SHAIK MAHAMMAD IRFAN

21BCE9547

ASSIGNMENT-3

VIT-AP UNIVERSITY

1.IMPORT NECESSARY LIBRARIES ¶

In [1]:

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

```

2.IMPORT DATASET

In [2]:

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

In [3]:

```
df
```

0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 211
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 175
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O 31012
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	1138
				Allen, Mr.					

In [4]:

```
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8

In [5]:

```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fa
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.

3.CHECKING NULL VALUES

In [6]:

```
# Checking for null values
df.isnull().any()
```

Out[6]:

```
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 [7]:

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

Out[7]:

```
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 [8]:

```
df.corr()
```

```
C:\Users\SMD IRFAN\AppData\Local\Temp\ipykernel_11360\1134722465.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only
valid columns or specify the value of numeric_only to silence this war
ning.
  df.corr()
```

Out[8]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.0126
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.2573
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.5495
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.0960
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.1596
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.2162
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.0000

In [9]:

```python
df.describe()
```

Out[9]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.00
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.20
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.69
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.00
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.91
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.45
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.00
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.32

In [10]:

```python
df['Age'].fillna(df['Age'].mean(), inplace=True)
```

In [11]:

```python
df['Cabin'].fillna(df['Cabin'].mode()[0],inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)
```

In [12]:

```python
# Evaluating null values

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

Out[12]:

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

4.DATA VISUALIZATION

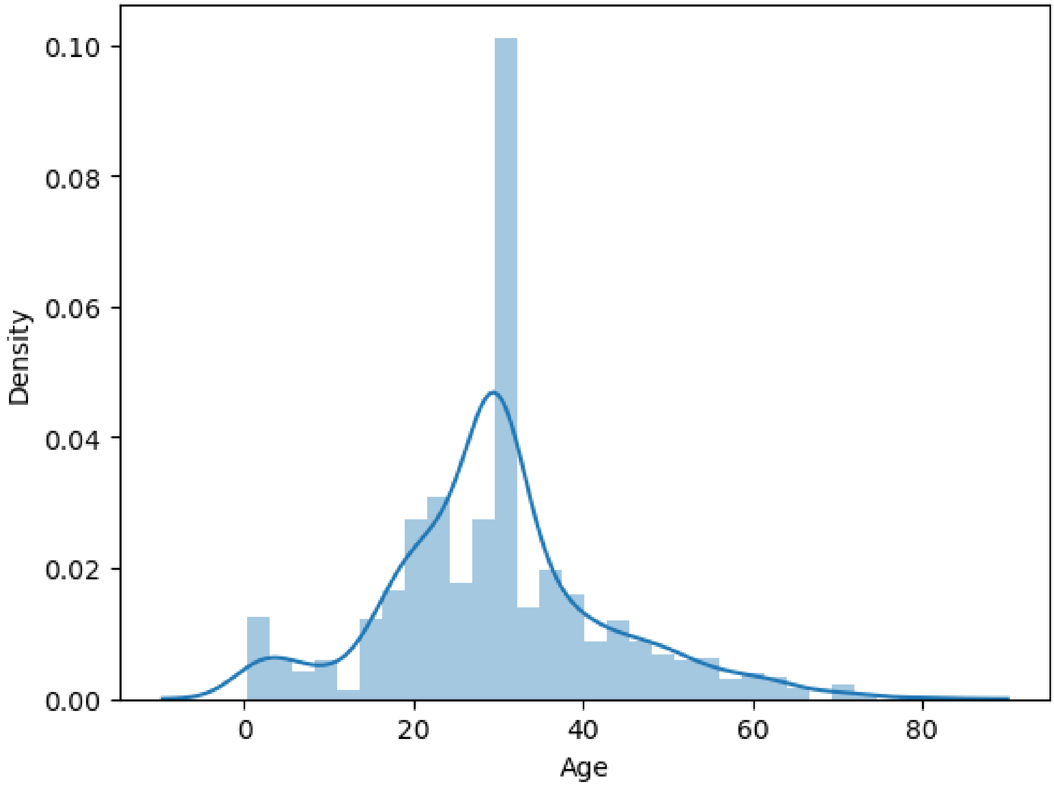
In [13]:

```
sns.distplot(df['Age'])
```

```
C:\Users\SMD IRFAN\AppData\Local\Temp\ipykernel_11360\3255828239.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.
14.0.

Please adapt your code to use either `displot` (a figure-level functio
n with
similar flexibility) or `histplot` (an axes-level function for histogr
ams).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['Age'])
```

