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

```
train_data = pd.read_csv('/content/train.csv')
test_data = pd.read_csv('/content/test.csv')
```

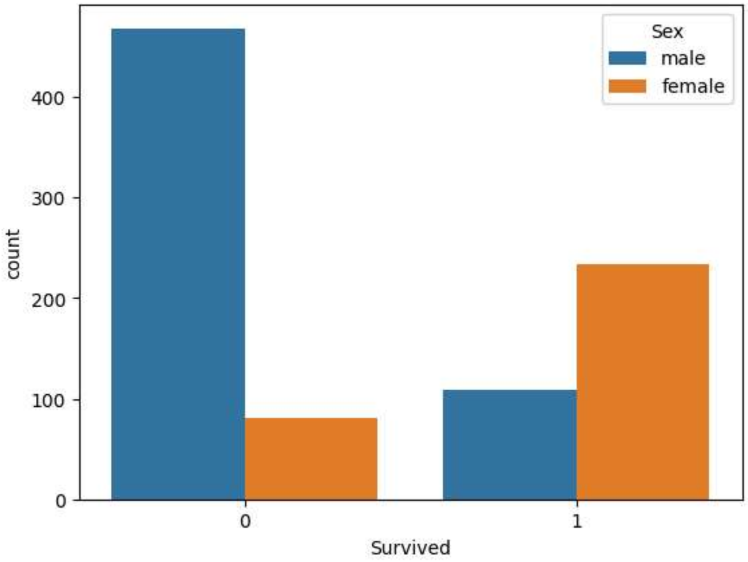
```
train_data.head()
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Far |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.250 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.283 |

```
train_data.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
```

```
sns.countplot(x='Survived',hue='Sex', data=train_data)
plt.show()
```



```
train_data['Age'].fillna(train_data['Age'].median(), inplace=True)
test_data['Age'].fillna(test_data['Age'].median(), inplace=True)
```

```
train_data['Embarked'].fillna(train_data['Embarked'].mode()[0], inplace=True)
test_data['Embarked'].fillna(test_data['Embarked'].mode()[0], inplace=True)
```

```
train_data.drop(['Cabin', 'Ticket'],axis=1, inplace=True)
test_data.drop(['Cabin', 'Ticket'],axis=1, inplace=True)
```

```
train_data = pd.get_dummies(train_data, columns=['Sex', 'Embarked'])
test_data = pd.get_dummies(test_data, columns=['Sex','Embarked'])
```

```
train_data, test_data = train_data.align(test_data, join='left',axis=1)
test_data.fillna(0, inplace=True)
```

```
X = train_data.drop(['Survived','Name','PassengerId'], axis=1)
y = train_data['Survived']
```

```
from sklearn.model_selection import train_test_split
x_train, X_val, y_train,y_val = train_test_split(X,y,test_size=0.2,random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier
model=  RandomForestClassifier (n_estimators=100,random_state=42)
model.fit(x_train,y_train)
```

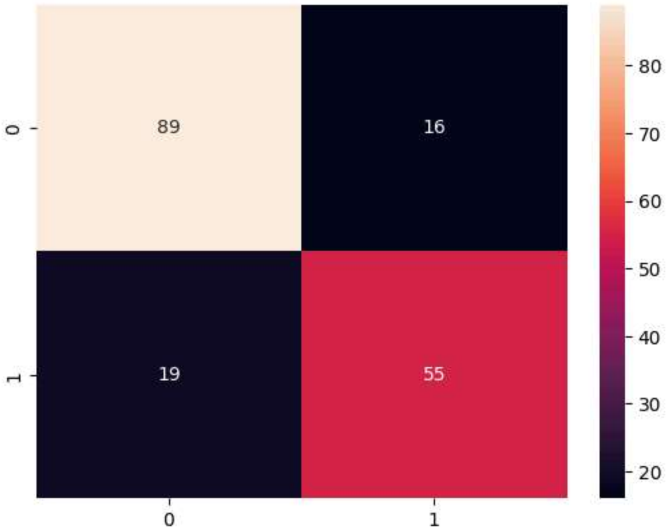
```
▾        RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
y_pred = model.predict(X_val)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
accuracy = accuracy_score(y_val, y_pred)
print(f'Validation_Accuracy: {accuracy:.4f}')

cm=confusion_matrix(y_val,y_pred)
sns.heatmap(cm,annot=True,fmt='d')
plt.show()
print(classification_report(y_val,y_pred))
```

```
Validation_Accuracy: 0.8045
```



```
              precision    recall  f1-score   support

           0       0.82      0.85      0.84       105
           1       0.77      0.74      0.76        74

    accuracy                           0.80       179
   macro avg       0.80      0.80      0.80       179
weighted avg       0.80      0.80      0.80       179
```

```
test_features = test_data.drop(['Name','PassengerId'],axis=1)
```

```
test_features = test_features.reindex(columns=x_train.columns,fill_value=0)
```

```
test_predictions = model.predict(test_features)


submission = pd.DataFrame({'PassengerId': test_data['PassengerId'],'Survived':test_predictions})


submission.to_csv('titanic_p.csv',index=False)

submission.head()
```

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

Next steps: Generate code with `submission` View recommended plots