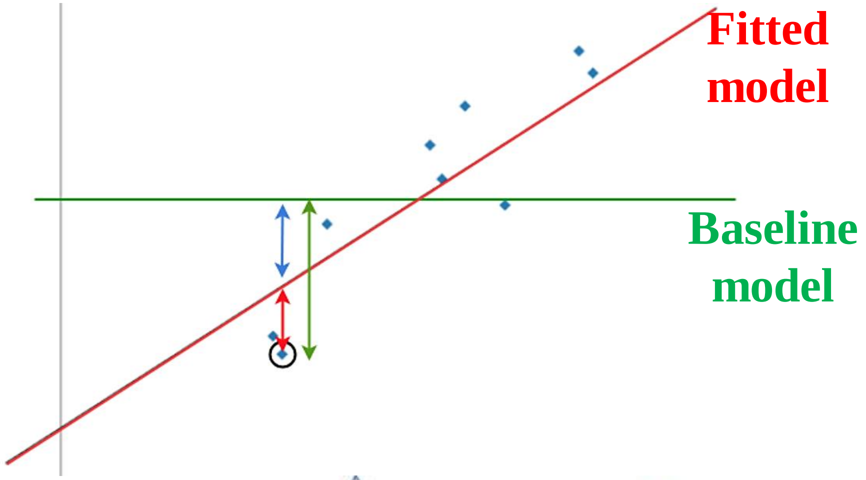
Linear Regression - Evaluation Metric: R^2 (Goodness-of-Fit)

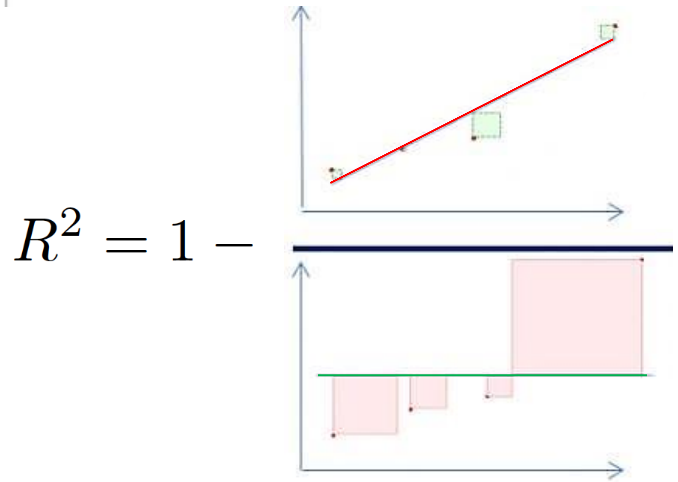
- **How well** the **regression line fits** the data wrt:
  - The correlation between the true value of the response variable and the predicted value
  - Baseline model's performance



RESidual **S**um of **S**quared errors of the regression model

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

TOTal **S**um of **S**quared errors that compares the actual y values to the baseline model (the mean)

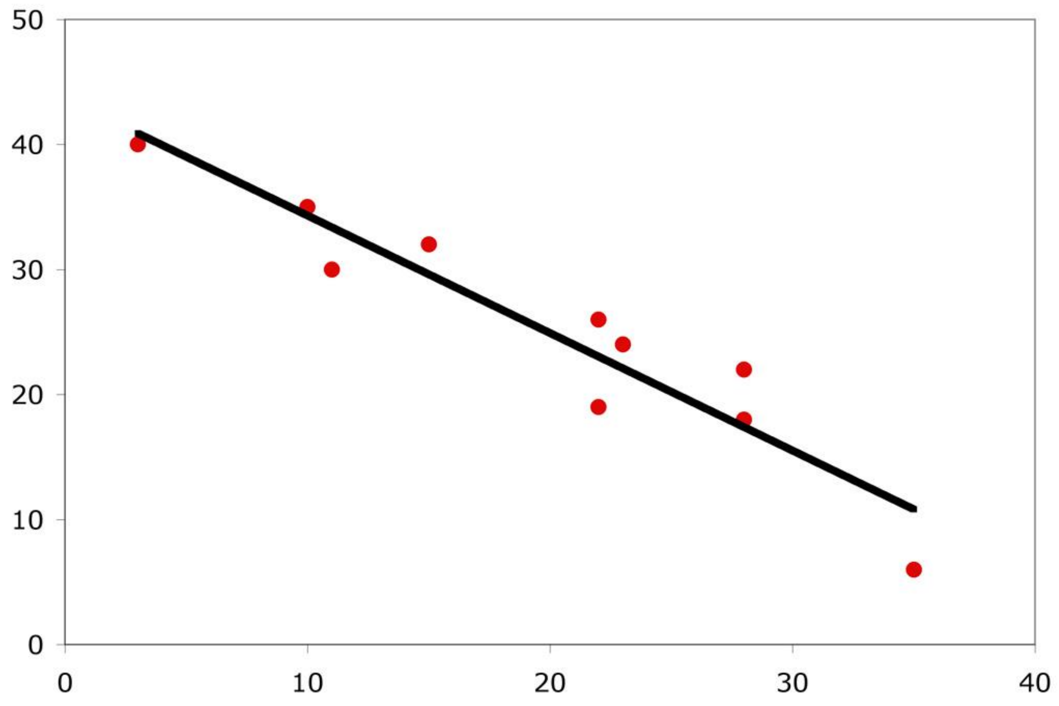


Question: What R^2 value should the model get closer to? 0 or 1

Linear Regression - Evaluation Metric: R^2 – Example

X	Y
3	40
10	35
11	30
15	32
22	19
22	26
23	24
28	22
28	18
35	6

Equation for Line of Best Fit: $y = -0.94x + 43.7$



Linear Regression - Evaluation Metric: R^2 – Example Cont.

X	Y	Predicted Y $y = -0.94x + 43.7$	Error	Error Squared	Distance between Y values and their mean	Mean distances squared
3	40					
10	35					
11	30					
15	32					
22	19					
22	26					
23	24					
28	22					
28	18					
35	6					
Mean:			**Sum:**		**Sum:**	

Linear Regression: $R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$ – Example

X	Y	Predicted Y $y = -0.94x + 43.7$	Error	Error Squared	Distance between Y values and their mean	Mean distances squared
3	40	40.88	.88	.77	14.8	219.04
10	35	34.30	-.70	.49	9.8	96.04
11	30	33.36	3.36	11.29	4.8	23.04
15	32	29.60	-2.40	5.76	6.8	46.24
22	19	23.02	4.02	16.16	-6.2	38.44
22	26	23.02	-2.98	8.88	.8	.64
23	24	22.08	-1.92	3.69	-1.2	1.44
28	22	17.38	-4.62	21.34	-3.2	10.24
28	18	17.38	-.62	.38	-7.2	51.84
35	6	10.80	4.8	23.04	-19.2	368.65
Mean:	**25.2**		**Sum:**	**91.81**	**Sum:**	**855.60**

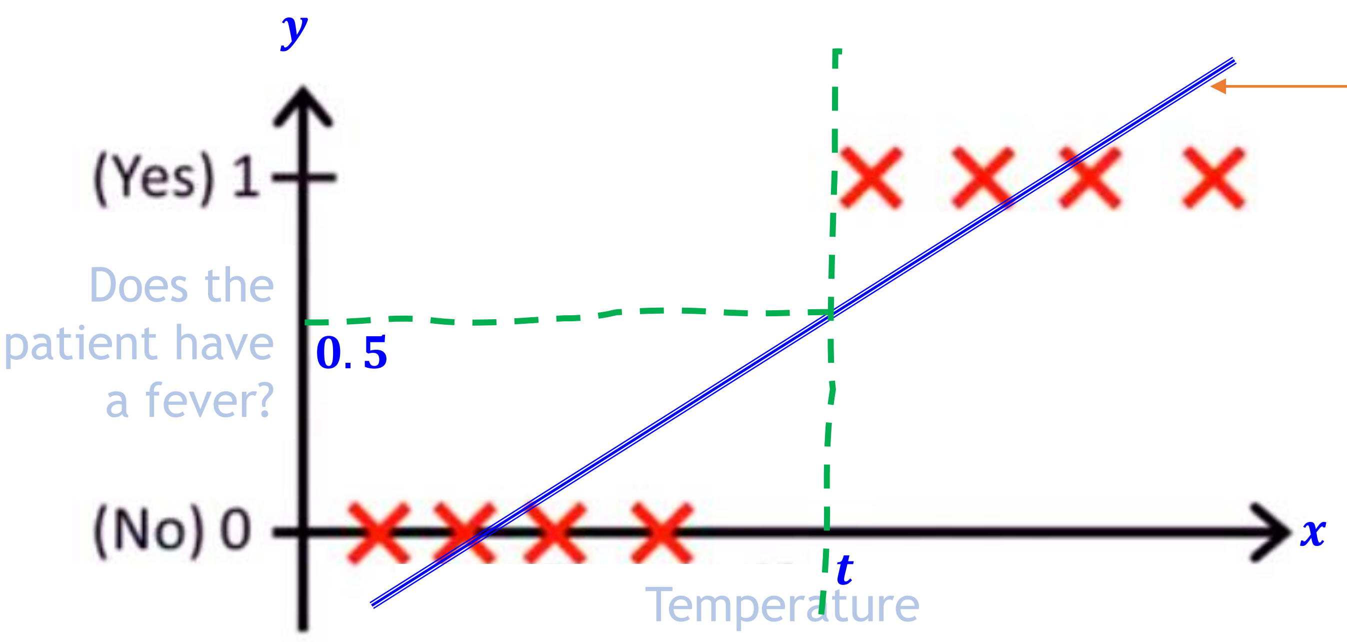
Logistic Regression
It is not a regressor! It is a classifier!!

Classification

- Disease: Exist or not
- Email: Spam or Ham
- Weather: Rain or Sunny
- Transaction: Fraudulent or Genuine
- Income: Wealthy or Poor
- Target variable $y \in \{0, 1\}$
  - 0 → Negative class, e.g., Ham
  - 1 → Positive class, e.g., Spam

binary classification problem with
class probability estimation

Binary Classification – Example



The fitted $h_\beta(x)$ using training data

- How to classify the samples:

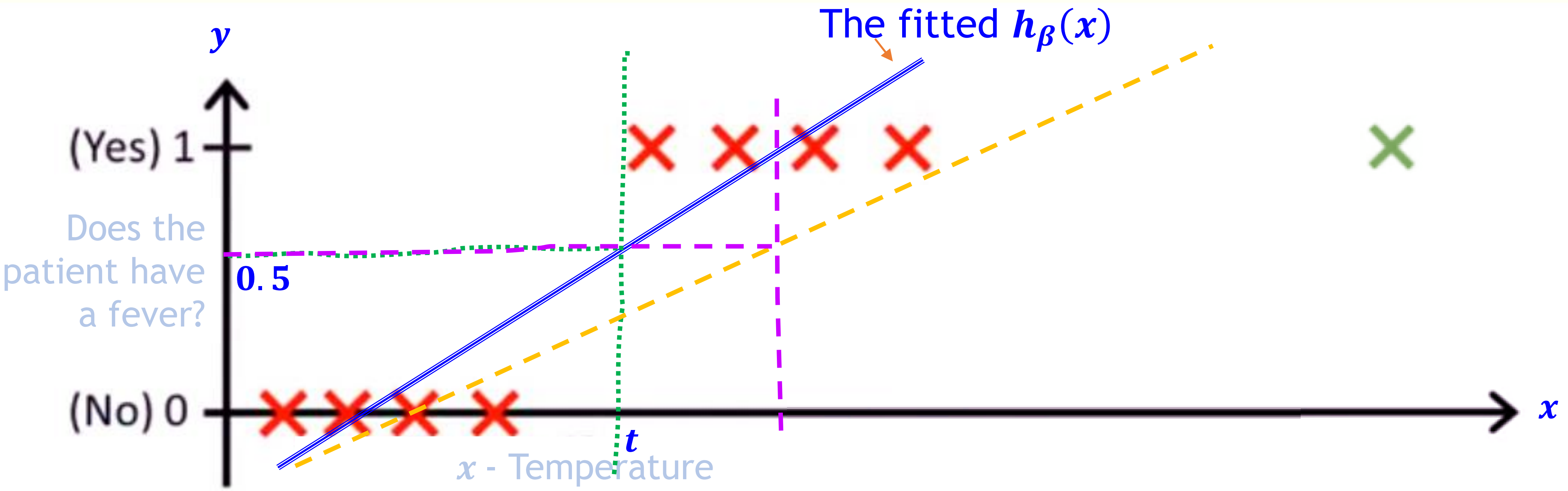
  - Set a threshold, $(\tau = 0.5)$
    - If $h_\beta(x) \geq \tau$ → y = 1
    - Else $h_\beta(x)$ → y = 0

- Samples to the left of t belongs to class 0 and
- Samples to the right of t belongs to class 1

- Let's approach this problem from what know
- Apply the ML model we learnt – LM:

$$h_\beta(x_i) = \sum_{j=1}^{p} x_{i,j} \cdot \beta_j$$

Binary Classification – Example Cont.



- Let's test it with a new sample
- From the set threshold, we know:
  - X < t belongs to class 0
  - X ≥ t belongs to class 1

- What if X was part of training sample

  New fitting causes miss classification

Binary Classification Cont.

- Observation
  - Target variable: y = 0 or y = 1
  - In LM, $h_\beta(x)$ can results a value < 0 or >1
  - It is not good enough to have the prediction in [0, 1]

- Solution
  - Logistic regression
    - $0 \leq h_\beta(x) \leq 1$

Logistic Regression Model Description

- It is based on the logistic (sigmoid) function $\sigma(z) = \frac{e^z}{1+e^z}$ for - ∞<z<∞.

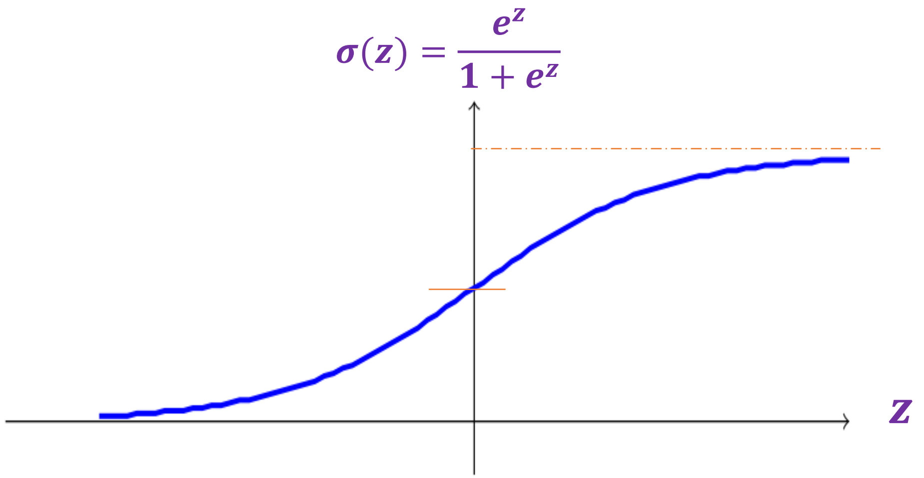
- To predict the likelihood of an outcome, y needs to be a function of the input variables, x.

- $z = h_\beta(x) \rightarrow$ linear function of the input variables:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} = X \cdot \beta.$$

- Based on the input variables, $\mathrm{X} = \{x_1, x_2, \dots, x_p\}$, and the set of parameters, β the probability of an event, **y** is given as:

$$p(y|X;\beta\}) = \sigma(z) = \frac{e^z}{1+e^z}$$



value of the logistic function varies from 0 to 1, as z increases

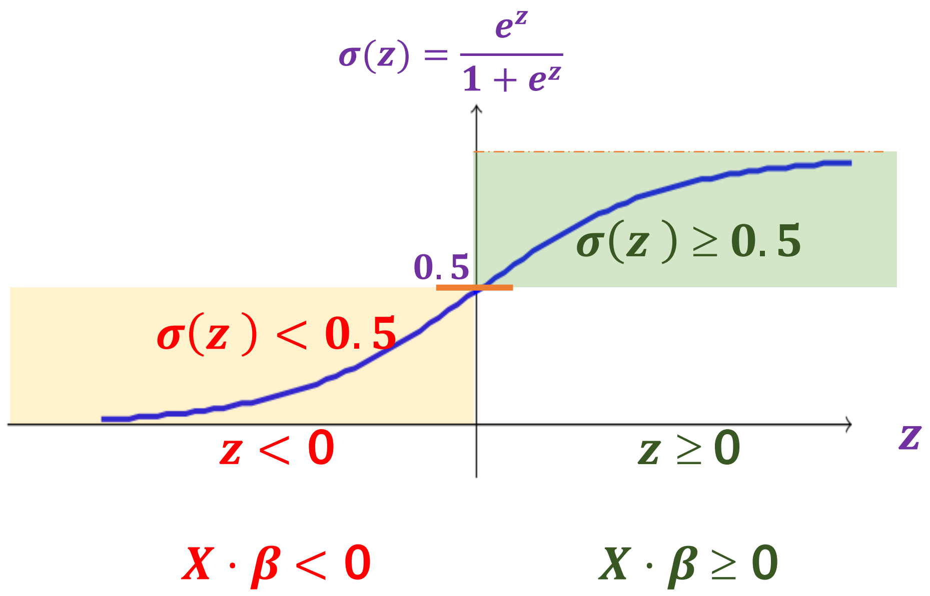
Logistic Regression - Classification

- For set of input variables, $\mathrm{X} = \{x_1, x_2, \dots, x_p\}$, and the set of parameters, $\boldsymbol{\beta}$ the probability of an event, **y** is given as:

$$p(y|X;\boldsymbol{\beta}\}) = \sigma(z) = \frac{e^z}{1+e^z}$$

- By setting a **threshold**, τ one can easily convert the likelihood probability, $\sigma(z)$ into a binary classification label.

- **Example:**
  - Predict “y=1” if $\sigma(z) \geq 0.5$
  - Predict “y=0” if $\sigma(z) < 0.5$



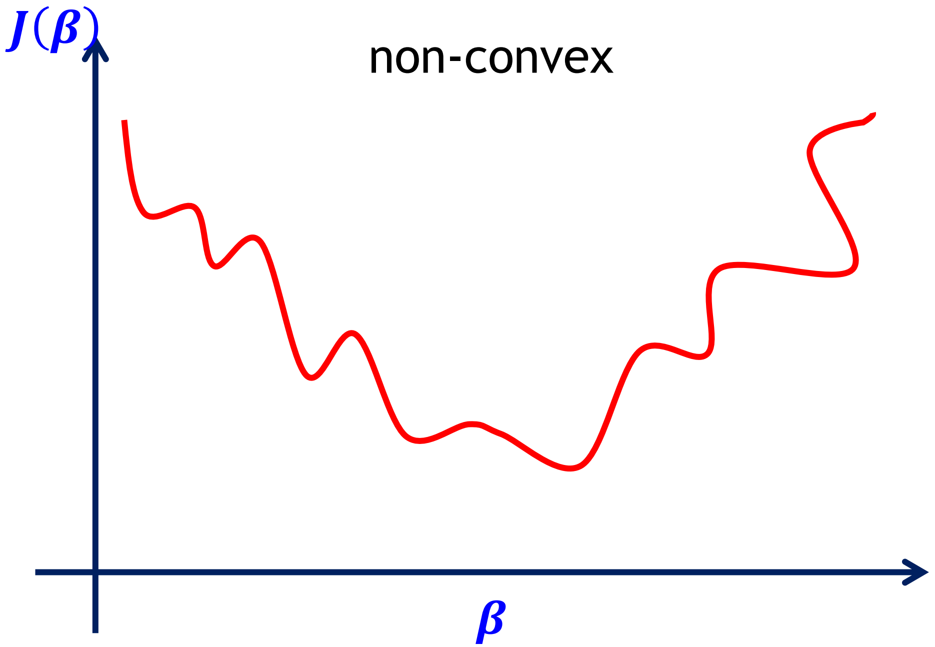
- **How do we estimate the best parameter $\boldsymbol{\beta}$?**
  - Consider an objective function, $J(\boldsymbol{\beta})$
  - Apply an optimizer (min or max) accordingly

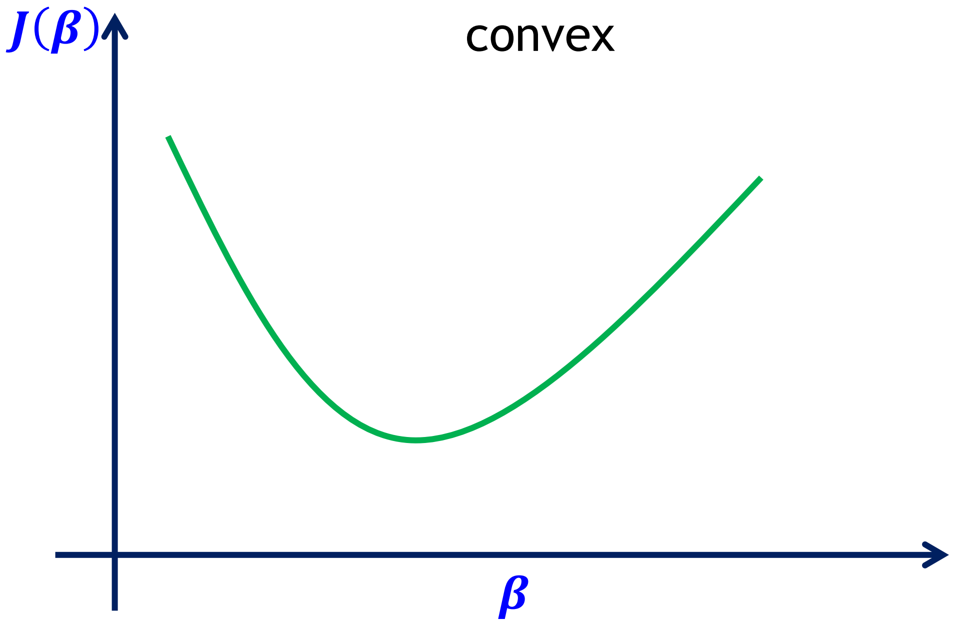
Logistic Regression – Objective Function

model

- From linear regression what we know: $J(\beta) = \frac{1}{n}\sum_{i=1}^{n}[h_\beta(x_i) - y_i]^2$

- Change the **model** to sigmoid function: $J(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{e^z}{1+e^z} - y_i\right]^2 ; z = h_\beta(x_i)$





Logistic Regression: Maximum Likelihood Estimator

- The sigmoid classifier is fit through **learning** the best values for the parameters $\boldsymbol{\beta}$ by **maximizing** the **log** (joint) **conditional likelihood** probabilities of the two classes.

- Given training sample $\langle x_i | y_i \rangle$ and assume y can only take two values of 0 or 1, the log conditional likelihood is:

  $\boldsymbol{log(p_i)} \leftarrow$ if $\boldsymbol{y_i = 1}$ and
  $\boldsymbol{log(1 - p_i)} \leftarrow$ if $\boldsymbol{y_i = 0}$.

  Note: $p_i = p(y = 1 | x_i ; \beta)$, i.e., probability function of y=1 given x_i parameterized by β.

- Then total log conditional likelihood (LCL):

$$LCL = \sum_{i:y_i=1} \log p_i + \sum_{i:y_i=0} \log(1 - p_i)$$

sum of the log conditional likelihood, by grouping together, the positive and negative training samples

Logistic Regression: MLE Cont.

- Unifying the individual class likelihood (l):

$$l(\beta) = \sum_{i=1}^{n} y_i \cdot log(p(x_i)) + (1 - y_i) \cdot log(1 - p(x_i))$$

<u>Note:</u> it is equivalent to previous eq. since, if y_i = 1 (true), then $1 - y_i$ = 0

- Now, let's substitute the expression for $p(y|x;\ \beta)$:

$$l(\beta) = \sum_{i=1}^{n} y_i \cdot log\left(\frac{1}{1 + e^{-\beta_i X_i}}\right) + (1 - y_i) \cdot log\left(1 - \frac{1}{1 + e^{-\beta_i X_i}}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i \cdot log\left(\frac{1}{1 + e^{-\beta_i X_i}}\right) + (1 - y_i) \cdot log\left(\frac{e^{-\beta_i X_i}}{1 + e^{-\beta_i X_i}}\right)$$

Why Logistic Regression: MLE Cont.

- $l(\beta) = \sum_{i=1}^{n} y_i \cdot log\left(\frac{1}{1+e^{-\beta_i X_i}}\right) + (1 - y_i) \cdot log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right)$ ← from previous slide

- Now, take y_i common term:

$$l(\beta) = \sum_{i=1}^{n} y_i \left[log\left(\frac{1}{1+e^{-\beta_i X_i}}\right) - log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right)\right] + log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right)$$

- Further simplify:

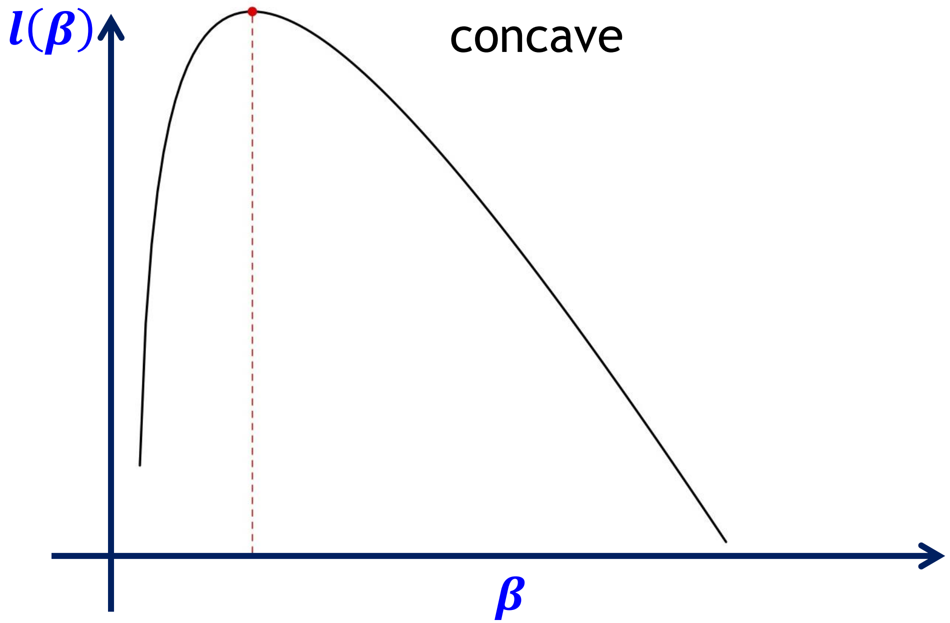
$$l(\beta) = \sum_{i=1}^{n} y_i\left[log\left(e^{\beta_i X_i}\right)\right] + log\left(\frac{e^{-\beta_i X_i}}{1+e^{-\beta_i X_i}}\right) = \sum_{i=1}^{n} y_i \beta X_i + log\left(\frac{1}{1+e^{\beta_i X_i}}\right)$$

$$\boxed{l(\beta) = \sum_{i=1}^{n} y_i \beta X_i - log\left(1 + e^{\beta_i X_i}\right)}$$

- **Optimal $\beta'_j s$:** Find via maximizing the objective function → $\max_{\beta} l(\beta)$

Logistic Regression: Maximum Likelihood Estimator Cont.

- $l(\beta) = \sum_{i=1}^{n} y_i \beta X_i - \log\left(1 + e^{\beta_i X_i}\right)$



- Now, we choose values of β that make this equation as large as possible: $\beta = \underset{\beta}{\arg\max}\, l(\beta)$
- Maximizing involves derivatives over multiple iterations.
- E.g., stochastic gradient **ascent**:

$$\beta_j := \beta_j + \lambda \frac{\partial}{\partial \beta_j} LCL$$

$l(\beta)$

  - The gradient-based update of the parameters
  - Slightly changes the parameter values to increase the log likelihood based on one example at a time.

Logistic Regression: Diagnostics

- **What we know:**
  - **Sigmoid classifier** is to assign class labels based on the **predicted probability,** $\sigma(z)$.
  - E.g., a customer can be classified with the label called "**Churn**" if $\sigma(z) \geq \tau$ (a high probability) that the customer will churn. Otherwise, i.e., $\sigma(z) < \tau$ a "**Remain**" label is assigned to the customer.
  - Generally, τ = **0.5** is used as the **default threshold** to distinguish between any two class labels.

- **Application specific τ:**
  - to **avoid false positives** (e.g., predict Churn when actually the customer will Remain)
  - to **avoid false negatives** (e.g., predict Remain when the customer will actually Churn).

- **How do we set an application specific τ:**
  - Using ROC graph, we can find it.
  - A ROC graph is a 2D plot that **summarizes a classifier performance over various threshold values** with **false positive rate** on the x axis **against true positive rate** on the y axis