

IMPLEMENT LOGISTICS REGRESSION AND FIND ACCURACY AND CONFUSION MATRIX

```
In [3]: import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sns
```

```
In [4]: #pp=pd.read_csv('C:\\Users\\91733\\Downloads\\titanic_data.csv')
```

```
In [5]: #pp
```

```
In [6]: df=pd.read_csv('C:\\satish (coding)\\csv files\\placement.csv')
```

```
In [7]: df
```

Out[7]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
998	8.62	46.0	1
999	4.90	10.0	1

1000 rows × 3 columns

```
In [8]: x=(df[['cgpa','placement_exam_marks']])
y=df.iloc[:,2]
```

```
In [9]: xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
In [10]: xtrain
```

```
Out[10]:
```

	cgpa	placement_exam_marks
105	6.72	12.0
68	6.58	30.0
479	6.38	33.0
399	5.74	43.0
434	7.01	5.0
...
835	6.67	65.0
192	6.67	38.0
629	8.15	11.0
559	6.61	58.0
684	8.02	67.0

700 rows × 2 columns

```
In [11]: xtest
```

```
Out[11]:
```

	cgpa	placement_exam_marks
993	6.73	21.0
859	6.52	22.0
298	6.16	48.0
553	7.46	17.0
672	7.75	13.0
...
167	6.65	48.0
998	8.62	46.0
984	6.89	16.0
491	7.77	49.0
10	6.82	16.0

300 rows × 2 columns

```
In [12]: ytest
```

```
Out[12]: 993    1
          859    0
          298    0
          553    0
          672    0
          ..
          167    1
          998    1
          984    1
          491    1
           10    1
          Name: placed, Length: 300, dtype: int64
```

```
In [13]: ytrain
```

```
Out[13]: 105    1
          68     0
          479    0
          399    1
          434    0
          ..
          835    1
          192    0
          629    1
          559    1
          684    0
          Name: placed, Length: 700, dtype: int64
```

```
In [14]: from sklearn.linear_model import LogisticRegression
          dt=LogisticRegression()
```

```
In [15]: dt.fit(xtrain,ytrain)
```

```
Out[15]: ▾ LogisticRegression
          LogisticRegression()
```

```
In [16]: p=dt.predict(xtest)
p
```

```
Out[16]: array([0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
                0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0,
                0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0,
                1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
                0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,
                0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

```
In [17]: import sklearn.metrics as mc
```

```
In [18]: mc.confusion_matrix(ytest,p)
```

```
Out[18]: array([[120,  31],
                [122,  27]], dtype=int64)
```

```
In [19]: mc.accuracy_score(ytest,p)
```

```
Out[19]: 0.49
```