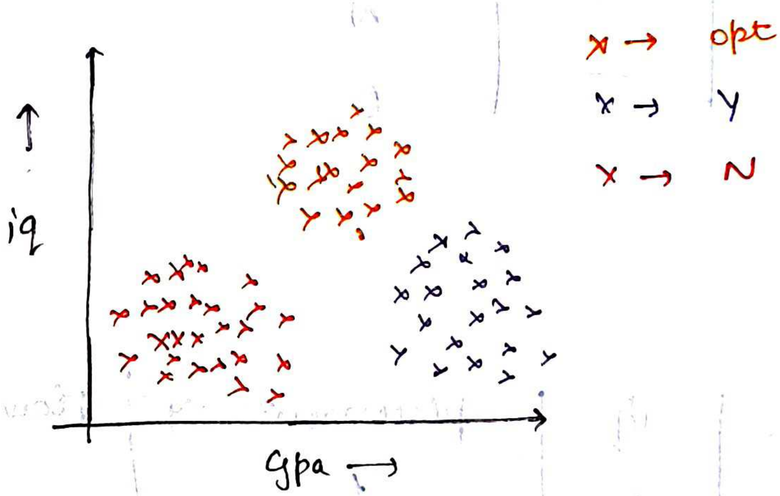
What is multiclass Classification?

gpa	iq	placement
9	90	Y
6	60	N
7	70	Opt

Y, N, Opt → 3 classes



How to Logistic Regression handle Multiclass Classification problems?

* mostly use in binary classification
* But now use in multiclassification with some technique

(i) OVR (one vs Rest) → OVA (One Vs All)

(ii) Multinomial Logistic Regression → Soft Max Reg

OVR Approach

train → prediction → code
↓

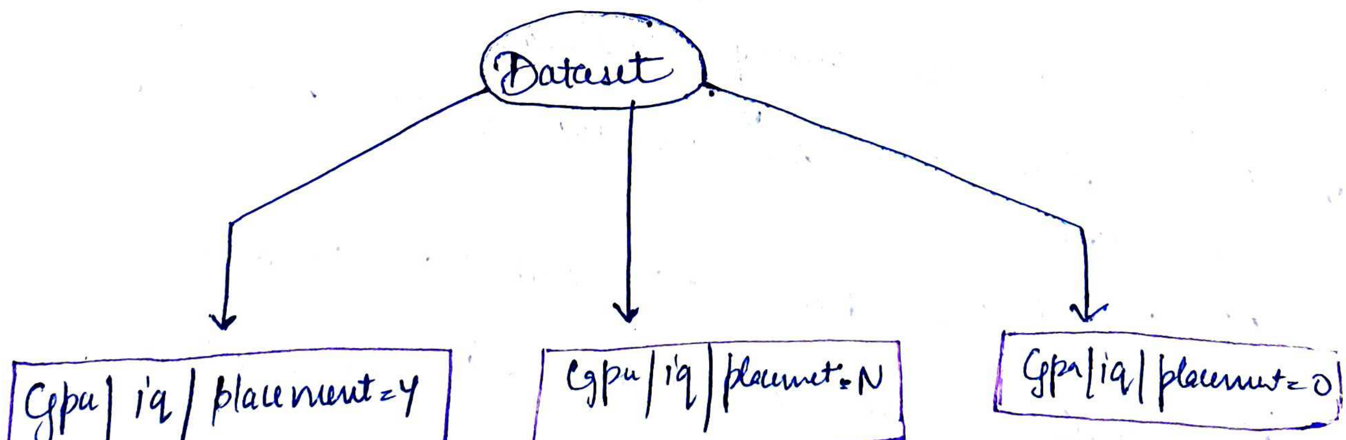
cgpa	iq	Placement
–	–	Y
–	–	N
–	–	O

(Y, N, O) → Logistic Reg model

* No. of category = apply No. of time Logistic Reg model.

Using One Hot Encoding:

cgpa	iq	placement = Y	placement = N	placement = O
–	–	1	0	0
–	–	0	1	0
–	–	0	0	1



Dataset 1 — $\beta_0 \; \beta_1 \; \beta_2$

Dataset 2 — $\beta_0 \; \beta_1 \; \beta_2$

Dataset 3 — $\beta_0 \; \beta_1 \; \beta_2$

* Apply Normal Logistic Regression on all three dataset.

example:-

Prediction {6.5, 65} → Y, N, O

P(Y) $\boxed{0.6}$

P(N) $\boxed{0.3}$

P(O) $\boxed{0.5}$

Normalize:

$$P(Y) = \frac{0.6}{0.6+0.3+0.5} = 0.42 = 42\%$$

$$P(N) = \frac{0.3}{0.6+0.3+0.5} = 0.23 = 23\%$$

$$P(O) = \frac{0.5}{0.6+0.3+0.5} = 0.35 = 35\%$$

$$P(Y) + P(N) + P(O) = 0.42 + 0.23 + 0.35 = 1$$

* Highest Probablity is output like 0.42 is Highest so, Output is Y.

* Not efficient with large dataset having high number of class.

code

Soft Max function

$$\boxed{\sigma(\vec{z})_i = \frac{\sigma^{z_i}}{\sum_{j=1}^{k} e^{z_i}}}$$

$z_1 \quad z_2 \quad z_3$

$$\boxed{\sigma(\vec{z_1}) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}}$$

$$\boxed{e(\vec{z_2}) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}}$$

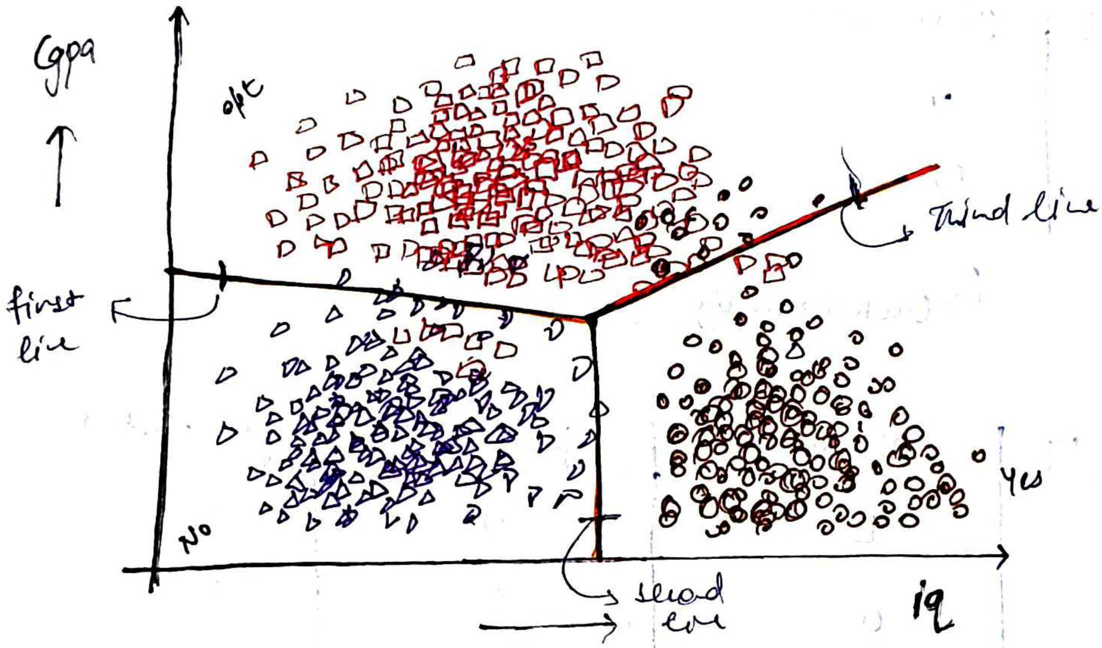
$$\boxed{e(\vec{z_3}) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}}$$

$$\sigma(\vec{z_1}) + e(\vec{z_2}) + e(\vec{z_3}) = 1$$

$\Downarrow$

multiclass

Softmax Logistic Regression / Multinomial LR



for K classes, Softmax LR will draw k lines to create the decision Region

\# for Binary we have $Ax + By + C = 0$

$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$

$\boxed{\beta_0 \quad \beta_1 \quad \beta_2}$

\# for 3 lines we have 3×3 total 9. parameters.

3 for each line

\# assumption for 2D data (2 feature)

for 3 (feature) = 3×4 = 12 parameter

category

$\beta_0 \ \beta_1 \ \beta_2 \ \beta_3$

training → Prediction → code
↓

Cgpa	iq	placement
-	-	Y
-	-	N
-	-	O

↳ Onehot encoding

Cgpa	iq	placement Y	placement N	placement O
-	-	1	0	0
-	-	0	1	0
-	-	0	0	1

Loss function:

general case →

$$\boxed{L = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{k} Y_i^k \log \hat{Y}_{ik}}$$

→ SoftMax LR
minimize

No. of classes

Binary :-

$$\boxed{L = -\frac{1}{n} \sum_{i=1}^{n} Y_i \log(\hat{Y}_i) + (1-Y_i) \log(1-\hat{Y}_i)}$$

↳ Special case only for K=2

cgpa	iq	placement = Y	placement = N	placement = 0
-	-	1	0	0
-	-	0	1	0
-	-	0	0	1

Let assume we have only 1 row

$n = 1$

so, $\sum_{i=1}^{n} = 1$

$$L = \sum_{k=1}^{k} y^k \log \hat{y}^k$$

$$L = y^{Yes} \log \hat{y}^{Yes} + \underset{0}{y^{No \to 0}} \log \hat{y}^{No} + y^{0 \to 0} \log \hat{y}^{0}$$

$$= \underline{(1) \log \hat{y}^{Yes}} + 0 + 0$$

For 2nd row

$$L = 0 + \underline{(1) \log \hat{y}^{No}} + 0$$

For 3rd row

$$L = 0 + 0 + \underline{(1) \log \hat{y}^{0}}$$

* In every row, we want those class which value has 1.

$$\boxed{L = \log \hat{Y}_1^{yes} + \log \hat{Y}_2^{No} + \log \hat{Y}_3^{\theta}}$$

$\hat{Y}_1^{yes} \to$ 1 row $\to \beta_0, \beta_1, \beta_2 \to$ 1st line

$\hat{Y}_2^{No} \to$ 2 row $\to \beta_0, \beta_1, \beta_2 \to$ 2nd line

$\hat{Y}_3^{\theta} \to$ 3 row $\to \beta_0, \beta_1, \beta_2 \to$ 3rd line

$\hat{Y}_1^{yes} = ?$ $\quad \hat{Y}_2^{No} = ?$ $\quad \hat{Y}_3^{\theta} = ?$

$\boxed{\hat{Y}_i} \to$ output of Logistic Regression

Sigmoid $\to$ (0 - 1)

$\hat{Y}_i = \sigma(Z_i)$

$$\boxed{Z_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}}$$

$\to$ for Normal logistic R.

$$\hat{Y}_1^{yes} = \sigma(Z_i) = \frac{e^{Z_{yes}}}{e^{Z_{yes}} + e^{Z_{no}} + e^{Z_{opt}}}$$

$\hookrightarrow$ softmax

$$Z_{yes} = \underbrace{\beta_0 + \overbrace{\beta_1 x_{i1} + \beta_2 x_{i2}}^{1\ \text{row}}}_{\text{coefficient and intersecpt of first line}}$$

$$Z_{No} = \underbrace{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}}_{\text{coeficient and intercept of second line}}$$

$$Z_{opt} \quad \underbrace{\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}}_{\text{Coeficient and intercept of Third line}}$$

$$\hat{Y}^{no}_{2} = \sigma(z_i) = \frac{e^{z_{no}}}{e^{z_{yes}} + e^{z_{no}} + e^{z_{opt}}}$$

$$\hat{Y}^{op}_{3} = \sigma(z_i) = \frac{e^{z_{opt}}}{e^{z_{yes}} + e^{z_{No}} + e^{z_{opt}}}$$

$$\left\{ \begin{bmatrix} \beta_0^1 & \beta_1^1 & \beta_2^1 \\ \beta_0^2 & \beta_1^2 & \beta_2^2 \\ \beta_0^3 & \beta_1^3 & \beta_2^3 \end{bmatrix} \begin{matrix} \rightarrow \text{1st line} \\ \rightarrow \text{2nd line} \\ \rightarrow \text{3nd line} \end{matrix} \right.$$

→ Start with 3 random line then start with Gradient descent

$$\beta_0^1 = \beta_0^1 - \eta \frac{\partial L}{\partial \beta_0^1}$$

$$\frac{\partial L}{\partial \beta_0^1} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k)$$

doing this for 9 different lines in 1000 times

* after 1000 times we get a value which less Loss and ~~good~~ best fit line.

Let assume $k = 2$ $(0, 1)$ then our Loss function will be binary cross entropy

$$= -\frac{1}{n}\sum_{i=1}^{n} y_i^1 \log(\hat{y}_i^1) + y_i^0 \log(\hat{y}_i^0)$$

$\boxed{y_i^1 \quad y_i^0}$ → Yper | iq | Placement $\overset{0}{N}, \overset{1}{Y}$

$1 \Rightarrow \quad y_i^1 = 1 \quad y_i^0 = 0$

$0 \Rightarrow \quad y_i^1 = 0 \quad y_i^0 = 1$

So,

$$= -\frac{1}{n}\sum_{i=1}^{n} \overset{\nearrow Y}{y_i^1} \left(\log \overset{\nearrow Y}{\hat{y}_i^1}\right) + (1-\overset{\nearrow N}{y_i^1}) \log(1-\overset{\nearrow N}{\hat{y}_i^1})$$

$$\boxed{L = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)}$$

→ Binary cross entropy

$\boxed{\text{prooved}}$

When to Use what?

Use One-vs-Rest (OVR) when:

1. Classes are Non-Mutually Exclusive: OVR is appropriate if an instance can belong to more than one class, as each classifier provides an independent probability for each class.

2. Dealing with Imbalanced Data: OVR might perform better when class distribution is highly imbalanced since each class gets a dedicated model.

Use Multinomial Logistic Regression (SoftMax Regression) when:

1. Computational Efficiency is Required: Softmax Regression is generally more efficient for large datasets and a high number of classes.

2. Classes are Mutually Exclusive: SoftMax Regression is a good choice when each instance can only belong to one class. The SoftMax function provides a set of probilities that sum to 1, fitting well with mutually exclusive classes.

3. Interpretability is important: The probabilities output by SoftMax Regression are more interpretable than those from OVR, as they always sum to 1. This can make model predictions easier to explain.

Gradient descent

Predict $\{6.5, 65\}$

line 1 →	β_0^1	β_1^1	β_2^1
line 2 →	β_0^2	β_1^2	β_2^2
line 3 →	β_0^3	β_1^3	β_2^3

$Z_1 = \beta_0^1 + 6.5\,\beta_1^1 + 65\,\beta_2^1$

$Z_2 = \beta_0^2 + 6.5\,\beta_1^2 + 65\,\beta_2^2$

$Z_3 = \beta_0^3 + 6.5\,\beta_1^3 + 65\,\beta_2^3$

Softmax ⇒ $$\sigma(Z_1) = \frac{e^{Z_1}}{e^{Z_1} + e^{Z_2} + e^{Z_3}}$$

$$\sigma(Z_2) = \frac{e^{Z_2}}{e^{Z_1} + e^{Z_2} + e^{Z_3}} \qquad \sigma(Z_3) = \frac{6^{Z_3}}{e^{Z_1} + e^{Z_2} + e^{Z_3}}$$

* get probability and high prob. is outcome or predicted outcome