In [15]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
```

In [28]:
```
df = pd.read_csv('Social_Network_Ads.csv')
```

In [29]:
```
df
```

Out[29]:

	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 × 5 columns

In [30]:
```
df.isna().sum()
df.drop("Gender", axis=1, inplace=True)
```

In [31]:
```
# spliting dataset into test and train sets
x = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.20, ra
```

In [12]:
```
x
```

Out[12]:
```
array([[15624510, 'Male', 19, 19000],
       [15810944, 'Male', 35, 20000],
       [15668575, 'Female', 26, 43000],
       ...,
       [15654296, 'Female', 50, 20000],
       [15755018, 'Male', 36, 33000],
       [15594041, 'Female', 49, 36000]], dtype=object)
```

In [32]:
```
y
```

Out[32]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1], dtype=int64)
```

In [33]:
```
# Train logistic regression model on training set
classifier = LogisticRegression(random_state=0)
classifier.fit(x_train, y_train)
```

Out[33]:
```
LogisticRegression
LogisticRegression(random_state=0)
```

In [40]:
```
y_pred = classifier.predict(x_test)
y_pred_df = pd.DataFrame(y_pred, columns=['Purchase'])
```

In [42]:
```
y_pred_df
```

Out[42]:

	Purchase
0	0
1	0
2	0
3	0
4	0
...	...
75	0
76	0
77	0
78	0
79	1

80 rows × 1 columns

In [43]:
```
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[43]:
```
array([[56,  2],
       [12, 10]], dtype=int64)
```

In [44]:
```
# Extract true positives (TP), false positives (FP), true negatives (TN), fals
TP = cm[1, 1]
FP = cm[0, 1]
TN = cm[0, 0]
FN = cm[1, 0]
```

In [45]:
```
accuracy = accuracy_score(y_test, y_pred)
error_rate = 1 - accuracy
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

In [46]:
```
cm
```

Out[46]:
```
array([[56,  2],
       [12, 10]], dtype=int64)
```

In [47]:
```
TP
```

Out[47]:
```
10
```

In [48]:
```
FP
```

Out[48]:
```
2
```

In [49]:
```
TN
```

Out[49]:
```
56
```

In [50]:
```
FN
```

Out[50]:
```
12
```

In [51]:
```
accuracy
```

Out[51]:
```
0.825
```

In [52]:
```
error_rate
```

Out[52]:
```
0.17500000000000004
```

In [53]:
```
precision
```

Out[53]:
```
0.8333333333333334
```

In [54]:
```
recall
```

Out[54]:
```
0.45454545454545453
```

In [55]:
```
f1
```

Out[55]:
```
0.5882352941176471
```

In [ ]:
```

```