15TH_SEPTEMBER_ASSIGNMENT

NAME:- CH.ABHIRAM

REG NO:- 21BCE8707

Steps:

1.import the necessary libraries

2.import the dataset

3.Handling null values

4.outlier detection---surya

5.Seperate Dependent and independent variables

6.Encoding

7.splitting into training and testing set

8.Feature scaling

1.import the necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

2.import the dataset

```
#.csv  .tsv ,json,.excel
dataset=pd.read_csv("Titanic-Dataset.csv")
#dataset=pd.read_csv(r"D:\SmartBridge\VIT_morning_slot\Churn_Modelling.csv")
```

```
dataset
```

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

```
dataset.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	7

```
dataset.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30

```
dataset.shape
```

```
(891, 12)
```

```
dataset.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
```

```
dataset.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.0
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.2
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.6
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.0
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.9
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.4
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.0
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.3

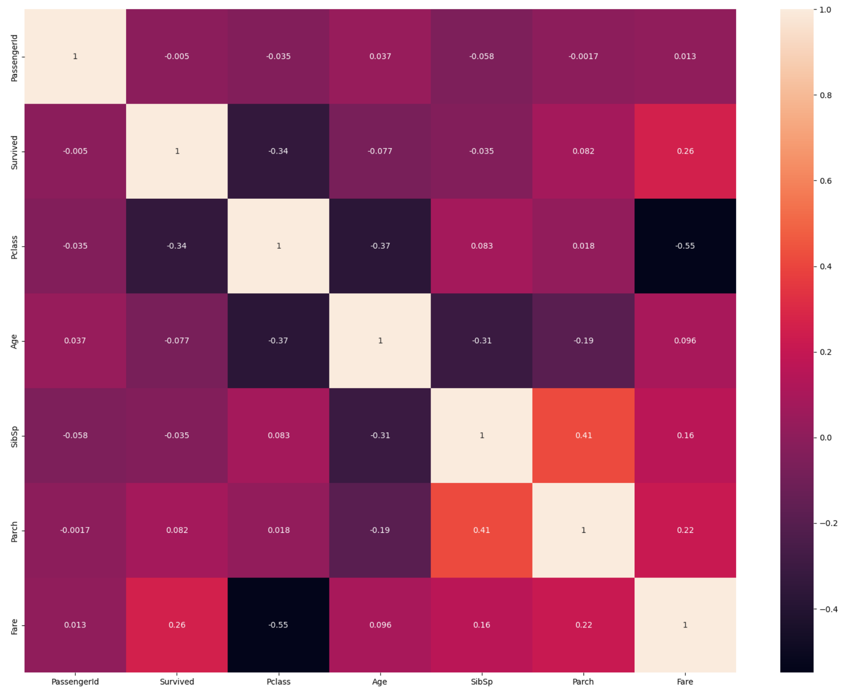
```
corr=dataset.corr()
corr
```

```
<ipython-input-9-f22ca9e9dc13>:1: FutureWarning: The default value of numeric_only
  corr=dataset.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	F
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159

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

```
<Axes: >
```



```
dataset.PassengerId.value_counts()
```

```
1      1
599    1
```

```
588    1
589    1
590    1
      ..
301    1
302    1
303    1
304    1
891    1
Name: PassengerId, Length: 891, dtype: int64
```

```
dataset.Survived.value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
dataset.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599

```
dataset.    Pclass.value_counts()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

3.Handling null values

```
dataset.isnull().any()
```

```
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
```

```
dataset.isnull().sum()
```

```
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
```

```
dataset["Age"].fillna(dataset["Age"].mean(),inplace=True)
```

```
dataset["Cabin"].fillna(dataset["Cabin"].mode()[0],inplace=True)
```

```
dataset["Embarked"].fillna(dataset["Embarked"].mode()[0],inplace=True)
```

```
dataset.isnull().sum()
```

```
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
```

```
corr=dataset.corr()
corr
```

```
<ipython-input-21-f22ca9e9dc13>:1: FutureWarning: The default value of numeric_onl
  corr=dataset.corr()
```

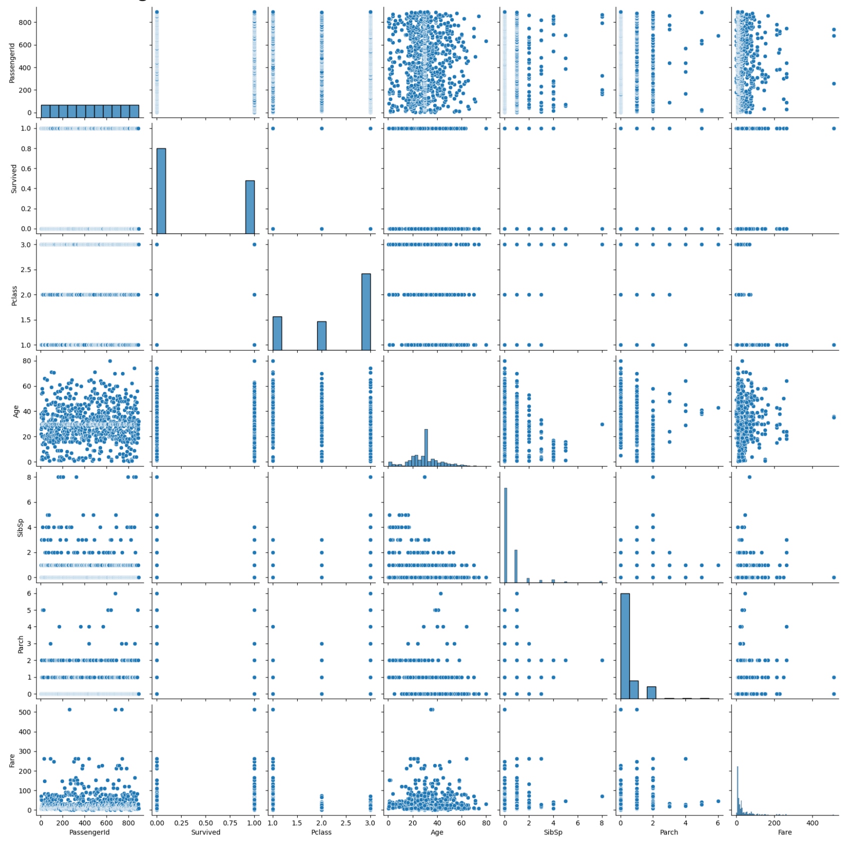
	PassengerId	Survived	Pclass	Age	SibSp	Parch	F
PassengerId	1.000000	-0.005007	-0.035144	0.033207	-0.057527	-0.001652	0.012
Survived	-0.005007	1.000000	-0.338481	-0.069809	-0.035322	0.081629	0.257
Pclass	-0.035144	-0.338481	1.000000	-0.331339	0.083081	0.018443	-0.549
Age	0.033207	-0.069809	-0.331339	1.000000	-0.232625	-0.179191	0.091
SibSp	-0.057527	-0.035322	0.083081	-0.232625	1.000000	0.414838	0.159
Parch	-0.001652	0.081629	0.018443	-0.179191	0.414838	1.000000	0.216
Fare	0.012658	0.257307	-0.549500	0.091566	0.159651	0.216225	1.000

```
dataset.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	7

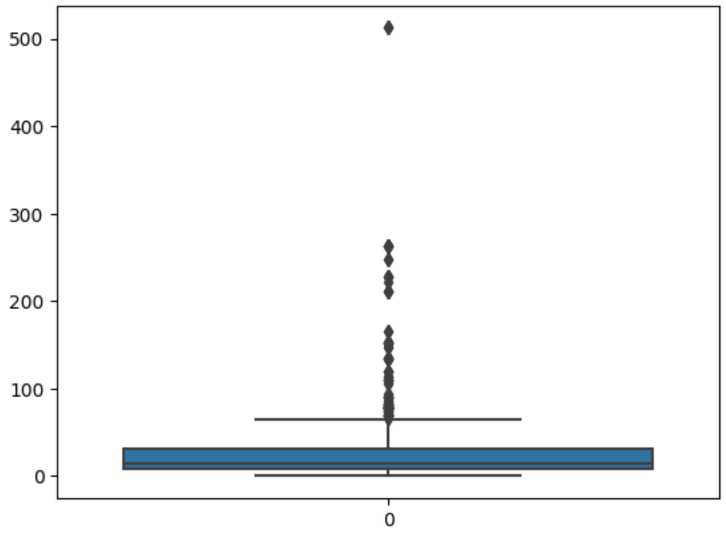
```
sns.pairplot(dataset)
```

```
<seaborn.axisgrid.PairGrid at 0x7b204c200040>
```



```
sns.boxplot(dataset.Fare)
```

```
<Axes: >
```



4.outliers

```
z_scores = np.abs(stats.zscore(dataset['Age']))
max_threshold=3
outliers = dataset['Age'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

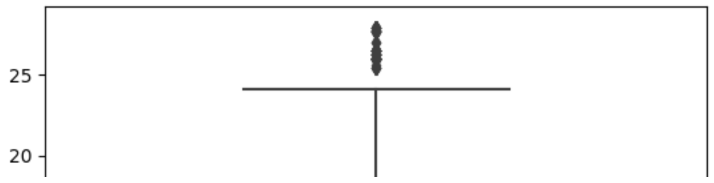
```
Outliers detected using Z-Score:
96     71.0
116    70.5
493    71.0
630    80.0
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

```
z_scores = np.abs(stats.zscore(dataset['Fare']))
max_threshold=3
outliers = dataset['Fare'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
27     263.0000
88     263.0000
118    247.5208
258    512.3292
299    247.5208
311    262.3750
341    263.0000
377    211.5000
380    227.5250
438    263.0000
527    221.7792
557    227.5250
679    512.3292
689    211.3375
700    227.5250
716    227.5250
730    211.3375
737    512.3292
742    262.3750
779    211.3375
Name: Fare, dtype: float64
```

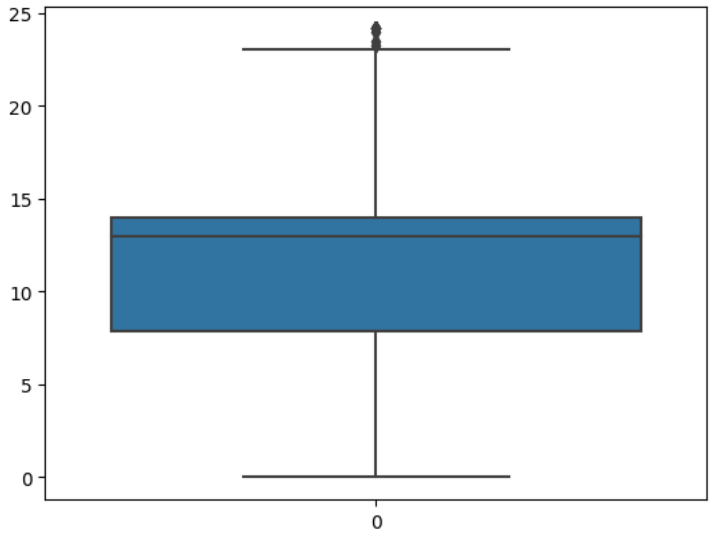
```
q1 = dataset.Fare.quantile(0.25)
q3 = dataset.Fare.quantile(0.75)
print(q1)
print(q3)
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
dataset.median()
dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
sns.boxplot(dataset.Fare)
```

```
7.8958
16.1
<ipython-input-86-20029ddbc2f9>:9: FutureWarning: The default value of numeric_onl
  dataset.median()
<ipython-input-86-20029ddbc2f9>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
<Axes: >
```



