

## Modul : Supervised Learning

## k-Nearest Neighbor

KK IF - Teknik Informatika- STEI ITB

Inteligensi Buatan  
(Artificial Intelligence)

EDUNEX ITB

## k-Nearest Neighbor

Supervised Learning

Instance-Based Classifier  
(Store all training data)

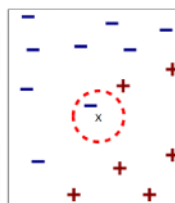
Lazy learner

No hypothesis

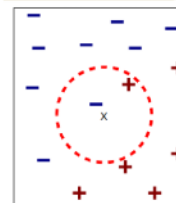
Unseen data prediction: Find class from  
similar stored datapendekatan machine learning  
menggunakan data berlabellazy learning,  
tidak membangun model eksplisit,  
melainkan menyimpan instance  
untuk melihat seberapa mirip  
data baru dengan contoh  
yang adahanya membandingkan,  
tidak ada generalisasitidak membangun model saat training  
model dibuat saat data yang hendak diprediksi tiba

EDUNEX ITB

## Classification (Predict unseen data)

Measures 'distance' of query (unseen data)  
to all instance (in training data)Symbolic attribute: 1 (different  
value), 0 (same value)Numeric attribute: Euclidean  
DistanceFind k 'most similar' instances  
(k nearest neighbor)Find the majority class from k  
nearest neighborClass/ Label Prediction: Majority Class of k  
nearest neighbor

(a) 1-nearest neighbor



(b) 2-nearest neighbor

biasanya paling  
bagus ketika k = 1  
kalau ada data training  
tapi sensitif noise (overfit)

distance = ukuran kemiripan

↳ bisa pakai euclidean / manhattan distance  
untuk atribut numerik, kalau untuk  
kategorikal bisa sama = 0, beda = 1

EDUNEX ITB

## Example: Play Tennis Dataset

outlook	temp.	humidity	windy	play	outlook	temp.	humidity	windy	play
sunny	hot	high	false	no	sunny	mild	high	false	no
sunny	hot	high	true	no	sunny	cool	normal	false	yes
overcast	hot	high	false	yes	rainy	mild	normal	false	yes
rainy	mild	high	false	yes	sunny	mild	normal	true	yes
rainy	cool	normal	false	yes	overcast	mild	high	true	yes
rainy	cool	normal	true	no	overcast	hot	normal	false	yes
overcast	cool	normal	true	yes	rainy	mild	high	true	no



$gamma = 0$ ,  $beda = 1$

## Classify New Instance: <Sunny, Cool, High, True>

outlook	temp.	humidity	windy	play	Distance	
sunny	hot	high	false	no	2	$k = 1 \rightarrow D-2 \rightarrow \text{Play: No}$
sunny	hot	high	true	no	1	
overcast	hot	high	false	yes	3	$k = 2 \rightarrow D-1, D-2, \rightarrow \text{Play: No}$
rainy	mild	high	false	yes	3	
rainy	cool	normal	false	yes	3	
rainy	cool	normal	true	no	2	$k = 3 \rightarrow D-1, D-2, D-6 \rightarrow \text{Play: No}$
overcast	cool	normal	true	yes	2	
sunny	mild	high	false	no	2	
sunny	cool	normal	false	yes	2	
rainy	mild	normal	false	yes	4	
sunny	mild	normal	true	yes	2	
overcast	mild	high	true	yes	2	
overcast	hot	normal	false	yes	4	
rainy	mild	high	true	no	2	



## Notes on k-Nearest Neighbor

### Advantages

Approximation can be less complex for complex target function

*Kadang fungsi target terlalu kompleks dan sulit ditulis dalam rumus*

*→ karena k-NN tidak membuat rumus, meskipun target function rumit, pendekatan jadi lebih sederhana*

### Disadvantages

*Untuk setiap data baru, k-NN harus menghitung jaraknya dengan semua data tsb → cost tinggi*

Cost of classifying new instance high

Consider all features → target function depends only on a few features

*k-NN memperhitungkan semua fitur, padahal bisa saja hanya beberapa fitur yang relevan → fitur tidak penting bisa mengganggu perhitungan jarak*



id	hobi	umur	pendidikan	kelas
1	Game	remaja	sma	1
2	Game	dewasa	s1	2
3	Game	dewasa	diploma	3
4	Baca	dewasa	s1	3
5	Olahraga	dewasa-muda	s1	2
6	Olahraga	dewasa-muda	diploma	3
7	Game	dewasa-muda	sma	1
8	Olahraga	dewasa-muda	sma	1
9	Baca	dewasa-muda	sma	1
10	Game	dewasa-muda	s1	3
11	Baca	Remaja	diploma	1
12	Game	remaja	diploma	2
13	game	dewasa	sma	3

Id	Jarak thd data baru
1	1+1+1=3
2	1+0+0=1 ✓
3	1+0+1=2 ✓
4	1+0+0=1 ✓
5	0+1+0=1 ✓
6	0+1+1=2 ✓
7	1+1+1=3
8	0+1+1=2
9	1+1+1=3
10	1+1+0=2
11	1+1+1=3
12	1+1+1=3
13	1+0+1=2

1. Hitung Jarak setiap instance data latih utk data uji berikut:  
hobi = olahraga; umur = dewasa; pendidikan = s1
2. Untuk k=5, maka kelas dari data uji di atas adalah: 3

EDUNEX ITB



kategorikal

↓ one-hot encoding

numerikal

supaya fitur punya tipe yang sama

## Distance Measurement on Numeric Attributes

$$D = \left( \sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}$$

Minkowski distance

jika,  
p:1 → Manhattan distance  
p:2 → Euclidean distance  
p:n → Chebyshev distance

$$D_m = \sum_{i=1}^n |p_i - q_i|$$

Manhattan distance

pergerakan GRID (vertikal / horizontal)

$$D_e = \left( \sum_{i=1}^n (p_i - q_i)^2 \right)^{1/2}$$

Euclidean distance

pergerakan garis lurus

EDUNEX ITB



Brightness	Saturation	Class
20	35	?
40	20	Red
50	50	Blue
60	90	Blue
10	25	Red
70	70	Blue
60	10	Red
25	80	Blue

$$\begin{aligned} d1 &= \sqrt{(20 - 40)^2 + (35 - 20)^2} \\ &= \sqrt{400 + 225} \\ &= \sqrt{625} \\ &= 25 \end{aligned}$$

$$\begin{aligned} d2 &= \sqrt{(20 - 50)^2 + (35 - 50)^2} \\ &= \sqrt{900 + 225} \\ &= \sqrt{1125} \\ &= 33.54 \end{aligned}$$

$$D_e = \left( \sum_{i=1}^n (p_i - q_i)^2 \right)^{1/2}$$

Euclidean distance

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

- $X_2$  = New entry's brightness (20).
- $X_1$  = Existing entry's brightness.
- $Y_2$  = New entry's saturation (35).
- $Y_1$  = Existing entry's saturation.

EDUNEX ITB



Brightness	Saturation	Class
40	20	Red
50	50	Blue
60	90	Blue
10	25	Red
70	70	Blue
60	10	Red
25	80	Blue

Brightness	Saturation	Class	Distance
40	20	Red	25
50	50	Blue	33.54
60	90	Blue	68.01
10	25	Red	10
70	70	Blue	61.03
60	10	Red	47.17
25	80	Blue	45

Brightness	Saturation	Class
20	35	?

Kalau 3 tetangga terdekat  
 RED  
 bisa out default value  
 Kalau ada majority class sama

07

THANK YOU

EDUNEX ITB

KNN2

01

Modul : Supervised Learning

Prediction Measurement

Nur ULFA Maulidevi

KK IF - Teknik Informatika- STEI ITB

Inteligensi Buatan  
 (Artificial Intelligence)

EDUNEX ITB

## Prediction Measurement

Supervised Learning

"Correct Class"

Prediction based on  
Model/ Hypothesis  
from Learning

		Prediction	
		True	False
Reality	True	<b>Tp</b> True-positive	<b>Fn</b> False-negative
	False	<b>Fp</b> False-positive	<b>Tn</b> True-negative

Handwritten notes:

- benar (True)
- salah (False)
- prediksi positif (True-positive)
- prediksi negatif (True-negative)
- salah prediksi (False-negative)
- salah prediksi (False-positive)

EDUNEX ITB



## Example

Instance	Correct Class	Prediction	
1	+	+	TP
2	-	-	TN
3	-	+	FP
4	+	-	FN
5	-	+	FP
6	+	-	FN
7	+	+	TP
8	-	-	TN
9	-	-	TN
10	+	+	TP

EDUNEX ITB



## Accuracy

		Prediction	
		True	False
Reality	True	<b>Tp</b> True-positive	<b>Fn</b> False-negative
	False	<b>Fp</b> False-positive	<b>Tn</b> True-negative

bagus digunakan ketika  
distribusi kelas seimbang

Fraction of all correct predictions over all predicted instances

$$Accuracy = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

Handwritten notes:

- prediksi benar (Tp + Tn)
- semua benar (Tp + Fp + Tn + Fn)

EDUNEX ITB



## Precision

		Prediction	
		True	False
Reality	True	<b>Tp</b> True-positive	<b>Fn</b> False-negative
	False	<b>Fp</b> False-positive	<b>Tn</b> True-negative

"dari semua ⊕, berapa yang benar ⊕?"

→ untuk kalau FP bertambah

Fraction of positive predictions that are correct

$$\text{Precision} = \frac{Tp}{Tp + Fp}$$

EDUNEX ITB



## Recall

		Prediction	
		True	False
Reality	True	<b>Tp</b> True-positive	<b>Fn</b> False-negative
	False	<b>Fp</b> False-positive	<b>Tn</b> True-negative

"dari semua yang aslinya ⊕, berapa yang berhasil diprediksi?"

→ kalau FN bertambah

Fraction of positive instances that are correctly predicted (retrieved/caught)

$$\text{Recall} = \frac{Tp}{Tp + Fn}$$

prediksi instance yang aslinya true

EDUNEX ITB



## Exercise: Find accuracy, precision and recall

Instance	Correct Class	Prediction
1	+	+
2	-	-
3	-	+
4	+	-
5	-	+
6	+	-
7	+	+
8	-	-
9	-	-
10	+	+
11	+	-
12	-	-
13	-	+
14	+	-

① accuracy

$$\frac{7}{14} = 50\%$$

② precision

$$\frac{3}{6} = 50\%$$

③ recall

$$\frac{3}{7} = 42,86\%$$

EDUNEX ITB



---

# THANK YOU

EDUNEX ITB

