1.import the necessary libraries

In [50]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2.import the dataset

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

Out[2]:

	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

In [3]:
```
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

In [4]:
```
df.tail()
```

Out[4]:

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

In [5]:
```
df.shape
```

Out[5]:
```
(891, 12)
```

In [6]:
```
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 [7]:
```
df.describe()
```

Out[7]:

	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

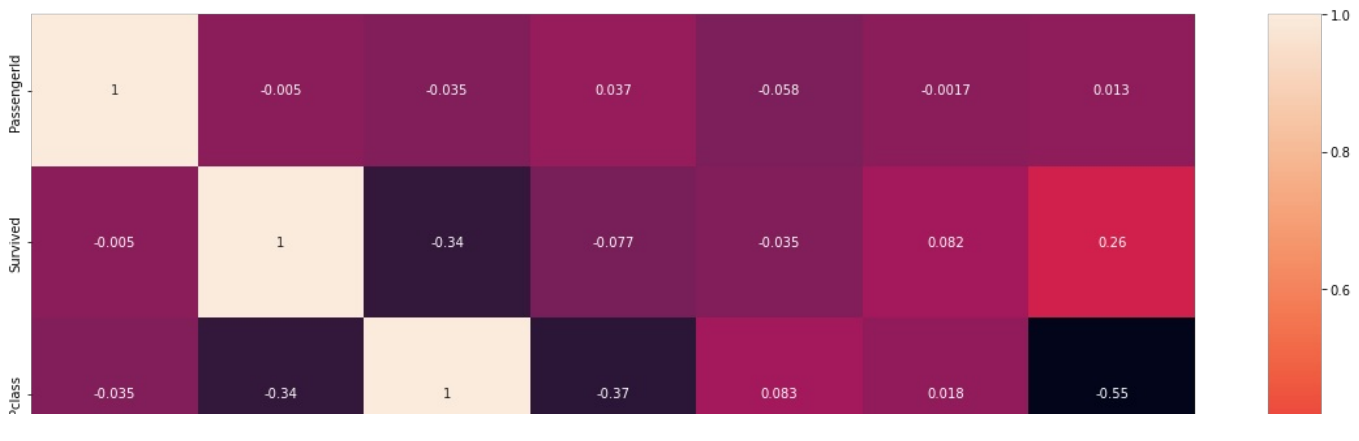
In [8]:
```
corr=df.corr()
corr
```

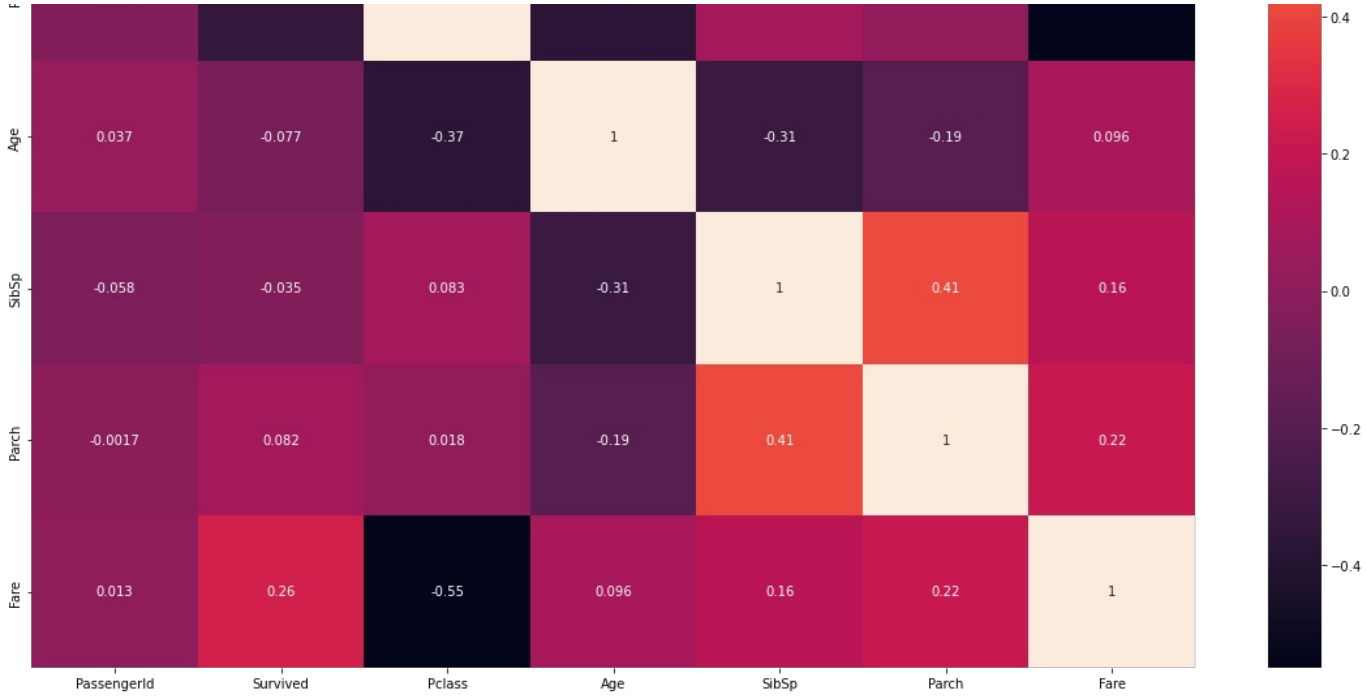
Out[8]:

	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 [9]:
```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

Out[9]:
```
<AxesSubplot:>
```





3.Handling null values

In [10]:
```
df.isnull().any()
```

Out[10]:
```
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 [11]:
```
df.isnull().sum()
```

Out[11]:
```
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 [12]:
```
df["Age"].fillna(df["Age"].mean(),inplace=True)
```

In [13]:
```
df["Cabin"].fillna(df["Cabin"].mode()[0],inplace=True)
```

In [14]:
```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

In [15]:
```
df.head()
```

Out[15]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
											B96	

0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	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 [16]:
```
df
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	B96 B98	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	B96 B98	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	B96 B98	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	B96 B98	Q

891 rows × 12 columns

4.Outliers

In [17]:
```
df
```

Out[17]:

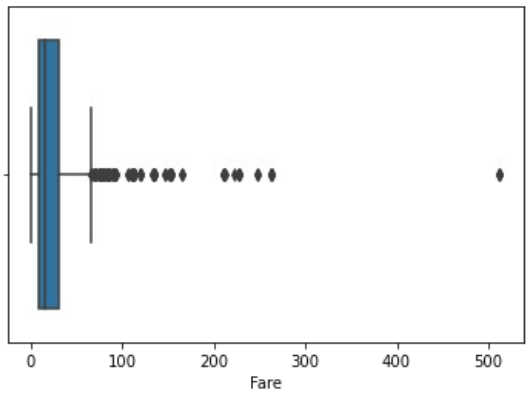
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	B96 B98	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	B96 B98	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	B96 B98	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	B96 B98	Q

891 rows × 12 columns

In [18]:
```
sns.boxplot(df["Fare"])
```

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

Out[18]:
```
<AxesSubplot:xlabel='Fare'>
```



In [20]:
```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
31.0
```

In [21]:
```
iqr=q3-q1
iqr
```

Out[21]:
```
23.0896
```

In [22]:
```
upperl=q3+1.5*iqr
upperl
```

Out[22]:
```
65.6344
```

In [23]:
```
lowerl=q1-1.5*iqr
lowerl
```

Out[23]:
```
-26.724
```

In [24]:
```
df.median()
```