```
q1 = dataset.Fare.quantile(0.25)
q3 = dataset.Fare.quantile(0.75)
print(q1)
print(q3)
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
dataset.median()
dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
sns.boxplot(dataset.Fare)
```

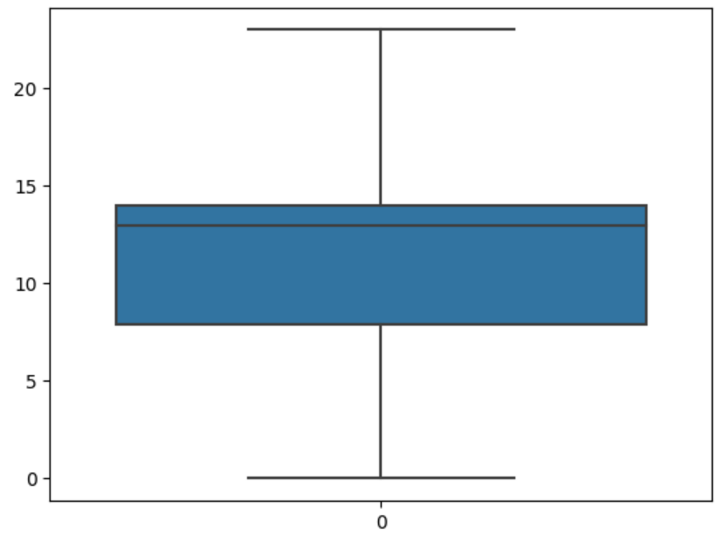
```
7.8958
14.4542
<ipython-input-87-20029ddbc2f9>:9: FutureWarning: The default value of numeric_onl
  dataset.median()
<ipython-input-87-20029ddbc2f9>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
<Axes: >
```



```
q1 = dataset.Fare.quantile(0.25)
q3 = dataset.Fare.quantile(0.75)
print(q1)
print(q3)
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
dataset.median()
dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
```

```
sns.boxplot(dataset.Fare)
```

```
7.8958
14.0
<ipython-input-88-20029ddbc2f9>:9: FutureWarning: The default value of numeric_onl
  dataset.median()
<ipython-input-88-20029ddbc2f9>:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable
  dataset["Fare"]=np.where(dataset["Fare"]>upperlimit,14,dataset['Fare'])
<Axes: >
```



```
dataset=dataset_cleaned
```

```
x=dataset.drop('Survived', axis=1)
y=dataset['Survived']
```

```
x.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.250
2	3	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.925
			Futrelle						

```
y.head()
```

```
0    0
2    1
3    1
4    0
5    0
Name: Survived, dtype: int64
```

5.Seperate dependent and independent variables

```
#datset.iloc[rows,column]
x=dataset.iloc[:,3:13]
y=dataset.iloc[:,13:14]
```

```
x.head()
```

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	B96 B98	S
2	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S
	Futrelle, Mrs								

```
y.head()
```

0
2
3
4
5

```
dataset.shape
```

```
(775, 12)
```

```
x.shape
```

```
(775, 9)
```

```
y.shape
```

```
(775, 0)
```

6.Encoding

Label encoding on Gender column

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
x["Sex"]=le.fit_transform(x["Sex"])
```

```
x["Sex"]
```

```
0      1
2      0
3      0
4      1
5      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 775, dtype: int64
```

```
x["Sex"].value_counts()
```

```
1    531
0    244
Name: Sex, dtype: int64
```

```
x["Sex"].nunique()
```

```
2
```

```
x.head()
```

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	Braund, Mr. Owen Harris	1	22.000000	1	0	A/5 21171	7.2500	B96 B98	S
2	Heikkinen, Miss. Laina	0	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	Futrelle, Mrs. Jacques Heath (Lily	0	35.000000	1	0	113803	53.1000	C123	S

