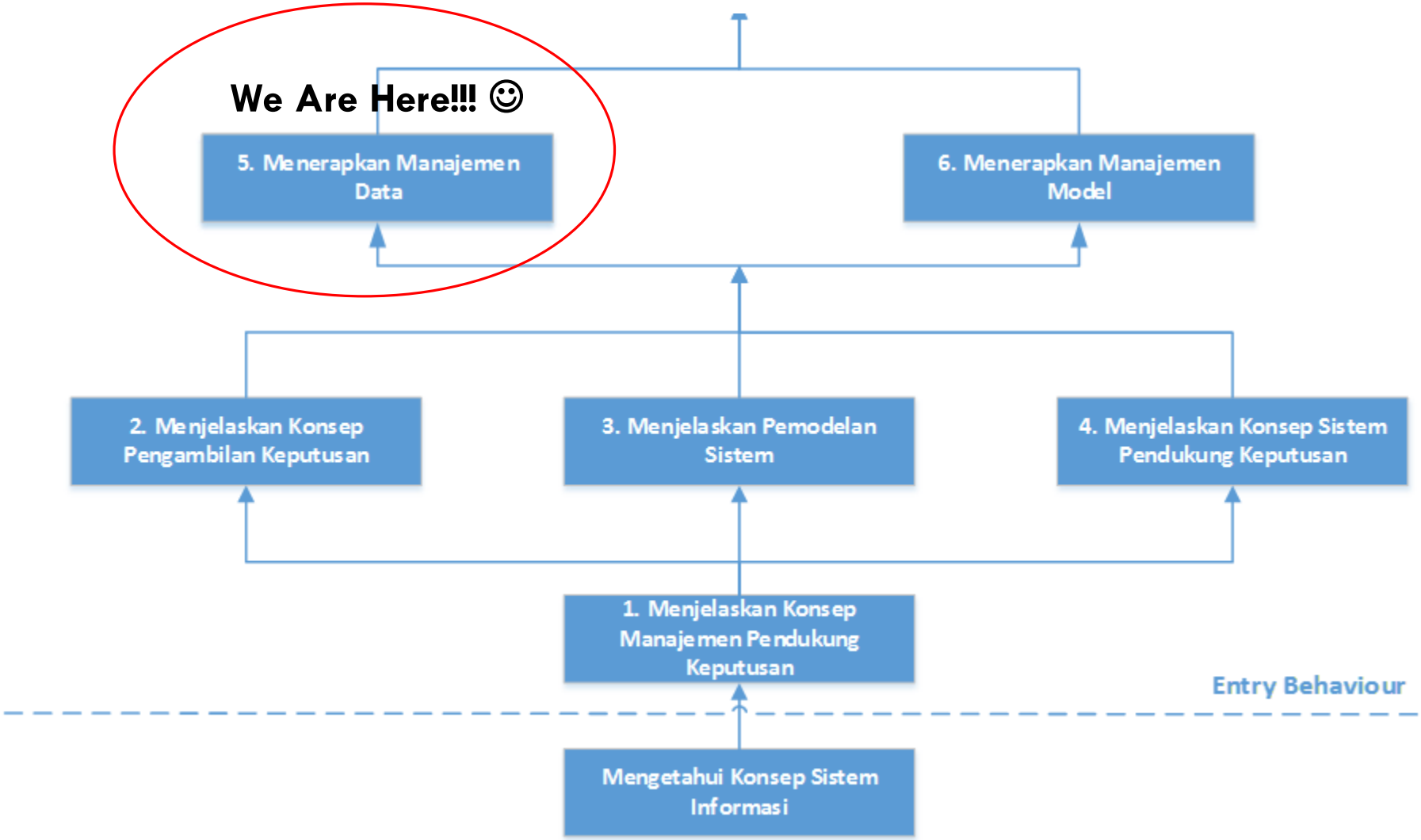
Manajemen Data

DECISION SUPPORT SYSTEM [D10K-5B01]

SUB CAPAIAN PEMBELAJARAN MK



AGENDA

1. Pendahuluan Subsistem Manajemen Data
2. Model Data Mining
3. Klasifikasi
4. Clustering

SUBSISTEM MANAJEMEN DATA

Terdiri database yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak Sistem Manajemen Database (DBMS).

Terdiri dari elemen berikut ini;

- DSS Database (Data dapat dimasukkan langsung kedalam model-model, atau di ekstraksi dari database yang lebih besar (datawarehouse))
- Sistem Manajemen Database (DBMS Relasional)
- Direktori Data (katalog semua data dalam database)
- Query Facility (seleksi dan manipulasi)

DATABASE

Konteks database disini berarti kumpulan data saling terkait yang dikelola untuk memenuhi kebutuhan dan dapat digunakan oleh lebih dari satu orang untuk lebih dari satu aplikasi.

Ekstraksi

Untuk membuat sebuah database DSS atau sebuah data warehouse, maka proses ekstraksi dilakukan untuk meng-capture data dari beberapa sumber.

SISTEM MANAJEMEN DATABASE

Sebuah database yang efektif dan manajemennya dapat mendukung banyak kegiatan manajerial; navigasi umum di antara record-record, mendukung pembuatan dan pemeliharaan sebuah kumpulan hubungan data yang berbeda-beda, dan laporan merupakan hasil yang umum.

Akan tetapi, kekuatan riil dari sebuah DSS terjadi ketika data diintegrasikan dengan model-modelnya.

QUERY FACILITY

Membangun dan menggunakan DSS sering memerlukan akses, manipulasi, dan query data.

Query Facility memasukkan sebuah bahasa query khusus (misal SQL).

Fungsi penting dari sebuah sistem DSS query adalah operasi seleksi dan manipulasi.

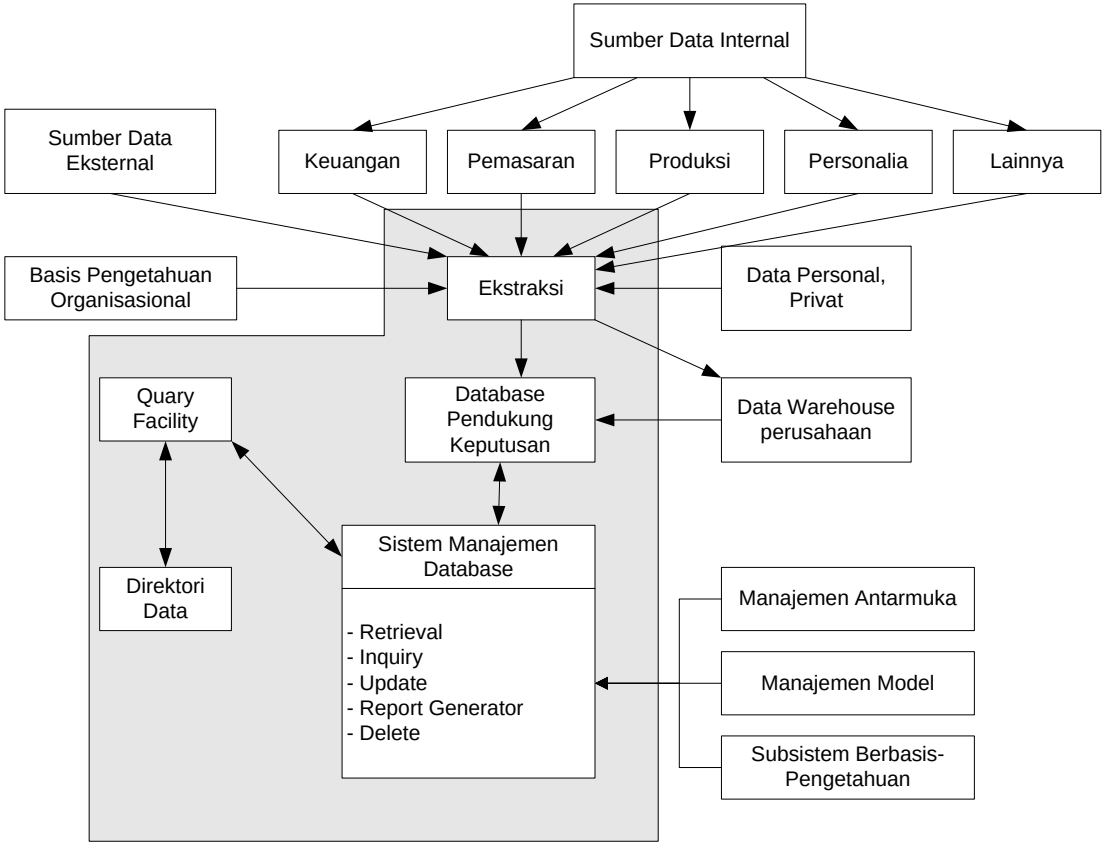
DIREKTORI DATA

Merupakan sebuah katalog dari semua data di dalam database.

Berisi definisi data, dan fungsi utamanya adalah untuk menjawab pertanyaan mengenai ketersediaan item-item data, sumbernya, dan makna eksak dari data.

Mendukung penambahan entri baru, menghapus, dan mendapatkan kembali informasi mengenal objek-objek khusus.

STRUKTUR SUBSISTEM MANAJEMEN DATA



MODEL DATA MINING

DEFINISI

"Mining": proses atau usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar yang telah ada.

DEFINISI

Beberapa faktor dalam pendefinisian data mining:

- Data mining adalah proses otomatis terhadap data yang dikumpulkan di masa lalu
- Objek dari data mining adalah data yang berjumlah besar atau kompleks
- Tujuan dari data mining adalah menemukan hubungan-hubungan atau pola-pola yang mungkin memberikan indikasi yang bermanfaat.

KATEGORI DALAM DATA MINING

Classification

Clustering

Statistical Learning

Association Analysis

Link Mining

Bagging and Boosting

Sequential Patterns

Integrated Mining

Rough Sets

Graph Mining

Classification

CLASSIFICATION

Klasifikasi adalah suatu proses pengelompokan data dengan didasarkan pada ciri-ciri tertentu ke dalam kelas-kelas yang telah ditentukan pula.

Dua metode yang cukup dikenal dalam klasifikasi, antara lain:

- Naive Bayes
- K Nearest Neighbours (K-NN)

Naïve Bayesian Classification

BAYESIAN CLASSIFICATION:

A statistical classifier: performs probabilistic prediction, i.e., predicts class membership probabilities

Foundation: Based on Bayes' Theorem.

Performance: A simple Bayesian classifier, naïve Bayesian classifier, has comparable performance with decision tree and selected neural network classifiers

Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data

Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

BAYES' THEOREM: BASICS

Total probability Theorem: $$P(B) = \sum_{i=1}^{M} P(B|A_i)P(A_i)$$

Bayes' Theorem: $$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Let **X** be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine P(H|**X**), (i.e., *posteriori probability*): the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*): the initial probability
  - E.g., **X** will buy computer, regardless of age, income, …
- P(**X**): probability that sample data is observed
- P(**X**|H) (likelihood): the probability of observing the sample **X**, given that the hypothesis holds
  - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

PREDICTION BASED ON BAYES' THEOREM

Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes' theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

Informally, this can be viewed as

posteriori = likelihood x prior/evidence

Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes

Practical difficulty: It requires initial knowledge of many probabilities, involving significant computational cost

CLASSIFICATION IS TO DERIVE THE MAXIMUM POSTERIORI

Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$

Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.

Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i | \mathbf{X})$

This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

Since P(X) is constant for all classes, only

needs to be maximized

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

NAÏVE BAYES CLASSIFIER

A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times ... \times P(x_n|C_i)$$

This greatly reduces the computation cost: Only counts the class distribution

If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i,D}|$ (# of tuples of C_i in D)

If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

and $P(x_k|C_i)$ is

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

NAÏVE BAYES CLASSIFIER: TRAINING DATASET

Class:

C_1:buys_computer = 'yes'

C_2:buys_computer = 'no'

Data to be classified:

X = (age <=30,
income = medium,
student = yes,
credit_rating = fair)

X → buy computer?

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	No	excellent	yes
31…40	high	Yes	fair	yes
>40	medium	No	excellent	no

NAÏVE BAYES CLASSIFIER: AN EXAMPLE

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643

P(buys_computer = "no") = 5/14= 0.357

Compute $P(X \mid C_i)$ for each class

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667

P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X \mid C_i)$: P(X | buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X | buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X \mid C_i)*P(C_i)$: P(X | buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P(X | buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

TAHAPAN ALGORITMA NAÏVE BAYES

1. Baca Data Training
2. Hitung jumlah class
3. Hitung jumlah kasus yang sama dengan class yang sama
4. Kalikan semua nilai hasil sesuai dengan data X yang dicari class-nya

1. BACA DATA TRAINING

Outlook	Temperature	Humidity	Windy	Play
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

TEOREMA BAYES

$$P(H \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid H)P(H)}{P(\mathbf{X})} = P(\mathbf{X} \mid H) \times P(H) / P(\mathbf{X})$$

- ***X*** → Data dengan class yang belum diketahui
- ***H*** → Hipotesis data *X* yang merupakan suatu class yang lebih spesifik
- ***P (H|X)*** → Probabilitas hipotesis *H* berdasarkan kondisi *X* (*posteriori probability*)
- ***P (H)*** → Probabilitas hipotesis *H* (*prior probability*)
- ***P (X|H)*** → Probabilitas *X* berdasarkan kondisi pada hipotesis *H*
- ***P (X)*** → Probabilitas *X*

2. HITUNG JUMLAH CLASS/LABEL

Terdapat 2 class dari data training tersebut, yaitu:

- C1 (Class 1) →Play = yes → 9 record
- C2 (Class 2) →Play = no → 5 record
- Total = 14 record

Maka:

- P (C1) = 9/14 = 0.642857143
- P (C2) = 5/14 = 0.357142857

Pertanyaan:

- Data *X = (outlook=rainy, temperature=cool, humidity=high, windy=true)*
- Main golf atau tidak?

3. HITUNG JUMLAH KASUS YANG SAMA DENGAN CLASS YANG SAMA

Untuk *P(Ci)* yaitu *P(C1)* dan *P(C2)* sudah diketahui hasilnya di langkah sebelumnya.

Selanjutnya Hitung *P(X|Ci)* untuk $i = 1$ dan 2

- *P(outlook="sunny"|play="yes")*=2/9=0.222222222
- *P(outlook="sunny"|play="no")*=3/5=0.6

- *P(outlook="overcast"|play="yes")*=4/9=0.444444444
- *P(outlook="overcast"|play="no")*=0/5=0

- *P(outlook="rainy"|play="yes")*=3/9=0.333333333
- *P(outlook="rainy"|play="no")*=2/5=0.4

3. HITUNG JUMLAH KASUS YANG SAMA DENGAN CLASS YANG SAMA

Jika semua atribut dihitung, maka didapat hasil akhirnya seperti berikut ini:

Atribute	Parameter	No	Yes
Outlook	value=sunny	0.6	0.2222222222222222
Outlook	value=cloudy	0.0	0.4444444444444444
Outlook	value=rainy	0.4	0.3333333333333333
Temperature	value=hot	0.4	0.2222222222222222
Temperature	value=mild	0.4	0.4444444444444444
Temperature	value=cool	0.2	0.3333333333333333
Humidity	value=high	0.8	0.3333333333333333
Humidity	value=normal	0.2	0.6666666666666666
Windy	value=false	0.4	0.6666666666666666
Windy	value=true	0.6	0.3333333333333333

4. KALIKAN SEMUA NILAI HASIL SESUAI DENGAN DATA X YANG DICARI CLASS-NYA

Pertanyaan:

- Data X = (outlook=rainy, temperature=cool, humidity=high, windy=true)
- Main Golf atau tidak?

Kalikan semua nilai hasil dari data X

- P(X|play="yes") = 0.333333333* 0.333333333* 0.333333333*0.333333333 = 0.012345679
- P(X|play="no") = 0.4*0.2*0.8*0.6=0.0384

- P(X|play="yes")*P(C1) = 0.012345679*0.642857143
  = 0.007936508
- P(X|play="no")*P(C2) = 0.0384*0.357142857
  = **0.013714286**

Nilai "no" lebih besar dari nilai "yes" maka class dari data X tersebut adalah "**No**"

NAÏVE BAYES CLASSIFIER: COMMENTS

Advantages

- Easy to implement
- Good results obtained in most of the cases

Disadvantages

- Assumption: class conditional independence, therefore loss of accuracy
- Practically, dependencies exist among variables, e.g.:
  - Hospitals Patients Profile: age, family history, etc.
  - Symptoms: fever, cough etc.,
  - Disease: lung cancer, diabetes, etc.
- Dependencies among these cannot be modeled by Naïve Bayes Classifier

How to deal with these dependencies? Bayesian Belief Networks

K-NEAREST NEIGHBOR (K-NN)

K-NEAREST NEIGHBOR - 1

Konsep dasar dari K-NN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga terdekatnya dalam data pelatihan.

Penghitungan jarak dilakukan dengan konsep Euclidean.

Jumlah kelas yang paling banyak dengan jarak terdekat tersebut akan menjadi kelas dimana data evaluasi tersebut berada.

K-NEAREST NEIGHBOR - 2

Algoritma

- Tentukan parameter K = jumlah tetangga terdekat.
- Hitung jarak antara data yang akan dievaluasi dengan semua data pelatihan.
- Urutkan jarak yang terbentuk (urut naik) dan tentukan jarak terdekat sampai urutan ke-K.
- Pasangkan kelas (C) yang bersesuaian.
- Cari jumlah kelas terbanyak dari tetangga terdekat tersebut, dan tetapkan kelas tersebut sebagai kelas data yang dievaluasi.

INSTANCE BASED CLASSIFIERS

Set of Stored Cases

Atr1	……	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- Store the training samples
- Use training samples to predict the class label of unseen samples

Unseen Case

Atr1	……	AtrN

INSTANCE BASED CLASSIFIERS

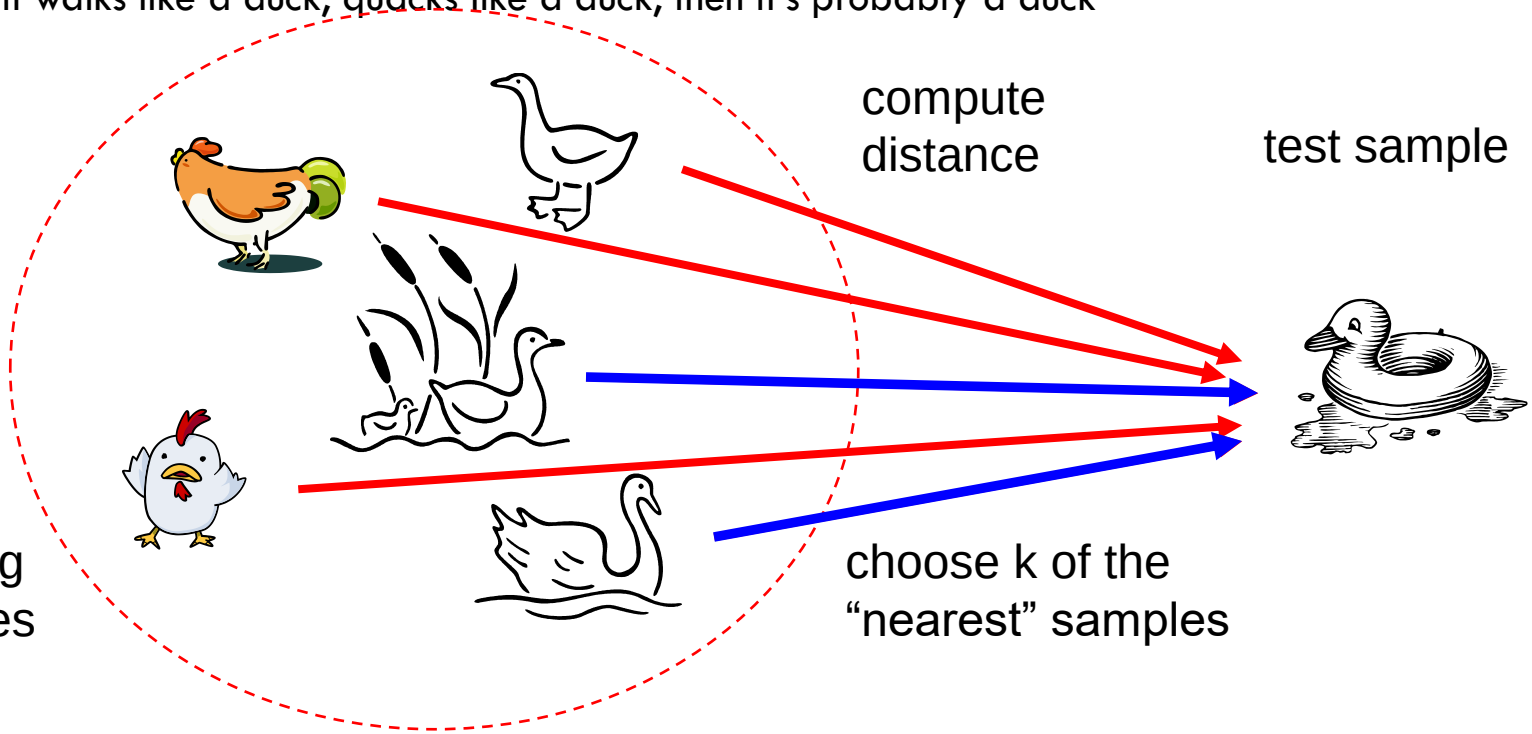
Examples:

- Rote learner
  - memorize entire training data
  - perform classification only if attributes of test sample match one of the training samples exactly
- Nearest neighbor
  - use *k* "closest" samples (nearest neighbors) to perform classification

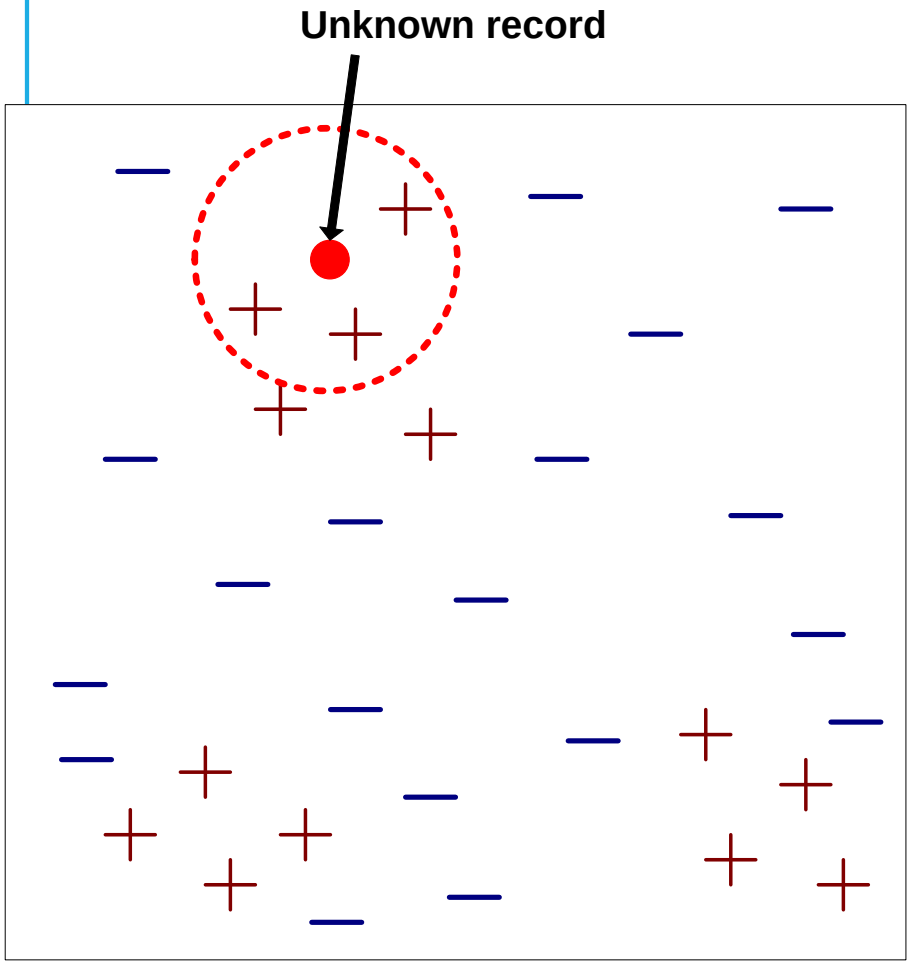
NEAREST NEIGHBOR CLASSIFIERS

Basic idea:

- If it walks like a duck, quacks like a duck, then it's probably a duck



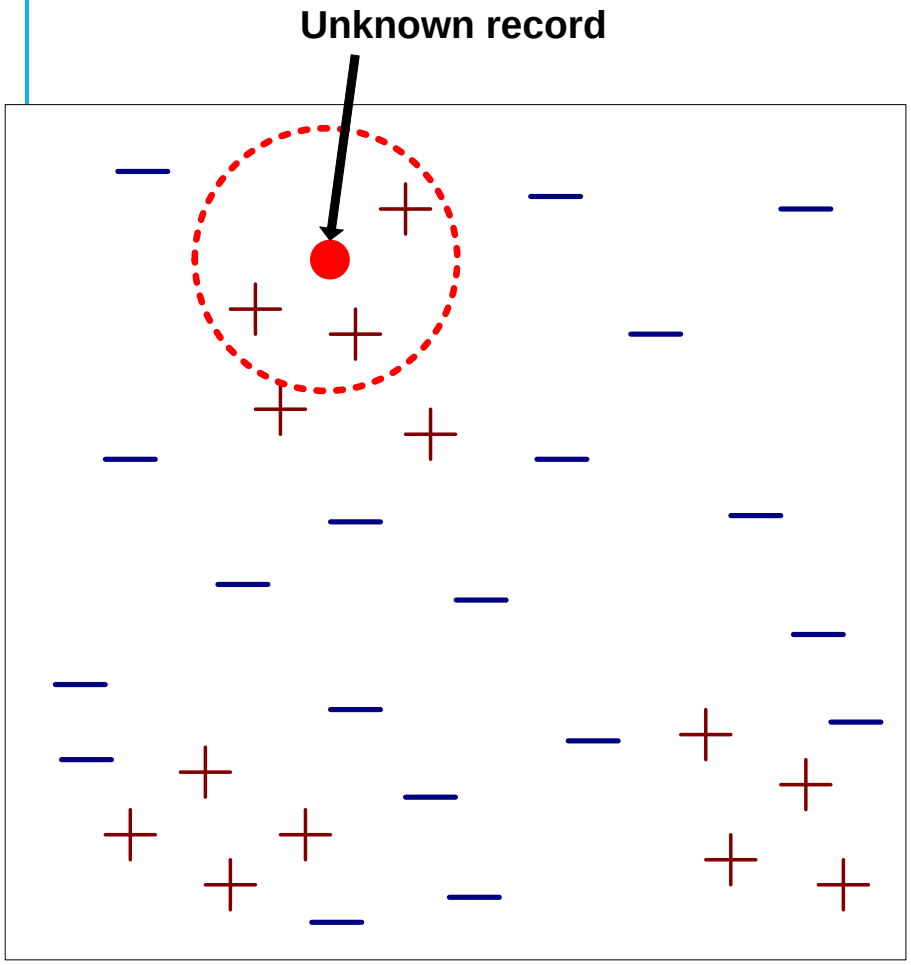
NEAREST NEIGHBOR CLASSIFIERS



Requires three inputs:

1. The set of stored samples
2. Distance metric to compute distance between samples
3. The value of *k*, the number of nearest neighbors to retrieve

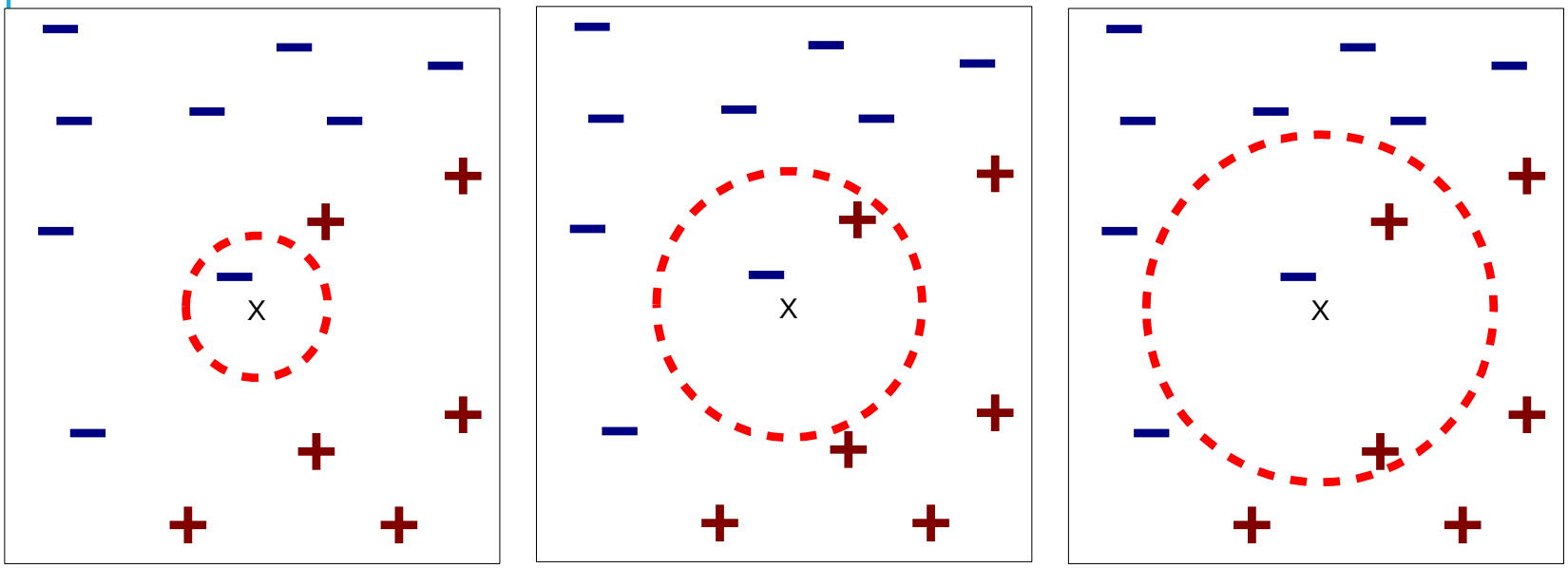
NEAREST NEIGHBOR CLASSIFIERS



To classify unknown record:

1. Compute distance to other training records
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

DEFINITION OF NEAREST NEIGHBOR



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k-nearest neighbors of a sample x are datapoints that have the *k* smallest distances to x

NEAREST NEIGHBOR CLASSIFICATION

Compute distance between two points:

- Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

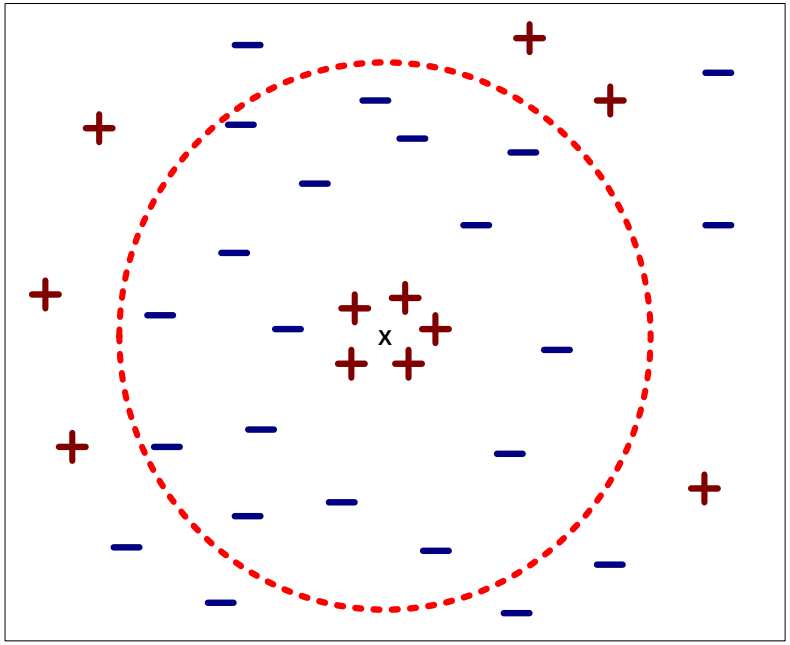
Options for determining the class from nearest neighbor list

- Take majority vote of class labels among the *k*-nearest neighbors
- Weight the votes according to distance
  - example: weight factor $w = 1 / d^2$

NEAREST NEIGHBOR CLASSIFICATION

Choosing the value of *k*:

- If *k* is too small, sensitive to noise points
- If *k* is too large, neighborhood may include points from other classes



NEAREST NEIGHBOR CLASSIFICATION

Scaling issues

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example:
  - height of a person may vary from 1.5 m to 1.8 m
  - weight of a person may vary from 90 lb to 300 lb
  - income of a person may vary from $10K to $1M

NEAREST NEIGHBOR CLASSIFICATION…

Problem with Euclidean measure:

- High dimensional data
  - curse of dimensionality
- Can produce counter-intuitive results

1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1	vs	0 0 0 0 0 0 0 0 0 0 0 1
d = 1.4142		d = 1.4142

- one solution: normalize the vectors to unit length

NEAREST NEIGHBOR CLASSIFICATION

k-Nearest neighbor classifier is a lazy learner

- Does not build model explicitly.
- Unlike eager learners such as decision tree induction and rule-based systems.
- Classifying unknown samples is relatively expensive.

k-Nearest neighbor classifier is a local model, vs. global model of linear classifiers.

CONTOH KASUS

X1 = Ketahanan Asam (detik)	X2 = Kekuatan (Kg / meter persegi)	Y = Klasifikasi
7	7	Buruk
7	4	Buruk
3	4	Baik
1	4	Baik

PENYELESAIAN

1. *Tentukan parameter K = jumlah tetangga terdekat*

   Misalkan menggunakan K = 3

2. *Hitung jarak antara permintaan dan contoh-contoh training semua*

   Koordinat query instance adalah (3, 7), kemudian untuk menghitung jarak, kita menghitung jarak kuadrat yang lebih cepat (tanpa akar kuadrat)

X1 = Ketahanan Asam (detik)	X2 = Kekuatan (Kg / meter persegi)	Jarak Kuadrat untuk Query-Instance (3,7)
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

PENYELESAIAN-2

3. *Urutkan jarak dan tentukan tetangga terdekat berdasarkan jarak terdekat ke-K*

X1 = Ketahanan Asam (detik)	X2 = Kekuatan (Kg / meter persegi)	Jarak Kuadrat untuk Query-Instance (3,7)	Urutan Jarak Terdekat	Apakah termasuk dalam 3-Nearest Neighbors?
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes
7	4	$(7-3)^2+(4-7)^2=25$	4	No
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes

PENYELESAIAN-3

4. *Kumpulkan kategori Y dari tetangga terdekat.* Perhatikan baris kedua pada kolom terakhir bahwa kategori tetangga terdekat (Y) tidak dimasukkan karena peringkat data ini lebih dari 3 (= K).

X1 = Ketahanan Asam (detik)	X2 = Kekuatan (Kg / meter persegi)	Jarak Kuadrat untuk Query-Instance (3,7)	Urutan Jarak Terdekat	Apakah termasuk dalam 3-Nearest Neighbors?	Y = Kategori dari Nearest Neighbor
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes	Buruk
7	4	$(7-3)^2+(4-7)^2=25$	4	No	-
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes	Baik
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes	Baik

PENYELESAIAN-4

5. *Gunakan mayoritas sederhana dari kategori tetangga terdekat sebagai nilai prediksi query-instance*

   Terdapat 2 baik dan 1 buruk pada masing-masing kategori, dimana 2 > 1 maka kita menyimpulkan bahwa kertas tisu baru yang lulus uji laboratorium dengan X1 = 3 dan X2 = 7 adalah termasuk dalam kategori **baik.**