墨学教育
—MELBSTUDY—

FIT5047 Module 5

Machine Learning

授课老师：Joe

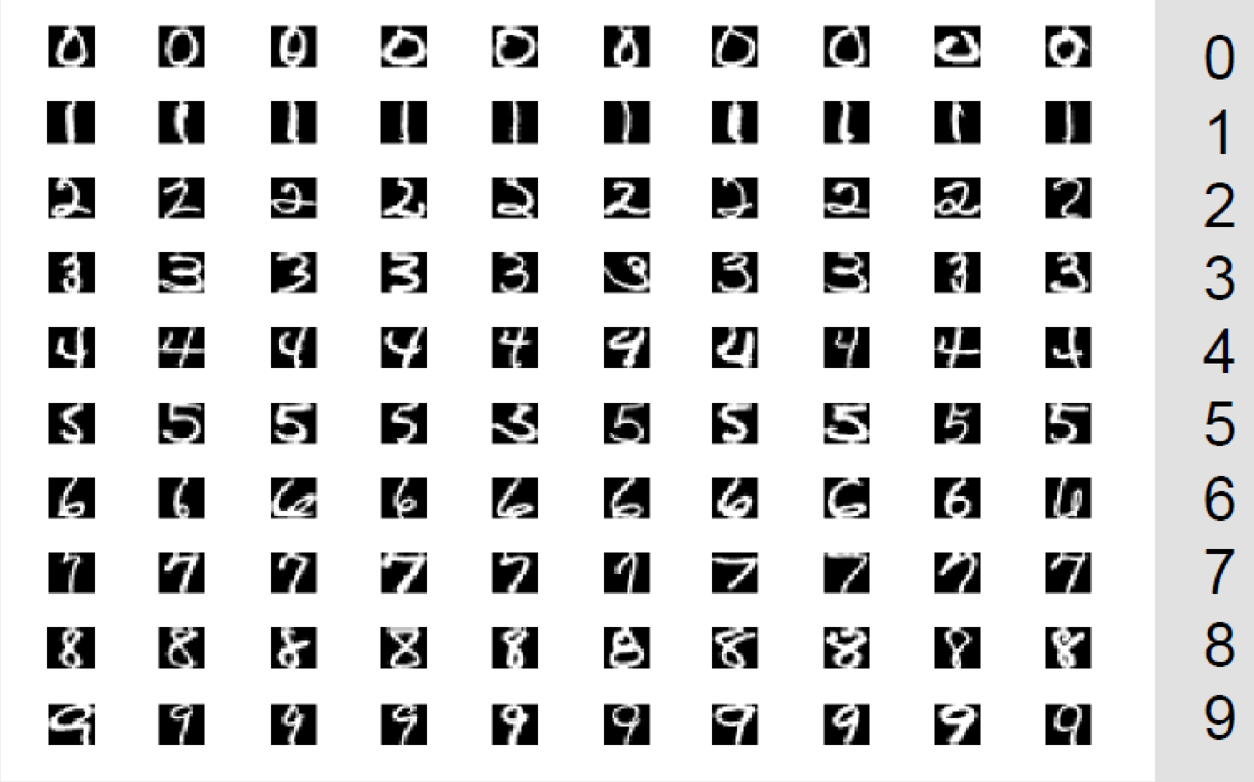
FIT5047平时班 – Module 5

- **Supervised machine learning**
  - Decision trees
  - Naïve Bayes
  - k Nearest Neighbour (k-NN)
  - Regression
  - (Logistic regression)
- **Clustering (Unsupervised learning)**
  - The clustering problem
  - Similarity measures
  - The K-means algorithm

- ***Supervised learning*: correct answers are provided for each input**
  - E.g., Decision Trees, Naïve Bayes, K-Nearest Neighbour (k-NN), Regression, Neural Nets
- ***Unsupervised learning*: correct answers are *not* given, must discover patterns in input data**
  - E.g., K-Means, Snob (Minimum Message Length Principle)

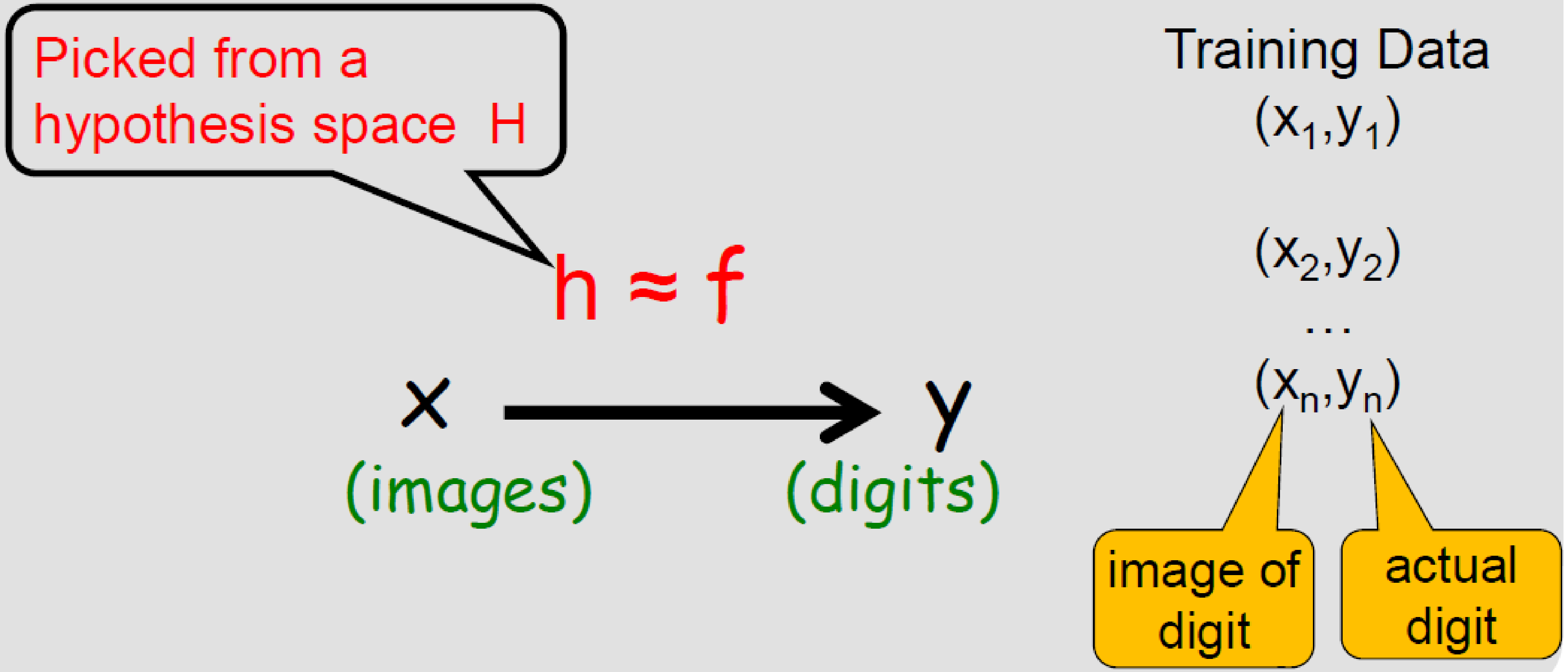
Supervised Learning

- **Classify images into one of the digits: 0, 1, …, 9**
  - Useful for the post office to automatically detect addresses



FIT5047平时班 – Module 5

- **Learn a function f from examples**



Picked from a hypothesis space H

$h \approx f$

$x \longrightarrow y$

(images) (digits)

Training Data

(x_1,y_1)

(x_2,y_2)

…

(x_n,y_n)

image of digit

actual digit

Setup:

- f is the **unknown** target function
- We are given some sample pairs from it (x, f(x))

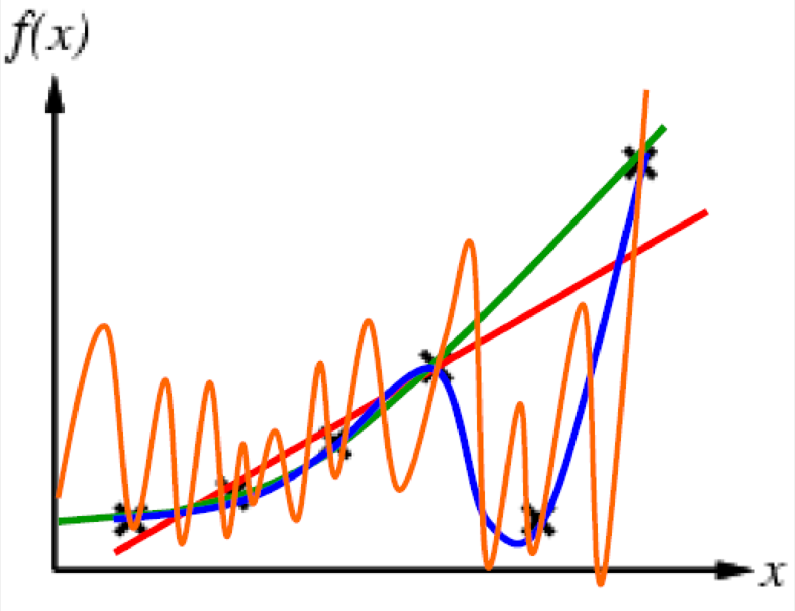
Problem: learn a function hypothesis h

- Based on the set of ***training*** examples
- Such that $h \approx f$ (h approximates f as best as possible)
  - **> h must generalize well on unseen examples**

FIT5047平时班 – Module 5

Curve fitting (regression):

h= Straight line? Quadratic? Cubic? Other?



Training Data

(x_1,y_1)

(x_2,y_2)

…

(x_n,y_n)

FIT5047平时班 – Module 5

- **Big Idea 1:**
  Pick h from the space H which agrees with f on the training set
  - Complete and consistent

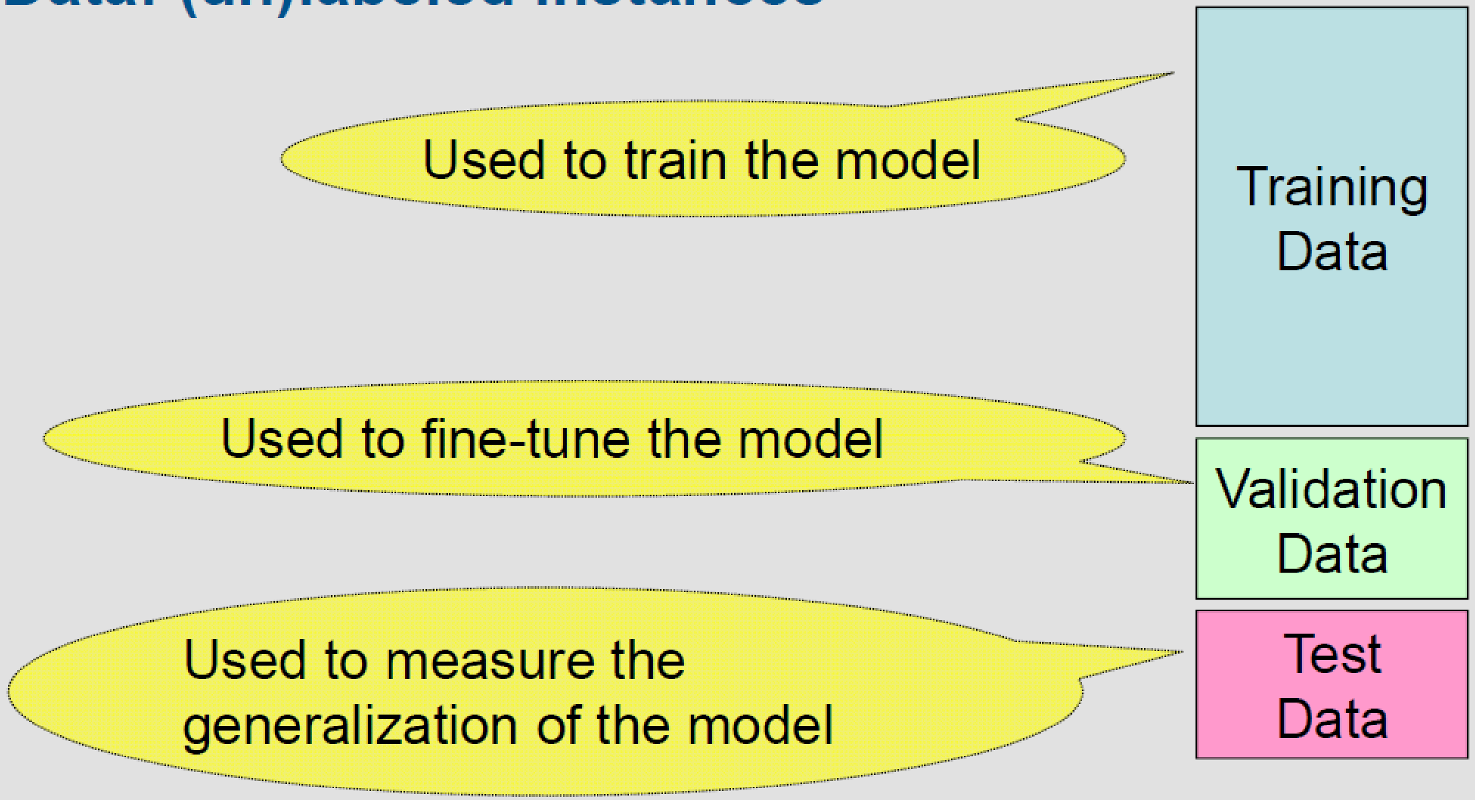
- **Big Idea 2:** **Prefer a simpler hypothesis to complex ones *provided both explain the data equally well***

Bias and Variance

- **Bias is the true error of the best classifier in the concept class**
  - Bias is high if the concept class cannot model the true data distribution well, e.g., it is too simple
  - High bias → both training and test error are high
- **Variance is the error of a trained classifier with respect to the best classifier in the concept class**
  - Variance decreases with more training data, and increases with more complicated classifiers
  - High variance → training error is low, and test error is high

Types of Data

- **Data: (un)labeled instances**



Used to train the model

Used to fine-tune the model

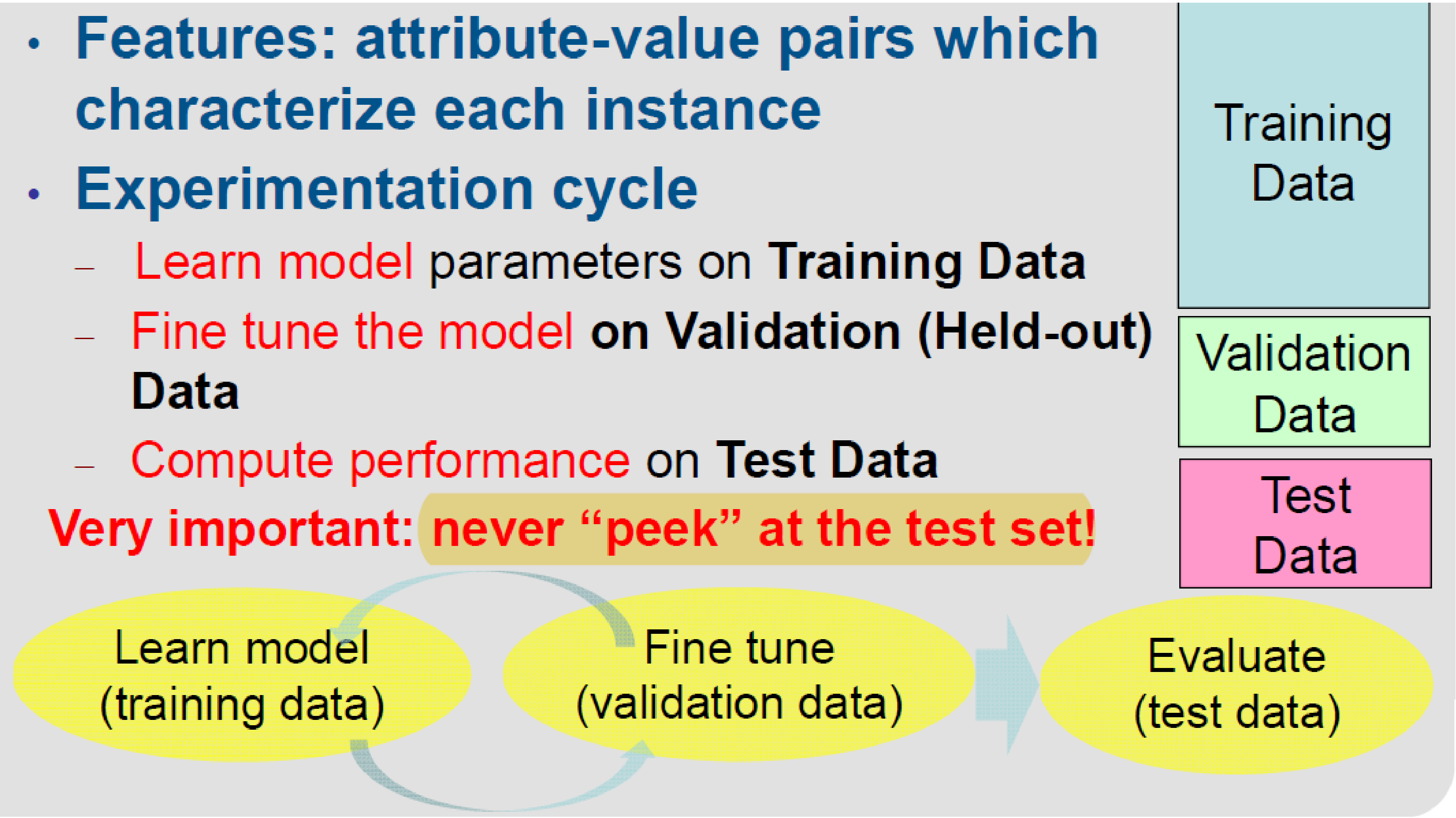
Used to measure the generalization of the model

Training Data

Validation Data

Test Data

FIT5047平时班 – Module 5



- **Features: attribute-value pairs which characterize each instance**
- **Experimentation cycle**
  - Learn model parameters on **Training Data**
  - Fine tune the model **on Validation (Held-out) Data**
  - Compute performance on **Test Data**

Very important: never "peek" at the test set!

Training Data

Validation Data

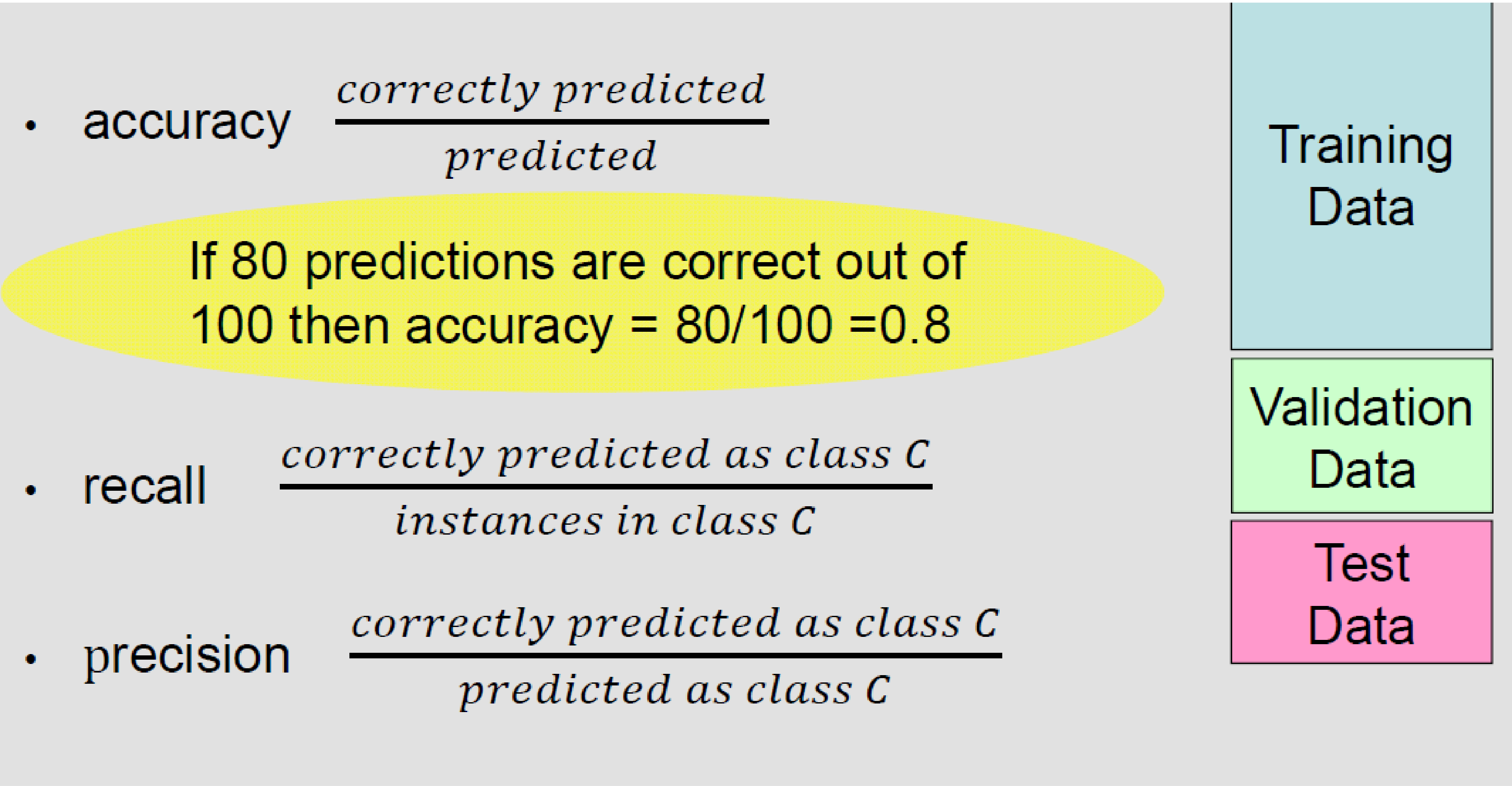
Test Data

Learn model (training data)

Fine tune (validation data)

Evaluate (test data)