```
x.Sex.value_counts()
```

```
1    531
0    244
Name: Sex, dtype: int64
```

One hot encoding on geography column

```
x.shape
```

```
(775, 9)
```

```
sex=pd.get_dummies(x["Sex"],drop_first=True)
```

```
sex
```

	1
0	1
2	0
3	0
4	1
5	1
...	...
886	1
887	0
888	0
889	1
890	1

775 rows × 1 columns

```
#concat
x=pd.concat([x,sex],axis=1)
```

```
x.head()
```

	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	1
0	Braund, Mr. Owen Harris	1	22.000000	1	0	A/5 21171	7.2500	B96 B98	S	1
2	Heikkinen, Miss. Laina	0	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S	0
	Futrelle, Mrs									

```
x.drop(["Sex"],axis=1,inplace=True)
```

```
x.head(10)
```

	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	1
0	Braund, Mr. Owen Harris	22.000000	1	0	A/5 21171	7.2500	B96 B98	S	1
2	Heikkinen, Miss. Laina	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98	S	0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.000000	1	0	113803	53.1000	C123	S	0
4	Allen, Mr. William Henry	35.000000	0	0	373450	8.0500	B96 B98	S	1
5	Moran, Mr. James	29.699118	0	0	330877	8.4583	B96 B98	Q	1
6	McCarthy, Mr. Timothy J	54.000000	0	0	17463	51.8625	E46	S	1

```
x.shape
```

```
(775, 9)
```

7.splitting into training and testing set

```
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)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(542, 9)
(233, 9)
(542, 0)
(233, 0)
```

```
a=[1,2,3,4,5,6]
b=[1,0,1,5,6,3]

for i in range(5):
    a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3,random_state=100)
    print("with random state",a_train)
```

```
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
with random state [5, 4, 6, 1]
```

```
a=[1,2,3,4,5,6]
b=[1,0,1,5,6,3]

for i in range(5):
    a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3)
    print("without random state",a_train)
```

```
without random state [6, 2, 3, 4]
without random state [2, 4, 6, 3]
without random state [2, 6, 4, 3]
without random state [2, 6, 1, 5]
without random state [6, 4, 5, 1]
```

8.Feature Scaling

```
scale = StandardScaler()
x[['Age', 'Fare']] = scale.fit_transform(x[['Age', 'Fare']])
```

```
x.head()
```

	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	1
0	Braund, Mr. Owen Harris	-0.556219	1	0	A/5 21171	-0.779117	B96 B98	S	1
2	Heikkinen, Miss. Laina	-0.243027	0	0	STON/O2. 3101282	-0.729373	B96 B98	S	0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0.461654	1	0	113803	2.599828	C123	S	0

