assessment-3

September 21, 2023

[26]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[27]:
```
from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Titanic-Dataset.csv to Titanic-Dataset (2).csv
```

[28]:
```
data=pd.read_csv('Titanic-Dataset.csv')
```

[29]:
```
data.head()
```

[29]:

	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

[30]:
```
data.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
```

[31]:
```
data.describe()
```

[31]:

	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

[32]:
```
data.isnull().any()
```

[32]:
```
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
```

[33]:
```
data.isnull().sum()
```

[33]:
```
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
```

[34]:
```
data['Age'].fillna(data['Age'].mean(),inplace=True)
```

[35]:
```
data['Cabin'].fillna('UnKnown',inplace=True)
```

[36]:
```
data['Embarked'].fillna('s',inplace=True)
```

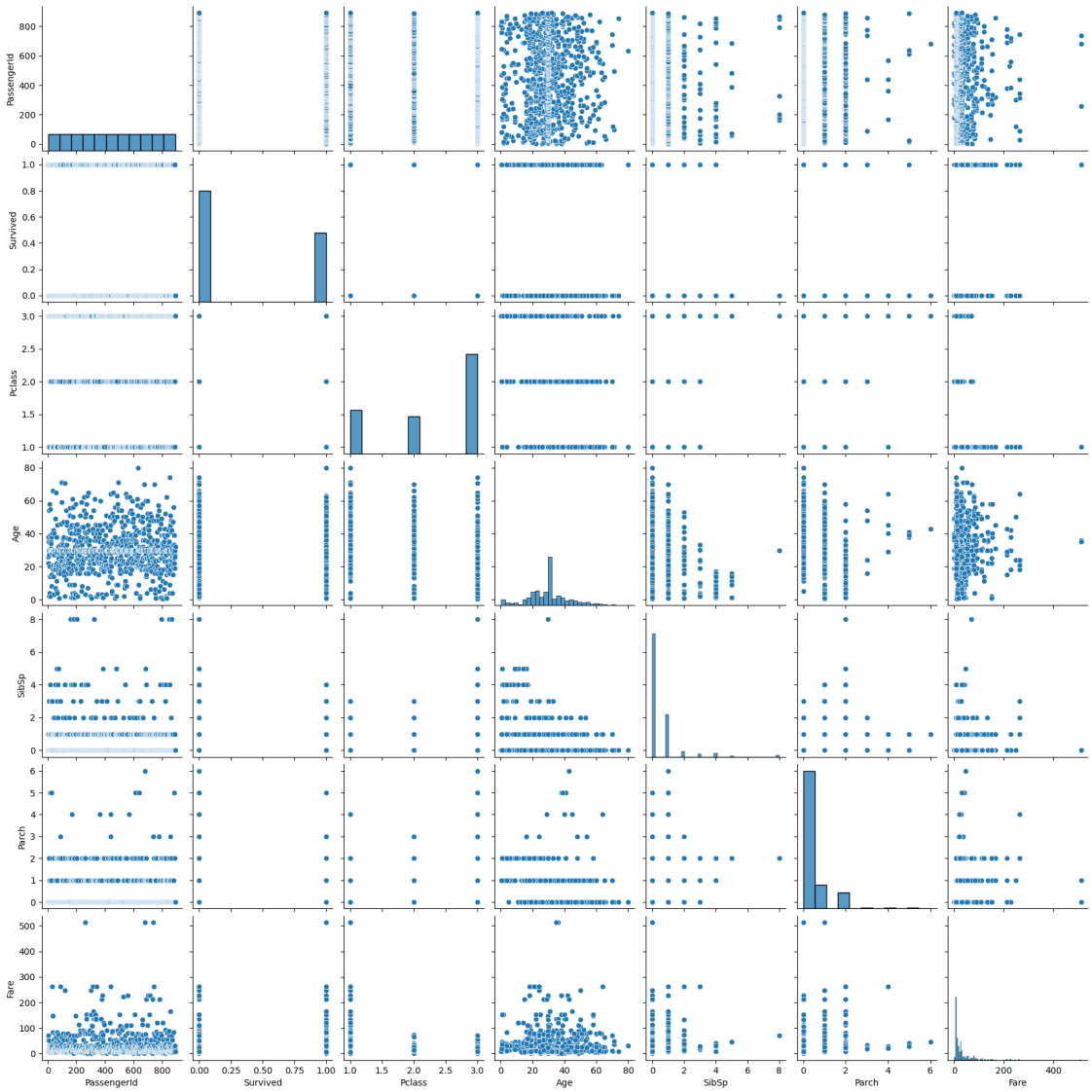
[37]:
```
data.isnull().any()
```

[37]:
```
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
```

[38]:
```
sns.pairplot(data)
```

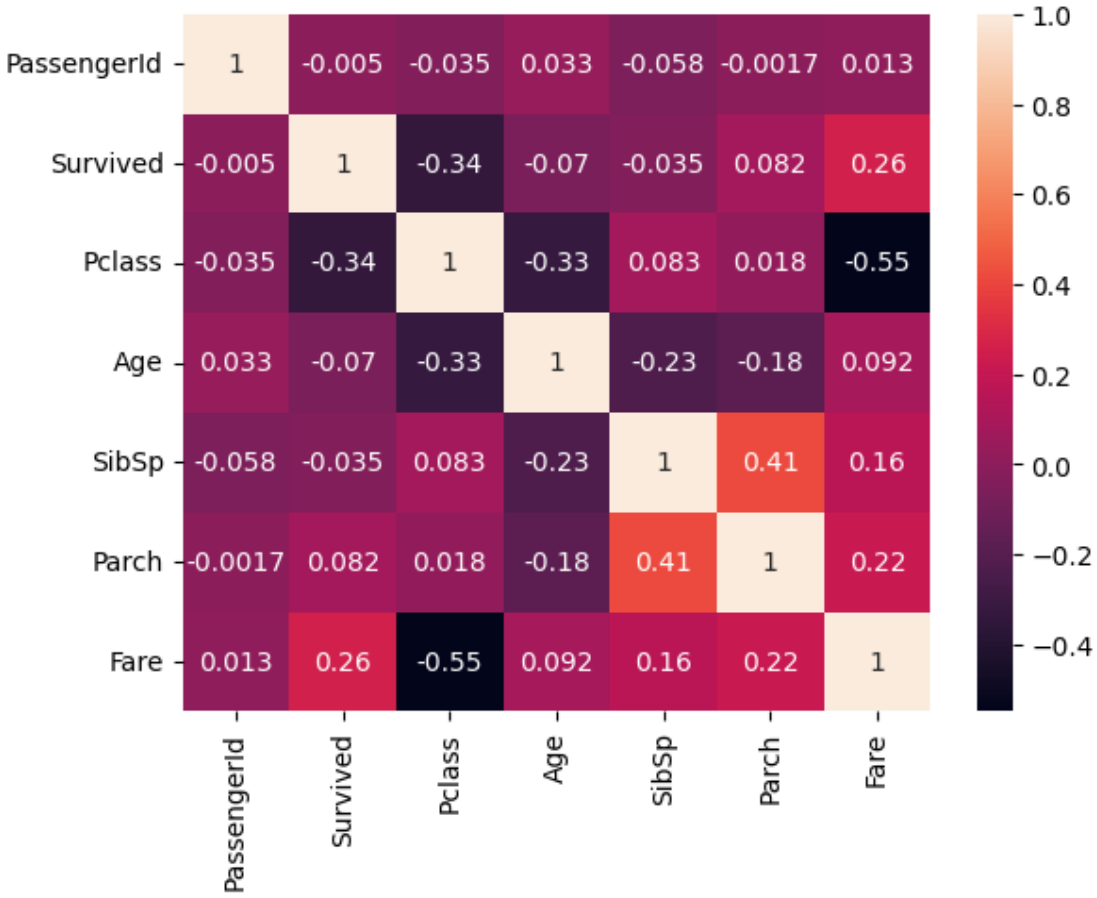
[38]:
```
<seaborn.axisgrid.PairGrid at 0x78c1fec29420>
```



[39]:
```
sns.heatmap(data.corr(),annot=True)
```

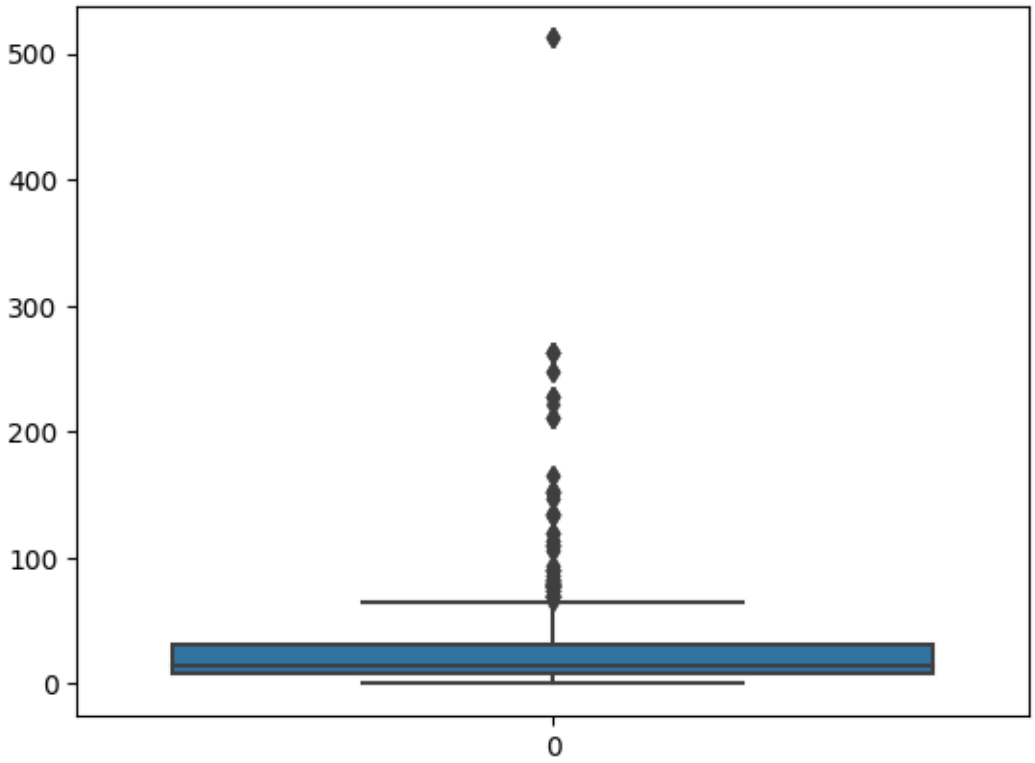
```
<ipython-input-39-6c71ac866e2e>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(data.corr(),annot=True)
```

[39]: <Axes: >



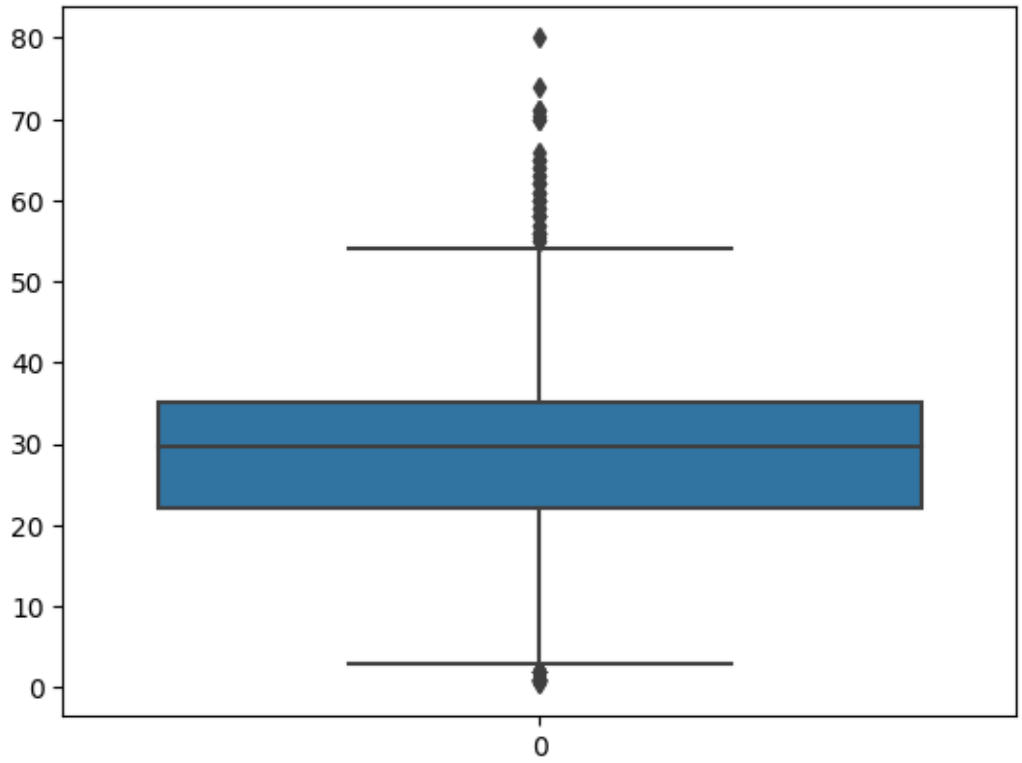
[40]:
```
sns.boxplot(data['Fare'])
```

[40]: <Axes: >



[41]:
```
sns.boxplot(data['Age'])
```

[41]: <Axes: >



[42]:
```
q1=data['Fare'].quantile(0.25)
q3=data['Fare'].quantile(0.75)
```

[43]:
```
print(q1)
print(q3)
```

```
7.9104
31.0
```

[44]:
```
IQR=q3-q1
```

[45]:
```
IQR
```

[45]: 23.0896

[46]:
```
upper_limit=q3+1.5*IQR
```

[47]:
```
upper_limit
```

[47]: 65.6344

[48]:
```
lower_limit=q1-1.5*IQR
```

[49]:
```
lower_limit
```

[49]:
```
-26.724
```

[50]:
```
data.median()
```

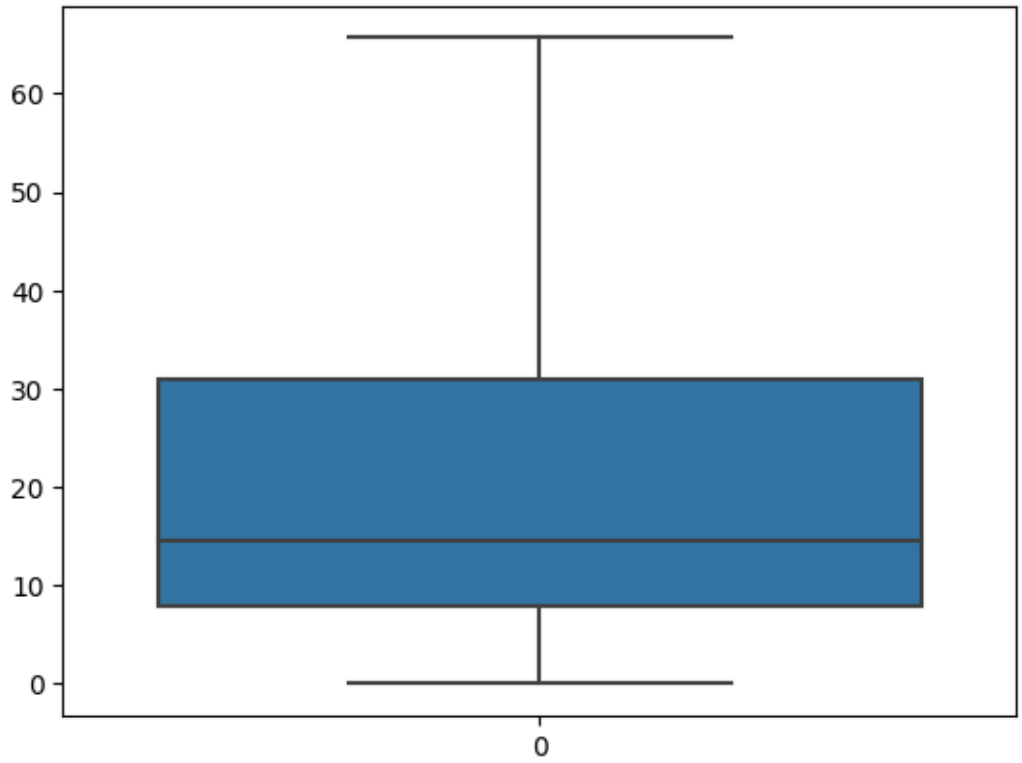
```
<ipython-input-50-135339ac59ce>:1: FutureWarning: The default value of
numeric_only in DataFrame.median is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  data.median()
```

[50]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

[51]:
```
data['Fare']=np.where(data['Fare']>upper_limit,upper_limit,np.
  ↪where(data['Fare']<lower_limit,lower_limit,data['Fare']))
```

[52]:
```
sns.boxplot(data['Fare'])
```

[52]:
```
<Axes: >
```



[53]:
```
q1=data['Fare'].quantile(0.25)
q3=data['Fare'].quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
31.0
```

[54]:
```
IQR=q3-q1
IQR
```

[54]: 23.0896

[55]:
```
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

[56]:
```
print(upper_limit)
print(lower_limit)
```

```
65.6344
-26.724
```

[57]:
```
data.median()
```

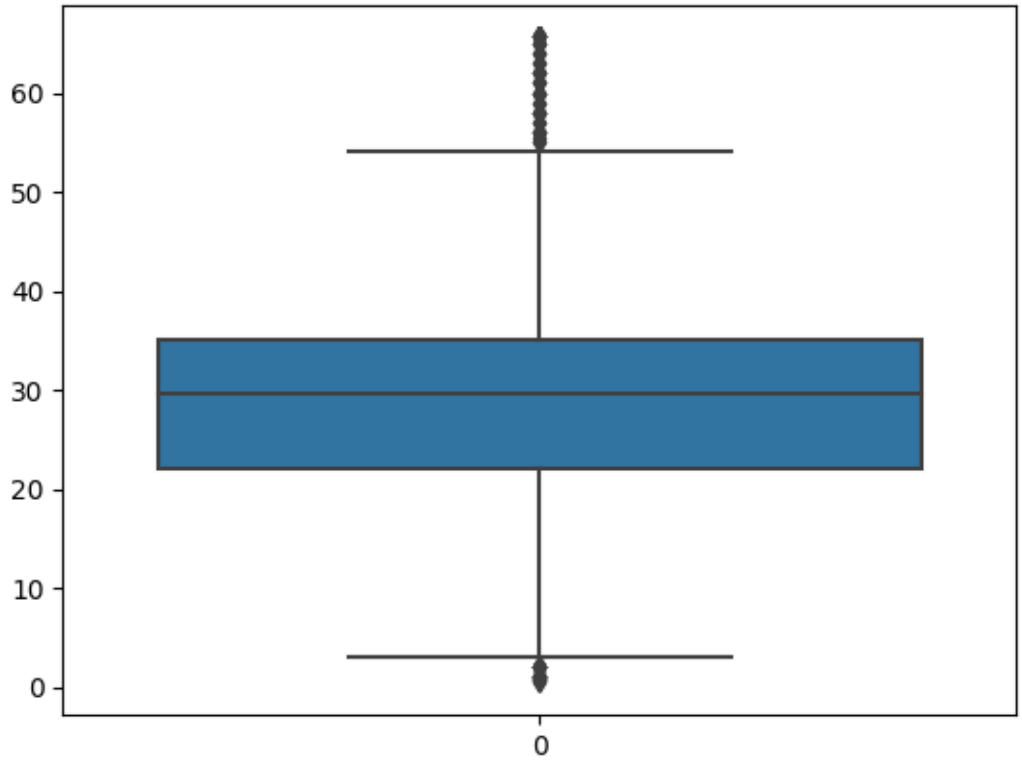
```
<ipython-input-57-135339ac59ce>:1: FutureWarning: The default value of
numeric_only in DataFrame.median is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  data.median()
```

[57]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

[58]:
```
data['Age']=np.where(data['Age']>upper_limit,upper_limit,np.
↪where(data['Age']<lower_limit,lower_limit,data['Age']))
```

[59]:
```
sns.boxplot(data['Age'])
```

[59]:
```
<Axes: >
```



[60]:
```
x = data.drop(columns=['Survived'],axis=1)
```

[61]:
```
x.head()
```

[61]:
```
   PassengerId  Pclass                                                Name  \
0            1       3                             Braund, Mr. Owen Harris
1            2       1  Cumings, Mrs. John Bradley (Florence Briggs Th…
2            3       3                              Heikkinen, Miss. Laina
3            4       1        Futrelle, Mrs. Jacques Heath (Lily May Peel)
4            5       3                            Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare    Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500  UnKnown        S
1  female  38.0      1      0          PC 17599  65.6344      C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250  UnKnown        S
3  female  35.0      1      0            113803  53.1000     C123        S
4    male  35.0      0      0            373450   8.0500  UnKnown        S
```

[62]:
```
y=data.Survived
y.head()
```

[62]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

[63]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

[64]:
```
x.Ticket=le.fit_transform(x.Ticket)
x.Cabin=le.fit_transform(x.Cabin)
x.Name=le.fit_transform(x.Name)
x.Sex=le.fit_transform(x.Sex)
x.Embarked=le.fit_transform(x.Embarked)
```

[65]:
```
x.head()
```

[65]:
```
   PassengerId  Pclass  Name  Sex   Age  SibSp  Parch  Ticket     Fare  Cabin  \
0            1       3   108    1  22.0      1      0     523   7.2500    147
1            2       1   190    0  38.0      1      0     596  65.6344     81
2            3       3   353    0  26.0      0      0     669   7.9250    147
3            4       1   272    0  35.0      1      0      49  53.1000     55
4            5       3    15    1  35.0      0      0     472   8.0500    147
```

