Implementation of Logistic Regression Model to Predict the Placement Status of Student

Aim:

To write a program to implement the the Logistic Regression Model to Predict the Placement Status of Student.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner

Algorithm

1. Import the standard libraries.
2. Upload the dataset and check for any null or duplicated values using .isnull() and .duplicated() function respectively.
3. Import LabelEncoder and encode the dataset.
4. Import LogisticRegression from sklearn and apply the model on the dataset.
5. Predict the values of array.
6. Calculate the accuracy, confusion and classification report by importing the required modules from sklearn.
7. Apply new unknown values

Program:

```
/*
Program to implement the the Logistic Regression Model to Predict the Placement Status of
Student.
Developed by: SANJAI A
RegisterNumber:  212220040142
*/
import pandas as pd
data=pd.read_csv("Placement_Data.csv")
data.head()
data1=data.copy()
data1=data1.drop(["sl_no","salary"],axis=1)
data1.head()
data1=data.copy()
data1=data1.drop(["sl_no","salary"],axis=1)
data1.head()
data.isnull().sum()
```

```python
data1.head()
data.duplicated().sum()
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data1["gender"]=le.fit_transform(data1["gender"])
data1["ssc_b"]=le.fit_transform(data1["ssc_b"])
data1["hsc_b"]=le.fit_transform(data1["hsc_b"])
data1["hsc_s"]=le.fit_transform(data1["hsc_s"])
data1["degree_t"]=le.fit_transform(data1["degree_t"])
data1["workex"]=le.fit_transform(data1["workex"])
data1["specialisation"]=le.fit_transform(data1["specialisation"])
data1["status"]=le.fit_transform(data1["status"])
print (data1)
x=data1.iloc[:,:-1]
y=data1["status"]
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
from sklearn.linear_model import LogisticRegression
c=LogisticRegression(solver="liblinear")
c.fit(x_train,y_train)
y_pred=c.predict(x_test)
print(y_pred)
from sklearn.metrics import accuracy_score
acur=accuracy_score(y_test,y_pred)
acur
from sklearn.metrics import confusion_matrix
con=confusion_matrix(y_test,y_pred)
print(con)
from sklearn.metrics import classification_report
class_report=classification_report(y_test,y_pred)
print(class_report)
print(c.predict([[1,80,1,90,1,1,90,1,0,85,1,85]]))
```

Output:

```
      gender  ssc_p  ssc_b  hsc_p  hsc_b  hsc_s  degree_p  degree_t  workex  \
0          1  67.00      1  91.00      1      1     58.00         2       0
1          1  79.33      0  78.33      1      2     77.48         2       1
2          1  65.00      0  68.00      0      0     64.00         0       0
3          1  56.00      0  52.00      0      2     52.00         2       0
4          1  85.80      0  73.60      0      1     73.30         0       0
..       ...    ...    ...    ...    ...    ...       ...       ...     ...
210        1  80.60      1  82.00      1      1     77.60         0       0
211        1  58.00      1  60.00      1      2     72.00         2       0
212        1  67.00      1  67.00      1      1     73.00         0       1
213        0  74.00      1  66.00      1      1     58.00         0       0
214        1  62.00      0  58.00      1      2     53.00         0       0

     etest_p  specialisation  mba_p  status
0       55.0               1  58.80       1
1       86.5               0  66.28       1
2       75.0               0  57.80       1
3       66.0               1  59.43       0
4       96.8               0  55.50       1
..       ...             ...    ...     ...
210     91.0               0  74.49       1
211     74.0               0  53.62       1
212     59.0               0  69.72       1
213     70.0               1  60.23       1
214     89.0               1  60.22       0

[215 rows x 13 columns]
[0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1
 1 1 1 0 0 1]
[[11  5]
 [ 3 24]]
              precision    recall  f1-score   support

           0       0.79      0.69      0.73        16
           1       0.83      0.89      0.86        27

    accuracy                           0.81        43
   macro avg       0.81      0.79      0.80        43
weighted avg       0.81      0.81      0.81        43
```

Result:

Thus the program to implement the the Logistic Regression Model to Predict the Placement Status of Student is written and verified using python programming.