pr5

May 4, 2024

[29]:
```
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, classification_report
```

[10]:
```
data=pd.read_csv("C:\\Users\\nayan\\Downloads\\Social_Network_Ads.csv")
data['Gender'].replace({'Male':0,'Female':1},inplace=True)
data
```

[10]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	0	19	19000	0
1	15810944	0	35	20000	0
2	15668575	1	26	43000	0
3	15603246	1	27	57000	0
4	15804002	0	19	76000	0
..	…	…	…	…	…
395	15691863	1	46	41000	1
396	15706071	0	51	23000	1
397	15654296	1	50	20000	1
398	15755018	0	36	33000	0
399	15594041	1	49	36000	1

[400 rows x 5 columns]

[3]:
```
data.head()
```

[3]:

	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

[12]:
```
data.columns
```

[12]:
```
Index(['User ID', 'Gender', 'Age', 'EstimatedSalary', 'Purchased'],
dtype='object')
```

[13]:
```
x=data[['User ID', 'Gender', 'Age', 'EstimatedSalary']]
y=data['Purchased']
```

[14]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
↪25,random_state=29)
```

[16]:
```
model = LogisticRegression()
model.fit(x_train, y_train)
```

[16]:
```
LogisticRegression()
```

[17]:
```
y_pred=model.predict(x_test)
```

[18]:
```
y_pred
```

[18]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0], dtype=int64)
```

[19]:
```
model.score(x_train,y_train)
```

[19]:
```
0.7833333333333333
```

[20]:
```
model.score(x,y)
```

[20]:
```
0.785
```

[21]:
```
cm=confusion_matrix(y_test,y_pred)
cm
```

[21]:
```
array([[64,  5],
       [16, 15]], dtype=int64)
```

[22]:
```
tn, fp, fn, tp = cm.ravel()
```

[23]:
```
print(tn, fp, fn, tp)
```

```
64 5 16 15
```

[24]:
```
a = accuracy_score(y_test,y_pred)
a
```

[24]: 0.79

[25]:
```
#error rate
e = 1 - a
e
```

[25]: 0.20999999999999996

[26]:
```
precision_score(y_test, y_pred)
```

[26]: 0.75

[27]:
```
recall_score(y_test, y_pred)
```

[27]: 0.4838709677419355

[30]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.80      0.93      0.86        69
           1       0.75      0.48      0.59        31

    accuracy                           0.79       100
   macro avg       0.78      0.71      0.72       100
weighted avg       0.78      0.79      0.78       100
```

[ ]: