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 / Jupyter notebook

Algorithm

Step1:

Import the standard libraries.

Step2:

Upload the dataset and check for any null or duplicated values using .isnull() and .duplicated() function respectively.

Step3:

LabelEncoder and encode the dataset.

Step4:

Import LogisticRegression from sklearn and apply the model on the dataset.

Step5:

Predict the values of array.

Step6:

Calculate the accuracy, confusion and classification report by importing the required modules from sklearn.

Step7:

Apply new unknown values

Program:

```
/*
Program to implement the the Logistic Regression Model to Predict the Placement Status of
Student.
Developed by: Manoj Kumar S
RegisterNumber: 212221230056
*/
```

import pandas as pd data=pd.read_csv("/content/Placement_Data.csv") data.head()

data1=data.copy() data1=data1.drop(["sl_no","salary"],axis=1) data1.head()

data1.isnull().sum()

data1.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] x

y=data1["status"] y

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 lr = LogisticRegression(solver = "liblinear") lr.fit(x_train,y_train) y_pred = lr.predict(x_test) y_pred

from sklearn.metrics import accuracy_score accuracy=accuracy_score(y_test,y_pred) accuracy

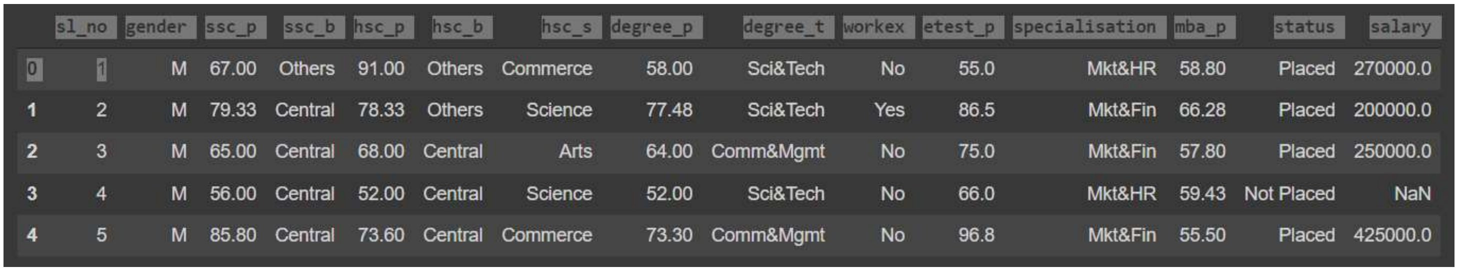
from sklearn.metrics import confusion_matrix confusion=confusion_matrix(y_test,y_pred) confusion

from sklearn.metrics import classification_report classification_report1=classification_report(y_test,y_pred) print(classification_report1)

```
lr.predict([[1,80,1,90,1,1,90,1,0,85,1,85]])
```

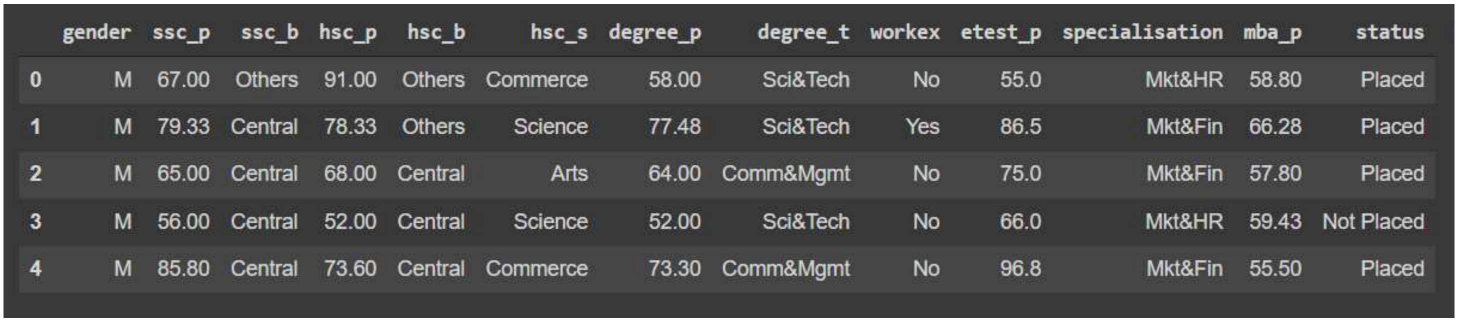
Output:

Original data(first five columns):



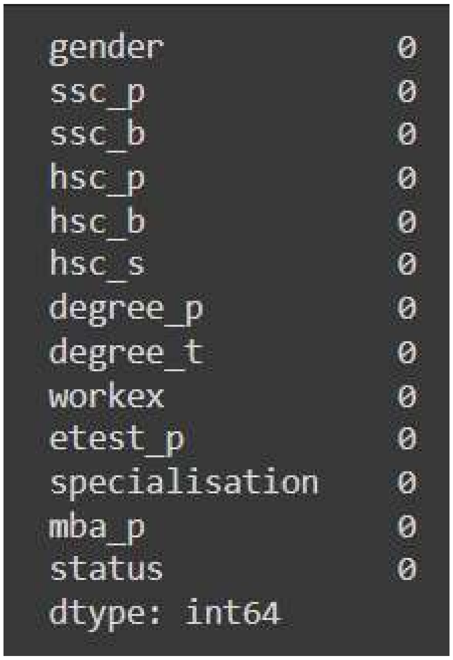
	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

Data after dropping unwanted columns(first five):



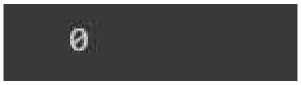
	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status
0	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed

Checking the presence of null values:



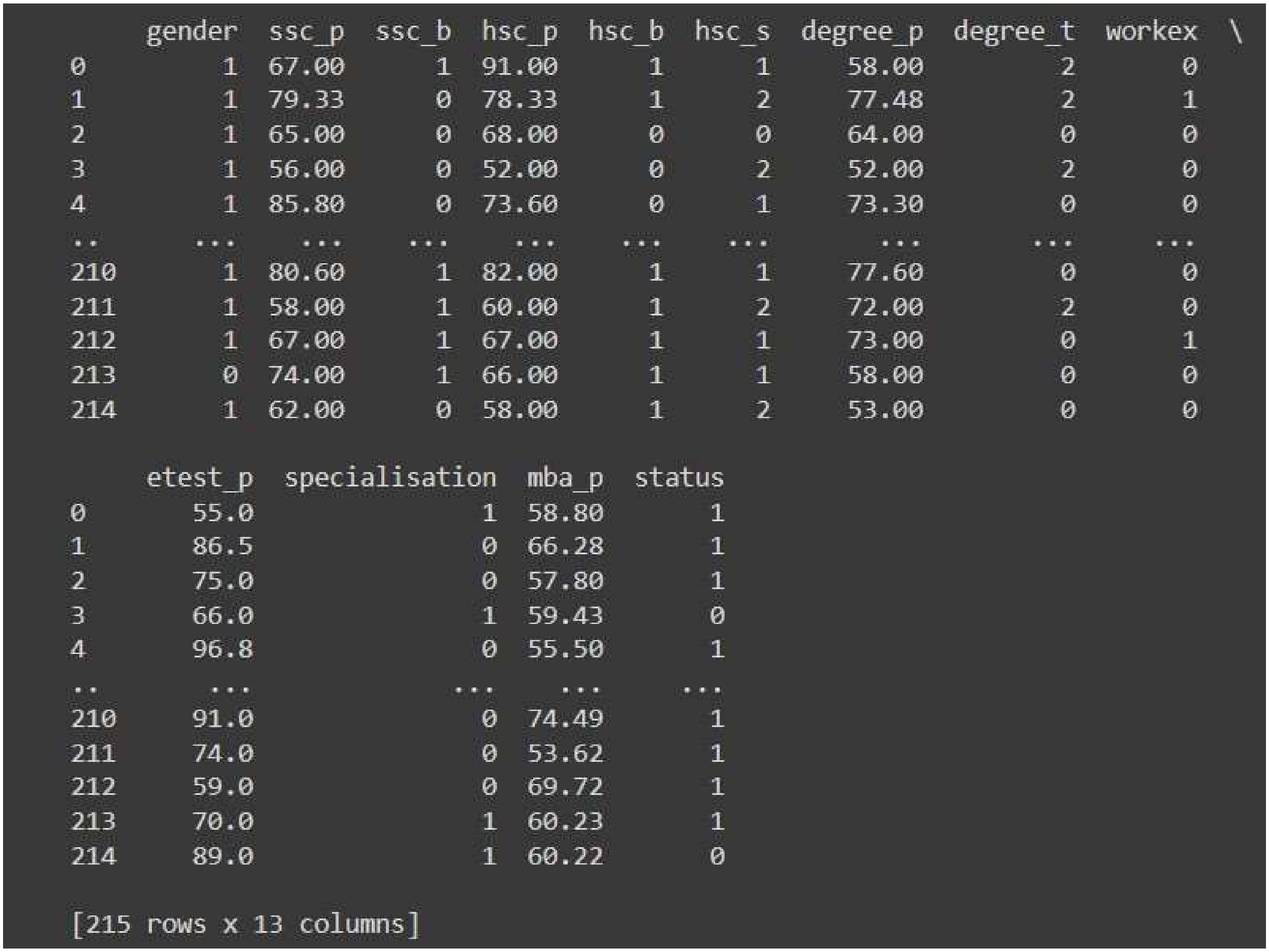
```
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
dtype: int64
```

Checking the presence of duplicated values:



```
 0
```

Data after Encoding:

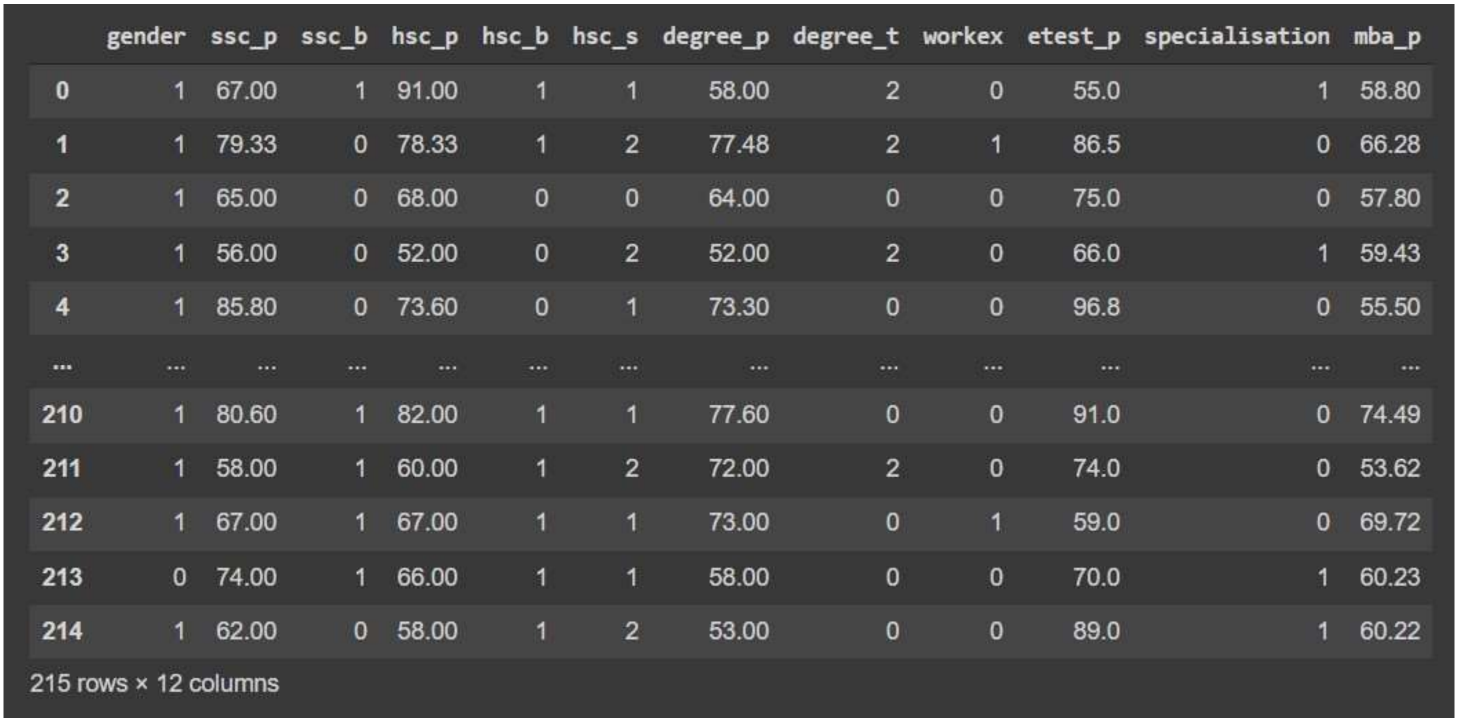


```
     gender  ssc_p  ssc_b  hsc_p  hsc_b  hsc_s  degree_p  degree_t  workex  \
0         1  67.00      1  91.00      1      1     58.00         2       0
1         1  79.33      0  78.33      1      2     77.48         2       1
2         1  65.00      0  68.00      0      0     64.00         0       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]
```

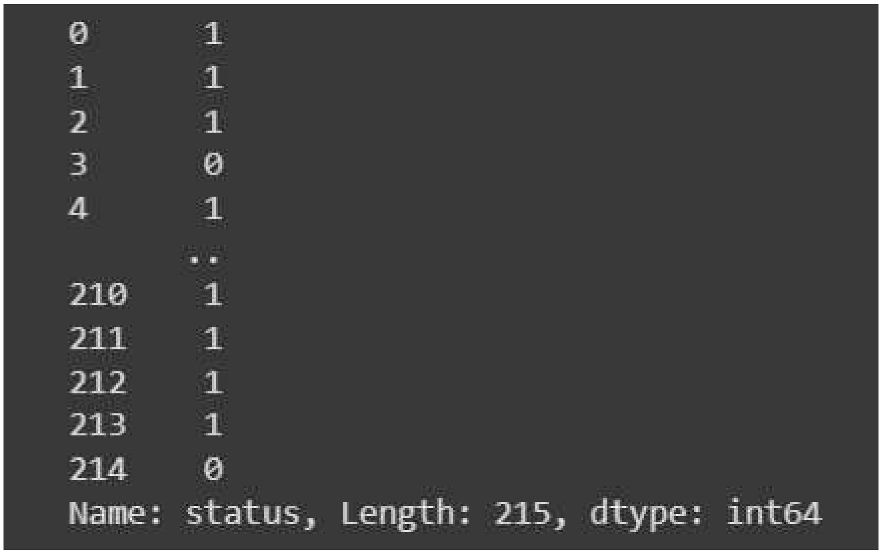
X Data:



	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p
0	1	67.00	1	91.00	1	1	58.00	2	0	55.0	1	58.80
1	1	79.33	0	78.33	1	2	77.48	2	1	86.5	0	66.28
2	1	65.00	0	68.00	0	0	64.00	0	0	75.0	0	57.80
3	1	56.00	0	52.00	0	2	52.00	2	0	66.0	1	59.43
4	1	85.80	0	73.60	0	1	73.30	0	0	96.8	0	55.50
...	...	...	...	...	...	...	...	...	...	...	...	...
210	1	80.60	1	82.00	1	1	77.60	0	0	91.0	0	74.49
211	1	58.00	1	60.00	1	2	72.00	2	0	74.0	0	53.62
212	1	67.00	1	67.00	1	1	73.00	0	1	59.0	0	69.72
213	0	74.00	1	66.00	1	1	58.00	0	0	70.0	1	60.23
214	1	62.00	0	58.00	1	2	53.00	0	0	89.0	1	60.22

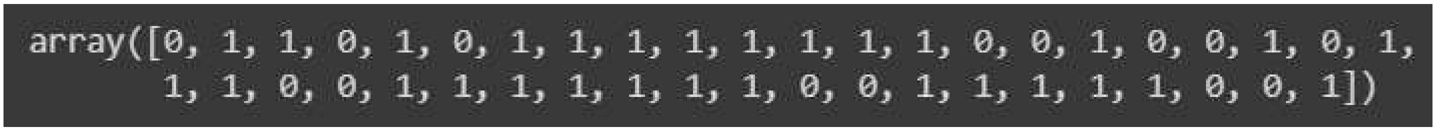
215 rows × 12 columns

Y Data:



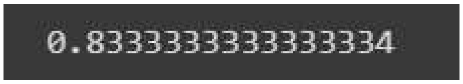
```
0      1
1      1
2      1
3      0
4      1
      ..
210    1
211    1
212    1
213    1
214    0
Name: status, Length: 215, dtype: int64
```

Predicted Values:



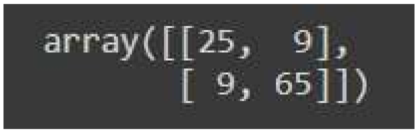
```
array([0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1])
```

Accuracy Score:



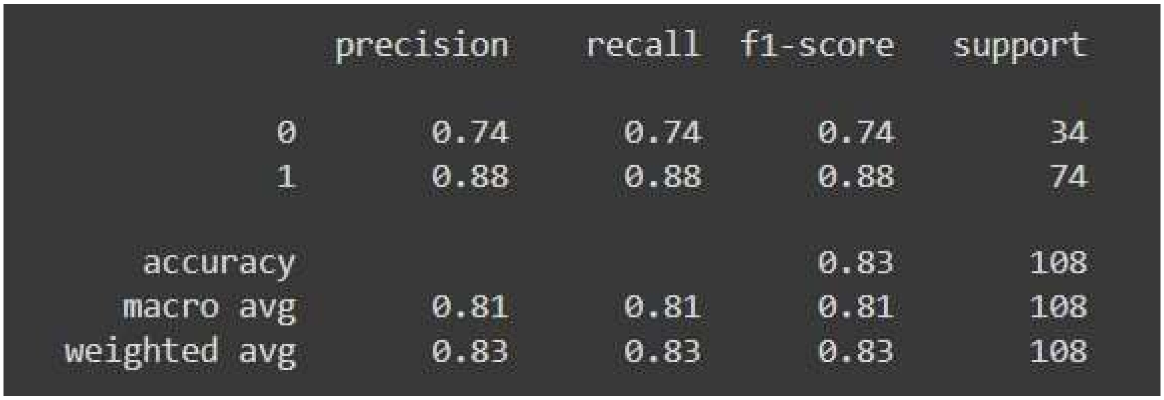
```
0.8333333333333334
```

Confusion Matrix:



```
array([[25,  9],
       [ 9, 65]])
```

Classification Report:

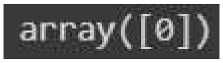


```
              precision    recall  f1-score   support

           0       0.74      0.74      0.74        34
           1       0.88      0.88      0.88        74

    accuracy                           0.83       108
   macro avg       0.81      0.81      0.81       108
weighted avg       0.83      0.83      0.83       108
```

Predicting output from Regression Model:



```
array([0])
```

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.