logistic-regression

April 4, 2024

0.1 To predict whether a person will purchase from Social Network Ads

[59]:
```
## Importing Necessary Libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,␣
  ↪confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
```

[60]:
```
## Loading dataset
data = pd.read_csv(r'C:\Users\ntpc\Desktop\Social_Network_Ads.csv')
data.head()
```

[60]:
```
    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
```

[61]:
```
data.dtypes
```

[61]:
```
User ID             int64
Gender             object
Age                 int64
EstimatedSalary     int64
Purchased           int64
dtype: object
```

[62]:
```
#checking each value counts
data['Purchased'].value_counts()
```

[62]:
```
Purchased
0    257
1    143
Name: count, dtype: int64
```

[63]:
```
#dropping User ID column
data = data.drop('User ID',axis = 1)
data.head()
```

[63]:
```
   Gender  Age  EstimatedSalary  Purchased
0    Male   19            19000          0
1    Male   35            20000          0
2  Female   26            43000          0
3  Female   27            57000          0
4    Male   19            76000          0
```

[64]:
```
# Encoding categorical variables
le = LabelEncoder()
data['Gender'] = le.fit_transform(data['Gender'])
```

[65]:
```
data.head()
```

[65]:
```
   Gender  Age  EstimatedSalary  Purchased
0       1   19            19000          0
1       1   35            20000          0
2       0   26            43000          0
3       0   27            57000          0
4       1   19            76000          0
```

[66]:
```
# Splitting the data into training and testing sets
X = data.drop('Purchased', axis=1)
y = data['Purchased']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[67]:
```
## Doing Feature Scaling for better accuracy
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[68]:
```
model = LogisticRegression()
```

[69]:
```
model.fit(X_train,y_train)
```

[69]:
```
LogisticRegression()
```

[70]:
```
# Making predictions on the testing data
y_pred = model.predict(X_test)
```

[71]:
```
# Evaluating the model
accuracy = accuracy_score(y_test, y_pred)
```

```python
print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.8875
```

[72]:

```python
# Printing classification report and confusion matrix
print('Classification Report:')
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.96      0.92        52
           1       0.91      0.75      0.82        28

    accuracy                           0.89        80
   macro avg       0.90      0.86      0.87        80
weighted avg       0.89      0.89      0.88        80
```

[73]:

```python
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
[[50  2]
 [ 7 21]]
```