

python 與機器學習-以 Abalone 資料為例

沈郁婷 副統計分析師

機器學習

在傳統的應用上，若要讓系統能夠達成媲美人工智慧執行方式，必須將規則及程式碼清楚的寫入系統當中，而機器學習(Machine Learning)則是讓電腦利用過去的資料，不斷從中自行學習、自行改善，最後漸漸得到該項規律的一種方式。

機器學習能夠應用在各種地方，例如：讓電腦判斷圖片內的動物是貓還是狗、由 google 開發的無人車，讓電腦代替人類進行自動駕駛、又或者是著名的 alphago，利用電腦自行從棋譜內學習下棋的策略，最後成功打敗了各個著名的棋士。由 alphago 的例子可以知道，圍棋的規則較為複雜、棋盤較大、會改變整體情況的因素太多，故不是將基本的圍棋規則寫入系統內系統就可以判斷最合適的下一步。而機器學習就適合用在這種多特徵、較無直觀規律可循的情境上。

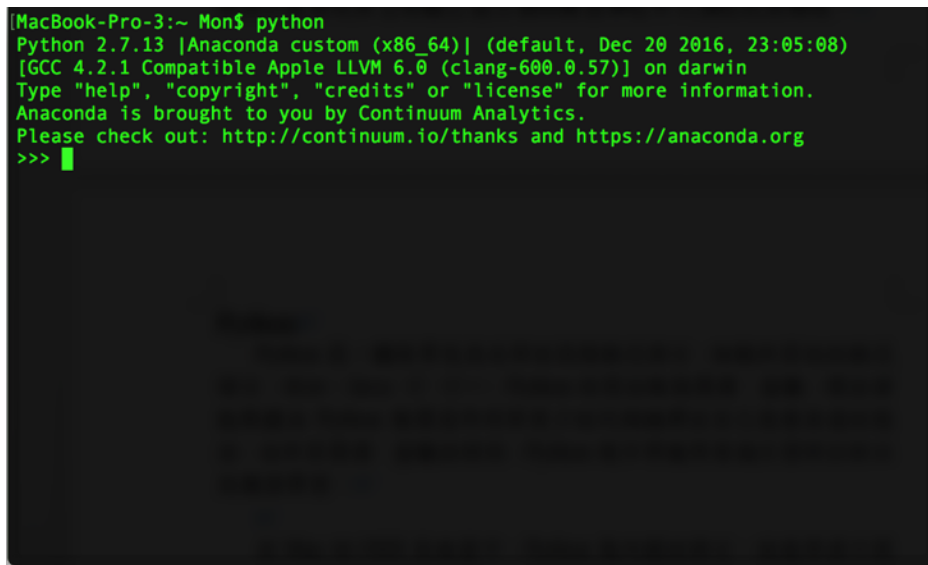
在傳統的機器學習中，一般來說可分為兩大類，監督式學習及非監督式學習。監督式學習是一種利用訓練資料集讓電腦從中學習、推理出一個模型，此模型是由訓練資料中的特徵值以及標記(label)所計算出來的，故有新資料時，可以利用新資料的特徵值讓此模型自動標示此資料之標記，在訓練模型時所用的標記為人為所標示的。

監督式學習通常被應用於解決“分類”問題，用於判斷行為是否有改變。非監督式學習則是電腦利用訓練資料內的特徵值進行學習、推理，最後得出一個模型，此模型會依照特徵相近的資料群聚在一起，故有新資料進來時，可以利用新資料的特徵值讓模型自動將資料群聚。而兩種學習方式最大的差別在於資料是否有人為標記，兩種學習的輸入皆為資料的特徵值，而輸出的差別在於監督式學習的輸出通常為訓練資料集中人為標記的結果。

