DigitalSkola

LEARNING PROGRESS REVIEW

Week 12

Entropy Team

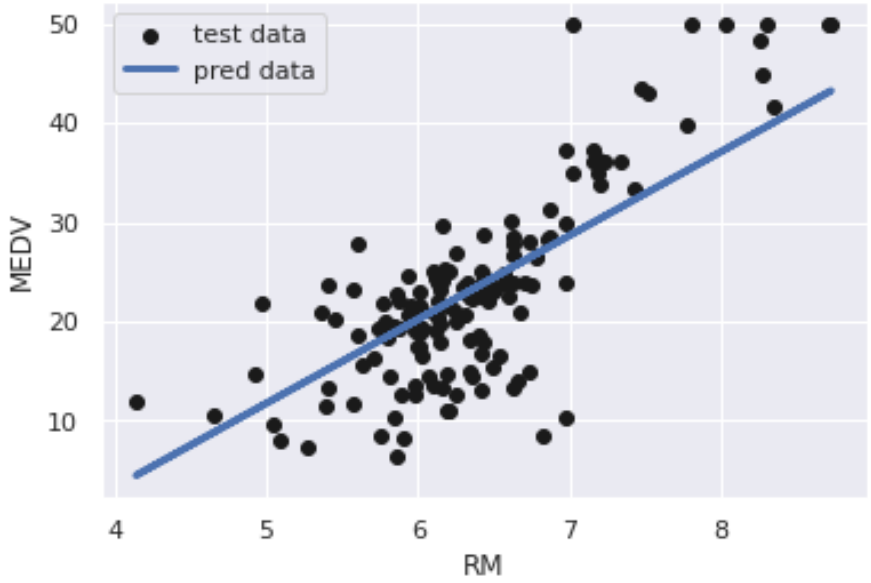
DAFTAR ISI

01

REGRESSION

Linear regression,
tree-based regression

Regresi



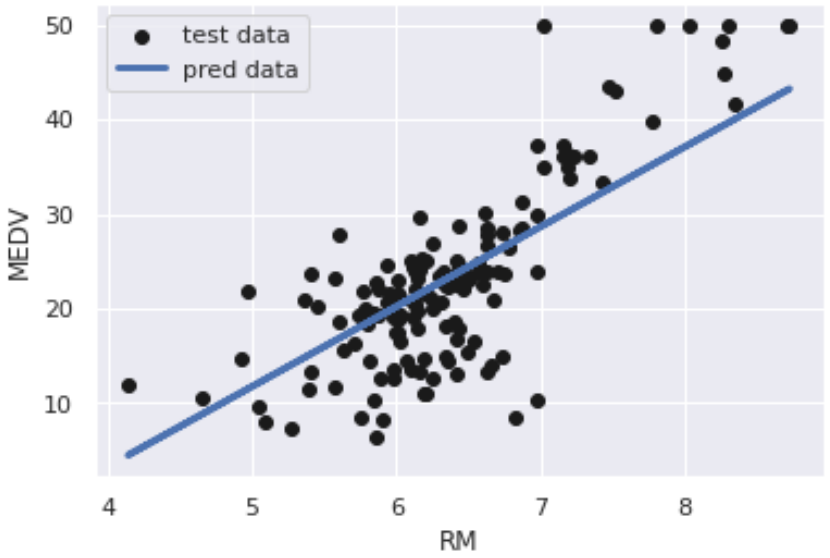
- **Regresi** merupakan salah satu jenis *supervised* ML di mana target berupa **nilai kontinu**

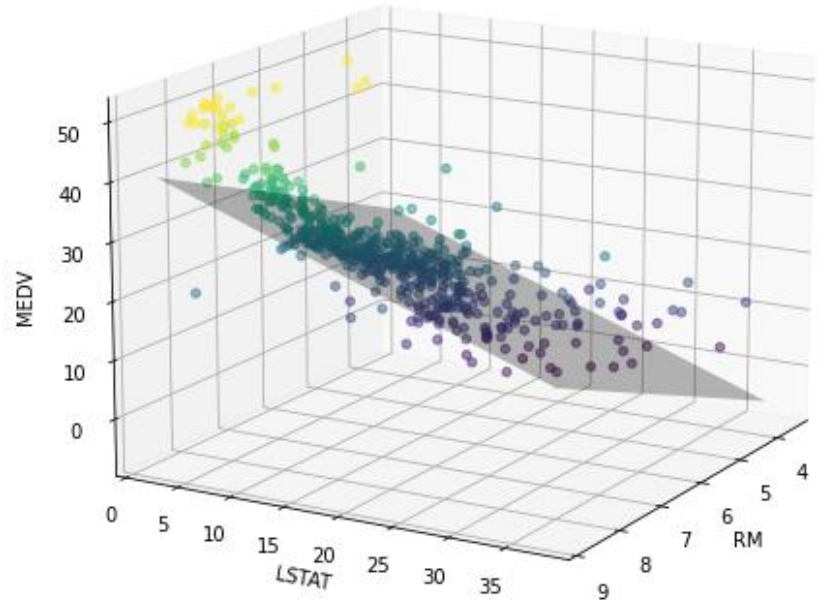
- **Regresi linier** merupakan salah satu jenis algoritma regresi yang digunakan untuk **membuat garis lurus** berdasarkan hubungan antara variabel dependen dengan variabel independen

Sumber: https://www.kaggle.com/adhang/boston-house-prices-linear-regression

DigitalSkola

Jenis Regresi Linier





Simple linear regression

Hanya terdapat 1 buah variabel independen

Multiple linear regression

Terdapat lebih dari 1 buah variabel independen

Sumber: https://www.kaggle.com/adhang/boston-house-prices-linear-regression

DigitalSkola

Regresi Linier

- **Regresi linier** akan membuat sebuah garis lurus untuk memenuhi persamaan regresi

- **Simple linear regression**

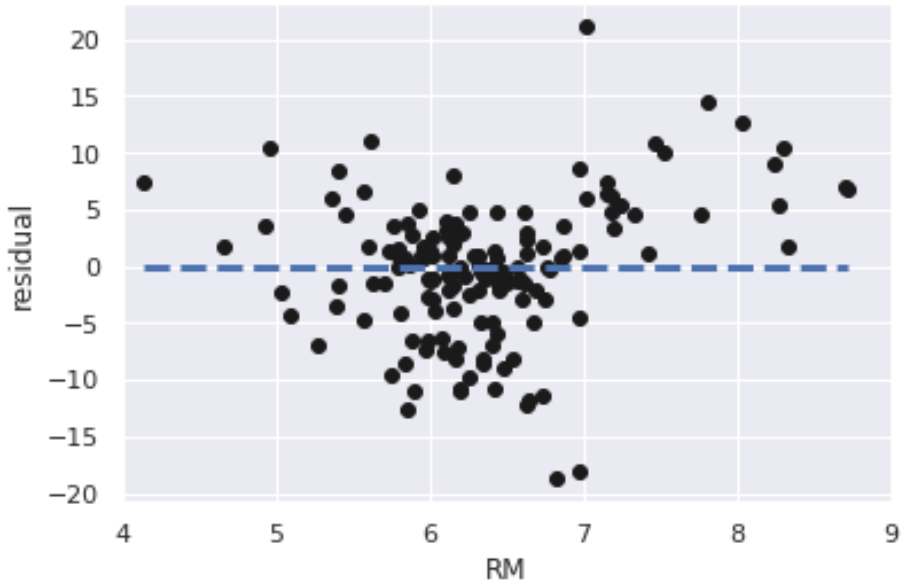
$$Y = a + bX + e$$

- **Multiple linear regression**

$$Y = a + b_1X_1 + b_2X_2 + \cdots + b_nX_n + e$$

- Di mana
  - Y : variabel dependen
  - X : variabel independen
  - a : intercept
  - b : slope
  - e : residual

Regresi Linier



- Algoritma regresi digunakan untuk mencari nilai ***intercept*** dan ***slope*** yang paling **optimal**
- Garis yang optimal yaitu memiliki nilai ***sum of squared residuals*** yang **rendah**

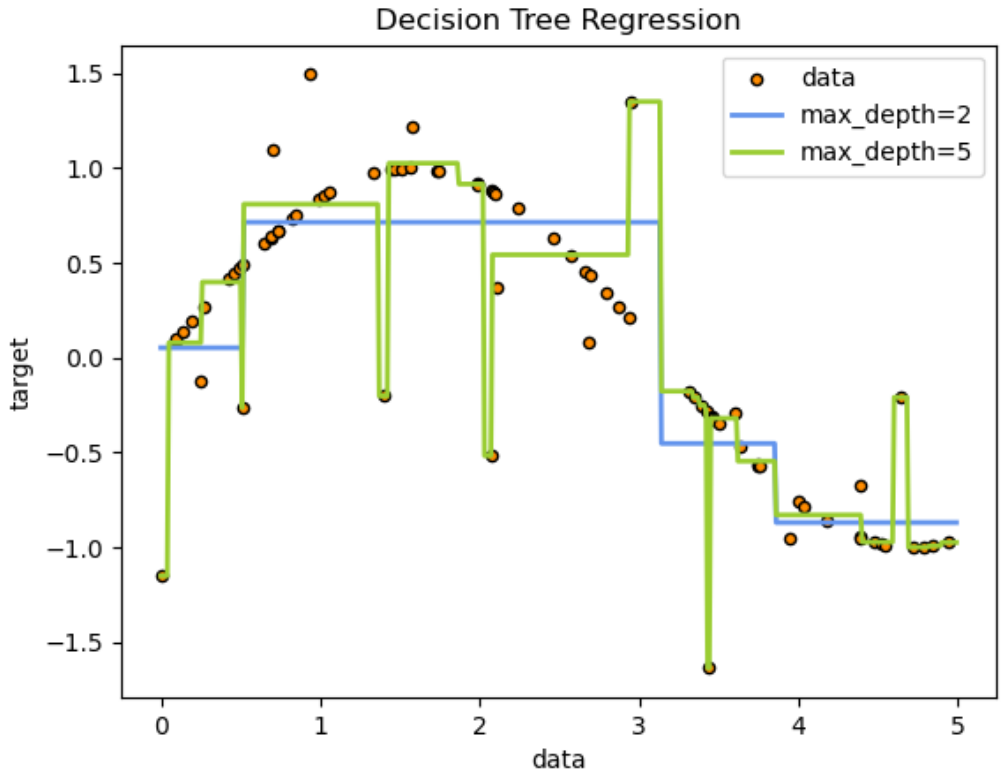
Sumber: https://www.kaggle.com/adhang/boston-house-prices-linear-regression

DigitalSkola

Regularization

- Pada *ordinary least square* (OLS), setiap *feature* (X) akan memiliki koefisien (*slope*), termasuk yang memiliki **kemampuan prediksi rendah**
- Hal tersebut dapat menyebabkan ***overfitting***
- Solusinya yaitu dengan ***regularization***, misal:
  - **Lasso** : menambahkan L1 norm
  - **Ridge** : menambahkan L2 norm
  - **Elastic net** : menambahkan L1 norm dan L2 norm

Regresi berbasis Tree



- ***Decision tree*** dan ***random forest*** juga dapat digunakan untuk menyelesaikan kasus regresi
- Pada ***random forest*,** hasil prediksi akhir merupakan **rata-rata prediksi** dari semua *tree*

Sumber: https://scikit-learn.org/stable/modules/tree.html

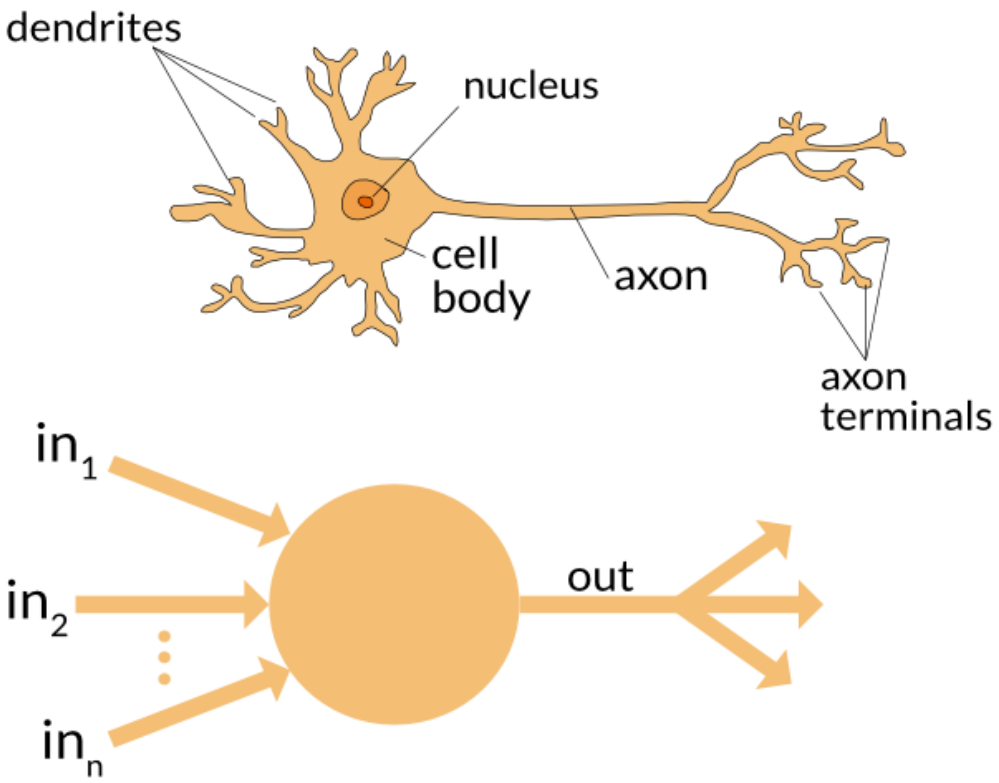
DigitalSkola

02

NEURAL NETWORK

Multi layer perceptron

Perceptron

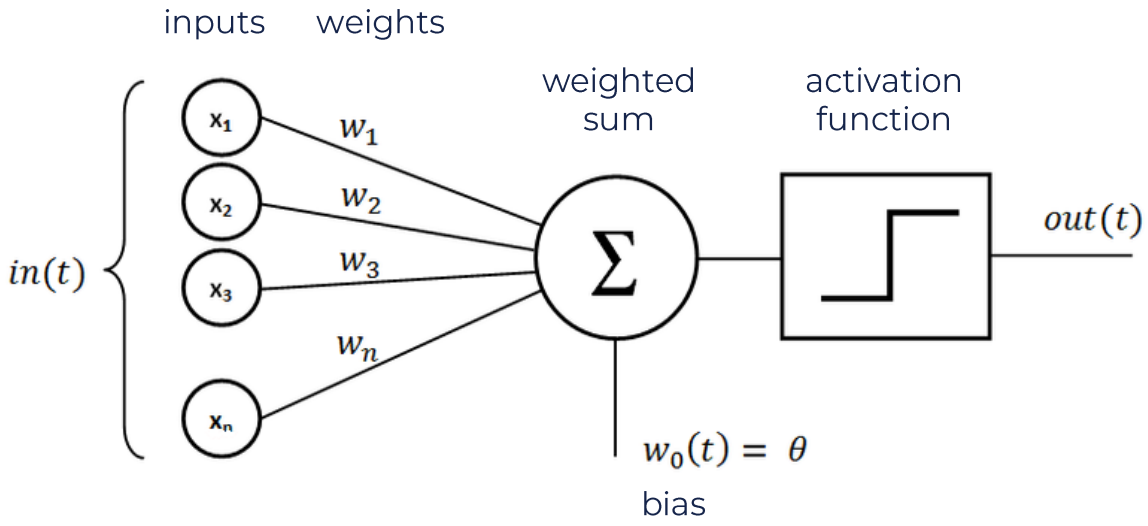


- *Neural network* (NN) merupakan algoritma *machine learning* yang **terinspirasi oleh cara kerja otak manusia**
- *Perceptron* merupakan bentuk paling dasar dari struktur *neural network*
- Secara umum, *single layer perceptron* terdiri dari ***input***, ***node***, dan ***output***

Sumber: https://www.codeproject.com/Articles/1205732/Build-Simple-AI-NET-Library-Part-Perceptron

DigitalSkola

Struktur Perceptron

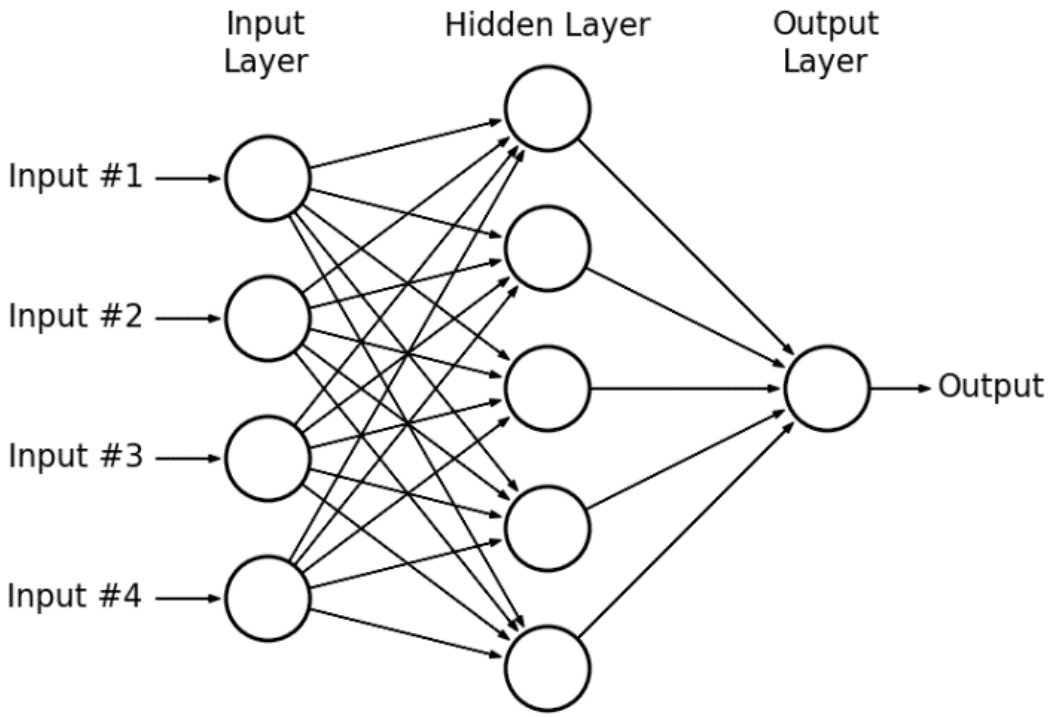


Pada *node*, terdapat 2 perhitungan:

- **Perhitungan *input* dari *node*** dengan cara menjumlahkan semua *input* yang telah diberi bobot (*weighted sum*)
- **Perhitungan *output* dari *node*** dengan cara memasukkan *weighted sum* ke fungsi aktivasi (*activation function*)

Sumber: https://commons.wikimedia.org/wiki/File:Perceptron_moj.png

Multi Layer Perceptron



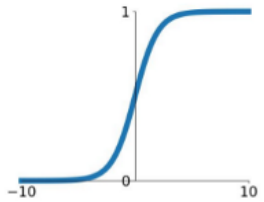
- ***Single layer perceptron*** hanya dapat digunakan untuk membuat **1 buah *decision boundary*** yang bersifat linear
- **Keterbatasan** itulah yang menjadi latar belakang terciptanya *multi layer perceptron* (MLP)
- Secara umum, MLP terdiri dari ***input*** *layer*, ***hidden*** *layer*, dan ***output*** *layer*

Sumber: https://www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065

Activation Function

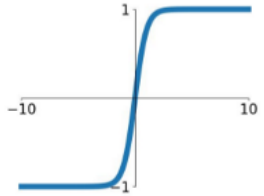
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



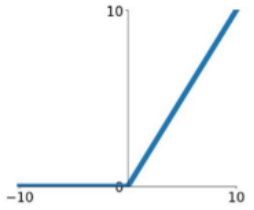
tanh

$\tanh(x)$



ReLU

$\max(0, x)$

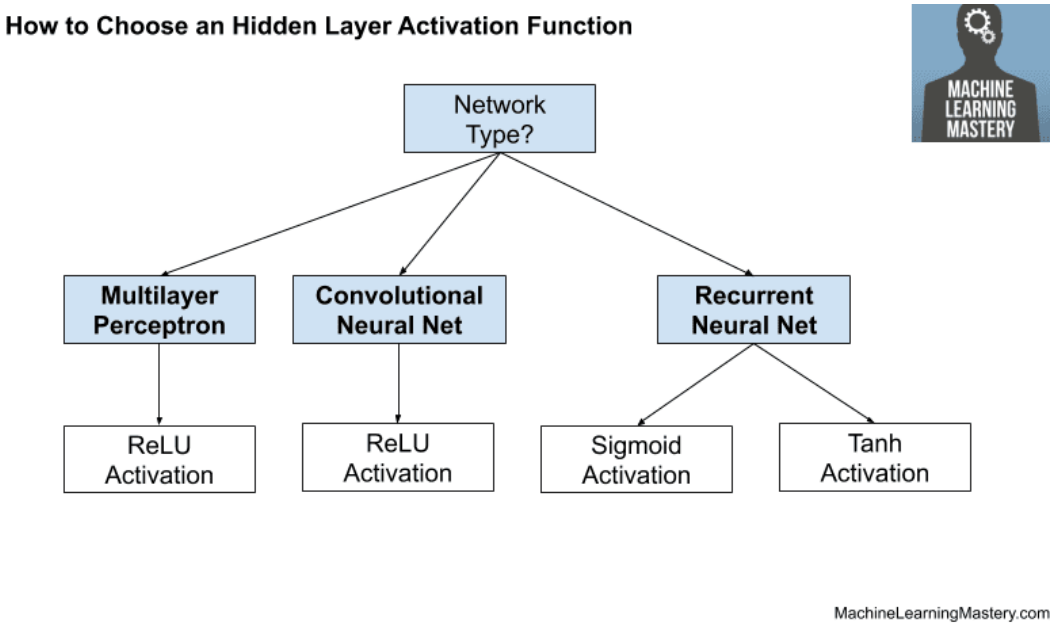


- **Setiap *node*** pada *hidden layer* dan *output layer* **memiliki *activation function***
- *Activation function* digunakan untuk **mengubah *input* menjadi *output*** dari sebuah *node*

Sumber: https://kotakode.com/blogs/3468/10-Konsep-Deep-Learning-yang-Harus-Kamu-Ketahui-untuk-Interview

DigitalSkola

Activation Function

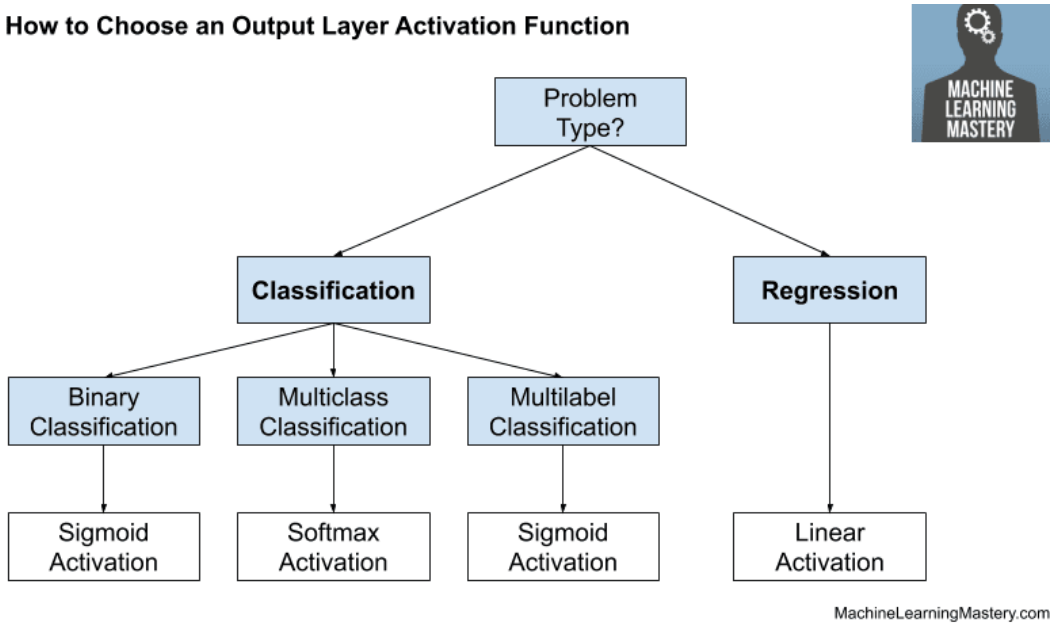


- Pada ***hidden layer*** MLP, umumnya menggunakan **ReLu** karena tidak rentan terhadap masalah *vanishing gradients*

Sumber: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

DigitalSkola

Activation Function

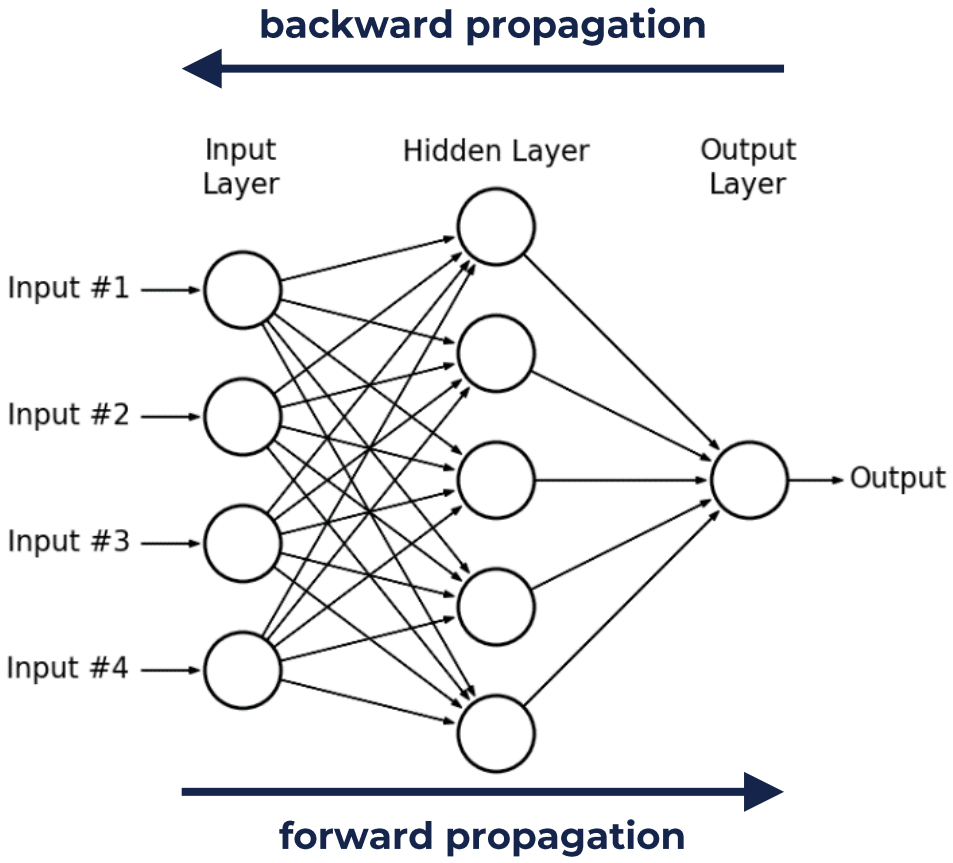


- Pada ***output layer*,** pemilihan *activation function* **tergantung** dari jenis **permasalahan** yang ingin diselesaikan

Sumber: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

DigitalSkola

Propagation



- ***Forward propagation***, yaitu proses untuk mengubah ***input*** menjadi ***output***
- ***Backward propagation***, yaitu proses untuk "menggerakkan" *error* dari *output layer* ke arah *input layer*. Pada tahap ini, NN akan **memperbarui *weight*** agar *error* menjadi minimal

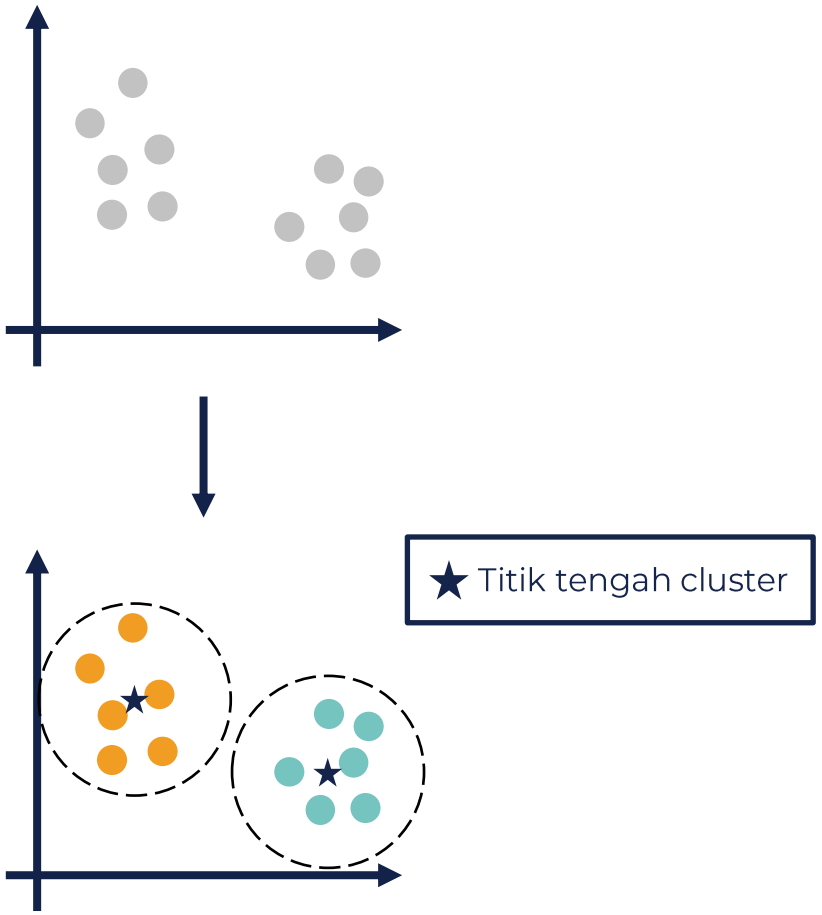
Sumber: https://www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065

03

UNSUPERVISED LEARNING

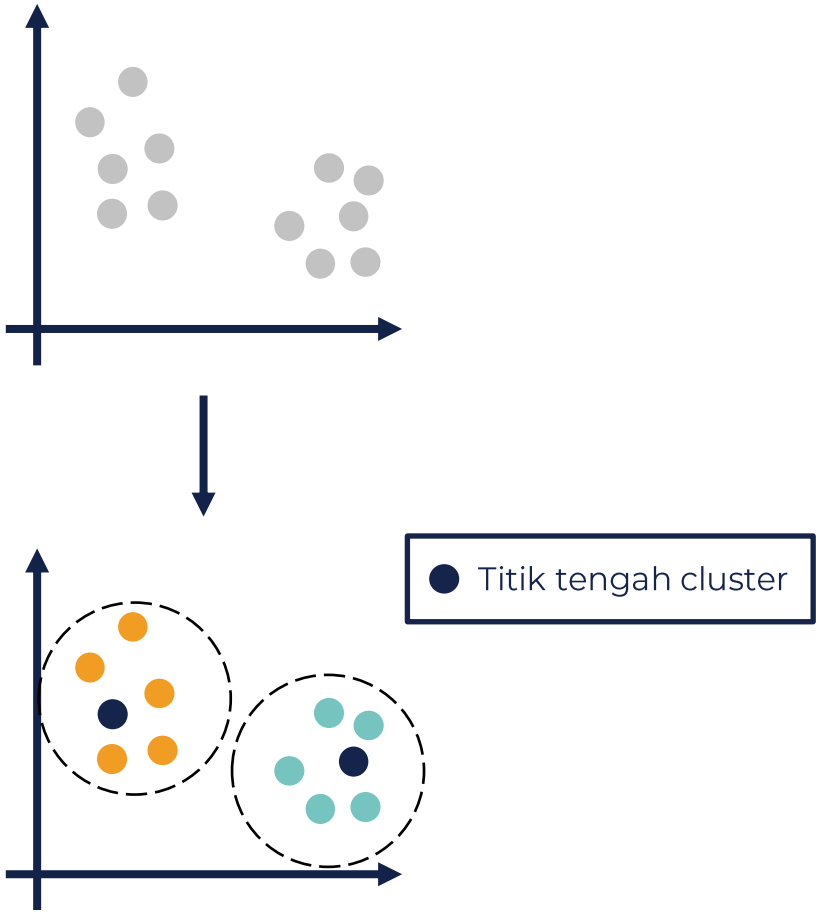
K-means, k-medoids, DBSCAN

K-Means



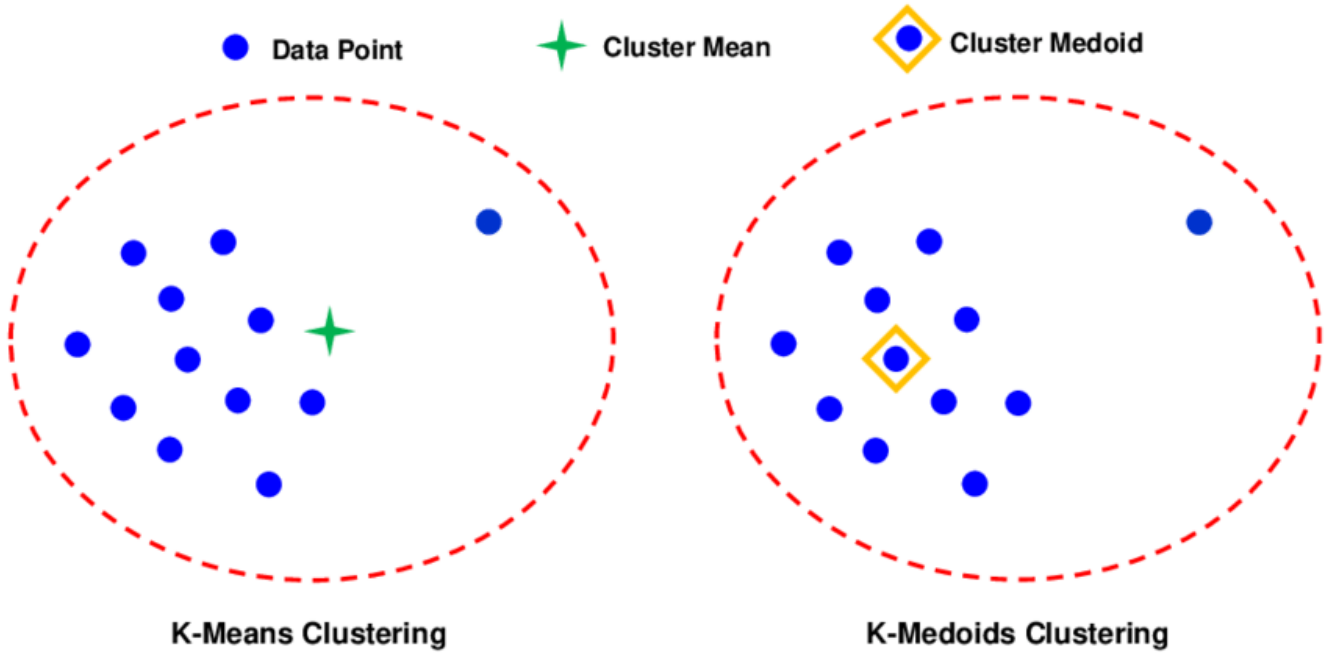
- K-means adalah salah satu algoritma ***clustering***
- K-means digunakan untuk membuat ***cluster* sebanyak K**
- Setiap *datapoint* akan masuk ke dalam *cluster* yang jaraknya paling dekat

K-Medoids



- K-medoids memiliki **konsep** yang **mirip** dengan k-means
- **Bedanya** yaitu cara menentukan titik tengah dari *cluster*
- K-medoids **tidak terlalu sensitif** terhadap ***outliers*** dibanding dengan k-means

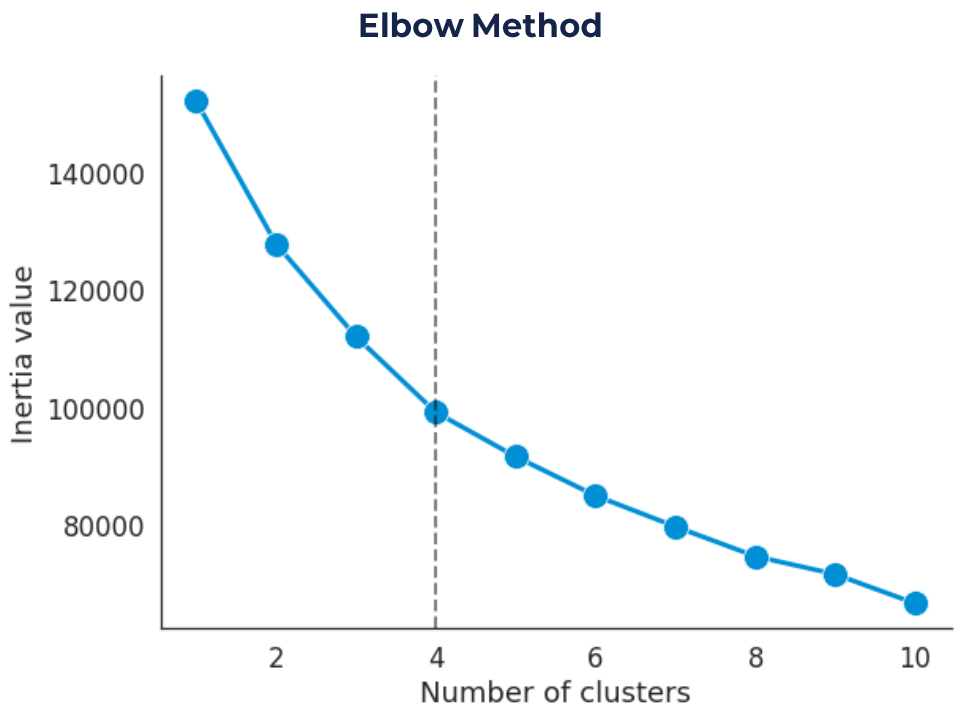
Titik Tengah Cluster



- Pada **k-means**, titik tengah *cluster* merupakan **rata-rata** dari semua *datapoint*
- Pada **k-medoids**, titik tengah *cluster* merupakan ***datapoint*** yang paling tengah

Sumber: https://www.researchgate.net/figure/The-graphical-representation-of-the-difference-between-the-k-means-and-k-medoids_fig1_342871651

Inertia

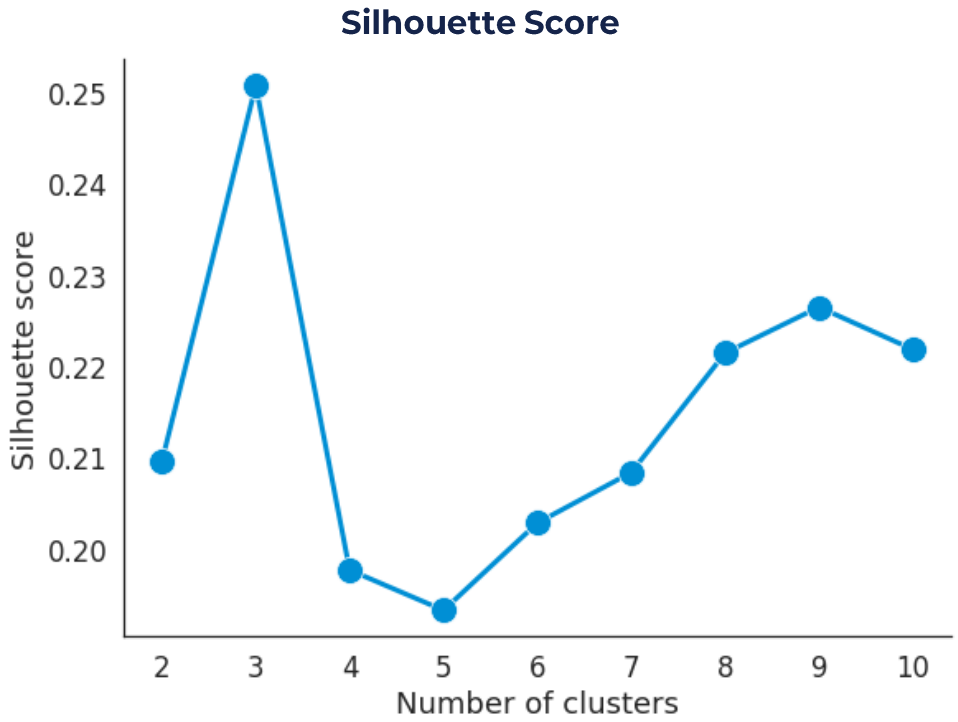


- *Inertia* merupakan salah satu metode untuk mengevaluasi algoritma *clustering*
- *Inertia* hanya fokus pada jarak ***intra-cluster***
- **Semakin rendah** nilai *inertia*, maka semakin kecil jarak antar-*datapoint*
- **Metode *elbow*** dapat digunakan untuk **menentukan jumlah *cluster*** yang optimal

Sumber: https://github.com/adhang/data-science-digital-skola

DigitalSkola

Silhouette Score

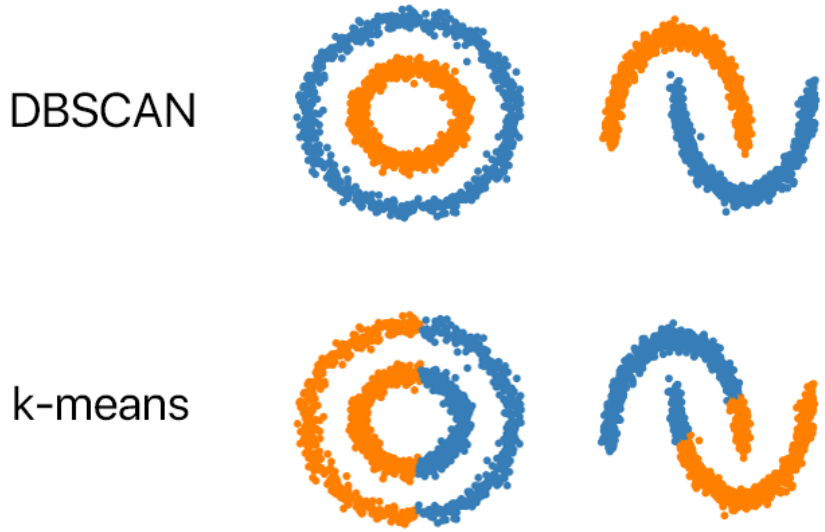


- *Silhouette score* merupakan salah satu metode untuk mengevaluasi algoritma *clustering*
- *Silhouette score* memperhitungkan jarak ***intra-cluster*** dan ***inter-cluster***
- *Silhouette score* yang bagus akan **mendekati nilai 1**

Sumber: https://github.com/adhang/data-science-digital-skola

DigitalSkola

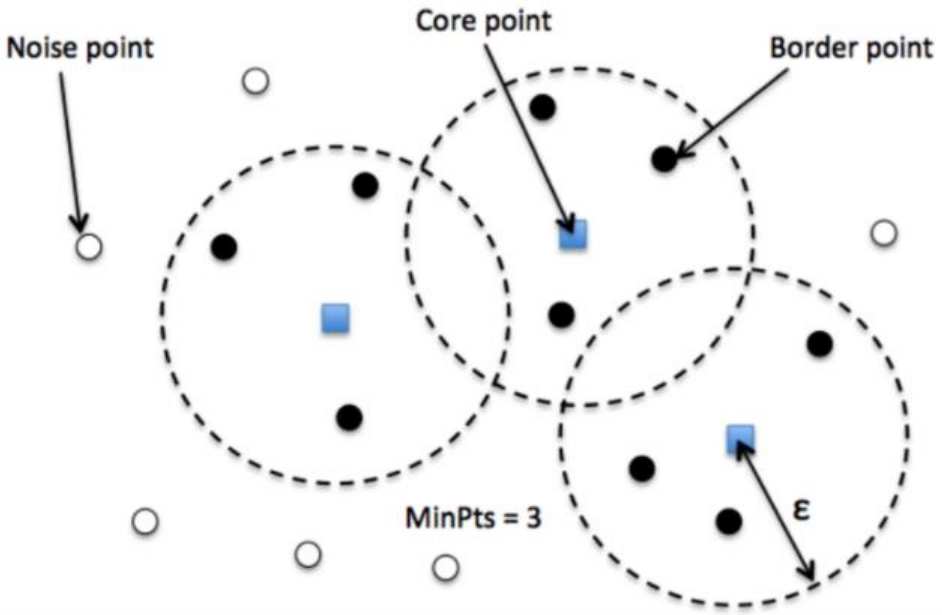
DBSCAN



- DBSCAN = Density-Based Spatial Clustering of Applications with Noise
- Sesuai namanya, DBSCAN merupakan algoritma *clustering* **berbasis *density***

Sumber: https://github.com/NSHipster/DBSCAN

Parameter DBSCAN



- DBSCAN memiliki 2 parameter utama:
  - **Epsilon** (ε), yaitu jarak antara titik tengah (*core point*) ke daerah sekitar
  - **MinPts** (min-points), yaitu **jumlah *datapoint*** minimum untuk **membentuk** sebuah ***cluster***
- Karena itulah, DBSCAN dapat digunakan untuk **mendeteksi** atau mengeliminasi ***outliers***/ *noise*

Sumber: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

THANKS

Entropy Team