Out[13]:

```
<Axes: xlabel='Age', ylabel='Density'>
```



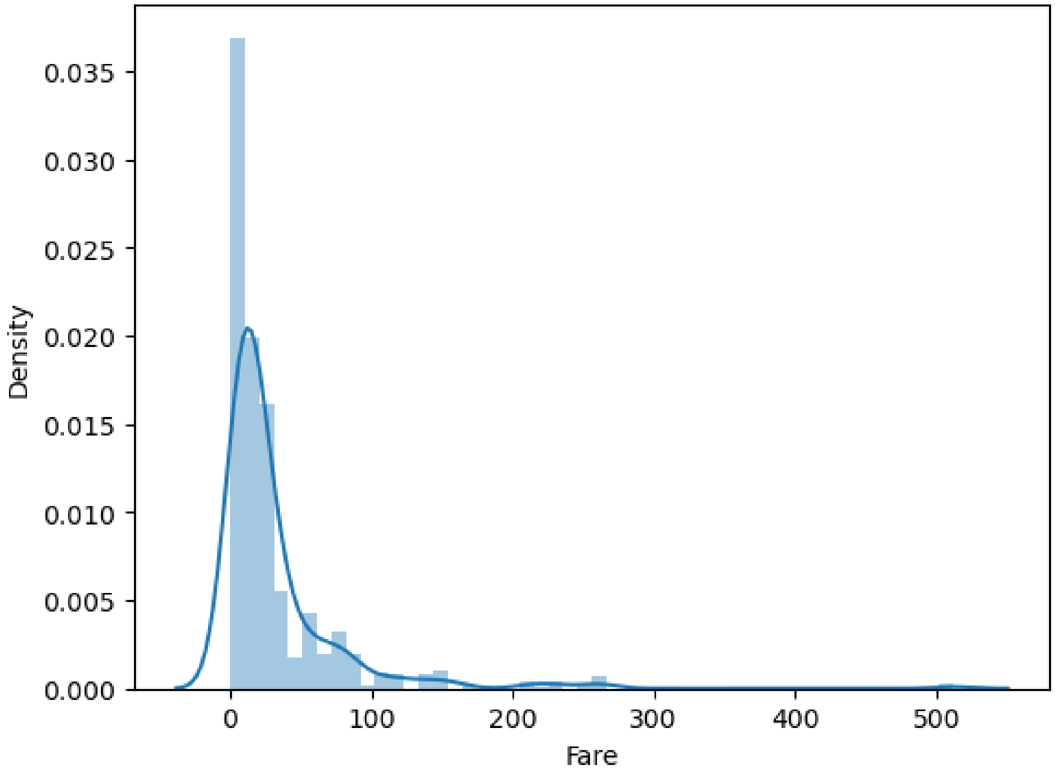
In [14]:

```
sns.distplot(df['Fare'])
```

```
C:\Users\SMD IRFAN\AppData\Local\Temp\ipykernel_11360\3425841524.py:1:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.
14.0.

Please adapt your code to use either `displot` (a figure-level functio
n with
similar flexibility) or `histplot` (an axes-level function for histogr
ams).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (http
s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['Fare'])
```

Out[14]:

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



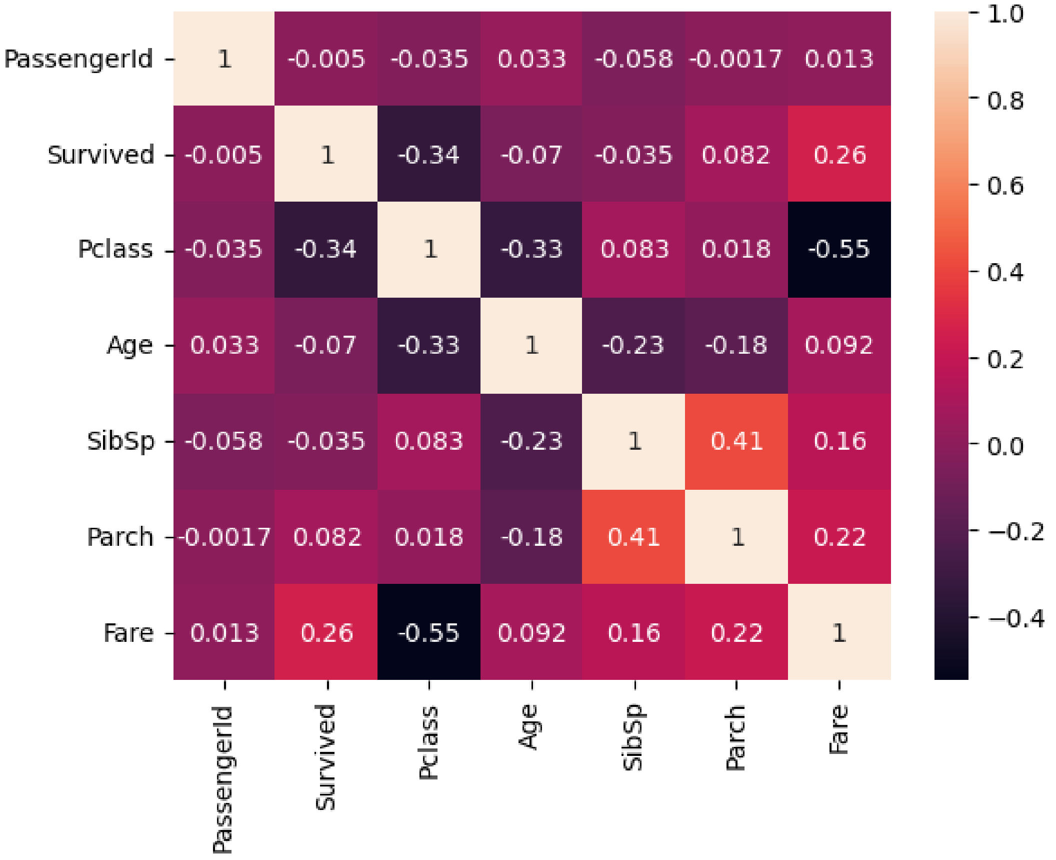
In [15]:

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

```
C:\Users\SMD IRFAN\AppData\Local\Temp\ipykernel_11360\4277794465.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only
valid columns or specify the value of numeric_only to silence this war
ning.
  sns.heatmap(df.corr(),annot=True)
```

Out[15]:

```
<Axes: >
```



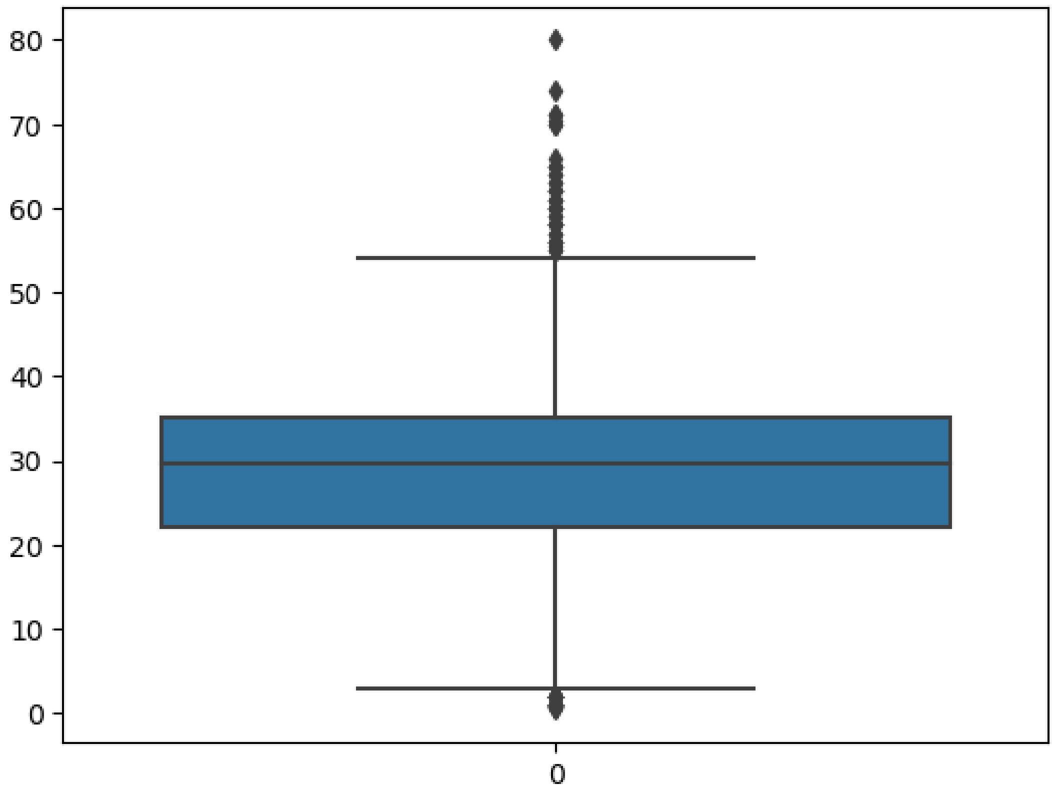
5.OUTLIER DETECTION

In [16]:

```
# Outlier Detection

sns.boxplot(df['Age'])
```

Out[16]:

```
<Axes: >
```

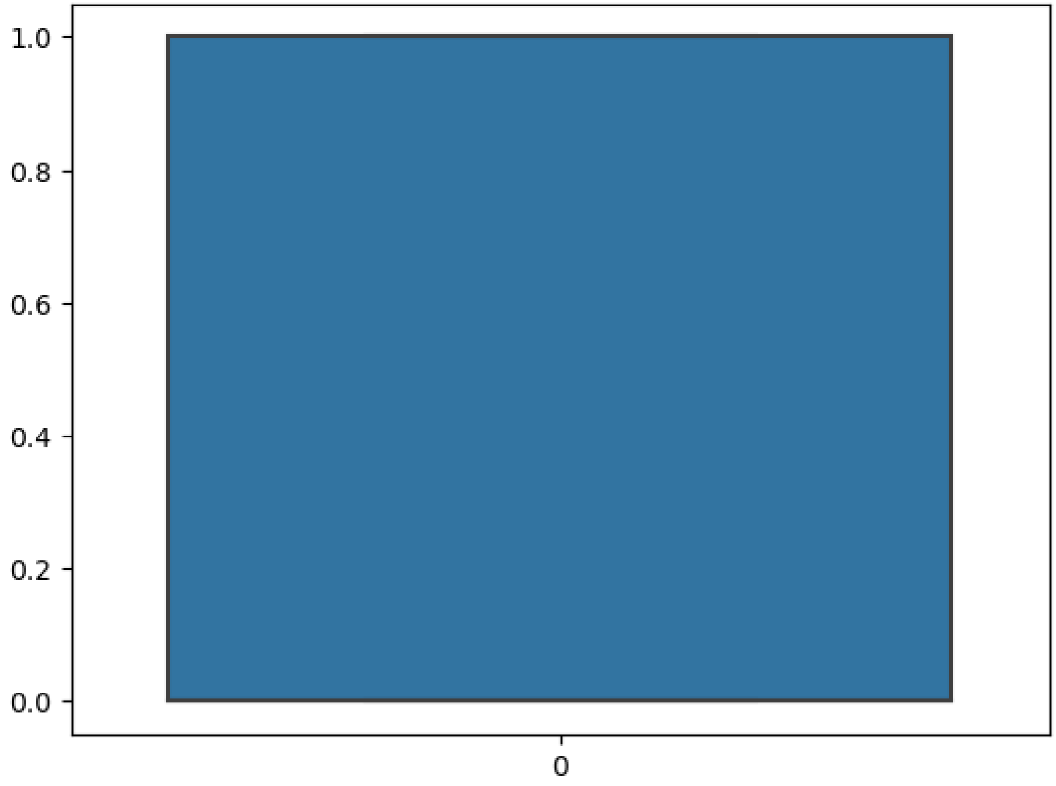


In [17]:

```
sns.boxplot(df['Survived'])
```

Out[17]:

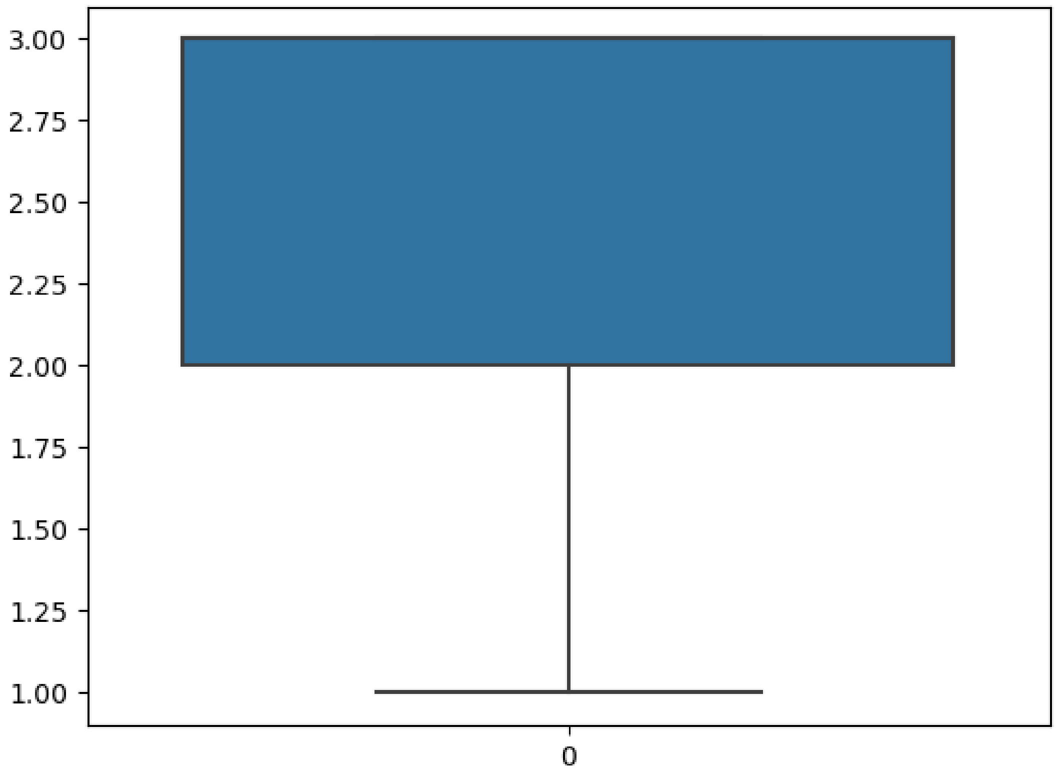
```
<Axes: >
```



In [18]:

```
sns.boxplot(df['Pclass'])
```

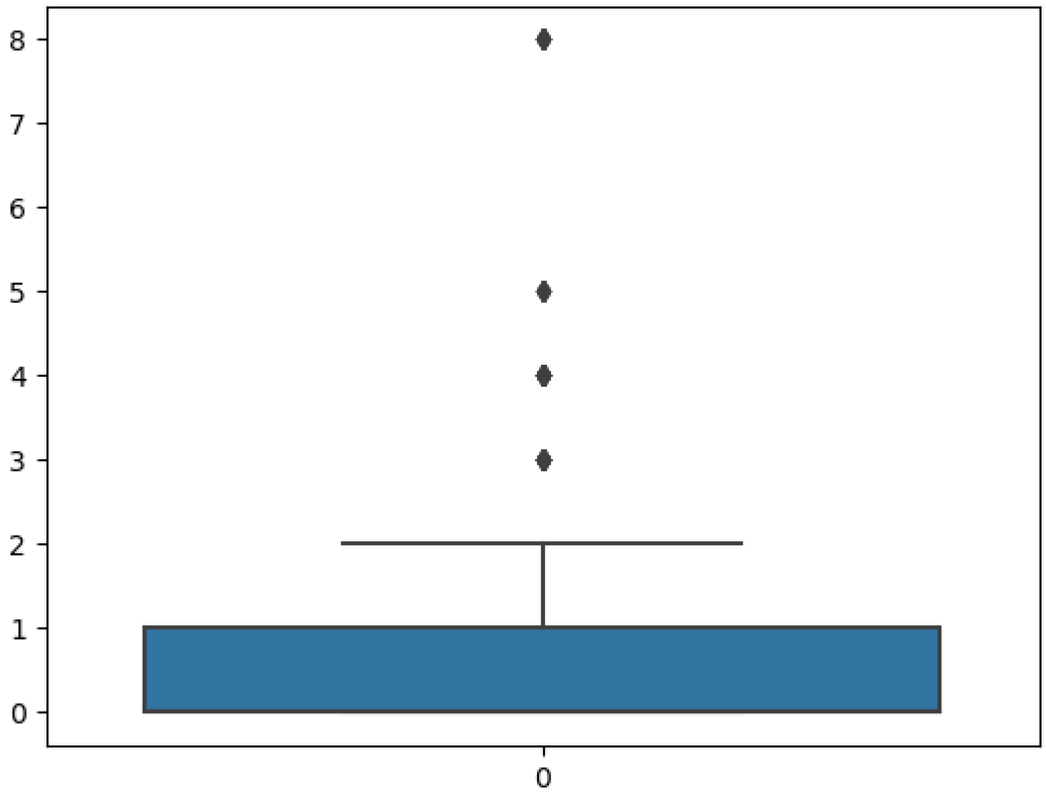
Out[18]: <Axes: >



In [19]:

```
sns.boxplot(df['SibSp'])
```

Out[19]: <Axes: >



In [20]:

```python
Q1 = np.percentile(df['Age'], 25)
Q3 = np.percentile(df['Age'],75)
IQR = Q3 - Q1
lower_bound = Q1 - 3* IQR  # Define k based on your requirement
upper_bound = Q3 + 3 * IQR
outliers = np.where((df['Age'] < lower_bound) | (df['Age'] > upper_
```

In [21]:

```python
print(outliers)
```

```
(array([630], dtype=int64),)
```

6.SPLITTING DEPENDENT AND INDEPENDENT VARIABLES

In [22]:

```python
x =df.iloc[:,2:]
x.head()
```

Out[22]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S

In [23]:

```python
y=df.iloc[:,1:2]
y.head()
```

Out[23]:

	Survived
0	0
1	1
2	1
3	1
4	0

In [24]:

```python
df.shape
```

Out[24]:

```
(891, 12)
```

In [25]:

```python
x.shape
```

Out[25]:

```
(891, 10)
```

In [26]:

```python
y.shape
```

Out[26]:

```
(891, 1)
```

7.PERFORMING ENCODING

In [27]:

```python
# Encoding
from sklearn.preprocessing import LabelEncoder
```

In [28]:

```python
le=LabelEncoder()
```

In [29]:

```python
x["Sex"]=le.fit_transform(x["Sex"])
x["Embarked"]=le.fit_transform(x["Embarked"])
x["Name"]=le.fit_transform(x["Name"])
x["Ticket"]=le.fit_transform(x["Ticket"])
x["Cabin"]=le.fit_transform(x["Cabin"])
```

In [30]:

```python
x["Sex"].value_counts()
x["Embarked"].value_counts()
```

Out[30]:

```
2    646
0    168
1     77
Name: Embarked, dtype: int64
```

In [31]:

```
x.head()
```

Out[31]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	108	1	22.0	1	0	523	7.2500	47	2
1	1	190	0	38.0	1	0	596	71.2833	81	0
2	3	353	0	26.0	0	0	669	7.9250	47	2
3	1	272	0	35.0	1	0	49	53.1000	55	2
4	3	15	1	35.0	0	0	472	8.0500	47	2

8.SPLITTING DATA INTO TRAIN AND TEST

In [32]:

```
# Splitting into test and train
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,ra
```

In [33]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[33]:

```
((623, 10), (268, 10), (623, 1), (268, 1))
```

In [34]:

```
a=[1,2,3,4,5]
b=[0,1,1,2,1]

for i in range(5):
    a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0
    print("with random state",a_train)
```

```
with random state [4, 5, 1]
with random state [4, 5, 1]
with random state [4, 5, 1]
with random state [4, 5, 1]
with random state [4, 5, 1]
```

9.FEATURE SCALING

In [35]:

```
#Feature Scaling
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

In [36]:

```
x_scaled
```

Out[36]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
0	1.0	0.121348	1.0	0.271174	0.125	0.000000	0.769118	0.014151	0.321918	
1	0.0	0.213483	0.0	0.472229	0.125	0.000000	0.876471	0.139136	0.554795	
2	1.0	0.396629	0.0	0.321438	0.000	0.000000	0.983824	0.015469	0.321918	
3	0.0	0.305618	0.0	0.434531	0.125	0.000000	0.072059	0.103644	0.376712	
4	1.0	0.016854	1.0	0.434531	0.000	0.000000	0.694118	0.015713	0.321918	
...	...	...	...	...	...	...	...	...	...	
886	0.5	0.615730	1.0	0.334004	0.000	0.000000	0.148529	0.025374	0.321918	
887	0.0	0.340449	0.0	0.233476	0.000	0.000000	0.020588	0.058556	0.205479	
888	1.0	0.464045	0.0	0.367921	0.125	0.333333	0.992647	0.045771	0.321918	
889	0.0	0.091011	1.0	0.321438	0.000	0.000000	0.011765	0.058556	0.410959	
890	1.0	0.247191	1.0	0.396833	0.000	0.000000	0.685294	0.015127	0.321918	

891 rows × 10 columns

In [ ]:

```

```