BinaryClassification

- The Red, Blue, Green :

$$\begin{bmatrix} 225 \\ 231 \\ \vdots \\ 255 \\ 134 \\ \vdots \\ 225 \end{bmatrix}$$

- The Binary :
- given the x, y :

$$(x, y), x \in \vec{R^{n_x}}, y \in \{0, 1\}$$

- The ruled the m is training example :

$$\{(x^1, y^1), (x^2, y^2), \ldots, (x^m, y^m)\}$$

- ruled the m_{test} = m test example
- The ruled a matrix X :

$$X = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ x^1 & x^2 & \cdots & x^m \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}, x \in \vec{R}^{n_x}, X \in \vec{R}^{n_x \times m}$$

- The ruled a matrix Y :

$$Y = [y^1 \quad y^2 \quad \cdots \quad y^m], y \in \{0, 1\}, y \in \vec{R}^{1 \times m}$$

> **Remember the X can't do the X^T**

> **In Python:**

> - X.shape(n_x, m), Y.shape(1, m)

> - The X row is n_x, the col is m

> - The Y row is 1, the col is m

Logistic Regression

- Given 'x',The probability is :

$$\hat{y} = P \times (y = 1|_x), x \in \vec{R}^{n_x}$$

- The logistic parameter is :

$$w \in R^{\vec{n_x}}, b \in R$$

- The Output is :

$$\hat{y} = w^T \times x + b, 0 \leq y \leq 1$$

- In this way, we can get the sigmoid function :

$$\hat{y} = \sigma(w^T \times x + b), z = w^T + b \Rightarrow \sigma(z) = \frac{1}{1 + e^{-z}}$$

- If the $z \to +\infty$:

$$\sigma(z) = \lim \frac{1}{1 + 0} = 1$$

- If the $z \rightarrow 0\ or -\infty$:

$$\sigma(z) = \lim \frac{1}{1+\infty} = 0$$

- But if someone to designed the $x_0 = 1$:

$$x \in \vec{R}^{n_x+1}, \hat{y} = \sigma(\theta^T \times x)\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{n_x} \end{bmatrix} 记 : b = \theta_0, w = \{\theta_1, \theta_2, \ldots, \theta_{n_x}\}$$

Logistic Regression Lost Function

> train the logistic regression's parameters : w and b

- Given the m ($ m_{test}$) training examples:

$$\{(x^1, y^1), (x^2, y^2), \ldots, (x^m, y^m)\}$$

- From them to gain the w and b, and to gain the $ \hat y^{(i)} \rightarrow y^{(i)} $

The Lost(Error) Function

> **To gain or maybe to say suit for the single training**

$$def L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

> **But the result leads to the error of the real, the image is uneven**

- To deal with the bug :

$$def L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

- If $ y = 1 $, $ L(\hat y, y) = -\log \hat y :$

$$\Rightarrow \log \hat{y} \to +\infty, 0 \leq \hat{y} \leq 1 \Rightarrow \lim \hat{y} = 1$$

- If $ y = 0 $, $ L(\hat{y}, y) = -\log(1 - \hat{y}): $

$$\Rightarrow \log(1 - \hat{y}) \to +\infty, 0 \leq \hat{y} \leq 1 \Rightarrow \lim \hat{y} = 0$$

Cost Function

> **For the whole training examples**

$$def J(w, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$