In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv("C:\\Users\\DELL\\Downloads\\train.csv")
df.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	E
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 [3]:
```
df.shape
```

Out[3]:
```
(891, 12)
```

In [4]:
```
df.columns
```

Out[4]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
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 [6]:
```
df1=df.drop(['Name','Ticket','Cabin','PassengerId'],axis=1)
```

In [7]:
```
df1
```

Out[7]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S
887	1	1	female	19.0	0	0	30.0000	S
888	0	3	female	NaN	1	2	23.4500	S
889	1	1	male	26.0	0	0	30.0000	C
890	0	3	male	32.0	0	0	7.7500	Q

891 rows × 8 columns

In [ ]:
```
```

In [8]:
```
df1.head()
```

Out[8]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

In [9]:
```
df1.isnull().sum()
```

Out[9]:
```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
dtype: int64
```

In [10]:
```
df1['Age'].fillna(df1['Age'].median(),inplace=True)
mode=df['Embarked'].mode()[0]
df1['Embarked'].fillna(mode,inplace=True)
```

In [11]:
```
df1
```

Out[11]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S
887	1	1	female	19.0	0	0	30.0000	S
888	0	3	female	28.0	1	2	23.4500	S
889	1	1	male	26.0	0	0	30.0000	C
890	0	3	male	32.0	0	0	7.7500	Q

891 rows × 8 columns

In [12]:
```
df1.isnull().sum()
```

Out[12]:
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

In [13]:
```
df1=pd.get_dummies(df1,drop_first=True)
```

In [ ]:
```

```

In [14]:
```
df1.head()
```

Out[14]:

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

In [15]:
```
df1.shape
```

Out[15]:
```
(891, 9)
```

In [16]:
```
#Feature Scaling
```

In [17]:
```
from sklearn.preprocessing import MinMaxScaler
mm=MinMaxScaler()
df1['Age']=mm.fit_transform(df1[['Age']])
df1['Fare']=mm.fit_transform(df1[['Fare']])
```

In [18]:
```
df1
```

Out[18]:

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked_Q	Embarked_S
0	0	3	0.271174	1	0	0.014151	1	0	1
1	1	1	0.472229	1	0	0.139136	0	0	0
2	1	3	0.321438	0	0	0.015469	0	0	1
3	1	1	0.434531	1	0	0.103644	0	0	1
4	0	3	0.434531	0	0	0.015713	1	0	1
...	...	...	...	...	...	...	...	...	...
886	0	2	0.334004	0	0	0.025374	1	0	1
887	1	1	0.233476	0	0	0.058556	0	0	1
888	0	3	0.346569	1	2	0.045771	0	0	1
889	1	1	0.321438	0	0	0.058556	1	0	0
890	0	3	0.396833	0	0	0.015127	1	1	0

891 rows × 9 columns

In [19]:
```
df1.describe()
```

Out[19]:

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_male	Embarked
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.0000
mean	0.383838	2.308642	0.363679	0.523008	0.381594	0.062858	0.647587	0.0864
std	0.486592	0.836071	0.163605	1.102743	0.806057	0.096995	0.477990	0.2811
min	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	0.000000	2.000000	0.271174	0.000000	0.000000	0.015440	0.000000	0.0000
50%	0.000000	3.000000	0.346569	0.000000	0.000000	0.028213	1.000000	0.0000
75%	1.000000	3.000000	0.434531	1.000000	0.000000	0.060508	1.000000	0.0000
max	1.000000	3.000000	1.000000	8.000000	6.000000	1.000000	1.000000	1.0000

In [20]:
```
#Data modelling
```

In [21]:
```
x=df1.drop(['Survived'],axis=1)
y=df1['Survived']
```

In [22]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [23]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [24]:
```
model.fit(x_train,y_train)
```

Out[24]:
```
LogisticRegression()
```

In [25]:
```
train_pred=model.predict(x_train)
test_pred=model.predict(x_test)
```

In [26]:
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_train,train_pred))
print(accuracy_score(y_test,test_pred))
```

```
0.797752808988764
0.7988826815642458
```

In [27]:
```
from sklearn.model_selection import cross_val_score
scores=cross_val_score(model,x,y,cv=5)
scores.mean()
```

Out[27]:
```
0.7890025735986441
```

In [28]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

In [29]:
```
dt_model=DecisionTreeClassifier()
rf_model=RandomForestClassifier()
```

```
gb_model=GradientBoostingClassifier()
xg_model=XGBClassifier()
```

In [30]:
```
dt_model.fit(x_train,y_train)
rf_model.fit(x_train,y_train)
gb_model.fit(x_train,y_train)
xg_model.fit(x_train,y_train)
```

Out[30]:
```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, ...)
```

In [31]:
```
y_pred5=dt_model.predict(x_test)
y_pred6=rf_model.predict(x_test)
y_pred7=gb_model.predict(x_test)
y_pred8=xg_model.predict(x_test)
```

In [32]:
```
print(accuracy_score(y_test,y_pred5))
print(accuracy_score(y_test,y_pred6))
print(accuracy_score(y_test,y_pred7))
print(accuracy_score(y_test,y_pred8))
```

```
0.7877094972067039
0.8268156424581006
0.8435754189944135
0.8491620111731844
```

In [33]:
```
train_pred1=dt_model.predict(x_train)
train_pred2=rf_model.predict(x_train)
train_pred3=gb_model.predict(x_train)
train_pred4=xg_model.predict(x_train)
```

In [34]:
```
print(accuracy_score(y_train,train_pred1))
print(accuracy_score(y_train,train_pred2))
print(accuracy_score(y_train,train_pred3))
print(accuracy_score(y_train,train_pred4))
```

```
0.9817415730337079
0.9817415730337079
0.9002808988764045
0.9634831460674157
```

In [35]:
```
scores=cross_val_score(dt_model,x,y,cv=5)
scores.mean()
```

Out[35]:
```
0.7811876216182286
```

In [36]:
```
scores=cross_val_score(rf_model,x,y,cv=5)
scores.mean()
```

Out[36]:
```
0.80585022911305
```

In [37]:
```
scores=cross_val_score(gb_model,x,y,cv=5)
scores.mean()
```

Out[37]:
```
0.8226853304877284
```

In [38]:
```
scores=cross_val_score(xg_model,x,y,cv=5)
scores.mean()
```

Out[38]:
```
0.8137342288619672
```

In [39]:
```
from sklearn.model_selection import GridSearchCV
estimator=DecisionTreeClassifier(random_state=0)
param_grid={'criterion':['gini','entropy'],
              'max_depth':[1,2,3,4]}
grid=GridSearchCV(estimator,param_grid,scoring='accuracy',cv=5)
grid.fit(x_train,y_train)
grid.best_params_
```

Out[39]:
```
{'criterion': 'gini', 'max_depth': 4}
```

In [40]:
```
dt_bhp=DecisionTreeClassifier(criterion='gini',max_depth=3,random_state=4)
dt_bhp.fit(x_train,y_train)

ypred_train=dt_bhp.predict(x_train)
ypred_test=dt_bhp.predict(x_test)

print(accuracy_score(ypred_train,y_train))
print(accuracy_score(ypred_test,y_test))

scores=cross_val_score(dt_bhp,x,y,cv=5)
scores.mean()
```

Out[40]:
```
0.8342696629213483
0.8212290502793296
0.8103132257862029
```

In [41]:
```
from sklearn.model_selection import GridSearchCV
estimator=RandomForestClassifier(random_state=0)
param_grid={'n_estimators':list(range(1,101))}
grid=GridSearchCV(estimator,param_grid,scoring='accuracy',cv=5)
grid.fit(x_train,y_train)
grid.best_params_
```

Out[41]:
```
{'n_estimators': 7}
```

In [42]:
```
rf_bhp=RandomForestClassifier(n_estimators=7,random_state=0)
rf_bhp.fit(x_train,y_train)

ypred_train=rf_bhp.predict(x_train)
ypred_test=rf_bhp.predict(x_test)

print(accuracy_score(ypred_train,y_train))
print(accuracy_score(ypred_test,y_test))
```

