Perform Data preprocessing on Titanic dataset

1.Data Collection.

Please download the dataset from

https://www.kaggle.com/datasets/yasserh/titanic-dataset

2.Data Preprocessing

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

Importing the Libraries

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

Importing the dataset

```
df=pd.read_csv('/content/Titanic-Dataset.csv')
```

```
df
```

	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

```
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
```

```
df.shape
```

```
(891, 12)
```

```
df.describe()
```

	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

Checking for null values

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

```
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
```

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

```
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
```

```
df['Age'].fillna(df['Age'].median(),inplace=True)
df.drop('Cabin',axis=1,inplace=True)
df.isnull().sum()
```

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

```
Mode_Embarked=df['Embarked'].mode()[0]
df['Embarked'].fillna(Mode_Embarked,inplace=True)
df1=df
df.isnull().sum()
```

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

Data Vizualization

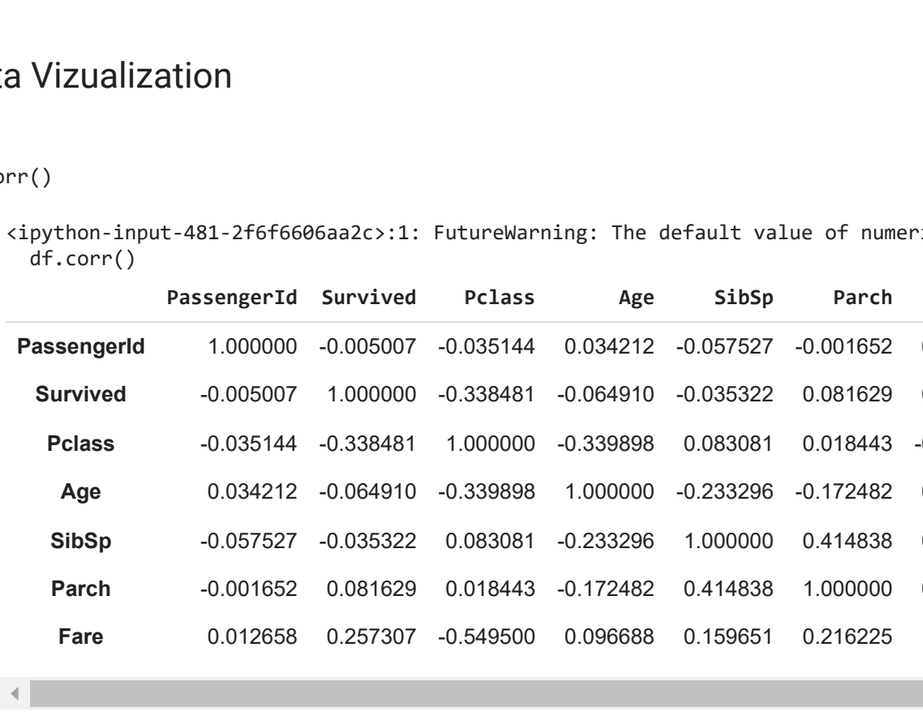
```
df.corr()
```

```
<ipython-input-481-2f6f6606aa2c>: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 colum
  df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.034212	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.064910	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.339898	0.083081	0.018443	-0.549500
Age	0.034212	-0.064910	-0.339898	1.000000	-0.233296	-0.172482	0.096688
SibSp	-0.057527	-0.035322	0.083081	-0.233296	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.172482	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096688	0.159651	0.216225	1.000000

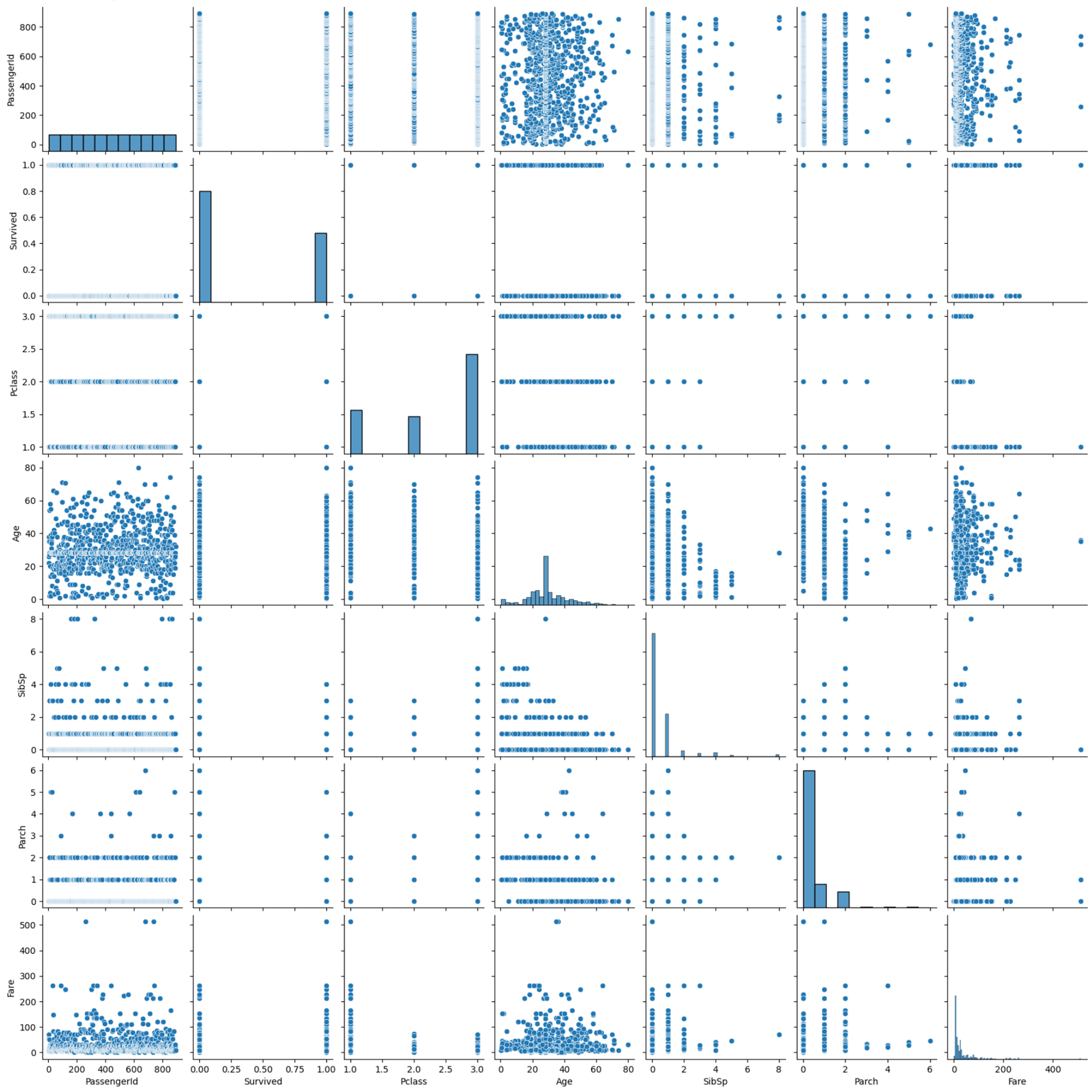
```
sns.heatmap(df.corr())
```

```
<ipython-input-482-aa4f4450a243>: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 colum
  sns.heatmap(df.corr())
<Axes: >
```

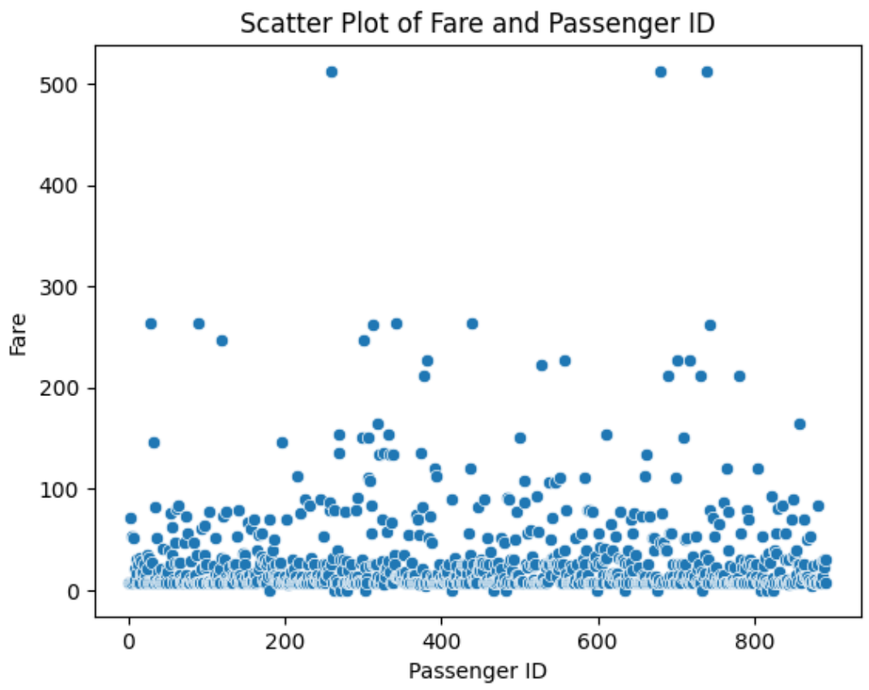


```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x782864b06140>
```

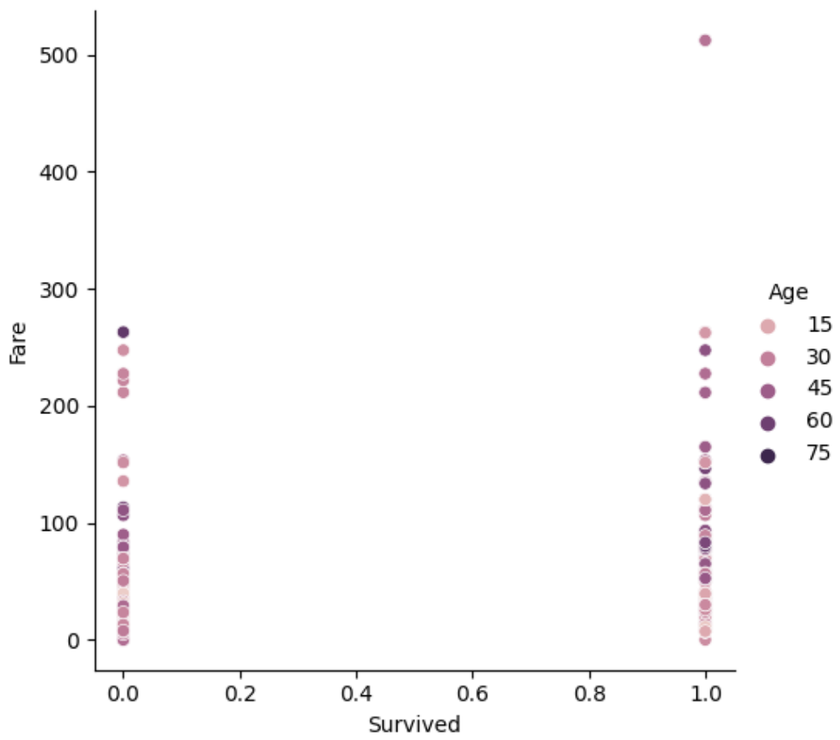



```
sns.scatterplot(x='PassengerId',y='Fare',data=df)
plt.xlabel('Passenger ID')
plt.ylabel('Fare')
plt.title('Scatter Plot of Fare and Passenger ID')
plt.show()
```



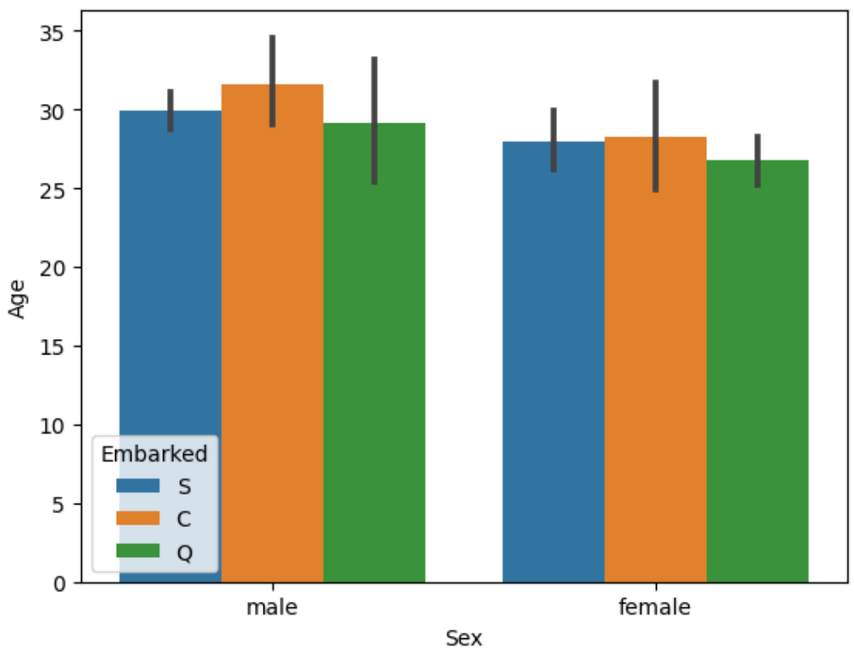
```
sns.relplot(x='Survived',y='Fare',data=df,hue='Age')
```

```
<seaborn.axisgrid.FacetGrid at 0x782863d5cd90>
```



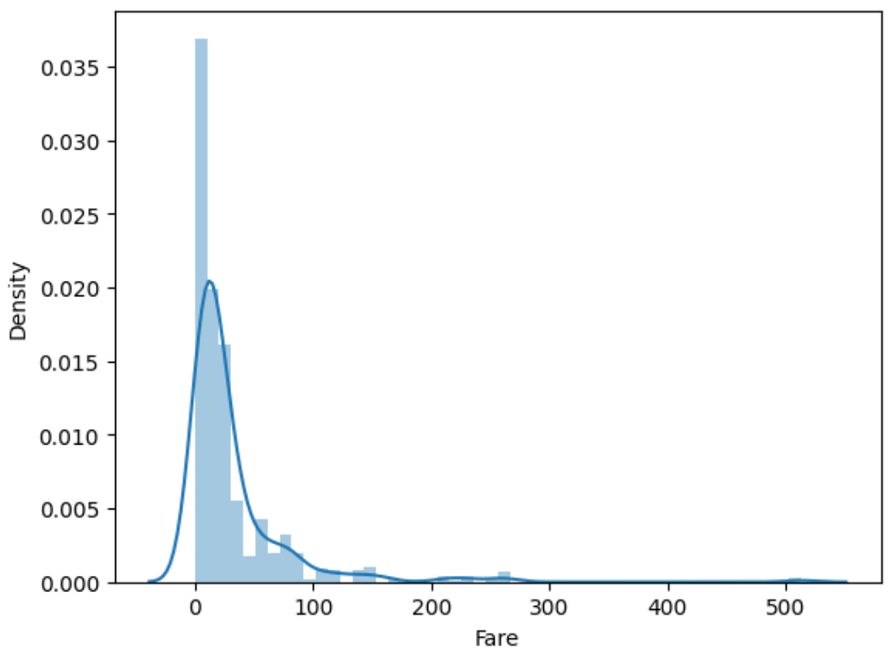

```
sns.barplot(x='Sex',y='Age',data=df,hue='Embarked')
```

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




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

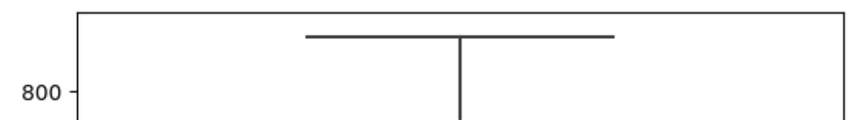
```
<ipython-input-458-70b4b4beb1b5>:1: UserWarning:

`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 function with
similar flexibility) or `histplot` (an axes-level function for histograms).

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

  sns.distplot(df['Fare'])
<Axes: xlabel='Fare', ylabel='Density'>
```



Outlier Detection

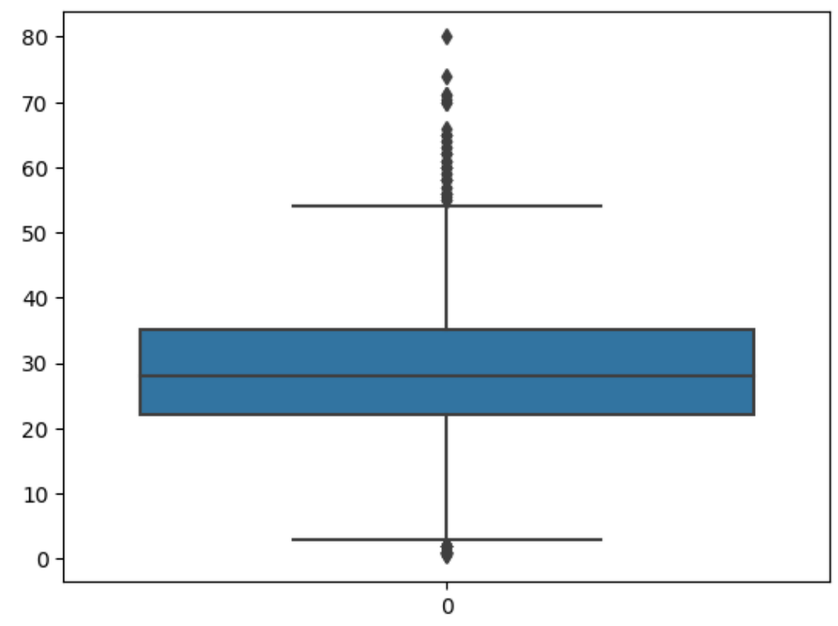
```
sns.boxplot(df['PassengerId'])
```

<Axes: >



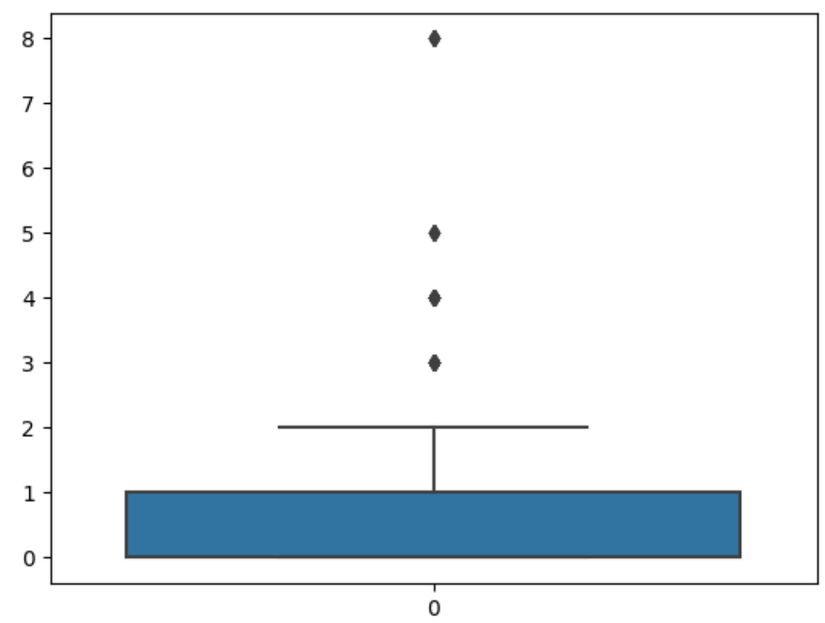

```
sns.boxplot(df['Age'])
```

<Axes: >



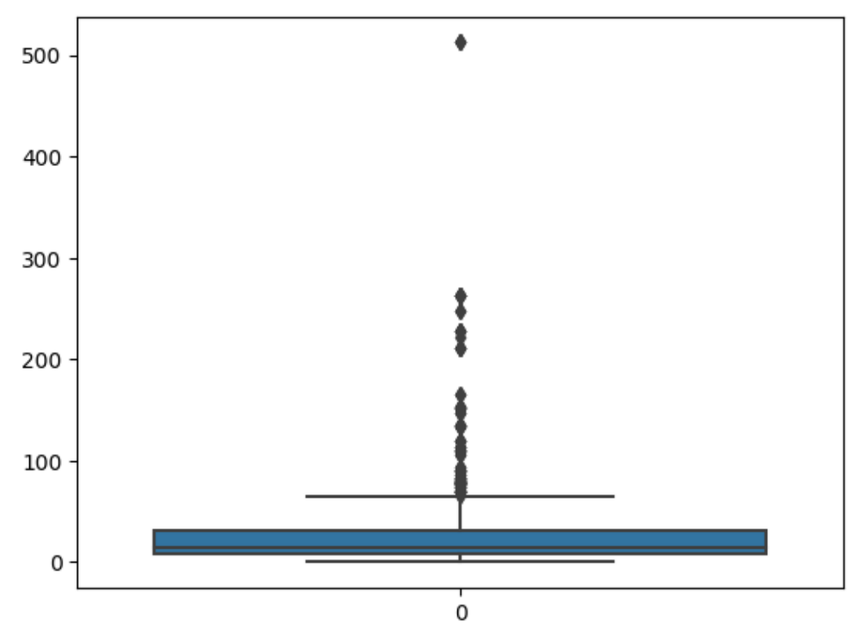

```
sns.boxplot(df['SibSp'])
```

<Axes: >



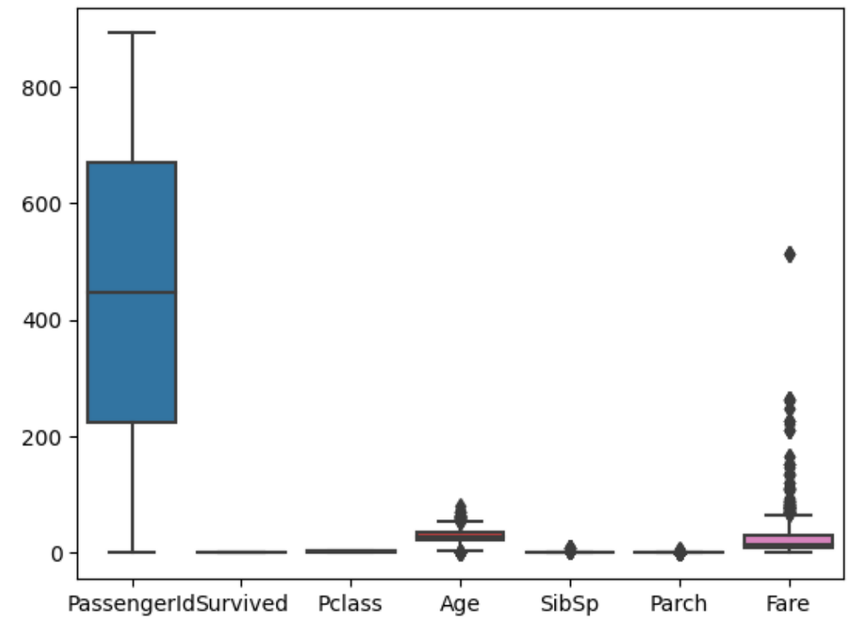

```
sns.boxplot(df['Fare'])
```

<Axes: >




```
sns.boxplot(df)
```

<Axes: >



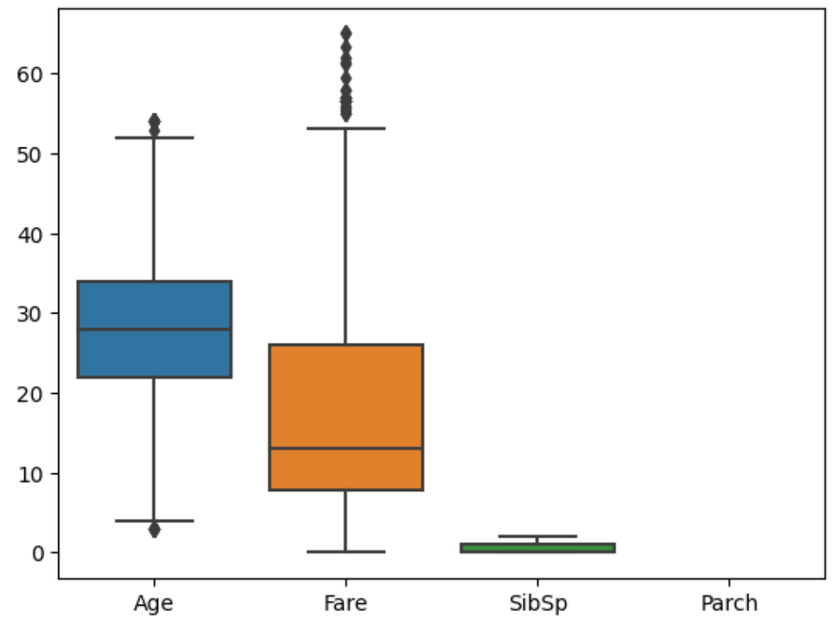

```
a=['Age','Fare','SibSp','Parch']


# Calculate the quartiles and IQR of the variable
Q1 = df[a].quantile(0.25)
Q3 = df[a].quantile(0.75)
IQR = Q3 - Q1
```

```
# Calculate the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Remove outliers from the DataFrame
df = df[(df[a] > lower_bound) & (df[a] < upper_bound)]
print(lower_bound)
print(upper_bound)
```

```
Age       2.500
Fare    -26.724
SibSp    -1.500
Parch     0.000
dtype: float64
Age      54.5000
Fare     65.6344
SibSp     2.5000
Parch     0.0000
dtype: float64
```

```
sns.boxplot(df[a])
```

```
<Axes: >
```



Splitting Dependent and Independent Variables

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  0 non-null      float64
 1   Survived     0 non-null      float64
 2   Pclass       0 non-null      float64
 3   Name         0 non-null      object
 4   Sex          0 non-null      object
 5   Age          825 non-null    float64
 6   SibSp        845 non-null    float64
 7   Parch        0 non-null      float64
 8   Ticket       0 non-null      object
 9   Fare         775 non-null    float64
 10  Embarked     0 non-null      object
dtypes: float64(7), object(4)
memory usage: 76.7+ KB
```

```
df1.head()
```

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

```
x=df1.iloc[:,4:11]
y=df1.iloc[:,1:2]
x=x.drop('Ticket',axis=1)
```

```
x.head()
```

	Sex	Age	SibSp	Parch	Fare	Embarked
0	male	22.0	1	0	7.2500	S
1	female	38.0	1	0	71.2833	C
2	female	26.0	0	0	7.9250	S
3	female	35.0	1	0	53.1000	S
4	male	35.0	0	0	8.0500	S

```
x.shape
```

```
(891, 6)
```

```
y.shape
```

```
(891, 1)
```

Perform Encoding

```
from sklearn.preprocessing import LabelEncoder
```

```
l=LabelEncoder()


x['Sex']=l.fit_transform(x['Sex'])


x['Sex']
```

```
     0      1
     1      0
     2      0
     3      0
     4      1
           ..
     886    1
     887    0
     888    0
     889    1
     890    1
     Name: Sex, Length: 891, dtype: int64
```

```
x['Embarked']=l.fit_transform(x['Embarked'])


x['Embarked']
```

```
     0      2
     1      0
     2      2
     3      2
     4      2
           ..
     886    2
     887    2
     888    2
     889    0
     890    1
     Name: Embarked, Length: 891, dtype: int64
```

```
x.head()
```

	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	22.0	1	0	7.2500	2
1	0	38.0	1	0	71.2833	0
2	0	26.0	0	0	7.9250	2
3	0	35.0	1	0	53.1000	2
4	1	35.0	0	0	8.0500	2

Splitting into test and train data

```
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,random_state=0)


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

```
((623, 6), (268, 6), (623, 1), (268, 1))
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()


x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)


x_train
```

```
array([[ 0.72592065,  1.64654836, -0.457246  , -0.47299765, -0.12253019,
         0.56710989],
       [-1.37756104,  1.4930717 ,  0.4033711 , -0.47299765,  0.91812372,
        -2.03075381],
       [ 0.72592065, -2.19036814,  3.8458395 ,  1.93253327,  0.29950338,
         0.56710989],
       ...,
       [ 0.72592065, -0.11843323, -0.457246  , -0.47299765, -0.51276504,
        -0.73182196],
       [-1.37756104,  0.49547341,  0.4033711 , -0.47299765, -0.31228976,
         0.56710989],
       [ 0.72592065,  2.33719333,  0.4033711 ,  0.72976781,  0.13566725,
         0.56710989]])
```

```
x_test
```

```
array([[ 0.76537495, -0.0724674 , -0.53120385, -0.47809977, -0.324475  ,
        -1.76531134],
       [ 0.76537495, -0.0724674 , -0.53120385, -0.47809977, -0.45513843,
         0.63014911],
       [ 0.76537495, -1.69302814,  3.68694819,  0.87064484, -0.04706937,
        -0.56758111],
       ...,
       [ 0.76537495, -0.14963696,  0.52333416, -0.47809977, -0.32455255,
        -1.76531134],
       [-1.30654916, -0.84416299, -0.53120385, -0.47809977, -0.45616356,
         0.63014911],
       [ 0.76537495, -0.0724674 , -0.53120385, -0.47809977, -0.07362838,
        -1.76531134]])
```