

Name : Shivam Indrabhan Borse

Roll No : 21119

Subject: Software Laboratory III (DATA SCIENCE)

Assignment No : 05

Problem statement:

Data Analytics II

1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

CODE :

----- Assignment No : 05 -----

Data Analytics II

1. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report, confusion_matrix
```

[47]

```
# Load the dataset
data = pd.read_csv('Social_Network_Ads.csv')
data
```

[48]

	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

```
data['Gender'] = pd.get_dummies(data['Gender'], drop_first=True)
X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

[3] ✓ 0.0s

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Train the logistic regression model
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

[4] ✓ 0.0s

... LogisticRegression(random_state=0)

+ Code

+ Markdown

```
# Predict the test set results
y_pred = classifier.predict(X_test)
y_pred
```

[5] ✓ 0.0s

... array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1], dtype=int64)

```
print(classification_report(y_test, y_pred))
```

[6] ✓ 0.0s

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

      0       0.82      0.97      0.89        58
      1       0.83      0.45      0.59        22

 accuracy      0.82          80
 macro avg       0.83      0.71      0.74          80
 weighted avg       0.83      0.82      0.81          80
```

+ Code

+ Markdown

```
# Compute the confusion matrix
cm = confusion_matrix(y_test, y_pred)
print('Confusion matrix:')
print(cm)
```

[7] ✓ 0.0s

```
... Confusion matrix:
[[56  2]
 [12 10]]
```

```
# Compute the accuracy, error rate, precision, recall, and F1-score
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)

print('Accuracy: ', accuracy)
print('Error rate: ', error_rate)
print('Precision: ', precision)
print('Recall: ', recall)
print('F1-score: ', f1)
```

[8] ✓ 0.0s

```
--- Accuracy: 0.825
Error rate: 0.17500000000000004
Precision: 0.8333333333333334
Recall: 0.45454545454545453
F1-score: 0.5882352941176471
```



--> SHIVAM BORSE