In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
sai=pd.read_csv("/home/placement/Downloads/Titanic Dataset.csv")
```

In [3]:
```
sai.describe()
```

Out[3]:

	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 [4]:
```
list(sai)
```

Out[4]:
```
['PassengerId',
 'Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

In [5]:
```
sai.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [6]:
```
sai.isna().sum()
```

Out[6]:
```
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 [7]:
```
sai['Pclass'].unique()
```

Out[7]:
```
array([3, 1, 2])
```

In [8]:

```
sai['Survived'].unique()
```

Out[8]:

```
array([0, 1])
```

In [9]:

```
sai['SibSp'].unique()
```

Out[9]:

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

In [10]:

```
sai['Parch'].unique()
```

Out[10]:

```
array([0, 1, 2, 5, 3, 4, 6])
```

In [11]:

```
#sai[sai['Cabin'].isnull()]
```

In [12]:

```
sai['Age'].unique()
```

Out[12]:

```
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 [13]: sai.head(10)
```

Out[13]:

	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 [14]:

```
o=sai.drop(['PassengerId','Name','SibSp','Cabin','Ticket','Parch'],axis=1)
o
```

Out[14]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	male	22.0	7.2500	S
1	1	1	female	38.0	71.2833	C
2	1	3	female	26.0	7.9250	S
3	1	1	female	35.0	53.1000	S
4	0	3	male	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	male	27.0	13.0000	S
887	1	1	female	19.0	30.0000	S
888	0	3	female	NaN	23.4500	S
889	1	1	male	26.0	30.0000	C
890	0	3	male	32.0	7.7500	Q

891 rows × 6 columns

In [15]:

```
o['Sex']=o['Sex'].map({'male':1,'female':2})
o
```

Out[15]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	S
887	1	1	2	19.0	30.0000	S
888	0	3	2	NaN	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

In [16]:

```
o[o['Age'].isnull()]
```

Out[16]:

	Survived	Pclass	Sex	Age	Fare	Embarked
5	0	3	1	NaN	8.4583	Q
17	1	2	1	NaN	13.0000	S
19	1	3	2	NaN	7.2250	C
26	0	3	1	NaN	7.2250	C
28	1	3	2	NaN	7.8792	Q
...	...	...	...	...	...	...
859	0	3	1	NaN	7.2292	C
863	0	3	2	NaN	69.5500	S
868	0	3	1	NaN	9.5000	S
878	0	3	1	NaN	7.8958	S
888	0	3	2	NaN	23.4500	S

177 rows × 6 columns

In [17]:

```

#k=o['Age'].fillna(o['Age'].mean(), inplace=True)
#k
```

In [18]:

```
k=o.fillna(29)
k
```

Out[18]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	3	1	22.0	7.2500	S
1	1	1	2	38.0	71.2833	C
2	1	3	2	26.0	7.9250	S
3	1	1	2	35.0	53.1000	S
4	0	3	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	2	1	27.0	13.0000	S
887	1	1	2	19.0	30.0000	S
888	0	3	2	29.0	23.4500	S
889	1	1	1	26.0	30.0000	C
890	0	3	1	32.0	7.7500	Q

891 rows × 6 columns

In [19]:

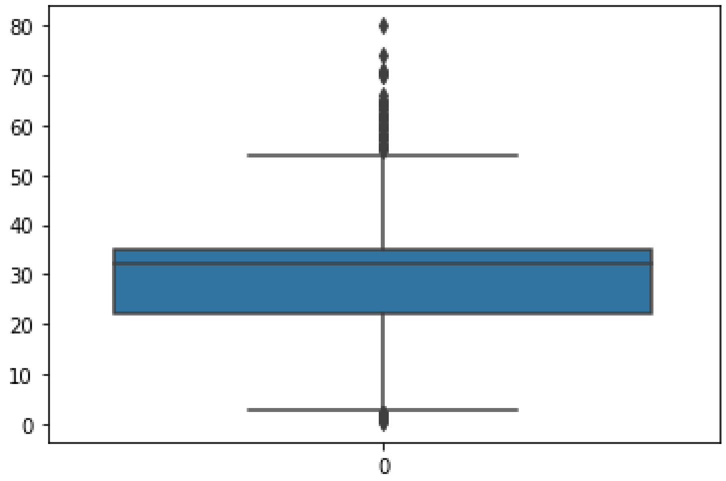
```
p=o.fillna(35,inplace=True)
p
```

In [20]:

```
import seaborn as sns
import matplotlib.pyplot as mp
sns.boxplot(o.Age)
```

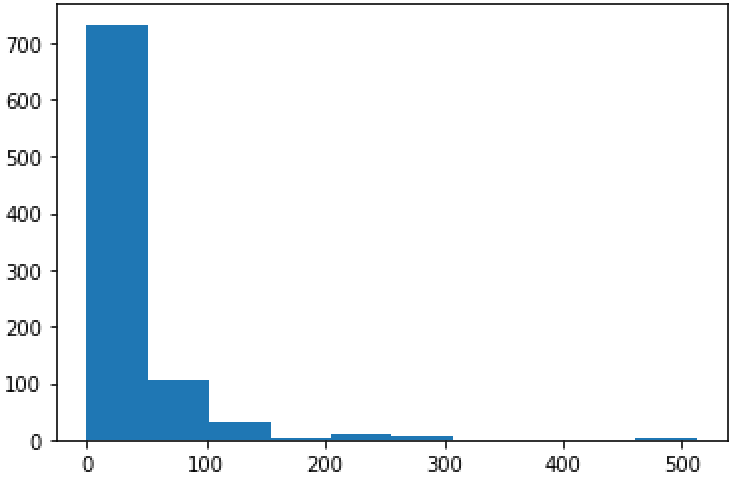
Out[20]:

```
<Axes: >
```



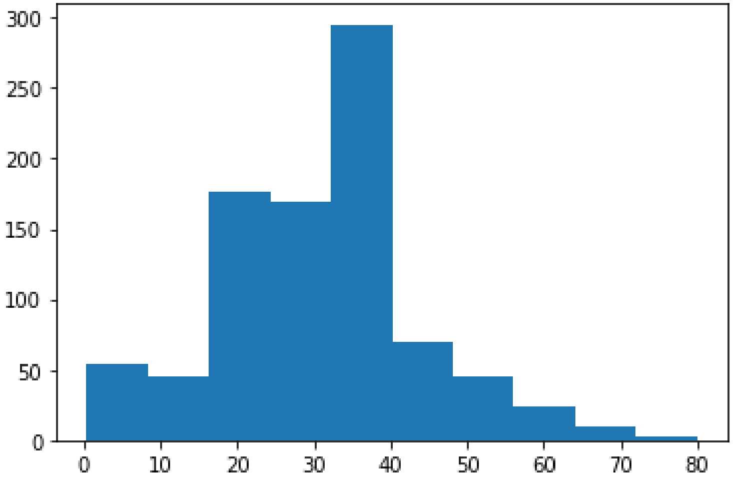
```
In [21]: mp.hist(o['Fare'])
```

```
Out[21]: (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>)
```



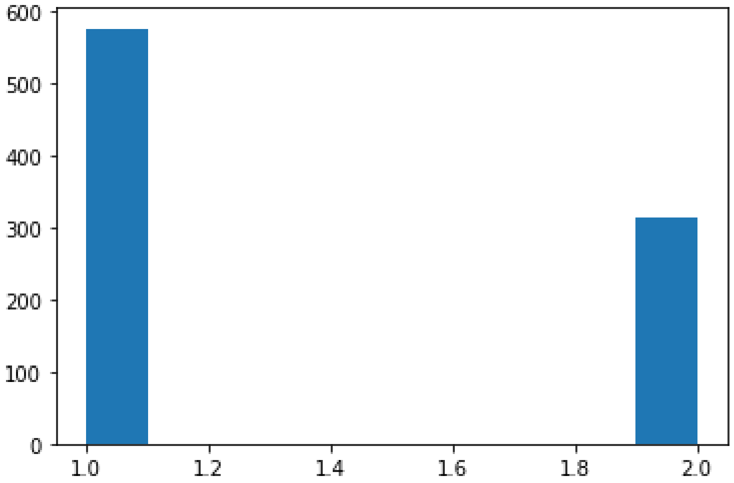
```
In [22]: mp.hist(o['Age'])
```

```
Out[22]: (array([ 54.,  46., 177., 169., 295.,  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 [23]: mp.hist(o['Sex'])
```

```
Out[23]: (array([577.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0., 314.]),
          array([1. , 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ]),
          <BarContainer object of 10 artists>)
```



```
In [24]: k.isna().sum()
```

```
Out[24]: Survived    0
         Pclass      0
         Sex         0
         Age         0
         Fare        0
         Embarked    0
         dtype: int64
```

is null

In [25]:
```
k.describe()
```

Out[25]:

	Survived	Pclass	Sex	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	1.352413	29.560236	32.204208
std	0.486592	0.836071	0.477990	13.005010	49.693429
min	0.000000	1.000000	1.000000	0.420000	0.000000
25%	0.000000	2.000000	1.000000	22.000000	7.910400
50%	0.000000	3.000000	1.000000	29.000000	14.454200
75%	1.000000	3.000000	2.000000	35.000000	31.000000
max	1.000000	3.000000	2.000000	80.000000	512.329200

In [26]:
```
k['Age'].unique()
```

Out[26]:
```
array([22.  , 38.  , 26.  , 35.  , 29.  , 54.  ,  2.  , 27.  , 14.  ,
        4.  , 58.  , 20.  , 39.  , 55.  , 31.  , 34.  , 15.  , 28.  ,
        8.  , 19.  , 40.  , 66.  , 42.  , 21.  , 18.  ,  3.  ,  7.  ,
       49.  , 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 [27]:
```
list(k)
```

Out[27]:
```
['Survived', 'Pclass', 'Sex', 'Age', 'Fare', 'Embarked']
```

In [28]:

```
k.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  891 non-null    int64  
 1   Pclass    891 non-null    int64  
 2   Sex       891 non-null    int64  
 3   Age       891 non-null    float64
 4   Fare      891 non-null    float64
 5   Embarked  891 non-null    object 
dtypes: float64(2), int64(3), object(1)
memory usage: 41.9+ KB
```

In [29]:

```
k.groupby(['Age']).count()
```

Out[29]:

	Survived	**Pclass**	**Sex**	**Fare**	**Embarked**
Age					
0.42	1	1	1	1	1
0.67	1	1	1	1	1
0.75	2	2	2	2	2
0.83	2	2	2	2	2
0.92	1	1	1	1	1
...	...	...	...	...	...
70.00	2	2	2	2	2
70.50	1	1	1	1	1
71.00	2	2	2	2	2
74.00	1	1	1	1	1
80.00	1	1	1	1	1

88 rows × 5 columns

```
In [30]: k["Age"]
```

```
Out[30]: 0      22.0
         1      38.0
         2      26.0
         3      35.0
         4      35.0
                ...
         886    27.0
         887    19.0
         888    29.0
         889    26.0
         890    32.0
         Name: Age, Length: 891, dtype: float64
```

```
In [31]: o['Pclass']=o['Pclass'].map({1:'F',2:'S',3:'T'})
         o
```

Out[31]:

	Survived	Pclass	Sex	Age	Fare	Embarked
0	0	T	1	22.0	7.2500	S
1	1	F	2	38.0	71.2833	C
2	1	T	2	26.0	7.9250	S
3	1	F	2	35.0	53.1000	S
4	0	T	1	35.0	8.0500	S
...	...	...	...	...	...	...
886	0	S	1	27.0	13.0000	S
887	1	F	2	19.0	30.0000	S
888	0	T	2	35.0	23.4500	S
889	1	F	1	26.0	30.0000	C
890	0	T	1	32.0	7.7500	Q

891 rows × 6 columns

In [32]:

```
o=pd.get_dummies(o,dtype=int)
o
```

Out[32]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_T	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	0	1	22.0	7.2500	0	0	1	0	0	0	1
1	1	2	38.0	71.2833	1	0	0	0	1	0	0
2	1	2	26.0	7.9250	0	0	1	0	0	0	1
3	1	2	35.0	53.1000	1	0	0	0	0	0	1
4	0	1	35.0	8.0500	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...
886	0	1	27.0	13.0000	0	1	0	0	0	0	1
887	1	2	19.0	30.0000	1	0	0	0	0	0	1
888	0	2	35.0	23.4500	0	0	1	0	0	0	1
889	1	1	26.0	30.0000	1	0	0	0	1	0	0
890	0	1	32.0	7.7500	0	0	1	0	0	1	0

891 rows × 11 columns

In [33]:

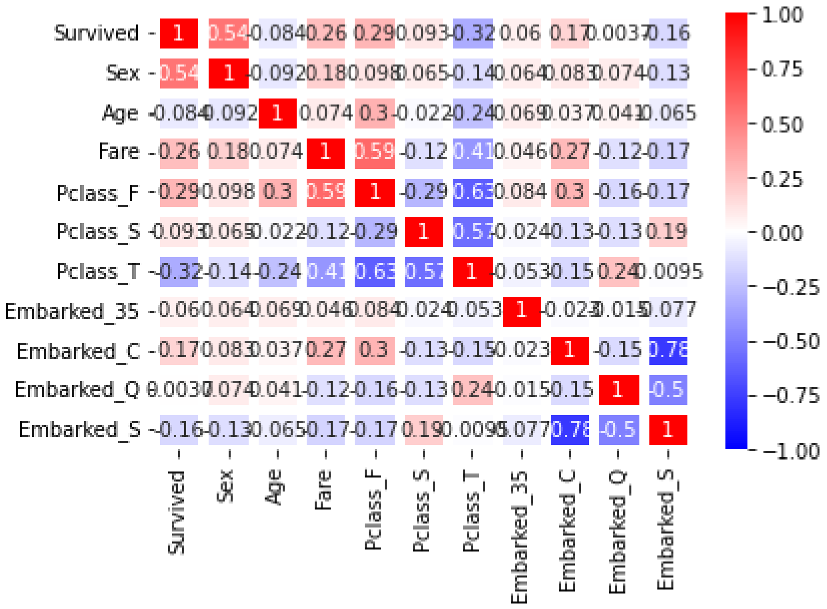
```
cor=o.corr()
cor
```

Out[33]:

	Survived	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_T	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
Survived	1.000000	0.543351	-0.083713	0.257307	0.285904	0.093349	-0.322308	0.060095	0.168240	0.003650	-0.155660
Sex	0.543351	1.000000	-0.091930	0.182333	0.098013	0.064746	-0.137143	0.064296	0.082853	0.074115	-0.125722
Age	-0.083713	-0.091930	1.000000	0.074199	0.302149	-0.022021	-0.242412	0.069343	0.036953	0.040528	-0.065062
Fare	0.257307	0.182333	0.074199	1.000000	0.591711	-0.118557	-0.413333	0.045646	0.269335	-0.117216	-0.166603
Pclass_F	0.285904	0.098013	0.302149	0.591711	1.000000	-0.288585	-0.626738	0.083847	0.296423	-0.155342	-0.170379
Pclass_S	0.093349	0.064746	-0.022021	-0.118557	-0.288585	1.000000	-0.565210	-0.024197	-0.125416	-0.127301	0.192061
Pclass_T	-0.322308	-0.137143	-0.242412	-0.413333	-0.626738	-0.565210	1.000000	-0.052550	-0.153329	0.237449	-0.009511
Embarked_35	0.060095	0.064296	0.069343	0.045646	0.083847	-0.024197	-0.052550	1.000000	-0.022864	-0.014588	-0.076588
Embarked_C	0.168240	0.082853	0.036953	0.269335	0.296423	-0.125416	-0.153329	-0.022864	1.000000	-0.148258	-0.778359
Embarked_Q	0.003650	0.074115	0.040528	-0.117216	-0.155342	-0.127301	0.237449	-0.014588	-0.148258	1.000000	-0.496624
Embarked_S	-0.155660	-0.125722	-0.065062	-0.166603	-0.170379	0.192061	-0.009511	-0.076588	-0.778359	-0.496624	1.000000

In [34]:
```
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[34]:
```
<Axes: >
```



In [35]:
```
sai.groupby('Survived').count()
```

Out[35]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Survived											
0	549	549	549	549	424	549	549	549	549	68	549
1	342	342	342	342	290	342	342	342	342	136	340

In [40]:

```
y=o['Survived']
x=o.drop('Survived',axis=1)
x
```

Out[40]:

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_T	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
0	1	22.0	7.2500	0	0	1	0	0	0	1
1	2	38.0	71.2833	1	0	0	0	1	0	0
2	2	26.0	7.9250	0	0	1	0	0	0	1
3	2	35.0	53.1000	1	0	0	0	0	0	1
4	1	35.0	8.0500	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...
886	1	27.0	13.0000	0	1	0	0	0	0	1
887	2	19.0	30.0000	1	0	0	0	0	0	1
888	2	35.0	23.4500	0	0	1	0	0	0	1
889	1	26.0	30.0000	1	0	0	0	1	0	0
890	1	32.0	7.7500	0	0	1	0	0	1	0

In [43]:

```
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 [44]:
```
x_test.head(10)
```

Out[44]:

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_T	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
709	1	35.0	15.2458	0	0	1	0	1	0	0
439	1	31.0	10.5000	0	1	0	0	0	0	1
840	1	20.0	7.9250	0	0	1	0	0	0	1
720	2	6.0	33.0000	0	1	0	0	0	0	1
39	2	14.0	11.2417	0	0	1	0	1	0	0
290	2	26.0	78.8500	1	0	0	0	0	0	1
300	2	35.0	7.7500	0	0	1	0	0	1	0
333	1	16.0	18.0000	0	0	1	0	0	0	1
208	2	16.0	7.7500	0	0	1	0	0	1	0
136	2	19.0	26.2833	1	0	0	0	0	0	1

In [45]:
```
x_train.head()
```

Out[45]:

	Sex	Age	Fare	Pclass_F	Pclass_S	Pclass_T	Embarked_35	Embarked_C	Embarked_Q	Embarked_S
6	1	54.0	51.8625	1	0	0	0	0	0	1
718	1	35.0	15.5000	0	0	1	0	0	1	0
685	1	25.0	41.5792	0	1	0	0	1	0	0
73	1	26.0	14.4542	0	0	1	0	1	0	0
882	2	22.0	10.5167	0	0	1	0	0	0	1

logistic regression

In [46]:

```
import warnings
warnings.filterwarnings('ignore')
from sklearn.linear_model import LogisticRegression
classifier=LogisticRegression()
classifier.fit(x_train,y_train)
```

Out[46]:

```
▾ LogisticRegression

LogisticRegression()
```

In [47]:

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

Out[47]:

```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 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, 0,
       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, 1, 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, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 0])
```

confusion matrix

In [48]:

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

Out[48]:

```
array([[153,  22],
       [ 34,  86]])
```

accuracy score

```
In [49]: from sklearn.metrics import accuracy_score
         accuracy_score(y_test,y_pred)

Out[49]: 0.8101694915254237
```

r2_score

```
In [50]: from sklearn.metrics import r2_score
         r2_score(y_test,y_pred)

Out[50]: 0.21333333333333337
```

```
In [ ]:
```