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

```
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

Next steps: Generate code with df | New interactive sheet

```
df.shape
```

```
(100, 4)
```

```
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

Next steps: Generate code with df | New interactive sheet

```
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

Next steps: Generate code with x | New interactive sheet

```
y
```

	placement
0	1
1	0
2	0
3	1
4	0
...	...
95	0
96	0
97	1
98	1
99	1

100 rows × 1 columns

dtype: int64

```
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
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipython-input-4182802953.py in <cell line: 0>()
----> 1 x-train

NameError: name 'train' is not defined
```

Next steps: Explain error

```
x_train
```

	cgpa	iq
52	7.0	175.0
85	5.8	166.0
38	6.5	160.0
29	7.0	112.0
62	6.0	102.0
...	...	...
57	6.5	130.0
2	5.3	121.0
82	6.5	37.0
99	6.2	113.0
73	4.9	61.0

90 rows × 2 columns

Next steps: Generate code with x_train New interactive sheet

```
x_test
```

	cgpa	iq
47	5.2	161.0
56	6.1	65.0
17	3.3	183.0
69	8.5	120.0
33	6.0	149.0
27	6.0	124.0
24	4.7	121.0
41	5.4	114.0
12	5.4	139.0
46	5.3	114.0

Next steps: Generate code with x_test | New interactive sheet

```
y_test
```

	placement
47	0
56	0
17	0
69	1
33	0
27	1
24	0
41	0
12	0
46	0

dtype: int64

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

```
scaler=StandardScaler()
```

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

```
x_train
```

```
       [ 1.22155739e+00,  2.22129741e-01],
       [ 1.75872433e+00, -1.08274571e+00],
       [-5.68999047e-01, -3.93377546e-01],
       [-2.27002767e+00,  2.70877918e+00],
       [ 1.31108522e+00,  8.62257319e-01],
       [-5.68999047e-01,  2.95990615e-01],
       [ 6.84390462e-01, -8.11922501e-01],
       [-1.19569380e+00, -1.74749358e+00],
       [-1.46427727e+00, -1.99369649e+00],
       [ 1.31108522e+00,  1.72889158e-01],
       [-8.37582513e-01,  1.23648575e-01],
       [-7.48054691e-01, -3.19516672e-01],
       [ 5.76957075e-02, -1.03350512e+00],
       [ 1.57966868e+00, -2.21035506e-01],
       [-1.28522162e+00,  5.66813822e-01],
       [-1.46427727e+00,  6.40674696e-01],
       [-1.10616598e+00,  1.13308053e+00],
       [-1.01663816e+00, -7.33137568e-02],
       [-3.00415581e-01,  4.92952947e-01],
       [ 1.40061304e+00,  1.23648575e-01],
       [ 7.73918284e-01,  3.94471781e-01],
       [ 2.02730779e+00,  1.10846023e+00],
       [ 5.76957075e-02,  8.13016736e-01],
       [ 1.31108522e+00, -1.52591095e+00],
       [ 1.13202957e+00, -9.10403667e-01],
       [ 1.13202957e+00, -1.79673416e+00],
       [ 4.15806996e-01,  1.72889158e-01],
       [-6.58526869e-01, -4.86934654e-02],
       [ 4.15806996e-01, -2.11679795e+00],
       [ 1.47223530e-01, -2.45655797e-01],
       [-1.01663816e+00, -1.52591095e+00]])
```

```
x_test=scaler.transform(x_test)
```

```
x_test
```

```
array([[-0.74805469,  0.93611819],
       [ 0.05769571, -1.42742979],
       [-2.44908331,  1.47776461],
       [ 2.20636344, -0.07331376],
       [-0.03183211,  0.6406747 ],
       [-0.03183211,  0.02516741],
       [-1.1956938 , -0.04869347],
       [-0.56899905, -0.22103551],
       [-0.56899905,  0.39447178],
       [-0.65852687, -0.22103551]])
```

```
from sklearn.linear_model import LogisticRegression
```

```
clf=LogisticRegression()
```

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

▾ LogisticRegression

```
LogisticRegression()
```

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

```
y_test
```

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

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

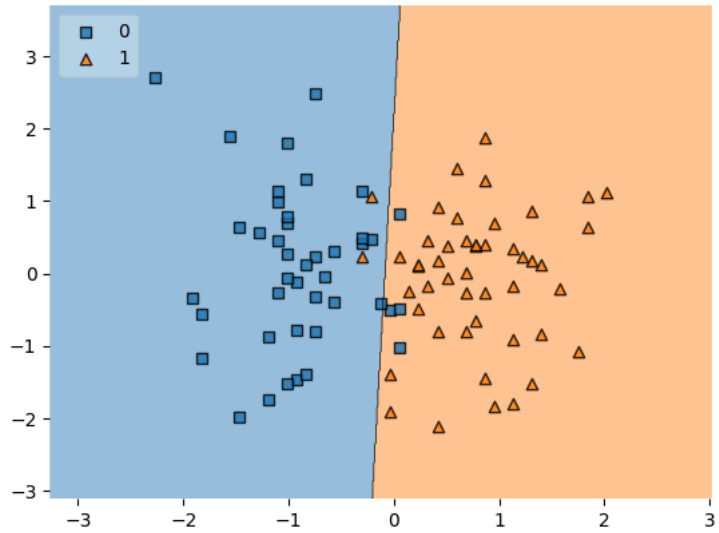
```
0.8
```

```
17    0
69    1
```

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

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

```
<Axes: >
```




```
import pickle
```

```
pickle.dump(clf,open('model.pkl','wb'))
```

Start coding or generate with AI.