Logistic regression

- It is a modification of linear regression used to predict the value of a binary variable, and thus it is useful for binary classification problems.

Relies on the logistic function

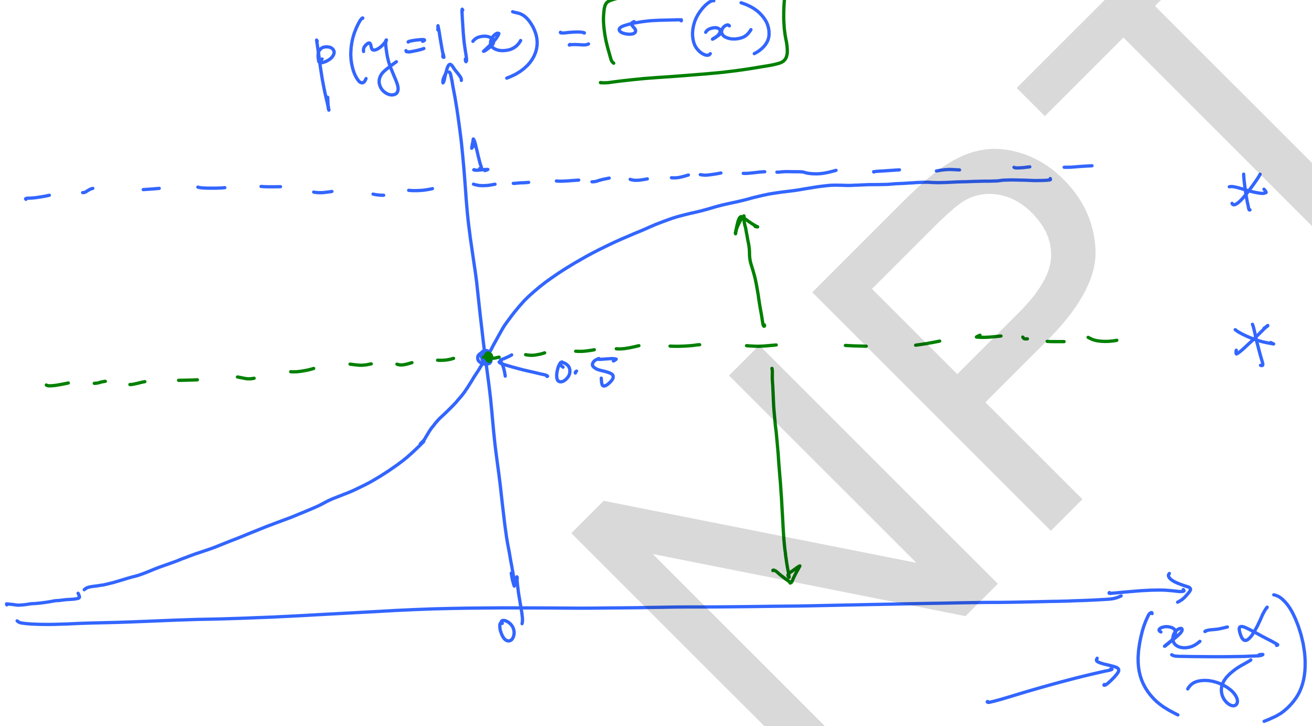
$$p(x) = \frac{1}{1 + \exp\left(-\frac{x-\alpha}{\gamma}\right)}$$

which is used to model the probability of the target variable taking the value 1. given a feature vector x.

Class I : $y = 0$

Class II : $y = 1$

$$\boxed{P(y=1 | x)}$$



* $x = \alpha$ is where $p = 0.5$

* γ is the width of the logistic function

Logistic function is a type of sigmoid

$P(y=1|x) < \boxed{0.5}$: Class I $(\hat{y} = 0)$

$P(y=1|x) \geq \boxed{0.5}$: Class II $(\hat{y} = 1)$.

classification threshold probability (p^*)

$$P(y=1|x) = \frac{1}{1 + \exp\left(-(\underline{\beta_0} + \underline{\beta}^T x)\right)}$$

$$\beta_0 = -\sum_{i=1}^{p} \frac{\alpha_i}{\gamma_i} \quad \text{and} \quad \beta = \left[\frac{1}{\gamma_1}, \frac{1}{\gamma_2}, \ldots, \frac{1}{\gamma_p}\right]$$

Binary cross-entropy loss function:

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

$$\hat{p}_i = P(y=1 \mid x = x_i)$$

y_i: true class of the datapoint.

→ Model is trained by numerical optimization.

Example:

(a) When $y_i = 1$ & $\hat{p}_i = 0$

$$l_{log.i} = -(1\log(0) + 0\log(1)) = -(-\infty + 0) \longrightarrow \infty$$

(b) When $y_i = 0$ & $\hat{p}_i = 1$

$$l_{log,i} = -\left(0 \log(1) + 1 \boxed{\log(0)}\right) = -\left(0 + (-\infty)\right) \rightarrow \infty$$

(c) When $y_i = 0$ and $\hat{p}_i = 0$

$$l_{log,i} = -\left(\underline{0 \log(0)} + 1 \log(1)\right) = -(0 + 0) = \boxed{0}$$

(d) When $y_i = 1$ & $\hat{p}_i = 1$

$$l_{log,i} = -\left(1 \log(1) + 0 \log(0)\right) = \boxed{0}.$$

Assessing the accuracy of a binary classification model:

Errors

- False positive (FP): $y = 0, \hat{y} = 1 \longrightarrow$ Type I error.
- False negative (FN): $y = 1, \hat{y} = 0 \longrightarrow$ Type II error.

Correct predictions

- True positive (TP): $y = 1, \hat{y} = 1$
- True negative (TN): $y = 0, \hat{y} = 0$.

Confusion matrix:

Confusion Matrix

$\hat{y}$ \ y	True values: 1	True values: 0
Predicted values: 1	True positive (TP)	False positive (FP)
Predicted values: 0	False negative (FN)	True negative (TN)

$$\text{Accuracy} = \left(\frac{TP + TN}{TP + TN + FP + FN} \right)$$

Simple accuracy fails in imbalanced datasets.
(eg. 95% semiconductors, 5% of metals) $\longrightarrow$ a model that always predicts semiconductors would have high accuracy!

Balanced accuracy is used instead.

Sensitivity (also called recall) = True positive rate (TPR)

$$= \left(\frac{TP}{TP + FP} \right)$$

$\longrightarrow$ of the predicted positives, how many are correct?

Specificity $\equiv$ True negative rate (TNR)

$$= \left(\frac{TN}{TN + FP} \right)$$

Balanced accuracy $= \left(\dfrac{\text{Sensitivity} + \text{Specificity}}{2} \right)$

$$= \left(\frac{TPR + TNR}{2} \right).$$

useful in imbalanced datasets.