Subject : LP-III- Machine Learning

Miniproject No. 2

Guided by:*Prof. Prajakta Khadkikar*

Submitted by:
Rohit James (41266)
Sufiya Sayyed (41278)
Aditya Wanjale (41281)

Problem Statement:

Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.).
Dataset Link: https://www.kaggle.com/competitions/titanic/data (https://www.kaggle.com/competitions/titanic/data)

Imports

Libraries

In [342]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Datasets

In [343]:
```
df_train = pd.read_csv("train.csv")
```

In [344]:
```
df_train
```

Out[344]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	C
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 12 columns

In [345]:
```
df_train.shape
```

Out[345]: (891, 12)

In [346]:
```
df_test = pd.read_csv("test.csv")
df_test
```

Out[346]:

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

418 rows × 11 columns

In [347]:
```
df_test.shape
```

Out[347]: (418, 11)

In [348]:
```
df_train.dtypes, df_test.dtypes
```

Out[348]:
```
(PassengerId      int64
 Survived         int64
 Pclass           int64
 Name            object
 Sex             object
 Age            float64
 SibSp            int64
 Parch            int64
 Ticket          object
 Fare           float64
 Cabin           object
 Embarked        object
 dtype: object,
 PassengerId      int64
 Pclass           int64
 Name            object
 Sex             object
 Age            float64
 SibSp            int64
 Parch            int64
 Ticket          object
 Fare           float64
 Cabin           object
 Embarked        object
 dtype: object)
```

Cleaning

In [349]:
```
df_train.isnull().sum()
```

Out[349]:
```
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 [350]:
```
df_train['Age'].fillna(df_train['Age'].mean(),inplace=True)
```

In [351]:
```
df_train['Embarked'].fillna(df_train['Embarked'].mode()[0],inplace=True)
```

In [352]:
```
df_train['Cabin'].fillna(df_train['Cabin'].mode()[0],inplace=True)
```

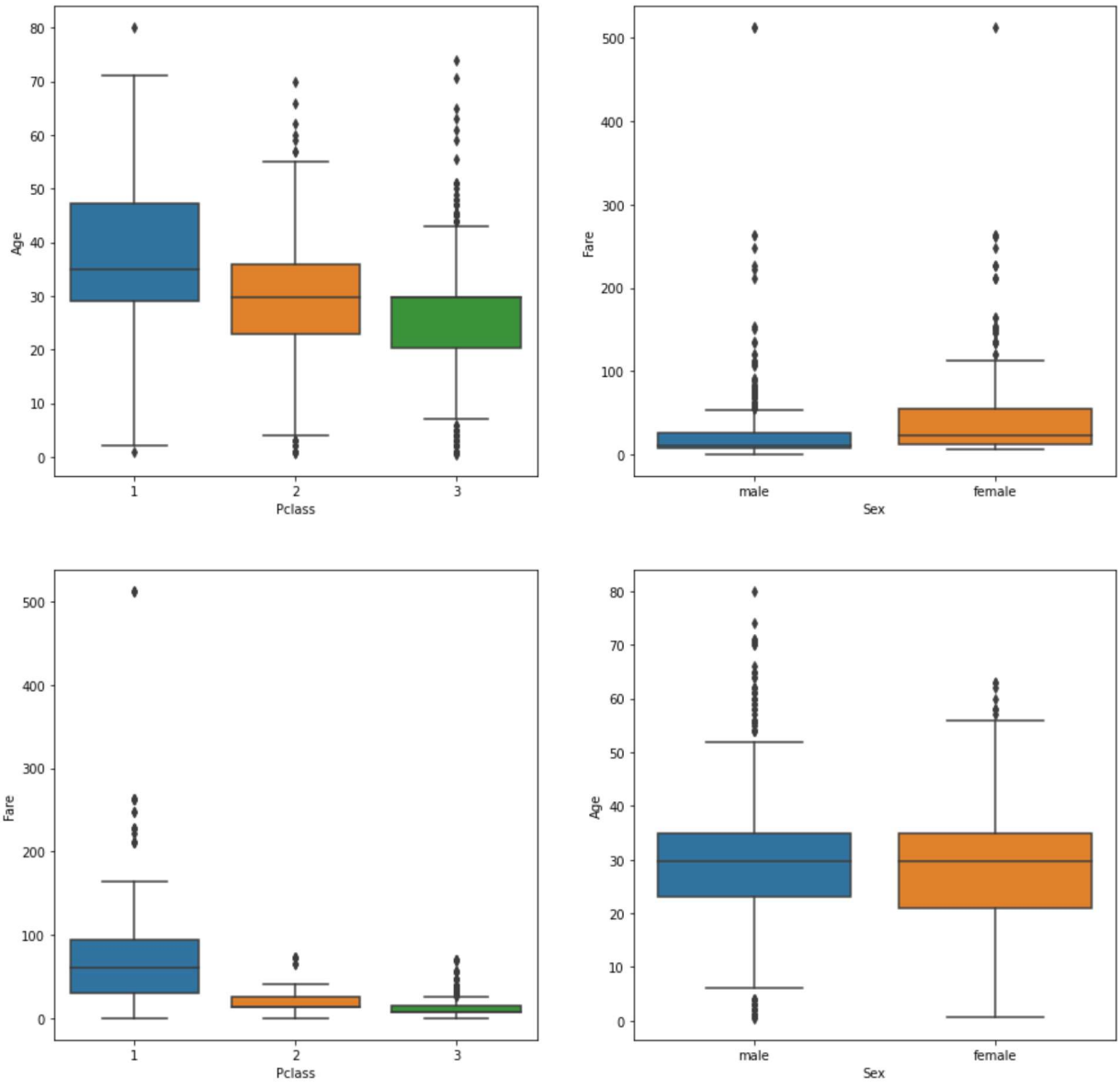
In [353]:
```
df_train.isnull().sum()
```

Out[353]:
```
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
```

Visualization

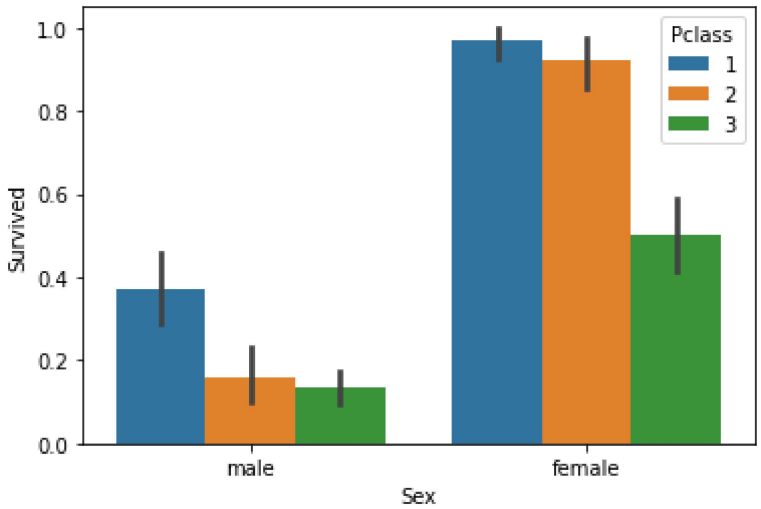
In [354]:
```
fig = plt.figure(figsize=(15,15))

ax1 = fig.add_subplot(221)
sns.boxplot(x='Pclass',y='Age',data=df_train)

ax2 = fig.add_subplot(222)
sns.boxplot(x='Sex',y='Fare',data=df_train)

ax3 = fig.add_subplot(223)
sns.boxplot(x='Pclass',y='Fare',data=df_train)

