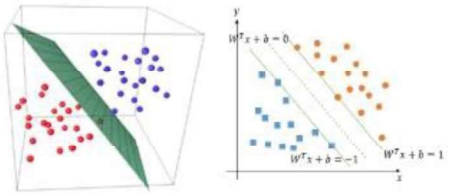
機器學習

對應不同的學習型態，有不同的機器學習演算法，主要列出幾種常見的演算法：

1. 迴歸分析Regression
   - 利用變數之間的關係建立迴歸模型，常見的迴歸演算法有Ordinary Least Squares、Logistic Regression、Linear Regression等
2. 聚類方法Clustering Method
   - k-means algorithm演算法是一個聚類演算法，把n的物件根據他們的屬性分為k個分割，程式試圖找到資料中自然聚類的中心，並假設物件屬性來自于空間向量，讓目標是使各個群組內部的均方誤差總和最小。
3. 決策樹Decision Tree Learning
   - 建立一個根據資料中實際值決策的模型，用來解決歸納和迴歸問題。常見的演算法有Classification and Regression Tree (CART)、C4.5及Random Forest等
4. 貝氏方法Bayesian method
   - 貝氏模型(Naive Bayes Model)發源於古典數學理論，有著堅實的數學基礎，以及穩定的分類效率。

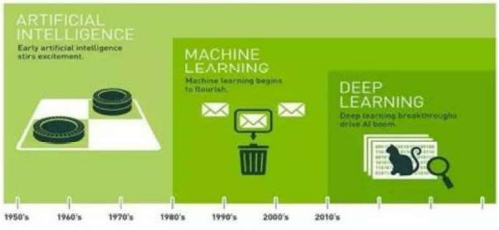
機器學習



對應不同的學習型態，有不同的機器學習演算法，主要列出幾種常見的演算法：

5. K最近鄰居(k-Nearest Neighbor，KNN)分類演算法
   - 樣本在特徵空間中的k個最相似(即特徵空間中最鄰近)的樣本中的大多數屬於某一個類別，則該樣本也屬於這個類別。
6. 聯合規則學習Association Rule Learning(決策樹)
   - Apriori algorithm是用來對資料間提取規律的方法，通過這些規律可以發現巨量多維空間資料之間的聯絡，而這些重要的聯絡可以被組織拿來使用。
7. 自適應增強演算法Adaptive Boosting(隨機森林)
   - Adaboost是一種反覆運算演算法，其核心思想是針對同一個訓練集訓練不同的分類器(弱分類器)，根據不同的錯誤率給定不同權重，把這些弱分類器集合起來，構成一個更強的最終分類器 (強分類器)。
8. 支援向量機Support Vector Machine
   - SVM將資料映射到一個更高維的空間裡，在這個空間裡建立有一個間隔超平面，在分開資料的超平面的兩邊建有兩個互相平行的超平面，分隔超平面使兩個平行超平面的距離最大化。假定平行超平面間的距離或差距越大，分類器的總誤差越小。

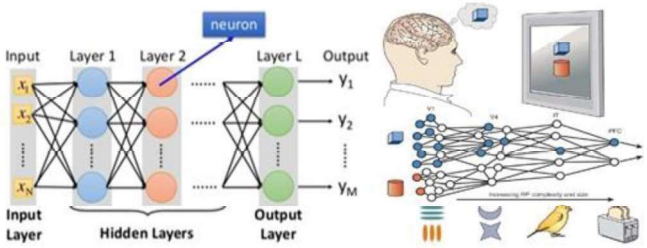
機器學習



對應不同的學習型態，有不同的機器學習演算法，主要列出幾種常見的演算法：

9. 類神經網路Neural networks

- Artificial Neural Networks(人工神經網路)是一種模仿生物神經網路結構和功能的數學模型或計算模型，用於對函式進行估計或近似，神經網路由大量的人工神經元聯結進行計算。**此演算法為非監督式學習的一種，也是開啟深度學習的始祖**。
- 人工智慧因機器學習起了頭，以監督式學習，透過資料特徵及特定標籤，讓機器可以根據給定的特徵去判斷應得的結果，但在大數據時代，許多資料往往沒有這麼完整，無法將每筆資料整理成特定格式，故無法利用監督式學習相關的演算法進行預測分析，只能選擇需要較多硬體資源的非監督式學習相關的演算法進行分析，像是類神經網路演算法。



類神經網路演算法，即仿造人腦的運作方式讓電腦進行計算，以**多節點及分層**的概念擷取資料特徵進行建模，結構上分為三部分：一、輸入層(Input layer)：多個神經元(Neuron)接受大量的非線性資料訊息，稱為輸入向量。二、輸出層(Output layer)： 訊息在神經元鏈接中傳輸分析而產生結果，稱為輸出向量。 三、隱藏層(Hidden layer)： 是輸入層和輸出層之間眾多神經元和鏈接組成的各個層面。隱藏層可以有多層，節點(神經元)數目不定，習慣上會用一層，數目越多則非線性越顯著，但計算時間較多。

機器學習(Machine Learning)

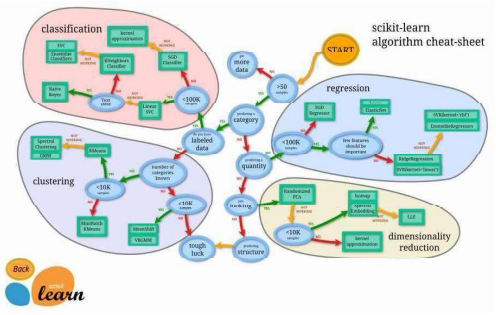
使用 Python 實作機器學習

- 機器學習是一門設計如何讓演算法能夠學習的電腦科學，讓機器能夠透過觀察已知的資料學習預測未知的資料。典型的應用包含**概念學習(Concept learning)**、**函數學習(Function learning)**、**預測模型(Predictive modeling)**、**分群(Clustering)與找尋預測特徵(Finding predictive patterns)**。終極目標是讓電腦能夠自行提升學習能力，預測未知資料的準確性能夠隨著已知資料的增加而提高，節省使用者人工調整校正的精力。
- 機器學習跟**知識發掘(Knowledge Discovery)**、**資料採礦(Data Mining)**、**人工智慧(Artificial Intelligence, AI)**以及**統計(Statistics)**有著密不可分的關係，應用範圍從學術研究到商業應用，從機器人科學家到垃圾郵件篩選與推薦系統，都可見其蹤影。
- 要使用的 Python 機器學習套件是 **scikit-learn(SKlearn)**，它建構於 NumPy、SciPy 與 Matplotlib 之上，是開源套件並可作為商業使用。

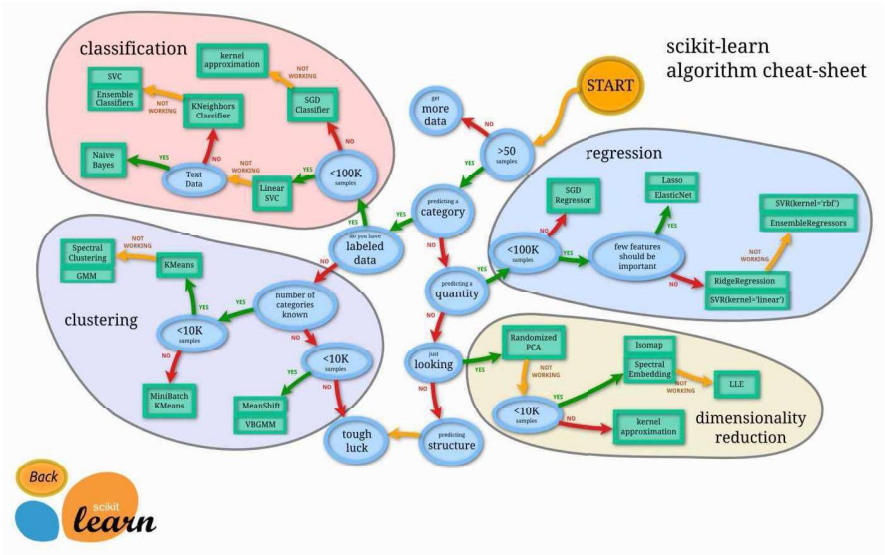
機器學習(Machine Learning)

使用 Python 實作機器學習

- SKlearn在python中提供大量常見的機器學習演算法和許多實用的資料集合，像是Iris以及手寫辨識數字的資料。而演算法的部分，可以在SKlearn官網中看到，將功能分為6個部分：**Classification**、**Regression**、**Clustering**、**Model selection**、**Preprocessing**、**Dimensionality reduction**，Classification、Regression和Clustering都算是做**學習演算法**的部分，Preprocessing和Dimensionality reduction比較算是**資料前處理**的部分，Model selection看來應該是**比較不同演算法的結果**。各個適合的演算法，在SKlearn中也有做相當清楚的圖表，呈現該演算法資料輸出的型態，相當方便。



機器學習(Machine Learning)



這是一幅很有用的流程圖，可以幫助決定何時該採用那種預測的方法，例如：

- 例如假如資料筆數**沒有超過50筆**，那麼應該去取得更多data再來進行。
- 如果要預測的是某個資料的類別(Category)，那麼有針對既有data定義Category了嗎？若沒有，那麼採用的方法就應是非監督學習當中的Clustering或Dimensionality reduction，反之則使用監督學習Classification，並視Sample資料筆數來決定Liner SVC或SGD Classifier的使用。
- 如果要預測的是數值而非類別，那麼Regression是唯一的方式。而Regression又分成好幾種方法，端視資料筆數以及是否加樣比重某些特徵值而定。

scikit-learn

從 scikit-learn 套件的首頁可以一目瞭然它的應用領域：

監督式學習(Supervised learning)

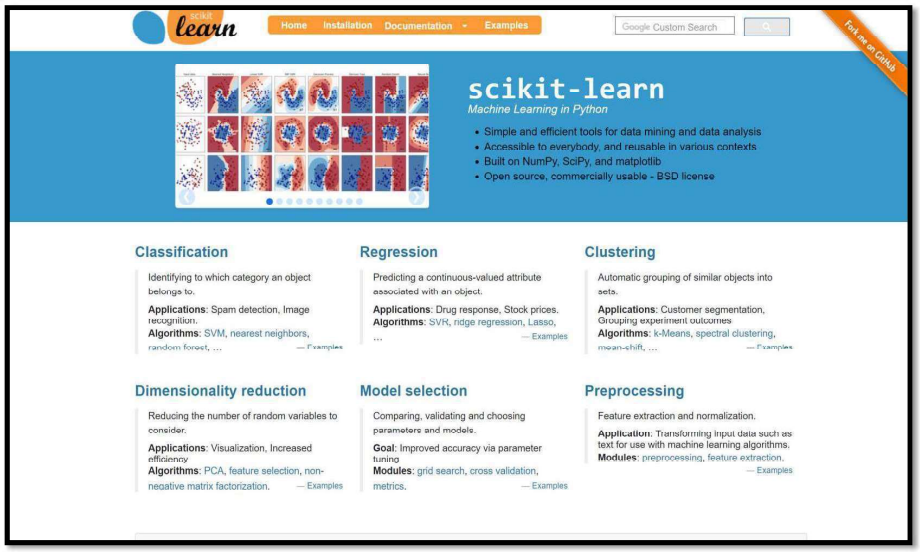
- 分類(Classification)
- 迴歸(Regression)

非監督式學習(Unsupervised learning)

- 分群(Clustering)
- 降維(Dimensionality reduction)

模型選擇(Model selection)

預(前)處理(Preprocessing)

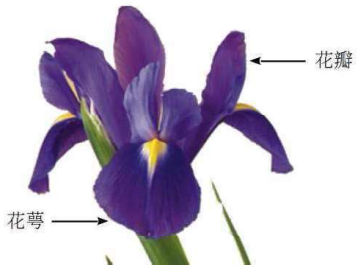


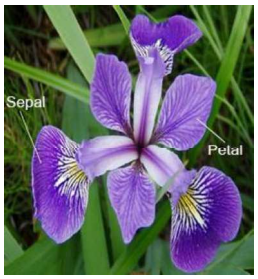
scikit-learn

在 scikit-learn 套件中，資料集以 Bunch 物件格式儲存，類似 Python 字典，包含鍵與值，優點是元素的存取除了一般的 <dict>['<key>'] 字典方式外，也可使用點號方式 <dict>.<key>

主要有以下陣列：

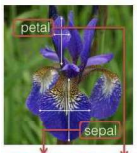
- 特徵陣列 (Feature matrix) 通常命名為 X：
  - 列 (Row)：資料集的樣本
  - 欄 (Column)：資料集的屬性
- 目標陣列 (Target matrix) 通常命名為 y：
  - 每個值即為每個樣本的類別





例如安德森鳶尾花資料集 (Anderson's Iris dataset)：鳶尾花的數據，包括花萼 (Sepal，下垂) 與花瓣 (Petal，直立) 的長度與寬度，品種則包括：山鳶尾 (Setosa)、變色鳶尾 (Versicolor)、 與維吉尼卡鳶尾 (Virginica)

scikit-learn



Height Width	Height Width	Class label
5.1,3.5	1.4,0.2,	Iris-setosa
4.9,3.0	1.4,0.2,	Iris-setosa
4.7,3.2	1.3,0.2,	Iris-setosa
4.6,3.1	1.5,0.2,	Iris-setosa
5.0,3.6	1.4,0.2,	Iris-setosa
5.4,3.9	1.7,0.4,	Iris-setosa

玩具資料(Toy datasets)

- 在練習資料視覺化或者機器學習的時候，除了可以自己產生資料以外，也可以用所謂的玩具資料(Toy datasets)，玩具資料並不是一個特定的資料，而是泛指一些小而美的標準資料。
- 使用 sklearn 的 datasets 物件的**load_iris()**方法來讀入鳶尾花資料。

```
from sklearn import datasets
import pandas as pd

iris = datasets.load_iris()
print(type(iris.data)) # 資料是儲存為 ndarray
print(iris.feature_names) # 變數名稱可以利用 feature_names 屬性取得
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names) # 轉換為 data fr
ame
iris_df.ix[:, "species"] = iris.target # 將品種加入 data frame
iris_df.head() # 觀察前五個觀測值
```

```
In [40]: from sklearn import datasets
         import pandas as pd

         iris = datasets.load_iris()
         print(type(iris.data)) # 資料是儲存為 ndarray
         print(iris.feature_names) # 變數名稱可以利用 feature_names 屬性取得
         iris_df = pd.DataFrame(iris.data, columns=iris.feature_names) # 轉換為 data frame
         iris_df.ix[:, "species"] = iris.target # 將品種加入 data frame
         iris_df.head() # 觀察前五個觀測值

         <class 'numpy.ndarray'>
         ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

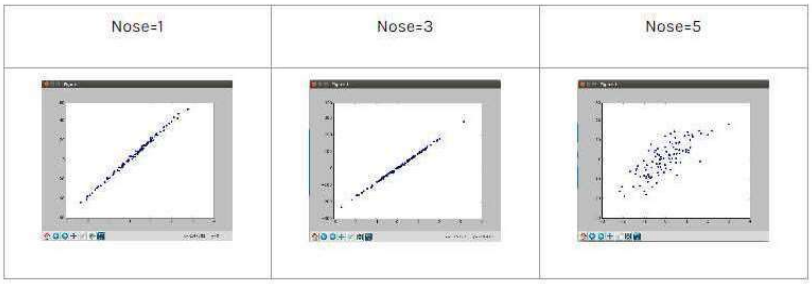
Out[40]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

- 還有其他更多的玩具資料，像是波士頓房地產資料可以透過**load_boston()**方法讀入，糖尿病病患資料可以透過**load_diabetes()**方法讀入。

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

scikit-learn



讓機器自己產生數據資料

- 很多時候，出於實驗測試或提供機器學習，除了這些實際搜集的數據之外，也可以讓機器自己來產生這些數據，以LinearRegression模型為例，用的是**make_regression**方法。先看看下方的例子：

```
from sklearn import datasets

import matplotlib.pyplot as plt

X,y = datasets.make_regression(n_samples=100, n_features=1, n_targets=1, noise=1)

plt.scatter(X, y)

plt.show()
```

在程式中引入了matplotlib模組，讓程式開發人員很容易就能將數字資料轉化成圖像。

make_regression參數中的n_samples表示要產生100組data，n_features及n_targets指定dataset的feature及target都是一個，noise=1則代表資料的離散程度。最後將feature、target所代表的x, y點畫出來。