In [ ]:
```
1
```

TASK

```
-->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 LIBRARIES

In [1]:
```
1
2
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import seaborn as sns
```

Importing the dataset.

In [2]:
```
1 df=pd.read_csv("Titanic-Dataset.csv")
```

In [3]:
```
1 df.head()
```

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

Checking for Null Values.

In [4]:
```
1 df.isnull().sum()
```

Out[4]:
```
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 [5]:
```
1 df["Age"].fillna(df["Age"].mean(),inplace=True)
```

In [6]:
```
df.Age.isnull().sum()
```

Out[6]:
```
0
```

In [7]:
```
df["Cabin"].fillna(df["Cabin"].mode().iloc[0],inplace=True)
```

In [8]:
```
df.Cabin.isnull().sum()
```

Out[8]:
```
0
```

In [9]:
```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

In [10]:
```
df.Embarked.isnull().sum()
```

Out[10]:
```
0
```

In [11]:
```
df.isnull().sum()
```

Out[11]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

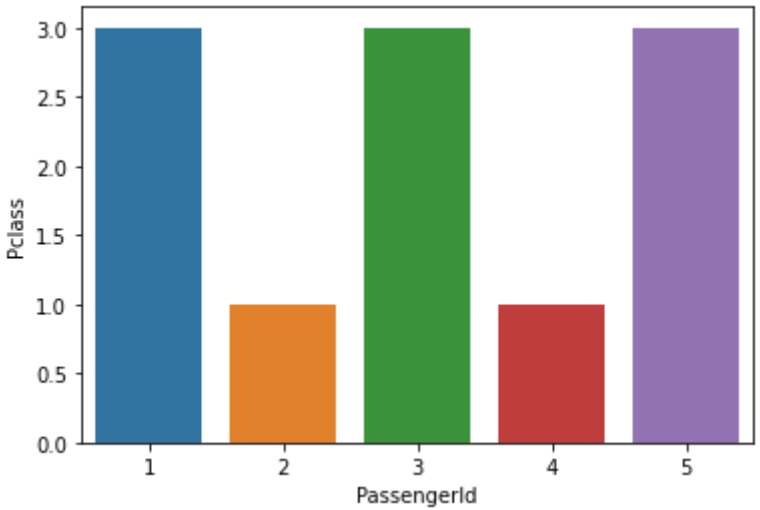
Data visualisation

In [12]:
```
1 data=df.head()
2 data
```

Out[12]:

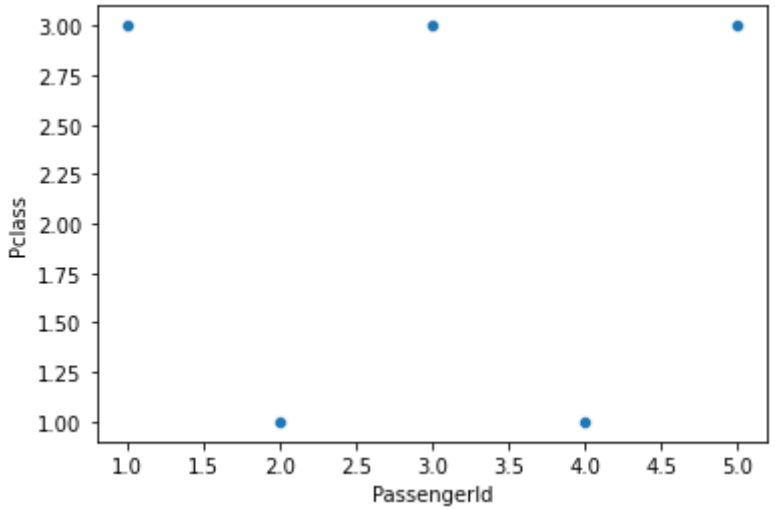
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2.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

In [13]:
```
1 sns.barplot(x="PassengerId",y="Pclass",data=data)
2 plt.show()
```



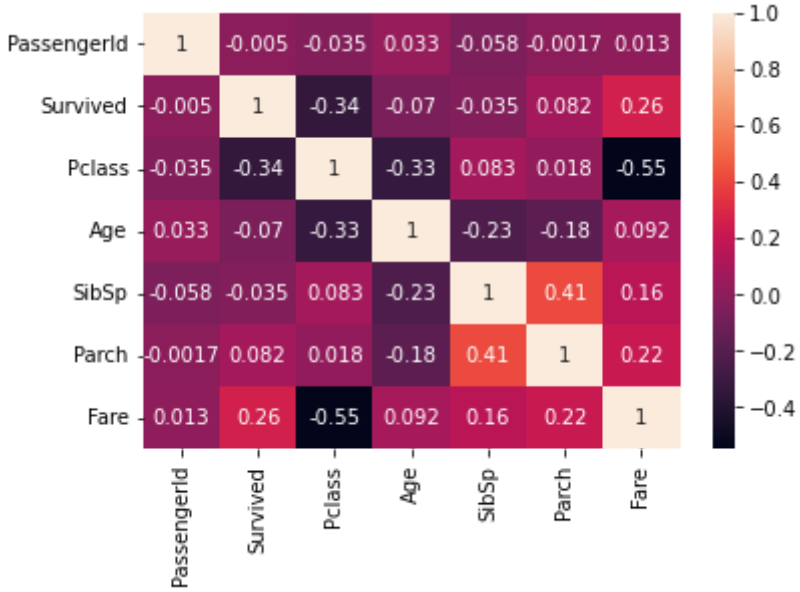
In [14]:
```
1 sns.scatterplot(x="PassengerId",y="Pclass",data=data)
```

Out[14]: <AxesSubplot:xlabel='PassengerId', ylabel='Pclass'>



In [15]:
```
1 sns.heatmap(df.corr(),annot=True)
```

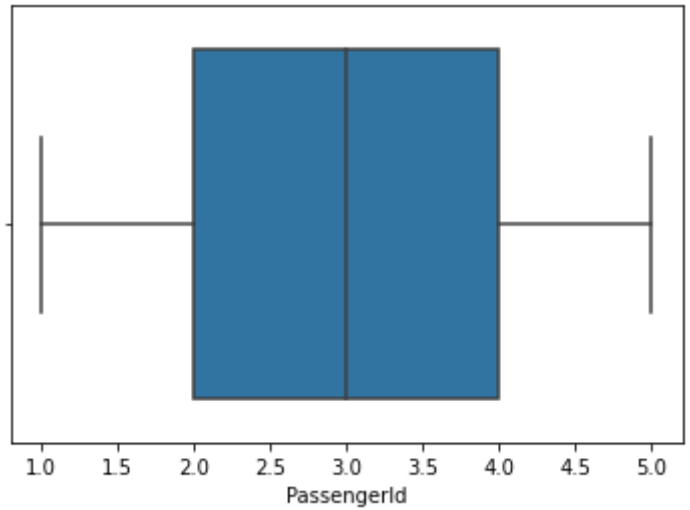
Out[15]: <AxesSubplot:>



In [16]:
```
1 sns.boxplot(data.PassengerId)
```

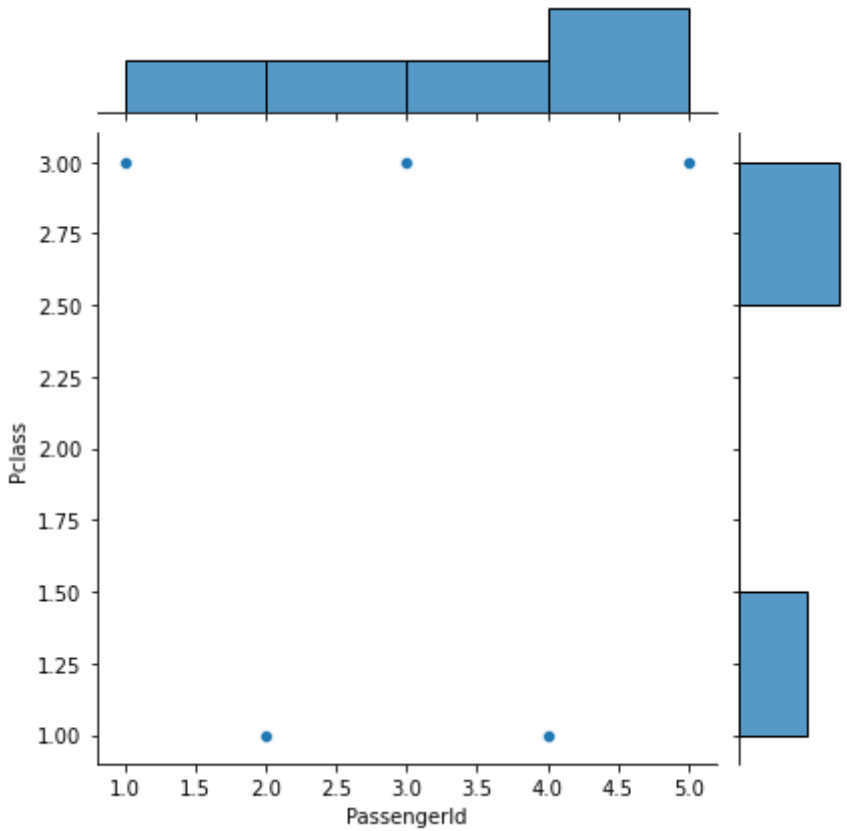
```
C:\Users\adarsha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argument
s without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[16]: <AxesSubplot:xlabel='PassengerId'>



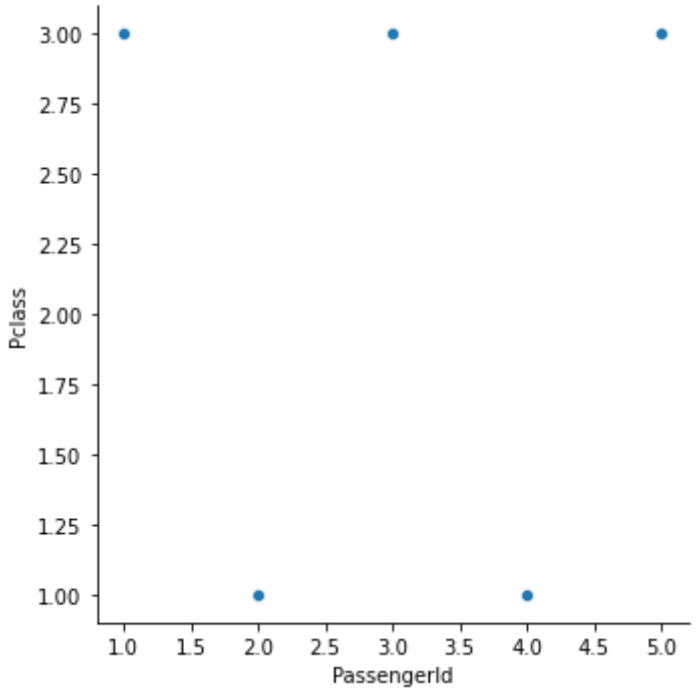
In [17]:
```
1 sns.jointplot(x="PassengerId",y="Pclass",data=data)
```