ax4 = fig.add_subplot(224)
sns.boxplot(x='Sex',y='Age',data=df_train)
```

Out[354]: <AxesSubplot:xlabel='Sex', ylabel='Age'>



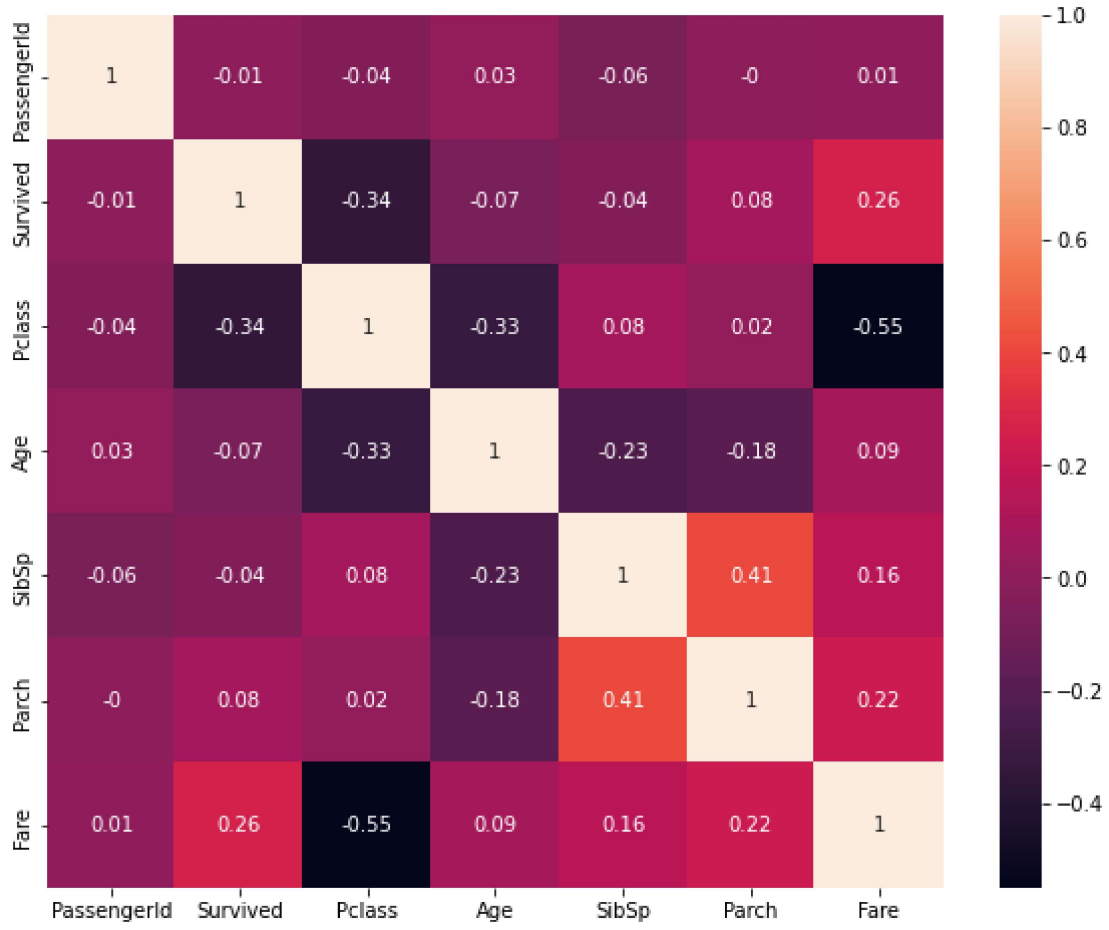
In [355]:
```
sns.barplot(x = 'Sex', y = 'Survived', hue = 'Pclass', data = df_train)
```

Out[355]: <AxesSubplot:xlabel='Sex', ylabel='Survived'>



In [356]:
```
fig = plt.figure(figsize=(10,8))
sns.heatmap(df_train.corr().round(2), annot=True)
```

Out[356]: <AxesSubplot:>



In [358]:
```
males = []
males = [1 if df_train['Sex'][i]=='male' else 0 for i in range(0,df_train.shape[0
df_train['Male'] = males
```

In [359]:
```
df_train.drop(['Sex','Name','Ticket','Cabin','Embarked'],axis = 1,inplace = True)
df_train.head(3)
```

Out[359]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Male
0	1	0	3	22.0	1	0	7.2500	1
1	2	1	1	38.0	1	0	71.2833	0
2	3	1	3	26.0	0	0	7.9250	0

Preparing training sets

In [360]:
```
X_train = df_train.drop('Survived',axis=1).values
X_train
```

Out[360]:
```
array([[  1.        ,   3.        ,  22.        , ...,   0.        ,
          7.25      ,   1.        ],
       [  2.        ,   1.        ,  38.        , ...,   0.        ,
         71.2833    ,   0.        ],
       [  3.        ,   3.        ,  26.        , ...,   0.        ,
          7.925     ,   0.        ],
       ...,
       [889.        ,   3.        ,  29.69911765, ...,   2.        ,
         23.45      ,   0.        ],
       [890.        ,   1.        ,  26.        , ...,   0.        ,
         30.        ,   1.        ],
       [891.        ,   3.        ,  32.        , ...,   0.        ,
          7.75      ,   1.        ]])
```

In [361]:
```
y_train = df_train['Survived'].values
#Y
```

Model Building

In [362]:
```
from sklearn.linear_model import LogisticRegression

