In [1]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,classification_report

import pandas as pd
```

In [2]:
```
df=pd.read_csv('Social_Network_Ads.csv')
```

In [3]:
```
data=pd.DataFrame(df)
print(data)
```

```
      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
..        ...     ...  ...              ...        ...
395  15691863  Female   46            41000          1
396  15706071    Male   51            23000          1
397  15654296  Female   50            20000          1
398  15755018    Male   36            33000          0
399  15594041  Female   49            36000          1

[400 rows x 5 columns]
```

In [4]:
```
X=df[['EstimatedSalary']]
y=df['Purchased']
```

In [5]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=30)
```

In [6]:
```
model=LogisticRegression()
model.fit(X_train,y_train)
```

Out[6]:
```
LogisticRegression()
```

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

In [8]:
```
print(y_pred)
```

```
[0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0
 0 0 1 0 0 0]
```

In [9]:
```
cm=confusion_matrix(y_test,y_pred)
```

In [10]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support
```

```
                0       0.71      1.00      0.83        49
                1       1.00      0.35      0.52        31

         accuracy                           0.75        80
        macro avg       0.86      0.68      0.68        80
     weighted avg       0.82      0.75      0.71        80
```

In [11]:

```
TP = cm[0, 0]
FP = cm[0, 1]
TN = cm[0, 0]
FN = cm[1, 0]
```

In [12]:

```
# Compute Accuracy, Error rate, Precision, Recall
accuracy = (TP + TN) / (TP + FP + TN + FN)
error_rate = 1 - accuracy
precision = TP / (TP + FP)
recall = TP / (TP + FN)
```

In [13]:

```
# Print the evaluation metrics
print("Accuracy:", accuracy)
print("Error Rate:", error_rate)
print("Precision:", precision)
print("Recall:", recall)
report =classification_report(y_test,y_pred,zero_division=1)
print(cm)
```

```
Accuracy: 0.8305084745762712
Error Rate: 0.1694915254237288
Precision: 1.0
Recall: 0.7101449275362319
[[49  0]
 [20 11]]
```

In [ ]: