STEP OF MACHINE LEARNING

1. PREPROCESSING + EDA + FEATURE SELECTION
2. EXTRACT INPUT AND OUTPUT COLUMNS
3. TRAIN TEST SPLIT
4. TRAIN THE MODEL
5. EVALUATE THE MODEL/MODEL SELECTION
6. DEPLOY THE MODEL

```
#IMPORTING THE NECESSARY LIBRARIES
import numpy as np
import pandas as pd

#Reading the data

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

```
     Unnamed: 0  cgpa   iq  placement
0             0   6.8  123          1
1             1   5.9  106          0
2             2   5.3  121          0
3             3   7.4  132          1
4             4   5.8  142          0
...         ...   ...  ...        ...
95           95   4.3  200          0
96           96   4.4   42          0
97           97   6.7  182          1
98           98   6.3  103          1
99           99   6.2  113          1

100 rows × 4 columns
```

```
data.head()
```

```
     Unnamed: 0  cgpa   iq  placement
0             0   6.8  123          1
1             1   5.9  106          0
2             2   5.3  121          0
3             3   7.4  132          1
4             4   5.8  142          0
```

```
data.tail()
```

```
     Unnamed: 0  cgpa   iq  placement
95           95   4.3  200          0
96           96   4.4   42          0
97           97   6.7  182          1
98           98   6.3  103          1
99           99   6.2  113          1
```

```
data.shape
```

```
(100, 4)
```

```
data.shape[0]
```

```
100
```

```
#removing missing values
data.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    int64  
 3   placement   100 non-null    int64  
dtypes: float64(1), int64(3)
memory usage: 3.2 KB
```

here as you can see there are no null values. so no data preprocessing required

```
data.head()
```

	Unnamed: 0	cgpa	iq	placement
0	0	6.8	123	1
1	1	5.9	106	0
2	2	5.3	121	0
3	3	7.4	132	1
4	4	5.8	142	0

```
data.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    int64  
 3   placement   100 non-null    int64  
dtypes: float64(1), int64(3)
memory usage: 3.2 KB
```

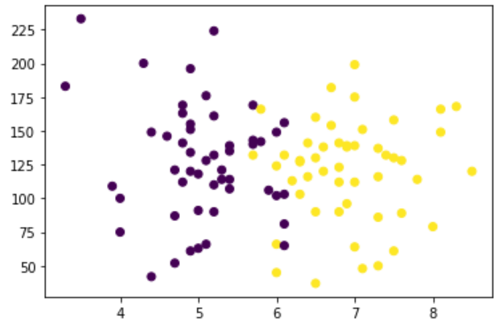
```
data = data.iloc[:,1:]
data
```

cgpa	iq	placement

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

```
plt.scatter(data['cgpa'],data['iq'],c = data['placement'])
#xaxis = cgpa , y axis = iq # colored the placement coloumnn
```

```
<matplotlib.collections.PathCollection at 0x7f9f1463fbe0>
```



*Using LOGISTIC REGRESSION NAMED LOGISTIC REGRESSION. *

```
#LOGISTIC REGRESSION BECAUSE THIS IS CLASSIFICATION PROBLEM
```

Independent Variable = Input = cgpa & iq dependent variable = output = Placement

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

```
x
```

	cgpa	iq
0	6.8	123
1	5.9	106
2	5.3	121
3	7.4	132
4	5.8	142
...	...	...
95	4.3	200
96	4.4	42
97	6.7	182
98	6.3	103
99	6.2	113

100 rows × 2 columns

```
y
```

```
0     1
1     0
2     0
3     1
4     0
     ..
95    0
```

```
96    0
97    1
98    1
99    1
Name: placement, Length: 100, dtype: int64
```

```
#train test split first
# then scale the values
```

```
from sklearn.model_selection import train_test_split
#train_test_split(inpdependant variable , dependent variable, give t
x_train,x_test, y_train, y_test = train_test_split(x,y,test_size = 0
```

```
x_train
```

	cgpa	iq
54	6.4	141
79	6.5	90
78	6.1	81
40	4.9	134
61	7.3	137
...	...	...
84	5.7	169
35	6.8	90
75	4.8	169
15	5.1	176
87	5.7	132

90 rows × 2 columns

```
x_test
```

	cgpa	iq
25	5.0	91
24	4.7	121
77	7.3	50
90	7.3	86
72	7.3	116
44	7.5	61
33	6.0	149
89	4.9	151
46	5.3	114
81	5.4	107

```
y_train
```

```
54    1
79    1
78    0
40    0
61    1
     ..
84    0
35    1
75    0
15    0
87    1
Name: placement, Length: 90, dtype: int64
```

```
y_test
```

```
25    0
24    0
77    1
90    1
72    1
44    1
33    0
89    0
46    0
81    0
Name: placement, dtype: int64
```

```
#scaling all values betweeen o and 1
#it is not necessary but you should do it

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

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

```
array([[ 0.36580376,  0.38287711],
       [ 0.45336317, -0.89245377],
       [ 0.10312553, -1.11751216],
       [-0.94758739,  0.2078317 ],
       [ 1.15383845,  0.28285116],
       [ 0.45336317, -2.21779762],
       [ 0.89116022, -1.54262245],
       [ 0.54092258, -0.14225913],
       [-0.86002798, -1.56762894],
       [ 2.20455136, -0.14225913],
       [-0.68490916,  2.45841561],
       [ 0.01556612, -0.04223318],
       [-2.34853795,  1.4331496 ],
       [ 0.27824435,  0.05779277],
       [ 0.10312553, -0.56736942],
       [-0.1595527 ,  1.00803931],
       [ 0.54092258,  0.30785765],
       [-0.59734975, -0.11725264],
       [ 0.7160414 ,  0.38287711],
       [ 0.97871963, -1.94272626],
       [ 0.62848199,  1.40814312],
       [-0.77246857, -1.49260948],
       [-1.12270621, -0.96747323],
       [ 0.27824435,  0.03278628],
       [ 0.36580376, -0.24228508],
       [-0.94758739, -1.61764191],
       [ 1.85431373,  1.00803931],
       [-0.50979034,  0.33286414],
       [ 1.41651668,  0.05779277],
       [-1.12270621, -1.84270031],
       [ 0.89116022, -0.34231103],
       [ 0.80360081, -0.74241484],
       [-0.24711211,  0.43289009],
       [-0.50979034,  0.23283819],
       [-1.0351468 ,  0.38287711],
       [ 0.97871963,  0.63294199],
       [ 1.41651668, -0.91746025],
       [-1.73562208, -1.26755108],
       [-1.47294385,  1.8582599 ],
       [-1.0351468 ,  0.93301985],
       [-0.77246857,  0.05779277],
       [-0.1595527 ,  0.4078836 ],
       [ 0.45336317,  0.10780575],
       [ 1.85431373,  0.58292902],
       [-0.68490916, -0.39232401],
       [ 0.89116022,  1.2330977 ],
       [ 1.32895727,  0.10780575],
       [-0.50979034, -0.29229806],
       [-1.73562208, -0.64238889],
       [ 0.80360081,  0.33286414],
       [-0.94758739,  0.73296794],
       [-0.07199329, -0.49234996],
       [ 0.89116022,  1.83325341],
       [ 0.19068494, -0.31730455],
       [-0.24711211,  0.35787063],
       [ 0.7160414 , -0.06723967],
       [-0.68490916,  0.15781872],
       [ 1.5916355 , -0.29229806]
```

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

```
array([[-0.97591834, -0.42941754],
       [-1.2495403 ,  0.51782704],
       [ 1.12185005, -1.72398513],
       [ 1.12185005, -0.58729164],
       [ 1.12185005,  0.35995294],
       [ 1.3042647 , -1.37666212],
       [-0.06384512,  1.40192198],
       [-1.06712566,  1.46507162],
       [-0.70229637,  0.2968033 ],
       [-0.61108905,  0.07577957]])
```

*TRAIN THE CLASSIFIER *

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

```
clf = LogisticRegression()
```

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

```
LogisticRegression()
```

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

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

```
y_test
```

```
25    0
24    0
77    1
90    1
72    1
44    1
33    0
89    0
46    0
81    0
Name: placement, dtype: int64
```

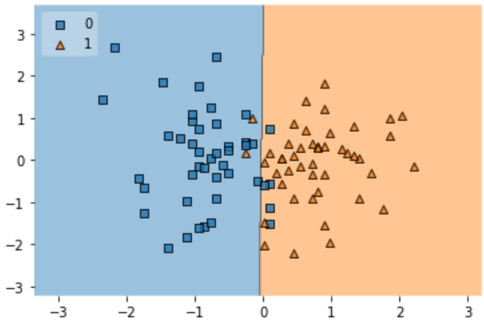
```
from sklearn.metrics import accuracy_score
```

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

```
1.0
```

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

```
/usr/local/lib/python3.8/dist-packages/mlxtend/plotting/decision_regions.py:244: MatplotlibDeprecationW
  ax.axis(xmin=xx.min(), xmax=xx.max(), y_min=yy.min(), y_max=yy.max())
<matplotlib.axes._subplots.AxesSubplot at 0x7f9f1456efd0>
```



Colab paid products - Cancel contracts here

✓ 0s completed at 12:04 PM