```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

##Importing data

```
train_df=pd.read_csv(r'/content/train.csv')
train_df
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         0       3

                                                  Name     Sex   Age
SibSp  \
0                              Braund, Mr. Owen Harris    male  22.0
1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                               Heikkinen, Miss. Laina  female  26.0
0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                             Allen, Mr. William Henry    male  35.0
0
..                                                 ...     ...   ...
...
886                              Montvila, Rev. Juozas    male  27.0
0
887                       Graham, Miss. Margaret Edith  female  19.0
0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN
1
889                              Behr, Mr. Karl Howell    male  26.0
0
890                                Dooley, Mr. Patrick    male  32.0
0

     Parch            Ticket     Fare Cabin Embarked
```

```
0         0         A/5 21171   7.2500   NaN        S
1         0          PC 17599  71.2833   C85        C
2         0  STON/O2. 3101282   7.9250   NaN        S
3         0            113803  53.1000  C123        S
4         0            373450   8.0500   NaN        S
..      ...               ...      ...   ...      ...
886       0            211536  13.0000   NaN        S
887       0            112053  30.0000   B42        S
888       2        W./C. 6607  23.4500   NaN        S
889       0            111369  30.0000  C148        C
890       0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
test_df=pd.read_csv(r'/content/test.csv')
test_df
```

```
     PassengerId  Pclass                                          Name
\
0            892       3                              Kelly, Mr. James

1            893       3              Wilkes, Mrs. James (Ellen Needs)

2            894       2                     Myles, Mr. Thomas Francis

3            895       3                              Wirz, Mr. Albert

4            896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)

..           ...     ...                                           ...

413         1305       3                            Spector, Mr. Woolf

414         1306       1                   Oliva y Ocana, Dona. Fermina

415         1307       3                   Saether, Mr. Simon Sivertsen

416         1308       3                           Ware, Mr. Frederick

417         1309       3                     Peter, Master. Michael J


        Sex   Age  SibSp  Parch              Ticket      Fare Cabin
Embarked
0      male  34.5      0      0              330911    7.8292   NaN
Q
1    female  47.0      1      0              363272    7.0000   NaN
S
2      male  62.0      0      0              240276    9.6875   NaN
Q
3      male  27.0      0      0              315154    8.6625   NaN
```

```
S
4    female  22.0      1      1              3101298   12.2875   NaN
S
..      ...   ...    ...    ...                 ...       ...   ...
...
413    male   NaN      0      0           A.5. 3236    8.0500   NaN
S
414  female  39.0      0      0            PC 17758  108.9000  C105
C
415    male  38.5      0      0  SOTON/O.Q. 3101262    7.2500   NaN
S
416    male   NaN      0      0              359309    8.0500   NaN
S
417    male   NaN      1      1                2668   22.3583   NaN
C

[418 rows x 11 columns]
```

```
train_df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
train_df.describe()
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

             Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```
train_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
```

```
train_df.corr( )
```

```
<ipython-input-7-583a47030d1b>: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.
  train_df.corr( )
```

```
             PassengerId  Survived    Pclass       Age     SibSp
Parch  \
PassengerId     1.000000 -0.005007 -0.035144  0.036847 -0.057527 -
0.001652
Survived       -0.005007  1.000000 -0.338481 -0.077221 -0.035322
0.081629
Pclass         -0.035144 -0.338481  1.000000 -0.369226  0.083081
0.018443
Age             0.036847 -0.077221 -0.369226  1.000000 -0.308247 -
0.189119
SibSp          -0.057527 -0.035322  0.083081 -0.308247  1.000000
0.414838
Parch          -0.001652  0.081629  0.018443 -0.189119  0.414838
1.000000
Fare            0.012658  0.257307 -0.549500  0.096067  0.159651
0.216225

                 Fare
PassengerId  0.012658
Survived     0.257307
Pclass      -0.549500
Age          0.096067
SibSp        0.159651
Parch        0.216225
Fare         1.000000
```

##Handling Null Values

```
train_df.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
```

```
train_df.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
```

```
train_df["Age"].fillna(train_df["Age"].median(), inplace = True)
```

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

```
train_df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
dtype: int64
```

```
train_df = train_df.drop('Cabin', axis=1)
```

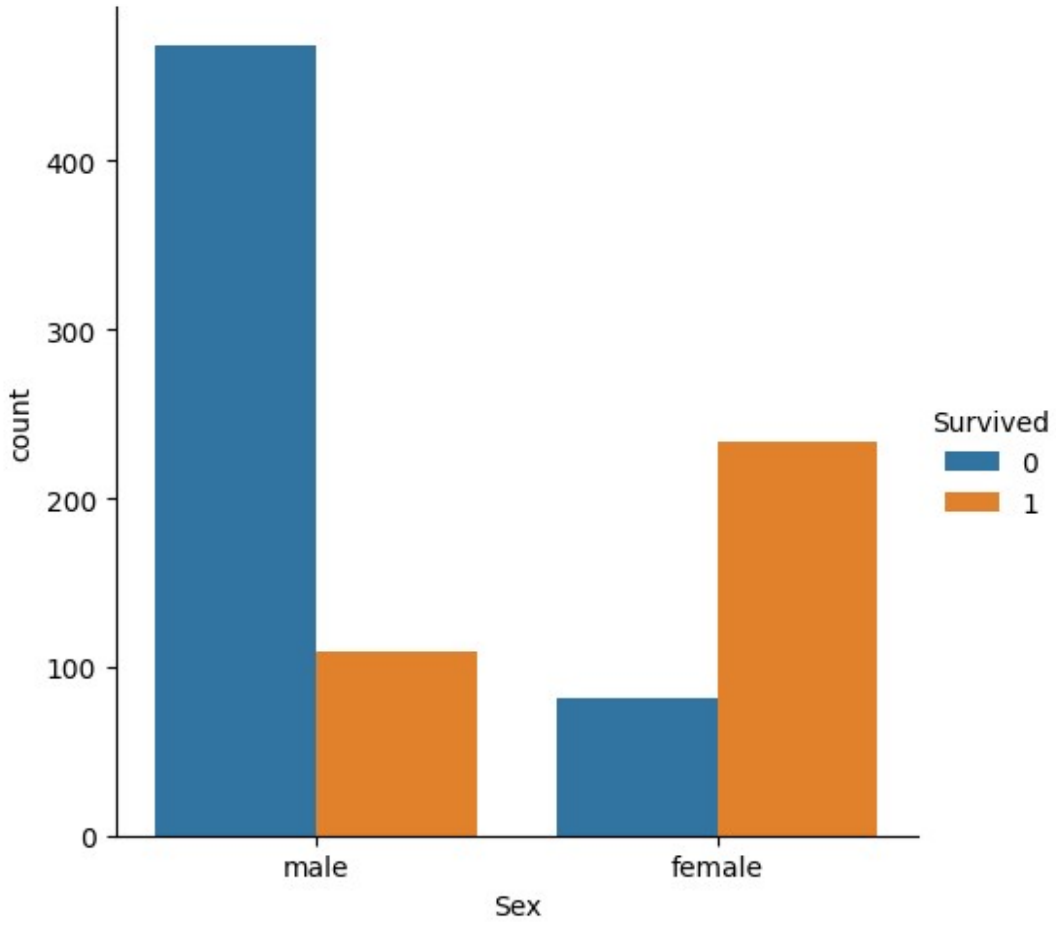
```
train_df.isnull().any()
```

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Embarked       False
dtype: bool
```

##Data visualsation

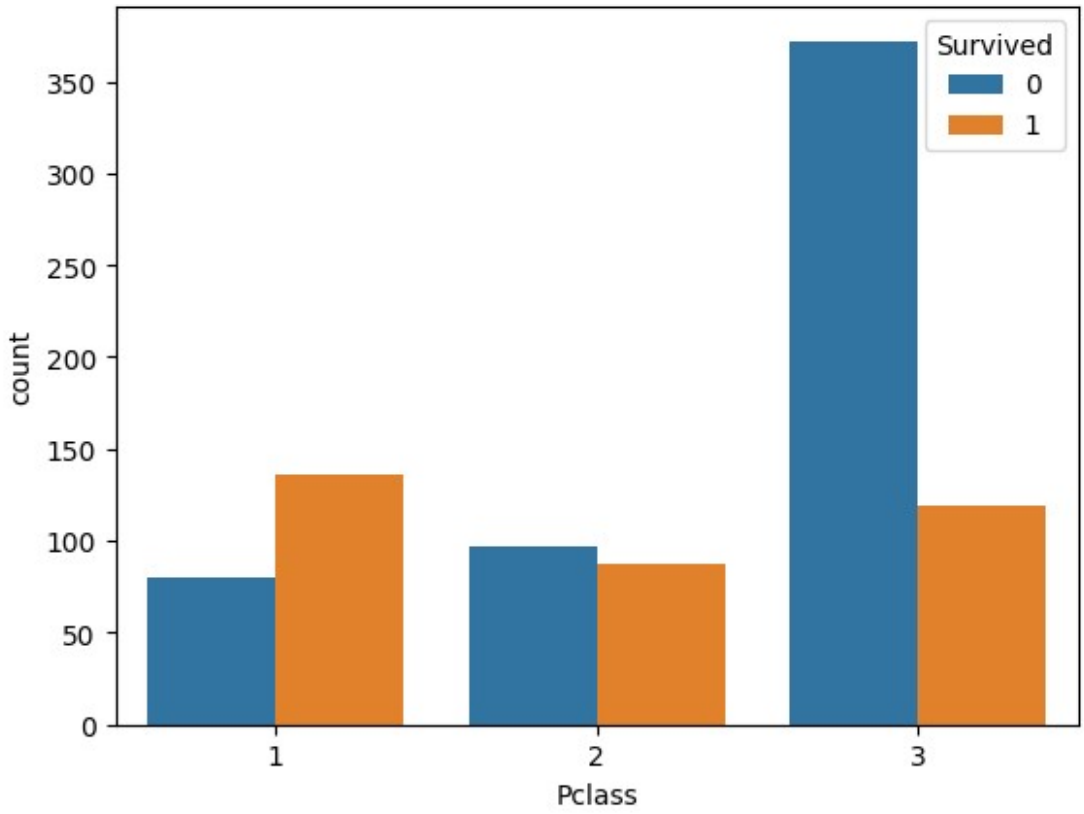
```
sns.catplot(x ="Sex", hue ="Survived",kind ="count", data = train_df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7b8803641930>
```




```
sns.countplot(x=train_df['Pclass'], hue=train_df['Survived'])
```

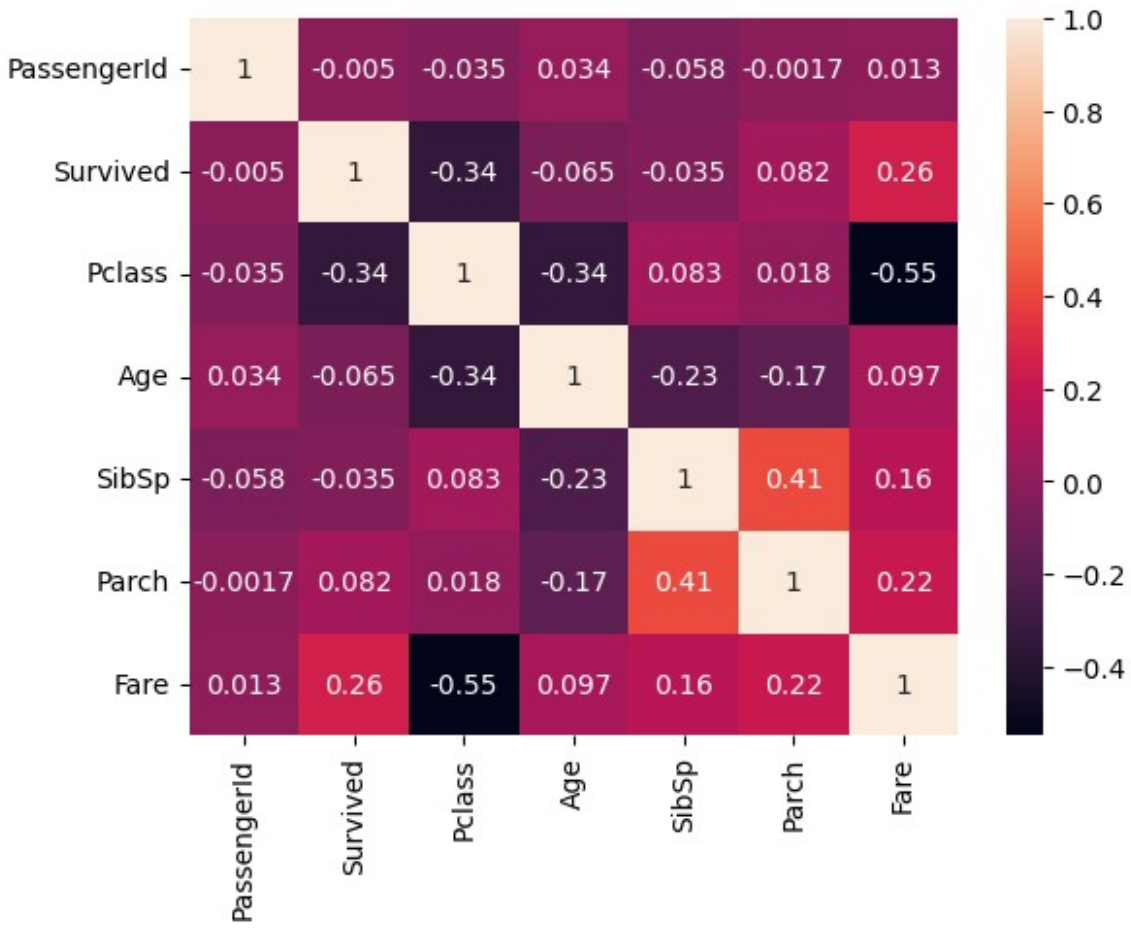
```
<Axes: xlabel='Pclass', ylabel='count'>
```




```
sns.heatmap(train_df.corr(), annot=True)
```

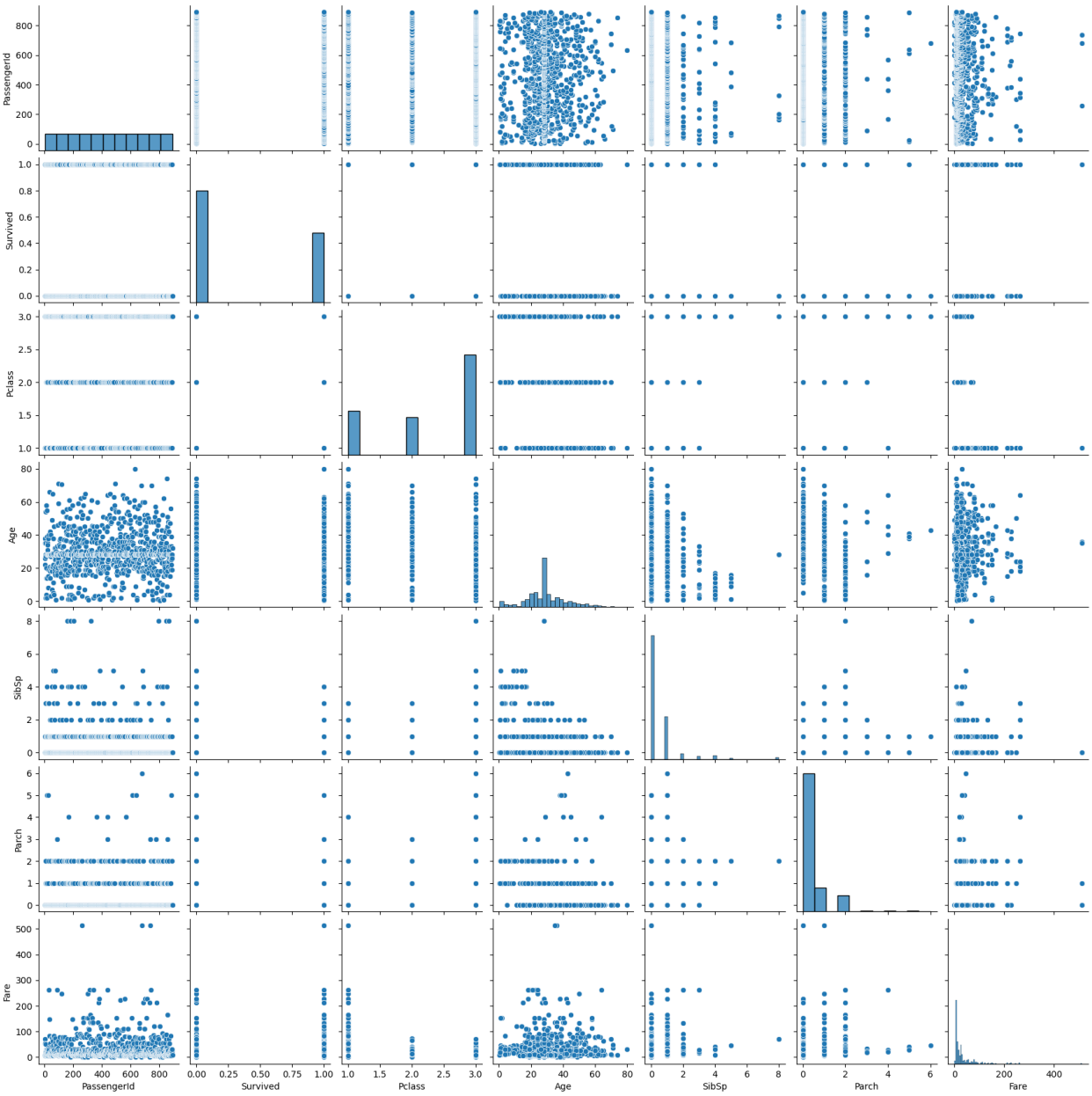
```
<ipython-input-17-5cffd907106f>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  sns.heatmap(train_df.corr(), annot=True)

<Axes: >
```

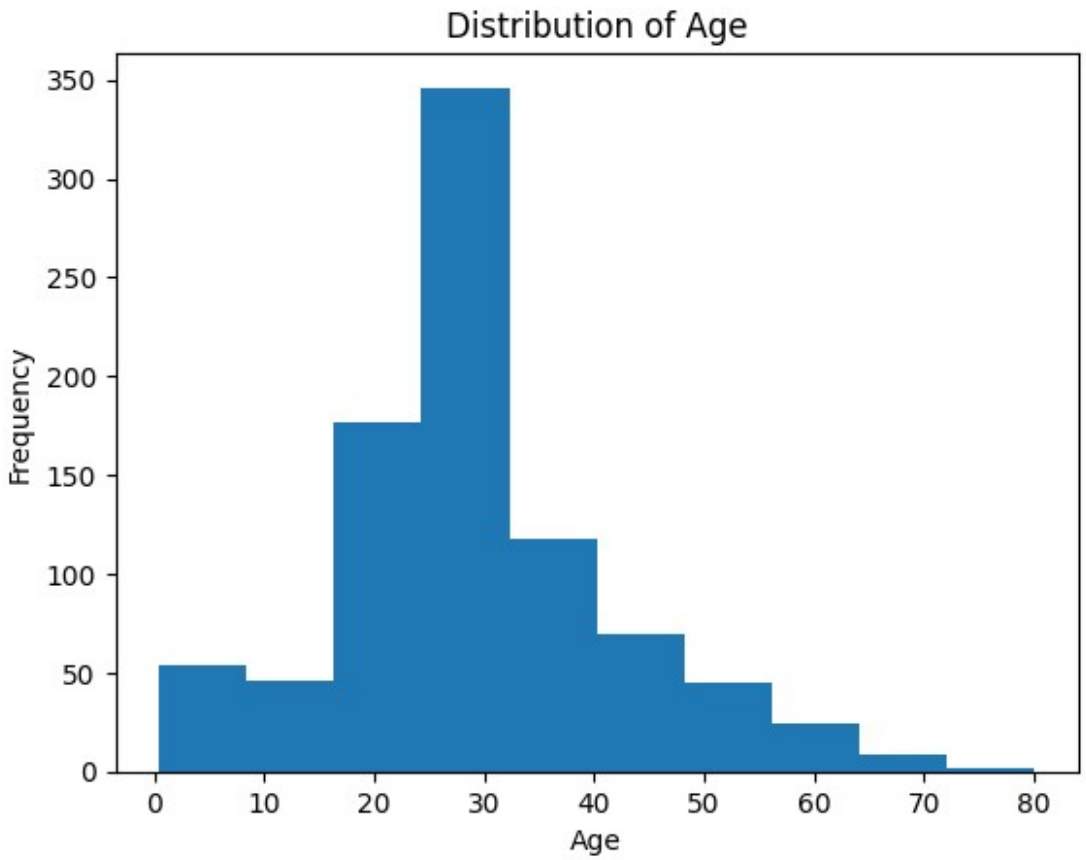



```
sns.pairplot(train_df)
```

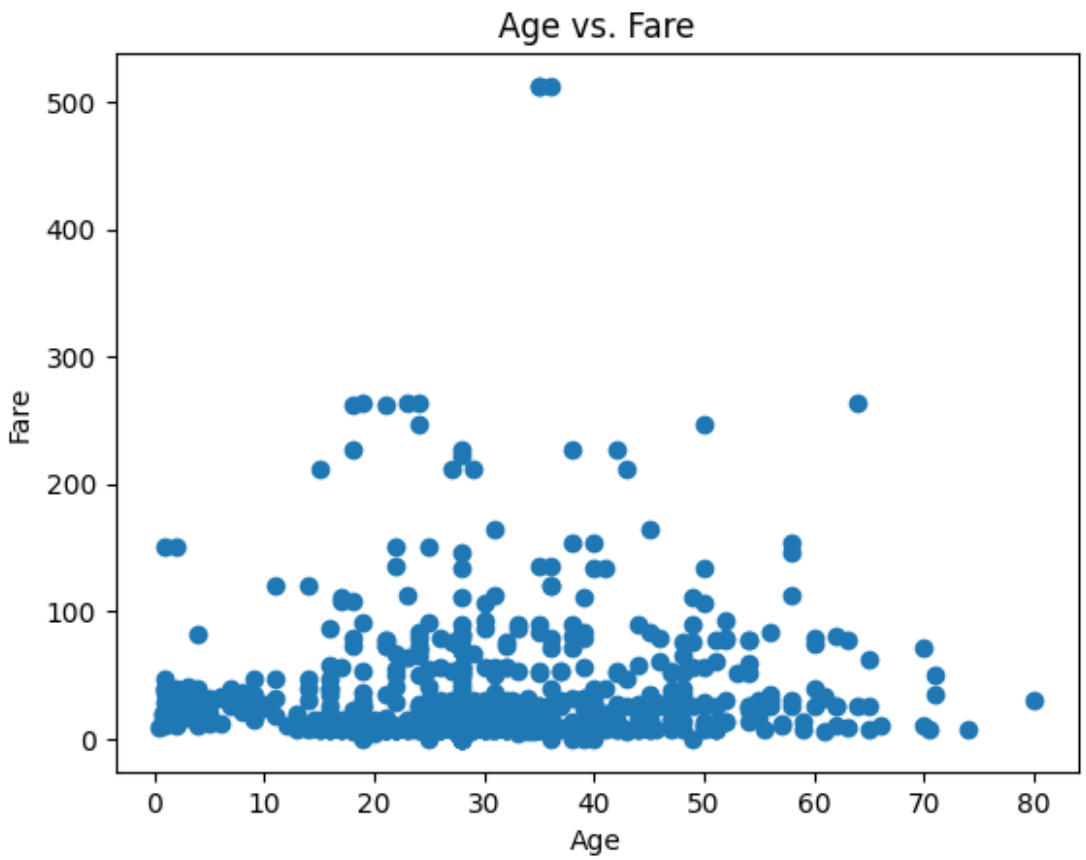
```
<seaborn.axisgrid.PairGrid at 0x7b87cb830bb0>
```