```
scores=cross_val_score(rf_bhp,x,y,cv=5)
scores.mean()
```

Out[42]:
```
0.9578651685393258
0.7932960893854749
0.7811938986880924
```

In [43]:
```
from sklearn.model_selection import GridSearchCV
estimator=GradientBoostingClassifier()
param_grid={'n_estimators':[1,5,10,20,40,100],
            'learning_rate':[0.1,0.2,0.3,0.5,0.8,1]}
grid=GridSearchCV(estimator,param_grid,scoring='accuracy',cv=5)
grid.fit(x_train,y_train)
grid.best_params_
```

Out[43]:
```
{'learning_rate': 0.3, 'n_estimators': 20}
```

In [63]:
```
gb_bhp=GradientBoostingClassifier(n_estimators=20,learning_rate=0.3)
gb_bhp.fit(x_train,y_train)

ypred_train=gb_bhp.predict(x_train)
ypred_test=gb_bhp.predict(x_test)

print(accuracy_score(ypred_train,y_train))
print(accuracy_score(ypred_test,y_test))

scores=cross_val_score(gb_bhp,x,y,cv=5)
scores.mean()
```

Out[63]:
```
0.8834269662921348
0.8379888268156425
0.8125918021467579
```

In [64]:
```
from sklearn.model_selection import GridSearchCV
estimator=XGBClassifier()
param_grid={'n_estimators':[10,20,40,100],
            'max_depth':[3,4,5],
            'gamma':[0,0.15,0.3,0.5,1]}

grid=GridSearchCV(estimator,param_grid,scoring='accuracy',cv=5)
grid.fit(x_train,y_train)
grid.best_params_
```

Out[64]:
```
{'gamma': 0.15, 'max_depth': 5, 'n_estimators': 10}
```

In [46]:
```
xg_bhp=XGBClassifier(n_estimators=10,gamma=0.15,max_depth=5)
xg_bhp.fit(x_train,y_train)

ypred_train=xg_bhp.predict(x_train)
ypred_test=xg_bhp.predict(x_test)

print(accuracy_score(ypred_train,y_train))
print(accuracy_score(ypred_test,y_test))

scores=cross_val_score(xg_bhp,x,y,cv=5)
scores.mean()
```

```
0.8806179775280899
0.8379888268156425
0.8238340342728014
```

Out[46]:

In [47]:
```
df2=pd.read_csv("C:\\Users\\DELL\\Downloads\\test.csv")
df2
```

Out[47]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
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 [48]:
```
df2.shape
```

Out[48]:
```
(418, 11)
```

In [49]:
```
df2.isnull().sum()
```

Out[49]:
```
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 [50]:
```
df2=df2.drop(['Name','Ticket','Cabin','PassengerId'],axis=1)
```

In [ ]:
```

```

In [51]:
```
df2.isnull().sum()
```

Out[51]:
```
Pclass       0
Sex          0
Age         86
SibSp        0
Parch        0
Fare         1
Embarked     0
dtype: int64
```

In [52]:
```
df2['Age'].fillna(df2['Age'].median(),inplace=True)
mode=df['Embarked'].mode()[0]
df2['Embarked'].fillna(mode,inplace=True)
```

In [53]:
```
df2['Fare']=df2['Fare'].fillna(df2['Fare'].median())
```

In [54]:
```
df2=pd.get_dummies(df2,drop_first=True)
```

In [55]:
```
df2.isnull().sum()
```

Out[55]:
```
Pclass        0
Age           0
SibSp         0
Parch         0
Fare          0
Sex_male      0
Embarked_Q    0
Embarked_S    0
dtype: int64
```

In [56]:
```
from sklearn.preprocessing import MinMaxScaler
mm=MinMaxScaler()
df2['Age']=mm.fit_transform(df2[['Age']])
df2['Fare']=mm.fit_transform(df2[['Fare']])
```

In [57]:
```
df2.isnull().sum()
```

Out[57]:
```
Pclass        0
Age           0
SibSp         0
Parch         0
Fare          0
Sex_male      0
Embarked_Q    0
Embarked_S    0
dtype: int64
```

In [ ]:
```
```

In [58]:
```
df2.isnull().sum()
```

Out[58]:
```
Pclass        0
Age           0
SibSp         0
Parch         0
Fare          0
Sex_male      0
Embarked_Q    0
Embarked_S    0
dtype: int64
```

In [59]:
```
pred=xg_bhp.predict(df2)
pred
```

Out[59]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 0, 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, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 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, 0, 1, 0, 1, 0, 1, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 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, 1, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0])
```

In [60]:
```
df3=pd.read_csv("C:\\Users\\DELL\\Downloads\\gender_submission.csv")
df3
```

Out[60]:

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

418 rows × 2 columns

In [61]:
```
df3['Survived']=pred
df3.head()
```

Out[61]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	1

In [62]:
```
df3.to_csv('sub6.csv',index=False)
```

In [ ]:

In [ ]:

In [ ]: