Data Preprocessing.

```
o   Import the Libraries.
o   Importing the dataset.
o   Checking for Null Values.
o   Data Visualization.
o   Outlier Detection
o   Splitting Dependent and Independent variables
o-  Encoding
o   Feature Scaling.
o   Splitting Data into Train and Test.
```

1.Import the Libraries

In [2]:

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

2.Importing the dataset

In [3]:

```
df=pd.read_csv("Titanic-Dataset.csv")
df
```

Out[3]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [4]:

```
df.shape
```

Out[4]:

```
(891, 12)
```

In [5]:

```
df.describe()
```

Out[5]:

	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 [6]:

```
df.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 [7]:

```
df.corr()
```

Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

In [8]:

```
df.corr().Fare.sort_values(ascending=False)
```

Out[8]:

```
Fare           1.000000
Survived       0.257307
Parch          0.216225
SibSp          0.159651
Age            0.096067
PassengerId    0.012658
Pclass        -0.549500
Name: Fare, dtype: float64
```

3.Checking for Null Values

In [9]:

```
df.isnull().any()
```

Out[9]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
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 [22]:

```
df["Age"].mode()
```

Out[22]:

```
0    24.0
Name: Age, dtype: float64
```

In [23]:

```
df["Age"]=df["Age"].fillna(df["Age"].mode())
```

In [24]:

```
df["Cabin"].mode()
```

Out[24]:

```
0    B96 B98
1         G6
Name: Cabin, dtype: object
```

In [25]:

```
df["Cabin"]=df["Cabin"].fillna(df["Cabin"].mode())
```

In [26]:

```
df["Embarked"].mode()
```

Out[26]:

```
0    S
Name: Embarked, dtype: object
```

In [27]:

```
df["Embarked"]=df["Embarked"].fillna(df["Embarked"].mode())
```

In [28]:

```
df.isnull().any()
```

Out[28]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [29]:

```
df.isnull().sum()
```

Out[29]:

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          685
Embarked         2
dtype: int64
```

In [19]:

```
df.Embarked.nunique()
```

Out[19]:

```
3
```

In [20]:

```
df.Embarked.unique()
```

Out[20]:

```
array(['S', 'C', 'Q', nan], dtype=object)
```

In [21]:

```
df.Embarked.value_counts()
```

Out[21]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

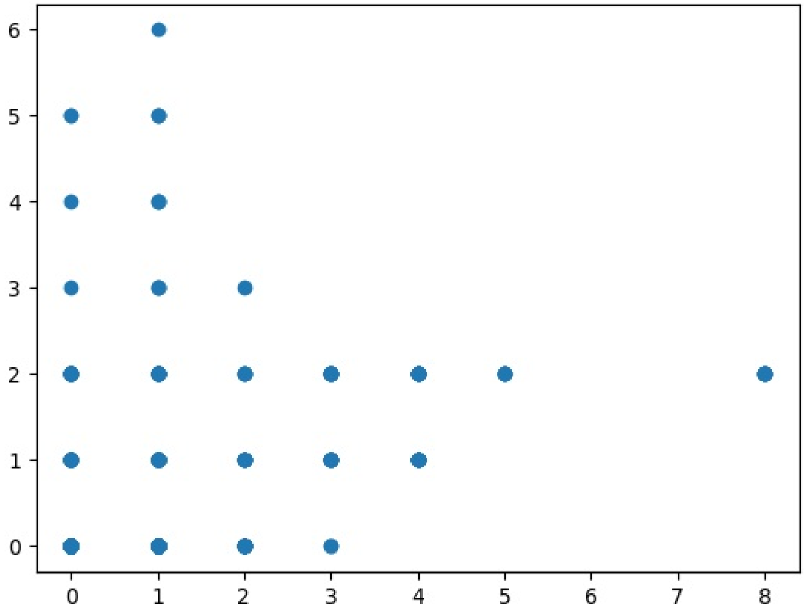
4.Data Visualization

In [30]:

```
plt.scatter(df["SibSp"],df["Parch"])
```

Out[30]:

```
<matplotlib.collections.PathCollection at 0x2a39156bcd0>
```

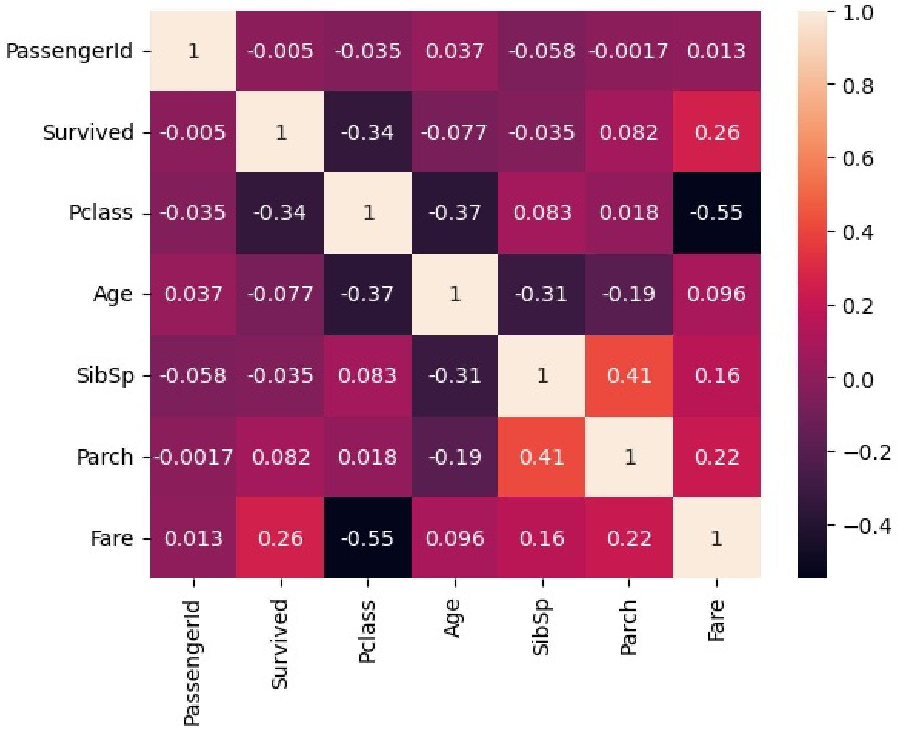


In [31]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[31]:

```
<Axes: >
```

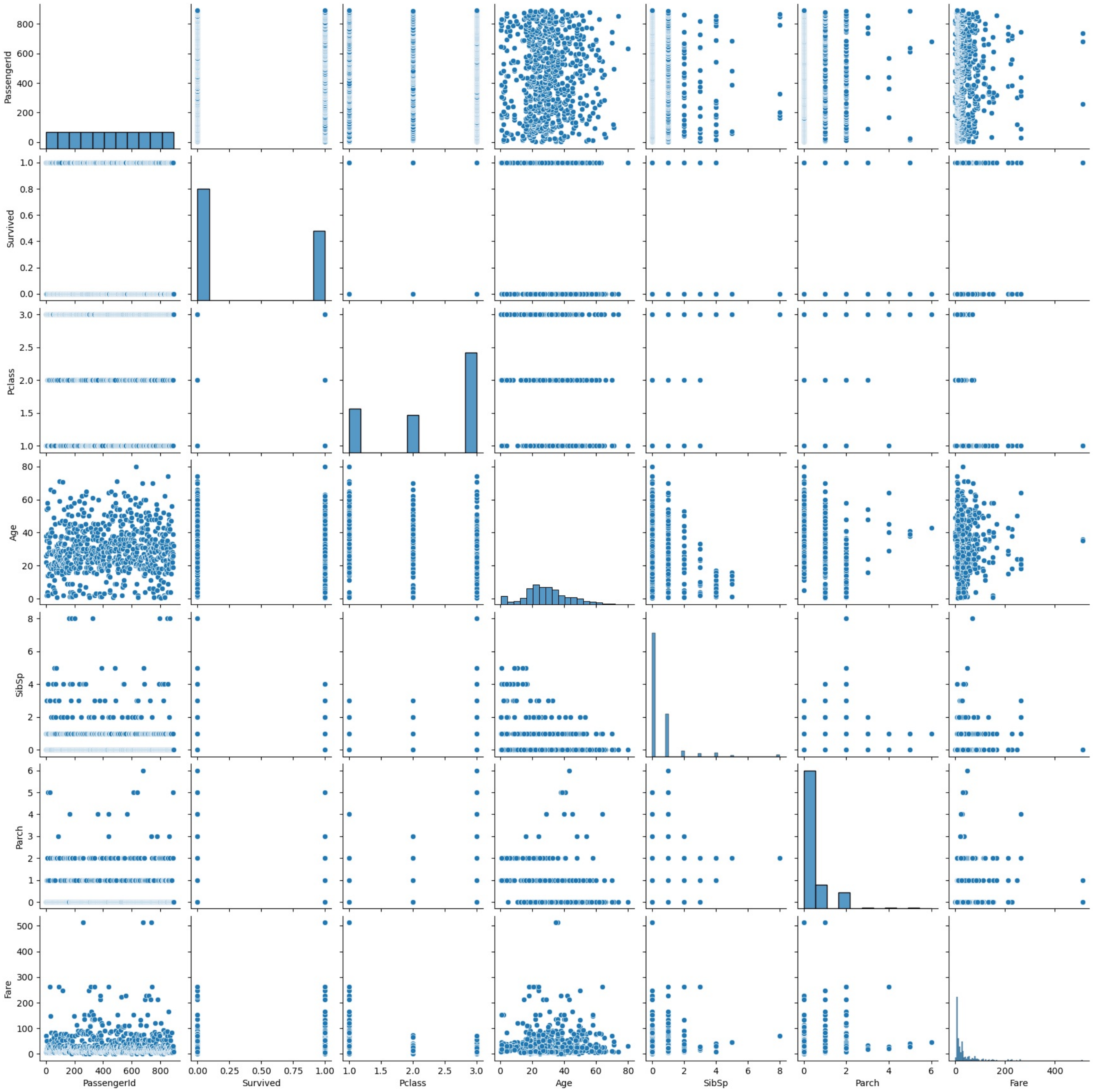


In [32]:

```
sns.pairplot(df)
```

Out[32]:

```
<seaborn.axisgrid.PairGrid at 0x2a391653cd0>
```

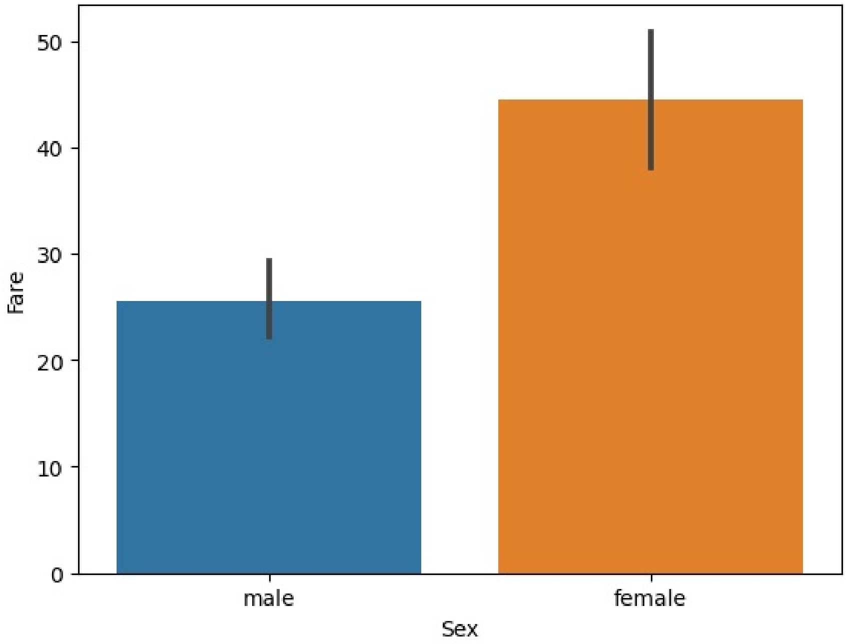


In [33]:

```
sns.barplot(x=df["Sex"],y=df["Fare"])
```

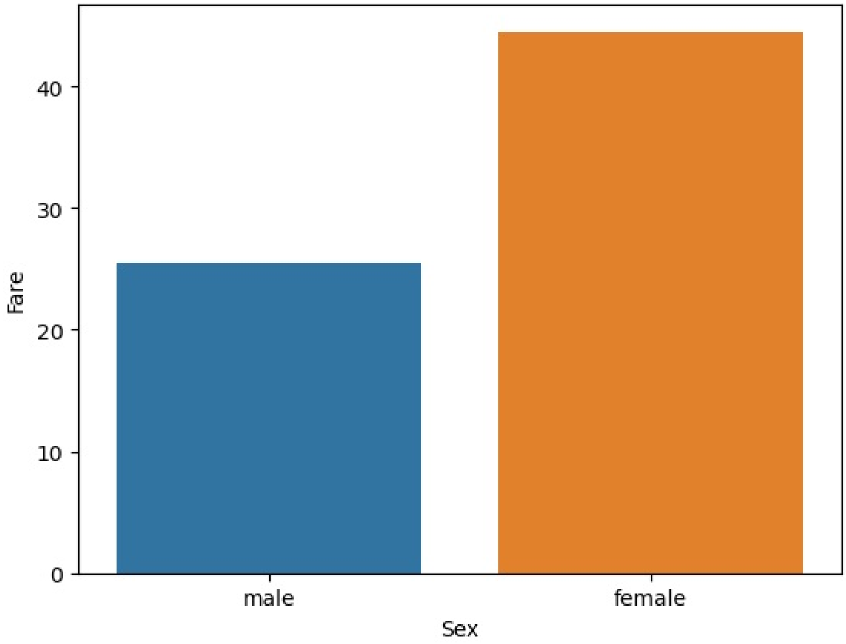
Out[33]:

```
<Axes: xlabel='Sex', ylabel='Fare'>
```



In [34]:
```
sns.barplot(x=df["Sex"],y=df["Fare"],ci=0)
```

Out[34]:
```
<Axes: xlabel='Sex', ylabel='Fare'>
```



5.Outlier Detection

In [35]:
```
df.head()
```

Out[35]:

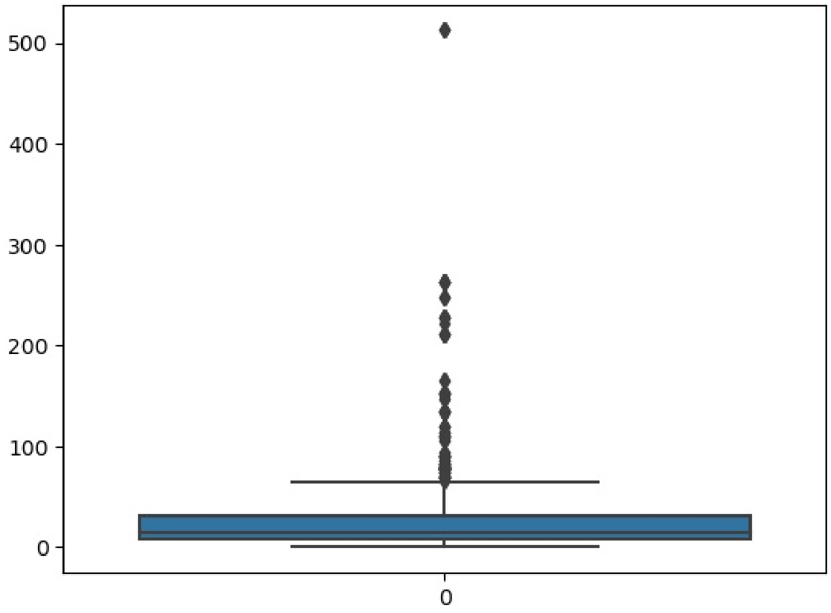
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	G6	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

In [36]:

```
sns.boxplot(df["Fare"])
```

Out[36]:

```
<Axes: >
```



In [37]:

```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
31.0
```

In [38]:

```
IQR=q3-q1
IQR
```

Out[38]:

```
23.0896
```

In [39]:

```
upper_limit=q3+1.5*IQR
upper_limit
```

Out[39]:

```
65.6344
```

In [40]:

```
lower_limit=q1-1.5*IQR
lower_limit
```

Out[40]:

```
-26.724
```

In [41]:

```
df.median()
```

Out[41]:

```
PassengerId    446.0000
Survived         0.0000
Pclass           3.0000
Age             28.0000
SibSp            0.0000
Parch            0.0000
Fare            14.4542
dtype: float64
```

In [42]:

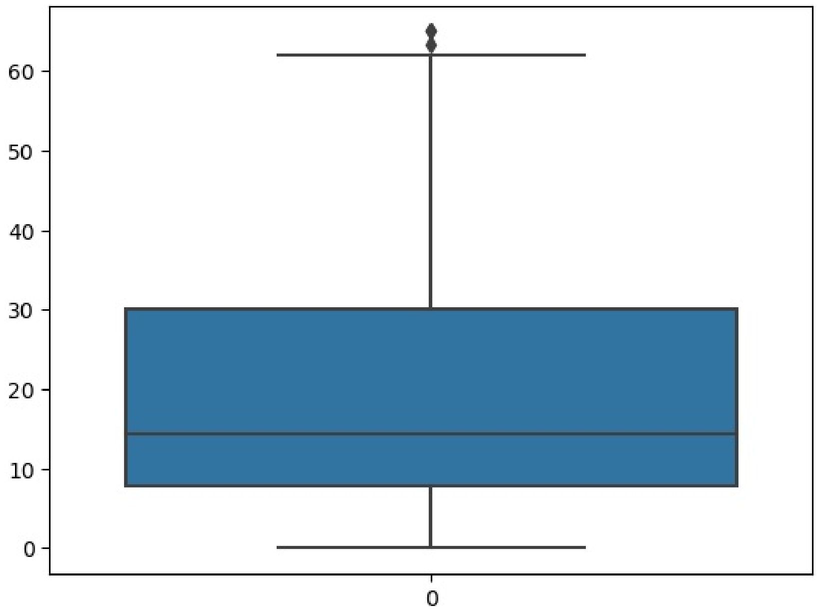
```
df['Fare']=np.where(df['Fare']>upper_limit,30,df['Fare'])
```

In [43]:

```
sns.boxplot(df["Fare"])
```

Out[43]:

```
<Axes: >
```



6.Splitting Dependent and Independent variables

In [44]:

```
df.head()
```

Out[44]:

	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	7.250	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	30.000	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	G6	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050	NaN	S

In [45]:

```
X=df.drop(columns=["Fare"],axis=1)
X.head()
```

Out[45]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	G6	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	NaN	S

In [46]:

```
X.shape
```

Out[46]:

```
(891, 11)
```

In [47]:

```
type(X)
```

Out[47]:

```
pandas.core.frame.DataFrame
```

In [48]:

```
y=df["Fare"]
y.head()
```

Out[48]:

```
0     7.250
1    30.000
2     7.925
3    53.100
4     8.050
Name: Fare, dtype: float64
```

In [49]:

```
type(y)
```

Out[49]:
```
pandas.core.series.Series
```

7.Encoding

In [51]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [52]:
```
X["Sex"]=le.fit_transform(X["Sex"])
```

In [53]:
```
X.head()
```

Out[53]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	G6	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	C123	S
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	NaN	S

In [54]:
```
print(le.classes_)
```

```
['female' 'male']
```

In [55]:
```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

Out[55]:
```
{'female': 0, 'male': 1}
```

8.Feature Scaling

In [56]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

In [ ]:

9.Splitting Data into Train and Test

In [60]:
```
from sklearn.model_selection import train_test_split
```

In [61]:
```
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

In [62]:
```
x_train
```

Out[62]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
140	141	0	3	Boulos, Mrs. Joseph (Sultana)	0	NaN	0	2	2678	NaN	C
439	440	0	2	Kvillner, Mr. Johan Henrik Johannesson	1	31.0	0	0	C.A. 18723	NaN	S
817	818	0	2	Mallet, Mr. Albert	1	31.0	1	1	S.C./PARIS 2079	NaN	C
378	379	0	3	Betros, Mr. Tannous	1	20.0	0	0	2648	NaN	C
491	492	0	3	Windelov, Mr. Einar	1	21.0	0	0	SOTON/OQ 3101317	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
835	836	1	1	Compton, Miss. Sara Rebecca	0	39.0	1	1	PC 17756	E49	C
192	193	1	3	Andersen-Jensen, Miss. Carla Christine Nielsine	0	19.0	1	0	350046	NaN	S
629	630	0	3	O'Connell, Mr. Patrick D	1	NaN	0	0	334912	NaN	Q
559	560	1	3	de Messemaeker, Mrs. Guillaume Joseph (Emma)	0	36.0	1	0	345572	NaN	S
684	685	0	2	Brown, Mr. Thomas William Solomon	1	60.0	1	1	29750	NaN	S

712 rows × 11 columns

In [63]:
```
x_test
```

Out[63]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
495	496	0	3	Yousseff, Mr. Gerious	1	NaN	0	0	2627	NaN	C
648	649	0	3	Willey, Mr. Edward	1	NaN	0	0	S.O./P.P. 751	NaN	S
278	279	0	3	Rice, Master. Eric	1	7.0	4	1	382652	NaN	Q
31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	0	NaN	1	0	PC 17569	B78	C
255	256	1	3	Touma, Mrs. Darwis (Hanne Youssef Razi)	0	29.0	0	2	2650	NaN	C
...	...	...	...	...	...	...	...	...	...	...	...
780	781	1	3	Ayoub, Miss. Banoura	0	13.0	0	0	2687	NaN	C
837	838	0	3	Sirota, Mr. Maurice	1	NaN	0	0	392092	NaN	S
215	216	1	1	Newell, Miss. Madeleine	0	31.0	1	0	35273	D36	C
833	834	0	3	Augustsson, Mr. Albert	1	23.0	0	0	347468	NaN	S
372	373	0	3	Beavan, Mr. William Thomas	1	19.0	0	0	323951	NaN	S

179 rows × 11 columns

In [64]:

```
y_train
```

Out[64]:

```
140    15.2458
439    10.5000
817    37.0042
378     4.0125
491     7.2500
        ...   
835    30.0000
192     7.8542
629     7.7333
559    17.4000
684    39.0000
Name: Fare, Length: 712, dtype: float64
```

In [65]:

```
y_test
```

Out[65]:

```
495    14.4583
648     7.5500
278    29.1250
31     30.0000
255    15.2458
        ...   
780     7.2292
837     8.0500
215    30.0000
833     7.8542
372     8.0500
Name: Fare, Length: 179, dtype: float64
```

In [67]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(712, 11)
(179, 11)
(712,)
(179,)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js