Data Preparation and ensuring the shape

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [2]:
```
## df=pd.read_csv("titanic.csv")
df=sns.load_dataset("titanic")
df.head()
```

Out[2]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_ma
0	0	3	male	22.0	1	0	7.2500	S	Third	man	Tr
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fal
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	Fal
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fal
4	0	3	male	35.0	0	0	8.0500	S	Third	man	Tr

In [3]:
```
df.describe()
```

Out[3]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype   
---  ------       --------------  -----   
 0   survived     891 non-null    int64   
 1   pclass       891 non-null    int64   
 2   sex          891 non-null    object  
 3   age          714 non-null    float64 
 4   sibsp        891 non-null    int64   
 5   parch        891 non-null    int64   
 6   fare         891 non-null    float64 
 7   embarked     889 non-null    object  
 8   class        891 non-null    category
 9   who          891 non-null    object  
 10  adult_male   891 non-null    bool    
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object  
 13  alive        891 non-null    object  
 14  alone        891 non-null    bool    
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

In [5]:
```
df["TravelAlone"]=np.where(df["sibsp"]+df["parch"] > 0,0,1).astype("uint8")
```

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

Out[6]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_ma
0	0	3	male	22.0	1	0	7.2500	S	Third	man	Tr
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fal
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	Fal
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fal
4	0	3	male	35.0	0	0	8.0500	S	Third	man	Tr

In [7]:
```
df1=df.drop(["sibsp","parch","deck","embark_town","alive","alone","who","a
```

In [8]:
```
df.isna().sum()
```

Out[8]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
TravelAlone      0
dtype: int64
```

In [9]:
```
df1.isna().sum()
```

Out[9]:
```
survived         0
pclass           0
sex              0
age            177
fare             0
embarked         2
TravelAlone      0
dtype: int64
```

In [10]:
```
df["age"].fillna(df1["age"].median(skipna=True),inplace=True)
```

In [11]:
```
df1.head()
```

Out[11]:

	survived	pclass	sex	age	fare	embarked	TravelAlone
0	0	3	male	22.0	7.2500	S	0
1	1	1	female	38.0	71.2833	C	0
2	1	3	female	26.0	7.9250	S	1
3	1	1	female	35.0	53.1000	S	0
4	0	3	male	35.0	8.0500	S	1

In [13]:
```
df_titanic=pd.get_dummies(df1,columns=["pclass","sex","embarked"],drop_firs
df_titanic.head()
```

Out[13]:

	survived	age	fare	TravelAlone	pclass_2	pclass_3	sex_male	embarked_Q	embark
0	0	22.0	7.2500	0	False	True	True	False	
1	1	38.0	71.2833	0	False	False	False	False	
2	1	26.0	7.9250	1	False	True	False	False	
3	1	35.0	53.1000	0	False	False	False	False	
4	0	35.0	8.0500	1	False	True	True	False	

In [14]:
```
X=df_titanic.drop(["survived"],axis=1)
y=df_titanic["survived"]
```

In [15]:
```
from sklearn.preprocessing import MinMaxScaler,StandardScaler
```

In [16]:
```
trans_MM=MinMaxScaler()
trans_SS=StandardScaler()
```

In [17]:
```
df_MM=trans_MM.fit_transform(X)
pd.DataFrame(df_MM)
```

Out[17]:

	0	1	2	3	4	5	6	7
0	0.271174	0.014151	0.0	0.0	1.0	1.0	0.0	1.0
1	0.472229	0.139136	0.0	0.0	0.0	0.0	0.0	0.0
2	0.321438	0.015469	1.0	0.0	1.0	0.0	0.0	1.0
3	0.434531	0.103644	0.0	0.0	0.0	0.0	0.0	1.0
4	0.434531	0.015713	1.0	0.0	1.0	1.0	0.0	1.0
...	...	...	...	...	...	...	...	...
886	0.334004	0.025374	1.0	1.0	0.0	1.0	0.0	1.0
887	0.233476	0.058556	1.0	0.0	0.0	0.0	0.0	1.0
888	NaN	0.045771	0.0	0.0	1.0	0.0	0.0	1.0
889	0.321438	0.058556	1.0	0.0	0.0	1.0	0.0	0.0
890	0.396833	0.015127	1.0	0.0	1.0	1.0	1.0	0.0

891 rows × 8 columns

In [18]:
```
df_SS=trans_SS.fit_transform(X)
pd.DataFrame(df_SS)
```

Out[18]:

	0	1	2	3	4	5	6	7
0	-0.530377	-0.502445	-1.231645	-0.510152	0.902587	0.737695	-0.307562	0.619306
1	0.571831	0.786845	-1.231645	-0.510152	-1.107926	-1.355574	-0.307562	-1.614710
2	-0.254825	-0.488854	0.811922	-0.510152	0.902587	-1.355574	-0.307562	0.619306
3	0.365167	0.420730	-1.231645	-0.510152	-1.107926	-1.355574	-0.307562	0.619306
4	0.365167	-0.486337	0.811922	-0.510152	0.902587	0.737695	-0.307562	0.619306
...	...	...	...	...	...	...	...	...
886	-0.185937	-0.386671	0.811922	1.960202	-1.107926	0.737695	-0.307562	0.619306
887	-0.737041	-0.044381	0.811922	-0.510152	-1.107926	-1.355574	-0.307562	0.619306
888	NaN	-0.176263	-1.231645	-0.510152	0.902587	-1.355574	-0.307562	0.619306
889	-0.254825	-0.044381	0.811922	-0.510152	-1.107926	0.737695	-0.307562	-1.614710
890	0.158503	-0.492378	0.811922	-0.510152	0.902587	0.737695	3.251373	-1.614710

891 rows × 8 columns

In [ ]:
```
```