classifier = LogisticRegression(random_state =0)
classifier.fit(X_train,y_train)
```

```
C:\Users\Lenovo\.conda\envs\myenv\lib\site-packages\sklearn\linear_model\_logis
tic.py:765: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-
learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on)
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[362]:
```
LogisticRegression(random_state=0)
```

Preparing testing set

In [363]:
```
df_test.head(3)
```

Out[363]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q

In [364]:
```
males = []
males = [1 if df_test['Sex'][i]=='male' else 0 for i in range(0,df_test.shape[0])
df_test['Male'] = males
```

In [365]:
```
df_test.drop(['Name','Ticket','Cabin','Embarked'],axis=1,inplace=True)
```

In [366]:
```
df_test
```

Out[366]:

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Male
0	892	3	male	34.5	0	0	7.8292	1
1	893	3	female	47.0	1	0	7.0000	0
2	894	2	male	62.0	0	0	9.6875	1
3	895	3	male	27.0	0	0	8.6625	1
4	896	3	female	22.0	1	1	12.2875	0
...	...	...	...	...	...	...	...	...
413	1305	3	male	NaN	0	0	8.0500	1
414	1306	1	female	39.0	0	0	108.9000	0
415	1307	3	male	38.5	0	0	7.2500	1
416	1308	3	male	NaN	0	0	8.0500	1
417	1309	3	male	NaN	1	1	22.3583	1

418 rows × 8 columns

In [367]:
```
df_test.isnull().sum()
```

Out[367]:
```
PassengerId     0
Pclass          0
Sex             0
Age            86
SibSp           0
Parch           0
Fare            1
Male            0
dtype: int64
```

In [368]:
```
df_test['Age'].fillna(df_test['Age'].mean(),inplace=True)
df_test['Fare'].fillna(df_test['Fare'].mean(),inplace=True)
```

In [369]:
```
df_test.isnull().sum()
```

Out[369]:
```
PassengerId    0
Pclass         0
Sex            0
Age            0
SibSp          0
Parch          0
Fare           0
Male           0
dtype: int64
```

In [370]:
```
#df_test = df_test.reset_index()
```

In [371]:
```
df_test.head(3)
```

Out[371]:

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Male
0	892	3	male	34.5	0	0	7.8292	1
1	893	3	female	47.0	1	0	7.0000	0
2	894	2	male	62.0	0	0	9.6875	1

In [372]:
```
df_test.drop('Sex',axis=1,inplace=True)
```

In [376]:
```
y_test = df_test.iloc[:,0:7].values
y_test
```

Out[376]:
```
array([[8.92000000e+02, 3.00000000e+00, 3.45000000e+01, ...,
        0.00000000e+00, 7.82920000e+00, 1.00000000e+00],
       [8.93000000e+02, 3.00000000e+00, 4.70000000e+01, ...,
        0.00000000e+00, 7.00000000e+00, 0.00000000e+00],
       [8.94000000e+02, 2.00000000e+00, 6.20000000e+01, ...,
        0.00000000e+00, 9.68750000e+00, 1.00000000e+00],
       ...,
       [1.30700000e+03, 3.00000000e+00, 3.85000000e+01, ...,
        0.00000000e+00, 7.25000000e+00, 1.00000000e+00],
       [1.30800000e+03, 3.00000000e+00, 3.02725904e+01, ...,
        0.00000000e+00, 8.05000000e+00, 1.00000000e+00],
       [1.30900000e+03, 3.00000000e+00, 3.02725904e+01, ...,
        1.00000000e+00, 2.23583000e+01, 1.00000000e+00]])
```

Prediction

In [377]:
```
y_predict_test = classifier.predict(y_test)
y_predict_test
```

Out[377]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0],
      dtype=int64)
```

In [380]:
```
df_test['Survived'] = y_predict_test
```

In [381]:
```
df_test
```

Out[381]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Male	Survived
0	892	3	34.50000	0	0	7.8292	1	0
1	893	3	47.00000	1	0	7.0000	0	0
2	894	2	62.00000	0	0	9.6875	1	0
3	895	3	27.00000	0	0	8.6625	1	0
4	896	3	22.00000	1	1	12.2875	0	1
...	...	...	...	...	...	...	...	...
413	1305	3	30.27259	0	0	8.0500	1	0
414	1306	1	39.00000	0	0	108.9000	0	1
415	1307	3	38.50000	0	0	7.2500	1	0
416	1308	3	30.27259	0	0	8.0500	1	0
417	1309	3	30.27259	1	1	22.3583	1	0

418 rows × 8 columns

Visualizing predicted data

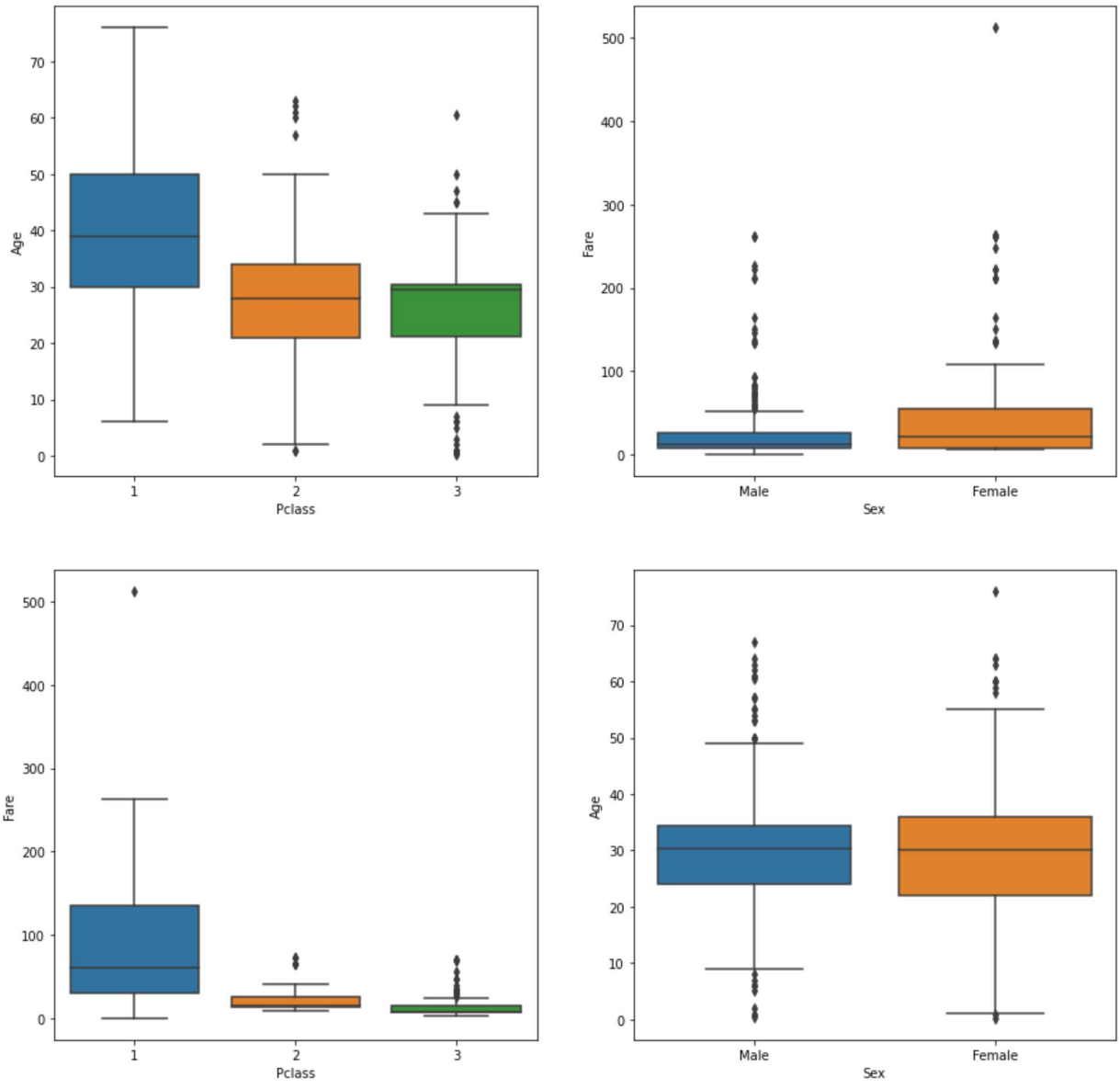
In [385]:
```
Sex = []
Sex = ['Male' if df_test['Male'][i]==1 else 'Female' for i in range(0,df_test.sha
df_test['Sex'] = Sex
df_test.head(3)
```

Out[385]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Male	Survived	Sex
0	892	3	34.5	0	0	7.8292	1	0	Male
1	893	3	47.0	1	0	7.0000	0	0	Female
2	894	2	62.0	0	0	9.6875	1	0	Male

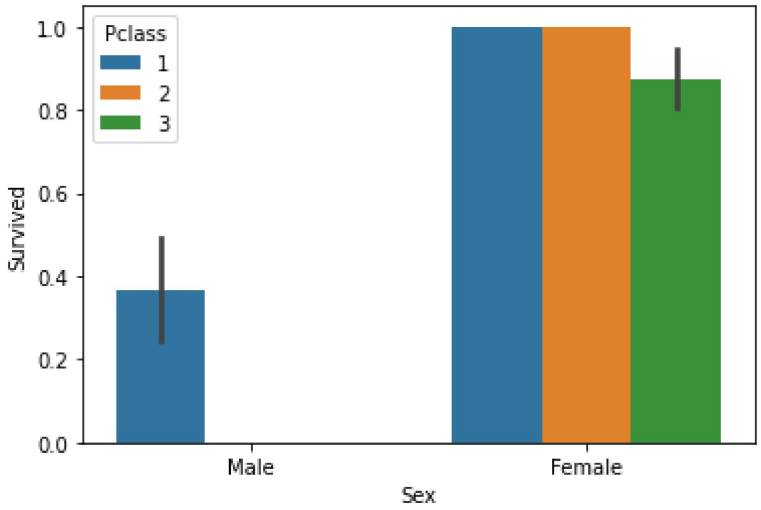
In [387]:
```
fig = plt.figure(figsize=(15,15))

ax1 = fig.add_subplot(221)
sns.boxplot(x='Pclass',y='Age',data=df_test)

ax2 = fig.add_subplot(222)
sns.boxplot(x='Sex',y='Fare',data=df_test)

ax3 = fig.add_subplot(223)
sns.boxplot(x='Pclass',y='Fare',data=df_test)

ax4 = fig.add_subplot(224)
sns.boxplot(x='Sex',y='Age',data=df_test)
```

Out[387]: <AxesSubplot:xlabel='Sex', ylabel='Age'>



In [388]:
```
sns.barplot(x = 'Sex', y = 'Survived', hue = 'Pclass', data = df_test)
```

Out[388]: <AxesSubplot:xlabel='Sex', ylabel='Survived'>



In [ ]:
```

```