**Project #2 Report**

名字：王康霖

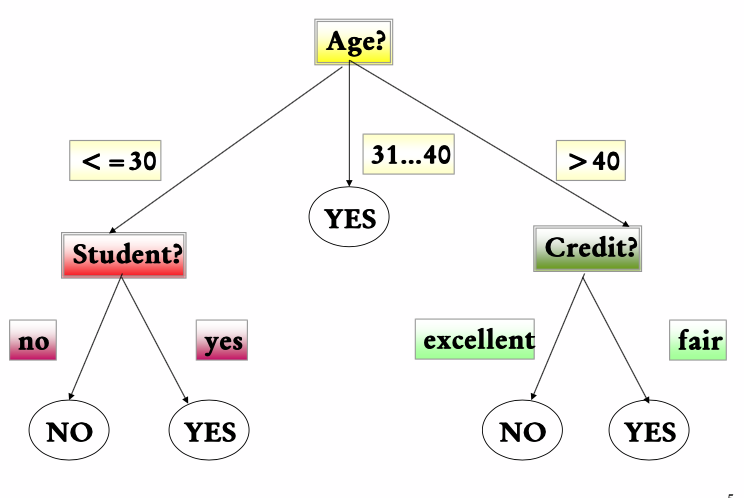
學號：N26077124

執行指令：python3 predict.py

**Dataset**

問題設計：客戶是否會買電腦

* My Decision Tree Rules



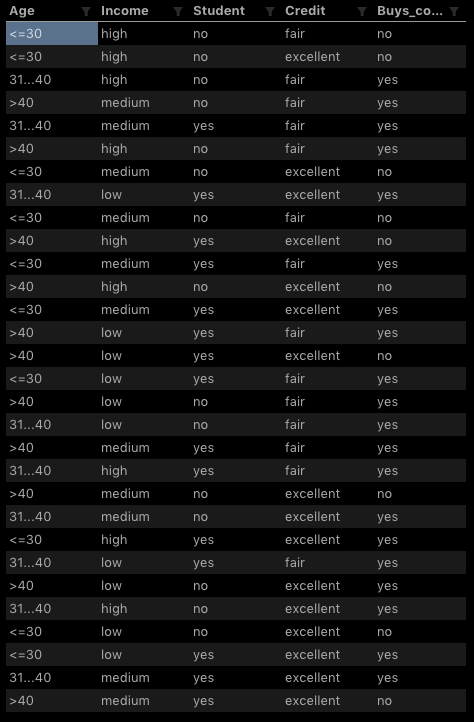
這是一個結果為“會/不會買電腦”的Decision Tree

依據上面的 Decision Tree 產生下列有30筆Data的Dataset

因此在這個問題上

M = 30, K = 4

* Dataset



**Data Preprocessing**

因為Dataset裡面的資料是以文字形式在儲存，但這樣無法拿來做使用。所以我們需要先用label encode的方式將同組轉換成數字。例如把 `Age` 欄位中的 ‘ <=30 ’, ’ 31…40 ’, ‘ >40 ’ 分別轉換成 1，2，3。我是使用 sklearn.preprocessing 所提供的 LabelEncoder。

再來就是要將 Feature 和 Answer 分開，Feature 就是前四個欄位：`Age`, `Income`, `Student`, `Credit`。Answer就是最後的欄位 `Buys\_computer`。

最後再將dataset split 成 train\_data 和 test\_data，使用的是sklearn.model\_selection 所提供的train\_test\_split功能。

參數如下：

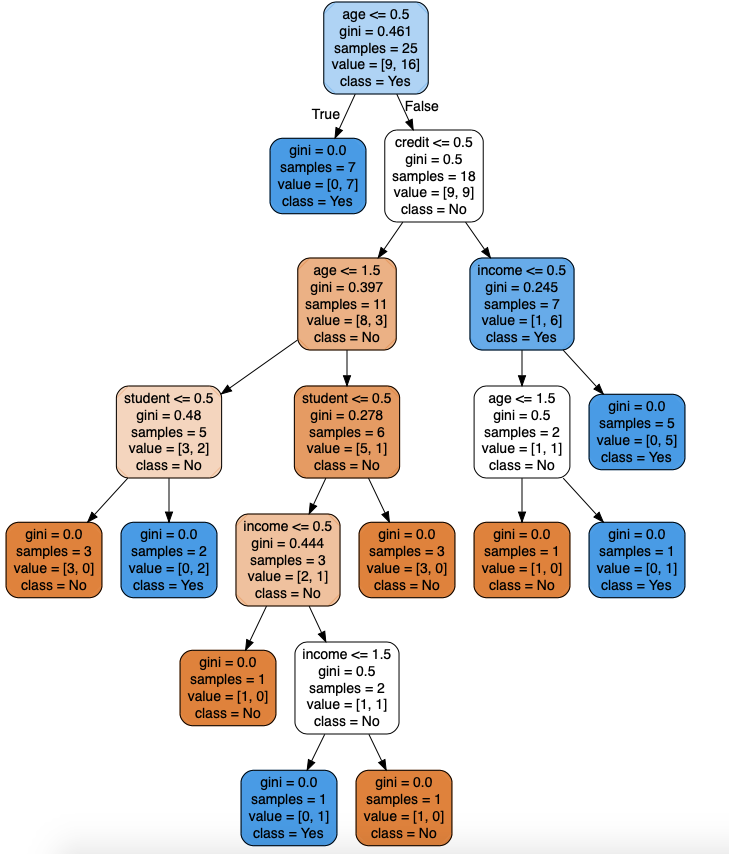
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.15, random\_state=42)

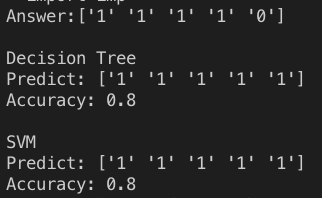
random\_state 會設42而不是亂數是為了讓data可以一樣

**Decision Tree**

* 結果

使用的是 sklearn 所提供的套件tree，數的生成是將規則輸出到 `dot\_tree.txt` 再用線上工具來繪畫（Webgraphiz, <http://webgraphviz.com>）



這個 Decision Tree 的準確率為 0.8

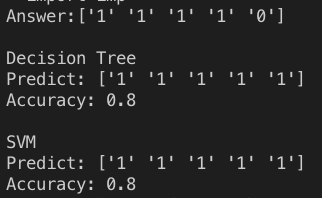
* 討論

其實在random\_state為其他數字的時候，也就是分割檔案的方式不一樣的時候所得到的準確率都有上下，所以這個分割參數還不是最佳的參數，但為了顯示差異所以選擇這個random\_state。

另外，由於dataset的M只有30，資料其實是非常不足的，而且由於dataset中沒有重複的資料，所以在split\_data的時候有可能會發生model從來沒學過的規則出現在test\_data中，所以給出錯誤的答案其實也是很合理的。

**SVM**

* 結果



可以看到準確率也是0.8，但其實這也是在train\_split\_test的時候選擇的random\_state的關係而已。在其他的分割方法的時候，準確率可能是SVM > Decision\_Tree亦或是反過來。

**總結**

在K和M都相同的情況下，我們可以發現會根據我們的Dataset的train和test的不同而有不同的結果。得出的結論是Decision Tree 結構簡單，使用上也不需要設定參數，但缺點就是“想像力”太差，如果出現的分類是自己沒有看過的，打錯的機率就很高。而其他分類器的方法，例如本次所用的SVM，因為還有很多參數可以做調整，而且資料集只有30筆資料，沒有辦法做出顯著的效果，並且也實在無法去跟Decision Tree來做相提並論和比較。