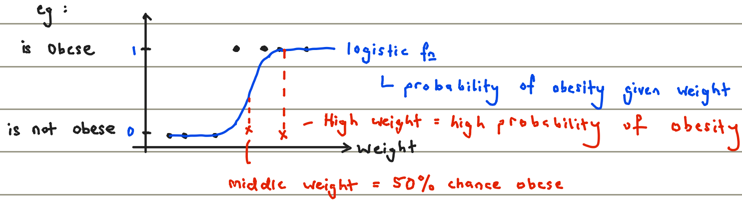
Logistic Regression

- Predicts whether something is True or False

eg :



- Typically used for classification
- Can have multi feature prediction

eg : obese → weight

obese → $\underbrace{\text{weight + Genotype + Age + weather}}_{\text{can be numerical / categorical}}$ ← may not be relevant

- Cannot compare $\underbrace{\text{complicated model}}_{\text{many features}}$ with $\underbrace{\text{simple model}}_{\text{lesser features}}$ easily

Wald's Test : To determine if feature is impt

- Logistic Regression no Residual (No least squares)
- Instead use maximum likelihood

How Logistic Regression Works

1) Logistic Regression starts with linear model

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_n x_n$$

↑ Intercept $\beta_1, \beta_2, \beta_3$: coefficients for predictor

2) Apply sigmoid fn : $\dfrac{1}{1+e^{-z}}$

└ map linear output z to a probability

3) set Decision Boundary (~ 0.5)

Training model :

Goal : Find optimal coefficients $\beta_0, \beta_1 \dots \beta_n$ by minimizing log - loss ← via gradient descent

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \ln(P(y_i \mid x_i)) + (1-y_i)\ln(1-P(y_i \mid x_i))]$$

N : No. samples

y_i : Actual sample

$P(y_i \mid x_i)$: Predicted Probability

Training process (continuous variable)

1) Init values for coefficients ($\beta_1 \dots \beta_n$) & Intercept β_0

↳ typically 0 or small random values

2) For each data point x_i, compute the predicted probability using sigmoid:

$$P(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 \dots + \beta_n x_n)}}$$

3) compute log loss : $-\frac{1}{m}\sum_{i=1}^{m} [y_i \log(p(x_i)) + (1-y_i)\log(1-p(x_i))]$

y_i : Actual label

$p(x_i)$: Predicted probability

4) Use gradient descent to minimize log loss

$$\frac{\partial}{\partial \beta_j}(\log \text{loss}) = \frac{1}{m}\sum_{i=1}^{m}(p(x_i) - y_i)\, x_{ij}$$

← j-th feature for i^{th} data point

$$\beta_{j\,new} = \beta_{j\,old} - \alpha \left(\frac{1}{m}\sum_{i=1}^{m}(p(x_i) - y_i)\, x_{ij}\right)$$

↑ learning rate

5) Iterative optimization

↳ repeat gradient computation & coefficient update
until log loss minimized