Machine Learning

K-NEAREST NEIGHBORS
DECISION TREE

Nguyen Quoc Thai

CONTENT

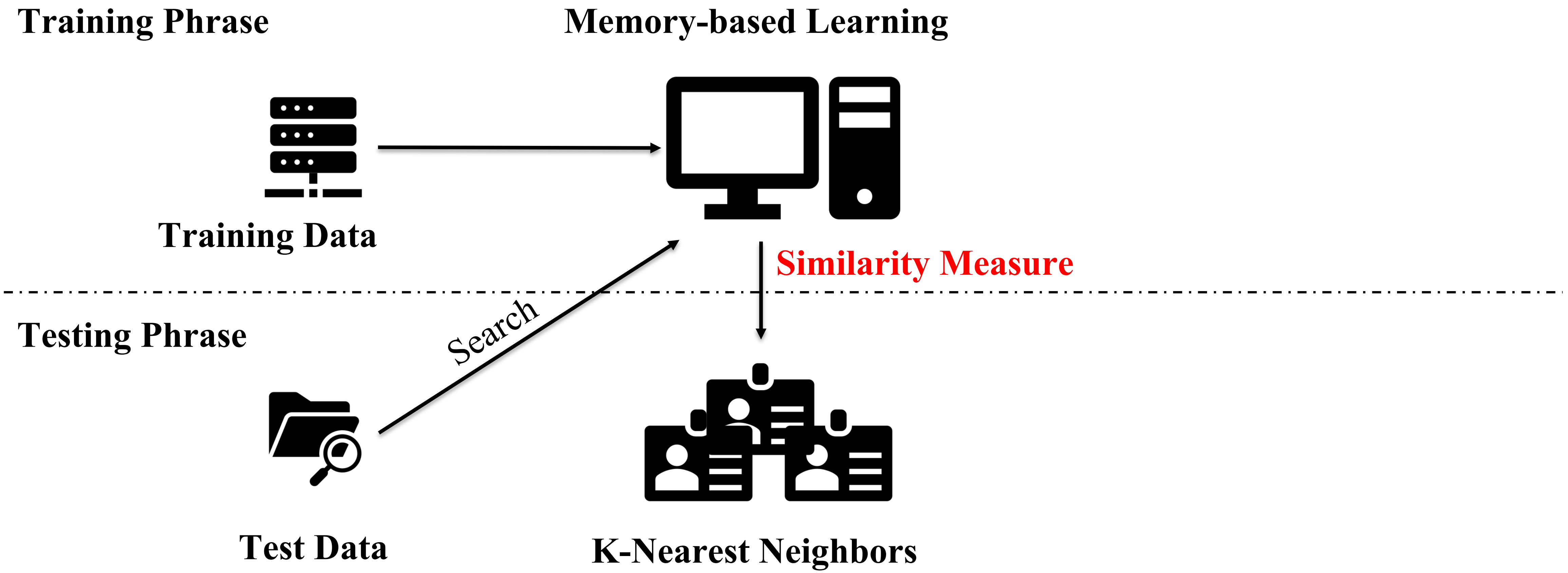
1 – K Nearest Neighbors

What is the KNN Algorithm?

- KNN is one of the simplest supervised machine learning algorithms
- Lazy Learning:
  - Does not "LEARN" until the test example is given
  - A new data is predicted based on K-Nearest Neighbors from the training data

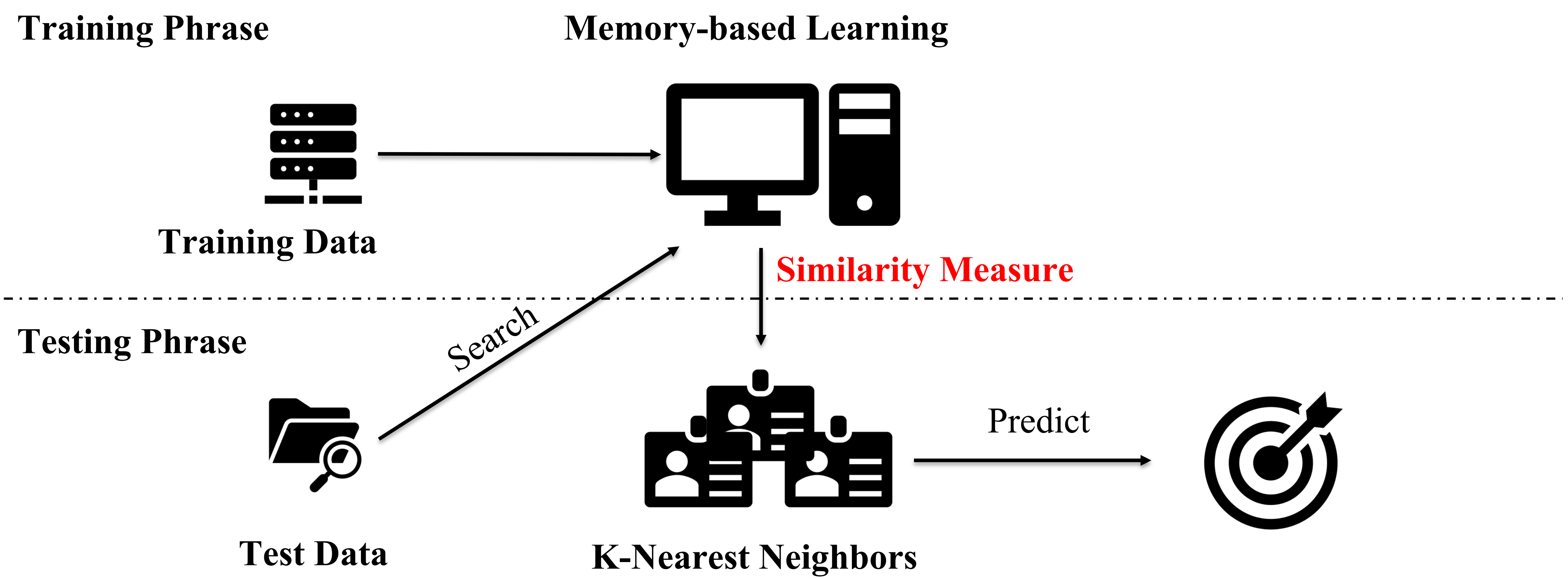
1 – K Nearest Neighbors

What is the KNN Algorithm?



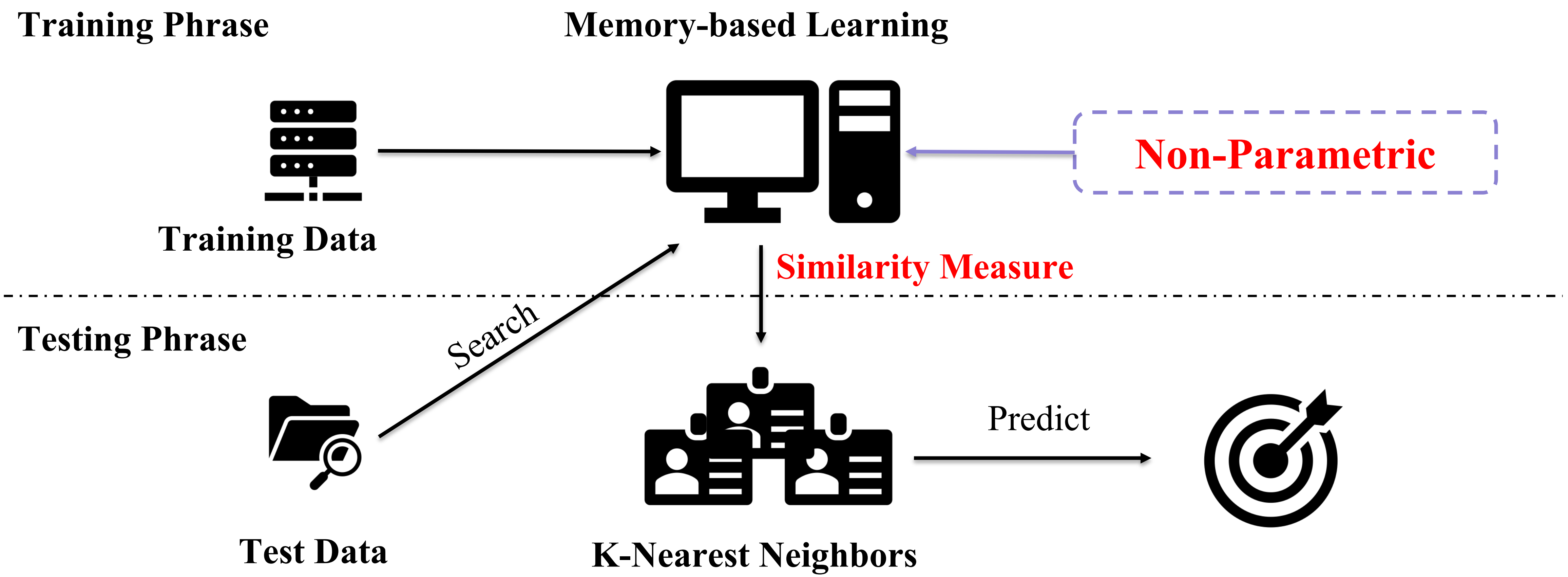
1 – K Nearest Neighbors

What is the KNN Algorithm?



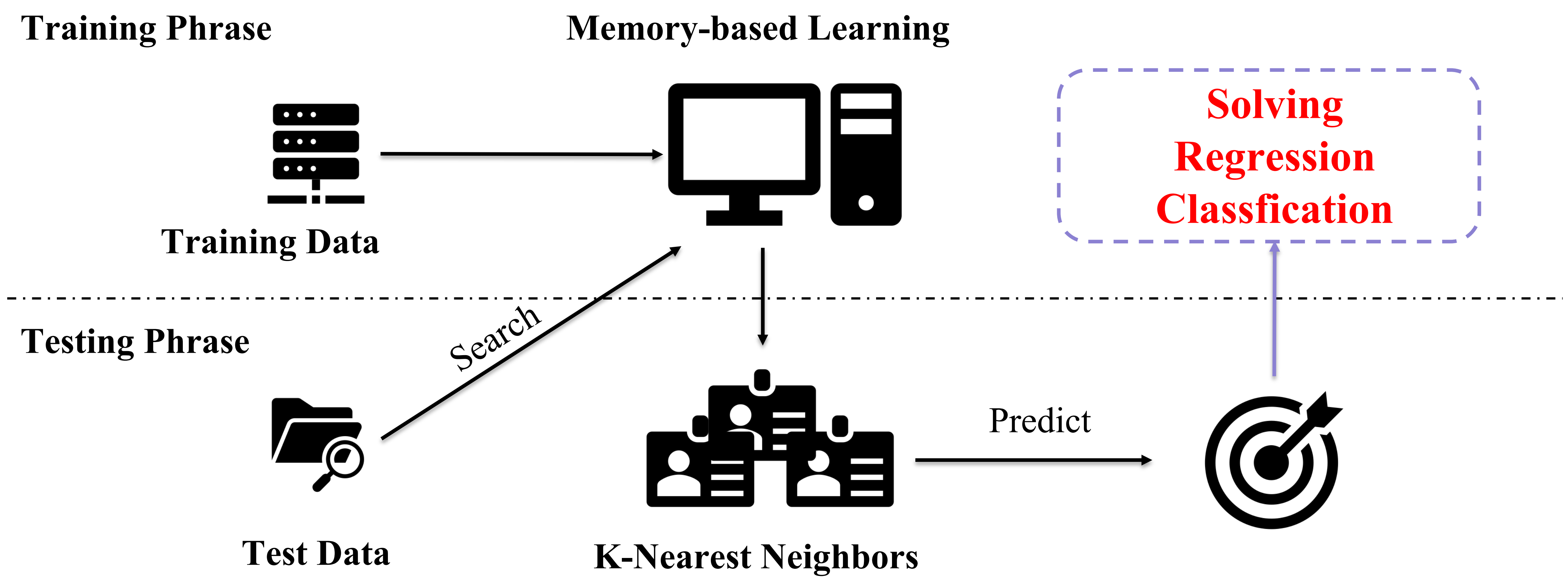
1 – K Nearest Neighbors

What is the KNN Algorithm?



1 – K Nearest Neighbors

What is the KNN Algorithm?



1 – K Nearest Neighbors

What is the KNN Algorithm?

Regression

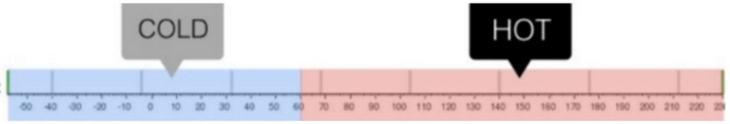
- Predict a continuous value based on the input variables

What will be the temperature tomorrow?

Classification

- Classify input variables to identify discrete output variables (labels, categories)

Will it be hot or cold tomorrow?



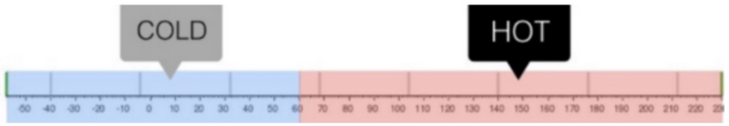
1 – K Nearest Neighbors

KNN: Classification Approach

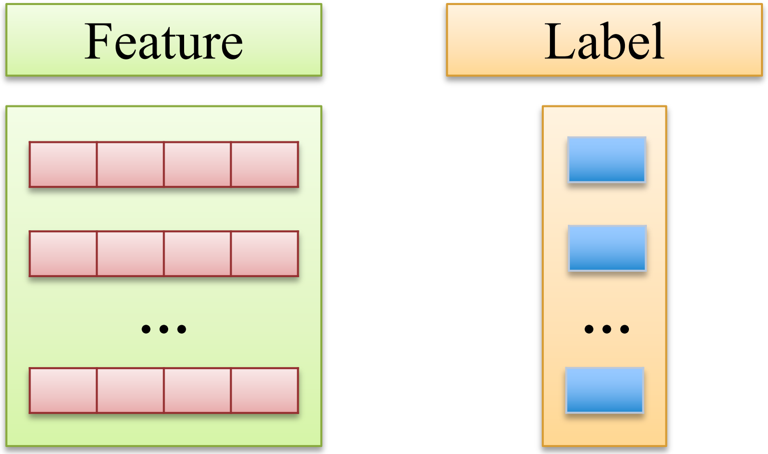
Classification

- Classify input variables to identify discrete output variables (labels, categories)

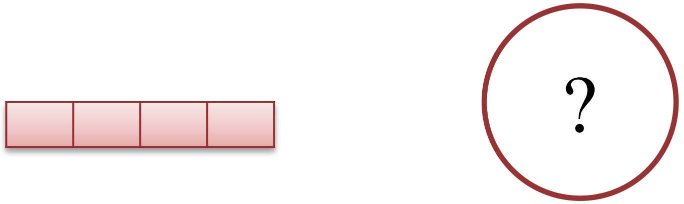
Will it be hot or cold tomorrow?



Training Data



Test Data



1 – K Nearest Neighbors

KNN: Classification Approach

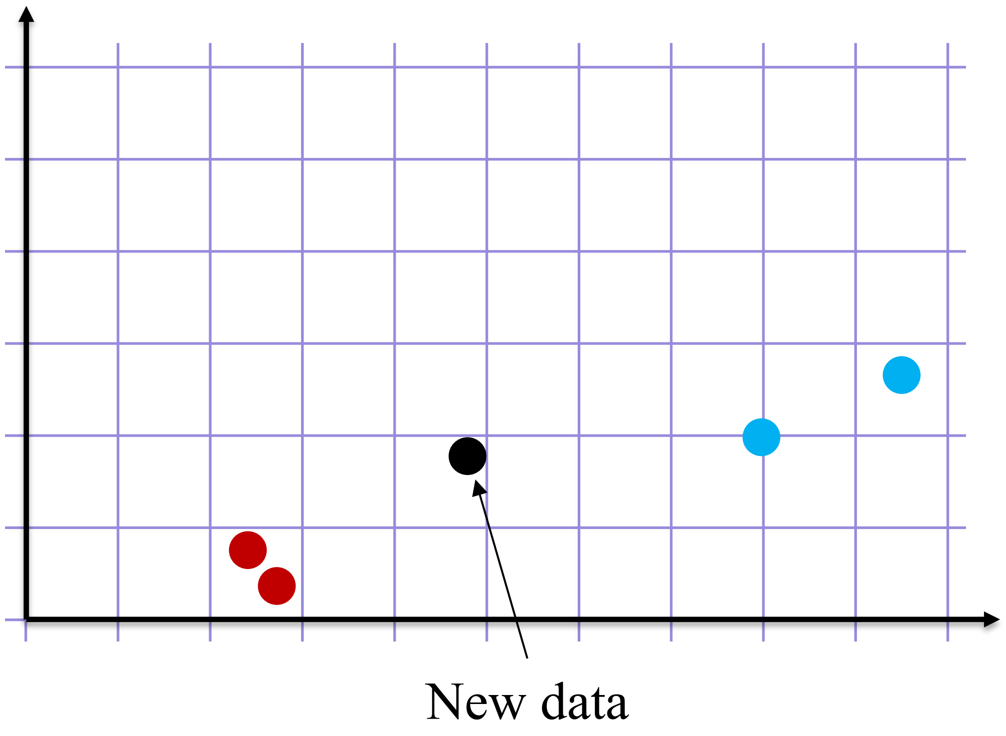
Step 1: Look at the data

Training Data

Petal_Length	Petal_Width	Label
1.4	0.2	0
1.3	0.4	0
4	1	1
4.7	1.4	1

Test Data

2.4	0.8



1 – K Nearest Neighbors

KNN: Classification Approach

Step 2: Calculate distances

Training Data

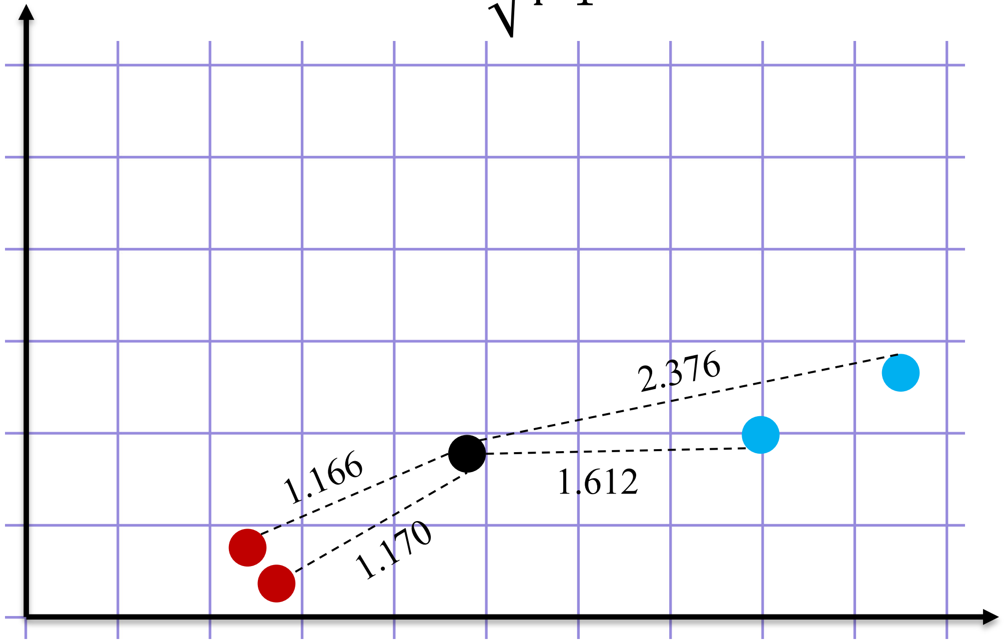
Petal_Length	Petal_Width	Label	Distance
1.4	0.2	0	1.166
1.3	0.4	0	1.170
4	1	1	1.612
4.7	1.4	1	2.376

Test Data

2.4	0.8

Euclidean Distance

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



1 – K Nearest Neighbors

KNN: Classification Approach

Step 3: Find neighbours

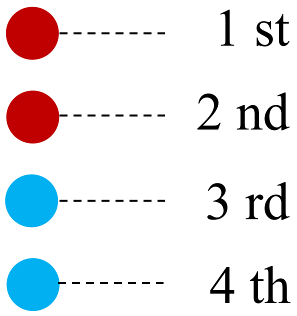
Training Data

Petal_Length	Petal_Width	Label	Distance
1.4	0.2	0	1.166
1.3	0.4	0	1.170
4	1	1	1.612
4.7	1.4	1	2.376

Test Data

2.4	0.8

Ranking points



Find the nearest neighbours by ranking points by increasing distance

1 – K Nearest Neighbors

KNN: Classification Approach

Step 4: Vote on labels

Training Data

Petal_Length	Petal_Width	Label	Distance
1.4	0.2	0	1.166
1.3	0.4	0	1.170
4	1	1	1.612
4.7	1.4	1	2.376

Test Data

2.4	0.8	→	0

K=3 Nearest neighbours

- 1 st
- 2 nd
- 3 rd

# of votes

- 2
- 1

Vote on the predicted class labels based on the class of the k nearest neighbors

1 – K Nearest Neighbors

KNN: Regression Approach

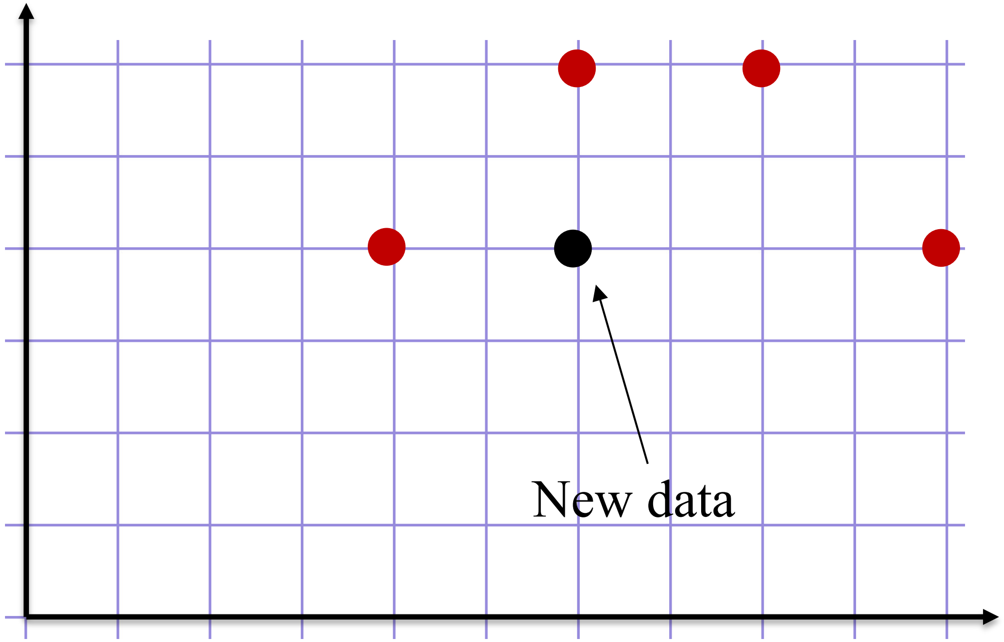
Step 1: Look at the data

Training Data

Length	Width	Price
2.0	2.0	2.0
3.0	3.0	2.5
4.0	3.0	3.5
5.0	2.0	5.0

Test Data

3.0	2.0



AI VIET NAM
@aivietnam.edu.vn

1 – K Nearest Neighbors

KNN: Regression Approach

Step 2: Calculate distances

Training Data

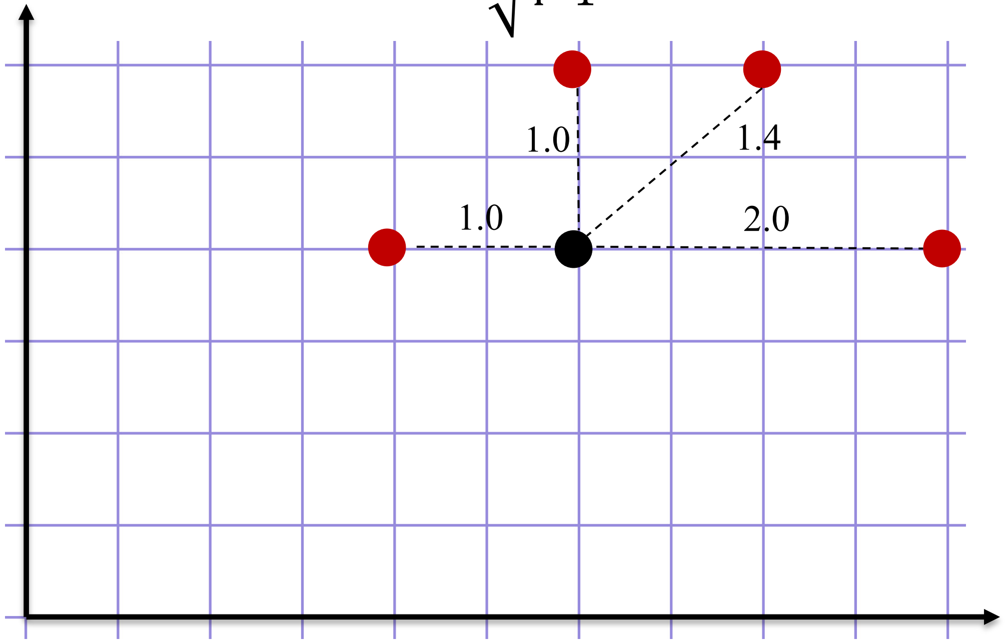
Length	Width	Price	Distance
2.0	2.0	2.0	1.0
3.0	3.0	2.5	1.0
4.0	3.0	3.5	1.4
5.0	2.0	5.0	2.0

Test Data

3.0	2.0

Euclidean Distance

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



1 – K Nearest Neighbors

KNN: Regression Approach

Step 3: Find neighbours

Training Data

Length	Width	Price	Distance
2.0	2.0	2.0	1.0
3.0	3.0	2.5	1.0
4.0	3.0	3.5	1.4
5.0	2.0	5.0	2.0

Test Data

3.0	2.0

Ranking points

1 st
2 nd
3 rd
4 th

Find the nearest neighbours by ranking points by increasing distance

AI VIET NAM
@aivietnam.edu.vn

1 – K Nearest Neighbors

KNN: Regression Approach

Step 4: Vote on labels

Training Data

Length	Width	Price	Distance
2.0	2.0	2.0	1.0
3.0	3.0	2.5	1.0
4.0	3.0	3.5	1.4
5.0	2.0	5.0	2.0

Test Data

3.0	2.0	→	3.25

$$Y_{pred} = \frac{1}{k}\sum_{x \in NB} y_x$$

K=4 Nearest neighbours

- 1 st
- 2 nd
- 3 rd
- 4 th

$$Y_{pred} = \frac{1}{4}(2.0 + 2.5 + 3.5 + 5.0)$$

$=3.25$

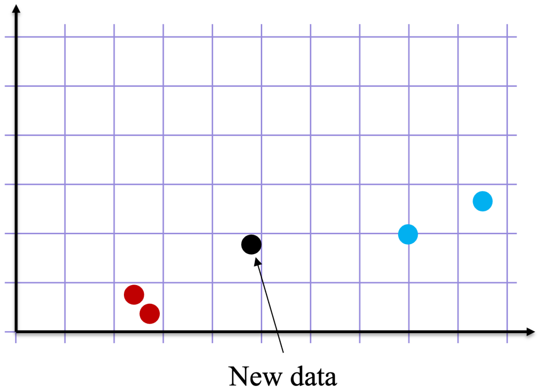
Compute the mean value of the k nearest neighbors

1 – K Nearest Neighbors

KNN: Summary

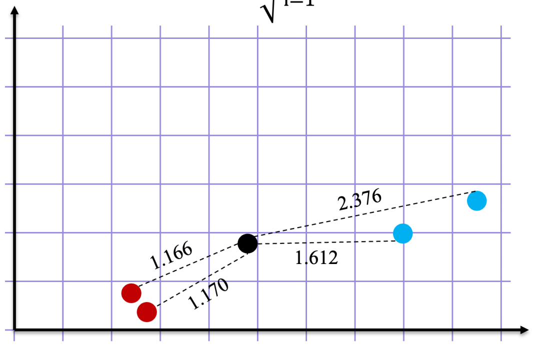
Step 1: Look at the data | **Step 2: Calculate distances** | **Step 3: Find neighbours** | **Step 4: Vote on labels**

Classification

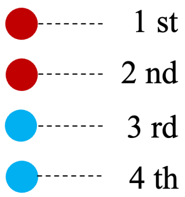


Euclidean Distance

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

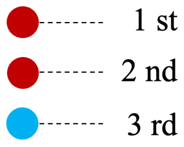


Ranking points

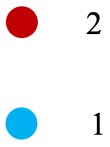


Find the nearest neighbours by ranking points by increasing distance

K=3 Nearest neighbours

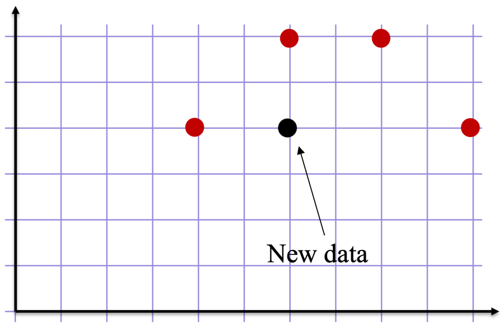


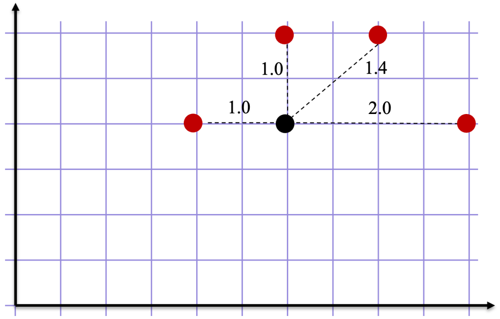
\# of votes



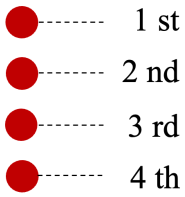
Vote on the predicted class labels based on the class of the k nearest neighbors

Regression



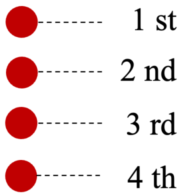


Ranking points



Find the nearest neighbours by ranking points by increasing distance

K=4 Nearest neighbours



$$Y_{pred} = \frac{1}{k}\sum_{x \in NB} y_x$$

Compute the mean value of the k nearest neighbors

1 – K Nearest Neighbors

Geometry Distance Functions

- **Euclidean (p=2)**

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

- **Manhattan (p=1)**

$$d(\mathrm{x},\mathrm{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

- **Minkowski (p-norm)**

$$d(\mathrm{x},\mathrm{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

- **Chebyshev (p=)**

$$d(\mathrm{x},\mathrm{y}) = \lim_{p\to\infty}\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} = \max_i |x_i - y_i|$$

1 – K Nearest Neighbors

Feature Scaling (Normalization)

- Standardize the range of independent variables (feature of data)

Strong Influence

Training Data

Petal_Length	L_Distance	Petal_Width	W_Distance	Label	Distance	Rank
1.4	1.8	0.2	0.4	0	1.844	3
1.3	1.9	0.4	0.2	0	1.910	4
4	0.8	1	0.4	1	0.894	1
4.7	1.3	1.4	0.8	1	1.526	2

Test Data

3.2	0.6

1 – K Nearest Neighbors

Feature Scaling (Normalization)

- Standardize the range of independent variables (feature of data)

Training Data

MinMaxScaler Normalization

Petal_Length	L_Distance	Petal_Width	W_Distance	Label	Distance
0.03	0.53	0.00	0.33	0	0.624
0.00	0.56	0.17	0.16	0	0.582
0.79	0.23	0.66	0.33	1	0.402
1.00	0.44	1.00	0.67	1	0.801

Rank
3
2
1
4

Test Data

0.56	0.33

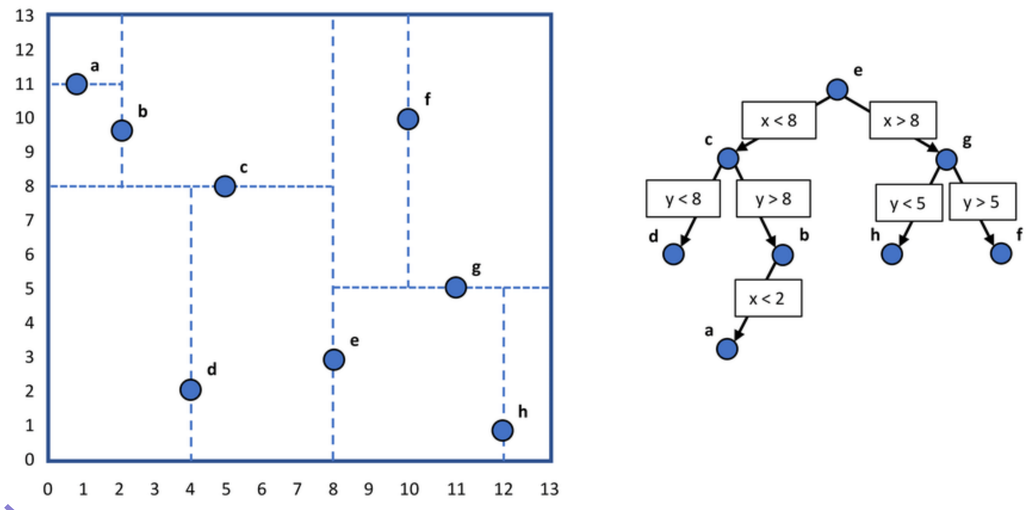
AI VIET NAM
@aivietnam.edu.vn

1 – K Nearest Neighbors

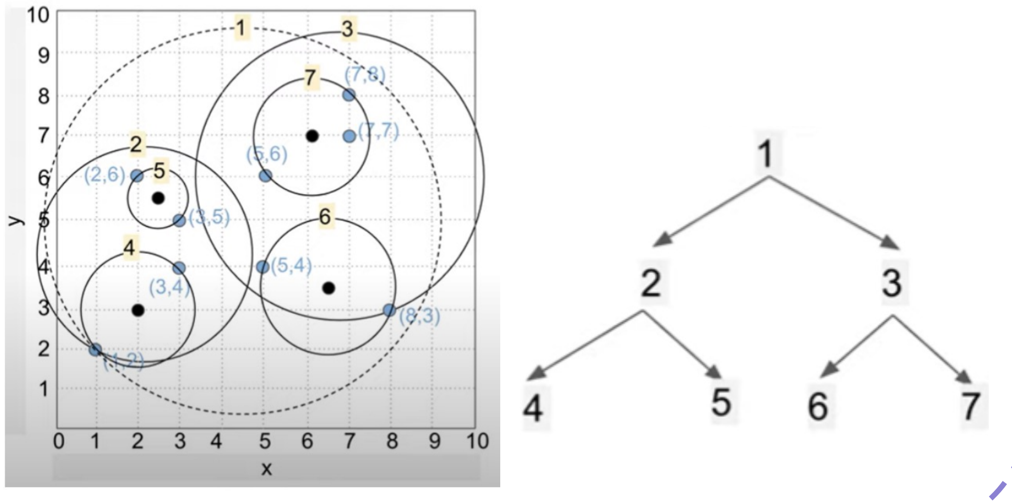
Searching in KNN

- Training dataset: N samples in D dimensions
- Brute Force: Naïve neighbor search – O[DN^2]

1 KD Tree O[NDlog(N)]



2 Ball Tree

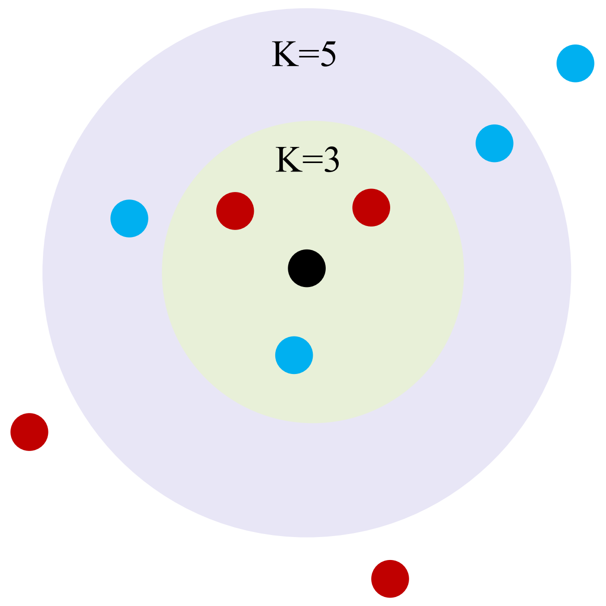


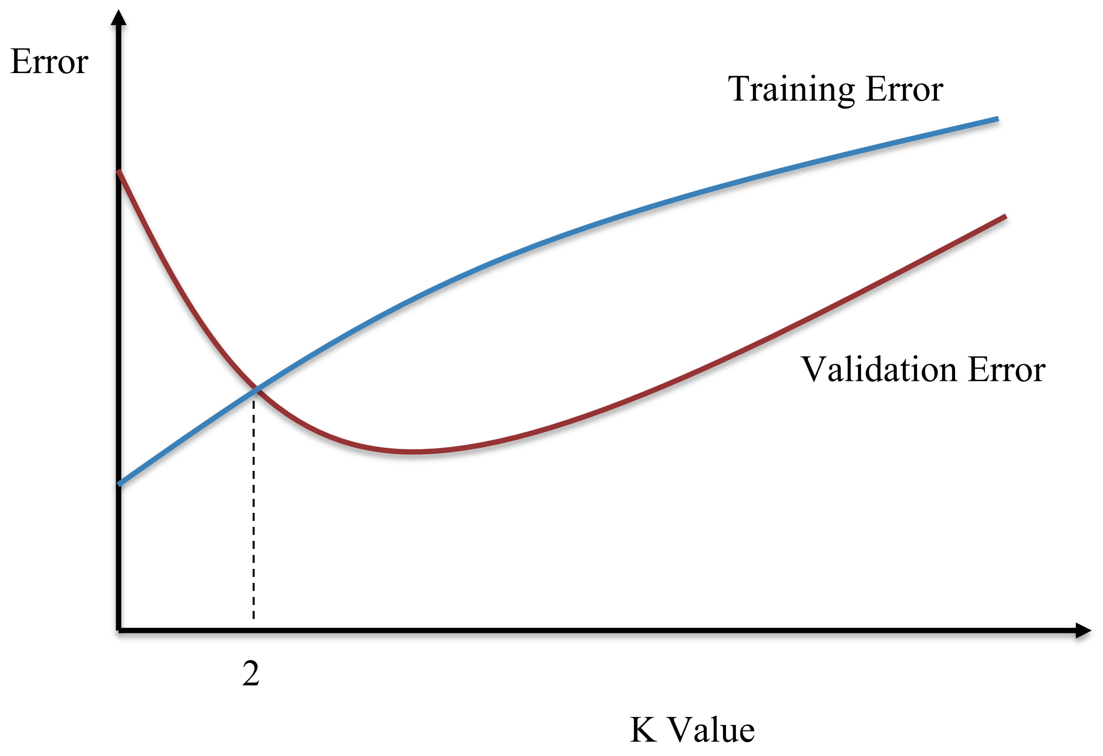
Source: https://varshasaini.in/kd-tree-and-ball-tree-knn-algorithm/

1 – K Nearest Neighbors

How to find the optimal value of K in KNN?

- Choose K based on the evaluation on the validation set (Accuracy, Error, F-Score,…)

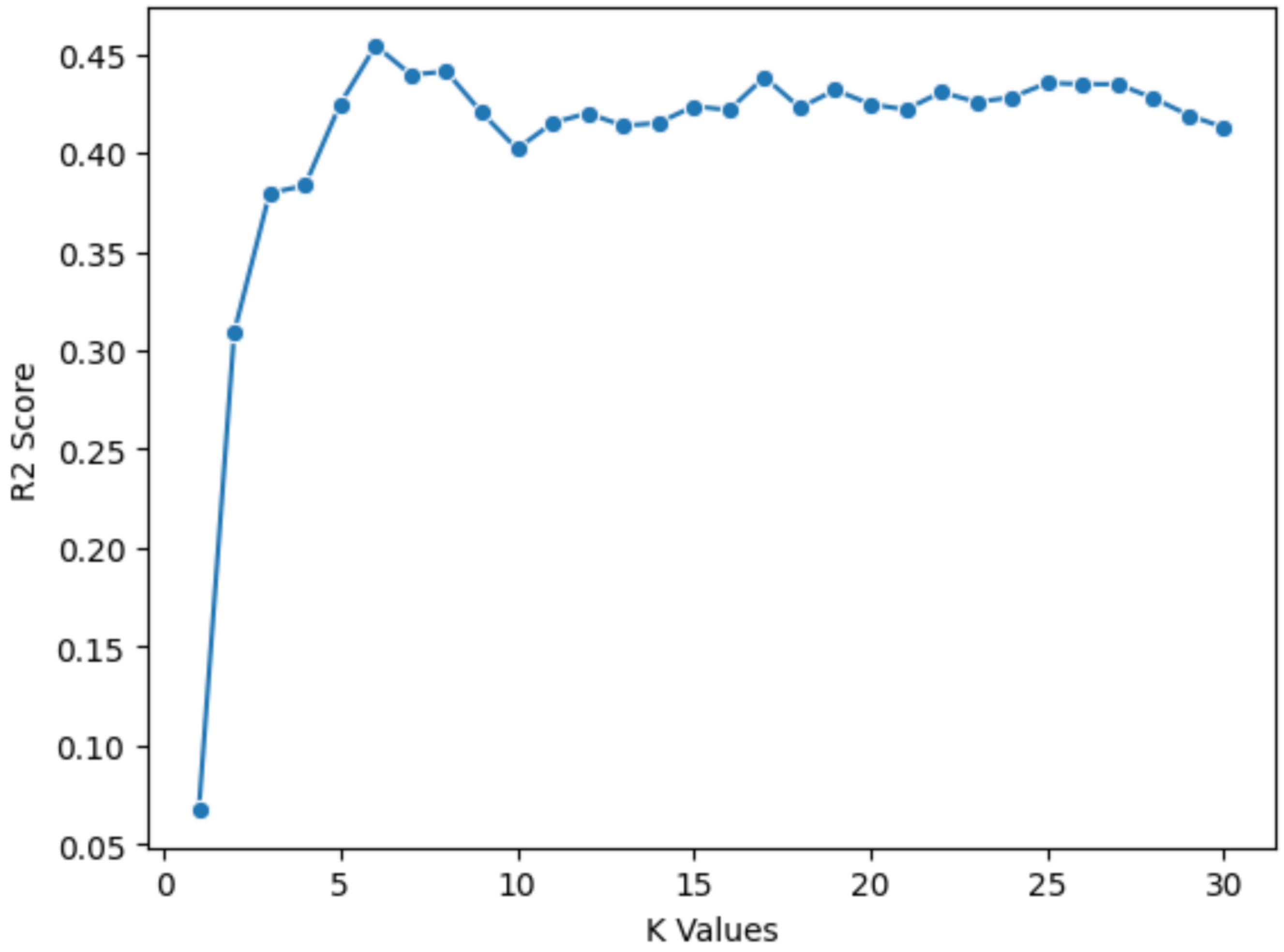




2 – KNN Applications

KNN for Regression: “Diabetes” Dataset

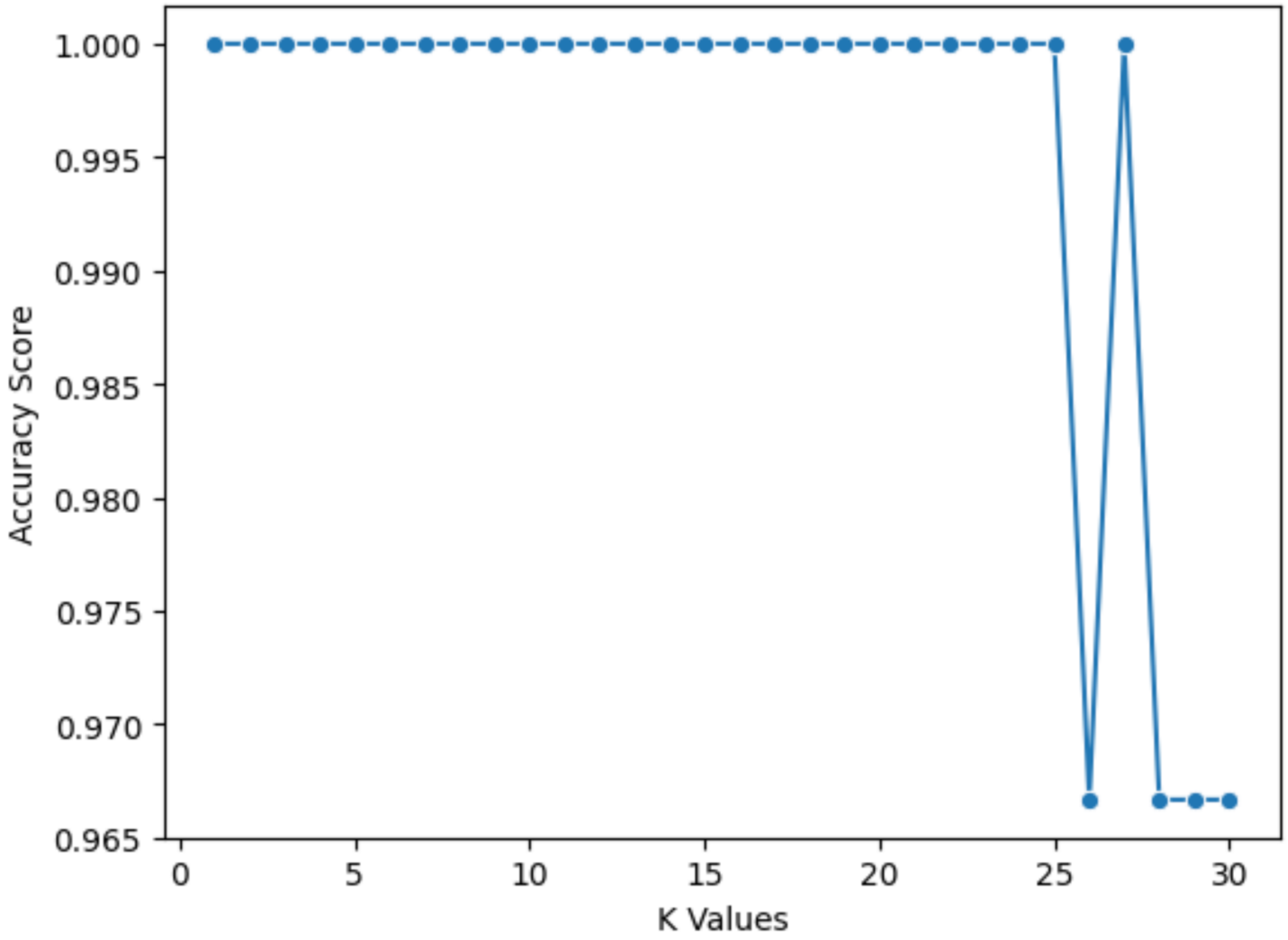
- Sample: 442
- Features: 10
- Target: 25 – 346
- R2-Score (Validation)



2 – KNN Applications

KNN for Classification: “Iris” Dataset

- Sample: 150
- Features: 4
- Classes: 3 (50 per class)
- Accuracy (Validation)

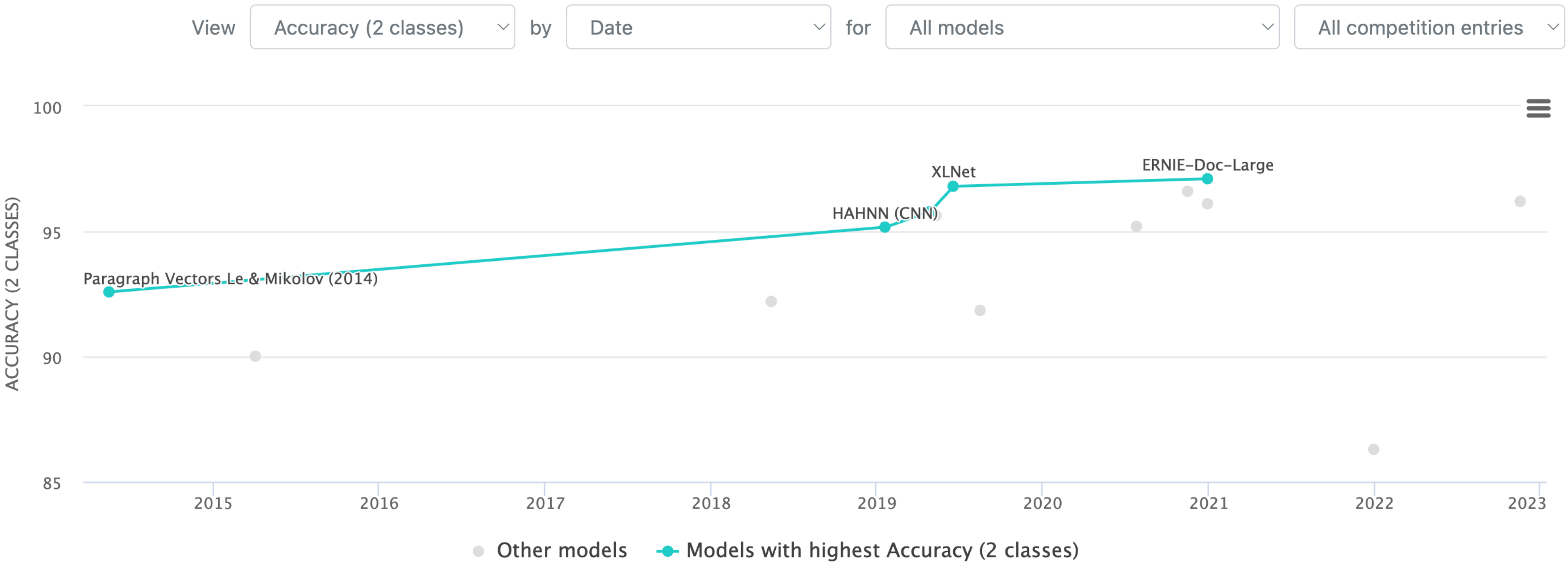


AI VIET NAM
@aivietnam.edu.vn

2 – KNN Applications

KNN for Text Classification: "IMDB" Dataset

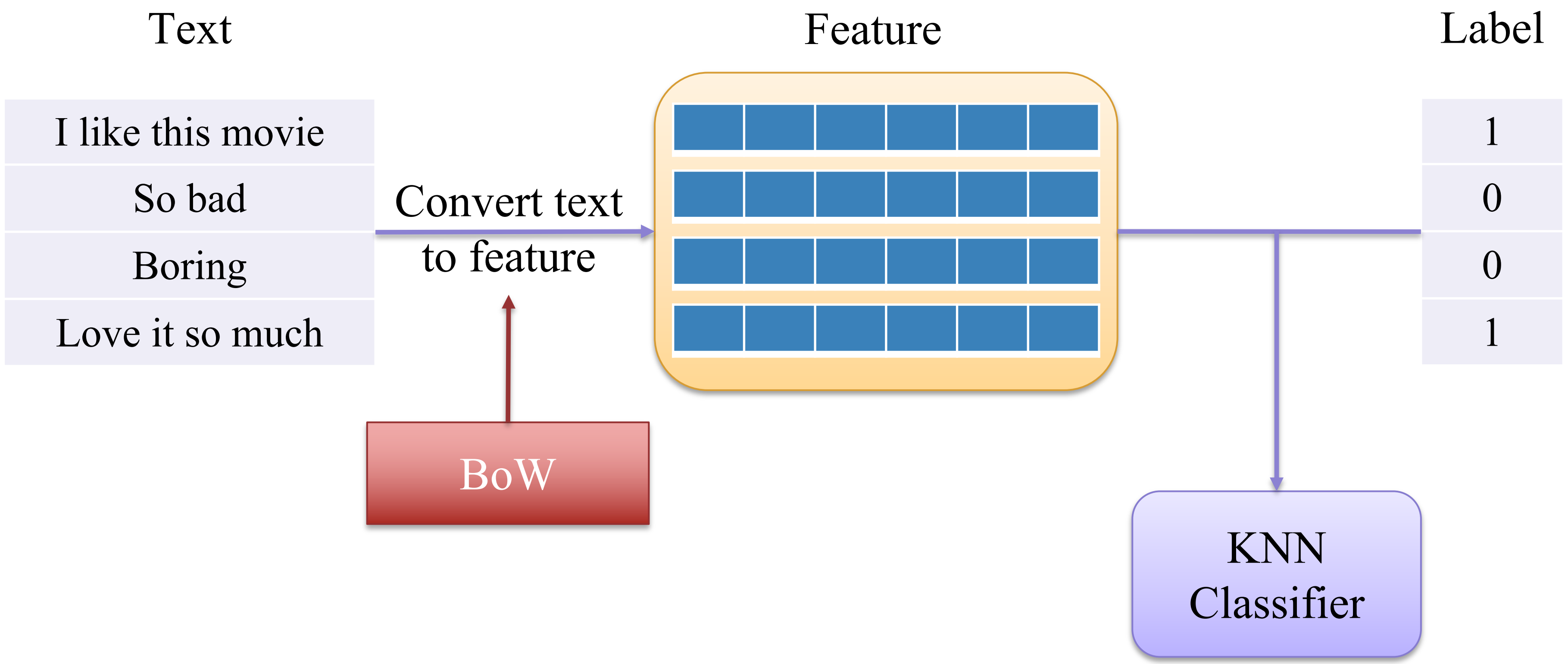
- Sample: 50.000 movie review
- Classes: 2 – Positive and Negative (25.000 per class)
- Accuracy, F1-Score



https://paperswithcode.com/sota/text-classification-on-imdb

2 – KNN Applications

KNN for Text Classification: "IMDB" Dataset



2 – KNN Applications

KNN for Text Classification: "IMDB" Dataset

- **Bag of Words (BoW)**
- **Document-Level:** Consider text as a bag (collection) of words
- **Represented by a V-dimensional**

Use: the number of occurrences of the word in the document

[dog, bites, man]
[man, bites, dog]
[dog, eats, meat]
[man, eats, food]

Vocabulary

IDX	0	1	2	3	4	5
Token	bites	dog	eats	food	man	meat

Counter

[dog, bites, man]

1	1	0	0	1	0

2 – KNN Applications

KNN for Text Classification: “IMDB” Dataset

- **Bag of Words (BoW)**
- **Document-Level:** Consider text as a bag (collection) of words
- **Represented by a V-dimensional**

[dog, bites, man]	1	1	0	0	1	0
[man, bites, dog]	1	1	0	0	1	0
[dog, eats, meat]	0	1	1	0	0	1
[man, eats, food]	0	0	0	1	1	0

2 – KNN Applications

! KNN for Text Classification: “IMDB” Dataset

- **Bag of Words (BoW)**
- **Out of vocabulary (OOV)**

Vocabulary

IDX	0	1	2	3	4	5
Token	bites	dog	eats	food	man	meat

Counter

[dog, bites, man]	1	1	0	0	1	0
[dog, and, dog, are, friends]	0	2	0	0	0	0

2 – KNN Applications

KNN for Text Classification: “IMDB” Dataset

- **Bag of Words (BoW)**
- Representation without considering frequency (Binary Representation)

Vocabulary

IDX	0	1	2	3	4	5
Token	bites	dog	eats	food	man	meat

Counter

[dog, and, dog, are, friends]

0	2	0	0	0	0

Convert to binary representation

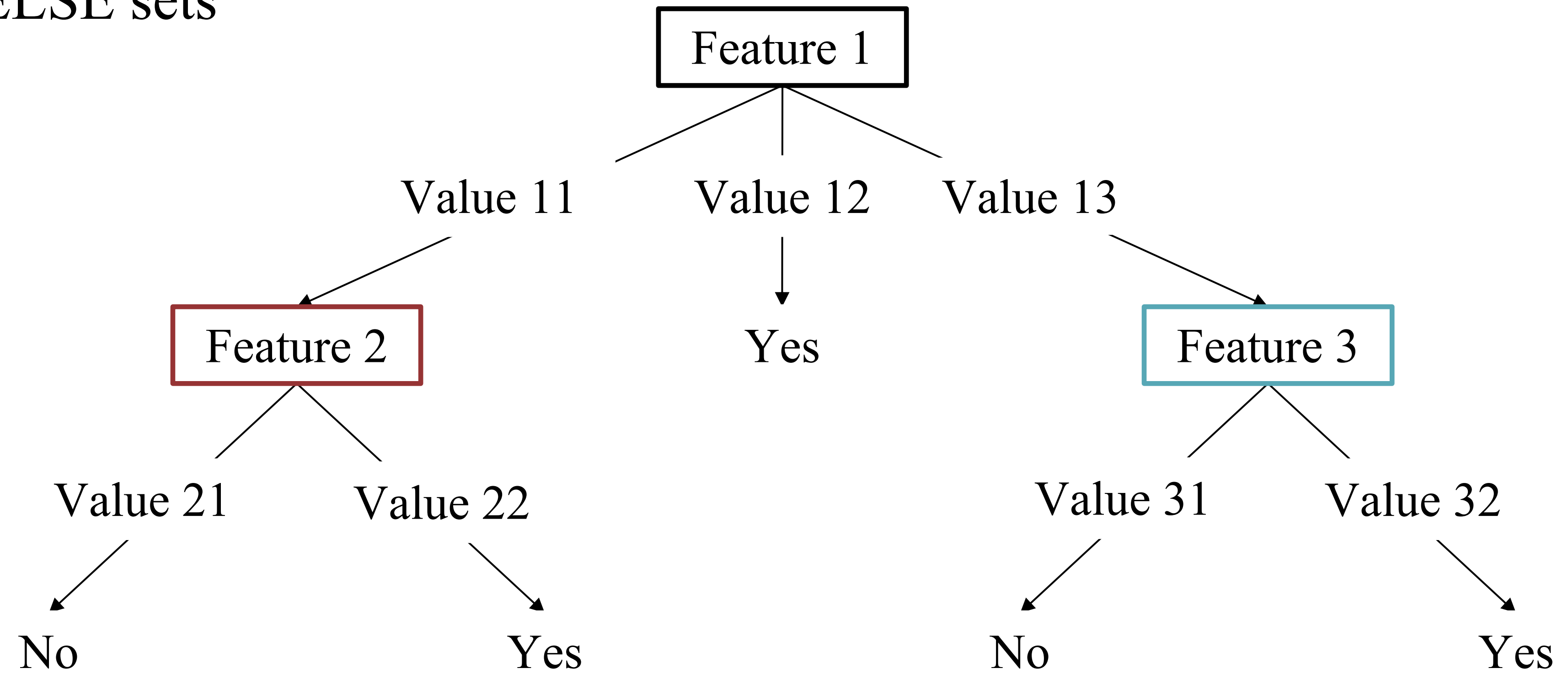
[dog, and, dog, are, friends]

0	1	0	0	0	0

3 – Decision Tree

Decision Tree

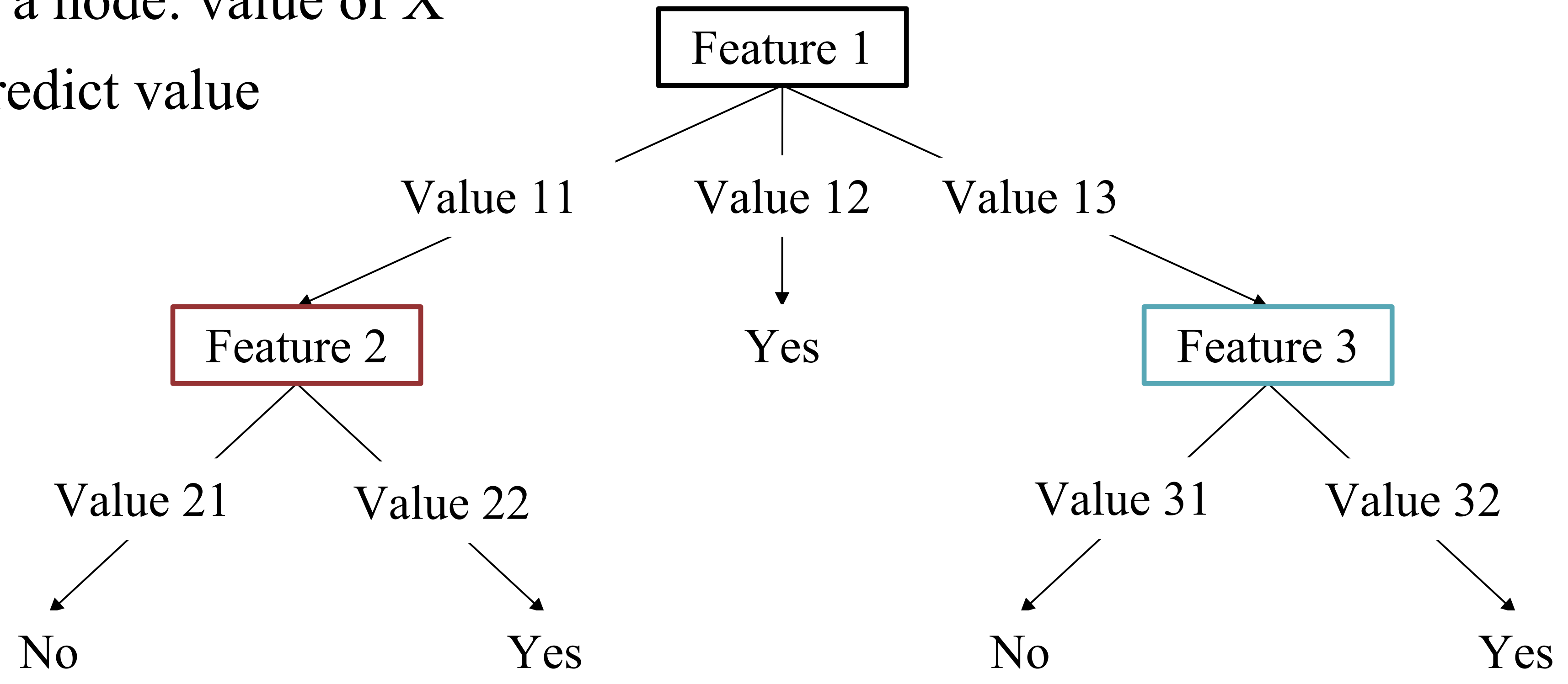
- A possible decision tree for the data
- Described by IF-ELSE sets



3 – Decision Tree

Decision Tree

- Each internal node: attribute X (feature)
- Each branch from a node: value of X
- Each leaf node: predict value



3 – Decision Tree

Decision Tree

➢ Example: Play Tennis dataset

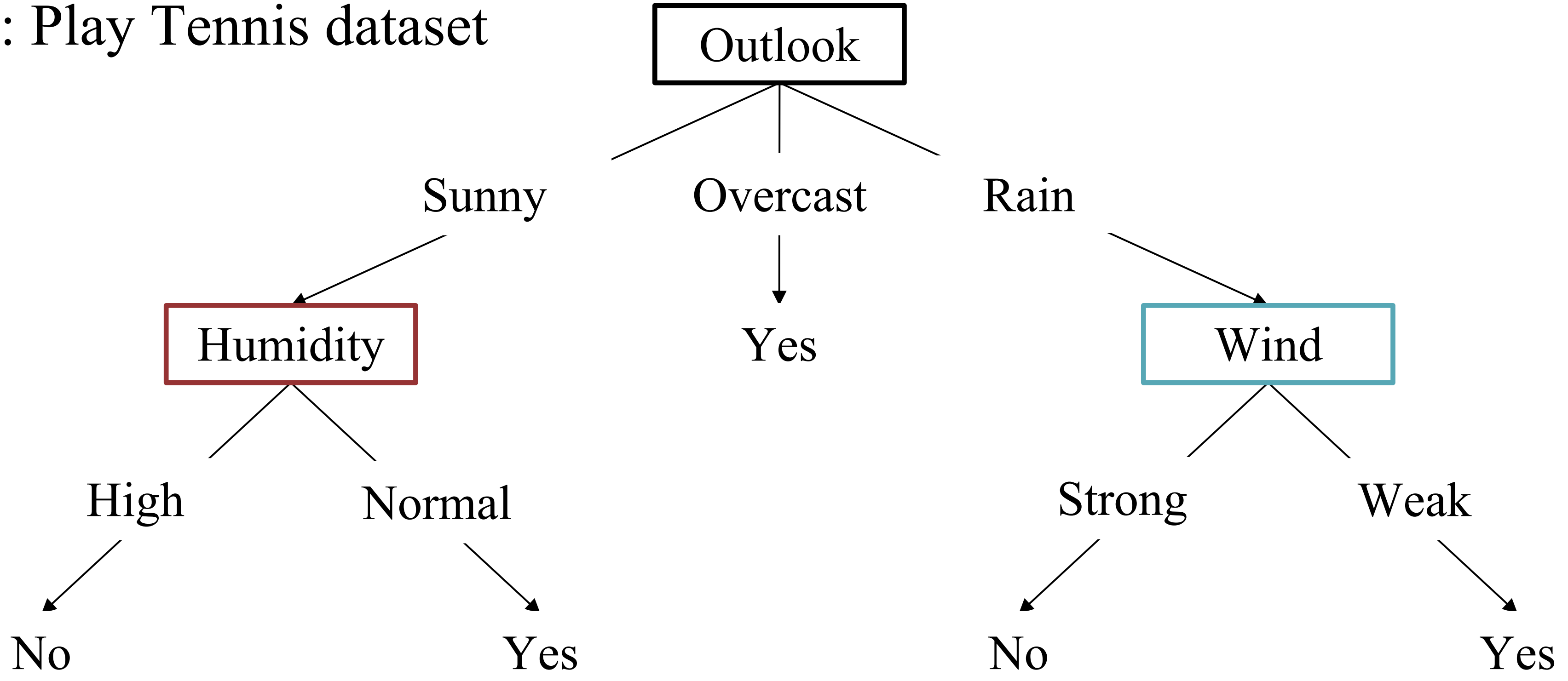
PlayTennis: training examples

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

3 – Decision Tree

Decision Tree

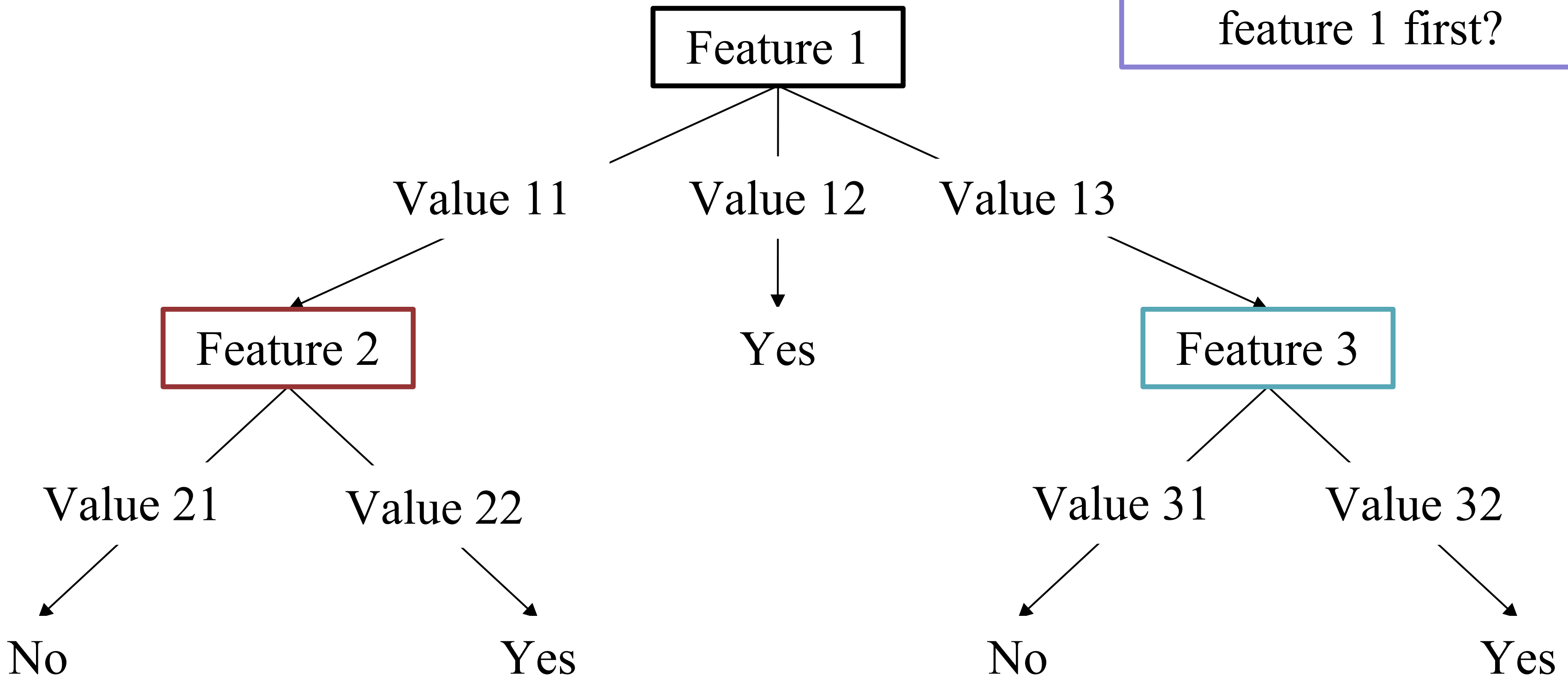
- Example: Play Tennis dataset



What prediction would we make for
<outlook=Sunny, temperature=Hot, humidity=High, wind=Weak> ?

3 – Decision Tree

Constructing Decision Tree: ID3

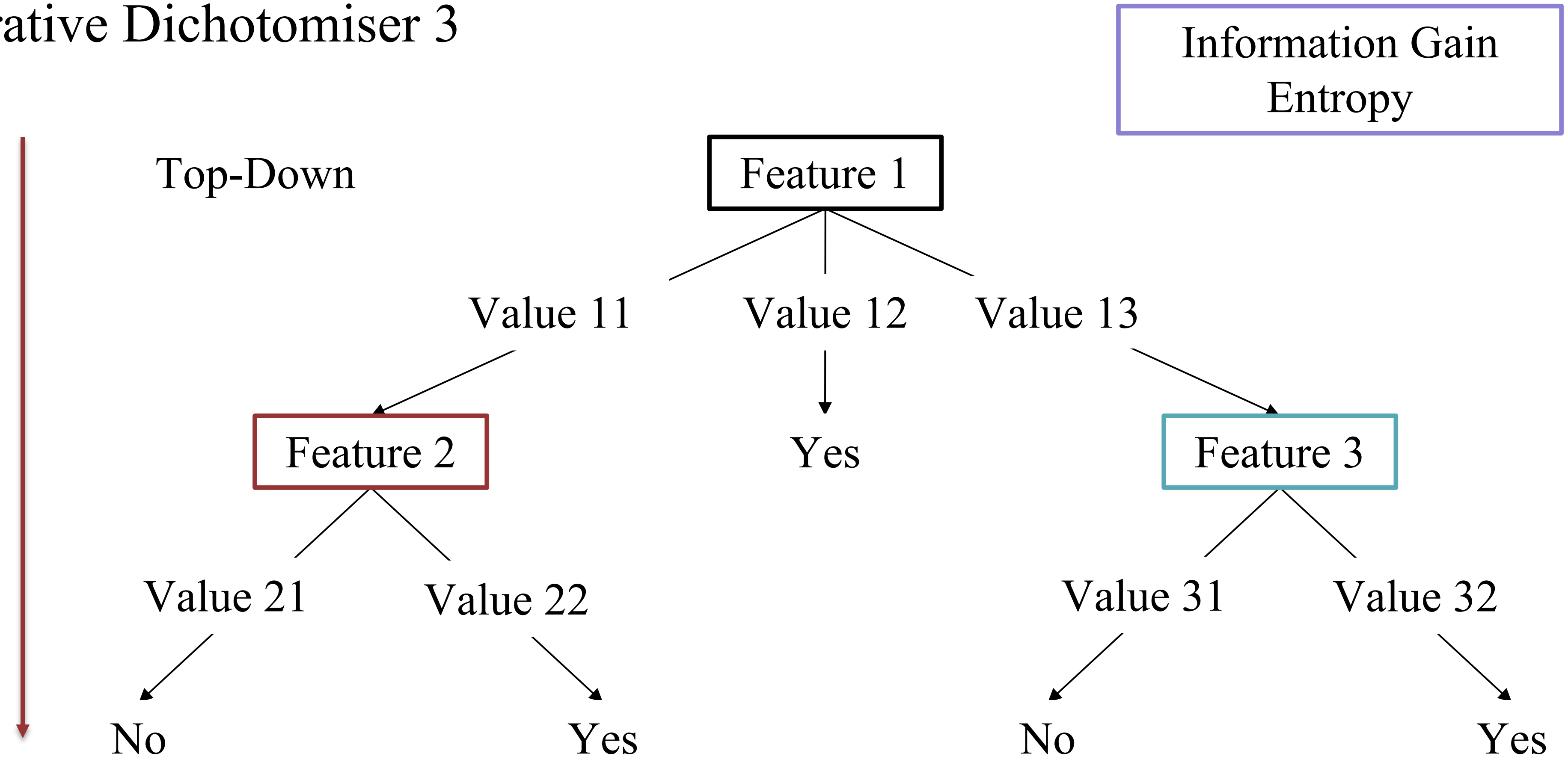


Why does it make more sense to test the feature 1 first?

3 – Decision Tree

Constructing Decision Tree: ID3

- Iterative Dichotomiser 3



3 – Decision Tree

Constructing Decision Tree: ID3

- Entropy is a measure of randomness/uncertainty of a set
- Data: a set S of examples with C many classes
- Probability vector a = $[p_1, p_2, \ldots, p_c]$ is the class distribution of the set S
- Entropy of the set S:

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

- If a set S of examples has

  Some dominant classes => small entropy of the class distribution

  Equiprobable classes => high entropy of the class distribution

3 – Decision Tree

Constructing Decision Tree: ID3

- Entropy of the set S:

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

- Example:

  S: 14 examples (9: class c1, 5: class c2)

  => E(S) = -(9/14).log2(9/14) –(5/14).log2(5/14) = 0.94

3 – Decision Tree

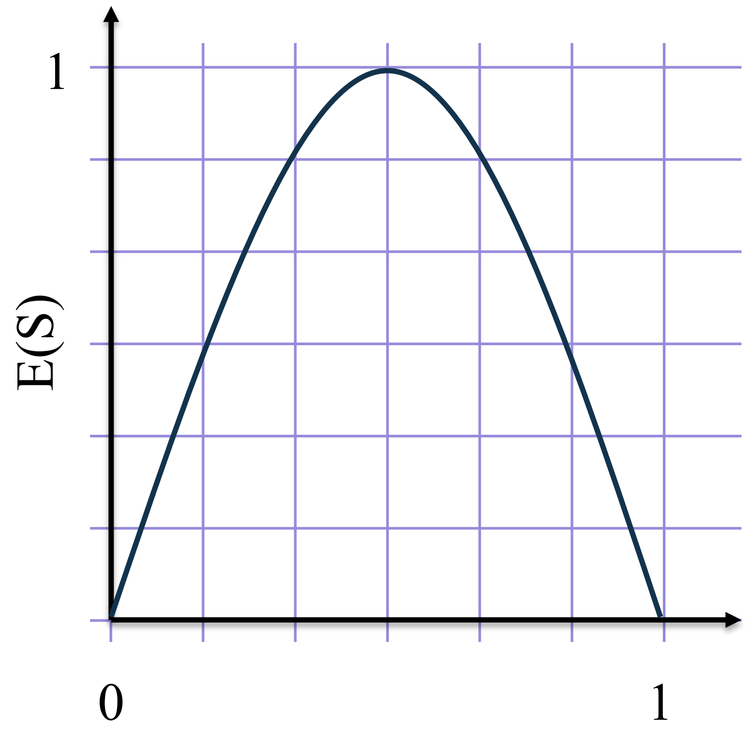
Constructing Decision Tree: ID3

- Entropy = 0

  => all samples class 1 (or 2)
- Entropy = 1

  => num samples class 1 = class 2
- Entropy ∈ (0,1)

  => num samples class 1 ≠ class 2



3 – Decision Tree

Constructing Decision Tree: ID3

- **Information Gain (IG)** on knowing the value of the feature F in S:

$$\mathrm{Gain}(S, F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

- S_f donates the subset of elements of S for which feature F has value f

3 – Decision Tree

Constructing Decision Tree: ID3

- **Information Gain (IG)**
- Gain(S, Wind) with Wind: Weak or Strong

$S = \{9: \text{Yes}, 5: \text{No}\}$

$S_{weak} = \{6: \text{Yes}, 2: \text{No}\}$

$W_{strong} = \{3: \text{Yes}, 3: \text{No}\}$

=> Gain(S, Wind)

$$= E(S) - \frac{8}{14} E(S_{weak}) - \frac{6}{14} E(S_{strong})$$

$$= 0.94 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1 = 0.048$$

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

3 – Decision Tree

Constructing Decision Tree: ID3

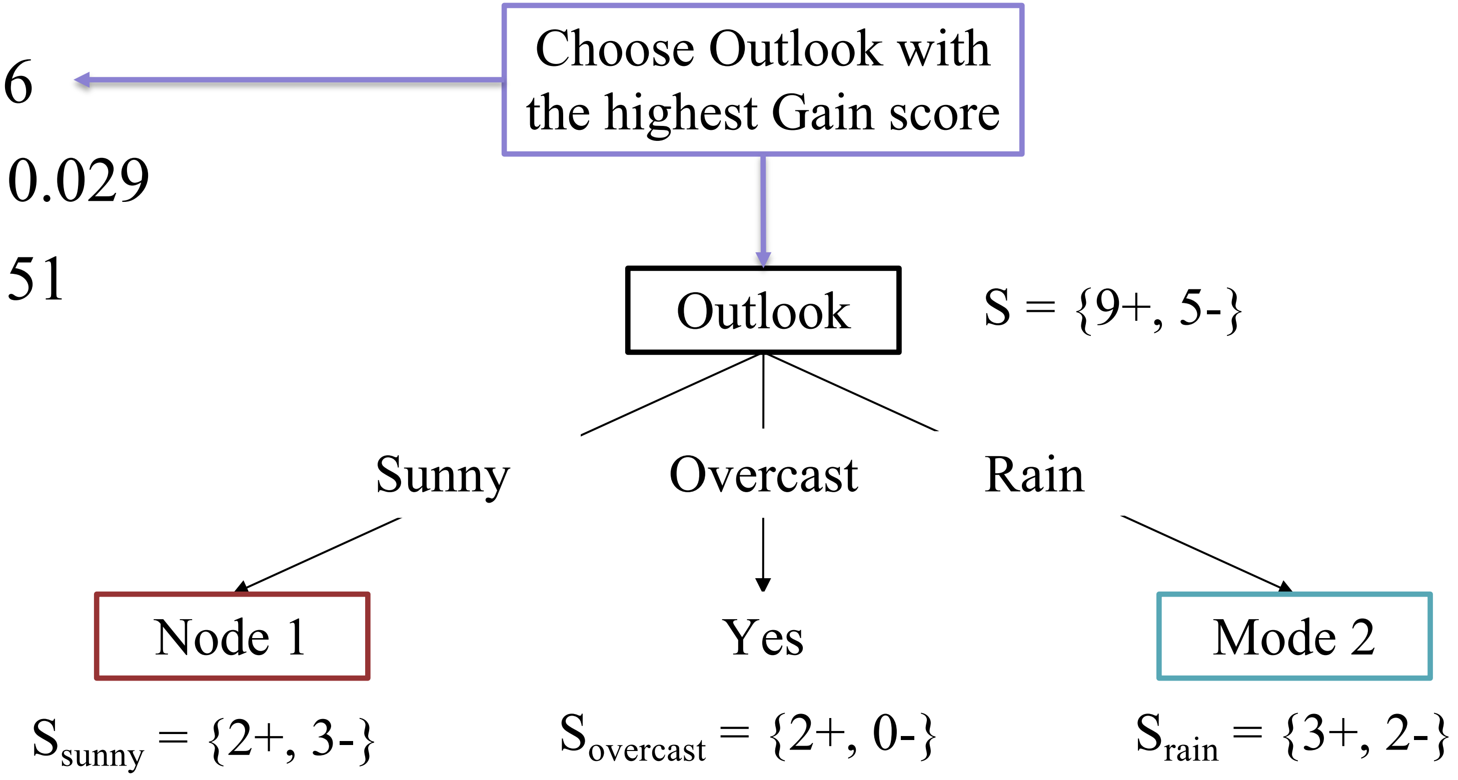
- **Information Gain (IG)**

  Gain(S, Outlook) = 0.246

  Gain(S, Temperature) = 0.029

  Gain(S, Humidity) = 0.151

  Gain(S, Wind) = 0.048



3 – Decision Tree

Constructing Decision Tree: ID3

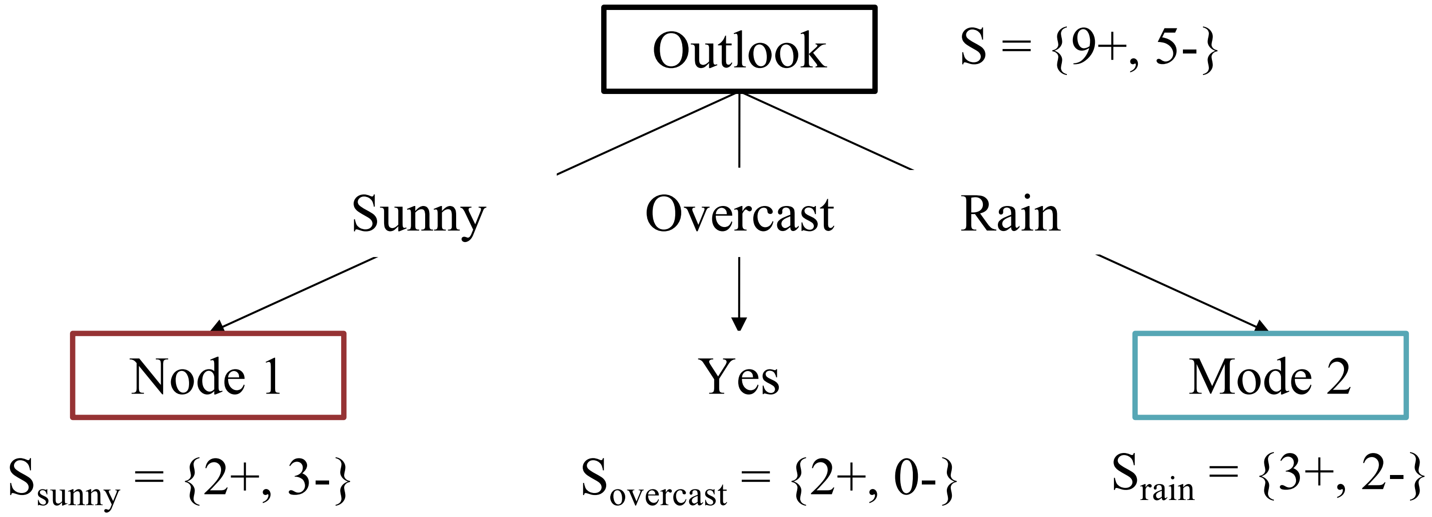
Information Gain (IG)

Fine “Node 1”: {Temperature, Humidity, Wind?

Gain(S_{sunny}, Temperature) = 0.57

Gain(S_{sunny}, Humidity) = 0.57

Gain(S_{sunny}, Wind) = 0.57



3 – Decision Tree

Constructing Decision Tree: ID3

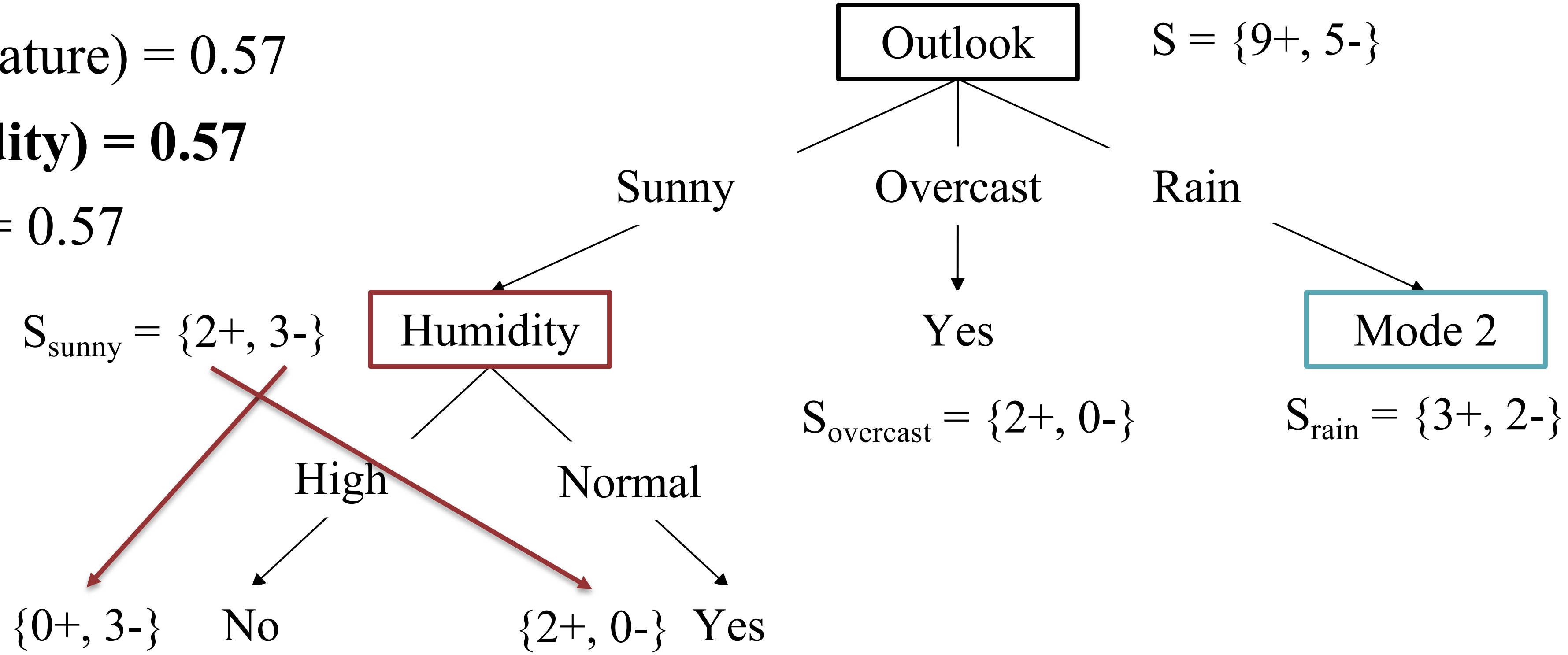
- **Information Gain (IG)**

Fine “Node 1”: {Temperature, Humidity, Wind?

Gain(S_{sunny}, Temperature) = 0.57

Gain(S_{sunny}, Humidity) = 0.57

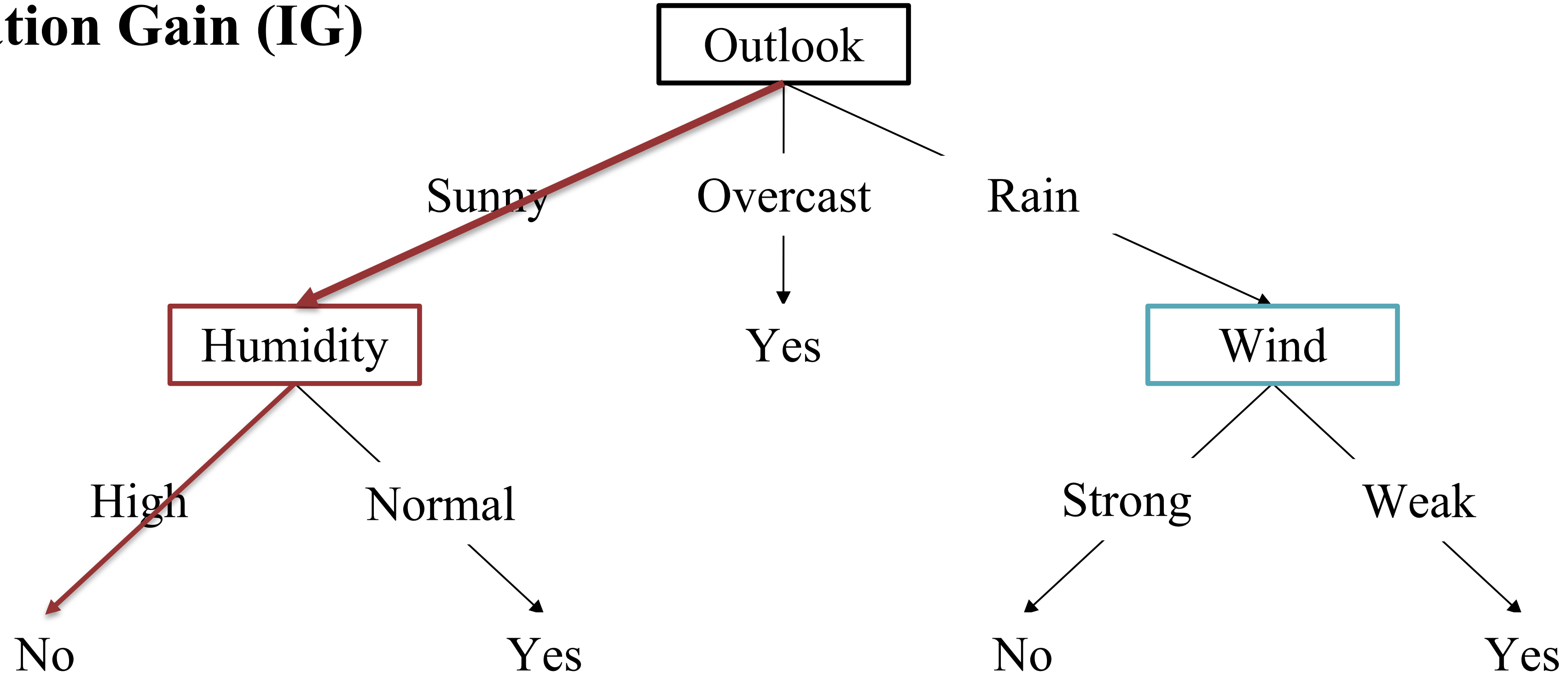
Gain(S_{sunny}, Wind) = 0.57



3 – Decision Tree

Constructing Decision Tree: ID3

- **Information Gain (IG)**

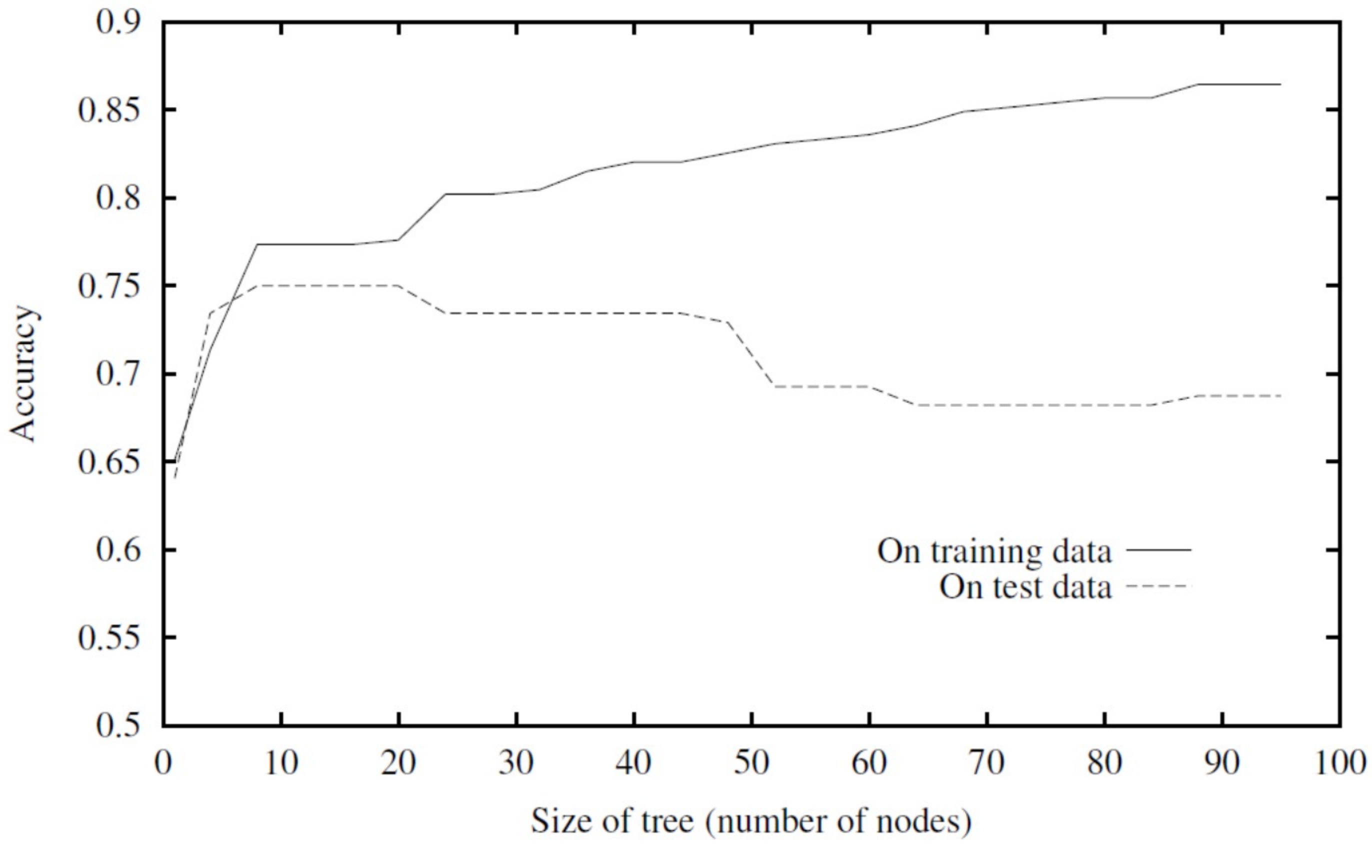


Test = <outlook=Sunny, temperature=Hot, humidity=High, wind=Weak> => No

3 – Decision Tree

Overfitting in Decision Trees

- Desired: a DT that is not too big in size, yet fits the training data reasonably

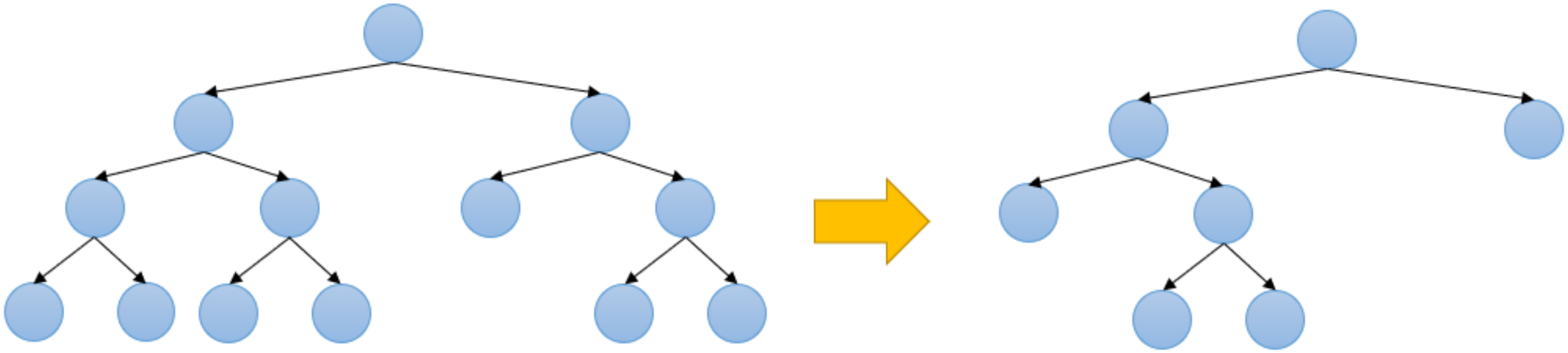


Source

3 – Decision Tree

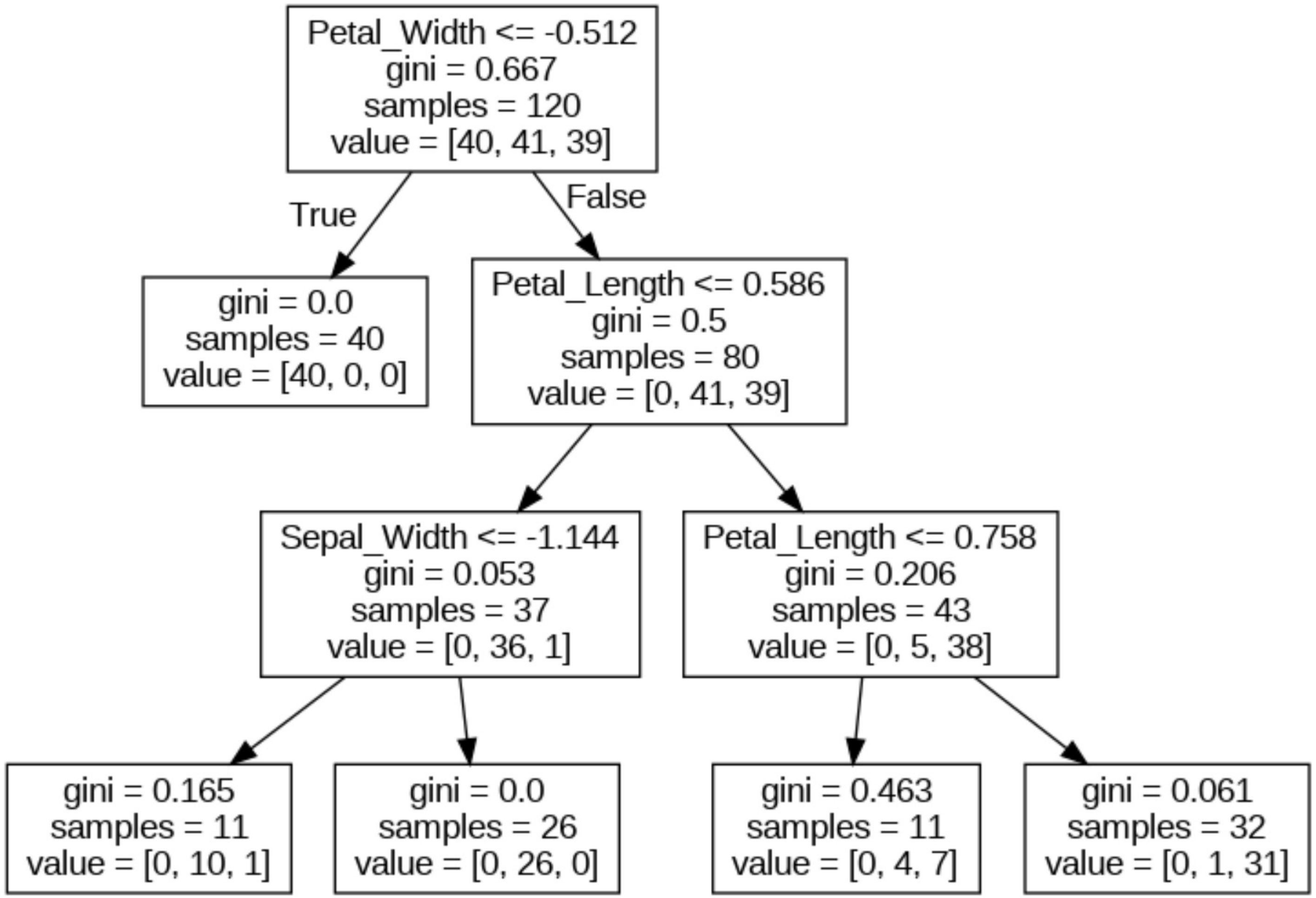
Overfitting in Decision Trees

- 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: evaluate validation set



3 – Decision Tree

Decision Tree for Iris Dataset

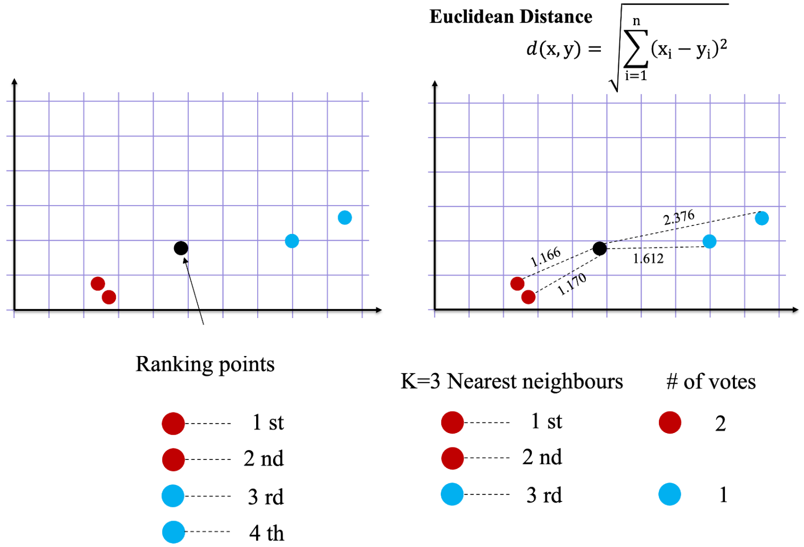


AI VIET NAM
@aivietnam.edu.vn

SUMMARY

KNN

- Predicted based on K-Nearest Neighbors from the training data through Geometry Distance Functions

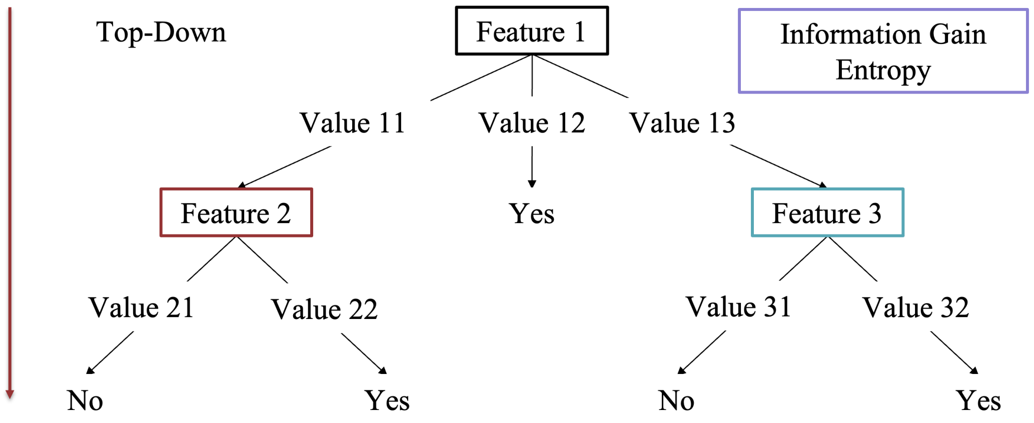


Find the nearest neighbours by ranking points by increasing distance

Vote on the predicted class labels based on the class of the k nearest neighbors

Decision Tree

- Build a decision tree (ID3)
- Defined by a hierarchy of rules (in form of a tree)



Thanks!

Any questions?