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

```
df = pd.read_csv('./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 = 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

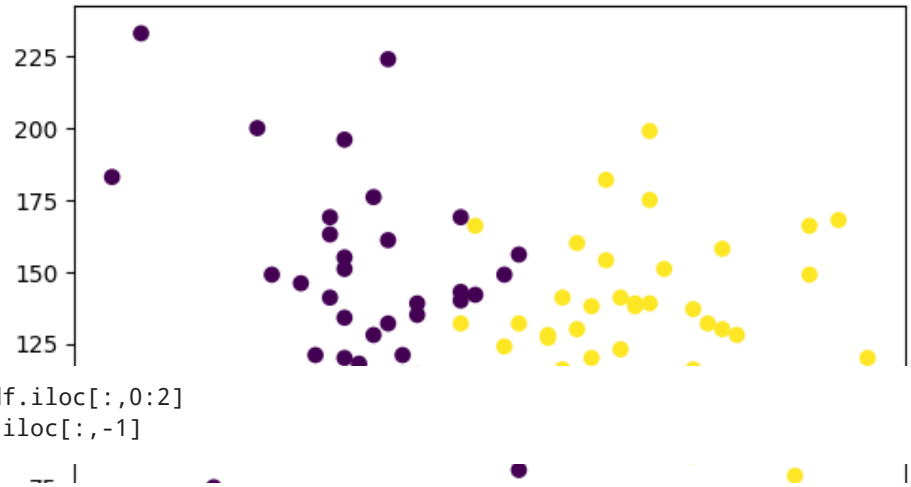
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   cgpa       100 non-null    float64
 1   iq         100 non-null    float64
 2   placement  100 non-null    int64  
dtypes: float64(2), int64(1)
memory usage: 2.5 KB
```

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

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

```
<matplotlib.collections.PathCollection at 0x7b0dc3db11e0>
```




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

```
x.head()
```

	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

```
y.head()
```

```
0    1
1    0
2    0
3    1
4    0
Name: placement, 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)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

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

```
array([[ 1.04743502,  0.64522148],
       [-1.08093718, -1.82849771],
       [ 1.93425677,  0.59524736],
       [-1.3469837 ,  0.59524736],
       [ 0.87007067,  0.34537673],
       [-0.72620848,  0.07051904],
       [ 0.51534197, -0.87898933],
       [ 0.95875285, -1.52865296],
       [-0.81489065, -0.85400227],
       [-0.72620848,  1.26989804],
       [-1.08093718, -0.95395052],
       [-0.10543325,  1.02002742],
       [-1.16961935,  0.52028617],
       [ 0.69270632,  0.72018267],
```

```
            [ 1.40216372,  0.82013092],
            [ 2.28898547, -0.12937746],
            [-0.54884413, -0.27929983],
            [ 0.95875285,  0.34537673],
            [ 1.8455746 , -1.15384702],
            [ 0.4266598 ,  0.39535086],
            [ 1.22479937, -1.87847183],
            [ 0.0719311 , -0.57914458],
            [ 0.51534197,  0.87010504],
            [-0.46016195, -0.45420927],
            [-0.6375263 ,  0.89509211],
            [ 1.22479937, -0.97893758],
            [ 0.16061327, -1.1038729 ],
            [ 1.40216372, -1.60361415],
            [-1.08093718, -0.10439039],
            [ 0.0719311 , -1.47867883],
            [ 0.95875285,  1.84460048],
            [ 1.22479937,  0.29540261],
            [-0.46016195,  0.34537673],
            [-0.90357283, -0.12937746],
            [ 0.33797762,  0.04553198],
            [ 0.60402415, -0.12937746],
            [ 0.16061327, -0.55415752],
            [ 1.31348155,  0.17046729],
            [ 2.11162112,  1.07000154],
            [-0.992255  ,  0.94506623],
            [ 0.7813885 , -0.05441627],
            [-0.6375263 , -0.37924808],
            [-0.90357283,  0.22044142],
            [ 0.7813885 ,  0.39535086],
            [ 0.33797762, -0.55415752],
            [-0.19411543,  0.44532498],
            [-0.6375263 , -0.87898933],
            [-0.81489065, -0.17935158],
            [-0.90357283,  0.64522148],
            [ 0.87007067,  0.32038967],
            [-0.19411543,  1.09498861],
            [-2.14512328,  2.69416061],
            [-0.81489065, -1.55364002],
            [-0.992255  , -0.32927396],
            [ 1.40216372,  0.12049317],
            [-2.32248763,  1.44480748],
            [-0.992255  ,  1.09498861],
            [ 1.7017124 ,  0.62911871]
```

```
x_test = scaler.fit_transform(x_test)
```

```
x_test
```

```
    array([[  7.8, 114. ],
           [  6.1,  65. ],
           [  6.8,  90. ],
           [  6. , 124. ],
           [  6.2, 113. ],
           [  6.8, 112. ],
           [  8.1, 166. ],
           [  6.6, 138. ],
           [  7.6, 128. ],
           [  4.4,  42. ]])
```

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

```
clf = LogisticRegression()
```

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

```
▾ LogisticRegression
LogisticRegression()
```

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

```
y_pred
```

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

```
y_test
```

```
55    1
56    0
35    1
27    1
99    1
93    1
65    1
48    1
30    1
96    0
Name: placement, dtype: int64
```

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

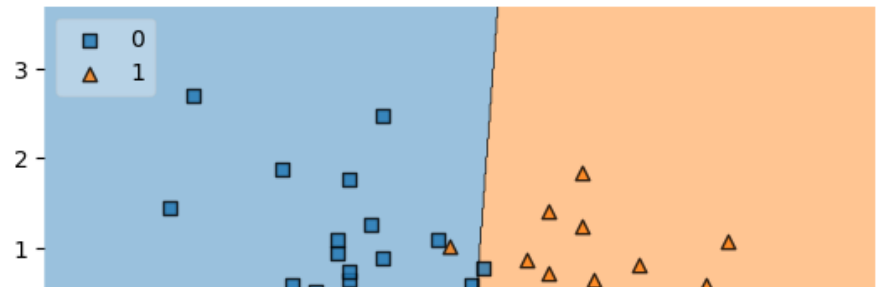
```
accuracy_score(y_test, y_pred)
```

```
0.7
```

```
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'))
```