```
x_train
```

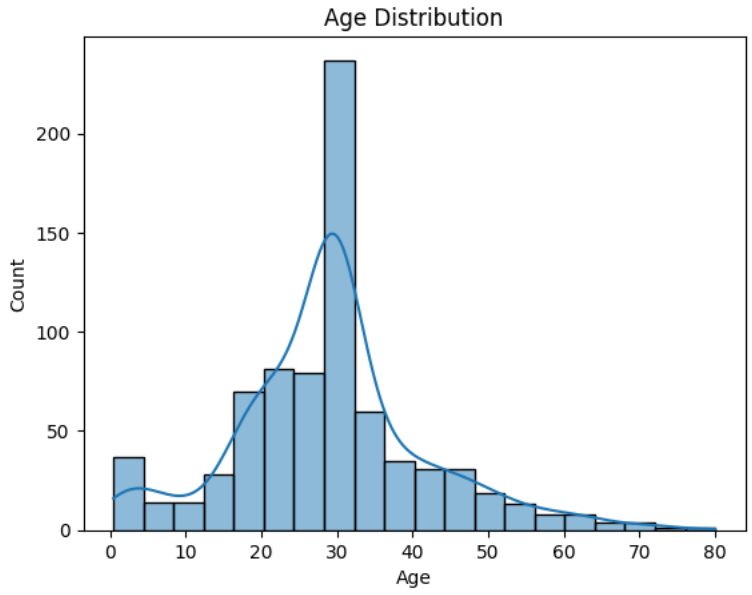
	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	1
654	Hegarty, Miss. Hanora "Nora"	18.000000	0	0	365226	6.7500	B96 B98	Q	0
38	Vander Planke, Miss. Augusta Maria	18.000000	2	0	345764	18.0000	B96 B98	S	0
646	Cor, Mr. Liudevit	19.000000	0	0	349231	7.8958	B96 B98	S	1
727	Mannion, Miss. Margareth	29.699118	0	0	36866	7.7375	B96 B98	Q	0
887	Graham, Miss. Margaret Edith	19.000000	0	0	112053	30.0000	B42	S	0
...	...	...	...	...	...	...	...	...	...
878	Laleff, Mr. Kristo	29.699118	0	0	349217	7.8958	B96 B98	S	1
	Cameron,				F.C.C		B96		

DATA VISUALIZATION

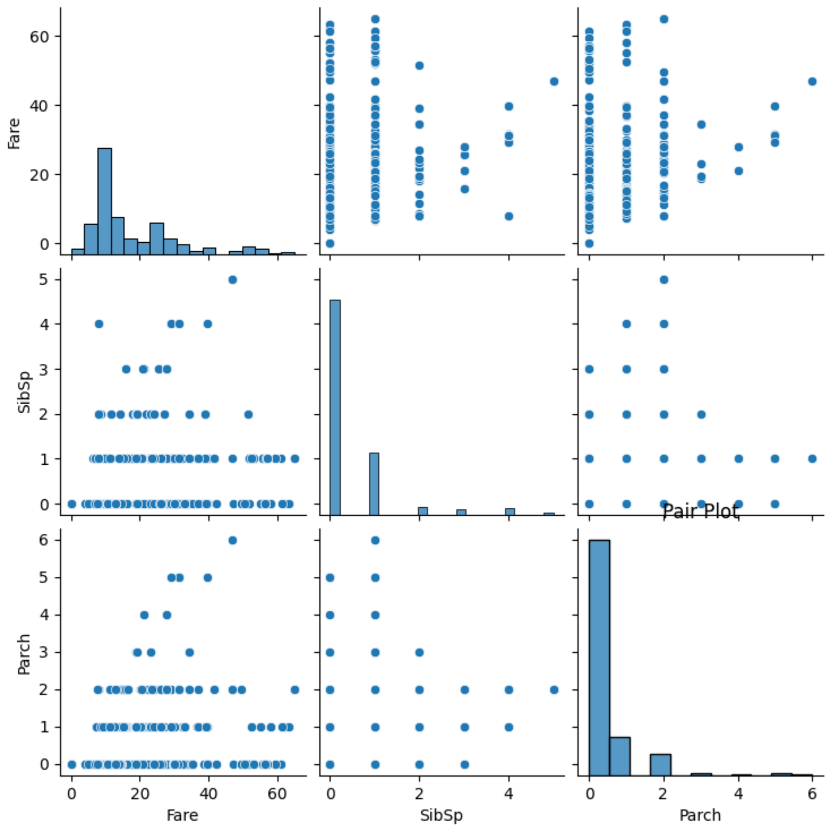
```
sns.countplot(data=dataset, x='Survived')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```

Survival Count

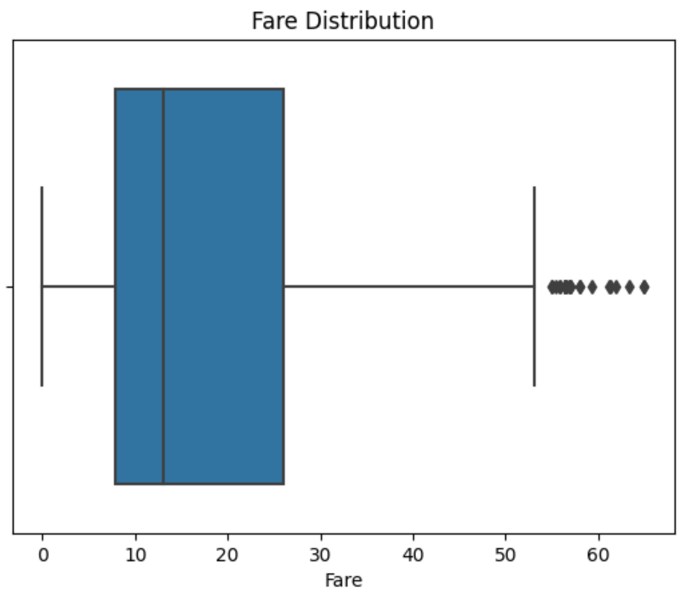
```
sns.histplot(data=dataset, x='Age', bins=20, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



```
sns.pairplot(data=dataset[['Fare', 'SibSp', 'Parch']])
plt.title('Pair Plot')
plt.show()
```

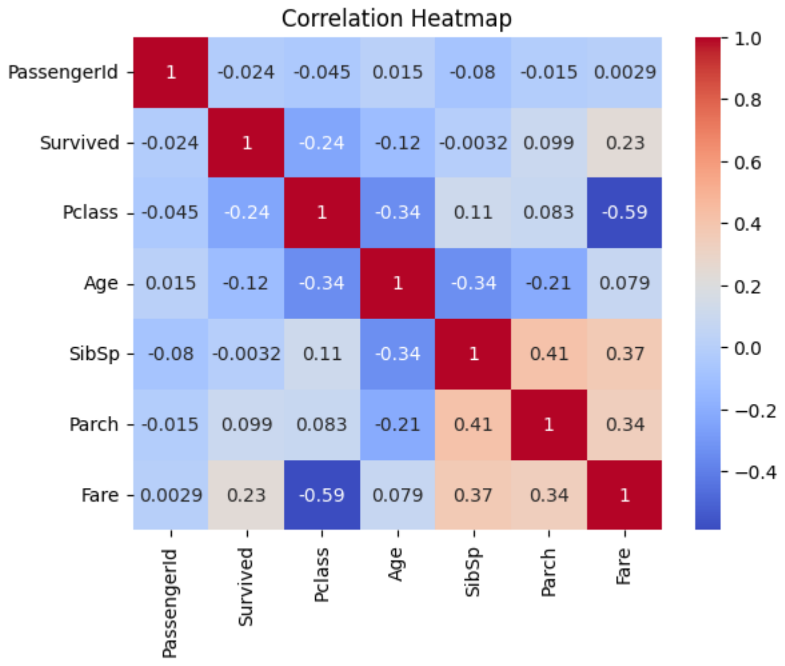


```
sns.boxplot(data=dataset, x='Fare')
plt.title('Fare Distribution')
plt.xlabel('Fare')
plt.show()
```



```
corr_matrix = dataset.corr()
sns.heatmap(corr_matrix, annot=True,cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

```
    <ipython-input-67-6ddef7c4acad>:1: FutureWarning: The default value of numeric_onl
      corr_matrix = dataset.corr()
```



	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1	-0.024	-0.045	0.015	-0.08	-0.015	0.0029
Survived	-0.024	1	-0.24	-0.12	-0.0032	0.099	0.23
Pclass	-0.045	-0.24	1	-0.34	0.11	0.083	-0.59
Age	0.015	-0.12	-0.34	1	-0.34	-0.21	0.079
SibSp	-0.08	-0.0032	0.11	-0.34	1	0.41	0.37
Parch	-0.015	0.099	0.083	-0.21	0.41	1	0.34
Fare	0.0029	0.23	-0.59	0.079	0.37	0.34	1