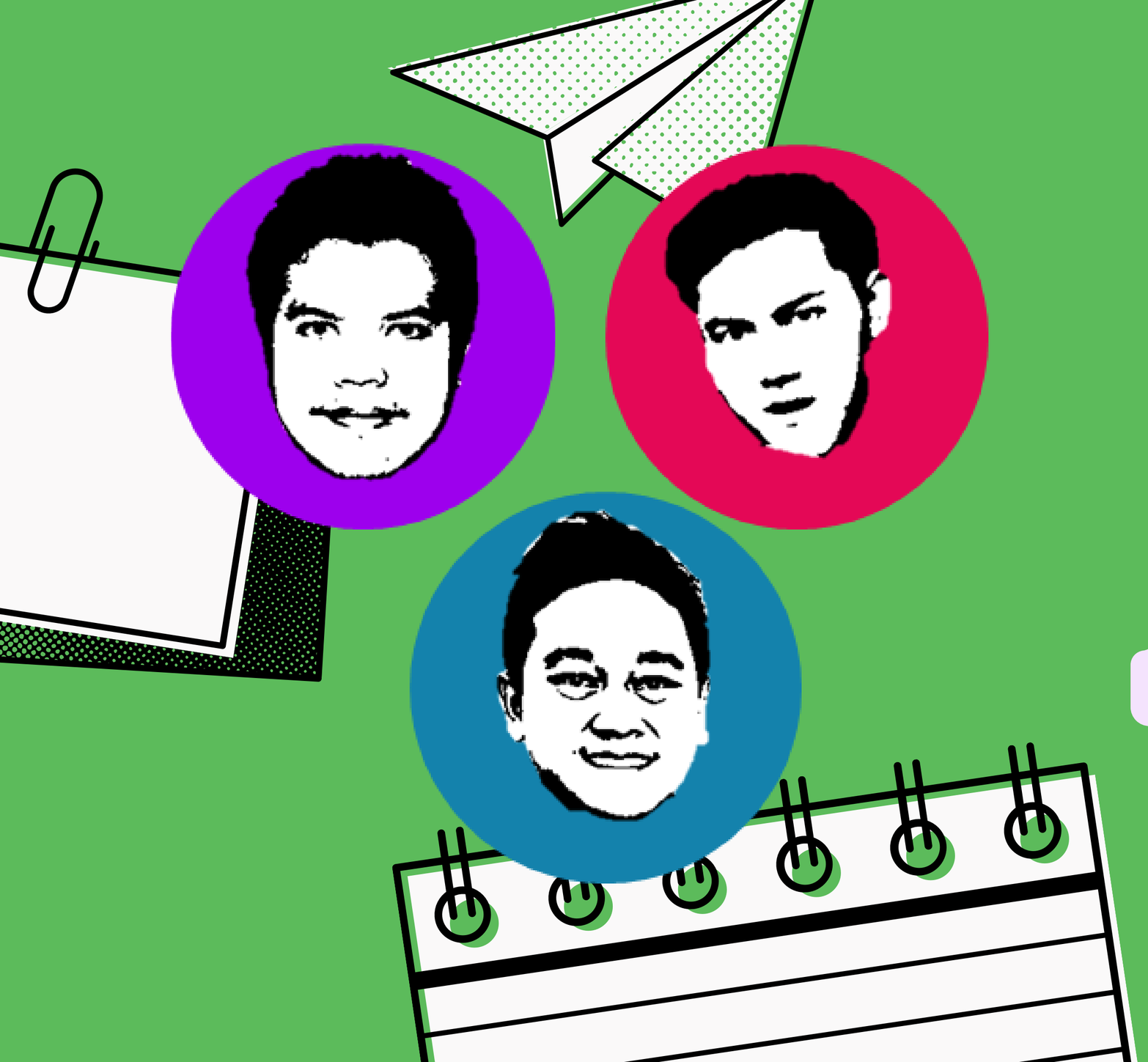
Data Science Bootcamp Batch 11

Learning Progress Review

Omicron

DigitalSkola

DigitalSkola

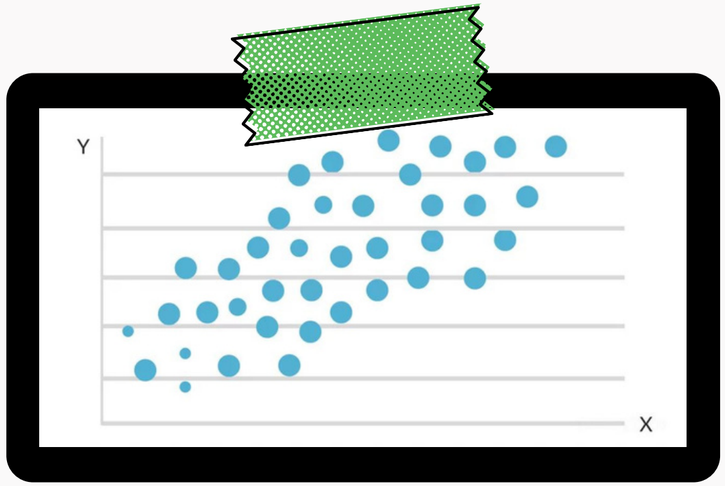


Omicron Memebers

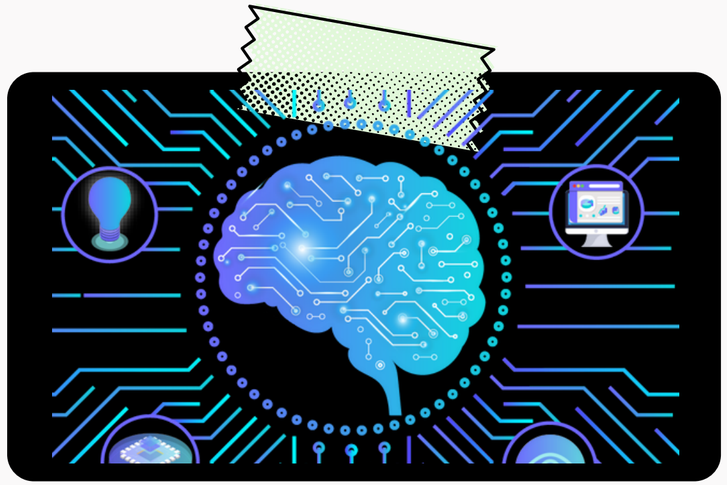
- Anugrah Yazid Ghani
- Edo M Hadad Gibran
- Muhammad Fikri Fadila

DigitalSkola

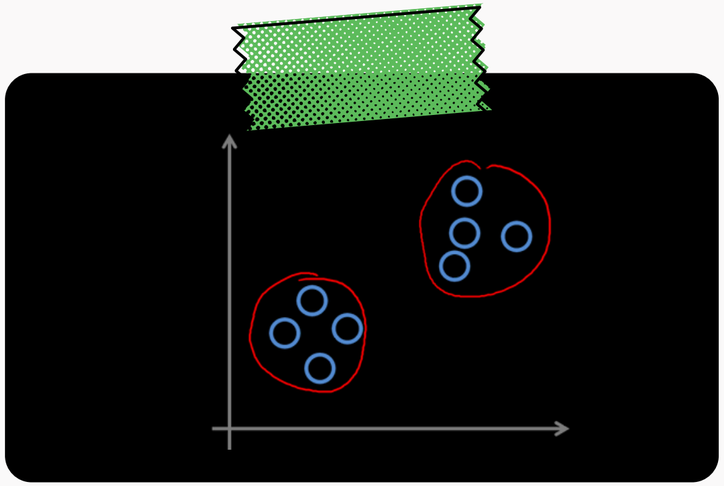
Content



Regression



Neural Network



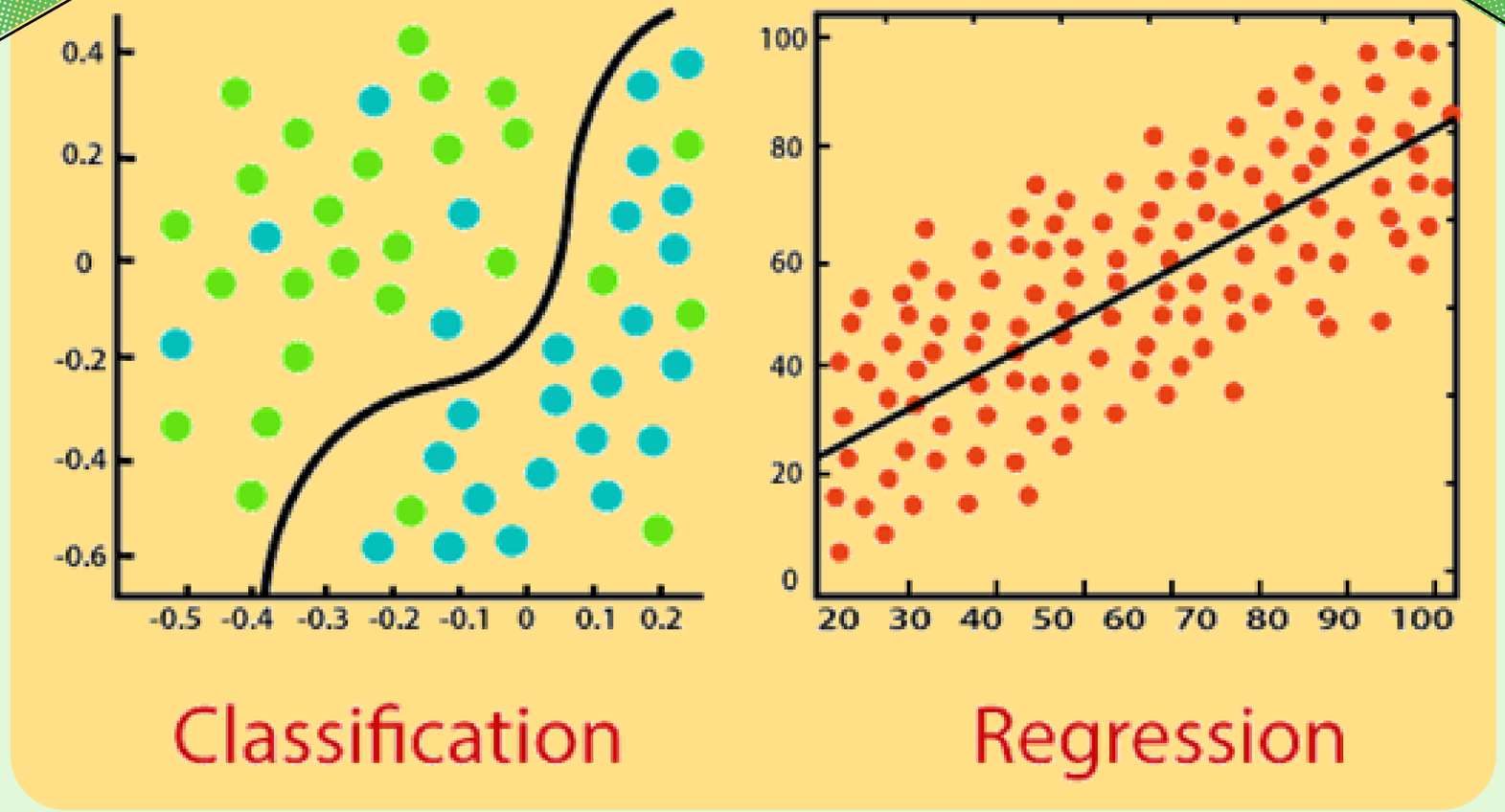
Unsupervised Learning

DigitalSkola

Regression

metode yang menentukan kekuatan dan karakter hubungan antara satu variabel dependen dan serangkaian variabel lainnya

DigitalSkola

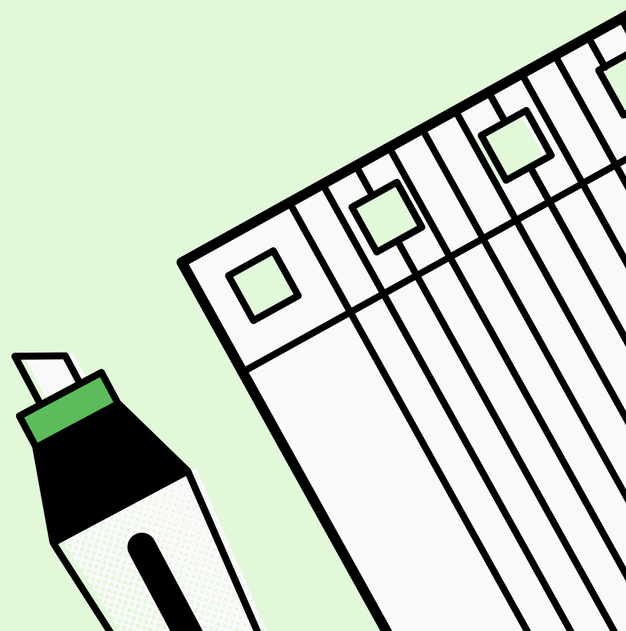


Regression

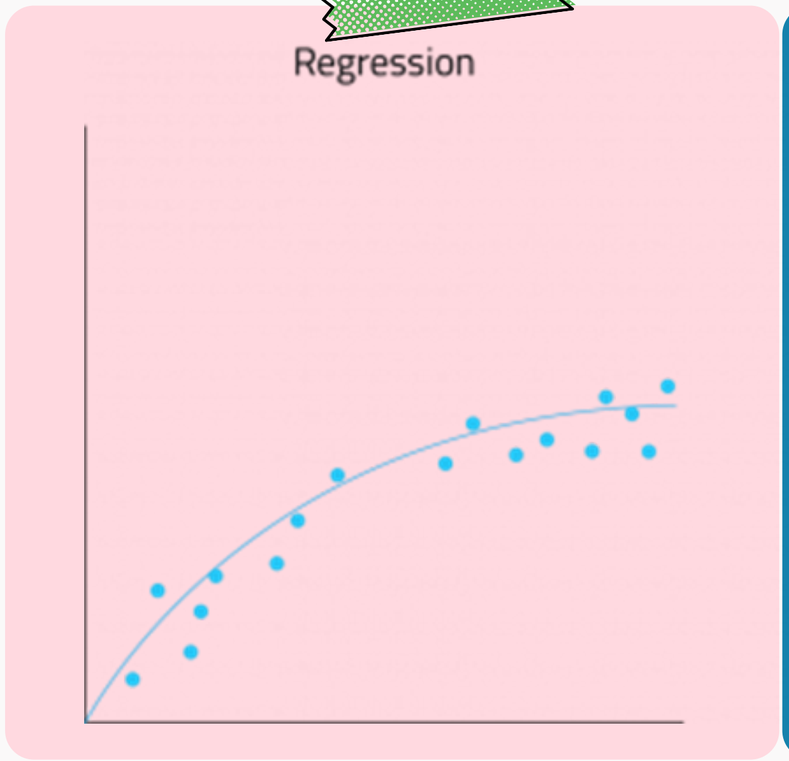
Regression dan Classification termasuk dalam kategori supervised learning

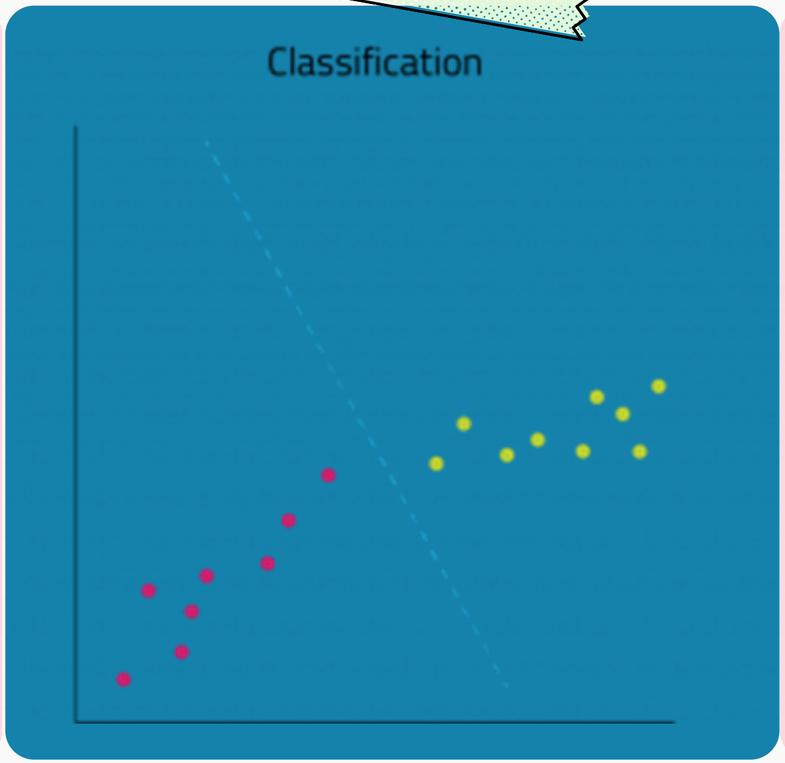
Classification algorithm : continuous value if it is in the form of a class label probability

Regression algorithm : discrete value which is in the form of an integer quantity



DigitalSkola





Regression

Classification

Clustering

Linear Regression

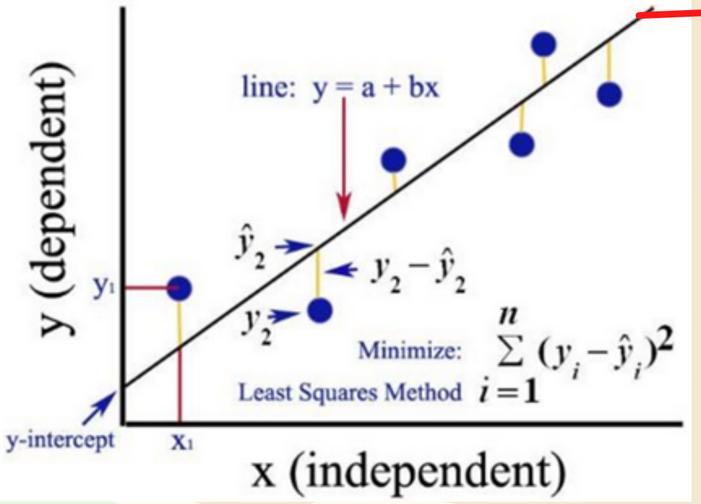
- Membangun hubungan diantara dua variables dengan garis lurus.
- Variabel independen merupakan variabel yang mempengaruhi atau menyebabkan perubahan.
- Variabel dependen adalah variabel yang dipengaruhi atau yang menjadi akibat karena adanya variabel independen.

- **Simple linear regression:** $Y = a + bX + u$
- **Multiple linear regression:** $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_tX_t + u$

Where:

- Y = the variable that you are trying to predict (dependent variable).
- X = the variable that you are using to predict Y (independent variable).
- a = the intercept.
- b = the slope.
- u = the regression residual.

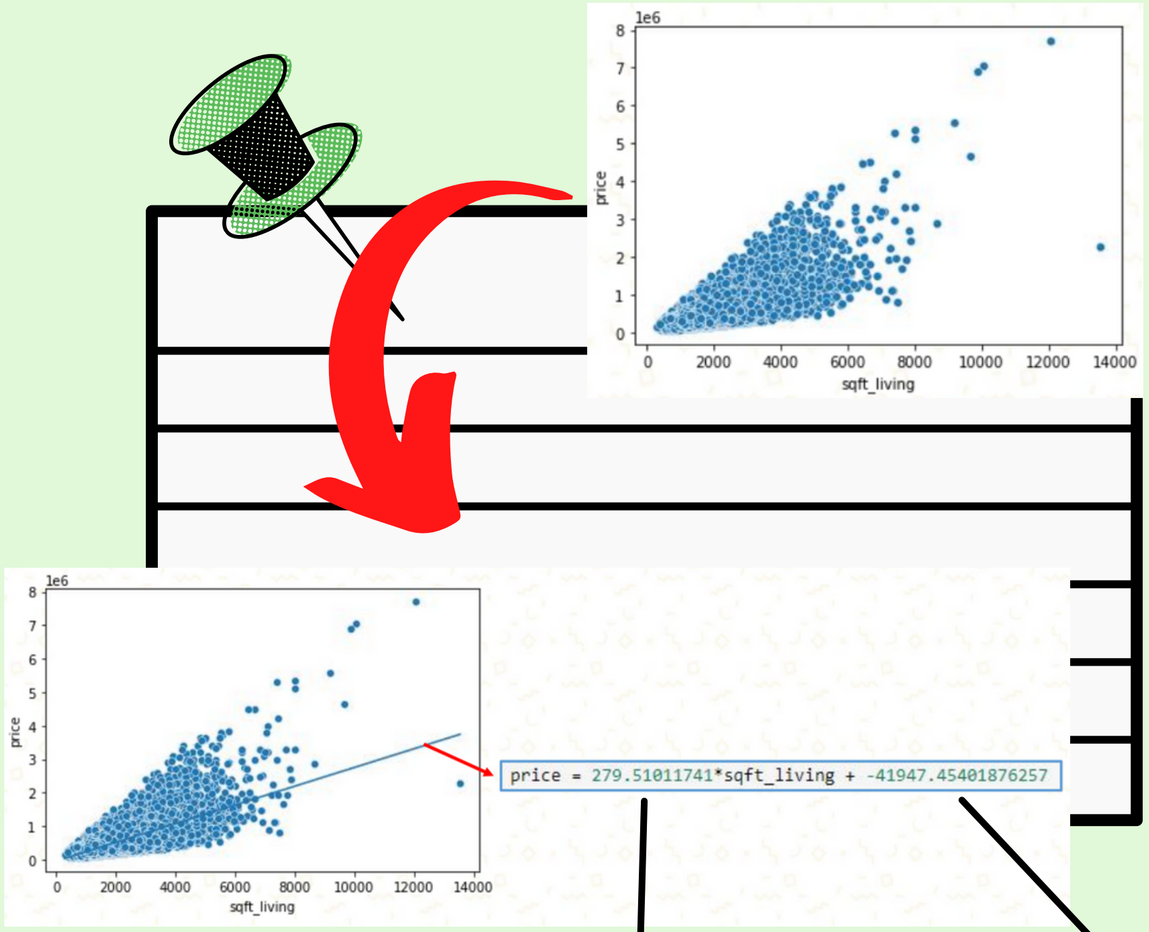
- Linear regression (regresi linier) mencoba menggambar garis yang paling dekat dengan data dengan menemukan slope dan intercept dan meminimalkan regression errors.
- Ordinary Least Squares (OLS) adalah metode estimasi yang paling umum untuk model linier



Garis optimal yang memberikan nilai ***sum of squared errors*** (SSE) terendah

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

DigitalSkola



Linear Regression

Example:

- y (dependent variable) = price (harga rumah)
- x (independent variable) = sqft_living (luas rumah)

Q : Rumah dengan luas 1000 square feet, berapa harganya kira kira?
A : USD 237562.663

DigitalSkola

Linear Regression

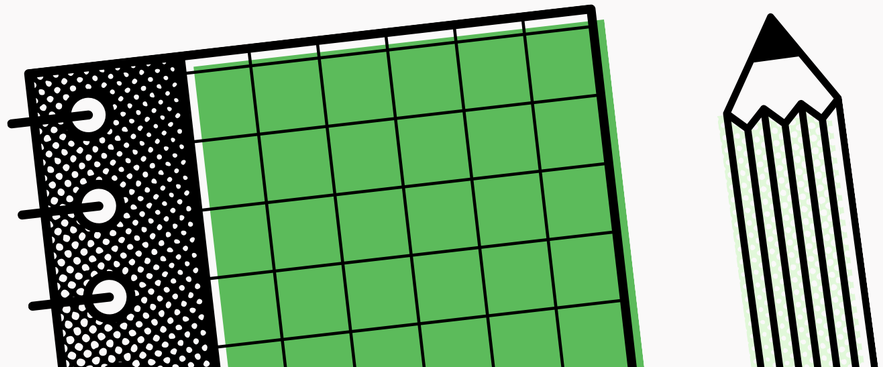
Kita bisa menggunakan library sklearn

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

df_X = df.drop(['id','date','price'],axis=1)
df_y = df['price']
X = df_X.astype(float).values
y = df_y.astype(float).values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
reg = LinearRegression()
reg.fit(X_train, y_train)
print('coefficient of determination of training set')
print(reg.score(X_train, y_train))
print('coefficient of determination of testing set')
print(reg.score(X_test, y_test))
print('coefficient')
print(reg.coef_)
print('intercept')
print(reg.intercept_)
print('prediction')
y_pred = reg.predict(X_test)
print(y_pred[:10])
print('real value')
print(y_test[:10])
```

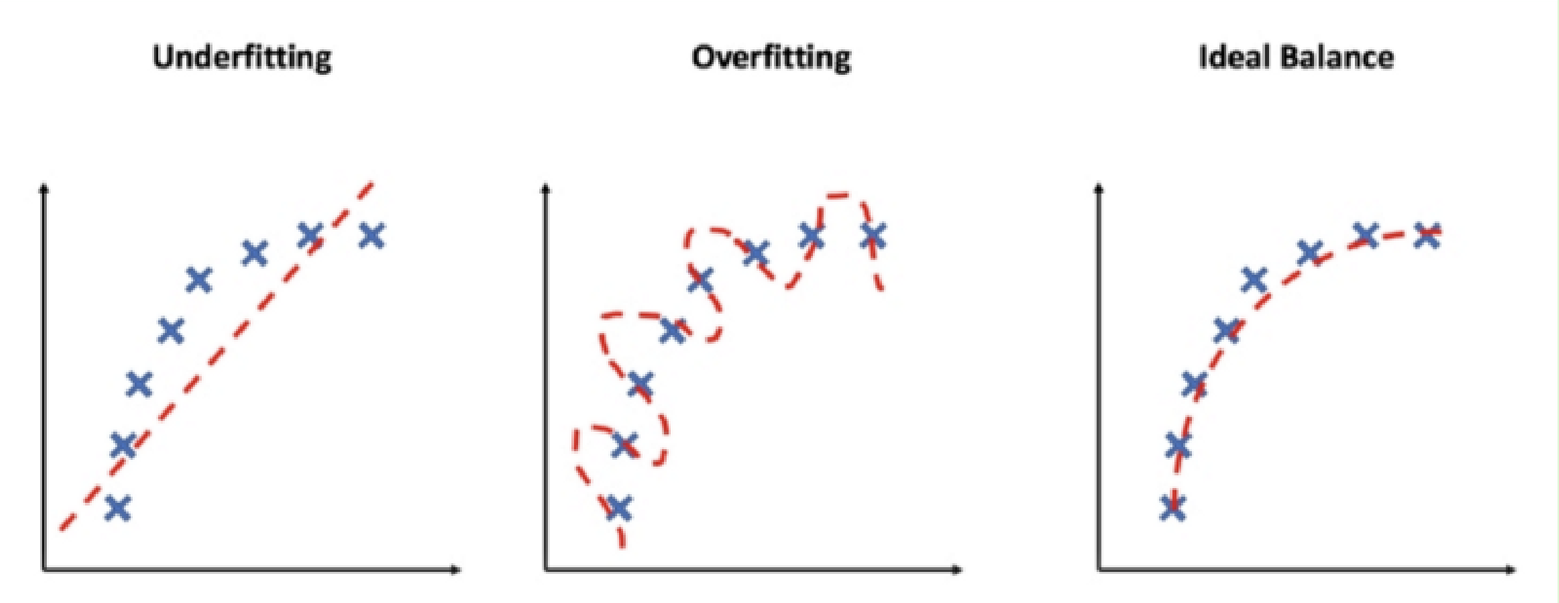
```
coefficient of determination of training set
0.6995155846436758
coefficient of determination of testing set
0.6994627057969862
coefficient
[-3.43081477e+04  4.03129700e+04  1.12001375e+02  9.91841247e-02
  5.27154218e+03  5.43877177e+05  5.50830616e+04  2.31460673e+04
  9.49081794e+04  7.22190669e+01  3.97823083e+01 -2.59441847e+03
  2.19209734e+01 -5.56358731e+02  5.95216324e+05 -1.96904658e+05
  1.62077488e+01 -3.30430480e-01]
intercept
6641646.708113588
prediction
[ 458597.0676416   748993.75994814 1243303.75799055 1665116.95095444
  737302.05741739  283239.58524974  831732.87582315  495383.02095338
  385779.81919026  474179.42285135]
real value
[ 365000.  865000. 1038000. 1490000.  711000.  211000.  790000.  680000.
  384500.  605000.]
```



DigitalSkola

Bias and Variance

- Linear regression mencari nilai coefficient yang meminimalkan nilai sum of squared errors (SSE).
- Tetapi mungkin ini bukan model terbaik, karena akan memberikan coefficient untuk semua features.
- Termasuk feature yang mempunyai “kemampuan prediksi yang rendah”.Ini akan menghasilkan model yang “high-variance, low bias”.
- Solusi = regularization (memodifikasi cost function untuk memberi batasan nilai coefficients).



Lasso and Ridge

L1 Regularization

- Lasso (least absolute shrinkage and selection operator) regression
- Lasso memberi tambahan “absolute value of magnitude” dari coefficient sebagai penalti untuk loss function
- Menambahkan sum of the coefficient values (the L-1 norm) dan mengalikan dengan constant lambda

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

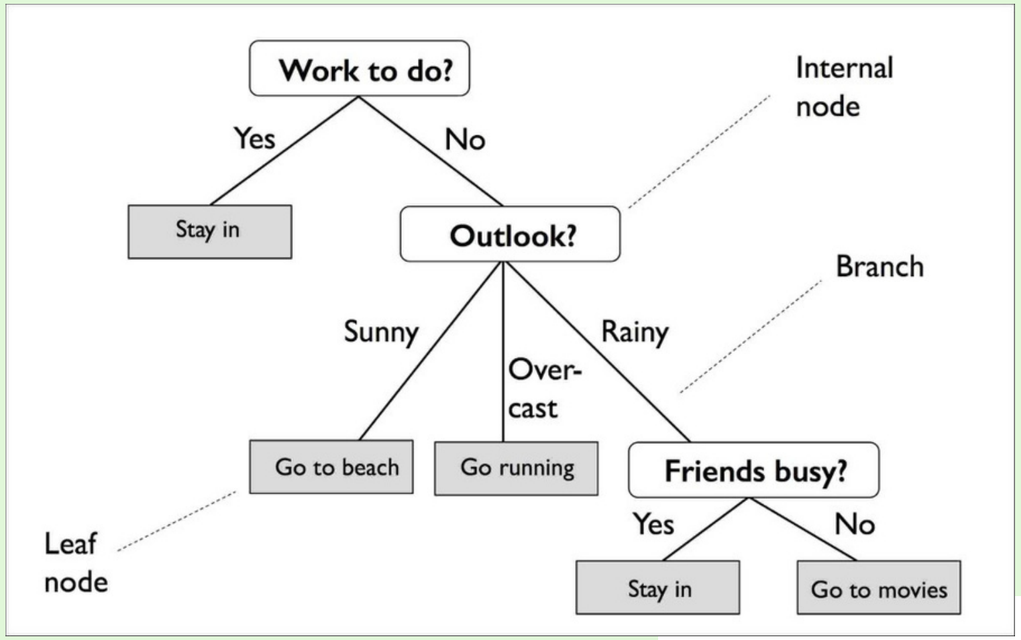
$$\text{Cost} = \underbrace{\sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

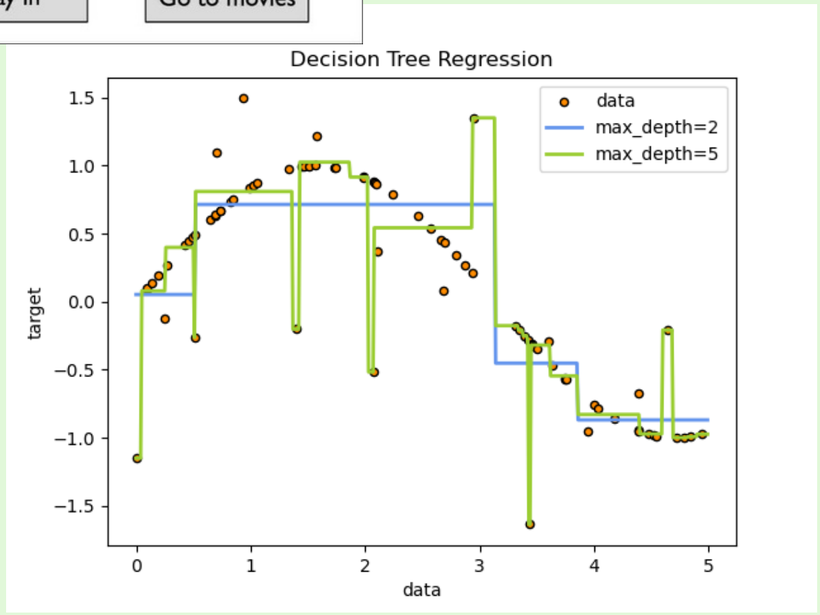
L2 Regularization

- Ridge regression
- Ridge regression menambahkan “squared magnitude” dari coefficient sebagai penalti untuk loss function
- Menambahkan sums the squares of coefficient values (the L-2 norm) dan mengalikan dengan constant lambda

Decision Tree Regression

- Decision trees bisa diaplikasikan pada kasus classification dan regression
- Keuntungan
  - Mudah dipahami dan diinterpretasikan.
- Kerugian
  - Bisa membuat “over-complex trees” yang tidak bisa generalise terhadap data baru atau disebut dengan overfitting.
  - Solusi : pruning

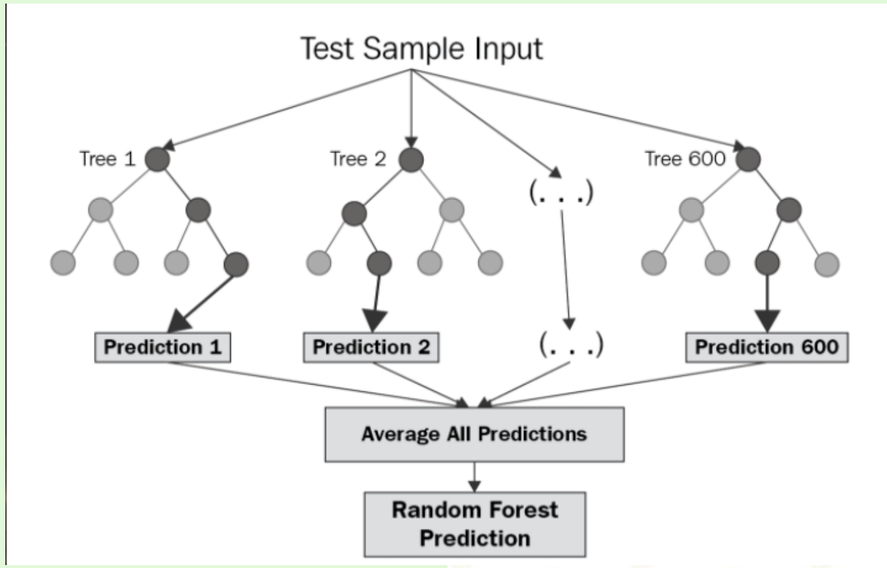


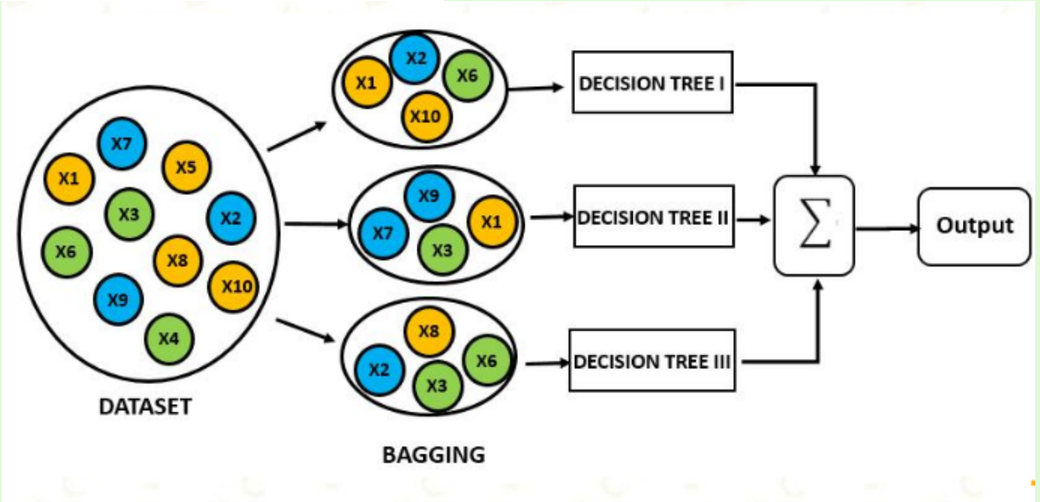


DigitalSkola

Random Forest Regression

- Random forest adalah algoritma dalam Supervised Learning yang menggunakan ensemble learning method untuk kasus classification dan regression.
- Hasil prediksi adalah label terbanyak (untuk kasus classification) atau rata rata hasil prediksi (untuk kasus regression) dari model tree yang banyak.





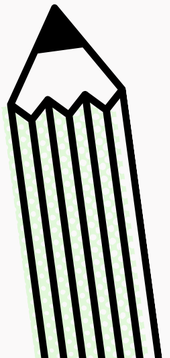
DigitalSkola

Evaluation metrics for Regression

- Pearson correlation coefficient (r) = mengukur kekuatan dan arah hubungan linier antara dua variabel (-1 to 1).
- Coefficient determination (r 2 or r square) = memberikan proporsi varians (fluktuasi) dari satu variabel yang diprediksi dari variabel lainnya (0 to 1).
- Root mean square error (RMSE) = merupakan besarnya tingkat kesalahan hasil prediksi. Semakin kecil (mendekati 0) semakin baik (prediction errors).

Mean squared error	$\text{MSE} = \frac{1}{n}\sum_{t=1}^{n} e_t^2$
Root mean squared error	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$
Mean absolute error	$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} \lvert e_t \rvert$
Mean absolute percentage error	$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n} \left\lvert \frac{e_t}{y_t} \right\rvert$

Performance Metric	Formula
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Pearson correlation coefficient (r)	$\frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}}_i)^2}}$
Coefficient determination (r^2)	$r^2 = [\text{Correlation Coefficient}]^2$

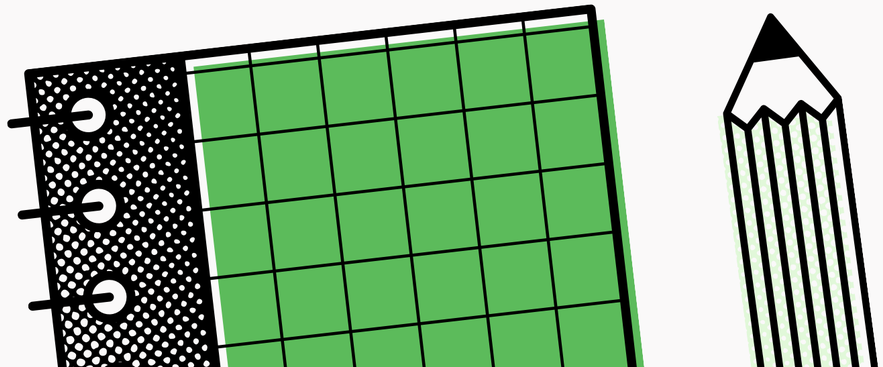


DigitalSkola

Performance comparison

Hasil perbandingan dari model regresi yang diaplikasikan pada house price dataset

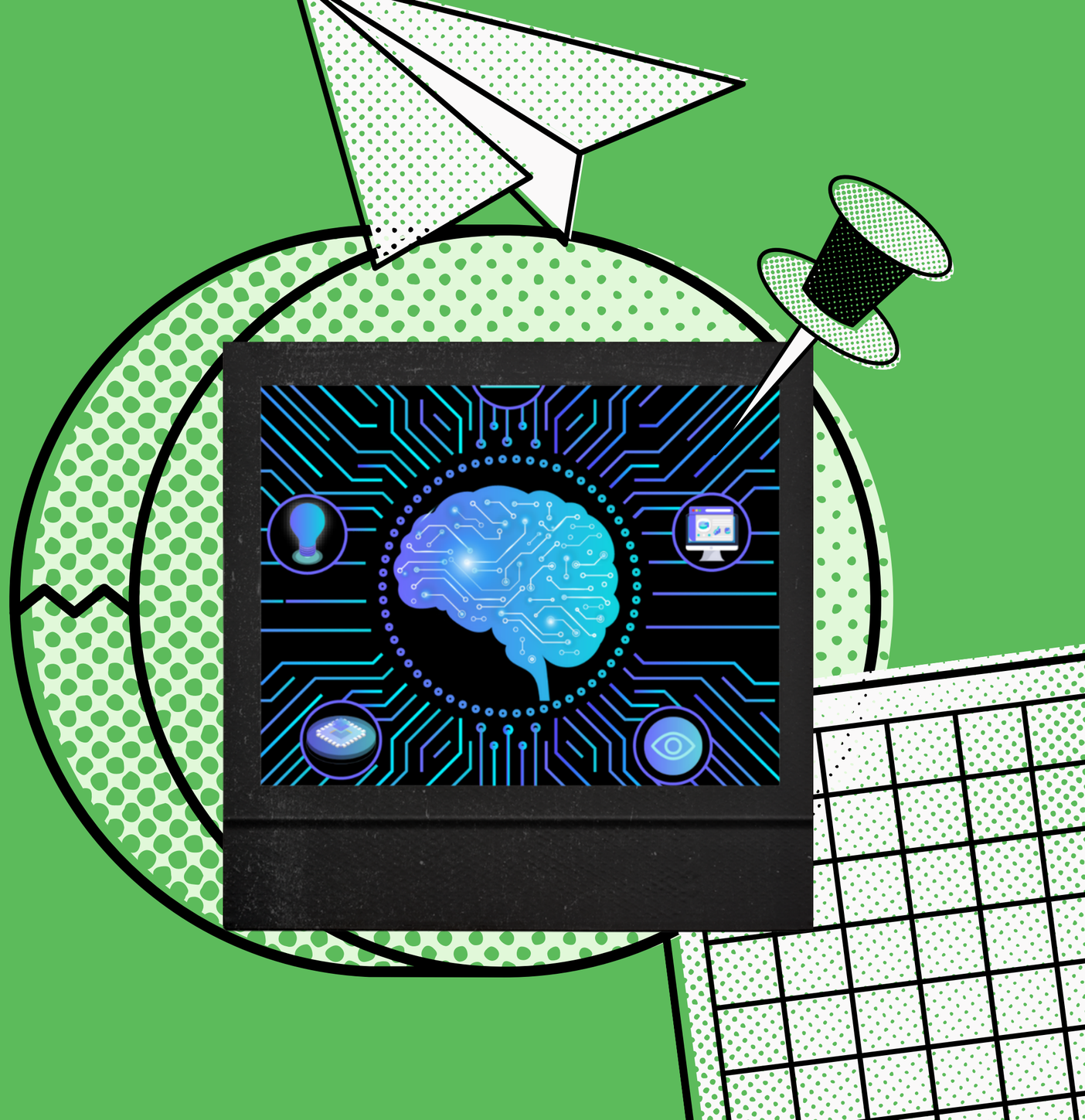
Model	RMSE	r2
Linear regression	208296	0.69
Lasso	208297	0.69
Ridge	208297	0.69
DT regression	192962	0.74
RF regression	144539	0.85



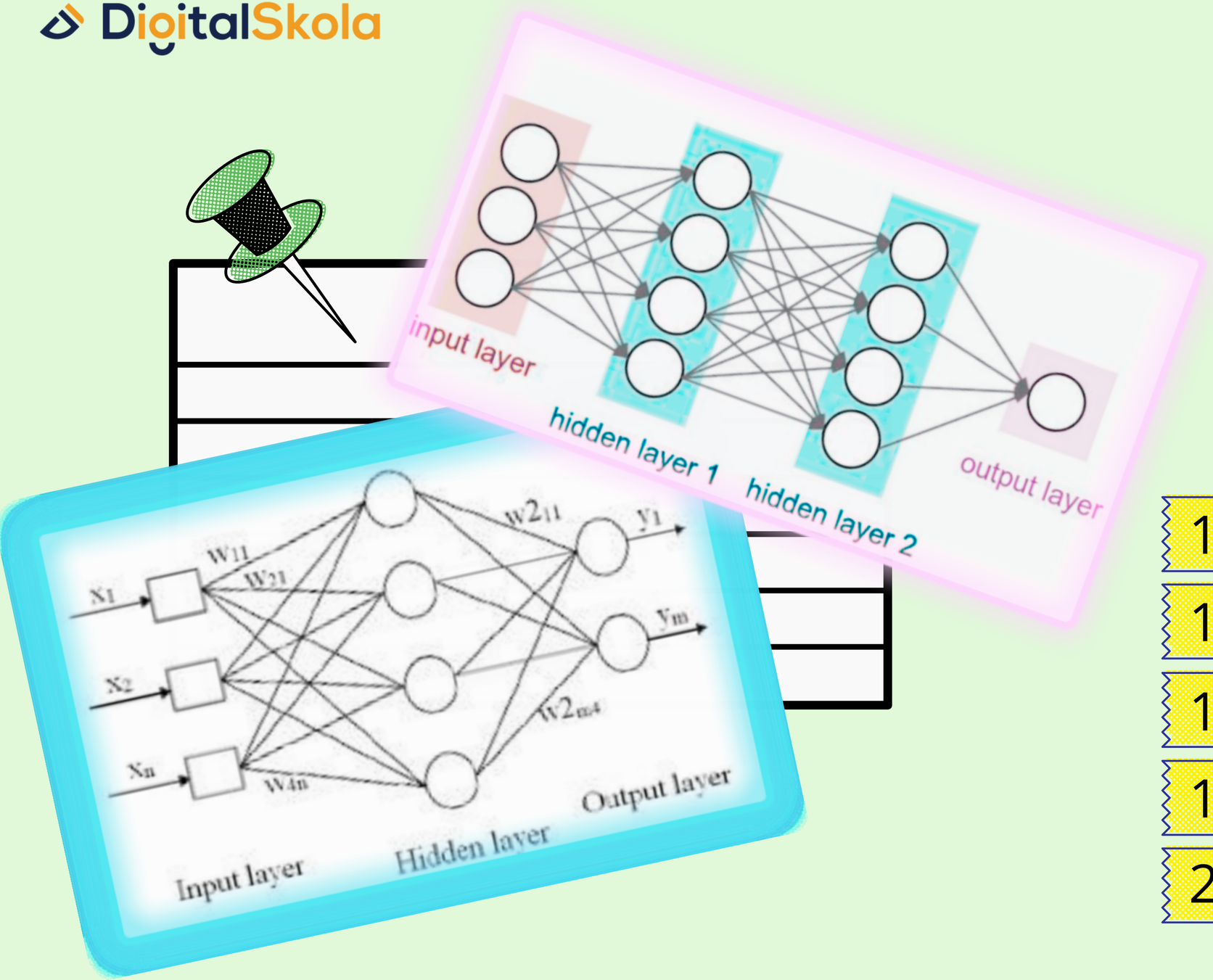
DigitalSkola

Neural Network

kumpulan unit pemrosesan sederhana yang berkomunikasi dengan mengirimkan sinyal satu sama lain melalui sejumlah besar koneksi berbobot



DigitalSkola



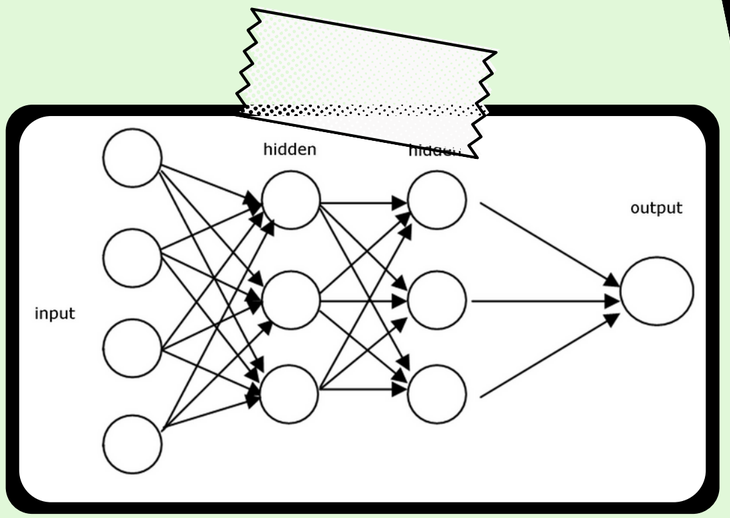
Artificial Neural Network

History

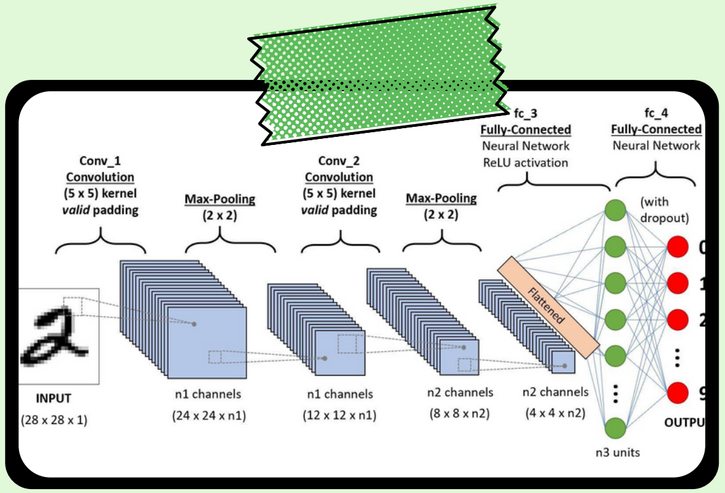
1943	**McCulloch and Pitts:** First mathematic model of neuron
1958	**Rosenblatt:** Perceptron - Single layer NN
1986	**Rumelhart:** Back Propagation algorithm
1995	**Y. LeCun, Y. Bengio, et al.:** Convolutional neural network
2006	**G. E. Hinton, et al.:** Deep belief nets

DigitalSkola

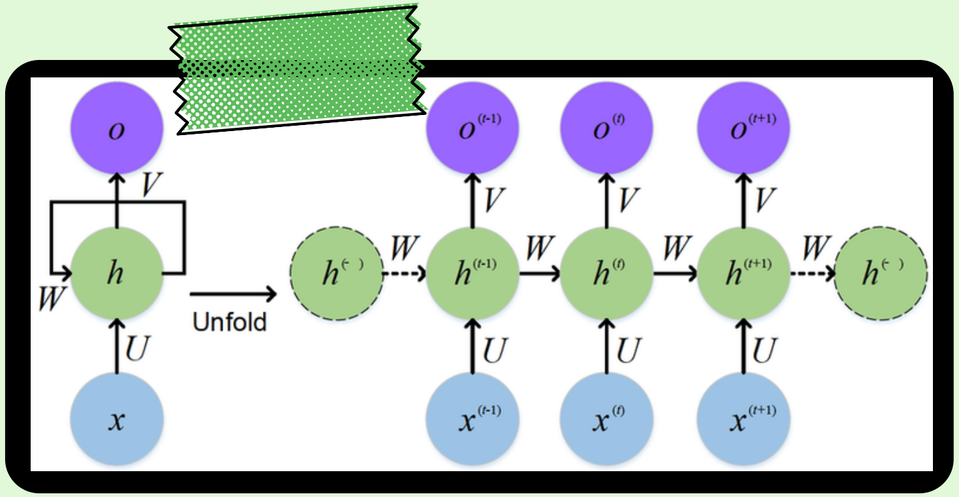
Types of ANN



Feed Forward Neural Network



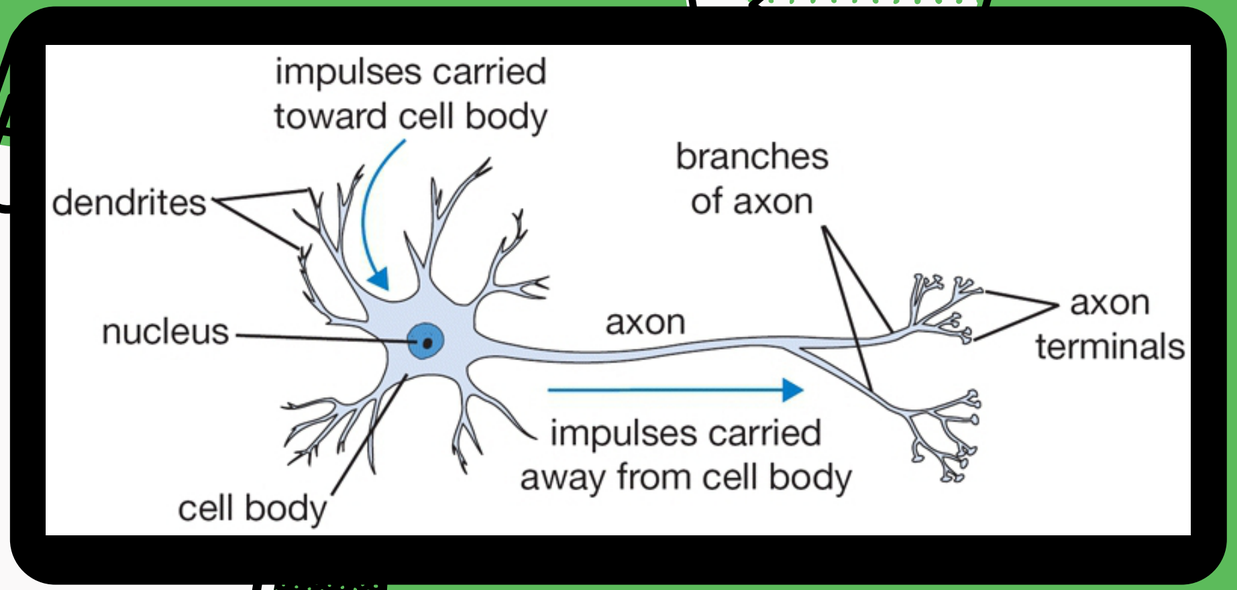
Convolutional Neural Network

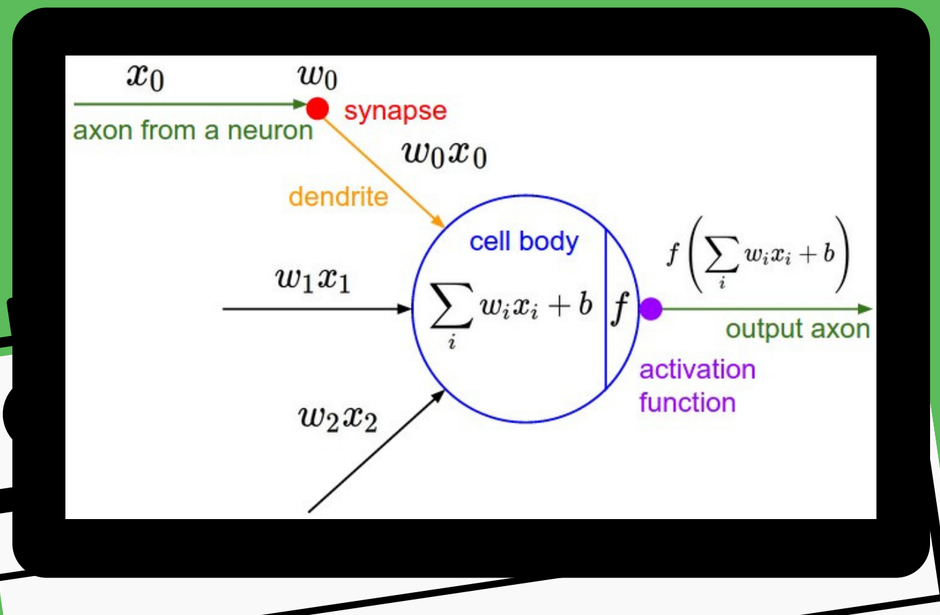


Recurrent Neural Network

DigitalSkola

Biological Motivation





- Neural Networks awalnya terinspirasi oleh tujuan pemodelan sistem saraf biologis.
- Unit komputasi dasar otak adalah neuron.
- Sekitar 86 miliar neuron dapat ditemukan di sistem saraf manusia

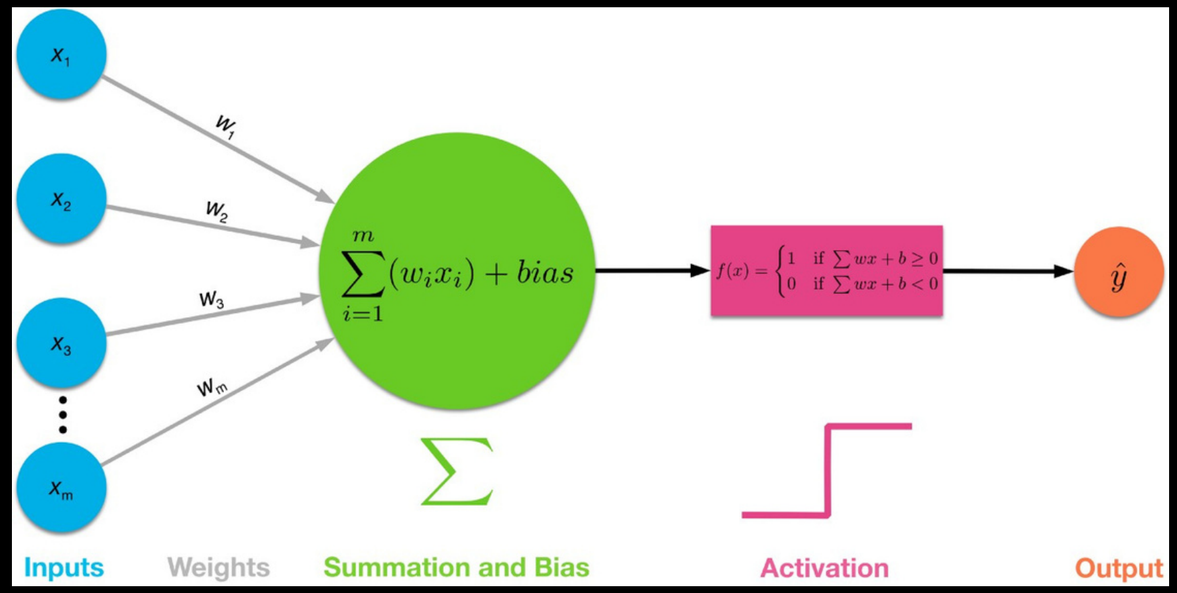
DigitalSkola

Perceptron

- Adalah pengklasifikasi linier (l*inear classifier*).
- Frank Rosenblatt, seorang psikolog Amerika, mengusulkan model perceptron klasik pada tahun 1958

Digunakan untuk mengklasifikasikan data input yang diberikan. *The perceptron* terdiri dari 4 bagian:

1. *Input values* atau *one input layer*
2. *Weights* dan Bias
3. *Net sum*
4. *Activation Function*



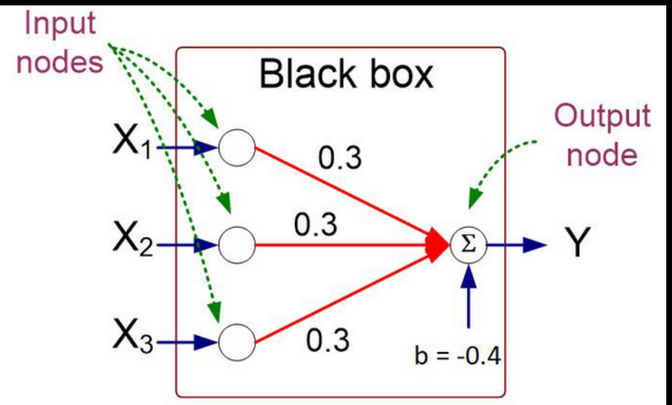
Bagaimana Perceptron bekerja?

Semua inputs x dikalikan dengan weights w

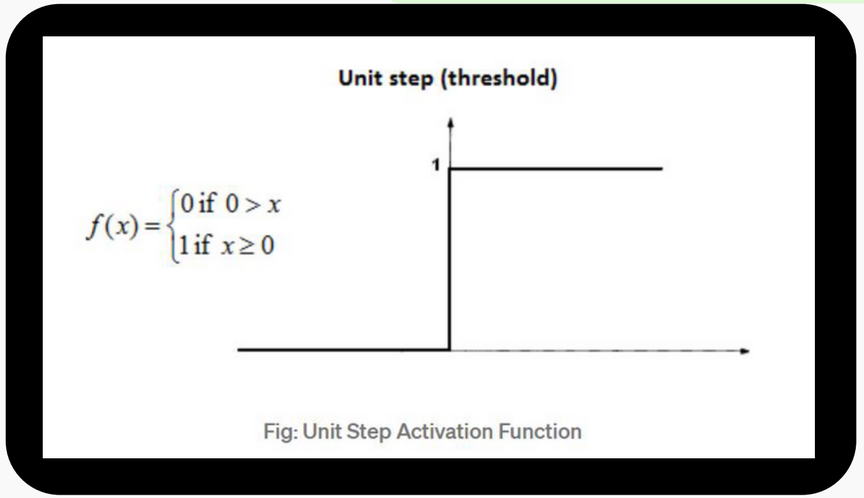
Jumlahkan semua hasil perkalikan tersebut (kita sebut *Weighted Sum*)

Mengaplikasikan *activation function* ke weighted sum

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



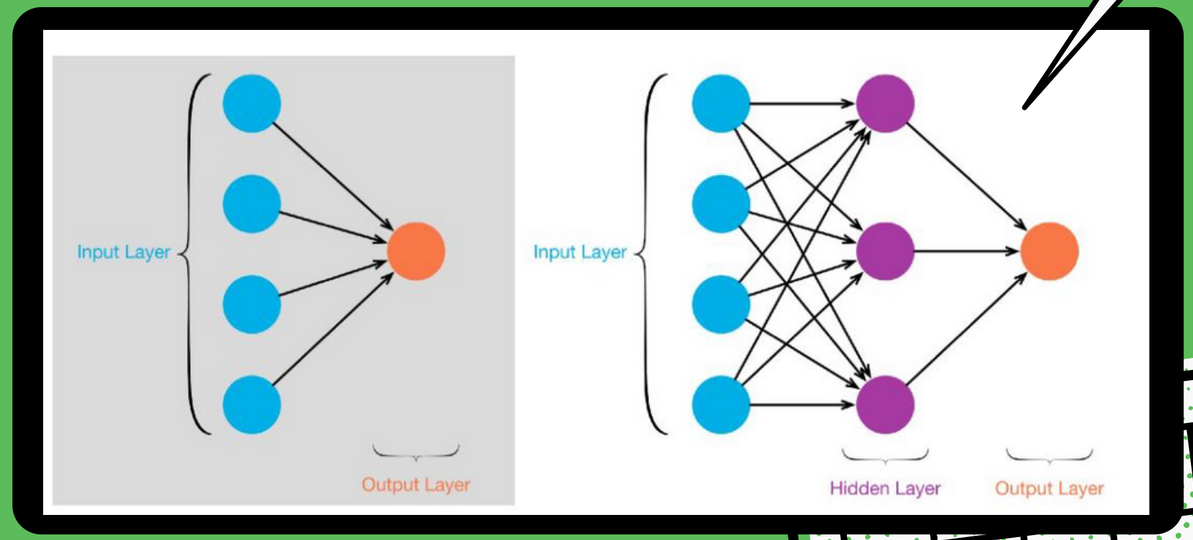
y = f(0.3*x1 + 0.3*x2 + 0.3*x3 - 0.4)



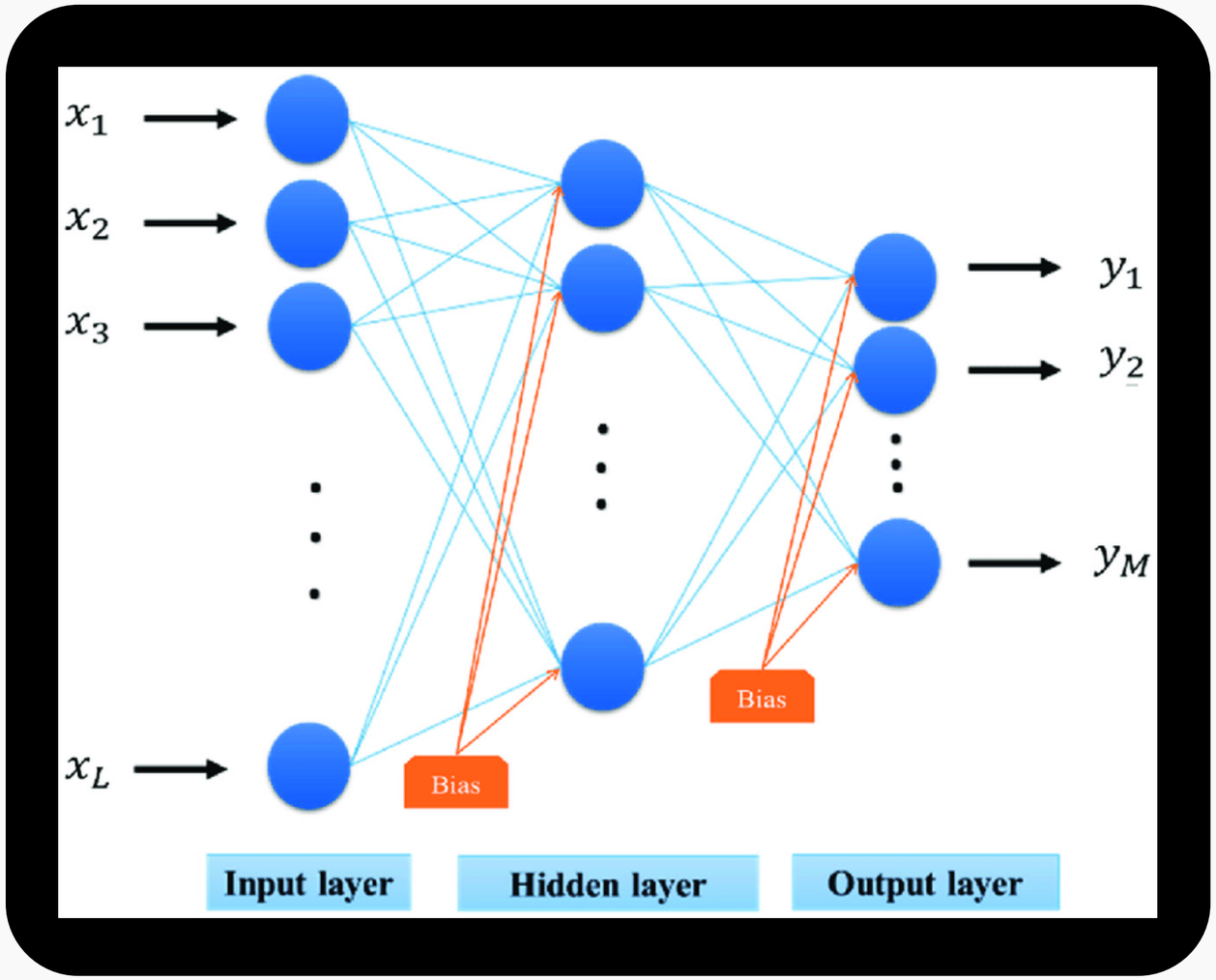
DigitalSkola

Multilayer Perceptron

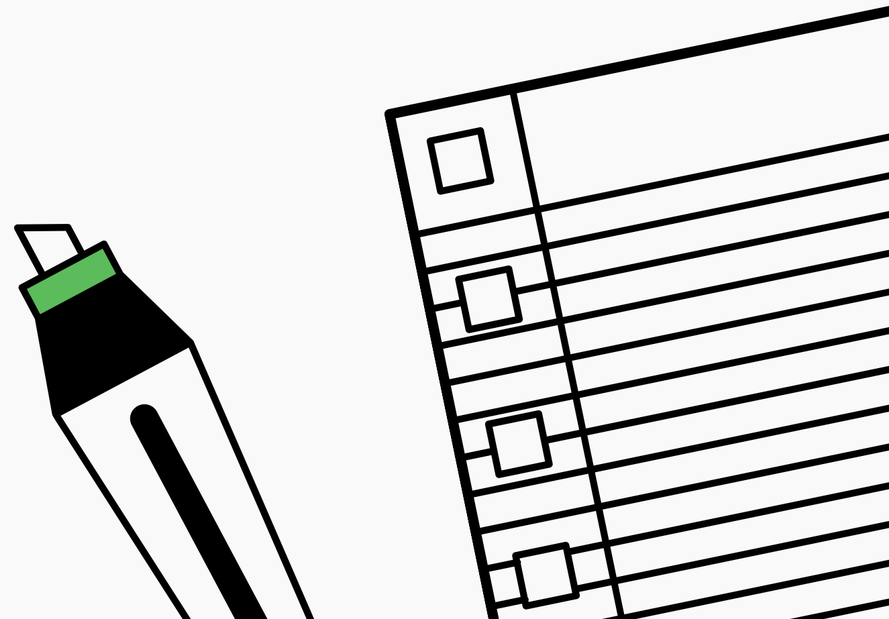
- Model ini terdiri dari tiga jenis layer — *input layer*, *hidden layer*, *output layer*.
- Kecuali node input, setiap node adalah neuron yang menggunakan fungsi aktivasi non-linier.
- MLP menggunakan *backpropagation* untuk *training*-nya.



General Structure of MLP



Training ANN means learning the weights of the neurons

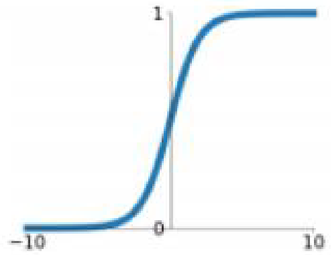


DigitalSkola

Activation Function

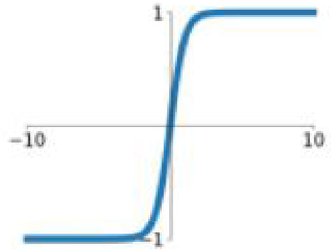
Sigmoid

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



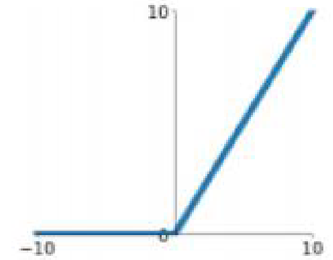
tanh

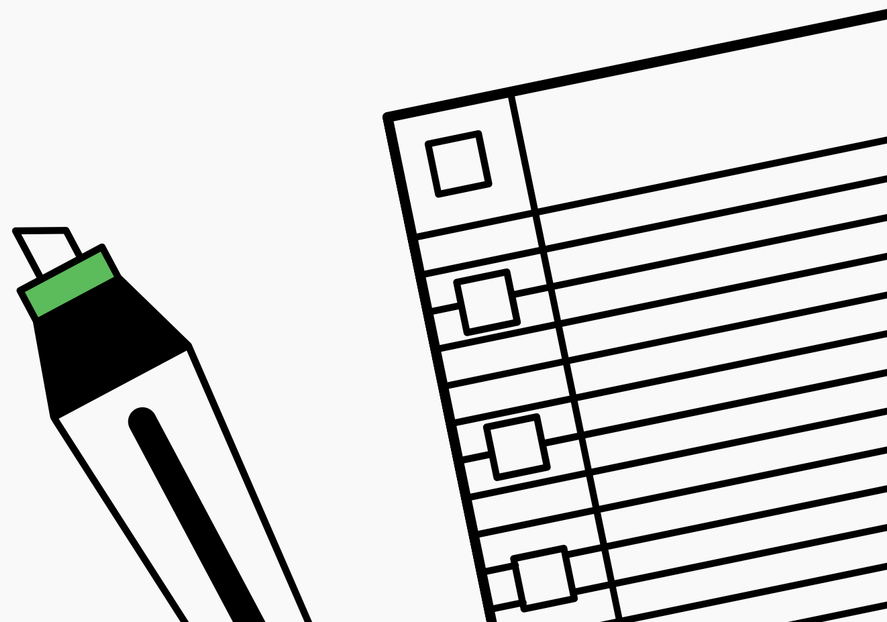
$$\tanh(x)$$



ReLU

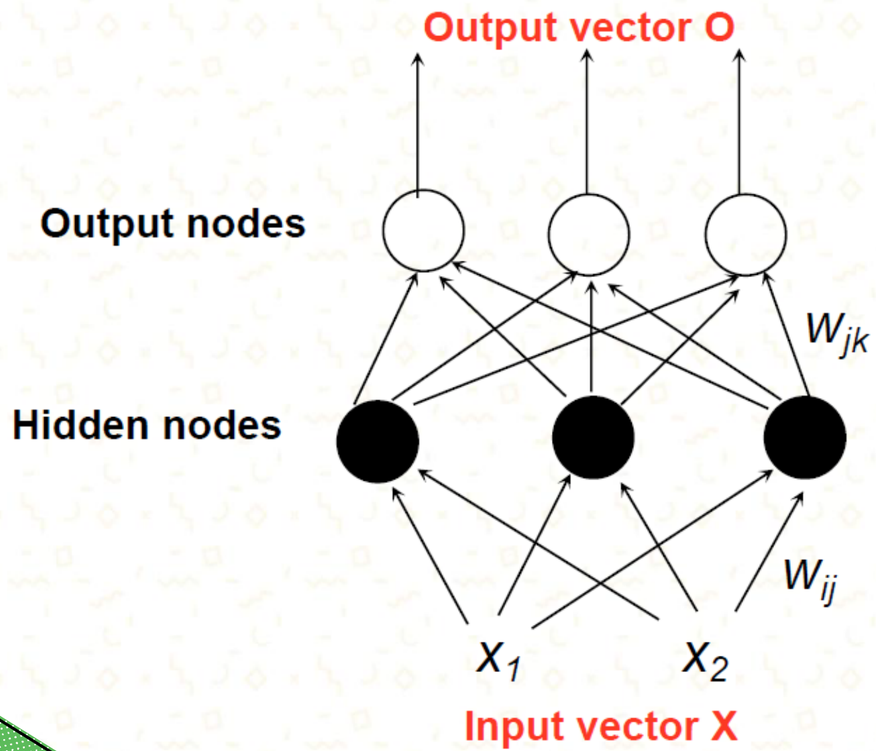
$$\max(0, x)$$





DigitalSkola

Multilayer Perceptron



$$O_k = \frac{1}{1+e^{-\sum h_j w_{jk} + \theta_k}}$$

$$h_j = \frac{1}{1+e^{-\sum x_i w_{ij} + \theta_j}}$$

DigitalSkola

Multilayer Perceptron

Tujuan

Mendapatkan weights yang membuat hampir semua sampel dalam training data dapat diklasifikasikan dengan benar

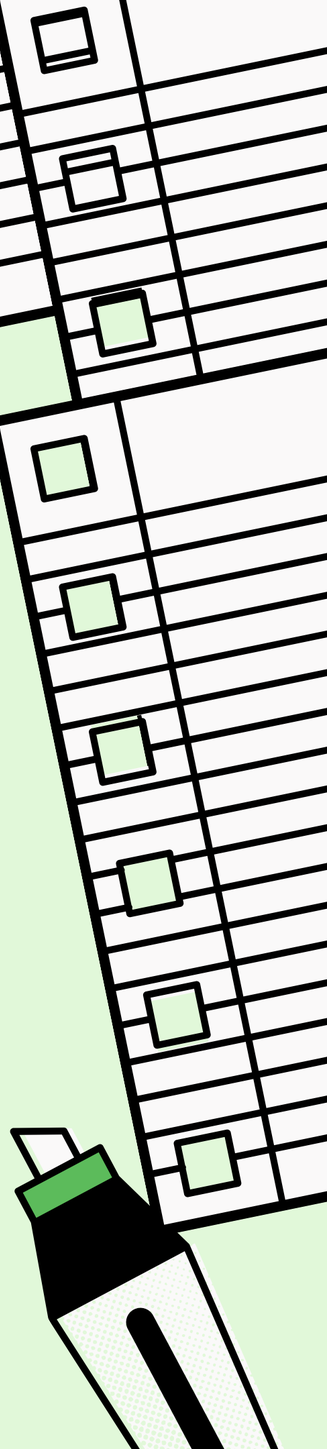
Langkah-langkah

Inisialisasi weights **wij** dengan nilai acak (random)

Masukkan (feed) sample training X ke dalam jaringan satu persatu

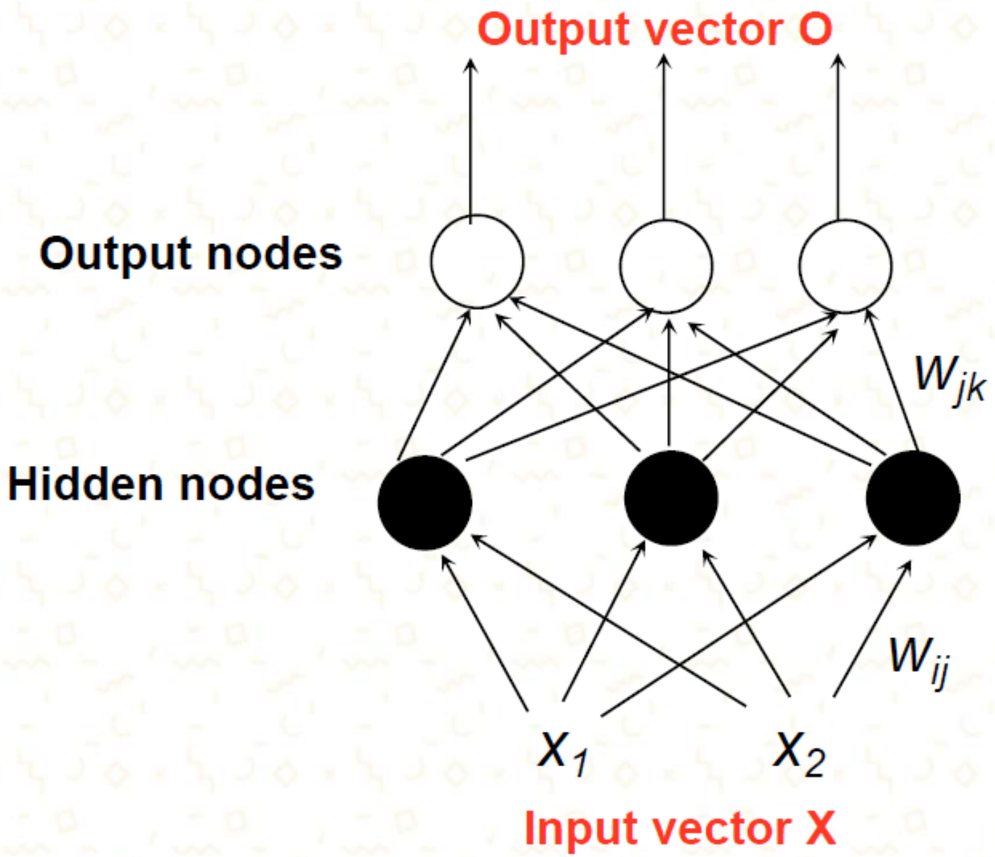
Untuk setiap unit

- Hitung *output value* **O** dengan mengaplikasikan *activation function*
- Hitung error **E**
- *Update weights* **wij** dan *biases*



DigitalSkola

Backpropagation Learning



$$O_k = \frac{1}{1+e^{-\sum h_j w_{jk} + \theta_k}}$$

$$h_j = \frac{1}{1+e^{-\sum x_i w_{ij} + \theta_j}}$$

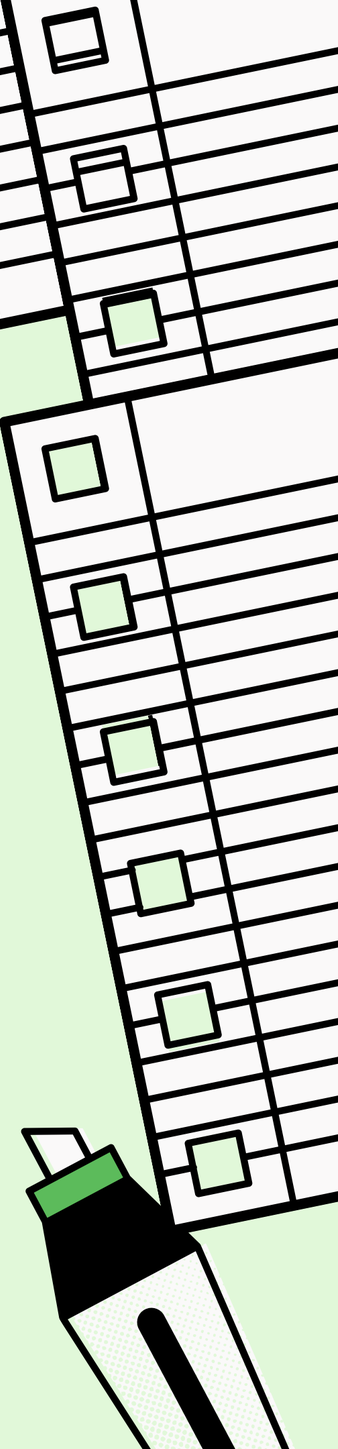
DigitalSkola

MLP in Sklearn

Sklearn menyediakan class MLPClassifier

class sklearn.neural_network.MLPClassifier(*hidden_layer_sizes=(100), activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000*) [source]

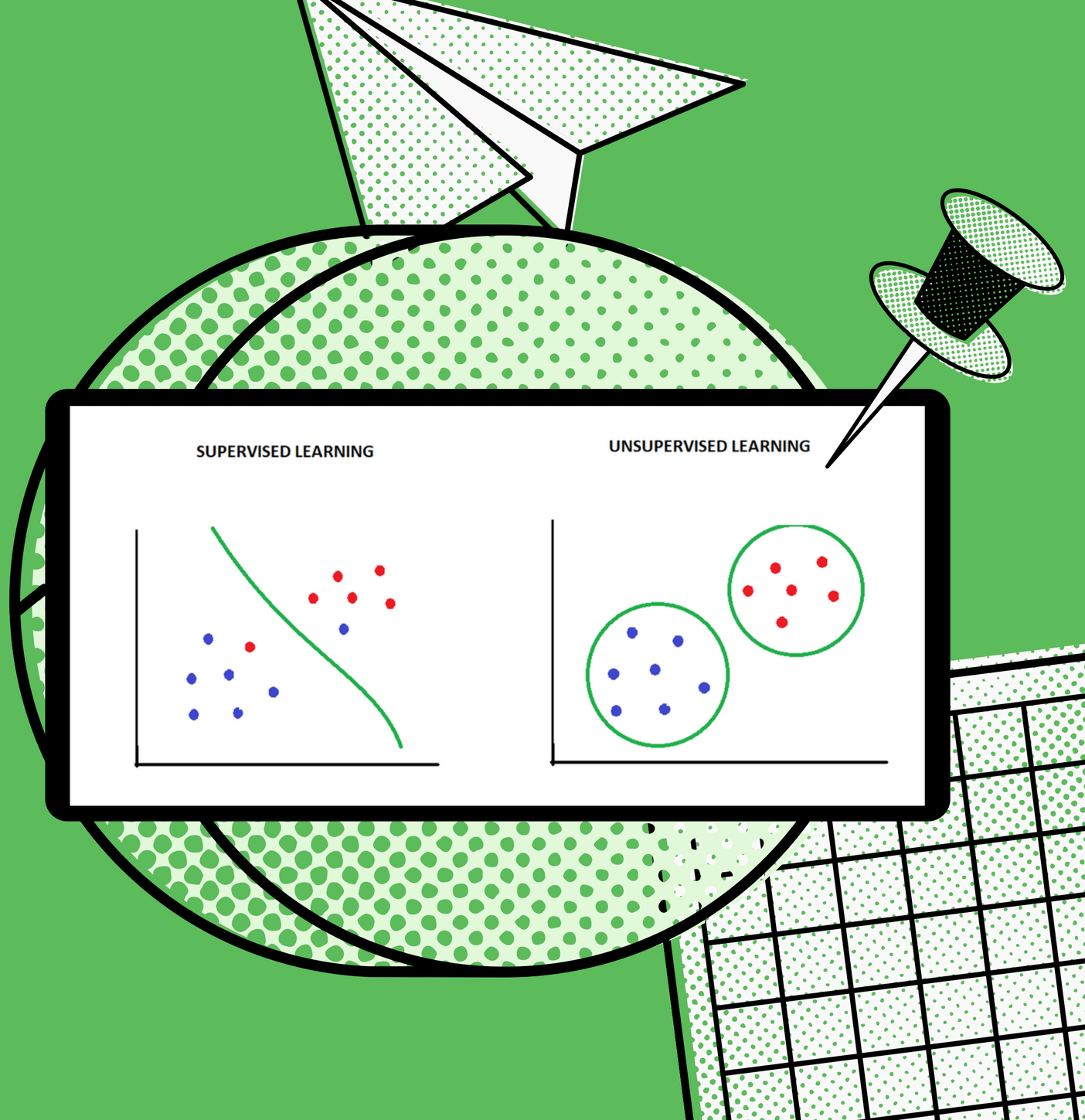
Hyperparameter	Description
hidden_layer_sizes	Jumlah hidden layer dan node-nya
activation	Activation function
max_iter	Jumlah iterasi



DigitalSkola

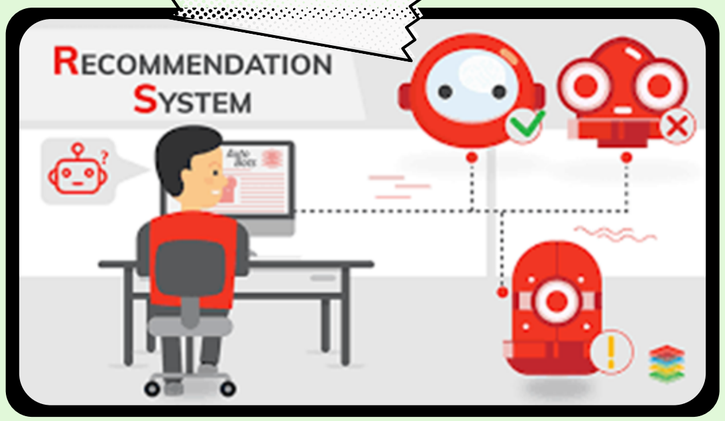
Unsupervised Learning

mencari pola atau karakteristik data

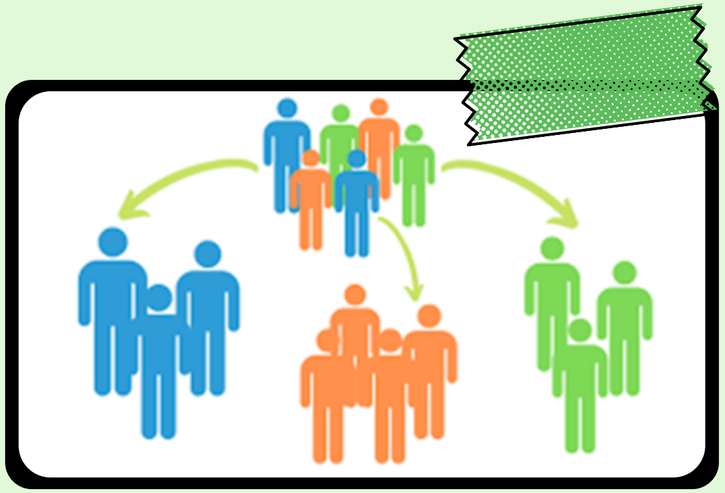


DigitalSkola

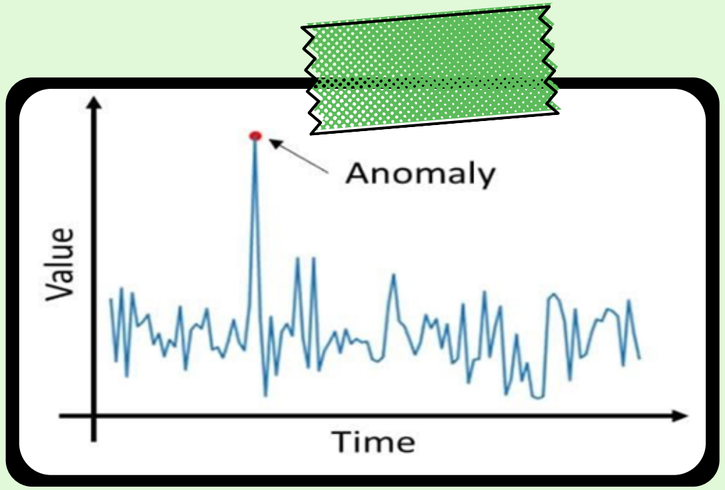
Penerapan Unsupervised Learning



Recommender System



Customer Segmentation

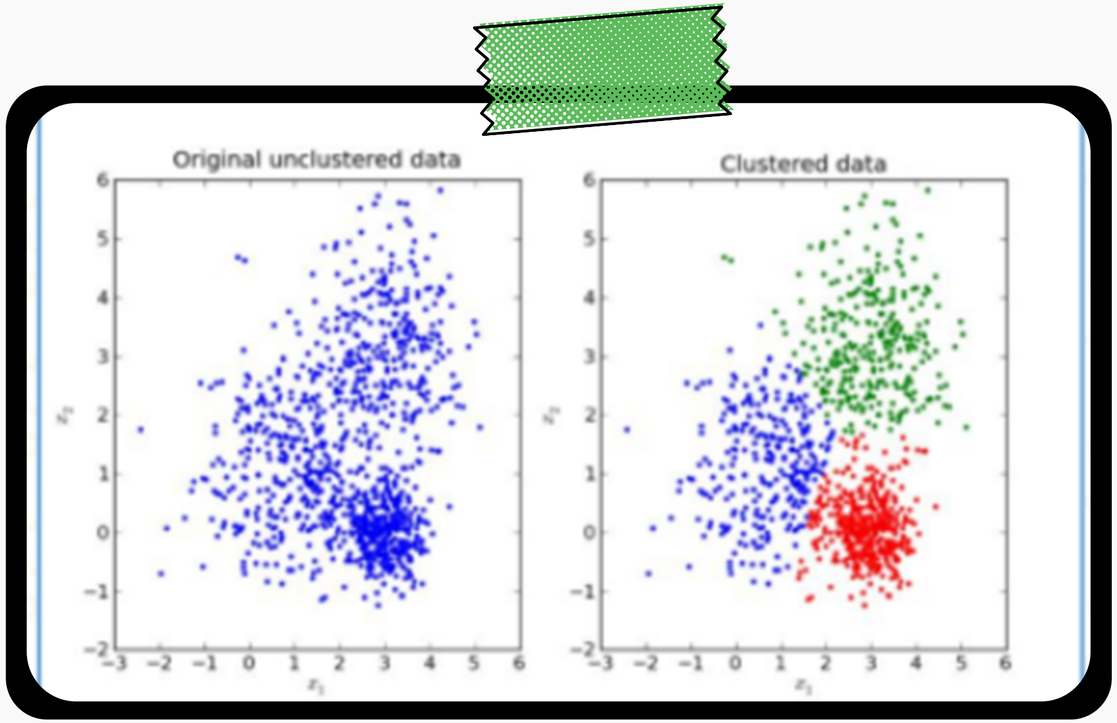


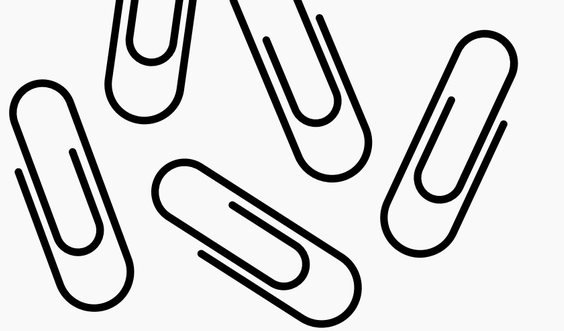
Anomaly Detection

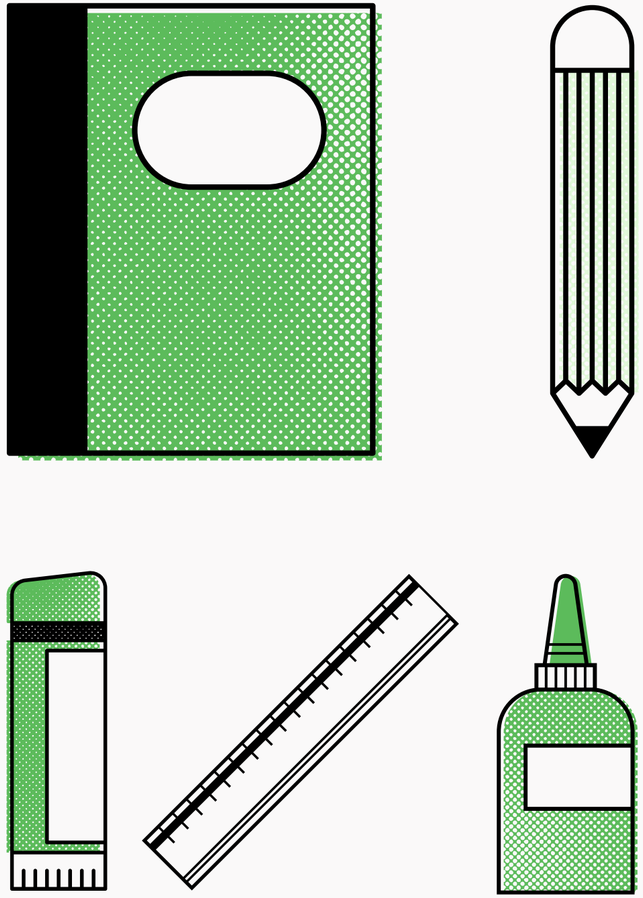
DigitalSkola

K-Means

- Algoritma K-Means mengelompokkan item ke dalam bentuk *cluster*
- Jumlah kelompok direpresentasikan sebagai **K**



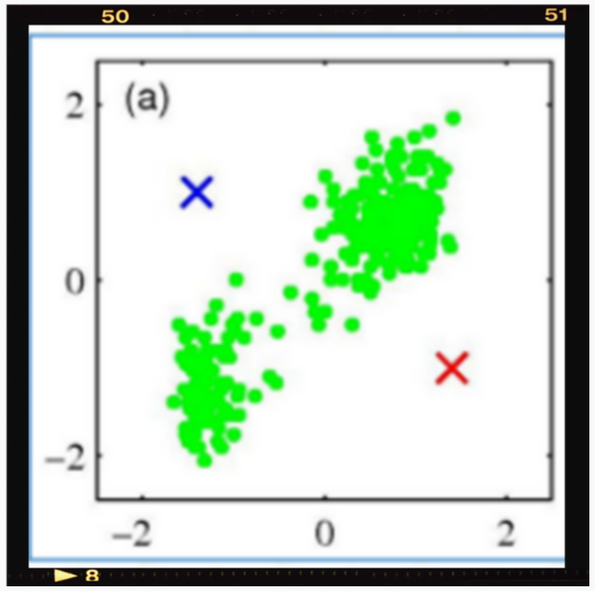




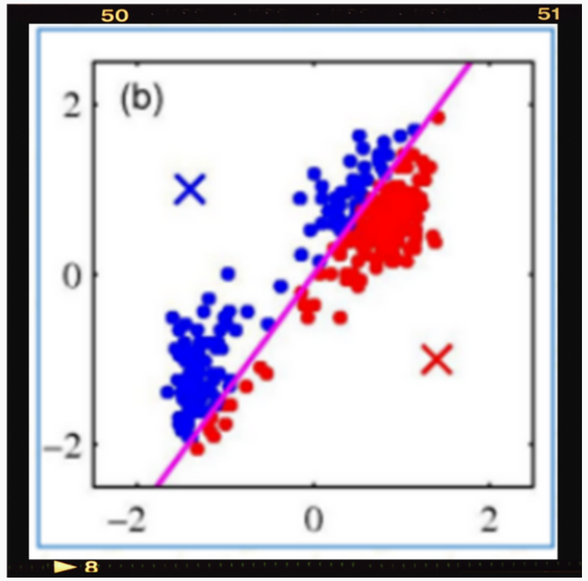
Bagaimana K-Means bekerja?

Bagaimana K-Means bekerja?

Memilih titik acak sejumlah **K** sebagai titik tengah (*means*). ex: **K** = 2

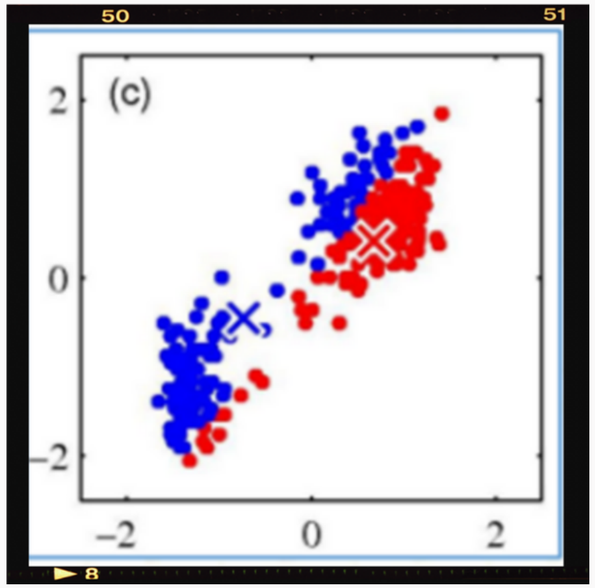


Iterative step 1.
Menetapkan data point ke titik tengah *cluster* terdekat

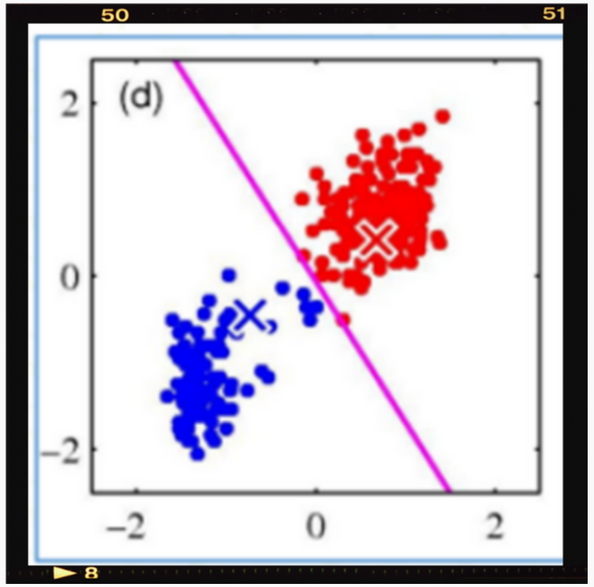


Bagaimana K-Means bekerja?

Iterative step 2.
- update titik tengah
- ganti titik tengah dengan *mean* poin dalam *cluster*



Mengulangi proses sampai mencapai ***convergence***



DigitalSkola

Evaluasi Performa Clustering

1. Inertia

•Jumlah Kuadrat (sum of squared distance) jarak dari setiap titik (xi) dengan clusternya (Ck).

•Jika Inertia kecil berarti poin dekat satu sama lain.

$$\sum_{i=1}^{n} (x_i - C_k)^2$$

2. Silhoutte Score

•a : jarak rata – rata ke semua titik lain dalam cluster.

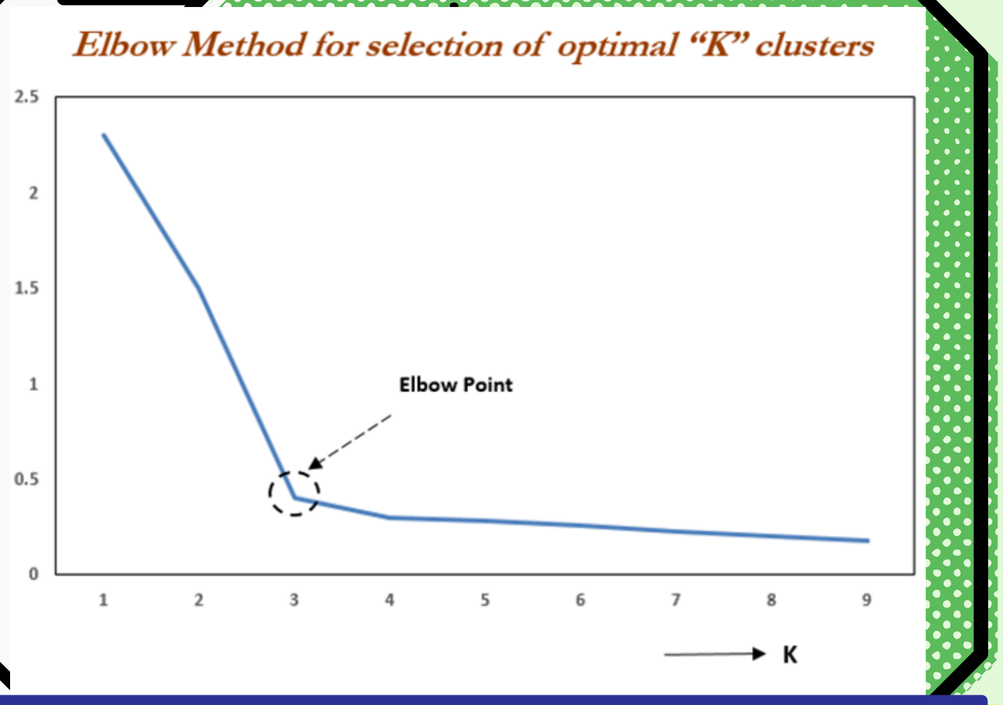
•b : jarak rata – rata ke semua titik lain di cluster terdekat berikutnya.

•Score di antara -1 sampai 1 (Semakin mendekati 1 semakin baik).

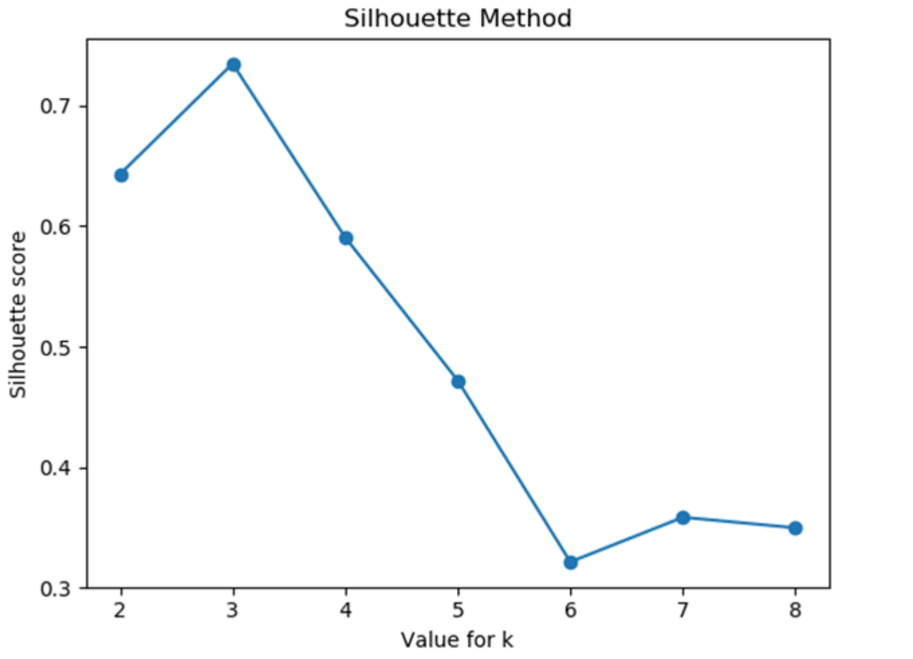
$$SC = \frac{b - a}{\max(a, b)}$$

DigitalSkola

Bagaimana memilih nilai K?



Elbow Method



Silhoutte Score

DigitalSkola

Keuntungan & Kekurangan K-Means

Keuntungan

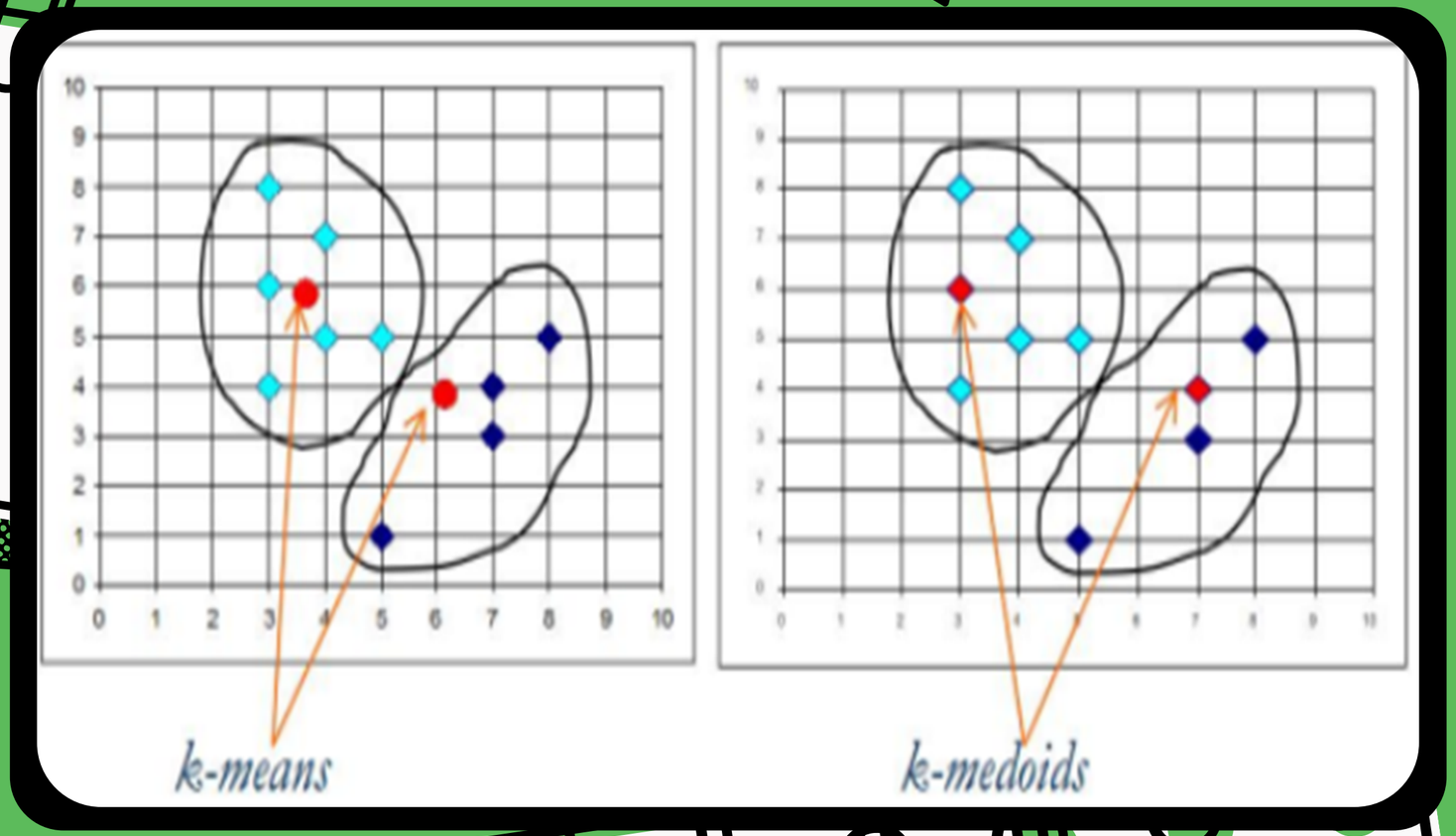
- Mudah untuk diimplementasikan
- Mampu menangani dataset yang besar
- Generalisasi cluster untuk berbagai bentuk dan ukuran

Kekurangan

- Sensitif terhadap nilai pencilan (*outliers*)
- Memilih nilai **k** butuh usaha lebih
- Ketika jumlah dimensi meningkat, skalabilitasnya menurun

DigitalSkola

K Medoids



- K-Medoids memilih representatif object sebanyak k yang disebut sebagai medoids.
- K-Medoids meminimalkan jumlah jarak (sum of distances) antara setiap titik dengan medoid dari clusternya.

DigitalSkola

Bagaimana K-Medoids Bekerja ?

Initialize

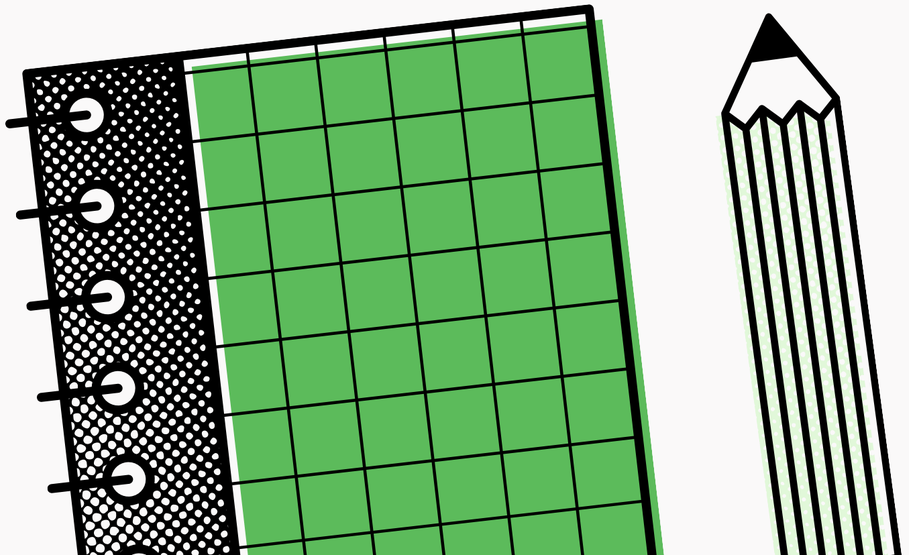
Pilih titik acak sebanyak **k** (disebut sebagai medoids)

Repeat

- Tetapkan setiap titik ke cluster dengan dengan jarak terdekat terhadap medoid m
- Memilih objek baru oi dalam cluster
- Hitung **total cost of swapping S**, antara medoid m dengan oi
- Jika S < 0 : Ganti m dengan oi untuk membentuk kumpulan medoids

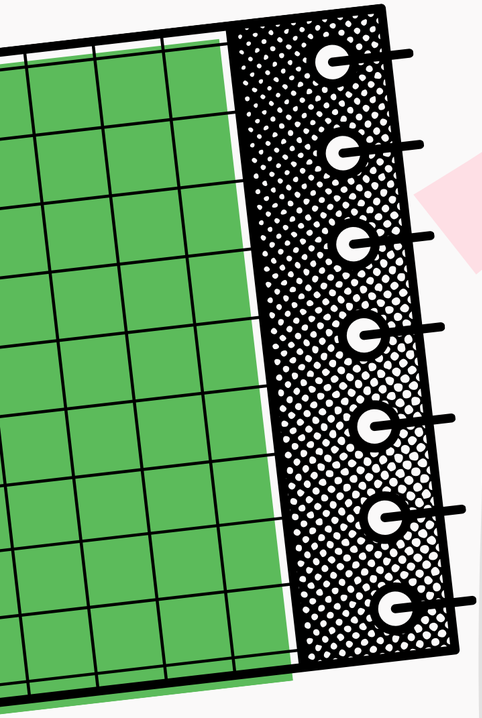
Keep Repeating

Mengulangi proses hingga tercapai ***convergence***



DigitalSkola

Keuntungan & Kekurangan K-Medoids



Keuntungan

- Mudah untuk diimplementasikan
- Algoritma K-Medoids bekerja dengan cepat dan dapat ***convergence*** dalam jumlah tahapan tertentu
- PAM tidak terlalu *sensitive* terhadap *outliers*

Kekurangan

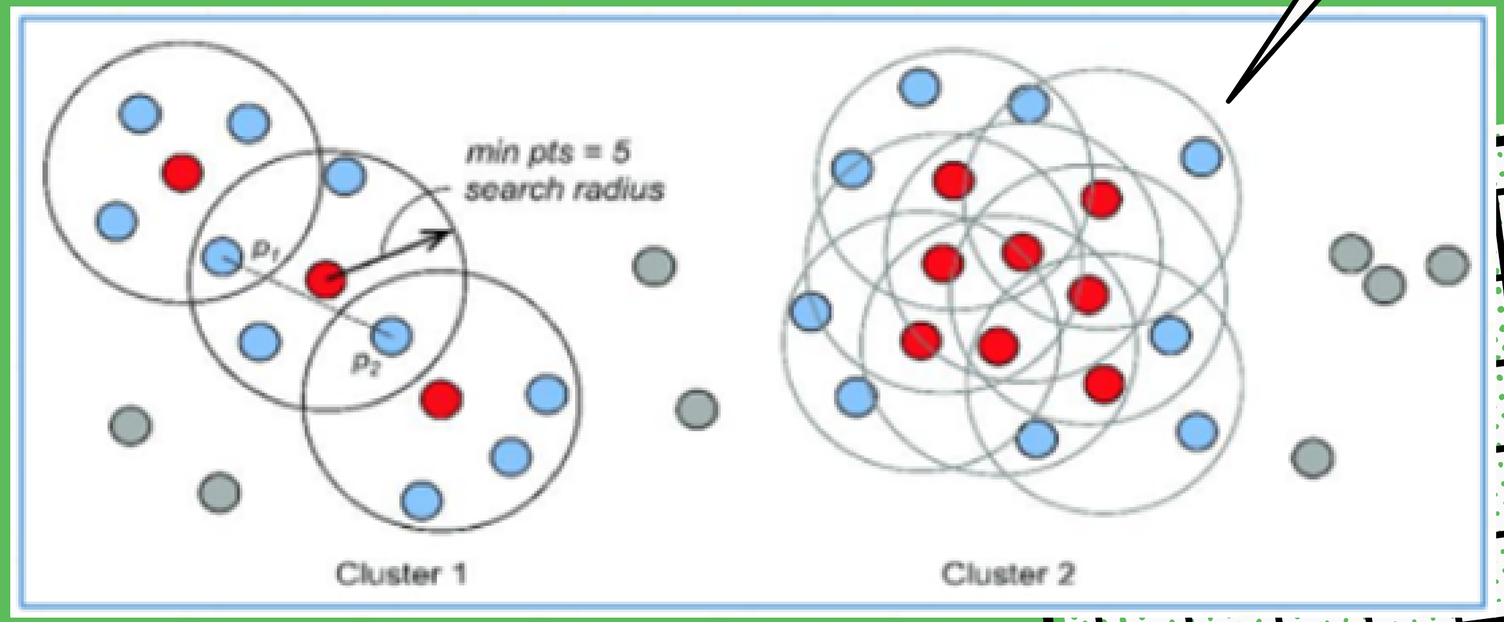
- Hasilnya bisa berbeda ketika dijalankan, karena nilai medoids k dipilih secara acak saat pertama kali
- Algoritma PAM dalam K-Medoids *clustering* bekerja baik pada dataset tapi tidak bisa "***scale well***" untuk dataset yang besar

DigitalSkola

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

- Adalah algoritma berbasis *density-based*
- Diusulkan oleh Martin Ester, Hans-Peter Kriegel, Jorg Sander, dan Xiaowei Xu pada tahun 1996



DigitalSkola

Bagaimana DBSCAN Bekerja ?

Grouping

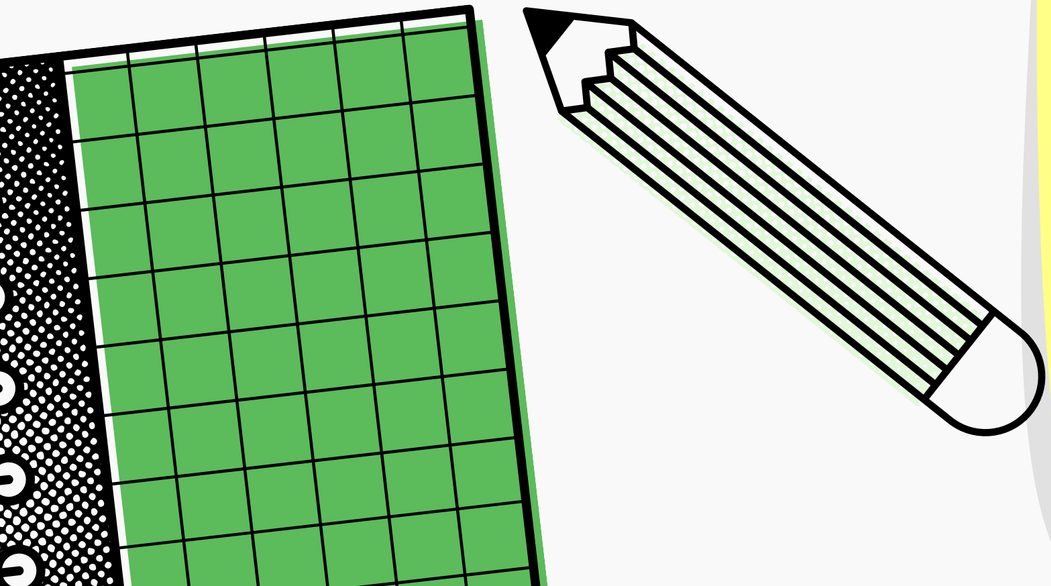
Mengelompokkan objek dalam area yang padat (*dense region*)

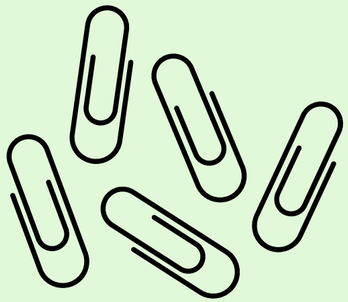
Density Parameters

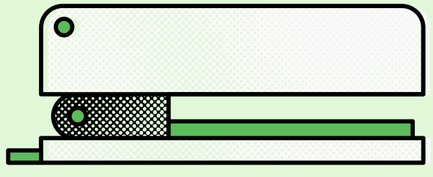
- Radius ε : Jarak untuk menentukan tetangga.
- MinPts : Jumlah minimum poin dalam lingkungan (*neighborhood*)

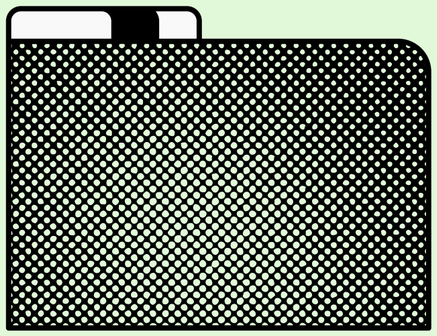
Fitur Utama

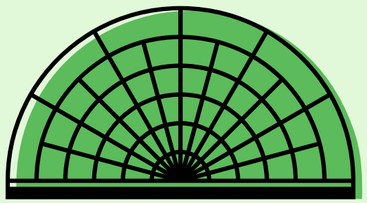
- Bisa menemukan *cluster* yang bentuknya *arbitrary*
- Bisa mengatasi *noise*
- Satu kali *scanning*
- Membutuhkan *density parameters*







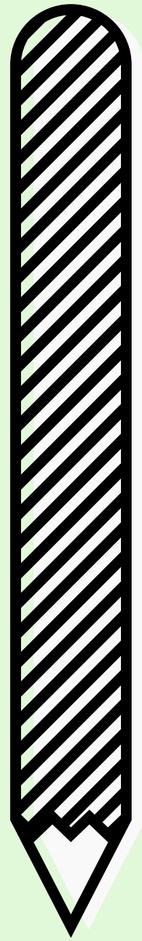




Cara Menentukan Nilai Radius & MinPts

Radius

- Menggunakan ***domain knowledge***
- Semakin besar dataset, semakin besar nilai MinPts
- Jika dataset > *noise* maka isi MinPts dengan nilai yang lebih besar
- Umumnya MinPts >= atau = jumlah dimensi dataset
- Untuk 2-dimensional data, gunakan MinPts = 4 (Ester et al., 1996)
- Untuk dataset > 2-dimensional data, gunakan MinPts = 2*dim. Dim = jumlah dimensi dataset (Sander et al., 1998)



MinPts

- Sorted k-dist graph

DigitalSkola

Terima Kasih!