```
# import libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

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,
accuracy_score,precision_score, recall_score

# load dataset
data = pd.read_csv('Social_Network_Ads.csv')

data.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

```
# drop unecssary colms only age and salary for x any purchased for y

X = data[['Age','EstimatedSalary']]
y = data['Purchased']

# split into train and test sets 75% training 25% testing

X_train, X_test , y_train, y_test  = train_test_split(X,y,test_size =
.25 , random_state = 0)

#feature scaling helps perform logistic regression better
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# train logistic regression model

classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train,y_train)
```

```
LogisticRegression(random_state=0)
```

```
# predict on test set
y_pred = classifier.predict(X_test)

# confusion matrix
cm = confusion_matrix(y_test,y_pred)
```

```python
#Extract TP, FP, TN, FN

TN, FP, FN, TP = cm.ravel()

#Metrics
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)

print(f"Confusion matrix: \n {cm}" )
print(f"True positive (TP): {TP}")
print(f"False positive (FP) : {FP}")
print(f"True Negative (TN) : {TN}")
print(f"Fase neagative (FN) : {FN}")
print(f"Accuracy: {accuracy}")
print(f"Error Rate: {error_rate}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
```

```
Confusion matrix:
 [[65  3]
 [ 8 24]]
True positive (TP): 24
False positive (FP) : 3
True Negative (TN) : 65
Fase neagative (FN) : 8
Accuracy: 0.89
Error Rate: 0.10999999999999999
Precision: 0.8888888888888888
Recall: 0.75
```