Out[17]: <seaborn.axisgrid.JointGrid at 0x2398023e4f0>



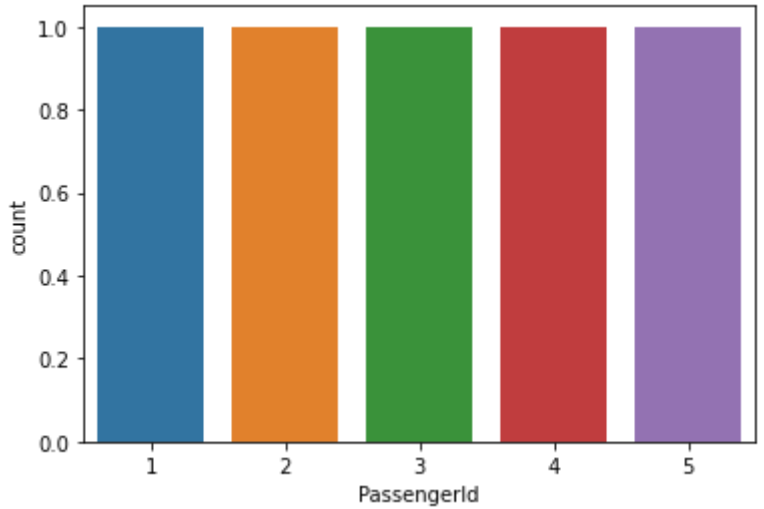
In [18]:
```
1 sns.relplot(x="PassengerId",y="Pclass",data=data)
```

Out[18]: <seaborn.axisgrid.FacetGrid at 0x239ff7fcd00>



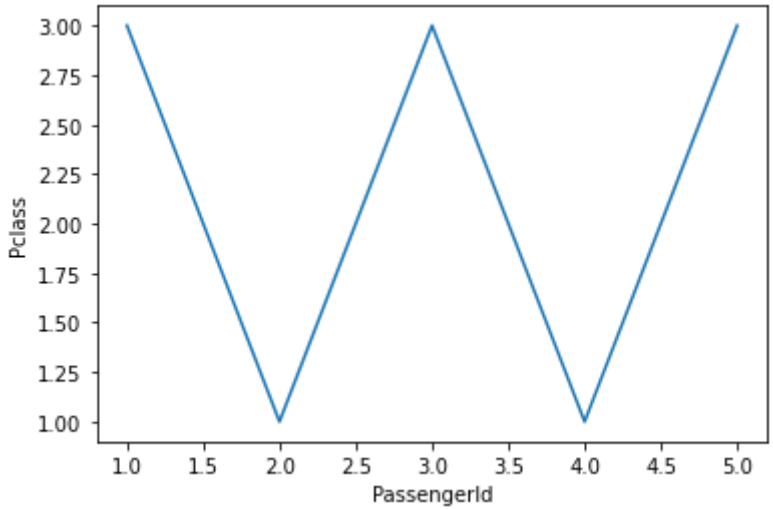
In [19]:
```
1 sns.countplot(x="PassengerId",data=data)
```

Out[19]: <AxesSubplot:xlabel='PassengerId', ylabel='count'>



In [20]:
```
1 sns.lineplot(x="PassengerId",y="Pclass",data=data)
```

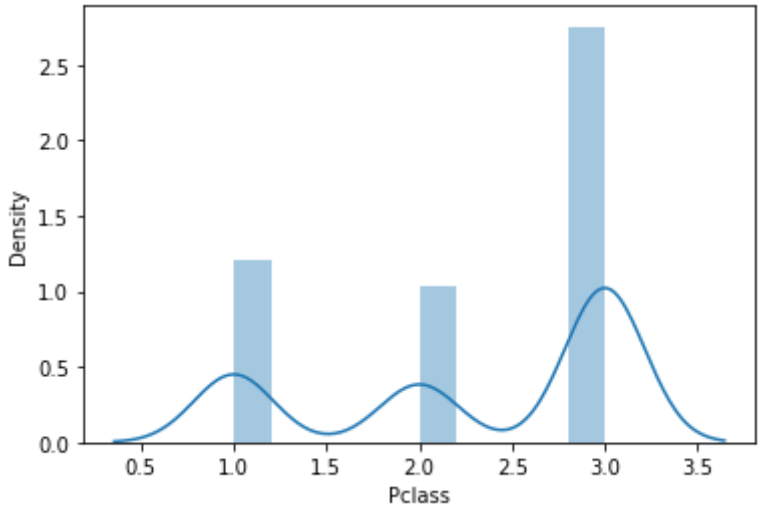
Out[20]: <AxesSubplot:xlabel='PassengerId', ylabel='Pclass'>



In [21]:
```
1 sns.distplot(df["Pclass"])
```

```
C:\Users\adarsha\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level f
unction with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[21]: <AxesSubplot:xlabel='Pclass', ylabel='Density'>

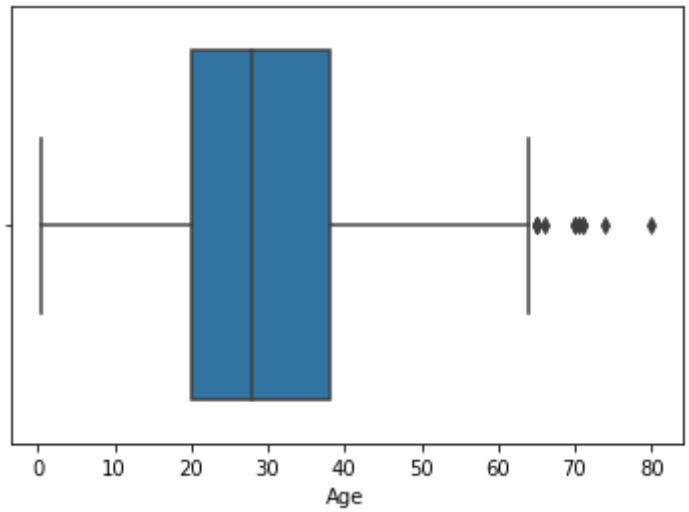


OUTLIER DETECTION

In [22]:
```
1 df=pd.read_csv("Titanic-Dataset.csv")
2
3 sns.boxplot(df.Age)
4
```

```
C:\Users\adarsha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argument
s without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[22]: <AxesSubplot:xlabel='Age'>



In [23]:
```
df.head()
```

Out[23]:

	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

In [24]:
```
df.shape
```

Out[24]:
```
(891, 12)
```

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

Out[25]:

	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 [26]:
```
q1=df.Age.quantile(0.25)
q3=df.Age.quantile(0.75)
print(q1)
print(q3)
```

```
20.125
38.0
```

In [27]:
```
iqr=q3-q1
iqr
```

Out[27]: 17.875

In [28]:
```
upper_limit=q3+1.5*iqr
upper_limit
```

Out[28]: 64.8125

In [29]:
```
lower_limit=q1-1.5*iqr
lower_limit
```

Out[29]: -6.6875

In [30]:
```
df.Age.median()
```

Out[30]: 28.0

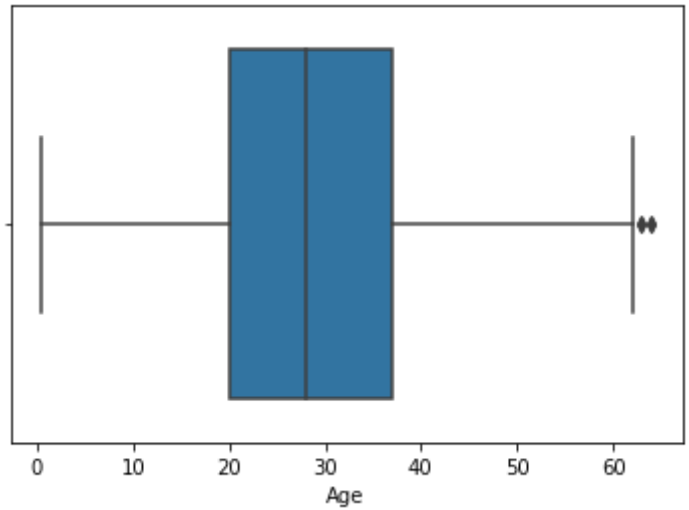
In [31]:
```

df['Age']=np.where(df['Age']>upper_limit,28,df['Age'])
```

In [32]:
```
1 sns.boxplot(df.Age)
```

```
C:\Users\adarsha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable a
s a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argument
s without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[32]: <AxesSubplot:xlabel='Age'>



Splitting Dependent and Independent variables

In [33]:
```
1 df.head()
```

Out[33]:

	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

In [34]:
```
1 x=df.iloc[:,4:]
2 y=df.iloc[:,1:2]
```

In [35]:
```
1 x.head()
```

Out[35]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	female	38.0	1	0	PC 17599	71.2833	C85	C
2	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	female	35.0	1	0	113803	53.1000	C123	S
4	male	35.0	0	0	373450	8.0500	NaN	S

In [36]:
```
1 y.head()
```

Out[36]:

	Survived
0	0
1	1
2	1
3	1
4	0

In [37]:
```
1 x.shape
```

Out[37]: (891, 8)

In [38]:
```
1 y.shape
```

Out[38]: (891, 1)

In [39]:
```
1 df.shape
```

Out[39]: (891, 12)

Encoding

In [40]:
```
1 x.head()
```

Out[40]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	female	38.0	1	0	PC 17599	71.2833	C85	C
2	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	female	35.0	1	0	113803	53.1000	C123	S
4	male	35.0	0	0	373450	8.0500	NaN	S

In [41]:
```
1 from sklearn.preprocessing import LabelEncoder
```

In [42]:
```
1 le=LabelEncoder()
```

In [43]:
```
1 x["Sex"]=le.fit_transform(x["Sex"])
2 x["Sex"]
```

Out[43]:
```
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 [44]:
```
x["Cabin"]=le.fit_transform(x["Cabin"])
x["Cabin"]
```

Out[44]:
```
0      147
1       81
2      147
3       55
4      147
      ...
886    147
887     30
888    147
889     60
890    147
Name: Cabin, Length: 891, dtype: int32
```

In [45]:
```
x["Embarked"]=le.fit_transform(x["Embarked"])
x["Embarked"]
```

Out[45]:
```
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: int32
```

In [46]:
```
x["Ticket"]=le.fit_transform(x["Ticket"])
x["Ticket"]
```

Out[46]:
```
0      523
1      596
2      669
3       49
4      472
      ...
886    101
887     14
888    675
889      8
890    466
Name: Ticket, Length: 891, dtype: int32
```

In [47]:
```
x.head()
```

Out[47]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	22.0	1	0	523	7.2500	147	2
1	0	38.0	1	0	596	71.2833	81	0
2	0	26.0	0	0	669	7.9250	147	2
3	0	35.0	1	0	49	53.1000	55	2
4	1	35.0	0	0	472	8.0500	147	2

In [48]:
```
x.shape
```

Out[48]:
```
(891, 8)
```

SPLITTING TAINING AND TESTING DATASET

In [49]:
```
1 from sklearn.model_selection import train_test_split
```

In [50]:
```
1 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [51]:
```
1 x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[51]: ((623, 8), (268, 8), (623, 1), (268, 1))

FEATURE SCALING

In [52]:
```
1 from sklearn.preprocessing import MinMaxScaler
2 ms=MinMaxScaler()
```

In [53]:
```
1 x_Scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

In [54]:
```
1 x_Scaled
```

Out[54]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1.0	0.339415	0.125	0.000000	0.769118	0.014151	1.000000	0.666667
1	0.0	0.591066	0.125	0.000000	0.876471	0.139136	0.551020	0.000000
2	0.0	0.402328	0.000	0.000000	0.983824	0.015469	1.000000	0.666667
3	0.0	0.543882	0.125	0.000000	0.072059	0.103644	0.374150	0.666667
4	1.0	0.543882	0.000	0.000000	0.694118	0.015713	1.000000	0.666667
...	...	...	...	...	...	...	...	...
886	1.0	0.418056	0.000	0.000000	0.148529	0.025374	1.000000	0.666667
887	0.0	0.292230	0.000	0.000000	0.020588	0.058556	0.204082	0.666667
888	0.0	NaN	0.125	0.333333	0.992647	0.045771	1.000000	0.666667
889	1.0	0.402328	0.000	0.000000	0.011765	0.058556	0.408163	0.000000
890	1.0	0.496697	0.000	0.000000	0.685294	0.015127	1.000000	0.333333

891 rows × 8 columns

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```