ASSIGNMENT-02

1) Write a program in Python to create a DEMO Machine Learning project.

Code Snippet ①

```
import numpy as np
import pandas as pd
df = pd.read_csv("placement.xls")
df = df.iloc[:,1:4]
df
```

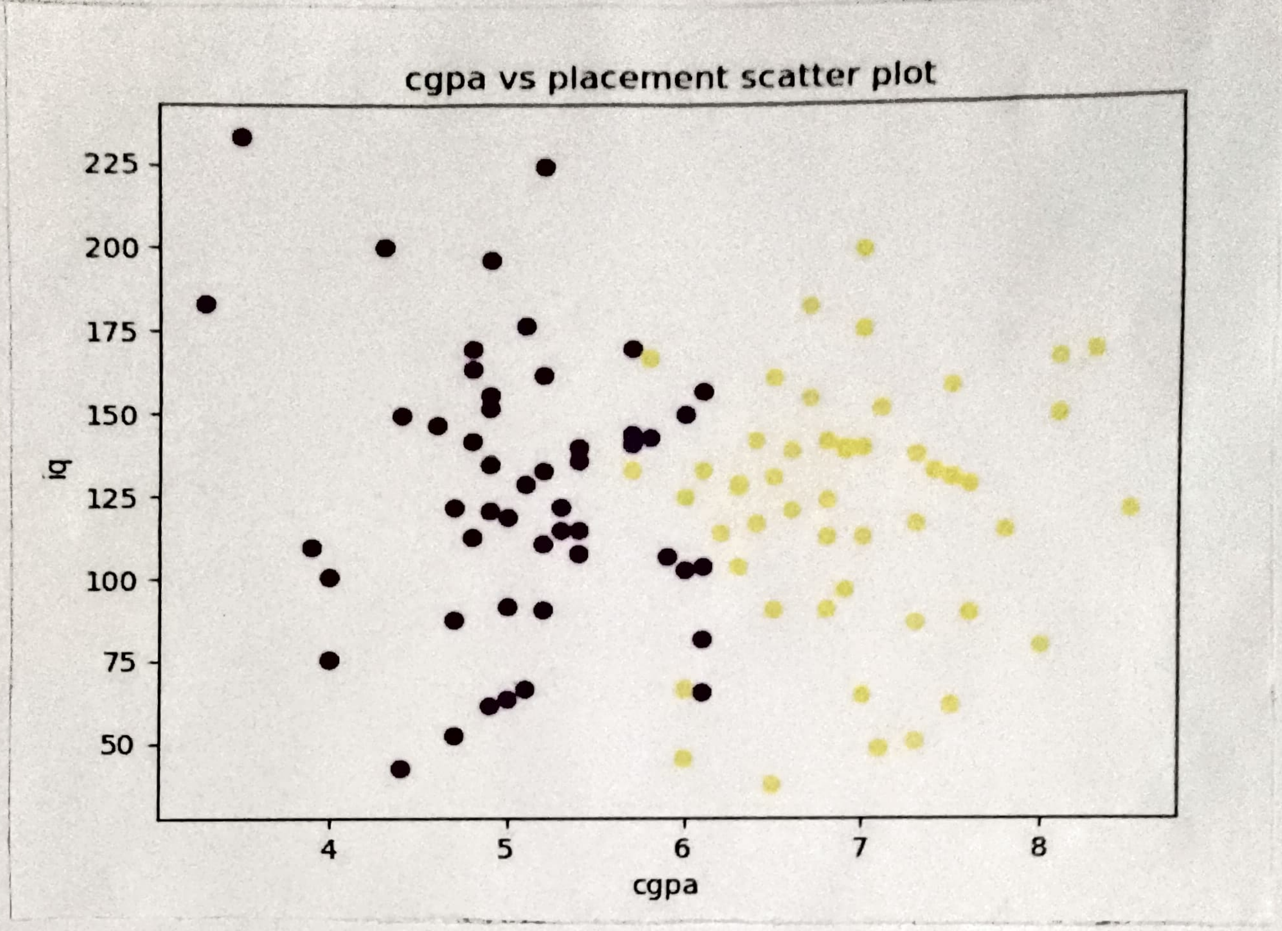
Output

```
      cgpa   iq     placement
0     6.8    123.0     1
1     5.9    106.0     0
2     5.3    121.0     0
3     7.4    132.0     1
4     5.8    142.0     0
...   ...    ...       ...
95    4.3    200.0     0
96    4.4    42.0      0
97    6.7    182.0     1
98    6.3    103.0     1
99    6.2    113.0     1
100 rows X 3 columns
```

Code Snippet ②

```
import matplotlib.pyplot as plt
plt.scatter (df['cgpa'], df['iq'], c = df['placement'])
plt.xlabel ('cgpa')
plt.ylabel('iq')
plt.title('cgpa vs placement scatter plot')
plt.show()
```

Output



Code Snippet ③

```
x = df.iloc[:,0:2]
y = df.iloc[:, -1]
x,y
```

Output

```
(    cgpa    iq
0    6.8    123.0
1    5.9    106.0
2    5.3    121.0
3    7.4    132.0
4    5.8    142.0
...  ...    ...
[100 rows X 2 columns]
0    1
1    0
2    0
3    1
4    0
...  ...
Name: placement, Length: 100, dtype: int64 )
```

Code Snippet ④

```
x.shape, y.shape
```

Output

```
((100,2),(100,))
```

Code Snippet ⑤

```
from sklearn.model-selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3)
x_train, x_test, y_train, y_test
```

Output

```
(     cgpa     iq
94    4.7     52.0
6     5.7    143.0
52    7.0    175.0
1     5.9    106.0
54    6.4    141.0
...   ...     ...
[70 rows x 2 columns],
      cgpa    iq
10    6.0    45.0
38    6.5   160.0
96    4.4    42.0
32    7.0   139.0
3     7.4   132.0
...   ...    ...
21    7.1   151.0
94     0
6      0
52     1
1      0
...   ...
Name: placement, Length: 70, dtype: int64,
10     1
38     1
96     0
32     1
3      1
...   ...
Name: placement, dtype: int64)
```

Code Snippet ⑥

```
from sklearn.preprocessing import StandardScaler
Scalar = StandardScaler()
x_train = scalar.fit_transform(x_train)
x_train
```

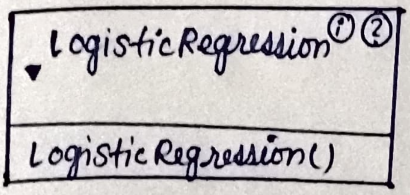
Output

```
array([[-1.20323409,-1.95718967],
       [-0.30048611, 0.63314862],
       [0.87308626, 1.54403681],
       [-0.11993652,-0.42006585],
       [0.33143747, 0.57621811],
          ...          ...
       [-0.8421349, 0.20616978]])
```

Code Snippet ⑦

```
x_test = scalar.transform(x_test)
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(x_train, y_train)
```

Output



```
▼ LogisticRegression ⓘ ?

LogisticRegression()
```

Code Snippet ⑧

```
y_pred = clf.predict(x_test)
y_pred
```

Output

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

Code Snippet ⑨

```
y_test
```

Output

```
10    1
38    1
96    0
32    1
 3    1
...   ...
21    1
Name: placement, dtype: int64
```

Code Snippet ⑩

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Output

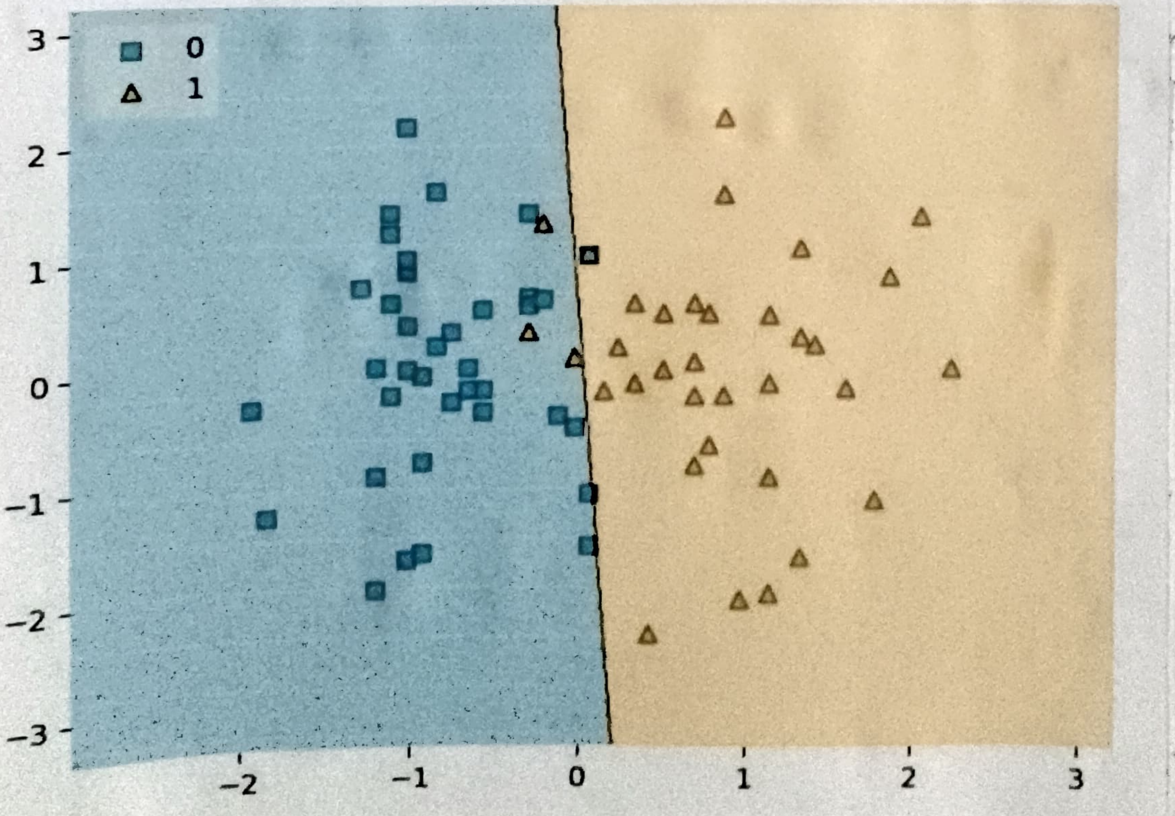
```
0.9
```

Code Snippet ⑪

```
from mlxtend.plotting import plot_decision_regions
plot_decision_regions(x_train, y_train.values, clf=clf, legend=2)
```

Output

```
<Axes: >
```



■ 0
▲ 1
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3