Python

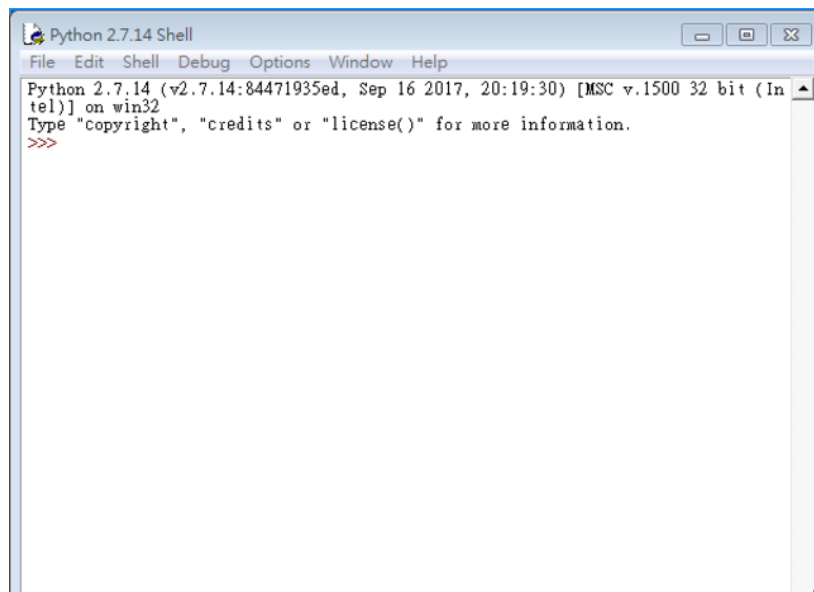
Python 是一種很常見及泛用的高階程式語言，相較於其他的程式語言，例如：Java、C、C++，Python 的寫法較為簡潔、直觀，開發者能夠藉由 Python 較簡易即利用更少的代碼編譯出自己想要表達的想法，由於其簡潔、直觀的特性，Python 現今常被用來進行資料分析以及機器學習。

在 Mac 的 OSX 系統當中，Python 為內建的語言，故使用者只需要在終端機內輸入”python”，即可進入 Python 的編譯器。如下圖所示：



```
MacBook-Pro-3:~ Mon$ python
Python 2.7.13 [Anaconda custom (x86_64)] (default, Dec 20 2016, 23:05:08)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>>
```

而在 Windows 系統中，由於 Python 並非內建的語言，故必須經由網路從官方網站中下載，安裝完成後即可從開始→所有程式中的 Python2.7 資料夾內打開 IDLE 進行編譯。如下圖所示：



```
Python 2.7.14 Shell
File Edit Shell Debug Options Window Help
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:19:30) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Abalone 資料集

UCI 資料庫是著名的開放資料庫，有著各式各樣的資料集，裡面的資料常常被學者用來進行分析及實驗，也是機器學習初學者取得自我學習資料庫的一個好地方。而 Abalone 資料集在 UCI 資料庫內是最被常使用的前 10 名。這個資料集的特性為資料容易解讀、是真實的資料、特徵值內容直觀、資料量夠大。

此資料集在一開始是被用於預測 Abalone 的年齡，由於要分辨 Abalone 的年齡必須利用顯微鏡觀察 Abalone 外殼上環的數量，這種分辨年齡的行為是無聊且繁瑣的，故資料科學家利用其他的特徵值對 Abalone 的年齡進行預測，而這一期 eNews 將利用此資料集，用各種 Abalone 表面的特徵值，對 Abalone 的肉大小進行預測，此預測在未來能夠幫助 Abalone 養殖農直接利用表面特徵來預測肉的大小，不需將 Abalone 剖開，如此能夠讓 Abalone 達到養殖的最大效益。

在此資料集當中，總共有九種特徵值，分別是 Sex、Length、Diameter、Height、Whole Weight、Shucked Weight、Viscera Weight、Shell Weight、Ring，每個特徵代表的意義如下表所示。

Feature	Mean
Sex	Abalone 的性別
Length	殼的最長測量值
Diameter	殼垂直測量的長度
Height	內部肉的垂直測量長度
Whole Weight	整個 Abalone 的重量(帶殼)
Shucked Weight	Abalone 的重量(不帶殼)
Viscera Weight	Abalone 的內臟重量
Shell Weight	Abalone 殼的重量
Ring	年齡

此次實驗將利用 Length、Diameter、Whole Weight 此三個表面特徵值，對 Shucked Weight 進行預測，此實驗將會將 Shucked Weight 先標記三個類型(大、中、小)，再利用選擇的三個特徵值對 Shucked Weight 進行監督式學習。

此次實驗將利用隨機森林演算法(Random Forest)當作是機器學習的演算法。

隨機森林演算法(Random Forest)

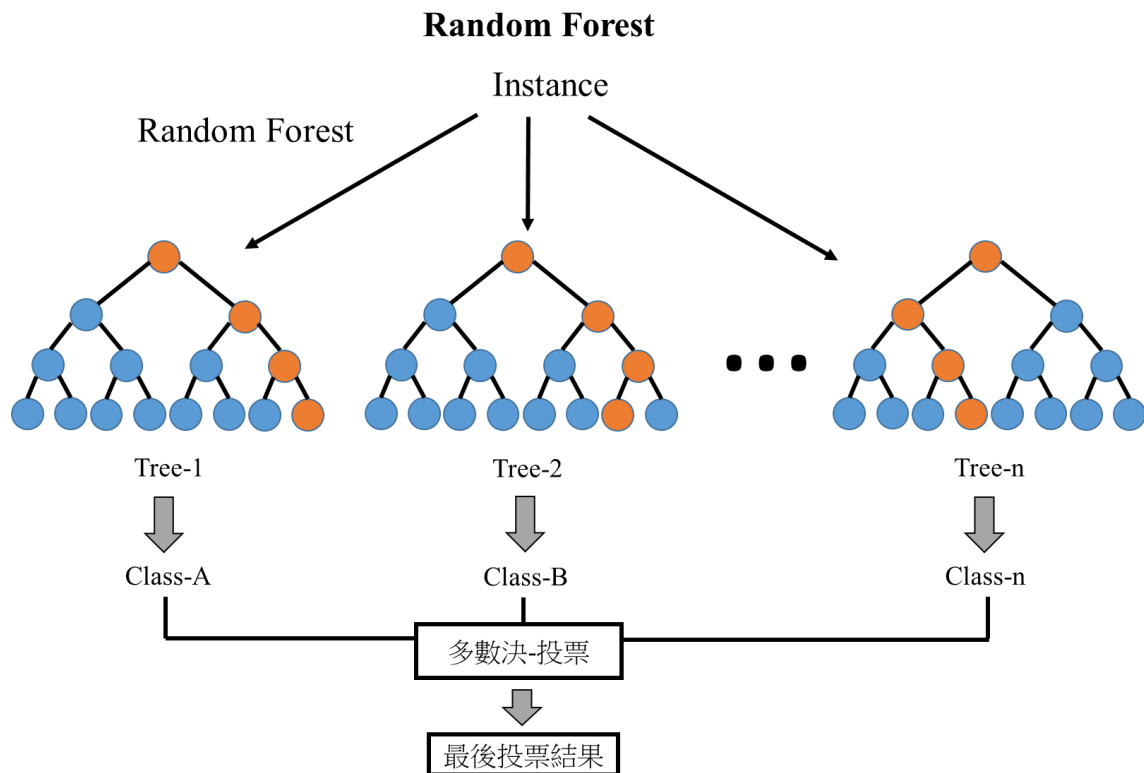
隨機森林是透過整體學習(Ensemble learning)的思想將多棵樹(決策樹 decision tree)集成的一種算法，它的基本單位是決策樹，而它的本質屬於機器學習的一大分支——整體學習（Ensemble Learning）方法。隨機森林的名稱中有兩個關鍵字，一個是「隨機」，一個就是「森林」。用最簡單的方式比喻「森林」，一棵叫做樹，那成千上百顆就可以稱做森林。而「隨機」的含義我會在下半部分解釋。

從直觀角度來解釋，每棵決策樹都是一個分類器（此次實驗針對的是分類問題），那麼對於一個輸入樣本，N 棵樹會有 N 個分類結果。而隨機森林集合了所有的分類投票結果，將投票次數最多的類別指定為最終的輸出，這就是一種最簡單的 Bagging 思想。

隨機森林的產生

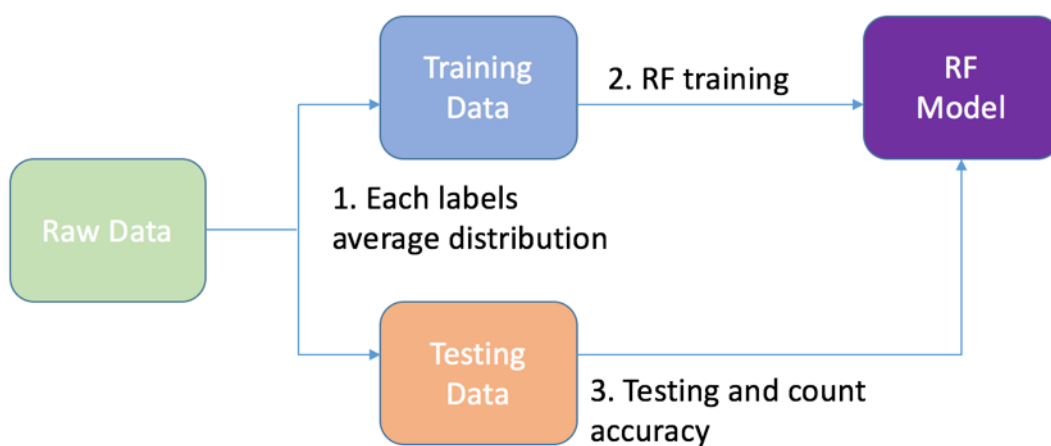
前面提到，隨機森林中有許多的分類樹。要將一個樣本進行分類，需要將樣本輸入到每棵樹中進行分類。用個方式比喻：森林中召開會議，討論某個動物到底是兔子還是松鼠，每棵樹都要獨立發表自己對這個問題的看法，也就是每棵樹都要投票。該動物到底是兔子還是松鼠，再依據投票情況來確定，獲得票數最多的類別就是森林的分類結果。森林中的每棵樹都是獨立的，99.9%不相關的樹做出的預測結果涵蓋所有的情況，這些預測結果將會彼此抵消。少數優秀的樹預測的結果將會超脫於芸芸眾生，做出一個好的預測。將若干個弱分類器的分類結果進行投票選擇，從而組成一個強分類器，這就是隨機森林 bagging 的思想。

下圖為隨機森林比喻示意圖：



實驗流程

實驗流程如下圖所示：



第一步會先將原始資料均分成兩份分別為：Training Data & Testing Data (Training Data：訓練模型用、Testing Data：用來測試模型的準確率)，第二步利用 Training Data 以及使用 Random Forest 演

算法建立模型，第三步再利用 Testing Data 去計算模型的準確率(準確率公式：預測成功數量/整體資料數量)，由準確率作為此模型的預測成功率。

Python 與機器學習實作-Abalone 資料集

此部分將講解如何使用 Python 做最基本的機器學習，從資料的標記到最後算出準確率皆會一一的敘述。在資料的標記部分，由於是要 Abalone 的表特徵預測內部肉的大小，故特徵值 Shucked Weight 將被用來當作標記，此階段會將數值型特徵值 Shucked Weight 依照大小來分為三類(大、中、小)，特徵值 Shucked Weight 從 0.001~1.488，計算後得到平均約 0.36，故將 0.36 作為區分的分水嶺，共分為三類(小<0.36、0.36<中<0.72、0.72<大)，並將小、中、大分別標記為 1、2、3(程式碼無法計算文字類型的值，故須將小、中、大依照數字表示)，處理完成之資料集如下圖所示：

Length	Diameter	Whole_weight	label
0.67	0.55	1.566	3
0.67	0.54	1.619	3
0.675	0.525	1.2835	2
0.675	0.51	1.382	2
0.68	0.52	1.4535	2
0.68	0.51	1.6075	2
0.685	0.52	1.3735	2
0.685	0.565	1.638	3
0.69	0.55	1.569	2
0.7	0.565	1.8565	3
0.7	0.535	1.773	2
0.705	0.545	1.58	2
0.71	0.575	2.009	3
0.71	0.57	1.9805	3
0.71	0.54	1.5805	3
0.71	0.56	2.015	3
0.72	0.57	1.8275	3
0.72	0.55	2.125	3
0.72	0.525	1.445	2
0.725	0.565	2.1425	3
0.73	0.56	1.9425	3
0.735	0.59	1.747	3
0.74	0.565	2.119	3
0.75	0.565	1.938	3
0.75	0.595	2.2205	3
0.77	0.62	2.5155	3
0.775	0.63	2.7795	3
0.275	0.175	0.2315	1
0.375	0.245	0.394	1
0.375	0.27	0.597	1

前三排分別為特徵值 Length、Diameter、Whole Weight 的數值，最後一排為處理過後的標記(Shucked Weight)的數值。

在此實驗中將使用到 Python 中常被用來作為資料分析用的套件 pandas，故須先將它載入到編譯器內，並將處理完成的資料集檔案(csv)載入到編譯器內，程式碼如下圖所示：

