Experiment 5: Titanic Kaggle Competition

Name: Anuj Chavan

UID: 2019120012

Class: BE EXTC

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

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
```

In [ ]:
```
train=pd.read_csv('../input/titanic/train.csv')
train.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [ ]:
```
train.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
```

In [ ]:
```
test=pd.read_csv('../input/titanic/test.csv')
test.head()
```

Out[ ]:

	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
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [ ]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
```

```
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

CLEANING TRAIN DATA

In [ ]:
```
print(train.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
```

In [ ]:
```
train.drop('Cabin',axis=1, inplace=True)
train.isnull().sum()
```

Out[ ]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Embarked         2
dtype: int64
```

In [ ]:
```
male_avg_age=train[train['Sex']=='male']['Age'].mean().round(0)
print("Average age of the male passenger:",male_avg_age)
female_avg_age=train[train['Sex']=='female']['Age'].mean().round(0)
print("Average age of the female passenger:",female_avg_age)
```

```
Average age of the male passenger: 31.0
Average age of the female passenger: 28.0
```

In [ ]:
```
print("Total null values in 'Age' column :",train['Age'].isnull().sum())
```

```
Total null values in 'Age' column : 177
```

In [ ]:
```
def new_age(df):
    age=df[1]
    sex=df[0]
    if pd.isnull(age):
        if sex=='male':
```

```
            return 31
        elif sex=='female':
            return 28
    else:
        return age
```

In [ ]:
```
train['Age']=train[['Sex','Age']].apply(new_age,axis=1)
```

In [ ]:
```
print("The total null values in 'Embarked' column :",train['Embarked'].isnull().sum())
print("\nThe value count in the column:\n",train['Embarked'].value_counts())
```

```
The total null values in 'Embarked' column : 2

The value count in the column:
 S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [ ]:
```
train['Embarked'].fillna('S',inplace=True)
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 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          891 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  Embarked     891 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

CLEANING TEST DATA

In [ ]:
```
print(test.isnull().sum())
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [ ]:

```
test.drop('Cabin',axis=1, inplace=True)
test.isnull().sum()
```

Out[ ]:
```
PassengerId     0
Pclass          0
Name            0
Sex             0
Age            86
SibSp           0
Parch           0
Ticket          0
Fare            1
Embarked        0
dtype: int64
```

In [ ]:
```
print("Average age of the male passenger:",male_avg_age)
print("Average age of the female passenger:",female_avg_age)
```

```
Average age of the male passenger: 31.0
Average age of the female passenger: 28.0
```

In [ ]:
```
print("Total null values in 'Age' column :",test['Age'].isnull().sum())
```

```
Total null values in 'Age' column : 86
```

In [ ]:
```
test['Age']=test[['Sex','Age']].apply(new_age,axis=1)
```

In [ ]:
```
print("Null values in 'Fare' column :", test['Fare'].isnull().sum())
```

```
Null values in 'Fare' column : 1
```

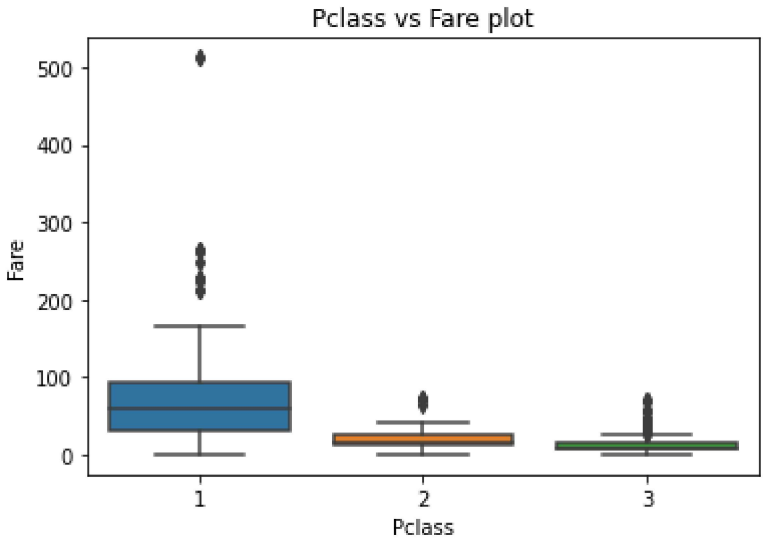
In [ ]:
```
test[test['Fare'].isnull()]
```

Out[ ]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
152	1044	3	Storey, Mr. Thomas	male	60.5	0	0	3701	NaN	S

In [ ]:
```
sns.boxplot(x=train['Pclass'],y=train['Fare'])
plt.title('Pclass vs Fare plot')
```

Out[ ]:
```
Text(0.5, 1.0, 'Pclass vs Fare plot')
```



In [ ]:
```
Fare_avg=train['Fare'].mean().round(4)
Fare_avg
```

Out[ ]: 32.2042

In [ ]:
```
test['Fare'].fillna(Fare_avg,inplace=True)
```

ML Model

//Deleting name and ticket column since redundant

In [ ]:
```
train.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	S

In [ ]:
```
train=pd.get_dummies(train.drop(['Name','Ticket'],axis=1),drop_first=True)
train.head()
```

Out[ ]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	1	0	3	22.0	1	0	7.2500	1	0	1
1	2	1	1	38.0	1	0	71.2833	0	0	0
2	3	1	3	26.0	0	0	7.9250	0	0	1
3	4	1	1	35.0	1	0	53.1000	0	0	1
4	5	0	3	35.0	0	0	8.0500	1	0	1

In [ ]:
```
test.head()
```

Out[ ]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	S

In [ ]:
```
test=pd.get_dummies(test.drop(['Name','Ticket'],axis=1),drop_first=True)
test.head()
```

Out[ ]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	892	3	34.5	0	0	7.8292	1	1	0
1	893	3	47.0	1	0	7.0000	0	0	1
2	894	2	62.0	0	0	9.6875	1	1	0
3	895	3	27.0	0	0	8.6625	1	0	1
4	896	3	22.0	1	1	12.2875	0	0	1

Model Building

In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,confusion_matrix, f1_score
```

In [ ]:
```
#Data split
X_train, X_test, y_train, y_test = train_test_split(train.drop('Survived',axis=1), trai
```

In [ ]:
```
random_forest = RandomForestClassifier(n_estimators=1000)
random_forest.fit(X_train, y_train)
predictions = random_forest.predict(X_test)
```

In [ ]:
```
print(confusion_matrix(y_test,predictions))
print('\n')
print(classification_report(y_test,predictions))
print("\n")
print("F1 Score :",f1_score(y_test,predictions))
```

```
[[91  8]
 [24 56]]


              precision    recall  f1-score   support

           0       0.79      0.92      0.85        99
           1       0.88      0.70      0.78        80

    accuracy                           0.82       179
   macro avg       0.83      0.81      0.81       179
weighted avg       0.83      0.82      0.82       179


F1 Score : 0.7777777777777777
```

In [ ]:
```
test.head()
```

Out[ ]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	892	3	34.5	0	0	7.8292	1	1	0
1	893	3	47.0	1	0	7.0000	0	0	1
2	894	2	62.0	0	0	9.6875	1	1	0
3	895	3	27.0	0	0	8.6625	1	0	1
4	896	3	22.0	1	1	12.2875	0	0	1

In [ ]:
```
test_predictions = random_forest.predict(test)
```

In [ ]:
```
test_predictions
```

Out[ ]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
```

```
1, 0, 1, 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, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0])
```

In [ ]:

```
df1=pd.DataFrame(test['PassengerId'],columns=['PassengerId'])
df2=pd.DataFrame(test_predictions,columns=['Survived'])
final_submission=pd.concat([df1,df2], axis=1)
final_submission
```

Out[ ]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	0
...	...	...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

418 rows × 2 columns

In [ ]:

```
final_submission.to_csv("TitanicPredictionsOptimized.csv",index=False)
```