FIT5047平时班 – Module 5

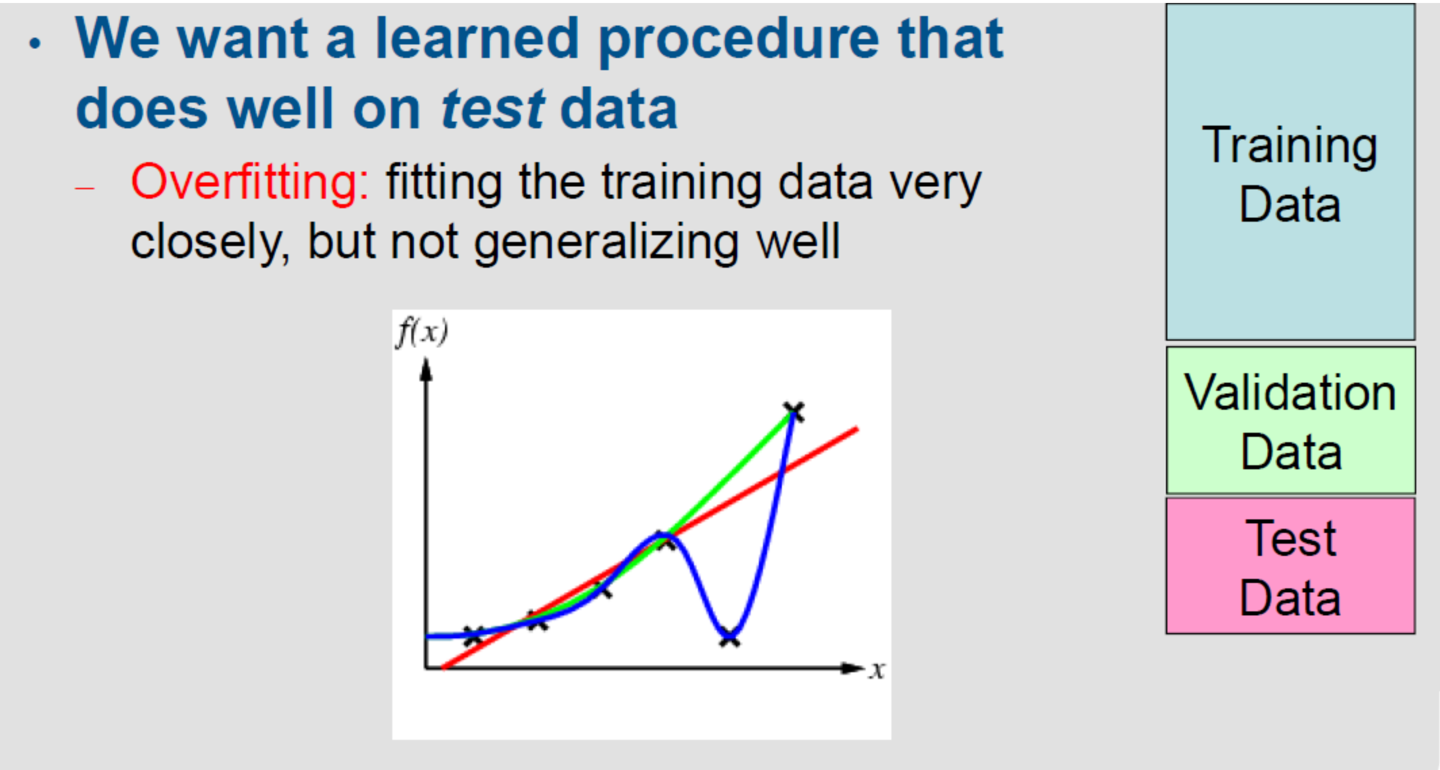


- accuracy $$\frac{correctly\ predicted}{predicted}$$

If 80 predictions are correct out of 100 then accuracy = 80/100 =0.8

- recall $$\frac{correctly\ predicted\ as\ class\ C}{instances\ in\ class\ C}$$
- precision $$\frac{correctly\ predicted\ as\ class\ C}{predicted\ as\ class\ C}$$

FIT5047平时班 – Module 5



- **We want a learned procedure that does well on *test* data**
  - Overfitting: fitting the training data very closely, but not generalizing well

Training Data

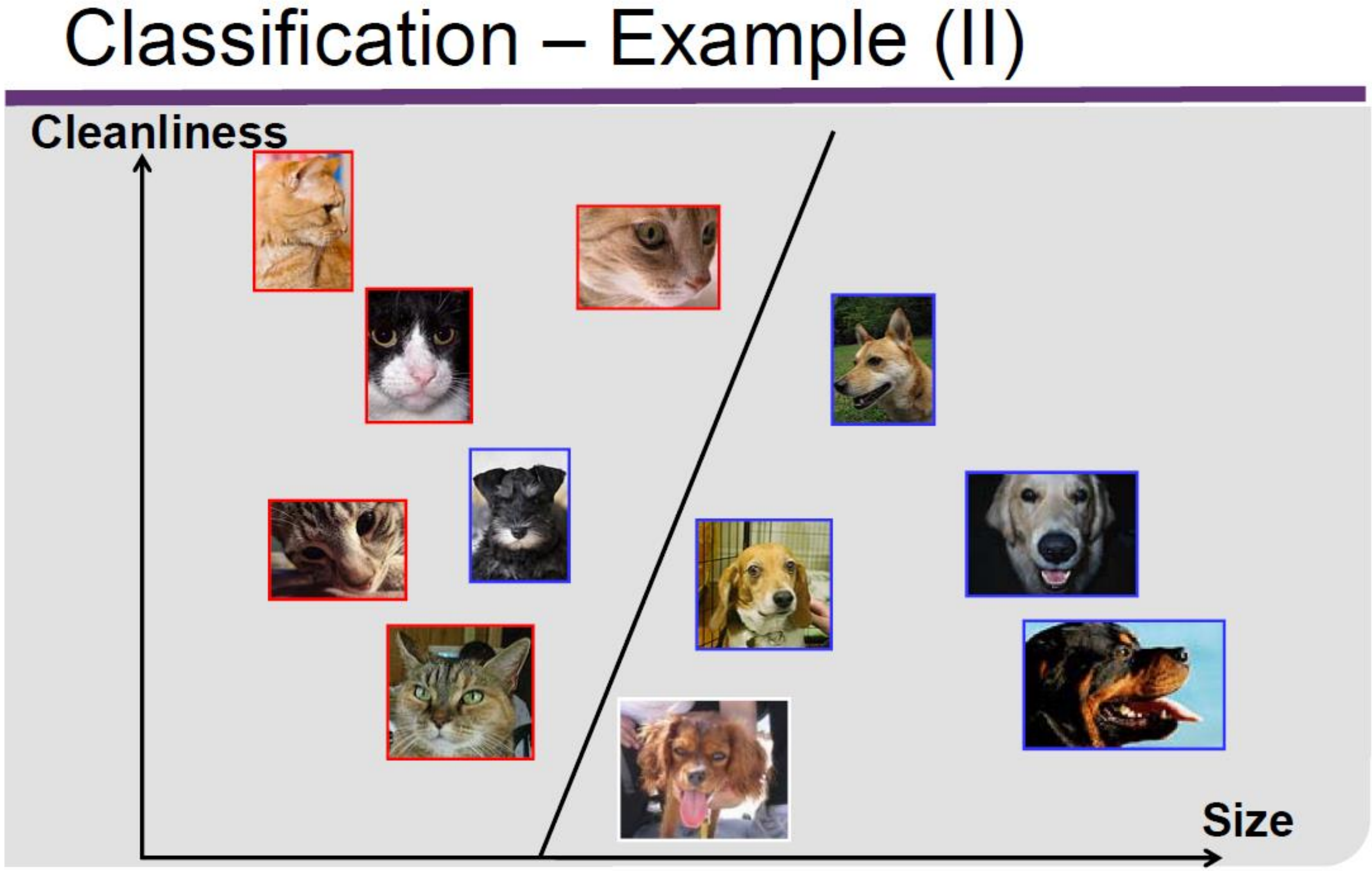
Validation Data

Test Data

Type of Inferred Value – Classification vs Regression

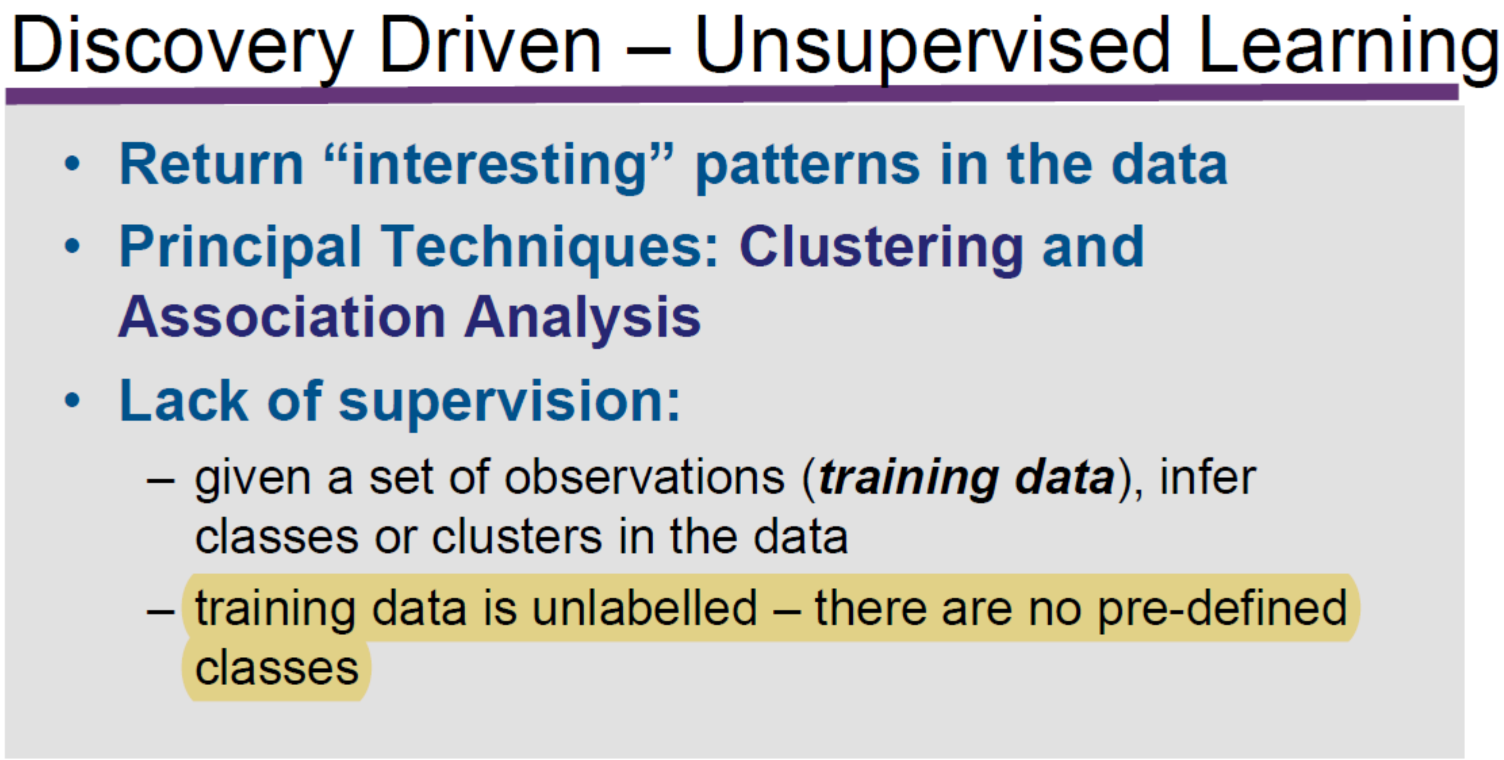
- **Classification**
  - Infers categorical or discrete values
- **Regression**
  - Infers continuous or ordered values

FIT5047平时班 – Module 5



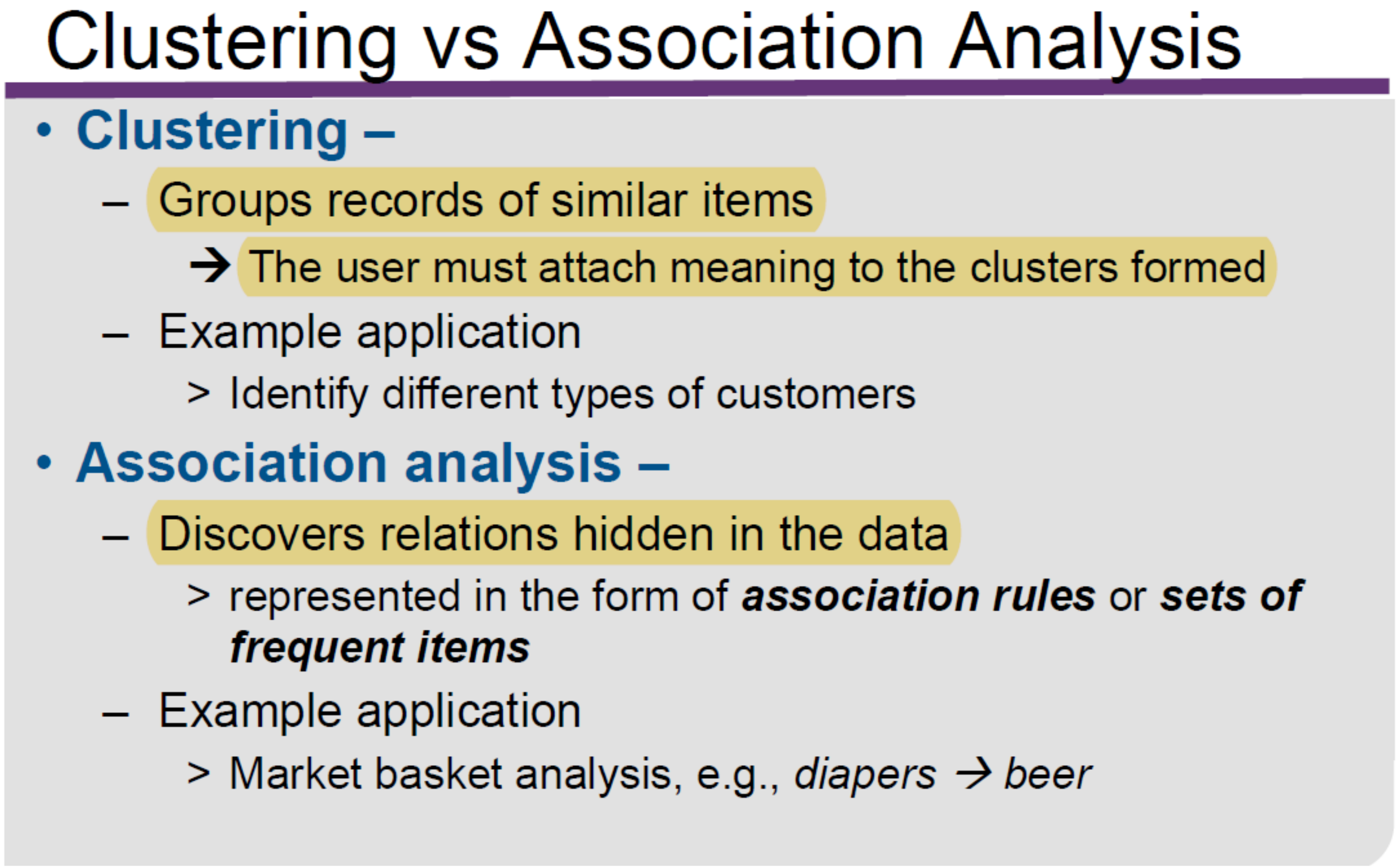
Classification – Example (II)

Regression – Example (I)



Discovery Driven – Unsupervised Learning

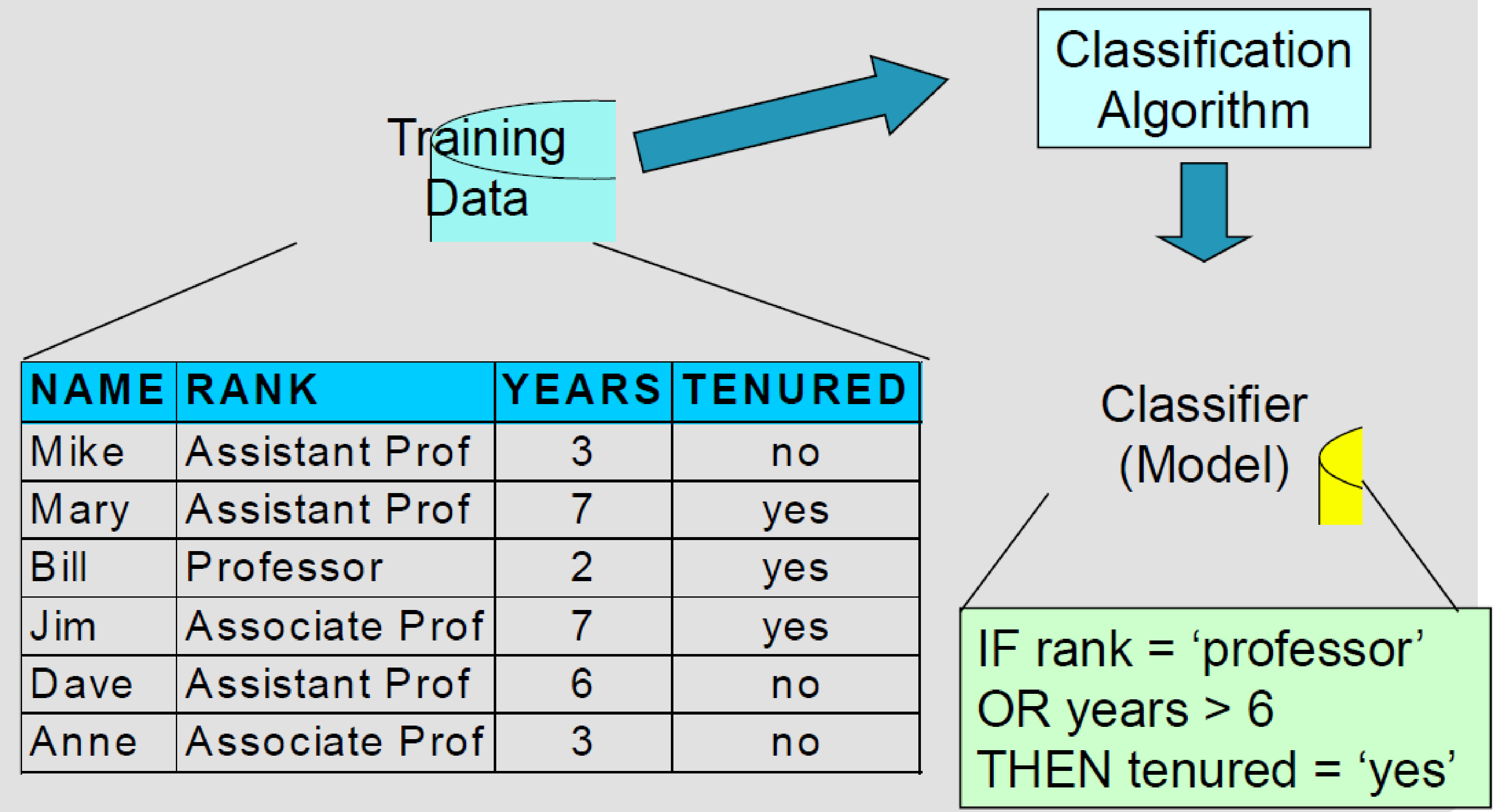
- **Return “interesting” patterns in the data**
- **Principal Techniques: Clustering and Association Analysis**
- **Lack of supervision:**
  - given a set of observations (***training data***), infer classes or clusters in the data
  - training data is unlabelled – there are no pre-defined classes



Clustering vs Association Analysis

- **Clustering –**
  - Groups records of similar items
    - → The user must attach meaning to the clusters formed
  - Example application
    - > Identify different types of customers
- **Association analysis –**
  - Discovers relations hidden in the data
    - > represented in the form of ***association rules*** or ***sets of frequent items***
  - Example application
    - > Market basket analysis, e.g., *diapers* → *beer*

FIT5047平时班 – Module 5



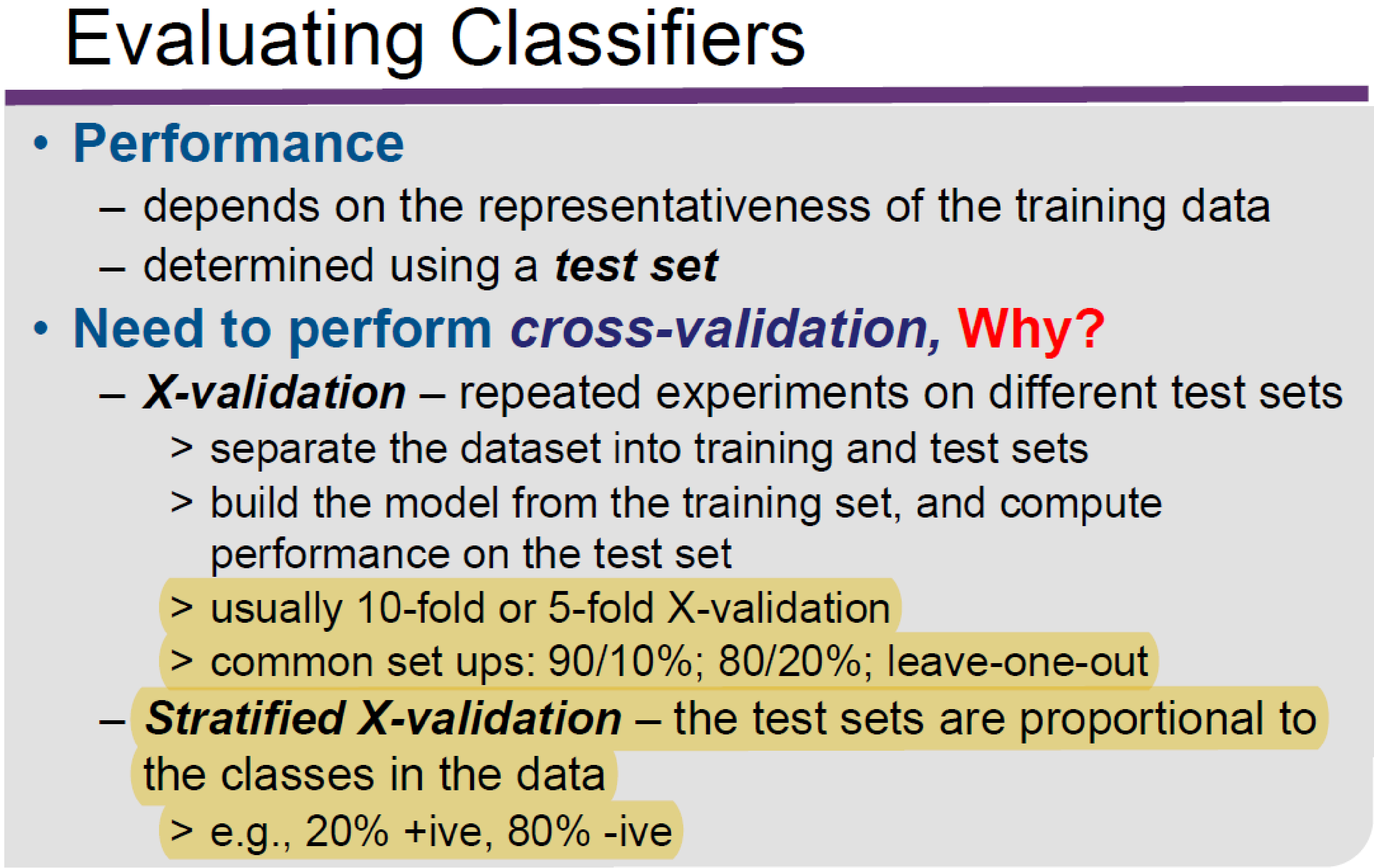
Training Data

Classification Algorithm

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classifier (Model)

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

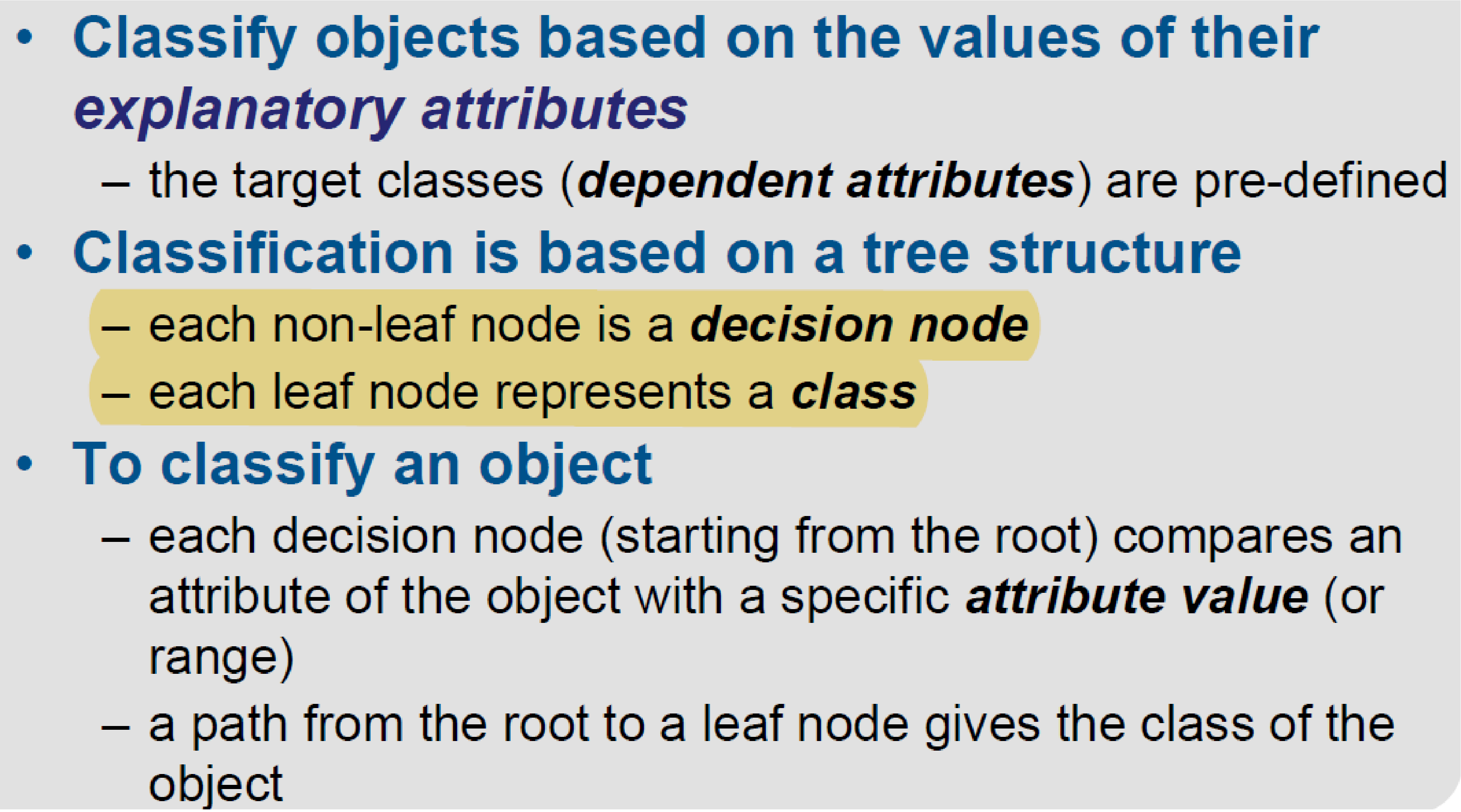


Evaluating Classifiers

- **Performance**
  - – depends on the representativeness of the training data
  - – determined using a ***test set***
- **Need to perform *cross-validation*, Why?**
  - – ***X-validation*** – repeated experiments on different test sets
    - > separate the dataset into training and test sets
    - > build the model from the training set, and compute performance on the test set
    - > usually 10-fold or 5-fold X-validation
    - > common set ups: 90/10%; 80/20%; leave-one-out
  - – ***Stratified X-validation*** – the test sets are proportional to the classes in the data
    - > e.g., 20% +ive, 80% -ive

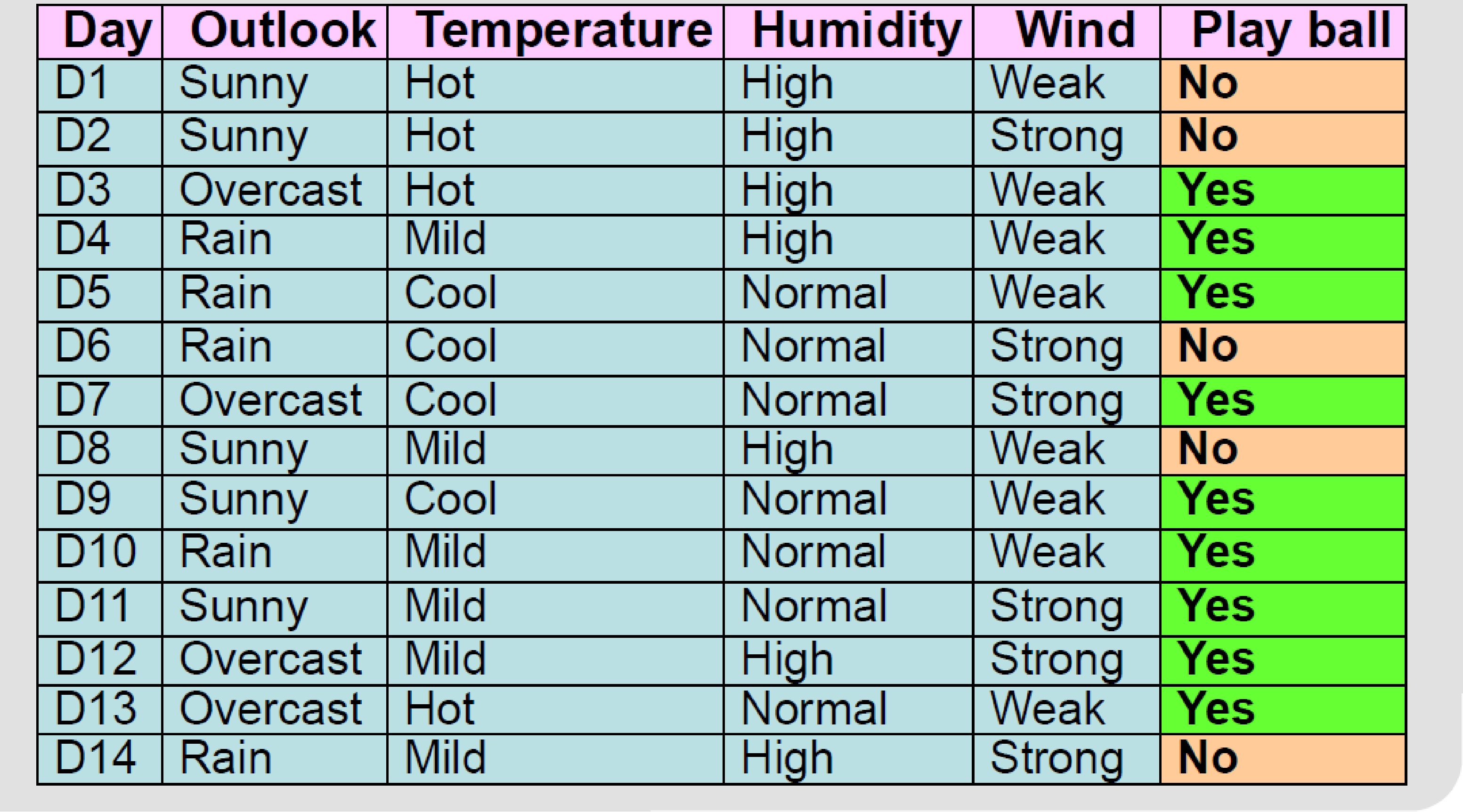
Decision Trees

FIT5047平时班 – Module 5



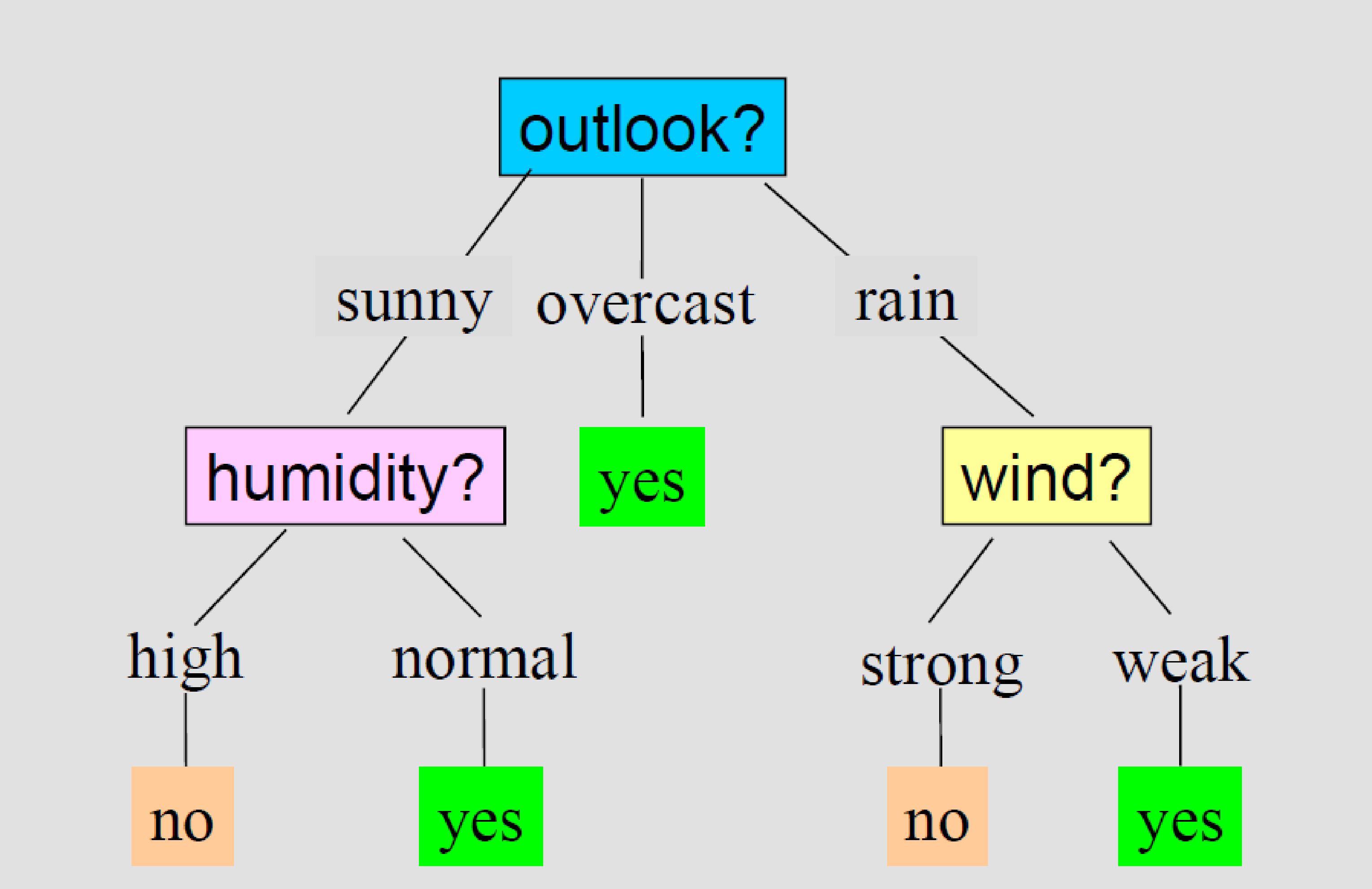
- **Classify objects based on the values of their *explanatory attributes***
  - the target classes (***dependent attributes***) are pre-defined
- **Classification is based on a tree structure**
  - each non-leaf node is a ***decision node***
  - each leaf node represents a ***class***
- **To classify an object**
  - each decision node (starting from the root) compares an attribute of the object with a specific ***attribute value*** (or range)
  - a path from the root to a leaf node gives the class of the object

DT Example – Training Dataset

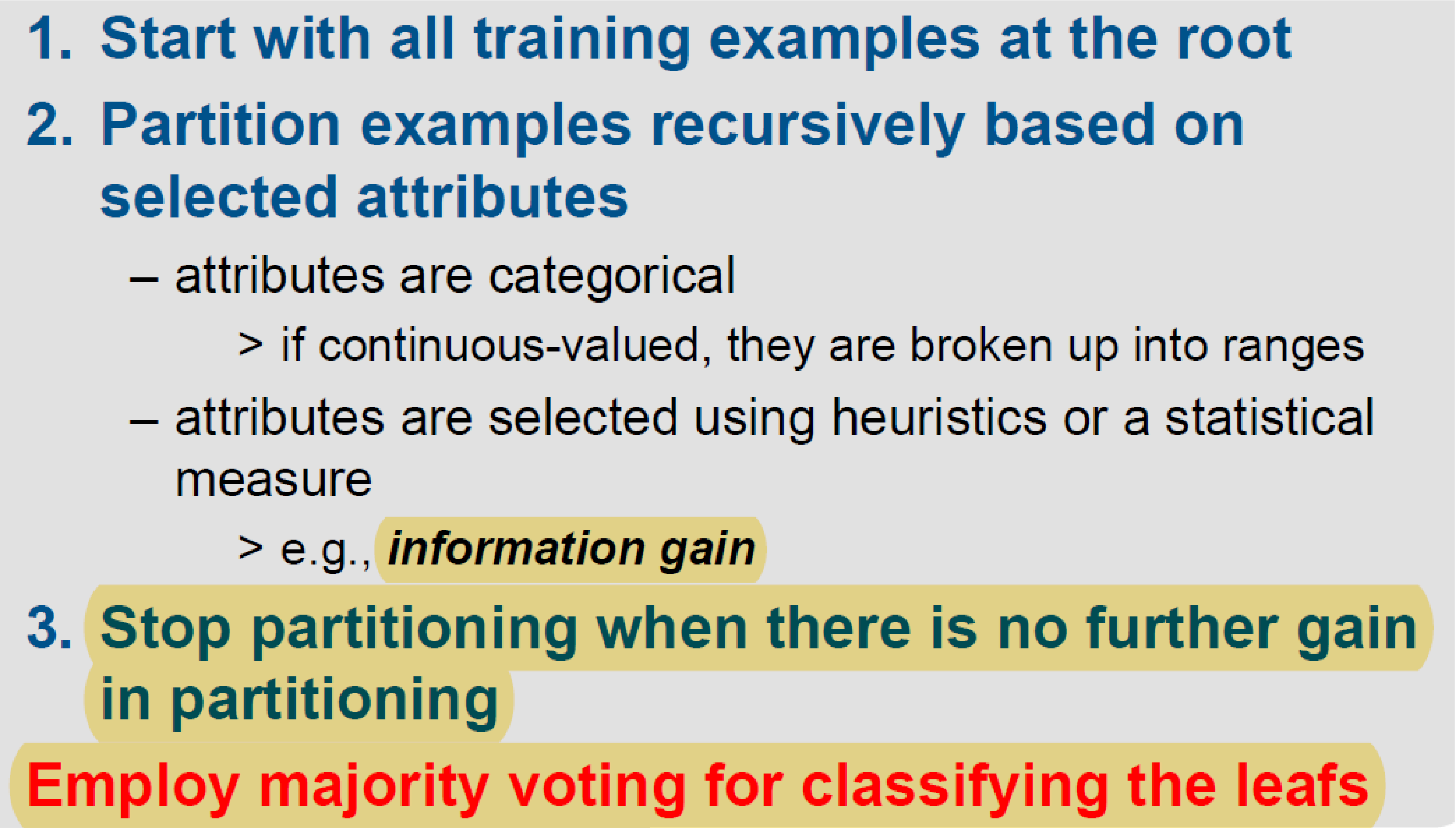


Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	**No**
D2	Sunny	Hot	High	Strong	**No**
D3	Overcast	Hot	High	Weak	**Yes**
D4	Rain	Mild	High	Weak	**Yes**
D5	Rain	Cool	Normal	Weak	**Yes**
D6	Rain	Cool	Normal	Strong	**No**
D7	Overcast	Cool	Normal	Strong	**Yes**
D8	Sunny	Mild	High	Weak	**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**

FIT5047平时班 – Module 5



FIT5047平时班 – Module 5

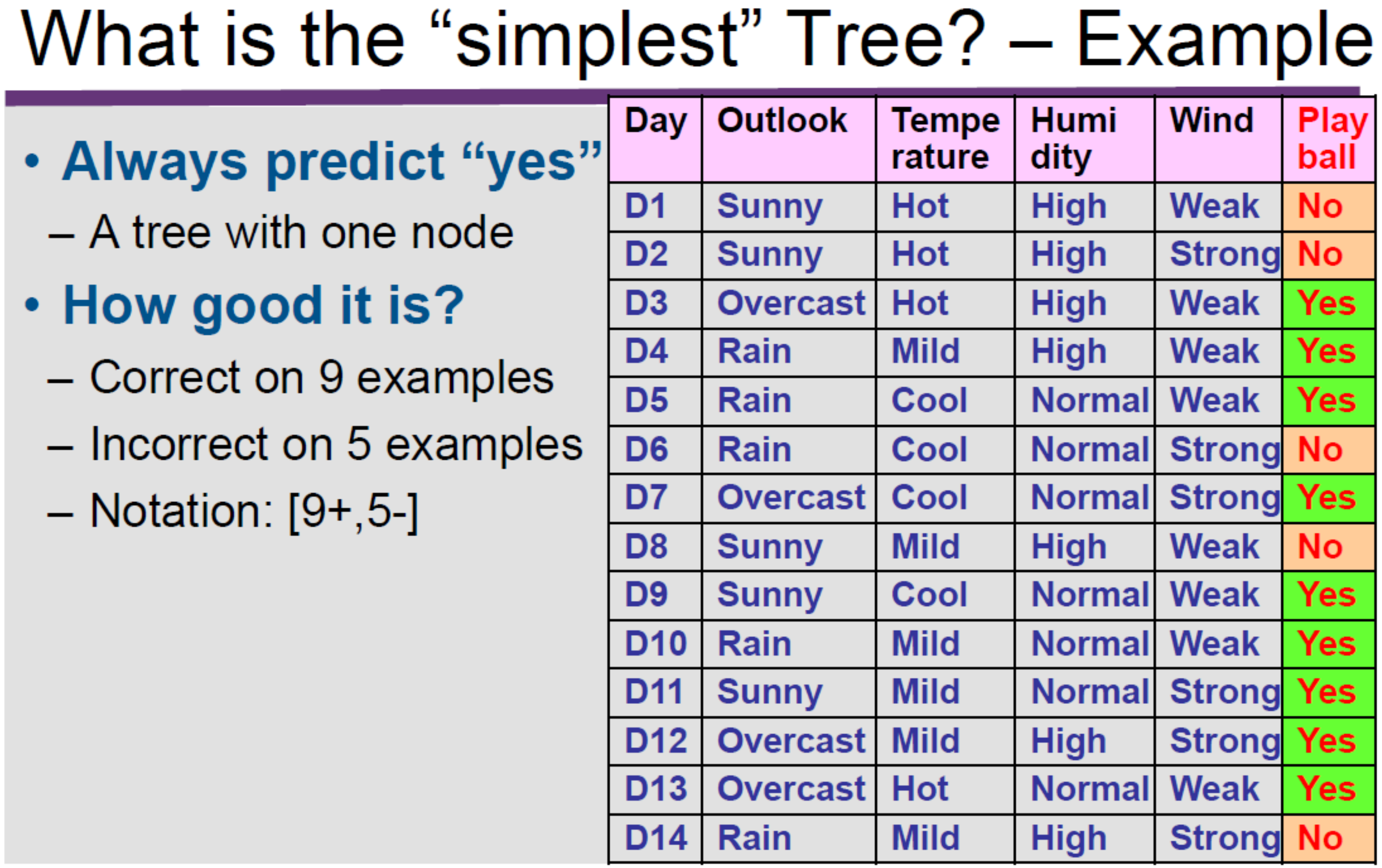


1. **Start with all training examples at the root**
2. **Partition examples recursively based on selected attributes**
   - attributes are categorical
     - > if continuous-valued, they are broken up into ranges
   - attributes are selected using heuristics or a statistical measure
     - > e.g., ***information gain***
3. **Stop partitioning when there is no further gain in partitioning**

Employ majority voting for classifying the leafs

FIT5047平时班 – Module 5

- **Approach – choose the attribute that best separates training examples into targeted classes**
- **Examples of proposed techniques**
  - ***Information Gain*** [Quinlan, 1975]

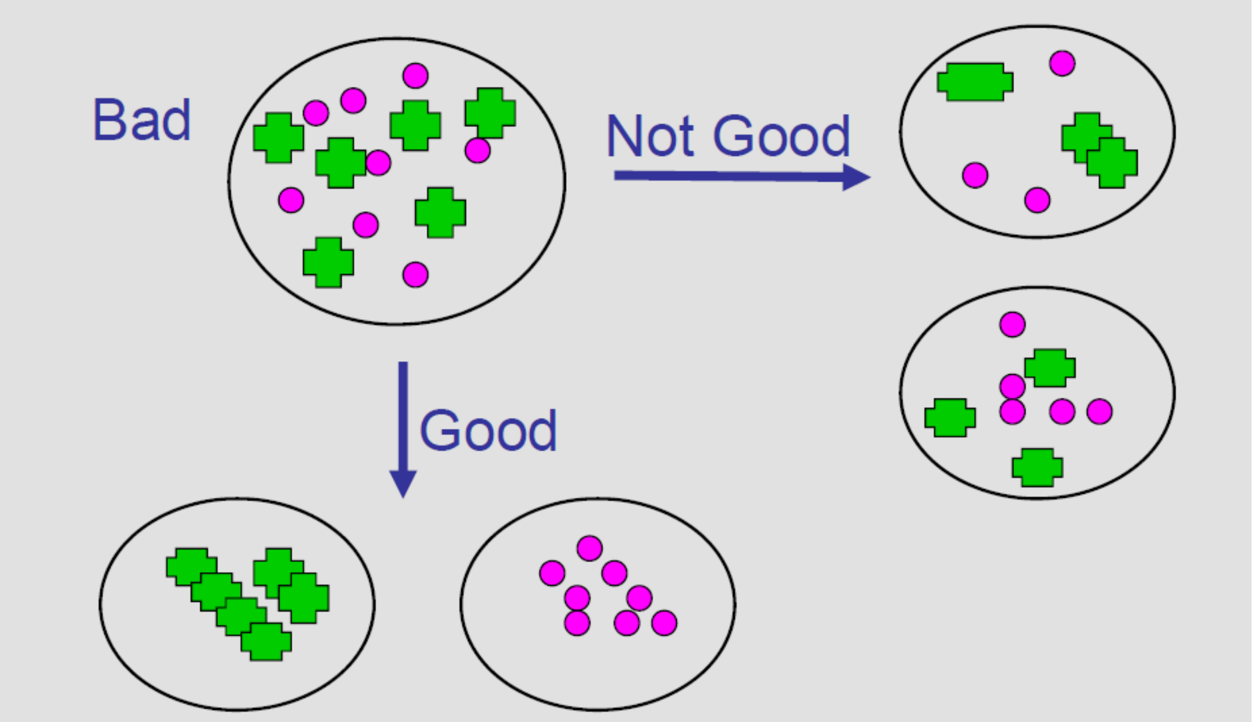


What is the “simplest” Tree? – Example

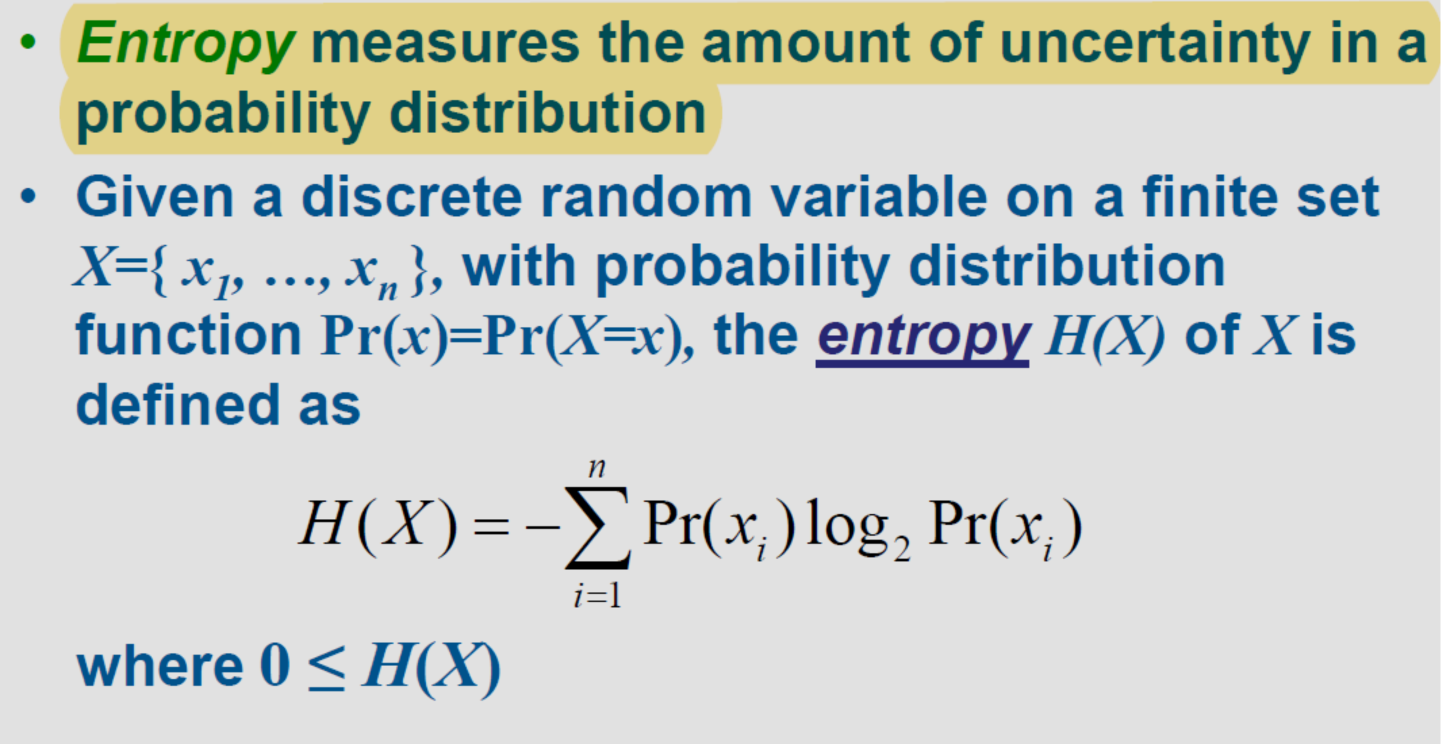
- **Always predict “yes”**
  - A tree with one node
- **How good it is?**
  - Correct on 9 examples
  - Incorrect on 5 examples
  - Notation: [9+,5-]

Day	Outlook	Temperature	Humidity	Wind	Play ball
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

FIT5047平时班 – Module 5



FIT5047平时班 – Module 5



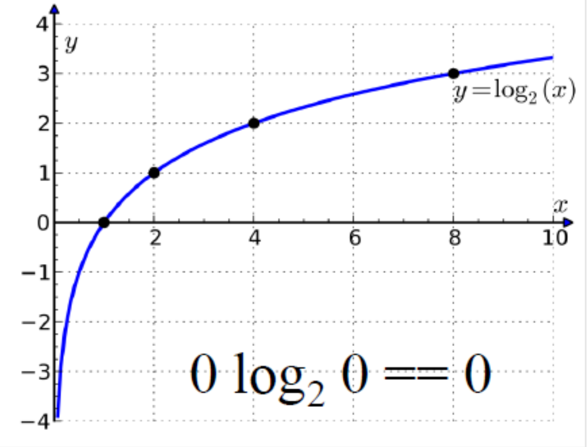
- *Entropy* measures the amount of uncertainty in a probability distribution
- Given a discrete random variable on a finite set $X=\{x_1, \ldots, x_n\}$, with probability distribution function $\Pr(x)=\Pr(X=x)$, the *entropy* $H(X)$ of X is defined as

$$H(X) = -\sum_{i=1}^{n} \Pr(x_i)\log_2 \Pr(x_i)$$

where $0 \le H(X)$

FIT5047平时班 – Module 5

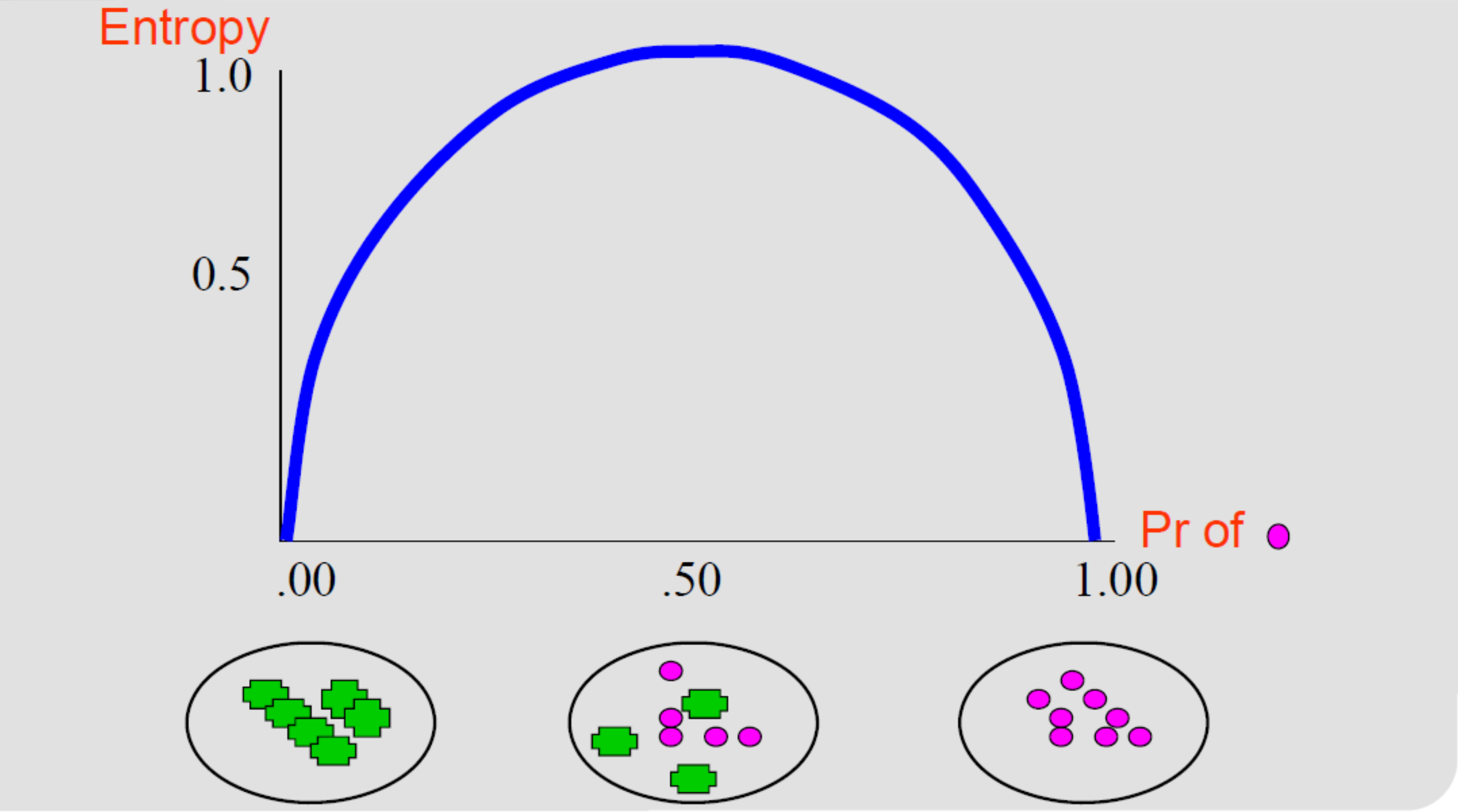
$$H(X) = -\sum_{i=1}^{n} \Pr(x_i)\log_2 \Pr(x_i)$$



$$0 \log_2 0 == 0$$

- **Suppose there is a random variable *S* that has value *a* or *b***
  - Let Pr(a) = 1 and Pr(b) = 0
  - Let Pr(a) = 0.9 and Pr(b) = 0.1
  - Let Pr(a) = 0.5 and Pr(b) = 0.5
  - Which probability assignment maximizes ***H(S)*** ?

Entropy

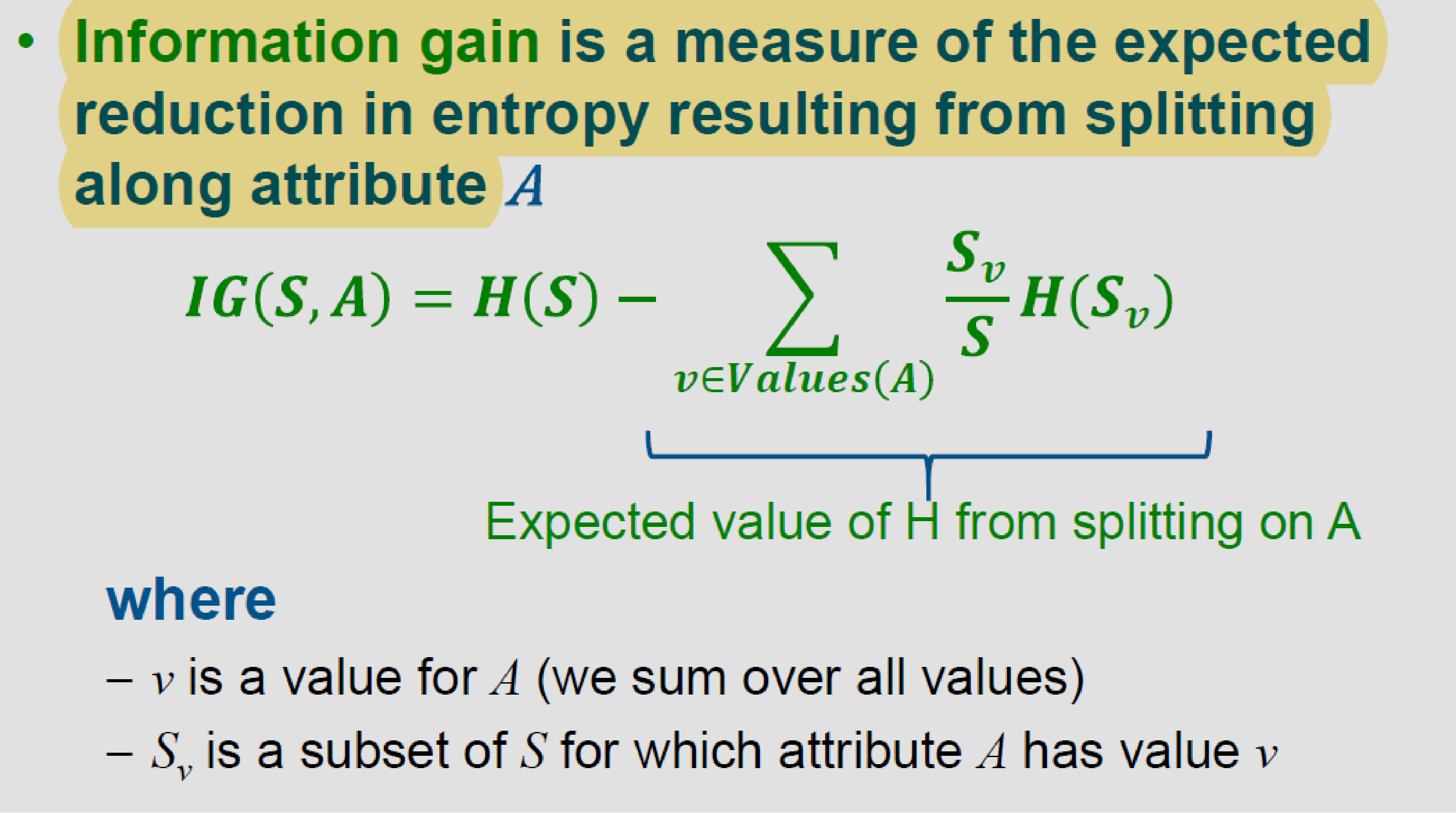


FIT5047平时班 – Module 5

- **Let S be a set of examples**
  - Labeled positive or negative
- **Entropy(S) = -P $\log_2(P)$ - N $\log_2(N)$**
  - **P** is the proportion of positive examples
  - **N** is the proportion of negative examples
  - 0 log 0 == 0
- **Example: S has 9 pos and 5 neg**
  - Entropy([9+, 5-])

$$= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.94$$

FIT5047平时班 – Module 5



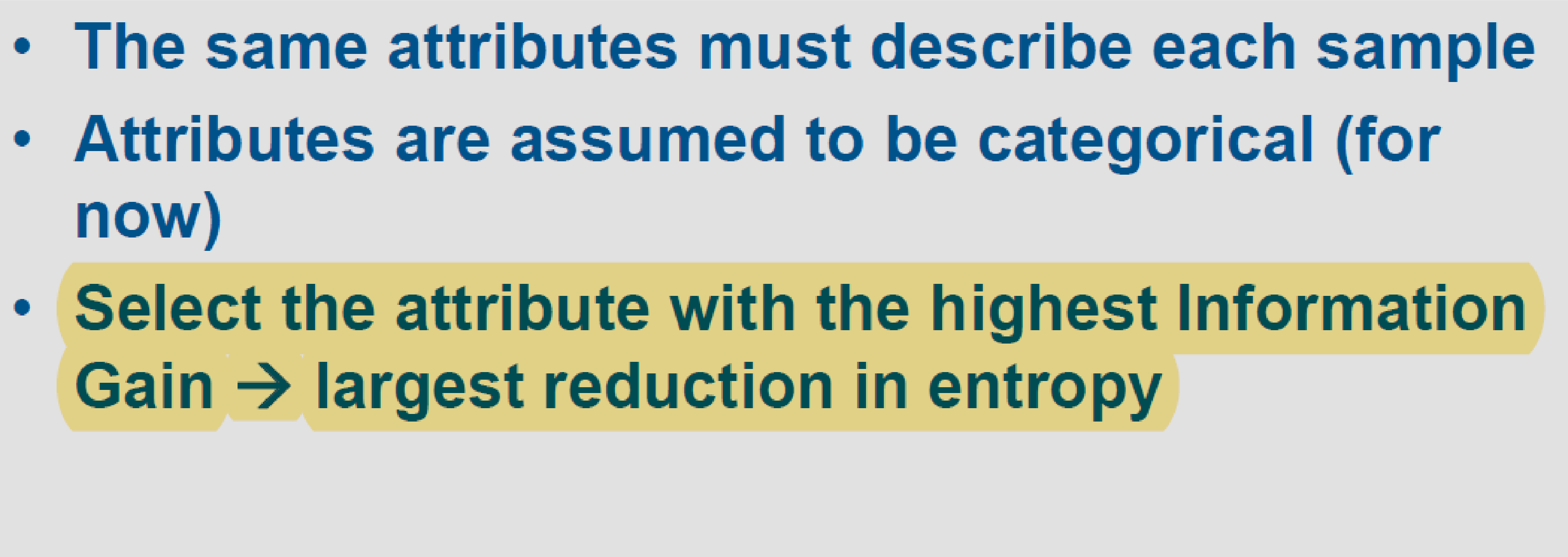
- Information gain is a measure of the expected reduction in entropy resulting from splitting along attribute A

$$IG(S,A) = H(S) - \underbrace{\sum_{v \in Values(A)} \frac{S_v}{S} H(S_v)}_{\text{Expected value of H from splitting on A}}$$

where

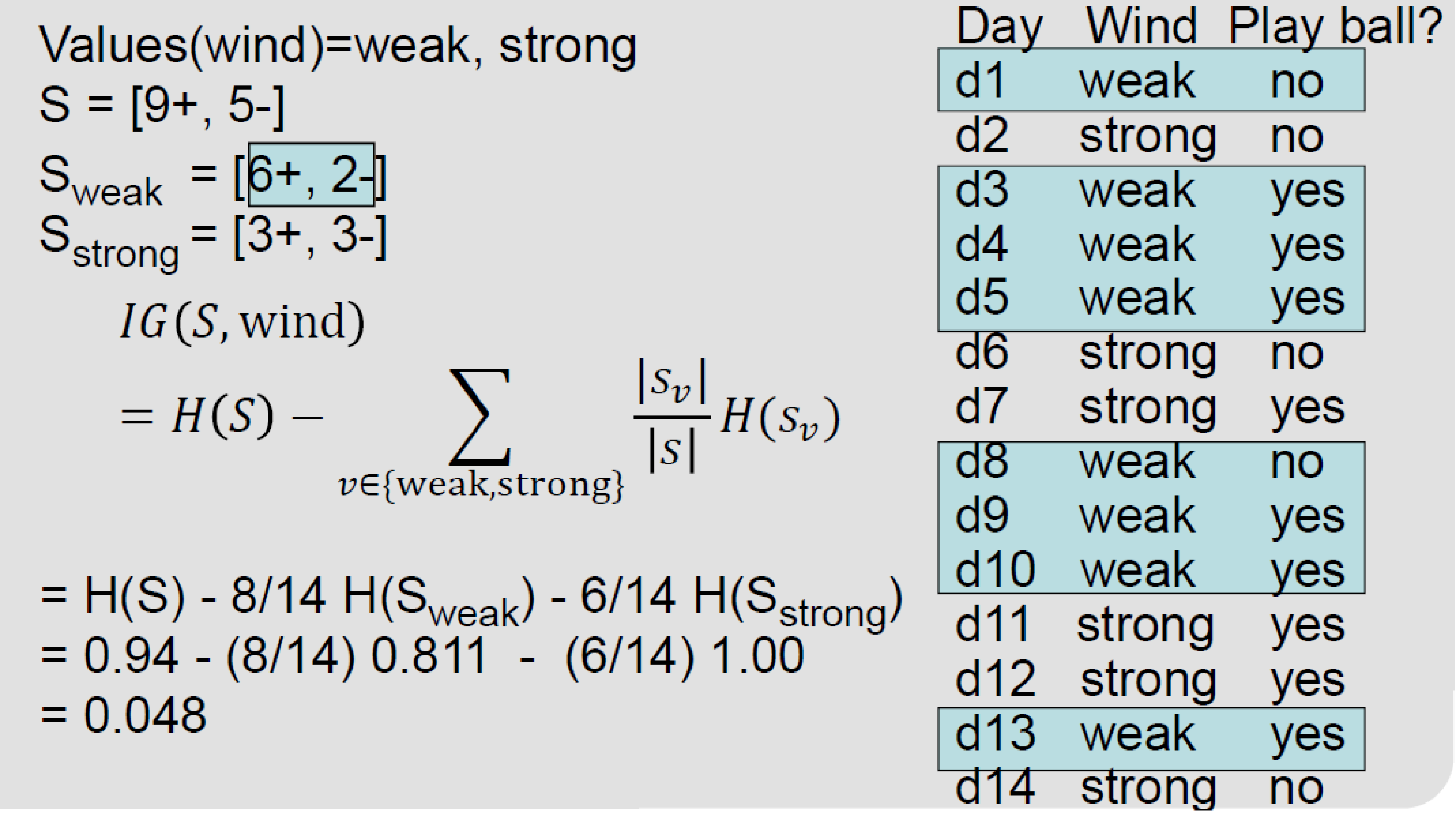
- v is a value for A (we sum over all values)
- S_v is a subset of S for which attribute A has value v

FIT5047平时班 – Module 5



- The same attributes must describe each sample
- Attributes are assumed to be categorical (for now)
- Select the attribute with the highest Information Gain → largest reduction in entropy

FIT5047平时班 – Module 5



Values(wind)=weak, strong

S = [9+, 5-]

S_{weak} = [6+, 2-]

S_{strong} = [3+, 3-]

$$IG(S, \text{wind})$$

$$= H(S) - \sum_{v \in \{\text{weak,strong}\}} \frac{|s_v|}{|s|} H(s_v)$$

= H(S) - 8/14 H(S_{weak}) - 6/14 H(S_{strong})

= 0.94 - (8/14) 0.811 - (6/14) 1.00

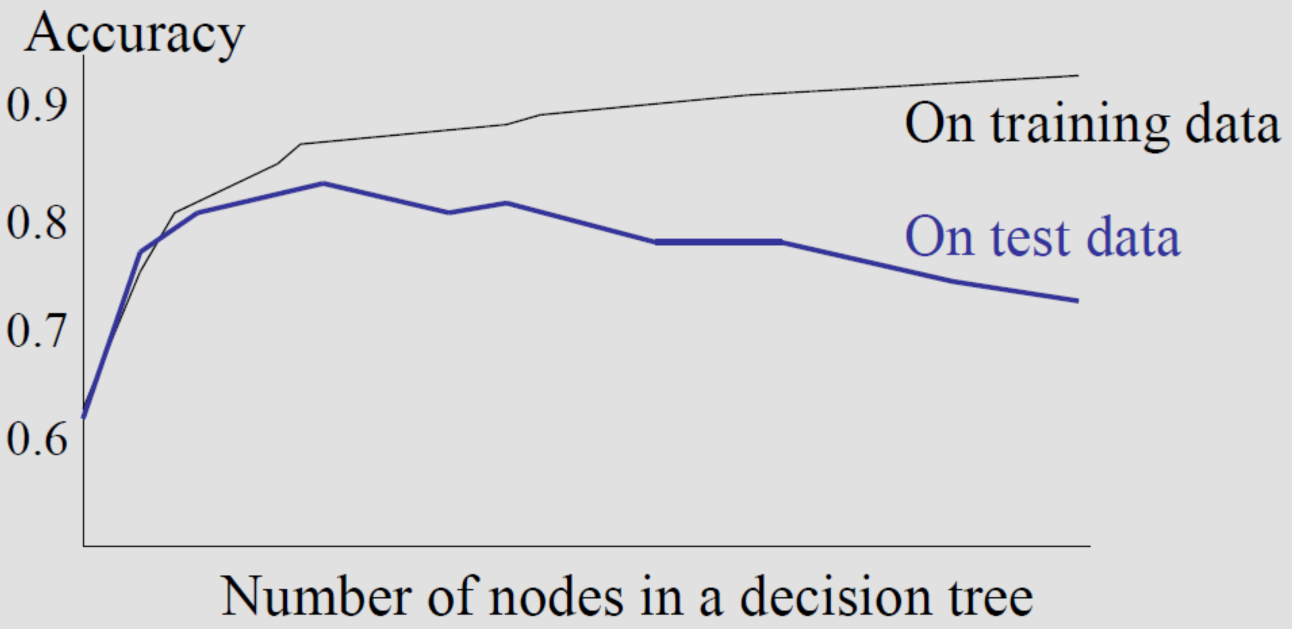
= 0.048

Day	Wind	Play ball?
d1	weak	no
d2	strong	no
d3	weak	yes
d4	weak	yes
d5	weak	yes
d6	strong	no
d7	strong	yes
d8	weak	no
d9	weak	yes
d10	weak	yes
d11	strong	yes
d12	strong	yes
d13	weak	yes
d14	strong	no

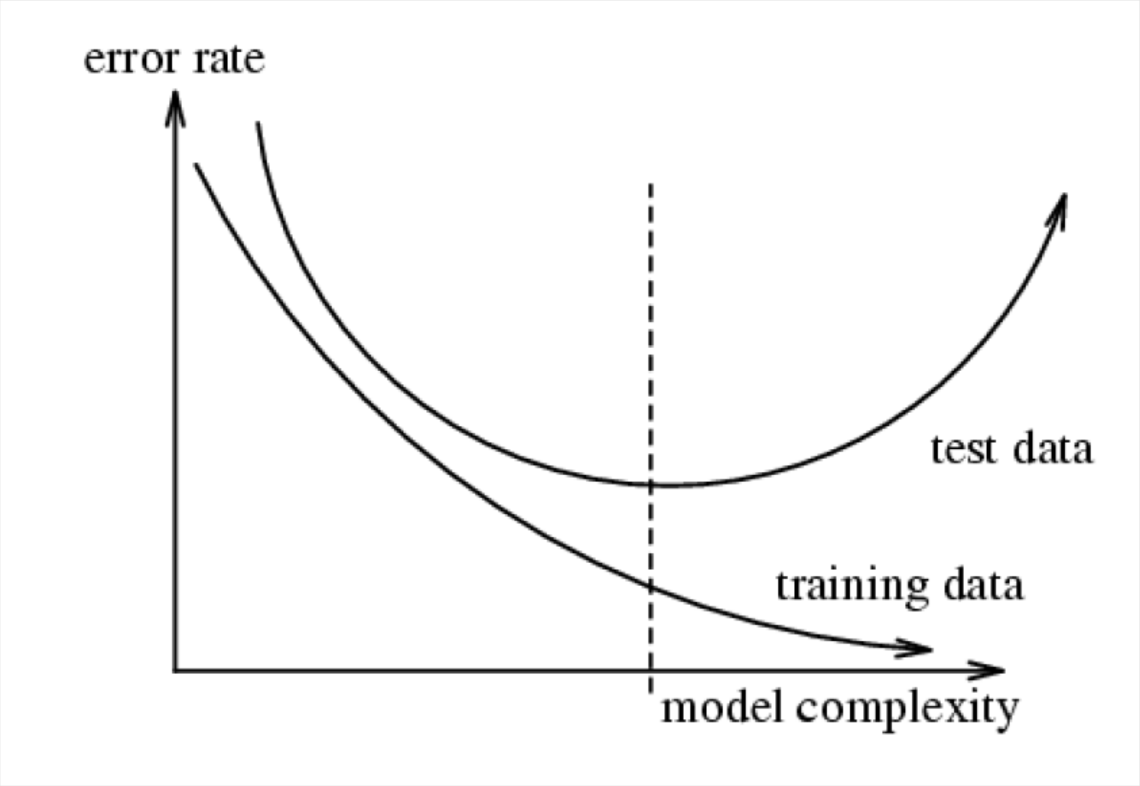
Overfitting (I)

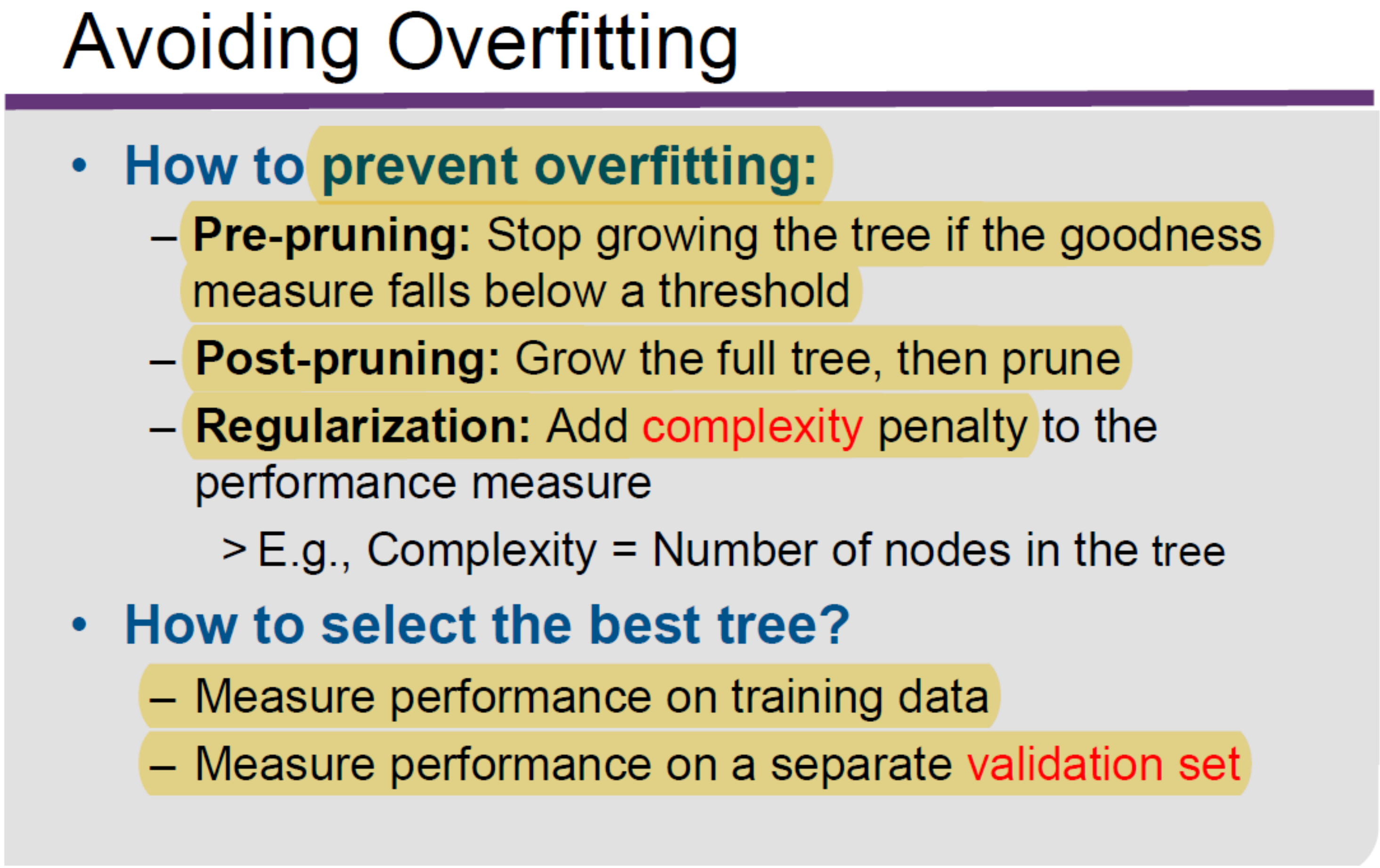
The generated tree may *overfit* the training data – try to fit noise or outliers

→ reduced accuracy for unseen samples



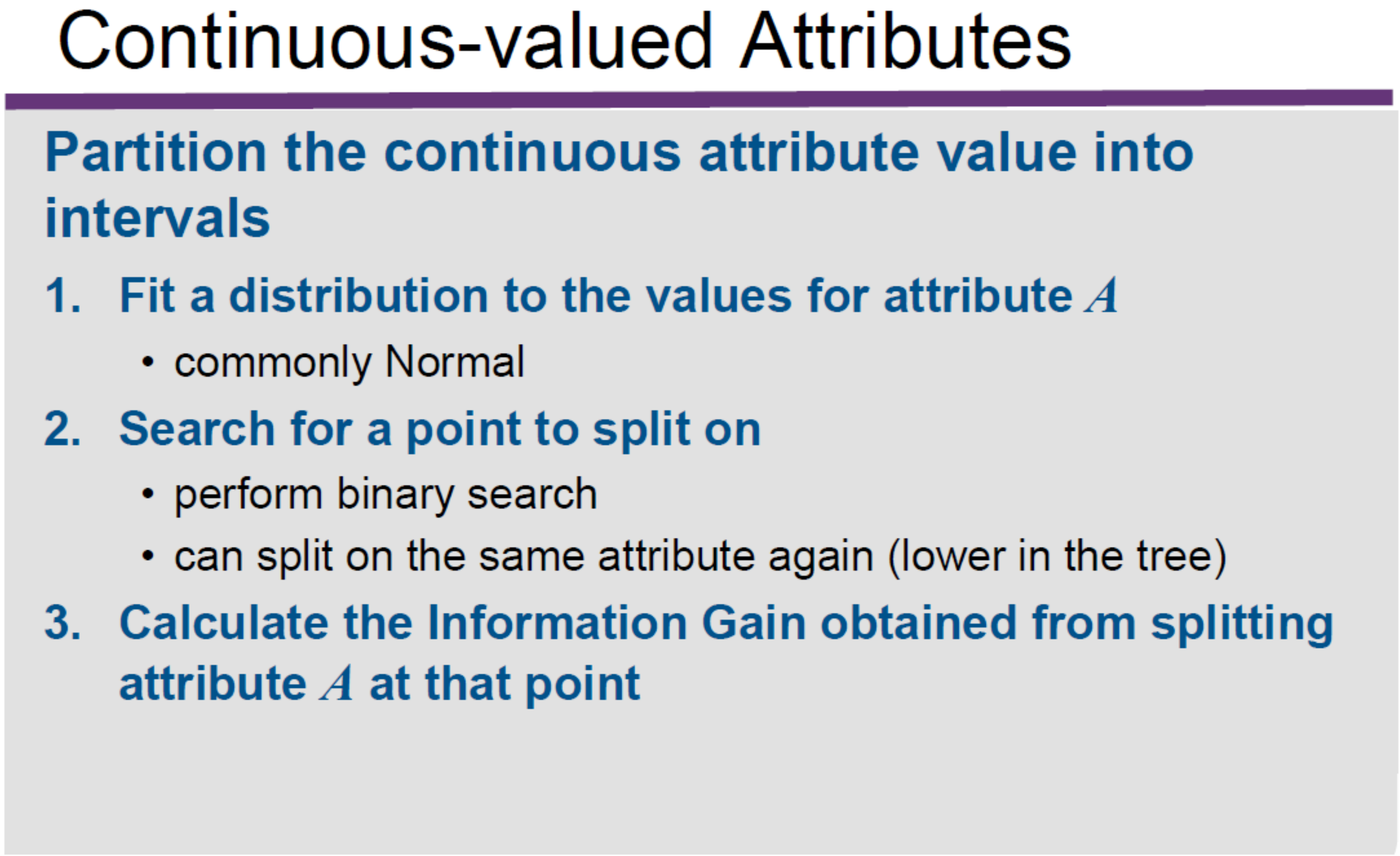
Overfitting (II)





Avoiding Overfitting

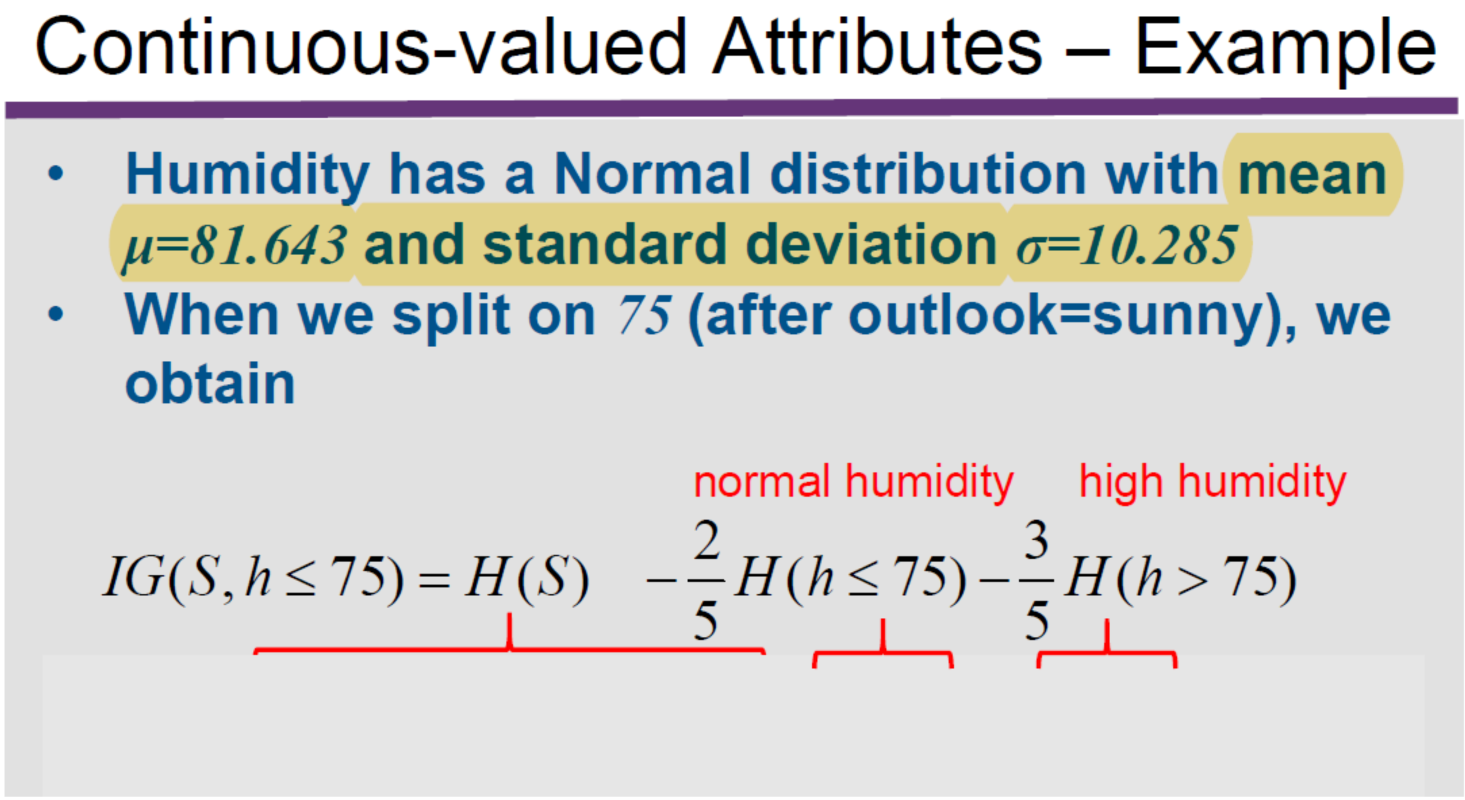
- How to prevent overfitting:
  - **Pre-pruning:** Stop growing the tree if the goodness measure falls below a threshold
  - **Post-pruning:** Grow the full tree, then prune
  - **Regularization:** Add complexity penalty to the performance measure
    - E.g., Complexity = Number of nodes in the tree
- How to select the best tree?
  - Measure performance on training data
  - Measure performance on a separate validation set



Continuous-valued Attributes

Partition the continuous attribute value into intervals

1. **Fit a distribution to the values for attribute A**
    - commonly Normal
2. **Search for a point to split on**
    - perform binary search
    - can split on the same attribute again (lower in the tree)
3. **Calculate the Information Gain obtained from splitting attribute A at that point**



Continuous-valued Attributes – Example

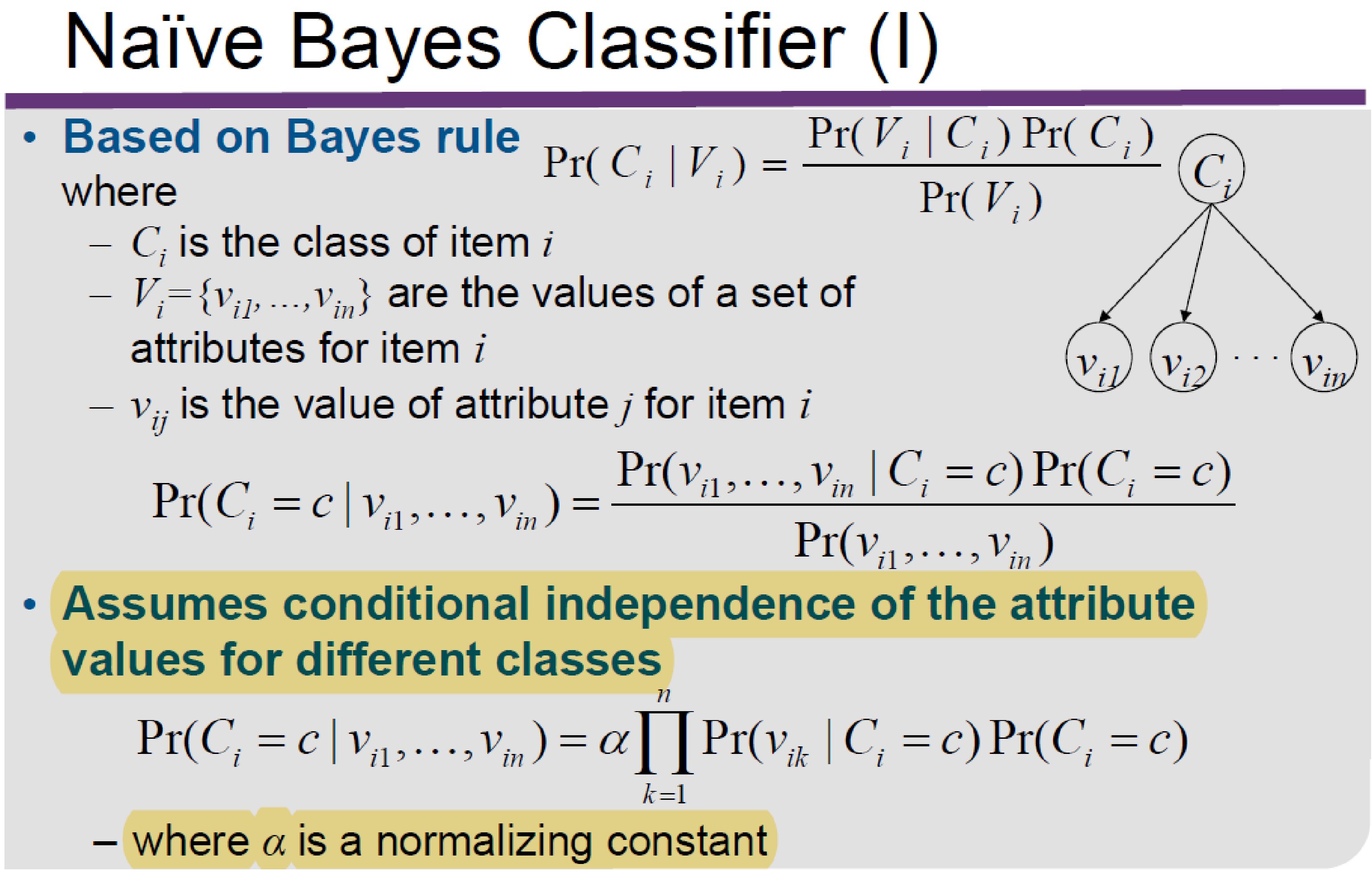
- **Humidity has a Normal distribution with mean $\mu=81.643$ and standard deviation $\sigma=10.285$**
- **When we split on 75 (after outlook=sunny), we obtain**

normal humidity high humidity

$$IG(S,h\le 75)=\underbrace{H(S)\quad -\frac{2}{5}}_{}\underbrace{H(h\le 75)}_{}-\frac{3}{5}\underbrace{H(h>75)}_{}$$

FIT5047平时班 – Module 5

Naïve Bayes Classifier



Naïve Bayes Classifier (I)

- **Based on Bayes rule** $\Pr(C_i \mid V_i) = \frac{\Pr(V_i \mid C_i)\Pr(C_i)}{\Pr(V_i)}$
  where
  - C_i is the class of item i
  - $V_i=\{v_{i1}, \ldots, v_{in}\}$ are the values of a set of attributes for item i
  - v_{ij} is the value of attribute j for item i

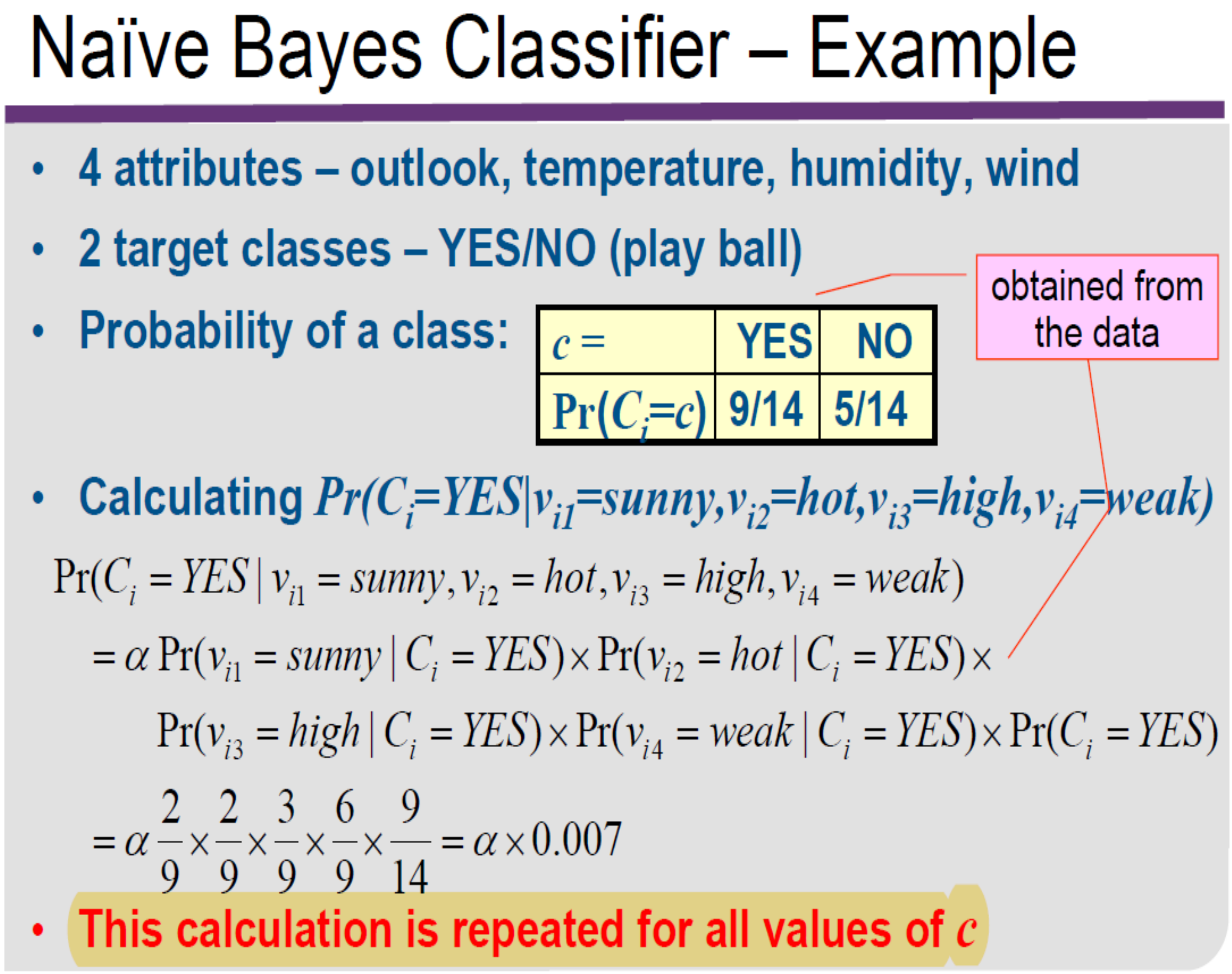
$$\Pr(C_i = c \mid v_{i1}, \ldots, v_{in}) = \frac{\Pr(v_{i1}, \ldots, v_{in} \mid C_i = c)\Pr(C_i = c)}{\Pr(v_{i1}, \ldots, v_{in})}$$

- **Assumes conditional independence of the attribute values for different classes**

$$\Pr(C_i = c \mid v_{i1}, \ldots, v_{in}) = \alpha \prod_{k=1}^{n} \Pr(v_{ik} \mid C_i = c)\Pr(C_i = c)$$

  – where α is a normalizing constant

FIT5047平时班 – Module 5



Naïve Bayes Classifier – Example

- 4 attributes – outlook, temperature, humidity, wind
- 2 target classes – YES/NO (play ball)
- Probability of a class:

$c =$	YES	NO
$\Pr(C_i=c)$	9/14	5/14

obtained from the data

- Calculating $Pr(C_i=YES|v_{i1}=sunny,v_{i2}=hot,v_{i3}=high,v_{i4}=weak)$

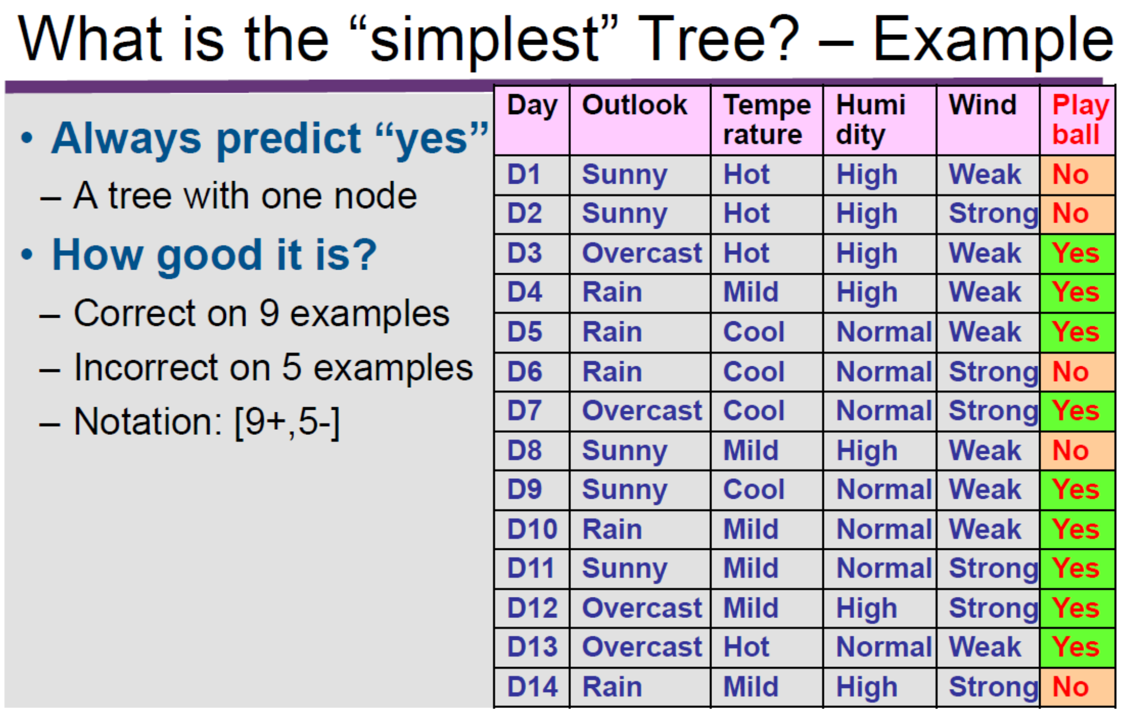
$$\Pr(C_i = YES \mid v_{i1} = sunny, v_{i2} = hot, v_{i3} = high, v_{i4} = weak)$$

$$= \alpha \Pr(v_{i1} = sunny \mid C_i = YES) \times \Pr(v_{i2} = hot \mid C_i = YES) \times$$

$$\Pr(v_{i3} = high \mid C_i = YES) \times \Pr(v_{i4} = weak \mid C_i = YES) \times \Pr(C_i = YES)$$

$$= \alpha \frac{2}{9} \times \frac{2}{9} \times \frac{3}{9} \times \frac{6}{9} \times \frac{9}{14} = \alpha \times 0.007$$

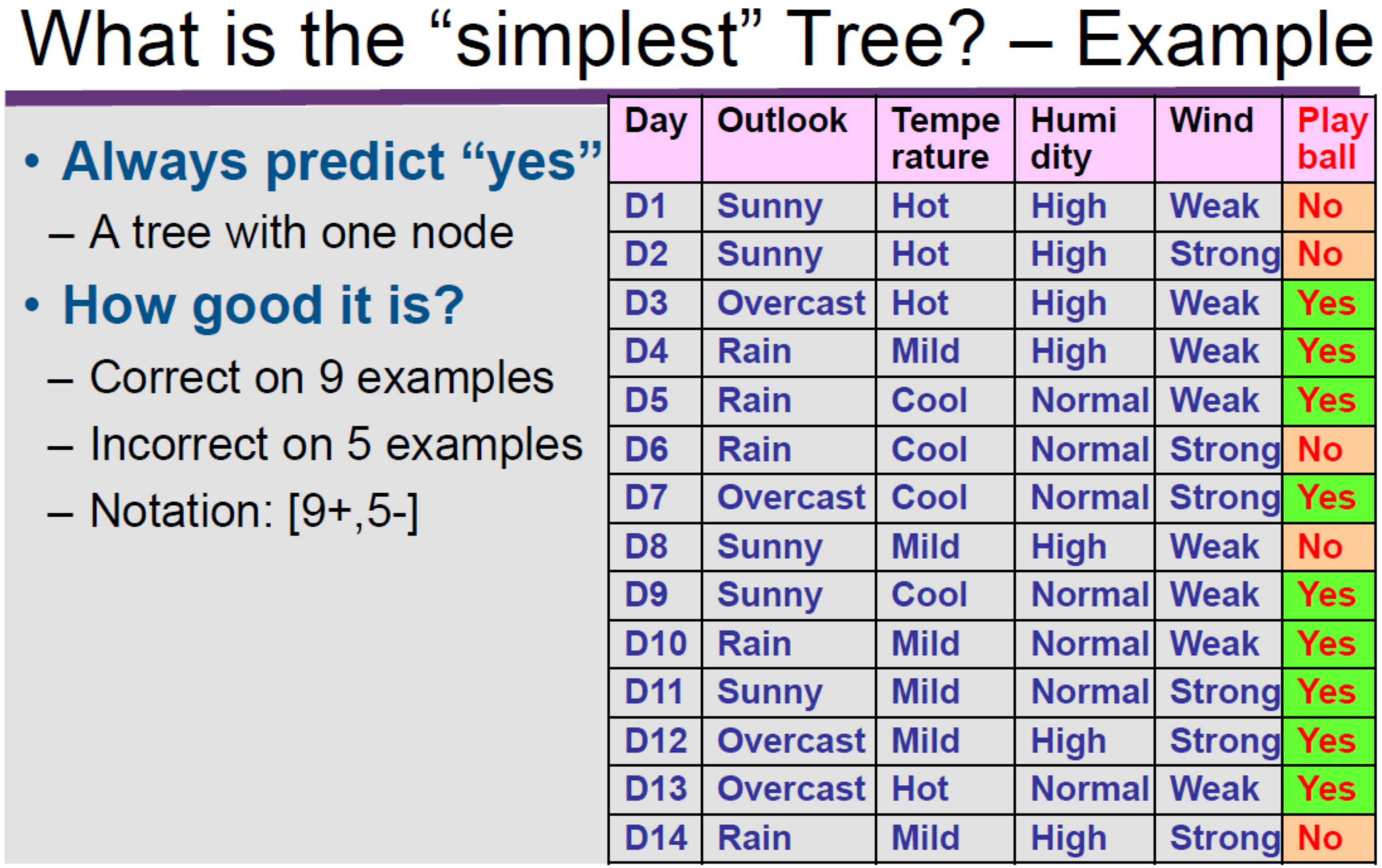
- This calculation is repeated for all values of c



What is the “simplest” Tree? – Example

- **Always predict “yes”**
  - A tree with one node
- **How good it is?**
  - Correct on 9 examples
  - Incorrect on 5 examples
  - Notation: [9+,5-]

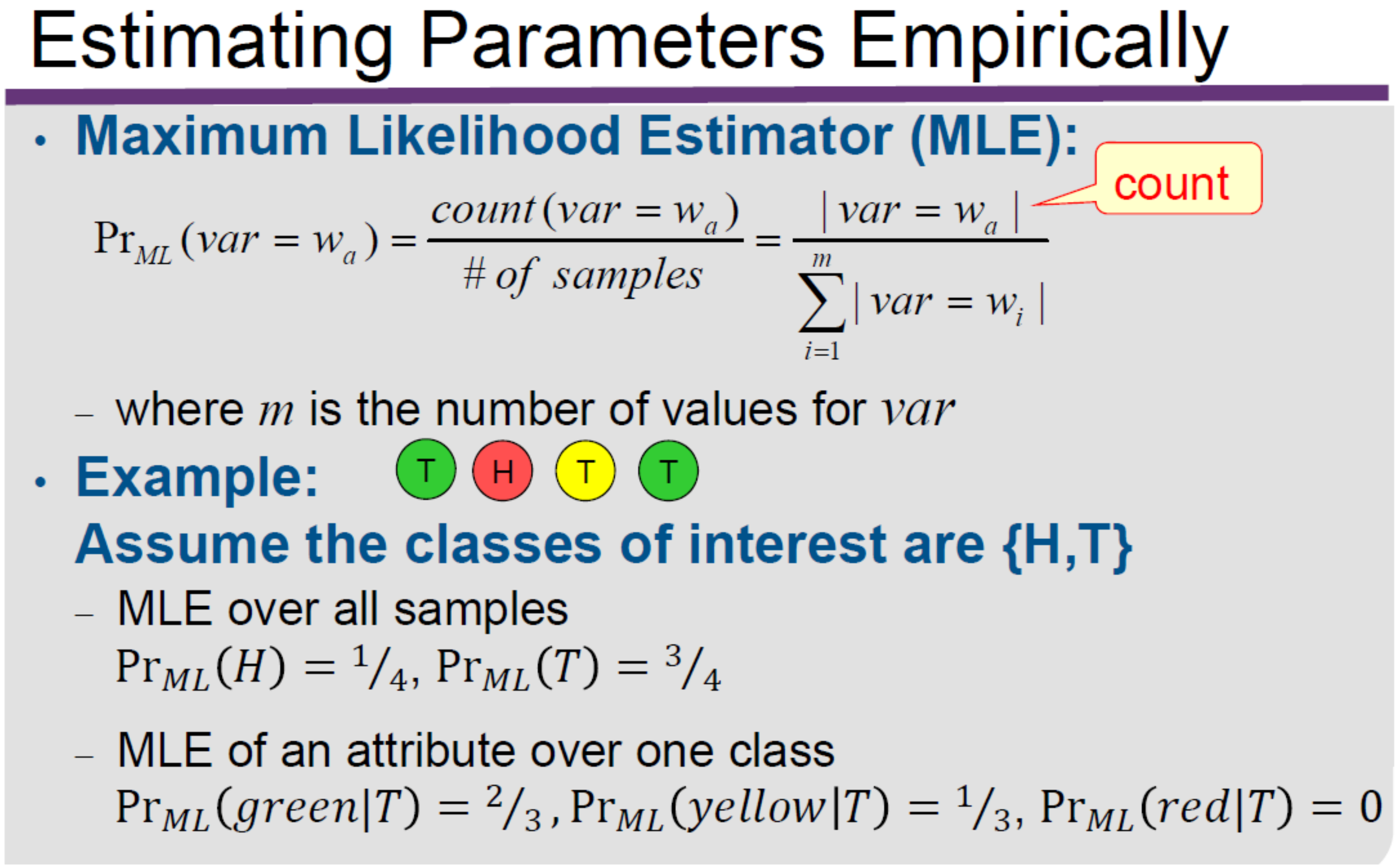
Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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



What is the “simplest” Tree? – Example

- **Always predict “yes”**
  - A tree with one node
- **How good it is?**
  - Correct on 9 examples
  - Incorrect on 5 examples
  - Notation: [9+,5-]

Day	Outlook	Temperature	Humidity	Wind	Play ball
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



Estimating Parameters Empirically

- **Maximum Likelihood Estimator (MLE):**

$$\Pr_{ML}(var = w_a) = \frac{count(var = w_a)}{\#\ of\ samples} = \frac{|\,var = w_a\,|}{\sum_{i=1}^{m} |\,var = w_i\,|}$$

count

  - where m is the number of values for var
- **Example:** T H T T

  Assume the classes of interest are {H,T}
  - MLE over all samples

    $\Pr_{ML}(H) = {}^{1}/_{4}, \Pr_{ML}(T) = {}^{3}/_{4}$
  - MLE of an attribute over one class

    $\Pr_{ML}(green|T) = {}^{2}/_{3}, \Pr_{ML}(yellow|T) = {}^{1}/_{3}, \Pr_{ML}(red|T) = 0$

The Sparse Data Problem – Smoothing

- **Not all instances are found in the data set or in a particular class**
  - If value w_a is not found in class c, w_a =0 → MLE for $\Pr(w_a|c)=0$
  - If variable var is not found in the training set, MLE for $\Pr(w_a)$ is undefined (denominator is zero)
- ***Expected Likelihood Estimator (ELE)***

$$\Pr_{EL}(var = w_a) \cong \frac{|w_a| + \varepsilon}{\sum_{i=1}^{m} \{|w_i| + \varepsilon\}} = \frac{|w_a| + \varepsilon}{\sum_{i=1}^{m} |w_i| + m\varepsilon}$$

  - where m is the number of values for var
  - If a variable is not found in the dataset, ELE is $1/m$
  - ELE is conservative
- **Use Smoothing when estimating the parameters of Naïve Bayes, i.e., $\Pr(C)$ and $\Pr(v|C)$**

Expected Likelihood Estimation – Example

T H T T

$$\Pr_{ML}(c=H) \cong \frac{|H|}{\sum_{i=1}^{2}|C_i|} = \frac{1}{\{1+3\}} = \frac{1}{4}$$

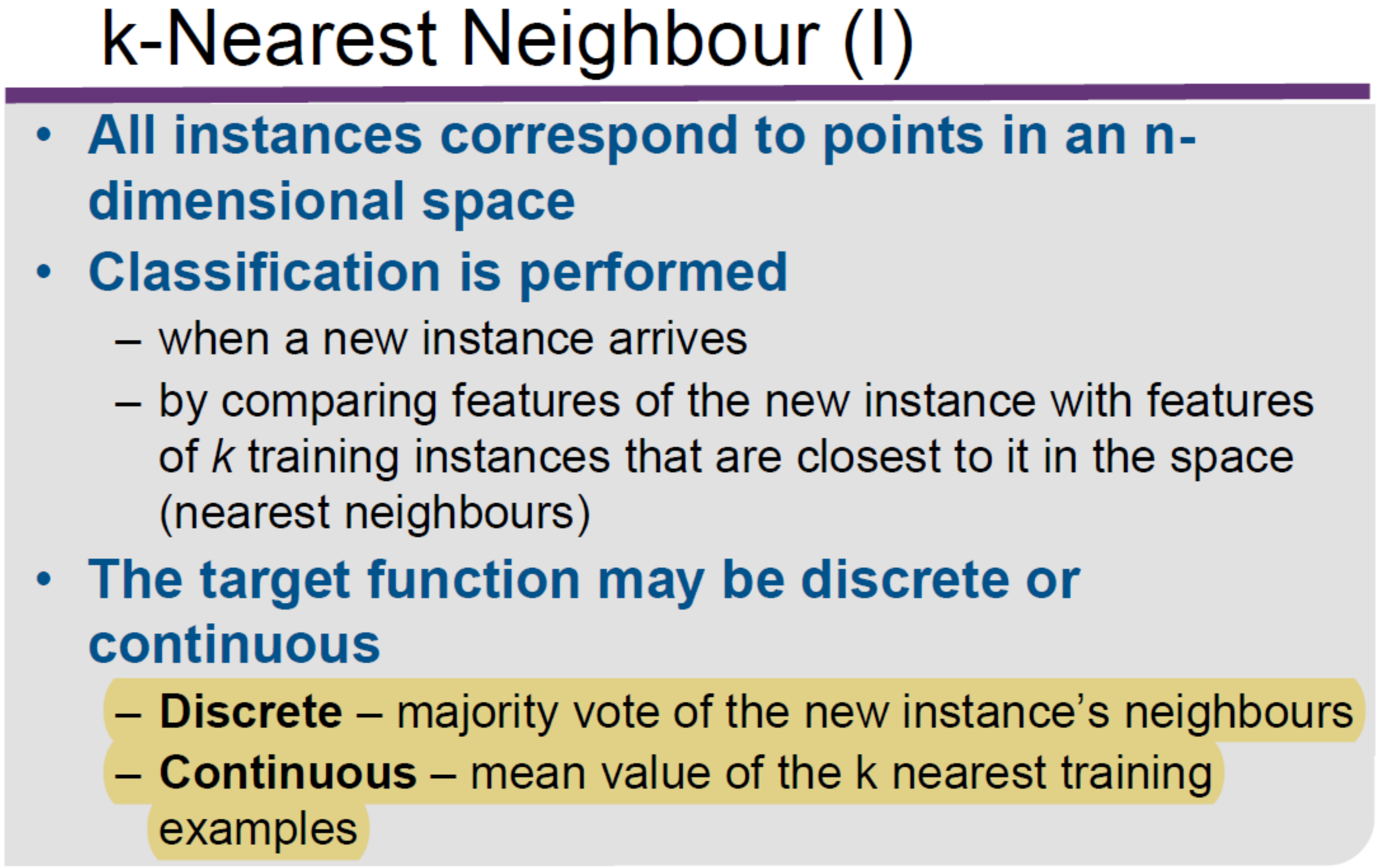
$$\Pr_{ML}(C) = \begin{array}{c} \text{H}\ \ \text{T} \\ \left\langle \frac{1}{4}, \frac{3}{4} \right\rangle \end{array}$$

$$\Pr_{EL}(c=H) \cong \frac{|H|+\varepsilon}{\sum_{i=1}^{2}|C_i|+2\varepsilon} = \frac{1+\varepsilon}{\{1+3\}+2\varepsilon} = \frac{1+\varepsilon}{4+2\varepsilon}$$

for ε=1

$$\Pr_{EL}(C) = \begin{array}{c} \text{H}\ \ \text{T} \\ \left\langle \frac{1}{3}, \frac{2}{3} \right\rangle \end{array}$$

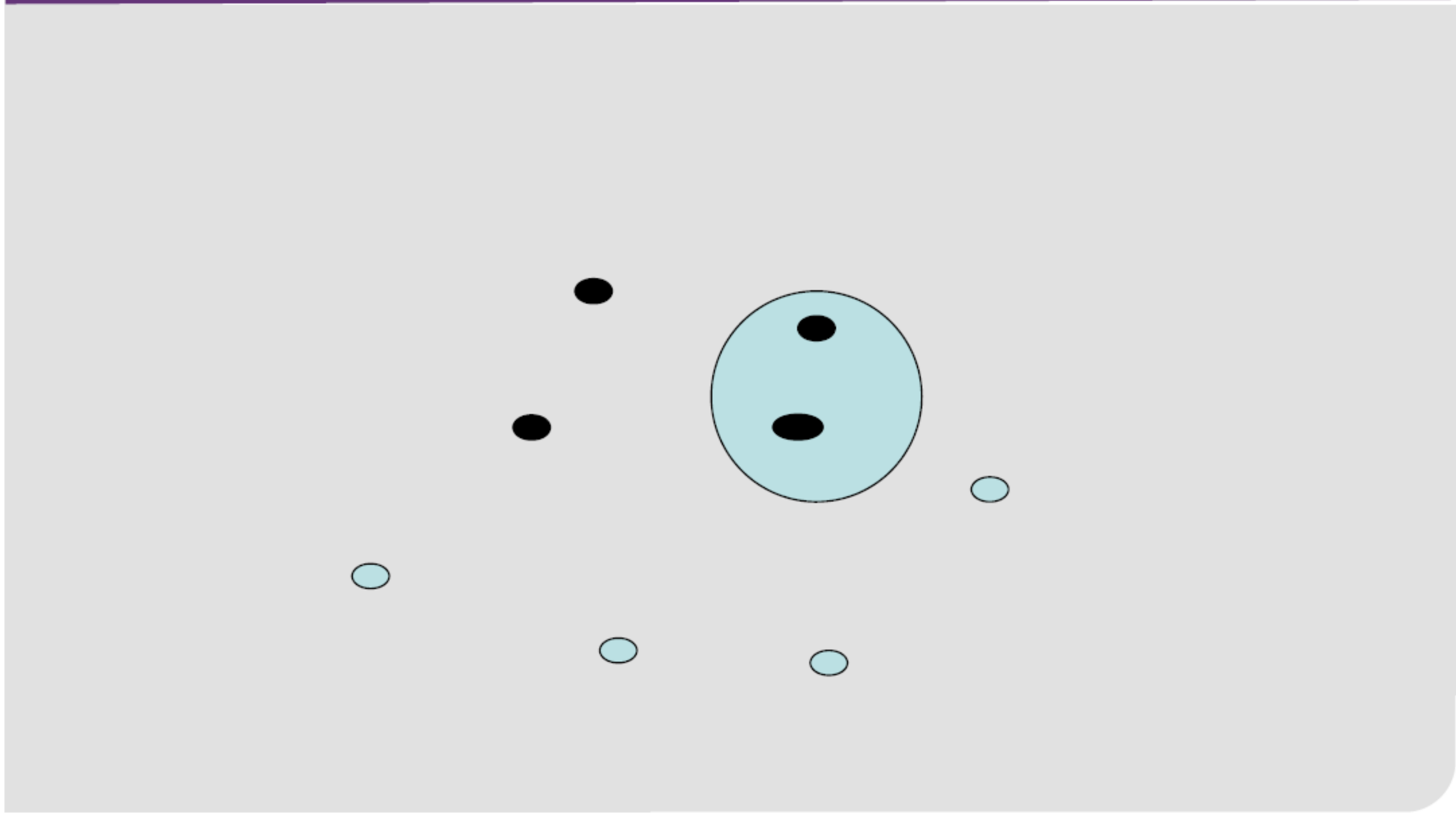
K Nearest Neighbour (k-NN)



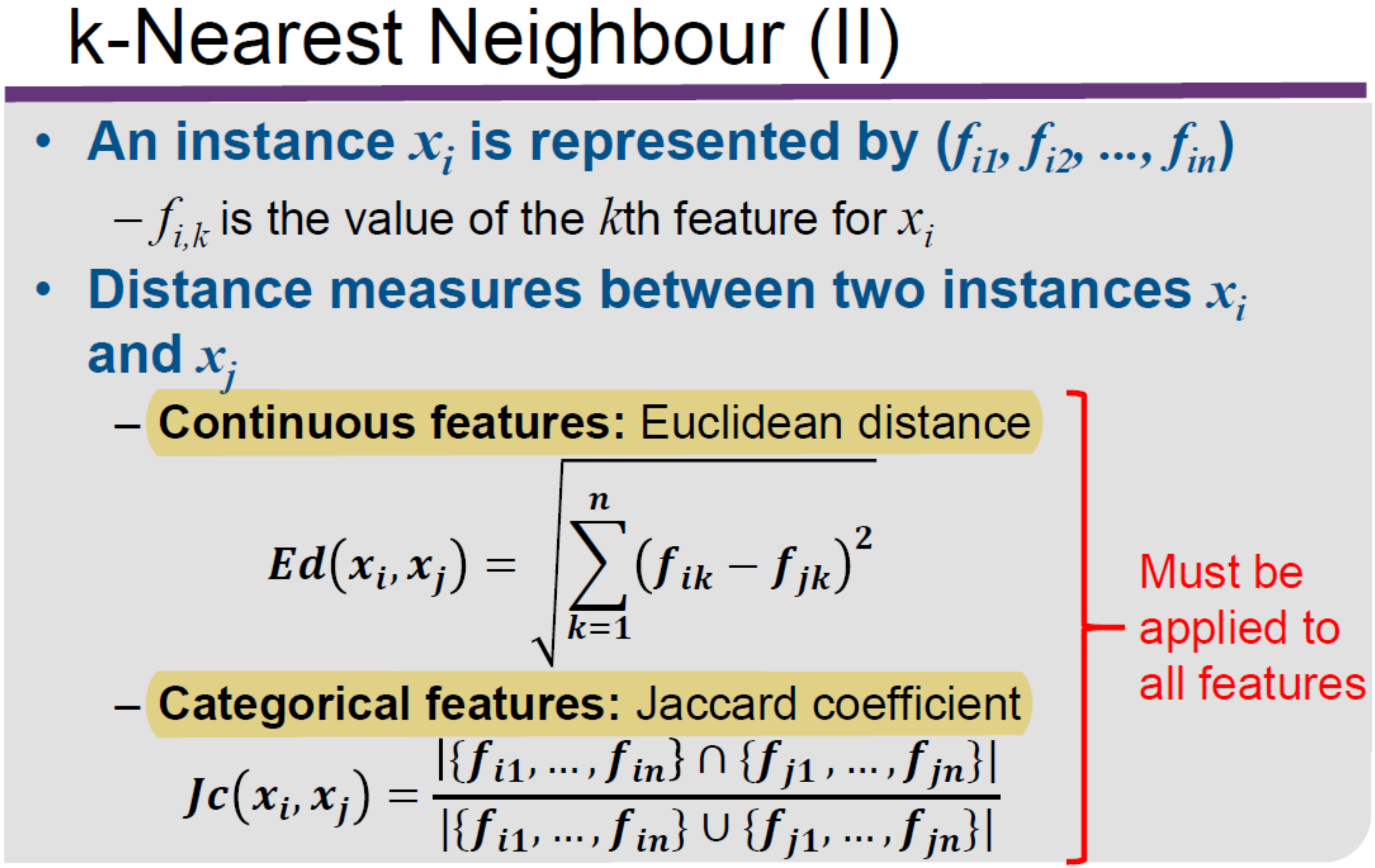
k-Nearest Neighbour (I)

- **All instances correspond to points in an n-dimensional space**
- **Classification is performed**
  - when a new instance arrives
  - by comparing features of the new instance with features of *k* training instances that are closest to it in the space (nearest neighbours)
- **The target function may be discrete or continuous**
  - **Discrete** – majority vote of the new instance’s neighbours
  - **Continuous** – mean value of the k nearest training examples

1-Nearest Neighbour



FIT5047平时班 – Module 5



k-Nearest Neighbour (II)

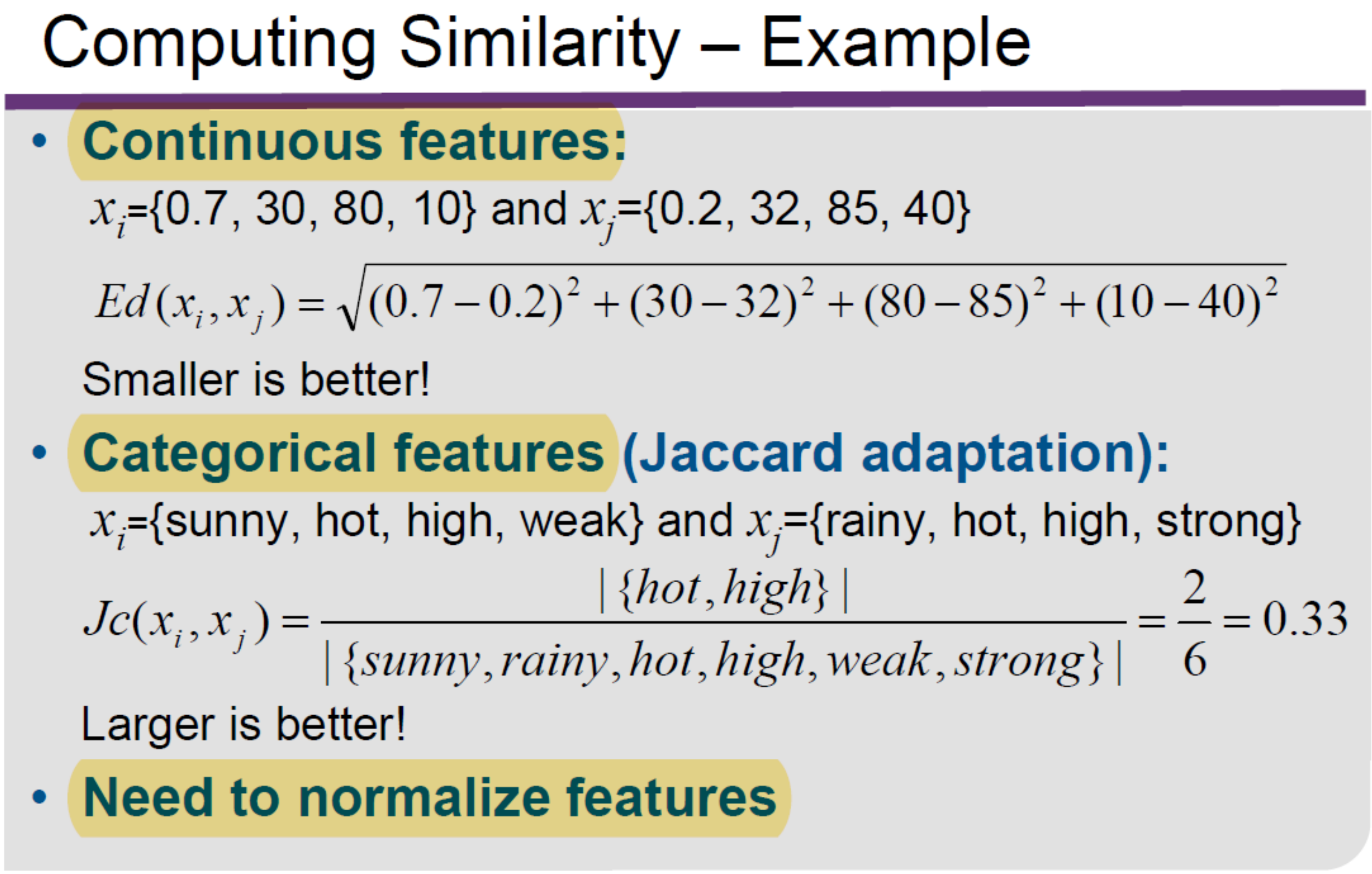
- **An instance x_i is represented by ($f_{i1}, f_{i2}, \ldots, f_{in}$)**
  - $f_{i,k}$ is the value of the kth feature for x_i
- **Distance measures between two instances x_i and x_j**
  - **Continuous features:** Euclidean distance

$$Ed(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (f_{ik} - f_{jk})^2}$$

  - **Categorical features:** Jaccard coefficient

$$Jc(x_i, x_j) = \frac{|\{f_{i1}, \ldots, f_{in}\} \cap \{f_{j1}, \ldots, f_{jn}\}|}{|\{f_{i1}, \ldots, f_{in}\} \cup \{f_{j1}, \ldots, f_{jn}\}|}$$

Must be applied to all features



Computing Similarity – Example

- **Continuous features:**

  x_i={0.7, 30, 80, 10} and x_j={0.2, 32, 85, 40}

  $$Ed(x_i, x_j) = \sqrt{(0.7-0.2)^2 + (30-32)^2 + (80-85)^2 + (10-40)^2}$$

  Smaller is better!

- **Categorical features (Jaccard adaptation):**

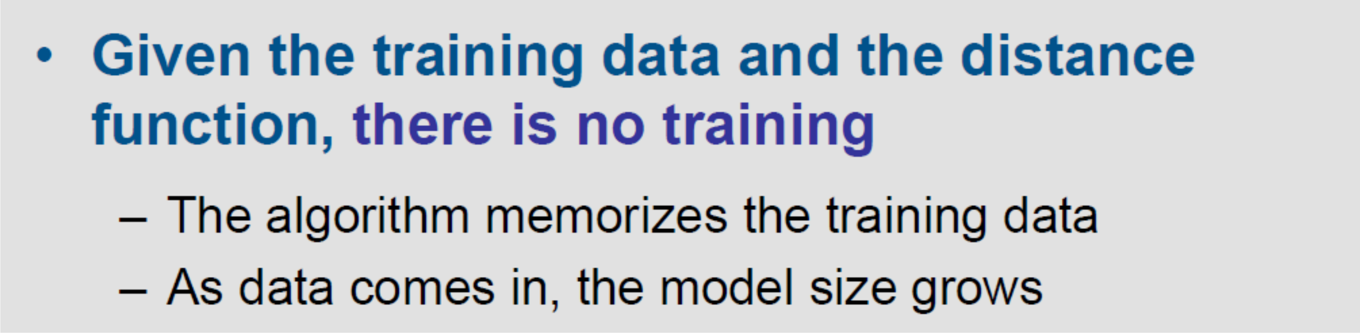
  x_i={sunny, hot, high, weak} and x_j={rainy, hot, high, strong}

  $$Jc(x_i, x_j) = \frac{|\{hot, high\}|}{|\{sunny, rainy, hot, high, weak, strong\}|} = \frac{2}{6} = 0.33$$

  Larger is better!

- **Need to normalize features**

FIT5047平时班 – Module 5



- **Given the training data and the distance function, there is no training**
  - The algorithm memorizes the training data
  - As data comes in, the model size grows

Regression

Error Function

- **Given a training set $\{(\boldsymbol{x}_1,t_1),\ldots,(\boldsymbol{x}_m,t_m)\}$**
  - assume $\boldsymbol{x}_i \in R^n, t_i \in R$
- **Linear regression model:** $t = \overset{\text{vector}}{\boldsymbol{w}}\cdot\boldsymbol{x} + \overset{\text{offset}}{w_0}$
  - we want to learn the parameters $\boldsymbol{w} \in R^n, w_0 \in R$
- **Error: Square of the difference between the true and predicted target value for x_i**

$$E(w) = \sum_{i=1}^{m} (\underbrace{t_i}_{\text{truth}} - \underbrace{(\boldsymbol{w}\cdot\boldsymbol{x}_i + w_0)}_{\text{prediction}})^2$$

Linear Regression – Example (I)

x_1,t_1 x_2,t_2 x_3,t_3

- **Training data: {(1,3),(2.1,0.5),(-5,6.2)}**
- **Linear regression model:** $t = w_1 x + w_0$
- **Error function:**

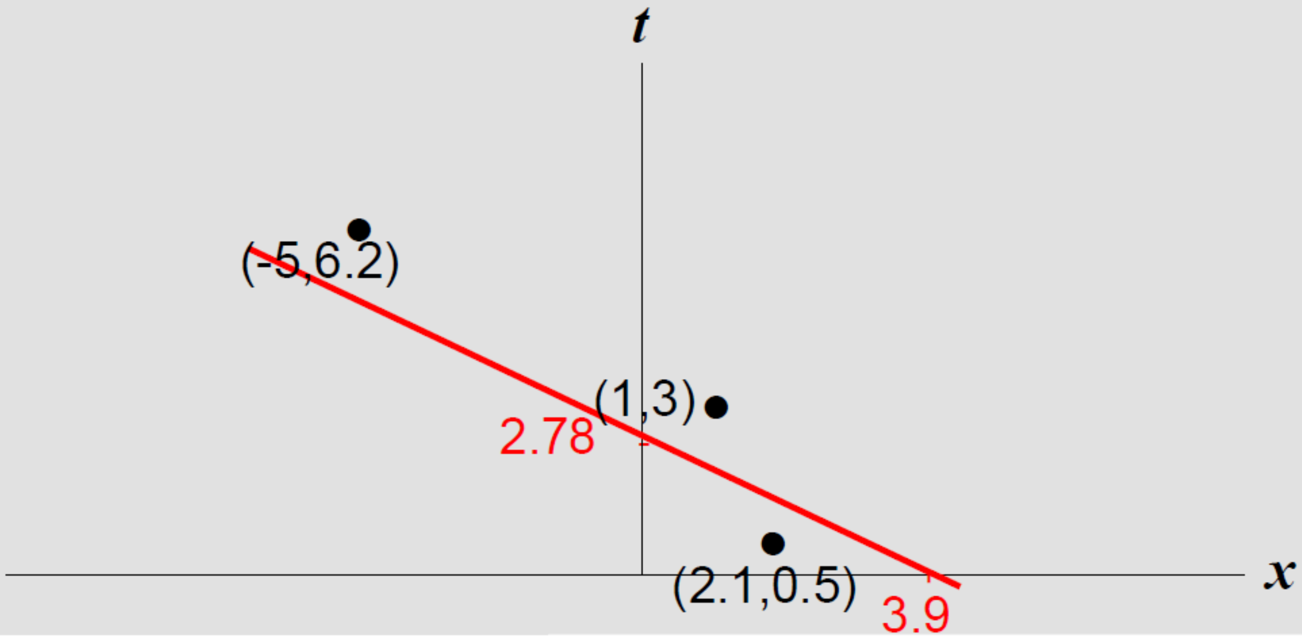
$$E(w) = (3 - 1w_1 - w_0)^2 + (0.5 - 2.1w_1 - w_0)^2 + (6.2 - (-5)w_1 - w_0)^2$$

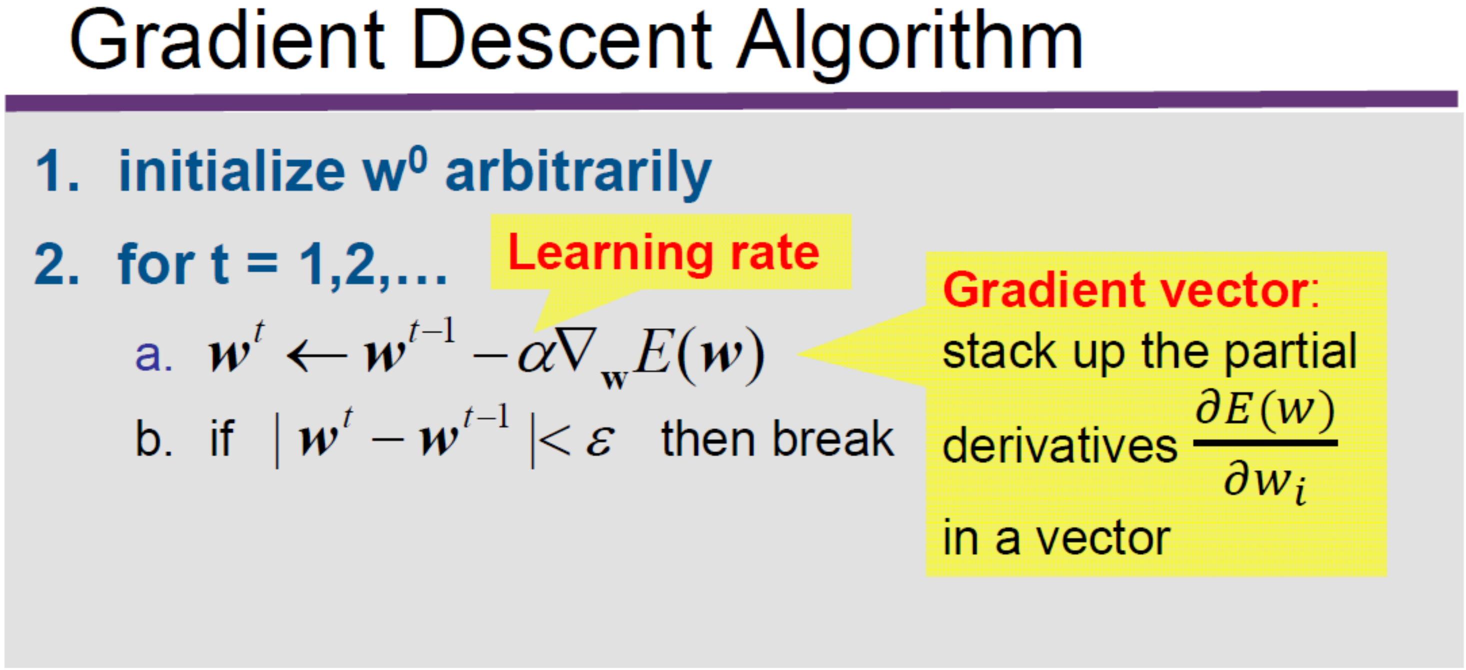
$$w_0 = \frac{(3+0.5+6.2) - w_1(1+2.1-5)}{3} = \frac{9.7 + w_1 1.9}{3} = 2.78$$

$$w_1 = \frac{3(1\times3 + 2.1\times0.5 + (-5)\times6.2) - (1+2.1-5)(3+0.5+6.2)}{3(1^2 + 2.1^2 + (-5)^2) - (1+2.1-5)^2} = -0.712$$

Linear Regression – Example (II)

- **Training data:** {(1,3),(2.1,0.5),(-5,6.2)}
- **Function:** $t = -0.712x + 2.78$





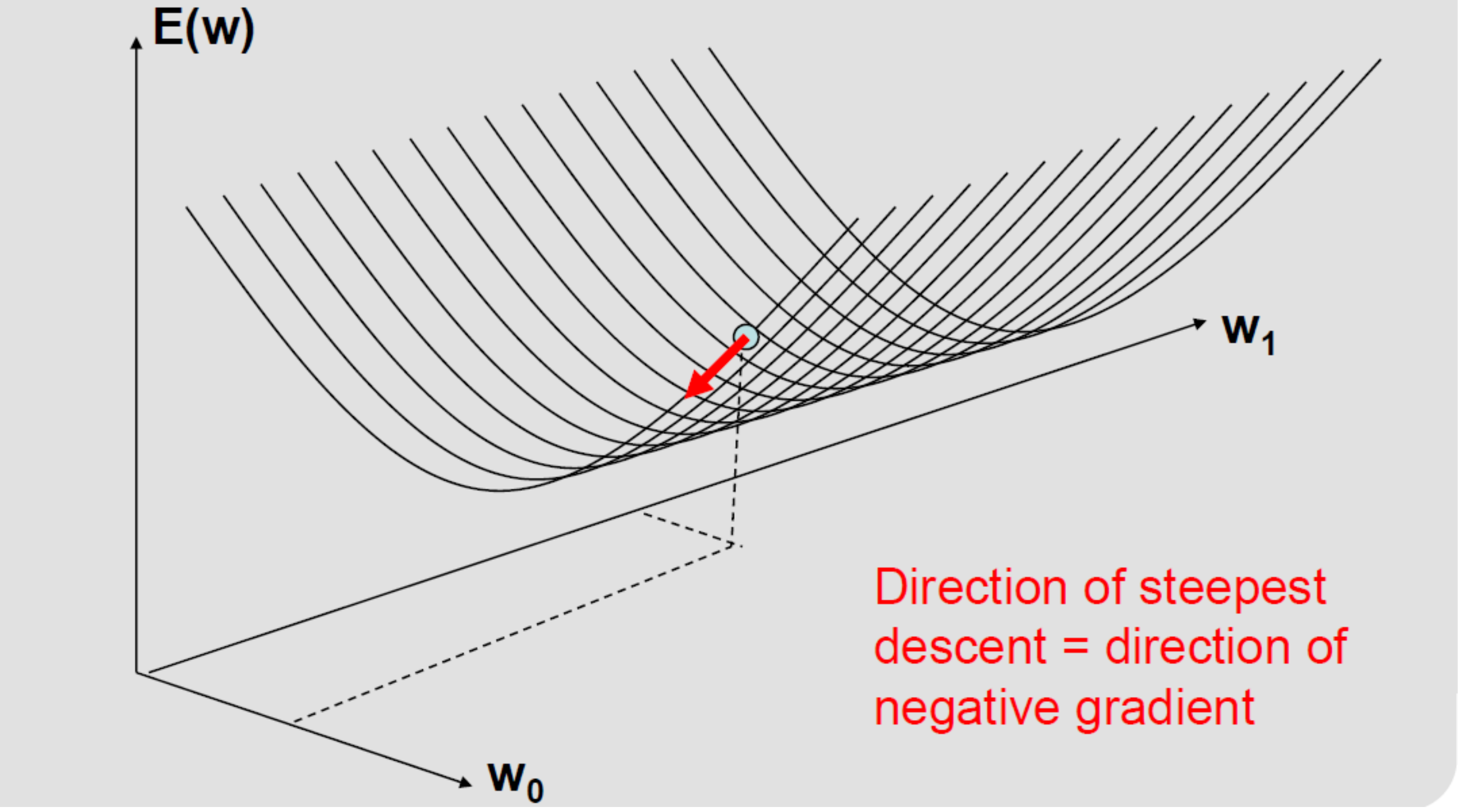
Gradient Descent Algorithm

```
1. initialize $w^0$ arbitrarily
2. for t = 1,2,…
    a. $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha \nabla_{\mathbf{w}} E(\mathbf{w})$
    b. if $|\mathbf{w}^t - \mathbf{w}^{t-1}| < \varepsilon$ then break
```

Learning rate

Gradient vector: stack up the partial derivatives $\frac{\partial E(w)}{\partial w_i}$ in a vector

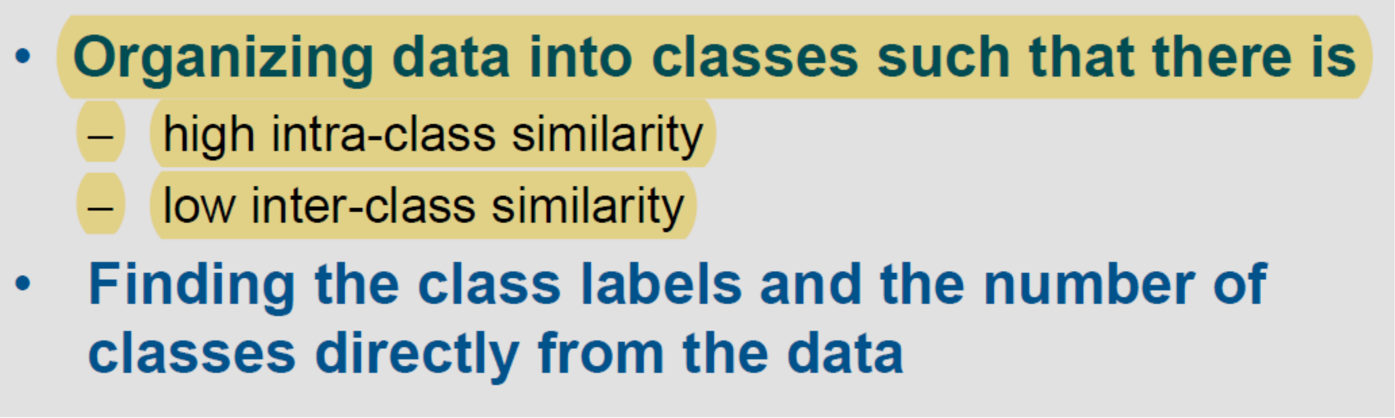
Illustration of Gradient Descent (I)



Direction of steepest descent = direction of negative gradient

Unsupervised Machine Learning – Clustering

FIT5047平时班 – Module 5

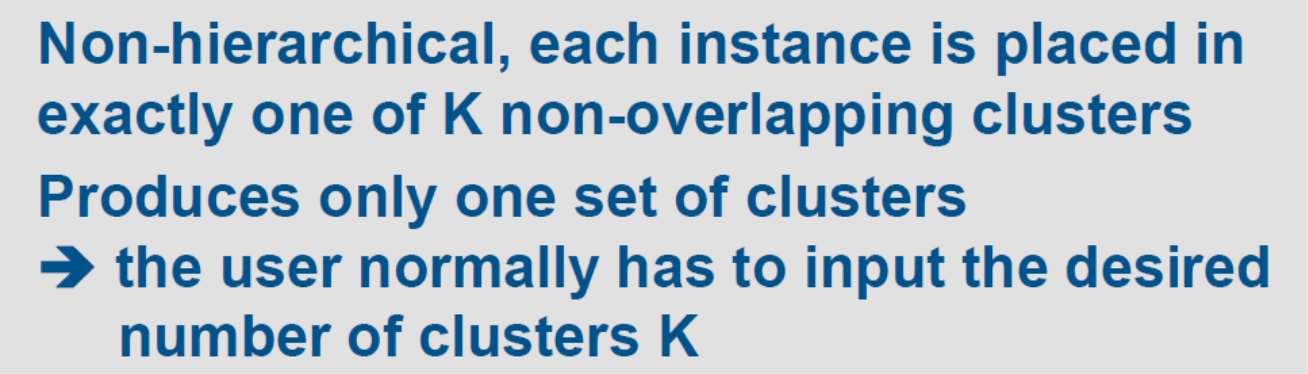


- Organizing data into classes such that there is
  - high intra-class similarity
  - low inter-class similarity
- Finding the class labels and the number of classes directly from the data

FIT5047平时班 – Module 5

K-means Algorithm

FIT5047平时班 – Module 5



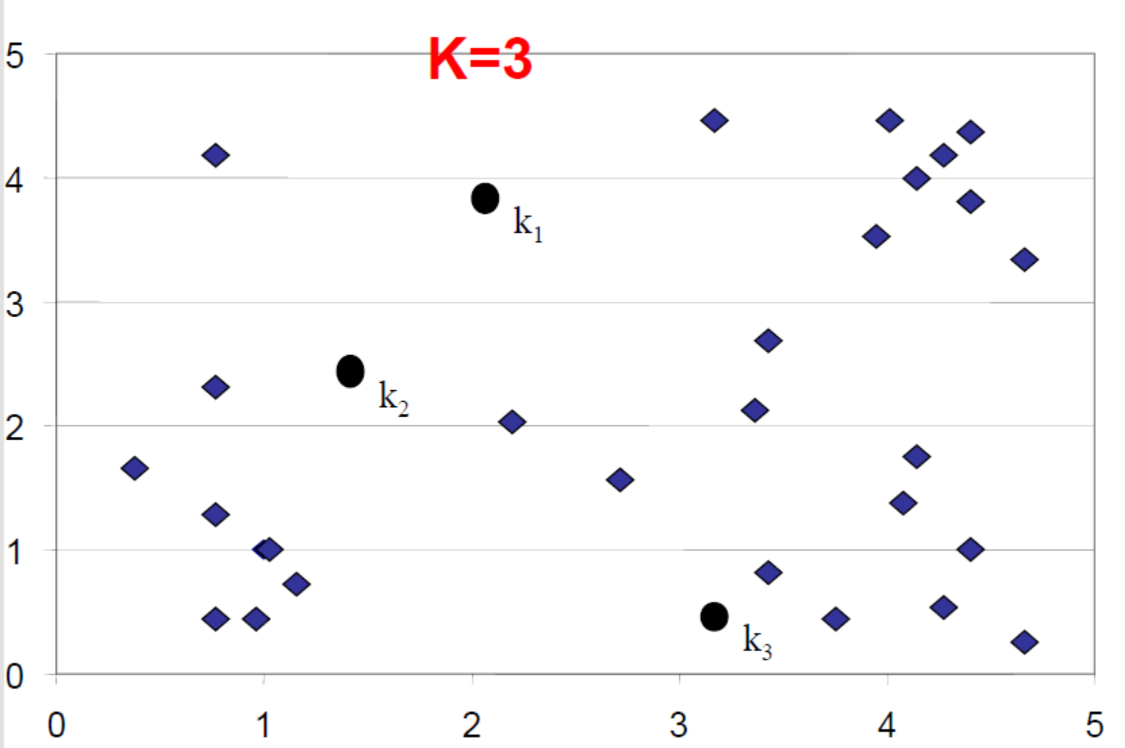
Non-hierarchical, each instance is placed in exactly one of K non-overlapping clusters

Produces only one set of clusters

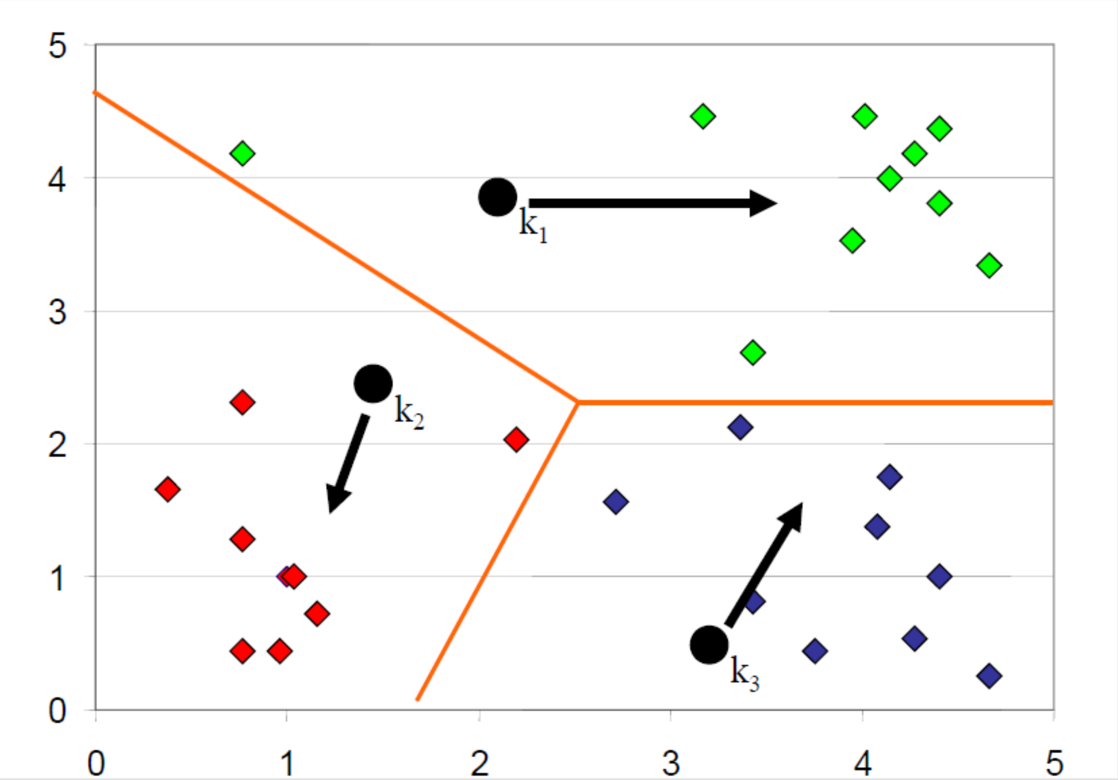
➔ the user normally has to input the desired number of clusters K

K-means Clustering: Steps 1 and 2

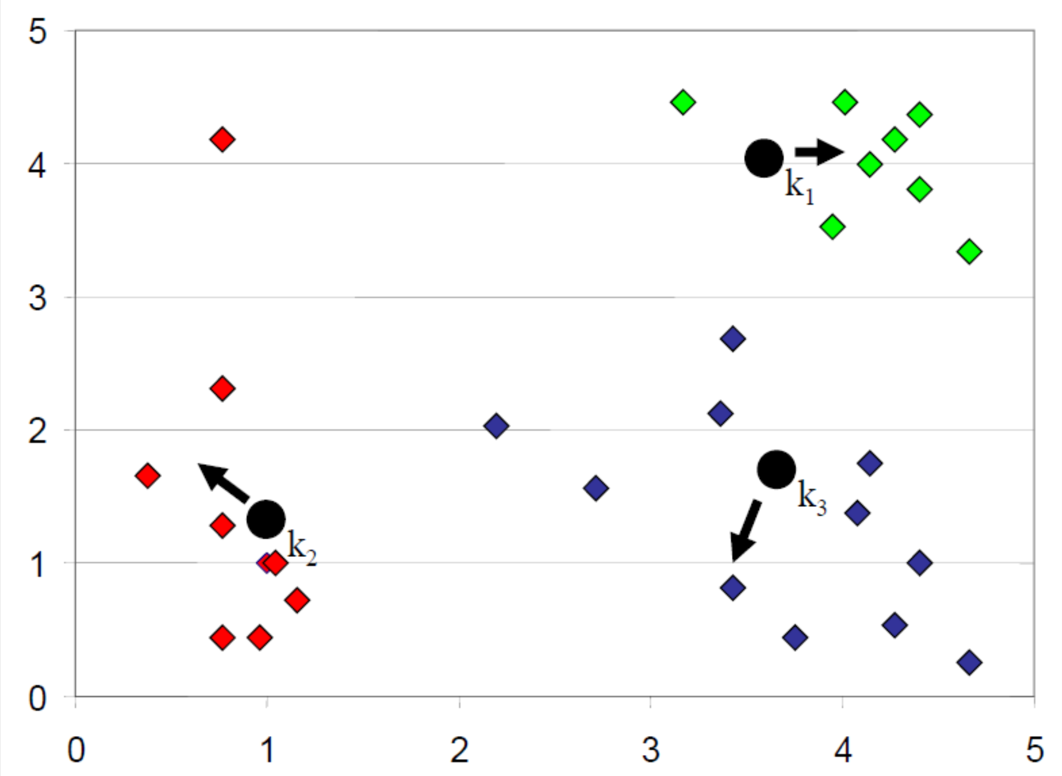
Distance Metric: Euclidean Distance



K-means Clustering: Steps 3 and 4

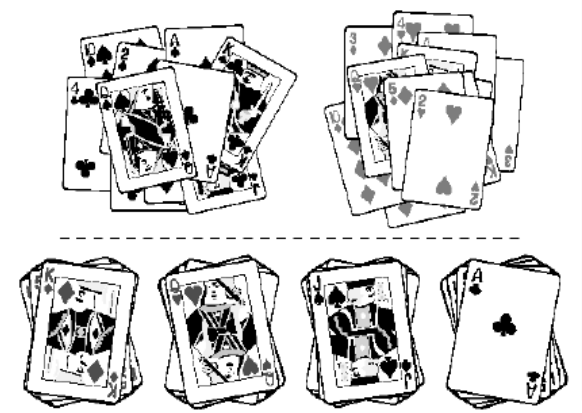


K-means Clustering: Steps 3 and 4



FIT5047平时班 – Module 5

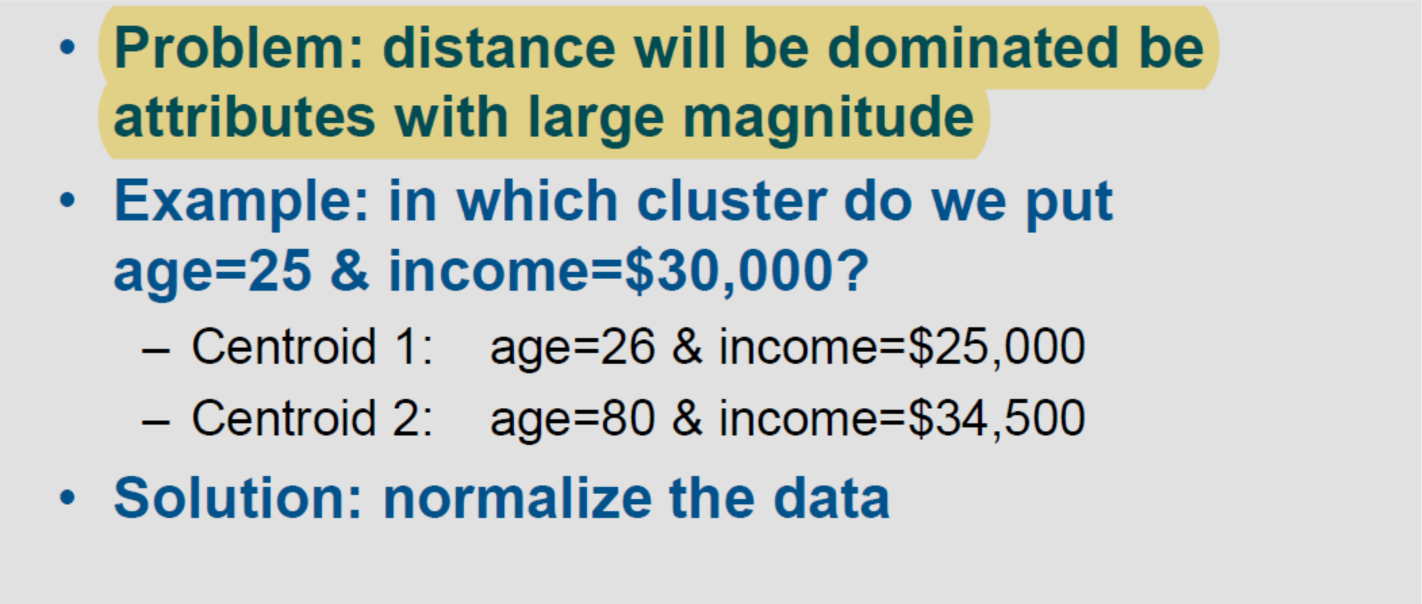
- **Clusters describe underlying structure in data**
  - – but structures in data can exist at different levels
- **In many cases, there is no a priori reason to select a particular value for k**
  - – Should consider several k-s, but different values of k can lead to different clusterings



k = 2

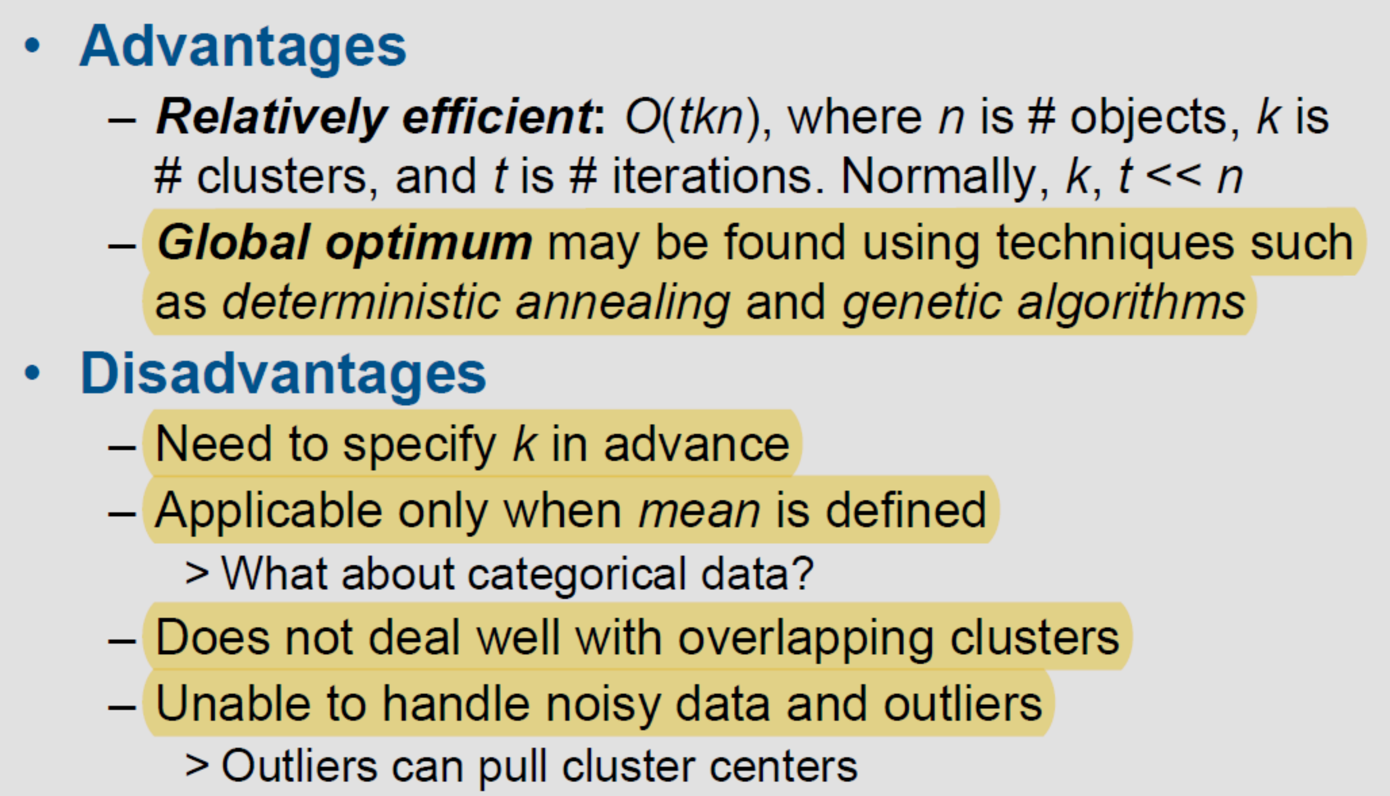
k = 4

FIT5047平时班 – Module 5



- Problem: distance will be dominated be attributes with large magnitude
- Example: in which cluster do we put age=25 & income=$30,000?
  - Centroid 1: age=26 & income=$25,000
  - Centroid 2: age=80 & income=$34,500
- Solution: normalize the data

FIT5047平时班 – Module 5



- **Advantages**
  - ***Relatively efficient***: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$
  - ***Global optimum*** may be found using techniques such as *deterministic annealing* and *genetic algorithms*
- **Disadvantages**
  - Need to specify k in advance
  - Applicable only when *mean* is defined
    - > What about categorical data?
  - Does not deal well with overlapping clusters
  - Unable to handle noisy data and outliers
    - > Outliers can pull cluster centers