6 November 2024

BRIN
BADAN RISET
DAN INOVASI NASIONAL

K-Nearest Neighbor Classifier & Naïve Bayes

Dr. Anto Satriyo Nugroho

Kepala Pusat Riset Kecerdasan Artifisial dan Keamanan Siber

Organisasi Riset Elektronika dan Informatika

Badan Riset dan Inovasi Nasional

Tiga hal yang harus diperhatikan

- **Memahami karakteristik Dataset**
  - Sample size → data medis sangat sedikit sample sizenya
  - Balans tidaknya class
  - Atribut apa saja yang tersedia dan bagaimana encoding yang tepat ? Atribut : Nominal, Ordinal, Interval, Ratio
  - Missing data, Noise, Anomali ?
  - Kualitas Data
  - Pengaruh sensor
- **Pemilihan model yang tepat dengan masalah yang diselesaikan**
  - Berbagai metode bisa dipakai : **Naïve Bayes**, **k-Nearest Neighbor Classifier**, Decision Tree, Multilayer Perceptron Neural Network, Deep Learning model, dsb.
  - Prinsip : Occam's Razor
- **Pemilihan metode evaluasi kinerja yang tepat**
  - Evaluasi error (confusion matrix, Error tipe I, Error tipe II)
  - Berbagai metrik : akurasi, precision, recall, arithmetical mean dan geometrical mean, dsb.
  - Receiver Operating Characteristics
  - Skenario pengukuran performa : Hold Out, k-Cross Validation, Leave-One-Out Cross Validation, dsb.

Agenda

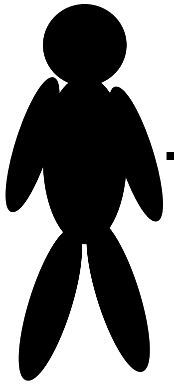
- **Nearest Neighbor Classifier**
- Naïve Bayes
- Eksperimen memakai MNIST Dataset

K-Nearest Neighbor Classifier

If it looks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck

Sumo vs Soccer Player Classification

- Feature : information to discriminate sumo player to soccer player, i.e. body's weight & height



weight & height measurement

Classification

Sumo player

Soccer player

http://en.wikipedia.org/wiki/Image:Asashoryu_fight_Jan08.JPG

Developing a Classification System

- Collecting/Creating Training set & Testing set
  - Training set: data used to design a classifier
  - Testing set: data used to evaluate the model performance
- Testing set should be independent (no intersection samples) from training set

Training Set

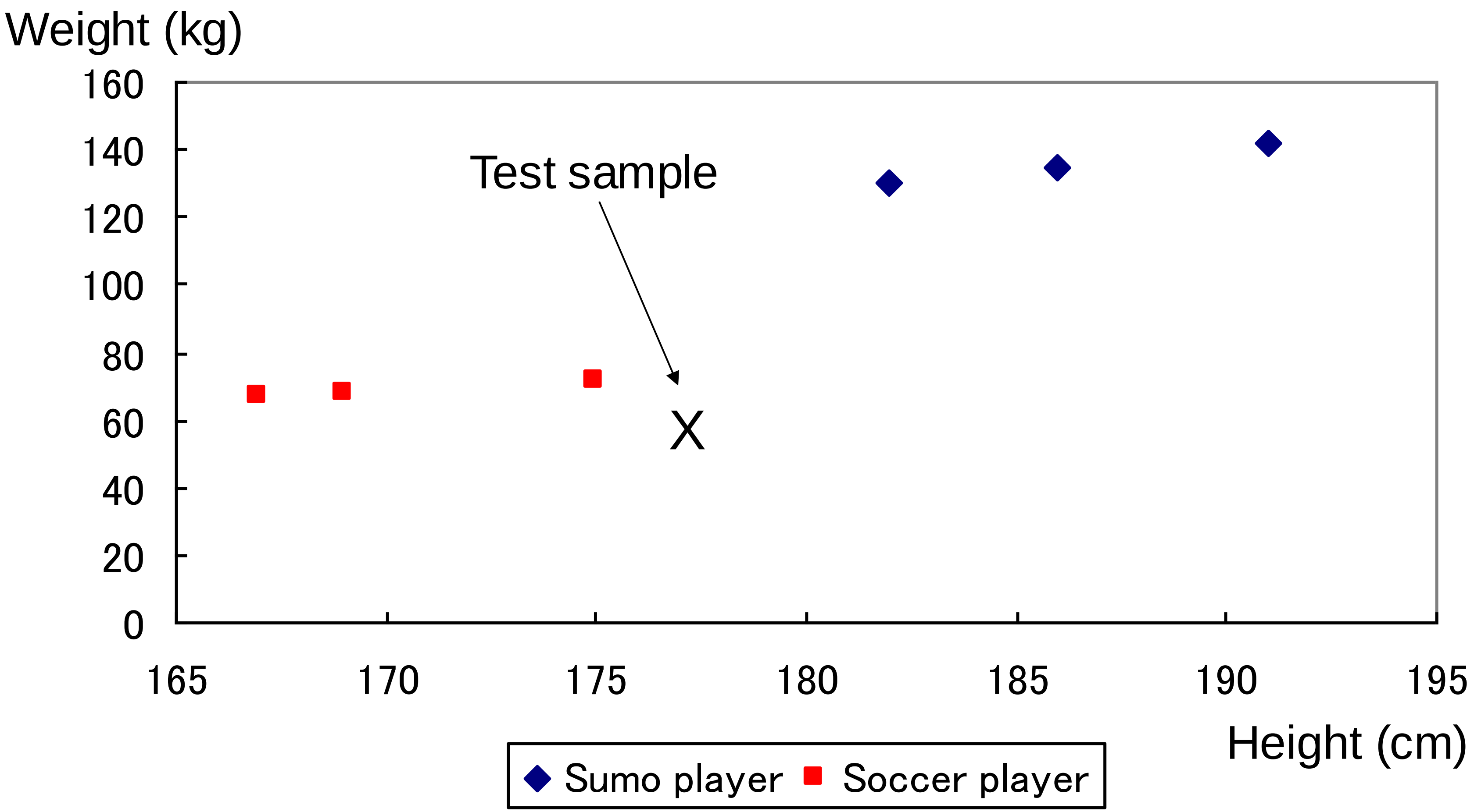
Training set		Soccer			Sumo		
		A1	**A2**	**A3**	**B1**	**B2**	**B3**
Feature	Weight (kg)	67	68	71	142	135	130
	Height (cm)	167	169	175	191	186	182

Task:

Used the training set to classify if a person with weight 60 kg and height 177 cm is a soccer or sumo player

Feature Space

- Feature space is vector space generated by the feature of the data
- The dimension of the data in this problem is 2 (weight & height)



Nearest Neighbor Rule

- Measure the distance between test sample with the whole training samples
- Result: class of the training sample with minimum distance is used to label the testing sample
- Euclidean distance between two vectors with dimensionality *d*

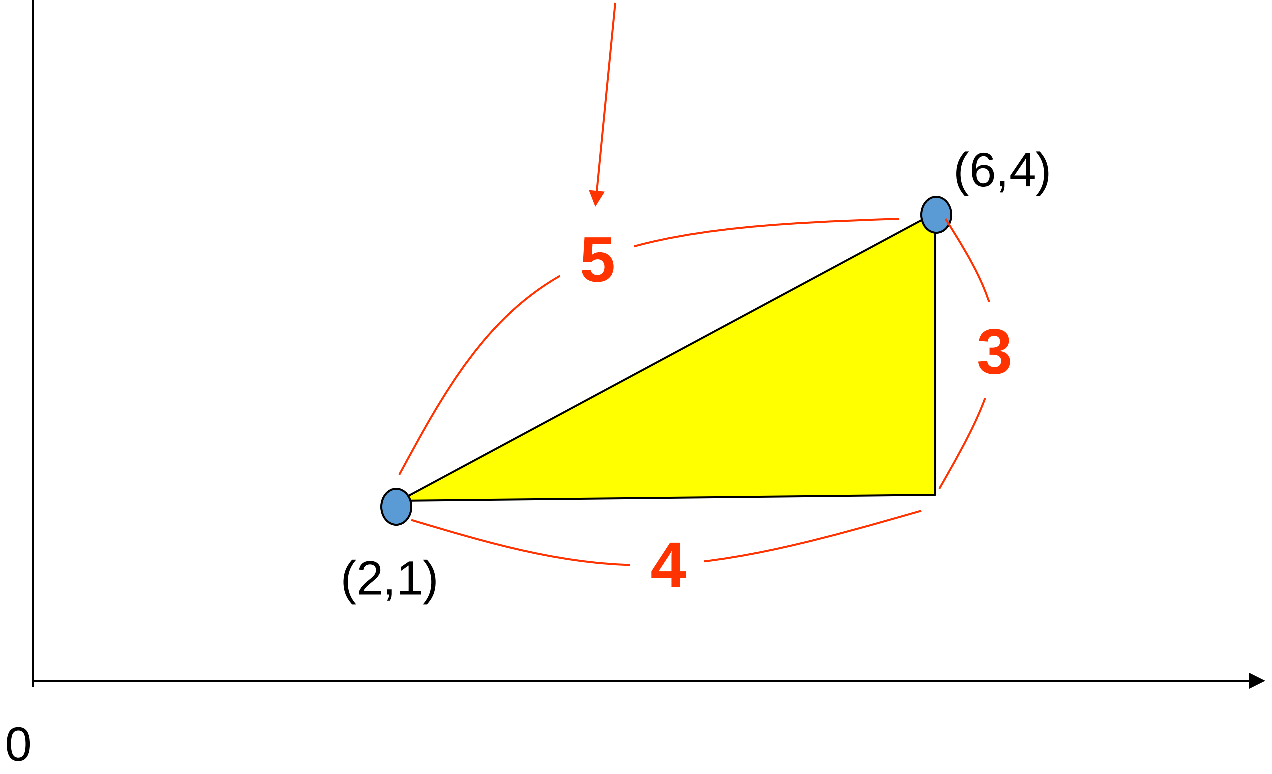
$$D(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

$$D(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

- Euclidean distance for TWO dimensional vector

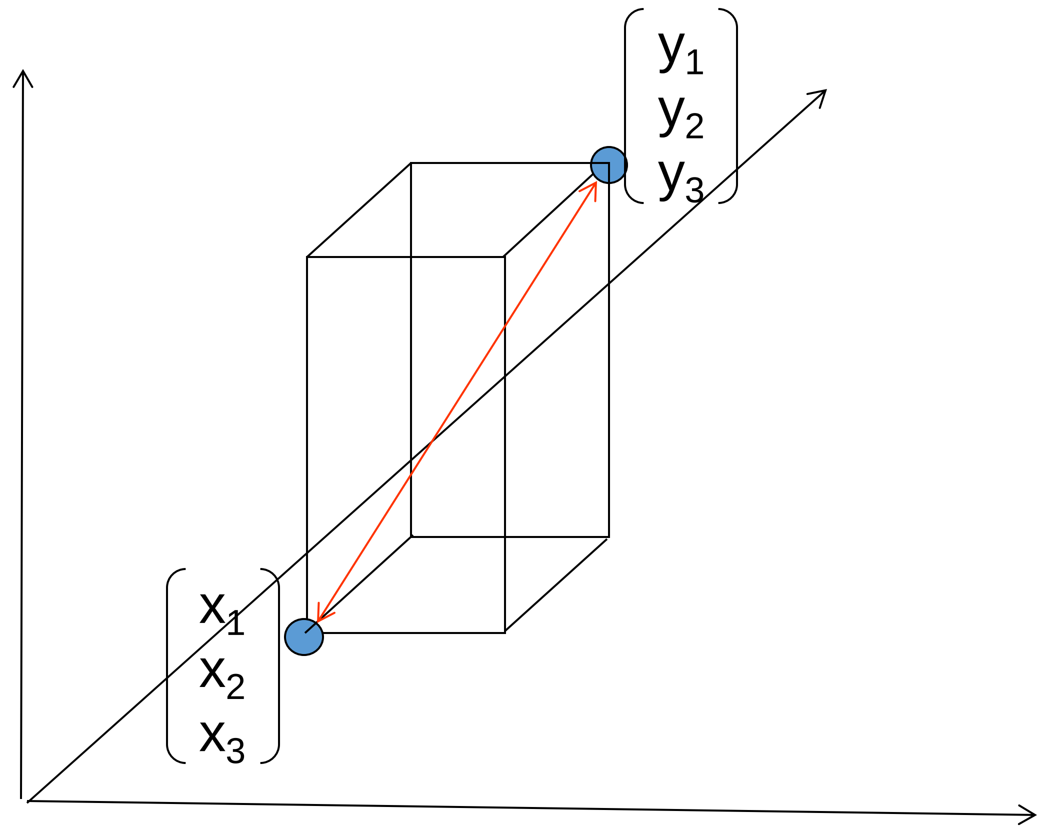
Euclidean Distance in 2D feature space

$$\sqrt{(6-2)^2+(4-1)^2}=\sqrt{4^2+3^2}=\sqrt{25}=5$$



$$D(\vec{x},\vec{y})=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2}$$

Euclidean Distance in 3D feature space



$$D(\vec{x},\vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

Training Set

Training set		Soccer			Sumo		
		A1	A2	A3	B1	B2	B3
Feature	Weight (kg)	67	68	71	142	135	130
	Height (cm)	167	169	175	191	186	182
Distance with testing sample (weight:60 kg height:177cm)		12.1	11.3	11.2	83.2	75.5	70.2
Order		3	2	1	6	5	4

Result: Class of test sample is “soccer player”

K-Nearest Neighbor

- k-Nearest Neighbor is generalization of Nearest Neighbor rule by choosing the nearest k samples (instead of 1) from the training set, then taking majority vote of their class
- k should be ODD number to avoid ambiguity
- Nearest neighbor:

  k-Nearest Neighbor with k=1

Training set	Soccer			Sumo		
	A1	A2	A3	B1	B2	B3
Order	3	2	1	6	5	4

- k=1 → {A3} →{soccer} →result: soccer
- k=3 → {A3,A2,A1} →{soccer,soccer,soccer} →result: soccer
- k=5 → {A3,A2,A1,B3,B2} →{soccer,soccer,soccer,sumo,sumo} →result: soccer

Mathematical Formulation of 1-NN

Training set which consists of *n* samples and *p* class is denoted as follows:

$$(\boldsymbol{x}_1, \theta_1), (\boldsymbol{x}_2, \theta_2), \cdots, (\boldsymbol{x}_n, \theta_n)$$

$$\theta_p \in \{\omega_1, \omega_2, \cdots, \omega_c\} \qquad (p = 1, \cdots, n)$$

1-Nearest Neighbor classification rules is defined as :

$$\min_{p=1,\cdots,n} \{D(\boldsymbol{x}, \boldsymbol{x}_p)\} = D(\boldsymbol{x}, \boldsymbol{x}_k) \quad \Longrightarrow \quad \boldsymbol{x} \in \theta_k$$

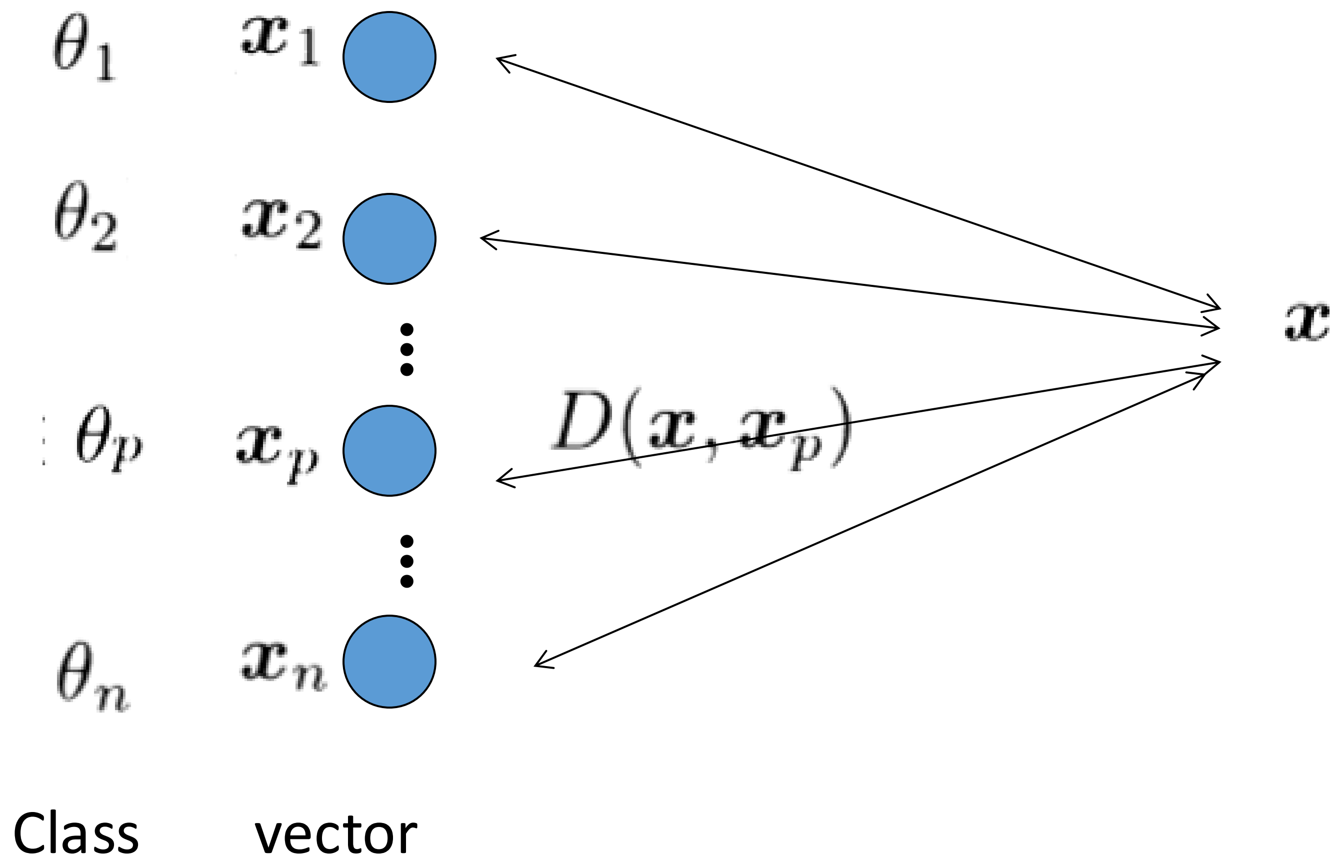
in which

$$\boldsymbol{x}_k \in \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$$

$$\theta_k \in \{\theta_1, \theta_2, \cdots, \theta_n\}$$

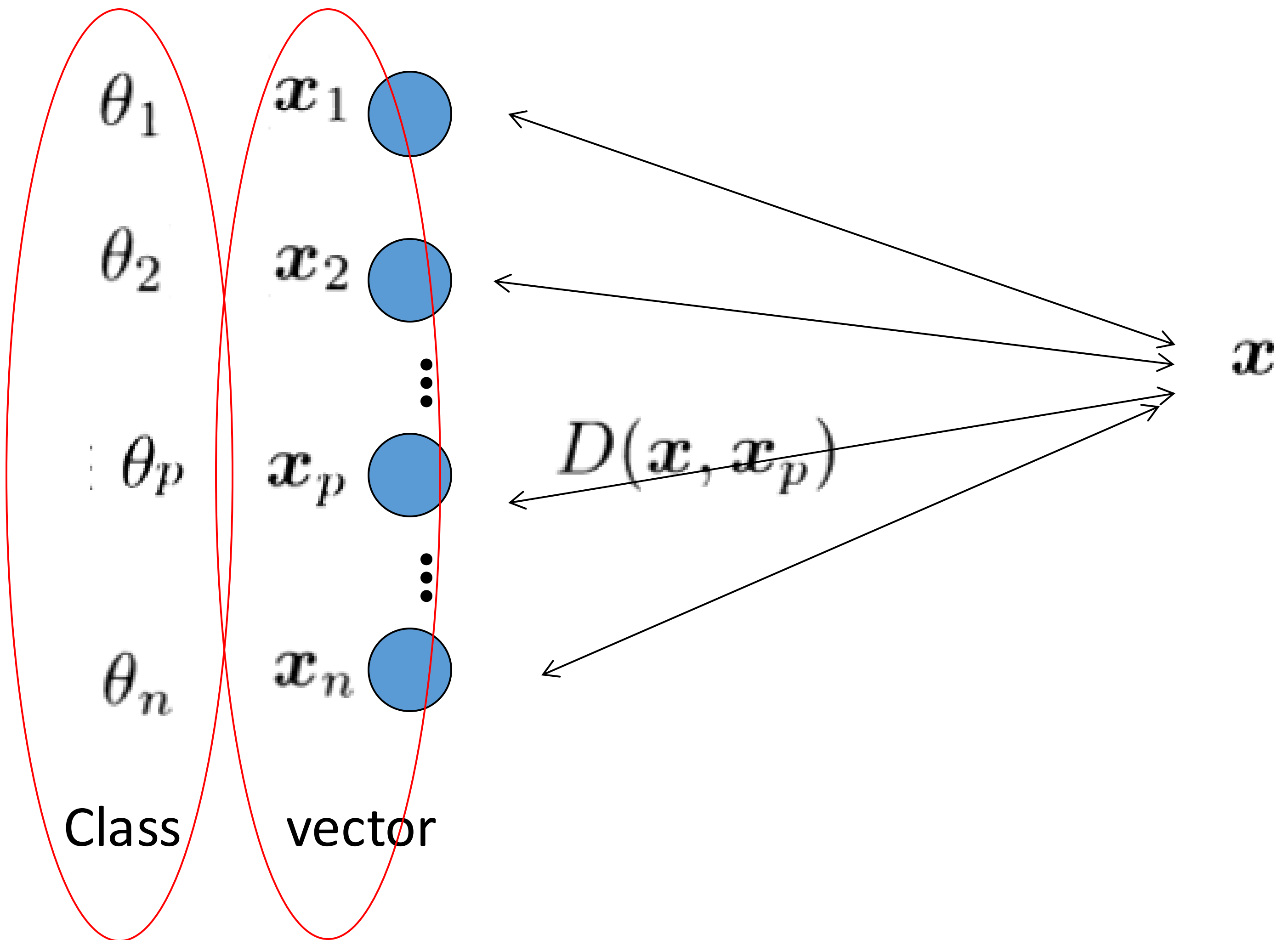
Mathematical Formulation of 1-NN

$$\min_{p=1,\cdots,n} \{D(\boldsymbol{x}, \boldsymbol{x}_p)\} = D(\boldsymbol{x}, \boldsymbol{x}_k) \quad \Longrightarrow \quad \boldsymbol{x} \in \theta_k$$



Mathematical Formulation of 1-NN

$$\min_{p=1,\cdots,n}\{D(\boldsymbol{x},\boldsymbol{x}_p)\} = D(\boldsymbol{x},\boldsymbol{x}_k) \quad \Longrightarrow \quad \boldsymbol{x} \in \theta_k$$



Agenda

- Nearest Neighbor Classifier
- **Naïve Bayes**
- Eksperimen memakai MNIST Dataset

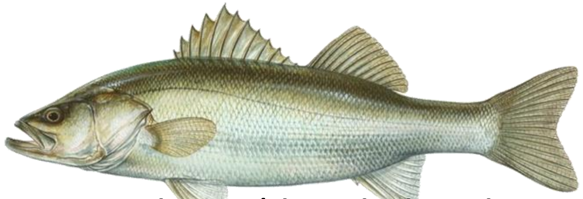
Naïve Bayes Classifier

Apakah Naïve Bayes ?

- Probabilistic classifer
- Generative learning algorithm : mencari model distribusi dari input suatu class
- Tidak mencari, fitur mana yang paling penting untuk membedakan satu class dengan yang lain

Introduction

- The sea bass/salmon example
  - State of nature, prior



sea bass (ikan kakap laut Asia)

    - State of nature is a random variable
    - The catch of salmon and sea bass is equiprobable

salmon

      - $P(\omega_1) = P(\omega_2)$ (uniform priors)
      - $P(\omega_1) + P(\omega_2) = 1$ (exclusivity and exhaustivity)

Decision Rule

- Decision rule with only the prior information
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$ otherwise decide ω_2

- Use of the class –conditional information

- $P(x \mid \omega_1)$ and $P(x \mid \omega_2)$ describe the difference in lightness between populations of sea and salmon

Probability Density

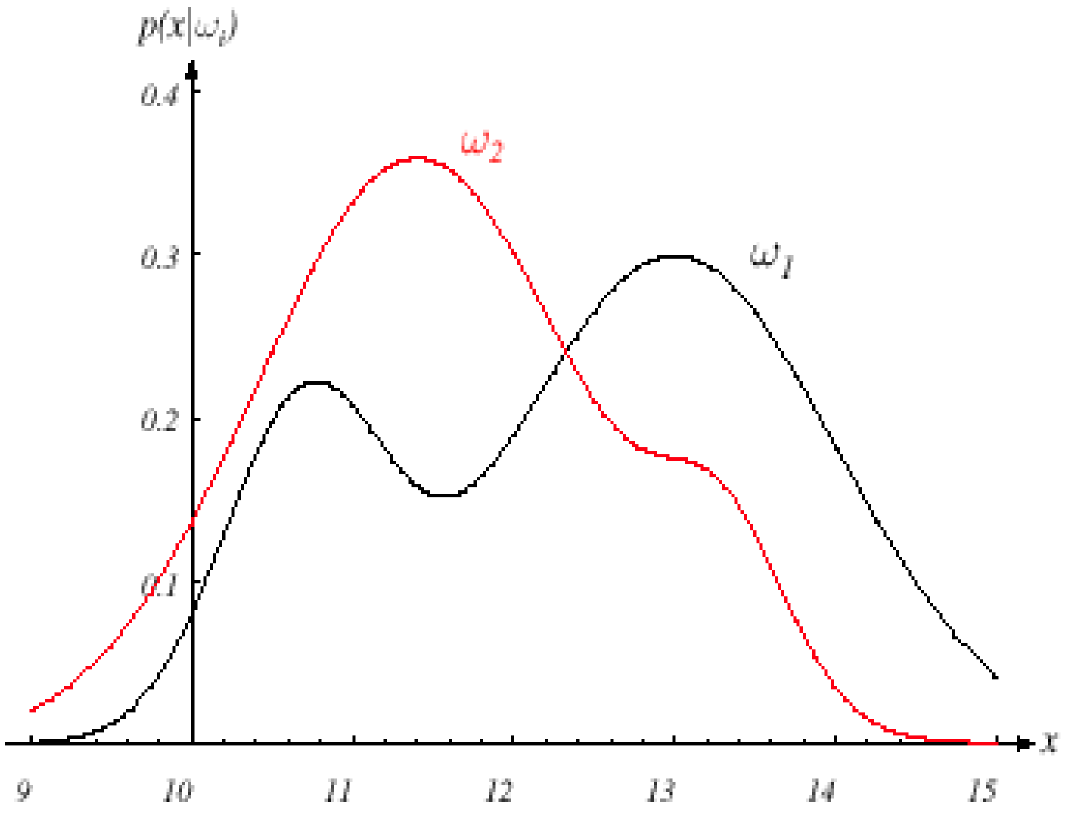


FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i. If x represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- Posterior, likelihood, evidence
  - $P(\omega_j \mid x) = P(x \mid \omega_j) \,.\, P(\omega_j) / P(x)$
  - Where in case of two categories

$$P(x) = \sum_{j=1}^{j=2} P(x \mid \omega_j) P(\omega_j)$$

  - Posterior = (Likelihood x Prior) / Evidence

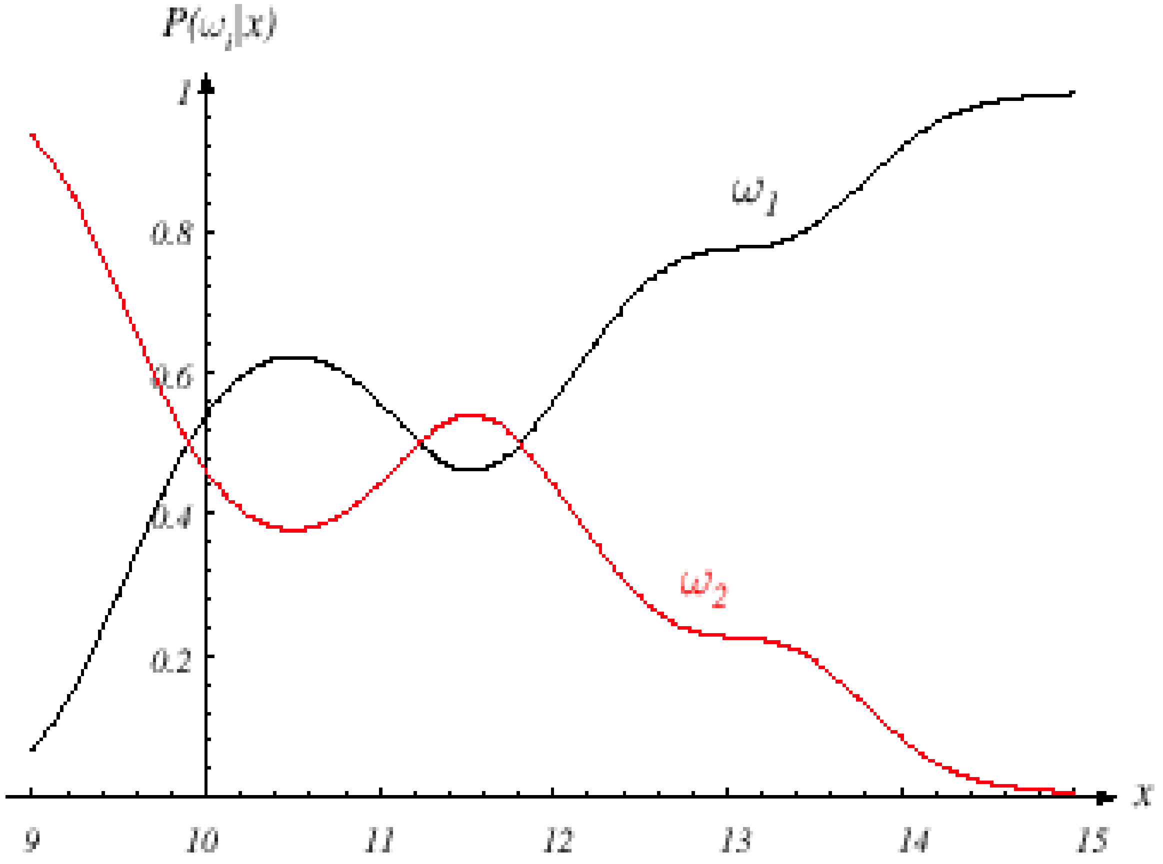


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Error Probability

- Decision given the posterior probabilities

  X is an observation for which:

  if $P(\omega_1 \mid x) > P(\omega_2 \mid x)$ $\Rightarrow$ True state of nature = ω_1
  if $P(\omega_1 \mid x) < P(\omega_2 \mid x)$ $\Rightarrow$ True state of nature = ω_2

  Therefore:

  whenever we observe a particular x, the probability of error is :

  $$P(error \mid x) = P(\omega_1 \mid x) \text{ if we decide } \omega_2$$
  $$P(error \mid x) = P(\omega_2 \mid x) \text{ if we decide } \omega_1$$

- Minimizing the probability of error

- Decide ω_1 *if* $P(\omega_1 \mid x) > P(\omega_2 \mid x)$; otherwise decide ω_2

  Therefore:

$$P(error \mid x) = min\ [P(\omega_1 \mid x), P(\omega_2 \mid x)]$$

(Bayes decision)

Characteristics of Naïve Bayes Classifier

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)

Contoh Soal

- Eksperimen memakai Iris Dataset
- Eksperimen memakai data sintetik

Example with Iris Dataset

- Training set:
  - Iris Setosa (ω_1): 25 samples (first half of the original dataset)
  - Iris Versicolor (ω_2): 25 samples (first half of the original dataset)
  - Iris Virginica (ω_3): 25 samples (first half of the original dataset)
- Testing set
  - Iris Setosa (ω_1): 25 samples (second half of the original dataset)
  - Iris Versicolor (ω_2): 25 samples (second half of the original dataset)
  - Iris Virginica (ω_3): 25 samples (second half of the original dataset)

- Suppose we want to classify a datum from Testing set, with the following characteristics (the actual class is Iris Versicolor):
  - Sepal length: 5.7
  - Sepal width: 2.6
  - Petal length: 3.5
  - Petal width: 1

Solution

1. Calculate the prior probability
2. Calculate the mean & variance of each feature
3. Calculate the likelihood
4. Calculate likelihood multiplication
5. Calculate the evidence
6. Calculate the posterior probability
7. Class decision based on posterior probability

$$\text{POSTERIOR} = \frac{\overset{\text{Step 1}}{\text{PRIOR}} \times \overset{\text{Step 2,3, 4}}{\text{LIKELIHOOD}}}{\underset{\text{Step 5}}{\text{EVIDENCE}}}$$

Step 1 : Prior Probability Calculation

- $P(\omega_1)$= number of ω_1 samples / total samples = 25/75 = 0.33
- $P(\omega_2)$= number of ω_2 samples / total samples = 25/75 = 0.33
- $P(\omega_3)$= number of ω_3 samples / total samples = 25/75 = 0.33

Step 2 : Mean & Variance Calculation

- Iris has continuous attributes, thus to calculate the likelihood we have to calculate the mean (μ) and variance (σ^2) of each class of each attributes.

		Sepal Length	Sepal Width	Petal Length	Petal Width
Iris Setosa	mean (μ)	5.028	3.48	1.46	0.248
	variance (σ^2)	0.160433333	0.13583333	0.03916667	0.0109333
Iris Versicolor	mean (μ)	6.012	2.776	4.312	1.344
	variance (σ^2)	0.300266667	0.1244	0.19693333	0.0425667
Iris Virginica	mean (μ)	6.576	2.928	5.64	2.044
	variance (σ^2)	0.5244	0.13043333	0.4175	0.0650667

Step 3 : Likelihood Calculation

- Suppose we want to classifiy a datum from Testing set, with the following characteristics (the actual class is Iris Versicolor):

  Sepal length: 5.7 Sepal width: 2.6 Petal length: 3.5   Petal width: 1

ω_1: Iris Setosa ω_2: Iris versicolor
ω_3: Iris Virginica

$$P(A_i \mid \omega_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{\frac{-(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

A_1:Sepal length=5.7,
A_2:sepal width=2.6
A_3:petal length=3.5
A_4:petal width=1

Hati-hati menulis formula matematika di Excel !

$$P(A_i \mid \omega_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{\frac{-(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

Ai = 5.7	Mean : 5.028 Variance : 0.16					
Perhitungan likelihood						
P(A_i = 5.7 | Iris Setosa)		=	1/SQRT(2*PI()*0.16)	*	EXP(-(5.7-5.028)*(5.7-5.028) / (2*0.16)	)
		=	0.24321			

Penting

$-x^2$ jangan ditulis –x^2 di Excel, karena akan ditafsirkan (–x) *(– x)

Tulislah $-x^2$ dengan cara: –x*x

Step 3 : Likelihood Calculation

P(sepal length=5.7 | Iris Setosa) = 0.241763
P(sepal width=2.6 | Iris Setosa) = 0.0625788
P(petal length=3.5 | Iris Setosa) = 1.7052 e-23 = 0
P(petal width=1 | Iris Setosa) = 2.23877 e-11

P(sepal length=5.7 | Iris Versicolor) = 0.619097
P(sepal width=2.6 | Iris Versicolor) = 0.998687
P(petal length=3.5 | Iris Versicolor) = 0.16855
P(petal width=1 | Iris Versicolor) = 0.481618

P(sepal length=5.7 | Iris Virginica) = 0.265044
P(sepal width=2.6 | Iris Virginica) = 0.731322
P(petal length=3.5 | Iris Virginica) = 0.00256255
P(petal width=1 | Iris Virginica) = 0.000360401

Step 4 : Calculate Likelihood Multiplication

- P(sepal length=5.7 | Iris Setosa)*P(sepal width=2.6 | Iris Setosa)*P(petal length=3.5 | Iris Setosa)*P(petal width=1 | Iris Setosa) = 0

- P(sepal length=5.7 | Iris Versicolor)*P(sepal width=2.6 | Iris Versicolor)*P(petal length=3.5 | Iris Versicolor)*P(petal width=1 | Iris Versicolor) = 0.05019

- P(sepal length=5.7 | Iris Virginica)*P(sepal width=2.6 | Iris Virginica)*P(petal length=3.5 | IrisVirginica)*P(petal width=1 | IrisVirginica) = 1.790×10^{-7}

Step 5 : Evidence Calculation

- Evidence = P(class ”Iris Setosa”) x p(5.7, 2.6, 3.5 | class ”Iris Setosa") +
  P(class ”Versicolor”) x p(5.7, 2.6, 3.5 | class ”Iris Versicolor") +
  P(class ”Iris Virginica”) x p(5.7, 2.6, 3.5 | class ”Iris Virginica")

  $= (0.33 \times 0) + (0.33 \times 0.0502) + (0.33 \times 1.790 \times 10^{-7})$

  $= 0.0167$

Step 6 : Posterior Probability Calculation

- Posterior (Iris Setosa | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 0/evidence = 0/0.00557 = 0

- Posterior (Iris Versicolor | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 0.0167/evidence = 0.0167/0.0167 = 1.0

- Posterior (Iris Virginica | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 1.790×10^{-7} /evidence = 1.790×10^{-7} /0.0167= 0

Step 7 : Decision based on Posterior Probability

- Posterior (Iris Setosa | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 0
- Posterior (Iris Versicolor | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 1.0
- Posterior (Iris Virginica | Sepal length: 5.7, Sepal width: 2.6, Petal length: 3.5, Petal width:1) = 0

- From the three posterior values above, the second one has the biggest value.
  Thus the class for datum with sepal length: 5.7   Sepal width:2.6 Petal length:3.5 Petal width:1   is **Iris Versicolor**

$$\boxed{\text{POSTERIOR} = \frac{\text{PRIOR} \times \text{LIKELIHOOD}}{\text{EVIDENCE}}}$$

Contoh Soal

- Eksperimen memakai Iris Dataset
- Eksperimen memakai data sintetik

Soal

- Diketahui data training (data pelatihan) sebagai berikut

feature (attribute) No.								class
1	2	3	4	5	6	7	8	
0.00165	0.00496	0.01495	0.135	0.00774	0.02321	26.822	-6.647379	1
0.00349	0.01406	0.02719	0.255	9.91483	0.0445	21.028	-4.649573	1
0.00398	0.01193	0.03209	0.307	0.01789	0.05368	20.767	-4.333543	1
0.00157	0.00472	0.01279	0.129	0.00617	0.01851	25.02	-4.913137	1
0.00241	0.00723	0.02008	0.221	0.00849	0.02548	24.743	-6.186128	1
0.00165	0.00496	0.01642	0.154	0.00728	0.02184	24.889	-5.660217	2
0.00232	0.00696	0.04137	0.37	0.02021	0.06062	19.493	-5.18696	2
0.0025	0.0075	0.01966	0.186	0.00889	0.02666	25.908	-6.18359	2
0.0025	0.00749	0.0919	0.198	0.00883	0.0265	25.119	-6.27169	2

- Tentukan class data testing, apakah class “1” atau “2” memakai metode Naïve Bayes

feature (attribute) No.								class
1	2	3	4	5	6	7	8	
0.0017	0.006	0.04932	0.2	0.02229	0.01614	26.369	-5.892061	?

7 Step perhitungan klasifikasi (Naïve Bayes)

1. Menghitung prior probability
2. Menghitung mean dan variance (σ^2) untuk tiap fitur masing-masing kelas
3. Menghitung likelihood
4. Menghitung perkalian likelihood
5. Menghitung evidence
6. Menghitung posterior probability
7. Class Decision (menentukan kelas)

Step 1 : menghitung Prior Probability

- P(class “1”) : 0.556
- P(class “2”) : 0.444

Step 2 : menghitung mean & variance tiap kelas

Class		feature (attribute) No.							
		1	2	3	4	5	6	7	8
1	mean	0.00262	0.00858	0.02142	0.2094	1.991024	0.033076	23.676	-5.345952
	variance (σ^2)	1.1735E-06	1.77699E-05	6.6319E-05	0.0059348	19.6208653	0.00023056	7.0790815	1.02421312
2	mean	0.0022425	0.0067275	0.0423375	0.227	0.0113025	0.033905	23.85225	-5.8256143
	variance (σ^2)	1.63225E-07	1.45209E-06	0.00121445	0.009433333	3.5819E-05	0.00032219	8.63622492	0.2541219

Step 3 : menghitung likelihood

Class	likelihood feature (attribute) No.							
	1	2	3	4	5	6	7	8
1	256.77	78.47	0.14	5.14	0.08	14.11	0.09	0.34
2	400.85	275.91	11.22	3.95	12.36	13.62	0.09	0.78

Contoh :

$$p(\ 0.0017 \mid \text{fitur 1 class 1}\ ) = \frac{1}{\text{SQRT}(\ 2 * \text{PI} * \text{fitur 1 var kelas 1}\ )}\ e^{\frac{-(\ 0.0017 - \text{mean 1 kelas 1}\ )^2}{2 * \text{var fitur 1 kelas 1}}}$$

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^{2}}} e^{-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

Step 4 : menghitung perkalian likelihood

p (0.0017, 0.006, ... | class "1") = 256.77 * 78.47 * 0.14 * 5.14 * 0.08 * 14.11 * 0.09 * 0.34 = 505.33

p (0.0017, 0.006, ... | class "2") = 400.85 * 275.91 * 11.22 * 3.95 * 12.36 * 13.62 * 0.09 * 0.78 = 60923229.08

Step 5 : menghitung evidence

Evidence = P(class ”1”) x p (0.0017, 0.006, ... | class "1") + P(class ”2”) x p (0.0017, 0.006, ... | class “2")

= (0.556 * 505.33) + (0.444 * 60923229.08)

= 27077271.44

Step 6 : menghitung Posterior Probability

P(class ”1” | 0.0017, 0.006, ... , -5.89) = 505.33 * 0.556/ 27077271.44 = 1.03681E-05 = 0.0000103681

P(class ”2” | 0.0017, 0.006, ... , -5.89) = 60923229.08 * 0.444/ 27077271.44 = 0.999989632

Posterior = (Likelihood x Prior Probability) / Evidence

Step 7 : Class Decision (menentukan class)

karena Posterior Probability class "2" > Posterior Probability class "1",

maka Testing Datum diklasifikasikan ke class "2"

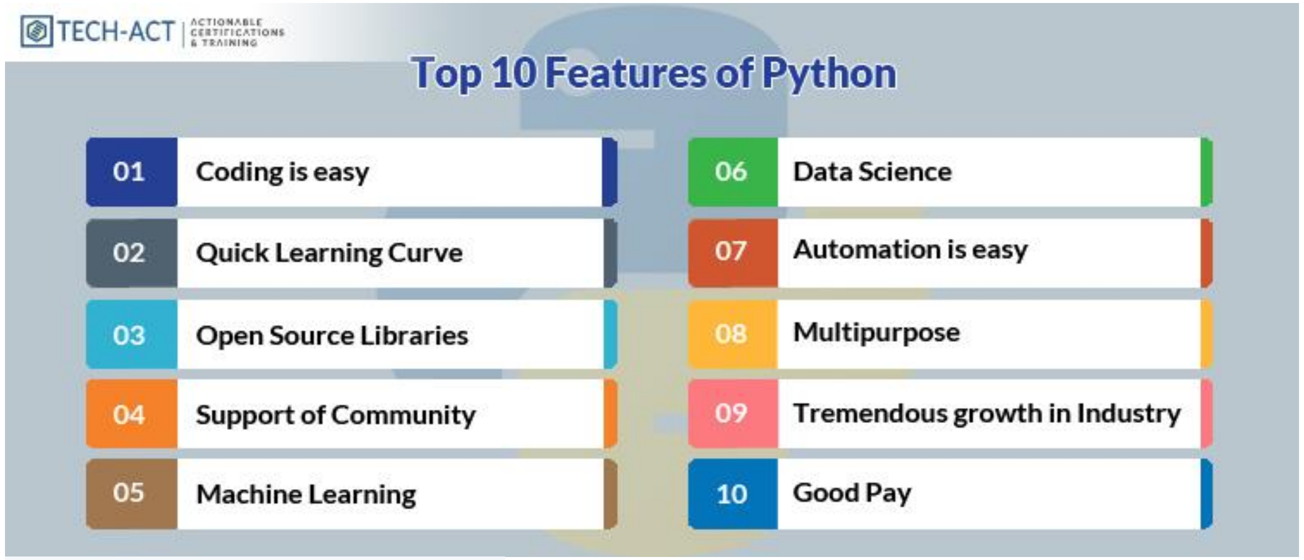
Agenda

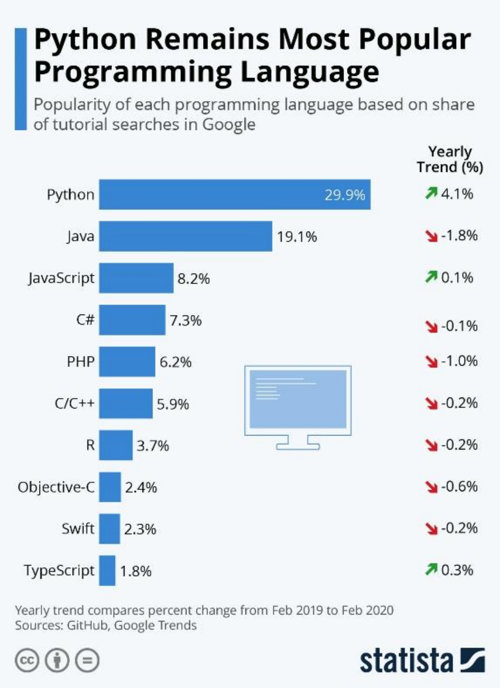
- Nearest Neighbor Classifier
- Naïve Bayes
- **Eksperimen memakai MNIST Dataset**

Implementasi Machine Learning (Naïve Bayes) Memakai Python

Mengapa Memakai Python ?

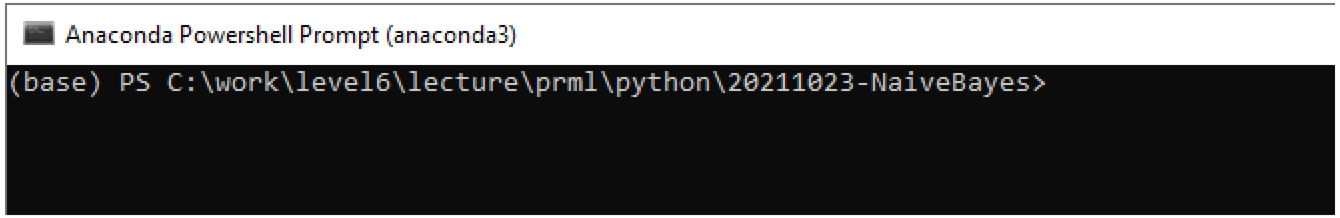
- Alasan mengapa belajar Python (https://rahard.wordpress.com/2018/10/29/mengapa-bahasa-python/)
  - Relatif mudah, interpreted, tidak perlu dicompile
  - Tersedia dalam berbagai sistem operasi (Windows, Linux, OSX, dsb)
  - Tersedia banyak pustaka/library
- Beberapa Link :
  - https://www.youtube.com/watch?v=cew2tMMB8zk (Budi Rahardjo)
  - https://www.youtube.com/watch?v=86tStUuz3B0





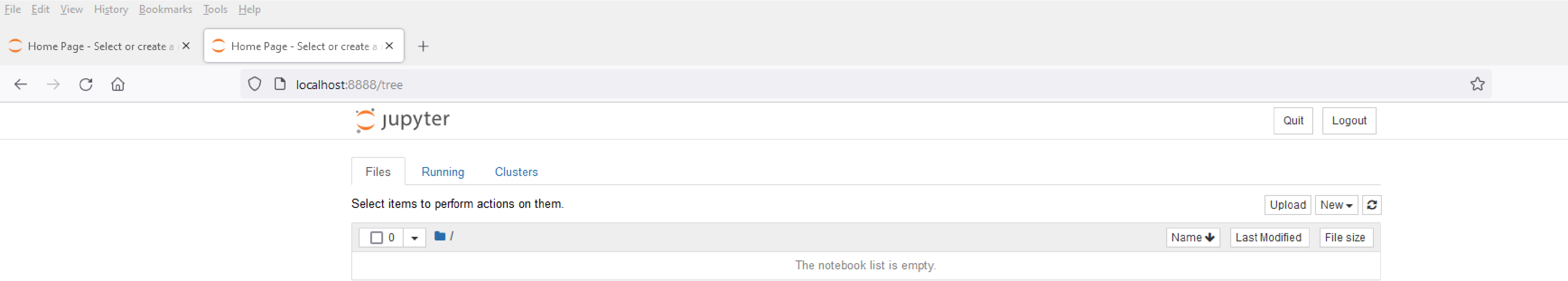
Instalasi Anaconda & Menjalankan Jupyter Notebook

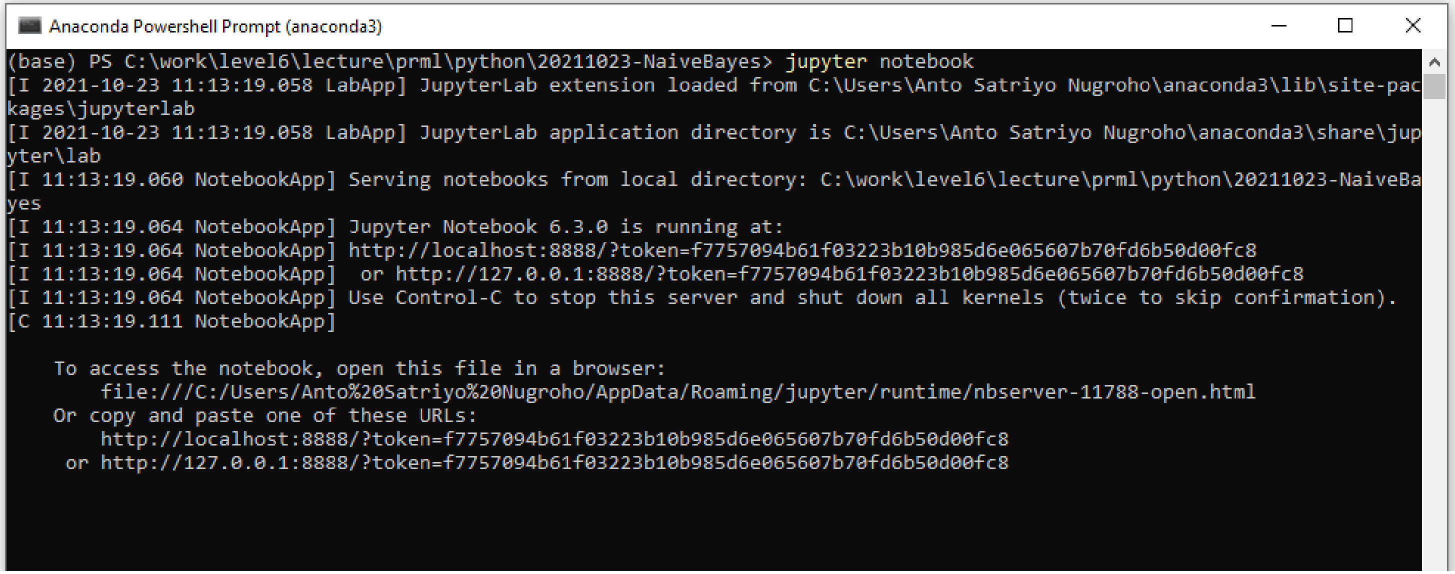
- Install lebih dahulu Anaconda. Anaconda adalah paket distribusi Python dari Continuum Analysis yang berisi paket Python dan beberapa paket tambahan.
- Jalankan Anaconda Powershell Prompt (anaconda3)
- Pindah ke folder yang diinginkan (misalnya C:\work\level6\lecture\prml\python\20211023-NaiveBayes)



- Jalankan jupyter notebook sehingga tampil notebook di layar

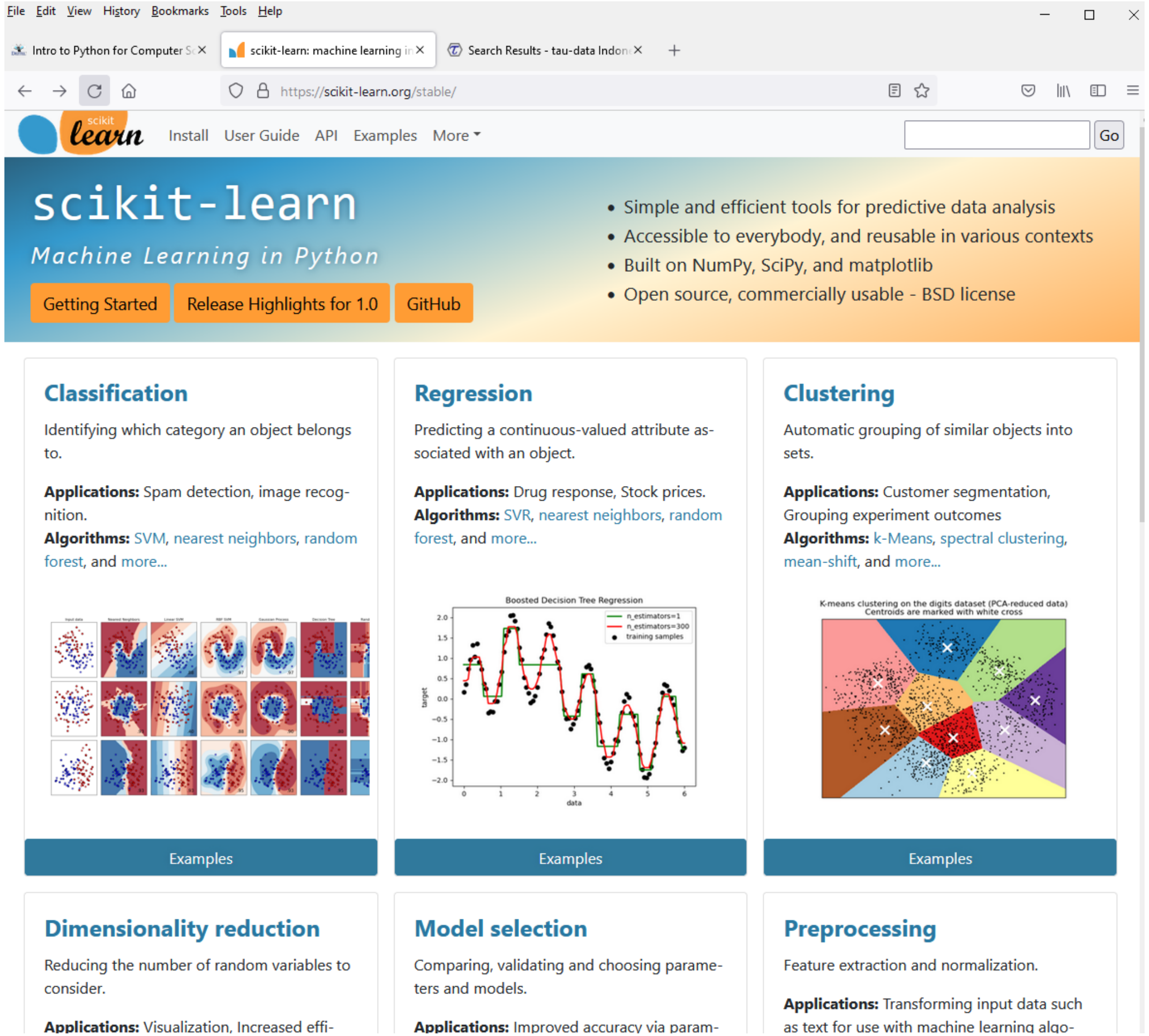
Instalasi Anaconda & Menjalankan Jupyter Notebook





scikit-learn : Machine Learning in Python

- https://scikit-learn.org/stable/
- Toolkit untuk implementasi machine learning.
- Mulai dikembangkan tahun 2007. Saat ini versi terbaru : 1.0 (24 September 2021)
- Scikit-learn telah mengimplementasikan berbagai modul untuk
  - Klasifikasi : Nearest neighbor, Naïve Bayes, Multilayer Perceptron Neural Network (Perceptron), dsb
  - Regresi :
  - Clustering : k-Means, Hierarchical clustering, dsb.
  - Dimensionality reduction
  - Model selection : comparing, validating dan pemilihan parameter & model
  - Preprocessing : Feature extraction dan normalization



Menyiapkan Dataset

- Buatlah data training dan testing memakai MS Excel dan simpanlah pada folder tersebut dengan nama data.xlsx

	A	B	C	D	E	F	G	H	I
1	ftr-1	ftr-2	ftr-3	ftr-4	ftr-5	ftr-6	ftr-7	ftr-8	Class
2	0.00165	0.00496	0.01495	0.135	0.00774	0.02321	26.822	-6.647379	1
3	0.00349	0.01406	0.02719	0.255	9.91483	0.0445	21.028	-4.649573	1
4	0.00398	0.01193	0.03209	0.307	0.01789	0.05368	20.767	-4.333543	1
5	0.00157	0.00472	0.01279	0.129	0.00617	0.01851	25.02	-4.913137	1
6	0.00241	0.00723	0.02008	0.221	0.00849	0.02548	24.743	-6.186128	1
7	0.00165	0.00496	0.01642	0.154	0.00728	0.02184	24.889	-5.660217	2
8	0.00232	0.00696	0.04137	0.37	0.02021	0.06062	19.493	-5.18696	2
9	0.0025	0.0075	0.01966	0.186	0.00889	0.02666	25.908	-6.18359	2
10	0.0025	0.00749	0.0919	0.198	0.00883	0.0265	25.119	-6.27169	2
11	0.0017	0.006	0.04932	0.2	0.02229	0.01614	26.369	-5.892061	2

Data training terdiri dari 9 sampel

Data terakhir dipakai sebagai data testing (1 sampel)

Implementasi Naïve Bayes dalam 18 baris

```
In [44]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import sklearn
         from sklearn.model_selection import train_test_split
         from sklearn.naive_bayes import GaussianNB
         from sklearn.metrics import accuracy_score
```

```
In [47]: dataset = pd.read_excel('data.xlsx')
         X = dataset.iloc[:,:8]
         y = dataset.iloc[:, 8]
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1, shuffle=False, stratify=None)
         classifier = GaussianNB()
         classifier.fit(X_train, y_train)
         y_pred = classifier.predict(X_test)
         ac = accuracy_score(y_test, y_pred)
         print('Hasil prediksi:', y_pred)
         print('Class:', y_test)
         print('Akurasi:',ac)
```

```
Hasil prediksi:  [2]
Class: 9    2
Name: Class, dtype: int64
Akurasi: 1.0
```

Hasil prediksi : class "2"

Class sebenarnya: "2"

Karena hasil prediksi sama dengan class sebenarnya, akurasi = 100%

Tugas

- Buatlah eksperimen memakai data Parkinson (dimensi : 22, class: PD atau Healthy). Gunakan 100 sampel pertama sebagai training set dan sisanya (95 sampel) sebagai testing set.

MNIST Character Recognition

Related Articles

- URL : http://yann.lecun.com/exdb/mnist/
- Articles on my blog
  - https://asnugroho.wordpress.com/2017/11/09/kelas-pattern-recognition-eksperimen-dengan-mnist-handwritten-digit-database/
  - https://asnugroho.wordpress.com/2017/11/10/fashion-mnist-database/
  - https://asnugroho.wordpress.com/2017/11/11/eksperimen-dengan-mnist-digit-fashion-database/
  - https://asnugroho.wordpress.com/2017/11/11/klasifikasi-memakai-naive-bayes-pada-dataset-mnist-handwritten-digit-fashion-mnist/

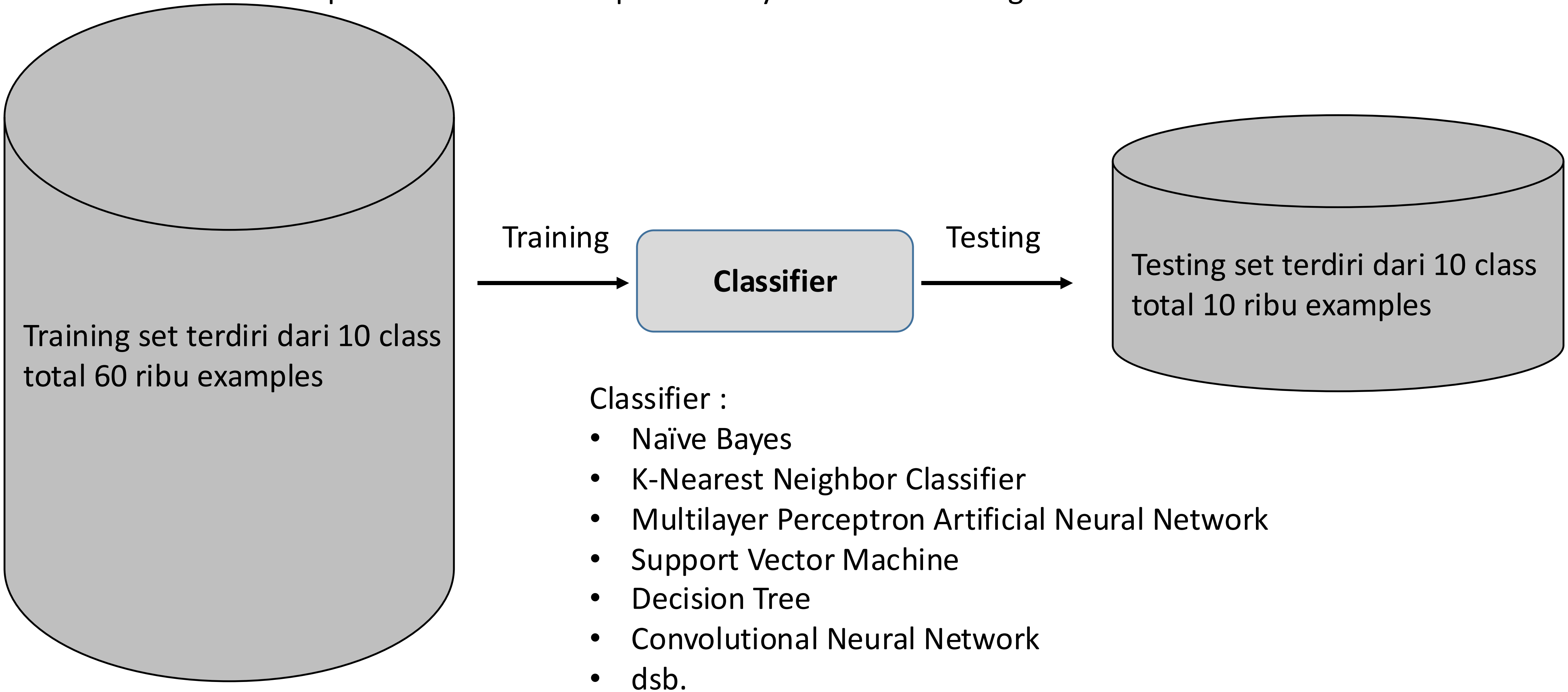
MNIST Dataset

- MNIST : database handwritten digits yang tersedia pada http://yann.lecun.com/exdb/mnist/
- MNIST : Modified National Institute of Standards and Technology database
- Empat data tersedia:
  - train-images-idx3-ubyte.gz: training set images
  - train-labels-idx1-ubyte.gz: training set labels
  - t10k-images-idx3-ubyte.gz: test set images
  - t10k-labels-idx1-ubyte.gz: test set labels
- Training Set : 60,000 examples Test Set : 10,000 examples
- Citra asli dari NIST berupa pixel hitam dan putih, yang telah dinormalisasikan ke 20x20 pixel, sambil menjaga aspect ratio-nya
- Citra kemudian dibuat agar berpusat pada kotak berukuran 28x28 piksel

	Training Set	Testing Set
Class	train-images-idx3-ubyte	t10k-images-idx3-ubyte
0	5923	980
1	6742	1135
2	5958	1032
3	6131	1010
4	5842	982
5	5421	892
6	5918	958
7	6265	1028
8	5851	974
9	5949	1009
Total	60000	10000

MNIST Dataset

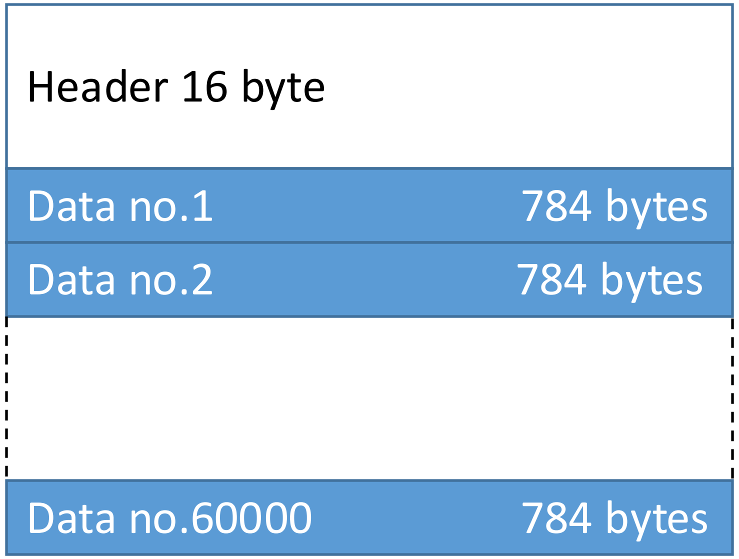
Skenario Pengujian : Hold-Out Method, data dibagi dua yaitu Training Set dan Testing Set. Training Set dipakai untuk pelatihan dan diukur performanya memakai Testing Set.



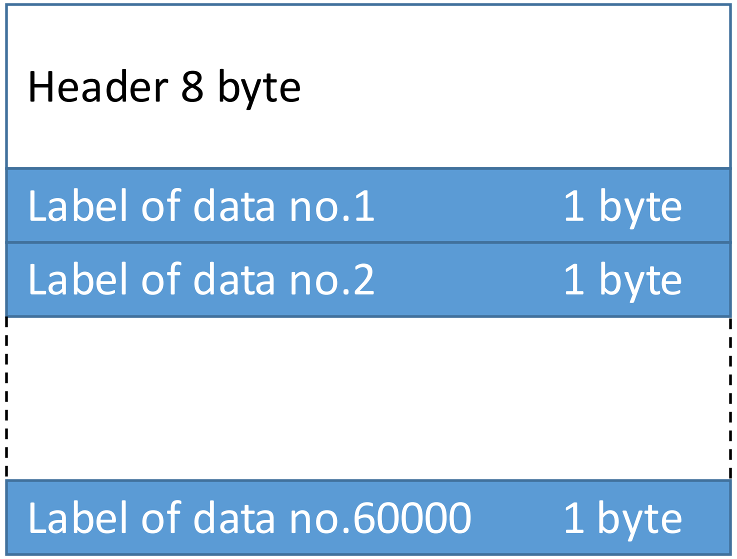
Classifier :

- Naïve Bayes
- K-Nearest Neighbor Classifier
- Multilayer Perceptron Artificial Neural Network
- Support Vector Machine
- Decision Tree
- Convolutional Neural Network
- dsb.

MNIST Dataset



train-images-idx3-ubyte: training set images



train-labels-idx1-ubyte: label of the dataset

- Size of one character is 28x28 pixel. One pixel is represented using one byte. File size of one character : 784 bytes. Thus, the size of train-images-idx3-ubyte is 16 + 60000*784 = 47040016 bytes. And the size of train-labels-idx1-ubyte is 8 + 60000*1 = 60008 bytes

Membaca Data MNIST

Beberapa catatan

- https://www.youtube.com/watch?v=c6otdpVMtXw
- https://www.youtube.com/watch?v=Zi4i7Q0zrBs

Keterangan

Beberapa catatan

- Error waktu import cv2 : https://stackoverflow.com/questions/76918044/cannot-import-mediapipe-typeerror-numpy-dtypemeta-object-is-not-subscripta
- pip install numpy==1.20.0
- Pip install tensorflow

Membaca Data MNIST

```
import cv2 as cv
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from tensorflow.keras.datasets import mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(60000, 784)
x_test  = x_test.reshape(10000, 784)
plt.imshow(x_train[59999].reshape((28, 28)), cmap = 'gray')
plt.show()
```

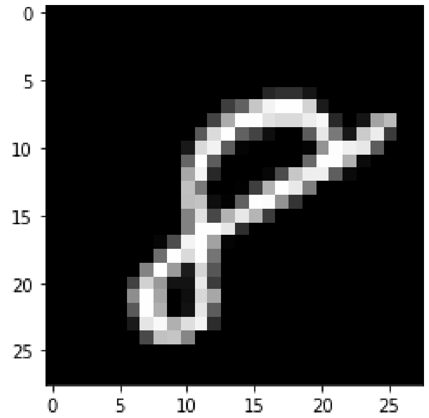
In [12]:
```
import cv2 as cv
import matplotlib.pyplot as plt
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from tensorflow.keras.datasets import mnist
```

In [14]:
```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(60000, 784)
x_test  = x_test.reshape(10000, 784)
```

In [15]:
```
x_test
```

Out[15]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

In [21]:
```
plt.imshow(x_train[59999].reshape((28, 28)), cmap = 'gray')
plt.show()
```



Pengenalan memakai Naïve Bayes

nb_model = GaussianNB()

fit_nb = nb_model.fit(x_train, y_train)

predictions = fit_nb.predict(x_test)

con_matrix = confusion_matrix(y_test, predictions)

print(con_matrix)

In [55]:
```
nb_model = GaussianNB()
fit_nb = nb_model.fit(x_train, y_train)
predictions = fit_nb.predict(x_test)
con_matrix = confusion_matrix(y_test, predictions)
print(con_matrix)
```

```
[[ 870    0    3    5    2    5   31    1   35   28]
 [   0 1079    2    1    0    0   10    0   38    5]
 [  79   25  266   91    5    2  269    4  271   20]
 [  32   39    6  353    2    3   51    8  409  107]
 [  19    2    5    4  168    7   63    7  210  497]
 [  71   25    1   20    3   44   40    2  586  100]
 [  12   12    3    1    1    7  895    0   26    1]
 [   0   15    2   10    5    1    5  280   39  671]
 [  13   72    3    7    3   11   12    4  648  201]
 [   5    7    3    6    1    0    1   13   18  955]]
```

Hasil Prediksi

		0	1	2	3	4	5	6	7	8	9
Class	**0**	870	0	3	5	2	5	31	1	35	28
	1	0	1079	2	1	0	0	10	0	38	5

banyaknya huruf "0" :	980
huruf "0"yang dikenali dengan benar	870
akurasi pengenalan angka "0"	88.8%

banyaknya huruf "1" :	1135
huruf "1"yang dikenali dengan benar	1079
akurasi pengenalan angka "1"	95.1%

Pengenalan memakai Naïve Bayes

```
def diagonal_sum(con_matrix):
   sum = 0
   for i in range(10):
      for j in range(10):
         if i==j: sum+= con_matrix[i, j]
   return sum

sum = diagonal_sum(con_matrix)
print(sum)
print(f'Accuracy %: {sum/10000}')
```

```
In [37]: def diagonal_sum(con_matrix):
             sum = 0
             for i in range(10):
                 for j in range(10):
                     if i==j: sum+= con_matrix[i, j]
             return sum

In [38]: sum = diagonal_sum(con_matrix)
         print(sum)
         print(f'Accuracy %: {sum/10000}')

         5558
         Accuracy %: 0.5558

In [ ]:
```

Pengenalan memakai Naïve Bayes

```
def diagonal_sum(con_matrix):
    total = 0
    for i in range(10):
        class_correct=0
        class_total=0
        for j in range(10):
            class_total+=con_matrix[i,j]
            if i==j: total+=con_matrix[i,j]
            if i==j: class_correct+= con_matrix[i, j]
        print (i,100*class_correct/class_total,class_correct,class_total)
    return
diagonal_sum(con_matrix)
print(f'Accuracy %: {100*total/10000}')
```

Confusion Matrix

- Matriks yang dipakai untuk menjelaskan performa sebuah classifier
- Contoh :
  - Baris pertama menunjukkan :
    - huruf A yang dikenali benar sebagai huruf A: 7
    - huruf A yang dikenali salah sebagai huruf B: 1
    - huruf A yang dikenali salah sebagai huruf C: 3
    - Akurasi huruf A = 7 / (7+1+3) = 63.5%
  - Baris kedua menunjukkan :
    - huruf B yang dikenali salah sebagai huruf A: 2
    - huruf B yang dikenali benar sebagai huruf B: 8
    - huruf B yang dikenali salah sebagai huruf C: 4
    - Akurasi huruf B = 8 / (2+8+4) = 57.1%
  - Baris ketigamenunjukkan :
    - huruf C yang dikenali salah sebagai huruf A: 1
    - huruf C yang dikenali salah sebagai huruf B: 2
    - huruf C yang dikenali benar sebagai huruf C: 9
    - Akurasi huruf C = 9 / (1+2+9) = 75%

		Hasil Prediksi		
		A	B	C
Class	A	7	1	3
	B	2	8	4
	C	1	2	9

Tugas

- Kerjakan hal serupa dengan metode k-Nearest Neighbor Classifier dan Multilayer Perceptron
  - https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
  - https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron

- Bandingkan hasil ketiga metode tersebut