In [ ]:
```
# Data Collection.
# Please download the dataset from
# https://www.kaggle.com/datasets/yasserh/titanic-datase
```

1.import the necessary libraries

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

2.import the dataset

In [4]:
```
df=pd.read_csv('Titanic-Dataset.csv')
df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

In [5]:
```
df.head()
```

Out[5]:

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

In [6]:
```
df.tail()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN

In [7]:
```
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 [8]:
```
df.describe()
```

Out[8]:

	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 [9]:
```
df.shape
```

Out[9]:
```
(891, 12)
```

In [10]:
```
corr=df.corr()
corr
```

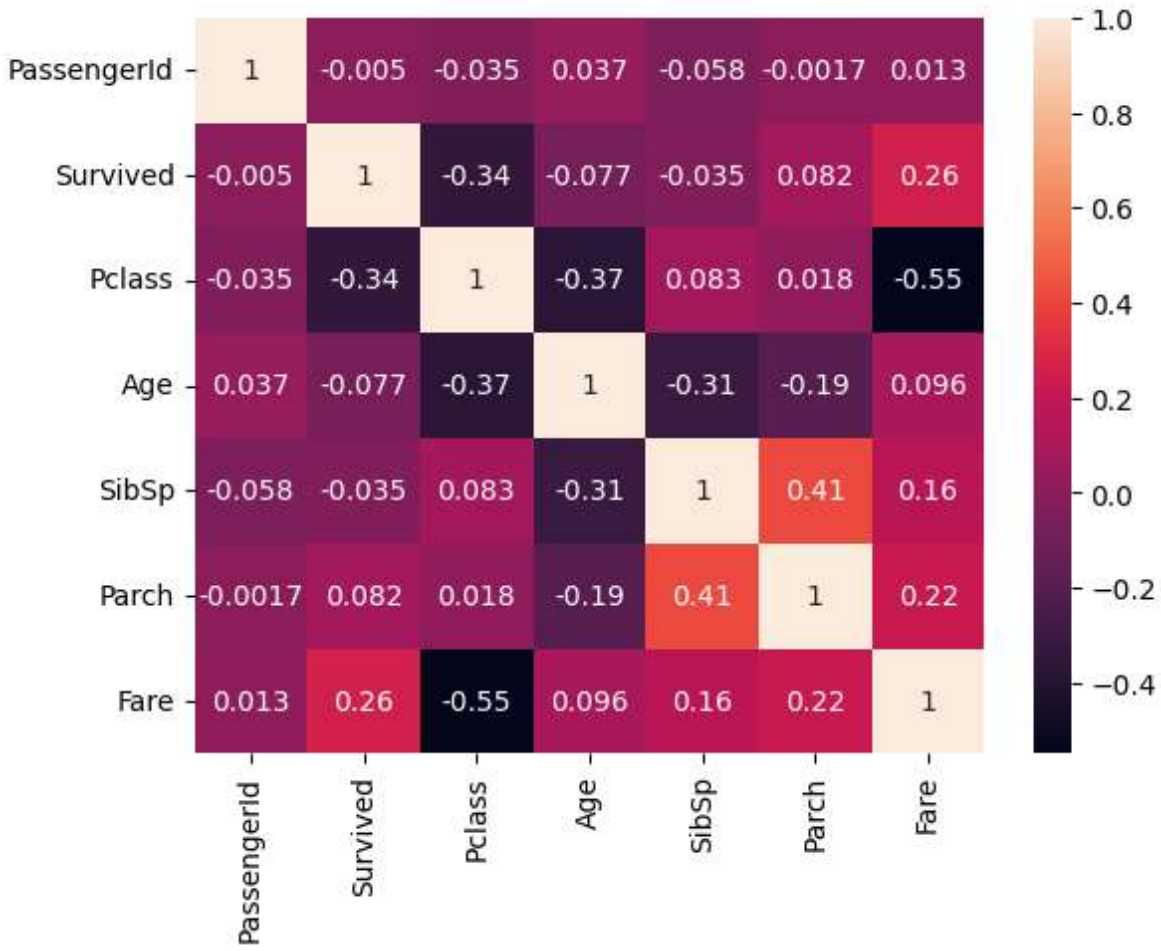
```
C:\Users\gurug\AppData\Local\Temp\ipykernel_16764\3182140910.py:1: FutureWarn
ing: The default value of numeric_only in DataFrame.corr is deprecated. In a
future version, it will default to False. Select only valid columns or specif
y the value of numeric_only to silence this warning.
  corr=df.corr()
```

Out[10]:

	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 [11]:
```
sns.heatmap(corr,annot=True)
```

Out[11]: <Axes: >



In [12]:
```
df.Survived.value_counts()
```

Out[12]:
```
0    549
1    342
Name: Survived, dtype: int64
```

In [13]:
```
df.Sex.value_counts()
```

Out[13]:
```
male      577
female    314
Name: Sex, dtype: int64
```

3.Handling null values

In [14]:
```
df.isnull().any()
```

Out[14]:
```
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 [15]:
```
df.isnull().sum()
```

Out[15]:
```
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 [20]:
```
df['Age'].fillna(df['Age'].mean(),inplace=True)
df['Cabin'].fillna(df['Cabin'].mode()[0],inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)
```

In [21]:
```
df
```

Out[21]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	I
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0
...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7

891 rows × 12 columns

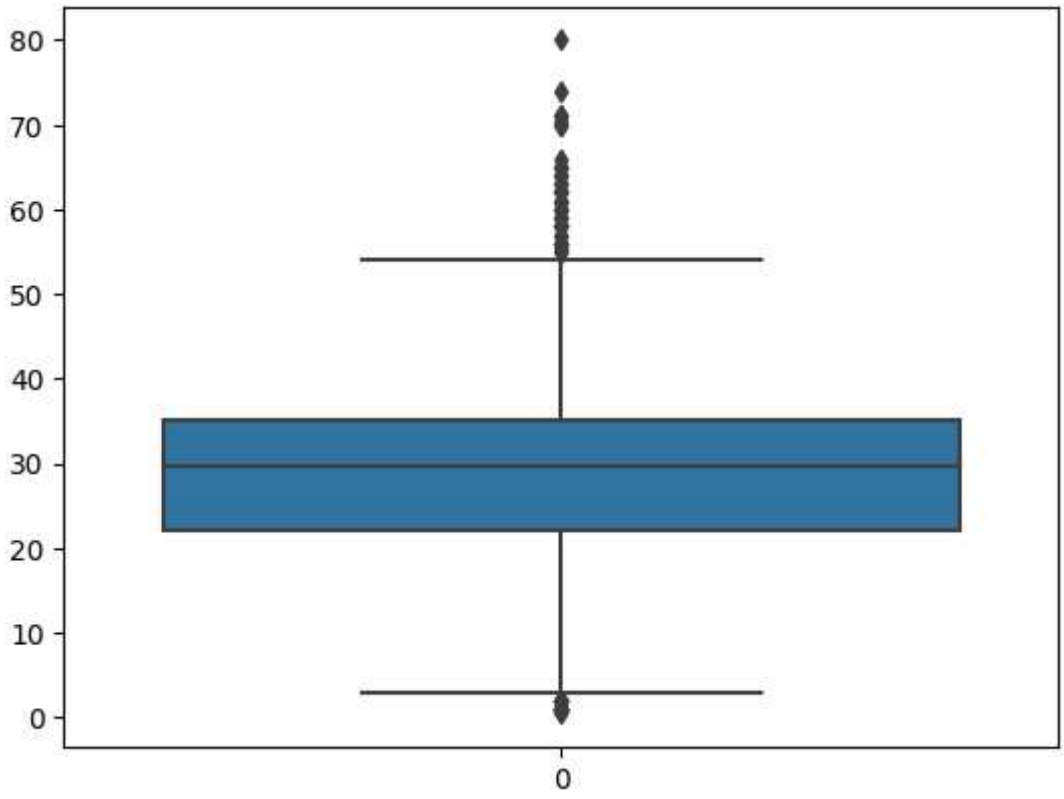
In [22]:
```
df.isnull().any()
```

Out[22]:
```
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.outliers

In [23]:
```
sns.boxplot(df.Age)
```

Out[23]:
```
<Axes: >
```



5.Seperate dependent and independent variables

In [24]:
```
x=df.iloc[:,2:12]
y=df.iloc[:,1:2]
```

In [25]:
```
df.head()
```

Out[25]:

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

In [26]:
```
y.head()
```

Out[26]:

	Survived
0	0
1	1
2	1
3	1
4	0

In [28]:
```
df.shape
```

Out[28]: (891, 12)

In [30]:
```
x.shape
```

Out[30]: (891, 10)

In [31]:
```
y.shape
```

Out[31]:
```
(891, 1)
```

6.Encoding

In [32]:
```
from sklearn.preprocessing import LabelEncoder
```

In [33]:
```
le=LabelEncoder()
```

In [34]:
```
x['Sex']=le.fit_transform(x['Sex'])
```

In [35]:
```
x['Sex']
```

Out[35]:
```
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: int32
```

In [36]:
```
x['Sex'].value_counts()
```

Out[36]:
```
1    577
0    314
Name: Sex, dtype: int64
```

In [37]:
```
x['Sex'].nunique()
```

Out[37]:
```
2
```

In [38]:
```
x.head()
```

Out[38]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	B96 B98	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	G6	S
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S
4	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	B96 B98	S

In [50]:
```
x['Name']=le.fit_transform(x['Name'])
```

In [51]:
```
x['Ticket']=le.fit_transform(x['Ticket'])
```

In [52]:
```
x['Cabin']=le.fit_transform(x['Cabin'])
```

In [60]:
```
x['Embarked']=le.fit_transform(x['Embarked'])
```

7.splitting into training and testing set

In [72]:
```
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=
```

In [73]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[73]:
```
((623, 10), (268, 10), (623, 1), (268, 1))
```

8.Feature Scaling

In [64]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

In [65]:
```
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

In [66]:
```
x_train
```

Out[66]:
```
array([[-1.5325562 , -0.9248674 ,  0.72592065, ..., -0.12253019,
         2.63667804,  0.56710989],
       [-1.5325562 , -0.3910637 , -1.37756104, ...,  0.91812372,
         1.90511193, -2.03075381],
       [ 0.84844757, -0.52743691,  0.72592065, ...,  0.29950338,
        -0.28958641,  0.56710989],
       ...,
       [ 0.84844757,  0.65706181,  0.72592065, ..., -0.51276504,
        -0.28958641, -0.73182196],
       [ 0.84844757,  1.75973661, -1.37756104, ..., -0.31228976,
        -0.28958641,  0.56710989],
       [-0.34205431, -1.2521631 ,  0.72592065, ...,  0.13566725,
        -0.28958641,  0.56710989]])
```

In [67]:
```
x_test
```

Out[67]:
```
array([[ 0.77963055,  1.59061177,  0.76537495, ..., -0.324475  ,
        -0.27189793, -1.76531134],
       [ 0.77963055,  1.53212698,  0.76537495, ..., -0.45513843,
        -0.27189793,  0.63014911],
       [ 0.77963055,  0.81471352,  0.76537495, ..., -0.04706937,
        -0.27189793, -0.56758111],
       ...,
       [ 0.77963055,  1.57111684,  0.76537495, ..., -0.32455255,
        -0.27189793, -1.76531134],
       [ 0.77963055,  1.09544053, -1.30654916, ..., -0.45616356,
        -0.27189793,  0.63014911],
       [-1.64991582,  1.2163091 ,  0.76537495, ..., -0.07362838,
        -0.27189793, -1.76531134]])
```

In [68]:
```
y_train=sc.fit_transform(y_train)
y_test=sc.fit_transform(y_test)
```

In [69]:
```
y_train
```

```
       [-0.79697591],
       [-0.79697591],
       [-0.79697591],
       [ 1.25474307],
       [ 1.25474307],
       [-0.79697591],
       [-0.79697591],
       [ 1.25474307],
       [ 1.25474307],
       [ 1.25474307],
       [-0.79697591],
       [-0.79697591],
       [-0.79697591],
       [ 1.25474307],
       [-0.79697591],
       [-0.79697591],
       [ 1.25474307],
       [-0.79697591],
       [ 1.25474307],
       [-0.79697591],
```

In [70]:
```
y_test
```

Out[70]:
```
array([[-0.77151675],
       [-0.77151675],
       [-0.77151675],
       [ 1.29614814],
       [ 1.29614814],
       [ 1.29614814],
       [ 1.29614814],
       [ 1.29614814],
       [ 1.29614814],
       [ 1.29614814],
       [-0.77151675],
       [ 1.29614814],
       [-0.77151675],
       [ 1.29614814],
       [ 1.29614814],
       [-0.77151675],
       [-0.77151675],
       [-0.77151675],
       [-0.77151675],
```

In [ ]: