In [1]:
```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
```

In [2]:
```
dataset = pd.read_csv("Social_Network_Ads.csv")
print(dataset.head())
```

```
    User ID  Gender  Age  EstimatedSalary  Purchased
0  15624510    Male   19            19000          0
1  15810944    Male   35            20000          0
2  15668575  Female   26            43000          0
3  15603246  Female   27            57000          0
4  15804002    Male   19            76000          0
```

In [3]:
```
X = dataset[['Age', 'EstimatedSalary']].values
y = dataset['Purchased'].values
```

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=0
)
```

In [5]:
```
sc = StandardScaler()

X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [6]:
```
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

Out[6]:
```
LogisticRegression(random_state=0)
```

In [7]:
```
y_pred = classifier.predict(X_test)
```

In [8]:
```
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)
```

```
Confusion Matrix:
 [[65  3]
 [ 8 24]]
```

In [12]:
```
TN, FP, FN, TP = cm.ravel()

print("TN =", TN)
print("FP =", FP)
print("FN =", FN)
print("TP =", TP)
```

```
TN = 65
FP = 3
FN = 8
TP = 24
```

In [10]:
```
accuracy = (TP + TN) / (TP + TN + FP + FN)
error_rate = 1 - accuracy
precision = TP / (TP + FP)
recall = TP / (TP + FN)

print("\n--- Metrics ---")
print("Accuracy =", accuracy)
print("Error Rate =", error_rate)
print("Precision =", precision)
print("Recall =", recall)
accuracy = (TP + TN) / (TP + TN + FP + FN)
error_rate = 1 - accuracy
precision = TP / (TP + FP)
recall = TP / (TP + FN)

print("\n--- Metrics ---")
print("Accuracy =", accuracy)
print("Error Rate =", error_rate)
print("Precision =", precision)
print("Recall =", recall)
```

```
--- Metrics ---
Accuracy = 0.89
Error Rate = 0.10999999999999999
Precision = 0.8888888888888888
Recall = 0.75
```