```
C:\Users\DHAIRY~1\AppData\Local\Temp/ipykernel_4572/530051474.py:1: FutureWarning: Dropping of nuisance columns i
n DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
  df.median()
```

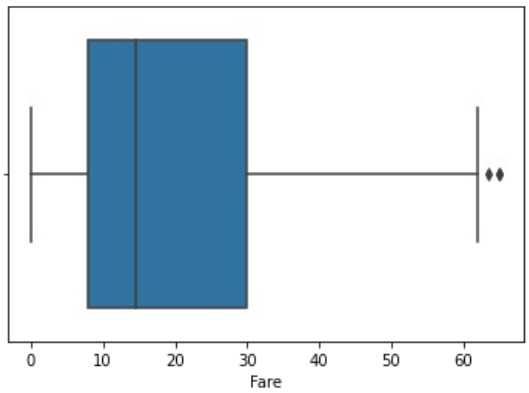
Out[24]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

In [25]:
```
df['Fare']=np.where(df['Fare']>upperl,30,df['Fare'])
```

In [26]:
```
sns.boxplot(df["Fare"])
```

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

Out[26]:
```
<AxesSubplot:xlabel='Fare'>
```



5.Seperate dependent and independent variables# 5.Seperate dependent and independent variables

In [27]:
```
df.head()
```

Out[27]:

	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	7.250	B96 B98	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	30.000	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	B96 B98	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050	B96 B98	S

In [28]:
```
X=df.drop(columns=["Fare"],axis=1)
X.head()
```

Out[28]:

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

In [29]:
```
X.shape
```

Out[29]:
```
(891, 11)
```

In [30]:
```
type(X)
```

Out[30]:
```
pandas.core.frame.DataFrame
```

In [31]:
```
Y=df["Fare"]
Y.head()
```

Out[31]:
```
0     7.250
1    30.000
2     7.925
3    53.100
4     8.050
Name: Fare, dtype: float64
```

In [32]:
```
type(Y)
```

Out[32]:
```
pandas.core.series.Series
```

6.Encoding

In [33]:
```
from sklearn.preprocessing import LabelEncoder
```

In [34]:
```
le=LabelEncoder()
```

In [35]:
```
X["Sex"]=le.fit_transform(X["Sex"])
```

In [36]:
```
X["Sex"]
```

Out[36]:
```
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 [37]:
```
X["Sex"].value_counts()
```

Out[37]:
```
1    577
0    314
Name: Sex, dtype: int64
```

In [38]:
```
X["Sex"].nunique()
```

Out[38]:
```
2
```

In [39]:
```
X.head()
```

Out[39]:

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

In [40]:
```
X["Pclass"].nunique()
```

Out[40]:
```
3
```

7.splitting into training and testing set

In [41]:
```
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)
```

In [42]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[42]:
```
((623, 11), (268, 11), (623,), (268,))
```

In [43]:
```
x_train
```

Out[43]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
857	858	1	1	Daly, Mr. Peter Denis	1	51.000000	0	0	113055	E17	S
52	53	1	1	Harper, Mrs. Henry Sleeper (Myna Haxtun)	0	49.000000	1	0	PC 17572	D33	C
386	387	0	3	Goodwin, Master. Sidney Leonard	1	1.000000	5	2	CA 2144	B96 B98	S
124	125	0	1	White, Mr. Percival Wayland	1	54.000000	0	1	35281	D26	S
578	579	0	3	Caram, Mrs. Joseph (Maria Elias)	0	29.699118	1	0	2689	B96 B98	C
...	...	...	...	...	...	...	...	...	...	...	...
835	836	1	1	Compton, Miss. Sara Rebecca	0	39.000000	1	1	PC 17756	E49	C
192	193	1	3	Andersen-Jensen, Miss. Carla Christine Nielsine	0	19.000000	1	0	350046	B96 B98	S
629	630	0	3	O'Connell, Mr. Patrick D	1	29.699118	0	0	334912	B96 B98	Q
559	560	1	3	de Messemaeker, Mrs. Guillaume Joseph (Emma)	0	36.000000	1	0	345572	B96 B98	S
684	685	0	2	Brown, Mr. Thomas William Solomon	1	60.000000	1	1	29750	B96 B98	S

623 rows × 11 columns

In [44]:
```
y_train
```

Out[44]:
```
857    26.5500
52     30.0000
386    46.9000
124    30.0000
578    14.4583
        ...
835    30.0000
192     7.8542
629     7.7333
559    17.4000
684    39.0000
Name: Fare, Length: 623, dtype: float64
```

In [45]:
```
x_test
```

Out[45]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Cabin	Embarked
495	496	0	3	Yousseff, Mr. Gerious	1	29.699118	0	0	2627	B96 B98	C
648	649	0	3	Willey, Mr. Edward	1	29.699118	0	0	S.O./P.P. 751	B96 B98	S
278	279	0	3	Rice, Master. Eric	1	7.000000	4	1	382652	B96 B98	Q

