

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

```
df=pd.read_csv("/content/gender_submission.csv")
```

```
df.head()
```

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

```
df.isnull().sum()
```

```
0
PassengerId  0
Survived     0
```

```
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

for col in df.select_dtypes(include=['object']).columns:
    data[col] = encoder.fit_transform(data[col])
```

```
x = df.iloc[:,0]
y = df.iloc[:,1]
```

```
x = np.array(x)
y = np.array(y)
```

```
x = x.reshape(-1,1)
y = y.reshape(-1,1)
```

```
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=42
)
```

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression(class_weight='balanced')
```

```
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:1408: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n,) or (n, 1) to avoid this warning.
y = column_or_1d(y, warn=True)
```

```
LogisticRegression
LogisticRegression(class_weight='balanced')
```

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

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy :", accuracy)
```

Accuracy : 0.5238095238095238

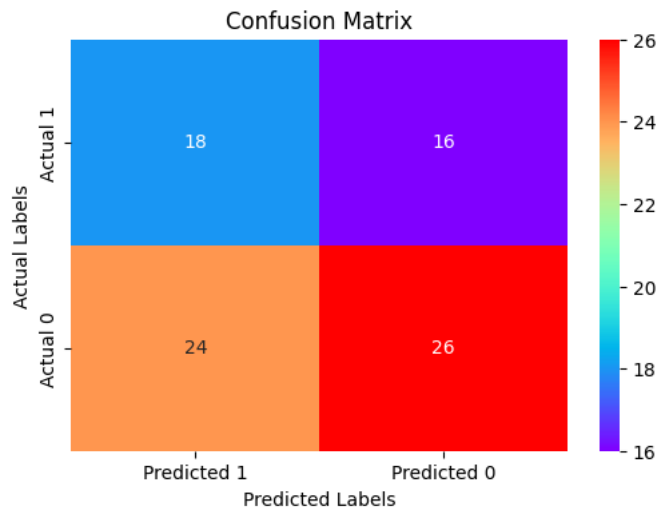
```
cm = confusion_matrix(y_test, y_pred, labels=[1, 0])
print("Confusion Matrix :")
print("TP FP")
print("FN TN")
print(cm)
```

Confusion Matrix :
TP FP
FN TN
[[18 16]
 [24 26]]

```
precision = precision_score(y_test, y_pred, zero_division=0)
print("Precision :", precision)
```

Precision : 0.42857142857142855

```
import seaborn as sns
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="rainbow",
            xticklabels=['Predicted 1', 'Predicted 0'],
            yticklabels=['Actual 1', 'Actual 0'])
plt.xlabel("Predicted Labels")
plt.ylabel("Actual Labels")
plt.title("Confusion Matrix")
plt.show()
```

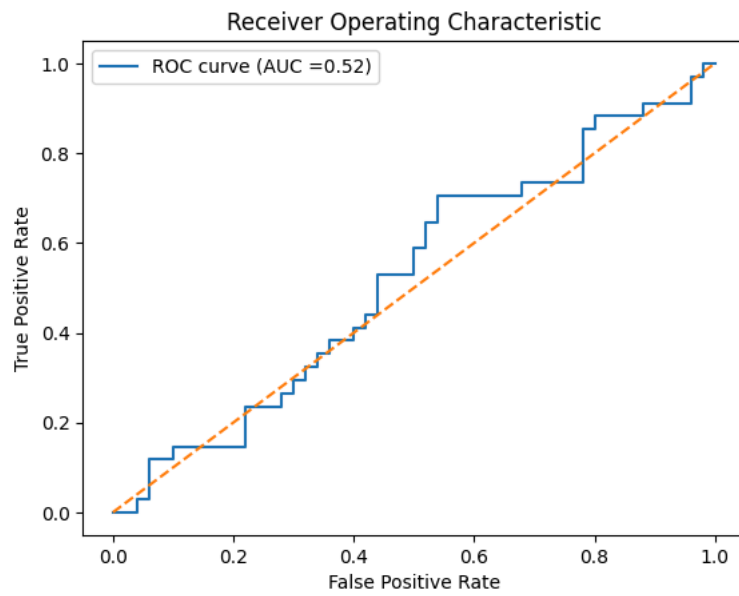


```
from sklearn.metrics import roc_curve, auc
```

```
y_prod = model.predict_proba(x_test)[:, 1]
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_prod)
```

```
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, label='ROC curve (AUC = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend()
plt.show()
```



Start coding or [generate](#) with AI.