

Pembelajaran Mesin

*Dr. Retno Kusumaningrum,
S.Si., M.Kom.*



The Course

Academic Year
2021/2022

About the Course

- Mata Kuliah : Pembelajaran Mesin
 - Machine Learning
- Kode Mata Kuliah : AlK21356
- Dosen Pengampu :
 - Pre Mid Term
Dr. Retno Kusumaningrum, S.Si., M.Kom
 - Post Mid Term
Rismiyati, B.Eng., M.Cs.
- Semester V

Assessment

Komponen Penilaian :

- Aktivitas Partisipatif & Hasil Proyek : 50%
- Tugas : 5%
- Quiz : 5%
- UTS : 20%
- UAS : 20%

Rentang Nilai :

- PAP (Panduan Acuan Patokan)

Aktivitas Partisipatif & Hasil Proyek

- Kelas akan dibagi menjadi 3 sampai 4 orang
- Masing-masing kelompok mengerjakan proyek untuk menerapkan algoritma pembelajaran mesin dalam kasus nyata
- Data kasus dapat diambil dari data real maupun public datasets
- Proyek harus menyertakan langkah-langkah pembelajaran mesin dengan jelas serta evaluasi kinerjanya
- Proyek diberikan pada TM 4, 6, 7 (Pra-Mid Term) dan TM 14 (Post-Mid Term)

Quiz, UTS, dan UAS

- Quiz diberikan pada:
 - Pra-Mid Term : TM₁ dan TM₃
 - Post-Mid Term : TM₁₂
- UTS
 - Multiple Choice
- UAS
 - Multiple Choice (Temporary)

The 2018 Top Programming Languages

Source: IEEE Spectrum



?

Praktikum



THE RANKINGS



RULES



Aturan Umum:

Untuk dapat mengikuti UAS setiap mahasiswa memiliki presensi kehadiran **sekurang-kurangnya 75%**.

Fakultas:

Perkuliahan Semester Gasal Tahun Ajaran 2020/2021 untuk semester III, V, dan VII dimulai tanggal **23 Agustus 2021**

Presensi kehadiran mahasiswa pada setiap mata kuliah, berlangsung melalui aplikasi SIAP. Jika terkendala sinyal, mahasiswa harus **menginformasikan kepada dosen pengampu paling lambat 3 (tiga) jam setelah perkuliahan selesai.**

Implementasi – Dr. Retno Kusumaningrum, S.Si., M.Kom.

- **Absensi** melalui **SIAP** hanya akan **dibuka 1x** dan akan diinformasikan **melalui kormat** atau saat ***online meeting***
- Bagi yang terkendala sinyal untuk proses absen bisa melaporkan **melalui kormat** dengan format:

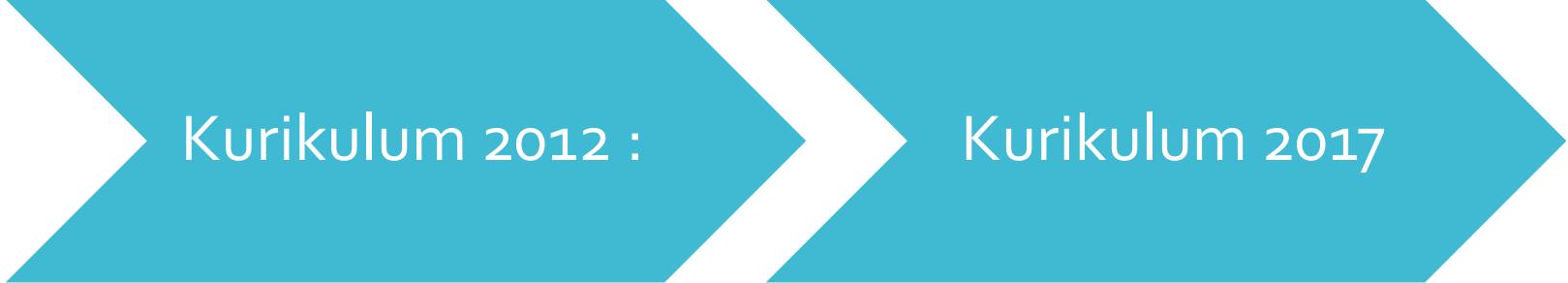
Nama :

NIM :

Penyebab Terkendala Sinyal :

Di Informatika UNDIP

Pembelajaran Mesin



Kurikulum 2012 :

- Machine Learning
- Mata Kuliah Pilihan
- Peserta Per Kelas : ± 10

Kurikulum 2017

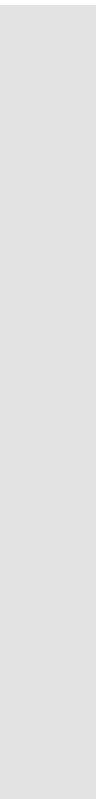
- Pembelajaran Mesin ≈ Machine Learning
- Mata Kuliah Wajib

Machine Learning From Zero to Hero

Machine Learning can be implemented in various fields:

- Software Engineering
- Information System & Information Technology
- Computation and Graphics
- Intelligent System

Why?



Di Informatika UNDI

Materi

Week 1

Introduction to Machine Learning

- Pengertian dan Karakteristik Machine Learning
- Learning Components & Solution Component
- Tipe Pembelajaran
- Pendekatan Implementasi Metode-metode Machine Learning
- Contoh Penerapan Metode Machine Learning dalam Dunia Nyata

QUIZ

Week 2

Konsep *Supervised Learning* dan Evaluasinya

- Konsep Dasar *Supervised Learning* (Jenis, Kategori berdasarkan jumlah kelas, Separability-Linearity)
- Proses di dalam *Supervised Learning*
- *Model Selection & Model Assesment*
- *Confusion Matrix*
- Ukuran Kinerja Klasifikasi
- ROC & AUC

Tugas

Week 3

K-Nearest Neighboor (KNN)

- Pengertian dan Karakteristiknya
- Algoritma KNN
- Normalisasi
- Ukuran-ukuran *Similarity Distance*
- Issue di dalam KNN

QUIZ

Week 4

Decision Tree

- Pengertian dan karakteristiknya
- Pengukuran informativeness dalam Decision Tree (Information Gain dan Entropi)
- Growing the Tree (incl. stopping criteria)
- Jenis-jenis algoritma Decision Tree (CART, ID3, dan C4.5)
- Overfitting di dalam Decision Tree
- Decision Tree Prunning sebagai teknik mengatasi overfitting
- Issue di dalam Decision Tree

PROJECT

Week 5

Bayesian Learning

- Pengertian Bayes Theorm dan penerapannya dalam Machine Learning
- Parameter Estimation (ML vs MAP)
- Algoritma terkait issue Data Diskrit dan Fitur Tunggal
- Algoritma terkait issue Data Diskrit dan Multi-Fitur
- Algoritma terkait issue Data Kontinyu dan Fitur Tunggal
- Algoritma terkait issue Data Kontinyu dan Multi-Fitur

Week 6

Naïve Bayes & Logistic Regression

- Pengertian dan karakteristik Naïve Bayes
- Algoritma Naïve Bayes untuk Data Diskrit
- Issue zero conditional probability dan teknik mengatasinya
- Algoritma Naïve Bayes untuk Data Kontinyu
- Pengertian dan karakteristik Logistic Regression
- Algoritma Logistic Regression

PROJECT

Week 7

Support Vector Machine (SVM)

- Pengertian Hyperplane
- Linear Classification via Hyperplane
- Concept of Margins
- Algoritma SVM (biclass - multiclass)
- Issue optimasi dalam SVM
- Generalisasi SVM (Large Margins)
- Pengenalan Kernel Methods (Non-linear SVM)

PROJECT

POST-MID TERM

- UNSUPERVISED LEARNING
 - K-Means Clustering
 - Hierarchical Clustering
 - EM-Algorithm
- EVALUASI PADA UNSUPERVISED LEARNING
- REINFORCEMENT LEARNING
- CURSE OF DIMENSIONALITY
- ENSEMBLE LEARNING

References

Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

Trevor Hastie, Robert Tibshirani and Jerome Friedman, Elements of Statistical Learning (2nd Edition), Springer Verlag, 2009

David MacKay, Information Theory, Inference, and Learning Algorithms (ver. 7.2), Cambridge University Press, 2005

David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press, 2016

Artikel-artikel Ilmiah

What is Machine Learning?

Definition and Its
History

ARTIFICIAL INTELLIGENCE



MACHINE LEARNING



DEEP LEARNING



A technique which enables computers to mimic human behavior

Classical AI :

- Rule Based System, Search Algorithms Depth First, Breadth First, A* Algorithm, Proportional Calculus, etc

ARTIFICIAL INTELLIGENCE



MACHINE LEARNING



DEEP LEARNING



A technique which
enables computers to
mimic human
behavior



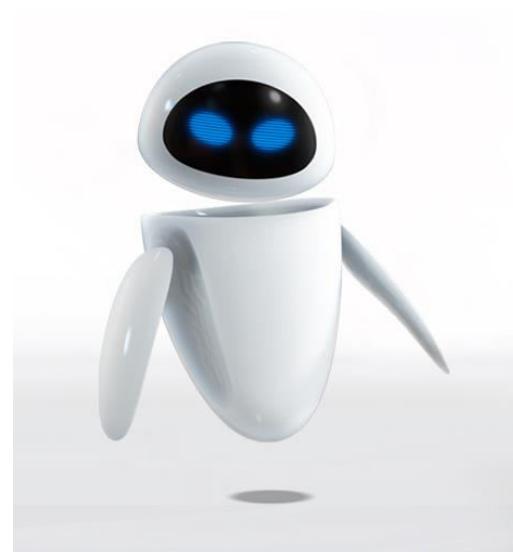
Learn from Experience



Human

data

Learn from Experience



Computer

Follow Instructions



Computer

What is Machine Learning?

- Subfield of Artificial Intelligence
- Can be seen as building blocks to enable computers or machine to improve its ability based on experiences, i.e. learns from data, thus it has intelligence like human
- It is a theoretical concept . There are various techniques with various implementations

ARTIFICIAL INTELLIGENCE



MACHINE LEARNING



DEEP LEARNING



A technique which enables computers to mimic human behavior



Subset of ML which use complex neural networks

Bigger Datasets



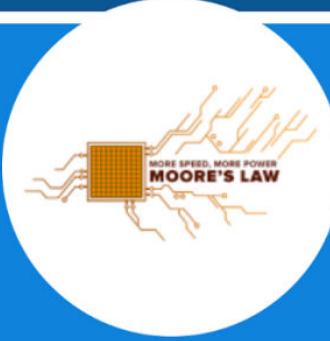
IMAGENET

10M labeled images

YouTube

8M categorized videos

Better Hardware

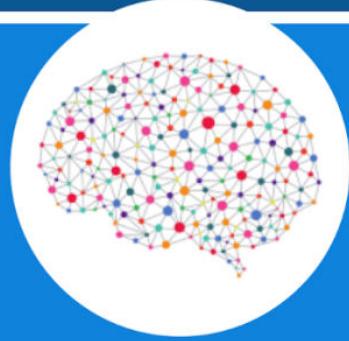


Moore's Law

Cost / GB in 1995: \$1000

Cost / GB in 2020: \$0.02

Smarter Algorithms



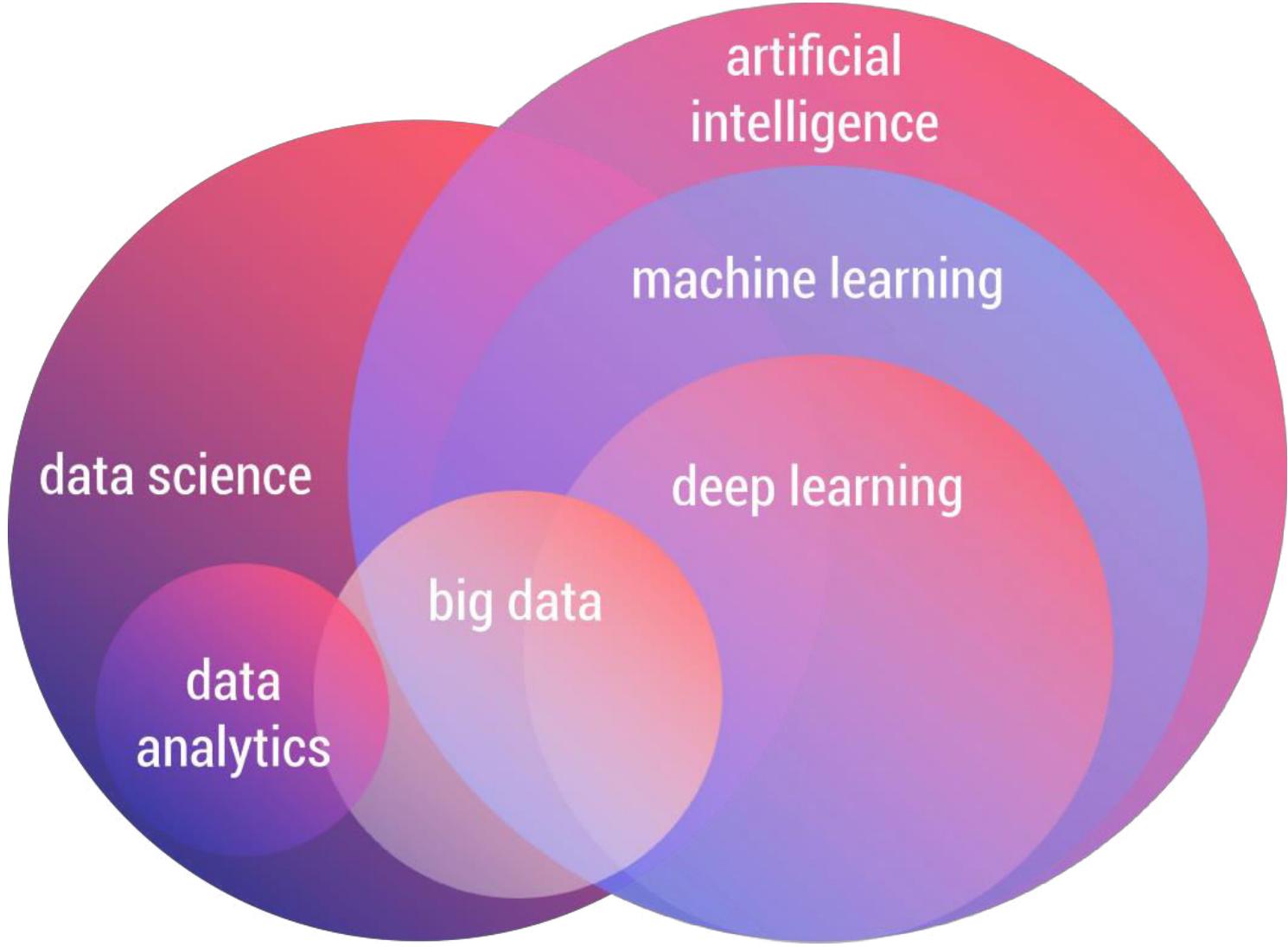
Recurrent Neural Networks

Convolutional Neural Nets

LSTM

<https://www.intel.com/content/www/us/en/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html>

Why Now?



Source image: <https://towardsdatascience.com/4-intersecting-domains-that-you-can-easily-confuse-with-artificial-intelligence-2233cb6ad7d1>

- Data terlalu 'besar' atau kompleks untuk dilakukan proses analisis dalam basisdata tradisional
- Dicirikan dengan 5 V

Volume

Variety

Velocity

Value*

Veracity*

Learning and Solution Components

Definition,
Formalization, and
Its Samples

Component of Learning

Contoh : credit approval

- Bank tidak memiliki ***magic formula*** untuk menentukan apakah seseorang tepat untuk diberikan kredit / tidak?
- Data historis kreditur bank dan bagaimana perilaku kreditur tersebut

Tell us about your business**Legal Business Name ***

Maximum of 36 characters.

Business Address * (no P.O. boxes) □ Suite/Apt# **Address * (no P.O. boxes) □ Suite/Apt#****City *****State *** Please select **ZIP Code****City *****State *** Please select **Business Phone Number ***

() -

Business' Legal Structure *

Sole Proprietorship  [What is this?](#) □

Authorizing Signature (must be one of the following) *

Please select 

Tell us about yourself [Why we need personal information?](#) □**First Name *** **MI** **Last Name *** same as business address**Address * (no P.O. boxes) □ Suite/Apt#** **Suite/Apt#****City *****State *** Please select **ZIP Co****Home Phone Number ***

() -

Date of birth(MM/DD/YYYY) *

/ /

□ Social Security Number *

- - [Why we need this?](#) □

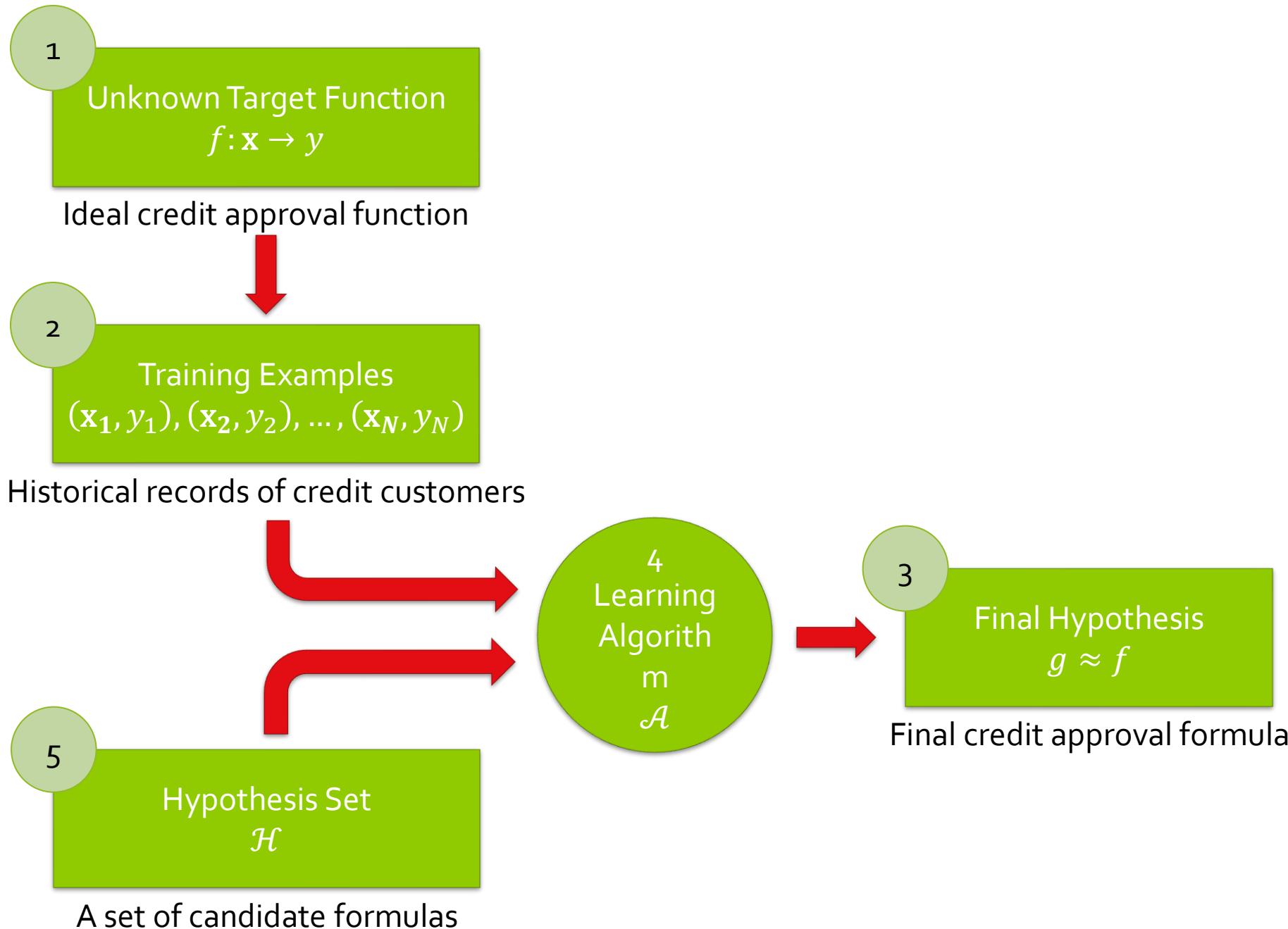
Do you rent or own your home? Own Rent Other**What is the amount of your rent or mortgage payment?**\$.00**Total household income ***\$.00 per year

Formalization

- Input : \mathbf{x}
 - customer application, vektor berdimensi d
- Output : $y (+1 \text{ or } -1)$
 - Good customer (+1) or bad customer (-1)
- Target function $f: \mathbf{x} \rightarrow y$
 - Ideal credit approval formula
 - Real condition : ***unknown to us***

- Why target function is unknown to us?
- Jika kita sudah tahu apa target function-nya maka yang bisa kita lakukan adalah cukup menerapkan rumusnya dan dihitung secara matematis...selesai...
- *Therefore, we need to learn from data*

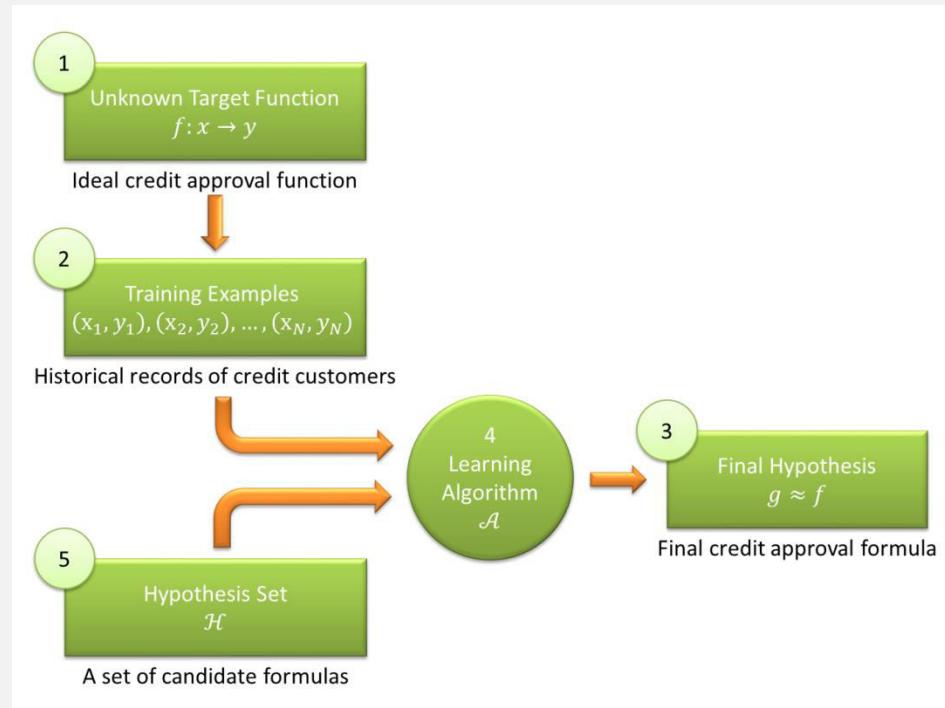
- Data :
 $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$
- Hypothesis: $g : \mathbf{x} \rightarrow y$
 - Hypothesis is a formula to approximate target function



Solution Components

- The two components of learning problem :
 - The Hypothesis Set $\mathcal{H} = \{h\}, g \in \mathcal{H}$
 - Learning Algorithm

Together, they are referred to as the **learning model**



Conclusions

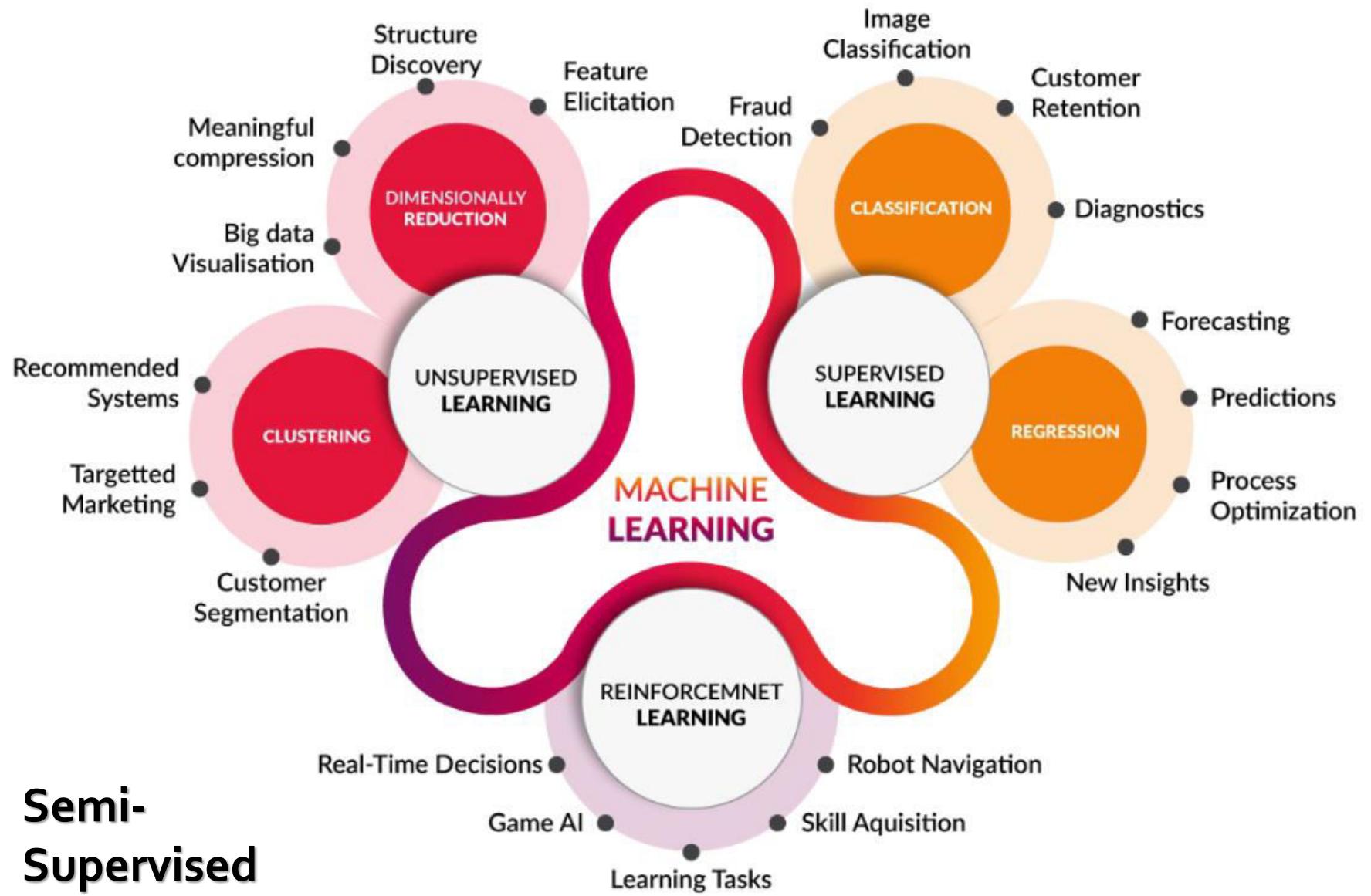
Learning is used when

.....

- We have data on it
- We cannot pin it down mathematically
- A pattern exists

Learning Methods

Three Types



LEARNING METHODS



Supervised Learning

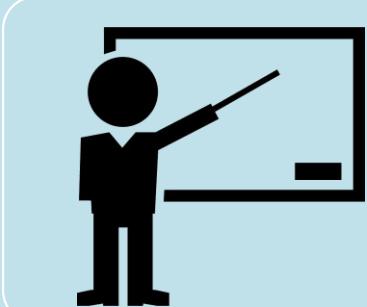


Unsupervised Learning

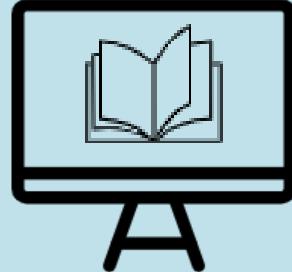


Reinforcement Learning

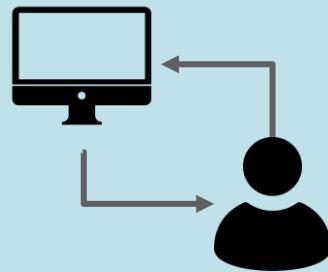
TYPES OF LEARNING METHOD



Supervised
Learning



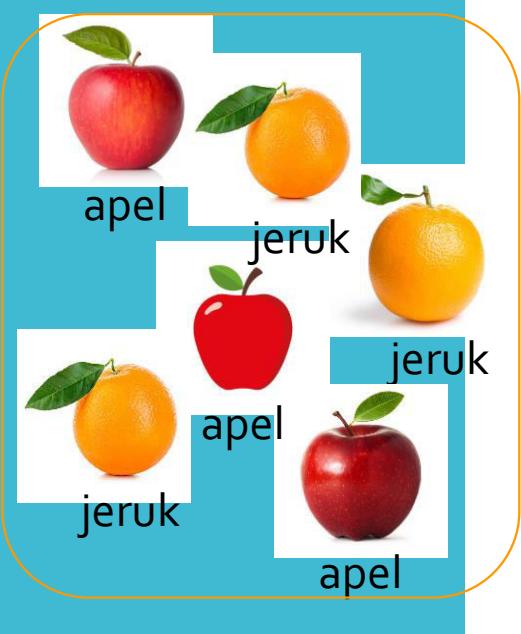
Unsupervised
Learning



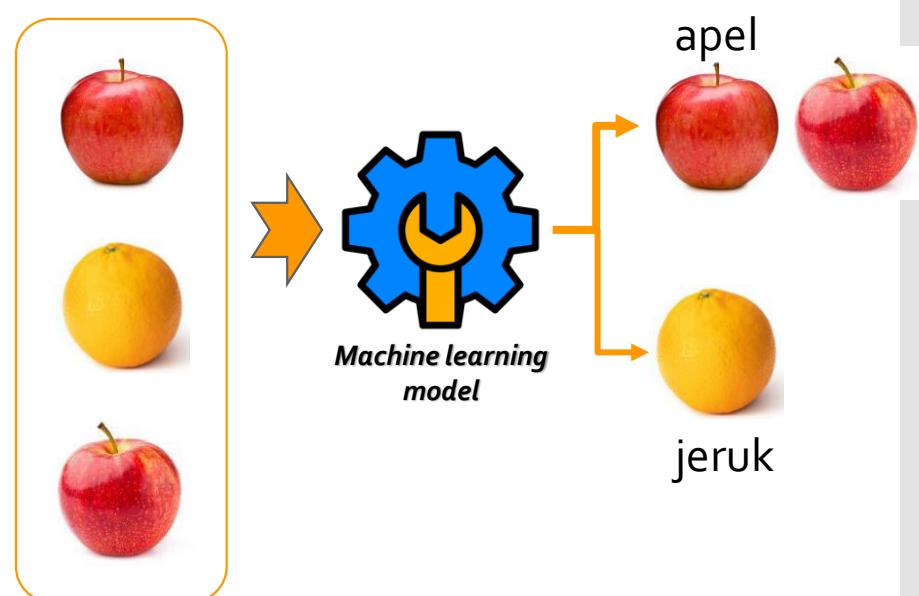
Reinforcement
Learning

SUPERVISED LEARNING – PEMBELAJARAN TERBIMBING

Provide the machine learning algorithm with categorized (labelled) data for training process



Provide the machine learning model with new (unlabelled) data to test if it tags new data appropriately

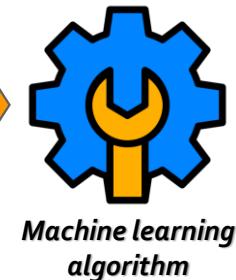
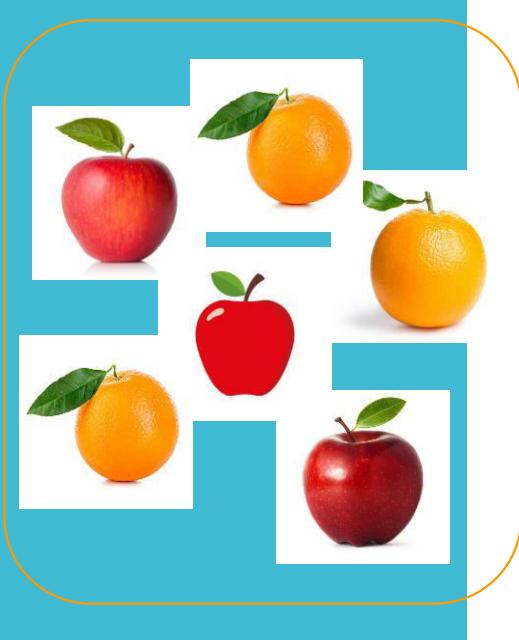




UNSUPERVISED PEMBELAJARAN TIDAK

Provide the machine learning algorithm with uncategorized (unlabelled) data to see what patterns it finds

Observe and learn from the patterns that the machine identifies



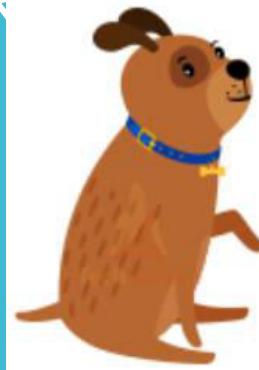
similar group



similar group



REINFORCING LEARNING



Sitting

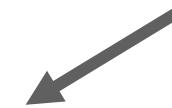
DOG (Agent)

STATE (Action)



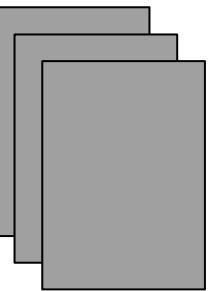
Walk

Reward

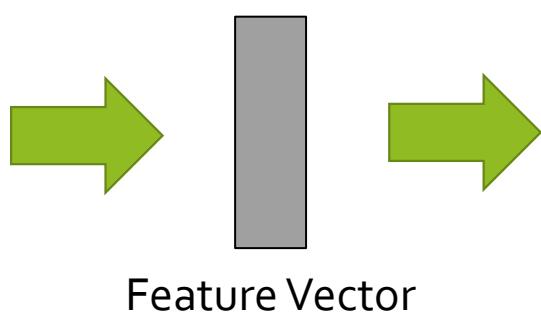


SUPERVISED LEARNING

Structure



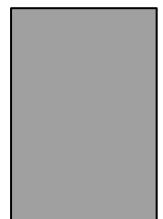
Labeled
Data



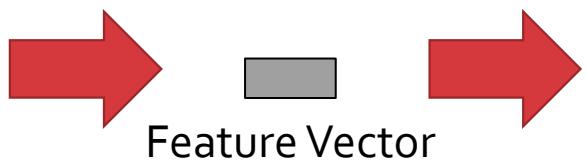
Machine
Learning
Algorithm

Training Process

Classification
Model



New Data



Classify /
Predict

Expected
Label / Class

Testing Process

1

K-Nearest
Neighboor

2

Decision Tree

3

Support
Vector
Machine
(SVM)

4

Bayesian
Learning
includes Naive
Bayes

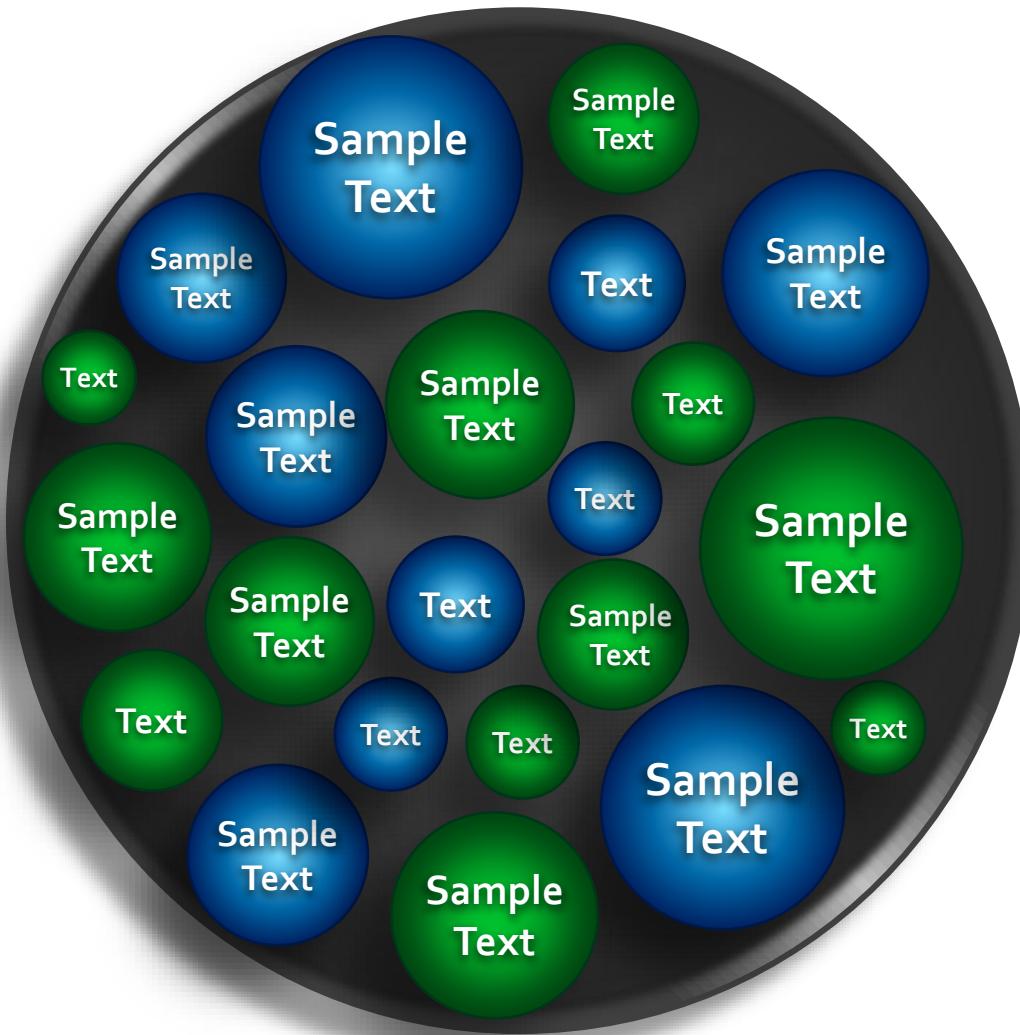
5

Logistic
Regression

Algorithm

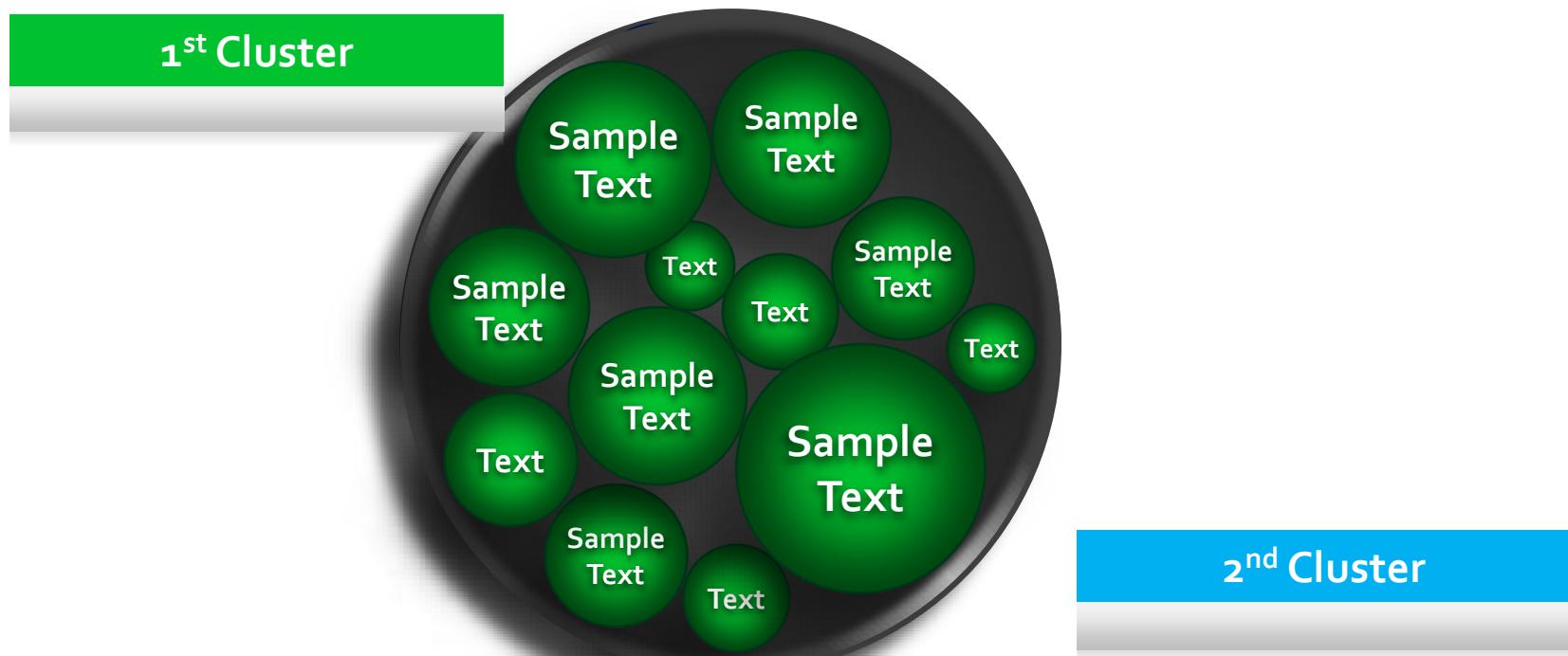
Unsupervised Learning

Illustration



Find “natural” grouping of instances given un-labeled data

identifying a finite set of categories or clusters to describe the data.



1

K-Means
Clustering

2

Hierarchical
Clustering

3

EM Algorithm

Algorithm

Reinforcement Learning

In Brief

Implementation Approach of Machine Learning Methods

Two Approaches

1

Batch
Learning

2

Online
Learning

Machine Learning In Everywhere

Example

Machine Learning in Software Engineering

Software quality (high-risk, or fault-prone component identification)

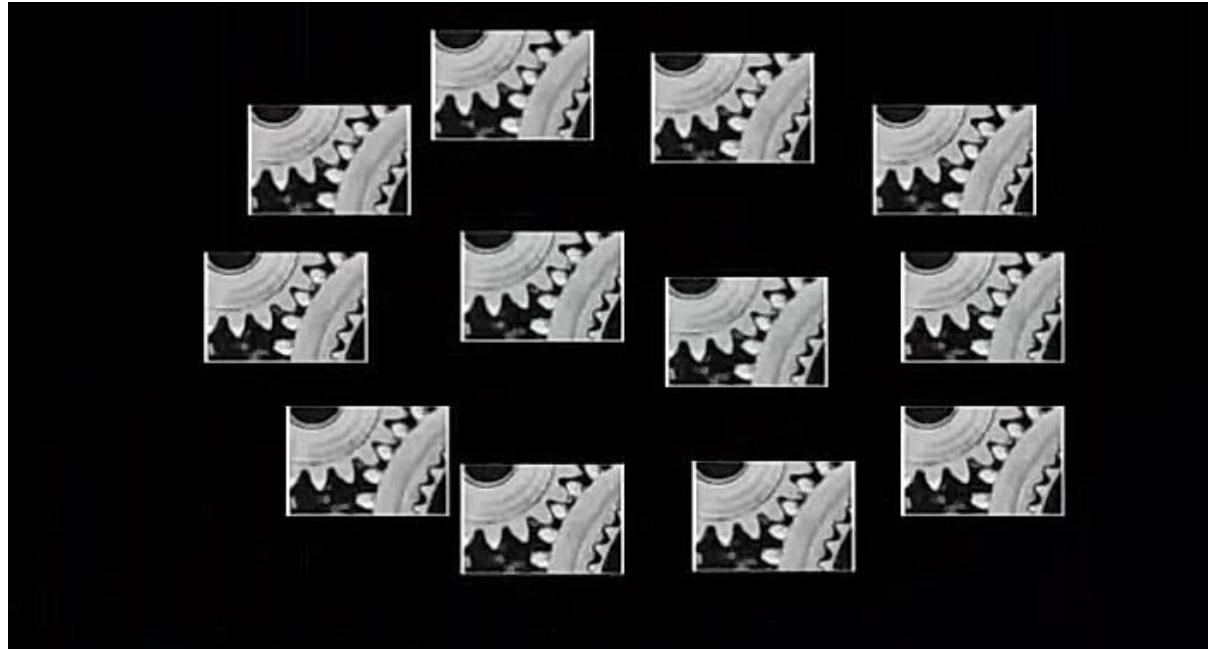
Software Defect Detection

Software Reliability

etc.

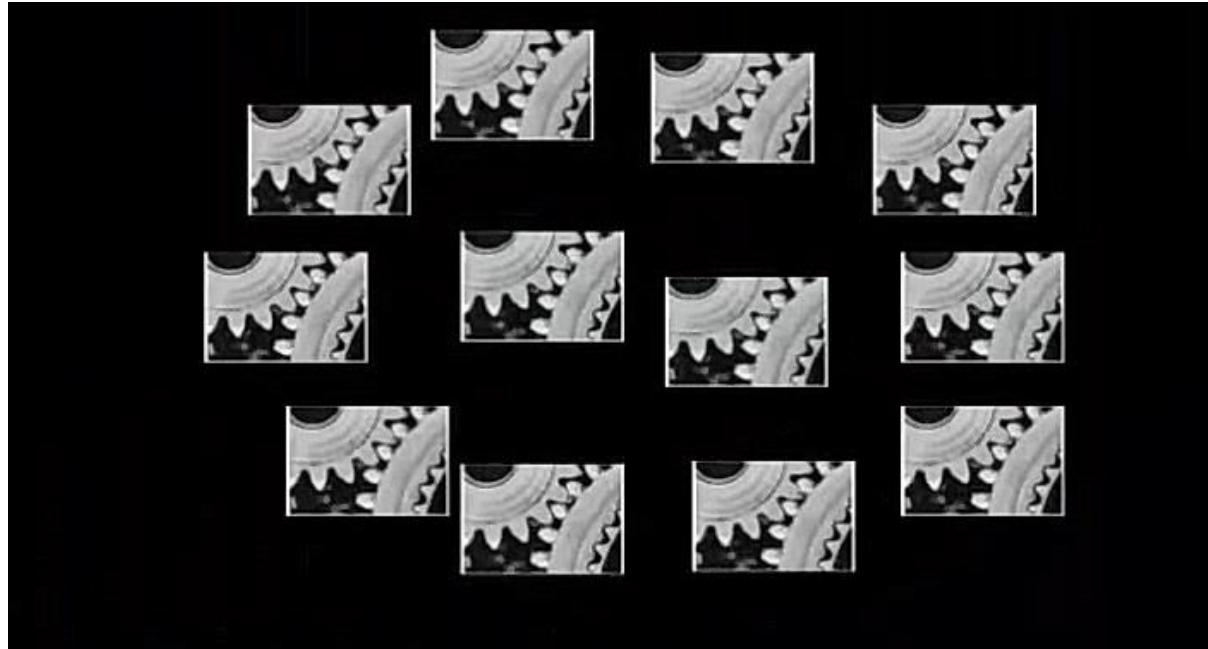
Series on Software Engineering and Knowledge Engineering: Volume 16
Machine Learning Applications in Software Engineering
Edited by: Du Zhang (California State University, USA), Jeffrey J P Tsai (University of Illinois, Chicago, USA)
Feb 2005

Software Code Is Composed of Several Components



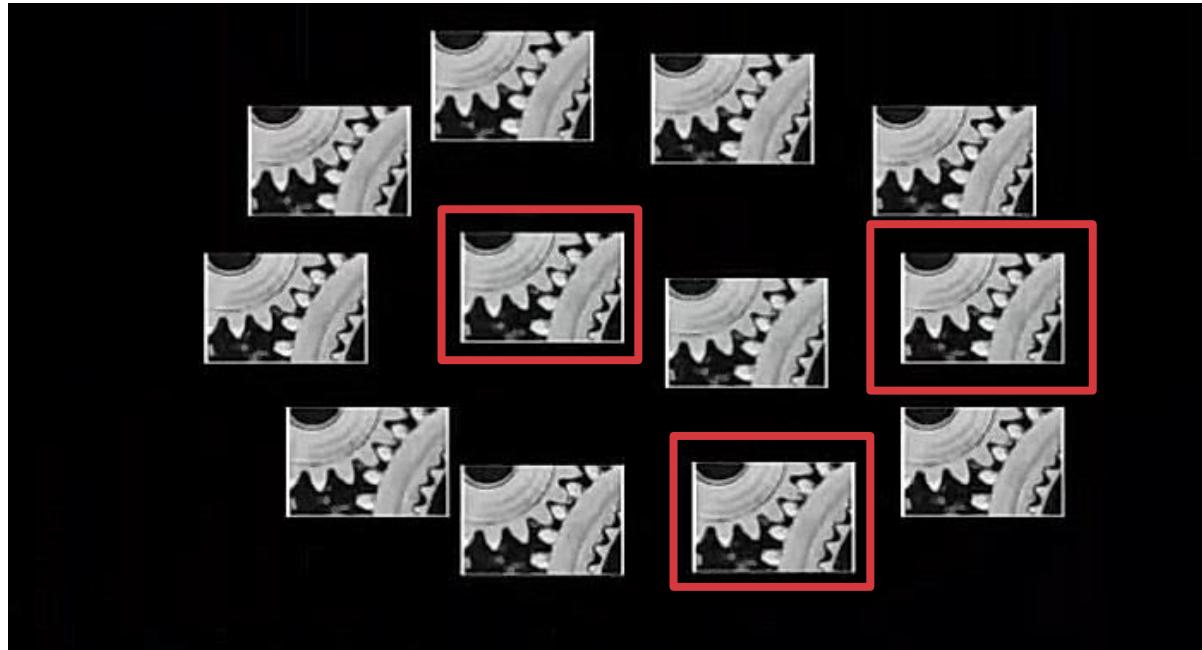
Software Defect Detection using Machine Learning

Testing all these components can be very expensive



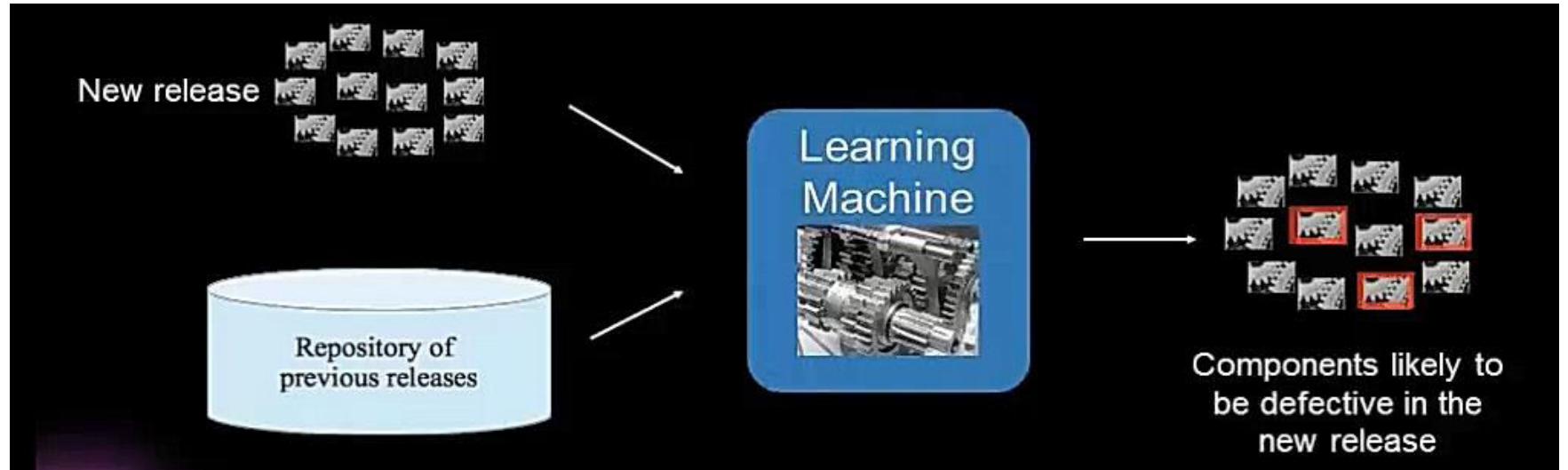
Software Defect Detection using Machine
Learning

If we know which components are likely to be defective,
we can increase testing cost-effectiveness



Software Defect Detection using Machine
Learning

Predictive model can be created to identify components likely to be defective by using past software releases and bug fixes as training data for machine learning



Software Defect Detection using Machine Learning



Software Defect Detection using Machine Learning

The diagram illustrates a machine learning dataset for software defect detection. It consists of a table with six columns. The first five columns represent input features: 'Branch count', 'Code + comment LOC', 'Halstead difficulty', 'Cyclomatic complexity', and an ellipsis column. The sixth column is the 'Defective?' target class. Above the table, a horizontal double-headed arrow spans the first five columns and is labeled 'Input features'. To the right of the table, another horizontal double-headed arrow spans the last two columns and is labeled 'Target class'.

Branch count	Code + comment LOC	Halstead difficulty	Cyclomatic complexity	...	Defective?
5	15	14.57	3		No
3	5	11.65	2		No
9	20	6.43	5		No
15	40	14.8	8		Yes
16	35	16.9	9		Yes

Software Defect Detection using Machine Learning

Machine Learning

in Information System & Information Technology

	Jenis Kelamin	Umur	Berat Badan	Tinggi Badan	Status Gizi
1	L	25	25	96	lebih
2	L	26	24	97	lebih
...
38	P	46	26.5	106	lebih
39	L	1	3.5	52	baik
40	L	1	4.5	52	baik
...
529	P	60	23	107	baik
530	L	7	6.2	65	rentan
531	L	8	6.4	67	rentan
...
592	P	56	12	93	rentan
593	L	10	5.8	68	kurang
594	L	33	9	77	kurang
...
612	P	54	10.9	90	kurang

Machine Learning in Computation and Graphics



Tennis Ball



Soccer Ball

How to Predict New Ball Pictures?



What is The Method?

- Supervised
- Unsupervised

What is/are The Feature(s)?

- Color?
- Shape?
- Size?
- Texture?
- Or ?

What is/are The Algorithm(s)?



Solutions ?????

Machine Learning in Intelligent System



What is The Method?

- Supervised
- Unsupervised

What is/are The Feature(s)?

- TF-IDF
- N-Grams
(Unigram,
Bigram, Trigram,
Quadgram, etc)
- PoS

What is/are The Algorithm(s)?



Solutions ?????

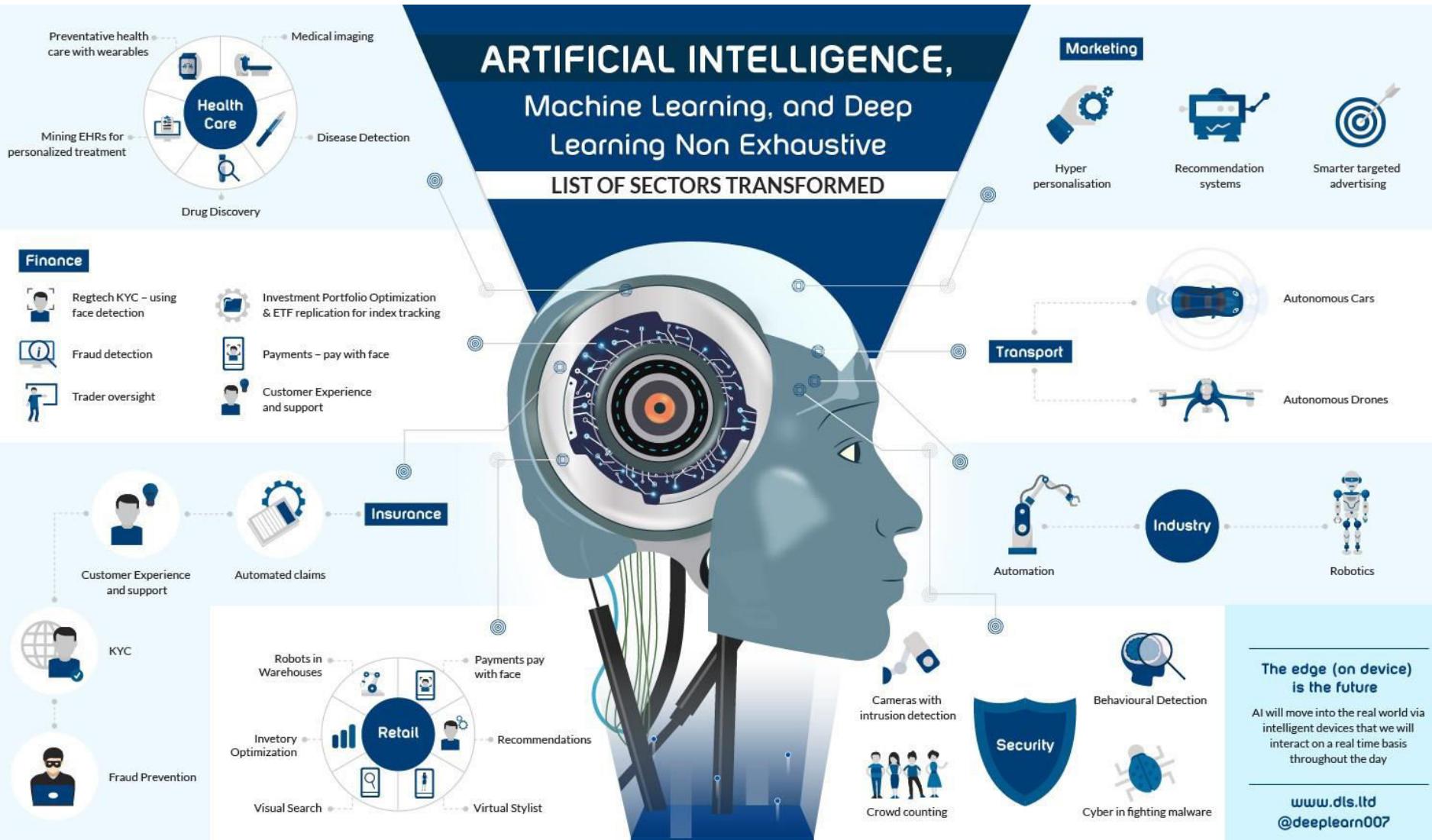


Image Source: https://www.bbntimes.com/images/AI_Transformation.jpeg

Concept of Supervised Learning & Its Performance Evaluation

Dr. Retno Kusumaningrum,
S.Si., M.Kom.



Supervised Learning

- Supervised learning menggunakan dataset yang terdiri atas pasangan variable input (x) dan output (y).

Patient ID	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Breast Cancer
1	48	23,5	70	2,707	0,467409	8,8071	9,7024	7,99585	417,114	No
2	83	20,69049	92	3,115	0,706897	8,8438	5,429285	4,06405	468,786	No
3	82	23,12467	91	4,498	1,009651	17,9393	22,43204	9,27715	554,697	No
4	45	20,83	74	4,56	0,832352	7,7529	8,237405	28,0323	382,955	Yes
...										
100	49	20,95661	94	12,305	2,853119	11,2406	8,412175	23,1177	573,63	Yes
	48	23,5	70	2,707	0,467409	8,8071	9,7024	7,99585	417,114	?

Supervised Learning (cont.)

- Supervised learning melakukan pembelajaran terhadap variable input (\mathbf{x}) dan output (y) untuk mendapatkan fungsi $f(\mathbf{x})$ yang tepat yang dapat memetakan setiap input \mathbf{x} dengan benar sebagai y .
- Variabel output y berperan sebagai teacher yang membimbing proses pembelajaran → supervised learning.

$$y = f(\mathbf{x})$$

- \mathbf{x} : variable/ fitur/ atribut input, biasanya berupa vector untuk satu buah data atau matriks untuk sekumpulan data dalam dataset
- y : variable output/ atribut target, biasanya berupa nilai scalar (kontinu atau diskret)
- $f()$: fungsi yang dicari dalam proses pembelajaran



Supervised Learning (cont.)

- Setelah fungsi f didapatkan, maka diharapkan fungsi tersebut dapat digunakan untuk memetakan (memprediksi) input data baru (\hat{x}) dengan tepat.
- Fungsi f yang dihasilkan dari proses pembelajaran disebut juga sebagai model.



JENIS-JENIS SUPERVISED LEARNING

Berdasarkan:



Jenis Supervised Learning

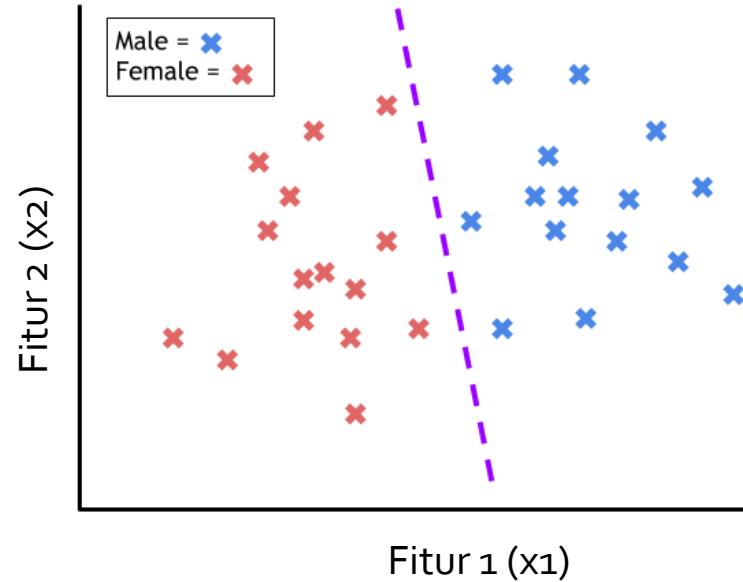
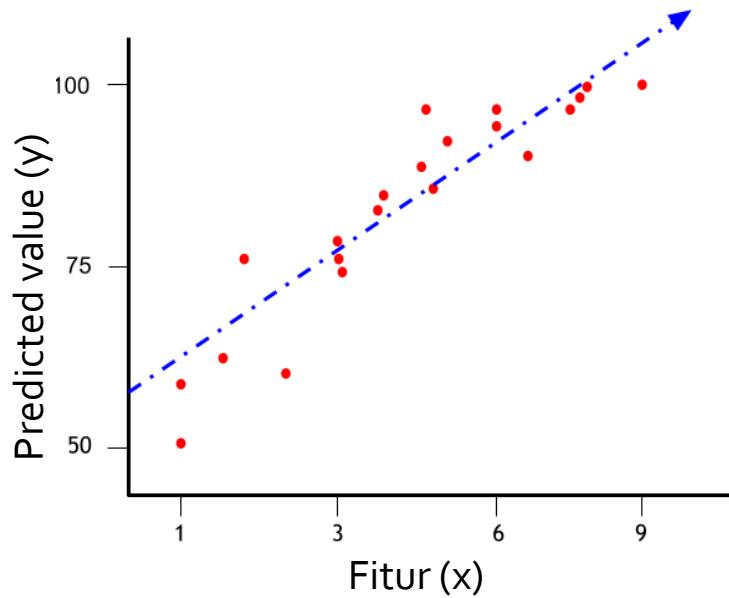
Berdasarkan Tipe Outputnya



Jenis Supervised Learning

Berdasarkan tipe nilai output-nya (y), supervised learning dibagi menjadi 2:

- Regresi : jika output y bernilai kontinu
- Klasifikasi : jika output y bernilai diskret/ kategori



Regresi

Contoh 1: Prediksi besarnya emisi gas CO₂

- Input: informasi engine
- Target class: besarnya emisi gas CO₂

1	MODEL	YEAR	MAKE	MODEL	VEHICLE	ENGINESIZE	CYLINDER	TRANSMIS	FUELTYPE	ION_CITY	ION_HWY	ON_CO2	ON_COMB	FUELCONC	CO2EMISSIONS
2	2014	ACURA	ILX	ILX	COMPACT	2,0	4	AS5	Z	9,9	6,7	8,5	33	196	
3	2014	ACURA	ILX	ILX	COMPACT	2,4	4	M6	Z	11,2	7,7	9,6	29	221	
4	2014	ACURA	ILX HYBRID	ILX HYBRID	COMPACT	1,5	4	AV7	Z	6,0	5,8	5,9	48	136	
5	2014	ACURA	MDX 4WD SUV - SMA	MDX 4WD SUV - SMA	3WD	3,5	6	AS6	Z	12,7	9,1	11,1	25	255	
6	2014	ACURA	RDX AWD SUV - SMA	RDX AWD SUV - SMA	3WD	3,5	6	AS6	Z	12,1	8,7	10,6	27	244	
7	2014	ACURA	RLX	RLX	MID-SIZE	3,5	6	AS6	Z	11,9	7,7	10,0	28	230	
8	2014	ACURA	TL	TL	MID-SIZE	3,5	6	AS6	Z	11,8	8,1	10,1	28	232	
9	2014	ACURA	TL AWD	TL AWD	MID-SIZE	3,7	6	AS6	Z	12,8	9,0	11,1	25	255	
10	2014	ACURA	TL AWD	TL AWD	MID-SIZE	3,7	6	M6	Z	13,4	9,5	11,6	24	267	
11	2014	ACURA	TSX	TSX	COMPACT	2,4	4	AS5	Z	10,6	7,5	9,2	31	212	
12	2014	ACURA	TSX	TSX	COMPACT	2,4	4	M6	Z	11,2	8,1	9,8	29	225	
13	2014	ACURA	TSX	TSX	COMPACT	3,5	6	AS5	Z	12,1	8,3	10,4	27	239	
14	2014	ASTON M/ DB9	MINICOM	MINICOM	2WD	5,9	12	A6	Z	18,0	12,6	15,6	18	359	
15	2014	ASTON M/ RAPIDE	SUBCOMP	SUBCOMP	2WD	5,9	12	A6	Z	18,0	12,6	15,6	18	359	
16	2014	ASTON M/ V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	2WD	4,7	8	AM7	Z	17,4	11,3	14,7	19	338	
17	2014	ASTON M/ V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	2WD	4,7	8	M6	Z	18,1	12,2	15,4	18	354	
18	2014	ASTON M/ V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	2WD	4,7	8	AM7	Z	17,4	11,3	14,7	19	338	
19	2014	ASTON M/ V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	V8 VANTAGE TWO-SEAT	2WD	4,7	8	M6	Z	18,1	12,2	15,4	18	354	
20	2014	ASTON M/ VANQUISH	MINICOM	MINICOM	2WD	5,9	12	A6	Z	18,0	12,6	15,6	18	359	
21	2014	AUDI	A4	A4	COMPACT	2,0	4	AV8	Z	9,9	7,4	8,8	32	202	
22	2014	AUDI	A4 QUATTRO	A4 QUATTRO	COMPACT	2,0	4	AS8	Z	11,5	8,1	10,0	28	230	



Regresi

Contoh 2: Prediksi harga rumah

- Input: informasi rumah
- Target class: harga rumah per meter

1	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
2	1	2012,917	32	84,87882	10	24,98298	121,5402	37,9
3	2	2012,917	19,5	306,5947	9	24,98034	121,5395	42,2
4	3	2013,583	13,3	561,9845	5	24,98746	121,5439	47,3
5	4	2013,500	13,3	561,9845	5	24,98746	121,5439	54,8
6	5	2012,833	5	390,5684	5	24,97937	121,5424	43,1
7	6	2012,667	7,1	2175,03	3	24,96305	121,5125	32,1
8	7	2012,667	34,5	623,4731	7	24,97933	121,5364	40,3
9	8	2013,417	20,3	287,6025	6	24,98042	121,5422	46,7
10	9	2013,500	31,7	5512,038	1	24,95095	121,4845	18,8
11	10	2013,417	17,9	1783,18	3	24,96731	121,5148	22,1
12	11	2013,083	34,8	405,2134	1	24,97349	121,5337	41,4
13	12	2013,333	6,3	90,45606	9	24,97433	121,543	58,1
14	13	2012,917	13	492,2313	5	24,96515	121,5373	39,3
15	14	2012,667	20,4	2469,645	4	24,96108	121,5104	23,8
16	15	2013,500	13,2	1164,838	4	24,99156	121,5340	34,3
17	16	2013,583	35,7	579,2083	2	24,9824	121,5461	50,5
18	17	2013,250	0	292,9978	6	24,97744	121,5445	70,1
19	18	2012,750	17,7	350,8515	1	24,97544	121,5311	37,4
20	19	2013,417	16,9	368,1363	8	24,9675	121,5445	42,3
21	20	2012,667	1,5	23,38284	7	24,96772	121,5410	47,7
22	21	2013,417	4,5	2275,877	3	24,96314	121,5115	29,3
23	22	2013,417	10,5	279,1726	7	24,97528	121,5454	51,6
24	23	2012,917	14,7	1360,139	1	24,95204	121,5484	24,6



Classification

Output y bernilai diskret/ kategori.

Contoh 1 : Prediksi / diagnose kanker pada pasien.

- Input : data medis pasien
- Target class: cancer or no cancer

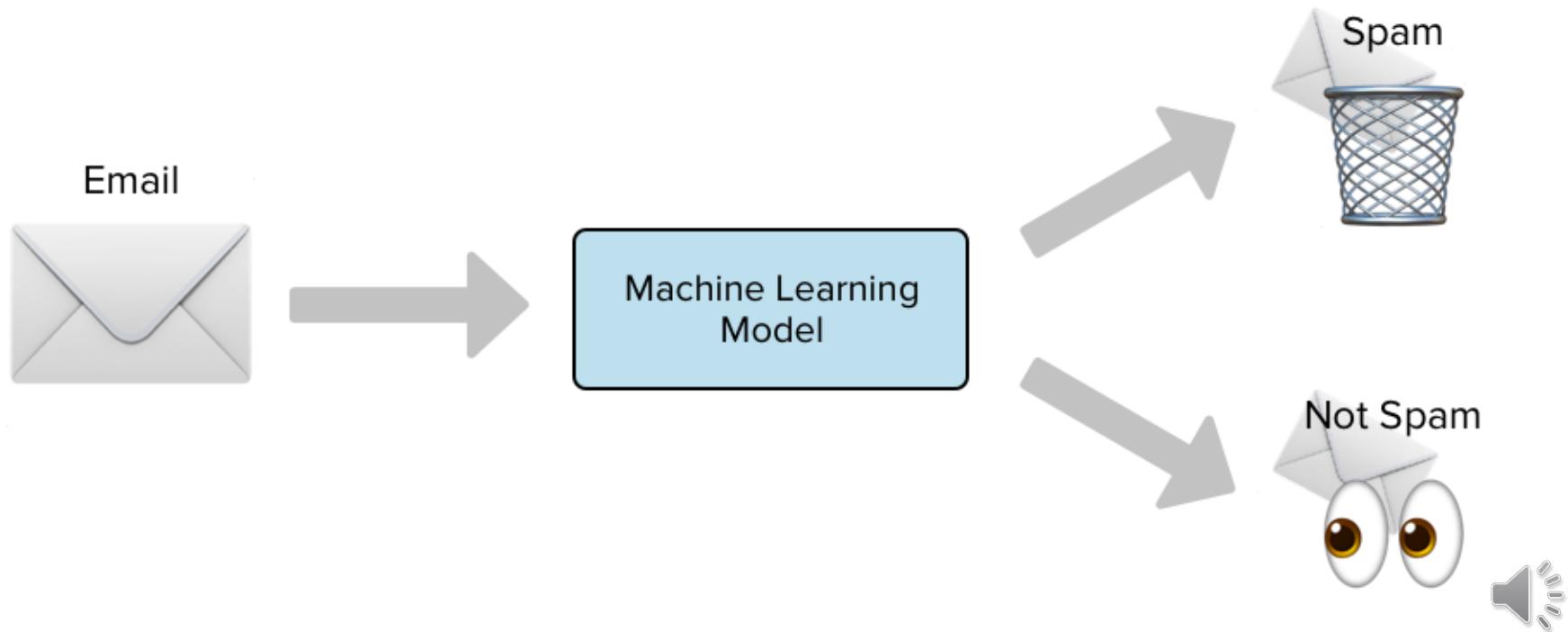
Patient ID	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Breast Cancer
1	48	23,5	70	2,707	0,467409	8,8071	9,7024	7,99585	417,114	No
2	83	20,69049	92	3,115	0,706897	8,8438	5,429285	4,06405	468,786	No
3	82	23,12467	91	4,498	1,009651	17,9393	22,43204	9,27715	554,697	No
4	45	20,83	74	4,56	0,832352	7,7529	8,237405	28,0323	382,955	Yes
...										
100	49	20,95661	94	12,305	2,853119	11,2406	8,412175	23,1177	573,63	Yes
	48	23,5	70	2,707	0,467409	8,8071	9,7024	7,99585	417,114	?



Classification

Contoh 2 : Prediksi spam

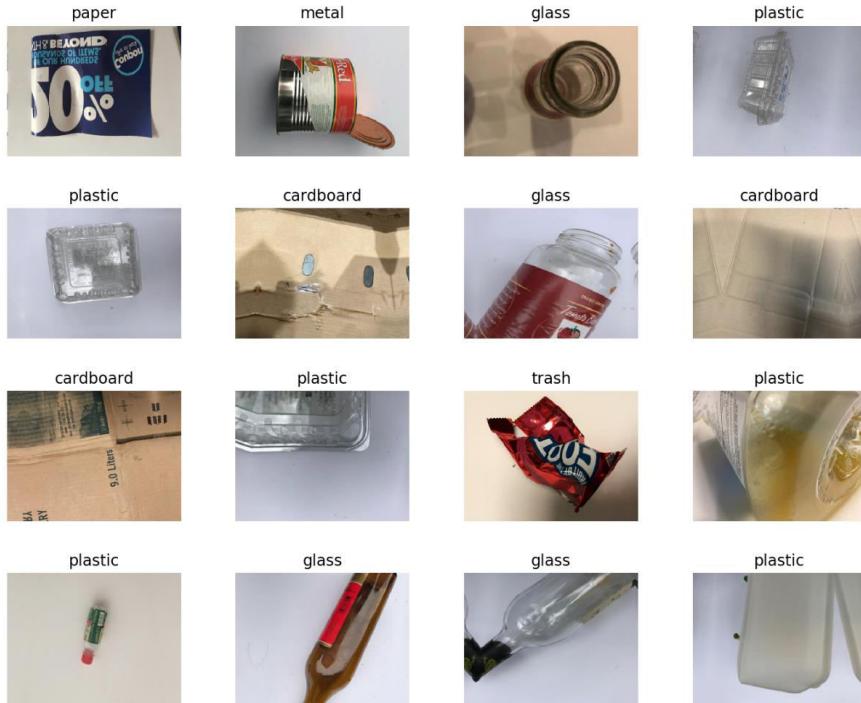
- Input: email message
- Target class: spam or not spam



Classification

Contoh 3: Prediksi jenis sampah

- Input: citra sampah
- Target class: plastic | cardboard | glass | trash



Classification

plastic

cardboard

glass

trash



Jenis Supervised Learning

Berdasarkan Jumlah Outputnya



Berdasarkan Jumlah Kelas

- Binary classification: target atribut hanya terdiri atas 2 ketagori/ kelas.
 - Contoh:
 - Prediksi kanker payudara, target class: yes or no
 - Prediksi spam, target class: spam or not spam
- Multiclass classification: target atribut terdiri atas lebih dari dua kelas.
 - Contoh:
 - Klasifikasi jenis sampah, target class: plastic, cardboard, glass, trash



Jenis Supervised Learning

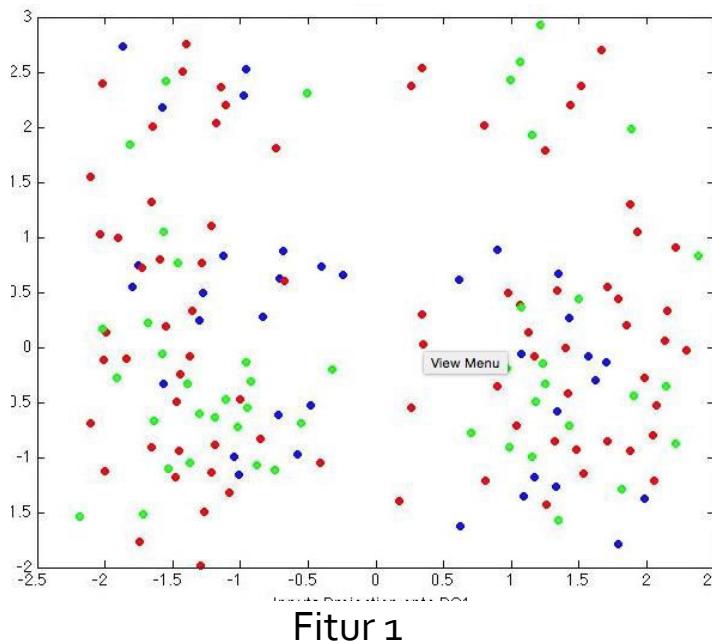
Berdasarkan Separability & Linearity



Classification Separability

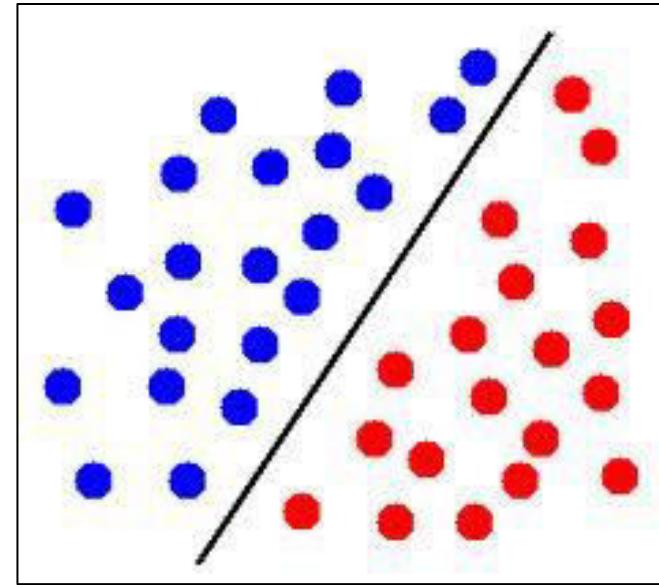
- Separable : kelompok yang berbeda menempati daerah pada ruang dimensi yang berbeda.
- Non-separable : kelompok yang berbeda bercampur dalam daerah yang sama dalam ruang dimensi --> transformasi ke bentuk lain.

Fitur 2



Fitur 1

Fitur 2

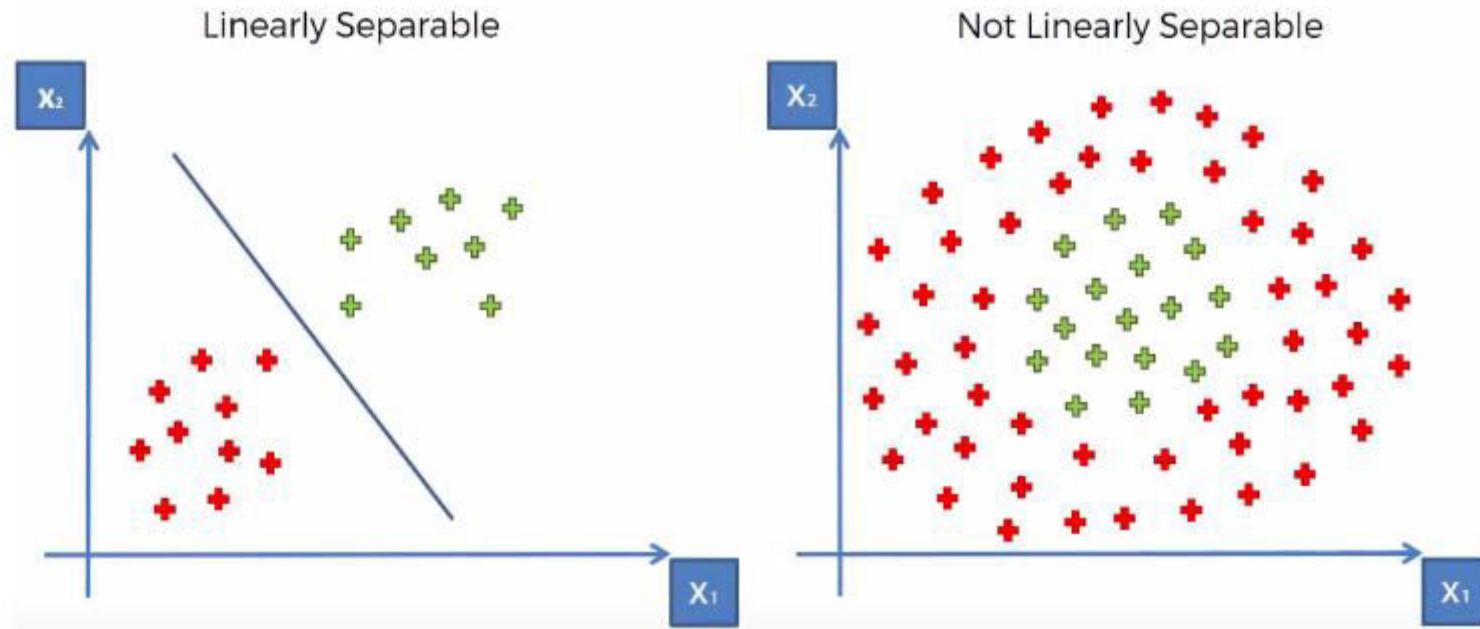


Fitur 1



Classification Linearity

- Linearly separable : kelompok yang berbeda dapat dipisahkan dengan sebuah garis lurus
- Non-linearly separable : kelompok yang berbeda tidak dapat dipisahkan dengan sebuah garis lurus (misal: 2 garis lurus atau kurva)

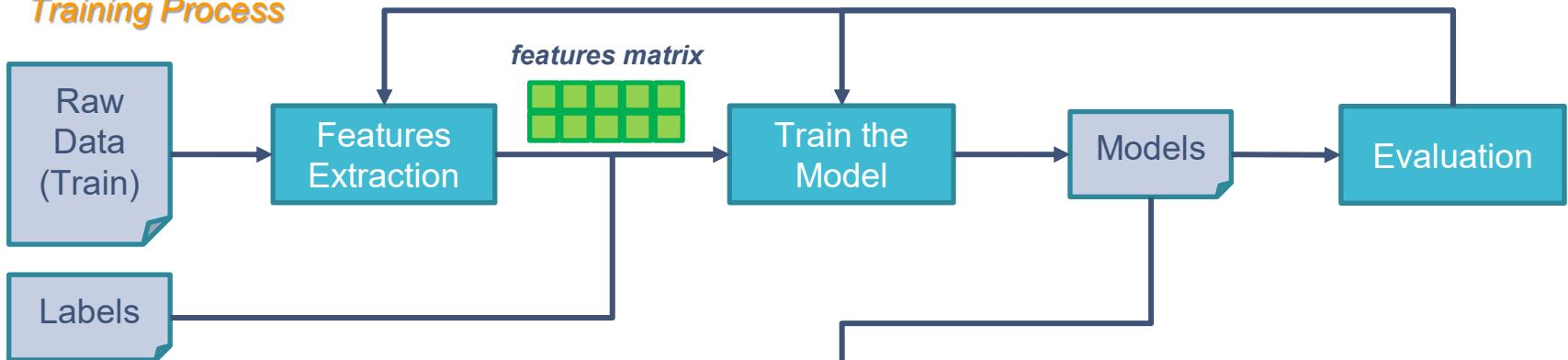


Supervised Learning Process

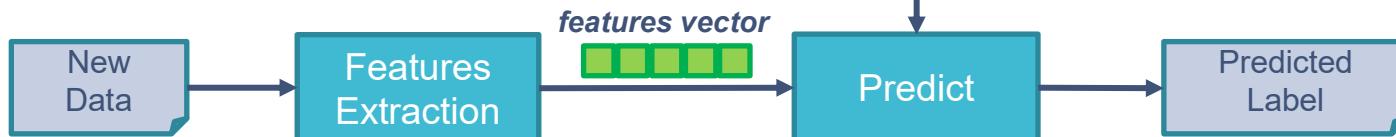


SUPERVISED LEARNING WORKFLOW - PIPELINE

Training Process



Testing Process



Supervised Learning Process

Ada 2 proses utama dalam supervised learning:

- Pelatihan/ training/ learning
 - Menggunakan dataset yang terdiri pasangan data input dan output → disebut data latih.
 - Tujuan: melakukan pembelajaran pada data latih untuk membangun model.
- Pengujian/ testing/ validation
 - Tujuan: menguji kemampuan model dari proses pelatihan.
 - Input data dimasukkan ke model sehingga didapat predicted output.
 - Predicted output dibandingkan dengan actual output (output yang sebenarnya).
 - Agar hasil pengujian fair → menggunakan subset data yang berbeda dengan data latih.



Supervised Learning Process (cont.)

Beberapa preprocessing dapat dilakukan terhadap dataset sebelum data dimasukkan ke algoritma klasifikasi, seperti:

- Normalisasi
- Seleksi atau ekstraksi fitur



Concept of Model Selection & Model Assesment

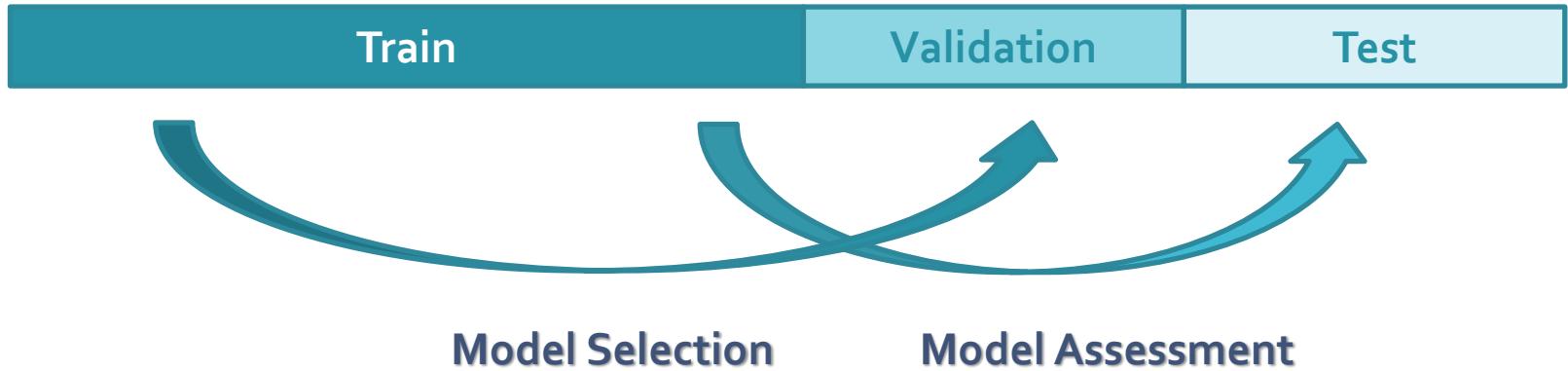
Data Training, Validation dan Test

Model Selection

- Estimating performances of different models to choose the best one (produces the minimum validation error)

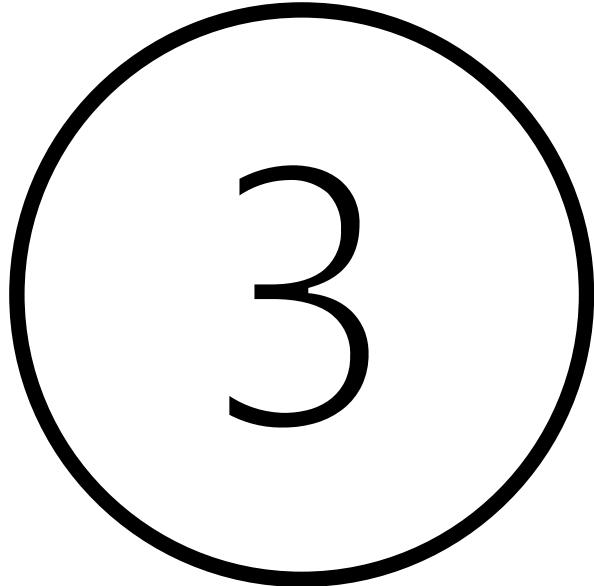
Model Assessment

- Having chosen a model, estimating the prediction error on new data



Pembagian Dataset

Data Training, Validation dan Test



2 Jenis

2 Subset Data

- Dataset dibagi menjadi 2 kelompok data yang disebut sebagai:
 - *Training set* (data pelatihan) : data ini digunakan sebagai pelatihan untuk membangun model klasifikasi / pengklasifikasi / *classifier*
 - *Testing set* (data pengujian) : data ini digunakan untuk menguji kinerja dari model klasifikasi yang dihasilkan
- Pembagian datanya mengikuti konsep *hold-out method*
 - Dataset dibagi menjadi dua bagian data yang tidak saling overlap dimana komposisi dari *training data* lebih besar dari *testing data*
- Komposisi yang umum digunakan adalah:
 - 70% - 30% atau 80% - 20%
 - 2 : 1

3 Subset Data

Dataset dibagi menjadi 3 kelompok data yang disebut sebagai:

- *Training set* (data pelatihan) :
 - Data ini digunakan sebagai pelatihan untuk membangun model klasifikasi / pengklasifikasi / *classifier*
- *Validation set* (data validasi)
 - Data ini digunakan sebagai dasar untuk memilih:
 - Best performing algorithm (NB or DT or)
 - Best model architecture (NN)
 - Learning / training parameters
- *Testing set* (data pengujian) :
 - Data ini digunakan untuk mengestimasi kinerja model klasifikasi pada *real word situation*
 - *Unseen dataset*

(lanjutan)

- Pembagian subset data ini dilakukan sejak awal.
Komposisi bisa berbeda-beda, pada umumnya:
 - *80% Training set, 10% Validation set, 10% Testing set*
 - *70% Training set, 20% Validation set, 10% Testing set*
- Proses pembagian data tetap dilakukan menggunakan konsep *hold-out method*
 - *Subset data* tidak saling overlap
- Umumnya diterapkan untuk dataset yang besar

Best Practice

Jika data yang dimiliki terlalu kecil untuk pembagian 3 subset data maka dapat dilakukan strategi sebagai berikut:

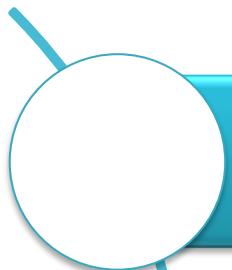
- Menggunakan *training set* dan *validation set* yang dibentuk menggunakan konsep *n-folds cross validation*
- Tetap membentuk *testing set* tersendiri

Misalkan kita memiliki 11.000 sampel data, maka:

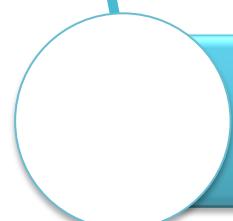
- 10.000 sampel data akan kita gunakan untuk proses *cross validation*
- 1.000 sampel data kita gunakan sebagai *testing set*

Cross Validation ???

Cross Validation



K-folds cross validation



Stratified K-folds cross validation



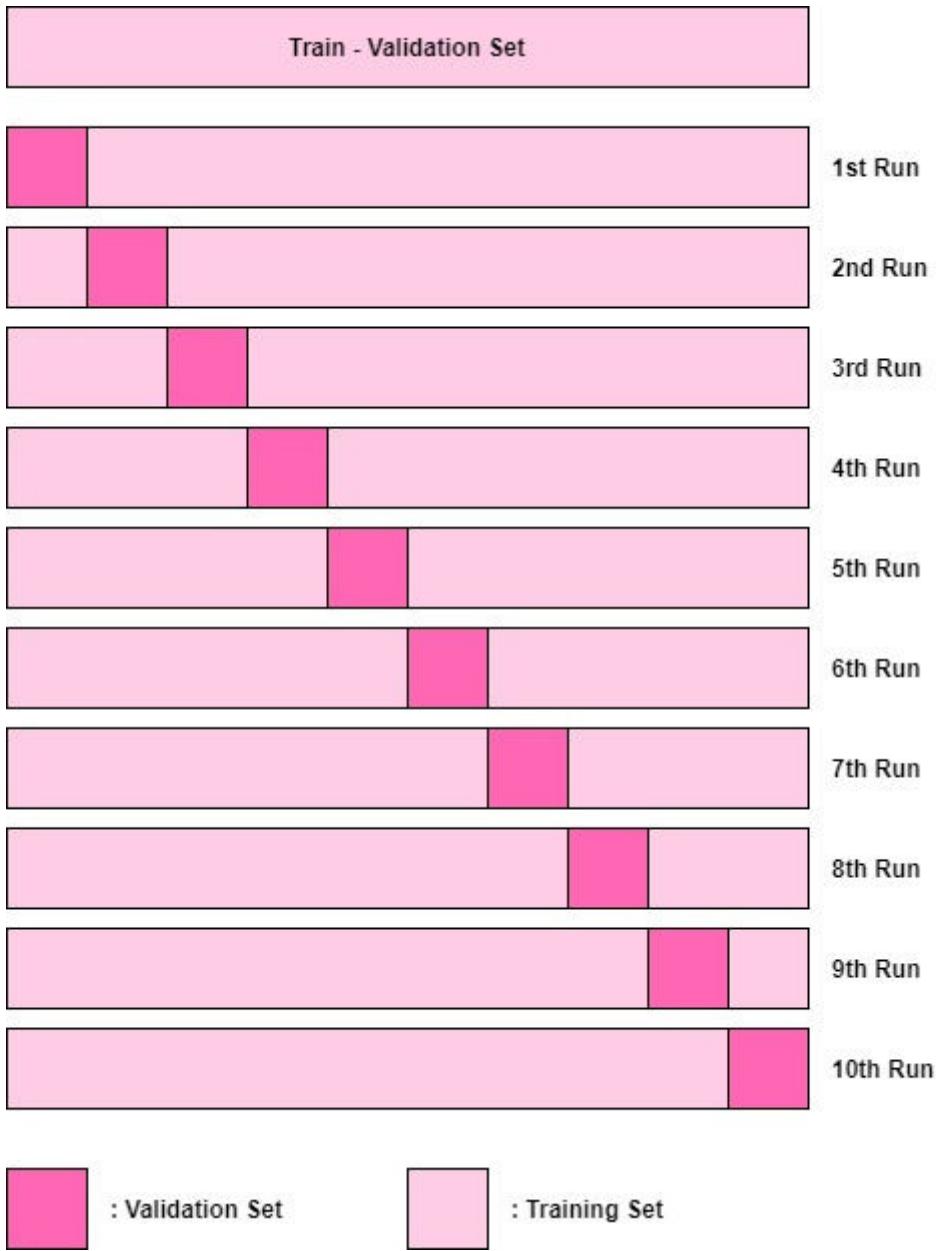
Leave-One-Out cross validation

K-folds Cross Validation

- Dataset akan kita bagi menjadi k bagian (fold) yang memiliki jumlah sampel data sama persis dan tidak saling overlap
- Untuk setiap kali *run the training process* (menjalankan proses pembelajaran) maka kita tetapkan satu bagian sebagai *validation set* dan $(k - 1)$ bagian yang lainnya sebagai *training set*
- Proses tersebut diulang sampai sebanyak k kali, sehingga semua bagian pernah berperan sebagai *validation set*

Proses pemilihan data untuk masing-masing fold dilakukan menggunakan random *random sampling*.

Illustration 10-folds CV



Stratified K-folds Cross Validation

- Prinsipnya sama dengan *K-Folds Cross Validation*, hanya saja perbedaannya pada proses pemilihan data untuk masing-masing fold dilakukan menggunakan *stratified sampling*.

(lanjutan)

Misalkan:

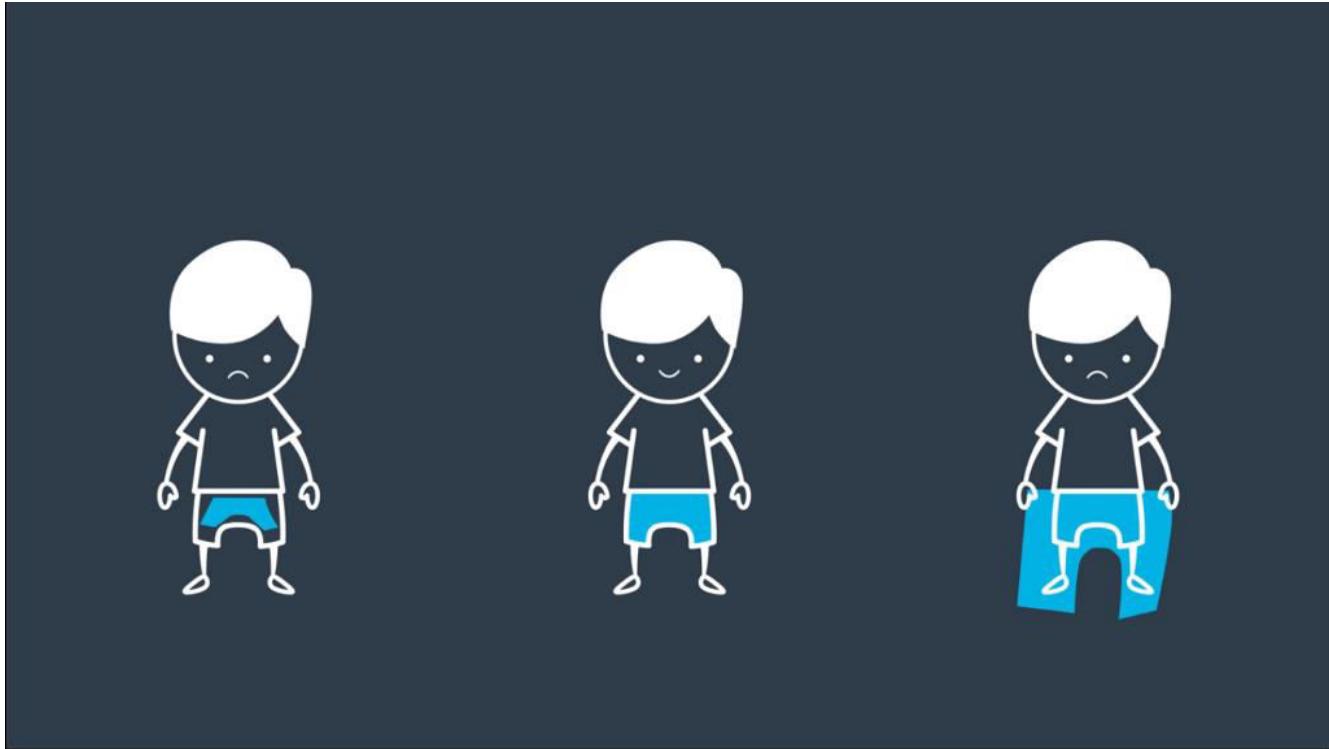
- Dataset train-validation sebanyak 100 buah sampel data yang terdiri dari 80 data kelas negatif (80%) dan 20 data kelas positif (20%)
- Kita terapkan 5-folds cross validation, maka setiap fold akan terdiri dari 20 sampel data
- Maka setiap folds terdiri dari:
 - Kelas negative = $80\% \times 20 = 16$ sampel data
 - Kelas positif = $20\% \times 20 = 4$ sampel data
- Satu kali running eksperimen:
 - Training set : 64 sampel data negative, 16 sampel data positif
 - Validation set : 16 sampel data negative, 4 sampel data positif

Leave-One-Out Cross Validation

- Merupakan bentuk khusus dari *K-folds cross validation*, dimana $K = n$, untuk n adalah total jumlah data
- Dilakukan n eksperimen dimana *training set* terdiri dari $n - 1$ samples data serta 1 buah sampel data untuk *validation set*.
- *Computationally expensive*
- *Leave-one-out cross-validation* tidak menjamin distribusi sampel data yang sama untuk masing-masing kelas

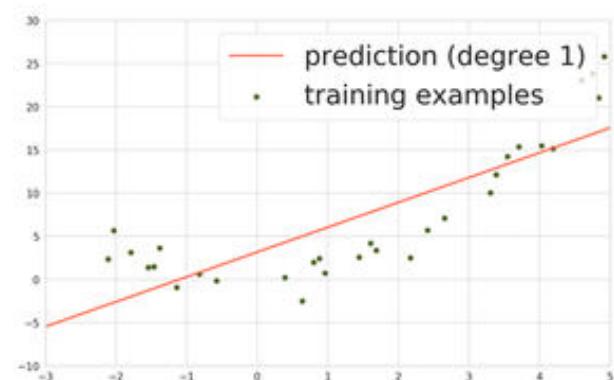


**Why we need to
evaluate the
performance of
classifier?**



Underfitting vs Good Fit vs Overfitting

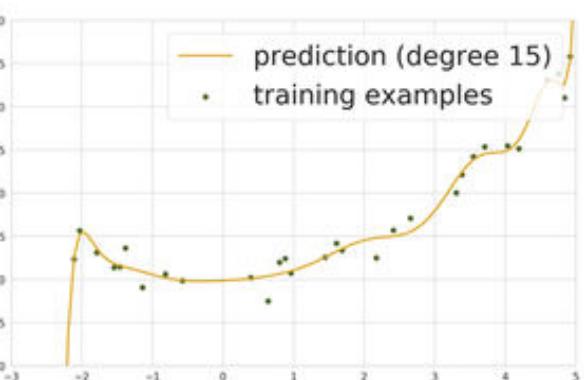
Underfit



Good Fit



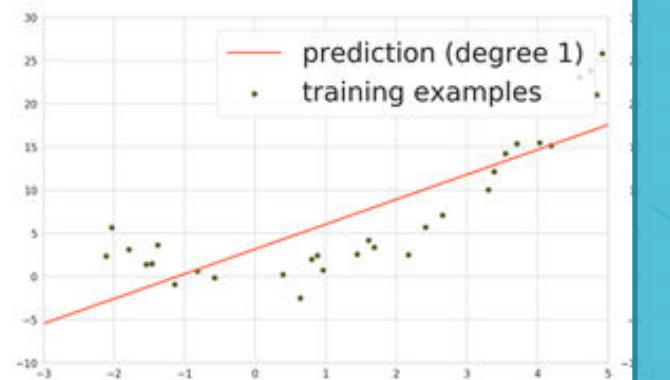
Overfit



Types of Model Fit

Underfit

Underfit



-----Good Fit-----

The model is too simple (tidak dapat menangkap pola yang tersirat dalam data) and is not able to predict values accurately at all (training data or validation/testing data)

Overfit

Types of Model Fit

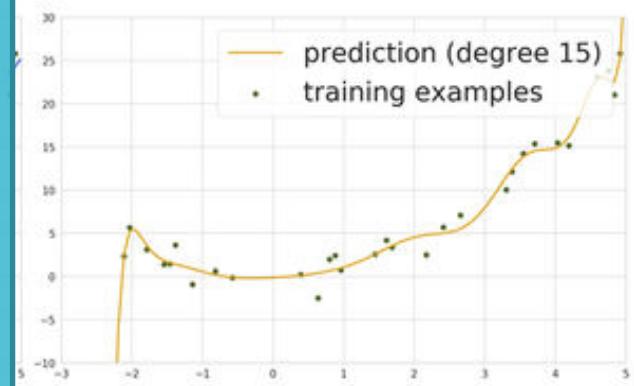
Underfit

Good Fit

Overfit

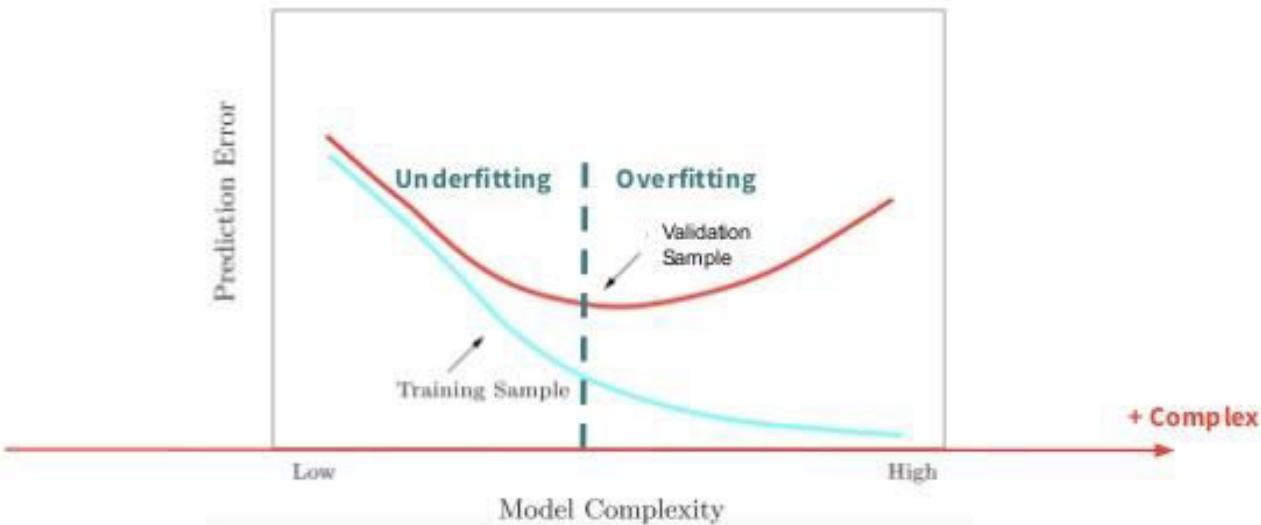
The model has not only learned from the signal but is also effected by noise

Overfit



Types of Model Fit

Underfitting and Overfitting

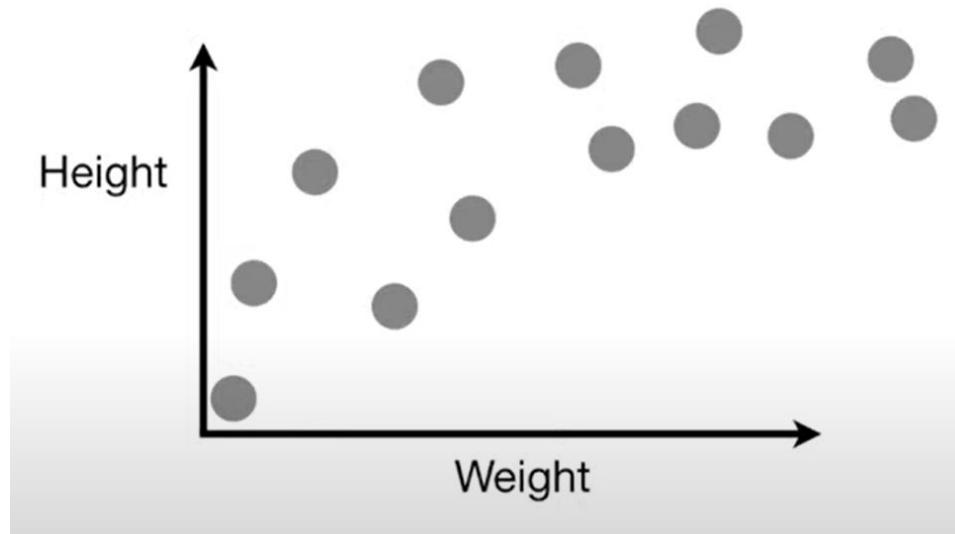


Model Complexity Graph

Bias & Variance in Machine Learning

Example

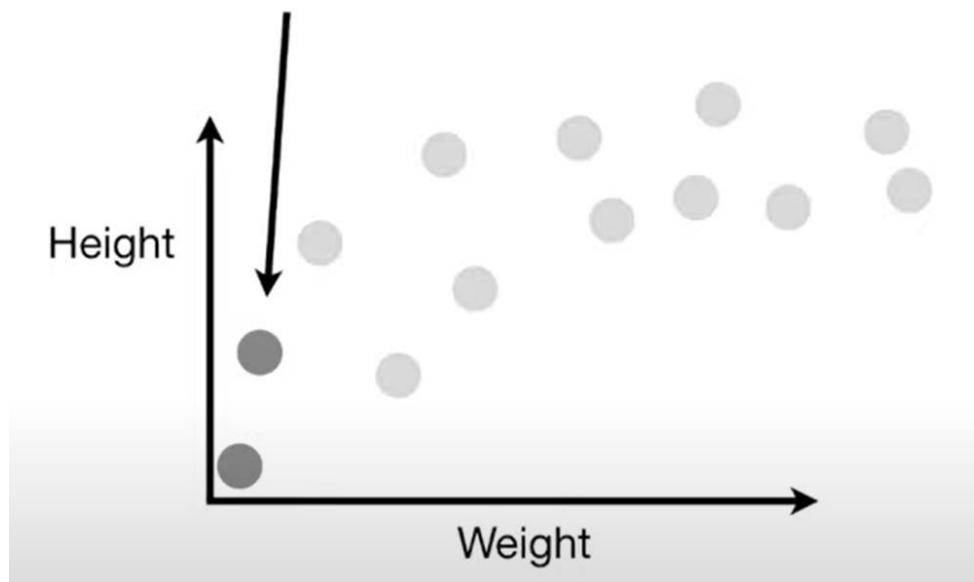
Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus



Example

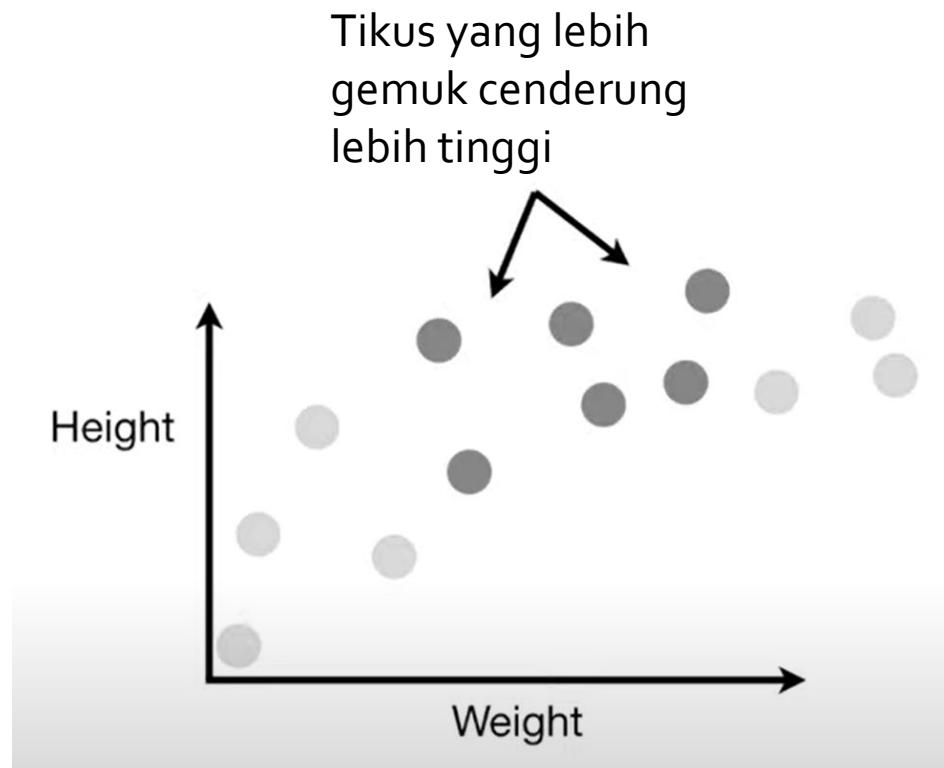
Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

Tikus yang kurus
cenderung pendek



Example

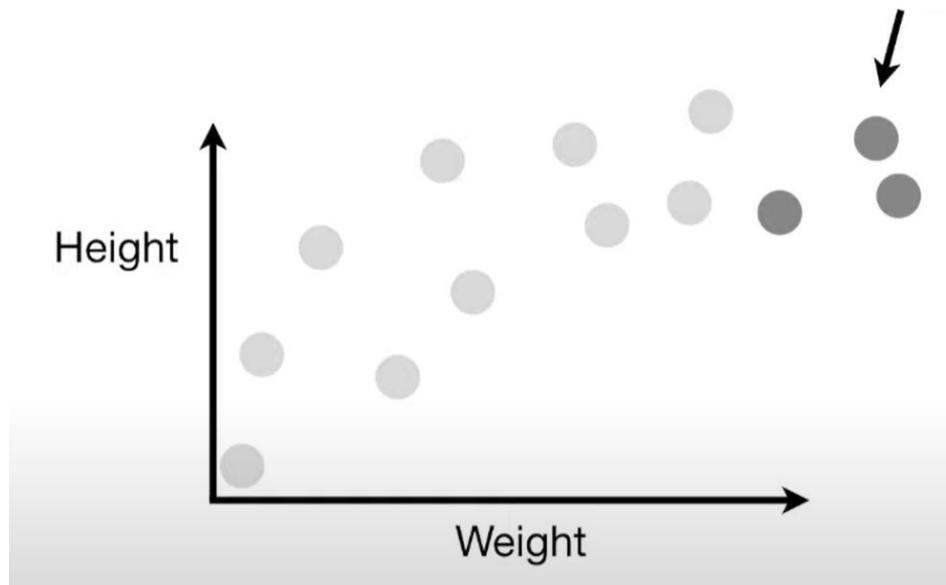
Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

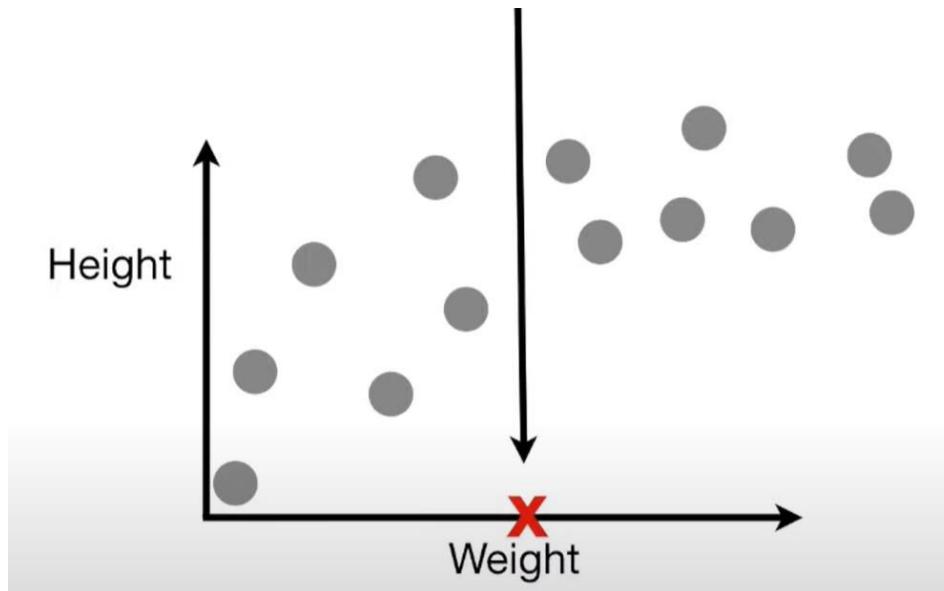
Pada berat tertentu, tikus-tikus tidak bertambah tinggi hanya mengalami obesitas



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

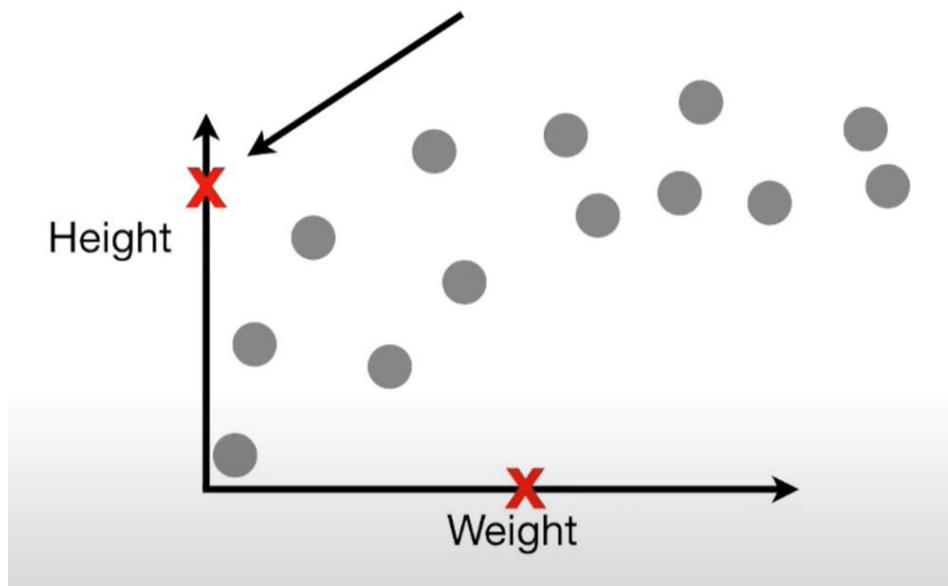
Misalkan diketahui berat badan sebuah tikus adalah titik berikut ini,



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

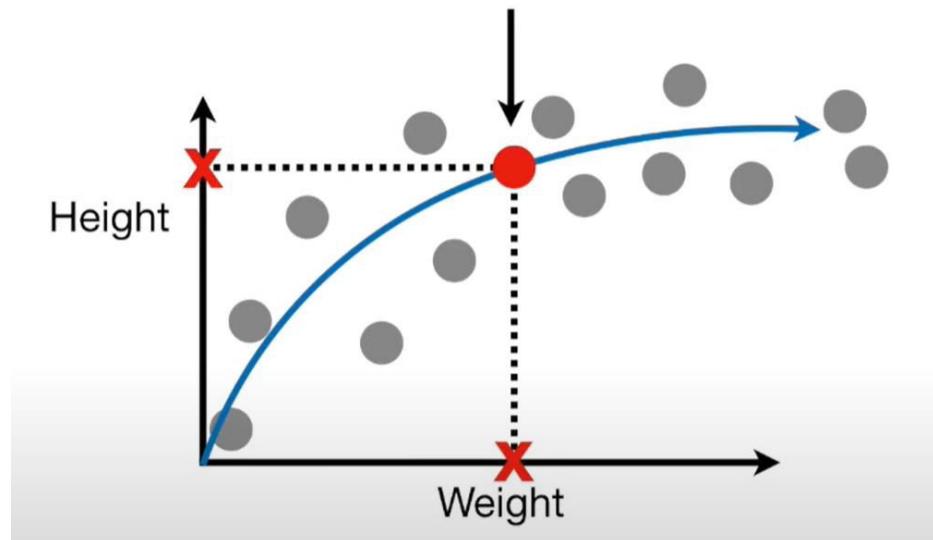
... , maka kita bisa memprediksi tingginya adalah sebagaimana ditunjukkan oleh titik berikut



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

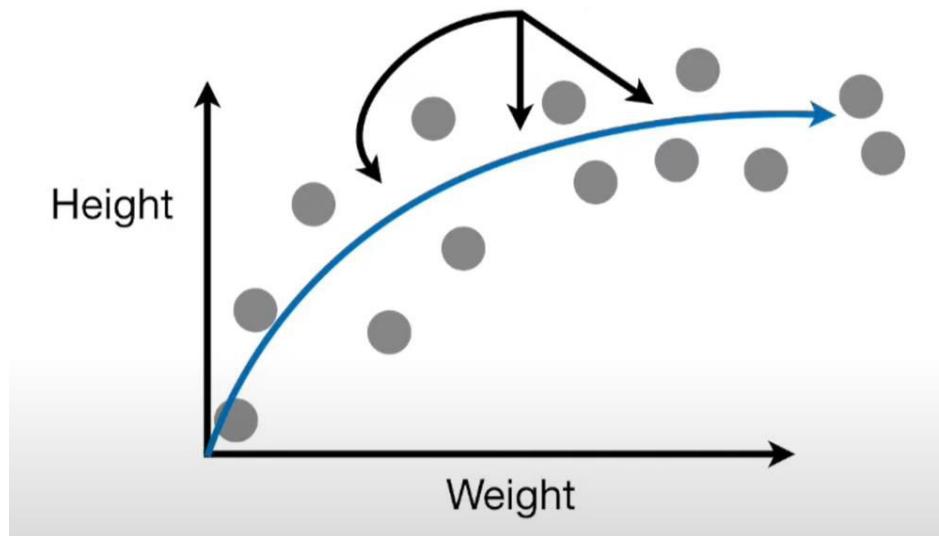
Idealnya kita tahu rumus matematisnya secara pasti yang mendeskripsikan hubungan antara berat dan tinggi



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

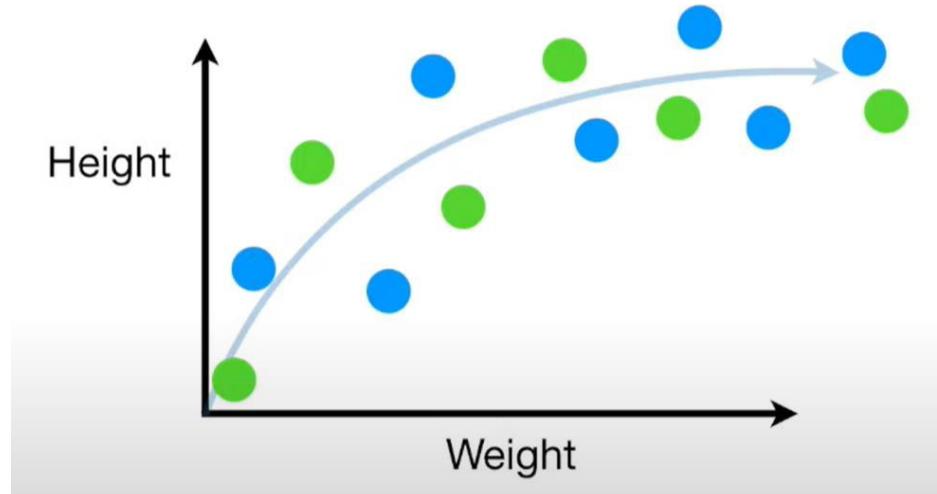
Tetapi pada kenyataannya kita tidak memiliki rumus pastinya tersebut, sehingga kita dapat menerapkan suatu algoritma machine learning untuk mengaproksimasi bagaimana hubungan antara berat dan tinggi tersebut?



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

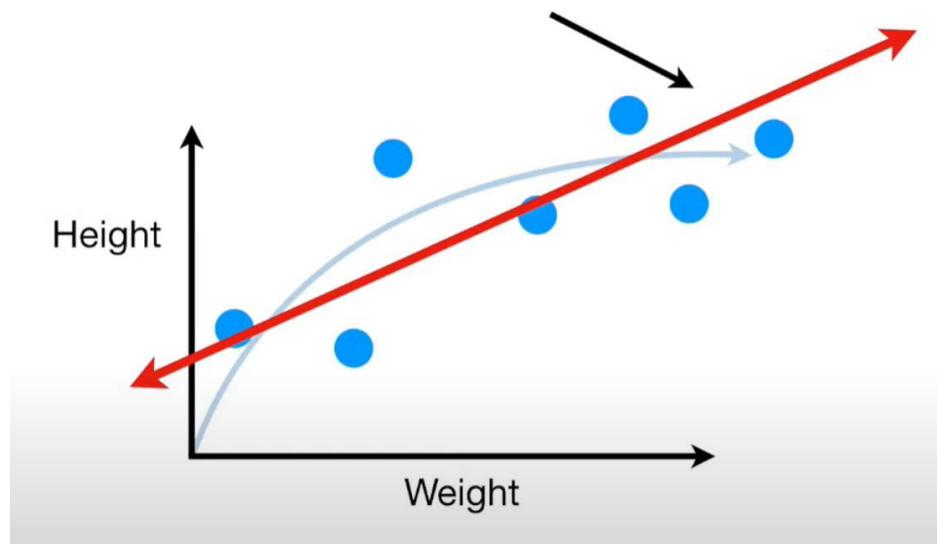
Misalkan sampel data tersebut terdiri dari dua subset, yakni training set berwarna biru, dan testing set berwarna hijau



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

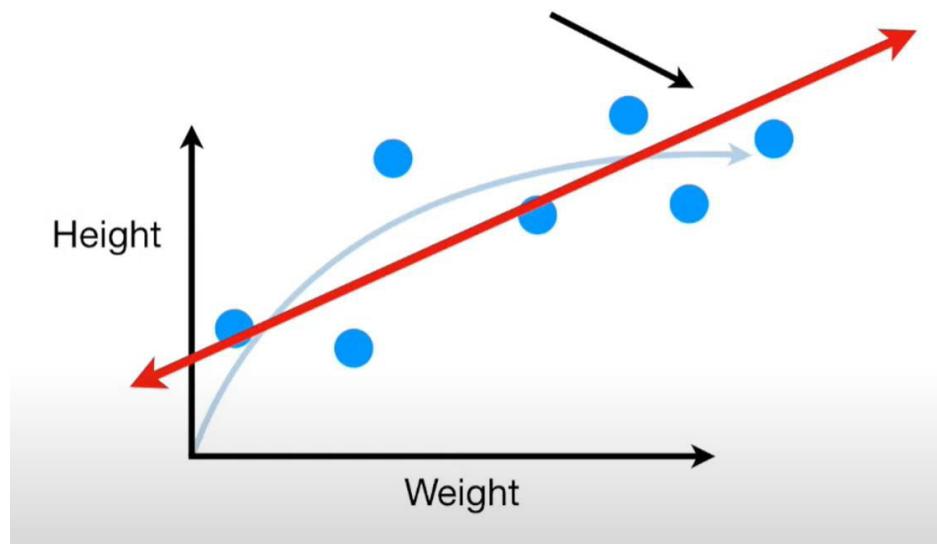
Jika sekarang hanya ada *training set* maka kita bisa membangun model prediksi salah satunya dengan *linear regression*



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

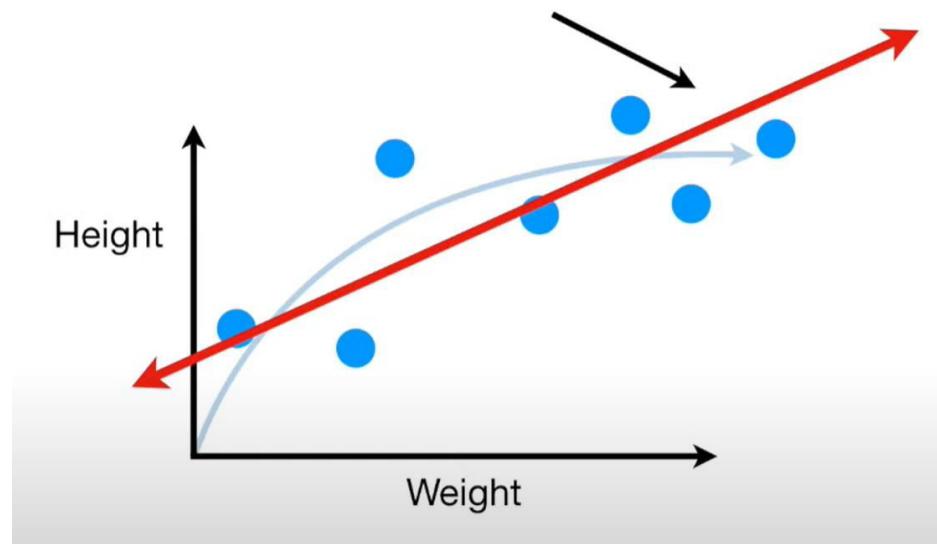
Linear regression menggunakan garis lurus untuk sebagai modelnya, dimana bentuk garis lurus ini tidak fleksibel untuk secara akurat dapat menggambarkan hubungan antara weight dan height seperti ditunjukkan dengan garis lengkung biru



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

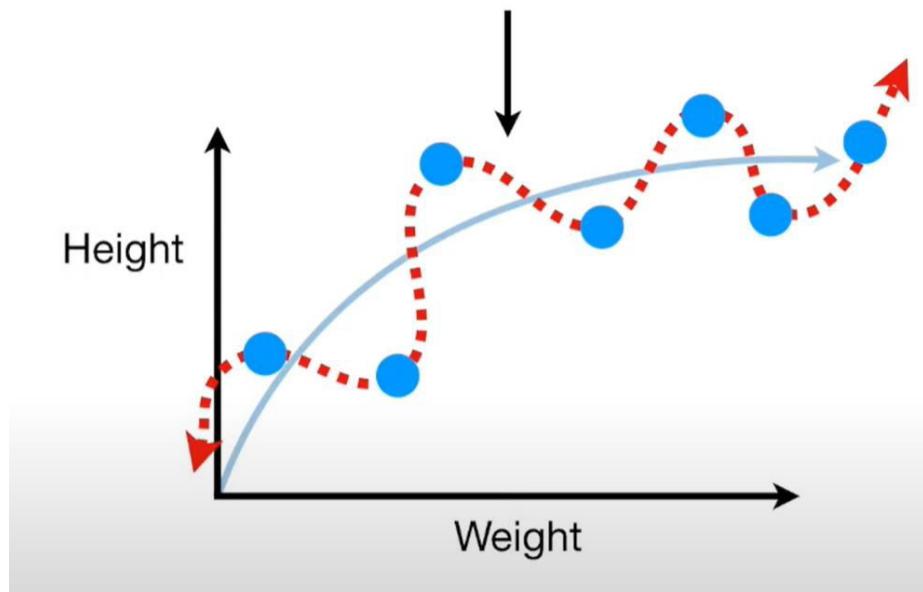
Ketidakmampuan metode/algoritma machine learning untuk menangkap hubungan yang sebenarnya tersebut disebut sebagai **bias**, yaitu metode/algoritma tersebut dikatakan memiliki bias yang tinggi.



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

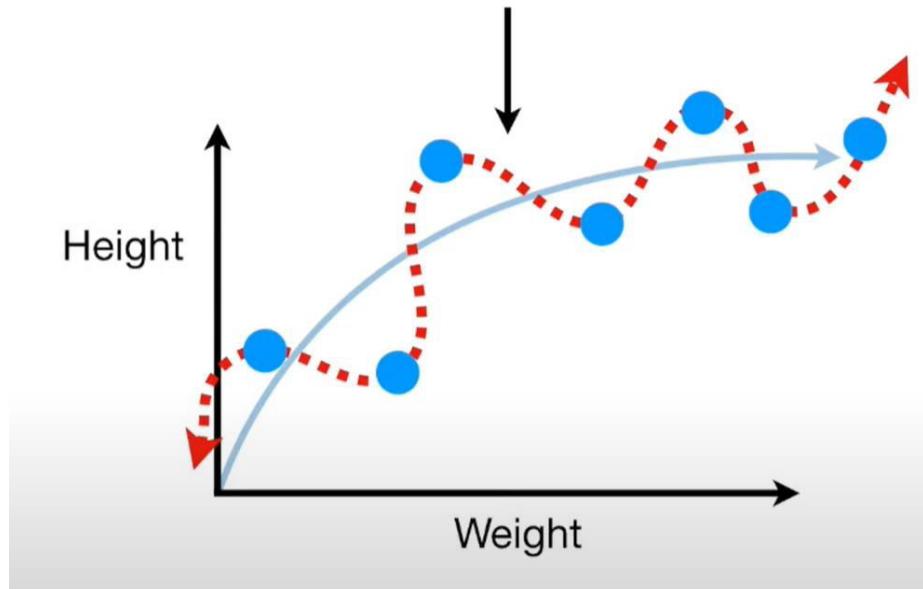
Metode/algoritma *machine learning* yang lainnya mungkin menghasilkan garis berkelok-kelok sebagai modelnya



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

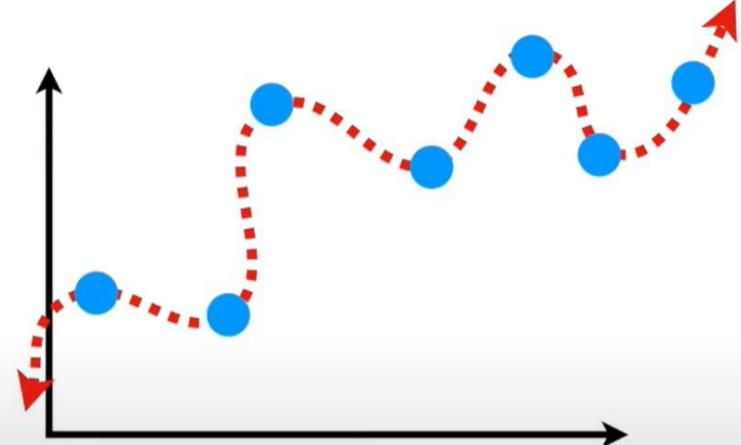
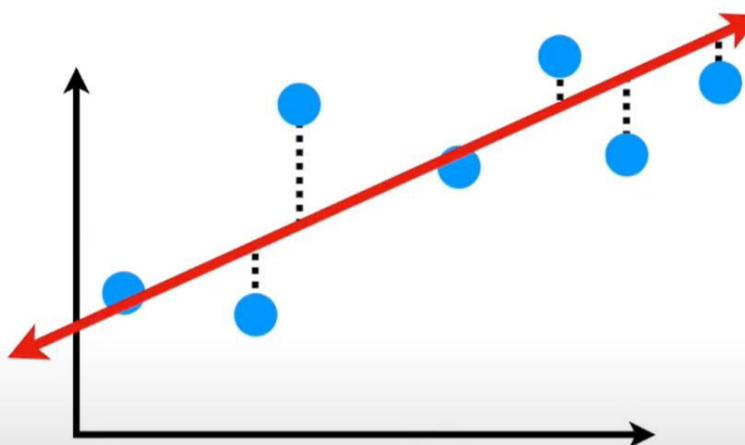
Bentuk garis ini sangat fleksibel dan dapat menggambarkan hubungan antara weight dan height dengan sangat baik untuk *training set* (titik-titik biru pada gambar) atau disebut memiliki nilai **bias** yang rendah.



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

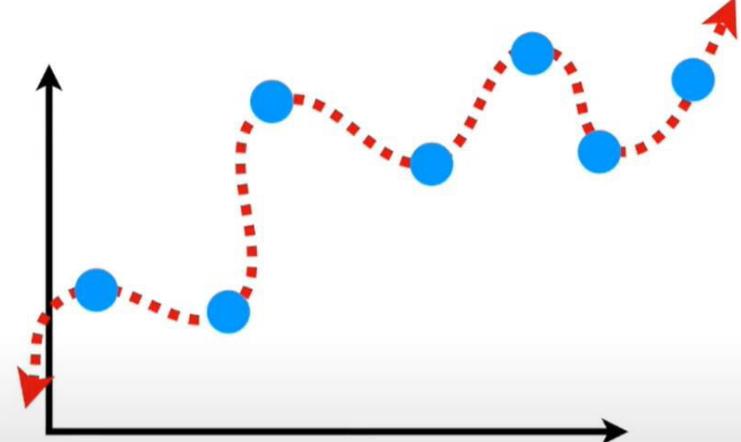
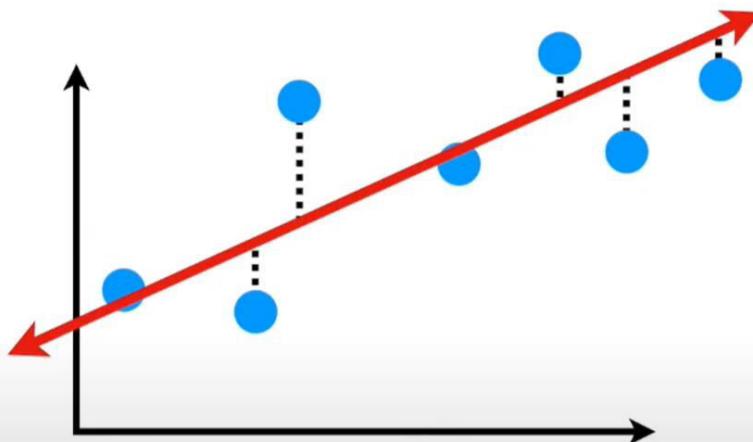
Untuk menentukan seberapa baik kedua model tersebut (fit/cocok) untuk *training set* maka dapat dilakukan dengan menghitung **sum of square errors (SSE)**.



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

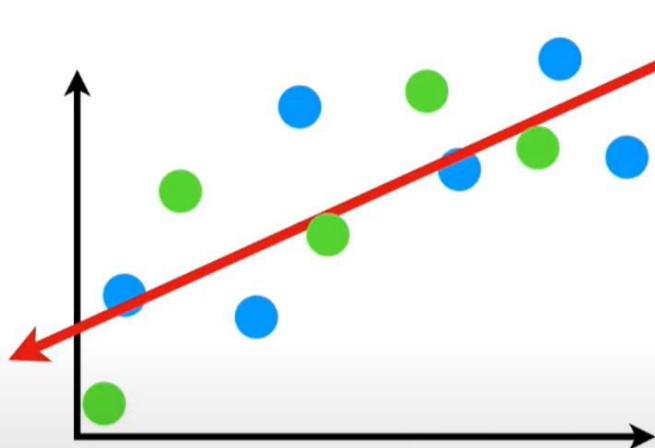
Pada model yang kanan maka bisa diperoleh nilai $SSE = 0$ untuk *training set*



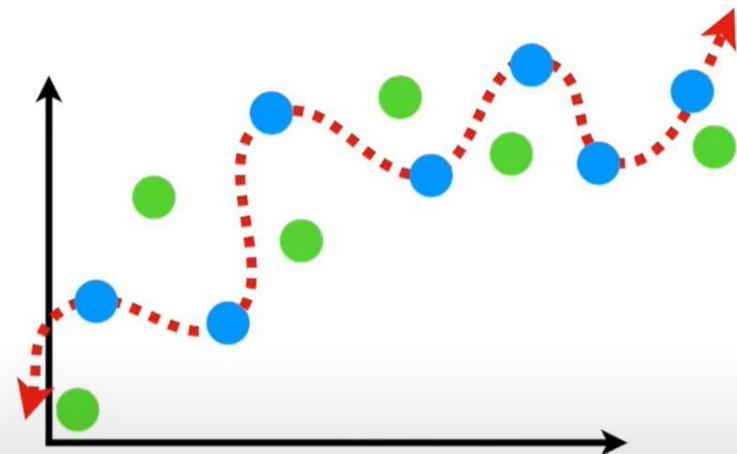
Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

Akan tetapi kita tidak hanya memiliki *training set* saja tetapi kita juga memiliki *testing set* berupa titik hijau



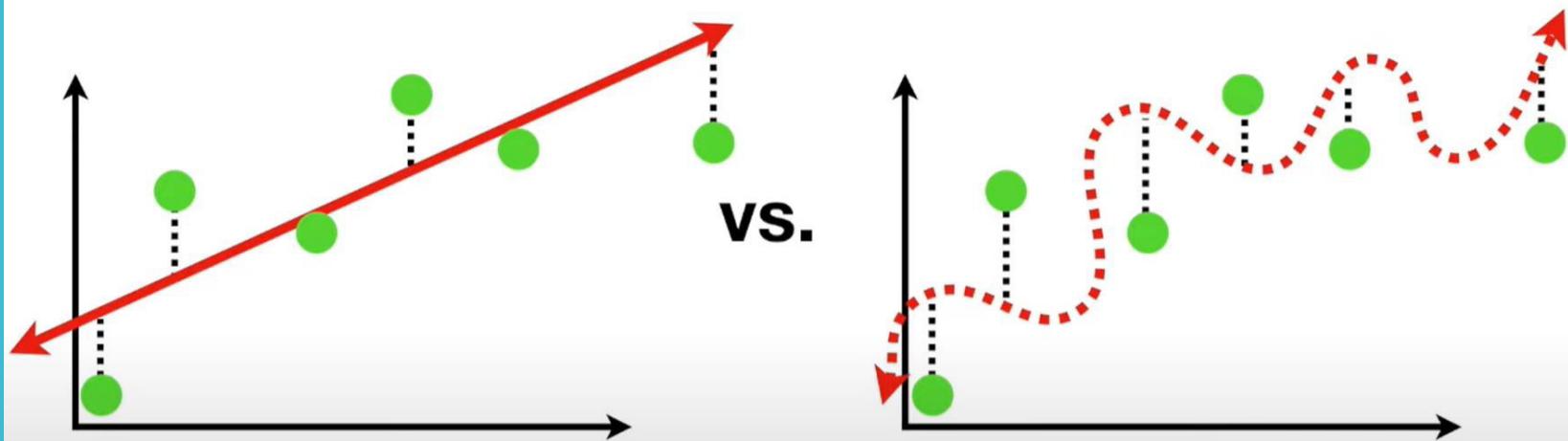
vs.



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

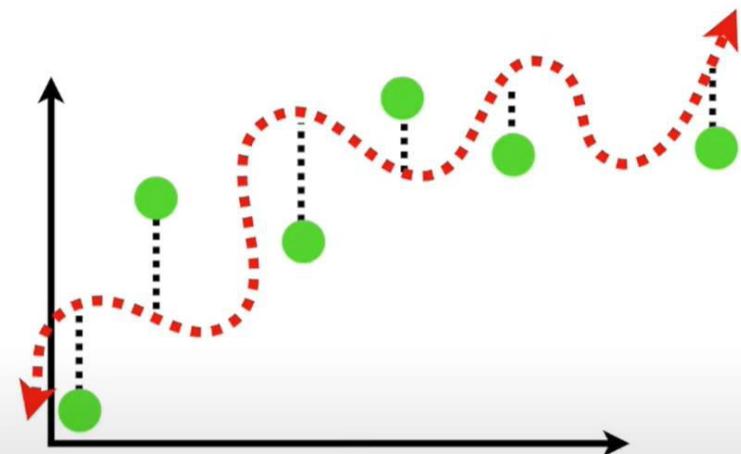
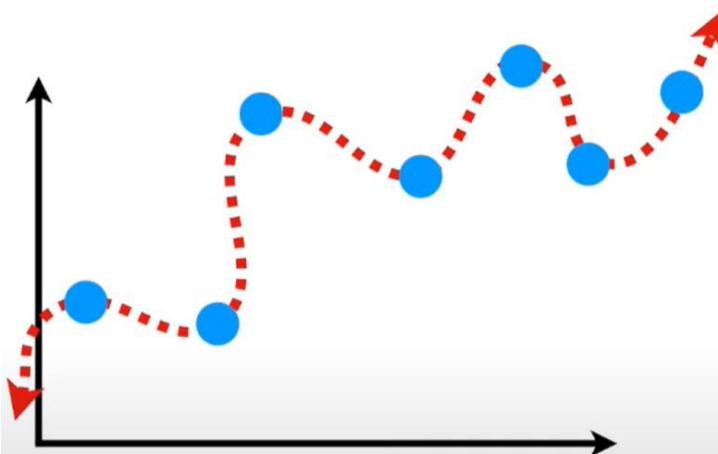
Apabila kita hitung nilai SSE terhadap *testing set* maka model sebelah kiri lebih bagus dengan nilai SSE yang lebih kecil



Example

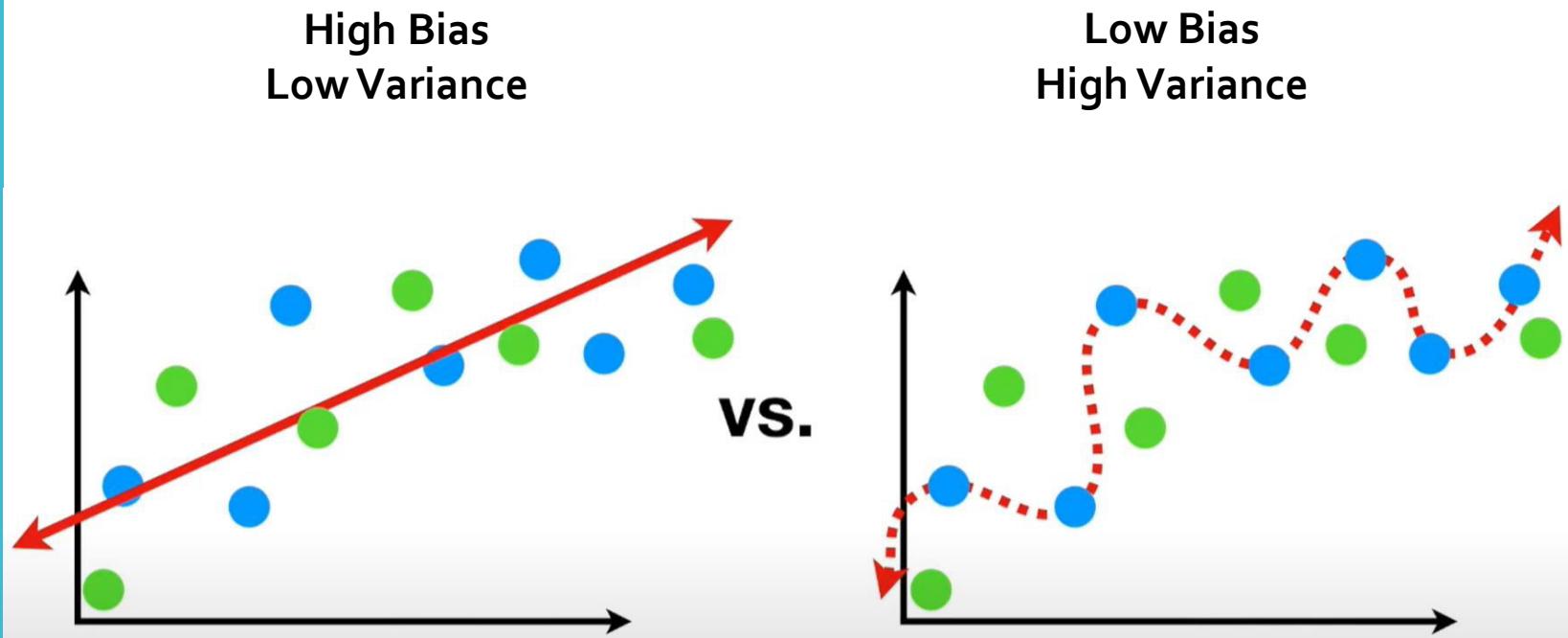
Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus

Dalam machine learning, perbedaan kemampuan suatu model untuk fit/cocok terhadap suatu dataset disebut sebagai **variance**. Pada contoh di sisi kanan, model tersebut dikatakan memiliki **variance yang tinggi**



Example

Perhatikan jika kita memiliki data berupa grafik yang menunjukkan berat dan tinggi dari tikus



Bagaimana idealnya Good Machine Learning Model?

**Low Bias
Low Variance**

Solusi: **Regularization, Boosting, Bagging**

Confusion Matrix

Confusion Matrix

- Matriks yang digunakan untuk meringkas data *correct/incorrect classification* yang dihasilkan oleh sebuah metode/model klasifikasi untuk suatu dataset.
- Baris dan kolom dari *confusion matrix* terkait dengan informasi *true classes* (*gold standard* / *actual class* / kelas yang sebenarnya) serta *predicted classes* (kelas yang diprediksi).
- Bukan termasuk performance metrics, tetapi merupakan representasi yang dapat digunakan sebagai acuan untuk menghitung performance metrics

Performance Metrics

Performance Metrics

1. **Accuracy** : proporsi data yang terklasifikasi dengan benar

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. **Error rate** : proporsi data yang tidak terklasifikasi dengan benar

$$error_rate = \frac{FP + FN}{TP + TN + FP + FN}$$

3. **Recall (Sensitivity / TPR)** : proporsi data dari kelas positif terklasifikasi sebagai kelas positif

$$TPR = \frac{TP}{TP + FN}$$

4. **Precision (Positive Predictive Value / PPV)**: proporsi data terprediksi sebagai kelas positif yang benar

$$precision = \frac{TP}{TP + FP}$$

Performance Metrics (2)

5. F-Measure

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

6. FPR (False Positive Rate) / False Alarm

$$FPR = \frac{FP}{FP + TN}$$

7. FNR (False Negative Rate)

$$FNR = \frac{FN}{FN + TP}$$

8. Specificity

$$Specificity = \frac{TN}{FP + TN} = 1 - FPR$$

Performance Evaluation for Binary Classification

Example

No	Actual Class	Predicted Class
1	Positif	Negatif
2	Negatif	Negatif
3	Positif	Positif
4	Positif	Positif
5	Negatif	Negatif
6	Negatif	Positif
7	Positif	Positif
8	Positif	Positif
9	Positif	Negatif
10	Negatif	Negatif

Look at The
Blackboard !!!

Hasil Eksperimen sebuah Algoritma Klasifikasi

Contoh Soal

Data Testing	Actual Class	Predicted Class
1	C1	C1
2	C1	C1
3	C1	C1
4	C1	C1
5	C1	C2
6	C2	C1
7	C2	C1
8	C2	C2

2 × 2 Confusion Matrix dari Hasil Eksperimen di Samping

		Predicted Class		$C = 5$
		C1	C2	
Actual Class	C1	4	1	$D = 3$
	C2	2	1	
		$A = 6$	$B = 2$	$T = 8$

$$TP = 4$$

$$FP = 1$$

$$TN = 1$$

$$FN = 2$$

$$accuracy = \frac{4 + 1}{8} = \frac{5}{8} = 0.625 = 62.5\%$$

$$error = \frac{2+1}{8} = \frac{3}{8} = 0.375 = 37.5\%$$

Performance in Unequal Importance of Classes

Performance in Unequal Importance of Classes

- Apabila terdapat dua kelas yang memiliki dua kelas yang memiliki tingkat kepentingan yang berbeda (*assymmetric*)
 - Lebih penting untuk memprediksi suatu data ke dalam kelas
 - Kelas 1 dibandingkan Kelas 2
 - Kelas + daripada Kelas –
- Accuracy bukan merupakan ukuran kinerja yang bagus untuk mengevaluasi sebuah pengklasifikasi
 - Gunakan sensitivity, specificity, FPR, FNR
- Misalnya untuk memprediksi status keuangan sebuah perusahaan dikatakan akan bangkrut atau tidak
 - Lebih penting untuk memprediksi bahwa suatu perusahaan mengalami kebangkrutan daripada tetap diprediksi sebagai perusahaan normal
- Tentukan data TP sebagai kondisi klasifikasi yang paling penting

Sensitivity

$$Sensitivity = TPR = \frac{TP}{TP+FN} = \frac{\text{Jumlah True Positive}}{\text{Jumlah Data Positive}}$$

Sensitivitas mengacu pada kemampuan tes untuk mendeteksi dengan benar pasien sakit yang memang memiliki kondisi tersebut.

- Tes dengan sensitivitas tinggi dapat diandalkan ketika hasilnya negatif karena jarang salah mendiagnosis mereka yang memiliki penyakit tersebut.
- Tes dengan sensitivitas 100% akan mengenali semua pasien dengan penyakit sebagai hasil tes positif.
 - Hasil tes negatif akan secara definitif menyatakan pasien tidak menderita penyakit terkait.

Sensitivity (cont'd)

- Namun, hasil positif dalam tes dengan sensitivitas tinggi tidak selalu berguna untuk menentukan seseorang tersebut benar-benar menderita penyakit terkait.
 - Misalkan alat tes 'palsu' dirancang untuk selalu memberikan hasil positif. Ketika digunakan pada pasien yang sakit, semua pasien dites positif, memberikan tes sensitivitas 100%.
 - Sensitivitas tidak memperhitungkan positif palsu (False Positive).
 - Tes palsu juga mengembalikan positif pada semua pasien yang sehat, memberikan tingkat positif palsu 100%, menjadikannya tidak berguna untuk mendeteksi atau "mengesampingkan" penyakit.

Specificity

$$Specificity = TNR = \frac{TN}{TN+FP} = \frac{\text{Jumlah True Negative}}{\text{Jumlah Data Negative}}$$

Spesifisitas berkaitan dengan kemampuan tes untuk mendeteksi pasien sehat dengan benar.

- Test dengan spesifisitas tinggi dapat diandalkan ketika hasilnya positif karena tes jarang memberikan hasil positif pada pasien yang sehat.
- Tes dengan spesifisitas 100% akan mengenali semua pasien sehat / tanpa penyakit dengan tes negative.
 - Hasil tes positif pasti akan menentukan adanya penyakit.

Specificity (cont'd)

- Namun, hasil negatif dari tes dengan spesifisitas tinggi tidak selalu berguna untuk menyingkirkan penyakit.
 - Misalnya, tes yang selalu mengembalikan hasil tes negatif akan memiliki spesifisitas 100%
 - Spesifisitas tidak mempertimbangkan negatif palsu (False Negative)
 - Tes seperti itu akan kembali negatif untuk pasien dengan penyakit, sehingga tidak berguna untuk mengesampingkan penyakit.

NOTE:

- A test with a higher sensitivity has a lower type II error rate (False Negative).
- A test with a higher specificity has a lower type I error rate (False Positive).

Contoh Soal

Data hasil klasifikasi citra *diabetic retinopathy* menunjukkan data *confusion matrix* sebagai berikut:

		Predicted Class				
		Normal	Abnormal			
Actual Class	Normal	4	1	C = 5	False Positive	
	Abnormal	2	1	D = 3	True Positive	
		A = 6	B = 2	T = 8		

True Negative
False Negative

$$sensitivity = \frac{TP}{TP + FN} = \frac{1}{1 + 2} = \frac{1}{3} = 33.3\%$$

$$specificity = \frac{TN}{FP + TN} = \frac{4}{1 + 4} = \frac{4}{5} = 80\%$$

$$FPR = \frac{FP}{FP + TN} = \frac{1}{1 + 4} = \frac{1}{5} = 20\%$$

Performance Evaluation for Multiclass Classification

Example

No	Actual Class	Predicted Class	No	Actual Class	Predicted Class
1	Ayam	Ayam	11	Kucing	Kucing
2	Kucing	Kucing	12	Ayam	Ayam
3	Ikan	Ayam	13	Ayam	Kucing
4	Kucing	Ayam	14	Kucing	Ikan
5	Ayam	Kucing	15	Ikan	Ikan
6	Ayam	Ayam	16	Ayam	Kucing
7	Kucing	Kucing	17	Ayam	Ayam
8	Ikan	Ikan	18	Ikan	Ayam
9	Ayam	Ayam	19	Ikan	Kucing
10	Kucing	Kucing	20	Ayam	Ayam

Example

		Predicted Class		
		Kucing	Ikan	Ayam
Actual Class	Kucing	4	1	1
	Ikan	1	2	2
	Ayam	3	0	6

Multi-level Confusion Matrix

Matriks $n \times n$, dimana n adalah jumlah kelas serta data (i, j) merepresentasikan jumlah elemen dari kelas i dan terprediksi sebagai kelas j

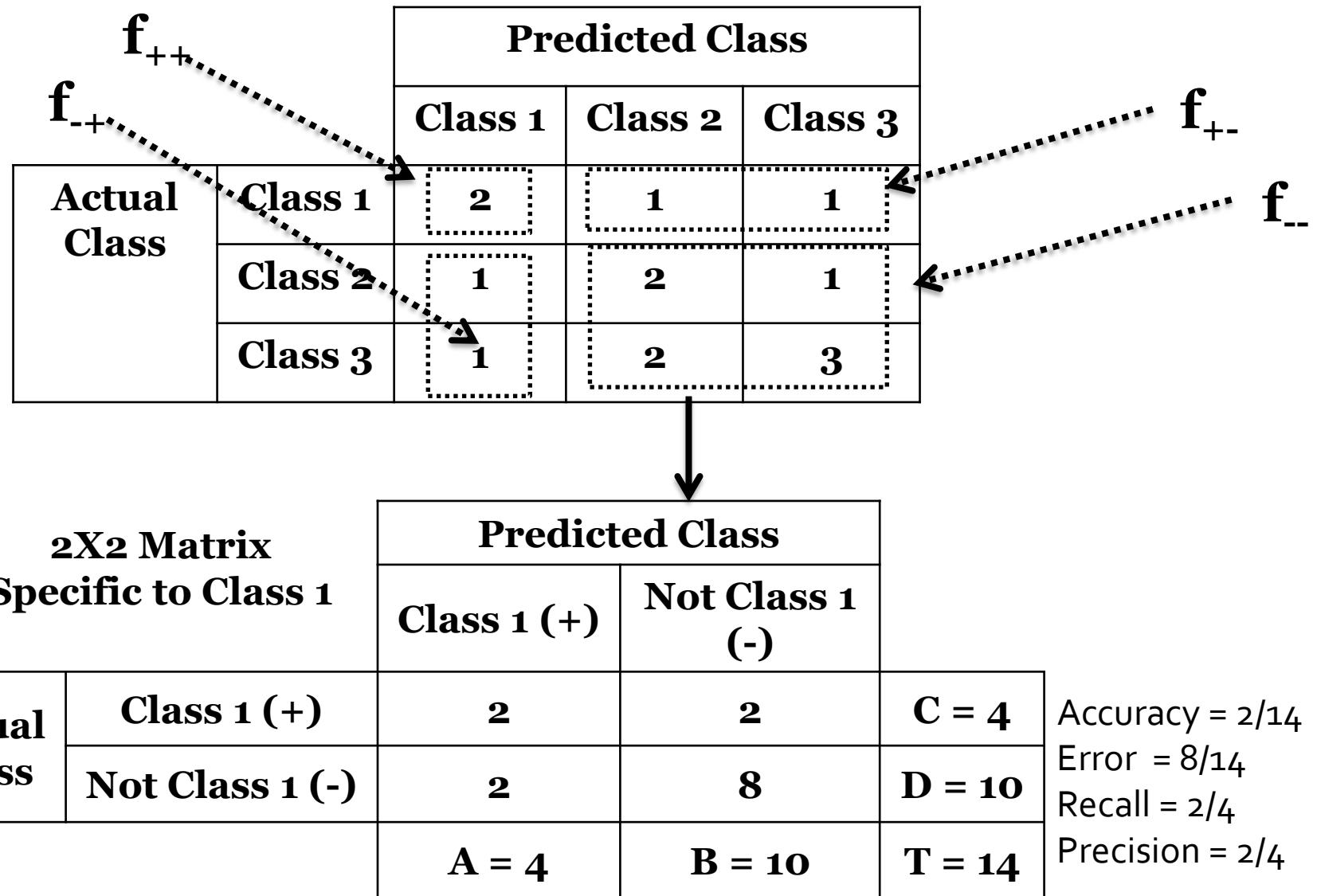
Multi-level Confusion Matrix		Predicted Class				Marginal Sum of Actual Values
		Class 1	Class 2	----	Class N	
Actual Class	Class 1	f_{11}	f_{12}	----	f_{1N}	$\sum_{j=1}^N f_{1j}$
	Class 2	f_{21}	f_{22}	----	f_{2N}	$\sum_{j=1}^N f_{2j}$

	Class N	f_{N1}	f_{N2}	----	f_{NN}	$\sum_{j=1}^N f_{Nj}$
Marginal Sum of Predictions		$\sum_{i=1}^N f_{i1}$	$\sum_{i=1}^N f_{i2}$	----	$\sum_{i=1}^N f_{iN}$	$T = \sum_{i=1}^N \sum_{j=1}^N f_{ij}$

Contoh - Multi-level Confusion Matrix

		Predicted Class			Marginal Sum of Actual Values
		Class 1	Class 2	Class 3	
Actual Class	Class 1	2	1	1	4
	Class 2	1	2	1	4
	Class 3	1	2	3	6
Marginal Sum of Predictions		4	5	5	T = 14

Konversi Multi-level Confusion Matrix ke Bentuk 2×2





ROC & AUC

Accuracy Measure - 2

- Berdasarkan Receiver Operating Characteristics (ROC)
 - Area under the ROC curve (AUC)
- Banyak digunakan pada bidang Medical Diagnosis
- Dominance Relationship:
 - A ROC curve A dominates another ROC curve B if A is always above and to the left of B in the plot

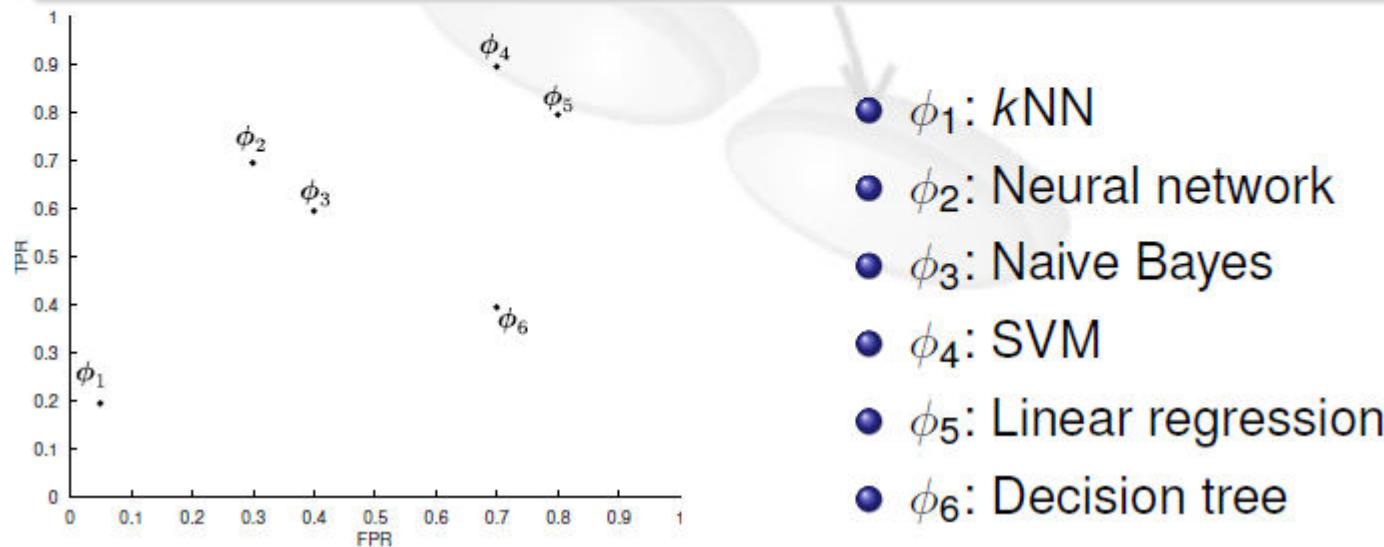
Source :

Lozano, J. A., Santafé, G., & Inza, I. (2010). *Classifier performance evaluation and comparison* (pp. 48–56).

Receiver Operating Characteristics (ROC)

ROC Space

Coordinate system used for visualizing classifiers performance where TPR is plotted on the Y axis and FPR is plotted on the X axis.



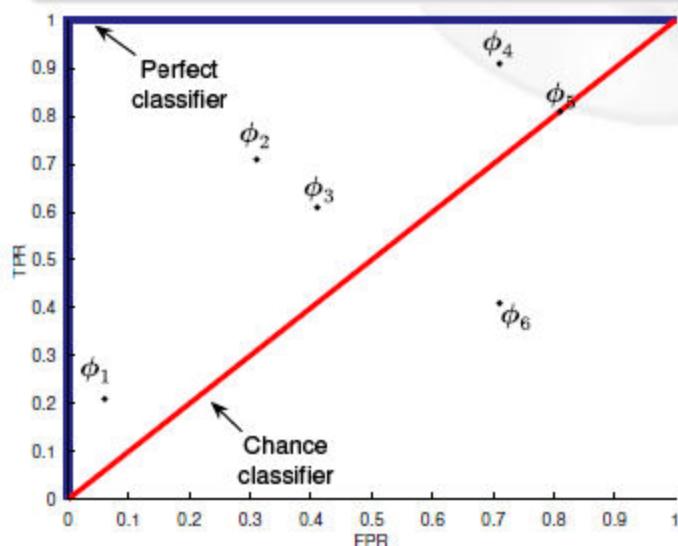
$$TPR = \text{Sensitivity}$$

$$FPR = 1 - \text{Specificity}$$

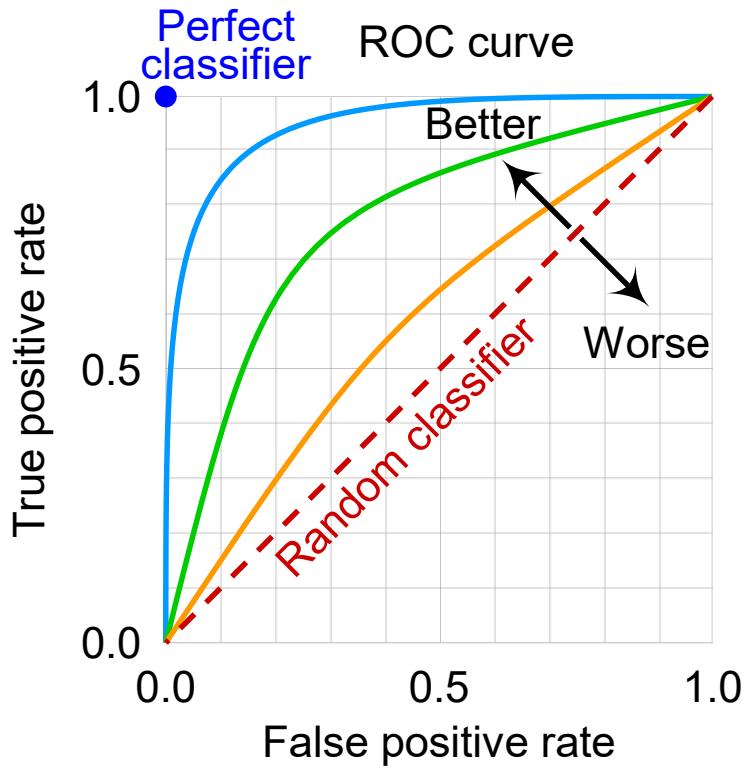
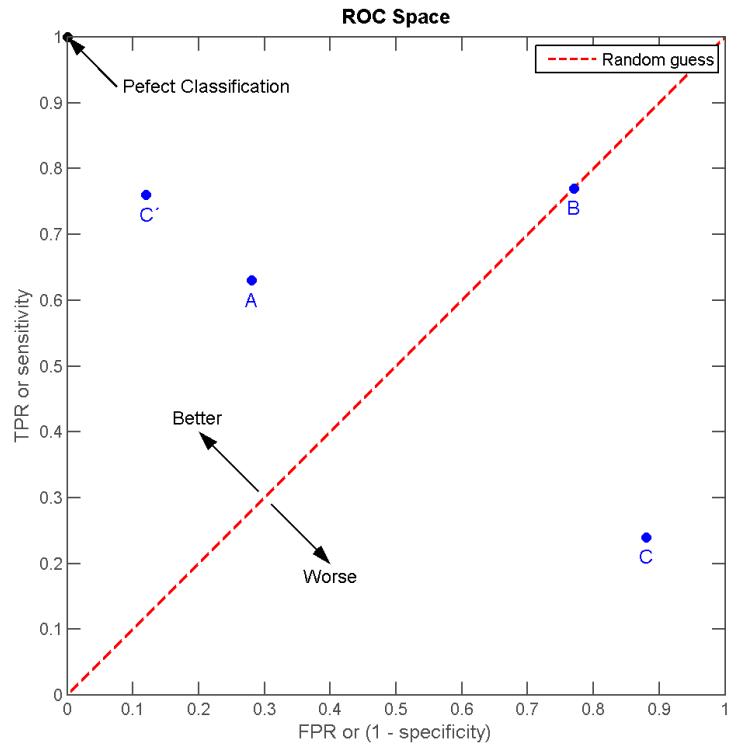
Receiver Operating Characteristics (ROC)

ROC Space

Coordinate system used for visualizing classifiers performance where TPR is plotted on the Y axis and FPR is plotted on the X axis.



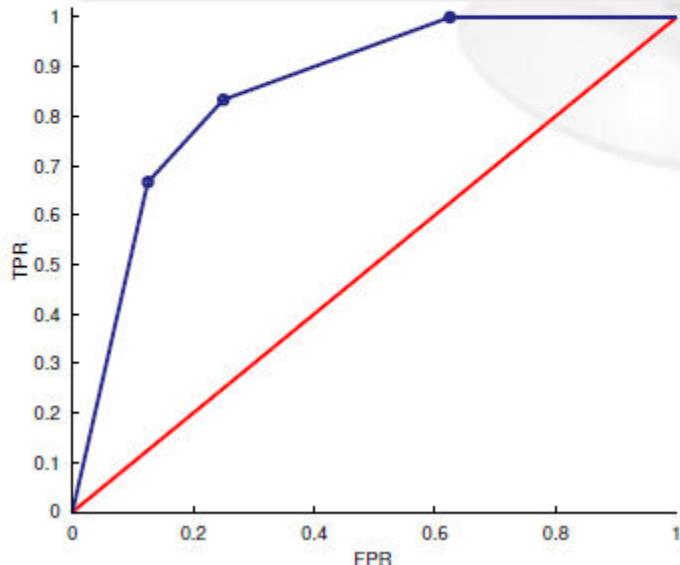
- ϕ_1 : kNN
- ϕ_2 : Neural network
- ϕ_3 : Naive Bayes
- ϕ_4 : SVM
- ϕ_5 : Linear regression
- ϕ_6 : Decision tree



Receiver Operating Characteristics (ROC)

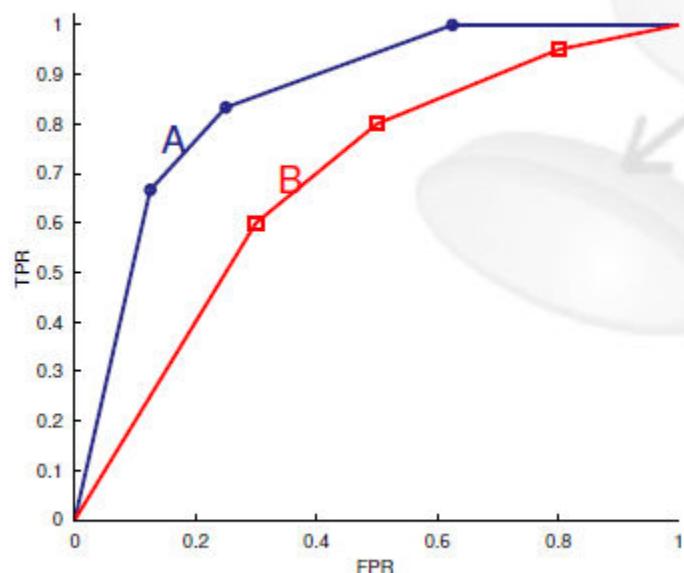
ROC Curve

For a probabilistic/fuzzy classifier, a ROC curve is a plot of the TPR vs. FPR as its discrimination threshold is varied



$p(c x)$	$T = 0,2$	$T = 0,5$	$T = 0,8$	C
0,99	c^+	c^+	c^+	c^+
0,90	c^+	c^+	c^+	c^+
0,85	c^+	c^+	c^+	c^+
0,80	c^+	c^+	c^+	c^-
0,78	c^+	c^+	c^-	c^+
0,70	c^+	c^+	c^-	c^-
0,60	c^+	c^+	c^-	c^+
0,45	c^+	c^-	c^-	c^-
0,40	c^+	c^-	c^-	c^-
0,30	c^+	c^-	c^-	c^-
0,20	c^+	c^-	c^-	c^+
0,15	c^-	c^-	c^-	c^-
0,10	c^-	c^-	c^-	c^-
0,05	c^-	c^-	c^-	c^-

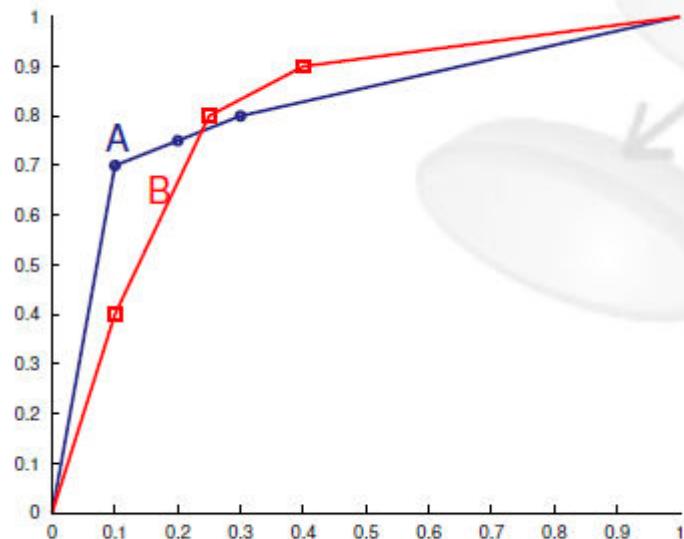
Receiver Operating Characteristics (ROC)



Dominance

- A dominates B throughout all the range of T
- A has a better predictive performance over any condition of cost and class distribution

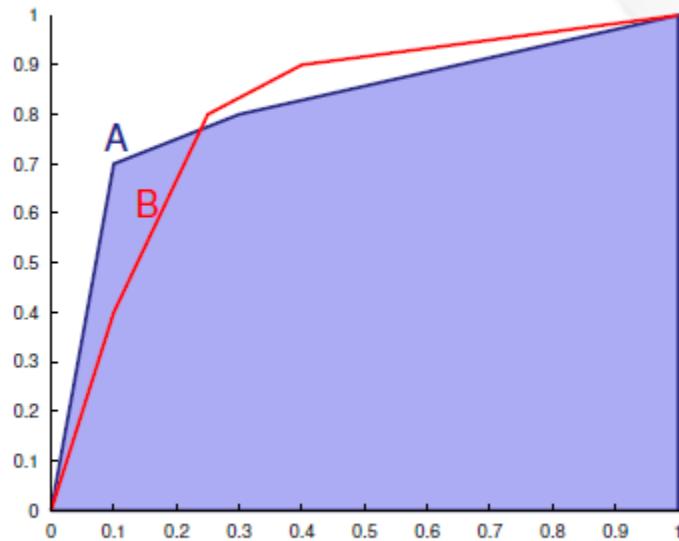
Receiver Operating Characteristics (ROC)



No-Dominance

- The dominance relationship may not be so clear
- No model is the best under all possible scenarios

Receiver Operating Characteristics (ROC)



Area Under ROC Curve

- Equivalent to Wilcoxon test
- If A dominates B :
 $AUC(A) \geq AUC(B)$
- If A does not dominate B
 AUC “cannot identify the best classifier”

k-Nearest Neighboor

*Dr. Retno Kusumaningrum,
S.Si., M.Kom.*



Outline



Characteristics and Definition



Intuition



Nearest Neighbour (NN) Classification



Drawbacks of NN Classification



k-Nearest Neighbour (k-NN) Algorithm



Example of k-NN Implementation



k-NN Practical Issues

Characteristics & Definition

k-NN

What are
The
Differences?



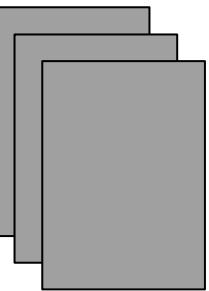
The Differences

Lazy Learning

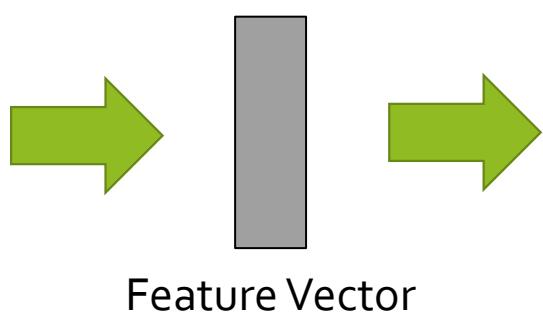
- On querying similarity between testing data and training data (which are stored in database) is calculated to predict the class of testing data

Eager Learning

- Generalized model is generated from training data
- Subsequently, using the model to predict the testing data



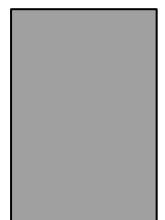
Labeled
Data



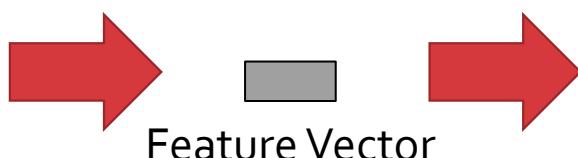
Machine
Learning
Algorithm

Training Process

Classification
Model



New Data



Classify /
Predict

Expected
Label / Class

Testing Process

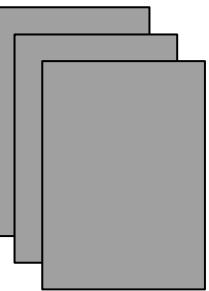
The Differences

Lazy Learning

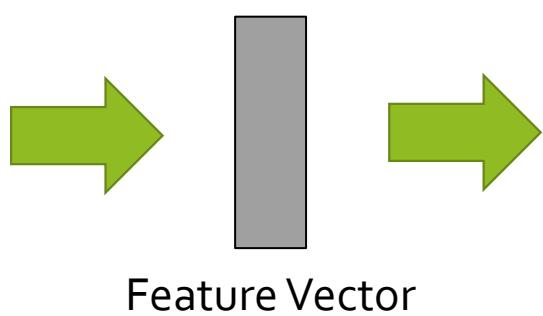
- On querying similarity between testing data and training data (which are stored in database) is calculated to predict the class of testing data

Eager Learning

- Generalized model is generated from training data
- Subsequently, using the model to predict the testing data



Labeled
Data

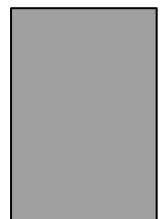


Machine
Learning
Algorithm

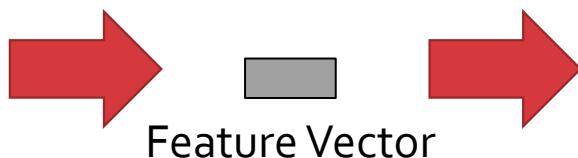
Training Process



Classification
Model



New Data



Classify /
Predict

Expected
Label / Class



Testing Process

Characteristics

- Is a type of **Instance Based Learning / Lazy Learning**
 - Computation is delayed until classification
- Simplest technique since there is no prior knowledge about the distribution of the data

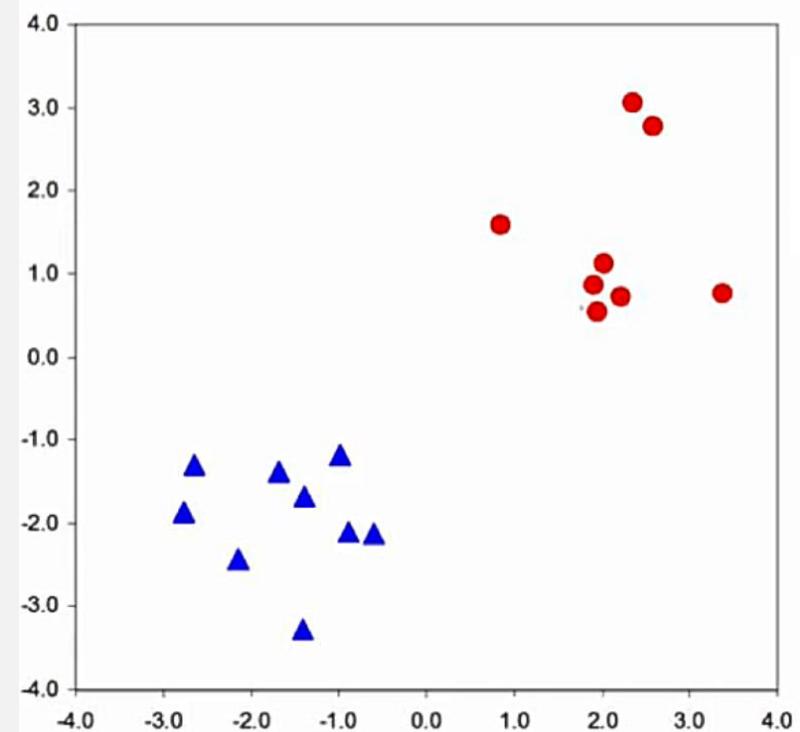


Intuition

NN

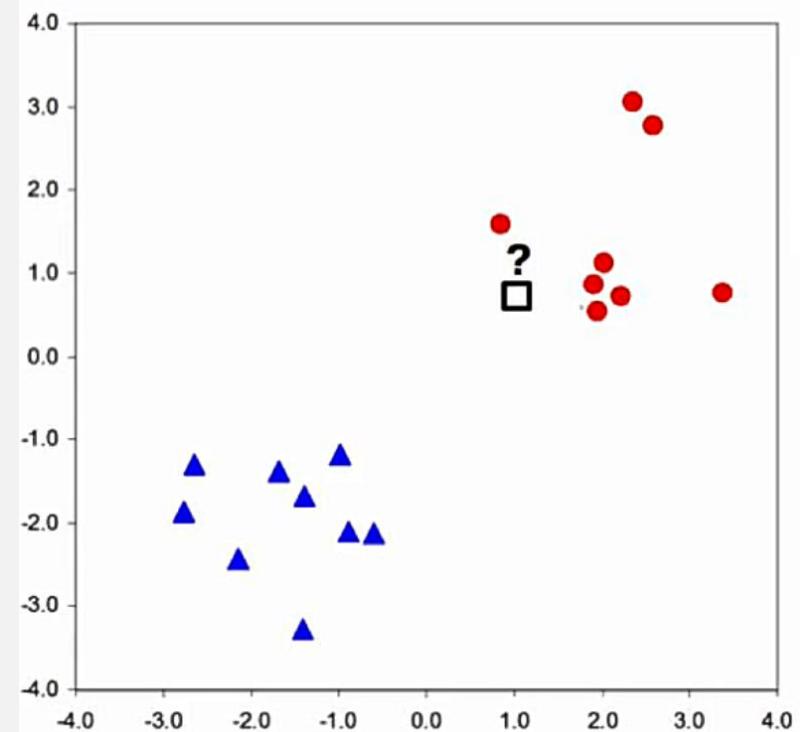
Example

- Set of points (x_1, x_2) :
 - The data consist of two features, i.e. x_1 and x_2
 - There are two classes, i.e. Red (Circle) or Blue (Triangle)



Example

- What is the box class?
Is it red or blue?
- Nearby point is red
 - We can use this as a basis of learning algorithm

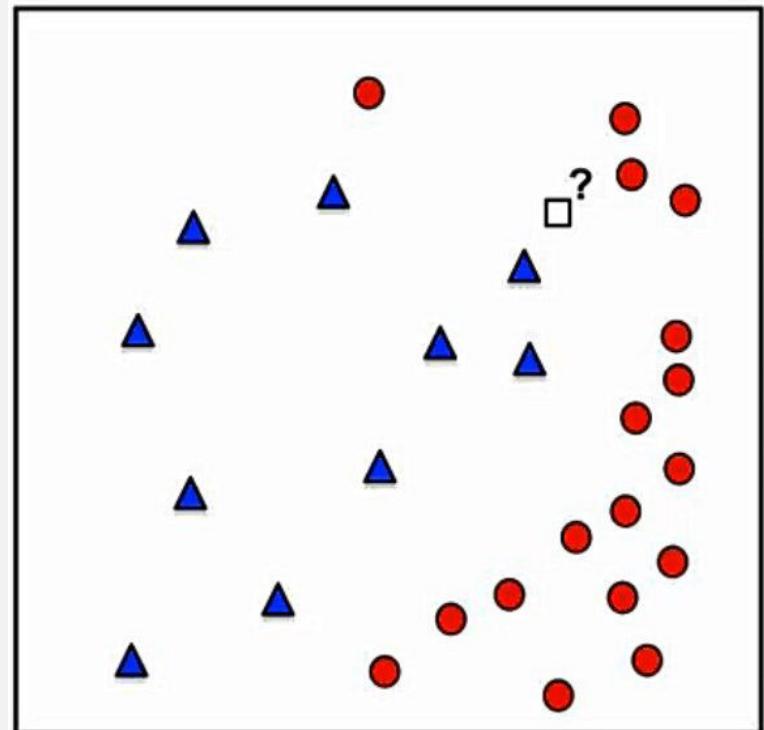


Nearest Neighboor Classification

NN

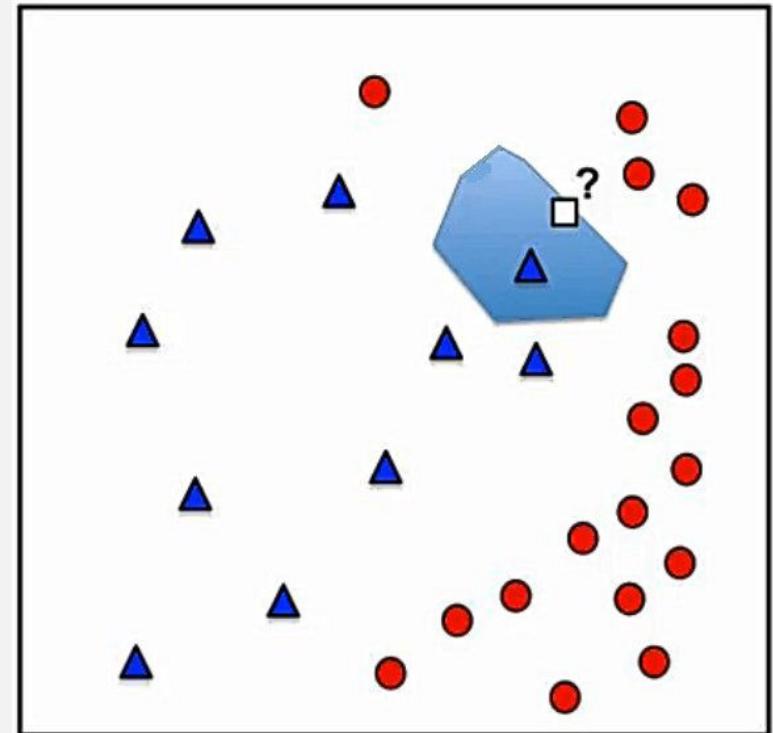
Example

- What about this?
 - Use the intuition to classify new point (box point)
 - Find the most similar training data
 - It should be **Blue** (Triangle)



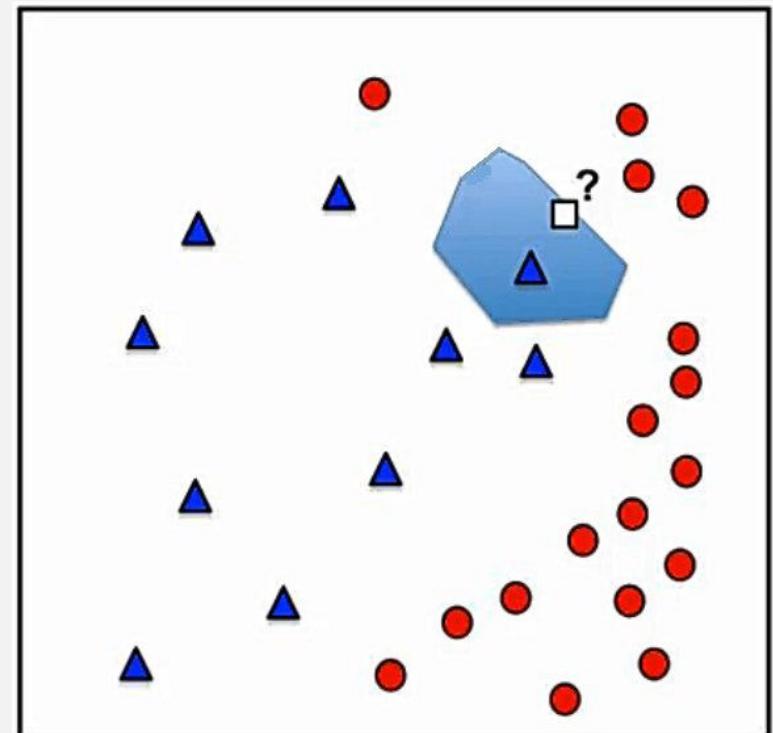
Example

- What about this?
 - It is not just this point (box point) that is going to be blue
 - All of the points around the blue point are should be **Blue** (Triangle)



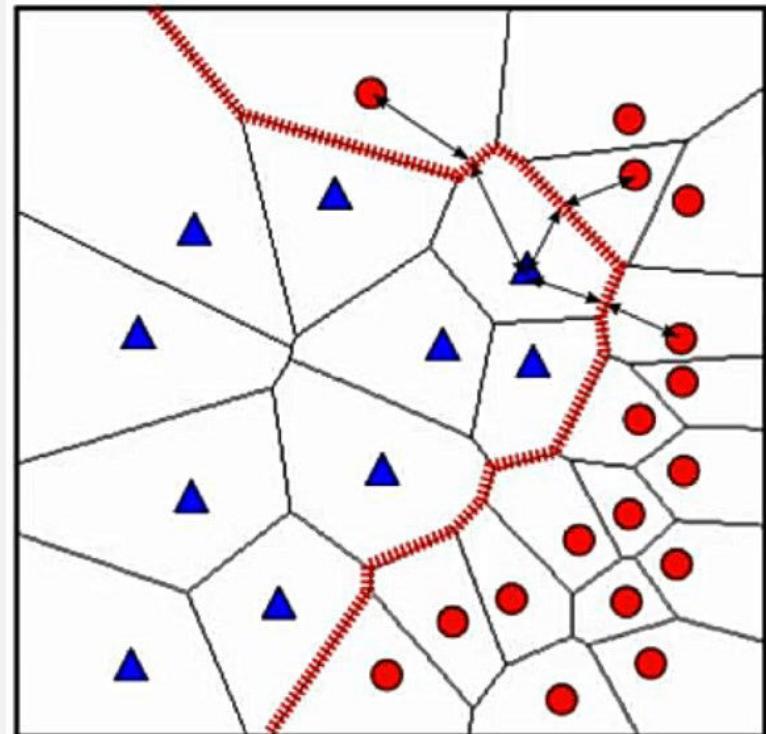
Example

- It is called as voronoi
- Voronoi Tesselation
 - Partition a space into non-overlapping regions
 - Generally, it contains a single point
 - Boundary : points at the same distance from two different training examples



Example

- Classification Boundary?
 - Non-linear
 - Reflects classes well
 - Impressive for simple method

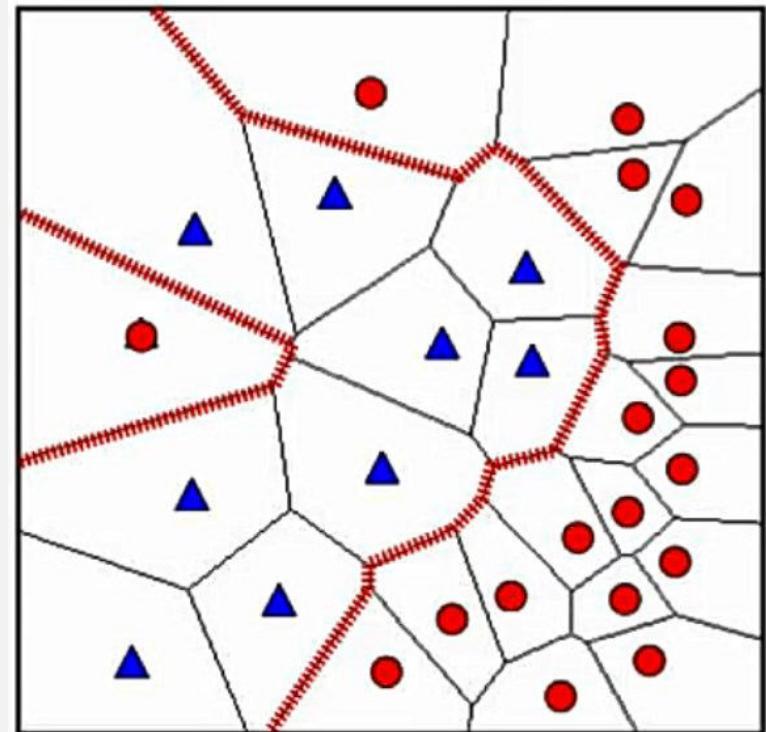


Nearest Neighboor Classification

Its Drawbacks

Nearest Neighboor : Outliers

- If we change a single blue point as red point (which is subsequently called as outlier data)
 - The decision or classification boundary dramatically changes



Drawbacks

- Algorithm is sensitive to outliers
 - Single mislabeled training data dramatically change the decision boundary
- No confidence $p(y|x)$
- Insensitive to class prior

Solution

- To make this algorithm more sensitive to class prior
- Idea :
 - Use more than one nearest neighbor to make decision
 - Count class labels in k most similar training data
- By using more than one nearest neighbor, it makes the classifier more stable

k-Nearest Neighbour

Classification

The Algorithm

- Given :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - \mathbf{x}_i : attribute values representation of examples
 - y_i : class label
 - Testing point \mathbf{x} that we want to classify
- Algorithm :
 - Compute distance \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - Select k closest instances $\mathbf{x}_{i1} \dots \mathbf{x}_{ik}$ and their label $y_{i1} \dots y_{ik}$
 - Output the class y^* which is more frequent in $y_{i1} \dots y_{ik}$

Distance Measures

- Key component of the k-NN algorithm :
 - Defines which training examples are similar and which aren't?
 - Can have strong effect on performance
- Some distance measures:
 - Euclidean Distance
 - Minkowski Distance
 - Hamming Distance
 - Manhattan Distace
 - Mahalanobis Distance
 - dll

Euclidean Distance

1

- Based on previous examples :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - We want to measure distance between testing point \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - There are d features for both training and testing examples
- Euclidean Distance :

$$D(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_d^2 (x_d - x_{i,d})^2}$$

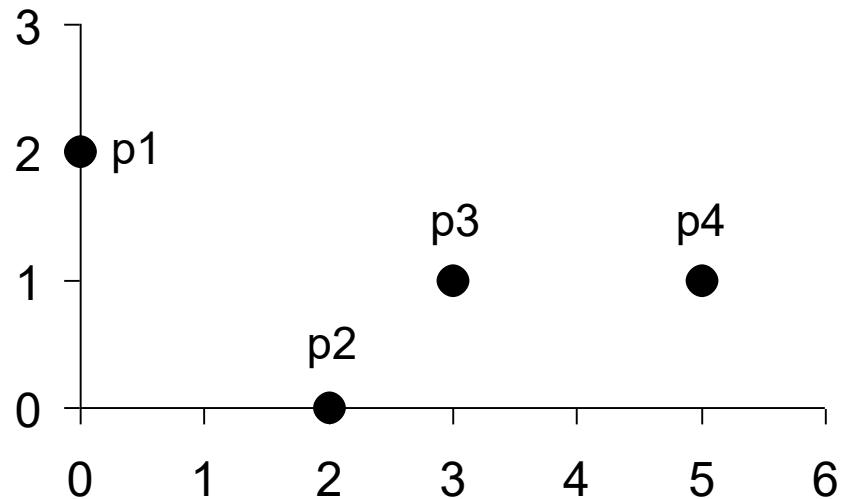
Symmetric,
Spherical, Threat all
Dimension Equally

Sensitive to
extreme differences
in single attribute

Behaves like
a “soft”
logical OR

Euclidean Distance Characteristics

Example



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Hamming Distance

2

- Based on previous examples :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - We want to measure distance between testing point \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - There are d features for both training and testing examples
 - It is categorical features

- Hamming Distance :

$$D(\mathbf{x}, \mathbf{x}_i) = \sum_d \mathbb{I}_{x_d \neq x_{i,d}}$$

Hamming distance count the number of features where two examples are disagree

Manhattan / City Block Distance

3

- Based on previous examples :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - We want to measure distance between testing point \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - There are d features for both training and testing examples
- Manhattan Distance :
$$D(\mathbf{x}, \mathbf{x}_i) = \sum_d |x_d - x_{i,d}|$$

Max / Sup / Chebyshev Distance

4

- Based on previous examples :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - We want to measure distance between testing point \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - There are d features for both training and testing examples
- Max Distance :

$$D(\mathbf{x}, \mathbf{x}_i) = \max_{1 \leq k \leq d} |x_d - x_{i,d}|$$

Minkowski Distance

5

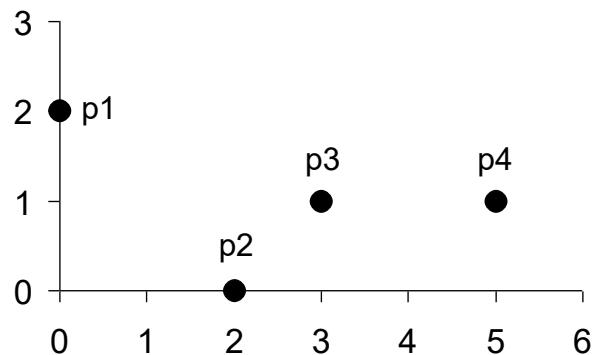
- Based on previous examples :
 - Training examples $\{\mathbf{x}_i, y_i\}$
 - We want to measure distance between testing point \mathbf{x} to every training examples \mathbf{x}_i , $D(\mathbf{x}, \mathbf{x}_i)$
 - There are d features for both training and testing examples
- Minkowski Distance :

$$D(\mathbf{x}, \mathbf{x}_i) = \sqrt[p]{|x_d - x_{i,d}|^p}$$

If $p = 1$, then

If $p = 2$, then

If $p = \infty$, then



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

p=1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

p=2	p1	p2	p3	p4
p1	0	2,828	3,162	5,099
p2	2,828	0	1,414	3,162
p3	3,162	1,414	0	2
p4	5,099	3,162	2	0

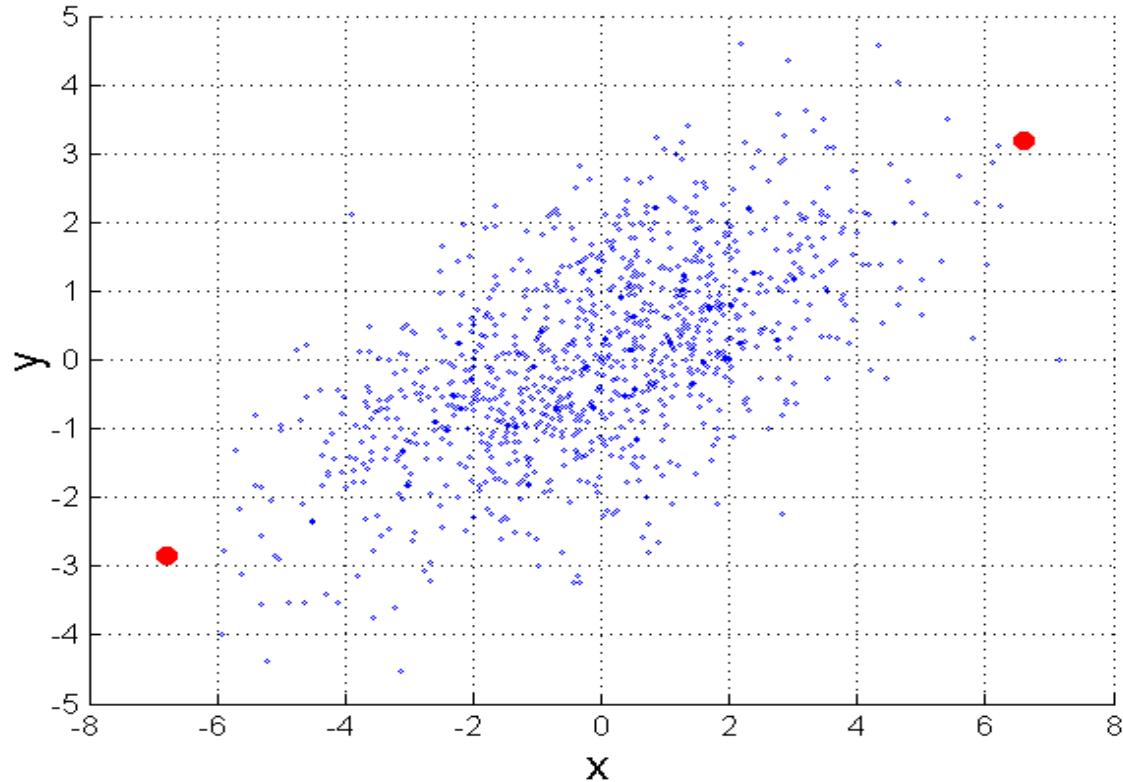
$\pi=\infty$	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

$$\begin{aligned}\text{mahalanobis}(\mathbf{x}, \mathbf{y}) \\ = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})\end{aligned}$$

Σ is the covariance matrix

Mahalanobis
Distance

6



For red points, the Euclidean distance is 14.7,
Mahalanobis distance is 6.

k-NN Classification

Example

Contoh :

Hasil survey untuk mengkategorikan apakah sebuah kertas tissue termasuk kualitas tinggi atau rendah menunjukkan data sebagai berikut :

Daya Tahan Keasamaan (x_1)	Kekuatan (x_2)	Kategori
7	7	R
7	4	R
3	4	T
1	4	T

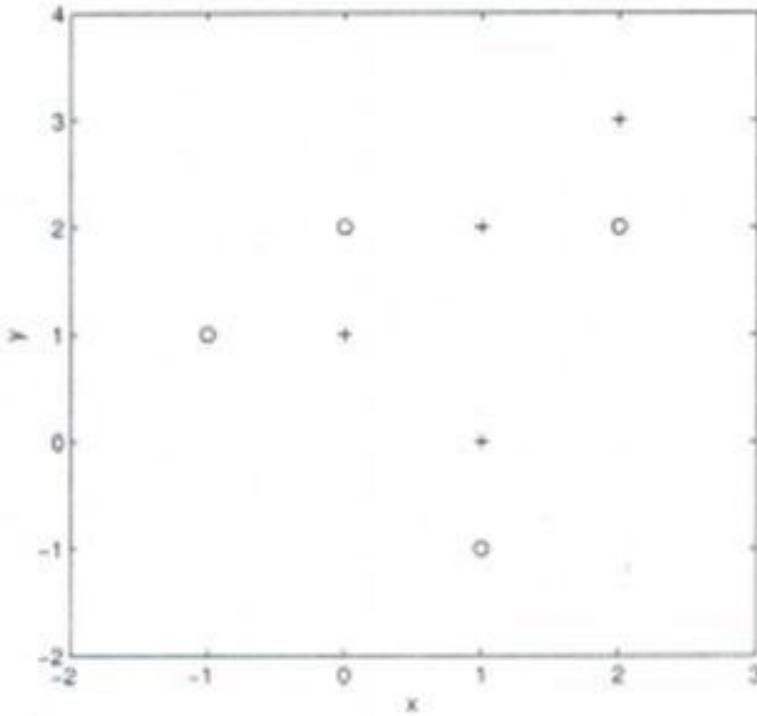
Sebuah pabrik memproduksi kertas tissue baru dan memiliki nilai daya tahan keasaman sebesar 3 serta kekuatan 7. Tentukan kategori dari tissue tersebut menggunakan algoritma KNN

	x_1	x_2	y	$D(u, l_i)$	Ranking
l_1	7	7	R	$D(u, l_1)$ $= \sqrt[2]{(3 - 7)^2 + (7 - 7)^2}$ $= \sqrt[2]{16 + 0} = 4$	3
l_2	7	4	R	$D(u, l_2)$ $= \sqrt[2]{(3 - 7)^2 + (7 - 4)^2}$ $= \sqrt[2]{16 + 9} = 5$	4
l_3	3	4	T	$D(u, l_3)$ $= \sqrt[2]{(3 - 3)^2 + (7 - 4)^2}$ $= \sqrt[2]{0 + 9} = 3$	1
l_4	1	4	T	$D(u, l_1)$ $= \sqrt[2]{(3 - 1)^2 + (7 - 4)^2}$ $= \sqrt[2]{4 + 9} = \sqrt{13}$ $= 3, 61$	2
u	3	7	???		

Misalkan $k = 1$ atau 1-NN , maka :
 $y_{pred} = T$

Misalkan $k = 3$ atau 3-NN , maka :
 $y_{pred} = T$

Soal



Perhatikan gambar di atas:

1. Jika kita ingin memprediksi data baru $x = 1, y = 1$ serta menggunakan euclidean distance dan 3-NN, maka tentukan kelas prediksi untuk data baru tersebut!
2. Untuk soal yang sama, bagaimana jika digunakan euclidean distance dan 7-NN?

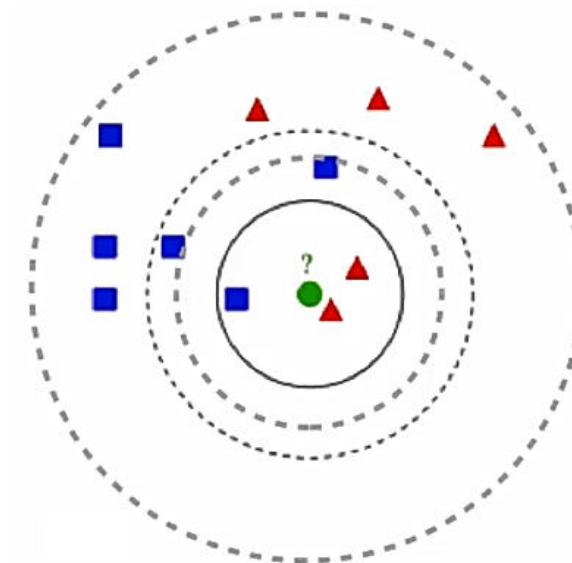
k-NN Practical Issues

- Resolving ties:
 - equal number of positive/negative neighbours
 - use odd k (doesn't solve multi-class)
 - breaking ties:
 - random: flip a coin to decide positive / negative
 - prior: pick class with greater prior
 - nearest: use 1-nn classifier to decide
- Missing values
 - have to “fill in”, otherwise can't compute distance
 - key concern: should affect distance as little as possible
 - reasonable choice: average value across entire dataset

- Features should be on the same scale
 - Example:
 - If one feature has its values in millimeters and another has in centimeters, we would need to normalize
 - One way is:
 - Replace x_{im} by $z_{im} = \frac{(x_{im} - \bar{x}_m)}{\sigma_m}$ (make them zero mean, unit variance)
 - $\bar{x}_m = \frac{1}{N} \sum_{i=1}^N x_{im}$: empirical mean of m^{th} feature
 - $\sigma_m^2 = \frac{1}{N} \sum_{i=1}^N (x_{im} - \bar{x}_m)^2$: empirical variance of m^{th} feature

Choosing the Value of k

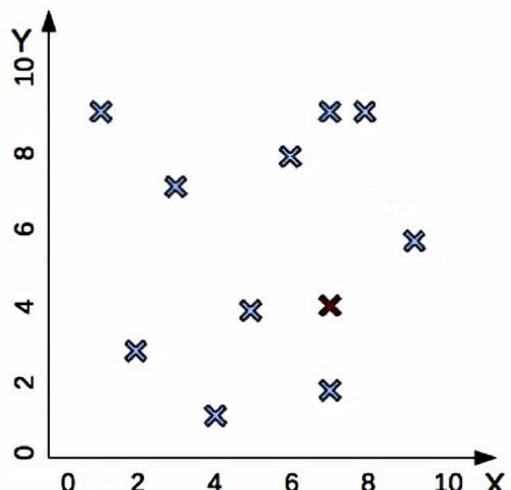
- Value of k has strong effect on kNN performance
 - large value \rightarrow everything classified as the most probable class: $P(y)$
 - small value \rightarrow highly variable, unstable decision boundaries
 - small changes to training set \rightarrow large changes in classification
 - affects “smoothness” of the boundary
- Selecting the value of k
 - set aside a portion of the training data (validation set)
 - vary k , observe training \rightarrow validation error



Making k-NN Fast

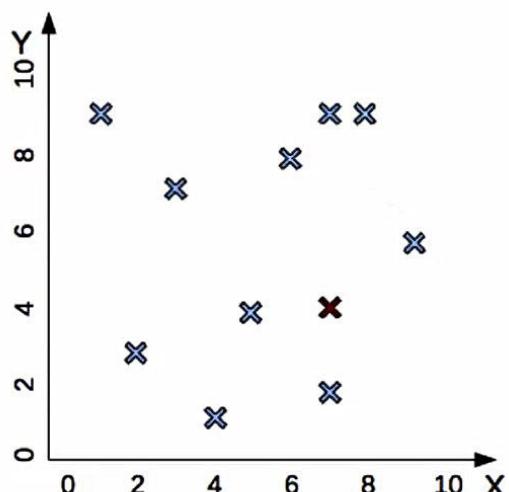
What You See

What Algorithm Sees



Find nearest neighbors
of the testing point (red)

What You See



Find nearest neighbors
of the testing point (red)

What Algorithm Sees

- Training set:
 $\{(1,9), (2,3), (4,1), (3,7), (5,4), (6,8), (7,2), (8,8), (7,9), (9,6)\}$
- Testing instance:
(7,4)
- Nearest neighbors?
compare one-by-one
to each training instance
- n comparisons
- each takes d operations

- Training : $O(d)$, but testing $O(nd)$
- Reduce d : dimensionality reduction
 - Simple feature selection
- Reduce n : don't compare to all training examples
 - Idea : quickly identify $m \ll n$ potential near neighbors
 - Compare only to those, pick k nearest neighbors
 - $O(md)$

Example of Reduce n

- K-D trees : low dimensional, real valued data
 - $O(d \log_2 n)$
 - Only works when $d \ll n$
 - Inexact : can miss neighbors

Example of Reduce n

- Inverted list : high dimensional, discrete (sparse) data
 - $O(n'd')$ where $d' \ll d$, $n' \ll n$
 - Only for sparse data (e.g. Text)
 - Exact

Example of Reduce n

- Local sensitive hashing :
high dimensional, real
valued or discrete data
 - $O(n'd)$ where $n' \ll n$
 - Example : Bits in
fingerprintings
 - Inexact : can miss
neighbors

Decision Tree

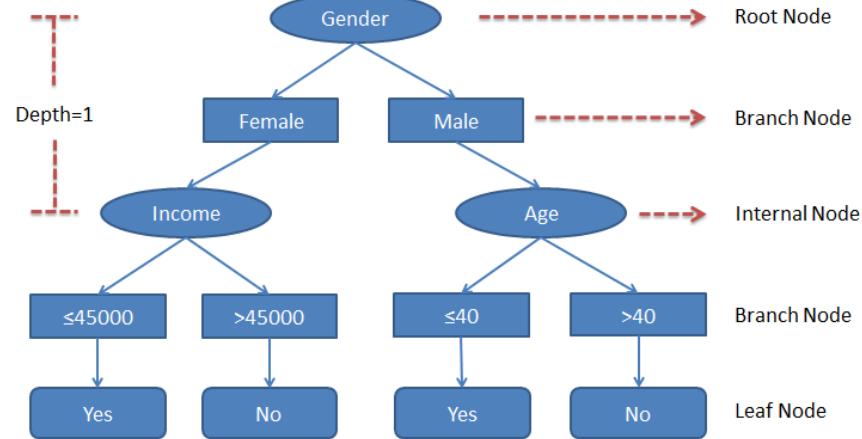
Dr. Retno Kusumaningrum,
S.Si., M.Kom.



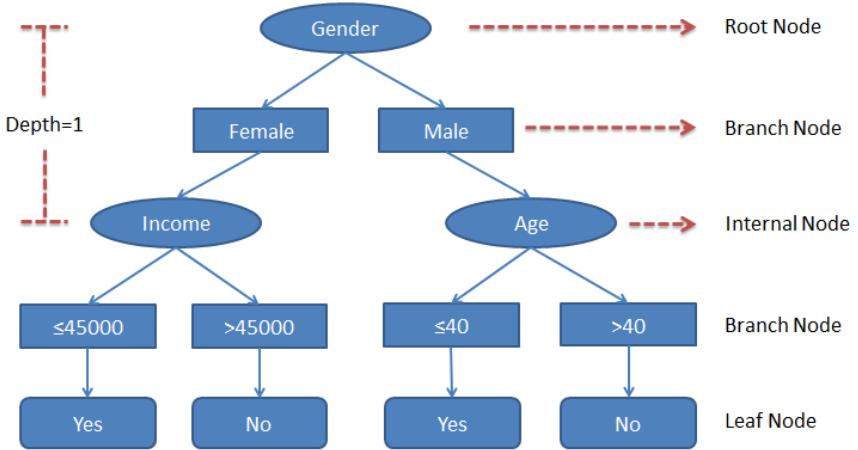
Definition

Decision Tree

A Decision Tree

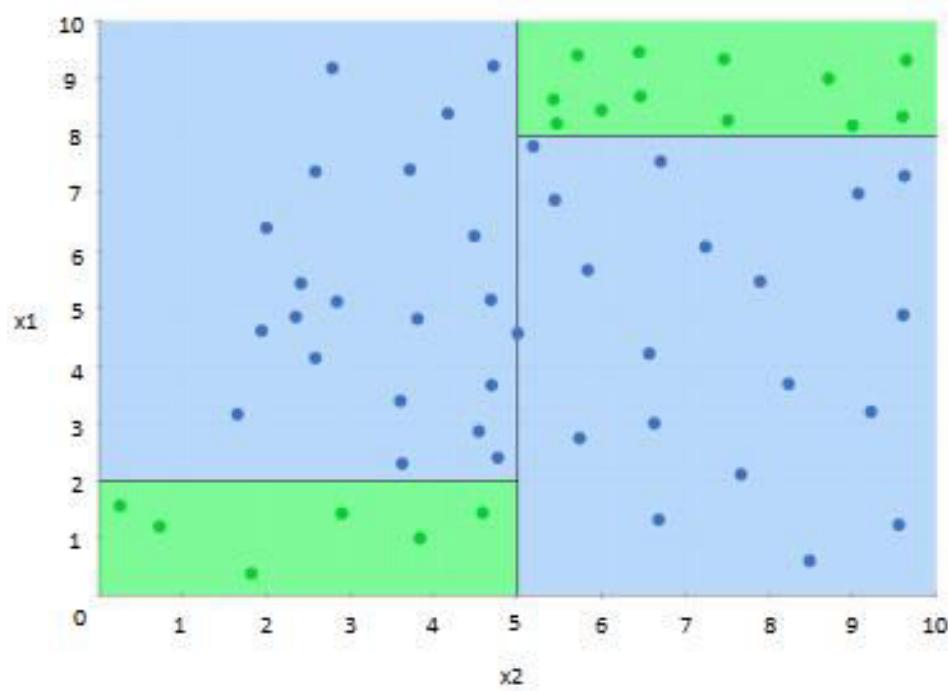


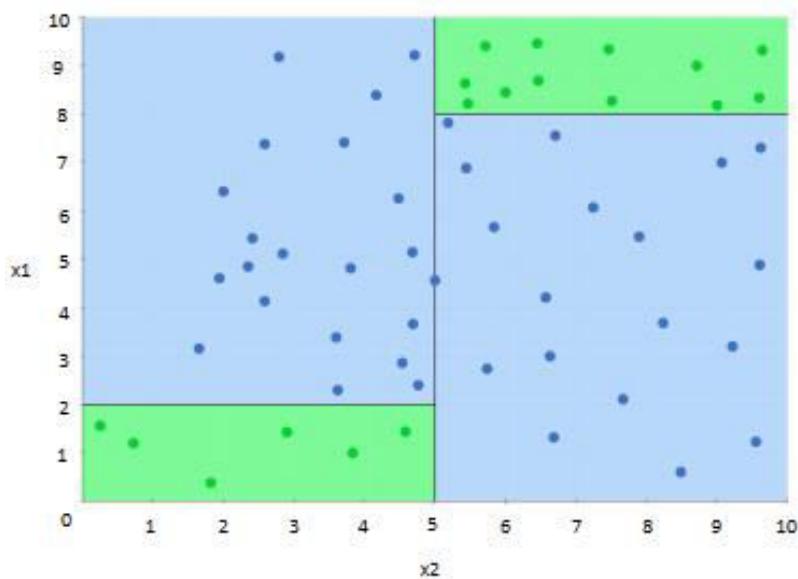
- Uses tree structure to specify sequences of decisions and consequences
- Employs a structure of nodes and branches
- The depth of a node is the minimum number of steps required to reach the node from the root node
- Eventually, a final point is reached and a prediction is made

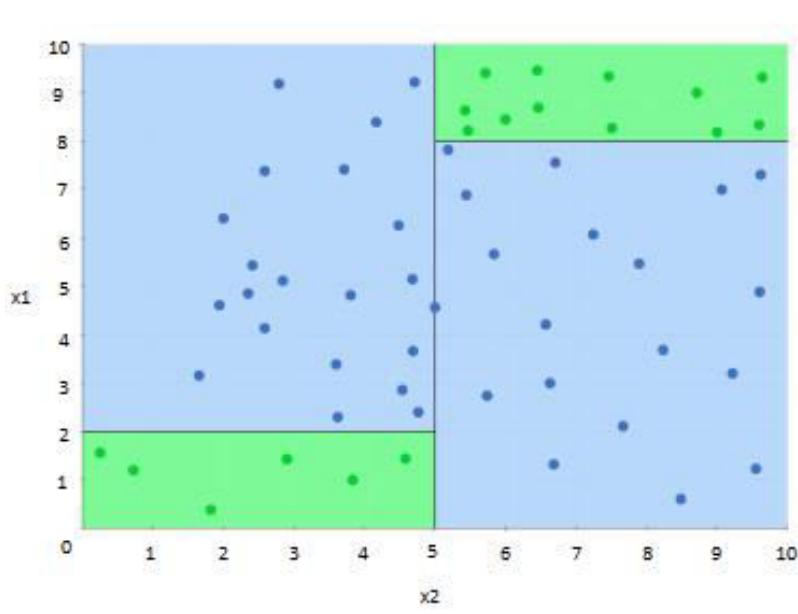


Identifying
the region
(blue or green)

A point lies in
a classification
problem (blue
vs green)







How It Works?

Decision Tree

Predict whether to **play** tennis or **not**?

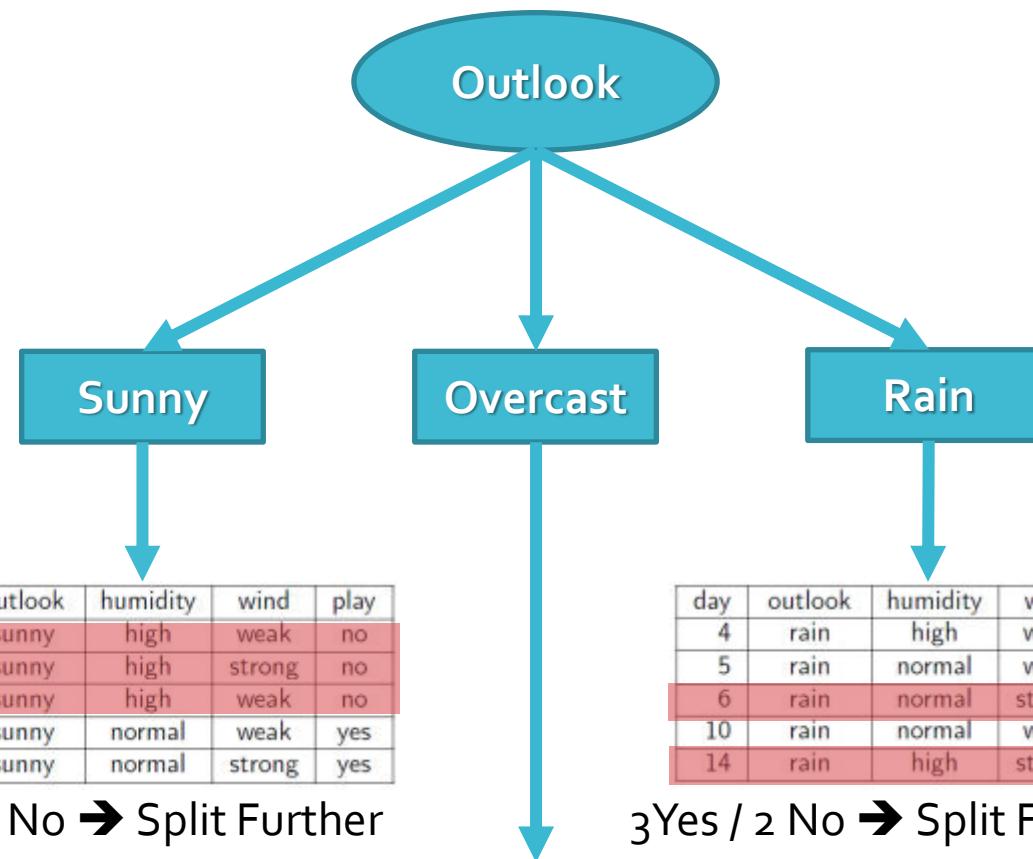
9 Yes
-
5 No

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no

New Data : rain high weak ?

- It is hard to guess
- Try to understand when John plays
- Divide and Conquer
 - Split into subset
 - Are they pure? (all yes or all No)
 - If Yes : stop
 - If No : continue
- See which subset new data falls into

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no



day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
8	sunny	high	weak	no
9	sunny	normal	weak	yes
11	sunny	normal	strong	yes

2 Yes / 3 No → Split Further

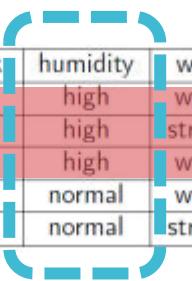
day	outlook	humidity	wind	play
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
10	rain	normal	weak	yes
14	rain	high	strong	no

3 Yes / 2 No → Split Further

day	outlook	humidity	wind	play
3	overcast	high	weak	yes
7	overcast	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes

4 Yes / 0 No → Pure Subset

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
8	sunny	high	weak	no
9	sunny	normal	weak	yes
11	sunny	normal	strong	yes



Outlook

Sunny

Overcast

Rain

Humidity

High

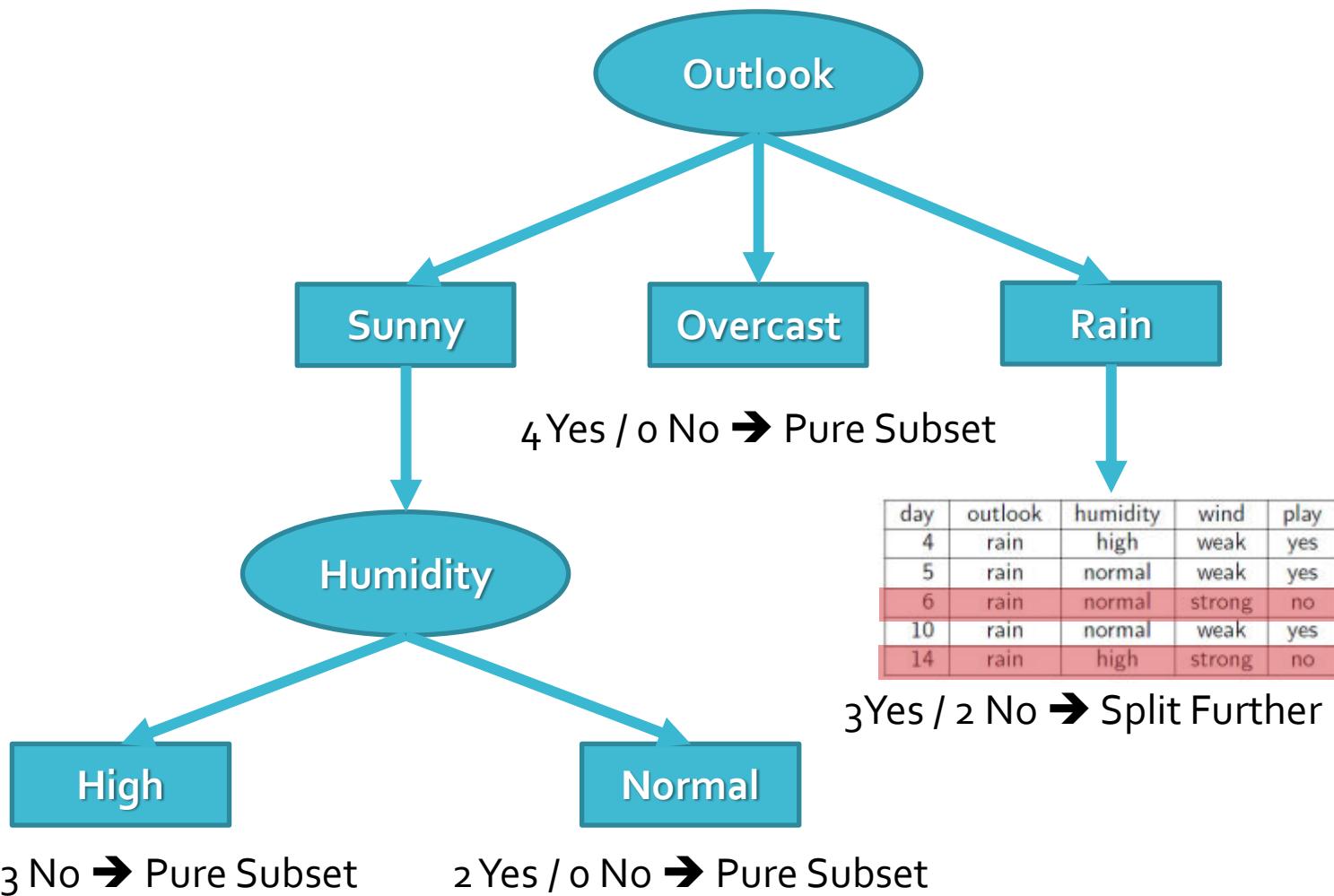
Normal

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
8	sunny	high	weak	no

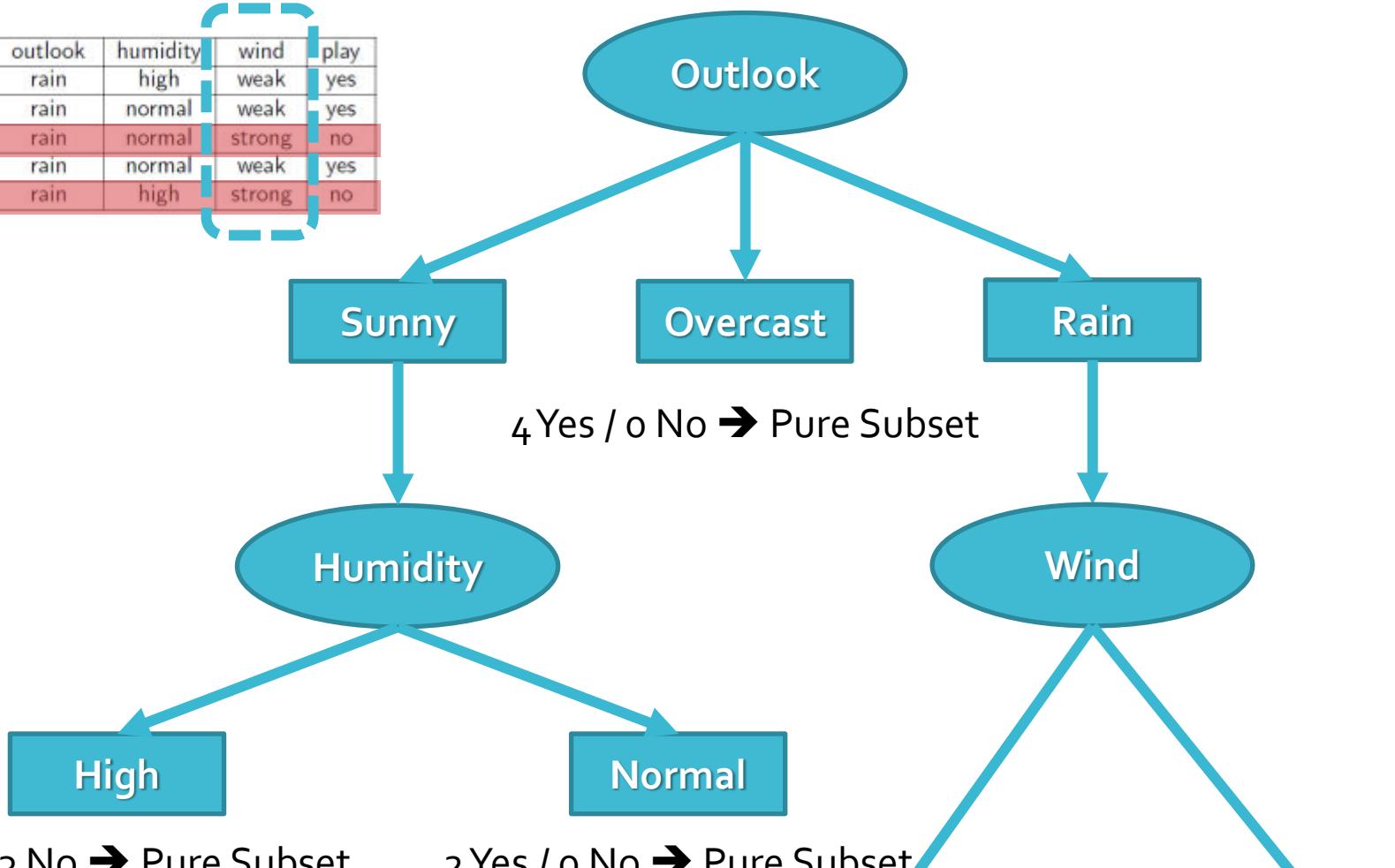
day	outlook	humidity	wind	play
9	sunny	normal	weak	yes
11	sunny	normal	strong	yes

2 Yes / 3 No → Pure Subset

o Yes / 3 No → Pure Subset



day	outlook	humidity	wind	play
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
10	rain	normal	weak	yes
14	rain	high	strong	no



Outlook

Sunny

Overcast

Rain

Humidity

High

Normal

Wind

Strong

Weak

4 Yes / o No → Pure Subset

o Yes / 3 No → Pure Subset

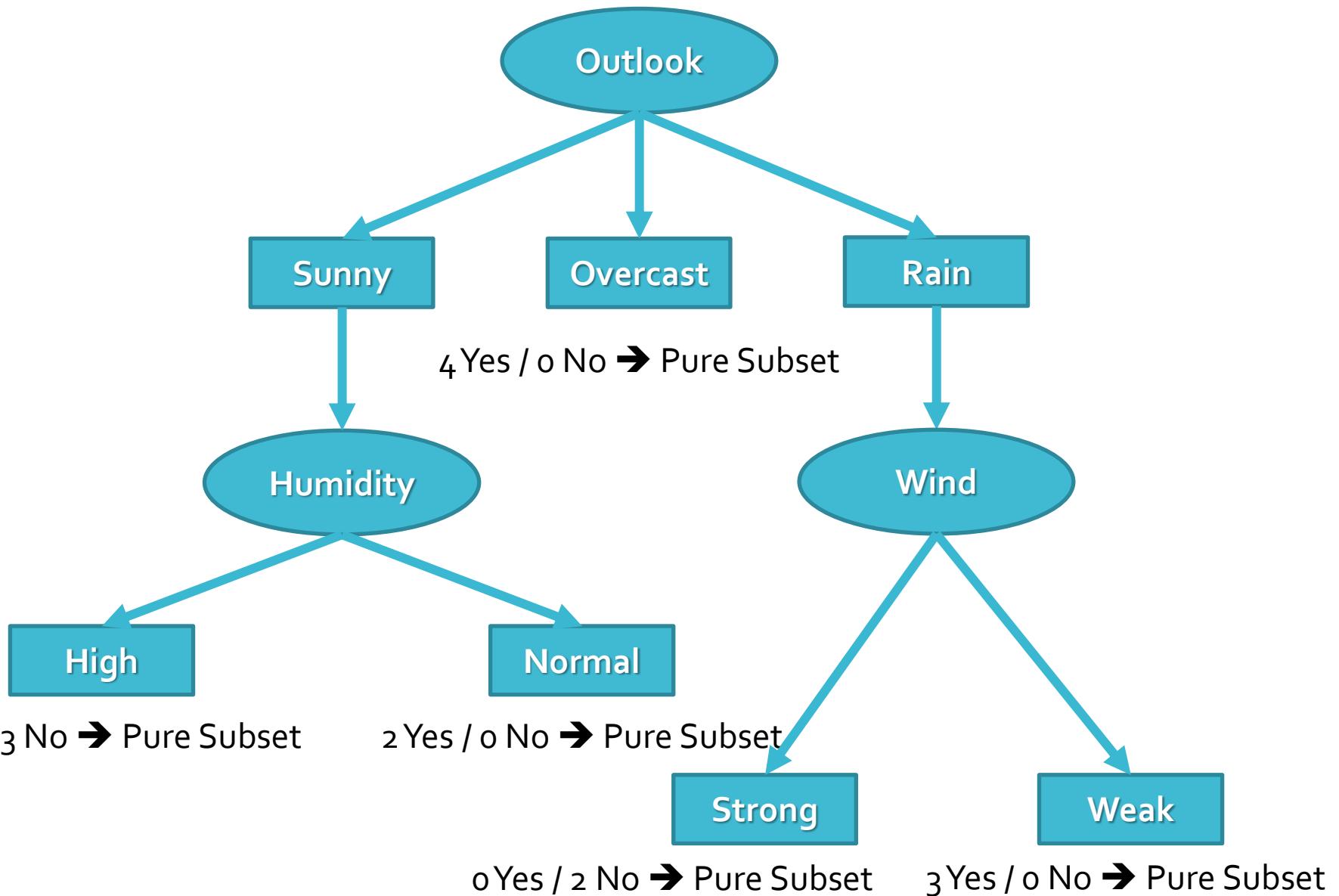
2 Yes / o No → Pure Subset

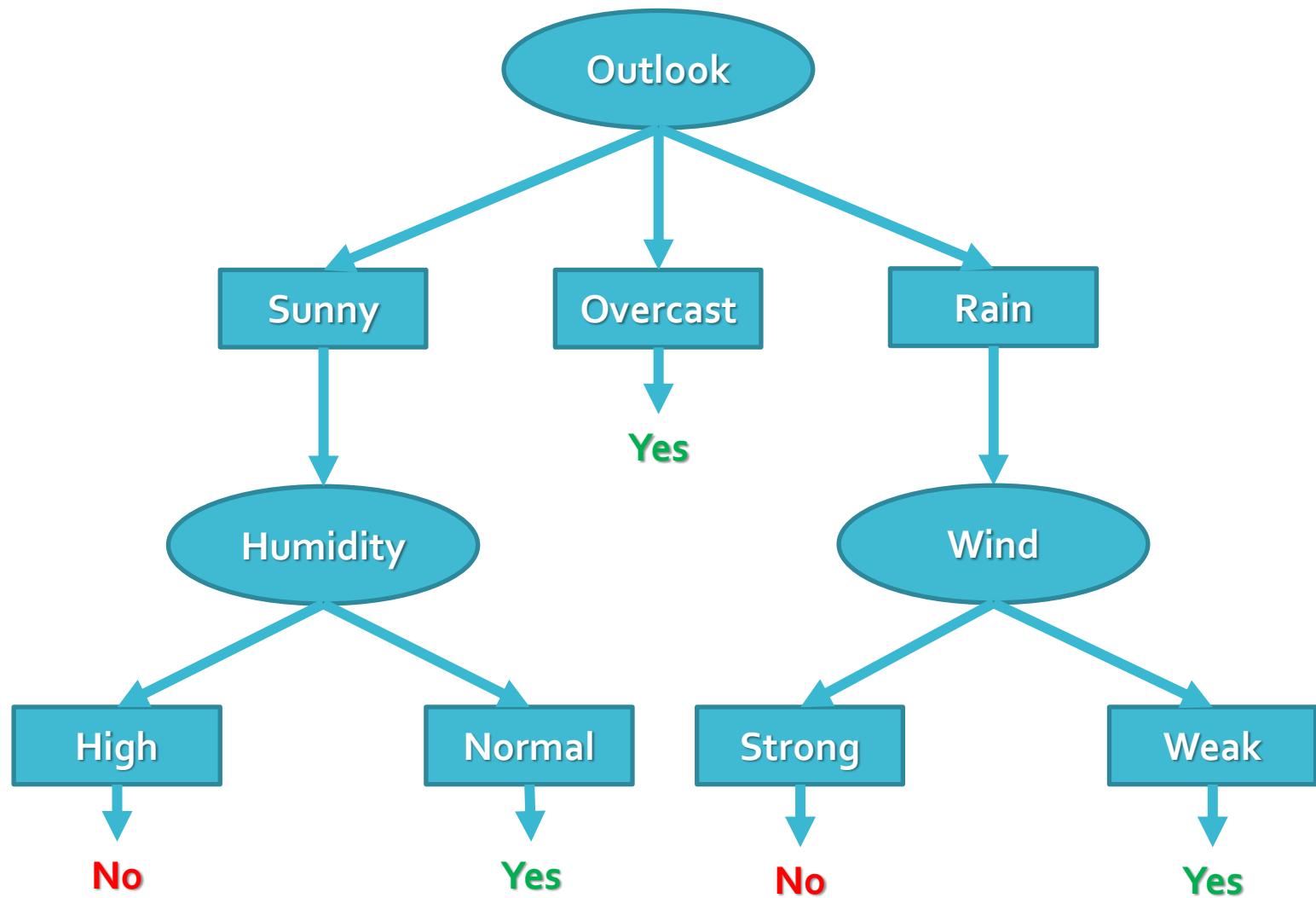
day	outlook	humidity	wind	play
6	rain	normal	strong	no
14	rain	high	strong	no

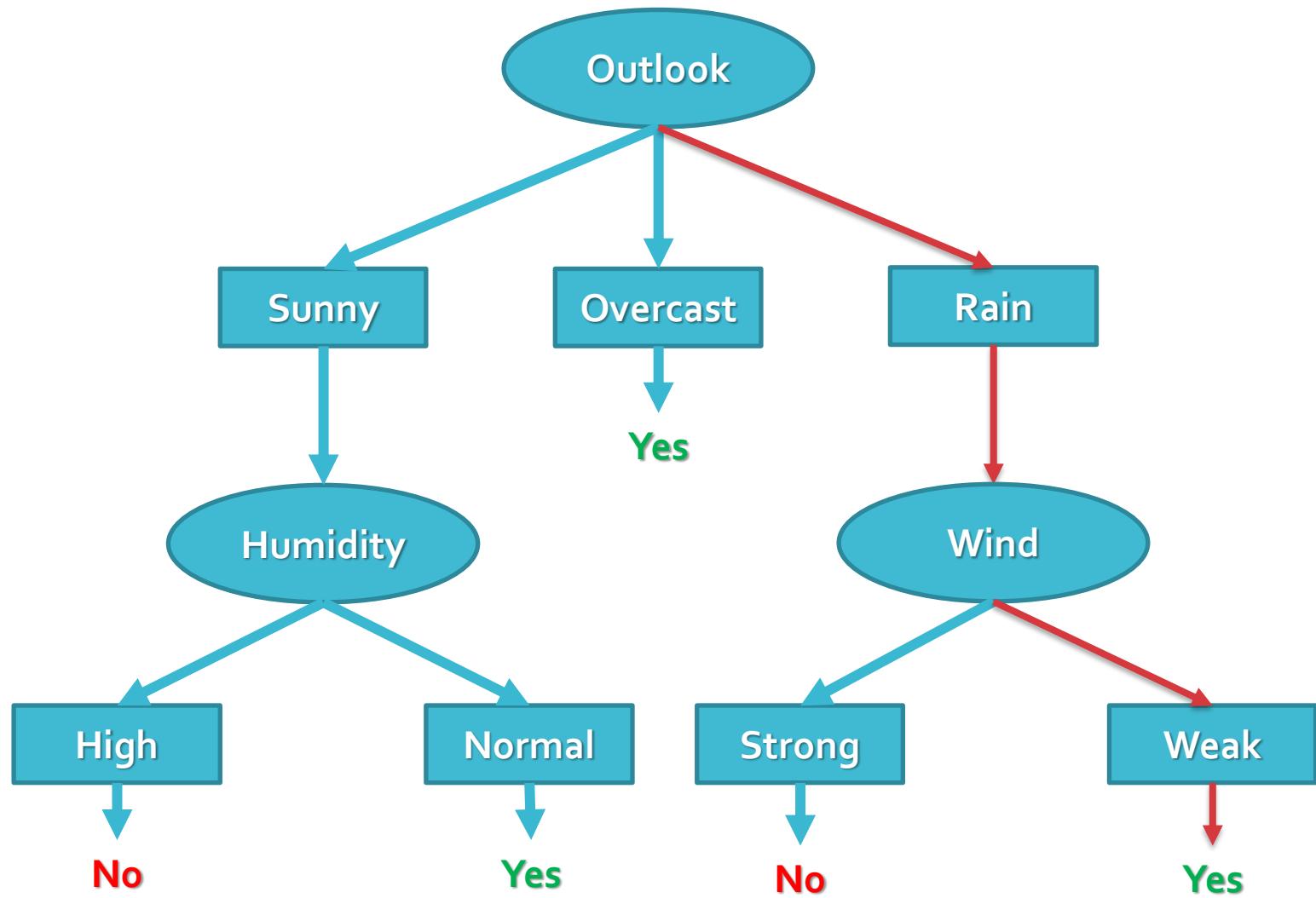
o Yes / 2 No → Pure Subset

day	outlook	humidity	wind	play
4	rain	high	weak	yes
5	rain	normal	weak	yes
10	rain	normal	weak	yes

3 Yes / o No → Pure Subset







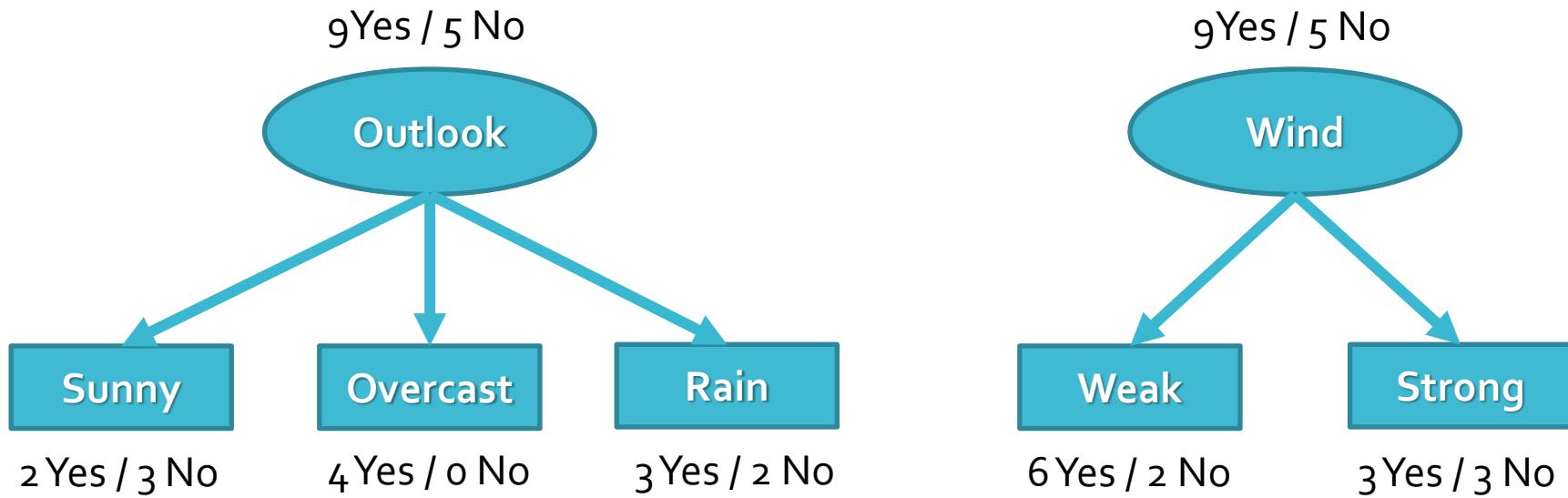
New Data : Outlook – rain, Humidity – high, and Wind – weak ?

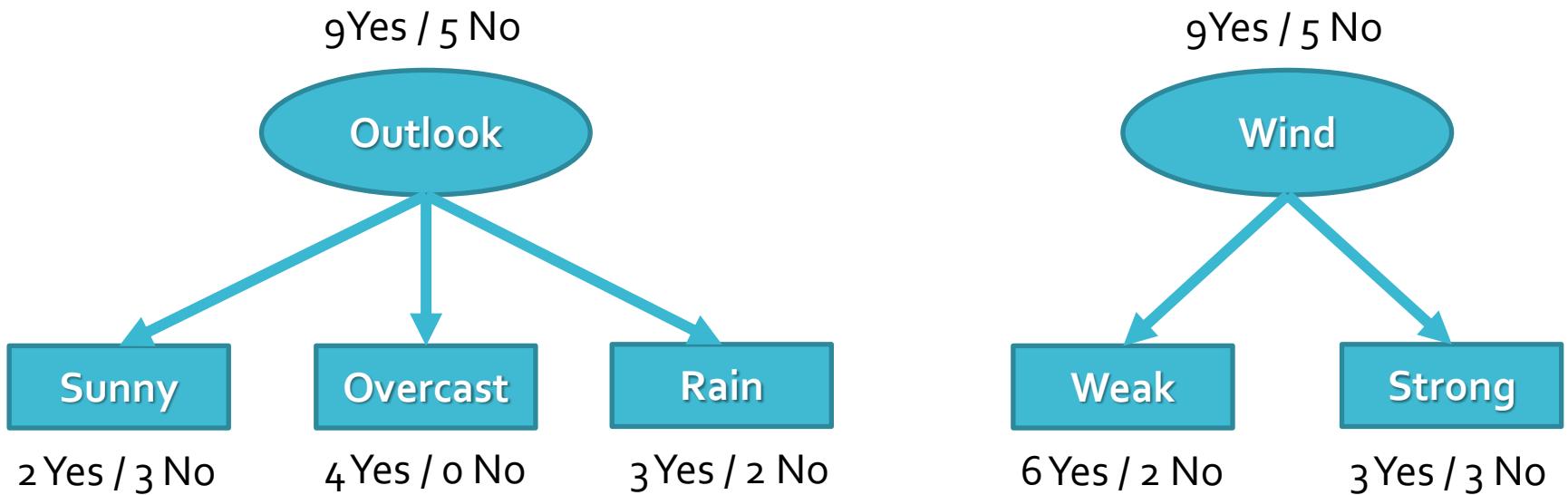
Prediction : Play Tennis

Which Attribute to Split On?

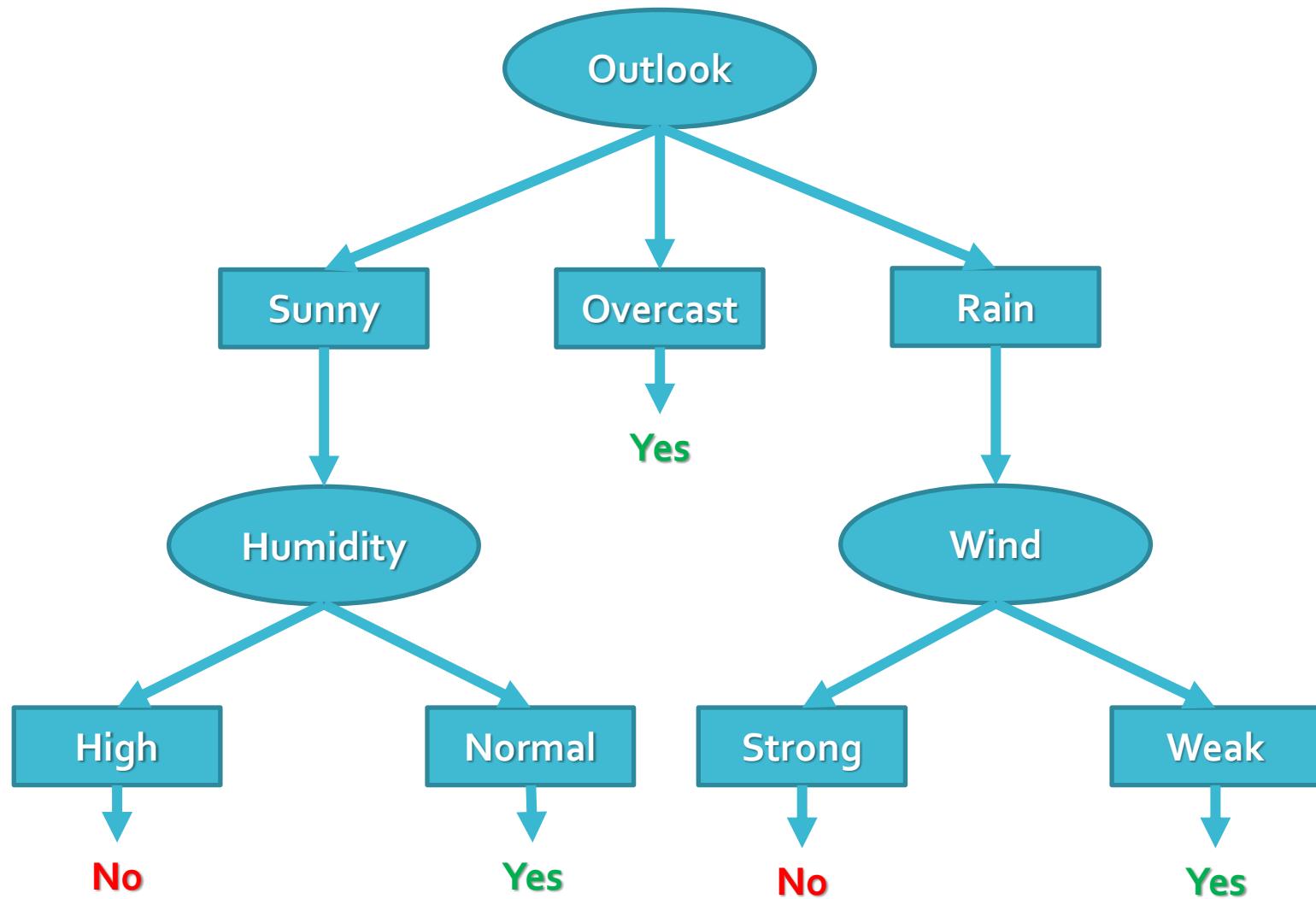
Decision Tree

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no





- Want to measure “purity” of the split
 - More certain about Yes / No after the split
 - Pure set (4 Yes / 0 No) → completely certain (100%)
 - Impure (3 Yes / 3 No) → completely uncertain (50%)
 - Can't use $p(\text{"Yes"}|\text{Set})$:
 - Must be symmetric : 4 Yes / 0 No as pure as 0 Yes / 4 No



Q : Why we choose outlook feature at the root node?

A : Outlook is the most informative feature, i.e. the purest feature

Entropy

- Measures *randomness(uncertainty)* in data
- For example S is a set of examples with C is the number of classes, then Entropy of S is :

$$H(S) = - \sum_{c \in C} p_c \log_2 p_c$$

Where p_c is the probability of an element S is included in class C

Examples :

- Impure (3 yes / 3 no) :

$$H(S) = - \sum_{c \in C} p_c \log_2 p_c$$

Since there are two classes, i.e. Yes and No

$$H(S) = -p_{yes} \log_2 p_{yes} - p_{no} \log_2 p_{no}$$

$$\begin{aligned} H(S) &= -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = -\frac{1}{2}(-1) - \frac{1}{2}(-1) \\ &= \frac{1}{2} + \frac{1}{2} = 1 \end{aligned}$$

- Pure set (4 Yes / 0 No) \rightarrow completely certain (100%)

$$\begin{aligned} H(S) &= -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = -1.0 - 0 \\ &= 0 \end{aligned}$$

Information Gain

- Assume that each of elements in S consists of a set of features
- Information Gain (IG) of features F

$$IG(S, F) = H(S) - \sum_{f \in F} \frac{|S_f|}{|S|} H(S_f)$$

Where S_f is the number of element S with the value of feature F is f

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no

Previous Examples :

Look at feature

“wind” $\in \{weak, strong\}$

Root node : $S = [9Y, 5N]$

$$\text{Entropi} : H(S) = - \left(\frac{9}{14} \right) \log_2 \left(\frac{9}{14} \right) - \left(\frac{5}{14} \right) \log_2 \left(\frac{5}{14} \right) = 0,94$$

$$S_{weak} = [6Y, 2N] \rightarrow H(S_{weak}) = - \left(\frac{6}{8} \right) \log_2 \left(\frac{6}{8} \right) - \left(\frac{2}{8} \right) \log_2 \left(\frac{2}{8} \right) = 0,811$$

$$S_{strong} = [3Y, 3N] \rightarrow H(S_{strong}) = - \left(\frac{3}{6} \right) \log_2 \left(\frac{3}{6} \right) - \left(\frac{3}{6} \right) \log_2 \left(\frac{3}{6} \right) = 1$$

$$\begin{aligned} IG(S, wind) &= H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong}) \\ &= 0,94 - \frac{8}{14} * 0,811 - \frac{6}{14} * 1 = 0,084 \end{aligned}$$

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no

Previous Examples :

Look at feature

“humidity” $\in \{high, normal\}$

Root node : $S = [9Y, 5N]$

$$\text{Entropi} : H(S) = - \left(\frac{9}{14} \right) \log_2 \left(\frac{9}{14} \right) - \left(\frac{5}{14} \right) \log_2 \left(\frac{5}{14} \right) = 0,94$$

$$S_{high} = [3Y, 4N] \rightarrow H(S_{high}) = - \left(\frac{3}{7} \right) \log_2 \left(\frac{3}{7} \right) - \left(\frac{4}{7} \right) \log_2 \left(\frac{4}{7} \right) = 0,99$$

$$S_{normal} = [6Y, 1N] \rightarrow H(S_{normal}) = - \left(\frac{6}{7} \right) \log_2 \left(\frac{6}{7} \right) - \left(\frac{1}{7} \right) \log_2 \left(\frac{1}{7} \right) = 0,59$$

$$\begin{aligned}
 IG(S, \text{humidity}) &= H(S) - \frac{|S_{high}|}{|S|} H(S_{high}) - \frac{|S_{normal}|}{|S|} H(S_{normal}) \\
 &= 0,94 - \frac{7}{14} * 0,99 - \frac{7}{14} * 0,59 = 0,152
 \end{aligned}$$

Previous Examples :

Look at feature

“outlook” $\in \{sunny, overcast, rain\}$

Root node : $S = [9Y, 5N]$

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
3	overcast	high	weak	yes
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
7	overcast	normal	strong	yes
8	sunny	high	weak	no
9	sunny	normal	weak	yes
10	rain	normal	weak	yes
11	sunny	normal	strong	yes
12	overcast	high	strong	yes
13	overcast	normal	weak	yes
14	rain	high	strong	no

Entropi : $H(S) = - \left(\frac{9}{14} \right) \log_2 \left(\frac{9}{14} \right) - \left(\frac{5}{14} \right) \log_2 \left(\frac{5}{14} \right) = 0,94$

$S_{sunny} = [2Y, 3N] \rightarrow H(S_{sunny}) = - \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) - \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) = 0,97$

$S_{overcast} = [4Y, 0N] \rightarrow H(S_{overcast}) = - \left(\frac{4}{4} \right) \log_2 \left(\frac{4}{4} \right) - \left(\frac{0}{4} \right) \log_2 \left(\frac{0}{4} \right) = 0$

$S_{rain} = [3Y, 2N] \rightarrow H(S_{rain}) = - \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) - \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) = 0,97$

$$\begin{aligned}
 IG(S, wind) &= H(S) - \frac{|S_{sunny}|}{|S|} H(S_{sunny}) - \frac{|S_{overcast}|}{|S|} H(S_{overcast}) - \frac{|S_{rain}|}{|S|} H(S_{rain}) \\
 &= 0,94 - \frac{5}{14} * 0,97 - \frac{4}{14} * 0 - \frac{5}{14} * 0,97 = 0,247
 \end{aligned}$$

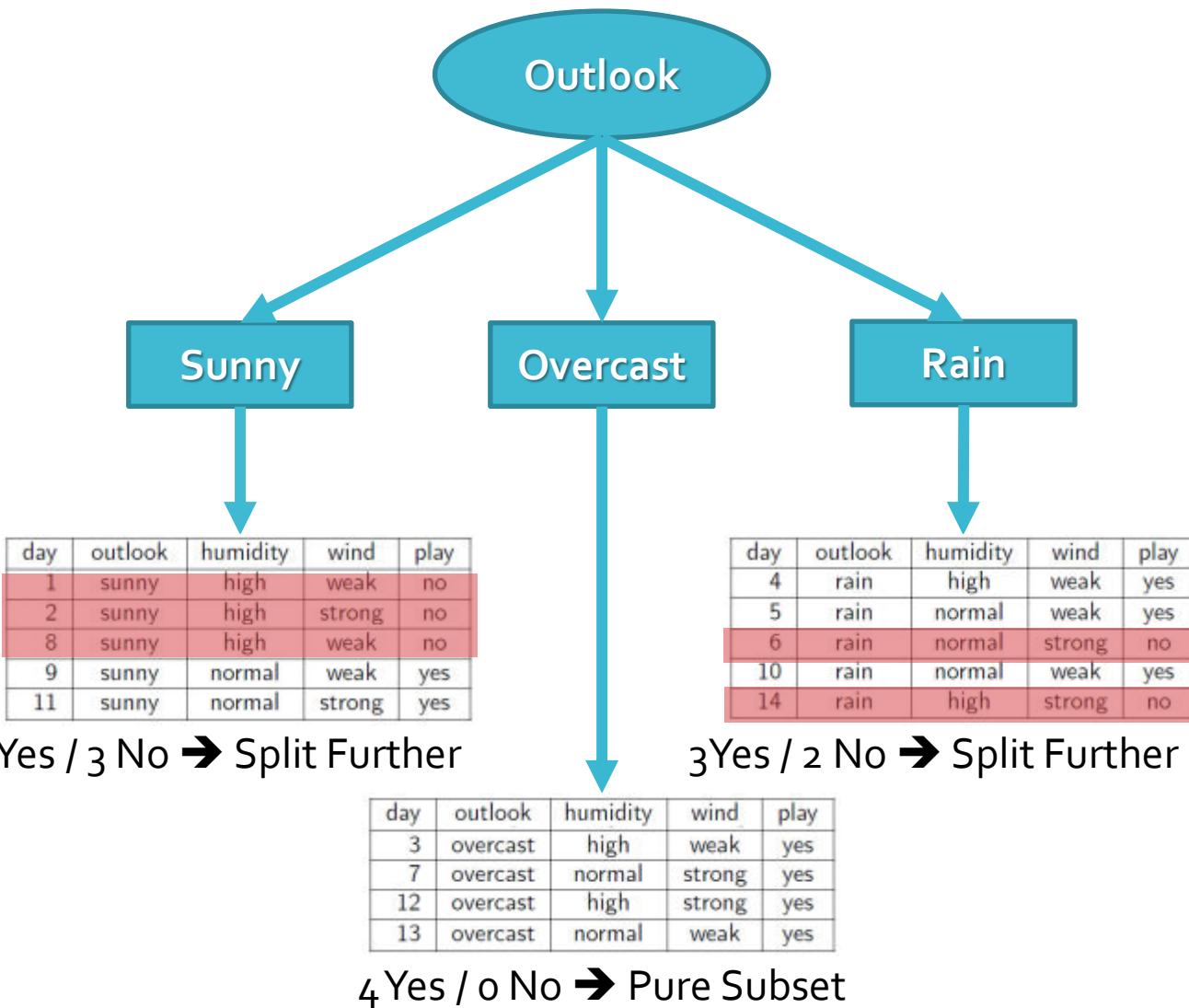
- The obtained value of information gain are :

$$IG(S, \text{wind}) = 0,048$$

$$IG(S, \text{humidity}) = 0,152$$

$$IG(S, \text{outlook}) = 0,247$$

- “Outlook” has the highest IG value
→ it is set as a root node
- Growing the tree
 - Iteratively select the feature with the highest information gain for each child of the previous node



Growing The Tree

Look at feature

“wind” $\in \{weak, strong\}$

Level 2, left node: $S = [2Y, 3N]$

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
8	sunny	high	weak	no
9	sunny	normal	weak	yes
11	sunny	normal	strong	yes

2 Yes / 3 No → Split Further

$$\text{Entropi : } H(S) = -\left(\frac{2}{5}\right) \log_2 \left(\frac{2}{5}\right) - \left(\frac{3}{5}\right) \log_2 \left(\frac{3}{5}\right) = 0,971$$

$$S_{weak} = [1Y, 2N] \rightarrow H(S_{weak}) = -\left(\frac{1}{3}\right) \log_2 \left(\frac{1}{3}\right) - \left(\frac{2}{3}\right) \log_2 \left(\frac{2}{3}\right) = 0,92$$

$$S_{strong} = [1Y, 1N] \rightarrow H(S_{strong}) = -\left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) = 1$$

$$\begin{aligned} IG(S, wind) &= H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong}) \\ &= 0,971 - \frac{3}{5} * 0,918 - \frac{2}{5} * 1 = 0,020 \end{aligned}$$

Growing The Tree

Look at feature

“humidity” $\in \{high, normal\}$

Level 2, left node: $S = [2Y, 3N]$

day	outlook	humidity	wind	play
1	sunny	high	weak	no
2	sunny	high	strong	no
8	sunny	high	weak	no
9	sunny	normal	weak	yes
11	sunny	normal	strong	yes

2 Yes / 3 No → Split Further

$$\text{Entropi : } H(S) = -\left(\frac{2}{5}\right) \log_2 \left(\frac{2}{5}\right) - \left(\frac{3}{5}\right) \log_2 \left(\frac{3}{5}\right) = 0,971$$

$$S_{high} = [0Y, 3N] \rightarrow H(S_{high}) = -(0) \log_2(0) - \left(\frac{3}{3}\right) \log_2 \left(\frac{3}{3}\right) = 0$$

$$S_{normal} = [2Y, 0N] \rightarrow H(S_{normal}) = -\left(\frac{2}{2}\right) \log_2 \left(\frac{2}{2}\right) - (0) \log_2(0) = 0$$

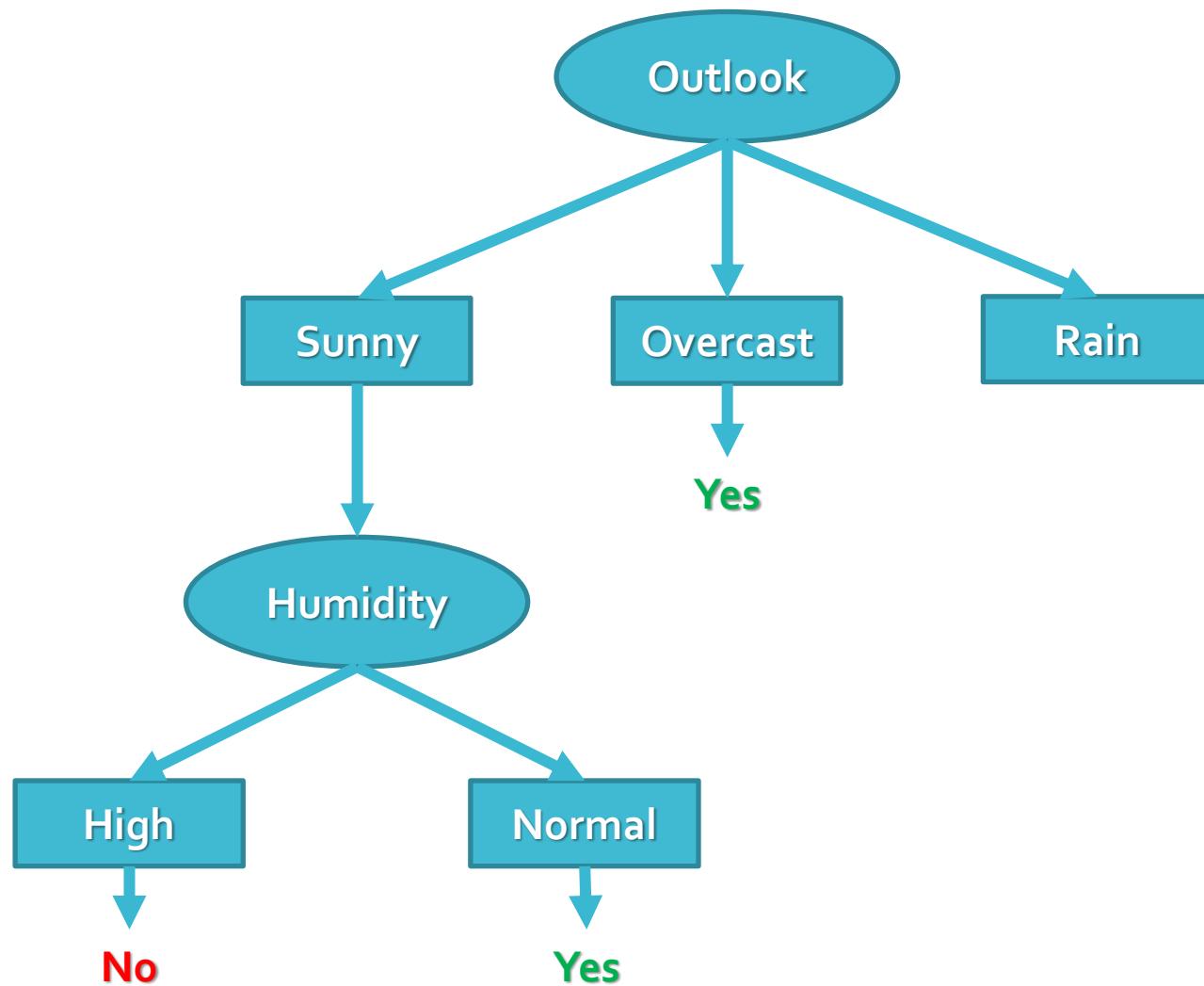
$$\begin{aligned} IG(S, \text{humidity}) &= H(S) - \frac{|S_{high}|}{|S|} H(S_{high}) - \frac{|S_{normal}|}{|S|} H(S_{normal}) \\ &= 0,971 - \frac{2}{5} * 0 - \frac{3}{5} * 0 = 0,971 \end{aligned}$$

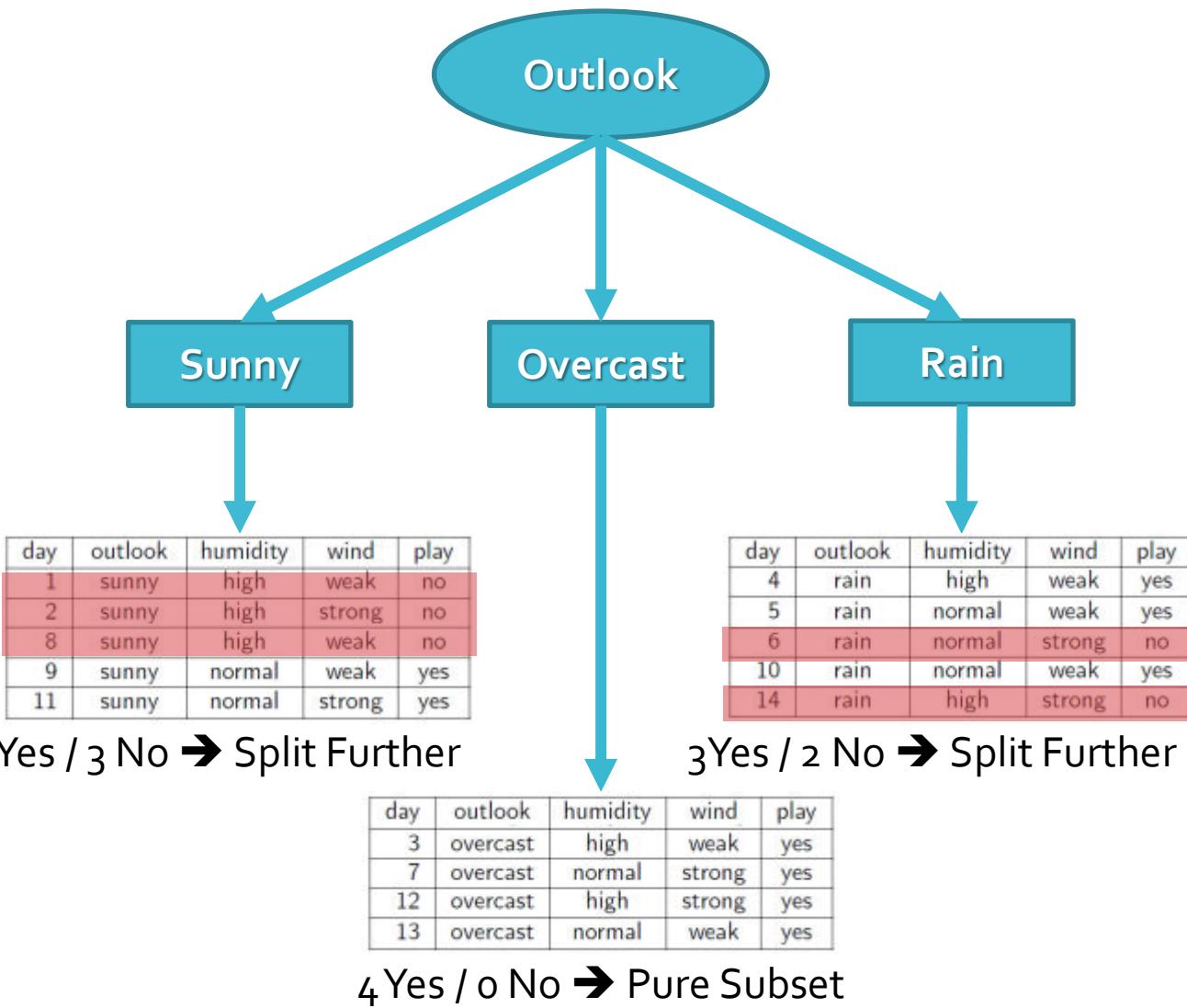
- The obtained value of information gain are :

$$IG(S, \text{wind}) = 0,020$$

$$IG(S, \text{humidity}) = 0,971$$

- “Humidity” has the highest IG value → it is set as a internal node for left side





Growing The Tree

Look at feature

“wind” $\in \{weak, strong\}$

Level 2, right node: $S = [3Y, 2N]$

day	outlook	humidity	wind	play
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
10	rain	normal	weak	yes
14	rain	high	strong	no

2 Yes / 3 No → Split Further

$$\text{Entropi : } H(S) = -\left(\frac{3}{5}\right) \log_2 \left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) \log_2 \left(\frac{2}{5}\right) = 0,971$$

$$S_{weak} = [3Y, 0N] \rightarrow H(S_{weak}) = -\left(\frac{3}{3}\right) \log_2 \left(\frac{3}{3}\right) - (0) \log_2(0) = 0$$

$$S_{strong} = [0Y, 2N] \rightarrow H(S_{strong}) = -(0) \log_2(0) - \left(\frac{2}{2}\right) \log_2 \left(\frac{2}{2}\right) = 0$$

$$IG(S, wind) = H(S) - \frac{|S_{weak}|}{|S|} H(S_{weak}) - \frac{|S_{strong}|}{|S|} H(S_{strong})$$

$$= 0,971 - \frac{3}{5} * 0 - \frac{2}{5} * 0 = 0,971$$

Growing The Tree

Look at feature

“humidity” $\in \{high, normal\}$

Level 2, right node: $S = [3Y, 2N]$

day	outlook	humidity	wind	play
4	rain	high	weak	yes
5	rain	normal	weak	yes
6	rain	normal	strong	no
10	rain	normal	weak	yes
14	rain	high	strong	no

2 Yes / 3 No → Split Further

$$\text{Entropi : } H(S) = -\left(\frac{3}{5}\right) \log_2 \left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) \log_2 \left(\frac{2}{5}\right) = 0,971$$

$$S_{high} = [1Y, 1N] \rightarrow H(S_{high}) = -\left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) - \left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) = 1$$

$$S_{normal} = [2Y, 1N] \rightarrow H(S_{normal}) = -\left(\frac{2}{3}\right) \log_2 \left(\frac{2}{3}\right) - \left(\frac{1}{3}\right) \log_2 \left(\frac{1}{3}\right) = 0,92$$

$$IG(S, \text{humidity}) = H(S) - \frac{|S_{high}|}{|S|} H(S_{high}) - \frac{|S_{normal}|}{|S|} H(S_{normal})$$

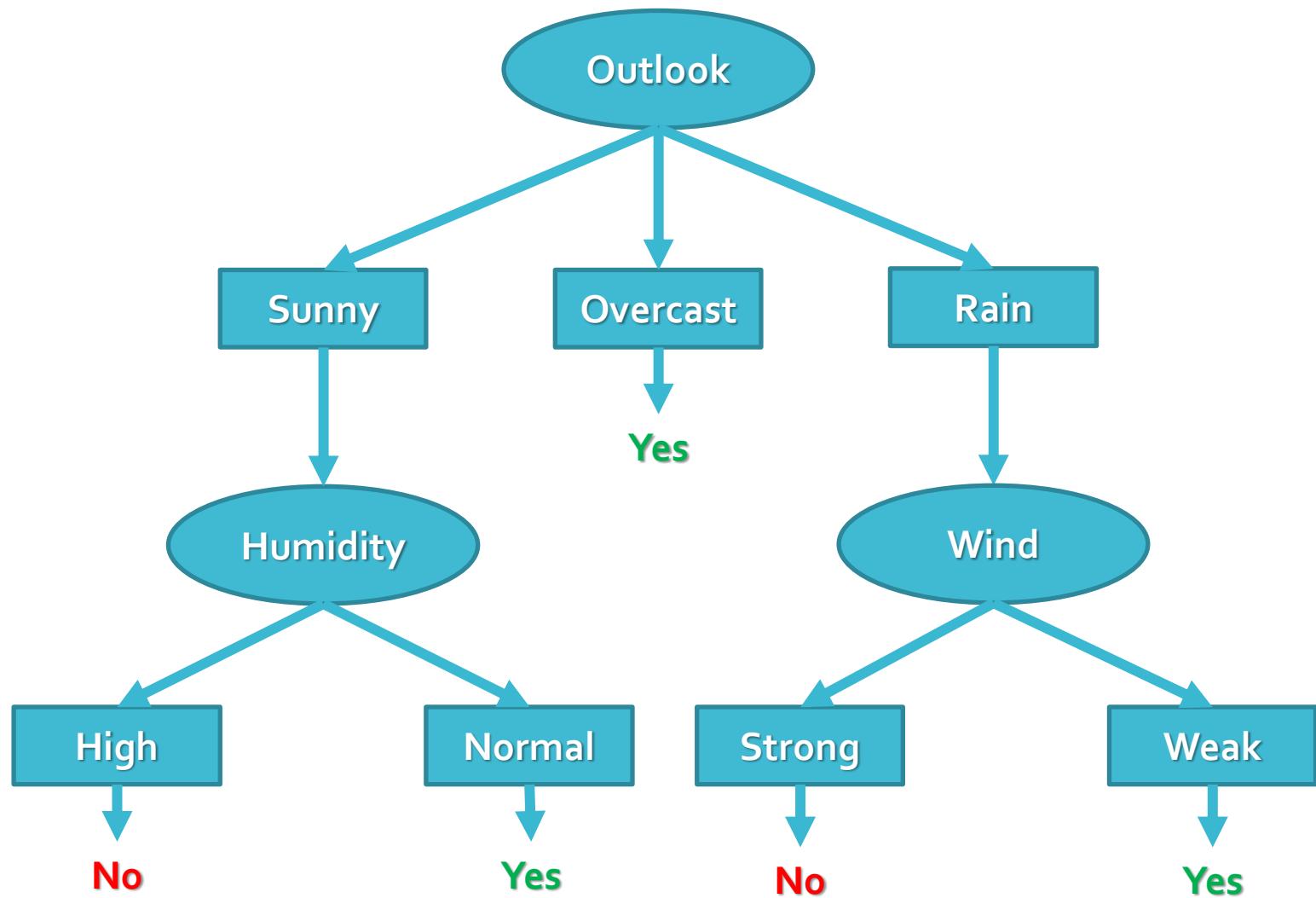
$$= 0,97 - \frac{2}{5} * 1 - \frac{3}{5} * 0,92 = 0,020$$

- The obtained value of information gain are :

$$IG(S, \text{wind}) = 0,971$$

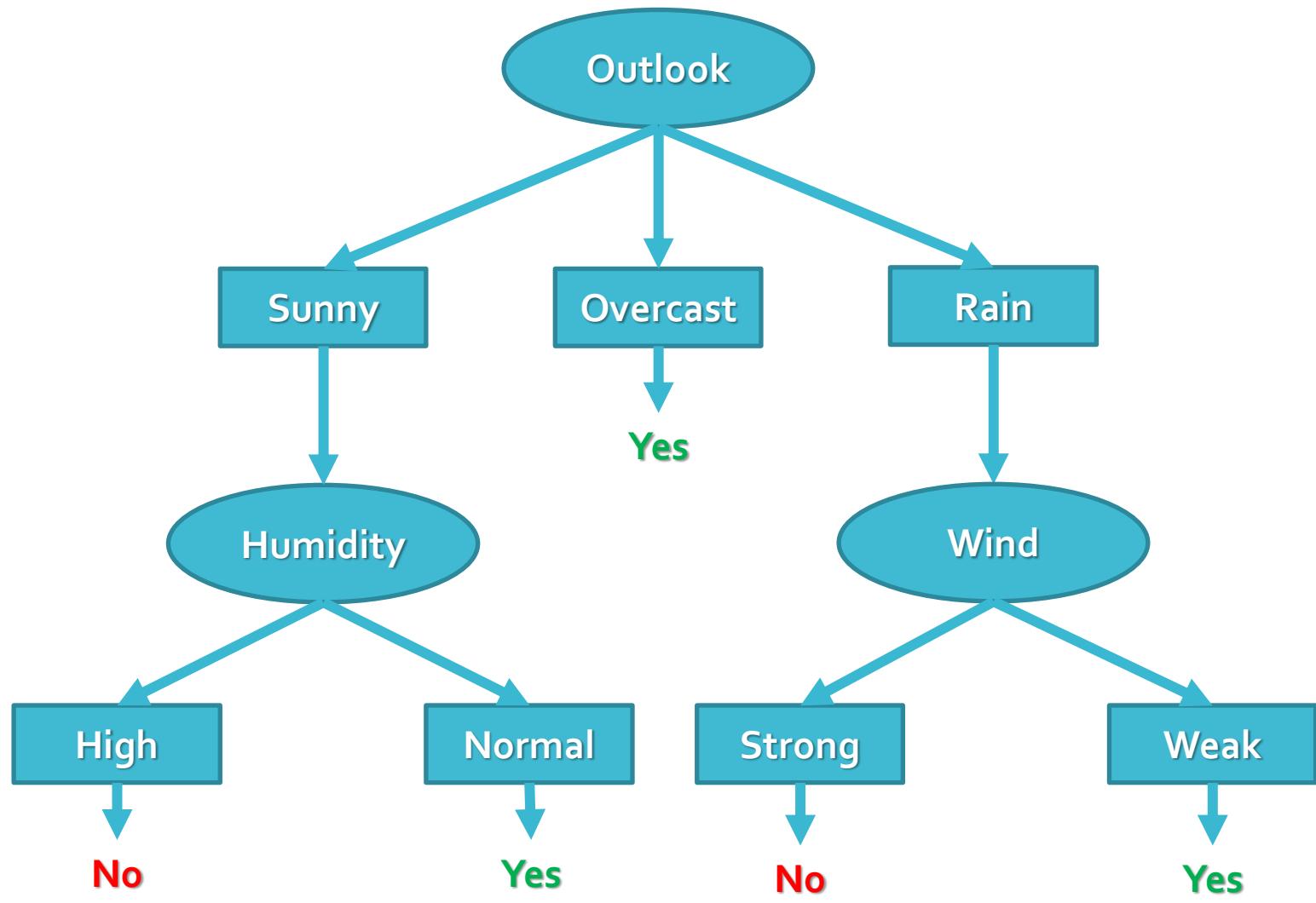
$$IG(S, \text{humidity}) = 0,020$$

- “Wind” has the highest IG value → it is set as a internal node for ride side



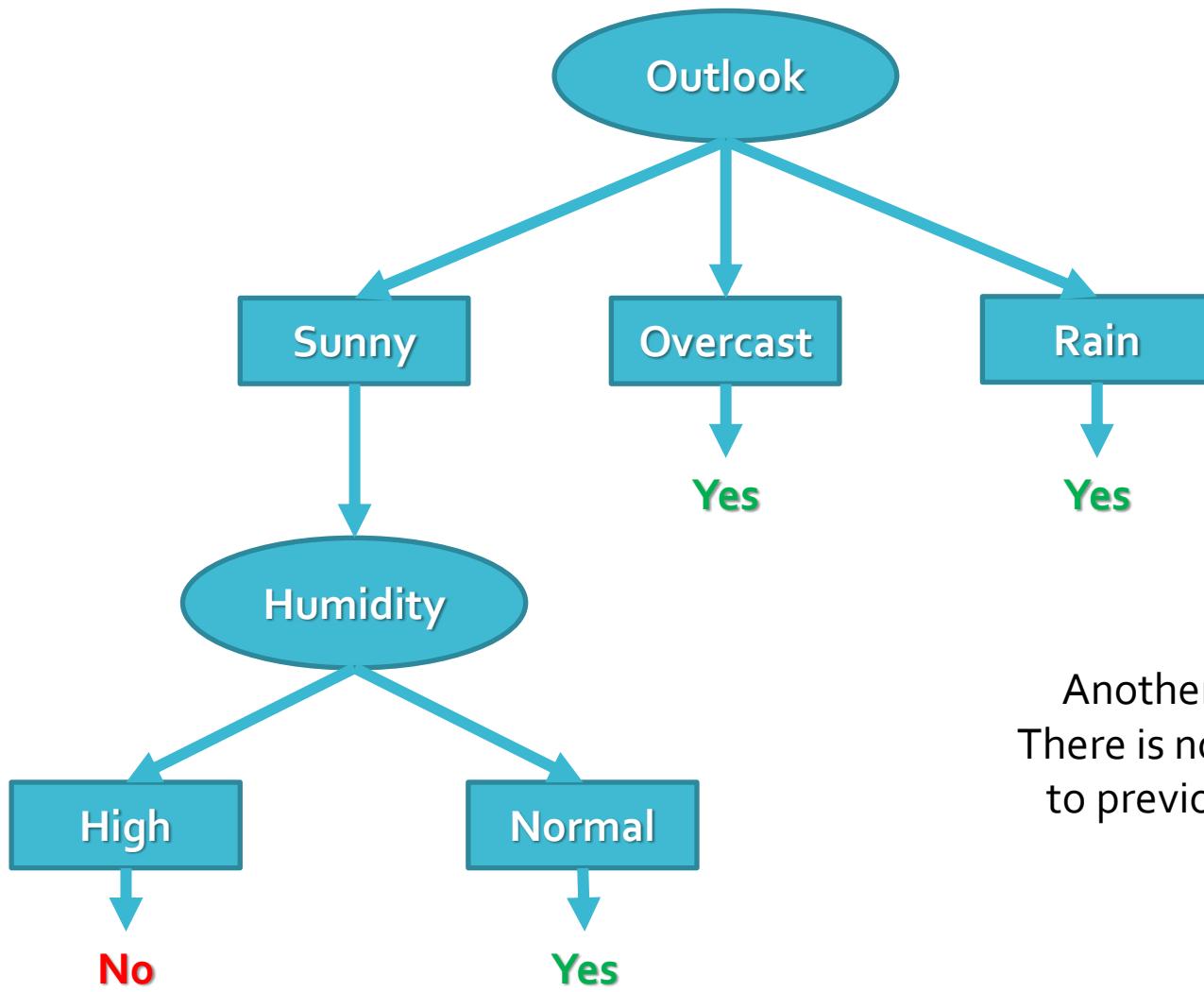
From Tree to Rules

Decision Tree



CONJUNCTION

IF Outlook = Sunny \wedge Humidity = High THEN PlayTennis = No



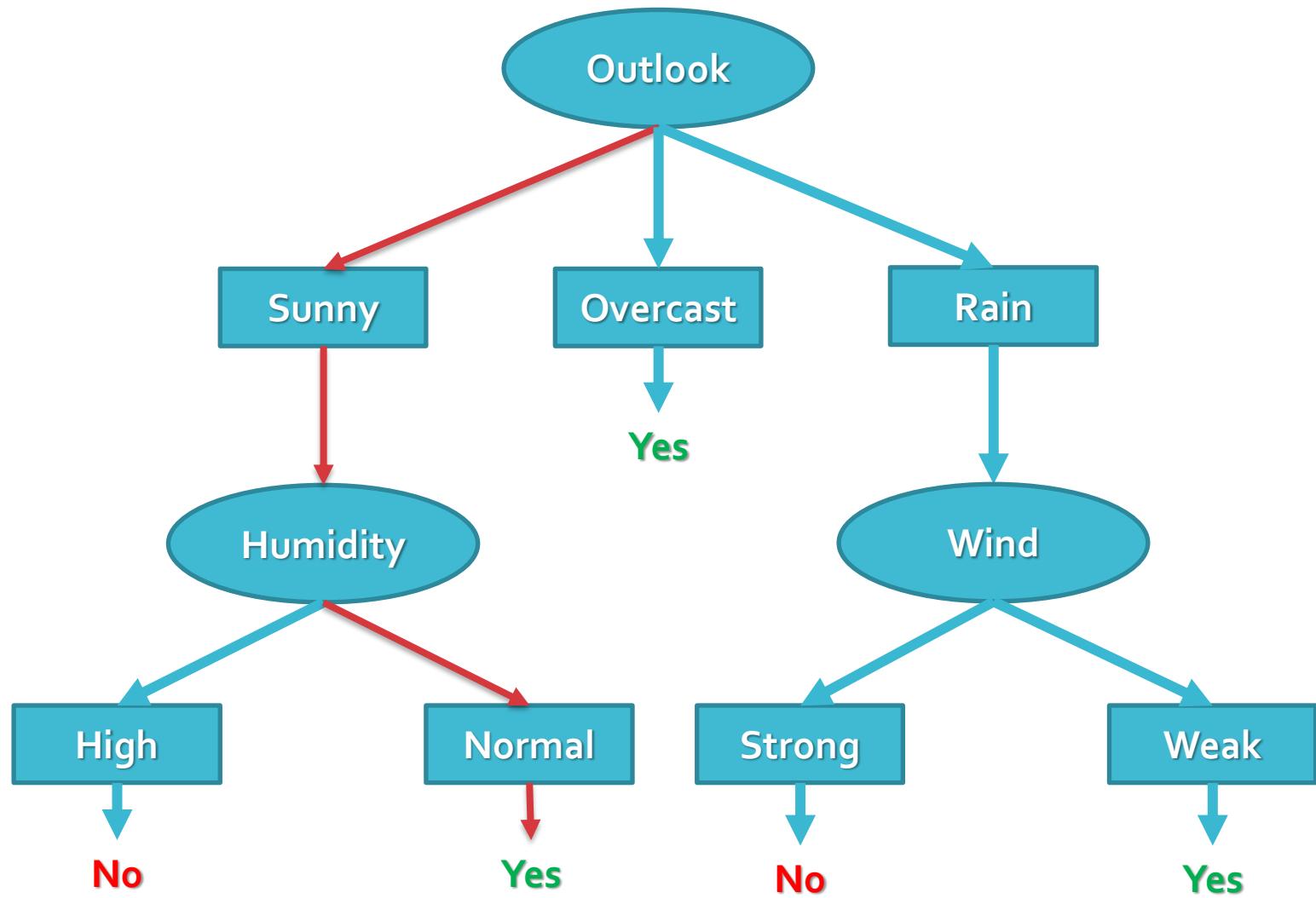
Another Example :
There is no relationship
to previous example

DISJUNCTION

IF Outlook = Overcast \vee Outlook = Rain THEN PlayTennis = Yes

Overfitting

Decision Tree

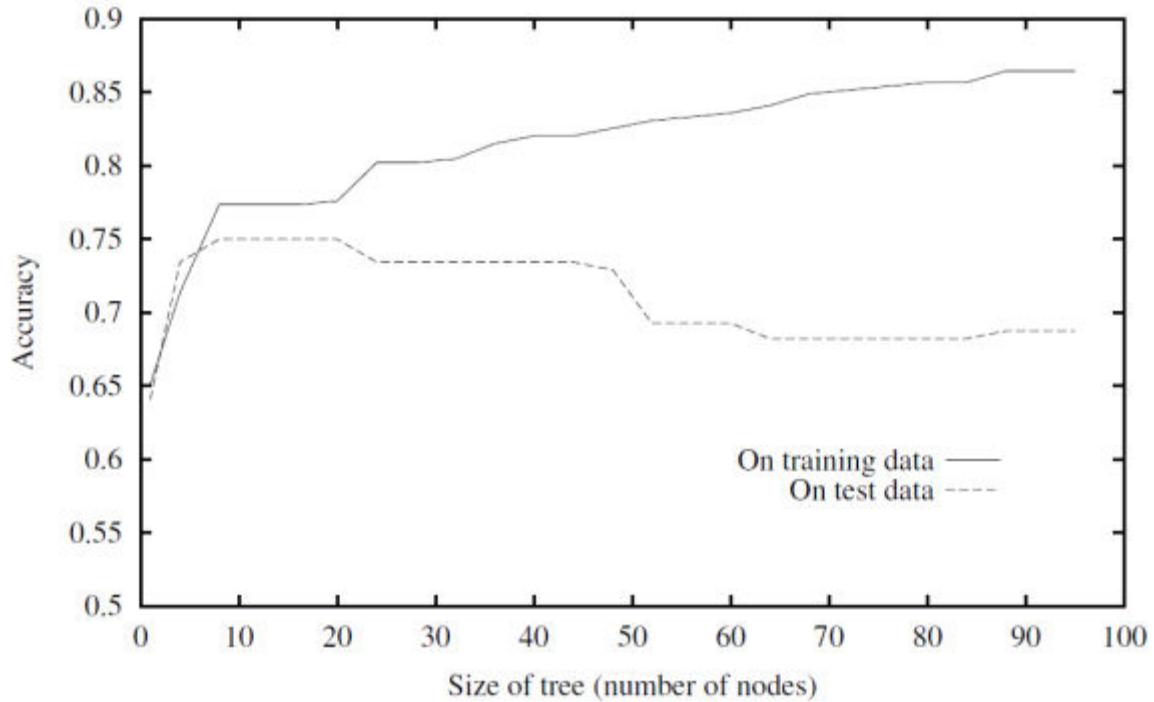


If we add noisy data to the dataset?

For example :

- Outlook=Sunny, Humidity=Normal, Wind=Strong, **PlayTennis = No**
- **Based on the tree, it will be classified as PlayTennis = Yes**

- This node will need to be expanded by testing some other feature
- The new tree would be more complex than the earlier one (trying to fit noise)
- The extra complexity may not be worth it ⇒ may lead to overfitting if the test data follows the same pattern as our normal training data
- Note: Overfitting may also occur if the training data is not sufficient



High training data accuracy doesn't necessarily imply high test data accuracy

Avoid Overfitting

- Desired: a DT that is not too big in size, yet fits the training data reasonably
- Mainly two approaches:
 - Prune while building the tree (**stopping early**)
 - Prune after building the tree (**post-pruning**)

- Criteria for judging which nodes could potentially be pruned
 - Use a validation set (separate from the training set)
 - Prune each possible node that doesn't hurt the accuracy on the validation set
 - Greedily remove the node that improves the validation accuracy the most
 - Stop when the validation set accuracy starts worsening

C4.5 Algorithm

Decision Tree

C4.5 Algorithm

- Motivation :
 - There is a limitation on ID3 Algoritm, i.e. overly sensitive to features with large numbers of values
 - Splitting attribute is evaluated by using gain ratio impurity method

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)}$$
$$SplitInfo(S, A) = - \sum_{i=1}^c \frac{S_i}{S} \log_2 \frac{S_i}{S}$$

Dimana S_1 to S_c adalah jumlah kasus untuk masing-masing nilai dari fitur A

Example: Attribute values on Continuous Interval

Decision Tree

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sun	Hot	85	Low	No
D2	Sun	Hot	90	High	No
D3	Overcast	Hot	78	Low	Yes
D4	Rain	Sweet	96	Low	Yes
D5	Rain	Cold	80	Low	Yes
D6	Rain	Cold	70	High	No
D7	Overcast	Cold	65	High	Yes
D8	Sun	Sweet	95	Low	No
D9	Sun	Cold	70	Low	Yes
D10	Rain	Sweet	80	Low	Yes
D11	Sun	Sweet	70	High	Yes
D12	Overcast	Sweet	90	High	Yes
D13	Overcast	Hot	75	Low	Yes
D14	Rain	Sweet	80	High	No

Sort Values of Humidity

$$= \{65, 70, 70, 70, 75, 78, 80, 80, 80, 85, 90, 90, 95, 96\}$$

Get distinct values

$$= \{65, 70, 75, 78, 80, 85, 90, 95, 96\}$$

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sun	Hot	85	Low	No
D2	Sun	Hot	90	High	No
D3	Overcast	Hot	78	Low	Yes
D4	Rain	Sweet	96	Low	Yes
D5	Rain	Cold	80	Low	Yes
D6	Rain	Cold	70	High	No
D7	Overcast	Cold	65	High	Yes
D8	Sun	Sweet	95	Low	No
D9	Sun	Cold	70	Low	Yes
D10	Rain	Sweet	80	Low	Yes
D11	Sun	Sweet	70	High	Yes
D12	Overcast	Sweet	90	High	Yes
D13	Overcast	Hot	75	Low	Yes
D14	Rain	Sweet	80	High	No

Interval	≤ 65	> 65
Yes	1	8
No	0	5

$$H(S) = -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) \\ = 0,94$$

$$S_{\leq 65} = [1Y, 0N] \rightarrow H(S_{\leq 65}) = -\left(\frac{1}{1}\right) \log_2 \left(\frac{1}{1}\right) - \left(\frac{0}{1}\right) \log_2 \left(\frac{0}{1}\right) = 0$$

$$S_{> 65} = [8Y, 5N] \rightarrow H(S_{> 65}) = -\left(\frac{8}{13}\right) \log_2 \left(\frac{8}{13}\right) - \left(\frac{5}{13}\right) \log_2 \left(\frac{5}{13}\right) = 0,961$$

$$Gain(S) = 0,94 - \left(\frac{1}{14}\right) * 0 - \left(\frac{13}{14}\right) * 0,9641 = 0,94 - 0,892 = 0,048$$

Perform against other threshold values

Data Representation

Decision Tree

DATA TO FEATURES

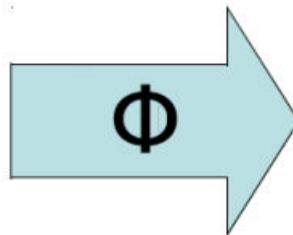
- Most learning algorithms require the data in some numeric representation (e.g., each input pattern is a vector)
- If the data naturally has numeric (real-valued) features, one way is to just represent it as a vector of real numbers
 - E.g., a 28×28 image by a 784×1 vector of its pixel intensities.
- What if the data has a non-numeric representation
 - E.g., An email (a text document)

Data to Features: A Text Document

A possible feature vector representation for a text document

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...



W=dear	:	1
W=sir	:	1
W=this	:	2
...		
W=wish	:	0
...		
MISSPELLED	:	2
NAMELESS	:	1
ALL_CAPS	:	0
NUM_URLS	:	0
...		

Data to Features: Symbolic/Categorical/Nominal Features

- Let's consider a dataset similar to the Tennis Playing example
- Features are nominal (Low/High, Yes/No, Overcast/Rainy/Sunny, etc.)
- Features with only 2 possible values can be represented as 0/1
- What about features having more than 2 possible values?
- Can't we just map Sunny to 0, Overcast to 1, Rainy to 2?

Y	Out	T	R
P	Sunny	Low	Yes
N	Sunny	High	Yes
N	Sunny	High	No
P	Overcast	Low	Yes
P	Overcast	High	No
P	Overcast	Low	No
N	Rainy	Low	Yes
P	Rainy	Low	No



Y	$\langle \text{Out} , \text{T} , \text{R} \rangle$
1	$\langle ? , 0 , 1 \rangle$
0	$\langle ? , 1 , 1 \rangle$
0	$\langle ? , 1 , 0 \rangle$
1	$\langle ? , 0 , 1 \rangle$
1	$\langle ? , 1 , 0 \rangle$
1	$\langle ? , 0 , 0 \rangle$
0	$\langle ? , 0 , 1 \rangle$
1	$\langle ? , 0 , 0 \rangle$

- Well, we could map Sunny to 0, Overcast to 1, Rainy to 2
- But such a mapping may not always be appropriate
 - Imagine color being a feature in some data
 - Let's code 3 possible colors as Red=0, Blue=1, Green=2
 - This implies Red is more similar to Blue than to Green

- Solution: For a feature with $K > 2$ possible values, we usually create K binary features, one for each possible value

Y	Out	T	R		Y	(S? , O? , R? , T , R)
P	Sunny	Low	Yes		1	(1 , 0 , 0 , 0 , 1)
N	Sunny	High	Yes		0	(1 , 0 , 0 , 1 , 1)
N	Sunny	High	No		0	(1 , 0 , 0 , 1 , 0)
P	Overcast	Low	Yes	→	1	(0 , 1 , 0 , 0 , 1)
P	Overcast	High	No		1	(0 , 1 , 0 , 1 , 0)
P	Overcast	Low	No		1	(0 , 1 , 0 , 0 , 0)
N	Rainy	Low	Yes		0	(0 , 0 , 1 , 0 , 1)
P	Rainy	Low	No		1	(0 , 0 , 1 , 0 , 0)



Bayesian Learning

Dr. Retno Kusumaningrum, S.Si., M.Kom.

Background

Bayes in
Machine
Learning ?



In Machine Learning?

- Interested in determining **the best hypothesis** from space H
- Given the observed training data D



The Best Hypothesis :

The most probable hypothesis given the data D plus any initial knowledge about the prior of the various hypothesis in H

BAYES THEOREM :

Provide a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself

Bayes Theorem

- **Goal:** To determine the most probable hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H .
- **Prior probability of h , $P(h)$:** it reflects any background knowledge we have about the chance that h is a correct hypothesis (before having observed the data).
- **Evidence of observation D , $P(D)$:** it reflects the probability that training data D will be observed given no knowledge about which hypothesis h holds.
- **Conditional Probability of observation D , $P(D|h)$:** it denotes the probability of observing data D given some world in which hypothesis h holds.

(cont')

- **Posterior probability of h , $P(h|D)$:** it represents the probability that h holds given the observed training data D .
 - It reflects our confidence that h holds after we have seen the training data D
 - It is the quantity that Machine Learning researchers are interested in.
- **Bayes Theorem** allows us to compute $P(h|D)$:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Bayes Theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Generally, It can be defined as follow :

$$\textit{Posterior} \propto \frac{\textit{likelihood} \times \textit{prior}}{\textit{evidence}}$$

evidence bisa dianggap sebagai faktor skala sehingga hasil penjumlahan posterior sama dengan satu

Bayesian Rules

$$P(h_j|D) = \frac{P(D|h_j)P(h_j)}{P(D)}$$

Jika terdapat 2 kelas data yaitu h_1 dan h_2 maka:

$$P(D) = P(D|h_1)P(h_1) + P(D|h_2)P(h_2)$$

$$P(h_1|D) = \frac{P(D|h_1)P(h_1)}{P(D|h_1)P(h_1) + P(D|h_2)P(h_2)}$$

$$P(h_2|D) = \frac{P(D|h_2)P(h_2)}{P(D|h_1)P(h_1) + P(D|h_2)P(h_2)}$$

Maximum A Posteriori
(MAP)

Bayes Learning

Maximum Likelihood
(ML)



Maximum A Posteriori (MAP)

- Berlaku untuk kondisi dimana kita memiliki beberapa alternatif hipotesis $h \in H$
- **Tujuan :**
 - Mendapatkan hipotesis yang paling mungkin h dari sekumpulan hipotesis kandidat H berdasarkan data observasi D
- **MAP Hypothesis (h_{MAP})**

$$\begin{aligned} h_{MAP} &= \operatorname{argmax}_{h \in H} P(h|D) \\ &= \operatorname{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \operatorname{argmax}_{h \in H} P(D|h)P(h) \end{aligned}$$

Maximum Likelihood (ML)

- Berlaku ketika dalam beberapa kasus kita akan mengasumsikan bahwa setiap hipotesis h dalam H mempunyai probabilitas prior yang sama, yakni :

$$P(h_i) = P(h_j), \forall h_i, h_j \in H$$

- Berdasarkan persamaan sebelumnya (h_{MAP}) maka kita bisa menyederhanakannya menjadi:

$$h_{ML} = \operatorname{argmax}_{h \in H} P(D|h)$$

Seperti yang kita ketahui bahwa $P(D|h)$ adalah *likelihood*, sehingga persamaan h_{ML} di atas selanjutnya sering disebut sebagai *maximum likelihood*.

Dalam konteks *machine learning* (*supervised learning*) perlu diingat bahwa D merupakan *data training* atau data pembelajaran dan h adalah kelas atau label data

Contoh Kasus 1

Terdapat permasalahan penentuan diagnosis penyakit kanker dengan dua hipotesis, yakni :

- Pasien mengidap penyakit kanker (h_1)
- Pasien tidak mengidap penyakit kanker (h_2)

Data yang tersedia pada uji laboratorium menunjukkan dua kemungkinan, yakni positif kanker ($D=p$) dan negatif kanker ($D=n$)

Selain itu terdapat informasi prior bahwa untuk keseluruhan populasi hanya terdapat 0.8% menderita kanker.

Hasil uji laboratorium uji lab mengembalikan hasil positif yang benar hanya 98% dari kasus dimana penyakit kanker benar-benar terjadi dan hasil negatif yang benar hanya 97% dari kasus dimana penyakit kanker tidak terjadi.

Jika seorang pasien datang menjalani uji laboratorium dengan hasil positif , maka bagaimana diagnosa terhadap pasien tersebut?

(lanjutan 1)

- Probabilitas *prior* (h_1 - mengidap kanker; h_2 - tidak mengidap kanker)
 - $P(h_1) = 0.8\% = 0.008 \rightarrow P(h_2) = 0.992$
- Data uji laboratorium (*likelihood*):
 - $P(D = p|h_1) = 0.98$ maka $P(D = n|h_1) = 0.02$
 - $P(D = n|h_2) = 0.97$ maka $P(D = p|h_2) = 0.03$

(lanjutan 2)

- Posterior Distribution :
 - $P(h_1|D = p)$
 $= P(D = p|h_1)P(h_1)$
 $= 0.98 \times 0.008 = 0.00784$
 - $P(h_2|D = p)$
 $= P(D = p|h_2)P(h_2)$
 $= 0.03 \times 0.992 = 0.02976$
- Berdasarkan data di atas maka $P(h_2|D = p) > P(h_1|D = p)$ sehingga h_{MAP} – *tidak kanker* dengan probabilitas posterior secara lengkap
$$P(h_1|D = p) = \frac{0.00784}{0.00784 + 0.02976} = 0.79$$

Contoh Kasus 2

~ Diskrit dan Multivariat ~

- Jika diketahui 40 buah data observasi yang menunjukkan gejala seseorang terkena penyakit demam berdarah dengan 3 gejala yakni adanya bintik merah (R), suhu badan tinggi (T), dan pusing (G). Masing-masing gejala tersebut dapat bernilai 1 jika seseorang mengalami gejala tersebut atau 0 jika berlaku sebaliknya.
- Jika diberikan data baru dengan gejala suhu tubuh tinggi dan tetapi tidak merasa pusing maka data tersebut menunjukkan apa?

R	T	G	C
0	1	0	N
1	1	1	P
0	0	0	N
1	0	0	P
1	1	0	P
0	0	0	N
1	1	1	N
0	1	1	P
1	0	0	N
1	1	1	P
1	0	1	N
1	0	1	P
1	1	0	P
0	1	0	P
0	1	0	N
1	1	1	P
0	0	0	N
1	1	0	P
0	1	1	N
1	0	1	P
0	1	0	N
0	0	0	N
1	1	0	P
0	0	1	N
1	0	0	P
0	1	0	N
0	0	1	P

R	T	G	C
1	0	0	N
0	0	0	N
1	0	1	P
0	1	0	P
1	1	1	P
1	0	1	N
0	0	0	N
1	0	1	N
1	0	0	P
0	0	1	N
1	1	0	P
0	1	1	N
1	0	1	P
0	1	0	N
0	0	0	N
1	1	0	P
0	0	1	N
1	0	0	P
0	1	0	N
0	0	1	P

Solusi – Proses Training

R	T	G	$n(D h = P)$	$P(D h = P)$	$n(D h = N)$	$P(D h = N)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Solusi – Proses Training

R	T	G
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

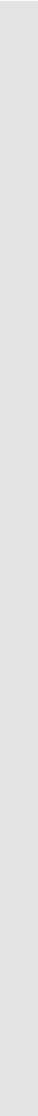
$n(h = P)$	$P(\mathbf{D} \mathbf{h} = \mathbf{P})$
1	$=1/20 = 0,05$
1	$=1/20 = 0,05$
2	$=2/20 = 0,1$
1	$=1/20 = 0,05$
3	$=3/20 = 0,15$
3	$=3/20 = 0,15$
4	$=4/20 = 0,2$
5	$=5/20 = 0,25$

$n(h = N)$	$P(\mathbf{D} \mathbf{h} = \mathbf{N})$
6	$=6/20 = 0,3$
3	$=3/20 = 0,15$
4	$=4/20 = 0,2$
1	$=1/20 = 0,05$
2	$=2/20 = 0,1$
3	$=3/20 = 0,15$
0	$=0/20 = 0$
1	$=1/20 = 0,05$

Solusi – Proses Testing

- Jika diberikan data baru dengan gejala tidak muncul bintik merah dan tidak merasa pusing, tetapi suhu tubuh tinggi maka data tersebut menunjukkan apa?

Jawab:



Soal

1

Diketahui hasil survey yang dilakukan sebuah lembaga kesehatan menyatakan bahwa 30% penduduk di dunia menderita penyakit jantung. Dari penduduk yang sakit jantung tersebut 60% adalah memiliki kolesterol tinggi, dan dari penduduk yang tidak menderita penyakit jantung 20% nya memiliki kolesterol tinggi. Dengan menggunakan MAP (Maximum Appropri Probability), hitunglah kemungkinan seseorang yang memiliki kolesterol tinggi mengidap penyakit jantung, dan kemungkinan orang tersebut tidak sakit jantung!

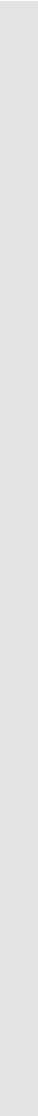
2

Diketahui suatu data sebagai berikut:

No	Outlook	Wind	Play Tennis
1	Sunny	Weak	Yes
2	Sunny	Weak	Yes
3	Rainy	Weak	No
4	Sunny	Strong	Yes
5	Rainy	Strong	No
6	Sunny	Weak	Yes
7	Sunny	Weak	No
8	Rainy	Weak	No
9	Rainy	Weak	Yes
10	Rainy	Strong	Yes

Tentukan kelas dari masing-masing data uji berikut ini !

No	Outlook	Wind
1	Sunny	Weak
2	Rainy	Weak



NEXT WEEK

Issue 1 ~ Continuous-valued Data of Single Features (Univariat)

Lebar	Buah
2.70"	Jeruk
2.52"	Jeruk
2.57"	Jeruk
2.22"	Jeruk
3.16"	Apel
3.58"	Apel
3.10"	Apel

Terdapat *training data* berisi 2 jenis buah yaitu apel dan jeruk dengan satu fitur yakni lebar buah. Adapun data-data tersebut seperti dijelaskan pada tabel di samping.

Jika terdapat buah berukuran 2.91", maka tentukan buah tersebut termasuk jeruk / apel?

Issue 2 ~ Continuous-valued Data of Multi Features (Multivariat)

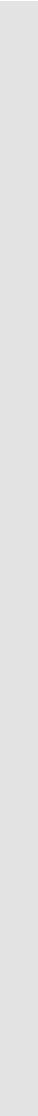
Lebar	Panjang	Ikan
2.70"	3.73"	Nila
2.52"	3.81"	Nila
2.57"	3.83"	Nila
2.22"	3.56"	Nila
3.16"	3.91"	Gurami
3.58"	4.04"	Gurami
3.10"	3.85"	Gurami

- Terdapat *training data* berisi 2 jenis ikan yaitu ikan gurami dan ikan nila dengan dua fitur yakni lebar ikan dan panjang ikan. Adapun data-data tersebut seperti dijelaskan pada tabel di samping.
- Jika terdapat ikan berukuran lebar 2.81" dan panjang 5.46", maka tentukan ikan tersebut termasuk ikan gurami / ikan nila?



Bayesian Learning

Dr. Retno Kusumaningrum, S.Si., M.Kom.



1st ISSUE

Kondisi :

- Continuous-valued Data of Single Features (Univariat)

Solusi :

- Memodelkan *likelihood* berdistribusi normal

$$P(D \mid h) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(D_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (average) of attribute values D_j of training data for which $h = h_i$

σ_{ji} : standard deviation of attribute values D_j of examples for which $h = h_i$

Contoh Soal :

Lebar	Buah
2.70"	Jeruk
2.52"	Jeruk
2.57"	Jeruk
2.22"	Jeruk
3.16"	Apel
3.58"	Apel
3.10"	Apel

Terdapat *training data* berisi 2 jenis buah yaitu apel dan jeruk dengan satu fitur yakni lebar buah. Adapun data-data tersebut seperti dijelaskan pada tabel di samping.

Jika terdapat buah berukuran 2.91", maka tentukan buah tersebut termasuk jeruk / apel?

Langkah-langkah penyelesaian

Training Process:

- Hitung rata-rata lebar jeruk dan rata-rata lebar apel

$$\overline{lebar}_{jeruk} = \frac{\sum_{j \in jeruk} lebar_j}{N_{jeruk}}$$

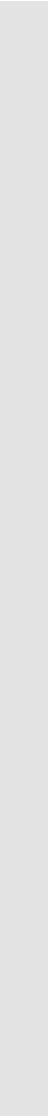
- Hitung standard deviasi dari jeruk dan standar deviasi dari apel

$$\sigma_{jeruk} = \sqrt{\frac{\sum_{j \in jeruk} (lebar_j - \overline{lebar}_{jeruk})^2}{N_{jeruk}}}$$

- Hitung probabilitas prior dari dari apel dan jeruk

Testing Process:

- Hitung probabilitas likelihood dari apel dan jeruk
- Hitung probabilitas posterior
- Tentukan keputusan akhir berdasarkan MAP



2nd ISSUE

Issue 2 ~ Continuous-valued Data of Multi Features (Multivariat)

Lebar	Panjang	Ikan
2.70"	3.73"	Nila
2.52"	3.81"	Nila
2.57"	3.83"	Nila
2.22"	3.56"	Nila
3.16"	3.91"	Gurami
3.58"	4.04"	Gurami
3.10"	3.85"	Gurami

- Terdapat *training data* berisi 2 jenis ikan yaitu ikan gurami dan ikan nila dengan dua fitur yakni lebar ikan dan panjang ikan. Adapun data-data tersebut seperti dijelaskan pada tabel di samping.
- Jika terdapat ikan berukuran lebar 2.81" dan panjang 5.46", maka tentukan ikan tersebut termasuk ikan gurami / ikan nila?

Langkah-langkah Penyelesaian

Training Process

- Labelling Dataset

$$X = \begin{bmatrix} 2.70 & 3.73 \\ 2.52 & 3.81 \\ 2.57 & 3.83 \\ 2.22 & 3.56 \\ 3.16 & 3.91 \\ 3.58 & 4.04 \\ 3.10 & 3.85 \end{bmatrix} \quad Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$



[Lanjutan]

- Memisahkan X berdasarkan kelas

$$X_1 = \begin{bmatrix} 2.70 & 3.73 \\ 2.52 & 3.81 \\ 2.57 & 3.83 \\ 2.22 & 3.56 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 3.16 & 3.91 \\ 3.58 & 4.04 \\ 3.10 & 3.85 \end{bmatrix}$$



[Lanjutan]

- Hitung nilai μ_i mean features dari kelas i dan μ mean global
- Hitung nilai *mean corrected* (nilai $X_i - \mu$) dinotasikan sebagai (X_i^0)
- Hitung matriks kovarian kelas i

$$\Sigma_i = \frac{(X_i^0)^T X_i^0}{n_i}$$

(lanjutan)

Testing Process:

- Hitung likelihood dari data



(lanjutan)

- Hitung probabilitas posterior
- Tentukan keputusan akhir berdasarkan MAP



NAÏVE BAYES

Naïve Bayes

- Bayes classification

$$P(h \mid D) \propto P(D \mid h)P(h) = P(d_1, \dots, d_n \mid h)P(h)$$

Difficulty: learning the joint probability $P(d_1, \dots, d_n \mid h)$

- Naïve Bayes classification

– Assumption that **all input attributes are conditionally independent!**

$$\begin{aligned} P(d_1, d_2, \dots, d_n \mid h) &= P(d_1 \mid d_2, \dots, d_n, h)P(d_2, \dots, d_n \mid h) \\ &= P(d_1 \mid h)P(d_2, \dots, d_n \mid h) \\ &= P(d_1 \mid h)P(d_2 \mid h) \cdots P(d_n \mid h) \end{aligned}$$

– MAP classification rule: for testing data $D' = (d'_1, d'_2, \dots, d'_n)$

$$[P(d'_1 \mid h_i) \cdots P(d'_n \mid h_i)]P(h_i) > [P(d'_1 \mid h_j) \cdots P(d'_n \mid h_j)]P(h_j), \quad h_i \neq h_j$$

- Naïve Bayes Algorithm (for discrete input attributes)
 - Learning Phase: Given a training set D ,

For each target value of $h (h = h_1, \dots, h_L)$

$P(h = h_i) \leftarrow$ estimate $P(h = h_i)$ with examples in D ;

For every attribute value δ_{jk} of each attribute d_j ($j = 1, \dots, n; k = 1, \dots, K$)

$P(d_j = \delta_{jk} | h = h_i) \leftarrow$ estimate $P(d_j = \delta_{jk} | h = h_i)$ with examples in D ;

Output: conditional probability tables; for $d_j, n \times K$ elements

- Test Phase: Given an unknown instance $D' = (d'_1, \dots, d'_m)$

Look up tables to assign the label h_i to D' if

$$[P(d'_1 | h_i) \cdots P(d'_n | h_i)]P(h_i) > [P(d'_1 | h_j) \cdots P(d'_n | h_j)]P(h_j), \quad h_i \neq h_j$$

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Bagaimana keputusan akan diambil jika terdapat *testing data* berupa informasi
Outlook=Sunny,
Temperature=Cool,
Humidity=High,
Wind=Weak

Example

- Learning Phase

Outlook	Play=Yes	Play=No
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=Yes	Play=No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=Yes	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=Yes	Play=No
Strong	3/9	3/5
Weak	6/9	2/5

$$P(\text{Play}=\text{Yes}) = 9/14 \quad P(\text{Play}=\text{No}) = 5/14$$

Example

- Test Phase
 - Given a new instance,
 $\mathbf{D}' = (\text{Outlook}=\text{Sunny}, \text{Temperature}=\text{Cool}, \text{Humidity}=\text{High}, \text{Wind}=\text{Weak})$
 - Look up tables

$$P(\text{Outlook}=\text{Sunny} | \text{Play}=\text{Yes}) = 2/9$$

$$P(\text{Temperature}=\text{Cool} | \text{Play}=\text{Yes}) = 3/9$$

$$P(\text{Humidity}=\text{High} | \text{Play}=\text{Yes}) = 3/9$$

$$P(\text{Wind}=\text{Weak} | \text{Play}=\text{Yes}) = 6/9$$

$$P(\text{Play}=\text{Yes}) = 9/14$$

$$P(\text{Outlook}=\text{Sunny} | \text{Play}=\text{No}) = 3/5$$

$$P(\text{Temperature}=\text{Cool} | \text{Play}=\text{No}) = 1/5$$

$$P(\text{Humidity}=\text{High} | \text{Play}=\text{No}) = 4/5$$

$$P(\text{Wind}=\text{Weak} | \text{Play}=\text{No}) = 2/5$$

$$P(\text{Play}=\text{No}) = 5/14$$

- MAP rule

$$P(\text{Yes} | \mathbf{D}') = [P(\text{Sunny} | \text{Yes}) P(\text{Cool} | \text{Yes}) P(\text{High} | \text{Yes}) P(\text{Weak} | \text{Yes})] P(\text{Play}=\text{Yes}) =$$

$$P(\text{No} | \mathbf{D}') = [P(\text{Sunny} | \text{No}) P(\text{Cool} | \text{No}) P(\text{High} | \text{No}) P(\text{Weak} | \text{No})] P(\text{Play}=\text{No}) =$$

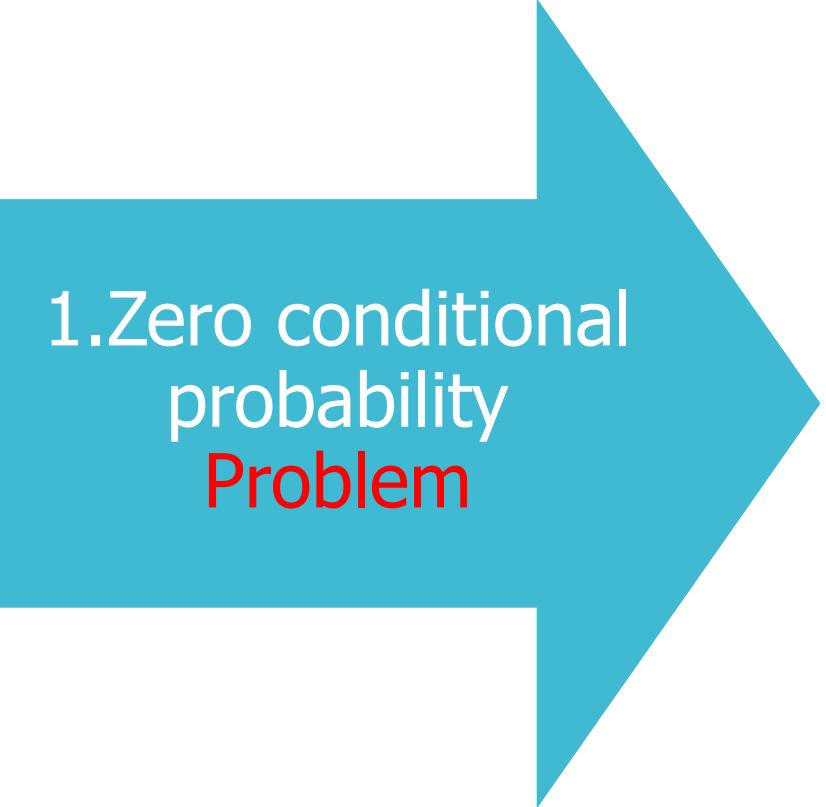
Given the fact $P(\text{Yes} | \mathbf{D}') > P(\text{No} | \mathbf{D}')$, we label \mathbf{D}' to be "Yes".

ISSUES

RELEVANT TO NAIVE BAYES



1.Violation of
Independence
Assumption



1.Zero conditional
probability
Problem

Violation of Independence Assumption

Events are correlated

- For many real world tasks $P(X_1, \dots, X_n | C) \neq P(X_1 | C) \cdots P(X_n | C)$
- Nevertheless, naïve Bayes works surprisingly well anyway!

Zero conditional probability Problem

- Such problem exists when no example contains the attribute value

$$P(X_1, \dots, X_n | C) \neq P(X_1 | C) \dots P(X_n | C)$$

- In this circumstance, $\hat{P}(x_1 | c_i) \dots \hat{P}(a_{jk} | c_i) \dots \hat{P}(x_n | c_i) = 0$ during test
- For a remedy, conditional probabilities are estimated with

$$X_j = a_{jk}, \hat{P}(X_j = a_{jk} | C = c_i) = 0$$

$$\hat{P}(X_j = a_{jk} | C = c_i) = \frac{n_c + mp}{n + m}$$

n_c : number of training examples for which $X_j = a_{jk}$ and $C = c_i$

n : number of training examples for which $C = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of X_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)

Example: $P(\text{outlook}=\text{overcast}|\text{no})=0$ in the play-tennis dataset

- Adding m "virtual" examples (m : up to 1% of #training example)
 - In this dataset, # of training examples for the "no" class is 5.
 - We can only add $m=1$ "virtual" example in our m -estimate remedy.
- The "outlook" feature can takes only 3 values. So $p=1/3$.
- Re-estimate $P(\text{outlook}|\text{no})$ with the m -estimate

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \quad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$

Another Problem **Continuous-valued Input Attributes**

- What to do in such a case?
 - Numberless values for an attribute
 - Conditional probability is then modeled with the normal distribution

$$\hat{P}(X_j | C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (average) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

- Learning Phase: for $\mathbf{X} = (X_1, \dots, X_n)$, $C = c_1, \dots, c_L$
Output: $n \times L$ normal distributions and $P(C = c_i)$ $i = 1, \dots, L$
- Test Phase: for $\mathbf{X}' = (X'_1, \dots, X'_n)$
 1. Calculate conditional probabilities with all the normal distributions
 2. Apply the MAP rule to make a decision

Example: Continuous-valued Features

- Temperature is naturally of continuous value.
Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8
No: 27.3, 30.1, 17.4, 29.5, 15.1
- Estimate mean and variance for each class

$$\mu = \frac{1}{N} \sum_{n=1}^N x_n, \quad \sigma^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \quad \sigma_{Yes} = 2.35$$

$$\mu_{No} = 23.88, \quad \sigma_{No} = 7.09$$

- **Learning Phase:** output two Gaussian models for $P(\text{temp}|\mathcal{C})$

$$\hat{P}(x | Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{2 \times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{11.09}\right)$$

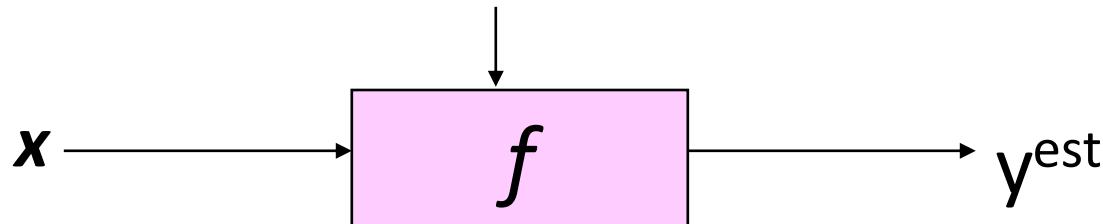
$$\hat{P}(x | No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{2 \times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{50.25}\right)$$

Support Vector Machine

*Dr. Retno Kusumaningrum,
S.Si., M.Kom.*



Linear Classifier



$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

Fungsi *signum* :

$$f = \text{sign}(x)$$

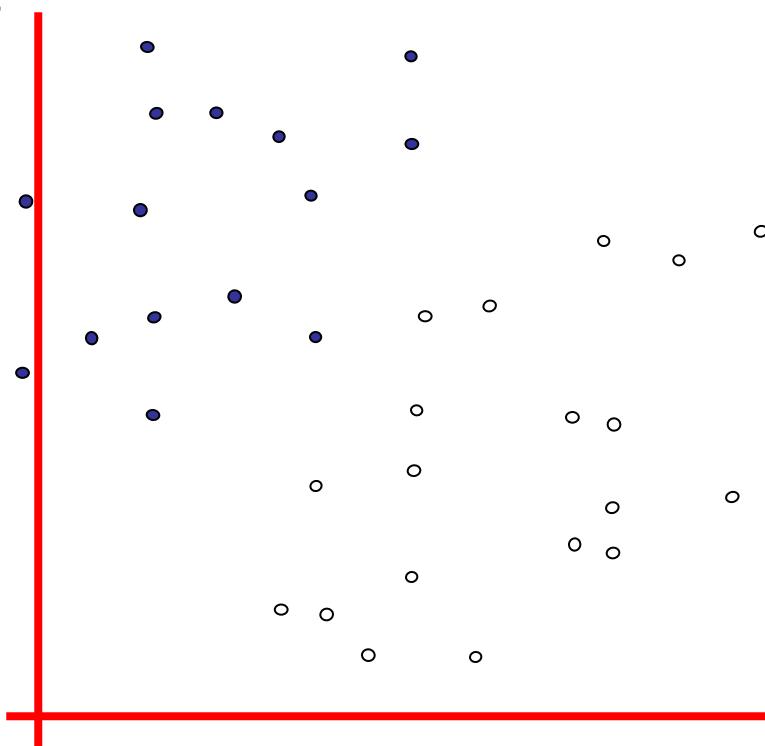
Fungsi f akan bernilai +1 jika x bernilai positif

Fungsi f akan bernilai -1 jika x bernilai negatif

Fungsi f akan bernilai 0 jika $x = 0$

Linear Classifiers

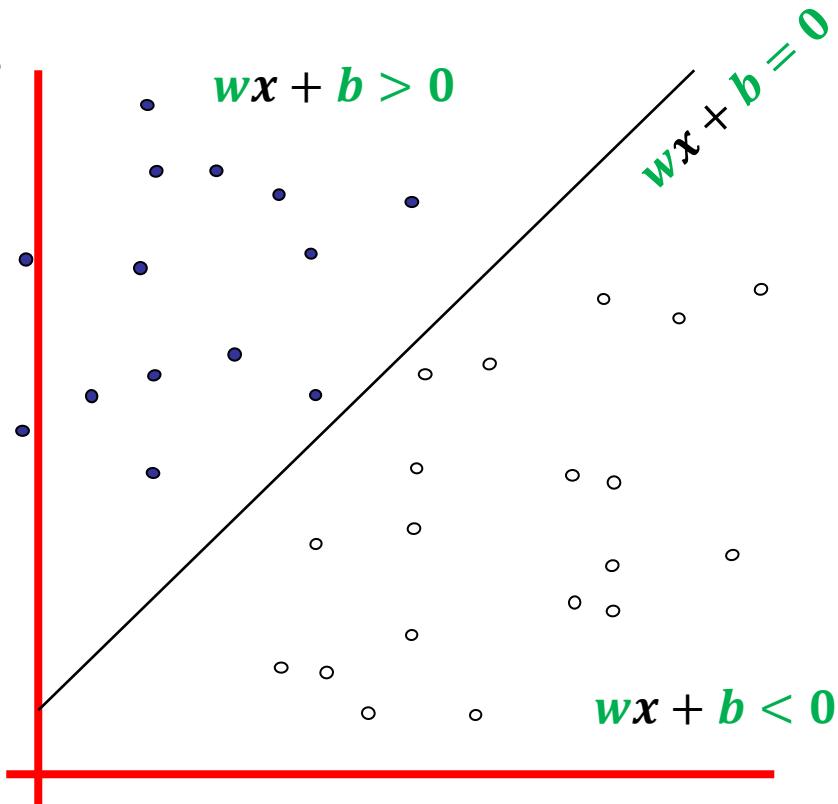
- denotes +1
- denotes -1



How would you
classify this data?

Linear Classifiers

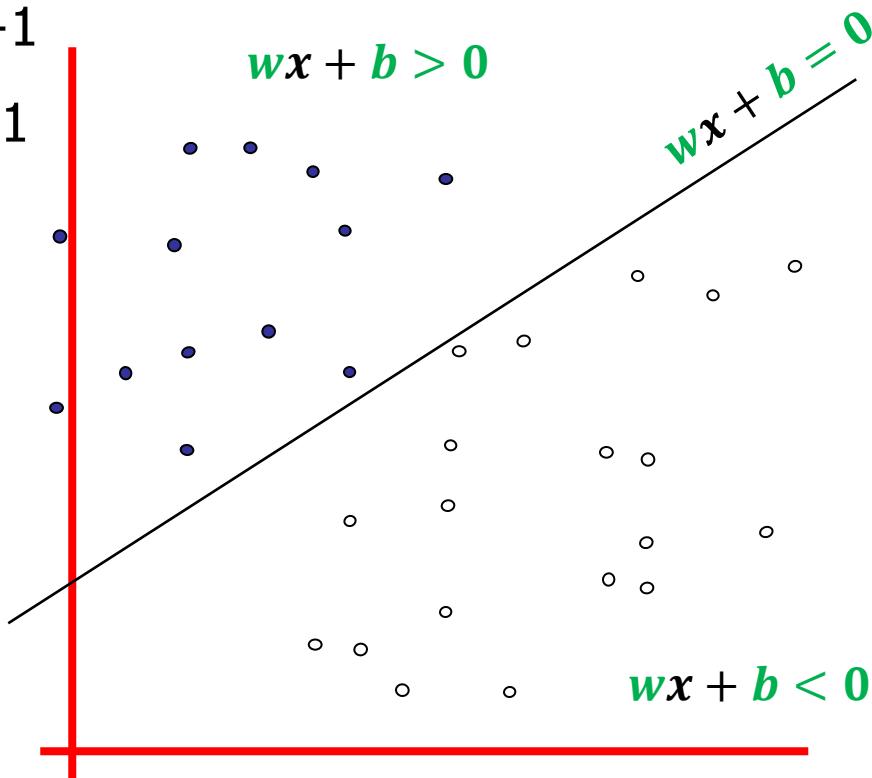
- denotes +1
- denotes -1



How would you
classify this data?

Linear Classifiers

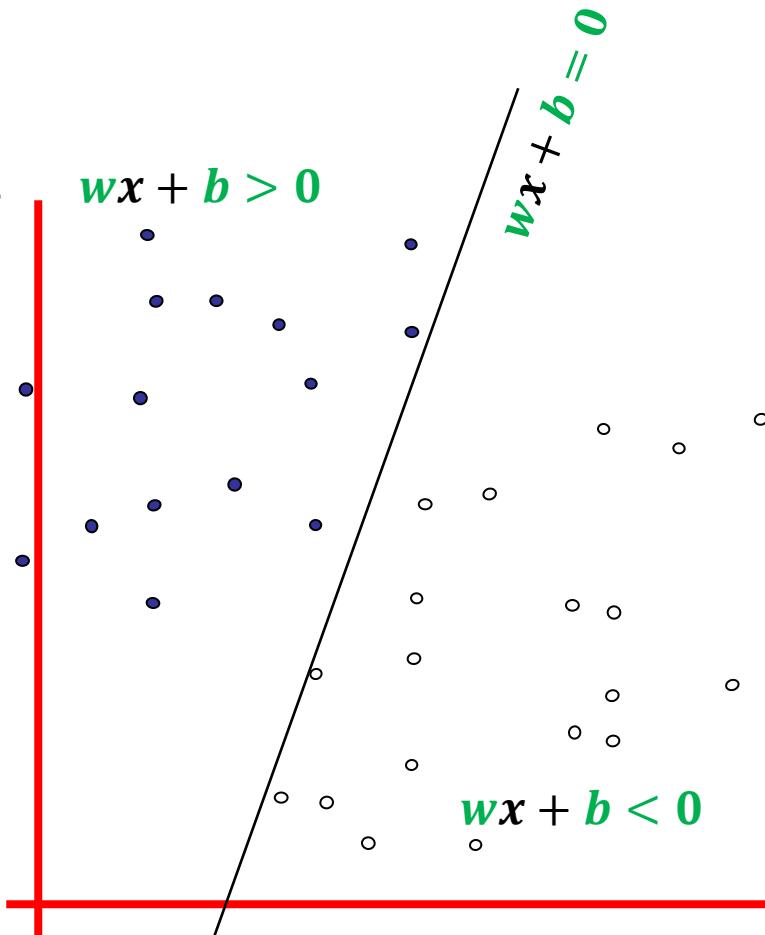
- denotes +1
- denotes -1



How would you
classify this data?

Linear Classifiers

- denotes +1
- denotes -1



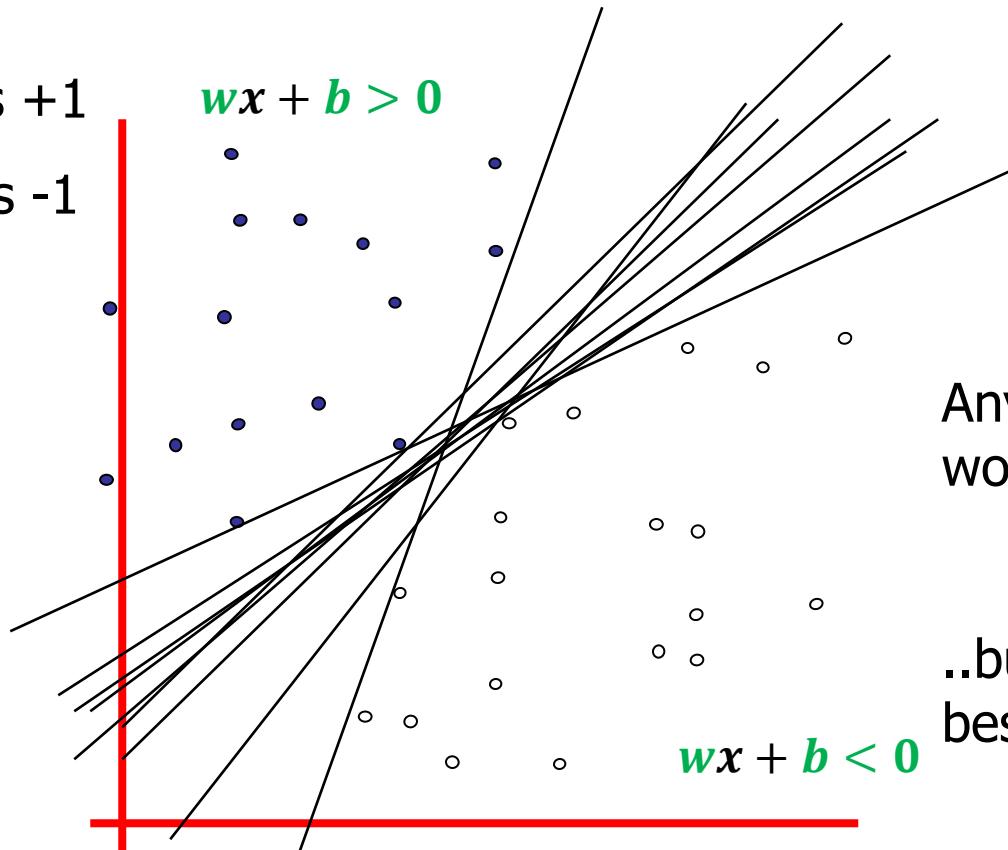
How would you
classify this data?

Linear Classifiers

- denotes +1
- denotes -1

$$\mathbf{w}x + b > 0$$

$$\mathbf{w}x + b < 0$$

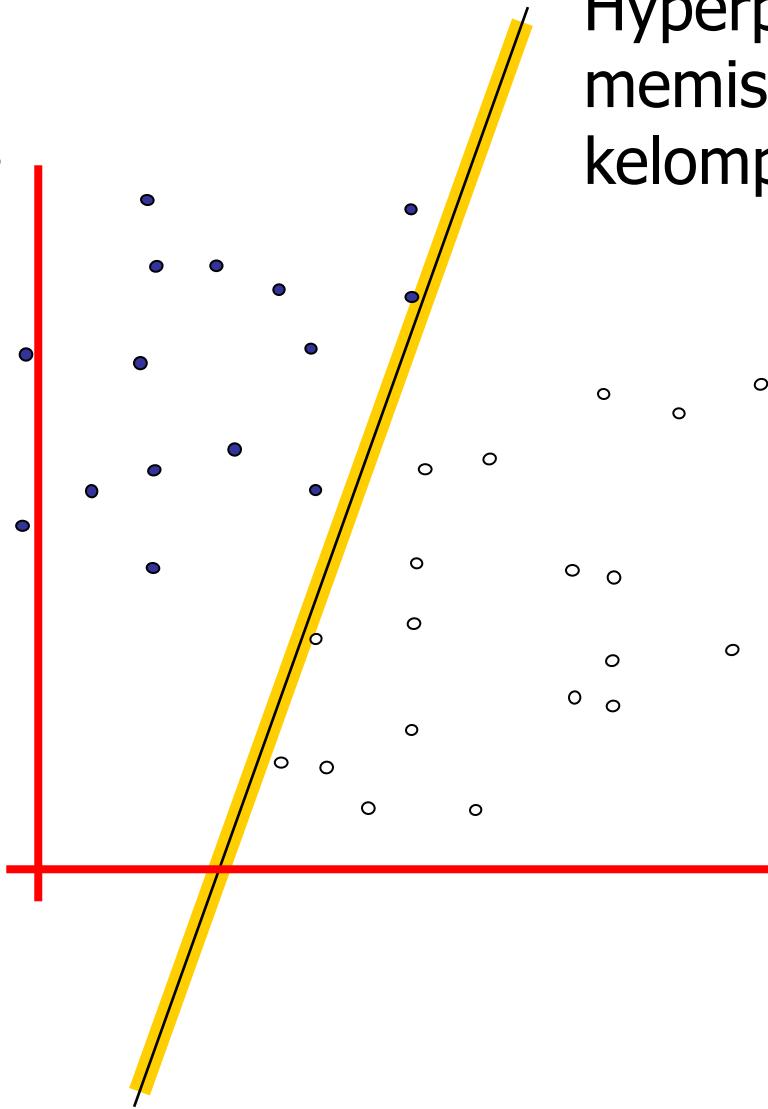


Any of these
would be fine..

..but which is
best?

Classifier Margin

- denotes +1
- denotes -1

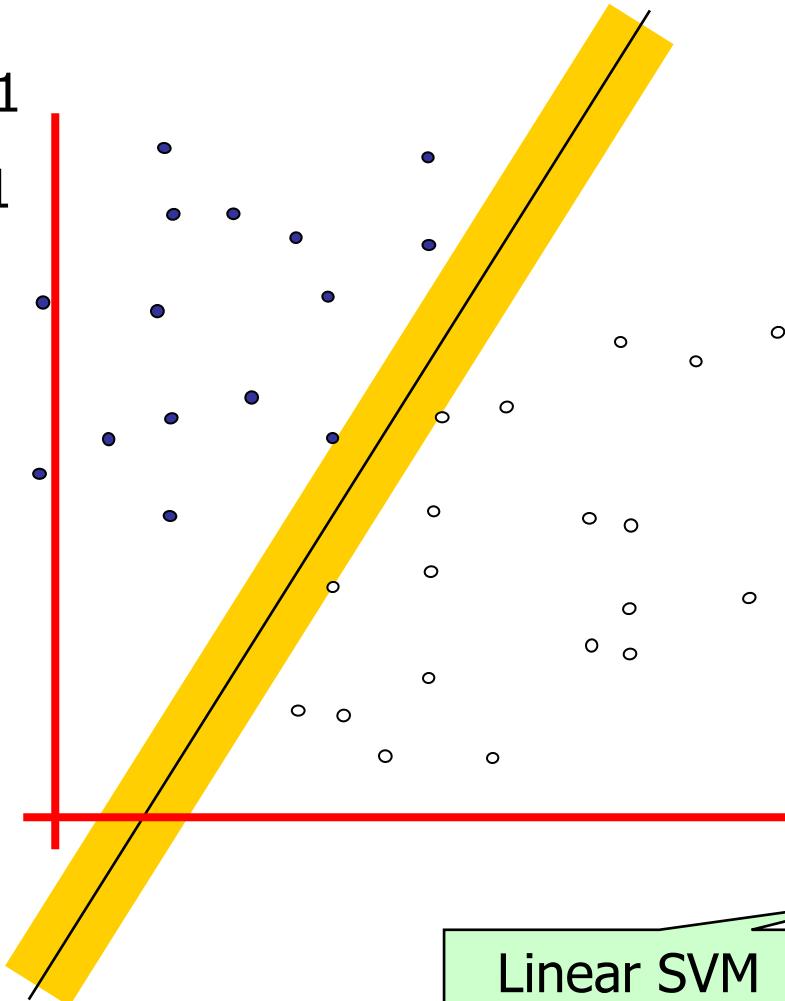


Hyperplane = garis yang memisahkan kedua kelompok

Menentukan **margin** sebagai jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing class

Maximum Margin Classifier

- denotes +1
- denotes -1



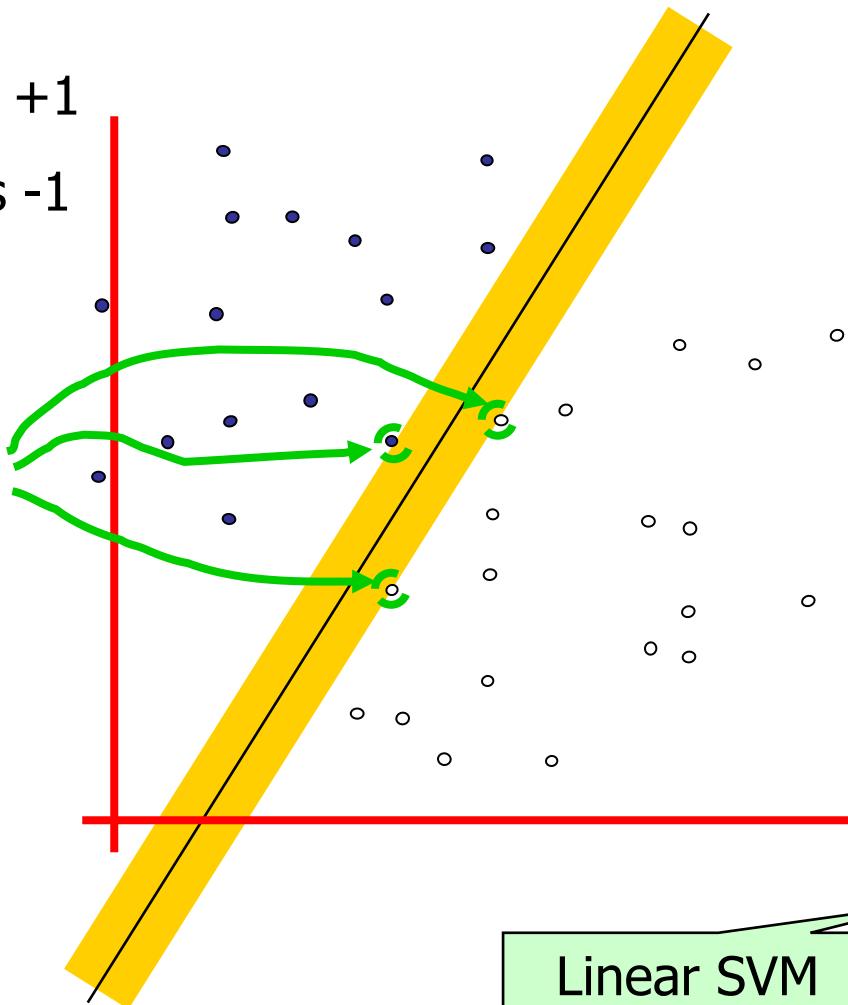
The **maximum margin linear classifier** is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Maximum Margin Classifier

- denotes +1
- denotes -1

Support Vectors
are those
datapoints that
the margin
pushes up
against

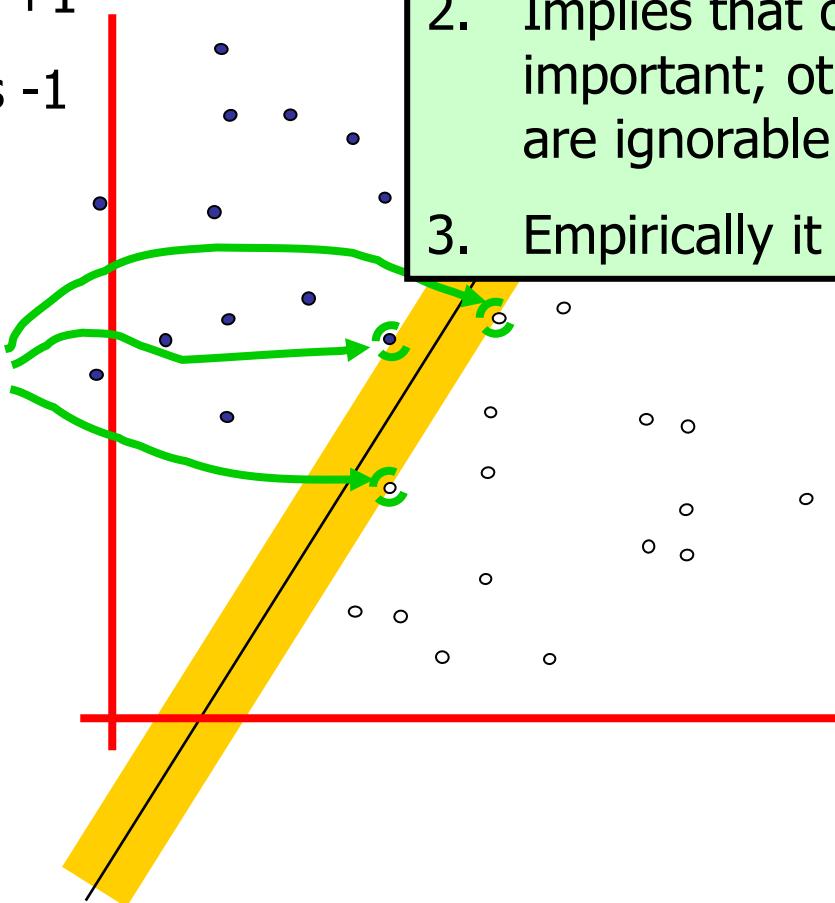


The **maximum margin linear classifier** is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Why Maximum Margin?

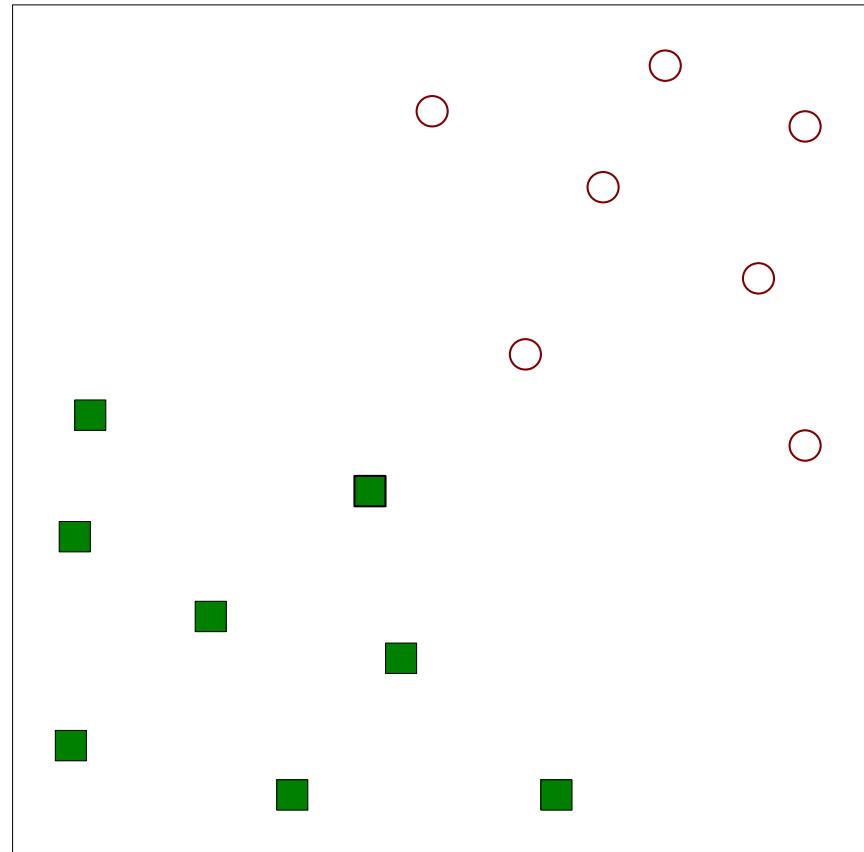
- denotes +1
 - denotes -1
- Support Vectors are those datapoints that the margin pushes up against



1. Intuitively this feels safest.
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

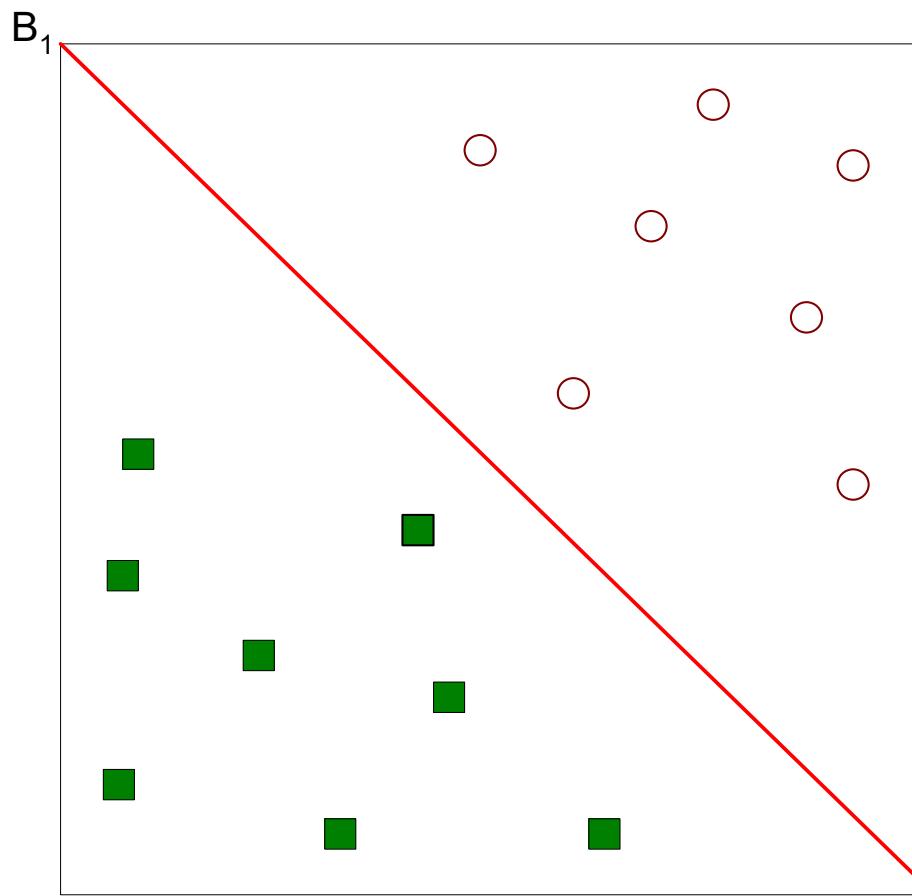
ANOTHER EXAMPLE

Support Vector Machines



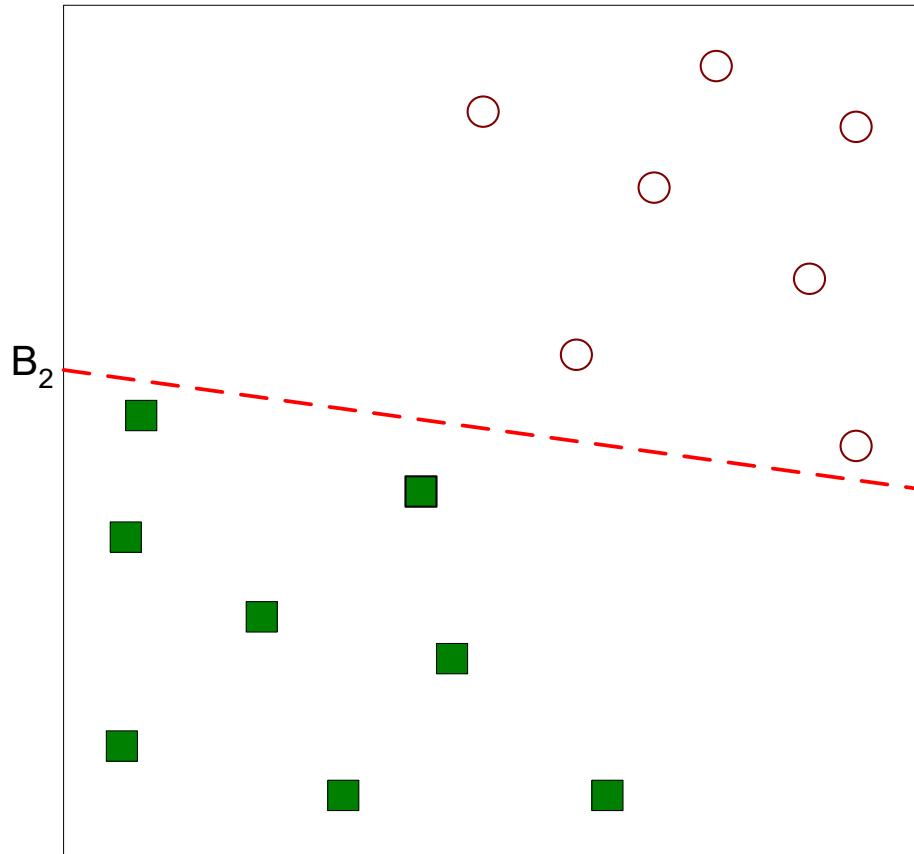
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



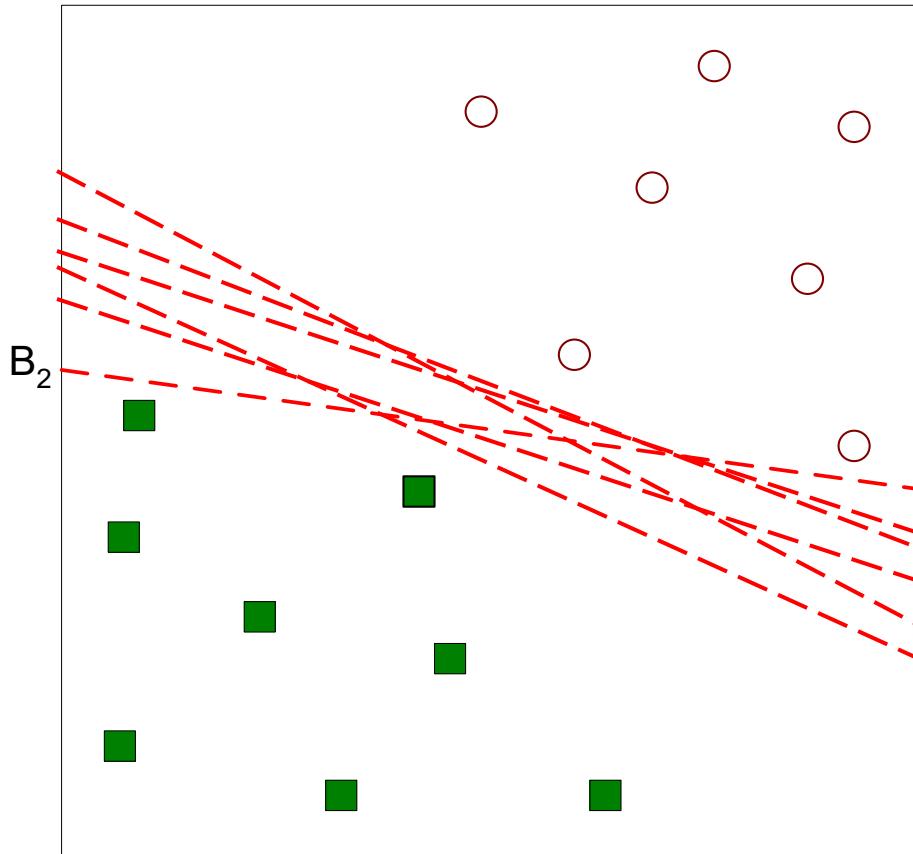
- One Possible Solution

Support Vector Machines



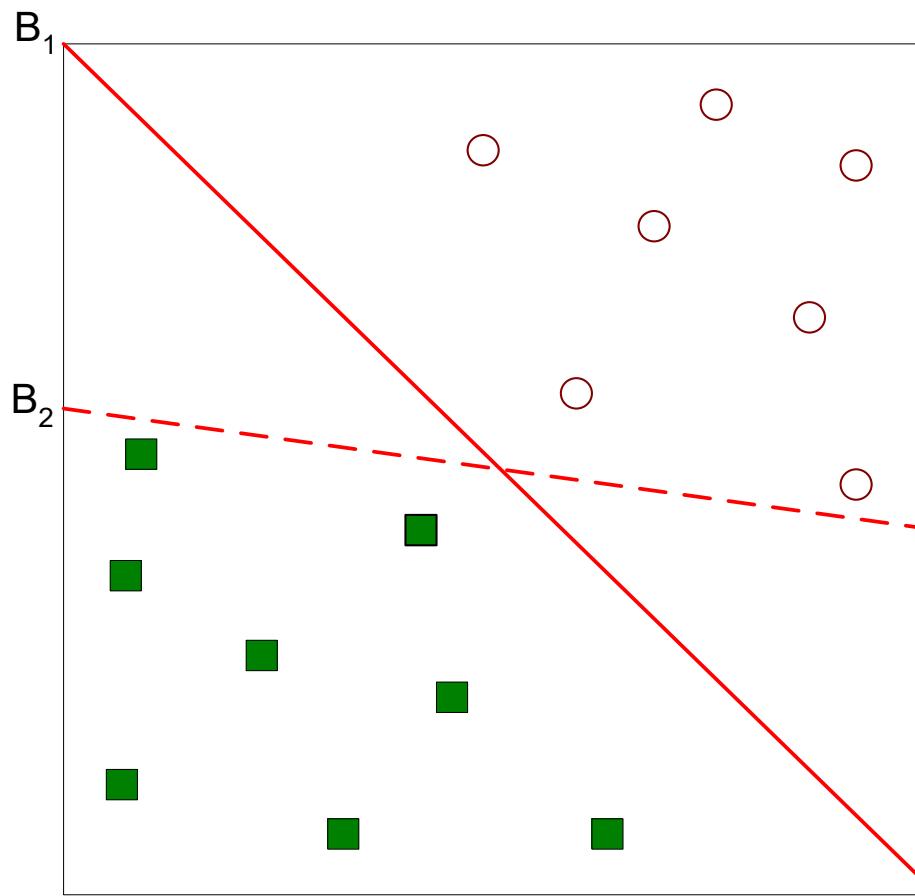
- Another possible solution

Support Vector Machines



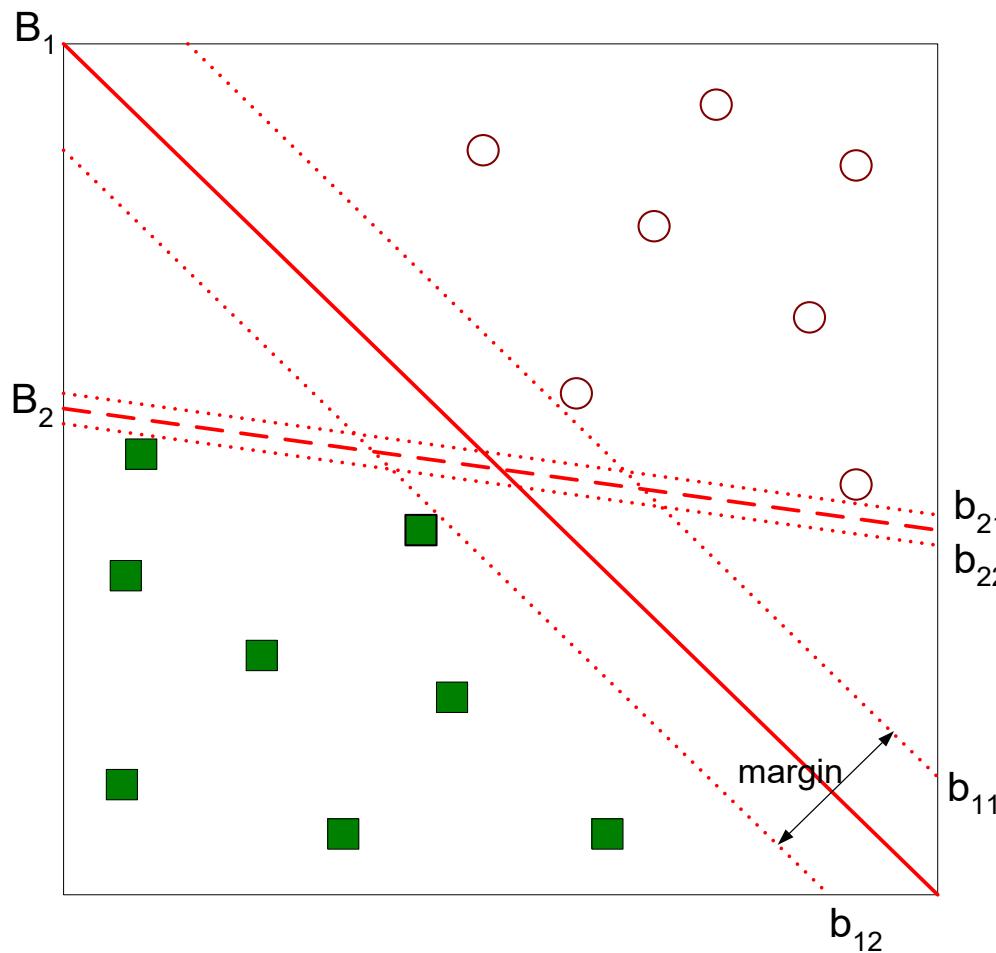
- Other possible solutions

Support Vector Machines



- Which one is better? B_1 or B_2 ?
- How do you define better?

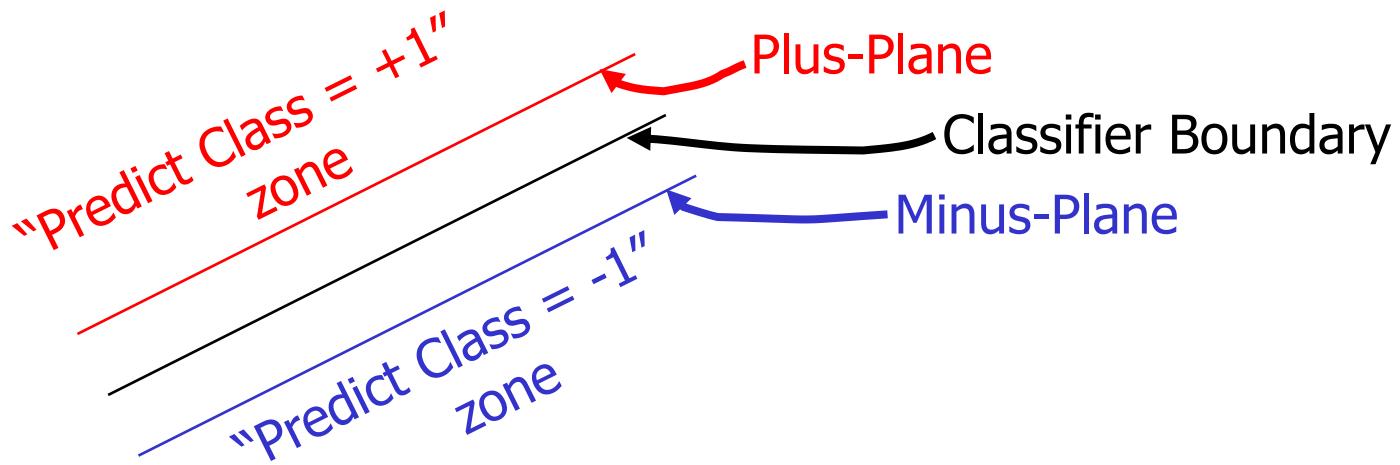
Support Vector Machines



- Find hyperplane **maximizes** the margin => B1 is better than B2

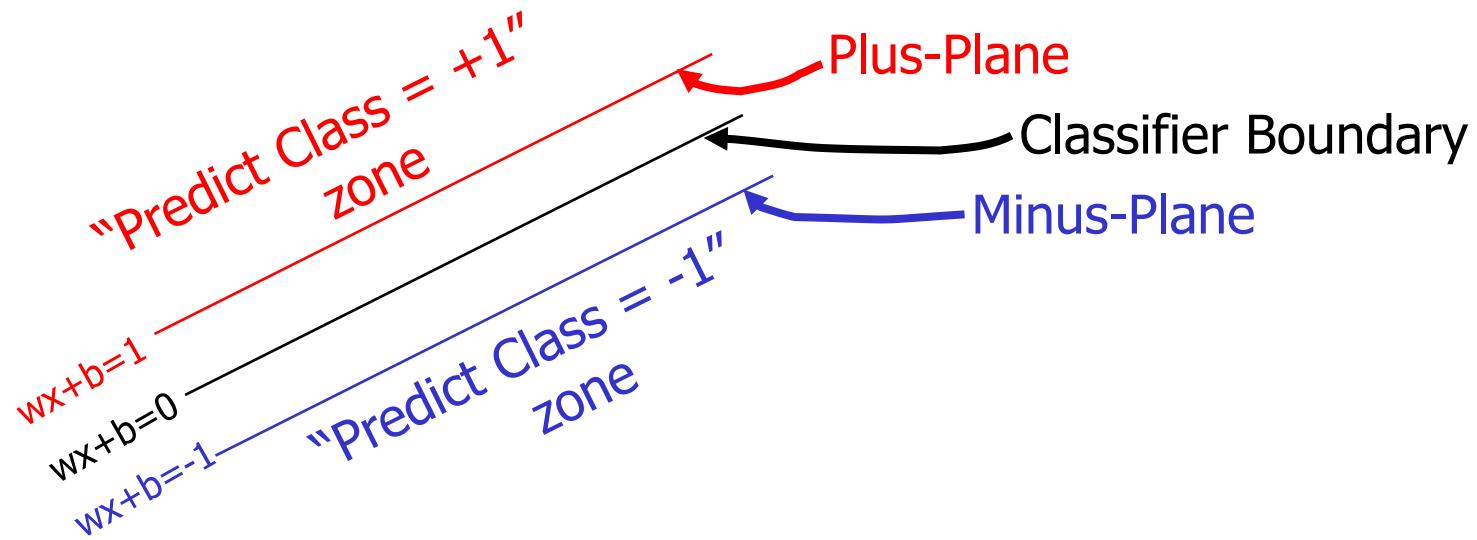
MATHEMATICAL CONCEPT

Specifying a line and margin



- How do we represent this mathematically?
- ...in m input dimensions?

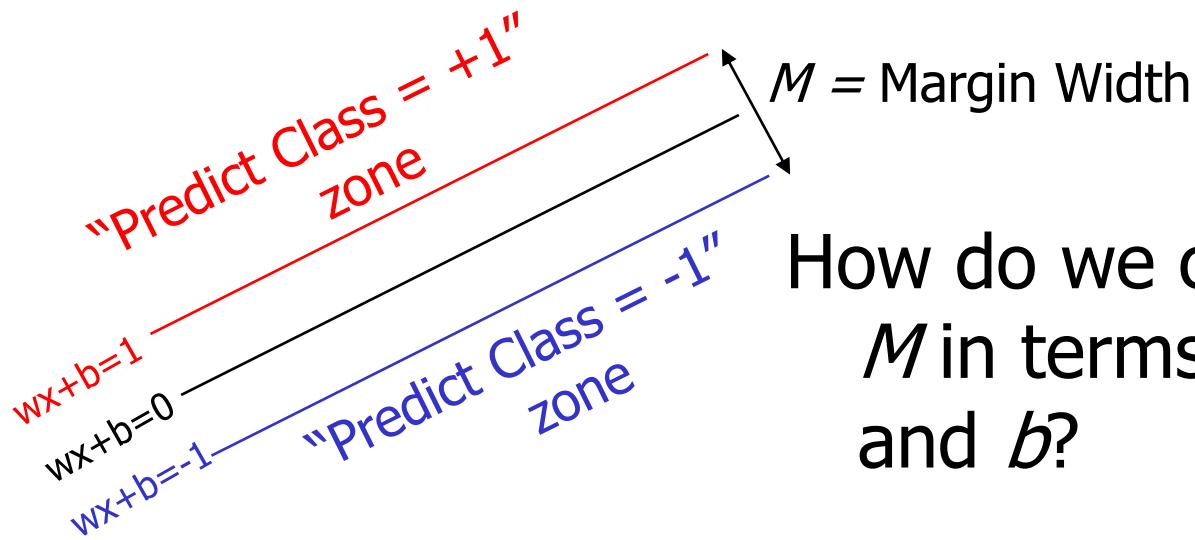
Specifying a line and margin



- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$

Classify as.. +1 if $\mathbf{w} \cdot \mathbf{x} + b \geq 1$
 -1 if $\mathbf{w} \cdot \mathbf{x} + b \leq -1$
 Universe if $-1 < \mathbf{w} \cdot \mathbf{x} + b < 1$
 explodes

Computing the margin width

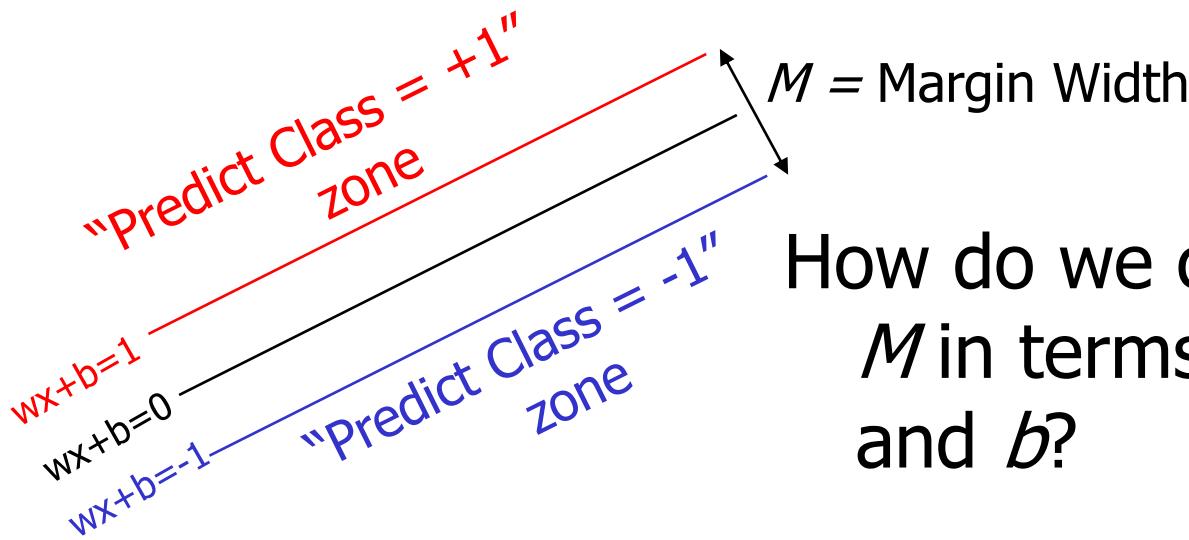


How do we compute M in terms of \mathbf{w} and b ?

- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$

Claim: The vector \mathbf{w} is perpendicular (tegak lurus) to the Plus Plane. Why?

Computing the margin width



How do we compute
 M in terms of \mathbf{w}
and b ?

- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$

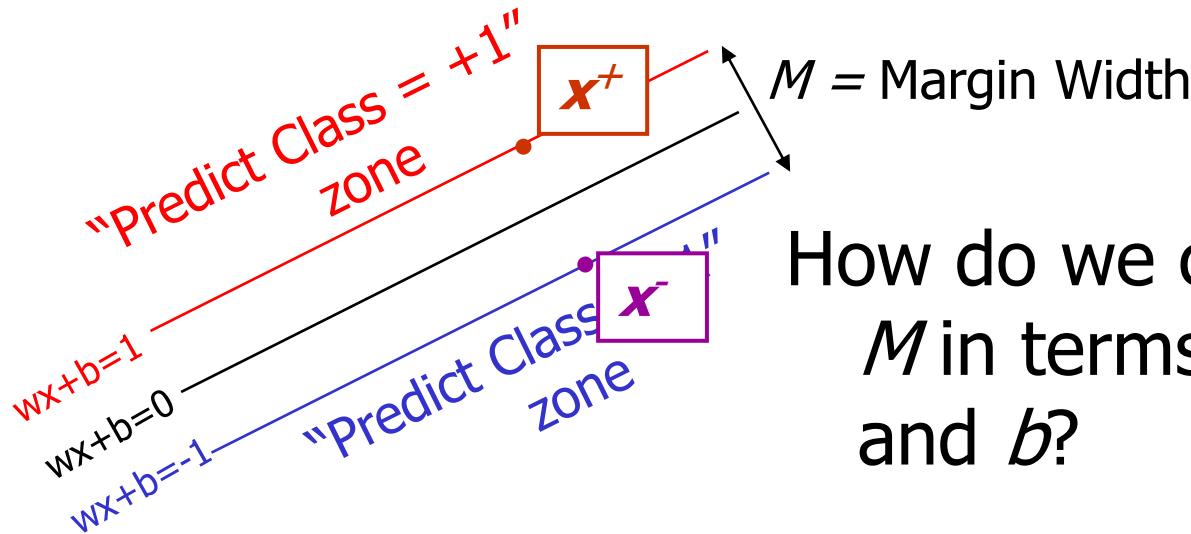
Claim: The vector \mathbf{w} is perpendicular to the Plus Plane. **Why?**

Let \mathbf{u} and \mathbf{v} be two vectors on the
Plus Plane. What is $\mathbf{w} \cdot (\mathbf{u} - \mathbf{v})$?

And so of course the vector \mathbf{w} is also
perpendicular to the Minus Plane

$\mathbf{w} \cdot (\mathbf{u} - \mathbf{v}) = 0 \rightarrow \text{orthogonal}$

Computing the margin width

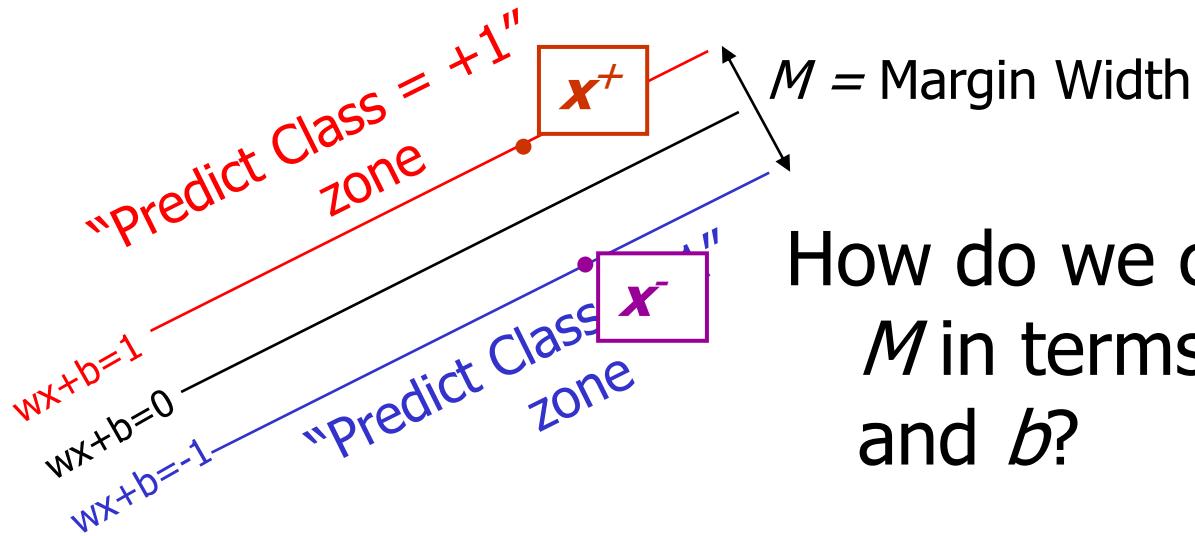


How do we compute
 M in terms of \mathbf{w}
and b ?

- Plus-plane $= \{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane $= \{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$
- The vector \mathbf{w} is perpendicular to the Plus Plane
- Let \mathbf{x}^- be any point on the minus plane
- Let \mathbf{x}^+ be the closest plus-plane-point to \mathbf{x}^- .

Any location in
 \mathbb{R}^m : not
necessarily a
datapoint

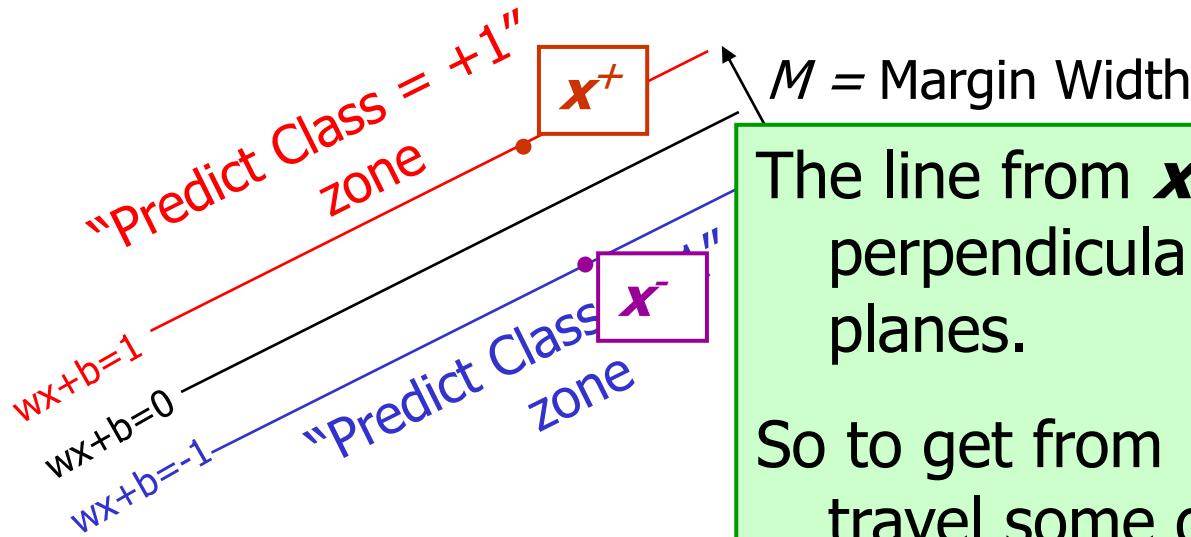
Computing the margin width



How do we compute
 M in terms of \mathbf{w}
and b ?

- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$
- The vector \mathbf{w} is perpendicular to the Plus Plane
- Let \mathbf{x}^- be any point on the minus plane
- Let \mathbf{x}^+ be the closest plus-plane-point to \mathbf{x}^- .
- **Claim:** $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$ for some value of λ . **Why?**

Computing the margin width

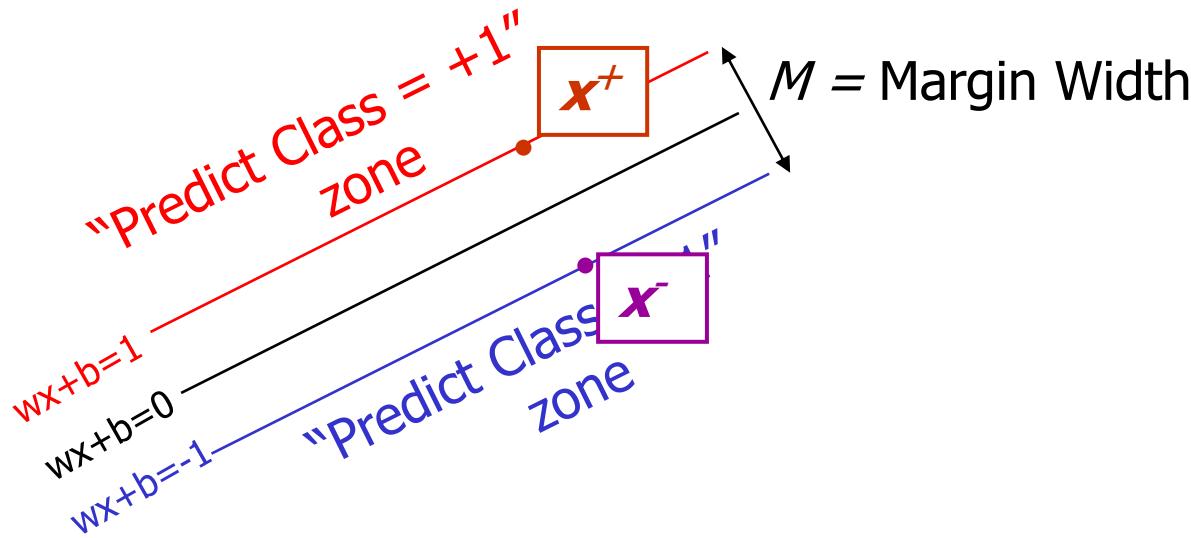


The line from x^- to x^+ is perpendicular to the planes.

So to get from x^- to x^+ travel some distance in direction w .

- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$
- The vector \mathbf{w} is perpendicular to the Plus Plane
- Let \mathbf{x}^- be any point on the minus plane
- Let \mathbf{x}^+ be the closest plus-plane-point to \mathbf{x}^- .
- **Claim:** $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$ for some value of λ . **Why?**

Computing the margin width

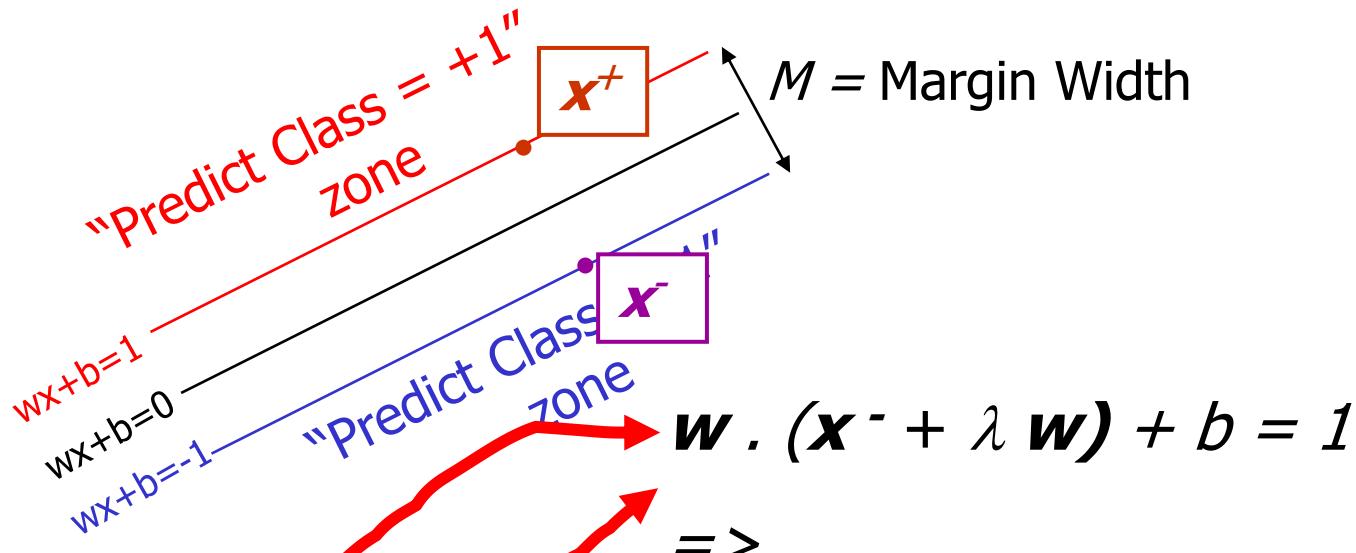


What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
- $|\mathbf{x}^+ - \mathbf{x}^-| = M$

It's now easy to get M
in terms of \mathbf{w} and b

Computing the margin width



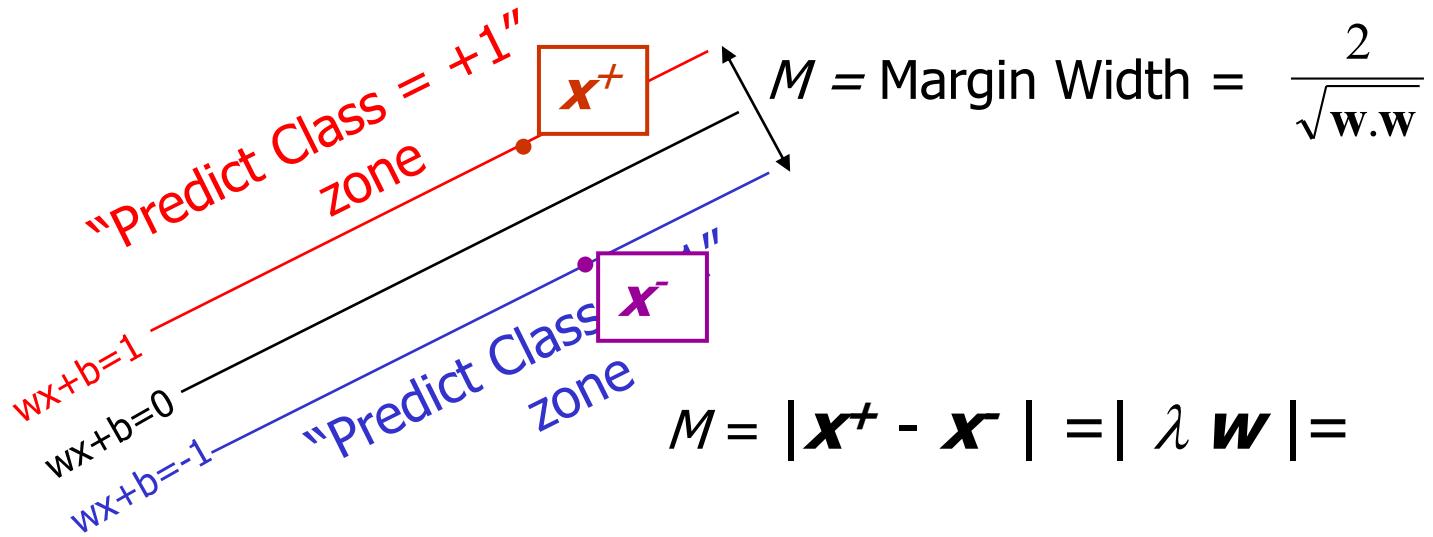
What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
 - $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
 - $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
 - $|\mathbf{x}^+ - \mathbf{x}^-| = M$
- $$\Rightarrow \mathbf{w} \cdot \mathbf{x}^- + b + \lambda \mathbf{w} \cdot \mathbf{w} = 1$$
- $$\Rightarrow -1 + \lambda \mathbf{w} \cdot \mathbf{w} = 1$$
- $$\Rightarrow$$

$$\lambda = \frac{2}{\mathbf{w} \cdot \mathbf{w}}$$

It's now easy to get M in terms of \mathbf{w} and b

Computing the margin width



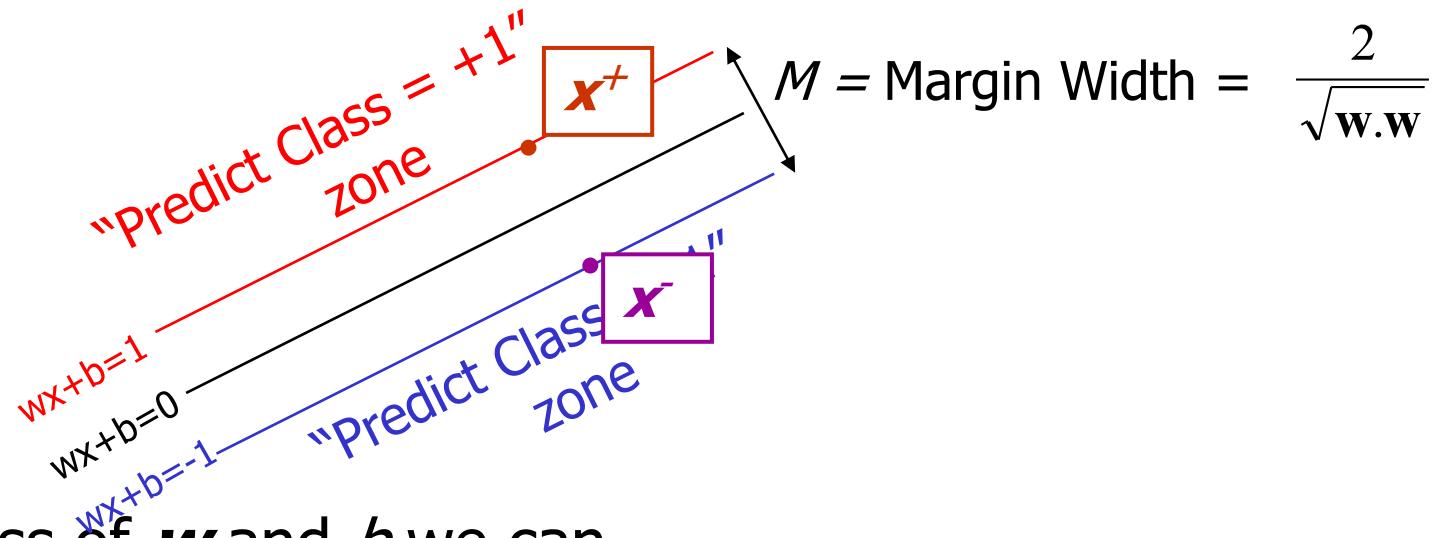
What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$
- $|\mathbf{x}^+ - \mathbf{x}^-| = M$
- $\lambda = \frac{2}{\mathbf{w} \cdot \mathbf{w}}$

$$= \lambda |\mathbf{w}| = \lambda \sqrt{\mathbf{w} \cdot \mathbf{w}}$$

$$= \frac{2\sqrt{\mathbf{w} \cdot \mathbf{w}}}{\mathbf{w} \cdot \mathbf{w}} = \frac{2}{\sqrt{\mathbf{w} \cdot \mathbf{w}}}$$

Learning the Maximum Margin Classifier



Given a guess of \mathbf{w} and b we can

- Compute whether all data points in the correct half-planes
- Compute the width of the margin

So now we just need to write a program to search the space of \mathbf{w} 's and b 's to find the widest margin that matches all the datapoints. *How?*

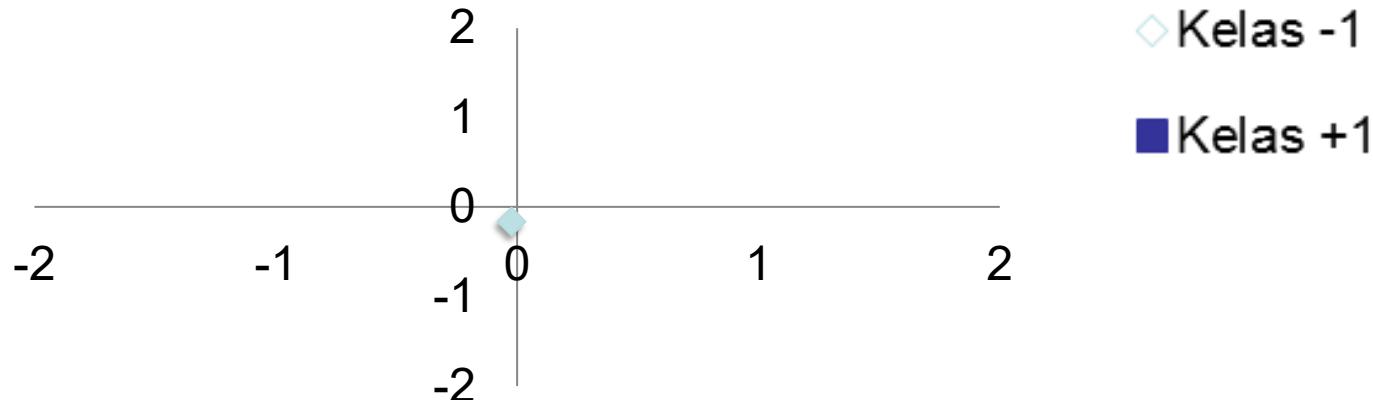
Quadratic Programming

Contoh Studi Kasus

- Contoh SVM Linier pada dataset berikut :
Tentukan Hyperplanenya !

x_1	x_2	Kelas (y)	Support Vector (SV)
1	1	1	1
1	-1	-1	1
-1	1	-1	1
-1	-1	-1	0

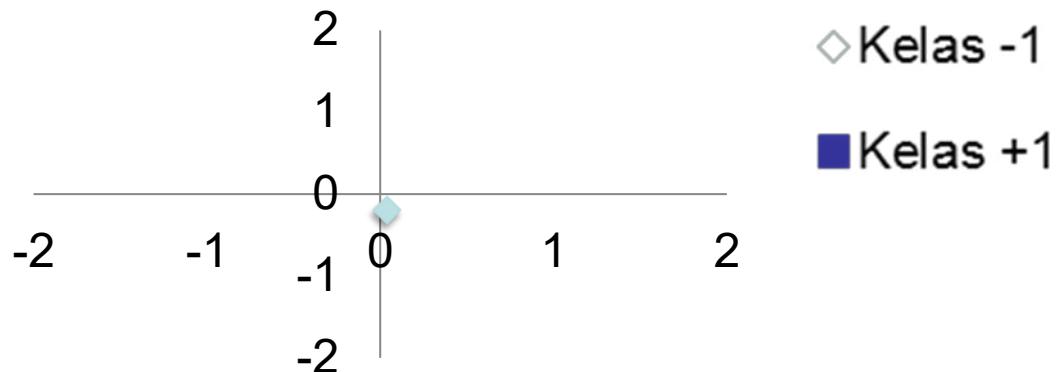
- Bentuk Visualisasi data :



Contoh Studi Kasus 1 (Cont.)

- Contoh SVM Linier :

x_1	x_2	Kelas (y)
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1



- Formulasi yang digunakan adalah sebagai berikut :
 - Meminimalkan nilai :

$$\frac{1}{2} \|w\|^2 = \frac{1}{2} (w_1^2 + w_2^2)$$

- Syarat :

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, 3, \dots, N$$

Contoh Studi Kasus 1 (Cont.)

- Karena ada dua fitur (x_1 dan x_2), maka w juga akan memiliki 2 fitur (w_1 dan w_2).
- Formulasi yang digunakan adalah sebagai berikut :

- Meminimalkan nilai margin :

$$\frac{1}{2} \|w\|^2 = \frac{1}{2} (w_1^2 + w_2^2)$$

- Syarat :

$$y_i(w \cdot x_i + b) \geq 1, \quad i = 1, 2, 3, \dots, N$$

$$y_i(w_1 \cdot x_1 + w_2 \cdot x_2 + b) \geq 1$$

x_1	x_2	Kelas (y)
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

Sehingga didapatkan beberapa persamaan berikut :

1. $(w_1 + w_2 + b) \geq 1$, untuk $y_1 = 1, x_1 = 1, x_2 = 1$
2. $(-w_1 + w_2 - b) \geq 1$, untuk $y_2 = -1, x_1 = 1, x_2 = -1$
3. $(w_1 - w_2 - b) \geq 1$, untuk $y_3 = -1, x_1 = -1, x_2 = 1$
4. $(w_1 + w_2 - b) \geq 1$, untuk $y_4 = -1, x_1 = -1, x_2 = -1$

Contoh Studi Kasus 1 (Cont.)

Didapatkan beberapa persamaan berikut :

1. $(w_1 + w_2 + b) \geq 1$
2. $(-w_1 + w_2 - b) \geq 1$
3. $(w_1 - w_2 - b) \geq 1$
4. $(w_1 + w_2 - b) \geq 1$

- Menjumlahkan persamaan (1) dan (2) :

$$\begin{aligned}(w_1 + w_2 + b) &\geq 1 \\ (-w_1 + w_2 - b) &\geq 1 \\ \hline &+ \\ 2w_2 &= 2\end{aligned}$$

Maka $w_2 = 1$

- Menjumlahkan persamaan (1) dan (3) :

$$\begin{aligned}(w_1 + w_2 + b) &\geq 1 \\ (w_1 - w_2 - b) &\geq 1 \\ \hline &+ \\ 2w_1 &= 2 \\ \text{Maka } w_1 &= 1\end{aligned}$$

- Menjumlahkan persamaan (2) dan (3) :

$$\begin{aligned}(-w_1 + w_2 - b) &\geq 1 \\ (w_1 - w_2 - b) &\geq 1 \\ \hline &+ \\ -2b &= 2 \\ \text{Maka } b &= -1\end{aligned}$$

Sehingga didapatkan persamaan hyperplane :

$$w_1x_1 + w_2x_2 + b = 0$$

$$x_1 + x_2 - 1 = 0$$

$$x_2 = 1 - x_1$$

Contoh Studi Kasus 1 (Cont.)

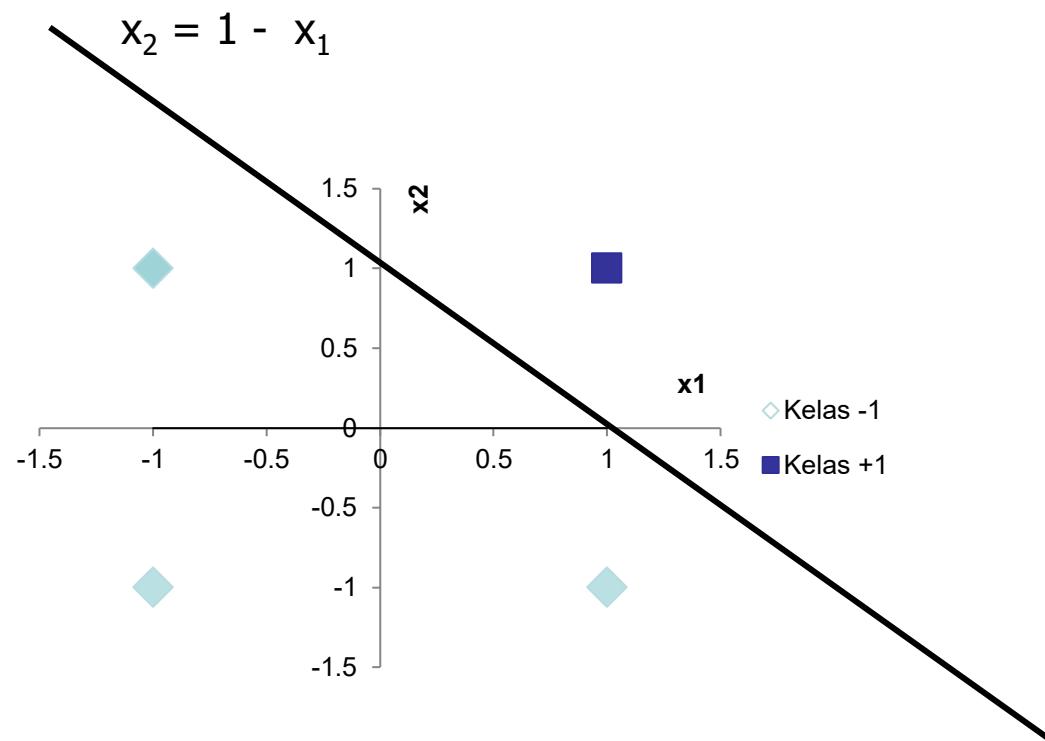
Visualisasi garis hyperplane (sebagai fungsi klasifikasi) :

$$w_1x_1 + w_2x_2 + b = 0$$

$$x_1 + x_2 - 1 = 0$$

$$x_2 = 1 - x_1$$

x_1	$x_2 = 1 - x_1$
-2	3
-1	2
0	1
1	0
2	-1



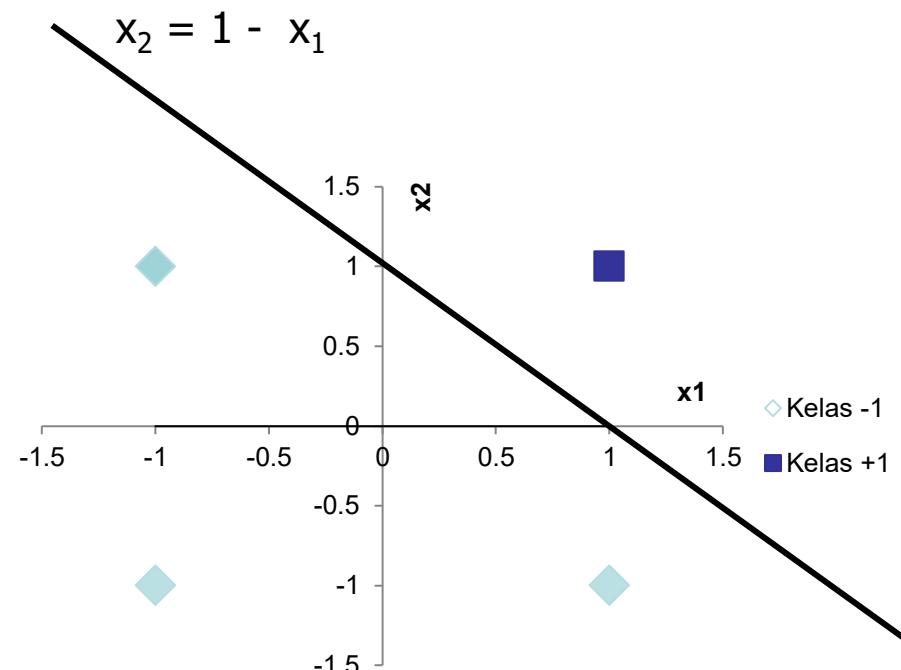
Contoh Studi Kasus 1 (Cont.)

Misalkan diketahui data uji/ data testing berikut :

$$\text{Diketahui : } f(x) = x_1 + x_2 - 1$$

$$\text{Kelas} = \text{sign}(f(x))$$

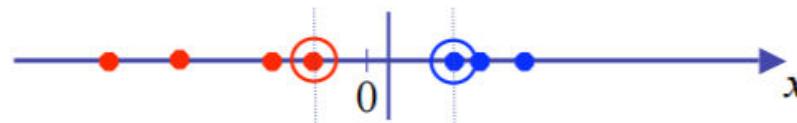
No	Data Uji		Hasil Klasifikasi $\text{Kelas} = \text{sign}(x_1 + x_2 - 1)$
	x_1	x_2	
1	1	5	$\text{sign}(1 + 5 - 1) = +1$
2	-1	4	$\text{sign}(-1 + 4 - 1) = +1$
3	0	7	$\text{sign}(0 + 7 - 1) = +1$
4	-9	0	$\text{sign}(-9 + 0 - 1) = -1$
5	2	-2	$\text{sign}(2 - 2 - 1) = -1$



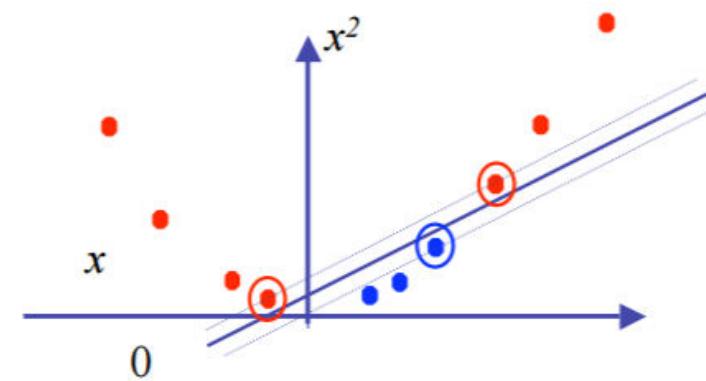
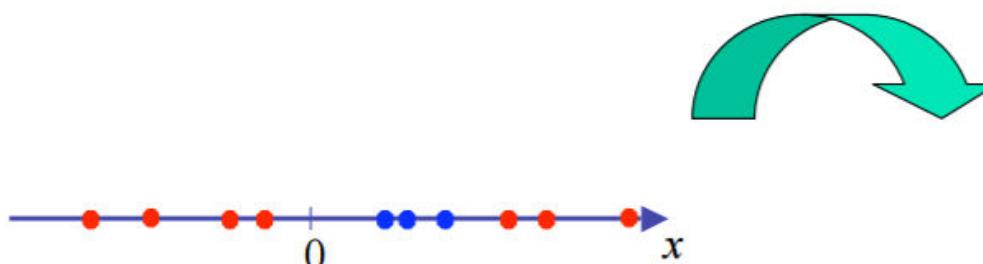
NON LINEAR SVM

Non Linear SVMs

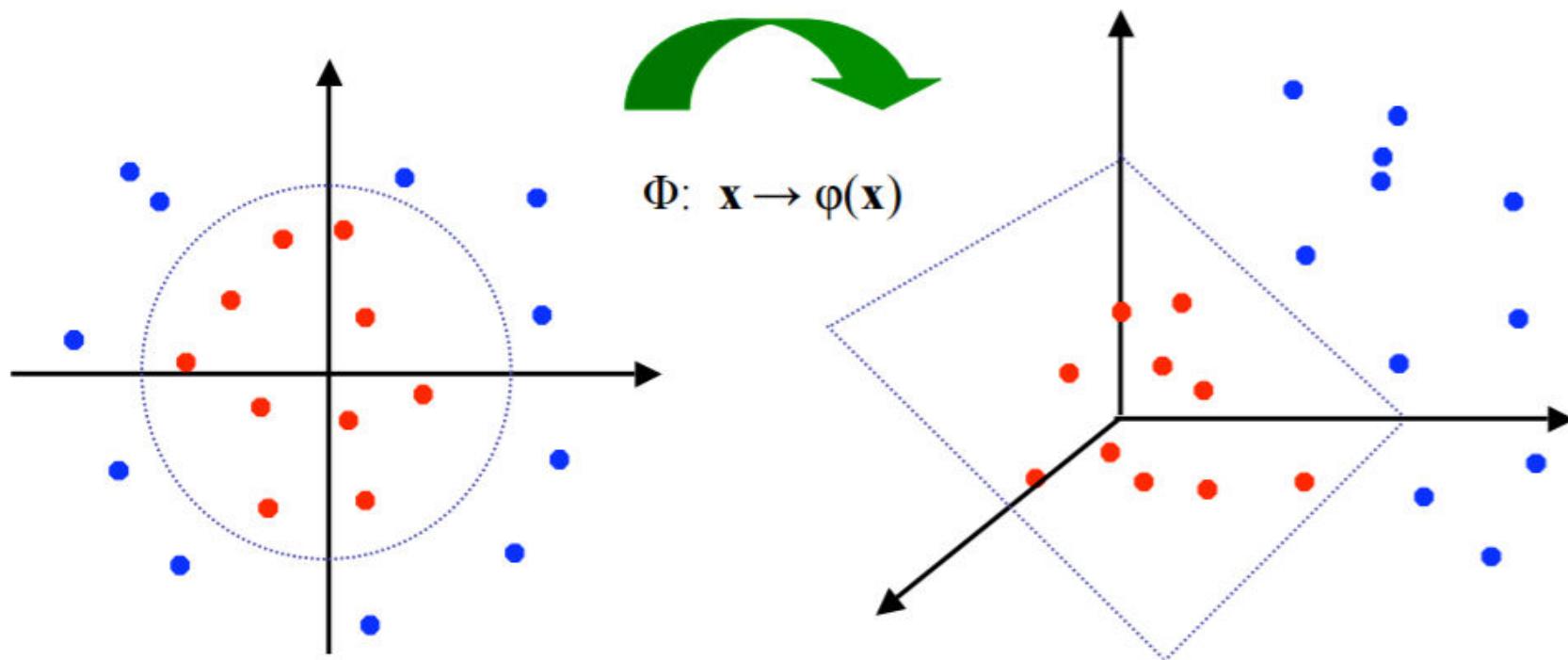
Linearly separable
data:



Map data to a
higher-dimensional space:



Non Linear SVMs : Feature Space



The original input space can always be mapped to some higher-dimensional feature space where the training set is separable:

Model SVM (Cont.)

- Beberapa Macam Fungsi Kernel Support Vector Machine (SVM) :

No	Nama Kernel	Definisi Fungsi
1	Linier	$K(x,y) = x \cdot y$
2	Polinomial of degree d	$K(x,y) = (x^T \cdot y)^d$
3	Polinomial of degree up to d	$K(x,y) = (x^T \cdot y + c)^d$
4	Gaussian RBF	$K(x,y) = \exp\left(\frac{-\ x-y\ ^2}{2\sigma^2}\right)$
5	Sigmoid (Tangen Hiperbolik)	$K(x,y) = \tanh(\sigma(x \cdot y) + c)$
6	Invers Multi Kuadratik	$K(x,y) = \frac{1}{\sqrt{\ x-y\ ^2 + c^2}}$
7	Additive	$K(x,y) = \sum_{i=1}^n K_i(x_i, y_i)$

- Kernel Linier digunakan ketika data yang akan diklasifikasi dapat terpisah dengan sebuah garis/hyperplane.
- Kernel non-Linier digunakan ketika data hanya dapat dipisahkan dengan garis lengkung atau sebuah bidang pada ruang dimensi tinggi (Kernel Trik, No.2 sampai 6).