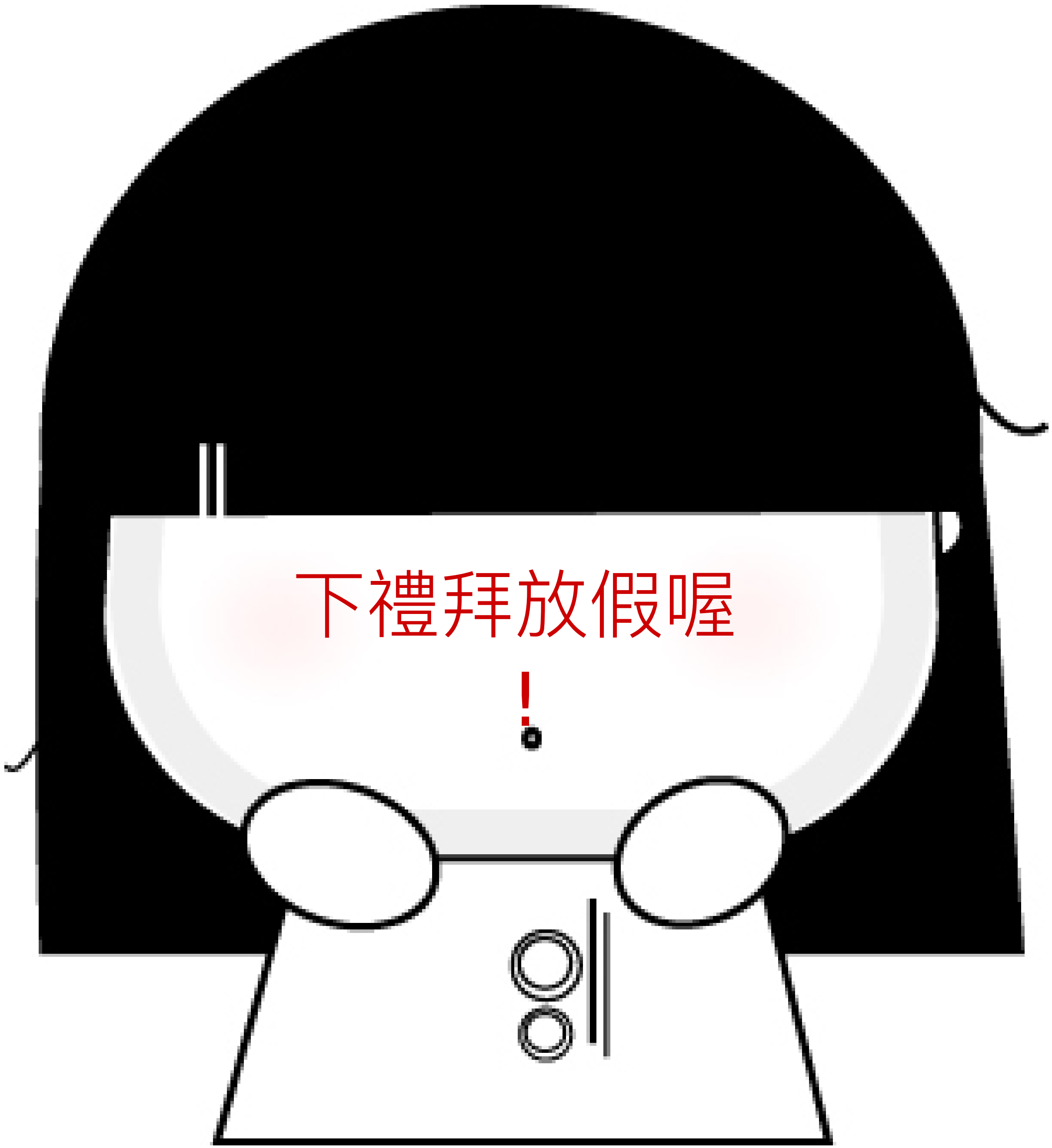
心理與神經資訊學
(Psychoinformatics & Neuroinformatics)

課號：Psy5261
識別碼：227U9340

教室： 博雅 101
時間： 四 234

黃從仁


下禮拜放假喔
!

AI & ML 總論

in·tel·li·gence

noun \in-ˈte-lə-jən(t)s

(1) the ability to learn or understand or to deal with new or trying situations

(2) the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria

為何電腦贏人腦？

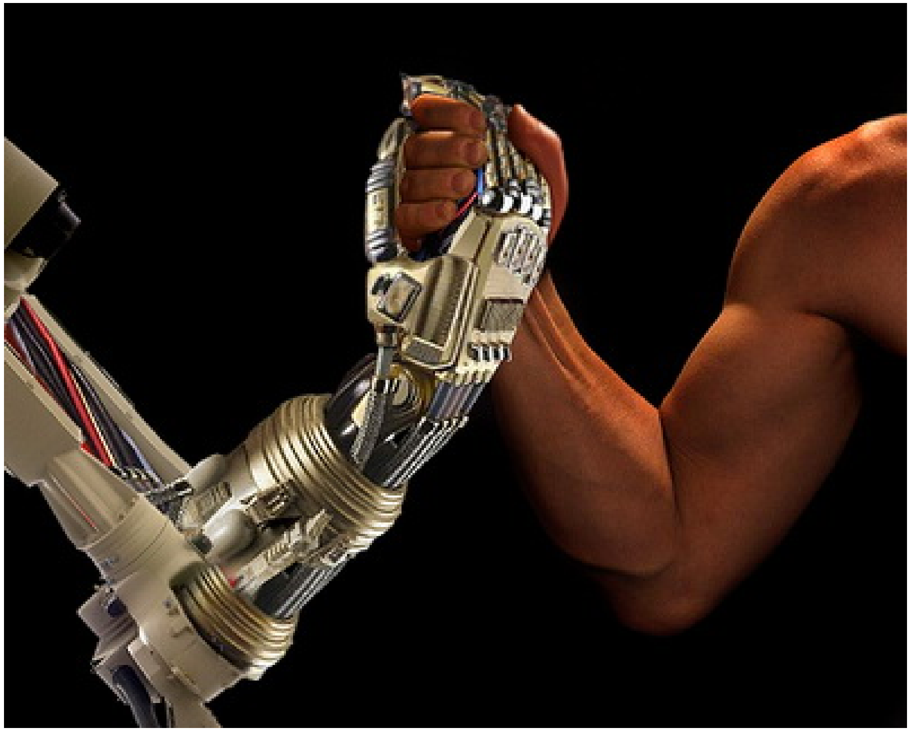
May 11th, 1997

Computer won world champion of chess

(Deep Blue) (Garry Kasparov)

(Reuters = Kyodo News)

電腦 vs. 人腦



- 不會累
- 計算非常精確
- 容量大的完美記憶

- 容易累
- 計算不精確
- 容量小的不完美記憶

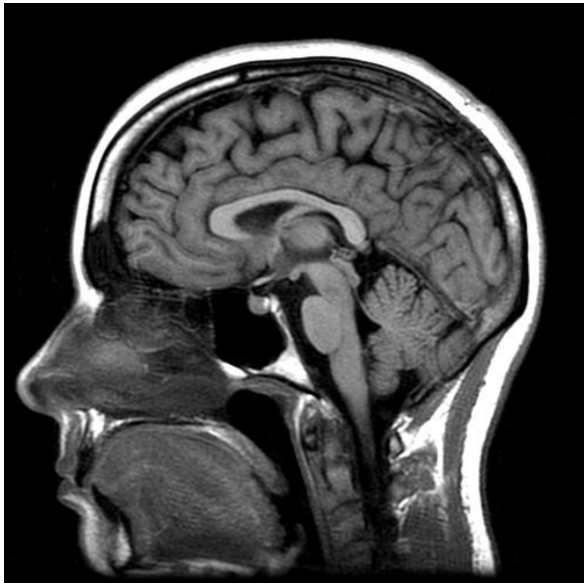
電腦適合透過大量樣本發覺並學習事物的微細規律性

人類記憶能力

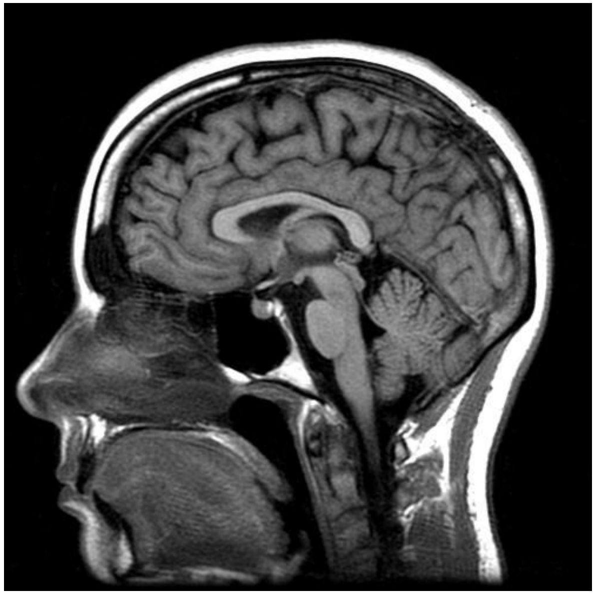
電腦記憶能力

計算區辨力

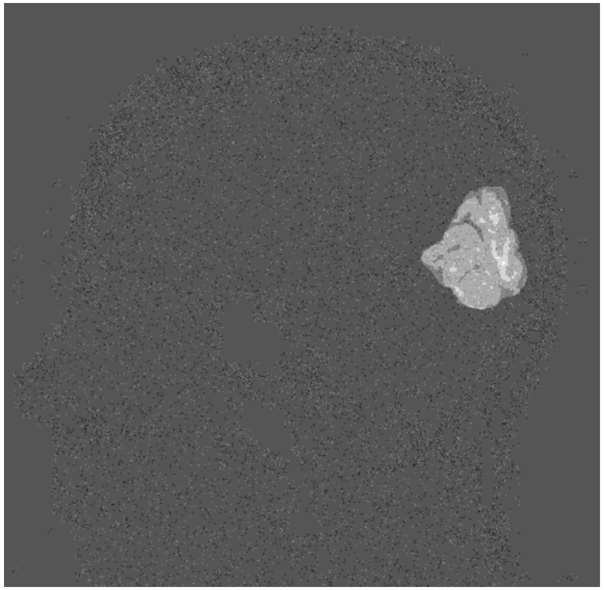
條件 A



條件 B



條件 B- 條件 A



機器學習 (Machine Learning)

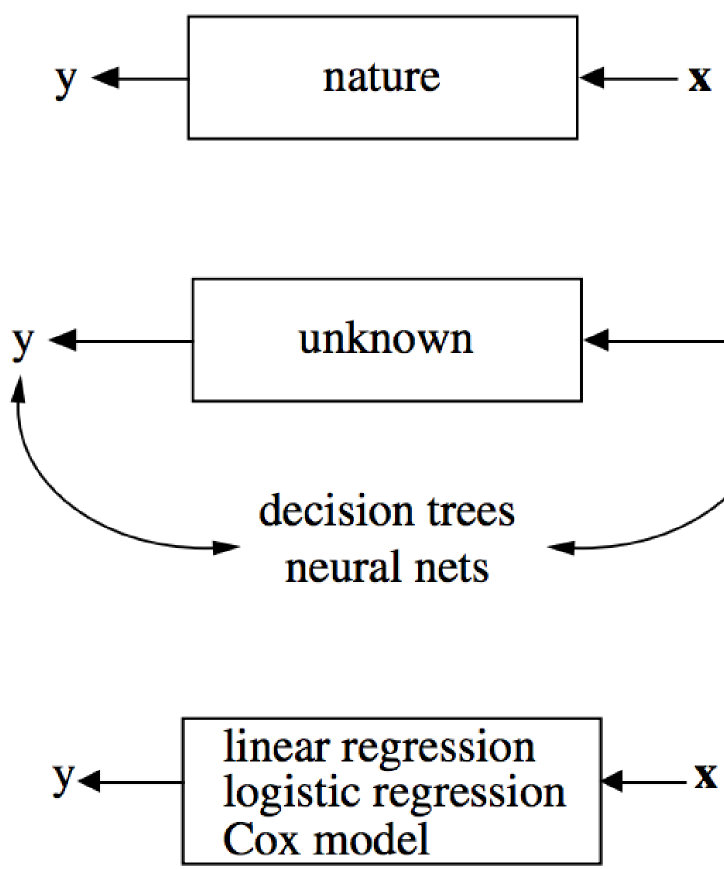
電腦可以幫我們從大量的資料中
找出微細的規律性
進而對資料做自動的分類與預測

統計模型建構的兩種文化

Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

Leo Breiman



心理學理論：解釋 vs. 預測

Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning

Perspectives on Psychological Science
2017, Vol. 12(6) 1100–1122

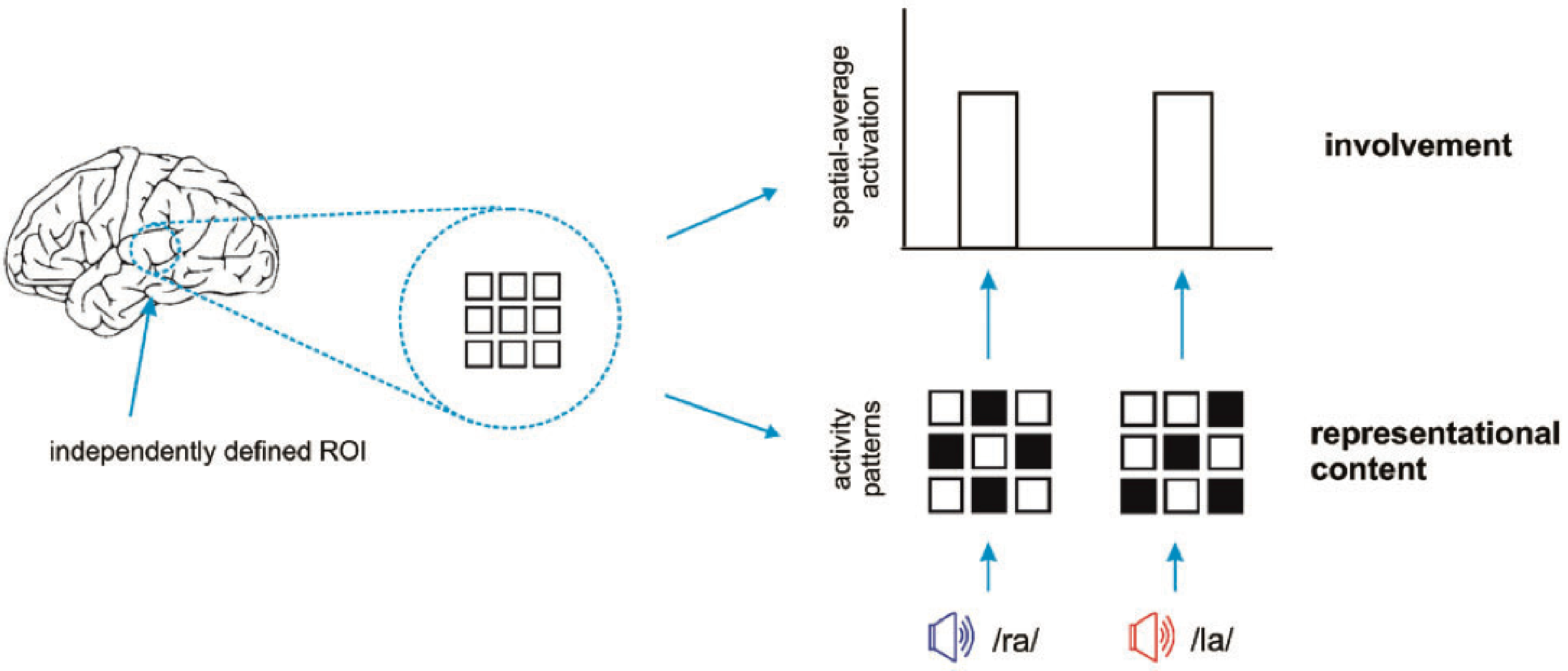
DOI: 10.1177/1745691617693393
www.psychologicalscience.org/PPS
SAGE

Tal Yarkoni and Jacob Westfall
University of Texas at Austin

Abstract
Psychology has historically been concerned, first and foremost, with explaining the causal mechanisms that give rise to behavior. Randomized, tightly controlled experiments are enshrined as the gold standard of psychological research, and there are endless investigations of the various mediating and moderating variables that govern various behaviors. We argue that psychology's near-total focus on explaining the causes of behavior has led much of the field to be populated by research programs that provide intricate theories of psychological mechanism but that have little (or unknown) ability to predict future behaviors with any appreciable accuracy. We propose that principles and techniques from the field of machine learning can help psychology become a more predictive science. We review some of the fundamental concepts and tools of machine learning and point out examples where these concepts have been used to conduct interesting and important psychological research that focuses on predictive research questions. We suggest that an increased focus on prediction, rather than explanation, can ultimately lead us to greater understanding of behavior.

腦科學案例研究：Brain Decoding

心理學家和算命師有何不同？



兩者都會讀心和解夢

分類→辨識→重建

Stimulus(S_i) → Brain activity pattern (P_i)

Classification (Which category)

Mapping a pattern (P_i) to a stimulus category ($c(S_i)$)

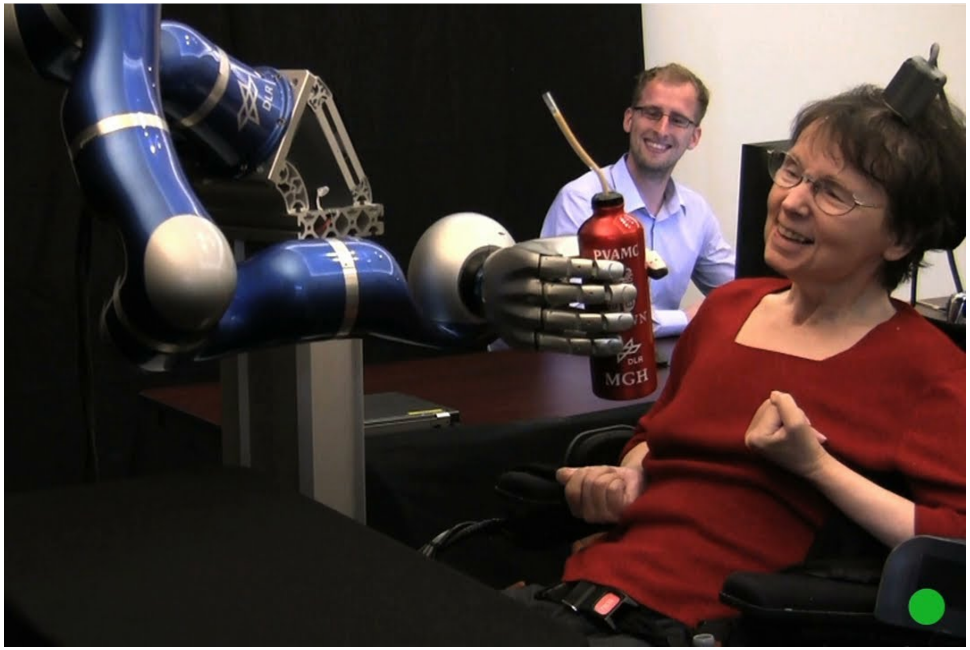
Identification (Which stimulus)

Mapping a pattern (P_i) to a stimulus (S_i) in a finite set

Reconstruction (What stimulus)

Using a pattern (P_i) to construct a stimulus (S_i)

神經義肢 (Neuroprosthetics)



可幫助身體障礙者
與世界互動

現在技術已可感受
物體的軟硬

腦對腦的直接溝通

比阿凡達還厲害

學習裡的兩難

Stability-Plasticity Dilemma: 學愈快忘愈快

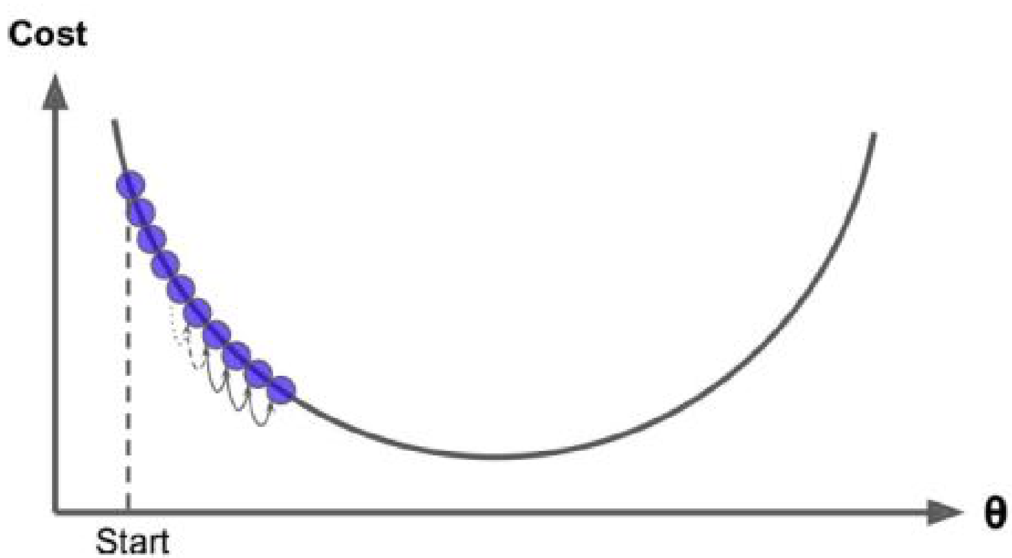
Bias-Variance Dilemma: 想太少還是想太多

Learning-Performance Dilemma: 要學還是要表現

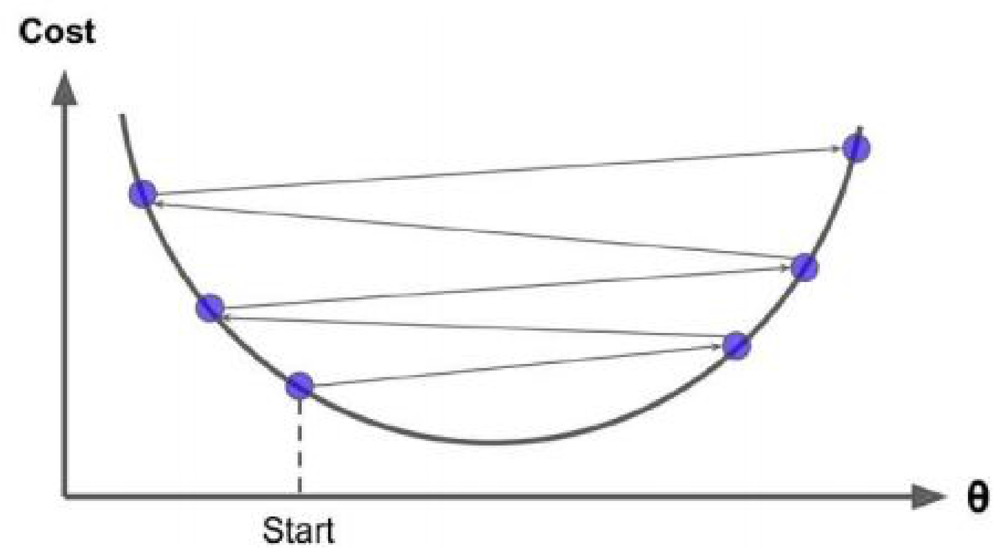
No Free Lunch: 天下沒有白吃的午餐

Stability-Plasticity Dilemma

穩定的系統學得慢 (左); 學得快的系統不穩定 (右)



Slow learning rate: Converges to minimum but very slowly

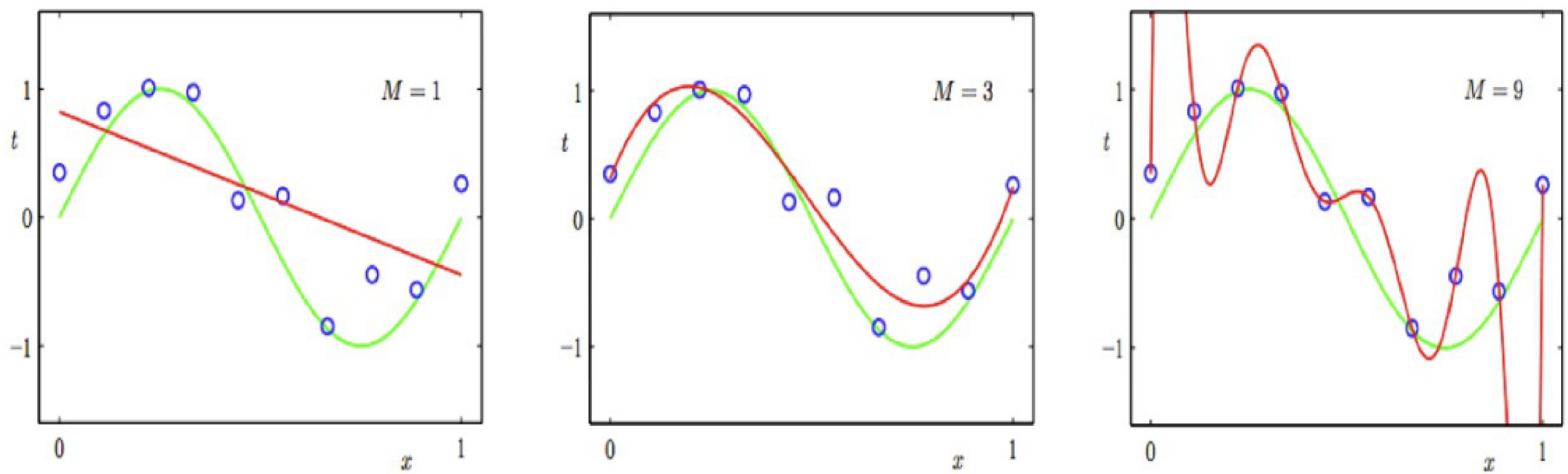


Fast learning rate: May not converge to minimum and error might keep increasing with further epochs

Bias-Variance Tradeoff

簡單的系統變化 (var) 少但估計 / 預測不精確 (bias)

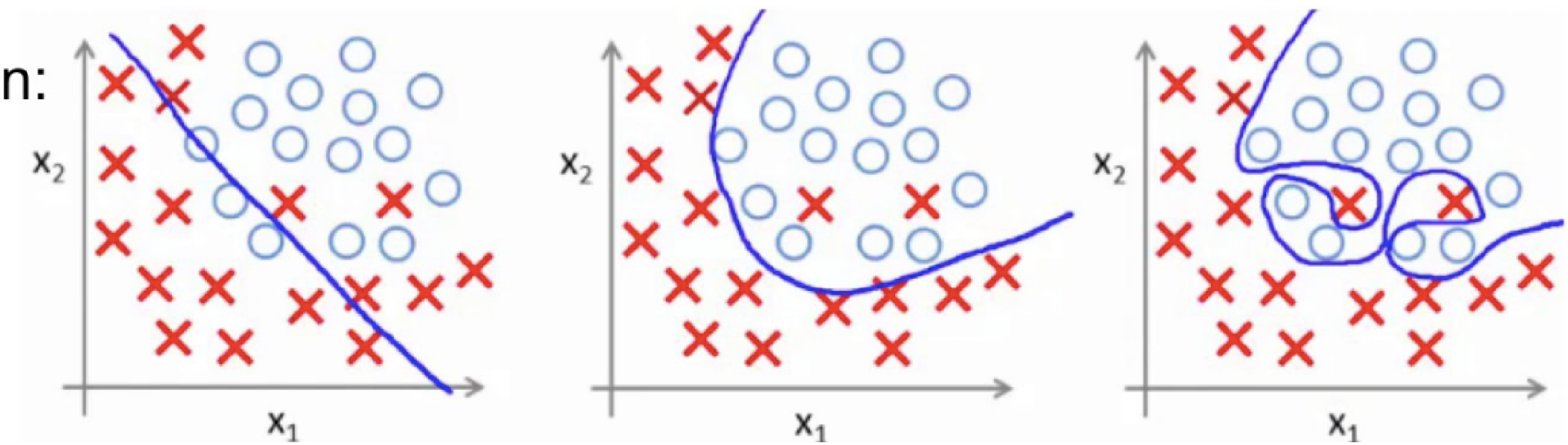
Regression:



predictor too inflexible:
cannot capture pattern

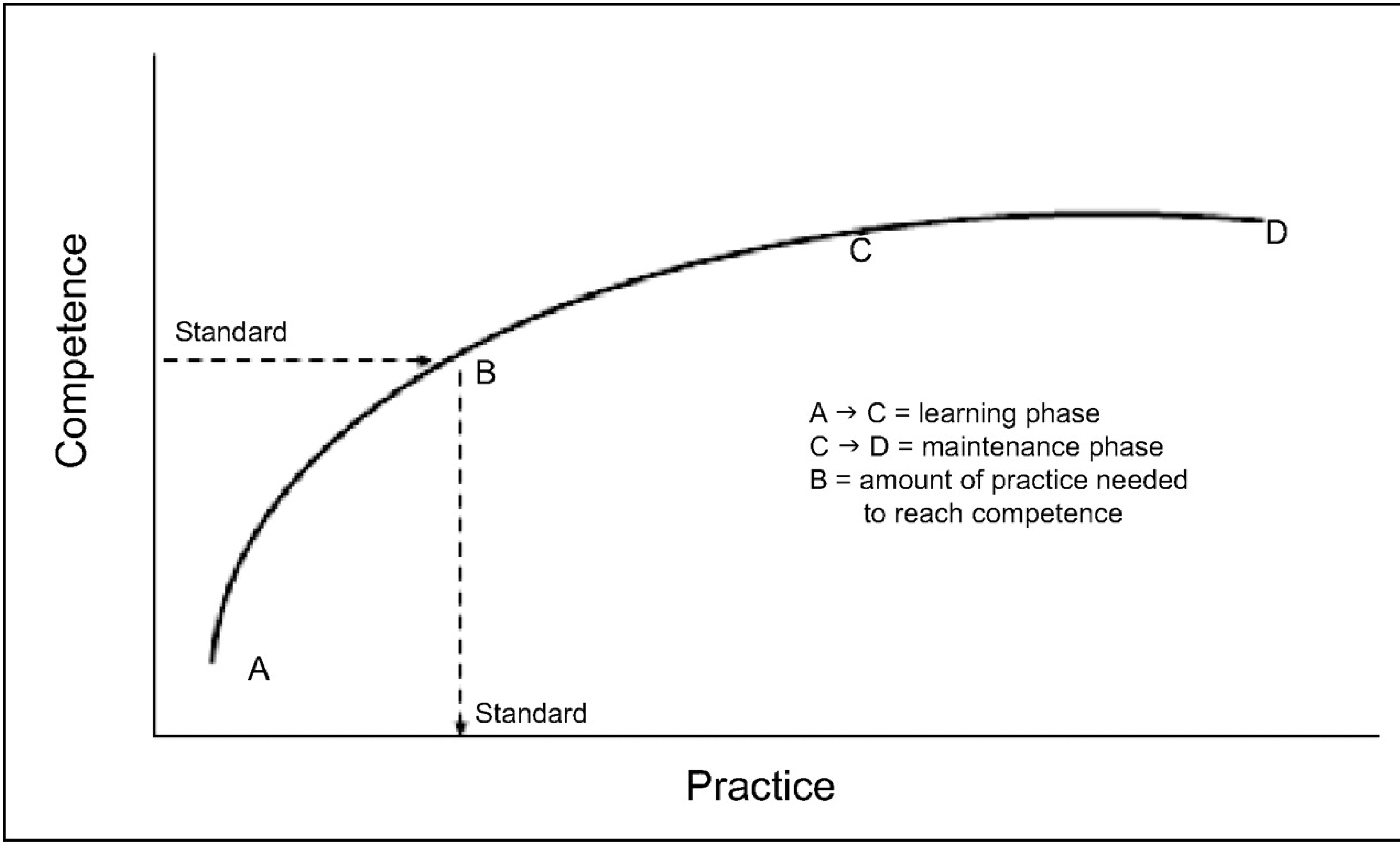
predictor too flexible:
fits noise in the data

Classification:



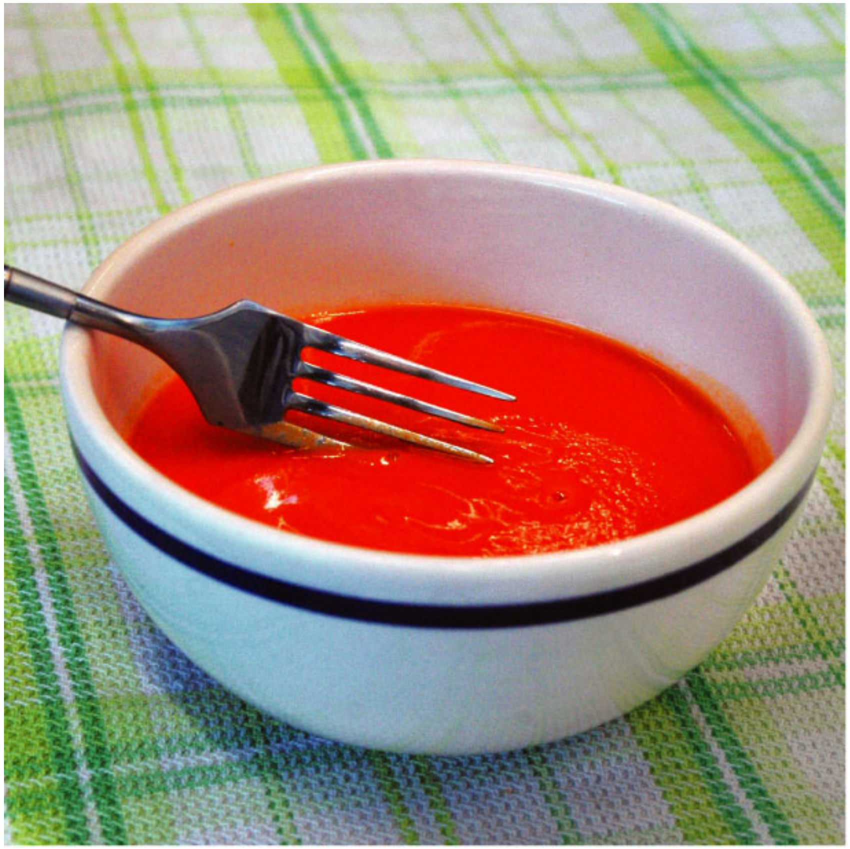
Learning-Performance 兩難

能表現得好時常沒學到什麼；學習新事物時表現常不好



No Free Lunch Theorem

Any two optimization algorithms are equivalent when their performance is averaged across all possible problems.



每種機器學習演算法有其各自的長短處

實務面
(scikit-learn)

市場需求

R&D 資料科學與工程研發	應徵職缺請 Email	hr@mobagel.com
職稱 (產業替代役可)	**年薪 (含配股獎金)**	**需求條件**
VP of Data Science	200萬~600萬	年資 (參考二者年資) 1. 5年以上資料分析專案或專業顧問經驗 2. 2年以上資料科學相關 PhD、PostDoc、教授等研究經驗 必要條件 (至少其中三項具備)： 1. 熟悉 Python 及 Numpy, Pandas, Scikit-learn 套件 2. 熟悉 R 及統計理論 3. 熟悉資訊視覺化設計 4. 具備商業分析專案經驗 5. 具備工業設備分析專案經驗 6. Kaggle 分析競賽排名至少前 10% 7. 英文語言能力佳 (多益860分以上) 或母語者 8. 日文語言能力佳 (日文 N1) 或母語者 加分： 1. 各項資訊專題競賽、黑客松、書卷獎 2. 資料分析相關學術論文發表
Chief Data Scientist	600萬~1200萬	年資 (參考二者年資) 1. 5年以上資料分析專案或專業顧問經驗 2. 2年以上資料科學相關 PhD、PostDoc、教授等研究經驗 必要條件 (至少其中四項具備)： 1. 熟悉機器學習/深度學習演算法及統計方法

Scikit-Learn

scikits learn
machine learning in Python

Classification

Identifying to which set of categories a new observation belong to.

Applications: Spam detection, Image recognition.
Algorithms: *SVM*, *nearest neighbors*, *random forest*, ...

— *Examples*

Regression

Predicting a continuous value for a new example.

Applications: Drug response, Stock prices.
Algorithms: *SVR*, *ridge regression*, *Lasso*, ...

— *Examples*

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: *k-Means*, *spectral clustering*, *mean-shift*, ...

— *Examples*

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: *PCA*, *Isomap*, *non-negative matrix factorization*.

— *Examples*

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: *grid search*, *cross validation*, *metrics*.

— *Examples*

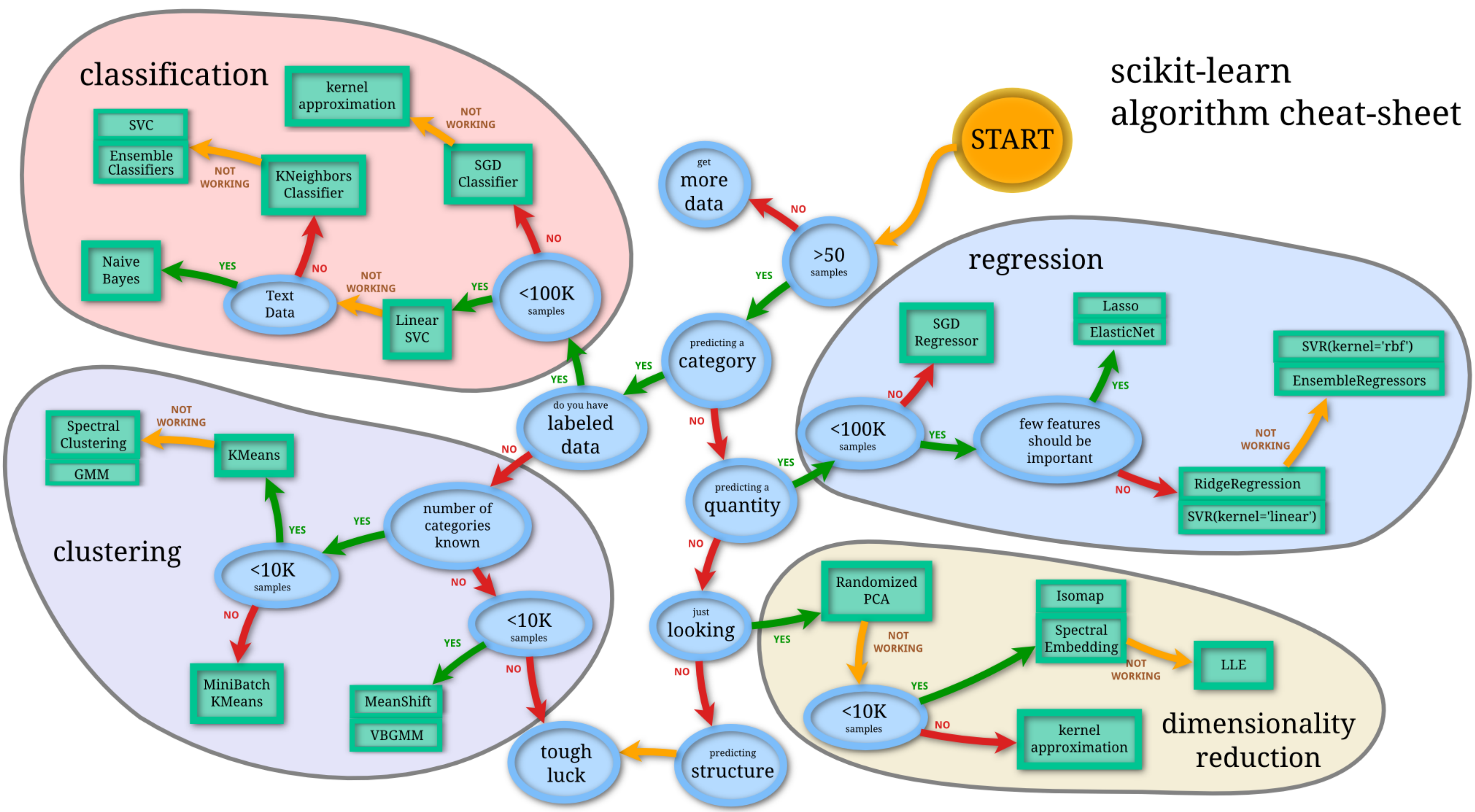
Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: *preprocessing*, *feature extraction*.

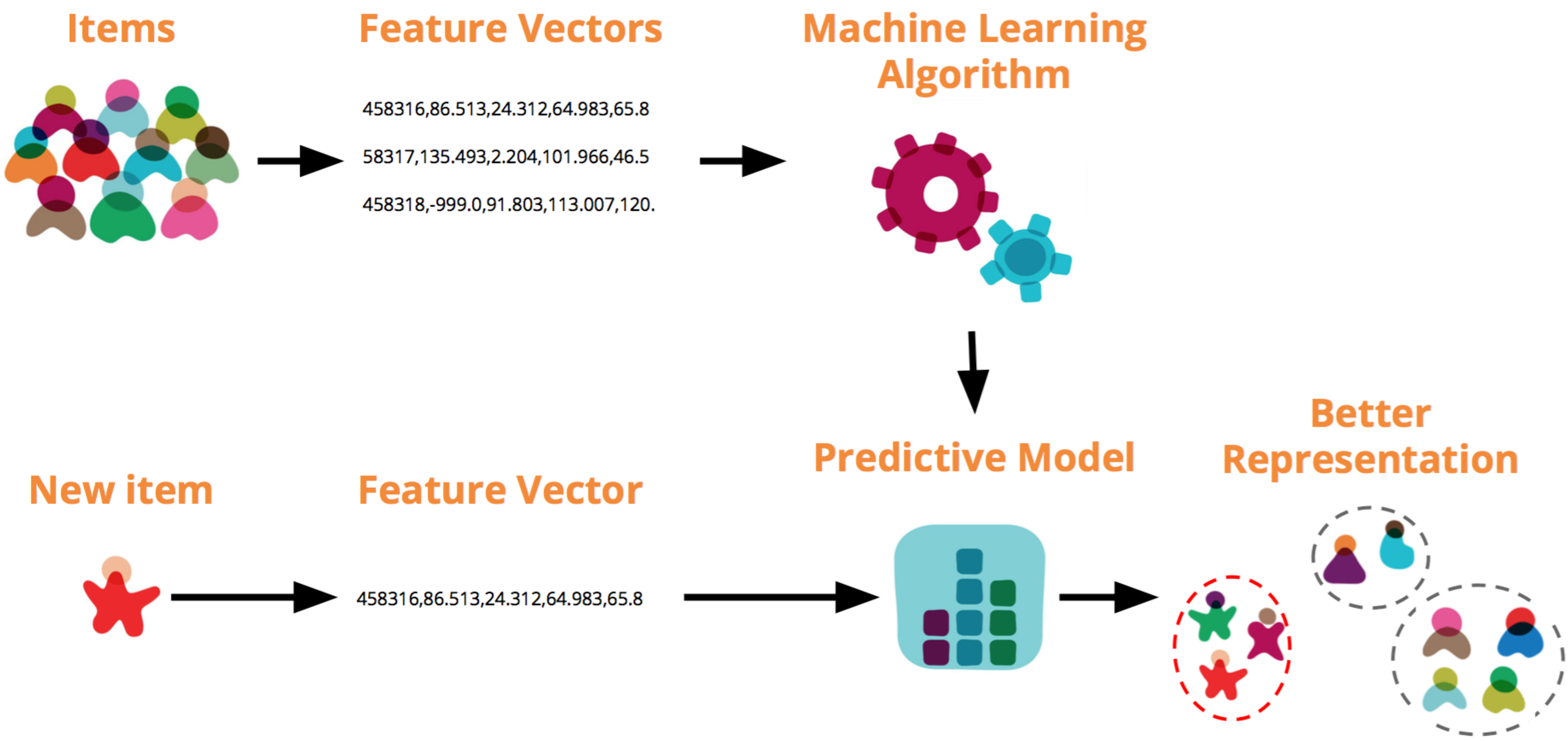
— *Examples*

scikit-learn推薦的 workflow



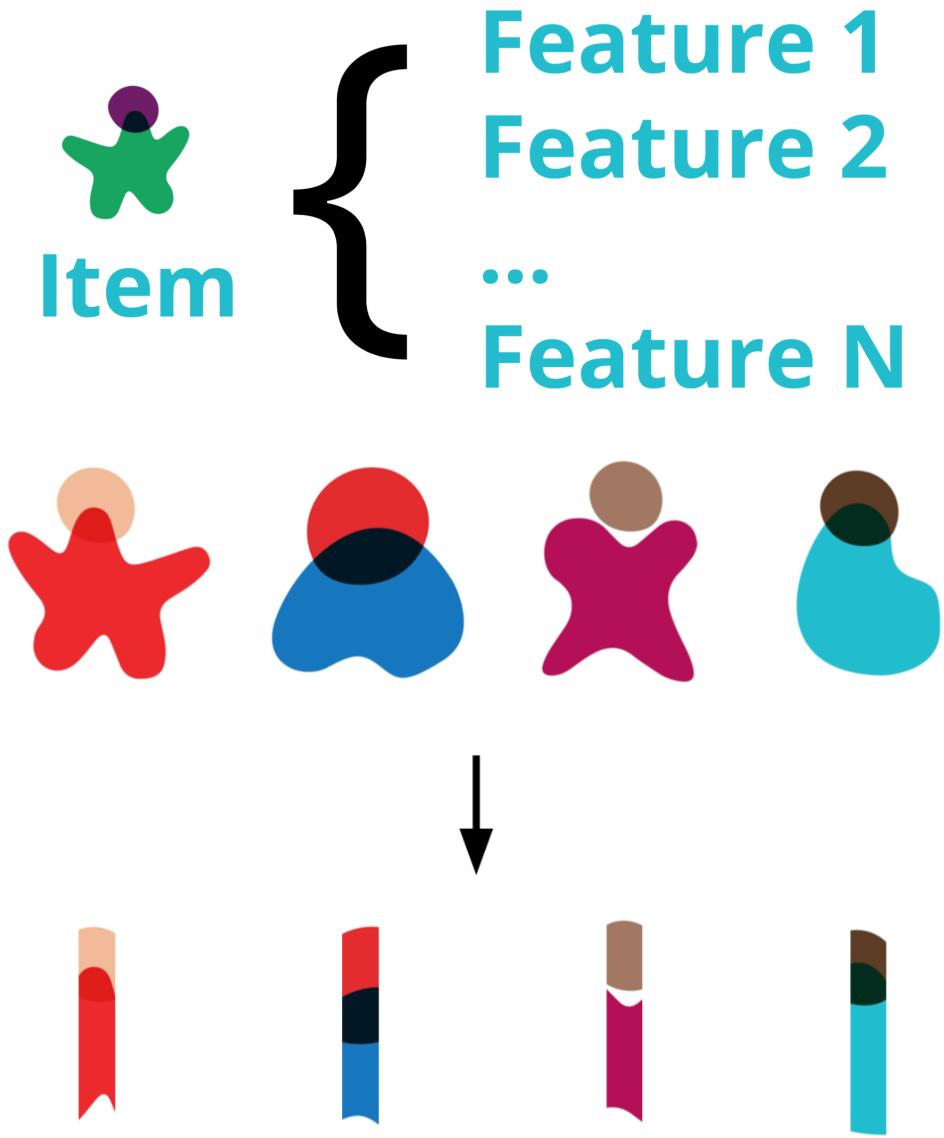
Features & Normalization

每個 feature 單位不同 (如身高 vs. 體重)
因此最好能去單位化 (normalization) 再一起比較



Dimensionality Reduction

Feature space 過大較不容易找出 regularities
最好能想辦法先降低維度 (如用 PCA)



