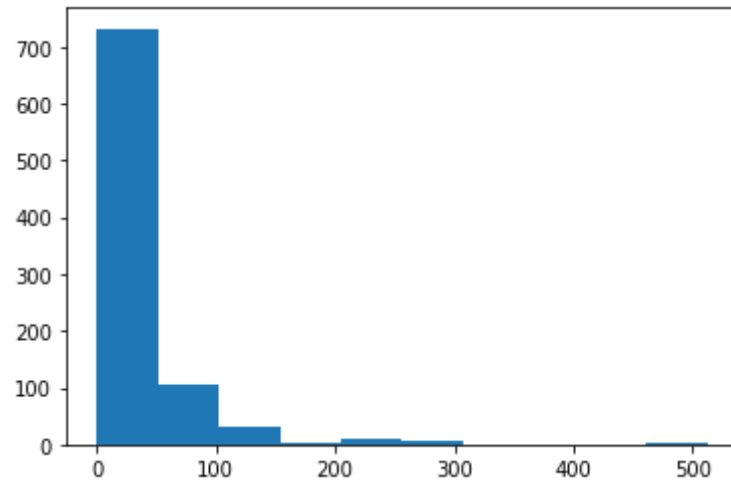Out[55]: <AxesSubplot:>
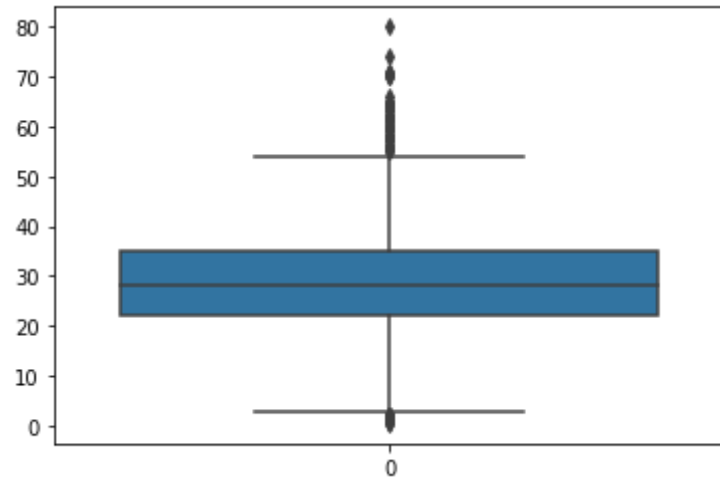
In [86]: `plt.hist(df2['Age'])`

Out[86]: (array([54., 46., 177., 346., 118., 70., 45., 24.,   9.,   2.]),
array([0.42 , 8.378, 16.336, 24.294, 32.252, 40.21 , 48.168, 56.126,
       64.084, 72.042, 80.   ]),
<BarContainer object of 10 artists>)

In [89]: `plt.hist(df['Fare'])`

Out[89]: (array([732., 106., 31.,   2., 11.,   6.,   0.,   0.,   0.,   3.]),
 array([0.   , 51.23292, 102.46584, 153.69876, 204.93168, 256.1646 ,
        307.39752, 358.63044, 409.86336, 461.09628, 512.3292]),
 <BarContainer object of 10 artists>)

In [88]: `df['Age'].unique()`

Out[88]:
```
array([22.   , 38.   , 26.   , 35.   ,   nan, 54.   ,  2.   , 27.   , 14.   ,
        4.   , 58.   , 20.   , 39.   , 55.   , 31.   , 34.   , 15.   , 28.   ,
        8.   , 19.   , 40.   , 66.   , 42.   , 21.   , 18.   ,  3.   ,  7.   ,
       49.   , 29.   , 65.   , 28.5  ,  5.   , 11.   , 45.   , 17.   , 32.   ,
       16.   , 25.   ,  0.83 , 30.   , 33.   , 23.   , 24.   , 46.   , 59.   ,
       71.   , 37.   , 47.   , 14.5  , 70.5  , 32.5  , 12.   ,  9.   , 36.5  ,
       51.   , 55.5  , 40.5  , 44.   ,  1.   , 61.   , 56.   , 50.   , 36.   ,
       45.5  , 20.5  , 62.   , 41.   , 52.   , 63.   , 23.5  ,  0.92 , 43.   ,
       60.   , 10.   , 64.   , 13.   , 48.   ,  0.75 , 53.   , 57.   , 80.   ,
       70.   , 24.5  ,  6.   ,  0.67 , 30.5  ,  0.42 , 34.5  , 74.   ])
```

In [63]: `df['Pclass']=df2['Pclassplt.hist(data2['Fare'])'].map({1:'First',2:'Second',3:'Third'})`

In [64]: `df2`

Out[64]:

	Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	22.0	0	0	1	0	0	1
1	1	1	38.0	0	1	0	1	0	0
2	1	3	26.0	0	1	0	0	0	1
3	1	1	35.0	0	1	0	0	0	1
4	0	3	35.0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	1	0	0	1
887	1	1	19.0	0	1	0	0	0	1
888	0	3	28.0	2	1	0	0	0	1
889	1	1	26.0	0	0	1	1	0	0
890	0	3	32.0	0	0	1	0	1	0

891 rows × 9 columns

In [66]:
```python
df2=pd.get_dummies(df2)
```

In [68]:
```python
df2
```

Out[68]:

	Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	22.0	0	0	1	0	0	1
1	1	1	38.0	0	1	0	1	0	0
2	1	3	26.0	0	1	0	0	0	1
3	1	1	35.0	0	1	0	0	0	1
4	0	3	35.0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...
886	0	2	27.0	0	0	1	0	0	1
887	1	1	19.0	0	1	0	0	0	1
888	0	3	28.0	2	1	0	0	0	1
889	1	1	26.0	0	0	1	1	0	0
890	0	3	32.0	0	0	1	0	1	0

891 rows × 9 columns

In [69]:
```python
df2['Age'].unique()
```

Out[69]:
```
array([22.  , 38.  , 26.  , 35.  , 28.  , 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  ,  8.  ,
       19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  , 49.  ,
       29.  , 65.  , 28.5 ,  5.  , 11.  , 45.  , 17.  , 32.  , 16.  ,
       25.  ,  0.83, 30.  , 33.  , 23.  , 24.  , 46.  , 59.  , 71.  ,
       37.  , 47.  , 14.5 , 70.5 , 32.5 , 12.  ,  9.  , 36.5 , 51.  ,
       55.5 , 40.5 , 44.  ,  1.  , 61.  , 56.  , 50.  , 36.  , 45.5 ,
       20.5 , 62.  , 41.  , 52.  , 63.  , 23.5 ,  0.92, 43.  , 60.  ,
       10.  , 64.  , 13.  , 48.  ,  0.75, 53.  , 57.  , 80.  , 70.  ,
       24.5 ,  6.  ,  0.67, 30.5 ,  0.42, 34.5 , 74.  ])
```

In [70]: `cor=df2.corr()`

In [71]: `cor`

Out[71]:

	Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	-0.338481	-0.064910	0.081629	0.543351	-0.543351	0.168240	0.003650	-0.155660
Pclass	-0.338481	1.000000	-0.339898	0.018443	-0.131900	0.131900	-0.243292	0.221009	0.081720
Age	-0.064910	-0.339898	1.000000	-0.172482	-0.081163	0.081163	0.030248	-0.031415	-0.014665
Parch	0.081629	0.018443	-0.172482	1.000000	0.245489	-0.245489	-0.011069	-0.081228	0.063036
Sex_female	0.543351	-0.131900	-0.081163	0.245489	1.000000	-1.000000	0.082853	0.074115	-0.125722
Sex_male	-0.543351	0.131900	0.081163	-0.245489	-1.000000	1.000000	-0.082853	-0.074115	0.125722
Embarked_C	0.168240	-0.243292	0.030248	-0.011069	0.082853	-0.082853	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.221009	-0.031415	-0.081228	0.074115	-0.074115	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	0.081720	-0.014665	0.063036	-0.125722	0.125722	-0.778359	-0.496624	1.000000

```
In [72]: sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[72]: <AxesSubplot:>
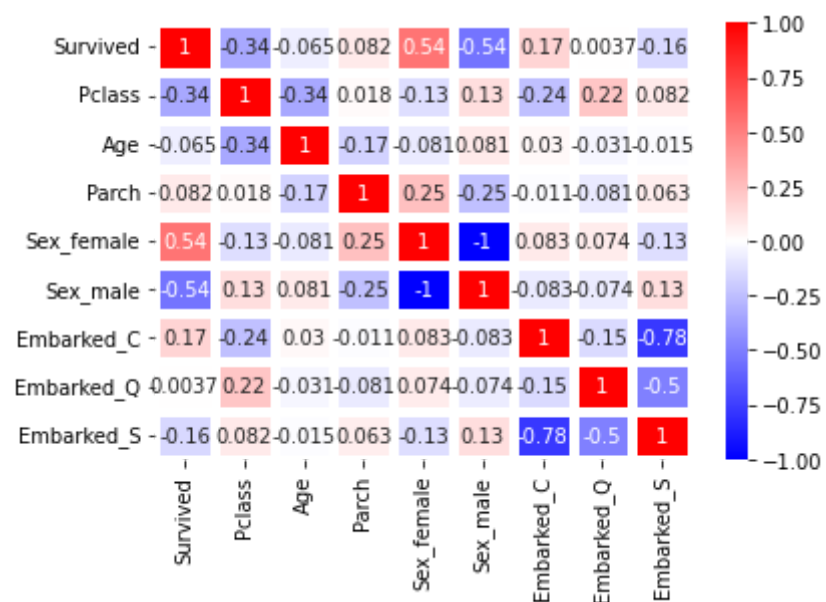
In [73]:
```python
df2.groupby(['Survived']).count()
```

Out[73]:

Survived	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	549	549	549	549	549	549	549	549
1	342	342	342	342	342	342	342	342

In [74]:
```python
y=df2['Survived']
x=df2.drop(columns='Survived')
```

In [75]:
```python
x
```

Out[75]:

	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	3	22.0	0	0	1	0	0	1
1	1	38.0	0	1	0	1	0	0
2	3	26.0	0	1	0	0	0	1
3	1	35.0	0	1	0	0	0	1
4	3	35.0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...
886	2	27.0	0	0	1	0	0	1
887	1	19.0	0	1	0	0	0	1
888	3	28.0	2	1	0	0	0	1
889	1	26.0	0	0	1	1	0	0
890	3	32.0	0	0	1	0	1	0

891 rows × 8 columns

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
```

In [80]: `x_train`

Out[80]:

	Pclass	Age	Parch	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
6	1	54.0	0	0	1	0	0	1
718	3	28.0	0	0	1	0	1	0
685	2	25.0	2	0	1	1	0	0
73	3	26.0	0	0	1	1	0	0
882	3	22.0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...
106	3	21.0	0	1	0	0	0	1
270	1	28.0	0	0	1	0	0	1
860	3	41.0	0	0	1	0	0	1
435	1	14.0	2	1	0	0	0	1
102	1	21.0	1	0	1	0	0	1

596 rows × 8 columns

In [81]:
```python
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[81]: LogisticRegression()

In [82]: `y_pred=classifier.predict(x_test)`

In [83]: `y_pred`

Out[83]:
```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0,
       0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0])
```

In [ ]: