

KECERDASAN BUATAN MODUL 4

METODE MACHINE LEARNING SUPERVISED LEARNING (K-NN DAN SVM)

Mochammad Zen Samsono Hadi

Course Definition

- Modul ini adalah bagian pertama dari Membangun Model
- Membangun Model yang dibahas adalah :
 - Merancang Skenario Model
 - Membangun Model Klasifikasi
- Terdapat beberapa algoritma yang akan dibahas yaitu k-Nearest Neighbor, Support Vector Maching, Decision Tree, dan Random Forest
- Pembangunan model menggunakan library

Machine Learning

Supervised Learning

- Classification
- Regression

Unsupervised Learning

- Clustering
- Dimension Reduction

Reinforcement Learning

- Dynamic Programming
- Monte Carlo Method

Kebutuhan

Cocok Pakai Fuzzy

Cocok Pakai Machine Learning

Sedikit data



Banyak data historis

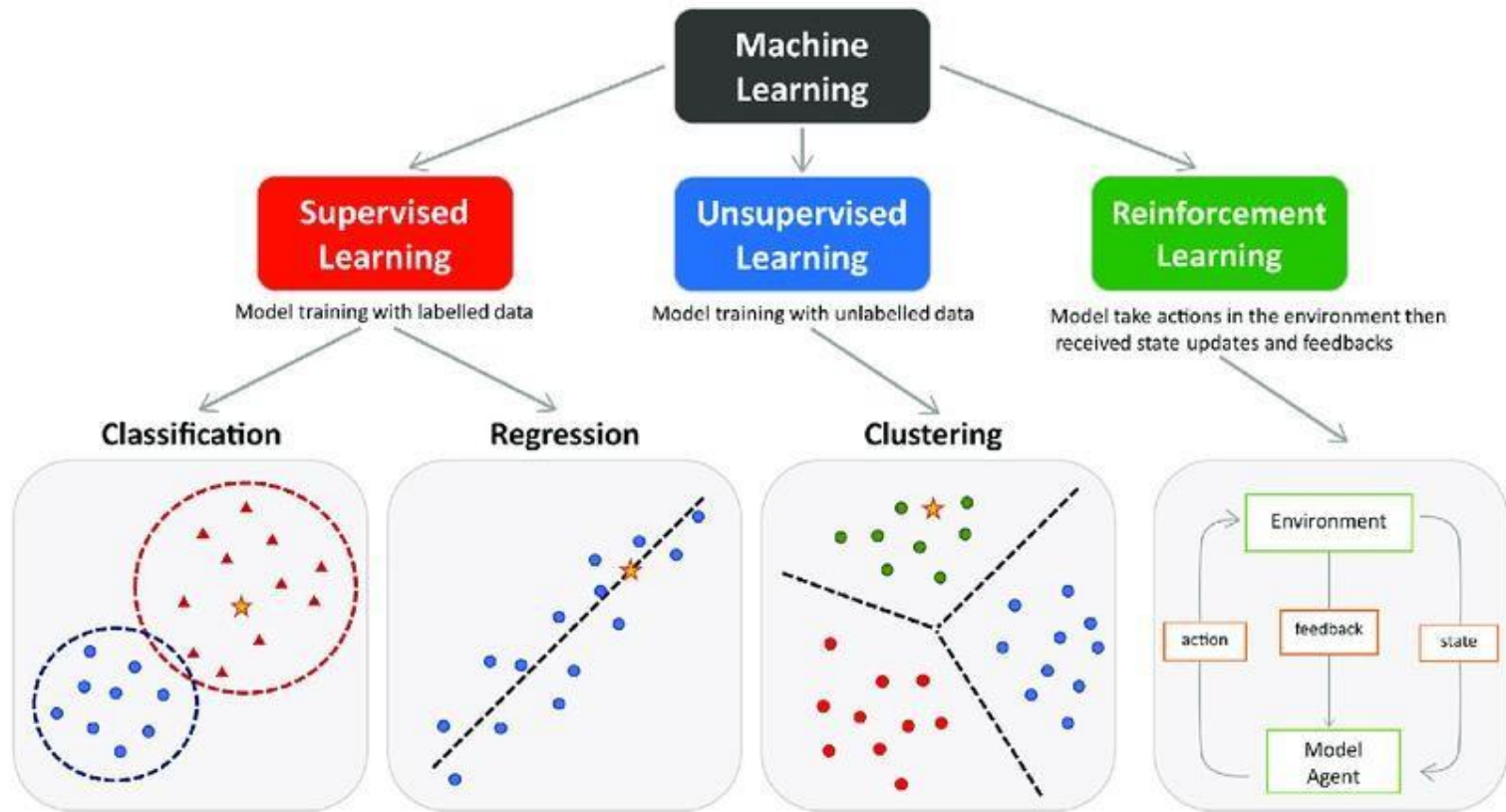


Logika manusia sulit ditulis dalam aturan



Sistem kontrol real-time



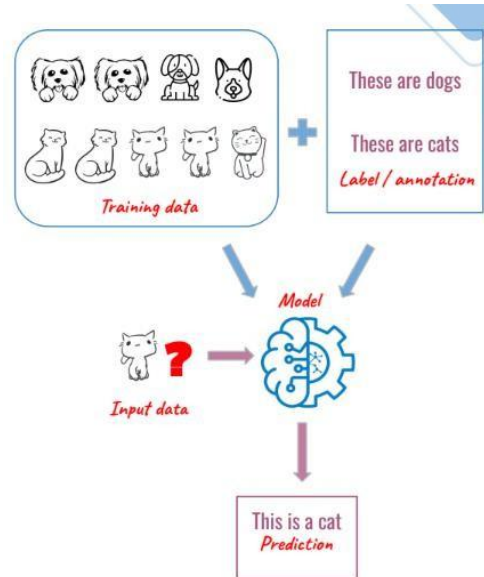


Source: [link](#)

Supervised Learning

pendekatan machine learning yang menggunakan **data-data yang sudah diberi label/class** atau **dataset-nya** sudah diketahui oleh perancangnya.

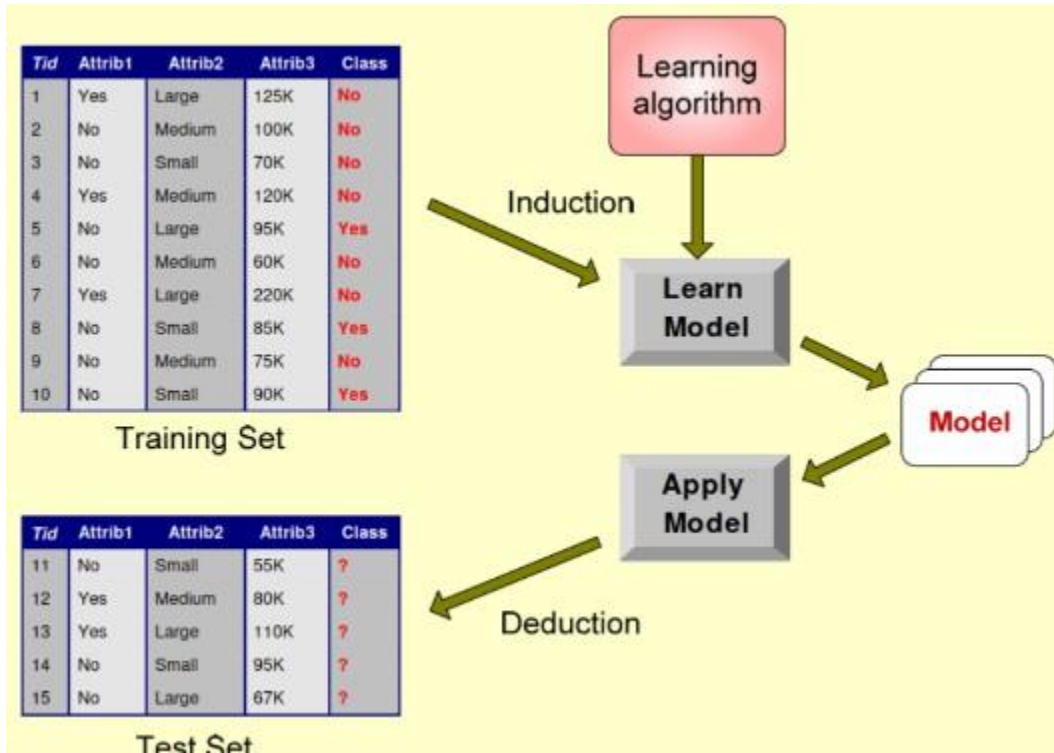
- Classification
 - Nearest Neighbors
 - Naïve Bayes
 - Decision Tree
 - etc...
- Regression
 - Linear Regression
 - etc...



Definisi: Klasifikasi

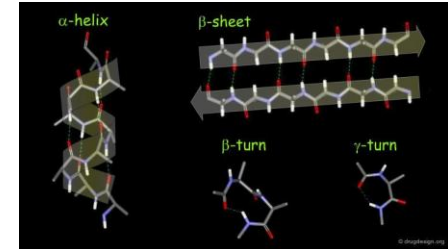
- Diberikan koleksi dari data-data atau records (**training set**)
 - Setiap record/data memiliki sekumpulan **atribut** (attributes), salah satu dari atribut adalah **kelas** (class)
- Temukan **model** untuk atribut kelas sebagai fungsi dari nilai atribut lainnya.
- Goal: data/record yang sebelumnya tidak memiliki kelas, harus **diberikan/ditentukan kelasnya seakurat mungkin**.
 - Sebuah **test set** digunakan untuk menentukan akurasi dari model. Biasanya **data set** yang diberikan **dibagi menjadi training dan test sets**, dimana **training set digunakan untuk membangun model** dan **test set untuk memvalidasinya**.

Ilustrasi Cara Kerja Klasifikasi



Contoh Pekerjaan Klafisikasi

- Prediksi sel tumor apakah jinak atau ganas
- Klasifikasi transaksi kartu kredit normal atau mencurigakan
- Mengklasifikasikan struktur sekunder protein sebagai alpha-helix, beta-sheet, atau random coil
- Mengkategorikan berita sebagai keuangan, cuaca, hiburan, olahraga, dll



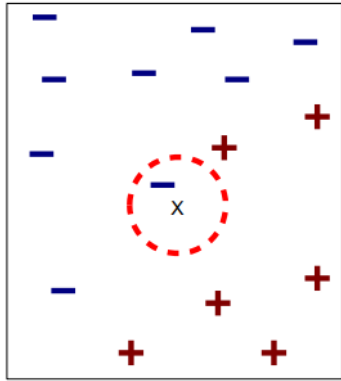
Outline

- **Membangun Model Klasifikasi :**
 - Algoritma klasifikasi yang diimplementasi menggunakan library, yaitu : k-NN, Decision Tree, Naïve Bayes, Support Vector Machine dan Boosting
 - Matriks Performansi Klasifikasi
- **Membangun Skenario Pemodelan :**
 - Pembagian data : data latih-uji, *k*-fold cross validation
 - Menentukan Langkah Eksperimen
 - Parameter Evaluasi

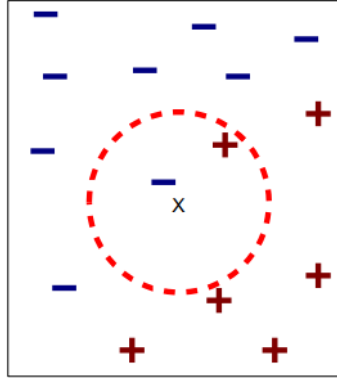
Nearest Neighbors

- Metode simple untuk mengklasifikasikan data baru berdasarkan **kesamaan/kedekatan** (*similarity*) dengan data berlabel (labelled data)
- Kedekatan(similarity) biasanya dihitung menggunakan **metriks jarak**
- Perhitungan jarak biasanya menggunakan *Euclidian*
- Mempunyai beberapa nama lain: lazy algorithm, memory-based, instance-based, exemplar-based, case-based, experience-based

k – Nearest Neighbor (k-NN) Classifier



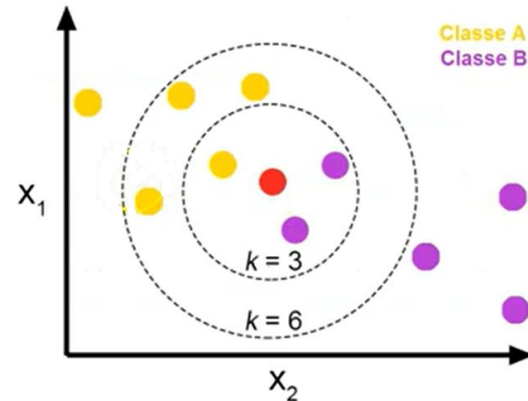
(a) 1-nearest neighbor



(b) 2-nearest neighbor

Nearest Neighbor terhadap data baru (x)

- Algoritma :
 - Menentukan nilai k
 - Menghitung jarak antara data baru terhadap semua training data
 - Mengidentifikasi k nearest neighbor
 - Menentukan label/kelas data baru berdasarkan kelas k-nearest neighbor (dapat menggunakan **majority voting**)



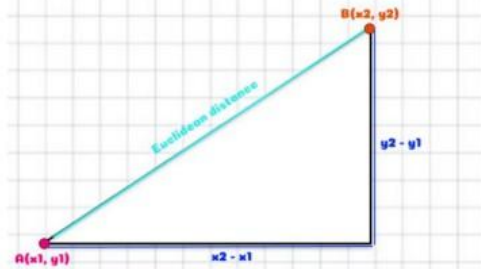
Metode Perhitungan Jarak

- **Jarak** merupakan pendekatan yang umum dipakai untuk menggambarkan **kedekatan data / atribut / fitur**. Semakin kecil nilai jaraknya, semakin dekat kesamaan kedua data / atribut / fitur tersebut.
- **Metode perhitungan jarak:**
 - Jarak Euclidean
 - Jarak City-Block
 - Jarak Chebychef
 - Jarak Minkowski
 - Jarak Canberra
 - Jarak Bray-Curtis
 - Divergensi Kullback Leibler
 - Divergensi Jensen Shannon

Klasifikasi dengan 1-NN

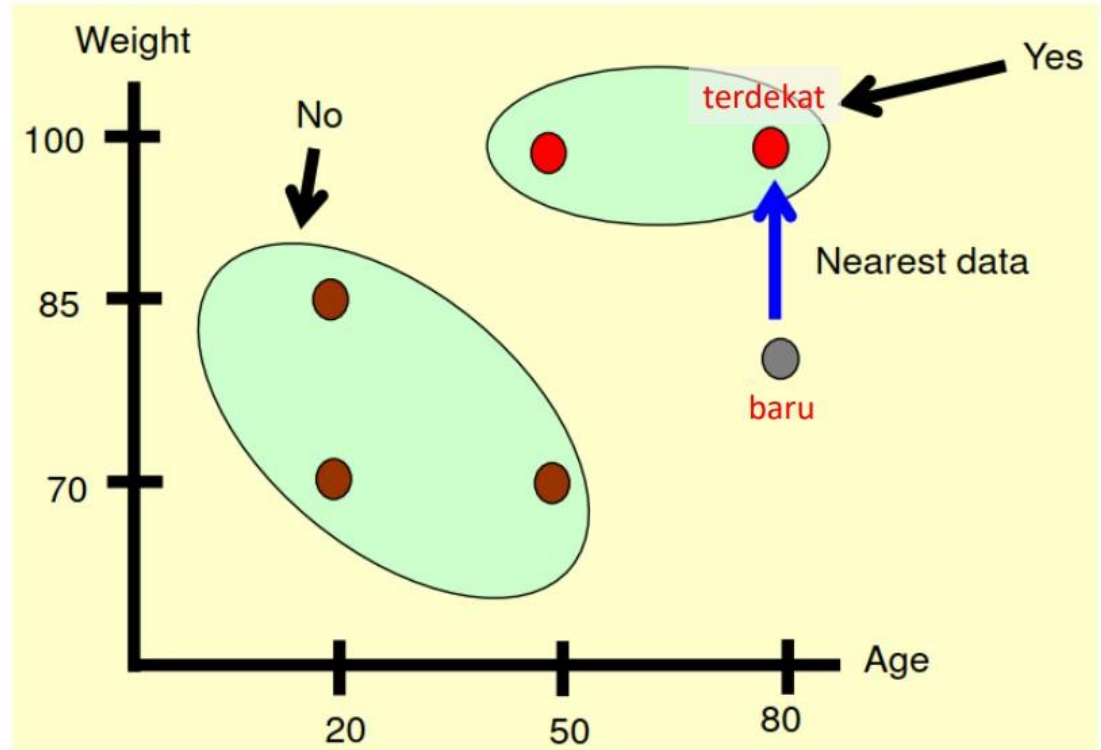
Euclidean Distance

$$\text{Euclidean}(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

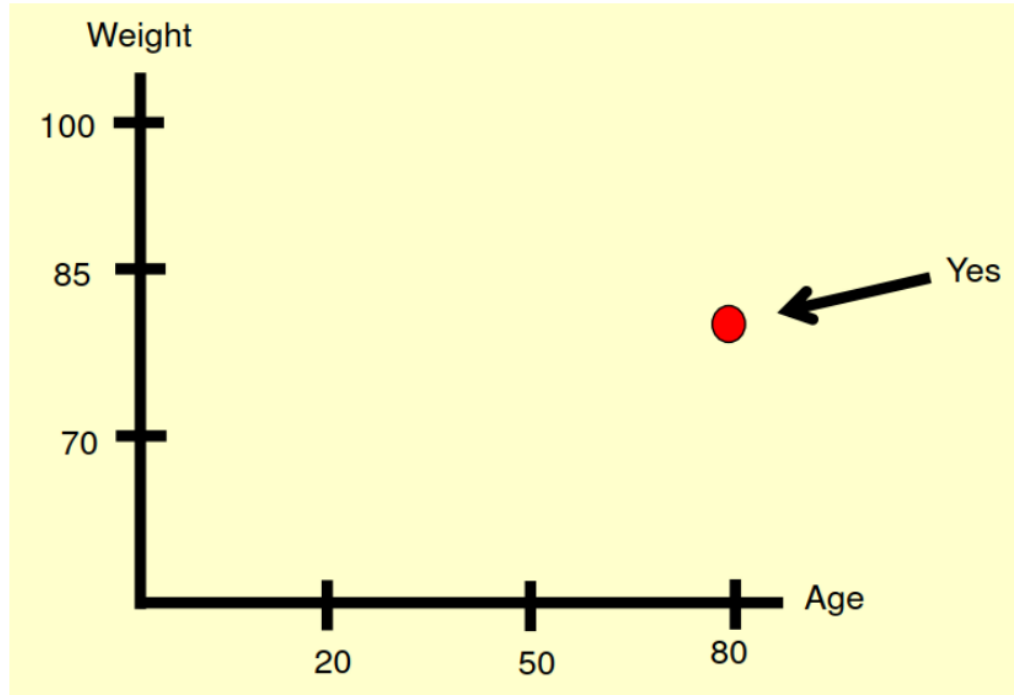


Jarak antara data baru dengan nearest data:

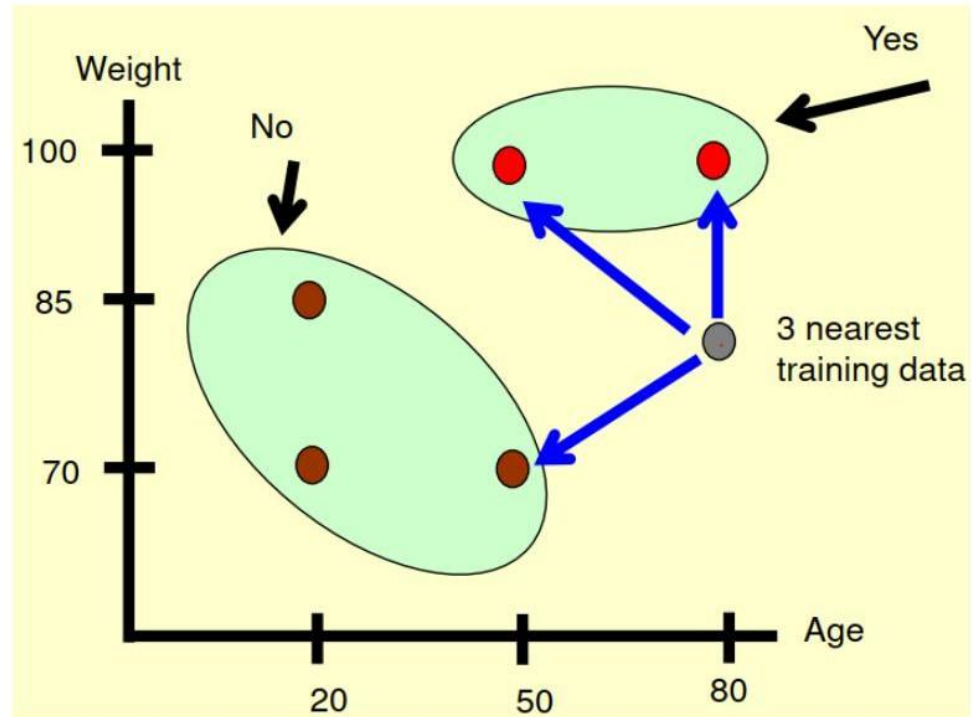
$$\begin{aligned} &= \sqrt{(80 - 80)^2 + (100 - 85)^2} \\ &= \sqrt{15^2} = 15 \end{aligned}$$



Klasifikasi dengan 1-NN

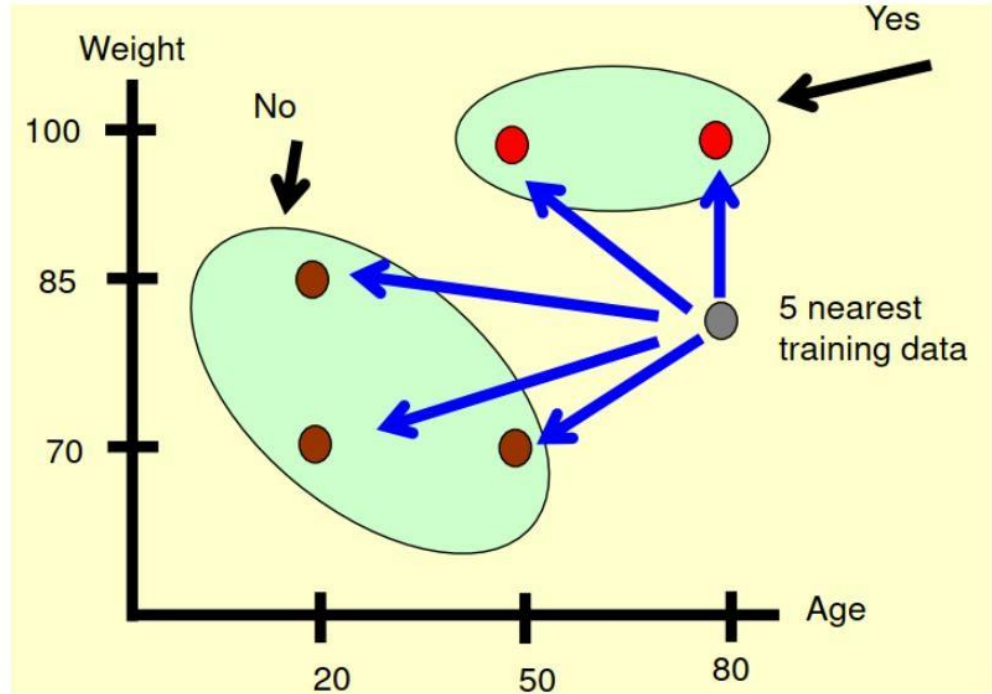


Contoh, $k=3$



Hasil Prediksi: Yes

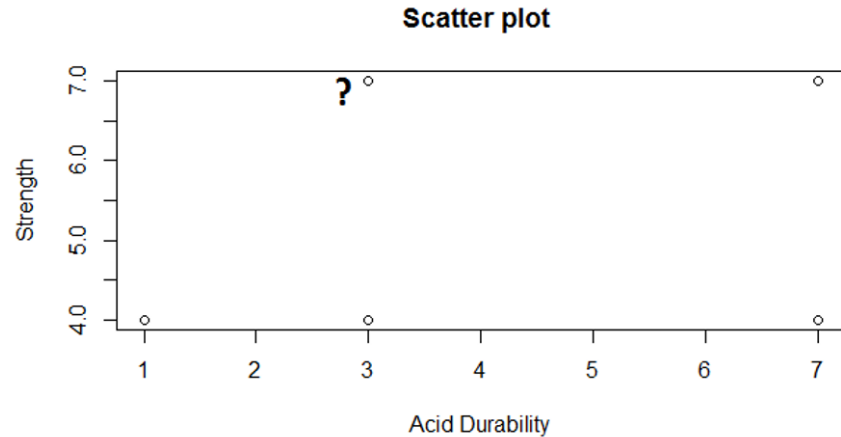
Contoh, $k=5$



Hasil Prediksi: No

Contoh kasus

- Terdapat 4 data latih (P1, P2, P3, P4). Data memiliki dua atribut (x1 dan x2)
- Data latih terdiri dari dua kelas (kelas BAD dan kelas GOOD)
- Diperlukan klasifikasi untuk menentukan kelas dari data uji (P5).
- X1: daya tahan terhadap asam
- X2: kekuatan kertas



Points	X1(Acid Durability)	X2(Strength)	Y(Classification)
P1	7	7	BAD
P2	7	4	BAD
P3	3	4	GOOD
P4	1	4	GOOD
P5	3	7	?

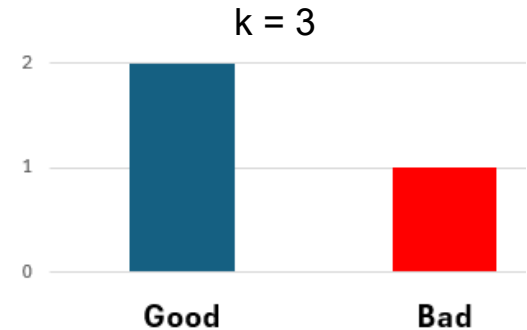
Contoh kasus (lanjutan)

	P1	P2	P3	P4
Euclidean Distance of P5(3,7) from	(7,7)	(7,4)	(3,4)	(1,4)
	$\text{Sqrt}((7-3)^2 + (7-7)^2) = \sqrt{16} = 4$	$\text{Sqrt}((7-3)^2 + (4-7)^2) = \sqrt{25} = 5$	$\text{Sqrt}((3-3)^2 + (4-7)^2) = \sqrt{9} = 3$	$\text{Sqrt}((1-3)^2 + (4-7)^2) = \sqrt{13} = 3.60$
Class	BAD	BAD	GOOD	GOOD

Points	X1(Durability)	X2(Strength)	Y(Classification)
P1	7	7	BAD
P2	7	4	BAD
P3	3	4	GOOD
P4	1	4	GOOD
P5	3	7	GOOD

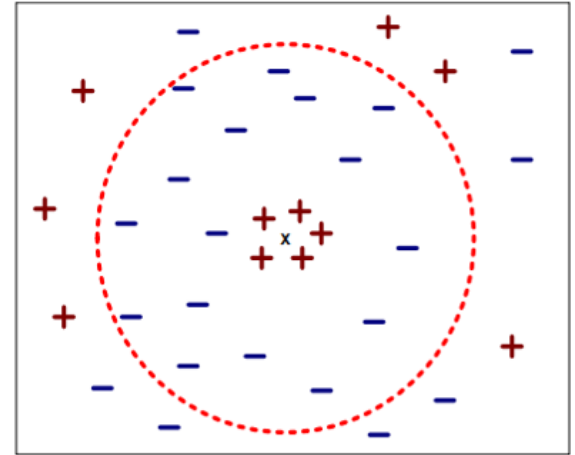


Bagaimana jika k = 4?



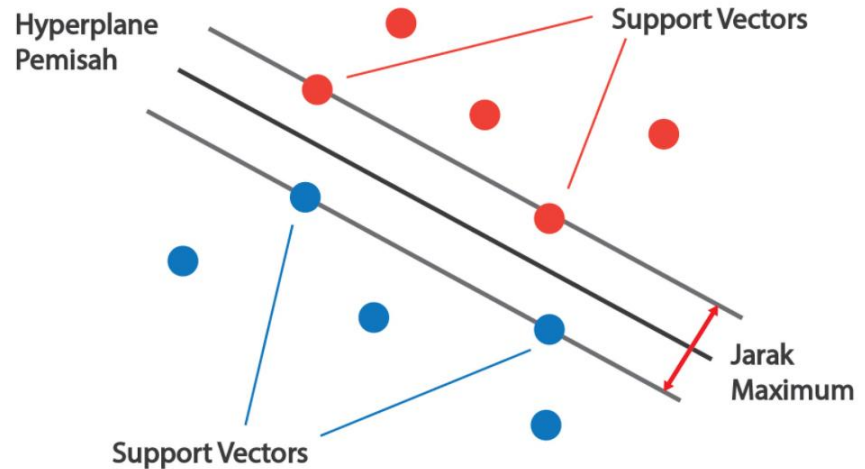
Kelebihan dan Kekurangan k -NN Classifier

- Cocok untuk data numerik
- Mudah dipahami dan diimplementasikan
- k -NN merupakan *lazy learner* (tidak membangun model secara eksplisit)
- Penentuan label/kelas data baru membutuhkan *computational cost* yang cukup tinggi
- Perlu menentukan nilai k yang sesuai
 - Jika k terlalu kecil, sensitif terhadap noise
 - Jika k terlalu besar, nearest neighbor mungkin mencakup data dari kelas lain

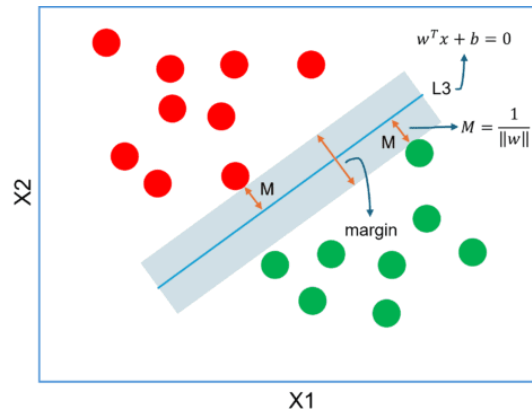
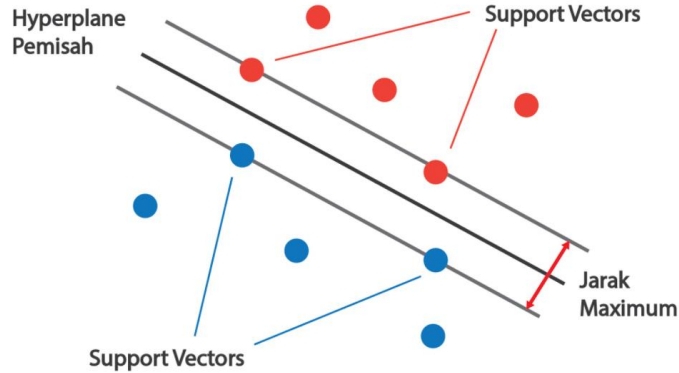


Support Vector Machine

- Ide utama dari SVM adalah untuk menemukan hyperplane terbaik yang memisahkan 2 daerah keputusan dengan baik
- Hyperplane adalah sebuah fungsi yang dapat digunakan sebagai pemisah antar kelas



Support Vector Machine



Algoritma SVM :

- Ide Fitur-fitur dari data dipetakan ke dalam ruang dimensi lebih tinggi menggunakan fungsi kernel (linear, polynomial, radial basis function, sigmoid).
- Mencari hyperplane terbaik yang memisahkan data yang telah dipetakan dalam ruang dimensi tinggi.
- Hyperplane terbaik dapat diperoleh dengan memaksimalkan jarak hyperplane dengan titik data terdekat (Support Vectors)

Kernel pada Support Vector Machine

- Fitur-fitur dari data dipetakan ke dalam ruang dimensi tinggi menggunakan fungsi kernel (linear, polynomial, sigmoid, atau radial basis function)
- Dengan menggunakan fungsi kernel, akan dihasilkan fitur-fitur baru yang akan digunakan untuk mengklasifikasi (tidak lagi menggunakan fitur-fitur lama)

Kernel *Linear*

$$K(x_i, x_j) = x_i^T x_j$$

Kernel *Polynomial*

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \quad \gamma > 0$$

Kernel *Sigmoid*

$$K(x_i, x_j) = \tanh(x_i^T x_j + r)$$

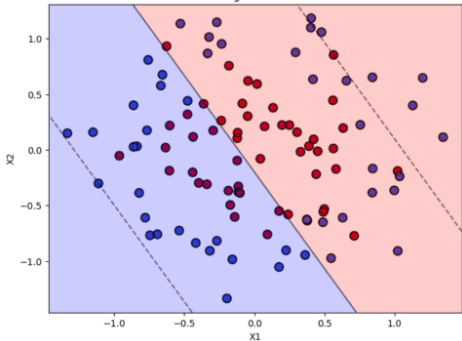
Kernel *Radial Basis Function*

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i^T - x_j\|^2\right), \quad \gamma > 0$$

Kernel pada Support Vector Machine

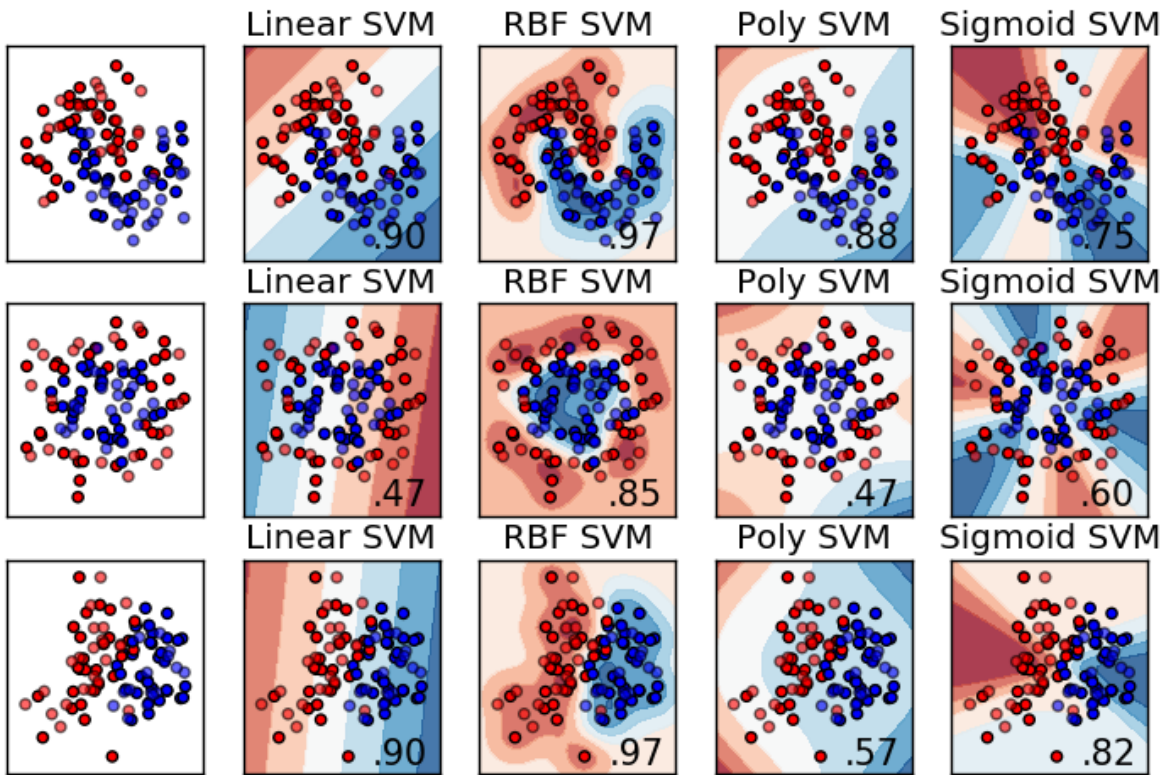
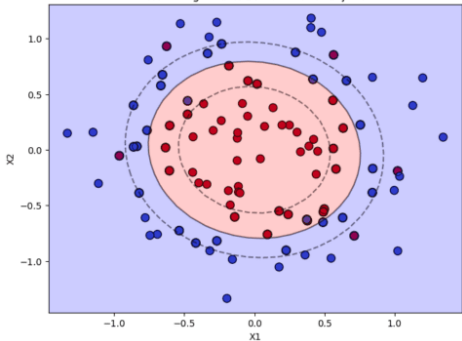
Linier (BAD)

SVM dengan Kernel Linier



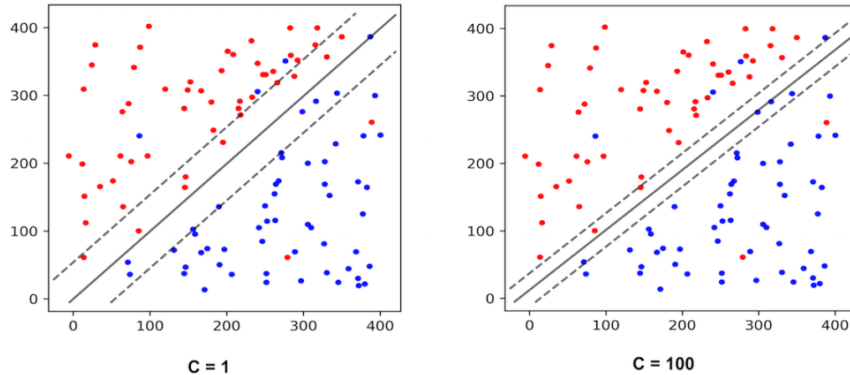
Polynomial (GOOD)

SVM dengan Kernel Polinomial Derajat 2



Parameter C pada Support Vector Machine

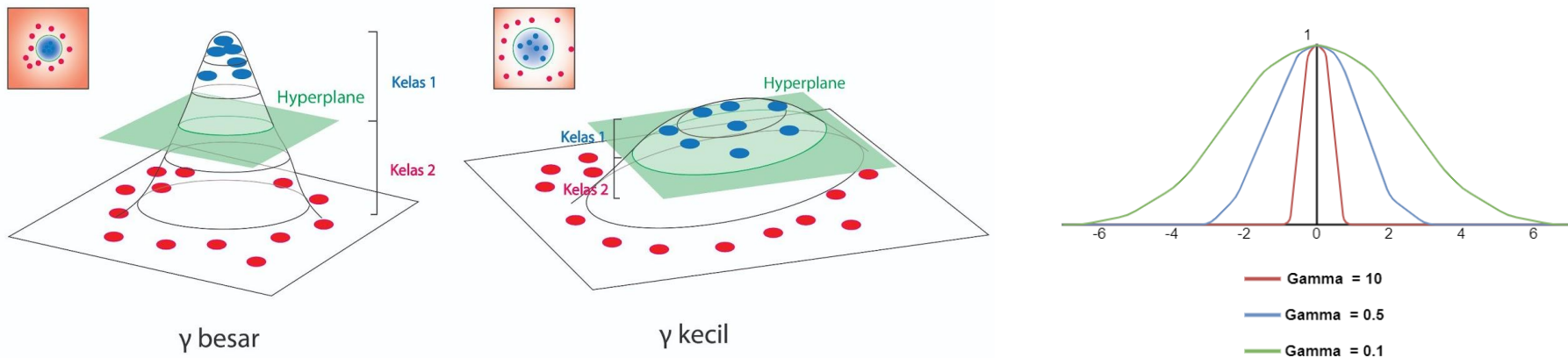
SVM Parameter C



Sumber: <https://learnopencv.com/svm-using-scikit-learn-in-python/>

- Nilai C yang terbaik harus dicari agar SVM terjadi keseimbangan antara jarak semaksimal dan kesalahan klasifikasi seminimal mungkin.
- Nilai C yang tinggi = model akan menerima lebih banyak penalti ketika model gagal mengklasifikasikan data dengan tepat.
- Nilai C yang rendah = model akan mentolerir data yang salah diklasifikasikan.
- Nilai C yang rendah biasanya baik digunakan pada data yang memiliki banyak noise, sebaliknya nilai C yang tinggi biasanya baik digunakan pada data yang memiliki sedikit noise.

Parameter γ pada Support Vector Machine



Nilai γ yang besar akan menghasilkan kelengkungan yang tajam pada ruang dimensi tinggi. Sebaliknya, nilai γ yang kecil akan menghasilkan ruang dimensi tinggi yang lebih landai.

Gamma	Pengaruh data	Boundary	Risiko
Terlalu kecil	Sangat luas	Terlalu halus	Underfitting
Sesuai	Seimbang	Cukup fleksibel	Generalisasi baik
Terlalu besar	Sangat lokal	Sangat tajam/kompleks	Overfitting

Hands On

Parameter	Keterangan	Contoh Nilai
kernel	Memilih kernel untuk SVM	- 'linear' untuk linear kernel - 'poly' untuk polynomial kernel - 'rbf' untuk radial basis function kernel - Nilai default : rbf
C	Parameter regularisasi SVM untuk semua kernel	- Bilangan Float positif (... , 0.1, 0.5, 1.0, 1.5 , ...) - Nilai default = 1.0
gamma	Parameter koefisien kernel untuk 'poly' dan 'rbf'	- Bilangan Float positif (... , 0.001, 0.01, 0.1, 1, ...)
degree	Derajat polynomial untuk kernel 'poly'	- Bilangan Integer (2,3,4,...) - Nilai default : 3

```
# Import Support Vector Machine Classifier dari sklearn
from sklearn.svm import SVC
```

```
svmLinear = SVC(
    kernel = 'linear',
    C = 1
)
```

```
svmPoly = SVC(
    kernel = 'poly',
    C = 1,
    gamma = 0.01,
    degree = 2
)
```

```
svmRBF = SVC(
    kernel = 'rbf',
    C = 1,
    gamma = 0.01
)
```

Kelebihan dan Kekurangan Support Vector Machine

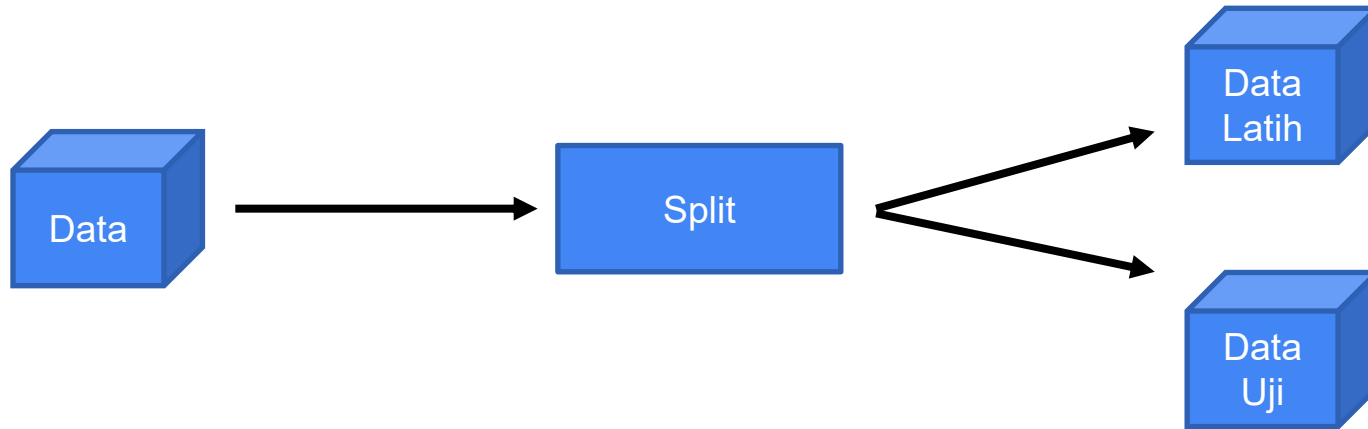
- SVM menggunakan strategi kernel sehingga dapat mengatasi data yang kompleks.
- Outlier dapat diatasi menggunakan soft margin constant C .
- Hyper parameter dan kernel perlu di-tuning untuk mendapatkan akurasi yang diharapkan
- Waktu training yang lebih lama untuk dataset yang lebih besar

Outline

- **Membangun Model Klasifikasi :**
 - Algoritma klasifikasi yang diimplementasi menggunakan library, yaitu : k-NN, Decision Tree, Naïve Bayes, Support Vector Machine dan Boosting
 - Matriks Performansi Klasifikasi
- **Membangun Skenario Pemodelan :**
 - Pembagian data : data latih-uji, *k*-fold cross validation
 - Menentukan Langkah Eksperimen
 - Parameter Evaluasi

Pembagian Data

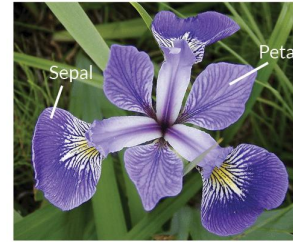
- Data dibagi menjadi 2 bagian :
 - Data Latih (*Training Data*) : untuk mengembangkan model
 - Data Uji (*Testing Data*) : untuk Mengukur performansi model



Pembagian Data

- Dataset Iris (<https://archive.ics.uci.edu/ml/datasets/iris>):
 - Data Latih (Training Data) : 70%
 - Data Uji (Testing Data) : 30%

X_train				y_train
Panjang Sepal	Lebar Sepal	Panjang Petal	Lebar Petal	Kelas
5.1	3.5	1.4	0.2	Iris Setosa
6.3	3.3	6	2.5	Iris Virginica
7	3	4.6	1.4	Iris Versicolour
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
5.8	3.3	6	2.4	Iris Virginica
6.8	3.1	4.5	1.5	Iris Versicolour
4.9	3	1.4	0.2	Iris Setosa
...	...	...	...	...
6.8	3.2	4.4	1.6	Iris Versicolour
X_test				y_test



Iris Versicolor



Iris Setosa

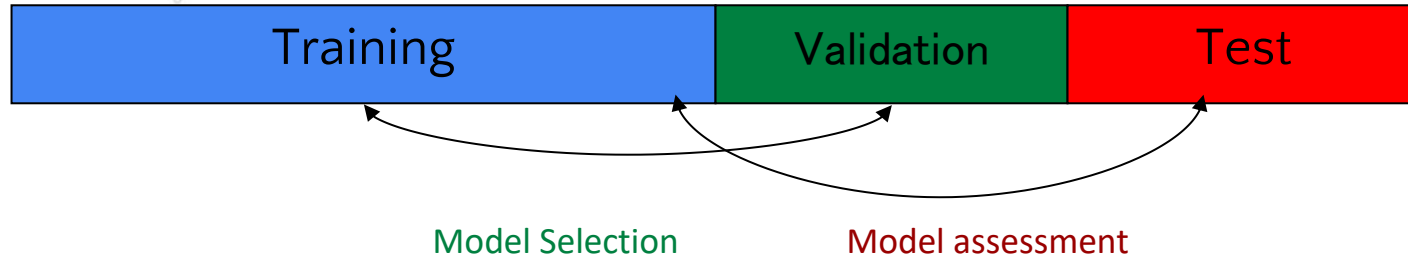


Iris Virginica

Training Data
70%

Testing Data
30%

Training – Validation – Testing Data



- Model Selection : Mengestimasi performa model - model yang berbeda untuk memilih model yang terbaik, yaitu model dengan minimum error
- Model Assessment : Dari model yang terpilih, mengestimasi error untuk data baru (data uji)

k -Fold Cross Validation

- k -Fold Cross Validation digunakan pada dataset dengan jumlah data yang relatif sedikit
- k -Fold Cross Validation dilakukan pada data latih
- Data latih dibagi menjadi k bagian kemudian secara iteratif, 1 bagian menjadi data validasi

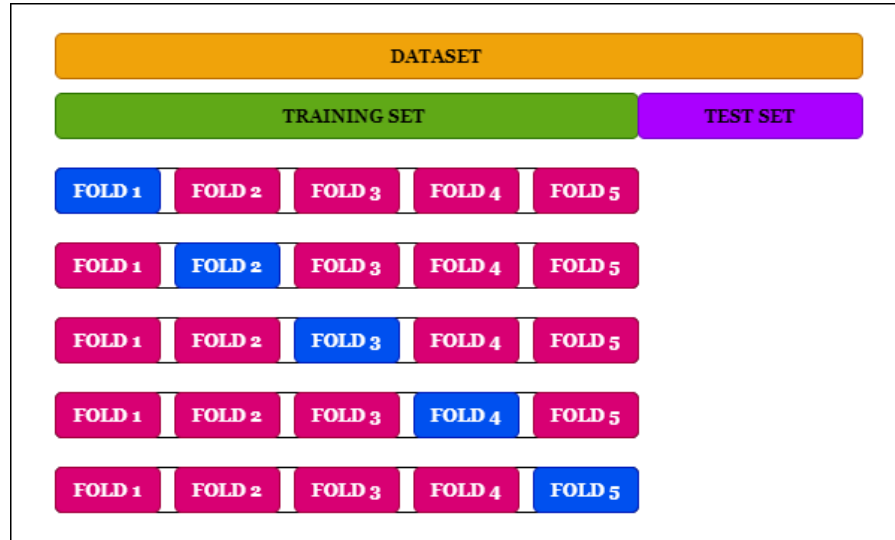
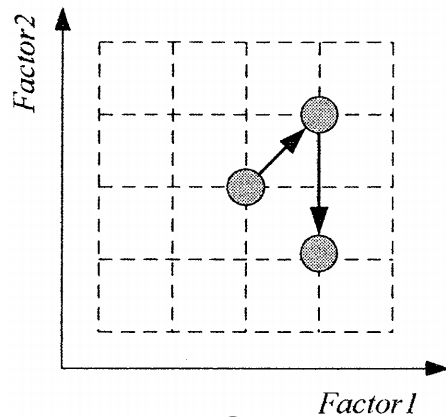


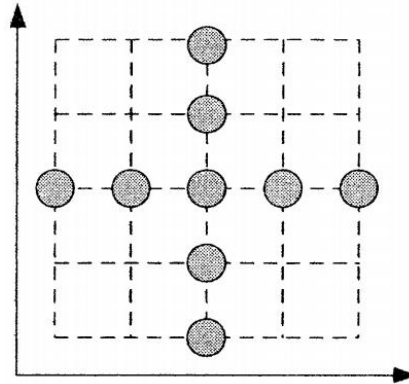
Image Source : <https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538>

Menentukan Langkah Eksperimen

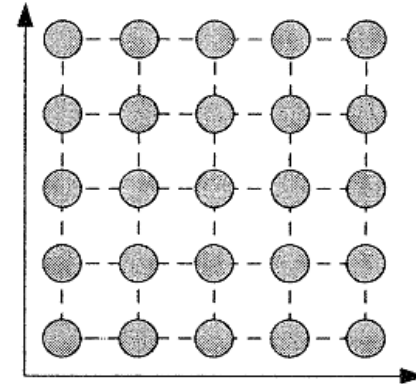
- Setiap metode memiliki parameter tertentu
- Dilakukan eksperimen dengan beberapa variasi parameter
- Parameter yang menghasilkan model performa terbaik akan digunakan selanjutnya
- Beberapa strategi pencarian parameter untuk menghasilkan model terbaik



Best Guess



One Factor at A Time



Grid Search

Parameter Evaluasi

- Klasifikasi
 - Akurasi
 - Presisi
 - Recall/Sensitivity
 - Specificity
 - F1-measure
 - ...
- Regresi
 - MSE (Mean Squared Error)
 - MAPE (Mean Absolute Percentage Error)
 - ...
- Klastering
 - Silhouette Score
 - Davies-Bouldin Index
 - ...

Referensi

- Achmad Sholichin, Algoritma K-Nearest Neighbor (KNN), Univ. Budi Luhur Jakarta, 2022
- Modul Training Data Science, Kominfo, 2021
- CM Bishop, 2006, Pattern Recognition and Machine Learning, Springer
- <https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538>
- <https://learnopencv.com/svm-using-scikit-learn-in-python/>
- <https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30>
- <https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning>
- <https://www.datacamp.com/community/tutorials/adaboost-classifier-python>
- <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>