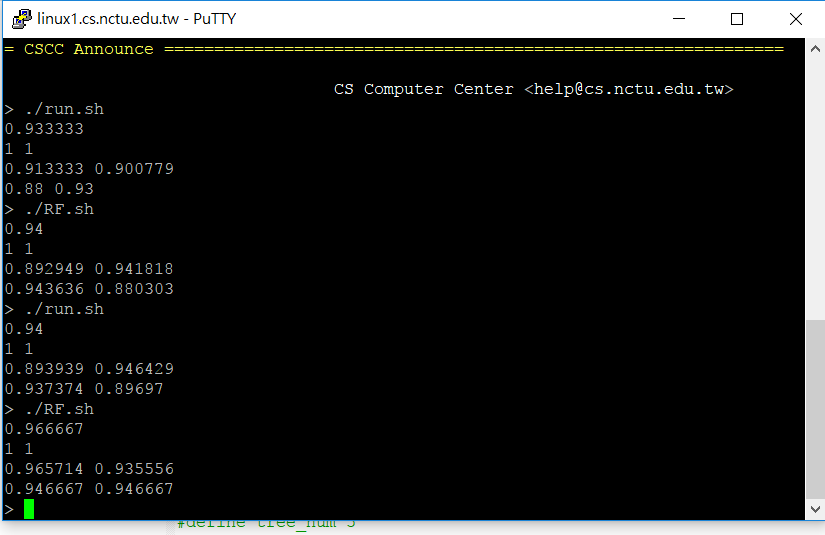
機器學習 作業1報告

0416094 黃兆宇

1. 執行結果:

./run.sh是Decision tree + k – fold

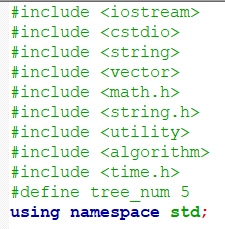
./RF.sh是增加了random forest (Decision tree + random forest + k – fold)



Random forest 的執行結果稍高 accuracy 0.94 ~ 0.96，但效果並不明顯

未使用的情況下約0.92 ~ 0.94。

1. 執行環境: 系工作站linux1站台
2. 程式語言: C++
3. 使用library:



String.h只要用來處理字串strcmp

Algorithm中使用random shuffle的函式

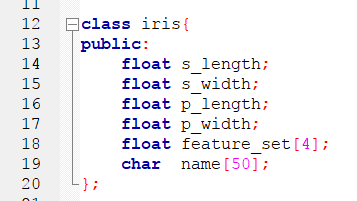
1. 執行方法 :

1.將程式main.cpp iris.data run.sh放至同一路徑執行./run.sh

2.將程式main\_rf.cpp iris.data RF.sh放至同一路徑執行./RF.sh(random forest)

6. 程式說明 :

1.設定資料結構 iris



先宣告儲存iris花朵資料的class 名為iris。

sepal length in cm = s\_length

sepal width in cm = s\_width

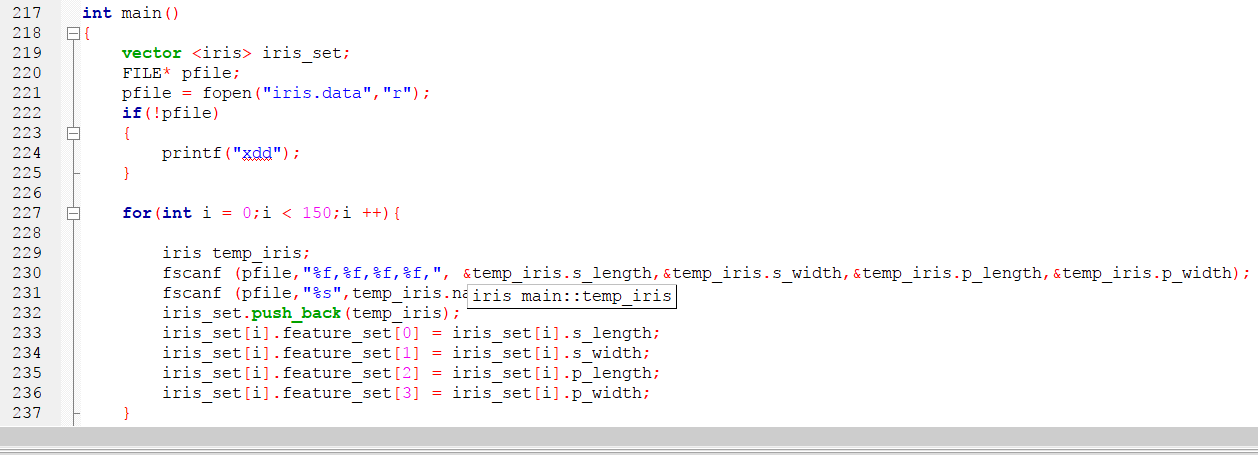
petal length in cm = p\_length

petal width in cm = p\_width

char name存花朵名

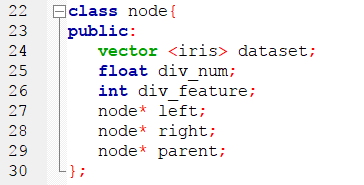
而這些feature同時以陣列的方式儲存，方便以後使用(continuous data)

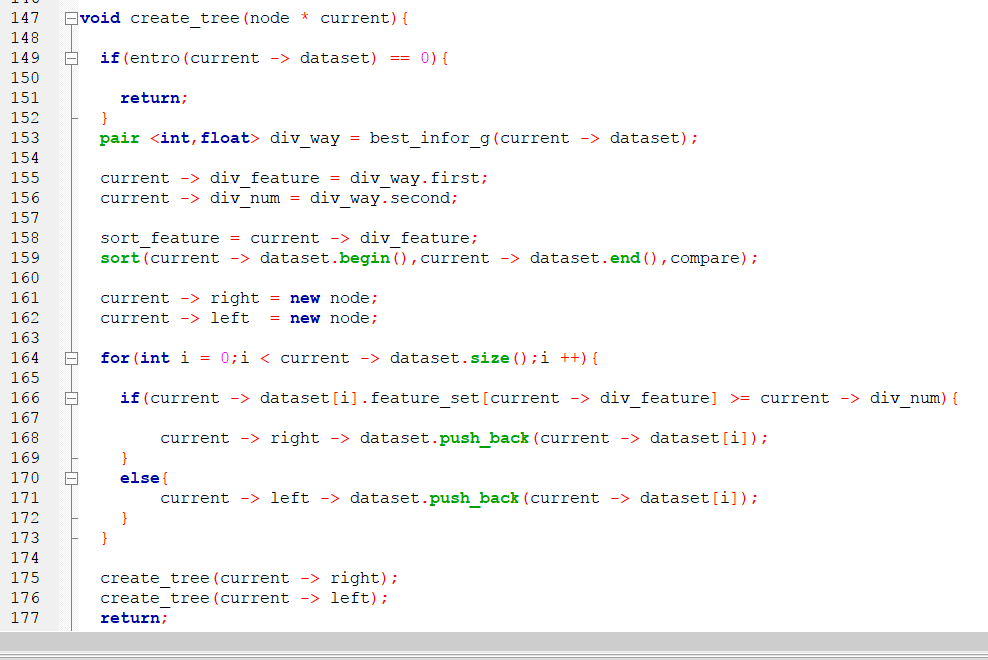
2.讀檔



之後讀取data中各花朵的資料至一存iris的vector，iris\_set之中，使用fscanf讀取。讀取到的資料亦存入feature\_set中。

3.建樹



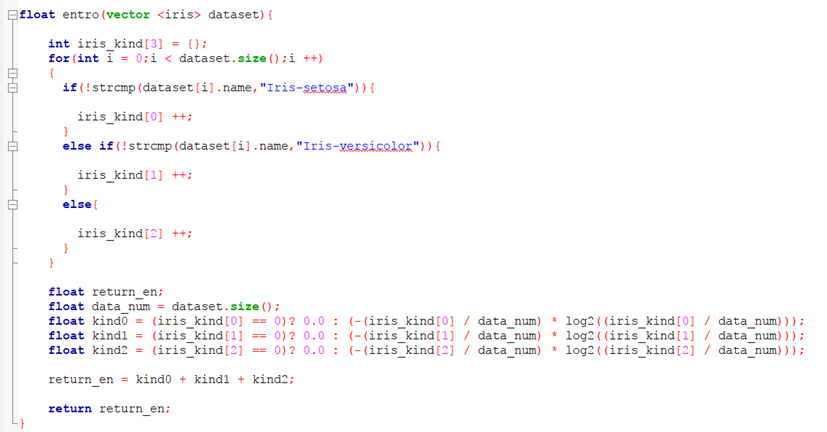


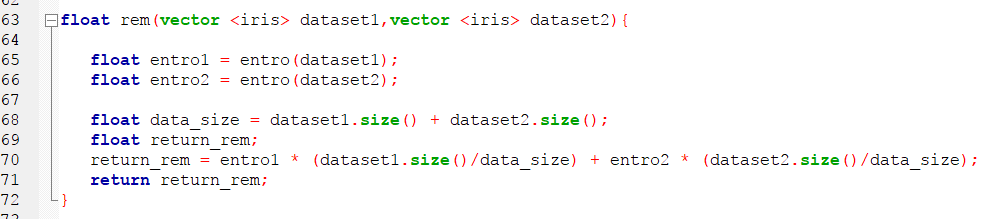
Decision tree是由一名為node的class組成

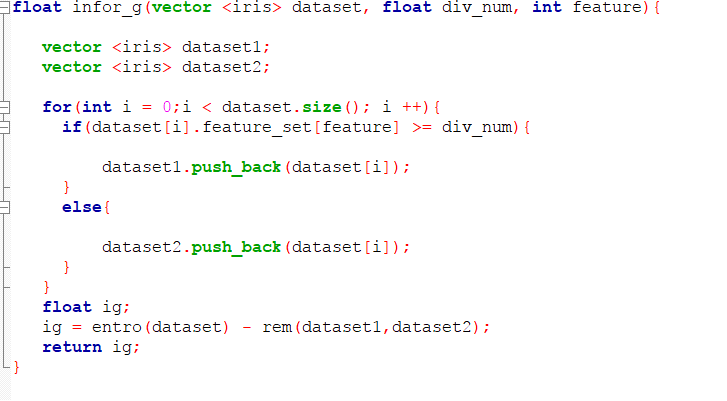
Node主要有三種功能

1. 記錄目前node中有哪一些iris(dataset)
2. 記錄此node的decision方式，如何被分開(div\_num被分開feature的數值，div\_feature 以哪一個feature為分開標準)
3. 指向Left right child的node的指標，確定下一個走向的節點(node \* left node \* right)

建樹的方法主要是依據best\_infor\_g回傳的兩個值(div\_num、div\_feature)做分類，將div\_feature小於div\_num的iris放入左節點，其他放入右節點中，再以遞迴的方式右走至entropy為0(entro是計算entropy的function)，return 再向左走，直到完成建樹。

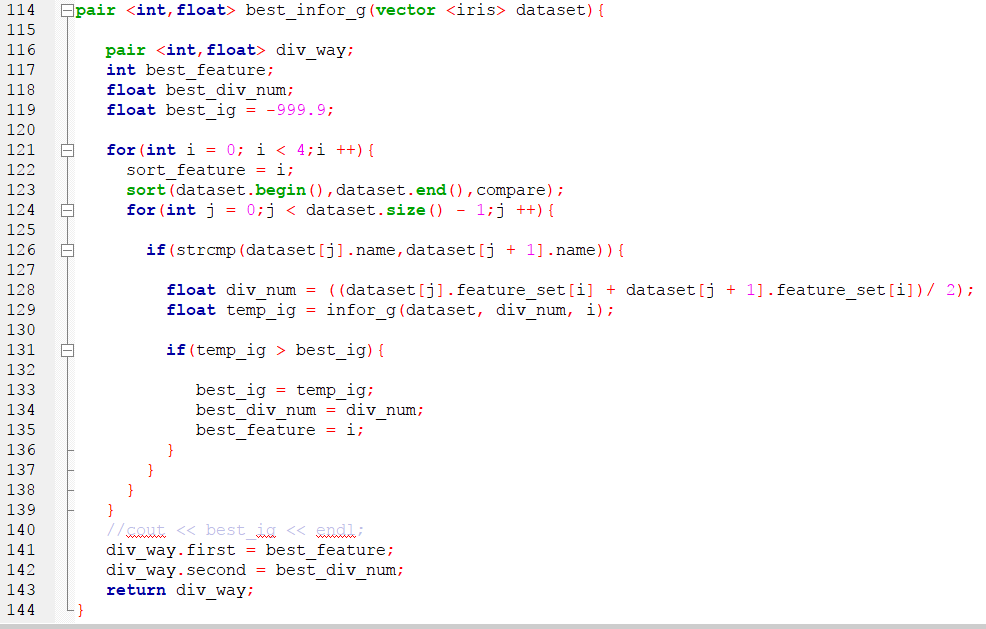






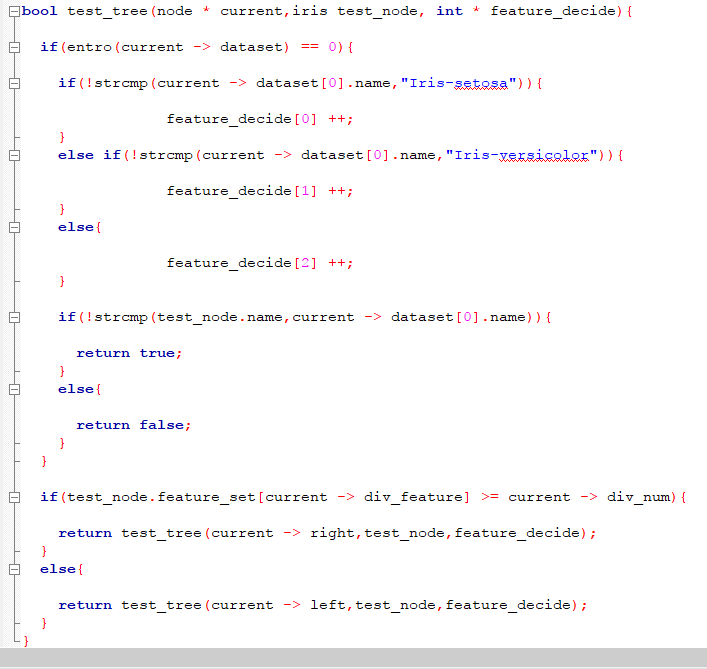
上三圖分別為計算entropy(entro)、remain(rem)、(infor\_g)information gain

的funtion

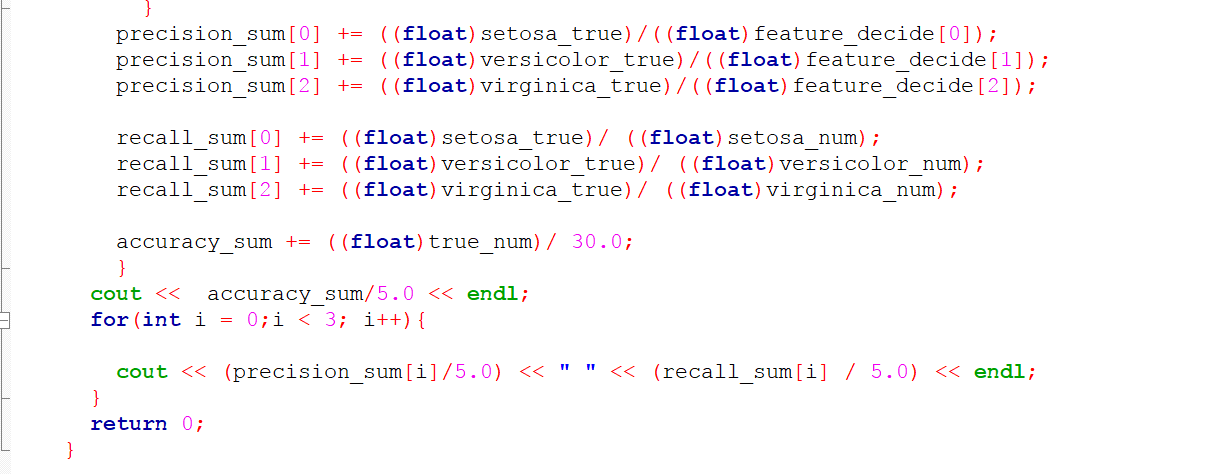


Best\_infor\_g function負責處理連續的資料，算出最佳information gain的區間和feature後，回傳給建樹函式。

1. 測試tree

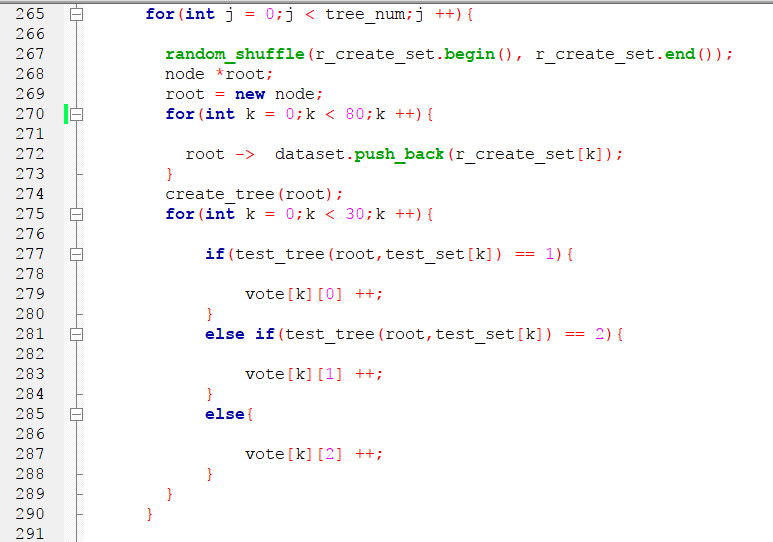


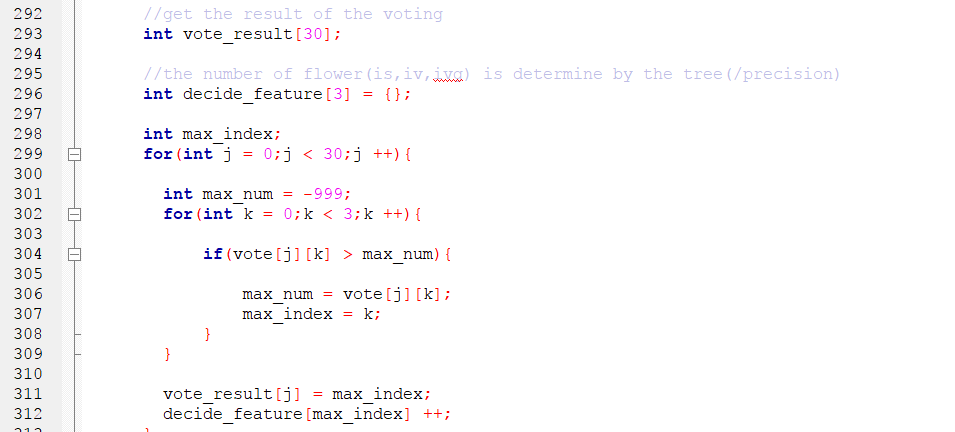
測試decision tree的方法是將先將原本的data set iris\_set以random shuffle的方式打亂順序，index取mod5，為0、1、2、3、4的四筆資料(各30筆)為test data，放入test\_tree中得到判斷結果，剩下的120筆則用來建樹，最後再將五棵樹結果取平均，即為k – fold。

最後印出結果(precision、recall、accuracy)。

以上是不包含random forest的部分

1. random forest





Random forest的做法是，將150筆資料中的120比取出，將此120筆的前80筆取出建樹，再將120筆資料random shuffle，從復建立五棵樹，再以投票的方式決定剩下30筆資料的feature。投票的結果是存在vote[30][3]這個二為陣列中，並依據投票的結果判斷五棵樹的accuracy，最後用與前面相同的方法k-fold取平均數得到最後結果。