

Lecture 3

K-Nearest-Neighbors Algorithm and Data Conversion

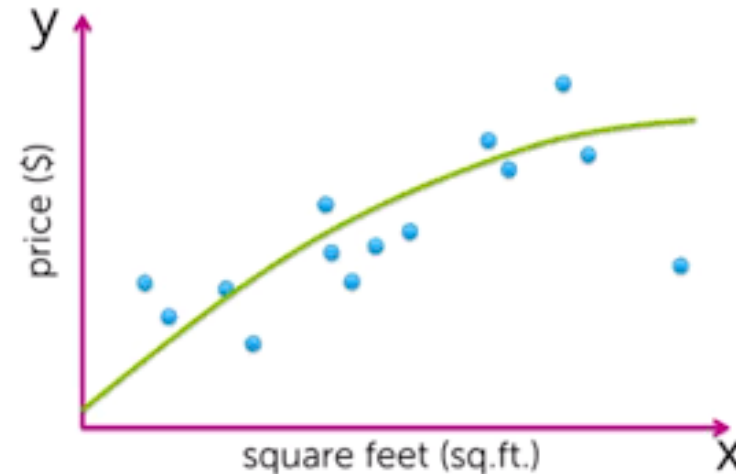
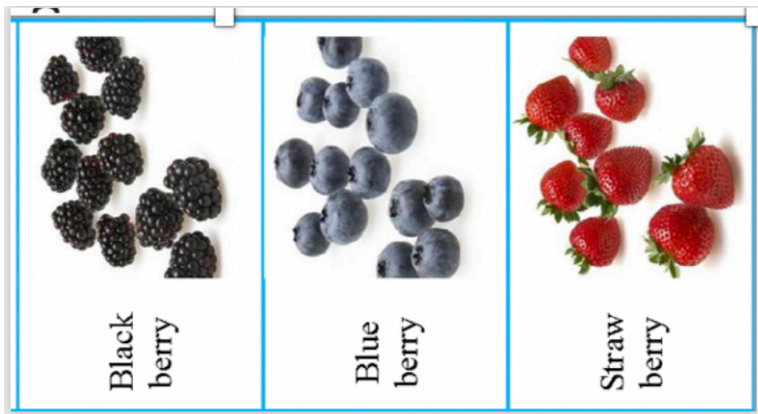
Lê Anh Cường

TDTU

Supervised Learning

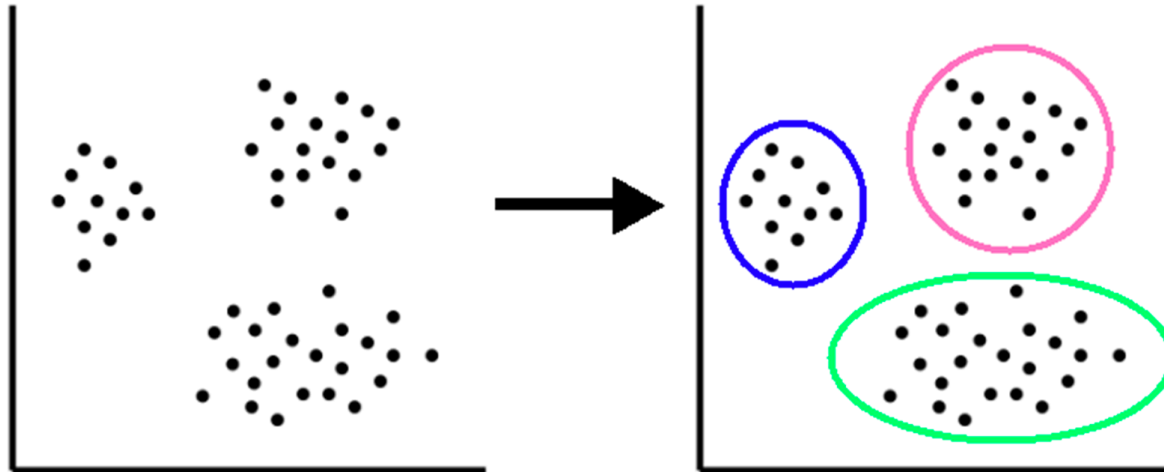
- Classification
- Regression

The main difference between them is that the output variable in **regression** is numerical (or continuous) while that for **classification** is categorical (or discrete).



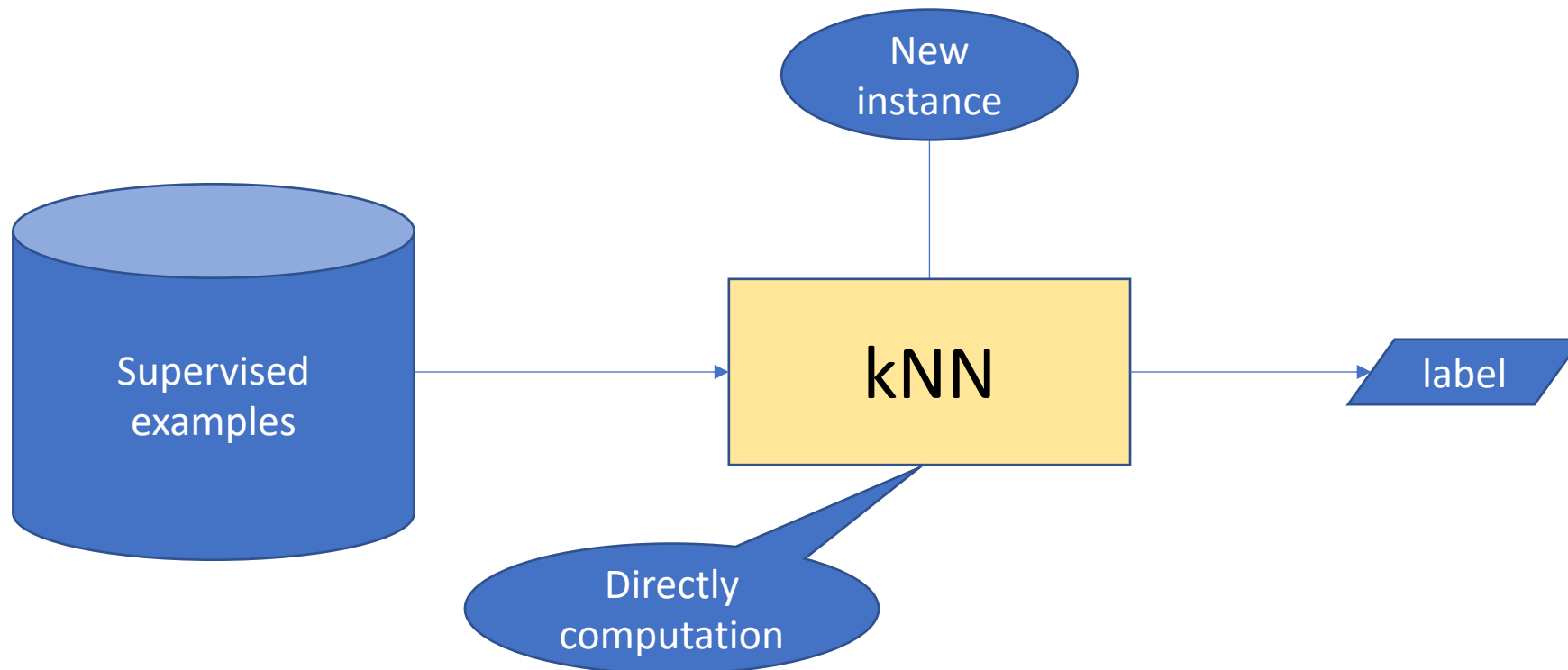
Unsupervised Learning

Unsupervised Learning is a **machine learning** technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected.



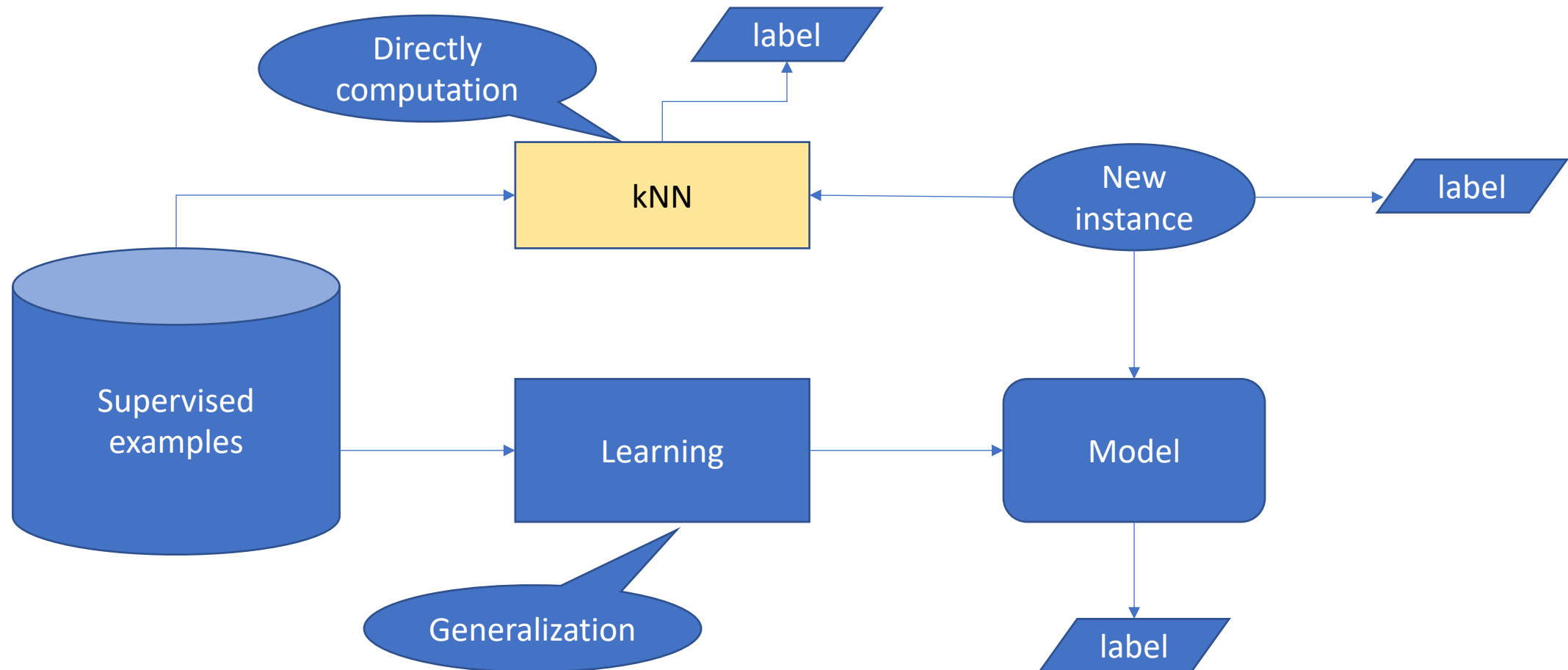
KNN is a Method in Instance-Based Learning

- Instance-based learning is often termed *lazy* learning, as there is typically no “transformation” of training instances into more general “statements”



KNN is a Method in Instance-Based Learning

- Instance-based learning is often termed *lazy* learning, as there is typically no “transformation” of training instances into more general “statements”

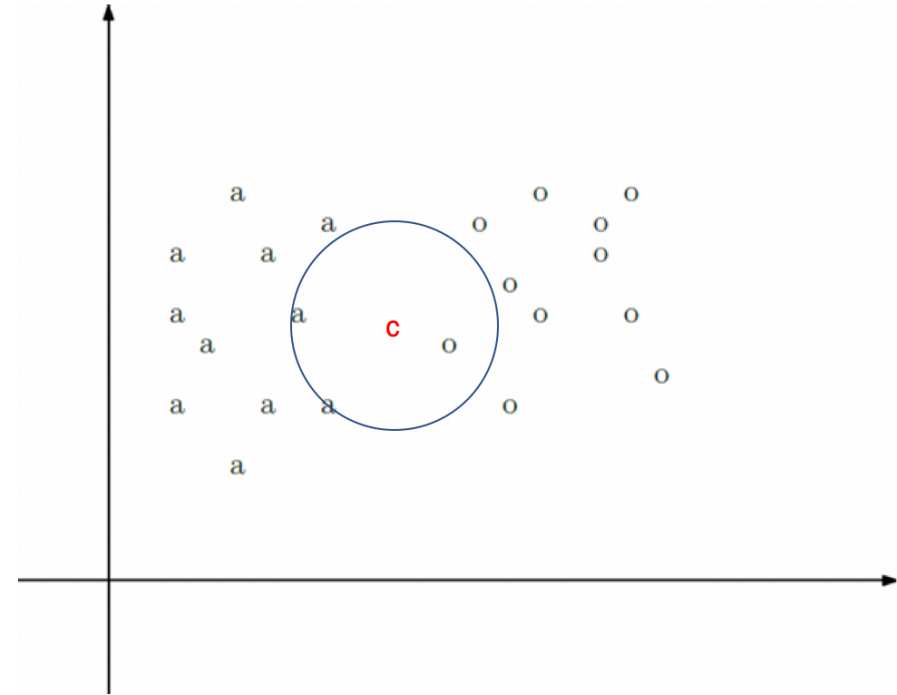


K-Nearest-Neighbors Algorithm

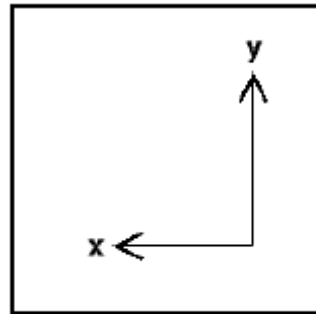
- A case is classified by a majority voting of its neighbors, with the case being assigned to the class most common among its K nearest neighbors measured by a Distance Function.
- If $K=1$, then the case is simply assigned to the class of its nearest neighbor

What is the most possible label for c?

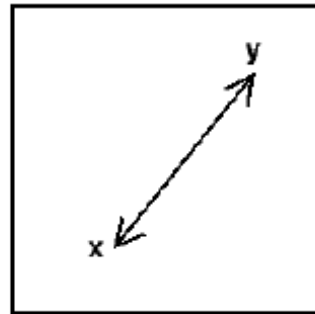
- Solution: Looking for the nearest K neighbors of c.
- Take the majority label as c's label
- Let's suppose $k = 3$:



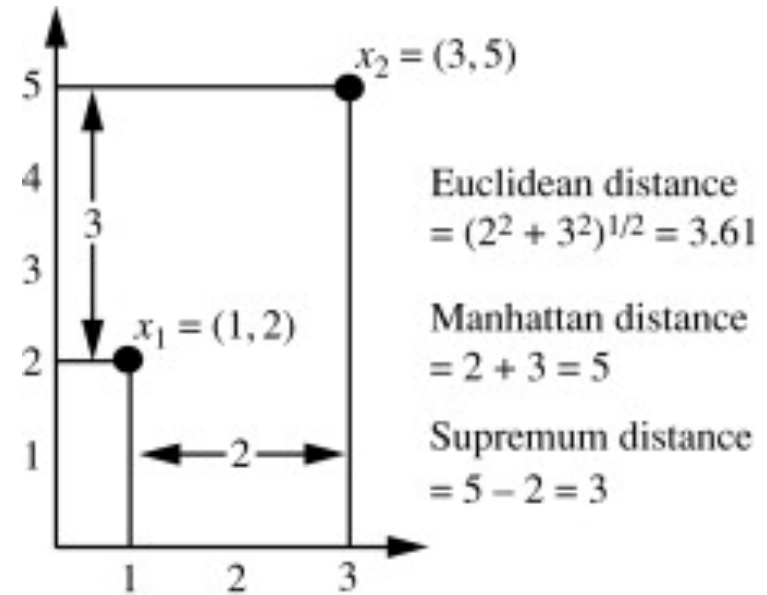
Distance Function Measurements



Manhattan



Euclidean

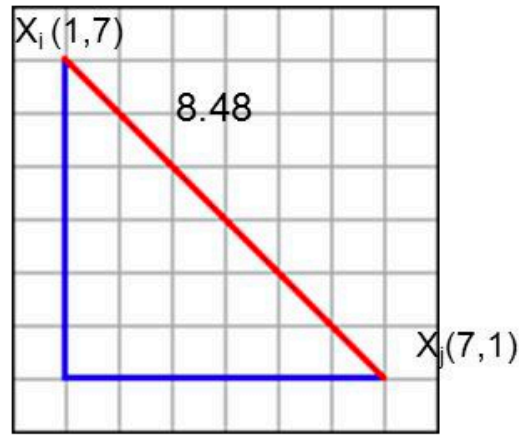


Minkowski Distance

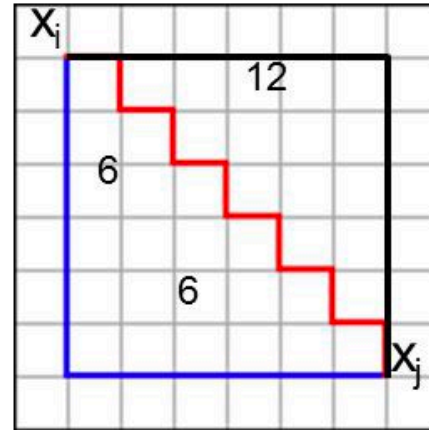
- Minkowski distance: a generalization

$$d(i, j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q} \quad (q > 0)$$

- If $q = 2$, d is Euclidean distance
- If $q = 1$, d is Manhattan distance

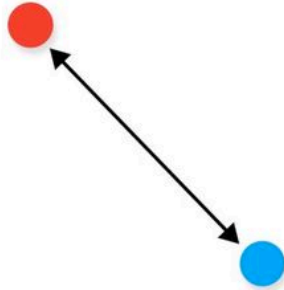


$q=2$

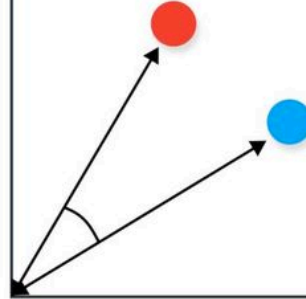


$q=1$

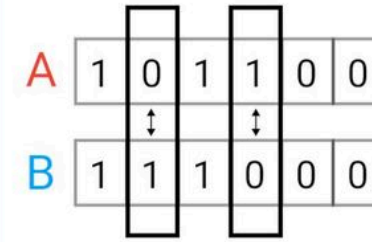
Euclidean



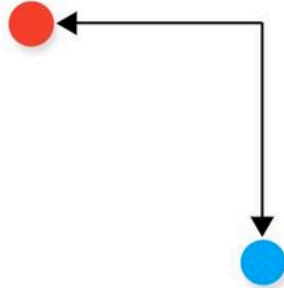
Cosine



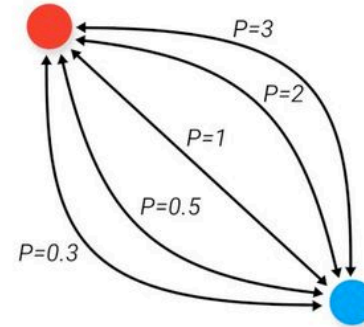
Hamming



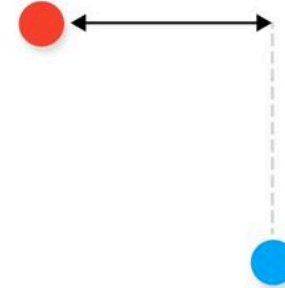
Manhattan



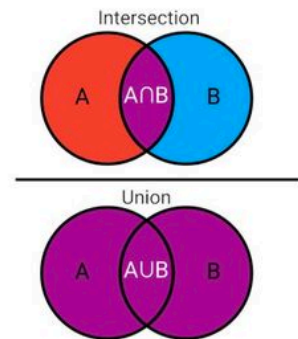
Minkowski



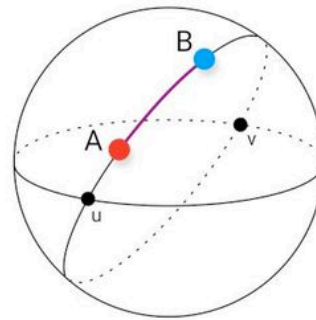
Chebyshev



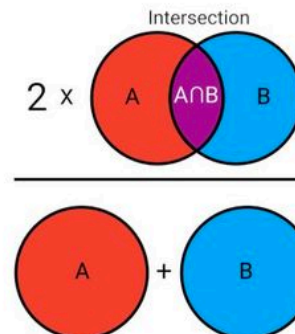
Jaccard



Haversine



Sørensen-Dice



Example

ID	Height	Age	class
1	5	45	H
2	5.11	26	L
3	5.6	30	M
4	5.9	34	M
5	4.8	40	H
6	5.8	36	M
7	5.3	19	L
8	5.8	28	M
9	5.5	23	L
10	5.6	32	M
11	5.5	38	

Example

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

kNN for Classification

- A simple implementation of **KNN classification** is a majority voting mechanism.
- Given: training examples $D=\{x_i, y_i\}$, and a new example x where
 - x_i : attribute-value **representation** of the example i^{th}
 - y_i : corresponding label or class of example i^{th}
- Algorithm:
 - Compute distance $D(x, x_i)$ for every x_i of the training data D
 - Select k closest instances x_{i_1}, \dots, x_{i_k} with their labels are y_{i_1}, \dots, y_{i_k}
 - **$y = \text{majority}(y_{i_1}, \dots, y_{i_k})$** is the predicted label of x

Example

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

kNN for Regression

- A simple implementation of **KNN regression** is to calculate the average of the numerical target of the K nearest neighbors.
- Given:
 - training examples $\{x_i, y_i\}$
 - x_i ... attribute-value representation of examples
 - y_i ... real-valued target (profit, rating on YouTube, etc)
 - testing point x that we want to predict the target
 - Algorithm:
 - compute distance $D(x, x_i)$ to every training example x_i
 - select k closest instances $x_{i1} \dots x_{ik}$ and their labels $y_{i1} \dots y_{ik}$
 - output the mean of $y_{i1} \dots y_{ik}$:
$$\hat{y} = f(x) = \frac{1}{k} \sum_{j=1}^k y_{i_j}$$

Example

ID	Height	Age	Weight	
1	5	45	77	H
2	5.11	26	47	L
3	5.6	30	55	M
4	5.9	34	59	M
5	4.8	40	72	H
6	5.8	36	60	M
7	5.3	19	40	L
8	5.8	28	60	M
9	5.5	23	45	L
10	5.6	32	58	M
11	5.5	38	?	

Distance-weighted k -NN

- Replace $\hat{f}(q) = \arg \max_{v \in V} \sum_{i=1}^k \delta(v, f(x_i))$ by:

$$\hat{f}(q) = \arg \max_{v \in V} \sum_{i=1}^k \frac{1}{d(x_i, x_q)^2} \delta(v, f(x_i))$$

Issues with Distance Metrics

- Most distance measures were designed for linear/real-valued attributes
- Two important questions in the context of machine learning:
 - How to handle nominal attributes
 - What to do when attribute types are mixed

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Hamming Distance

- For category variables, Hamming distance can be used.

Hamming Distance

$$D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

Normalization

Age	Loan	Default	Distance	
25	\$40,000	N	102000	
35	\$60,000	N	82000	
45	\$80,000	N	62000	
20	\$20,000	N	122000	
35	\$120,000	N	22000	2
52	\$18,000	N	124000	
23	\$95,000	Y	47000	
40	\$62,000	Y	80000	
60	\$100,000	Y	42000	3
48	\$220,000	Y	78000	
33	\$150,000	Y	8000	1
48	\$142,000	?		

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Normalization

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

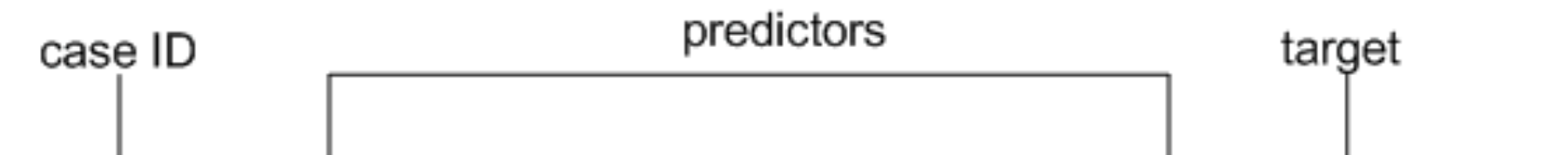
Exercise

Customer	Age	Loan	Default	E
John	25	40000	N	
Smith	35	60000	N	
Alex	45	80000	N	
Jade	20	20000	N	
Kate	35	120000	N	
Mark	52	18000	N	
Anil	23	95000	Y	
Pat	40	62000	Y	
George	60	100000	Y	
Jim	48	220000	Y	
Jack	33	150000	Y	
Andrew	48	142000	?	

What to do when attribute types are mixed

- Convert categorical values into numerical values
 - Binary values: convert to 0 and 1, for example 'male', 'female'
 - Multiple degrees, such as 'low', 'average', and 'high': convert to 1, 2, 3
 - if the values are "Red", "Green", "Blue" (or more generally, something that has no intrinsic order): convert to [1,0,0], [0,1,0], [0,0,1]
- Normalize or scale the data into the same interval.

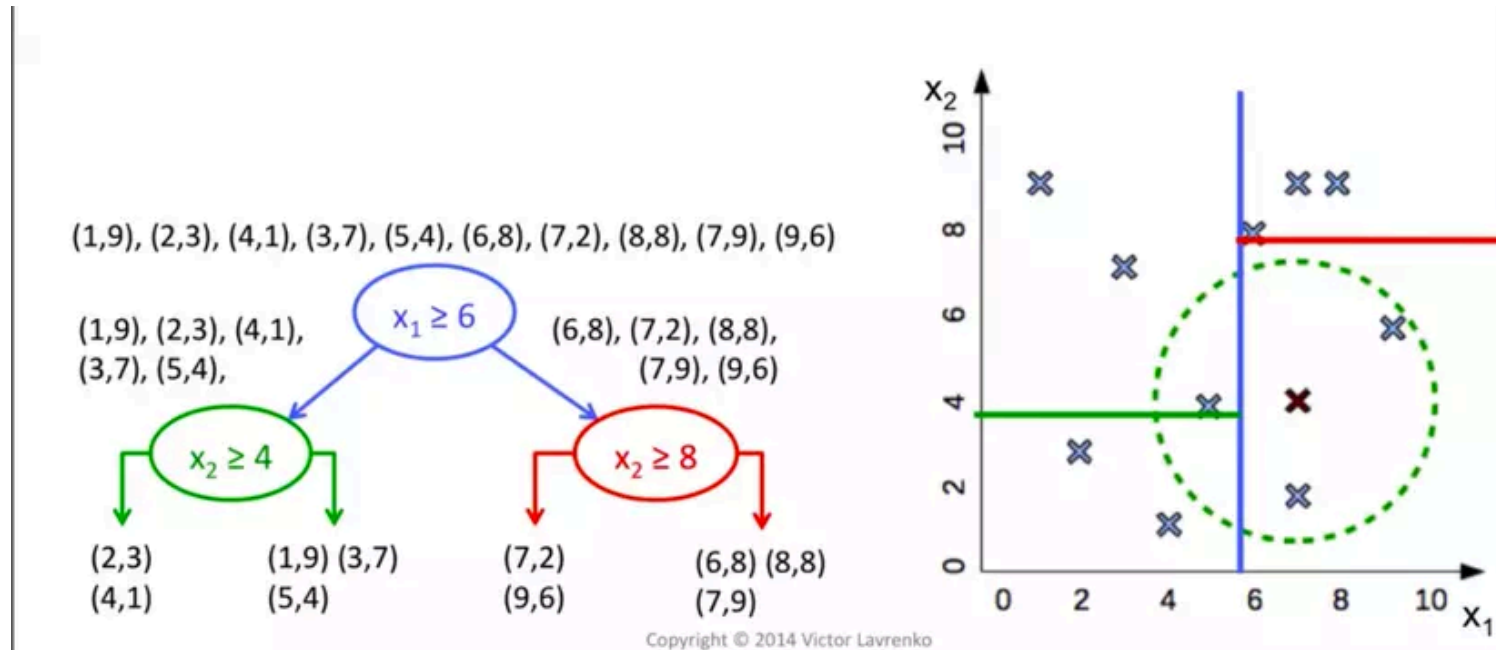
Exercise



The diagram illustrates the structure of the data table. A bracket labeled "case ID" points to the CUST_ID column. A larger bracket labeled "predictors" spans the CUST_GENDER, EDUCATION, OCCUPATION, and AGE columns. A bracket labeled "target" points to the AFFINITY_CARD column.

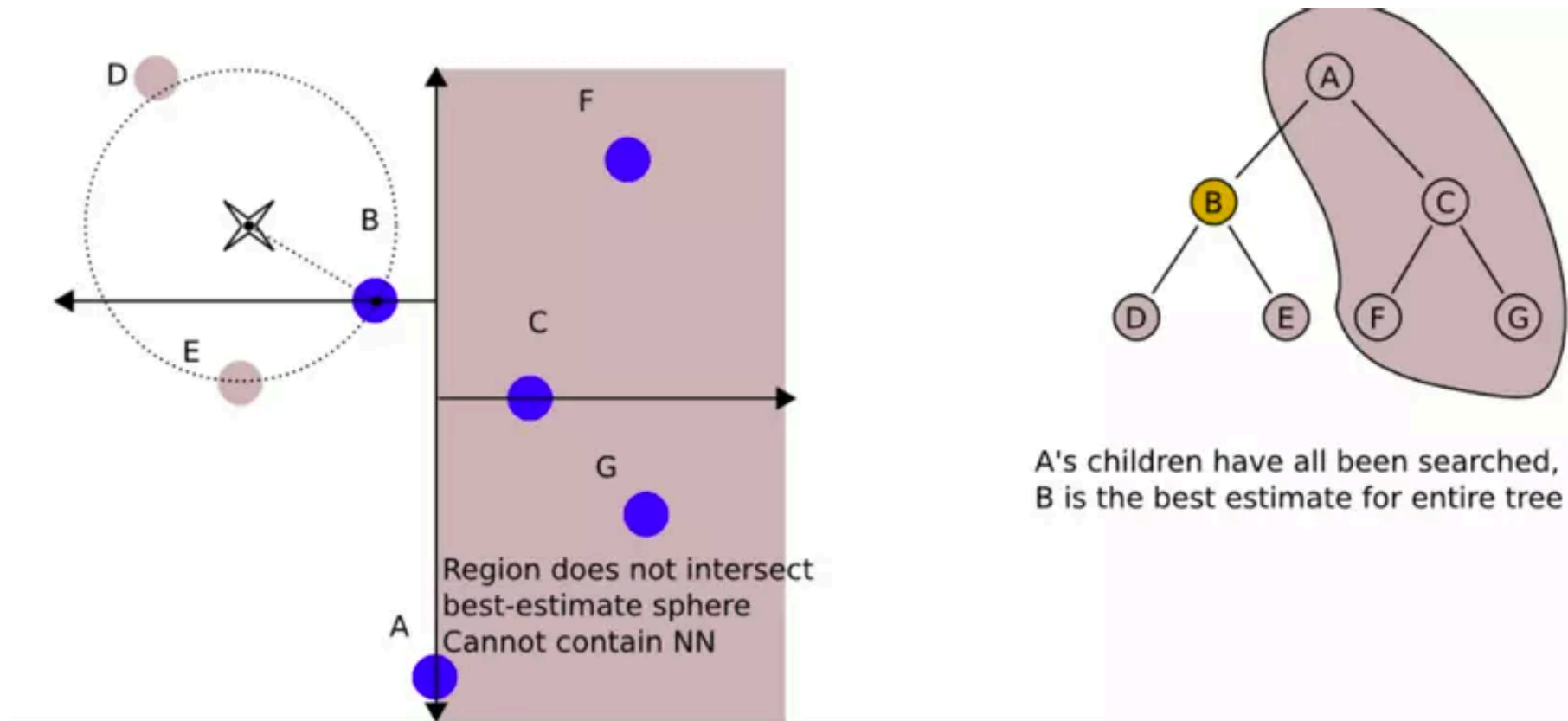
CUST_ID	CUST_GENDER	EDUCATION	OCCUPATION	AGE	AFFINITY_CARD
101501	F	Masters	Prof.	41	0
101502	M	Bach.	Sales	27	0
101503	F	HS-grad	Cleric.	20	0
101504	M	Bach.	Exec.	45	1
101505	M	Masters	Sales	34	1
101506	M	HS-grad	Other	38	0
101507	M	< Bach.	Sales	28	0
101508	M	HS-grad	Sales	19	0
101509	M	Bach.	Other	52	0
101510	M	Bach.	Sales	27	1

KDTree



<https://viblo.asia/p/gioi-thieu-thuat-toan-kd-trees-nearest-neighbour-search-RQqKLvjzl7z>

Nearest Neighbor via KDTree



Summary

- kNN can deal with complex and arbitrary decision boundaries.
- Despite its simplicity, researchers have shown that the classification accuracy of kNN can be quite strong and in many cases as accurate as those elaborated methods.
- kNN is slow at the classification time
- kNN does not produce an understandable model