five-tcod74

April 20, 2024

Assignment 5 TCOD74 - Bendre Anushka A.

[7]:

```
import pandas as pd
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
```

[9]:

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

[10]:

```
# Display the first few rows of the dataset
print("First few rows of the dataset:")
print(data.head())
```

```
First few rows of the dataset:
    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
```

[23]:

```
# Separate features and target variable
X = data.iloc[:, [2, 3]].values
y = data.iloc[:, 4].values
```

[24]:

```
# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,␣
  ↪random_state=0)
```

[25]:

```
# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[26]:

```python
# Logistic Regression model
classifier = LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

[26]:

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

[27]:

```python
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

[28]:

```python
# Compute Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(conf_matrix)
```

```

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

[29]:

```python
# Calculate TP, FP, TN, FN
TP = conf_matrix[1, 1]
FP = conf_matrix[0, 1]
TN = conf_matrix[0, 0]
FN = conf_matrix[1, 0]
```

[30]:

```python
# Compute Accuracy
accuracy = accuracy_score(y_test, y_pred)
```

[31]:

```python
# Compute Error Rate
error_rate = 1 - accuracy
```

[32]:

```python
# Compute Precision
precision = precision_score(y_test, y_pred)
```

[33]:

```python
# Compute Recall
recall = recall_score(y_test, y_pred)
```

[34]:

```python
# Print Metrics
print("\nAccuracy:", accuracy)
print("Error Rate:", error_rate)
print("Precision:", precision)
print("Recall:", recall)
```

```

Accuracy: 0.89
Error Rate: 0.10999999999999999
Precision: 0.8888888888888888
```

```
Recall: 0.75
```

```
[ ]:
```