資料科學導論 exetcise2 大氣4A 黃展皇 106601015

[原檔案，採用datasets.load\_iris()資料情況]

程式碼如附件code\_106601015.py

Requestment：sklearn、numpy、math

1. The accuracy on both training and test data

使用gini index作為分類依據：

gini tree train accuracy: 0.952381

gini tree test accuracy: 0.977778

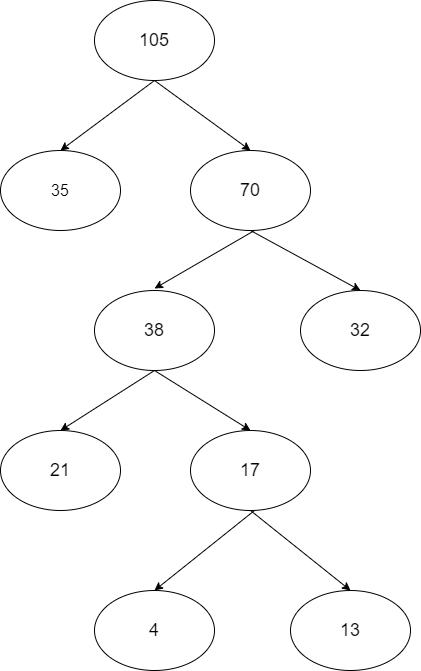
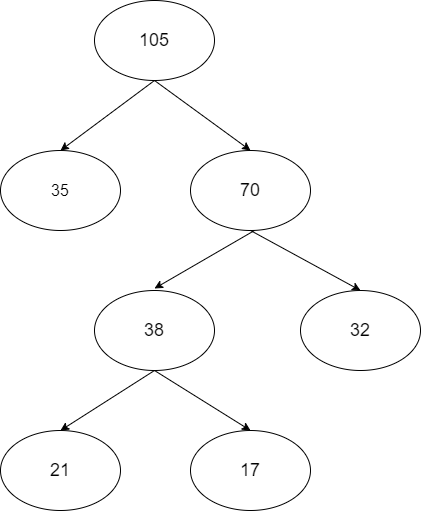
使用information entropy作為分類依據：

entropy tree train accuracy: 0.952381

entropy tree test accuracy: 0.977778

1. The effect of different parameters (Ex: criterion、max\_depth)

針對max\_depth做更改，繼續加深深度並不會影響準確度，但當深度減少到1時便會產生顯著的誤差，原因是因為本分類器在這個case已經完成分類，所以加深深度對於提升準確度無明顯作用，但是若減少深度則可能產生未完全分類的情況，導致準確率下降。至於criterion是gini或是entropy在此案例產生了兩種不同的樹：



左圖為gini tree, 右圖為entropy tree，可以看到在depth=1兩種方式都先將35個0 class分離出來再針對1、2做分類，接著幾乎都做出了相同的分類，最後在17的node中gini停下而entropy則細分為4、13node。

1. A brief discussion of the results

可以看到在演算法無誤的情況下，此二元決策分類樹可以找出多X輸入中對於y label的相對統計關係，並且建立決策樹，讓所有的測試data都能歸納至一個在此關係下的node，得出結論，且運算時間不長，也可以讓人類了解內部演算結構(no black box)。

最後的準確度來到95%以上相當良好，不過在針對零星資料的雜質時無論是哪種樹都無法”清理”得很乾淨，追根究柢仍然是演算法本身greedy，且無法省略零星雜質的汙染(gini、entropy值皆會受到影響)。

[更改後檔案，採用pd.read\_csv資料情況]

程式碼如壓縮檔附件code\_106601015.py

Requestment：sklearn、numpy、math

1. The accuracy on both training and test data

使用gini index作為分類依據：

gini tree train accuracy: 0.638444

gini tree test accuracy: 0.627660

使用information entropy作為分類依據：

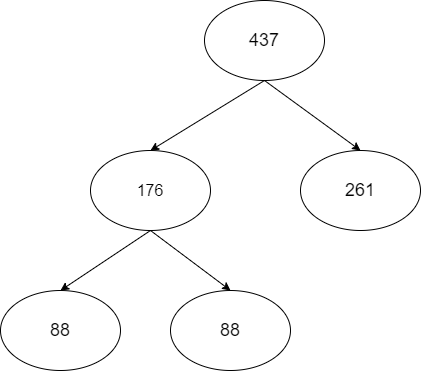
entropy tree train accuracy: 0.638444

entropy tree test accuracy: 0.627660

1. The effect of different parameters (Ex: criterion、max\_depth)

針對max\_depth做更改，max\_depth =0時(不分群時) train accuracy: 0.462243，tree test accuracy: 0.457447；max\_depth =1時(分群一次時) train accuracy: 0.638444，tree test accuracy: 0.627660；max\_depth =1時(分群一次時) train accuracy: 0.638444，tree test accuracy: 0.627660，max\_depth=2以上時無論是train或是test的accuracy都不會再改變，代表著資料已經分類完成，即使容許的樹最大深度可增加仍不會提升accuracy。

至於criterion是gini或是entropy在此案例產生了一樣的樹：



可以看到在depth=1兩種方式都將437個資料在best\_idx and best\_thr: 3 and -0.008013分割成176/261，接著左方的176個資料在best\_idx and best\_thr: 3 and -0.708364分割成88/88，兩種評估資料雜質的技術都做出了相同的決判。

1. A brief discussion of the results

可以看到在演算法無誤的情況下，此二元決策分類樹可以找出多X輸入中對於y label的相對統計關係，並且建立決策樹，讓所有的測試data都能歸納至一個在此關係下的node(目前看起來是都抓label 3)，得出結論，且運算時間不長，也可以讓人類了解內部演算結構(no black box)。

最後的準確度來到60%左右，基本上可以看到跟之前的資料90%以上的準確度有頗大的差距，也就是說二元決策分類樹對於資料本身的要求相對高，在雜亂的資料中雖能找出相應的label與閾值，但是硬分割的結果將導致不夠好的準確度，也導致分類時無法有效取得關鍵labels，也可能殘留greedy的問題。