

資料科學課程

機器學習

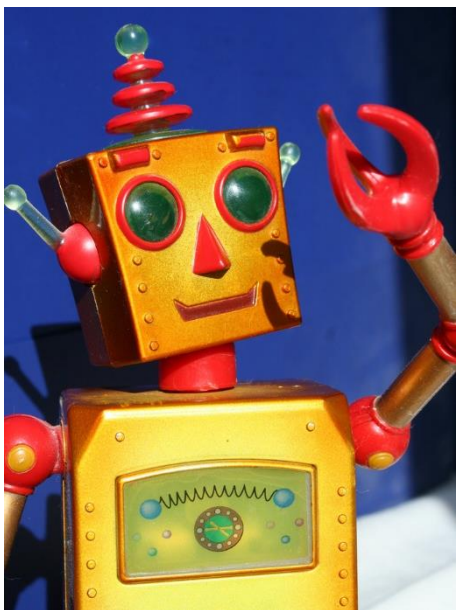
Scikit-learn

蔡岳霖

2019 / 04 / 12

甚麼是機器學習？

- 從資料中自動分析而了解資料的規律/趨勢/特性
- 可用相同規則應用在未知資料做出預測



$$f(x) = y$$



機器學習的流程



- 定義問題
- 依照問題與資料狀況選擇模型
- 資料前處理
- 建立模型
- 模型評估

機器學習的類型

- 監督式學習
 - 回歸
 - 分類
- 非監督式學習
 - 分群
 - 維度縮減
- 強化學習

Scikit - learn

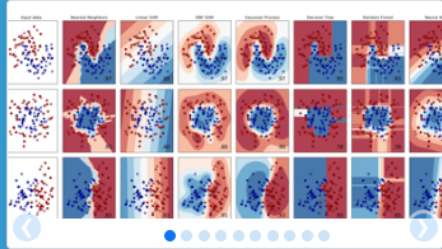
- Python 程式語言內用於機器學習的套件

- 功能

- 資料前處理

- 各類機器學習模型

- 模型評估指標

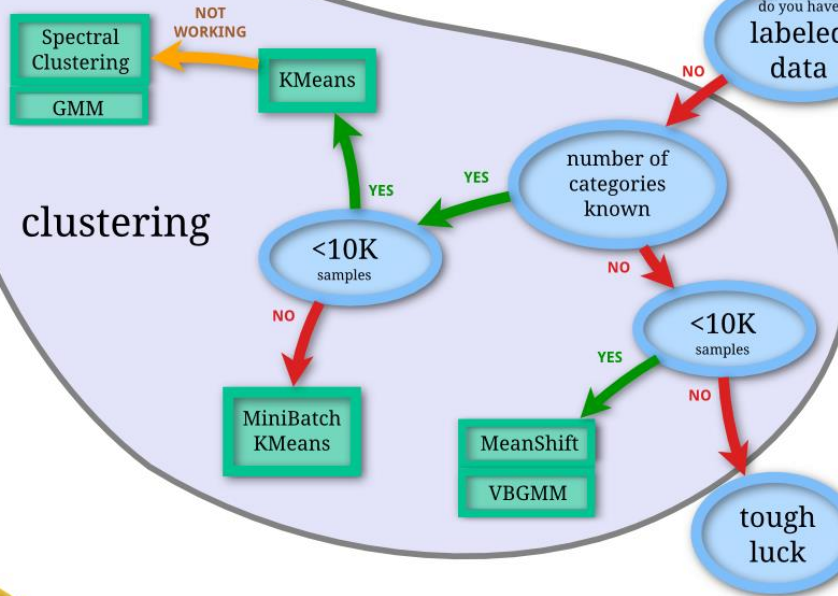


scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification	Regression	Clustering
Identifying to which category an object belongs to. Applications: Spam detection, Image recognition. Algorithms: SVM, nearest neighbors, random forest, ... — Examples	Predicting a continuous-valued attribute associated with an object. Applications: Drug response, Stock prices. Algorithms: SVR, ridge regression, Lasso, ... — Examples	Automatic grouping of similar objects into sets. Applications: Customer segmentation, Grouping experiment outcomes Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples
Dimensionality reduction	Model selection	Preprocessing
Reducing the number of random variables to consider. Applications: Visualization, Increased efficiency Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples	Comparing, validating and choosing parameters and models. Goal: Improved accuracy via parameter tuning Modules: grid search, cross validation, metrics. — Examples	Feature extraction and normalization. Application: Transforming input data such as text for use with machine learning algorithms. Modules: preprocessing, feature extraction. — Examples

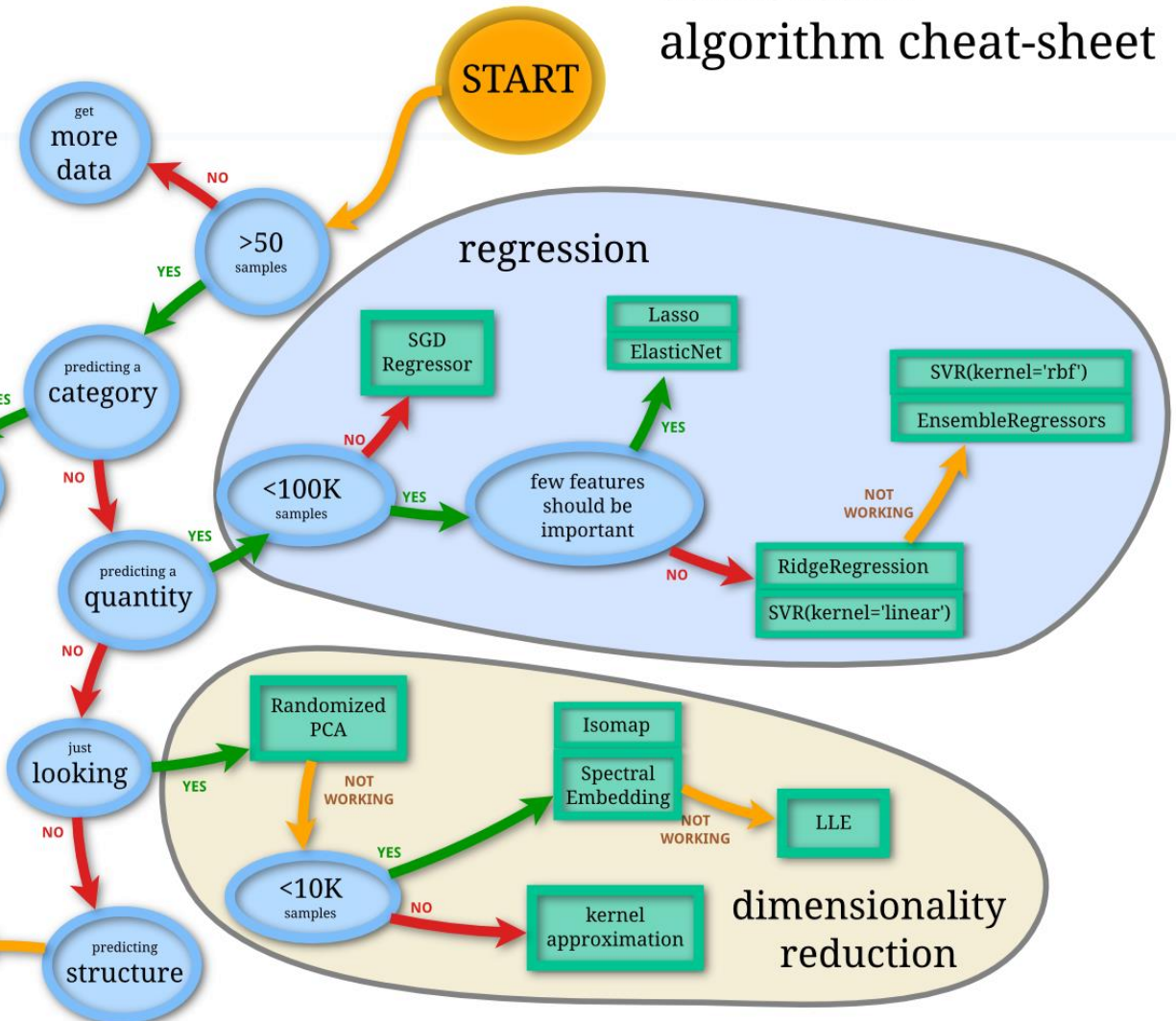
classification



```

graph TD
    Start([do you have labeled data]) --> Q1([number of categories known])
    Q1 -- YES --> KMeans1[KMeans]
    Q1 -- NO --> Q2([<10K samples])
    Q2 -- YES --> KMeans1
    Q2 -- NO --> MiniBatchKMeans[MiniBatch KMeans]
    KMeans1 -- NOT WORKING --> SpectralClustering[Spectral Clustering]
    KMeans1 -- NOT WORKING --> GMM[GMM]
    Q2 -- YES --> MeanShift[MeanShift]
    Q2 -- YES --> VBGM[VBGM]
    MeanShift -- NO --> ToughLuck([tough luck])
    VBGM -- NO --> ToughLuck
  
```

The flowchart is titled "clustering". It begins with a decision point "do you have labeled data". If the answer is "YES", it proceeds to "number of categories known". If "YES", it leads to "KMeans". If "NO", it leads to "<10K samples". From "<10K samples", if "YES", it leads to "KMeans", and if "NO", it leads to "MiniBatch KMeans". From "KMeans", if "NOT WORKING", it leads to "Spectral Clustering" and "GMM". From "<10K samples", if "YES", it leads to "MeanShift" and "VBGM". From "MeanShift" and "VBGM", if "NO", it leads to "tough luck".

The logo for Scikit-Learn, featuring three overlapping circles: a yellow one at the top left with the word "Back" in black, a blue one at the bottom left, and a large orange one on the right containing the text "scikit" in a small sans-serif font and "learn" in a large, bold, black script font.

Source

一頁簡報就上手sklearn

```
import pandas as pd
from sklearn import preprocessing, linear_model, model_selection, metrics
```

```
data = pd.read_csv('example_data.csv')

data_y = data['target']
data = data.drop('target', axis = 1, inplace = True)
```

```
one_hot_data = pd.get_dummies(data)

ss = preprocessing.StandardScaler()
scale_data = ss.fit_transform(data)
```

```
train_x, test_x, train_y, test_y = model_selection.train_test_split(data, data_y, test_size = 0.2, random_state = 99)
```

```
model = linear_model.LinearRegression() # LogisticRegression()
model.fit(train_x, train_y)

test_prediction = model.predict(test_x)
print('r-square of linear regression : {:.3f}'.format(metrics.r2_score(test_prediction, test_y)))
```

監督式學習

一些比程式碼更重要的事情

機器學習的流程



- 定義問題
- 依照問題與資料狀況選擇模型
- 資料前處理
- 建立模型
- 模型評估

資料前處理



- 遺漏值處理
 - 刪除有遺漏的資料 (row or column)
 - 遺漏值填補
- 極端值處理
 - 資料分布轉換
 - 取代極端值

資料前處理

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

- 將資料轉為數值型態
 - Label encoding
 - One-hot encoding

姓名	分數
HowHow	93
蔡哥	92
阿滴	90
HowHow	88
阿滴	95

姓名	分數
0	93
1	92
2	90
0	88
2	95

Label encoding

HowHow	蔡哥	阿滴	分數
1	0	0	93
0	1	0	92
0	0	1	90
1	0	0	88
0	0	1	95

One-hot encoding

資料前處理

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
```

- 將資料範圍限縮
 - Standard scale
 - Min-max scale

姓名	分數
HowHow	93
蔡哥	92
阿滴	90
HowHow	88
阿滴	95

姓名	分數
HowHow	0.518
蔡哥	0.148
阿滴	-0.592
HowHow	-1.332
阿滴	1.258

Standard scale

$$x_{standard} = \frac{x - \mu}{\sigma}$$

$$x_{minmax} = \frac{x - Min(x)}{Max(x) - Min(x)}$$

姓名	分數
HowHow	0.714
蔡哥	0.571
阿滴	0.286
HowHow	0.000
阿滴	1.000

Min-max scale

選擇模型

```
from sklearn.linear_model import LinearRegression, LogisticRegression
```

- 線性模型

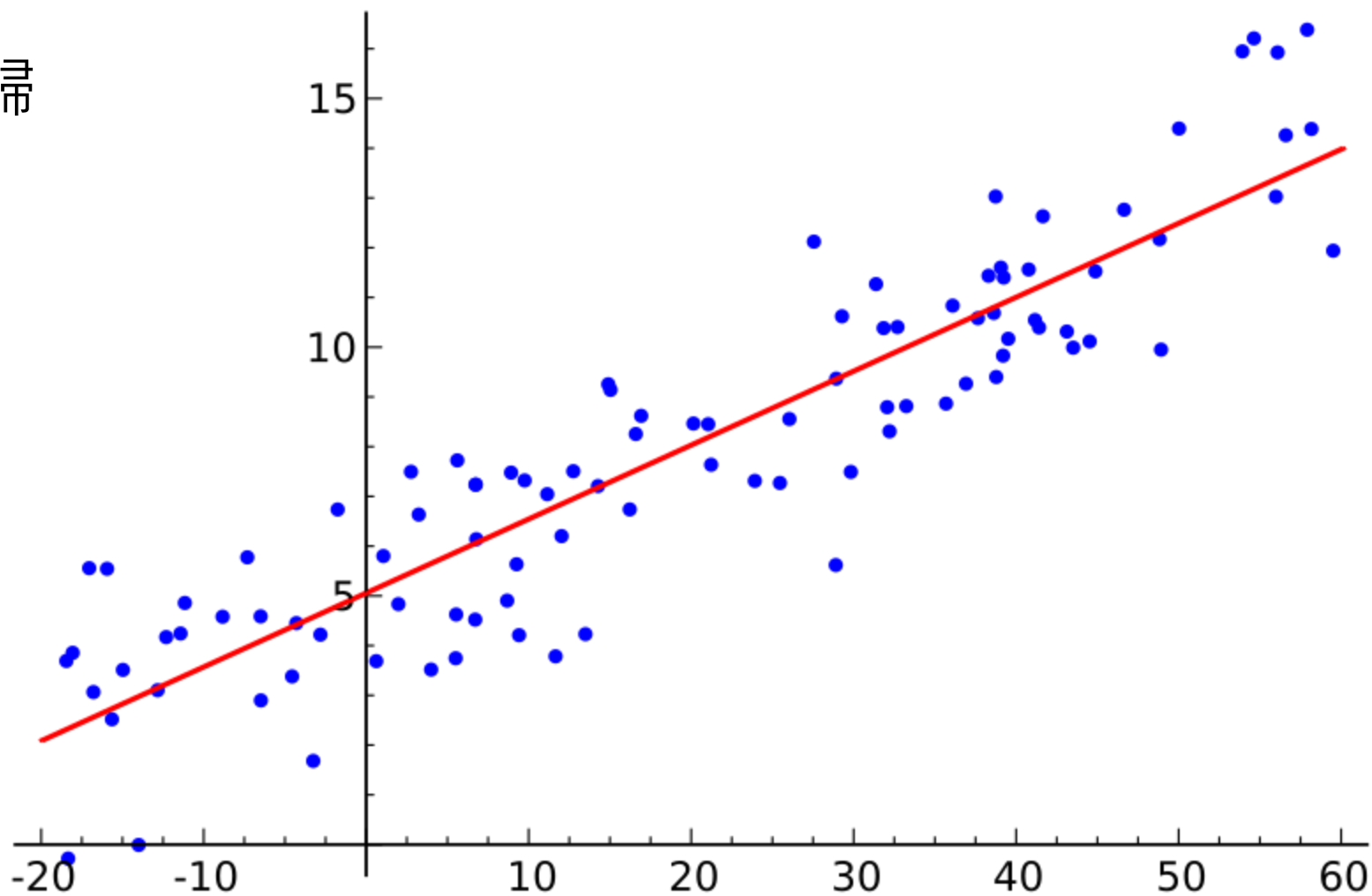
$$y = f(a_0 + a_1x_1 + a_2x_2 + \dots)$$

- 線性回歸 (Linear Regression)
- 邏輯式回歸 (Logistic Regression)
- Ridge regression
- Lasso regression

線性模型要找一條線能夠讓
資料與線的距離(誤差)最小

線性模型

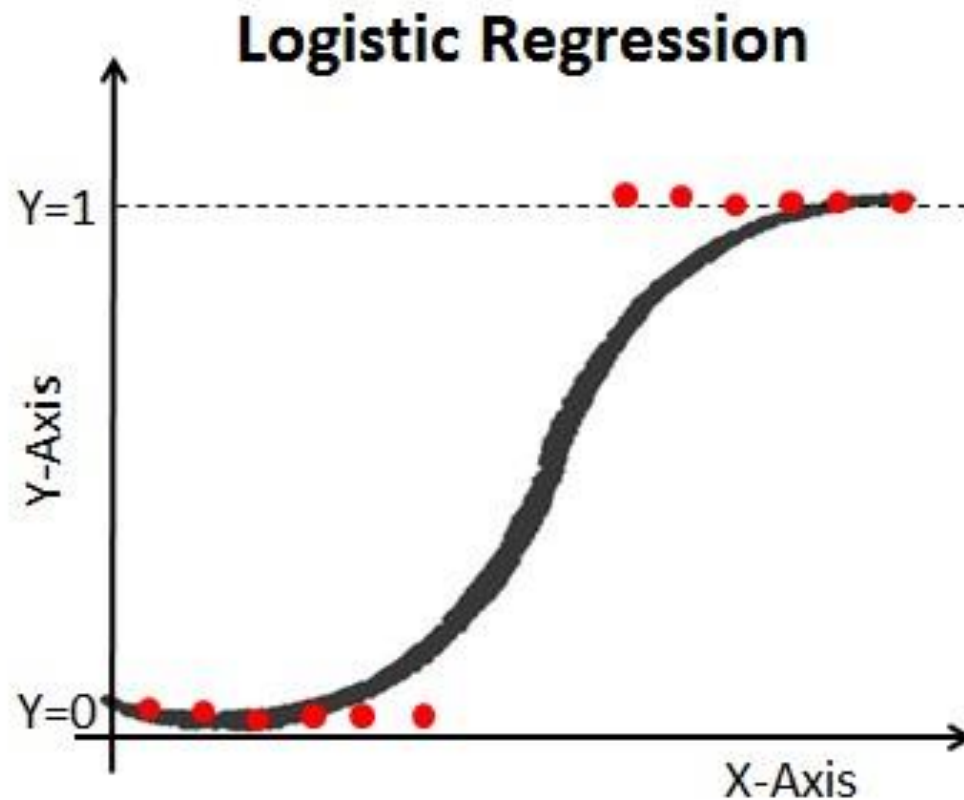
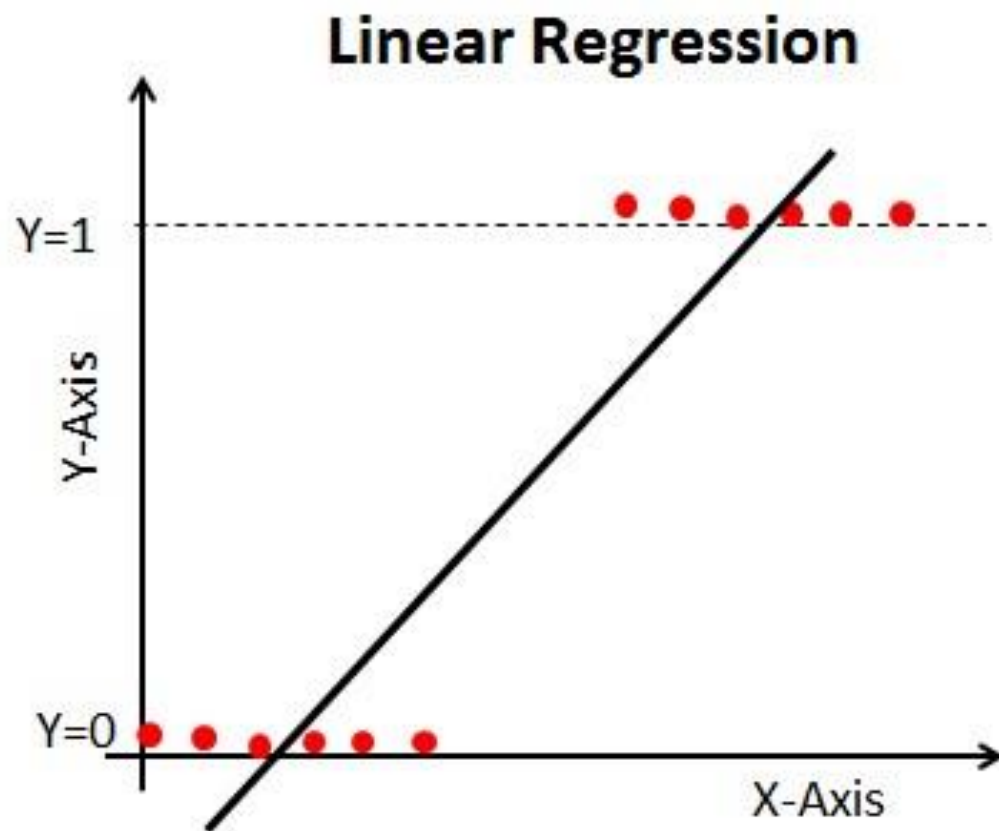
- 線性回歸



線性模型

- 邏輯式回歸

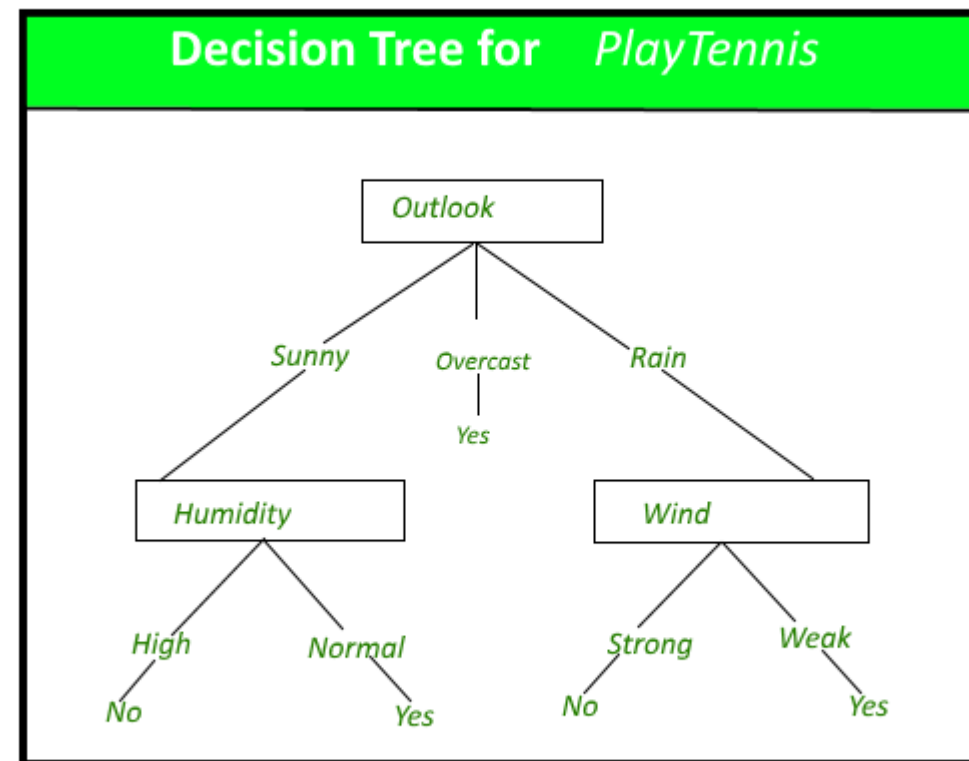
$$f(x) = \frac{1}{1 + e^{-(x)}}$$



選擇模型

```
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor  
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
```

- 樹狀模型
 - 決策樹 (Decision Tree)
 - 隨機森林 (Random Forest)



樹狀模型要不斷地找好的切分點將資料分割

選擇模型

- 模型特性比較

- 線性模型

- 著重整體趨勢
 - 有資料的假設，容易受極端值影響

encoding 方式會影響模型

資料需作標準化

- 樹狀模型

- 沒有資料假設，單一樹規則明確
 - 對資料局部關係敏感，容易過度配適

資料不需作標準化

選擇模型



- 其他模型
 - 支持向量機 (Support Vector Machine, SVM)
 - 最近鄰居法 (K-Nearest Neighbor, KNN)
 - 集成學習 (ensemble methods)
 - 神經網路 (neural network)

模型評估

```
from sklearn.model_selection import train_test_split
```

- 評估模型在未知的資料上的表現



在模型配適前，通常會留一份有正確答案的資料作為驗證資料

模型評估 - 回歸

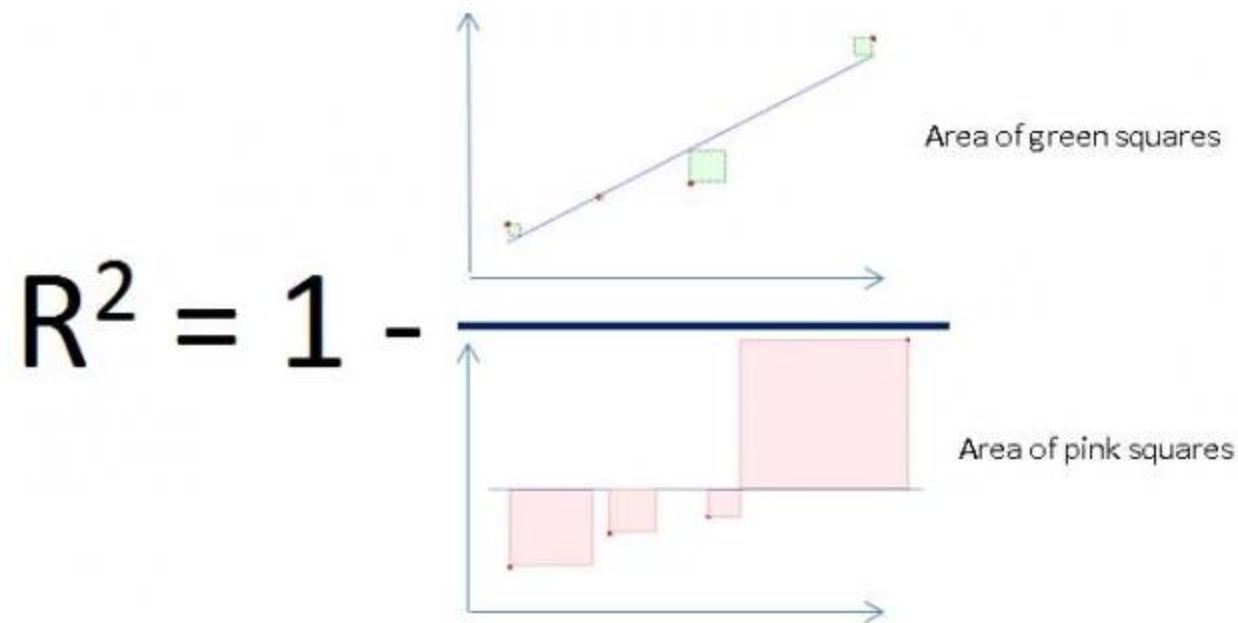
```
from sklearn.metrics import r2_score, mean_squared_error,  
mean_absolute_error
```

- 回歸
 - 解釋量 (R^2)
 - 均方差 (Mean Squared Error, MSE)
 - 平均絕對差 (Mean Absolute Error, MAE)

模型評估 - 回歸

- 解釋量 (R^2)

- 介於 0~1 之間
- 數值越高代表誤差越小



模型評估 - 回歸

- 均方差 (MSE)
- 平均絕對差 (MAE)
 - 與被預測的數值範圍有關
 - 數值越低代表誤差越小

$$MSE = \frac{1}{n} \sum \underbrace{\left(y - \hat{y} \right)^2}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}}$$

$$MAE = \underbrace{\frac{1}{n}}_{\substack{\text{Divide by the total} \\ \text{number of data points}}} \sum \underbrace{\left| \underbrace{y}_{\substack{\text{Actual output value}}} - \underbrace{\hat{y}}_{\substack{\text{Predicted output value}}} \right|}_{\substack{\text{Sum of} \\ \text{The absolute value of the} \\ \text{residual}}}$$

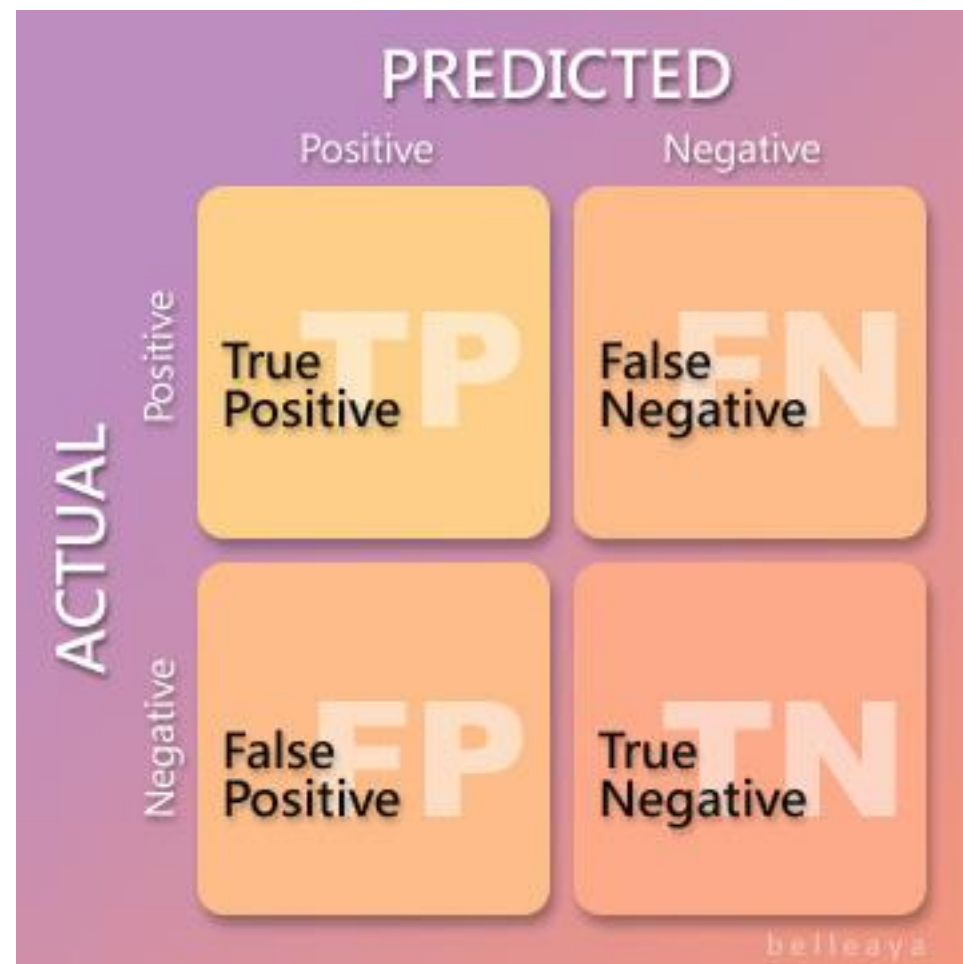
模型評估 – 分類

```
from sklearn.metrics import confusion_matrix, accuracy_score,  
precision_score, recall_score
```

- 分類
 - 混淆矩陣 (Confusion matrix)
 - 正確率 (Accuracy)
 - 準確率 (Precision)、召回率 (Recall)
 - [AUC](#)、[f1 score](#)

模型評估 - 分類

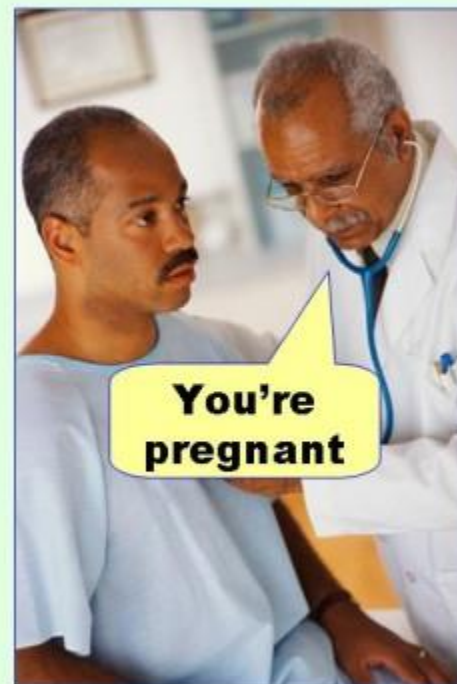
- 混淆矩陣 (Confusion matrix)
 - 詳細呈現模型預測狀況



模型評估 – 分類

- 混淆矩陣 (Confusion matrix)
 - False Negative 與 False Positive

Type I error
(false positive)



Type II error
(false negative)



模型評估 - 分類

- 正確率 (Accuracy)

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive TP	False Negative FN
	Negative	False Positive FP	True Negative TN

belleya

模型評估 - 分類

- 準確率 (Precision) 與 召回率 (Recall, Specificity)

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Precision} = \frac{TP}{TP + FP}$$

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Specificity} = \frac{TN}{TN + FP}$$

其他議題 – 監督式學習

- 特徵工程與特徵選取
- 更強大/複雜的模型
- 過度配適與模型泛化
- 參數選取

非監督式學習

一些比程式碼更重要的事

機器學習的流程



- 定義問題
- 依照問題與資料狀況選擇模型
- 資料前處理
- 建立模型
- 模型評估

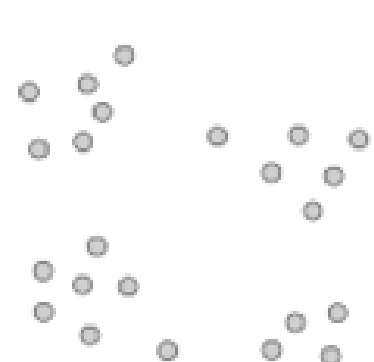
定義問題與選擇模型

- 分群 (clustering)
 - K-means 分群法
 - 階層式分群法
 - DBSCAN

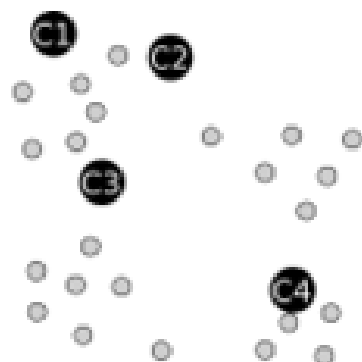
分群的目的是將相似特性的資料分作同一個群體

K-means 分群

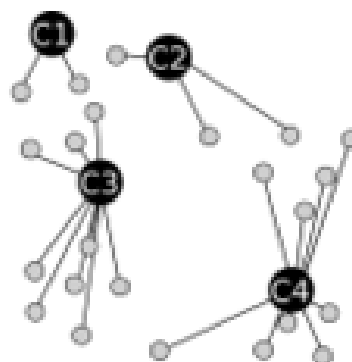
```
from sklearn.cluster import KMeans
```



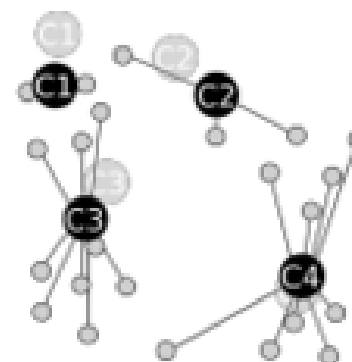
0a. Données d'entrée



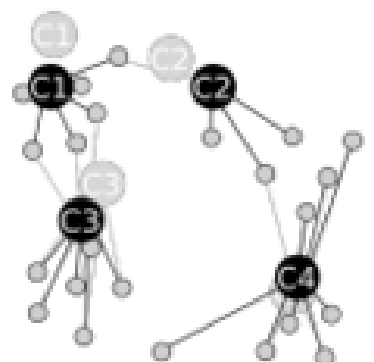
0b. intialisation



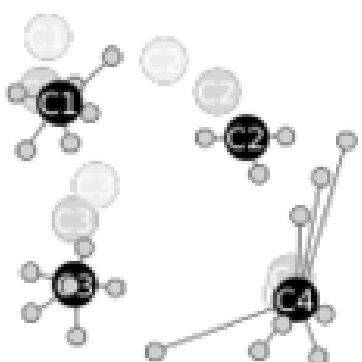
1a. assignation



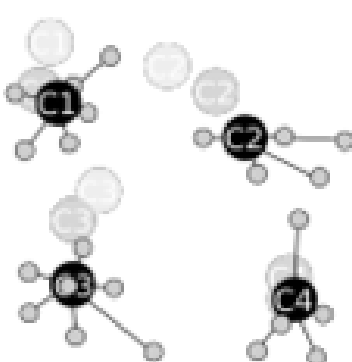
1b. calcul des points moyens



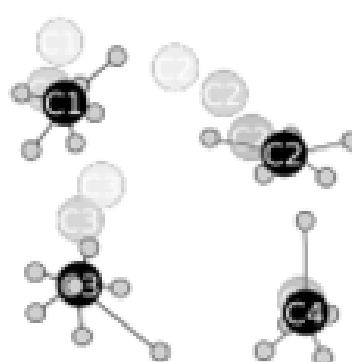
2a. assignation



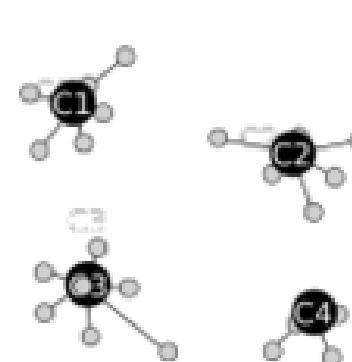
2b. calcul des points moyens



3a. assignation



3b. calcul des points moyens



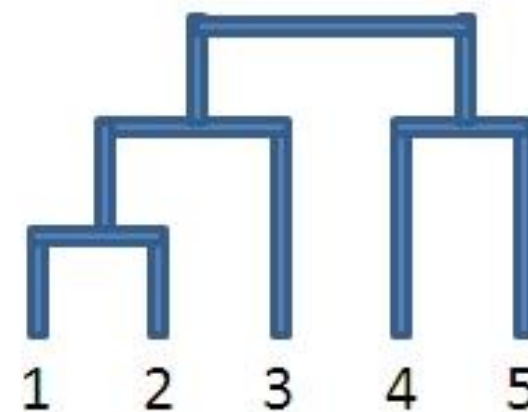
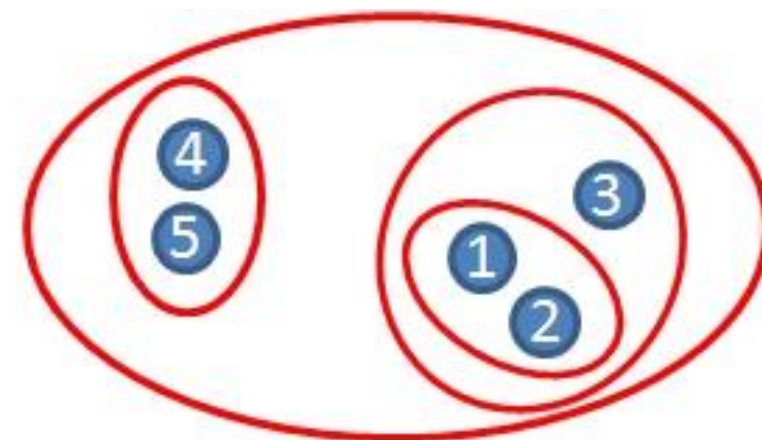
4a. assignation
clusters stables (fin)

階層式分群

```
from sklearn.cluster import AgglomerativeClustering
```

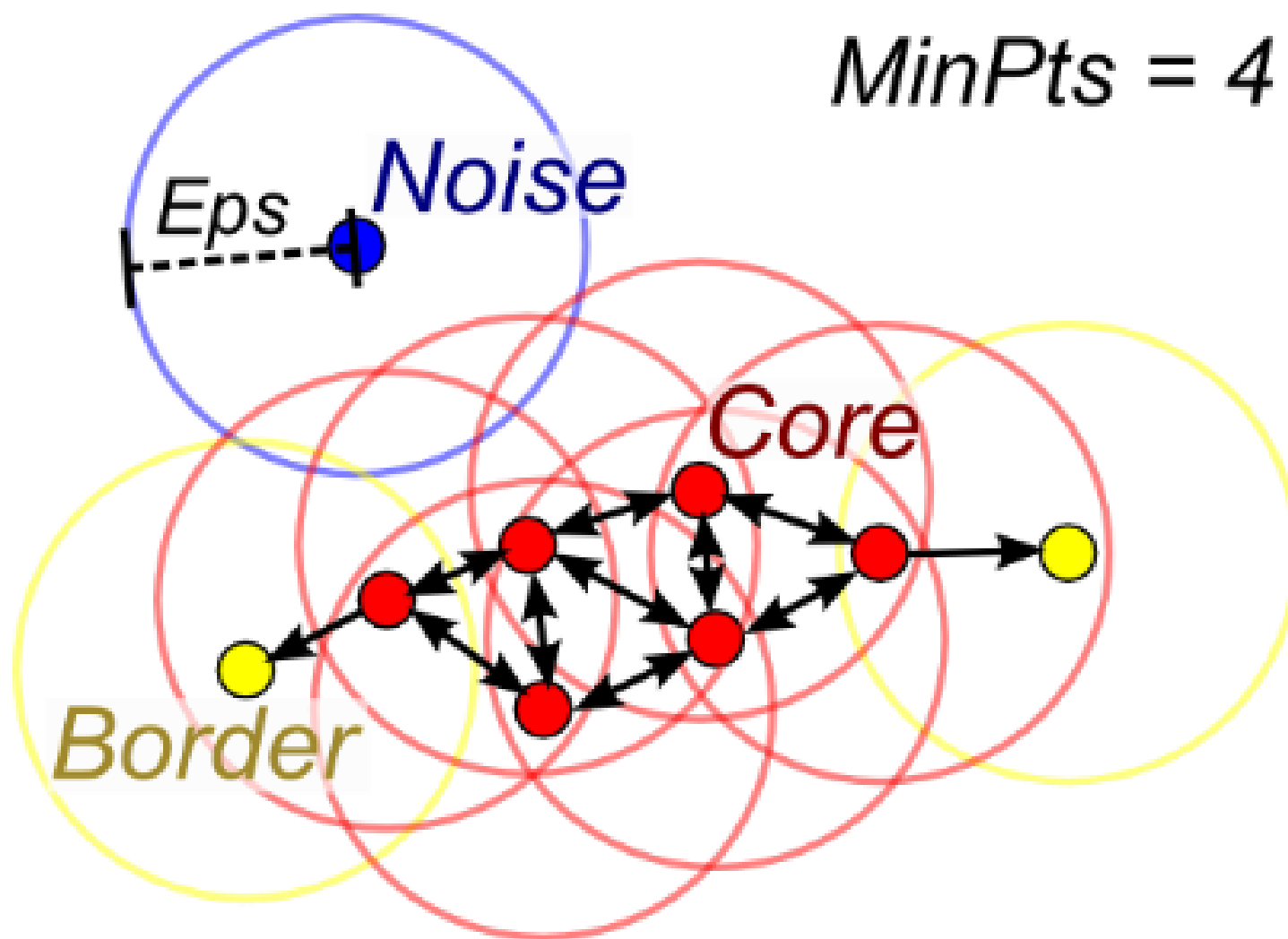


階層式



DBSCAN分群

```
from sklearn.cluster import DBSCAN
```



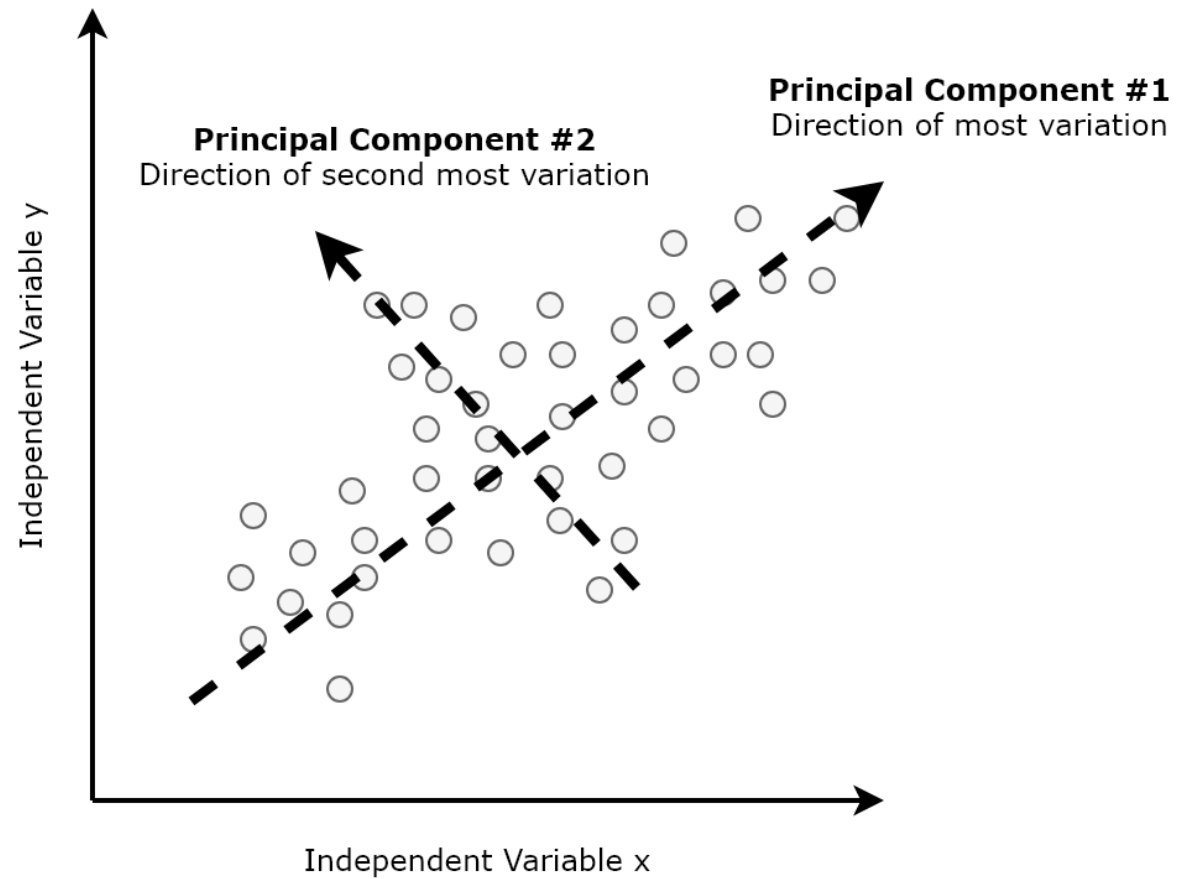
定義問題與選擇模型

```
from sklearn.decomposition import PCA
```

- 維度縮減 (dimension reduction)
 - Principal component analysis, PCA

維度縮減的目的是試圖用較低維度的座標描述高維度的資料

PCA



PCA是對資料作座標轉換，
並且依序分離出重要程度較高的軸

模型評估



- 分群
 - 分群質量評估
- 維度縮減
 - 訊息量損失比例 ($1 - \text{可解釋變異}$)

補充資料

特徵工程與特徵選取

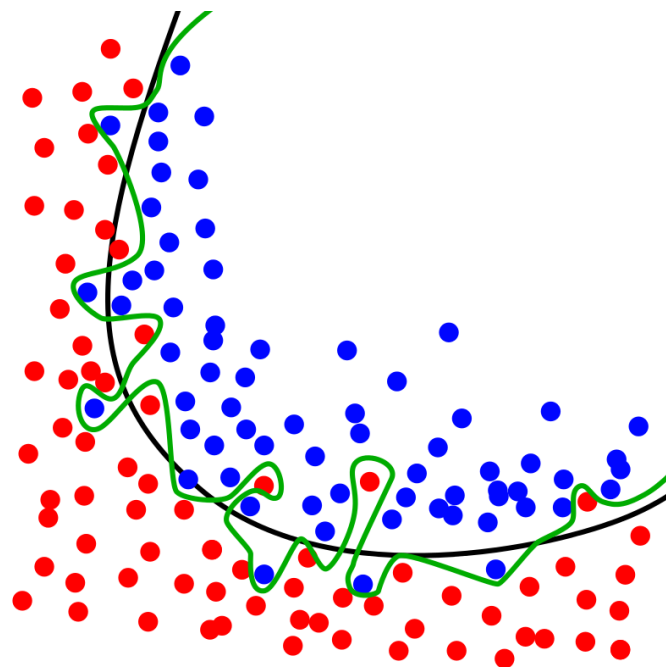
- 產生其他特徵
 - 領域知識
 - 資料探索
- 選擇有效的特徵
 - 相關係數
 - Lasso, Ridge regression
 - 特徵重要程度

其他機器學習模型

- 目前比賽優勝者較常使用的模型
 - Bagging : [Random Forest](#)
 - Boosting : [XGBoost](#), [LightGBM](#), [CatBoost](#)
 - Neural Network
 - Stacking model

過度配適與模型泛化

- 只要模型能力夠強，就一定能完全正確預測訓練資料
- 如何發現模型已過度配適資料？
- 如何解決降低過度配適的狀況？
 - 降低模型能力
 - 使用ensemble方法



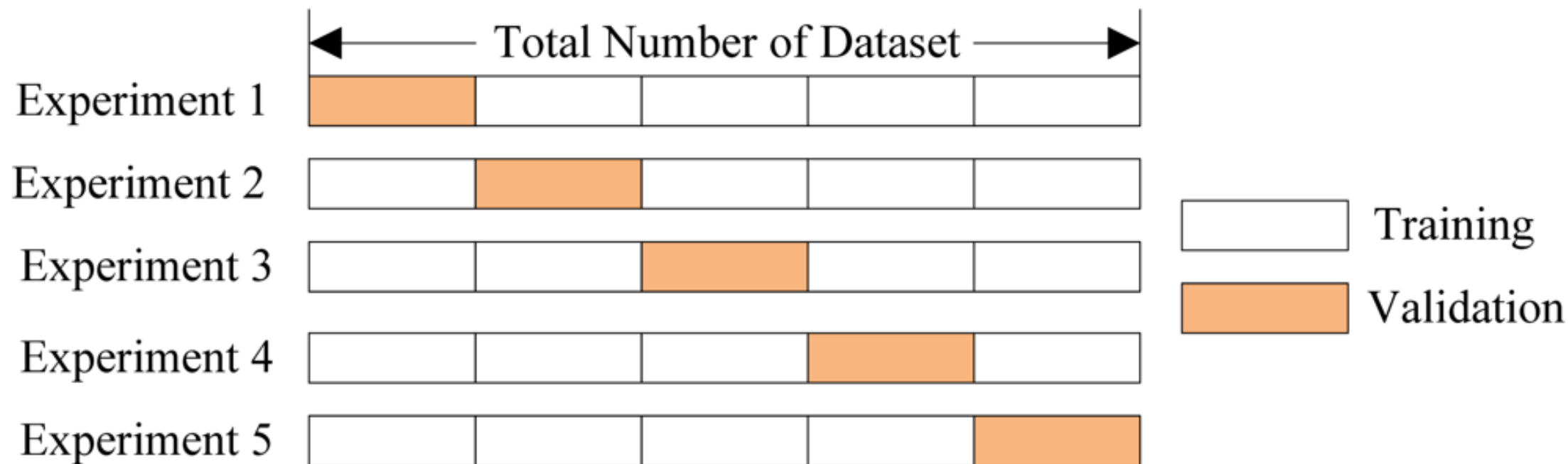
參數選取

```
from sklearn.model_selection import GridSearchCV
```

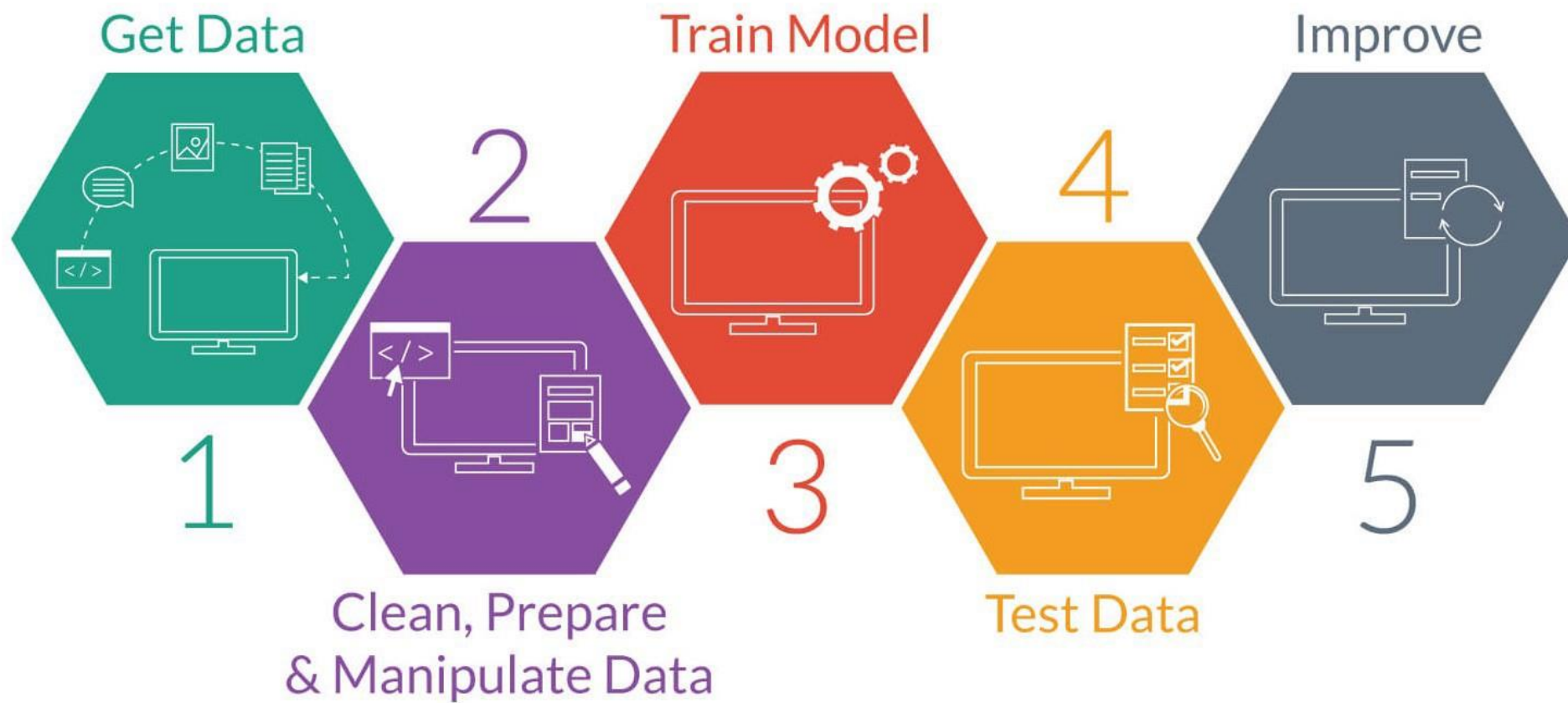
- 改變模型的參數會稍稍影響模型的能力，進而對於資料的預測能力更好
- 如何選擇參數？
- 如何確認選擇的參數不會造成過度配適？

參數選取

Cross Validation



機器學習總結



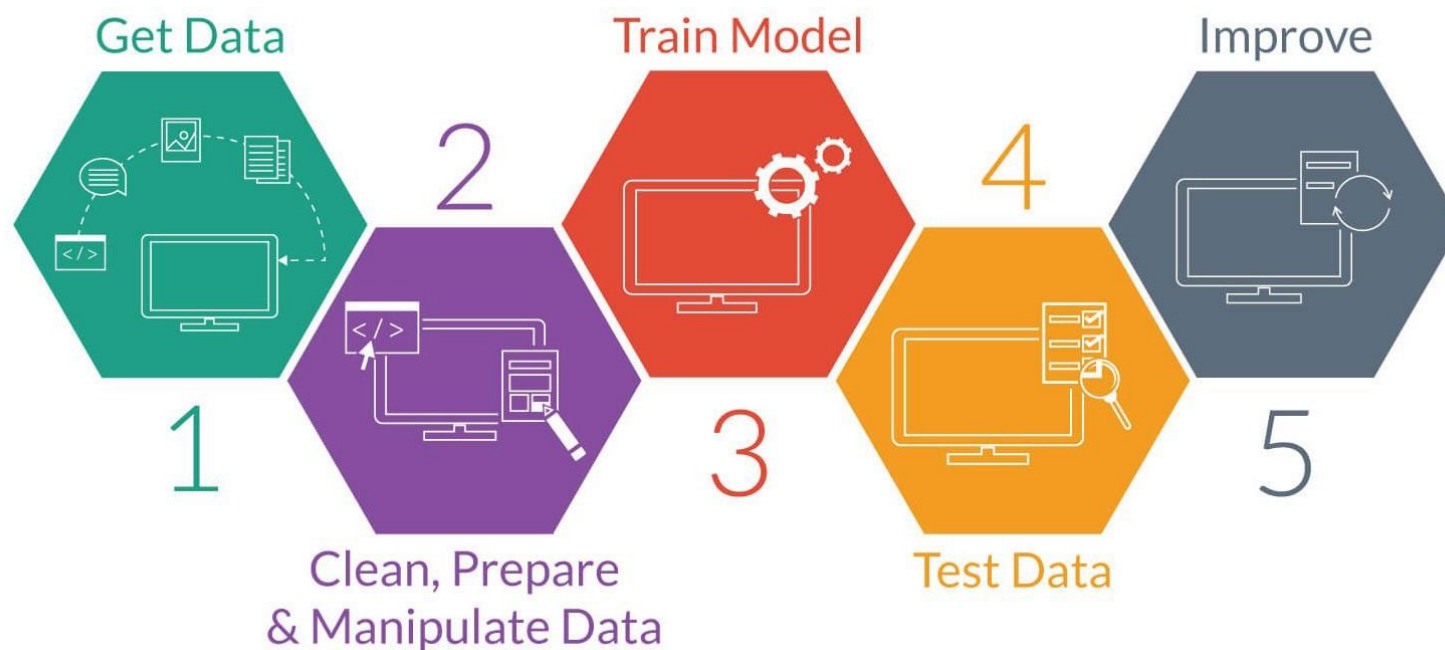
如何得到一個好的模型？

- 資料面

- 資料前處理
- 資料探索
- 特徵工程

- 模型面

- 參數調整
- 交叉驗證



相關資源

- 課程
 - [台大李宏毅老師個人網站](#)
 - [機器學習基石 – 台大林軒田老師](#)
- 實戰
 - [Kaggle](#)
 - [AIdea](#)
 - [Tbrain](#)
- 概念
 - [圖解機器學習](#)