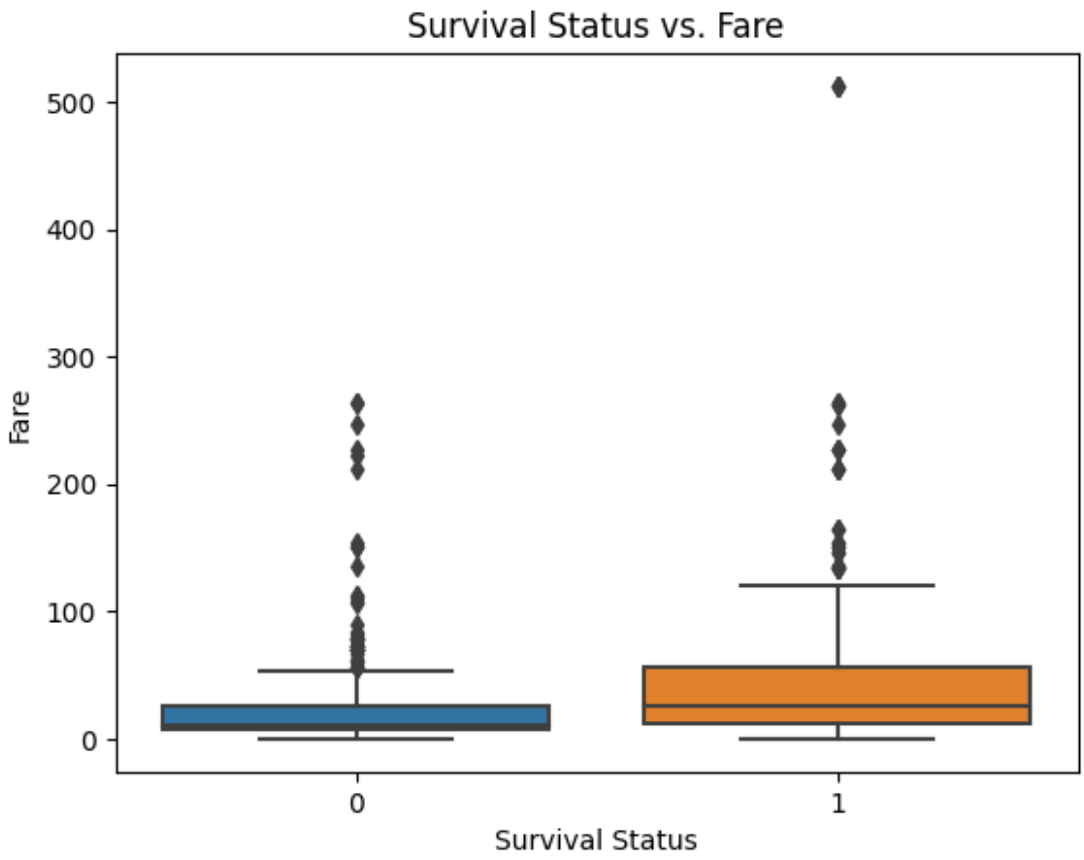

```
plt.hist(train_df['Age'], bins=10)
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()
```




```
plt.scatter(train_df['Age'], train_df['Fare'])
plt.xlabel('Age')
plt.ylabel('Fare')
plt.title('Age vs. Fare')
plt.show()
```

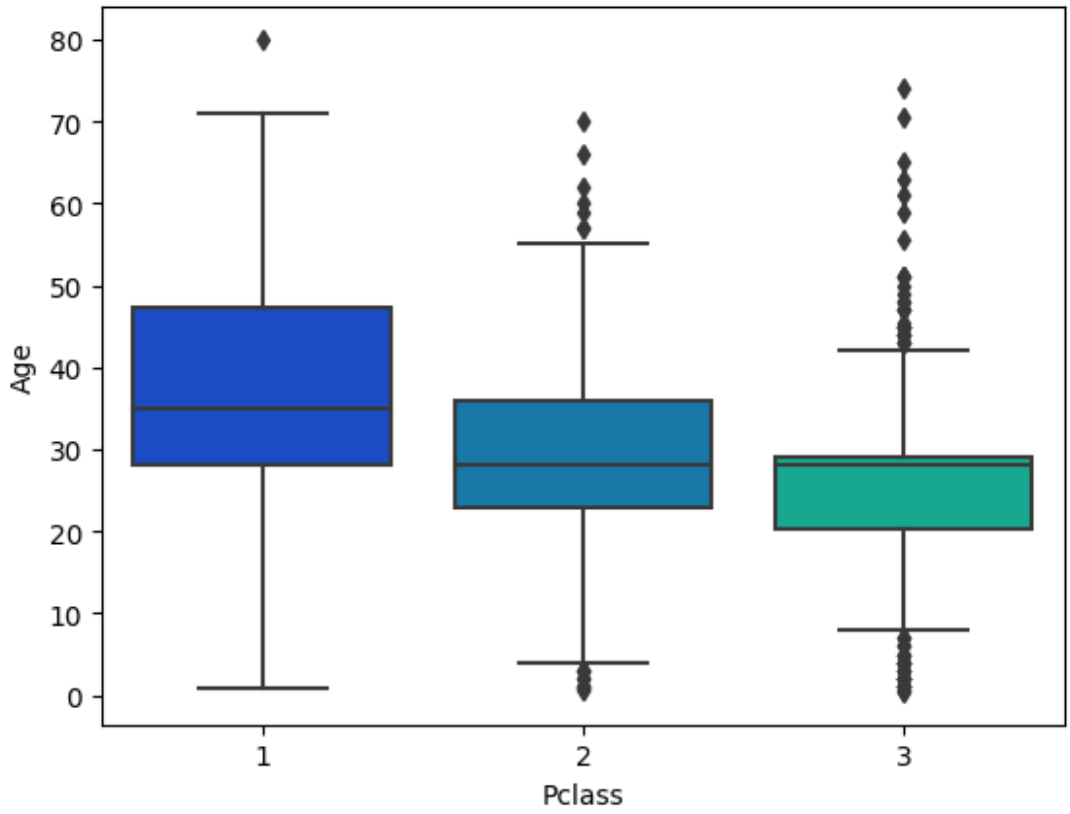



```
sns.boxplot(x=train_df['Survived'], y=train_df['Fare'])
plt.xlabel('Survival Status')
plt.ylabel('Fare')
plt.title('Survival Status vs. Fare')
plt.show()
```



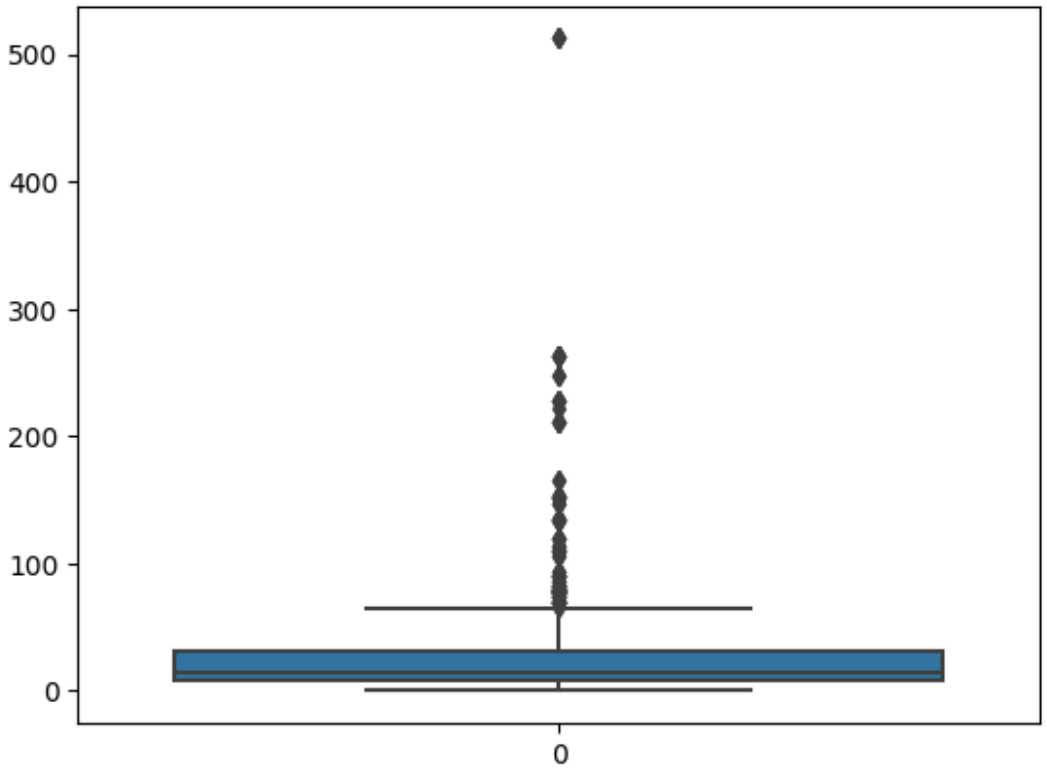

```
sns.boxplot(x='Pclass',y='Age',data=train_df,palette='winter')
```

```
<Axes: xlabel='Pclass', ylabel='Age'>
```



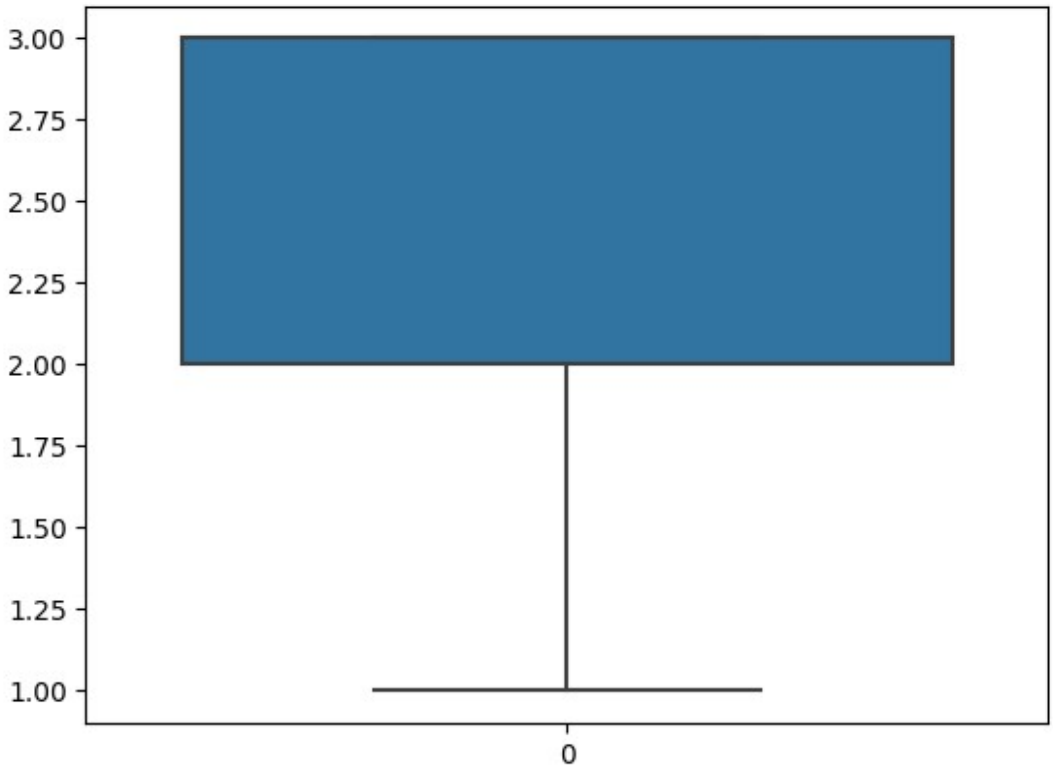

```
sns.boxplot(train_df["Fare"])
```

```
<Axes: >
```



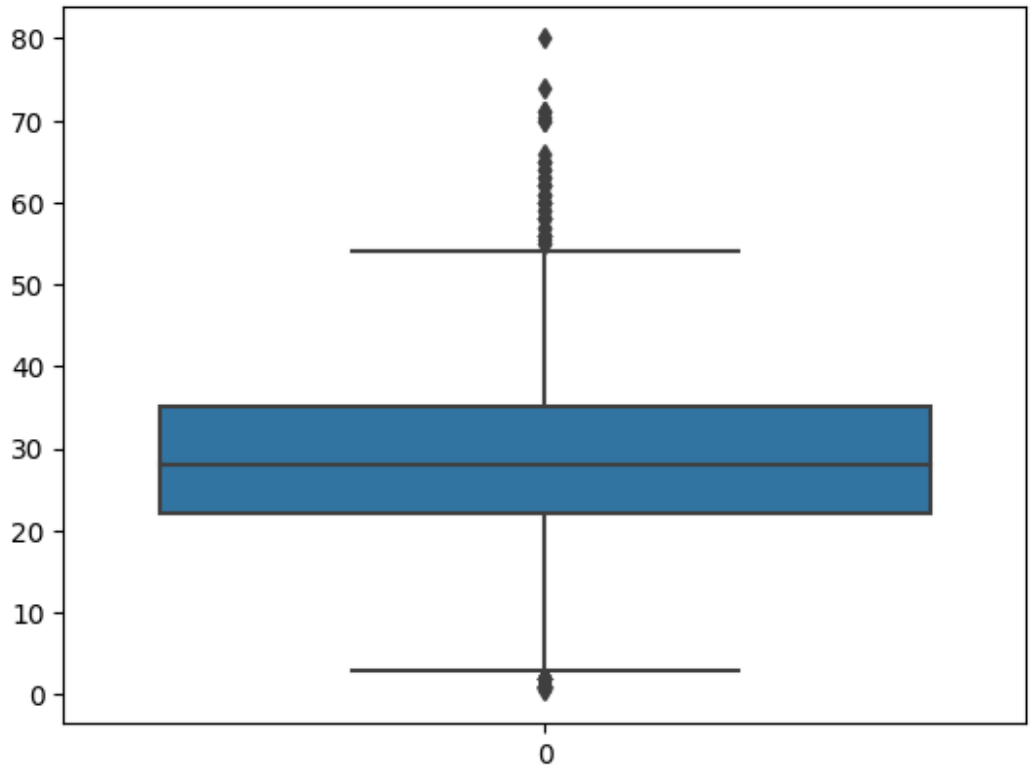

```
sns.boxplot(train_df["Pclass"])
```

```
<Axes: >
```



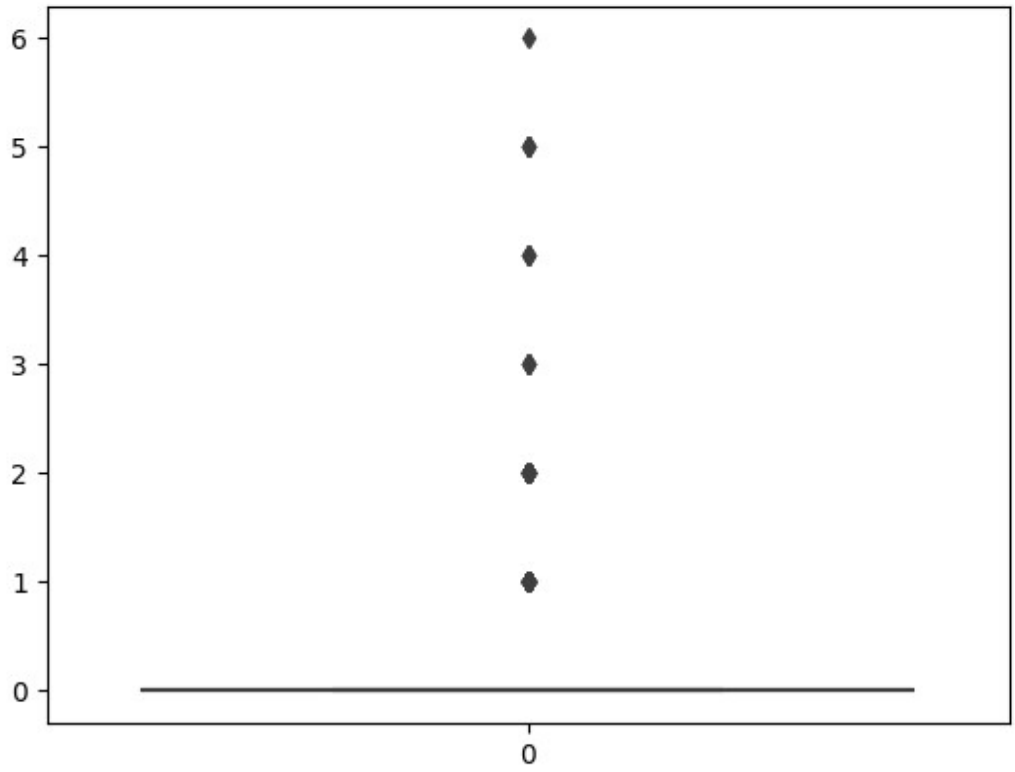

```
sns.boxplot(train_df["Age"])
```

```
<Axes: >
```




```
sns.boxplot(train_df["Parch"])
```

```
<Axes: >
```



##Outlier removal

```
Q1 = train_df.quantile(0.25)
Q3 = train_df.quantile(0.75)
IQR = Q3 - Q1
```

```
<ipython-input-28-6673476a5ffd>:1: FutureWarning: The default value of
numeric_only in DataFrame.quantile 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.
  Q1 = train_df.quantile(0.25)
<ipython-input-28-6673476a5ffd>:2: FutureWarning: The default value of
numeric_only in DataFrame.quantile 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.
  Q3 = train_df.quantile(0.75)
```

```
threshold = 1.5

outlier_indices = ((train_df < (Q1 - threshold * IQR)) | (train_df >
(Q3 + threshold * IQR))).any(axis=1)
```

```
<ipython-input-32-85372db698c9>:1: FutureWarning: Automatic reindexing
on DataFrame vs Series comparisons is deprecated and will raise
ValueError in a future version. Do `left, right = left.align(right,
axis=1, copy=False)` before e.g. `left == right`
```

```
  outlier_indices = ((train_df < (Q1 - threshold * IQR)) | (train_df > (Q3 + threshold * IQR))).any(axis=1)

titanic_data_no_outliers = train_df[~outlier_indices]
```

##Splitting independant and dependant variables

```
X = titanic_data_no_outliers.drop('Survived', axis=1)

y = titanic_data_no_outliers['Survived']

X_encoded = pd.get_dummies(X, columns=['Sex', 'Embarked'], drop_first=True)

X_encoded
```

```
     PassengerId  Pclass                                               Name
Age  \
0              1       3                            Braund, Mr. Owen Harris
22.0
2              3       3                             Heikkinen, Miss. Laina
26.0
3              4       1     Futrelle, Mrs. Jacques Heath (Lily May Peel)
35.0
4              5       3                           Allen, Mr. William Henry
35.0
5              6       3                                   Moran, Mr. James
28.0
..           ...     ...                                              ...
...
884          885       3                           Sutehall, Mr. Henry Jr
25.0
886          887       2                            Montvila, Rev. Juozas
27.0
887          888       1                    Graham, Miss. Margaret Edith
19.0
889          890       1                            Behr, Mr. Karl Howell
26.0
890          891       3                              Dooley, Mr. Patrick
32.0

     SibSp  Parch            Ticket     Fare  Sex_male  Embarked_Q
Embarked_S
0        1      0         A/5 21171   7.2500         1           0
1
2        0      0  STON/O2. 3101282   7.9250         0           0
1
3        1      0            113803  53.1000         0           0
1
4        0      0            373450   8.0500         1           0
```

```
1
5         0      0            330877   8.4583          1            1
0
..      ...    ...               ...      ...        ...          ...
...
884       0      0   SOTON/OQ 392076   7.0500          1            0
1
886       0      0            211536  13.0000          1            0
1
887       0      0            112053  30.0000          0            0
1
889       0      0            111369  30.0000          1            0
0
890       0      0            370376   7.7500          1            1
0

[577 rows x 11 columns]
```

```
X_encoded.drop('Name',axis=1, inplace=True)

X_encoded.drop('Ticket',axis=1, inplace=True)
```

##Scaling and Splitting

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_encoded)

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=42)


print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (461, 9)
Shape of X_test: (116, 9)
Shape of y_train: (461,)
Shape of y_test: (116,)

X_train

array([[-1.32870581,  0.67832969, -0.15253451, ...,  0.57401488,
        -0.35320863,  0.5766832 ],
       [ 1.37041894,  0.67832969,  0.19857883, ...,  0.57401488,
        -0.35320863,  0.5766832 ],
```

```
       [-1.35184116, -0.68542024,  0.54969217, ...,  0.57401488,
        -0.35320863,  0.5766832 ],
       ...,
       [-0.09096431, -0.68542024, -0.15253451, ..., -1.7421151 ,
        -0.35320863,  0.5766832 ],
       [ 0.88843238,  0.67832969, -0.15253451, ..., -1.7421151 ,
         2.83118791, -1.73405434],
       [-1.13976707,  0.67832969, -0.15253451, ...,  0.57401488,
        -0.35320863,  0.5766832 ]])
```

```
from sklearn.linear_model import LinearRegression
```

```
le = LinearRegression()
```

```
le.fit(X_train, y_train)
```

```
LinearRegression()
```

```
predict = le.predict(X_test)
```