31	32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)	0	29.699118	1	0	PC 17569	B78	C
255	256	1	3	Touma, Mrs. Darwis (Hanne Youssef Razi)	0	29.000000	0	2	2650	B96 B98	C
...	...	...	...	...	...	...	...	...	...	...	...
263	264	0	1	Harrison, Mr. William	1	40.000000	0	0	112059	B94	S
718	719	0	3	McEvoy, Mr. Michael	1	29.699118	0	0	36568	B96 B98	Q
620	621	0	3	Yasbeck, Mr. Antoni	1	27.000000	1	0	2659	B96 B98	C
786	787	1	3	Sjoblom, Miss. Anna Sofia	0	18.000000	0	0	3101265	B96 B98	S
64	65	0	1	Stewart, Mr. Albert A	1	29.699118	0	0	PC 17605	B96 B98	C

268 rows × 11 columns

In [46]:
```
y_test
```

Out[46]:
```
495    14.4583
648     7.5500
278    29.1250
31     30.0000
255    15.2458
        ...
263     0.0000
718    15.5000
620    14.4542
786     7.4958
64     27.7208
Name: Fare, Length: 268, dtype: float64
```

8.Feature Scaling

In [47]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

In [51]:
```
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\DHAIRY~1\AppData\Local\Temp/ipykernel_4572/273440110.py in <module>
----> 1 x_train=sc.fit_transform(x_train)
      2 x_test=sc.fit_transform(x_test)

~\anaconda3\lib\site-packages\sklearn\base.py in fit_transform(self, X, y, **fit_params)
    697         if y is None:
    698             # fit method of arity 1 (unsupervised transformation)
--> 699             return self.fit(X, **fit_params).transform(X)
    700         else:
    701             # fit method of arity 2 (supervised transformation)

~\anaconda3\lib\site-packages\sklearn\preprocessing\_data.py in fit(self, X, y, sample_weight)
    728         # Reset internal state before fitting
    729         self._reset()
--> 730         return self.partial_fit(X, y, sample_weight)
    731
    732     def partial_fit(self, X, y=None, sample_weight=None):

~\anaconda3\lib\site-packages\sklearn\preprocessing\_data.py in partial_fit(self, X, y, sample_weight)
    764         """
    765         first_call = not hasattr(self, "n_samples_seen_")
--> 766         X = self._validate_data(X, accept_sparse=('csr', 'csc'),
    767                                 estimator=self, dtype=FLOAT_DTYPES,
    768                                 force_all_finite='allow-nan', reset=first_call)

~\anaconda3\lib\site-packages\sklearn\base.py in _validate_data(self, X, y, reset, validate_separately, **check_p
arams)
    419             out = X
    420         elif isinstance(y, str) and y == 'no_validation':
--> 421             X = check_array(X, **check_params)
    422             out = X
    423         else:
```

```
~\anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in check_array(array, accept_sparse, accept_large_spars
e, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    671                     array = array.astype(dtype, casting="unsafe", copy=False)
    672                 else:
--> 673                     array = np.asarray(array, order=order, dtype=dtype)
    674             except ComplexWarning as complex_warning:
    675                 raise ValueError("Complex data not supported\n"

~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order, like)
    100         return _asarray_with_like(a, dtype=dtype, order=order, like=like)
    101
--> 102     return array(a, dtype, copy=False, order=order)
    103
    104


~\anaconda3\lib\site-packages\pandas\core\generic.py in __array__(self, dtype)
   1991
   1992     def __array__(self, dtype: NpDtype | None = None) -> np.ndarray:
-> 1993         return np.asarray(self._values, dtype=dtype)
   1994
   1995     def __array_wrap__(

~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order, like)
    100         return _asarray_with_like(a, dtype=dtype, order=order, like=like)
    101
--> 102     return array(a, dtype, copy=False, order=order)
    103
    104


ValueError: could not convert string to float: 'Daly, Mr. Peter Denis '
```

In [ ]: