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

#Now we will import the dataset and select only age and salary as the features
dataset = pd.read_csv("/content/sample_data/Social_Network_Ads.csv")
dataset.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19.0	19000.0	0
1	15810944	Male	35.0	20000.0	0
2	15668575	Female	26.0	43000.0	0
3	15603246	Female	27.0	57000.0	0
4	15804002	Male	19.0	76000.0	0

```
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

```
#Splitting data for training and testing
from sklearn.model_selection  import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state =
```

```
#Scale the features to avoid variation and let the features follow a normal distribution
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
#The preprocessing part is over. It is time to fit the model
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

     LogisticRegression(random_state=0)
```

```
#We fitted the model on training data. We will predict the labels of test data.
y_pred = classifier.predict(X_test)
```

```
#The prediction is over. Now we will evaluate the performance of our model.
from sklearn.metrics import confusion_matrix, classification_report
cm = confusion_matrix(y_test, y_pred)
print(cm,"\n")

     [[65  3]
      [ 8 24]]
```

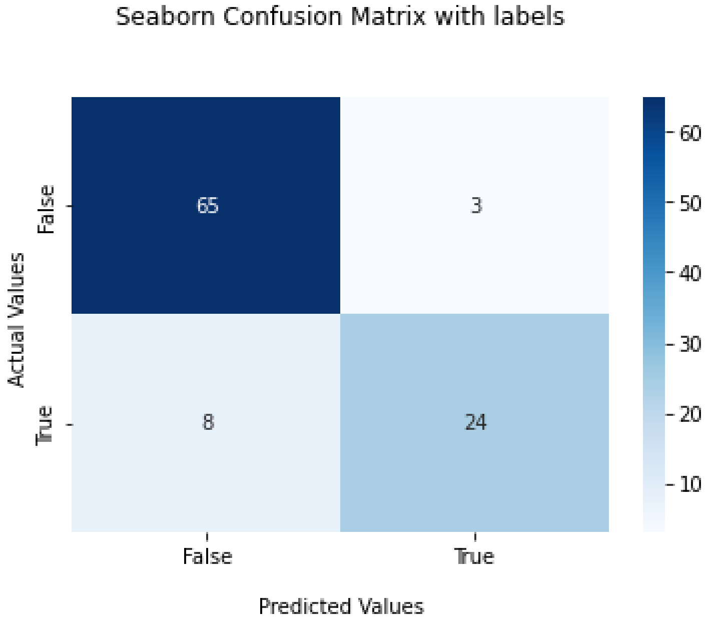
```
#Display TP,FP,TN,FN
tp=cm[1][1]
fp=cm[0][1]
tn=cm[0][0]
fn=cm[1][0]
print("\nTrue Positive :", tp)
print("False Positive:", fp)
print("True Negative :", tn)
print("False Negative:", fn)
```

```
     True Positive : 24
     False Positive: 3
     True Negative : 65
     False Negative: 8
```

```
import seaborn as sns
ax = sns.heatmap(cm, annot=True, cmap='Blues')
ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ');

# Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(['False','True'])
ax.yaxis.set_ticklabels(['False','True'])

# Display the visualization of the Confusion Matrix.
plt.show()
```



```
#Classification report displaying accuracy, precision,recall and fscore
print("\nClassification Report")
cl_report=classification_report(y_test,y_pred)
print("\n",cl_report)
```

```
Classification Report

              precision    recall  f1-score   support

           0       0.89      0.96      0.92        68
           1       0.89      0.75      0.81        32

    accuracy                           0.89       100
   macro avg       0.89      0.85      0.87       100
weighted avg       0.89      0.89      0.89       100
```

✓ 0s completed at 5:21 PM ● ✕