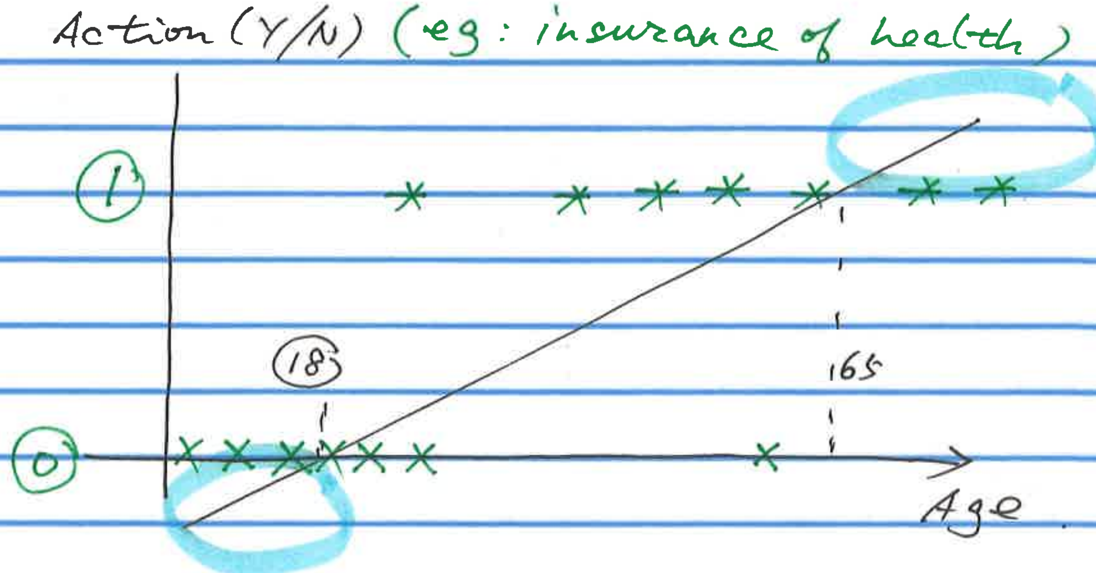
Lesson B01(b):
Logistic Regression

Logistic Regression: one of the most ML algorithms for binary classification.

Main Implementation Step:

1. How to compute logistic function
2. How to learn the coefficient for a logistic regression using Stochastic gradient decent
3. How to predict?



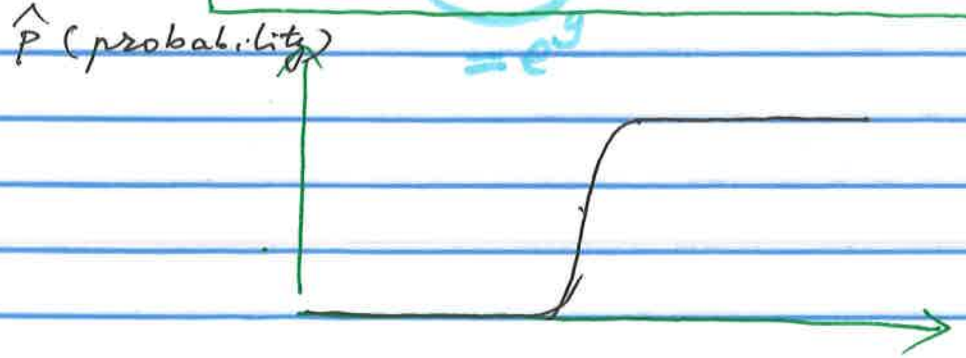
why Linear regression is not applicable on above problem? → Range of DV may lie outside of [0, 1].

Solution :

$y = \beta_0 + \beta_1 x$.
(Linear regression)

$P = \frac{1}{1+e^{-y}}$
(sigmoid function)

$$\boxed{\ln\left(\underbrace{\frac{P}{1-P}}_{=e^y}\right) = \beta_0 + \beta_1 \cdot x}$$



principle of logistic Regression.

$\longrightarrow$ linear regression of LOGIT.

(or $\ln(\text{odds})$)

① Define P as the probability of an Event.

$$\text{odds} = \frac{P}{1-P} \iff P = \frac{\text{odds}}{1+\text{odds}}.$$

② $\underbrace{\ln(\text{Odds})}_{\text{LOGIT}} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p \cdot x_p + e.$

$$P \in [0,1] \longrightarrow \ln(\text{Odds}) \in (-\infty, +\infty).$$

e.g: $P = 0 \Rightarrow \text{odds} = 0 \Rightarrow \ln(\text{odds}) = -\infty$

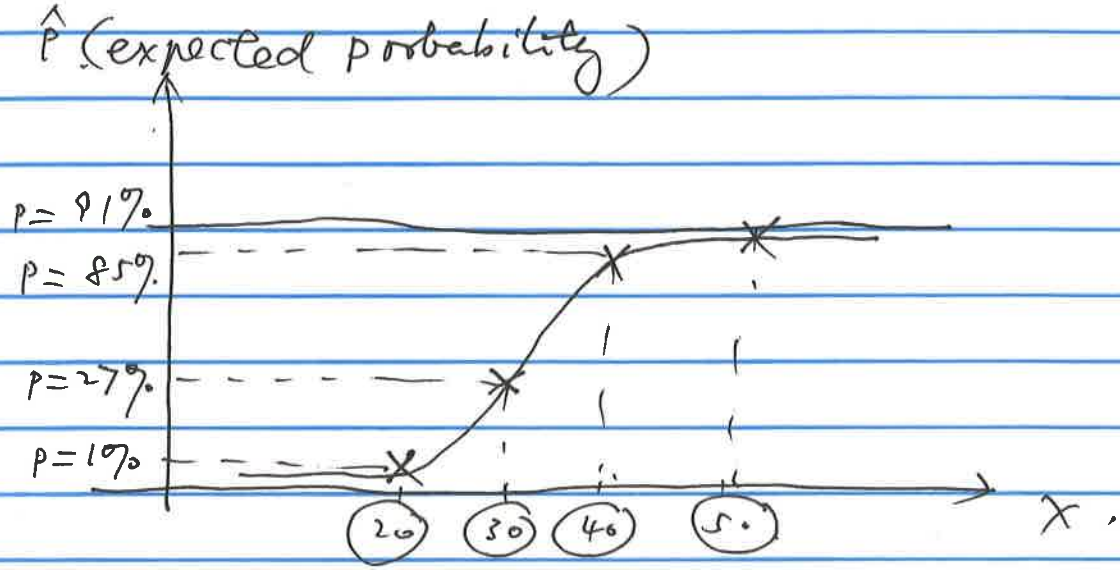
e.g: $P = 1 \Rightarrow \text{odds} = +\infty \Rightarrow \ln(\text{odds}) = +\infty.$

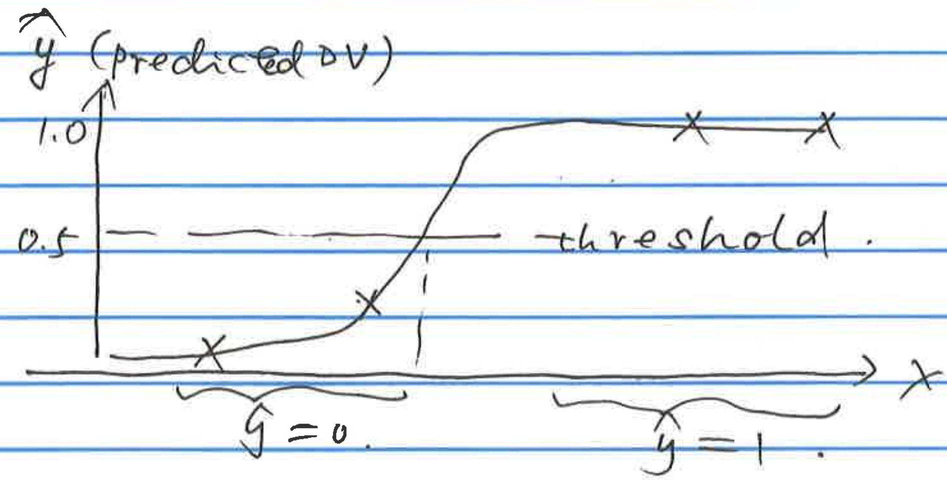
③ $\ln(\text{Odds}) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$

$\Rightarrow \text{odds} = e^{\beta_0 + \beta_1 x_1 + \dots \beta_p x_p}.$

sigmoid

$$\Rightarrow P = \frac{e^{\beta_0 + \beta_1 x_1 + \dots}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots}} = 1 - \frac{1}{1 + e^{\beta_0 + \beta_1 \dots}}$$





Logistic Regression.

① Supervised Learning for binary classification.

② logit = log odds.

$$\text{odds} = \frac{P(\text{event})}{1 - P(\text{event})} \quad (\in(-\infty, +\infty))$$

$$P(X) = Pr(y=1/X)$$

③ Sigmoid func: $P(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$.

$$\Rightarrow \ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x.$$

Parameter Estimation

The goal of Learning is to estimate parameter vector $\hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$.

Logistic Regression use Maximum Likelihood to estimate $\hat{\beta}$.

Given N sample with Label 0/1.

$$\begin{cases} \rightarrow \text{sample labelled "1"}: & \hat{P}(x) \rightarrow 1. \\ \rightarrow \text{sample labelled "0"}: & 1 - \hat{P}(x) \rightarrow 1. \end{cases}$$

likilyhood function

$$L(\vec{\beta}) = \prod_{s \in y_i = 1} P(x_i) \cdot \prod_{s \in y_i = 0} (1 - P(x_i))$$

$$= \prod_{s} P(x_i)^{y_i} \cdot (1 - P(x_i))^{(1-y_i)}$$

$$\Rightarrow \log L(\vec{\beta}) = \sum_{i=1}^{N} \{ y_i \log (p(x_i)) + (1 - y_i) \log (1 - P(x_i)) \}$$

$$= \sum_{i=1}^{N} \left[y_i \log \left(\frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}} \right) + (1 - y_i) \log \left(\frac{e^{-\beta_0 - \beta_1 x_i}}{1 + e^{-\beta_0 - \beta_1 x_i}} \right) \right]$$

$$\Rightarrow \log L(\vec{\beta}) = \sum_{i=1}^{N} \left[y_i (\beta_0 + \beta_1 x_i) x_i - \log \left(1 + e^{\beta_0 + \beta_1 x} \right) \right]$$

$$\vec{\beta} = \underset{\vec{\beta}}{\operatorname{argmax}} \log(L(\vec{\beta})).$$

$$\log(L(\vec{\beta})) = \sum_{i=1}^{N} \Big[y_i(\beta_0 + \beta_1 x_i)\, x_i - \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) \Big].$$

— transcendental Equation

Define $\ell(\vec{\beta}) = \log(L(\vec{\beta}))$.

$$\vec{\beta} = \underset{\vec{\beta}}{\operatorname{argmax}}\ \ell(\vec{\beta}).$$

2 Methods: (Iterative)

1. Gradient.

$$\vec{\beta}^{t+1} = \vec{\beta}^{t} + \alpha \cdot \nabla_\beta \ell(\vec{\beta}^{t})$$

2. Newton Raphson.

$$\vec{\beta}^{t+1} = \vec{\beta}^{t} - \frac{\nabla_\beta \ell(\vec{\beta}^{t})}{\nabla^2_\beta \ell(\vec{\beta}^{t})}.$$

Hessian

$$\nabla_{\vec{\beta}}\, \ell(\vec{\beta}) = \nabla_{\vec{\beta}} \sum_{i=1}^{N} \left[y_i(\beta_0 + \beta_1 x_i) - \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) \right].$$

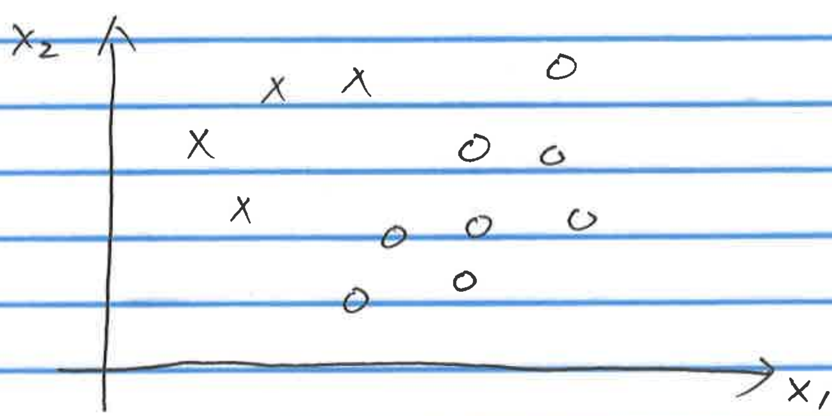
$$= \sum_{i=1}^{N} \left[y_i - p(x_i) \right] \cdot x_i.$$

$$\nabla^2_{\vec{\beta}}\, \ell(\vec{\beta}) = - \sum_{i=1}^{N} p(x_i)\left(1 - p(x_i)\right) x_i^T x_i$$

Hands Exercise of Logistic Regression (Training based on Gradient Ascent)

⓪ Given a training set

x_1	x_2	y
2.7	2.5	0
1.4	2.3	0
3.3	4.4	0
3.06	3.05	0
5.3	2.75	1

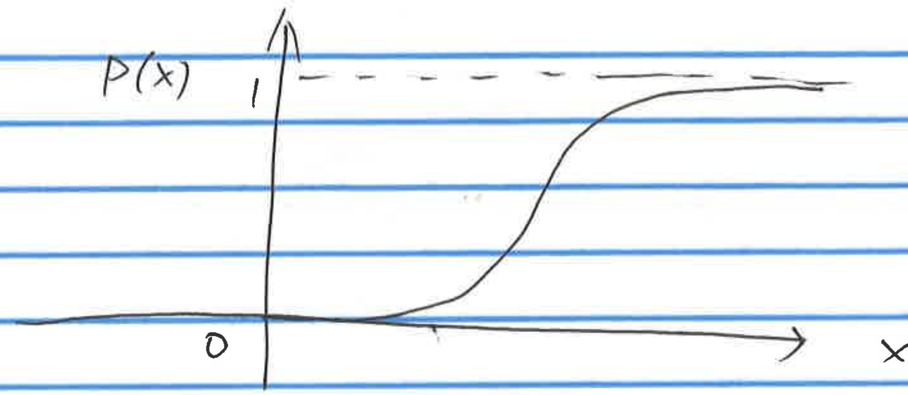


Observations:

① The data is linearly separable

② We need to transform data points using LOGIT or sigmoid functions.

Step ② Find the parameter of model.



$$P(\vec{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

(threshold = 0.5)

✓ Initially, we assume $\beta_0 = 0$. $\beta_1 = 0$. $\beta_2 = 0$.

✓ we will calculate the prediction using the Above model parameter

⇐ e.g. for 1^{st} observation:

$$\boxed{x_1 = 2.7, \quad x_2 = 2.5, \quad y = 0}$$

$$\text{prediction} = \frac{1}{1 + e^{-(0 + 0.0*2.7 + 0.0*2.5)}} = 0.5$$

Next we will update parameter via:

learn_rate

$$\beta = \beta + \alpha * (y - \text{prediction}) * \text{prediction} * (1 - \text{prediction}) * x$$

④ β_0, β_1, or β_2 $\Rightarrow$ $\beta_0 = -0.0375$, $\beta_1 = -0.1$; $\beta_2 = -0.095$

step 3

use the newly parameter

compute the prediction again.

☆ One round of calculating the predicted value for all training set using newly updated parameter is "Epoch".