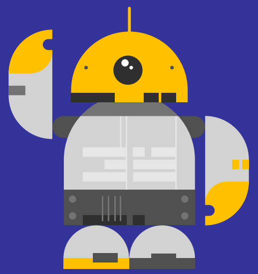


Database Systems

Lecture 8 – Cont. Machine Learning for Data Analytics

Dr. T. Akilan
takilan@lakeheadu.ca

This Session

- Taking the missed cosine similarity
- Quick recap on Regression
  - Linear regression
  - Goodness-of-Fit - R^2
- Classification
  - Logistic regression
  - Maximum likelihood estimation
- Evaluation metrics
  - ROC
  - AUC

Cosine Distance Similarity

- Good for dealing with sparse matrix of attributes
  - Large matrix with a lot of zeros
- **Example**: term-frequency vector
  - Typical in Natural Language Processing
  - Represent text document by counting words

Document	**team**	**coach**	**hockey**	**baseball**	**soccer**	**penalty**	**score**	**win**	**loss**	**season**
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Document Vector or Term-Frequency Vector

Cosine Similarity

- Cosine measure as a similarity function: $sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}||\,||\boldsymbol{y}||}$

$$\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$$

- E.g.,:

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- **How similar are Doc 1 and Doc2?**
- Let x and y represent the feature vectors of Doc 1 and Doc2:

$$\begin{aligned} x &= (5, 0, 3, 0, 2, 0, 0, 2, 0, 0) \\ y &= (3, 0, 2, 0, 1, 1, 0, 1, 0, 1) \end{aligned}$$

Now, compute the cos-sim(x. y):

$$\boldsymbol{x}^t \cdot \boldsymbol{y} = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25$$

$$||\boldsymbol{x}|| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$$

$$||\boldsymbol{y}|| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$$

$$sim(\boldsymbol{x}, \boldsymbol{y}) = 0.94$$

Next → Getting back to ML Models

Recap - Model Description of Linear Regression

- **Assumption:** a linear relationship between the input variables and the outcome variable.

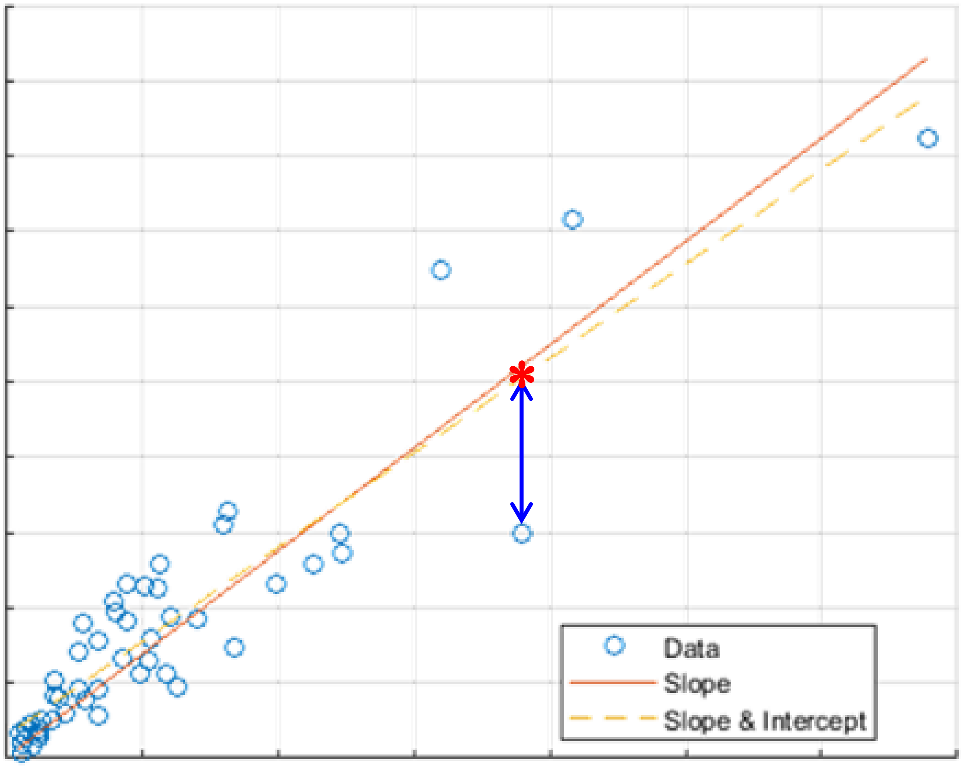
- **General model:**

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

- **Cost function (e.g., MSE):**

$$J(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left[h_{\beta}(x_i) - y_i\right]^2$$

- **Optimal $\beta_j^{'}$s:** Find via minimizing the cost function → $\min_{\beta} J(\beta)$



Linear Regression w/ Categorical Variables

$$\text{income} = b_0 + b_1\text{age} + b_2\text{yearsOfEducation} + b_3\text{gender} + b_4\text{state}$$

- ***Gender*** is **categorical**, but **binary**
  - one variable: *Male*, which is 0 for females
- ***State*** is a **categorical** variable:
  - **50 possible values**
  - Expand it to 49 indicators (0/1) variables:
  - The remaining level is the **default level**, i.e., all indicators set to 0

```
results3 <- lm(Income~Age + Education,Gender,
     + Alabama,
     + Alaska,
     + Arizona,
     .
     .
     .
     + WestVirginia,
     + Wisconsin,
     income_input)
```

In regression, a proper way to implement a categorical variable that can take on m **different values** is to add $m - 1$ **binary variables** to the regression model.

Linear Regression - Overfitting

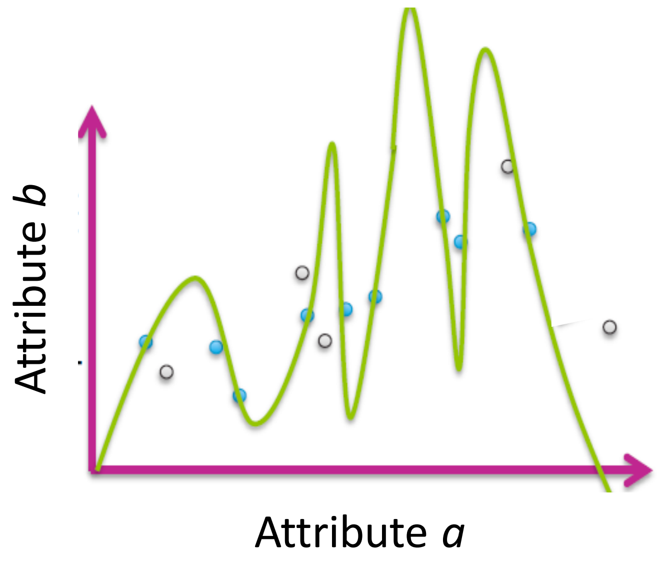
- Overfitting associated with **too many regression coefficients** to be estimated.

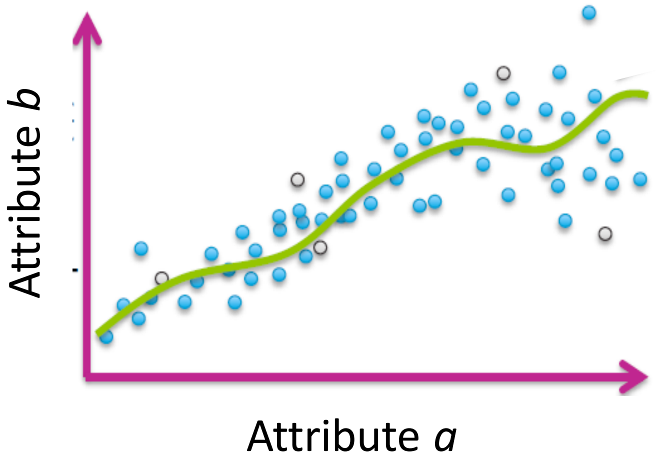
- Just adding more variables to explain a given dataset may not improve the explanatory nature of the model.

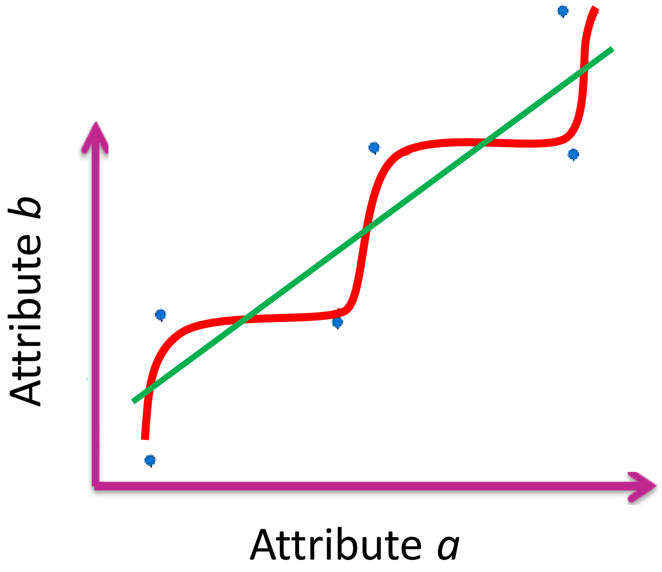
- Example: $f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + w_3x_3$
  - Let's add a fourth attribute $x_4 = x_1^2$ and add another new attribute $x_5 = \frac{x_2}{x_3}$
  - Now, the model needs to learn the parameters (weights) of the following $f(x)$. $f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$

  - Potentially, it can lead to overfitting and reduce model's generalizability outside the original dataset.

Linear Regression - Overfitting







- Few observations (small N)
- ➢ rapidly overfit, as model complexity increases

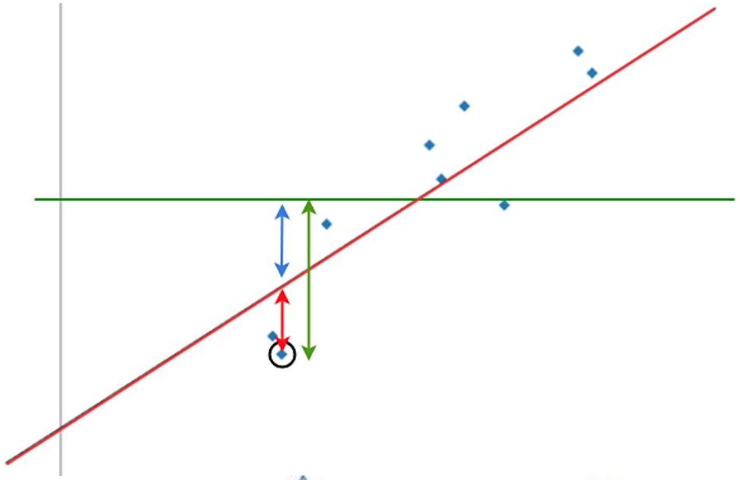
- Many N (very large N)
- ➢ harder to overfit

- Red model is overfitted, since it almost memorized all the data points.
- Green model can be an optimal solution.

Residual sum of squared errors of the regression model

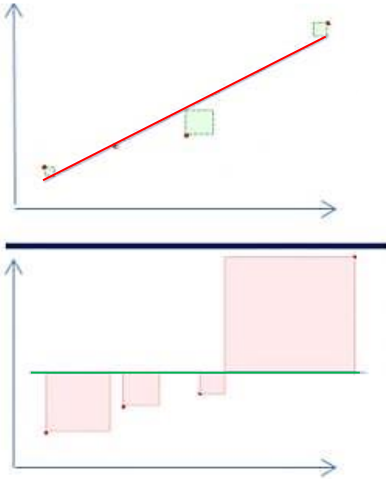
$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

total sum of squared errors that compares the actual y values to the baseline model the mean



- It is also the square of the correlation between the true output and the predicted output
- R-Squared checks if the fitted regression line will predict better than the mean line
- How well the regression line fits the data.

$R^2 = 1 -$



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

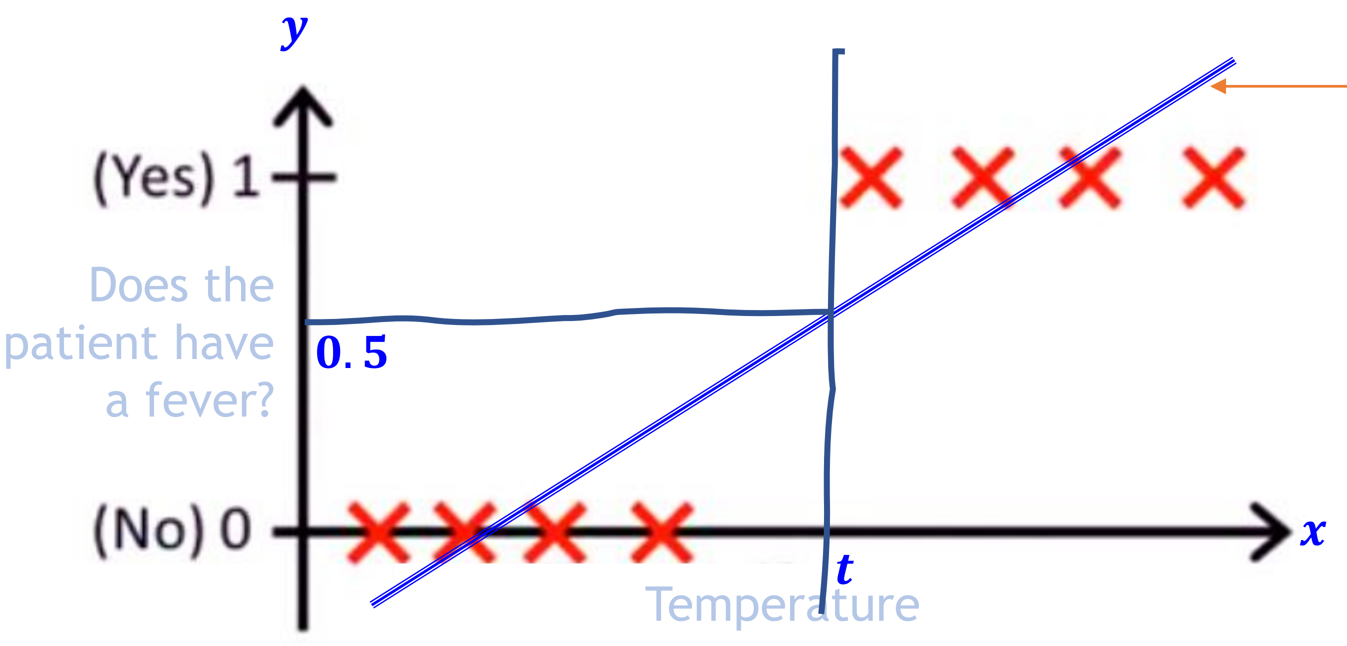
Logistic Regression
It is not a regressor! It is a classifier!!

Classification

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

a binary classification problem

Binary Classification



The fitted $h_\beta(x)$ on the Training data

- How to classify the samples:

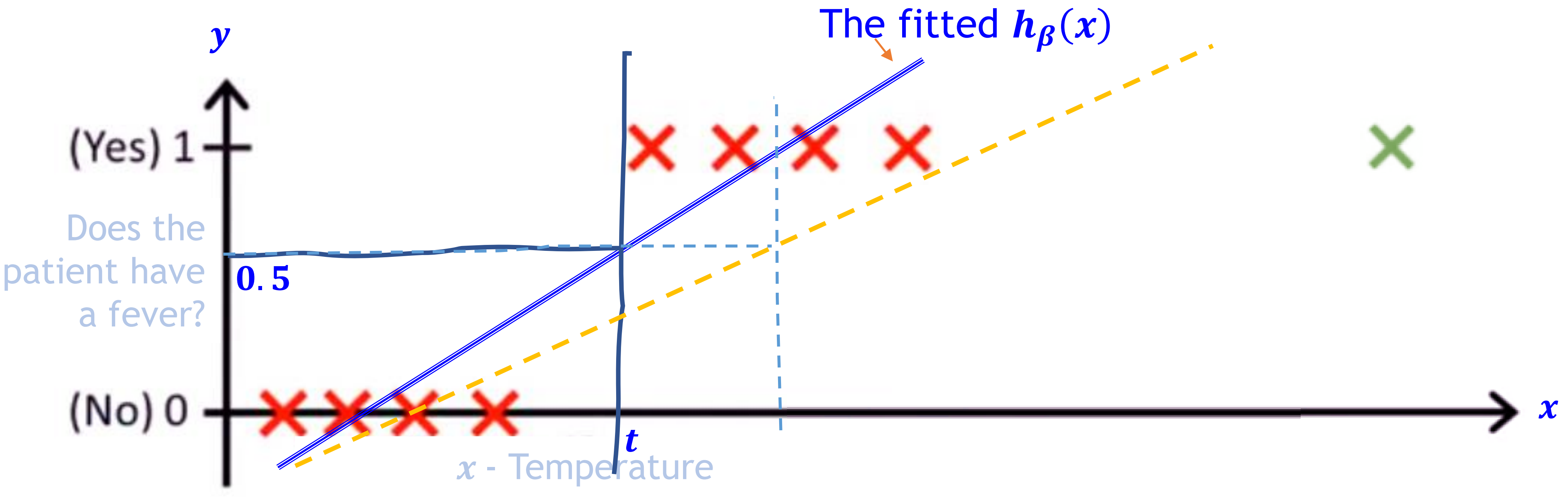
  - Set a threshold, $(\tau = 0.5)$
    - If $h_\beta(x) \geq \tau$ → y = 1
    - Else $h_\beta(x)$ → y = 0

- 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 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<∞.

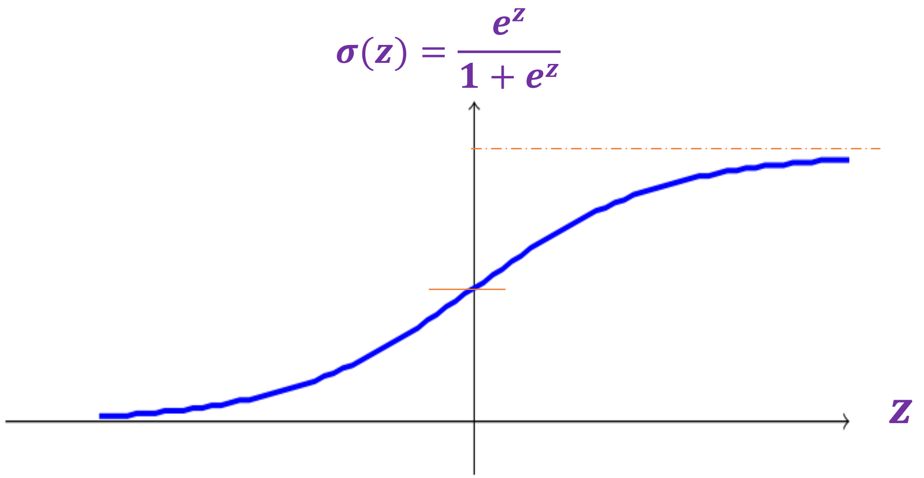
- To predict the likelihood of an outcome, y needs to be a function of the input variables, $\boldsymbol{x}$.

- $z = \ h_\beta(x)$ → 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, $\mathrm{X} = \{x_1, x_2, \ldots, x_p\}$, 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

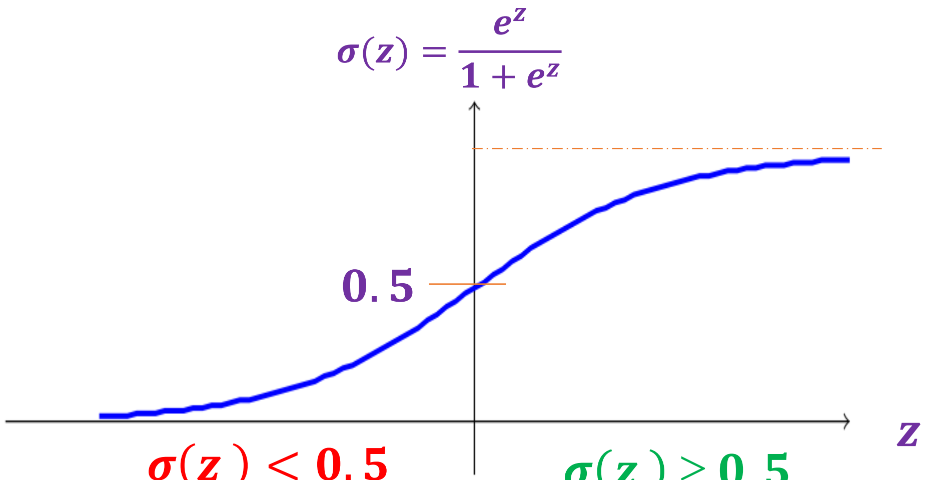
Logistic Regression - Classification

- 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:

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

- 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 $\sigma(z) < 0.5$



$\sigma(z) < 0.5$	$\sigma(z) \geq 0.5$
$z < 0$	$z \geq 0$
$X \cdot \beta < 0$	$X \cdot \beta \geq 0$