```
import pandas as pd
data = pd.read_csv("abalone_process.csv")
```

這段程式碼先將套件 pandas 載入，並將其用“pd”表示，並利用 pandas 中的語法“read_csv”將先前處理好的資料集載入，載入後的輸出結果如下圖所示：

	Length	Diameter	Whole_weight	label
0	0.455	0.365	0.5140	1
1	0.350	0.265	0.2255	1
2	0.530	0.420	0.6770	1
3	0.440	0.365	0.5160	1
4	0.330	0.255	0.2050	1
5	0.425	0.300	0.3515	1
6	0.530	0.415	0.7775	1
7	0.545	0.425	0.7680	1
8	0.475	0.370	0.5095	1
9	0.550	0.440	0.8945	1
10	0.525	0.380	0.6065	1
11	0.430	0.350	0.4060	1
12	0.490	0.380	0.5415	1
13	0.535	0.405	0.6845	1
14	0.470	0.355	0.4755	1
15	0.500	0.400	0.6645	1
16	0.355	0.280	0.2905	1
17	0.440	0.340	0.4510	1
18	0.365	0.295	0.2555	1
19	0.450	0.320	0.3810	1
20	0.355	0.280	0.2455	1
21	0.380	0.275	0.2255	1
22	0.565	0.440	0.9395	2
23	0.550	0.415	0.7635	1
24	0.615	0.480	1.1615	2
25	0.560	0.440	0.9285	2
26	0.580	0.450	0.9955	2
27	0.590	0.445	0.9310	1
28	0.605	0.475	0.9365	2
29	0.575	0.425	0.8635	2
...	...	...	...	...
4147	0.695	0.550	1.6645	3
4148	0.770	0.605	2.0505	3
4149	0.280	0.215	0.1240	1
4150	0.330	0.230	0.1400	1

接下來將標記所挑出來，並用“label”所表示，另外需將載入資料集中的標記移除，此步驟的目的是因為在機器學習訓練模型的階段不能夠將標記一起跟特徵值進行訓練，因為這樣會讓訓練出來的模

型被標記所影響，故此步驟須將資料集中的標記所移除，其程式碼如下圖所示：

```
label = data["label"]
data.pop("label")
```

將移除標記後的資料集輸出可以發現標記欄位(label)已經被移除了，其結果如下圖所示：

	Length	Diameter	Whole_weight
0	0.455	0.365	0.5140
1	0.350	0.265	0.2255
2	0.530	0.420	0.6770
3	0.440	0.365	0.5160
4	0.330	0.255	0.2050
5	0.425	0.300	0.3515
6	0.530	0.415	0.7775
7	0.545	0.425	0.7680
8	0.475	0.370	0.5095
9	0.550	0.440	0.8945
10	0.525	0.380	0.6065
11	0.430	0.350	0.4060
12	0.490	0.380	0.5415
13	0.535	0.405	0.6845
14	0.470	0.355	0.4755
15	0.500	0.400	0.6645
16	0.355	0.280	0.2905
17	0.440	0.340	0.4510
18	0.365	0.295	0.2555
19	0.450	0.320	0.3810
20	0.355	0.280	0.2455
21	0.380	0.275	0.2255
22	0.565	0.440	0.9395
23	0.550	0.415	0.7635
24	0.615	0.480	1.1615
25	0.560	0.440	0.9285
26	0.580	0.450	0.9955
27	0.590	0.445	0.9310
28	0.605	0.475	0.9365
29	0.575	0.425	0.8635
...	...	...	...
4147	0.695	0.550	1.6645
4148	0.770	0.605	2.0505
4149	0.280	0.215	0.1240
4150	0.330	0.230	0.1400

接下來的步驟由實驗流程圖可知，須將資料均等的切分為兩部分，分別是訓練資料集(training data)及測試資料集(testing data)，在此部分將利用機器學習套件 sklearn 中的 model_selection 語法 train_test_split，此語法會將資料依照使用者需求切分成使用者訂定的訓練資料集及測試資料集的比例，在此語法中需要輸入三個參數，第一個為特徵值，第二個為標記，第三個為測試資料的比例(test_size)，由於在實驗中訓練資料集和測試資料集的比例為 1:1，故

在此部分將測試資料的比例參數設為 0.5，此段程式碼如下圖所示：

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data, label, test_size=0.5)
```

由圖中可知最後可得四個輸出結果，分別為測試資料集的特徵值(X_train)、測試資料集的特徵值(X_test)、訓練資料集的標記(y_train)、測試資料集的標記(y_test)。

接下來將訓練資料集利用隨機森林演算法進行訓練並輸出訓練後的模型，在此部分將利用 sklearn 中的 ensemble 語法 RandomForestClassifier 進行訓練，由下圖可知此步驟需將訓練資料集的特徵值(X_train)及訓練資料集的標記(y_train)輸入。

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
```

最後將測試資料放入訓練完成後的模型進行預測，並利用預測後的結果跟原先測試資料集的標記進行比較，計算準確率，在此部分使用 sklearn 套現中的 metrics 語法 accuracy_score 計算準確率，此語法需要兩個輸入值，分別為測試資料集的標記(y_test)及利用測試資料及預測後的標記結果(y_predict)，最後將其輸出即為準確率，此段程式碼如下圖所示：

```
from sklearn.metrics import accuracy_score
y_predict = clf.predict(X_test)
print accuracy_score(y_test, y_predict)
```

經由計算準確率後能夠得知此模型訓練後的結果是否優良，若是訓練後的結果不夠優良，需考慮是否在特徵值選取時需要選取更能夠代表此資料的特徵值或者是資料的標記間需要更明確，而此次分析結果準確率約為 0.9(如下圖所示)，此數值代表著若未來 Abalone 農利用此三個特徵值對新的 Abalone 進行預測，在 10 次預測中有 9 次是正確的，是一個還不錯的預測模型。

```
accuracy = 0.909526089038
```

結論

在這次的 eNews 中簡易的介紹了 Python 以及機器學習，利用公開資料集中所提供的特徵值，進行了特徵值的選取以及資料的標記，並設計了簡易的實驗，最後完成了基本的監督式學習。在現實生活中有很多資料能夠利用機器學習進行分析及預測，而若需分析更深層的资料並非此次 eNews 如此簡易，需考慮到更多條件。機器學習和統計分析間有相同及不同的部分，若能將兩者靈活運用且互相比較，能夠對資料有更多元的解讀。

參考文獻

1. “機器學習：聊聊機器學習模型集成學習算法——Bagging”.
<https://kknews.cc/zh-tw/other/axjaq46.html>
2. Wikipedia.” 隨機森林”. <https://zh.wikipedia.org/wiki/隨機森林>
3. Wikipedia.” 機器學習”. <https://zh.wikipedia.org/wiki/機器學習>
4. [Machine Learning] Random Forest 隨機森林.
<https://read01.com/mEjD8O.html>
5. UCI ”Abalone Data Set ”.
<http://archive.ics.uci.edu/ml/datasets/Abalone>