Unsupervised vs. Supervised Learning

非監督式的學習：沒有教師提供正確答案

監督式的學習：有教師提供正確答案

半監督式的學習：有教師提供<u>部分</u>正確答案

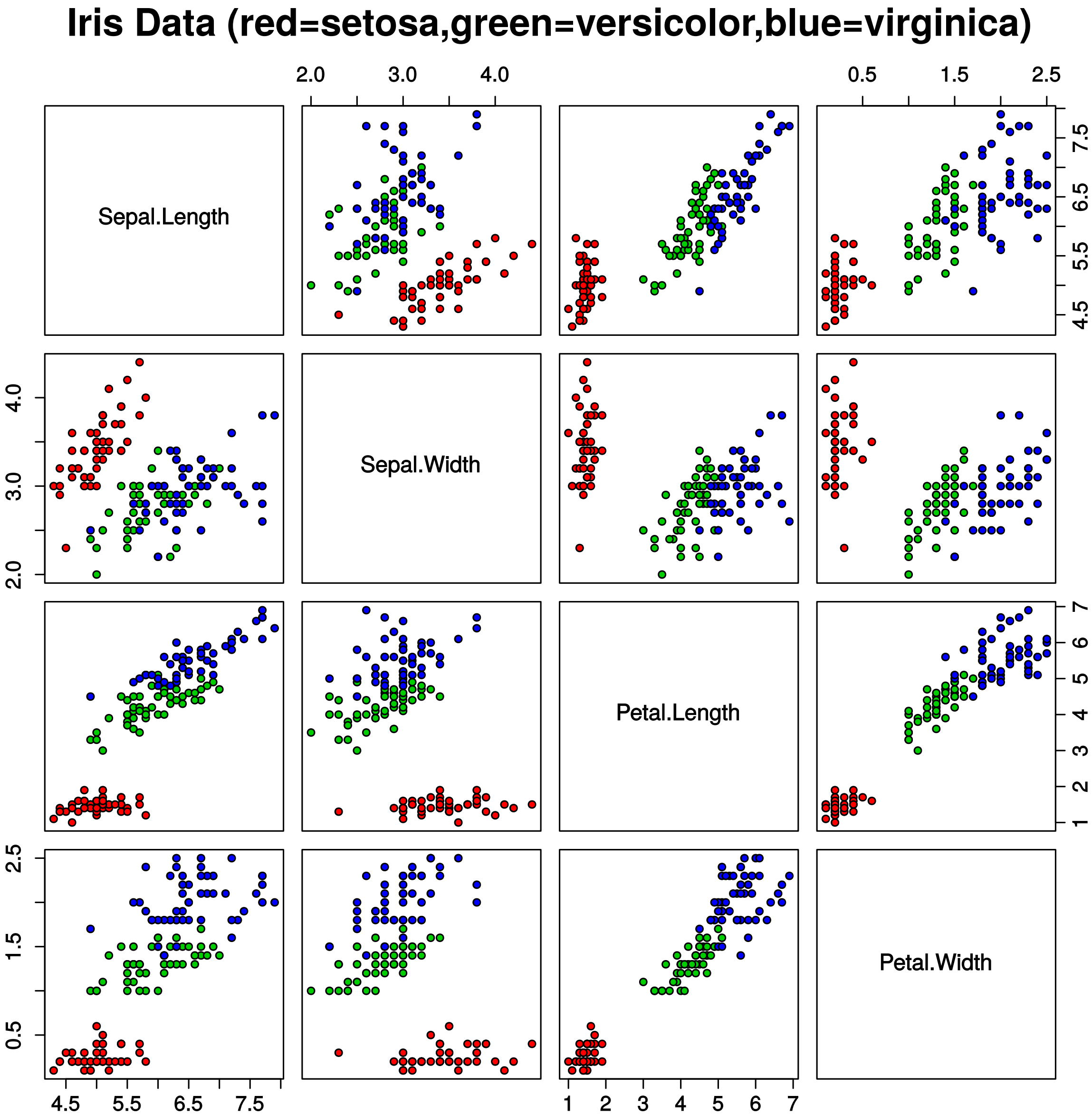
根據花瓣花萼大小的不同，鳶尾花似乎有 3 種

鳶尾花有 3 種，這個樣本是第 k 種

分類問題上主要是看有無提供類別標記以供學習

The Iris Dataset

常用來測試各種新的演算法



非監督式學習
(Unspervised Learning)

琳琅滿目的演算法

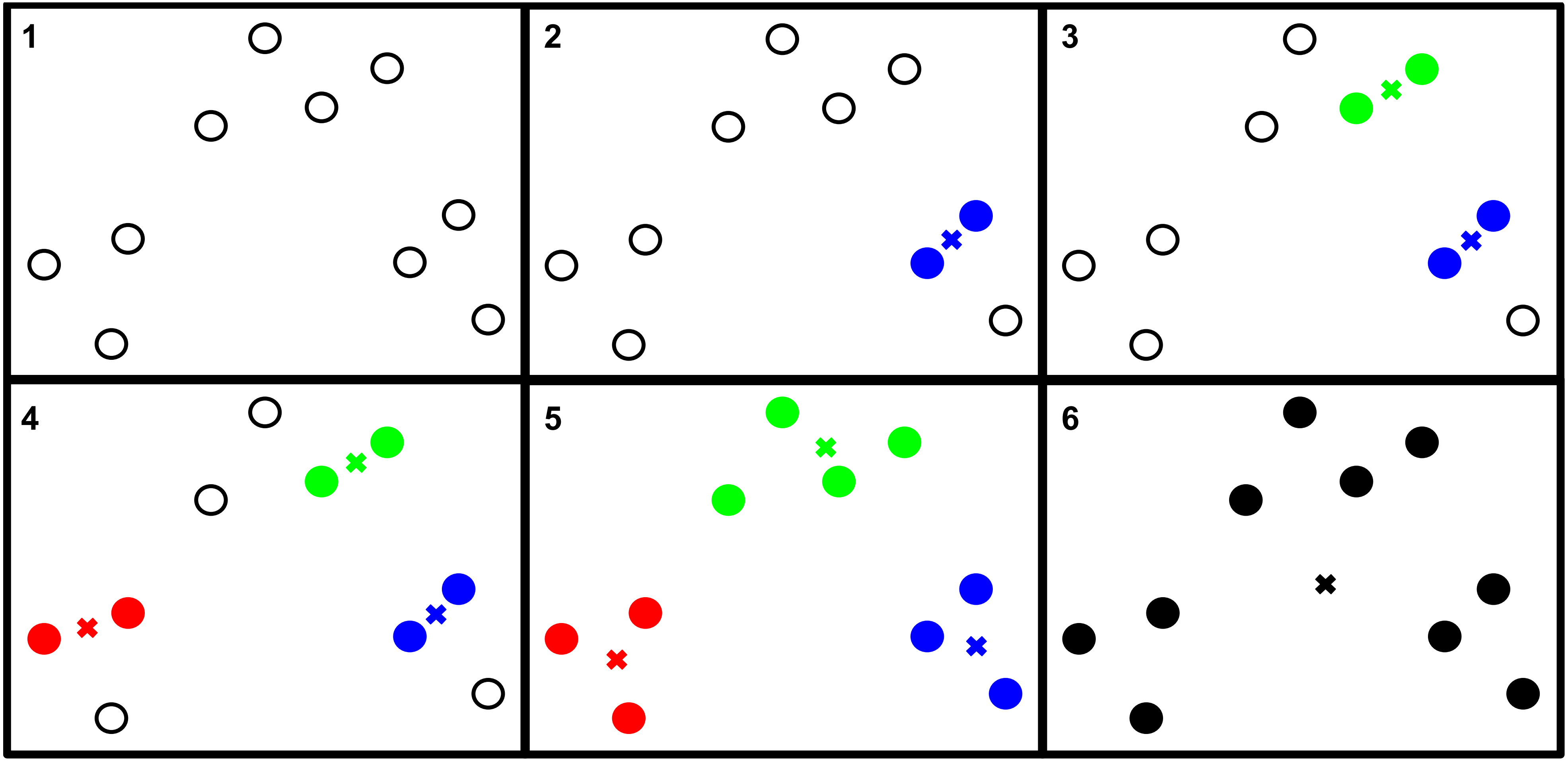
對演算法的原理有一些了解才知道其特性

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

Hierarchical Clustering

每一步驟最相近 / 相似的兩組合成一群

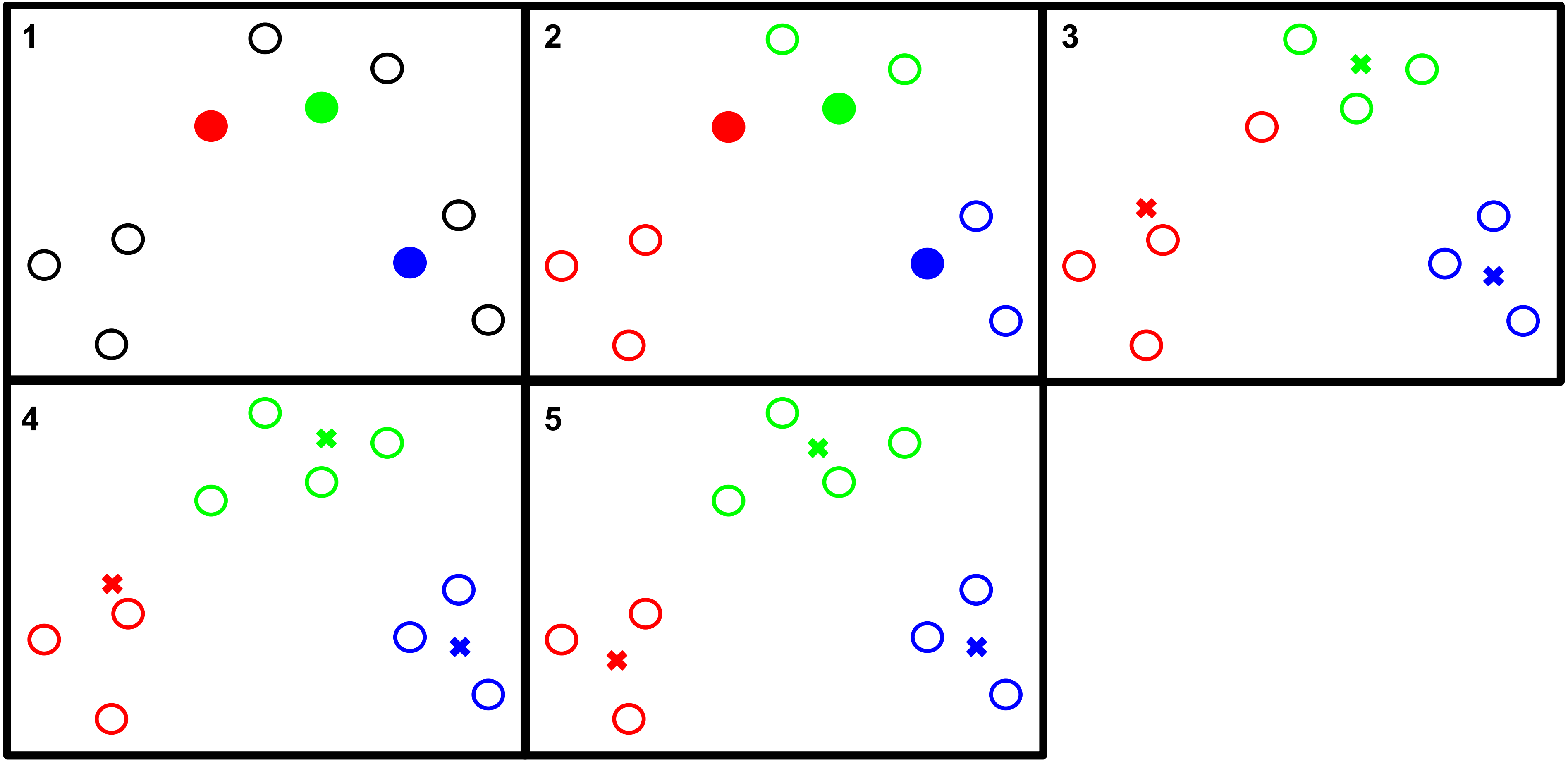
但定義兩群組的距離的方法有很多種




```
model=AgglomerativeClustering(n_clusters=3)
model.fit(X); print(model.labels_)
```

K-Means Clustering

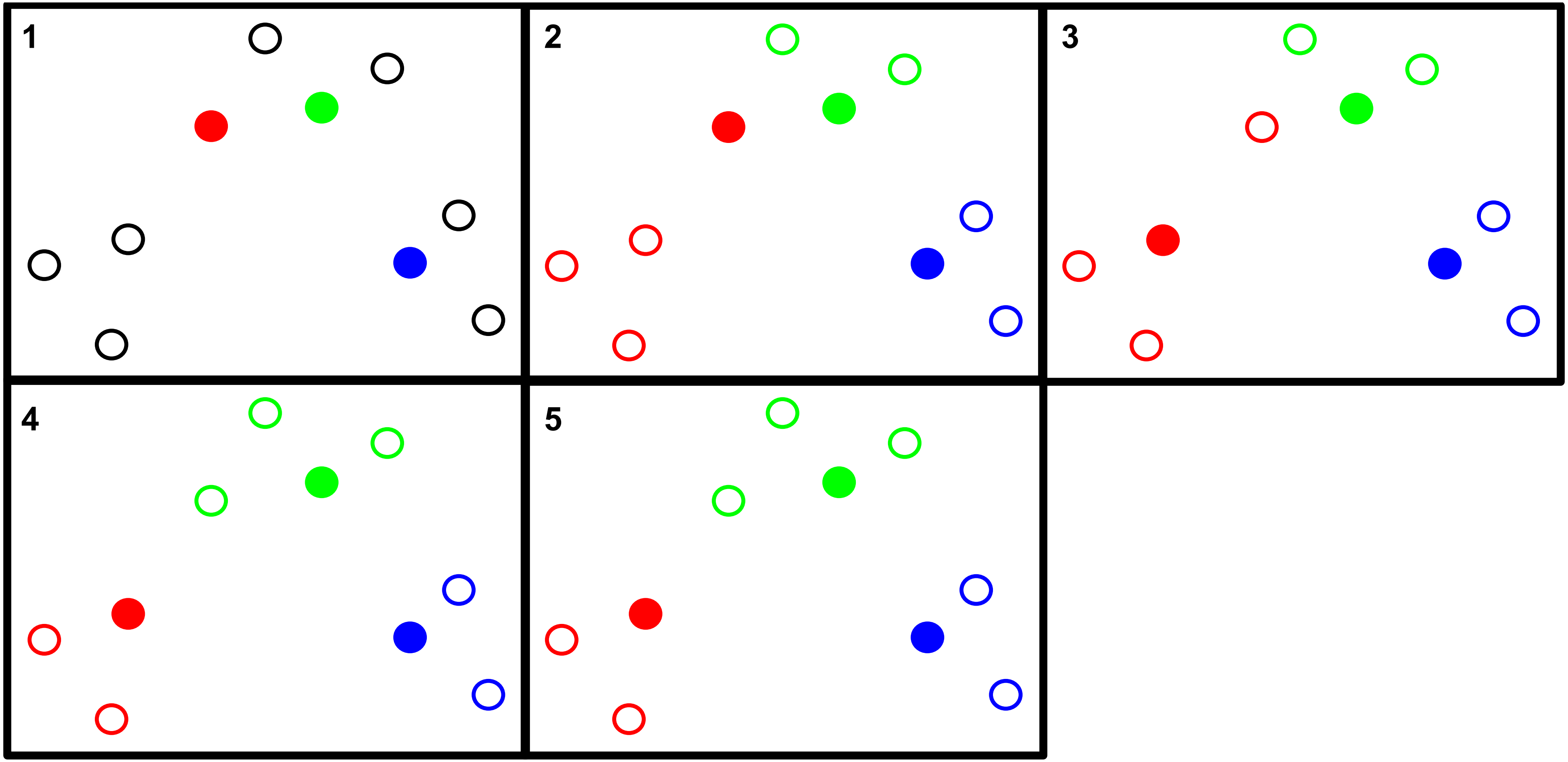
指派各點 x_j 到不同的組群 C_i 使得每個點到其群組中心位置 μ_i 為 k 種可能中最近




```
model=KMeans(n_clusters=3)
model.fit(X); print(model.labels_)
```

K-Medoids Clustering

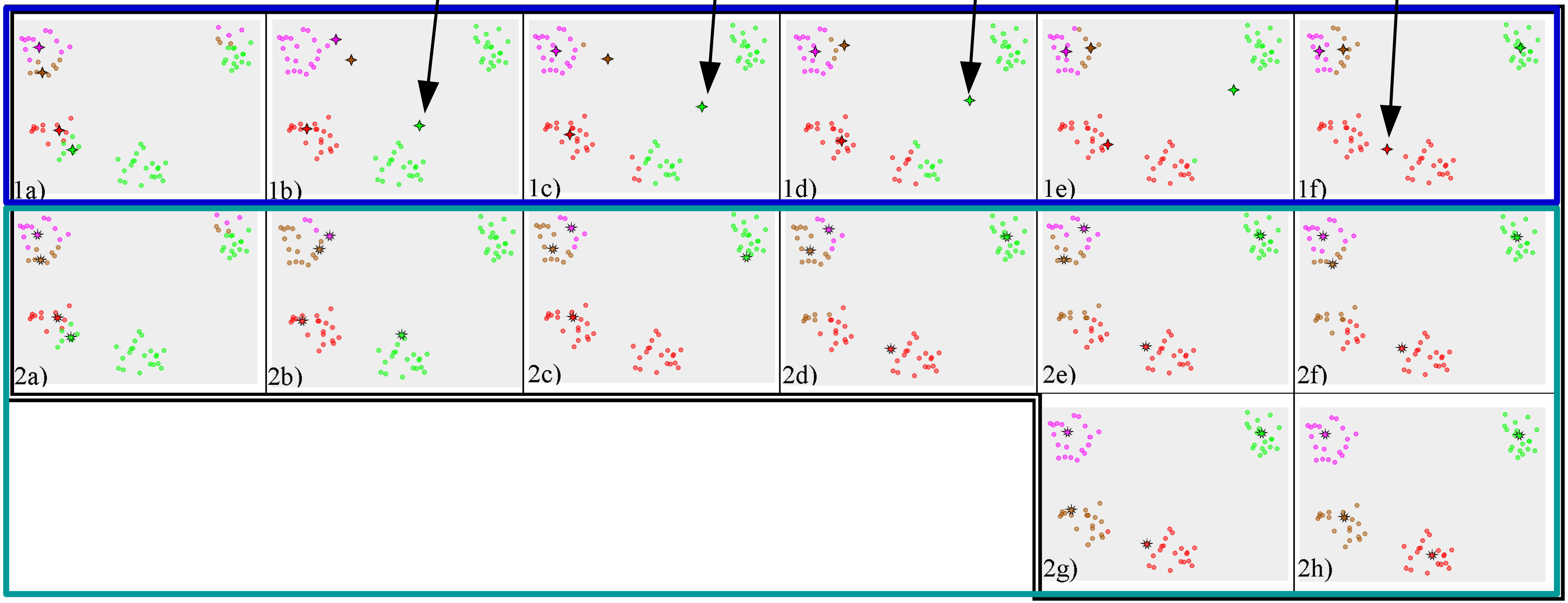
用某一樣本當作每群的原型以避免極值影響
此原型與其他同群內的樣本距離之和為最小



N/A in scikit-learn
Find implementations in Google Search/GitHub

K-Means vs. K-Medoids

K-Means 容易產生四不像的群組原型 (prototype)！



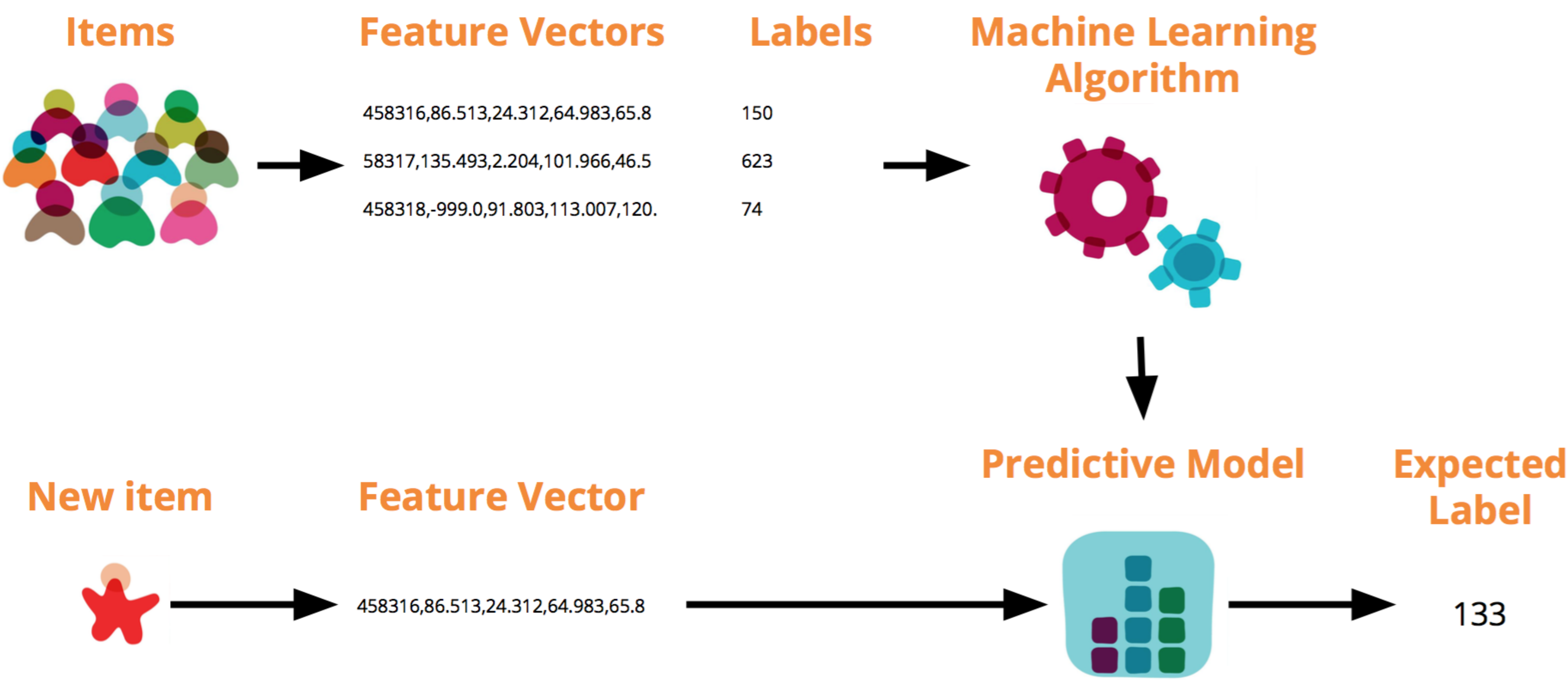




監督式學習
(Supervised Learning)

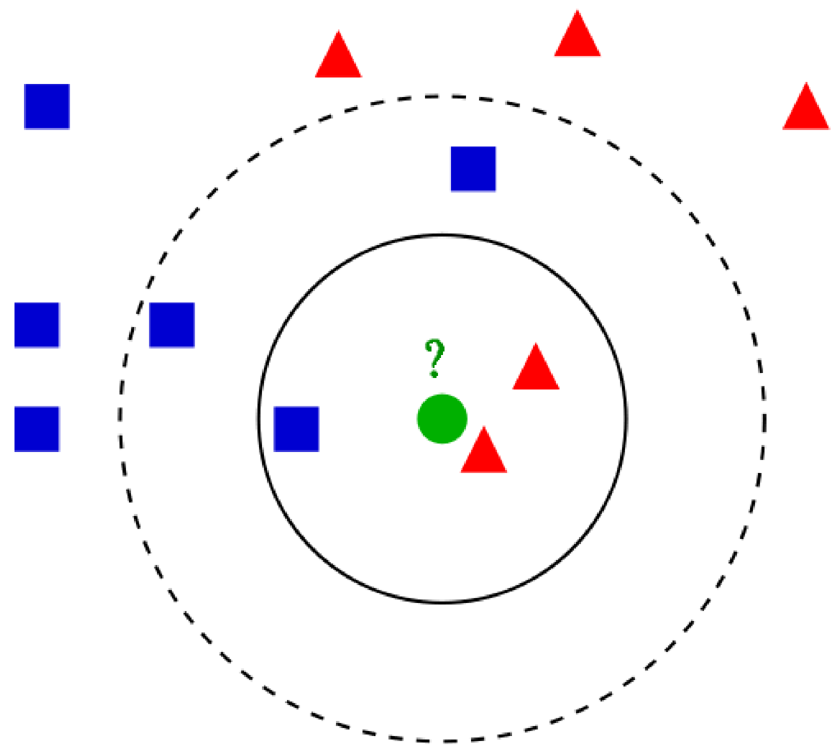
整體流程

比 unsupervised learning 多出了 labels 的部分



K-Nearest Neighbor (kNN)

用回憶中最相似的 k 個範例，以多數決判斷類別 / 數值



k=3 ●= ▲

k=5 ●= ■

k=11 ●= ?

```
from sklearn import *
clf=neighbors.KNeighborsClassifier(1) #try 1
clf.fit(X,Y) #training
print(np.mean(clf.predict(X)==Y)) #testing
```

Distance-Weighted KNN

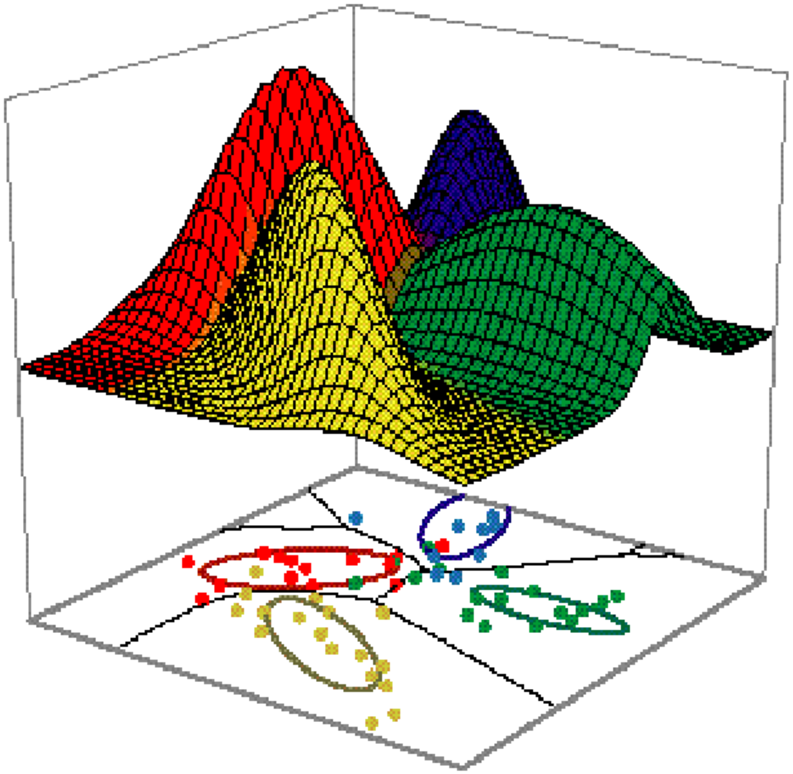
$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \quad \text{with } w_i = \frac{1}{d(x_q, x_i)^2}$$

改進 KNN 使得投票比重和距離成反比

```
from sklearn import *
clf=neighbors.KNeighborsClassifier(3,'distance')
clf.fit(X,Y) #training
print(np.mean(clf.predict(X)==Y)) #testing
```

Naive Bayes Classifier

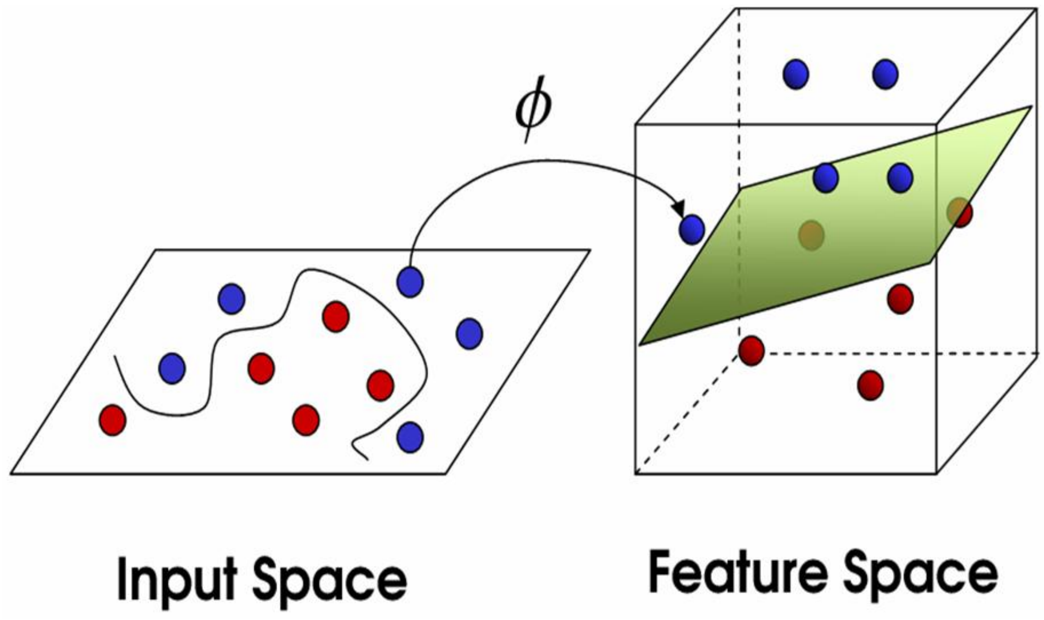
$$P\left(C_i|\vec{F}\right)=\frac{P\left(\vec{F}|C_i\right)P\left(C_i\right)}{P\left(\vec{F}\right)}$$

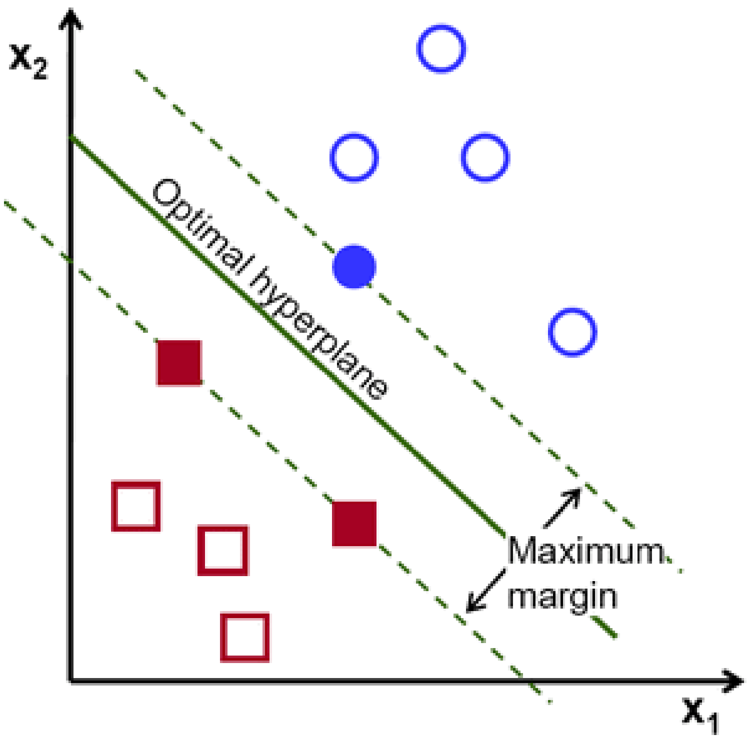


用觀察到的範例估計母體分佈，
判斷一個樣本最有可能的類別。

```
from sklearn import *
clf=naive_bayes.GaussianNB()
clf.fit(X,Y) #training
print(np.mean(clf.predict(X)==Y)) #testing
```

Support Vector Machine

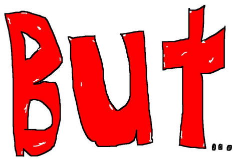





```
from sklearn import *
clf=svm.SVC()
clf.fit(X,Y) #training
print(np.mean(clf.predict(X)==Y)) #testing
```

機器學習的各種亂做

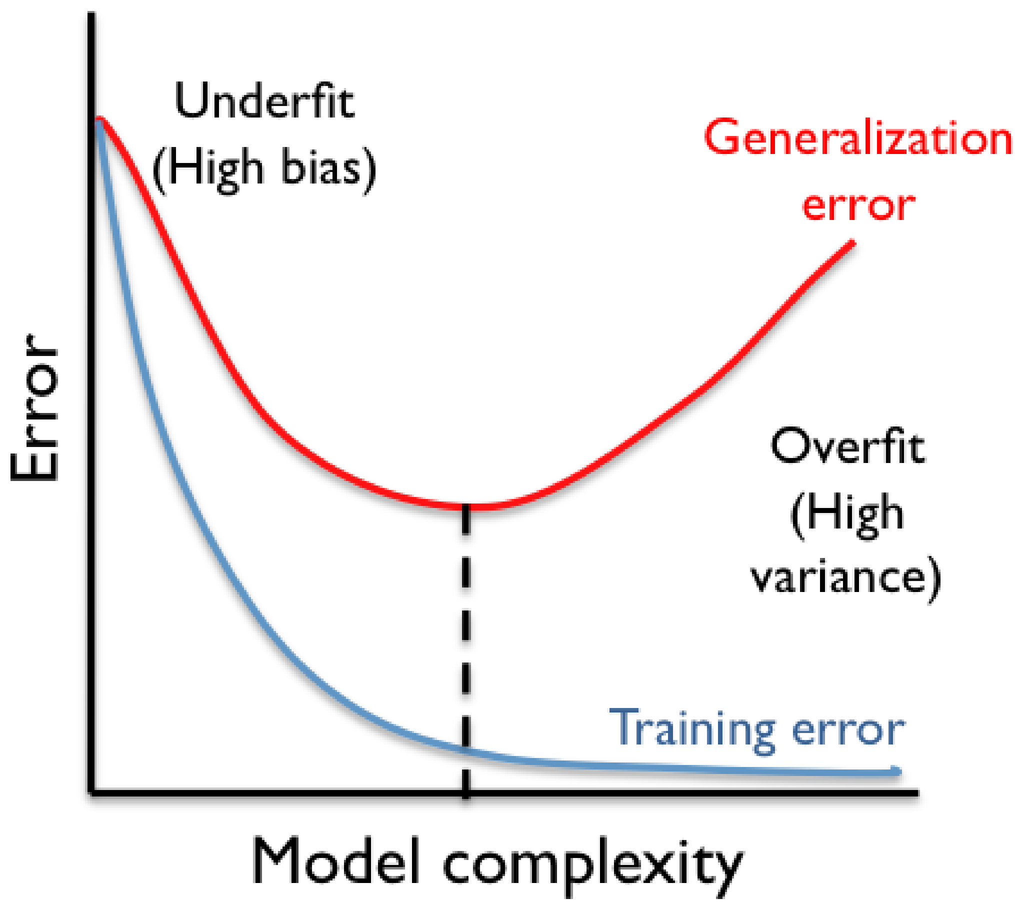
機器學習常見的錯誤 (1/3)




```
X2=np.random.rand(150,4) # or randint for y
clf=neighbors.KNeighborsClassifier(1)
#clf=naive_bayes.GaussianNB()
#clf=svm.SVC()
clf.fit(X2,Y);
pred=clf.predict(X2)
print(np.mean(pred==Y))
print(metrics.confusion_matrix(Y,pred))
```

訓練 (training) 適可而止

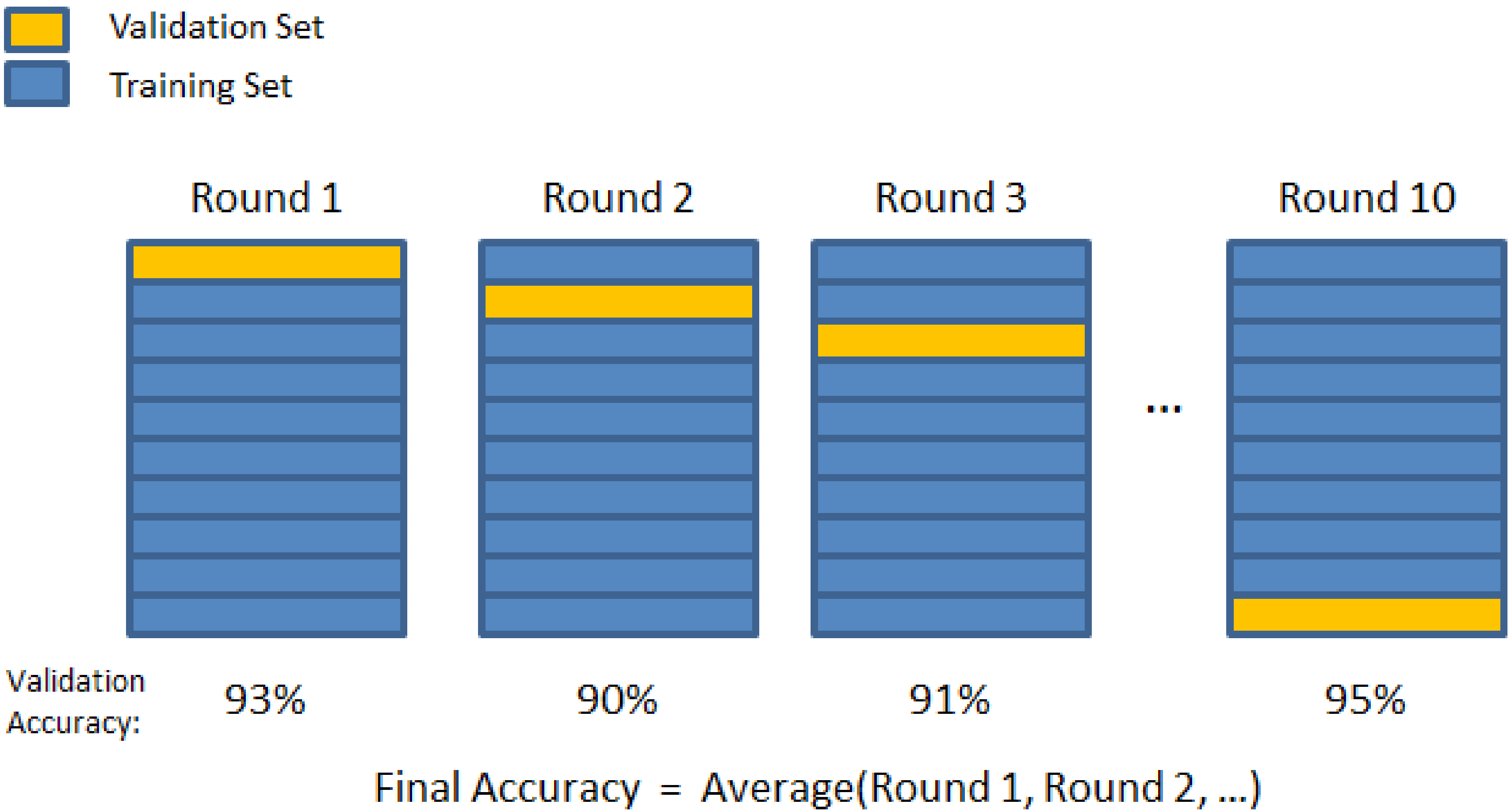
training vs. testing errors
又叫 In- vs. out-sample errors



監督式學習的目標在 minimize testing errors

監督式學習需要 Cross-Validation

10-fold validation 的例子：



當 k=N 時稱為 leave-one-out cross-validation

機器學習常見的錯誤 (2/3)

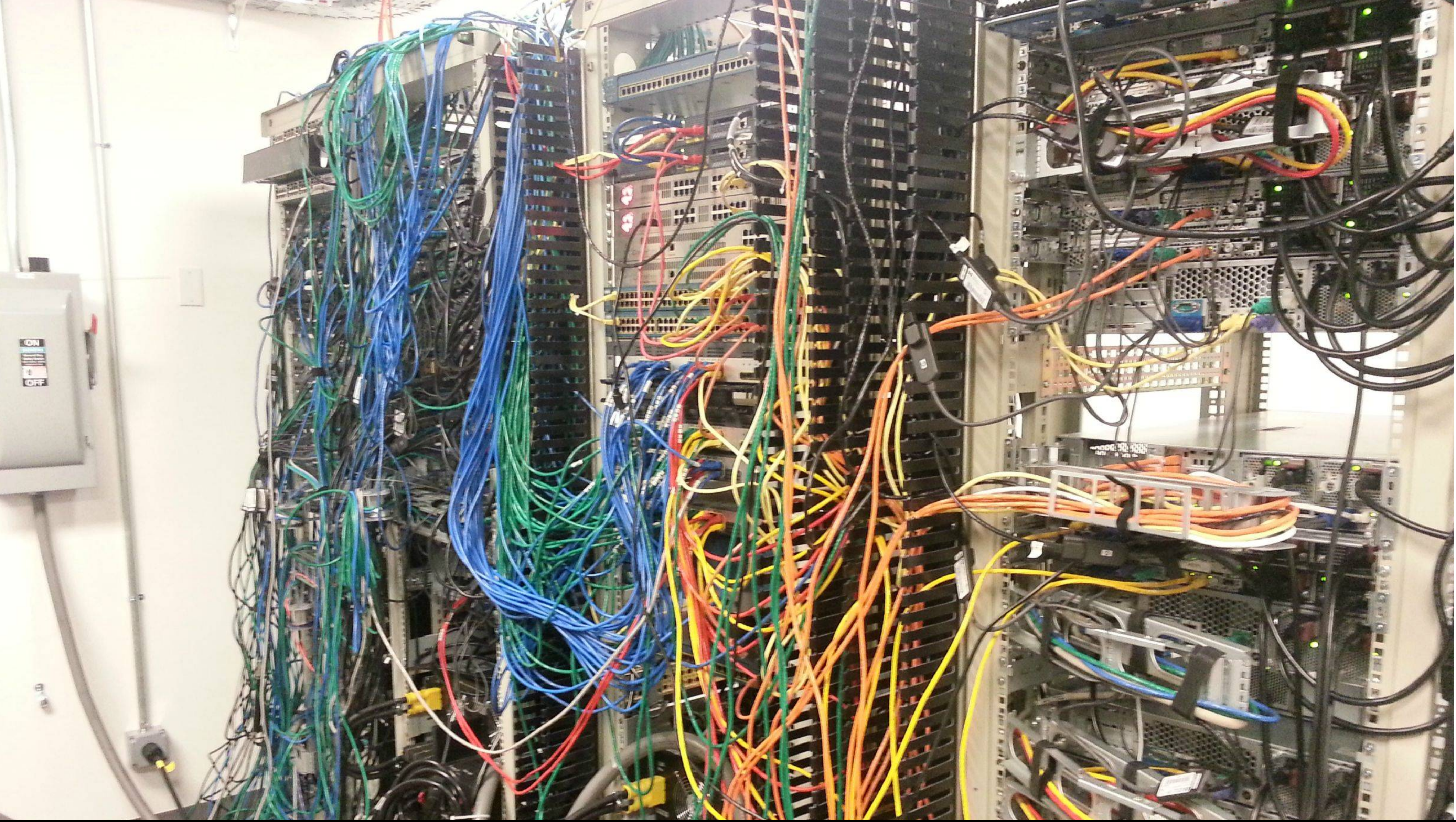

```
clf=svm.SVC() # try other supervised classifers
kf=cross_validation.KFold(len(Y),5)
s1=cross_validation.cross_val_score(clf,X,Y,cv=kf)
s2=cross_validation.cross_val_score(clf,X2,Y,cv=kf)
print(s1.mean(),s2.mean())
```

機器學習常見的錯誤 (3/3)

Supervised learners 常會學到 prior distributions

```
from numpy import *
from sklearn.svm import *
from sklearn.model_selection import *
from sklearn.metrics import *

x=random.rand(100,3) # 3-d random features
y=random.permutation([0]*90+[1]*10) # 2 categories
yp=cross_val_predict(SVC(),x,y,cv=KFold(100)) # leave-1-out
print('Accuracy:',mean(y==yp)) # mean accuracy
print('C. Matrix:\n',confusion_matrix(y,yp)) # confusion matrix
```



聽魯師一個勸：

每個 ML pipeline 都要打亂測過

Game Over