In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

Uploading the data into df

In [2]:

```
df=pd.read_csv("tested.csv")
```

In [3]:

```
df
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	

418 rows × 12 columns

Get the information about the dataset using info function

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

Identify the missing values

In [5]:

```
for i in df:
    print(df[i].value_counts())
```

```
892     1
1205    1
1177    1
1176    1
1175    1
       ..
1028    1
1027    1
1026    1
1025    1
1309    1
Name: PassengerId, Length: 418, dtype: int64
0    266
1    152
Name: Survived, dtype: int64
3    218
1    107
2     93
Name: Pclass, dtype: int64
Kelly, Mr. James               1
Carr, Miss. Jeannie            1
Dennis, Mr. William            1
Rosblom, Miss. Salli Helena    1
Touma, Miss. Maria Youssef     1
                              ..
Zakarian, Mr. Mapriededer      1
Carlsson, Mr. Carl Robert      1
Dintcheff, Mr. Valtcho         1
Thomas, Mr. Charles P          1
Peter, Master. Michael J       1
Name: Name, Length: 418, dtype: int64
```

```
male      266
female    152
Name: Sex, dtype: int64
21.0    17
24.0    17
22.0    16
30.0    15
18.0    13
        ..
76.0     1
28.5     1
22.5     1
62.0     1
38.5     1
Name: Age, Length: 79, dtype: int64
0    283
1    110
2     14
3      4
4      4
8      2
5      1
Name: SibSp, dtype: int64
0    324
1     52
2     33
3      3
4      2
9      2
6      1
5      1
Name: Parch, dtype: int64
PC 17608    5
CA. 2343    4
113503      4
PC 17483    3
220845      3
           ..
349226      1
2621        1
4133        1
113780      1
2668        1
Name: Ticket, Length: 363, dtype: int64
7.7500     21
26.0000    19
13.0000    17
8.0500     17
7.8958     11
           ..
7.8208      1
8.5167      1
78.8500     1
52.0000     1
22.3583     1
Name: Fare, Length: 169, dtype: int64
B57 B59 B63 B66    3
B45                2
C89                2
C55 C57            2
A34                2
                  ..
E52                1
D30                1
E31                1
C62 C64            1
```

```
C105                    1
Name: Cabin, Length: 76, dtype: int64
S    270
C    102
Q     46
Name: Embarked, dtype: int64
```

Handling the missing values

In [6]:

```
from sklearn.impute import SimpleImputer
```

In [7]:

```
si=SimpleImputer(missing_values=np.nan,strategy="most_frequent")
df["Age"]=si.fit_transform(df[["Age"]])
```

In [8]:

```
df
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	
413	1305	0	3	Spector, Mr. Woolf	male	21.0	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	0	3	Ware, Mr. Frederick	male	21.0	0	0	359309	8.0500	NaN	
417	1309	0	3	Peter, Master. Michael J	male	21.0	1	1	2668	22.3583	NaN	

418 rows × 12 columns

Dropping the unwanted columns from a dataset

In [9]:
```
df.drop(['PassengerId','Name','Ticket','Cabin','Fare','SibSp','Parch'],axis=1,inplace=Tr
```

In [10]:
```
df
```

Out[10]:

	Survived	Pclass	Sex	Age	Embarked
0	0	3	male	34.5	Q
1	1	3	female	47.0	S
2	0	2	male	62.0	Q
3	0	3	male	27.0	S
4	1	3	female	22.0	S
...	...	...	...	...	...
413	0	3	male	21.0	S
414	1	1	female	39.0	C
415	0	3	male	38.5	S
416	0	3	male	21.0	S
417	0	3	male	21.0	C

418 rows × 5 columns

Separate the feature and target columns

In [11]:
```
target=df["Survived"]
```

In [12]:
```
features=df.iloc[:,1:]
```

In [13]:
```
features
```

Out[13]:

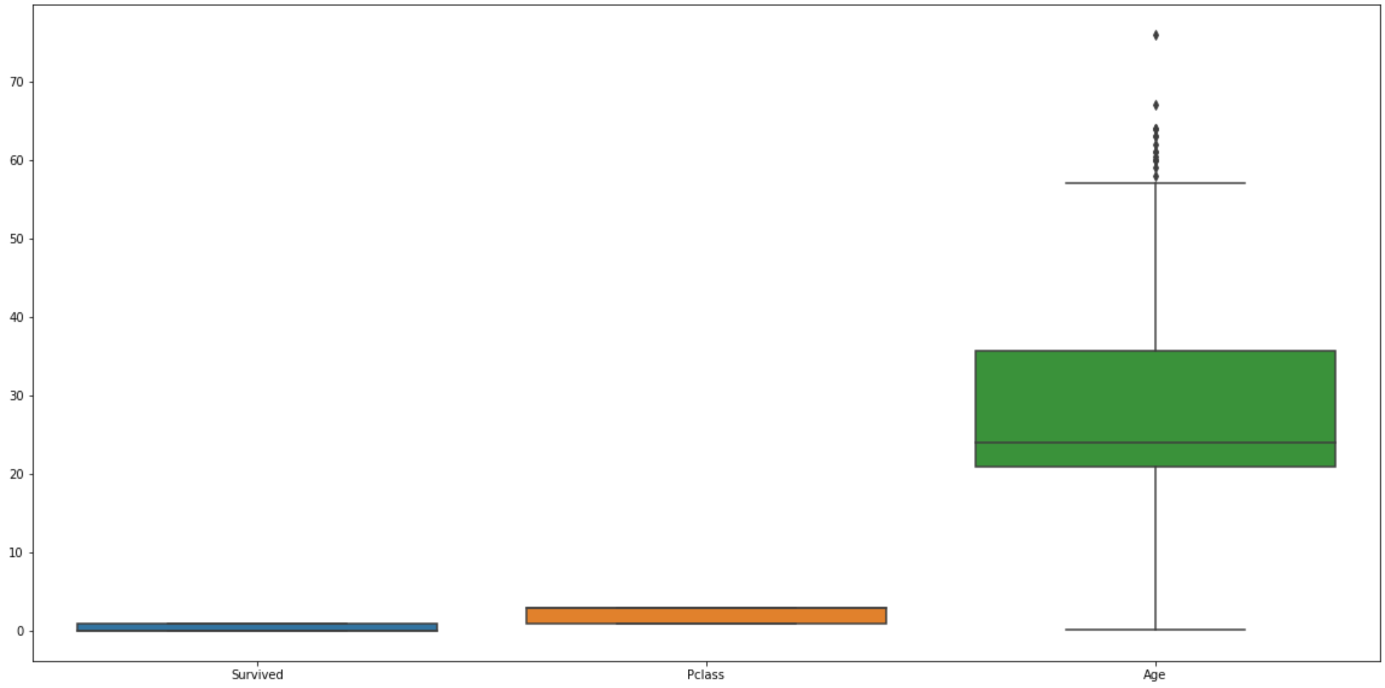
	Pclass	Sex	Age	Embarked
0	3	male	34.5	Q
1	3	female	47.0	S
2	2	male	62.0	Q
3	3	male	27.0	S
4	3	female	22.0	S
...	...	...	...	...
413	3	male	21.0	S
414	1	female	39.0	C
415	3	male	38.5	S
416	3	male	21.0	S
417	3	male	21.0	C

418 rows × 4 columns

Finding the outliers

In [14]:

```
plt.figure(figsize=(20,10))
sns.boxplot(data=df);
```



Finding the correlation between the variables in the dataset

In [15]:

```
pd.concat([features,target],axis=1).corr().style.background_gradient()
```

Out[15]:

	Pclass	Age	Survived
Pclass	1.000000	-0.503026	-0.108615
Age	-0.503026	1.000000	0.021962
Survived	-0.108615	0.021962	1.000000

In [16]:

```
print('skewness value of Age: ',df['Age'].skew())
```

```
skewness value of Age:  0.8188583248551901
```

In [17]:

```
df.describe()
```

Out[17]:

	Survived	Pclass	Age
count	418.000000	418.000000	418.000000
mean	0.363636	2.265550	28.364833
std	0.481622	0.841838	13.180116
min	0.000000	1.000000	0.170000

25%	0.000000	1.000000	21.000000
50%	0.000000	3.000000	24.000000
75%	1.000000	3.000000	35.750000
max	1.000000	3.000000	76.000000

Handling the outliers

In [18]:

```
q1=np.quantile(df["Age"],0.25)
q3=np.quantile(df["Age"],0.75)
iqr=q3-q1
uw=q3+1.5*iqr
lw=q1-1.5*iqr
print(q1,q3)
print(uw,lw)
index=df['Age'][(df['Age']>uw)|(df['Age']<lw)].index
#print(index)
#df.drop(index,inplace=True)
```

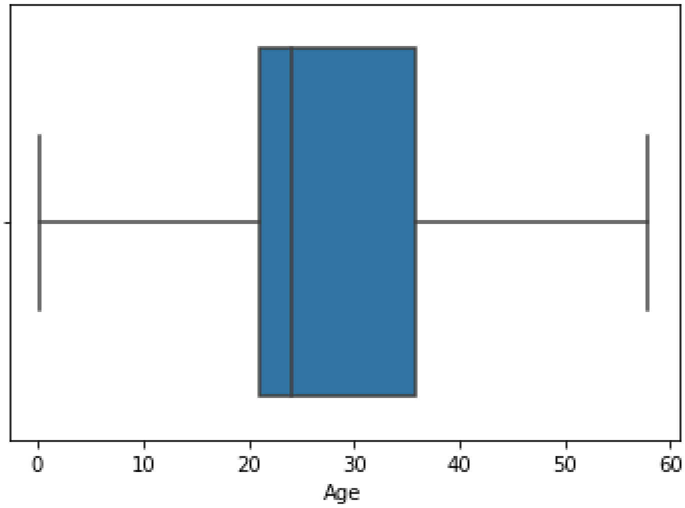
```
21.0 35.75
57.875 -1.125
```

In [19]:

```
for i in df["Age"]:
    #print(i)
    if i>uw:
        df["Age"]=df["Age"].replace(i,uw)
    elif i<lw:
        df["Age"]=df["Age"].replace(i,lw)
```

In [20]:

```
sns.boxplot(df['Age'],data=df);
```



In [21]:

```
print('skewness value of Age: ',df['Age'].skew())
```

```
skewness value of Age:  0.6480019342560972
```

Encoding the categorical data

In [22]:

```
from sklearn.preprocessing import OrdinalEncoder
```

In [23]:

```
ordinal=OrdinalEncoder()
df["Sex"]=ordinal.fit_transform(df[["Sex"]])
```

In [24]:

```
df
```

Out[24]:

	Survived	Pclass	Sex	Age	Embarked
0	0	3	1.0	34.500	Q
1	1	3	0.0	47.000	S
2	0	2	1.0	57.875	Q
3	0	3	1.0	27.000	S
4	1	3	0.0	22.000	S
...	...	...	...	...	...
413	0	3	1.0	21.000	S
414	1	1	0.0	39.000	C
415	0	3	1.0	38.500	S
416	0	3	1.0	21.000	S
417	0	3	1.0	21.000	C

418 rows × 5 columns

Scaling

In [30]:

```
from sklearn.preprocessing import StandardScaler
s=StandardScaler()
pd.DataFrame(s.fit_transform(df.iloc[:,:-1]))
```

Out[30]:

	0	1	2	3
0	-0.755929	0.873482	0.755929	0.498027
1	1.322876	0.873482	-1.322876	1.483332
2	-0.755929	-0.315819	0.755929	2.340548
3	-0.755929	0.873482	0.755929	-0.093156
4	1.322876	0.873482	-1.322876	-0.487278
...	...	...	...	...
413	-0.755929	0.873482	0.755929	-0.566103
414	1.322876	-1.505120	-1.322876	0.852737
415	-0.755929	0.873482	0.755929	0.813325
416	-0.755929	0.873482	0.755929	-0.566103
417	-0.755929	0.873482	0.755929	-0.566103

418 rows × 4 columns

values between -1 to 1

In [ ]: