logistic_reg

May 11, 2022

[2]:
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

[4]:
```
# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
print(dataset)
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

```
      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 x 5 columns]
```

[3]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,␣
 ↪random_state = 0)
```

[4]:
```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
[5]: # Fitting Logistic Regression to the Training set
     from sklearn.linear_model import LogisticRegression
     log_reg = LogisticRegression(random_state = 0)
     log_reg.fit(X_train, y_train)
```

```
[5]: LogisticRegression(random_state=0)
```

```
[7]: # Predicting the Test set results
     y_pred = log_reg.predict(X_test)
```

```
[10]: cmp = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
      cmp.head(20)
```

```
[10]:     Real Values  Predicted Values
      0             0                 0
      1             0                 0
      2             0                 0
      3             0                 0
      4             0                 0
      5             0                 0
      6             0                 0
      7             1                 1
      8             0                 0
      9             0                 1
      10            0                 0
      11            0                 0
      12            0                 0
      13            0                 0
      14            0                 0
      15            0                 0
      16            0                 0
      17            0                 0
      18            1                 1
      19            0                 0
```

```
[12]: # Making the Confusion Matrix
      from sklearn.metrics import confusion_matrix
      cm = confusion_matrix(y_test, y_pred)
      cm
```

```
[12]: array([[65,  3],
             [ 8, 24]])
```

```
[14]: from sklearn.metrics import accuracy_score
      print ("Accuracy : ", accuracy_score(y_test, y_pred))
```

```
Accuracy :  0.89
```

[15]:
```
from sklearn.metrics import precision_score
print ("precision : ",precision_score(y_test, y_pred,average='macro'))
```

```
precision :  0.8896499238964992
```

[16]:
```
from sklearn.metrics import recall_score
print ("recall : ",recall_score(y_test, y_pred,average='macro'))
```

```
recall :  0.8529411764705883
```

[ ]:
```
```