***Data Mining Project 2***

***Classification***

**資訊所碩一 P76074575 潘昌義**

1. **作業簡述**
2. **資料集製作與評估**
3. **Decision tree**
4. **Random Forest**
5. **討論與延伸**
6. **總結**
7. **作業簡述**

在主題三中，我們為了能區別不同種類item的相異性，使用不同classification的方式嘗試將data正確分類。在這個作業中，我們需要自己製作或找一個資料集，且這個資料集具有一個絕對的分類方式(absolutely right rule)，以方便我們未來檢驗這棵樹的正確性。關於演算法的部分，我們可以使用sklearn裡面的模型，例如decision tree classifier或random forest classifier等方法，最後使用graphviz將結果繪製成圖，並檢驗是否與自己原先假設或資料給定的absolutely right rule相符或類似。

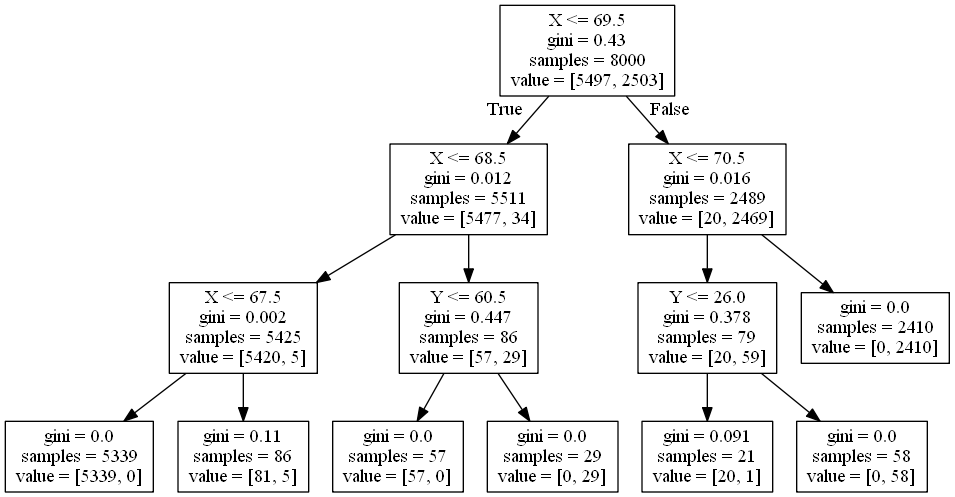
1. **資料集製作與評估**

由於我們需要一個具有absolutely right rule的資料，因此我選擇自己以一個簡單的數學方程式來進行資料的生成，其中加入一點點的小誤差使資料不會完全按照absolutely right rule走，以觀察稍晚的模型能不能不被這點”現實因素”所影響。

為了使最後觀察結果簡化，我選擇採用兩個特徵X與Y，構成一個二元二次方程式，對每一個變數X和Y都取一個介於1~100的隨機值，其中b是一個隨機震盪的數值，介在-3~3之間。由於我們可以由方程式變數的定義域，先行推斷值域位於-1~10403之間，因此我假設分類的觀察結果為”該數是否大於5000”，若大於5000則設為1，反之設為0。

1. **Decision tree**

我們以sklearn.tree裡面的DecisionTreeClassifier將建好的資料進行分類，分類完成後以同樣在sklearn.tree裡面的export\_graphviz將結果輸出成.dot檔，最後再以graphviz將圖片輸出，其結果如下：

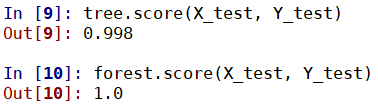


我們不難從上面看出，decision tree的第一層分類馬上幫我們從X=69.5的地方區分為左右兩側，對這個結果不會感到太意外是因為我們對X做了平方的處理，自然他的比重會相對較大，另外 = 70.7，也就非常接近上述分類的69.5了。

至於為什麼會比70.7來得小呢？其實原因很簡單，因為我們在Y的部分配了一個1~100的隨機數，對方程式的整體獎勵是4~400，因此要達到5000這個值會比原先還要來的簡單一些，所以X就比70.7來得小了。

1. **Random Forest**

除了上面用到的decision tree外，在課堂上我們也有提到隨機森林(Random forest)是個良好的分類器，他是由多個decision tree所構成的分類模型，藉由設定tree的數量來提高精確率，但需要付出的代價就是運算所需的時間。我們從sklearn.ensemble內將RandomForestClassifier引進程式後，除了將樹的數量設定為10之外，其餘設定不變跑相同的training data，以相同的testing data測試不同分類器的score，結果如下所示：



雖然沒什麼進展，但還是比較準哪

由於random forest是由多個decision tree所構成的，所以如果要做視覺化的圖就需要個別對每個tree做圖，~~太麻煩了可以不要做嗎。~~

1. **討論與延伸**

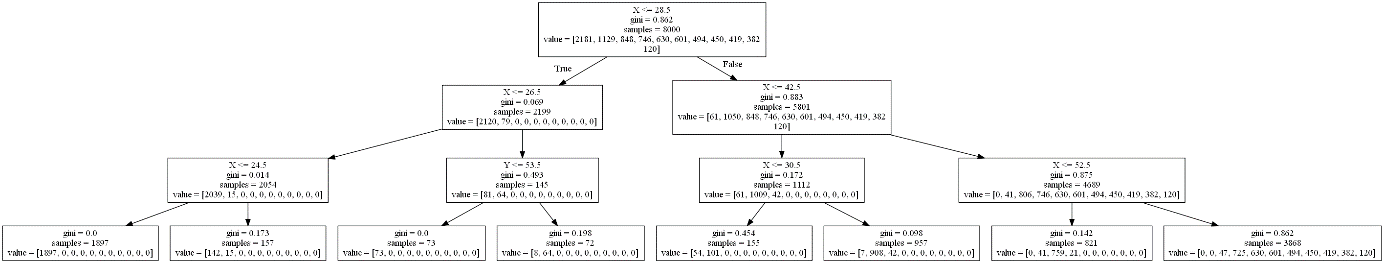
除了完成上述的測試外，這邊分享一下我前面嘗試的奇怪(？)組合：

* 1. 分類”餘數”

不得不說這個真的不太行，因為餘數不能由數字的大小來直接判斷，所以分類器做出來的效果之差無法想像，用decision tree分類的socre只有0.145分，而他的圖呢…~~直接砍掉重練了…~~

* 1. 分類”商”

前面都試過餘數了，當然會想試商數阿，然而這個分類結果卻是非常的壯觀，不是說他不準，而是圖的規模莫名其妙的大…

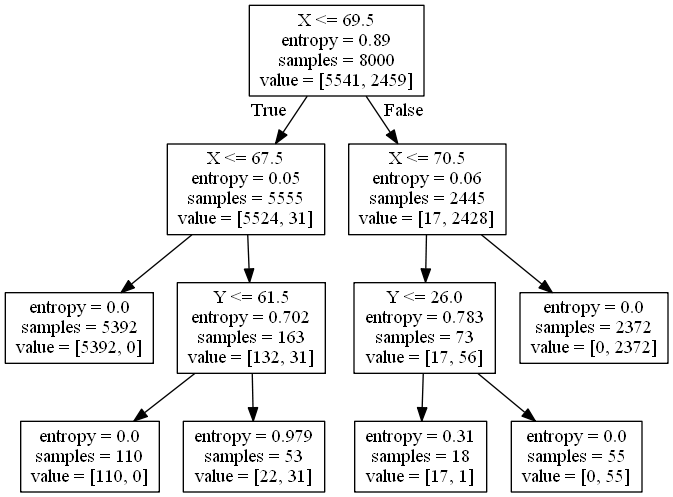


有附檔，~~有興趣可以點開看看~~

我想會出現這種問題大概是因為我的數值介在0~10000之間，而我又把1000作為一個單位，自然出現的結果就大到爆了…

* 1. 嘗試以不同的criterion(分類標準)做

模型的預設基本上都是gini，這邊我選擇改用entropy作criterion，其結果如下：



其實還是能夠清楚的看出在第一層與第二層能夠輕鬆以X的值區分類別，而底下用Y區分導致entropy較高也在我們的預期下，~~畢竟本來我就沒打算要模型用Y去辨識了。~~

1. **總結**

整體來說，我們可以看到absolutely right rule中我們假定影響比較大的feature (例如我擬定的X，他是平方項)，對分類結果的影響較大，在不同的分類標準下皆在第一層就優先被區分了；而比較不重要的feature (例如僅有一次項的Y)則會被放到底層的部份去做進一步的區分，然而它的效果通常不太好，可能是因為深度不夠，也有可能是因為本來就很難拿一個冪次較低的特徵來做良好的分辨。

此外，上述實驗有個小小的失敗，就是我們幾乎沒能看出誤差項b對整體分類是否有構成影響，我想大概是因為我設的數字太小了吧，反而Y比較像是這個實驗的誤差項，觀察一次項能否影響二次項對整體結果的改變。