

TRƯỜNG CNTT & TRUYỀN THÔNG

KHOA KHOA HỌC MÁY TÍNH

Phương pháp k láng giềng
K nearest neighbors - KNN

Nội dung

- **Giới thiệu về KNN**
- Kết luận và hướng phát triển

K nearest neighbors – K láng giềng

Phương pháp K láng giềng

Tell me
who your friends are
and I'll tell you
who you are



- Giới thiệu về KNN
- kết luận và hướng phát triển

K nearest neighbors – K láng giềng

Tell me
who your friends
are and I'll tell you
who you are

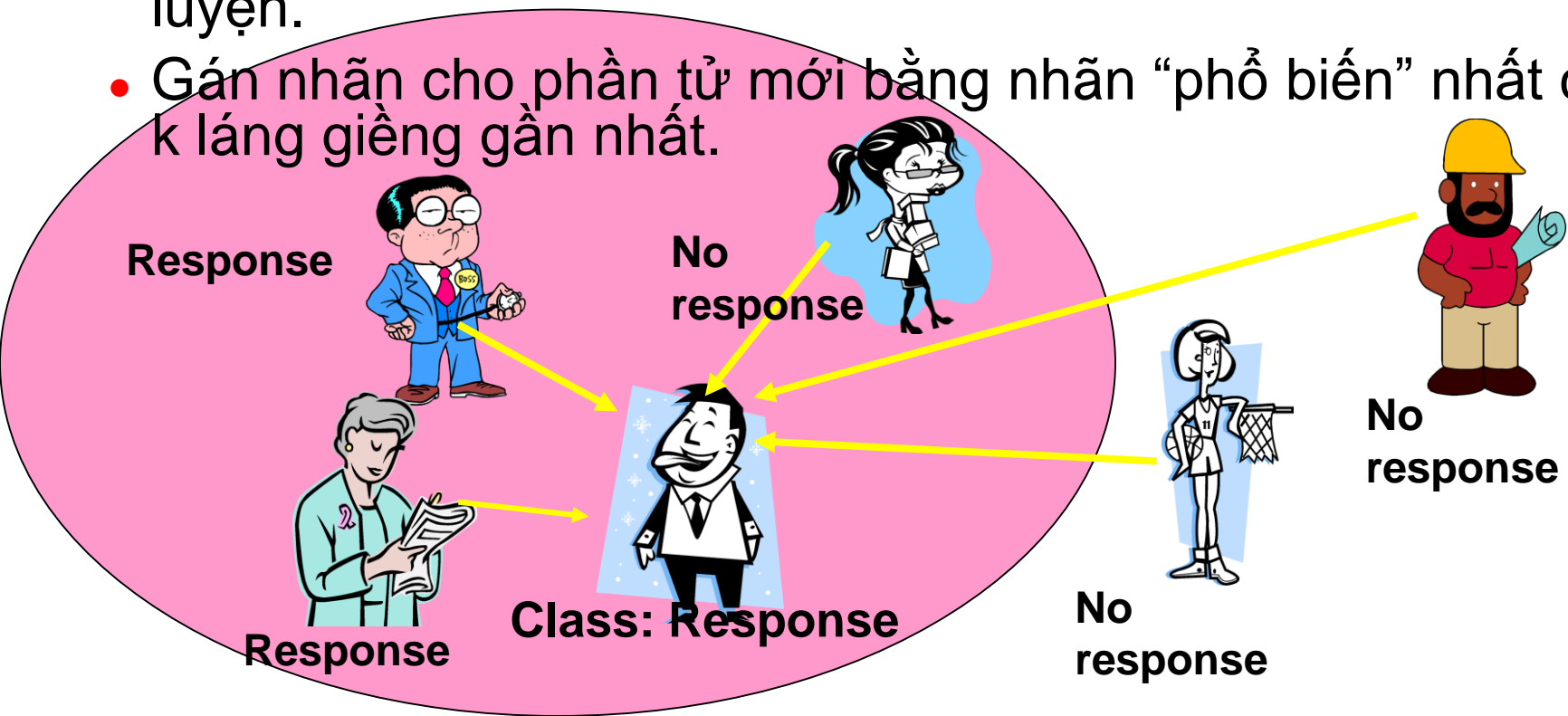


K nearest neighbors – K láng giềng

- **K láng giềng - KNN** là một thuật toán phân lớp (classification) các trường hợp mới đến dựa trên một số lượng thông tin “phổ biến” nhất của **k** láng giềng gần nhất với nó.
- K láng giềng (K-Nearest Neighbors) còn được gọi bằng các tên khác như Lazy Learning, Instance-Based Learning
- KNN được Fix và Hodges đề xuất năm 1952
- Học có giám sát (supervised classification)

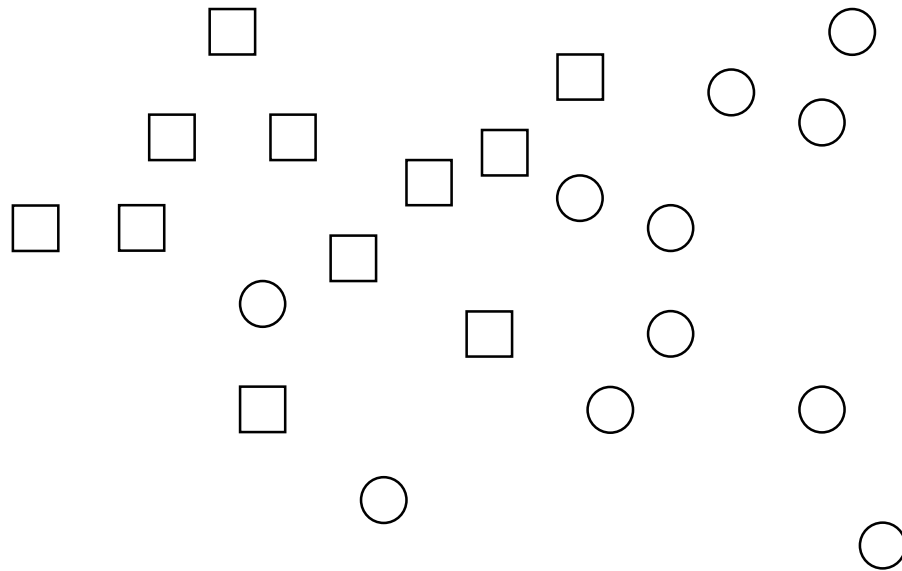
K-Nearest Neighbor – K láng giềng

- Để xác định được lớp của phần tử mới đến
 - Tính toán khoảng cách từ phần tử mới đến các phần tử còn lại trong tập huấn luyện.
 - Chọn K phần tử gần nhất với phần tử mới trong tập huấn luyện.
 - Gán nhãn cho phần tử mới bằng nhãn “phổ biến” nhất của k láng giềng gần nhất.



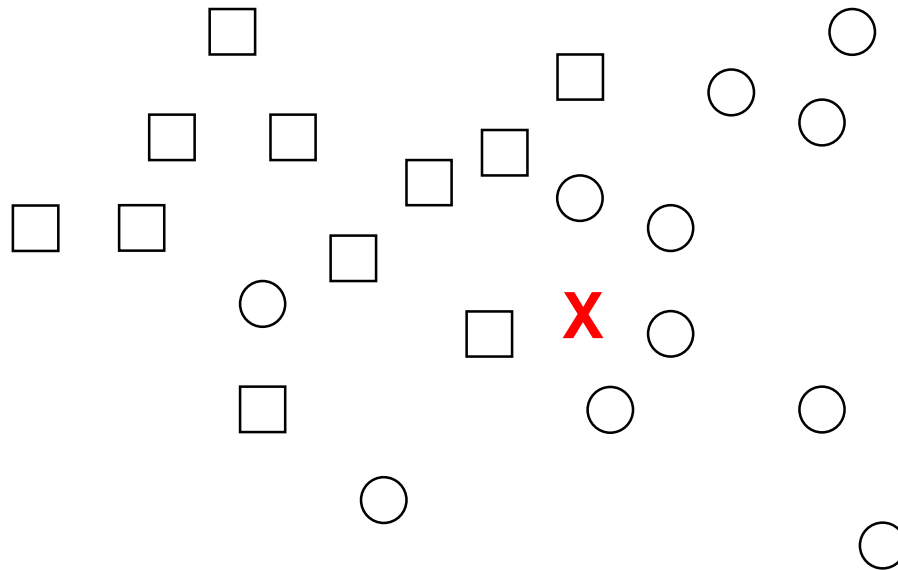
- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN



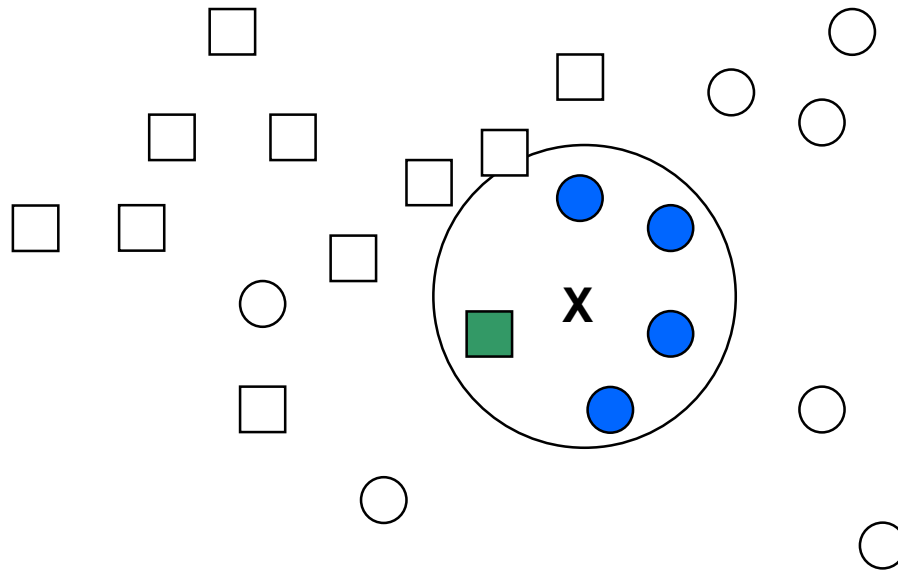
- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN



- Giới thiệu về KNN
- kết luận và hướng phát triển

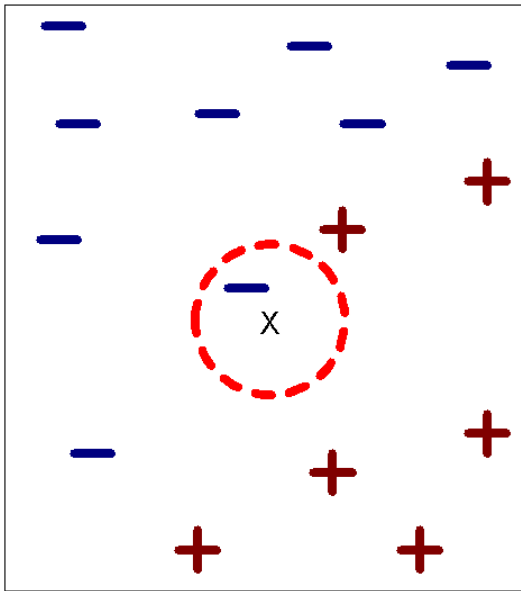
Phương pháp KNN



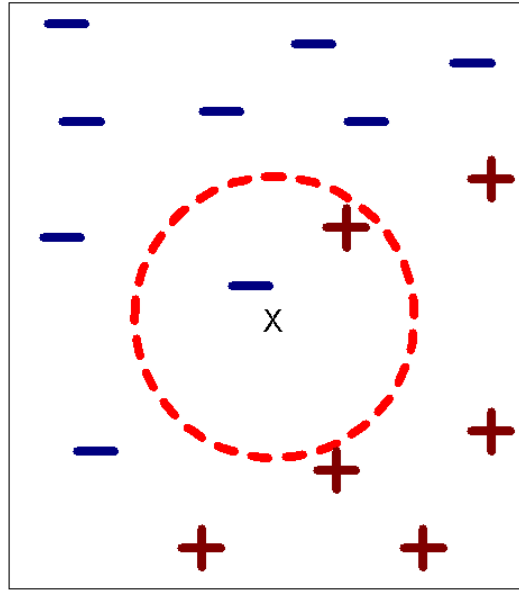
K = ?

Nhãn của phần tử mới đến là hình vuông hay hình tròn?

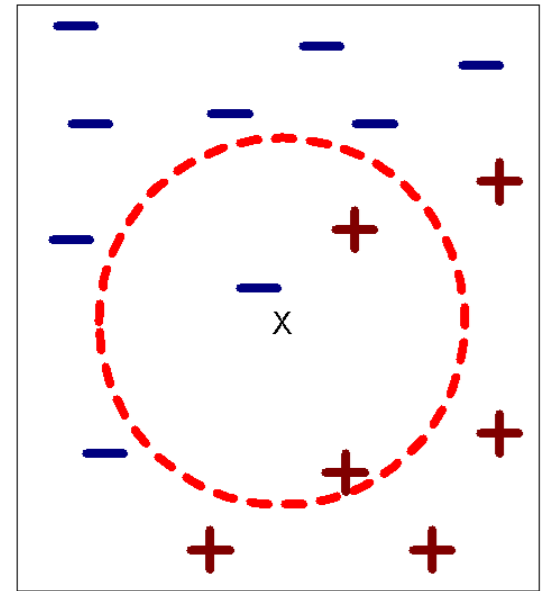
K-Nearest-Neighbor Strategy



(a) 1-nearest neighbor



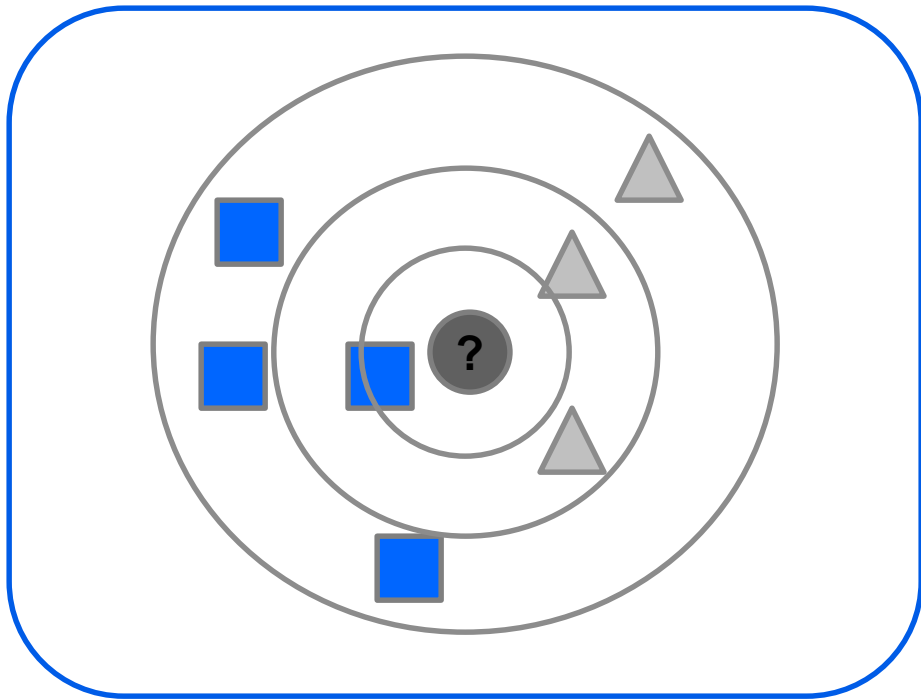
(b) 2-nearest neighbor



(c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Phương pháp KNN



□ Phần tử mới xuất hiện thuộc nhóm nào khi

□ $k = 1$:

□ $k = 3$:

□ $k = 7$

□ 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
- Choose an odd value for k , to eliminate ties

Các độ đo khoảng cách

- Khoảng cách được tính theo từng kiểu của dữ liệu
 - Kiểu số,
 - Kiểu rời rạc (nominal type),
 - Nhị phân

Các độ đo khoảng cách - Kiểu số

- Khoảng cách *Minkowski*

$$d(i, j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \dots + |x_{i_p} - x_{j_p}|^q)}$$

$i = (x_{i_1}, x_{i_2}, \dots, x_{i_p})$ và $j = (x_{j_1}, x_{j_2}, \dots, x_{j_p})$ là 2 phần tử dữ liệu trong p -dimensional, q là số nguyên dương

- nếu $q = 1$, d là khoảng cách Manhattan

$$d(i, j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \dots + |x_{i_p} - x_{j_p}|$$

- nếu $q = 2$, d là khoảng cách Euclid

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

Các độ đo khoảng cách – Kiểu rời rạc (nominal type)

- VD: thuộc tính color có giá trị là red, green, blue, etc.
- phương pháp so khớp đơn giản,
 - m là số lượng trùng khớp và
 - p là tổng số biến (thuộc tính),
 - khoảng cách được định nghĩa :

$$d(i, j) = \frac{p - m}{p}$$

Kiểu rời rạc (nominal type)

$$d(i, j) = \frac{p - m}{p}$$

- m là số lượng trùng khớp và
- p là tổng số biến (thuộc tính),

	Màu tóc	Màu mắt	Chiều cao	Cân nặng	Trình độ
Nam	Đen	Đen	Cao	Trung bình	Cao đẳng
Lan	Nâu	Đen	Thấp	Trung bình	Đại học
Điệp	Nâu	Đen	Cao	Trung bình	Cao đẳng

$d(\text{Nam}, \text{Lan}) = ?$

$d(\text{Nam}, \text{Điệp}) = ?$

K nearest neighbors – K láng giềng

Cách tính khoảng cách giữa các phần tử

- Mỗi phần tử được biểu diễn bởi tập hợp các thuộc tính



John:

Tuổi = 35

Thu nhập = 95K

Số thẻ tín dụng = 3



Mary:

Tuổi = 41

Thu nhập = 215K

Số thẻ tín dụng = 2

- Sử dụng khoảng cách Euclide giữa 2 phần tử.

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

Khoảng cách (John, Mary) =

$$\text{sqrt} [(35-41)^2 + (95\text{K}-215\text{K})^2 + (3-2)^2]$$

K nearest neighbors – K láng giềng

- **K = 3 : 3 nearest neighbors**

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Mary	22	50K	2	Yes
Hannah	63	200K	1	No
Tom	59	170K	1	No
Nellie	25	40K	4	Yes
David	37	50K	2	?

K nearest neighbors – K láng giềng

- K = 3 : 3 nearest neighbors**

Customer	Age	Income (K)	No. cards	Resp
John	35	35	3	No
Mary	22	50	2	Yes
Hannah	63	200	1	No
Tom	59	170	1	No
Nellie	25	40	4	Yes
David	37	50	2	? = Yes

Distance from David

$$\text{sqrt} [(35-37)^2 + (35-50)^2 + (3-2)^2] = \mathbf{15.16}$$

$$\text{sqrt} [(22-37)^2 + (50-50)^2 + (2-2)^2] = \mathbf{15}$$

$$\text{sqrt} [(63-37)^2 + (200-50)^2 + (1-2)^2] = \mathbf{152.23}$$

$$\text{sqrt} [(59-37)^2 + (170-50)^2 + (1-2)^2] = \mathbf{122}$$

$$\text{sqrt} [(25-37)^2 + (40-50)^2 + (4-2)^2] = \mathbf{15.74}$$

K nearest neighbors

- Phương pháp KNN (tên khác instance-based, lazy)
 - rất đơn giản, **không có quá trình học**, Không có mô hình được xây dựng: Lưu trữ tất cả các dữ liệu huấn luyện
 - Hệ thống không thực hiện công việc gì cho đến khi có một trường hợp mới đến cần được phân loại. khi phân loại **mất nhiều thời gian so với một mô hình**, do quá trình tìm kiếm k dữ liệu lân cận, sau đó phân loại dựa trên majority vote (nếu là bài toán **hồi quy, giá trị dự báo tính dựa trên giá trị trung bình k láng giềng**)
 - kết quả phụ thuộc vào việc chọn **khoảng cách** sử dụng

K nearest neighbors

- phương pháp KNN
 - có thể làm việc trên **nhiều loại dữ liệu** khác nhau
 - giải quyết các vấn đề về **phân loại, hồi quy**
 - cho kết quả tốt, tuy nhiên **độ phức tạp** của quá trình phân loại **khá lớn**
 - được ứng dụng thành công trong hầu hết các lĩnh vực **tìm kiếm thông tin, nhận dạng, phân tích dữ liệu**, etc.

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

X1	X2	Lớp
0.45	5	?

X1	X2	Lớp
0.1	10	+1
0.2	25	+1
0.3	0	+1
0.5	11	-1
0.8	100	-1
0	50	+1
1	70	-1

D(Manhattan)
?
?
?
?
?
?
?

⇒ **Lớp ?**

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

X1	X2	Lớp
0.45	5	?

X1	X2	Lớp
0.1	10	+1
0.2	25	+1
0.3	0	+1
0.5	11	-1
0.8	100	-1
0	50	+1
1	70	-1

D(Manhattan)
5.35
20.25
5.15
6.05
95.35
45.45
65.55

1NN  lớp = +1

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

Height	Age	Weight
5.5	38	?

Height	Age	Weight
5	45	77
5.11	26	47
5.6	30	55
5.9	34	59
4.8	40	72
5.8	36	60
5.3	19	40

D(Manhattan)
?
?
?
?
?
?
?

⇒ **Weight=?**

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

Height	Age	Weight
5.5	38	?

Height	Age	Weight
5	45	77
5.11	26	47
5.6	30	55
5.9	34	59
4.8	40	72
5.8	36	60
5.3	19	40

D(Manhattan)
7.5
12.39
8.1
4.4
2.7
2.3
19.2

1NN  **Weight = 60**

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

- Giới thiệu về KNN
- kết luận và hướng phát triển

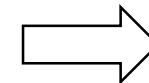
Phương pháp KNN

Height	Age	Weight
5.5	38	?

Height	Age	Weight
5	45	77
5.11	26	47
5.6	30	55
5.9	34	59
4.8	40	72
5.8	36	60
5.3	19	40

D(Manhattan)
7.5
12.39
8.1
4.4
2.7
2.3
19.2

3NN



Weight =
(59 + 72 + 60)/3
≈ 63.7

$$d(i, j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \dots + |x_{i_p} - x_{j_p}|$$

Nhận xét

- Thuộc tính X_2 có miền giá trị (0..100) trong khi thuộc tính X_1 có miền giá trị 0..1
- Kết quả phụ thuộc nhiều vào X_2
(chênh lệch X_2 lớn hơn so với X_1)
- nên chuẩn hóa dữ liệu
(chuẩn hóa thuộc tính X_2 về giá trị 0..1)

$$new_val = (val - min)/(max - min)$$

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

X1	X2	Lớp
0.45	5	?

X1	X2	Lớp
0.1	0.1	+1
0.2	0.25	+1
0.3	0	+1
0.5	0.11	-1
0.8	1	-1
0	0.5	+1
1	0.7	-1

1NN →

- Giới thiệu về KNN
- kết luận và hướng phát triển

Phương pháp KNN

X1	X2	Lớp
0.45	0.05	?

X1	X2	Lớp
0.1	0.1	+1
0.2	0.25	+1
0.3	0	+1
0.5	0.11	-1
0.8	1	-1
0	0.5	+1
1	0.7	-1

D(Manhattan)
0.4
0.45
0.2
0.11
1.3
0.9
1.2

1NN  **lớp = -1**

KNN Classification – Distance

Tuổi	Thu nhập	Nhãn	Khoảng cách
25	\$40,000	N	
35	\$60,000	N	
45	\$80,000	N	
20	\$20,000	N	
35	\$120,000	N	
52	\$18,000	N	
23	\$95,000	Y	
40	\$62,000	Y	
60	\$100,000	Y	
48	\$220,000	Y	
33	\$150,000	Y	

48

\$142,000

?

Euclidean Distance

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

Nội dung

- Giới thiệu về KNN
- **Kết luận và hướng phát triển**

Phương pháp KNN

- thường rất chính xác, nhưng chậm do phải duyệt qua dữ liệu để tìm phần tử gần
- giả sử các thuộc tính có độ quan trọng như nhau
 - gán trọng số quan trọng cho mỗi thuộc tính
- chịu đựng được nhiễu
- thống kê đã sử dụng k -NN từ những năm 50s
 - khi dữ liệu lớn ($n \rightarrow \infty$) và $k/n \rightarrow 0$, lỗi gần với giá trị nhỏ nhất

Hướng phát triển

- tăng tốc cho quá trình tìm kiếm phần tử lân cận
 - cấu trúc index
- chọn thuộc tính quan trọng
- gán trọng số cho các thuộc tính

The End