EXPERIMENT 9

Aim:

To build a Logistic Regression model using the Social Network Ads dataset to predict whether a user will purchase a product based on their age and estimated salary.

Algorithm:

1. Import necessary libraries such as NumPy, Pandas, and Scikit-learn.
2. Load the dataset using Pandas.
3. Display the dataset and inspect the first few records.
4. Extract the feature columns (Age, Estimated Salary) and label (Purchased).
5. Split the dataset into training and testing sets using train_test_split().
6. Train the Logistic Regression model using the training data.
7. Evaluate the model accuracy using score() on both training and testing datasets.
8. Identify the best random state by iterating over multiple random states to maximize test accuracy.
9. Display the classification report to evaluate performance using precision, recall, and F1-score.

Code:

```
import numpy as np

import pandas as pd

df = pd.read_csv('Social_Network_Ads - Social_Network_Ads.csv')

df
```

Output:

```
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      230000          1
397 15654296   Female  50      20000           0
398 15755018   Male    36      33000          0
399 15594041   Female  49      36000           1
400 rows × 5 columns
```

```
df.head()
```

Output:

```
  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
```

```
features = df.iloc[:, [2, 3]].values
label = df.iloc[:, 4].values
features
```

Output:

```
array([[   19, 19000],
      [   35, 20000],
      [   26, 43000],
      [   27, 57000],
      [   19, 76000],
      [   27, 58000],
      [   27, 84000],
      [   32, 150000],
      [   25, 33000],
      [   35, 65000],
      [   26, 80000],
      [   26, 52000],
      [   20, 86000],
      [   32, 18000],
      [   18, 82000],
      [   29, 80000],
      [   47, 25000],
      [   45, 26000],
      [   46, 28000],
      [   48, 29000],
      ...
      [   49, 36000]])
```

```
label
```

Output:

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 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, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 0, 1])
```

```
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression


for i in range(1, 401):

  x_train, x_test, y_train, y_test = train_test_split(features, label, test_size=0.2, random_state=i)

  model = LogisticRegression()

  model.fit(x_train, y_train)

  train_score = model.score(x_train, y_train)

  test_score = model.score(x_test, y_test)

  if test_score > train_score:

    print("Test {} Train{} Random State {}".format(test_score, train_score, i))
```

Output:

```
Test 0.9 Train0.840625 Random State 4

Test 0.8625 Train0.85 Random State 5

Test 0.8625 Train0.859375 Random State 6

Test 0.8875 Train0.8375 Random State 7

Test 0.8625 Train0.8375 Random State 9

...

Test 0.95 Train0.81875 Random State 352

Test 0.9625 Train0.81875 Random State 217

Test 0.9375 Train0.821875 Random State 364
```

```
x_train, x_test, y_train, y_test = train_test_split(features, label, test_size=0.2, random_state=42)

finalModel = LogisticRegression()

finalModel.fit(x_train, y_train)
```

```
print(finalModel.score(x_train, y_train))

print(finalModel.score(x_test, y_test))
```

Output:

```
0.8375
```

```
0.8875
```

```
from sklearn.metrics import classification_report

print(classification_report(label, finalModel.predict(features)))
```

Output:

```
          precision   recall  f1-score   support


        0     0.85     0.93     0.89      257

        1     0.85     0.70     0.77      143


   accuracy                    0.85     400

  macro avg     0.85     0.81     0.83      400

weighted avg     0.85     0.85     0.84      400
```

Result:

Thus, the Logistic Regression model was successfully trained and tested using the Social Network Ads dataset.
The model achieved an accuracy of **88.75%** on the test data and **85% overall**, effectively predicting whether users are likely to purchase a product based on their age and estimated salary.