Supervised Learning

Achmad Wildan Al Aziz
Universitas Airlangga, 27 November 2018

Outline

1. Introduction
2. Logistic Regression
3. K-Nearest Neighbour
4. Support Vector Regression
5. Evaluation Metrics
6. Cross Validation

Machine Learning

DATA SCIENCE INDONESIA

Supervised

Unsupervised

Clasification

Parametric

- Logistic Regression
- Discriminant Analysis

Non Parametric

- SVM
- Deep Learning
- KNN
- etc

Regression

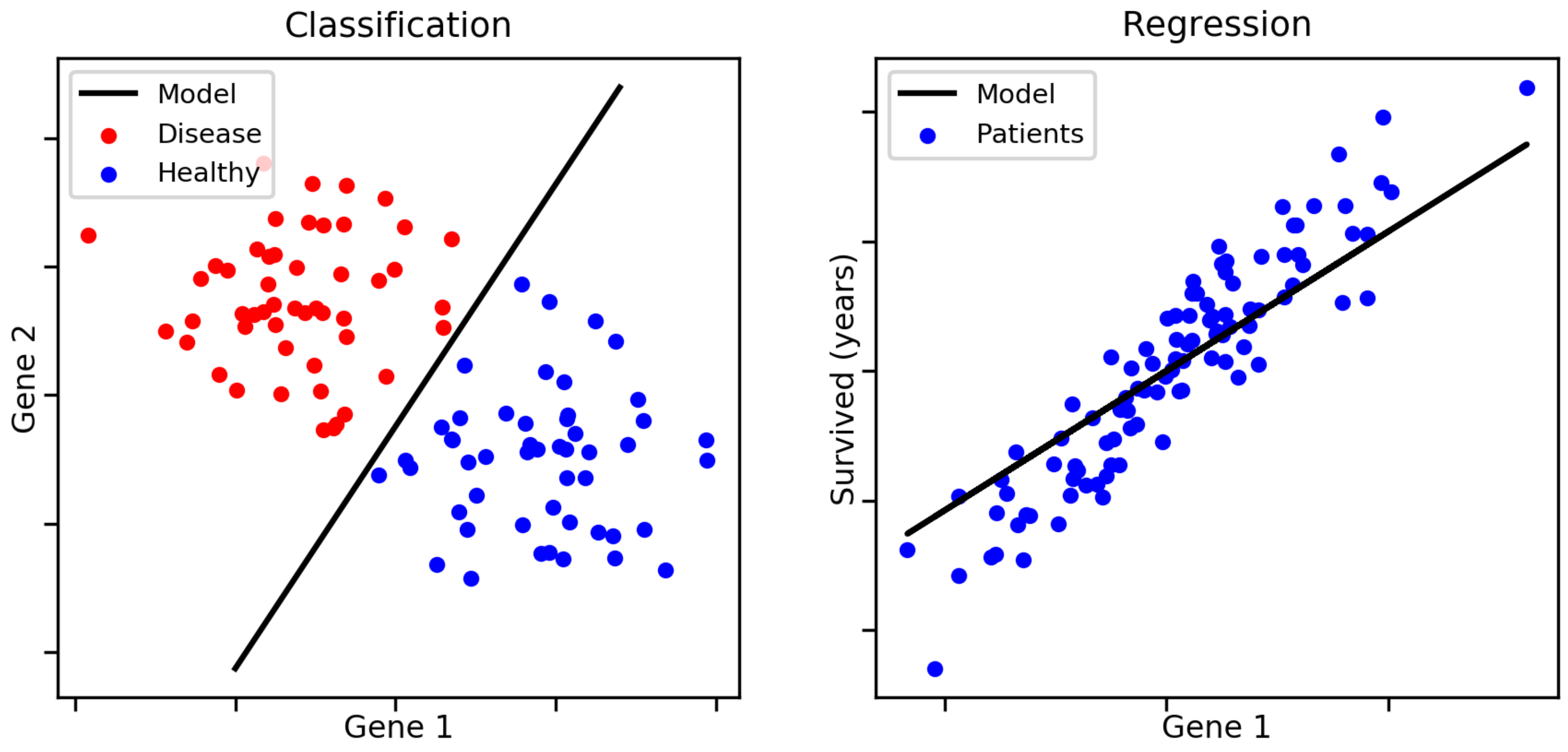
Parametric

- Multiple Linear Regression

Non Parametric

- SVR
- Deep Learning
- KNN
- etc

Introduction



https://aldro61.github.io/microbiome-summer-school-2017/figures/figure.classification.vs.regression.png

Data Structure

Observations	Y	X_1	X_2	.	.	.	X_k
1	Y_1	X_{11}	X_{12}	.	.	.	X_{1k}
2	Y_2	X_{21}	X_{22}	.	.	.	X_{2k}
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
n	Y_n	X_{n1}	X_{n2}	.	.	.	X_{nk}

Classificatin Case

Y: Nominal / Ordinal

X: Nominal / Ordinal / Interval / Rasio

Regression Case

Y: Interval / Rasio

X: Nominal / Ordinal / Interval / Rasio

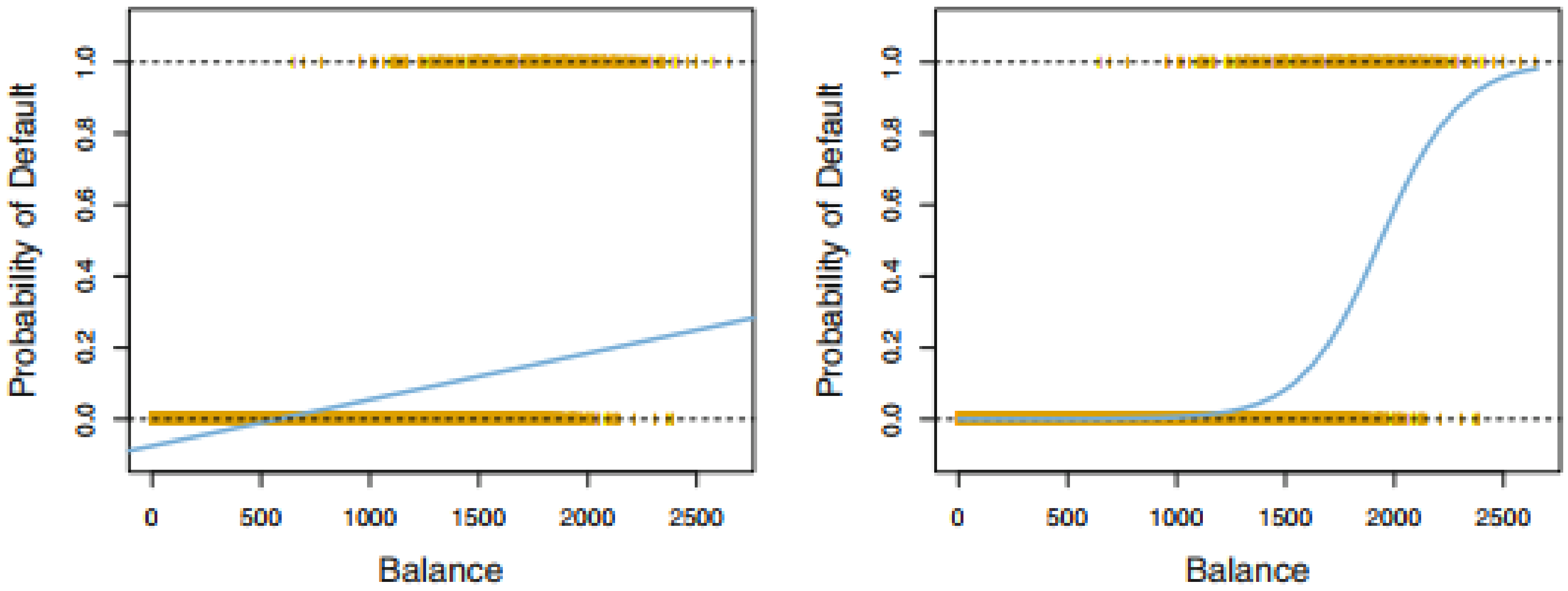
Parametric vs. nonparametric algorithms

DATA MODELING

	Pros	Cons
Parametric algorithms	**Simpler** Easier to understand and to interpret **Faster** Very fast to fit your data **Less data** Require "few" data to yield good perf.	**Limited complexity** Because of the specified form, parametric algorithms are more suited for "simple" problems where you can guess the structure in the data
Nonparametric algorithms	**Flexibility** Can fit a large number of functional forms, which doesn't need to be assumed **Performance** Performance will likely be higher than parametric algorithms as soon as data structures get complex	**Slower** Computations will be significantly longer **More data** Require large amount of data to learn **Overfitting** We'll see in a bit what this is, but it affects model performance

https://www.slideshare.net/CharlesVestur/building-a-performing-machine-learning-model-from-a-to-z

Logistic Regression



Logistic Regression

- Binary Logistic Regression

Binary Logistic Regression Formula :

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)}$$

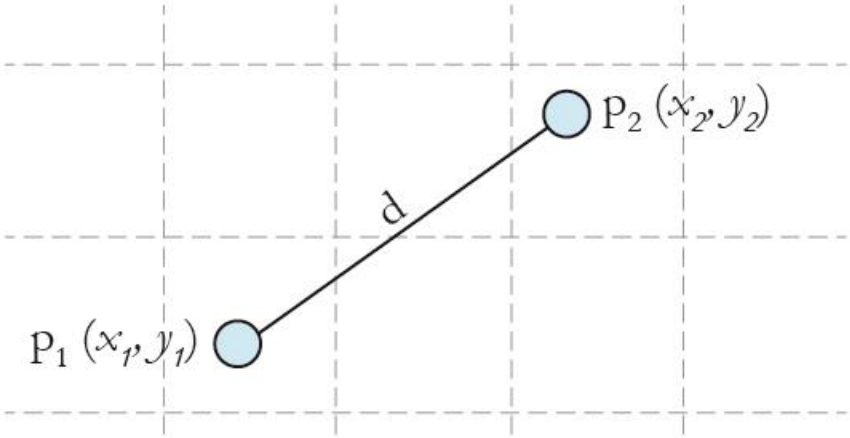
$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}$$

Logistic Regression

- Type of logistic regression : Binary, Ordinal, dan Multinomial
- Assumption
  - Independent each observation
  - No Multicolinearity
  - linearity of independent variables and log odds

K Nearest Neighbours

- KNN can be used for classification and regression
- Algorithm :
  1. Calculate distance each observation
  2. Sort the calculated distances in ascending order based on distance values
  3. Get k smallest distances from the sorted array, and calculate the average (regression) or count the majority vote (classification)

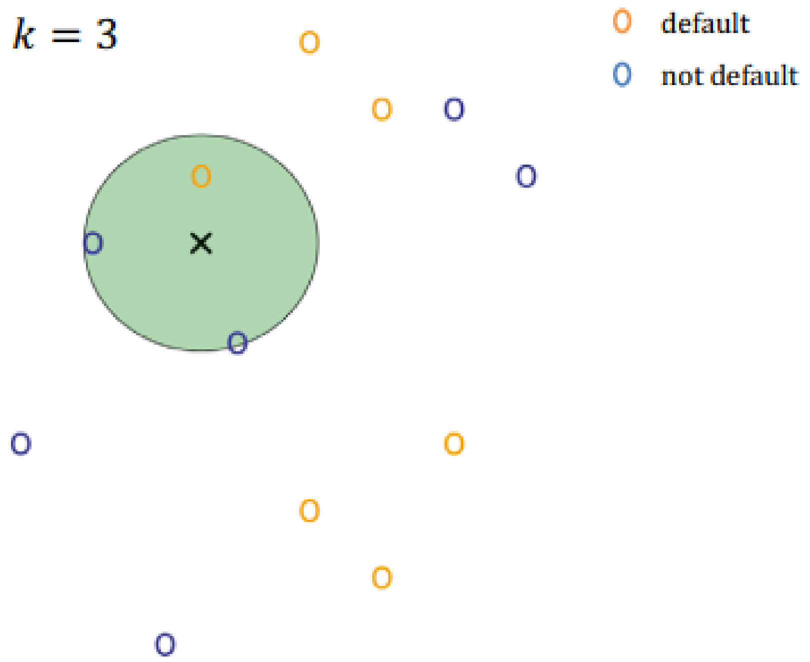


$$\text{Euclidean distance (d)} = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Source : *Gamesetmap*

K Nearest Neighbours

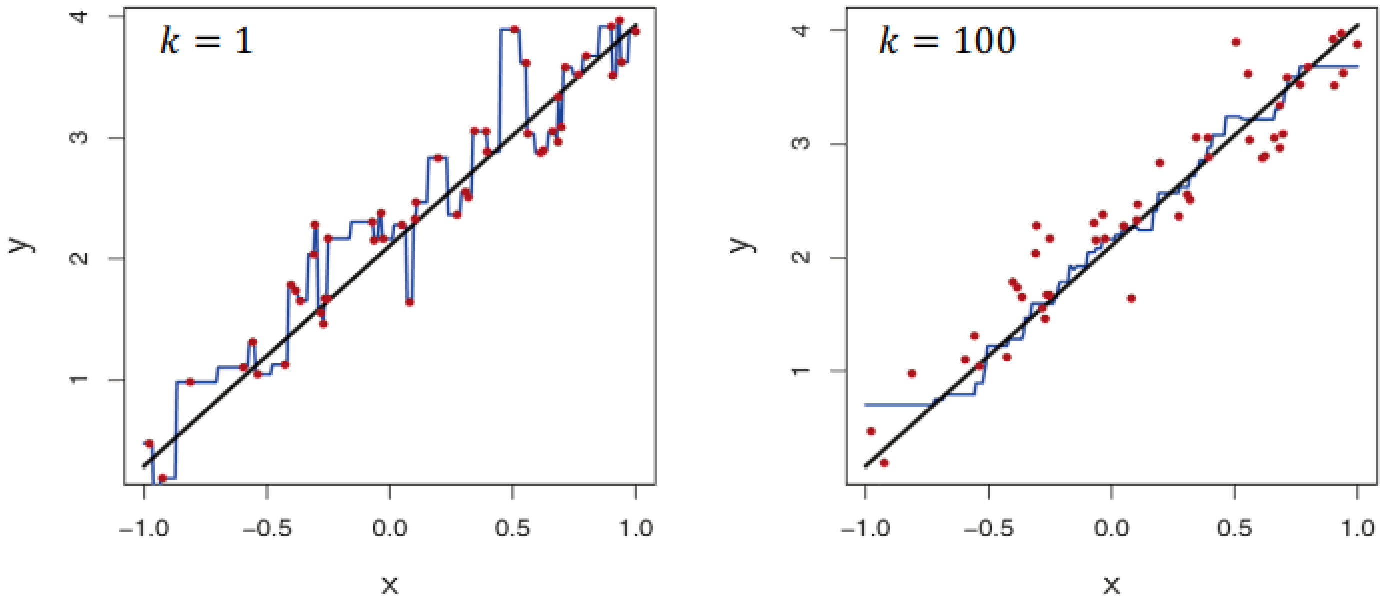
KNN Classfification



Picture from: Intoduction to Statistical Learning

K Nearest Neighbours

- Example of KNN with different K



Picture from: Intoduction to Statistical Learning

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

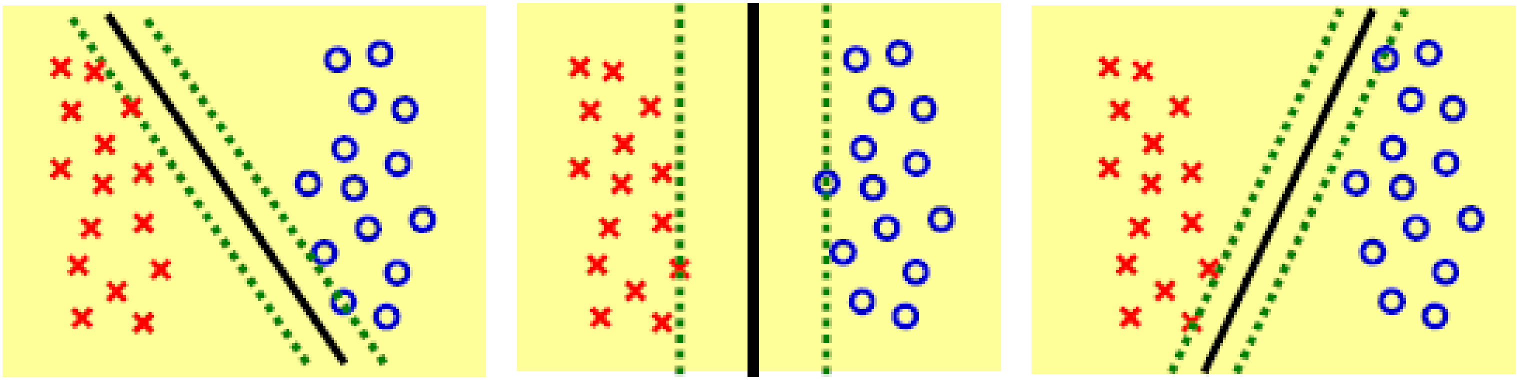
Source : *http://www.saedsayad.com*

KNN Classification

```
library(caret)
library(class)
data(iris)

inTrain <- createDataPartition(iris$Species, p = .8)[[1]]

train <- iris[inTrain,]

testX <- iris[-inTrain,-5]
testY <- iris[-inTrain,5]

fit <- knn (train[,-5], testX, train$Species, k = 3, prob = F)
confusionMatrix(testY,fit)
```

Support Vector Machine

- Main idea SVM is maximize hyperplane

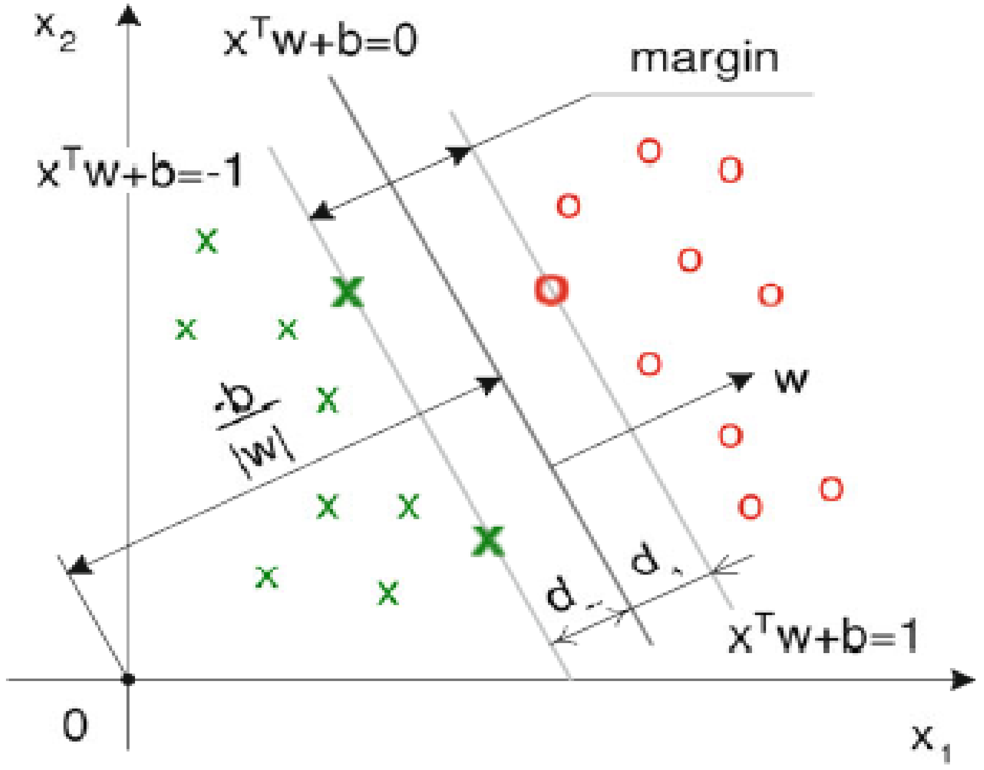


(a) Small margin (b) Large margin (c) Small margin

Source : Introduction to Statistical Machine Learning (Mashashi Sugiyama)

Support Vector Machine

DATA SCIENCE INDONESIA



Hyperplane concept (Haerdle, *et.al.*, 2011)

Dapat direpresentasikan dalam pertidaksamaan

$$y_i(x_i^T w + b) - 1 \geq 0, \quad i = 1, 2, \ldots, n$$

Secara matematis, formulasi problem optimasi SVM untuk klasifikasi linier dalam primal space adalah

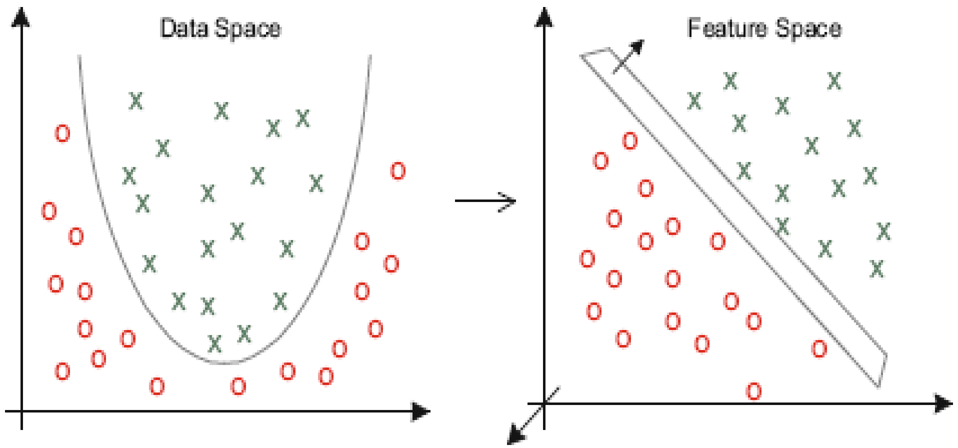
$$\min \frac{1}{2}\|w\|^2$$

Dengan demikian permasalahan optimasi dengan konstrain dapat dirumuskan menjadi

$$L_{pri}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \{ y_i(x_i^T w + b) - 1 \}$$

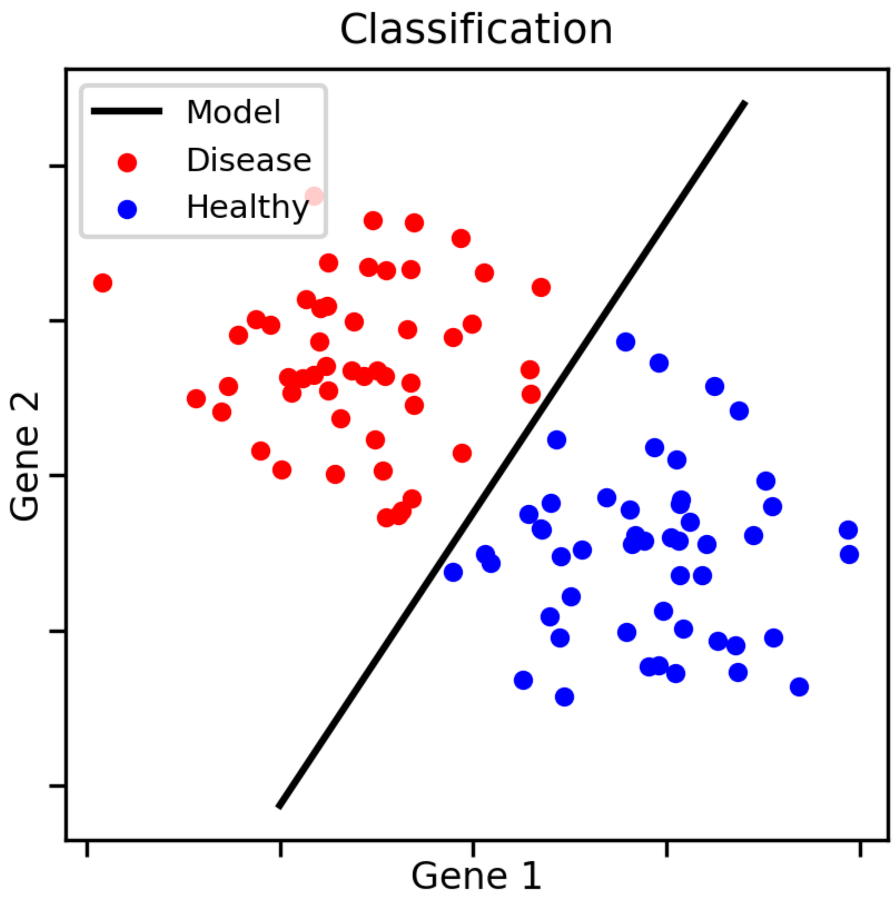
Support Vector Machine

Dalam mencari solusi masalah nonlinier digunakan “kernel trick” yaitu menambahkan fungsi kernel ke dalam persamaan SVM



1. Kernel Linier
2. Kernel Polynomial
3. Fungsi Kernel Radial Basis Function (RBF)
4. Kernel Eksponensial

Evaluation Metrics (Classification Case)



Evaluation Metrics (Regression Case)

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_\text{samples}} \sum_{i=0}^{n_\text{samples}-1} 1(\hat{y}_i = y_i)$$

$$\text{recall} = \frac{tp}{tp + fn},$$

$$\text{precision} = \frac{tp}{tp + fp},$$

$$F_\beta = (1 + \beta^2) \frac{\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}.$$

Thank You