請測試以cascade架構下的Adaboost、KNN二種分類器進行訓練，以各種

不同的參數，以及不同的訓練集、測試集比例，觀察調整以上變數的情

況下，對於訓練出來的準確率有什麼變化。

# Adaboost

主要是透過 python sklearn.ensemble 函式使用，裡面其中最重要的三個參數如下：

* **n\_estimators**

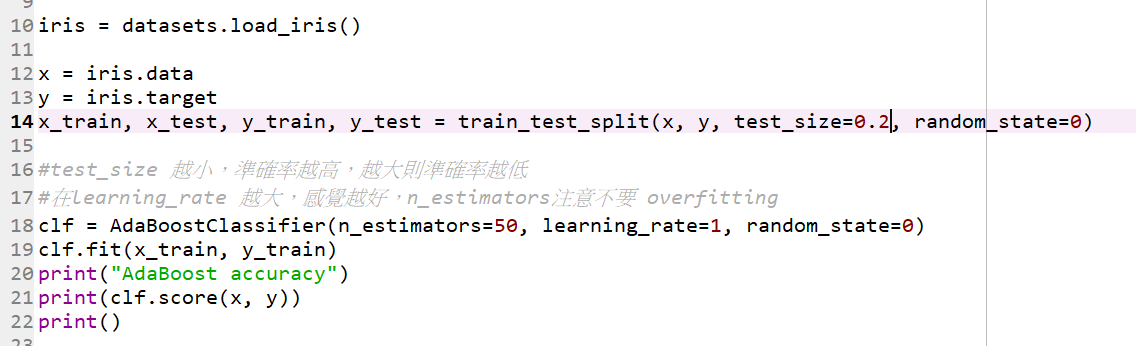
**為公式****的可最大疊代次數，其中 就是每次的疊代次數記錄，通常標準是 50，如果太大可能會導致 overfitting，太小則又不夠精確。**

* **learning\_rate**

為每個分類器的權重，如果每個分類器的權重越大，那建議 **n\_estimators 可以小點，如果每個分類器的權重越小，則建議n\_estimators 要更大些。**

透過鳶尾花資料集進行練習，我發現最主要的問題是 train\_test\_split 中的 test\_size 比例越小時則其準確性越高，反之則越大。

## 程式碼



## 輸出結果

其中最酷的是 n\_estimators在 40 的情況時，準確率會降低，但在 n\_estimators 是 30 的情況下時，準確率卻又會上升至 n\_estimators 等於 50。

學習連結如下

[sklearn.ensemble.AdaBoostClassifier()函数解析（最清晰的解释）by 我是管小亮](https://blog.csdn.net/TeFuirnever/article/details/100276569?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-4&spm=1001.2101.3001.4242)

[sklearn.ensemble.AdaBoostClassifier by scikit learn](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html)

[[Day26]機器學習：KNN分類演算法！by iT邦幫忙](https://ithelp.ithome.com.tw/articles/10197110)

# KNN

主要是透過 python sklearn.neighbors函式使用，裡面其中最重要的三個參數如下：

* n\_neighbors

預設是5，是 KKN 演算法中已離你最近的 K 的點為主，將自己也視為此類別。

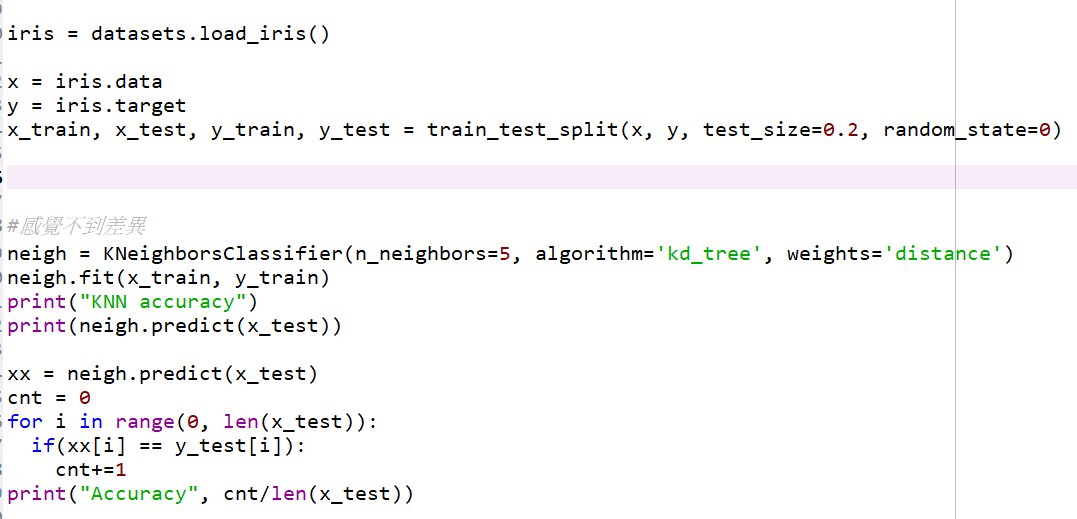
* weights

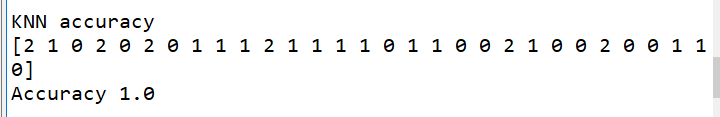
以要被判斷的點為中心，有兩種方法 uniform 為只要在此範圍內的點權重都相同，而 distance 每個點權重都不同，取決於每個點到中心點的距離。

* algorithm

使用的演算法，預設是 auto，也就是函式自動使用適合的演算法，brute 是暴力搜尋，當訓練集大時非常耗時，而 kd\_tree and ball\_tree 則是用來改善 brute，而產生的演算法。

**程式碼**



**結果**

其中由於 KNN 會出來的是對於每個點的預測，於是我有在第 30行到第35行簡單的對每筆資料進行判斷。

其中得出我們的 weights 如果使用 distance 的準確率會比 uniform 來的更好。

而 n\_neighbors 在這邊我們則不會感受到太大的差異，主要是因為資料量太小且 5 是標準函式預設的 n\_neighbors，如果再小可能就會導致 overfitting。

注意：n\_neighbors 不可以設定為 1，此函式出來的結果畢竟會是100%準確，沒辦法發揮此函式的功用。

學習連結

[sklearn.neighbors.KNeighborsClassifier()函数解析（最清晰的解释）by 我是管小亮](https://tefuirnever.blog.csdn.net/article/details/99818078?utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&dist_request_id=1328769.82134.16177873596971781&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control)

[sklearn.neighbors.KNeighborsClassifier by scikit learn](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html)

[[Day26]機器學習：KNN分類演算法！by iT邦幫忙](https://ithelp.ithome.com.tw/articles/10197110)