```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv('/content/placement.csv')
```

```
df.head()
```

	Unnamed: 0	cgpa	iq	placement
0	0	6.8	123.0	1
1	1	5.9	106.0	0
2	2	5.3	121.0	0
3	3	7.4	132.0	1
4	4	5.8	142.0	0

```
df.shape
```

```
(100, 4)
```

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  100 non-null    int64
 1   cgpa        100 non-null    float64
 2   iq          100 non-null    float64
 3   placement   100 non-null    int64
dtypes: float64(2), int64(2)
memory usage: 3.2 KB
```

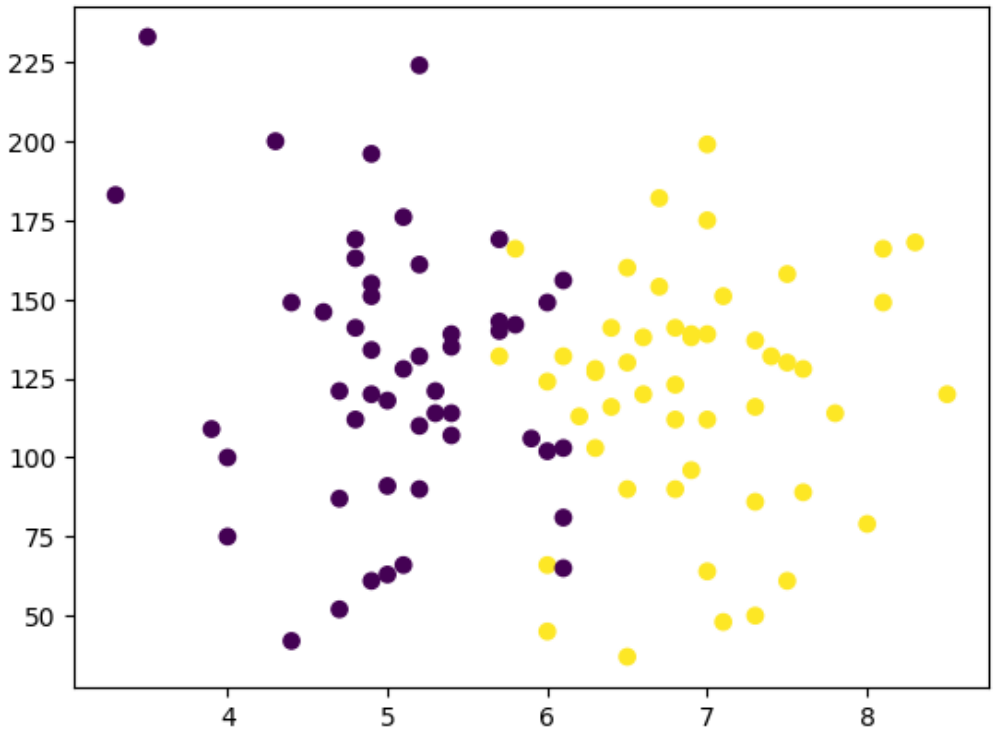
```
df=df.iloc[:, 1:]
df.head()
```

	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

```
import matplotlib.pyplot as plt
```

```
plt.scatter(df['cgpa'],df['iq'],c=df['placement'])
plt.figure(figsize=(4, 4))
```

<Figure size 400x400 with 0 Axes>



<Figure size 400x400 with 0 Axes>

```
#steps
# 0. Preprocess+EDA+ Disk: 26.32 GB/107.72 GB
# 1. Extract input a[illegible] output columns.
# 2. Scale the values.
# 3. Train test split.
# 4. Train the model.
# 5. Evaluate the model/ model selection.
# 6. Deploy the model.
```

```
# step 1. extract input and output columns. here in this 'cgpa' and 'iq' both are the input values and can
#as independent values while 'placement' is the output and is also referred as dependent variable as it is
```

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

```
x
```

	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
...	...	...
95	4.3	200.0
96	4.4	42.0
97	6.7	182.0
98	6.3	103.0
99	6.2	113.0

100 rows × 2 columns

```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement alr                                1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-le
Requirement alr                                1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-lea
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-le
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from sc
```



```
y.shape
```

```
(100,)
```

```
#train test split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1)
```

```
x_train
```

	cgpa	iq
25	5.0	91.0
59	4.8	112.0
51	4.8	141.0
92	5.2	110.0
86	5.1	128.0
...	...	...
13	6.4	116.0
69	8.5	120.0
34	4.8	163.0
30	7.6	128.0
42	7.6	89.0

90 rows × 2 columns

x_test

	cgpa	iq
22	4.9	120.0
37	8.1	149.0
80	4.9	196.0
33	6.0	149.0
53	8.3	168.0
49	5.4	135.0
81	5.4	107.0
2	5.3	121.0
71	6.1	132.0
36	5.7	140.0

y_train

```
25    0
59    0
51    0
92    0
86    0
     ..
13    1
69    1
34    0
30    1
42    1
Name: placement, Length: 90, dtype: int64
```

y_test

```
22    0
37    1
```

```
80    0
33    0
53    1
49    0
81    0
2     0
71    1
36    0
Name: placement, dtype: int64
```

```
from sklearn.preprocessing import StandardScaler
```

```
scaler=StandardScaler()
```

```
x_train=scaler.fit_transform(x_train)
```

```
x_train
```

```
array([[-0.87085983, -0.75296061],
       [-1.04698878, -0.23565943],
       [-1.04698878,  0.47870887],
       [-0.69473088, -0.28492621],
       [-0.78279535,  0.15847481],
       [-1.83956908, -0.3095596 ],
       [-1.7515046 , -1.14709485],
       [ 0.7143
       [ 1.2426
       [-2.19182699,  2.74498073],
       [ 1.15462315,  0.38017531],
       [ 0.89042971,  1.90744548],
       [ 0.09784942,  0.84820972],
       [-0.69473088,  2.52328022],
       [-0.87085983, -1.44269553],
       [ 0.71430076,  0.47870887],
       [ 0.62623628,  0.79894294],
       [-0.69473088,  0.97137667],
       [ 0.71430076, -0.23565943],
       [ 0.27397837, -0.45735994],
       [ 1.77107449, -1.04856129],
       [-0.6066664 , -0.18639265],
       [-0.16634401,  0.50334226],
       [-1.7515046 , -0.53126011],
       [-1.13505326, -1.71366281],
       [ 0.97849419,  0.72504277],
       [-0.25440849,  0.25700837],
       [ 0.36204285,  0.47870887],
       [ 0.09784942, -0.99929451],
       [ 0.53817181, -0.03859231],
       [ 0.45010733,  0.20774159],
       [ 1.15462315, -0.87612756],
       [-1.13505326, -0.01395892],
       [ 1.3307521 ,  0.8974765 ],
       [-0.16634401,  1.09454362],
       [-0.07827954, -0.38345977],
       [ 1.3307521 ,  0.20774159],
       [ 1.15462315, -0.13712587],
       [-0.95892431, -1.4919623 ],
       [-0.25440849,  0.52797565],
       [-0.69473088,  0.25700837],
       [ 0.80236524,  0.42944209],
       [-0.25440849,  1.16844379],
       [ 0.80236524,  0.4048087 ],
       [ 0.00978494, -1.88609654],
       [-1.39924669,  0.67577599],
       [ 0.27397837,  0.15847481],
       [ 1.85913896,  1.09454362],
```

```
            [-0.95892431,  0.72504277],
            [ 0.00978494, -1.36879536],
            [-2.36795594,  1.51331124],
            [-0.78279535,  1.34087751],
            [ 0.09784942, -0.45735994],
            [-0.78279535, -1.36879536],
            [-1.22311774,  0.60187582],
            [ 0.89042971,  0.42944209],
            [ 0.89042971, -1.41806214],
            [ 0.53817181,  0.4048087 ],
```

```
#training the model
from sklearn.linear_model import LogisticRegression
```

```
clf=LogisticRegression()
```

```
clf.fit(x_train, y_train)
```

```
LogisticRegression()
```

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

```
y_test
```



```
22    0
37    1
80    0
33    0
53    1
49    0
81    0
2     0
71    1
36    0
Name: placement, dtype: int64
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,y_pred)
```

```
0.9
```

```
from mlxtend.plotting import plot_decision_regions
```

```
plot_decision_regions(x_train,y_train.values,clf=clf,legend=2)
```

```
<Axes: >
```