```
   Embarked
0         2
1         0
2         2
3         2
4         2
```

[66]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

[67]:
```
x_scaled
```

[67]:
```
     PassengerId  Pclass      Name  Sex       Age  SibSp     Parch    Ticket  \
0       0.000000     1.0  0.121348  1.0  0.330909  0.125  0.000000  0.769118
1       0.001124     0.0  0.213483  0.0  0.576253  0.125  0.000000  0.876471
2       0.002247     1.0  0.396629  0.0  0.392245  0.000  0.000000  0.983824
3       0.003371     0.0  0.305618  0.0  0.530251  0.125  0.000000  0.072059
4       0.004494     1.0  0.016854  1.0  0.530251  0.000  0.000000  0.694118
..           …     …        …   …       …    …       …        …
886     0.995506     0.5  0.615730  1.0  0.407579  0.000  0.000000  0.148529
887     0.996629     0.0  0.340449  0.0  0.284906  0.000  0.000000  0.020588
888     0.997753     1.0  0.464045  0.0  0.448967  0.125  0.333333  0.992647
889     0.998876     0.0  0.091011  1.0  0.392245  0.000  0.000000  0.011765
890     1.000000     1.0  0.247191  1.0  0.484249  0.000  0.000000  0.685294

         Fare     Cabin  Embarked
0    0.110460  1.000000  0.666667
1    1.000000  0.551020  0.000000
2    0.120745  1.000000  0.666667
3    0.809027  0.374150  0.666667
4    0.122649  1.000000  0.666667
..        …       …        …
886  0.198067  1.000000  0.666667
887  0.457077  0.204082  0.666667
888  0.357282  1.000000  0.666667
889  0.457077  0.408163  0.000000
890  0.118078  1.000000  0.333333

[891 rows x 11 columns]
```

[68]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.
↪2,random_state=0)
```

[69]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[69]:
```
((712, 11), (179, 11), (712,), (179,))
```

[70]:
```
x_train.head()
```

[70]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
140	0.157303	1.0	0.111236	0.0	0.448967	0.000	0.333333	0.298529	0.232284	1.0	0.000000
439	0.493258	0.5	0.502247	1.0	0.468915	0.000	0.000000	0.804412	0.159977	1.0	0.666667
817	0.917978	0.5	0.566292	1.0	0.468915	0.125	0.166667	0.908824	0.563793	1.0	0.000000
378	0.424719	1.0	0.095506	1.0	0.300240	0.000	0.000000	0.269118	0.061134	1.0	0.000000
491	0.551685	1.0	0.978652	1.0	0.315574	0.000	0.000000	0.954412	0.110460	1.0	0.666667

[70]:
```

```