沒有想像中簡單的簡單分類器 KNN

kaggle-預測未來腳踏車租借數量

15/8/12 NCKU TienYang

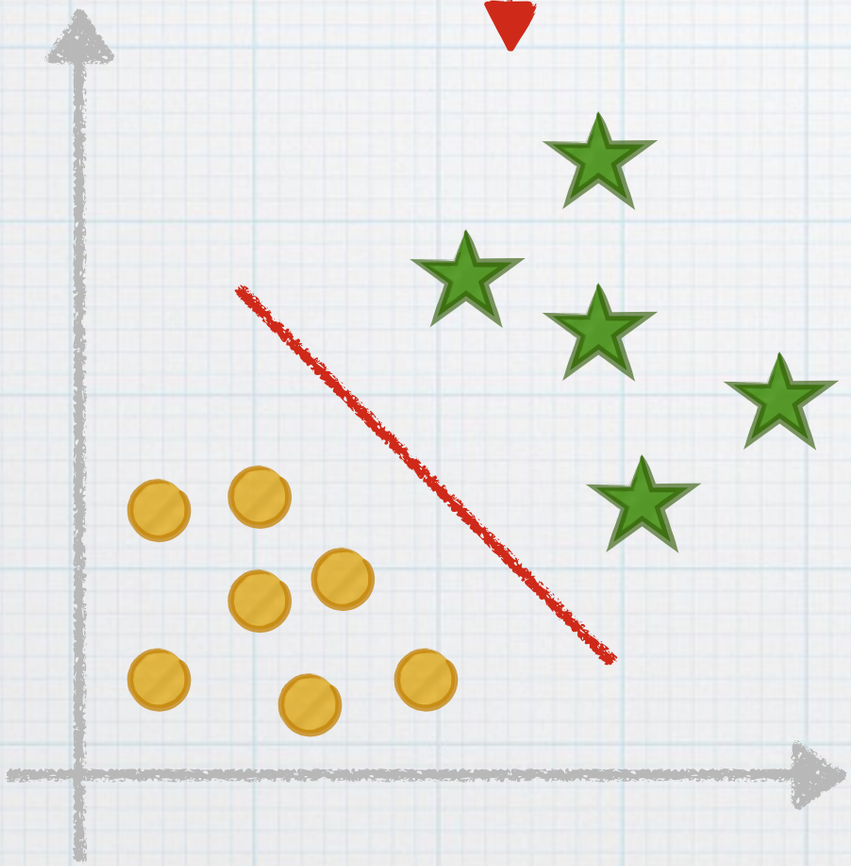
先來瞭解一下機器學習的方式可分為這兩種

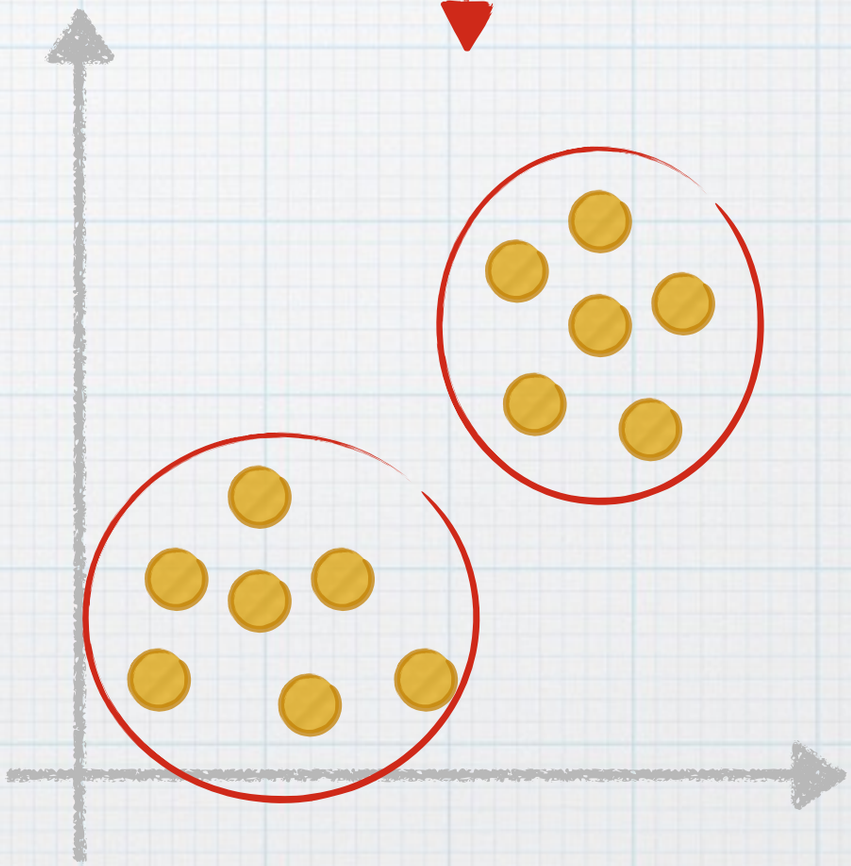
supervised learning

unsupervised learning

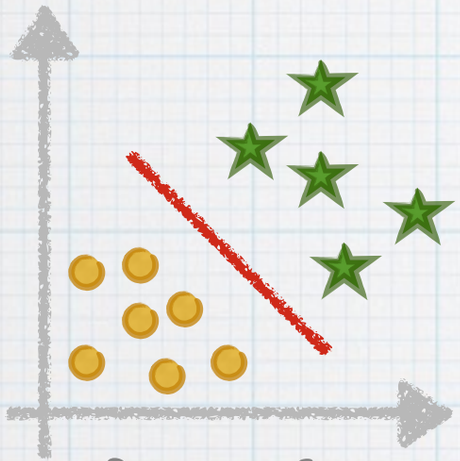
瞭解一下

supervised learning　unsupervised learning

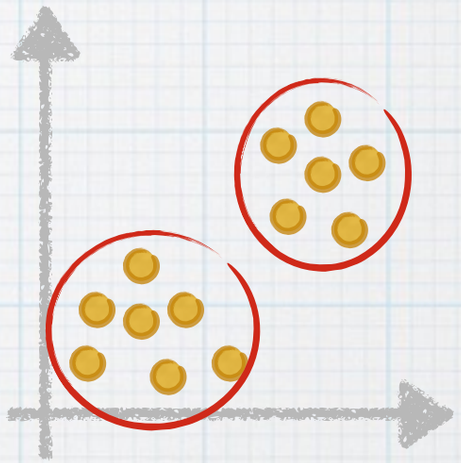




再來瞭解一下



classification v.s clustering



- Labeled data points
- Want a "rule" that assigns labels to new points
- Supervised learning

- Data is not labeled
- Group pointed that are "close" to each other
- Unsupervised learning

知道了

supervised learning v.s unsupervised learning

classification v.s clustering

知道了

supervised learning v.s unsupervised learning

classification v.s clustering

那什麼是KNN?

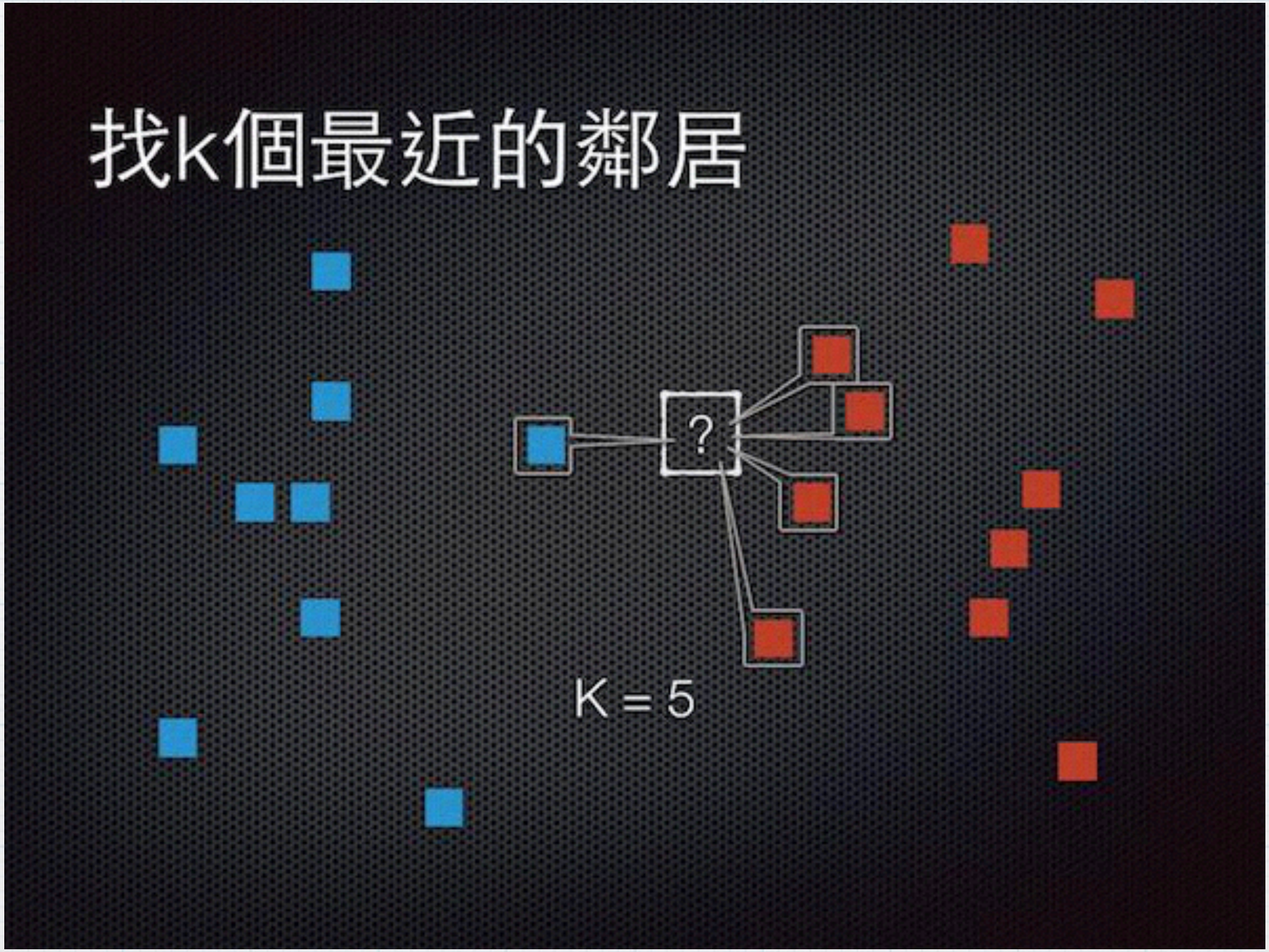
KNN

此演算法在2007年IEEE統計排名前十名資料採礦演算法之一，以目前來說是廣泛使用、非常有效而且是易於掌握的演算法。

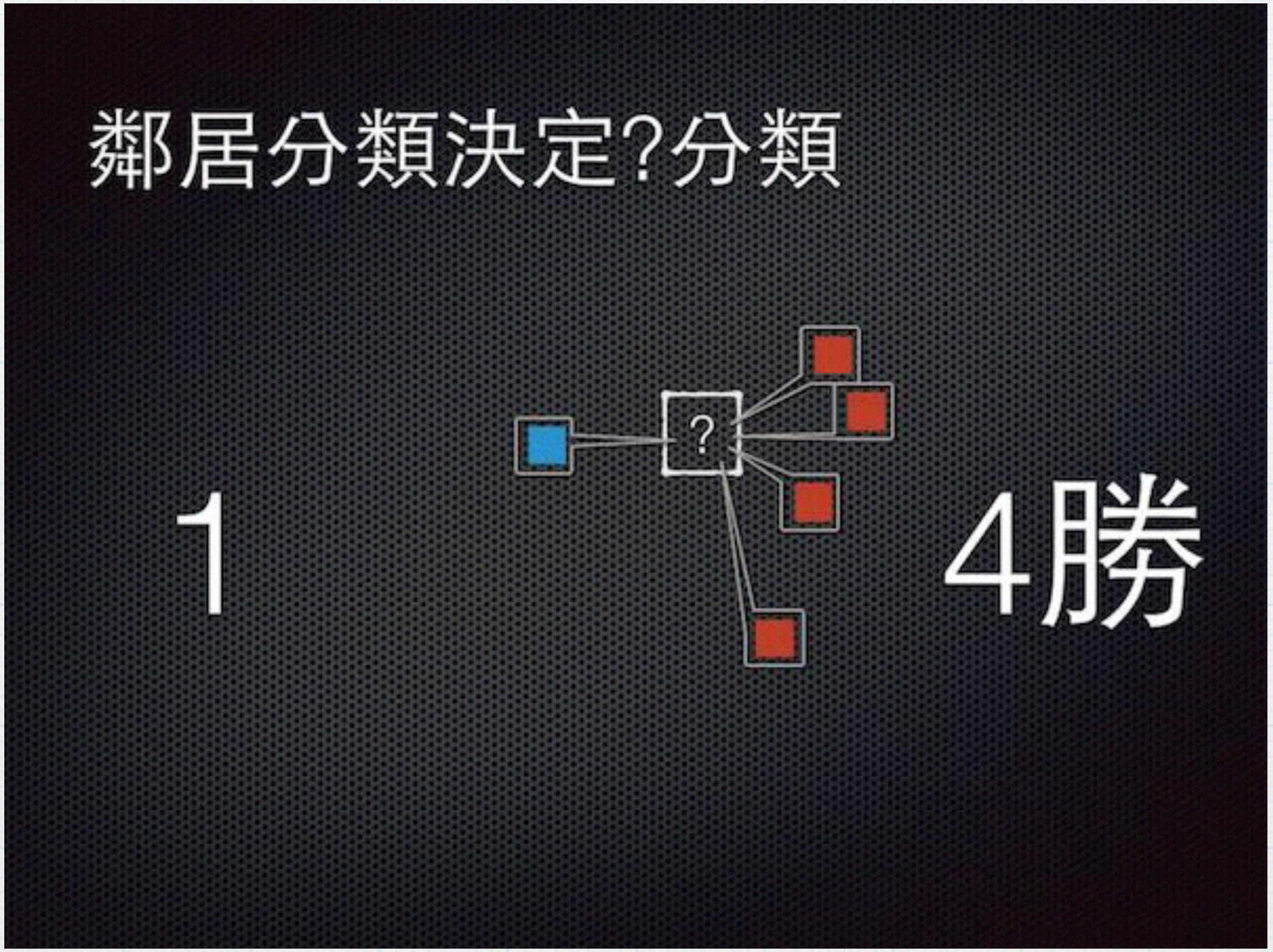
KNN

此演算法在2007年IEEE統計排名前十名資料採礦演算法之一，以目前來說是廣泛使用、非常有效而且是易於掌握的演算法。

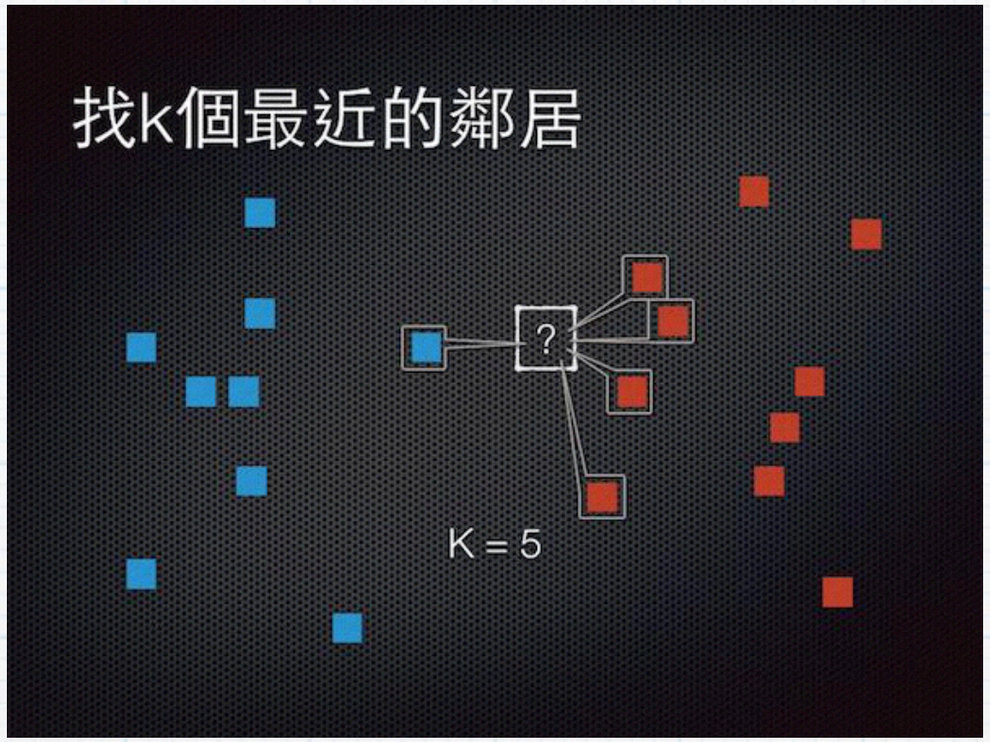
接下來給你看兩張圖你就知道什麼是KNN了！

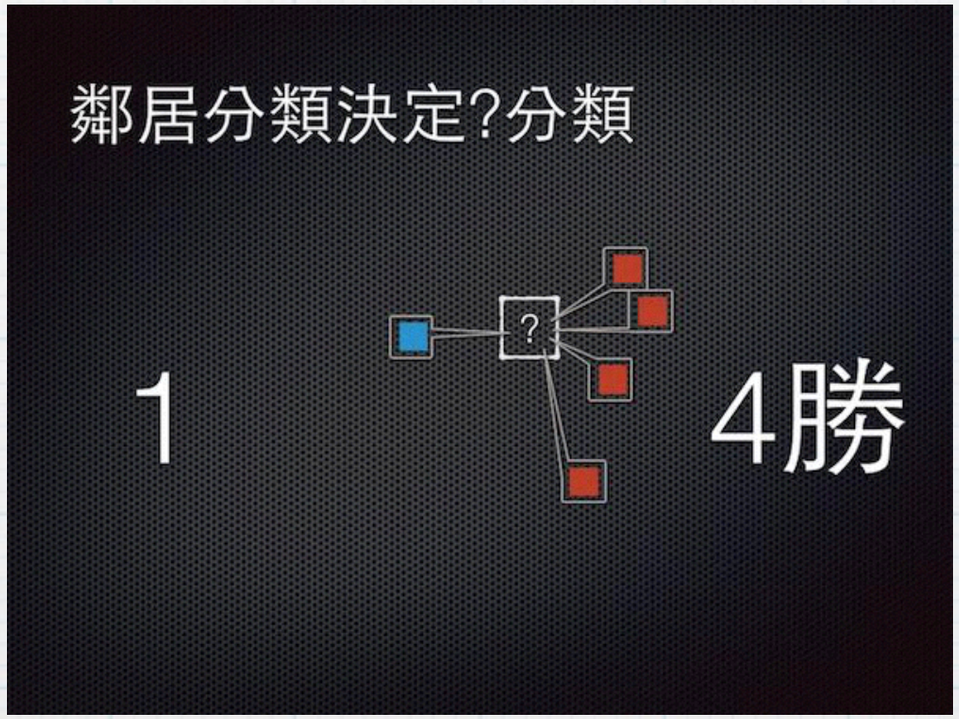

找k個最近的鄰居
?
K = 5

鄰居分類決定?分類



先





後

看起來簡單透了

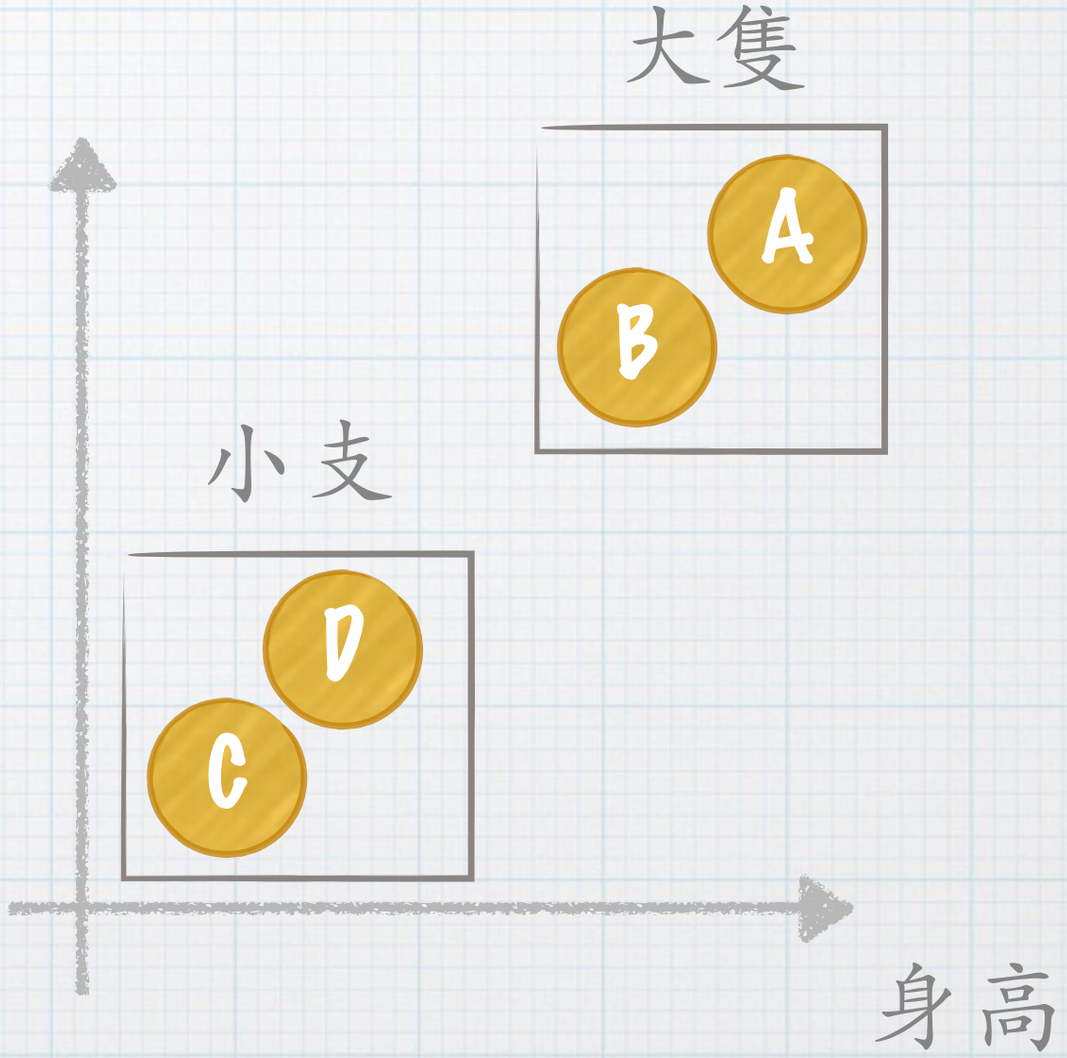
但我要怎樣找出

最近的鄰居？

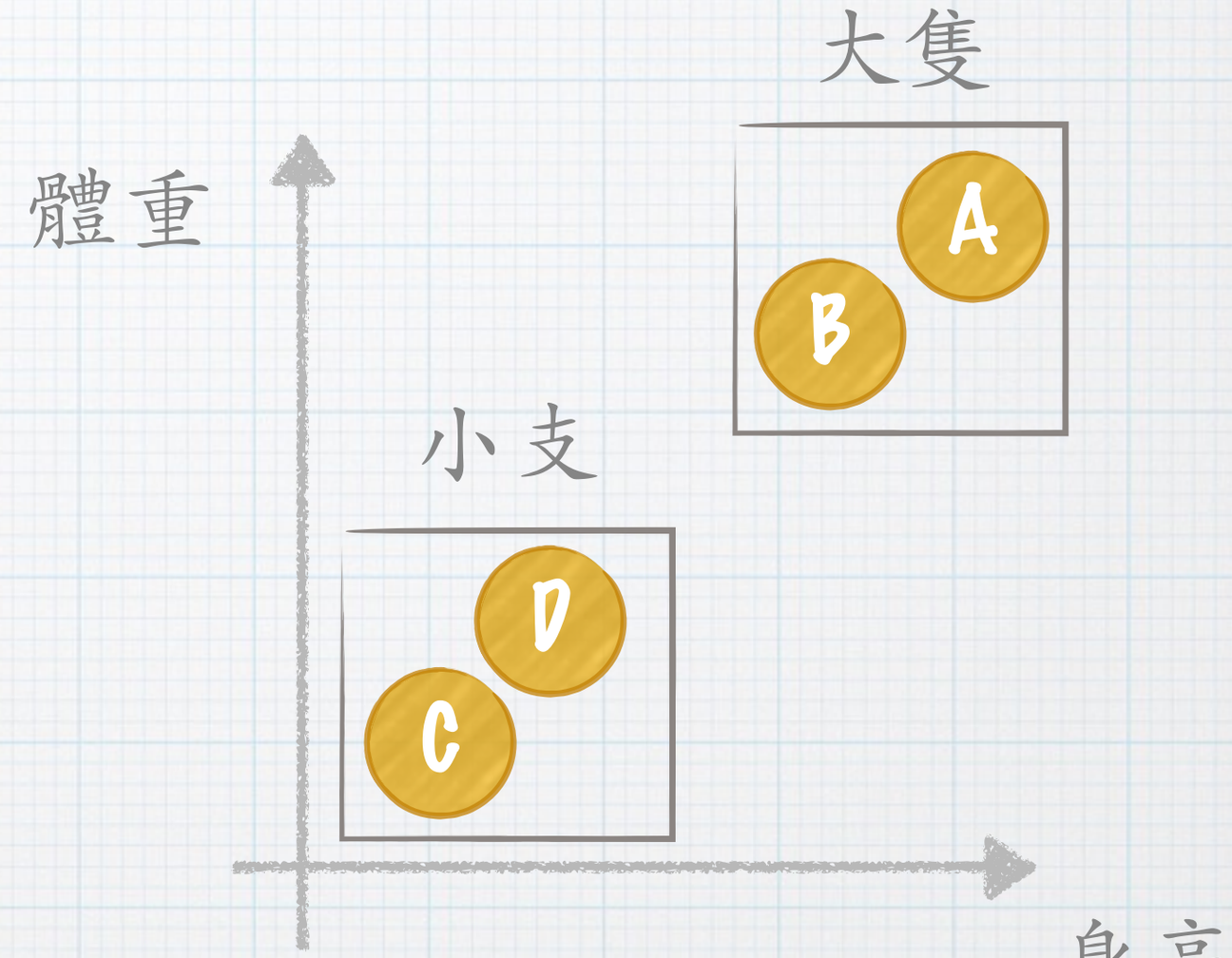
先來看一下資料

資料

	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支



	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

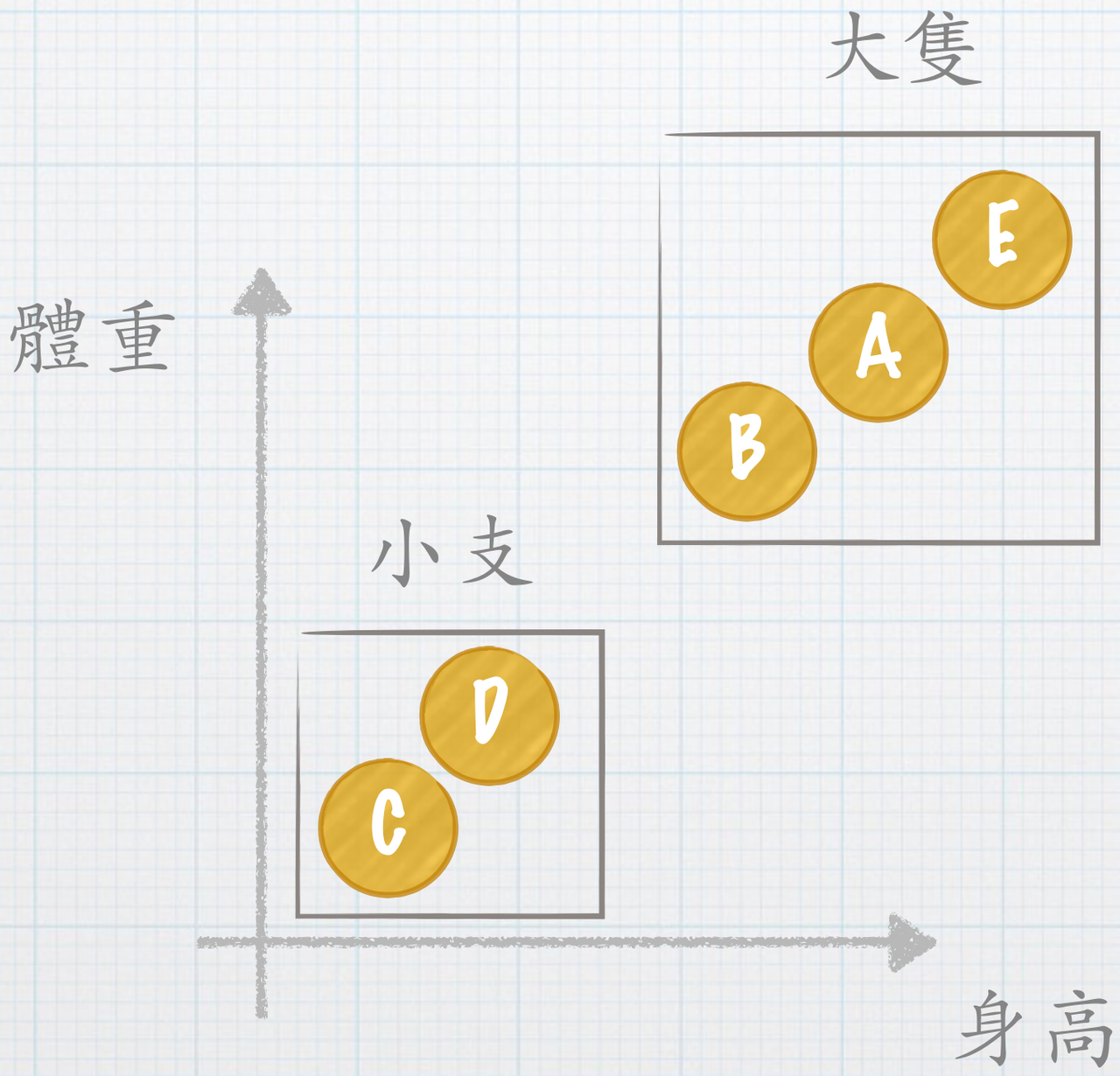


現在加入了一個 E

身高190 體重100 他會

是哪一類？

用膝蓋想都知道 E 是屬於大隻！



現在我們來教機器如何學習

1. 先找k個最近鄰居
2. 鄰居決定分類

先來找最近的K

K = 1

所以最近的是什麼類 E就是哪一類

Euclidean distance

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

E 身高190 體重100

E 和 A 的距離

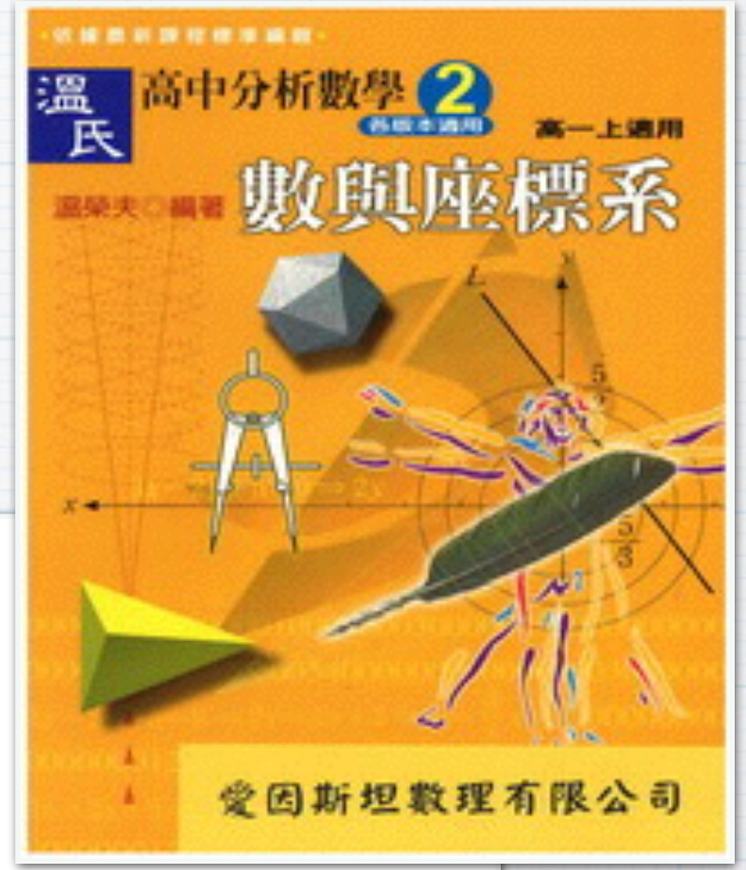
	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

$$\sqrt{(190-180)^2+(100-90)^2}$$

$$\rightarrow \sqrt{200}$$

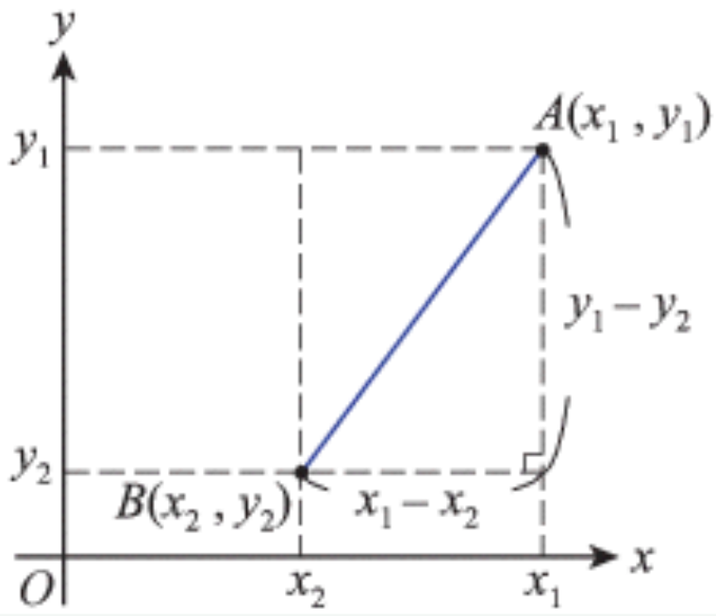
Euclidean distance

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$



距離公式

坐標平面上兩點 $A(x_1, y_1)$, $B(x_2, y_2)$，則 $\overline{AB} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。



誰最近呢？

	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

E 和大夥的距離

$\sqrt{200}$

$\sqrt{450}$

$\sqrt{3925}$

$\sqrt{3125}$

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

E 和 A 最近，所以A是大隻

那E也是大隻

你有想過剛剛是
怎樣找最近距離
的嗎？

是不是每個點都要算距離然後找最近的？

Brute Force

N samples in D dimensions

N = 4　　D = 2

	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

找出所有距離 N*D

找出最近的點 N

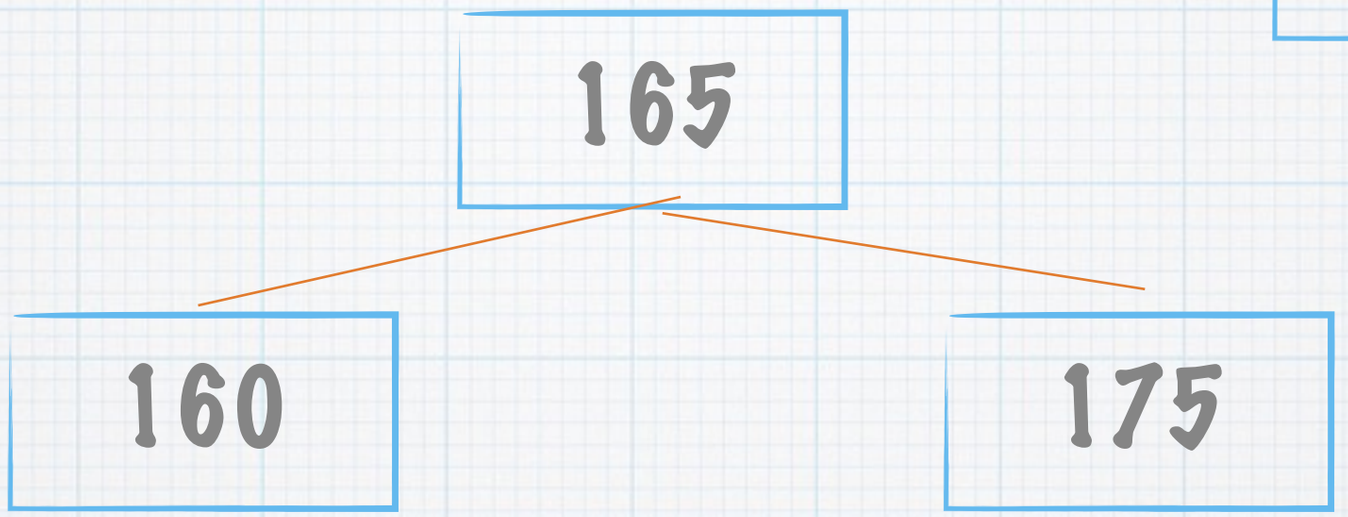
時間複雜度

O[DN²]

感覺還可以更快

一個一個找微蠢

K-D Tree

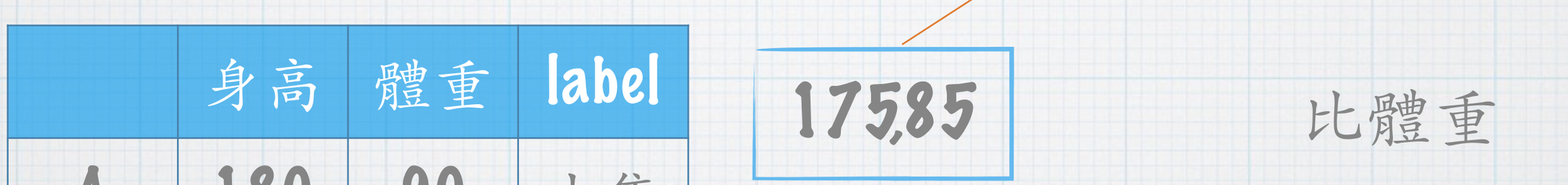


	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

190

180

K-D Tree



	身高	體重	label
A	180	90	大隻
B	175	85	大隻
C	160	45	小支
D	165	50	小支

時間複雜度

O[DNlogN]

KD-tree感覺頗威

但是當你的維度 D

太高時，建樹會見

到哭出來

救星來了 ball -tree

Where KD trees partition data along Cartesian axes, ball trees partition data in a series of nesting hyper-spheres.

問題來了？

什麼叫做最近？

為什麼算距離一定要是Euclidean distance

Euclidean distance
的缺點

A房子：200坪,用10年,10元的畫

B房子：50坪,用10年,10萬元的畫

C房子：199坪,用10年,9萬元的畫

Euclidean distance 的缺點

A房子：200坪,用10年,10元的畫

B房子：50坪,用10年,100000元的畫

C房子：199坪,用10年,90000元的畫

C房子離B房子比較近，所以B,C的房價一樣

A房子：200坪,用10年,10元的畫

B房子：50坪,用10年,10萬元的畫

C房子：199坪,用10年,9萬元的畫

C房子離B房子比較近，所以B,C的房價一樣

聽你在放屁，A,C的房價要差不多才對，因為坪數差不多

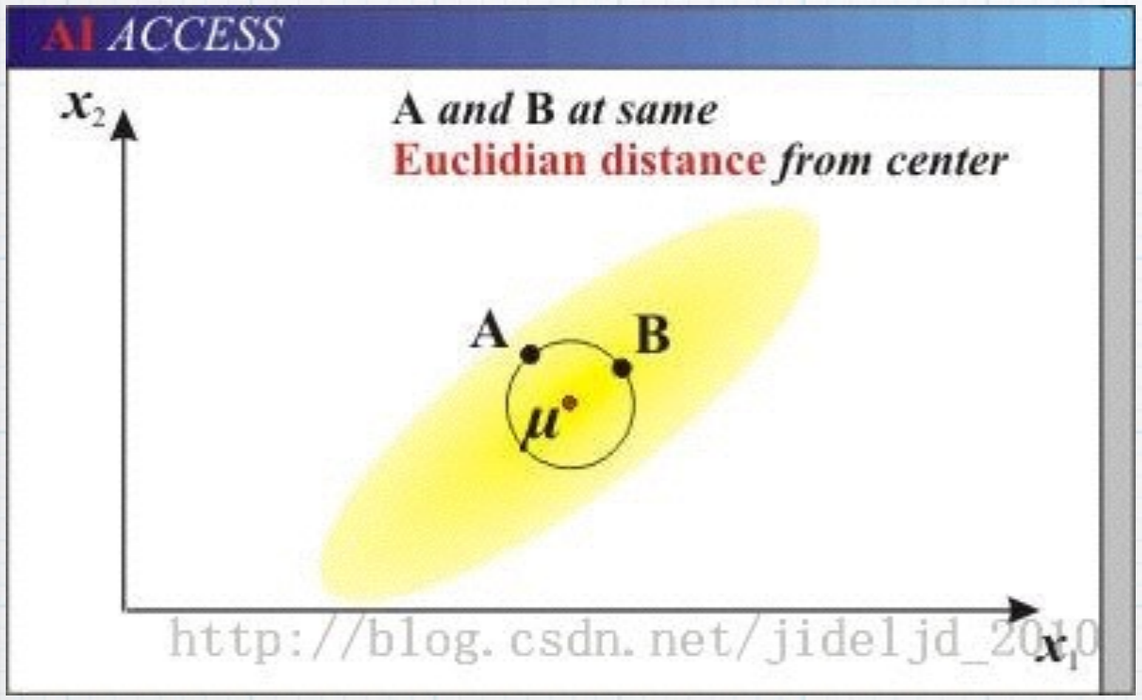
Euclidean distance 的缺點，不同維度會等同看待

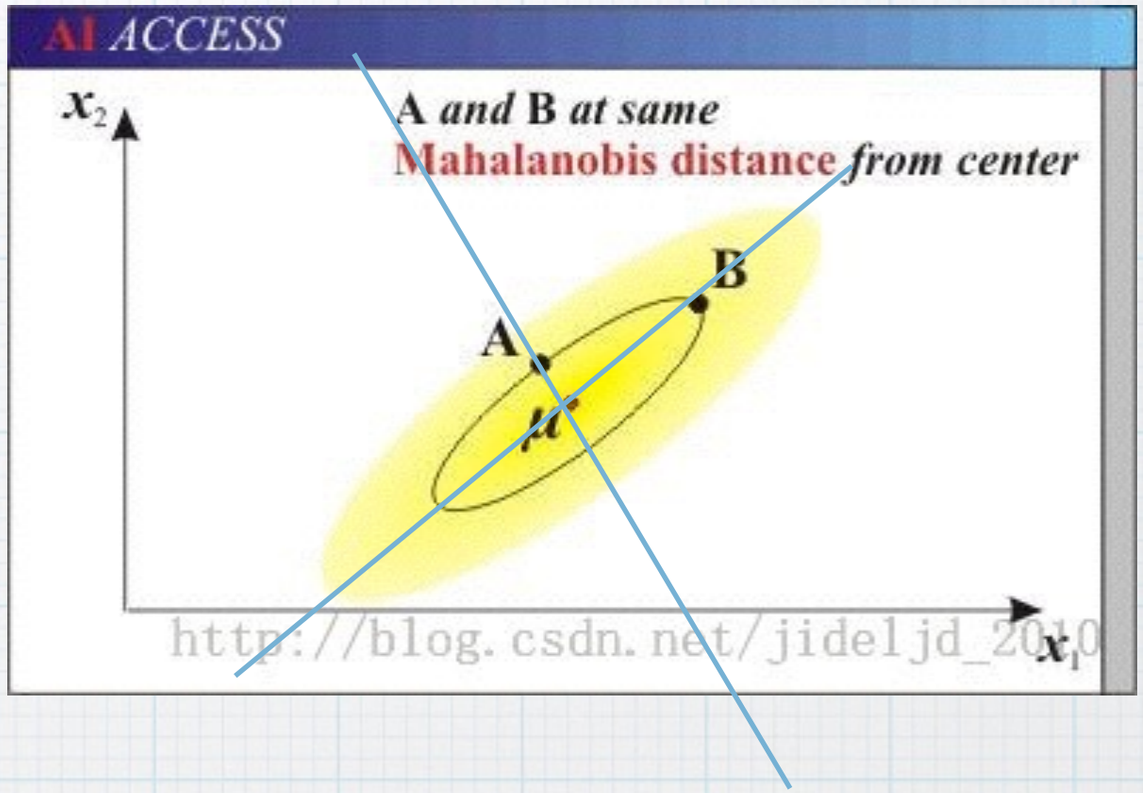
Mahalanobis distance降臨

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1} (x-\mu)}.$$

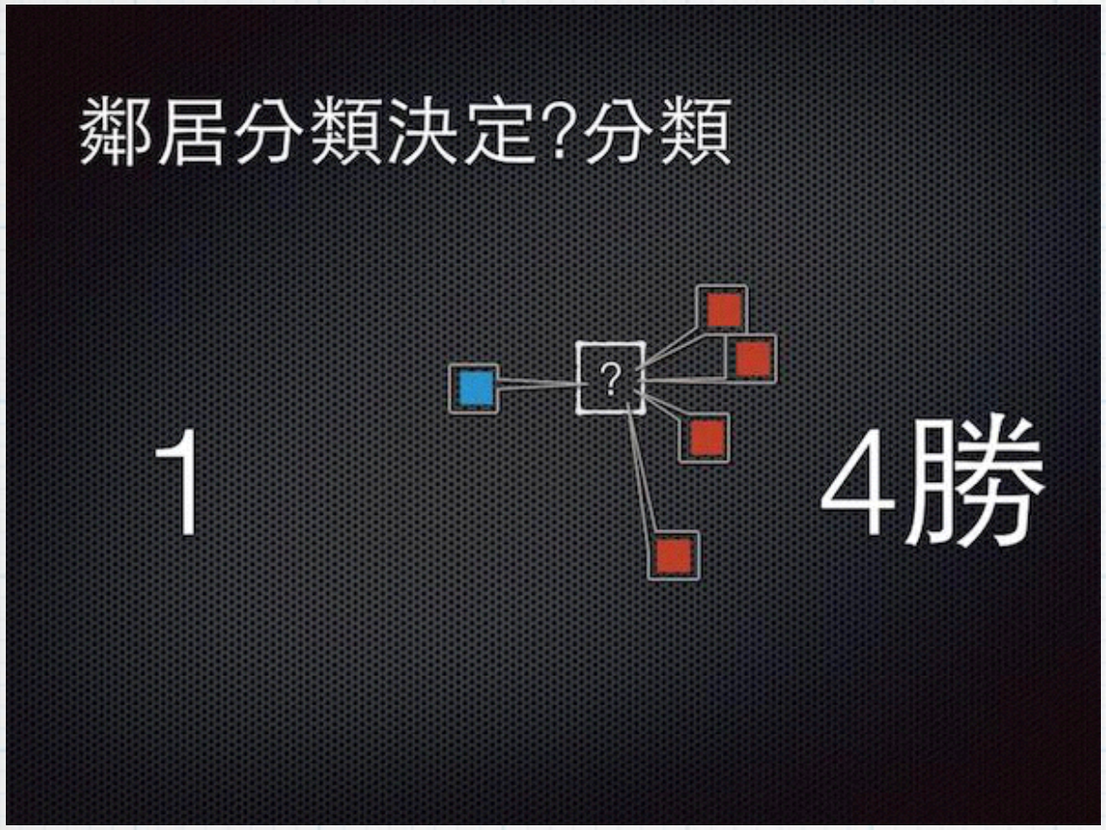
可以解決不同維度會等同看待的問題

Mahalanobis distance





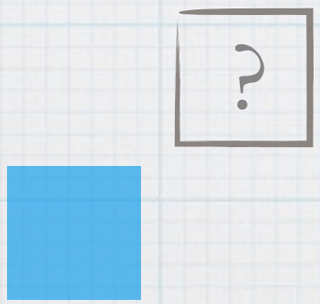
旋轉座標軸



我覺得雖然？

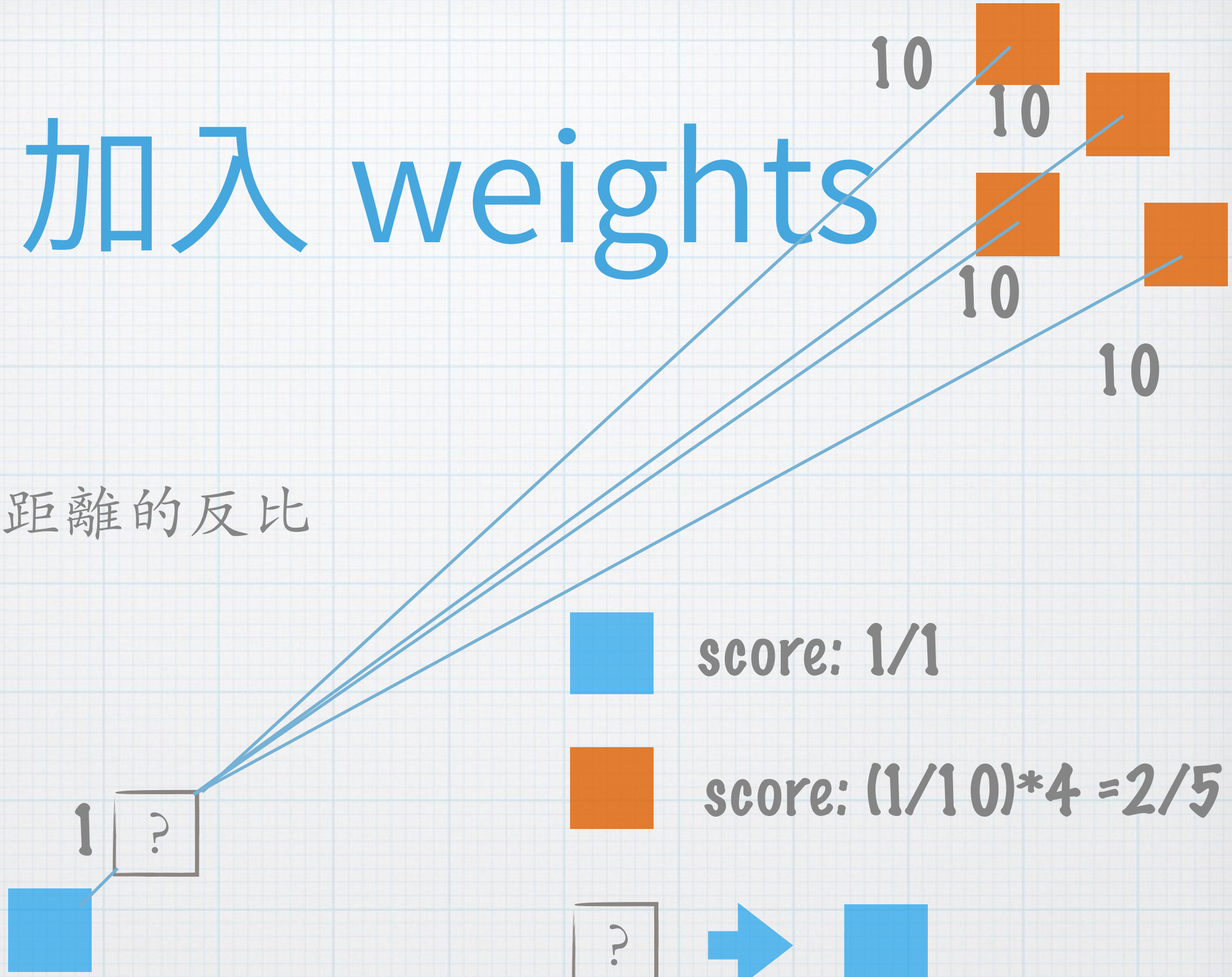
有四票，但是他離藍色真的太近了

這樣很不公平



我覺得這樣不太公平，應該要對距離近的比較好一點才對

加入 weights



weights:距離的反比

終於可以開始玩

DATA了

kaggle™

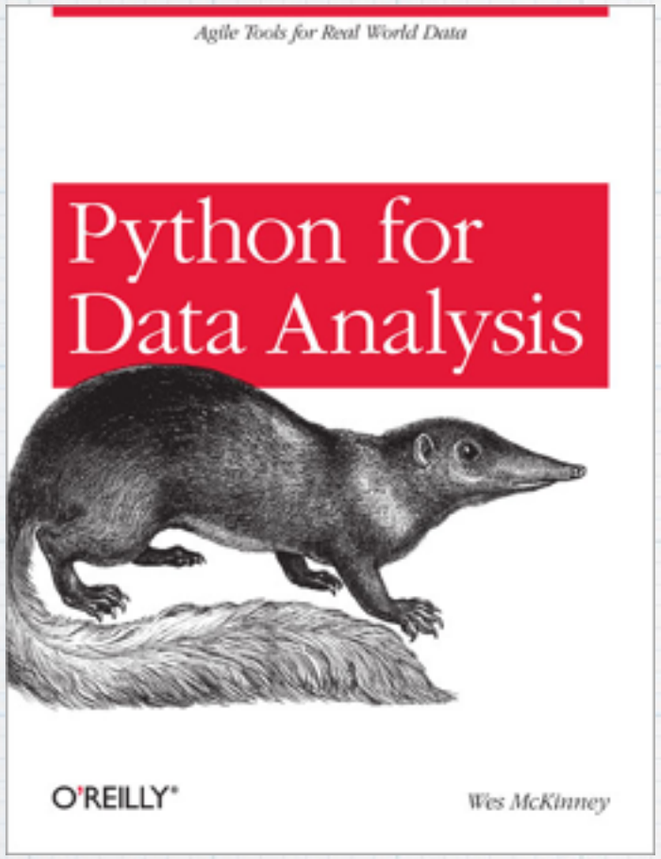
Completed • Knowledge • 3,252 teams

Bike Sharing Demand

Wed 28 May 2014 – Fri 29 May 2015 (2 months ago)

You must predict the total count of bikes rented during each hour

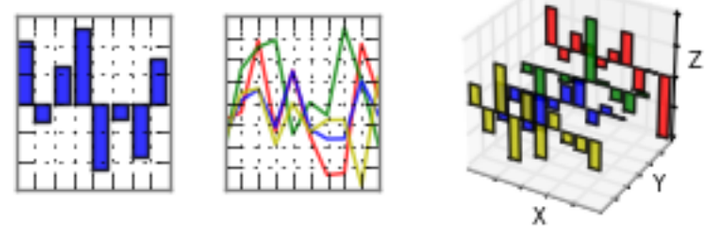
datetime	hourly date + timestamp
season	1 = spring, 2 = summer, 3 = fall, 4 = winter
weather	1: Clear, Few clouds, Partly cloudy, Partly cloudy
temp	temperature in Celsius
humidity	relative humidity
windspeed	wind speed
holiday	whether the day is considered a holiday
workingday	whether the day is neither a weekend nor holiday



資料分析

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



地表最強 # 資料探勘 & 機器學習

scikit learn

DEMO

Github

https://github.com/wy36101299/knn-Bike-Sharing-Demand

reference

scikit-learn
http://scikit-learn.org/stable/modules/neighbors.html

[Machine Learning] kNN分類演算法« Big O(1)
http://enginebai.logdown.com/posts/241676/knn

pandas: powerful Python data analysis toolkit
http://pandas.pydata.org/pandas-docs/stable/