**Project 2 Report**

工資所 R36074121 辛政達

1. 程式架構

本次程式以python進行撰寫，並且根據創建決策樹所需用到的不同功能分別撰寫凾式，以下針對不同的主要功能function進行基本的說明。

1. 創建資料集:

本次作業我蒐集了兩份不同資料(下節會進行資料集的說明)，由於其資料是由英文字串表示，因此我將資料轉變為數值變量，以利於建模，此外在此部分，我也將資料拆分為80%的訓練資料與20%的測試資料。

1. 計算Entropy:

根據理論與公式，撰寫了一個用以計算Entropy的function。

Entropy的定義為資訊的期望值，用來表示隨機變數不確定性的度量，透過計算資料與各項特徵的Entropy，可以計算各特徵值的資訊增益(information gain)，進而判斷該用資料集的何種特徵值構建決策樹，以進行資料的劃分。

1. 選取對分類結果影響最大的特徵值(計算資訊增益):

根據理論與公式，撰寫了一個用以計算資訊增益的function

要如何選擇劃分的特徵值，需要看資訊增益，若資訊增益越大，則該特徵對於最終分類結果的影響也會越大，因此應該選擇資訊增益最大的特徵值來做為決策樹構建的依據。

1. 創建決策樹:

本次作業，我使用ID3演算法來構建決策樹，其核心為在決策樹各個結點上對應資訊增益準則選擇特徵，遞迴地建構決策樹。具體做法式:從root node開始，對結點計算所有可能特徵值的資訊增益，選擇資訊增益最大的特徵作為結點的特徵，並由該特徵的不同取值建立子結點，再對子結點遞迴地呼叫以上方法，構建決策樹，直到所有特徵值的資訊增益均很小或沒有特徵可以選擇為止，最後得到一個決策樹。

並且使用dictionary儲存決策樹結構，分析出的決策樹表示如下圖



1. 使用決策樹執行分類，並計算準確率:

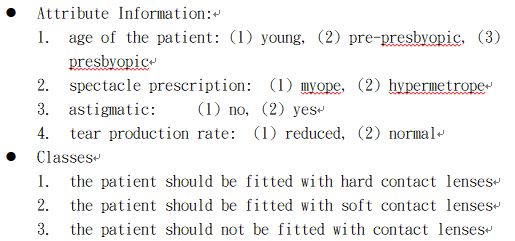
在此部分，我將測試資料集輸入至訓練好的決策樹模型中，並建立了”featattributes”的list以記錄決策樹分類過程中用到的結點特徵值。

測試集輸入後得到的分類結果再與實際分類結果進行比較，計算預測分類的準確率。

.

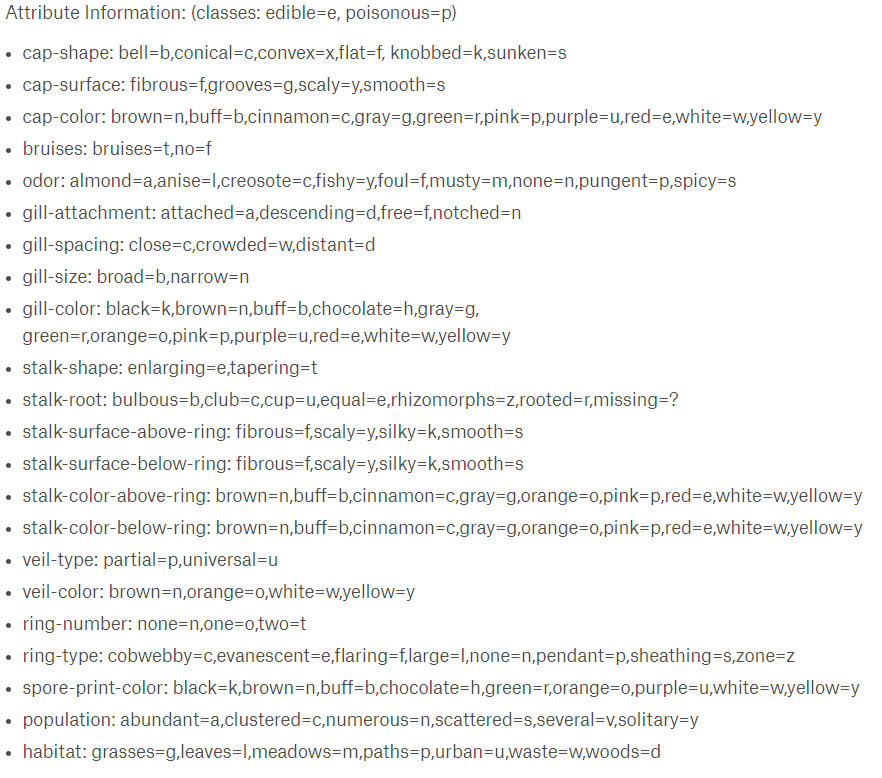
1. 資料集說明

本次作業總共使用了兩份資料，第一份資料為來自UCI資料庫的”隱形眼鏡資料集”(https://archive.ics.uci.edu/ml/datasets/Lenses)，一共有24組樣本資料，4種特徵值，和3種分類屬性，以下為此資料集特徵值和分類屬性的介紹:



此份資料集的資料量小，主要是用來驗證實作決策樹的正確性。

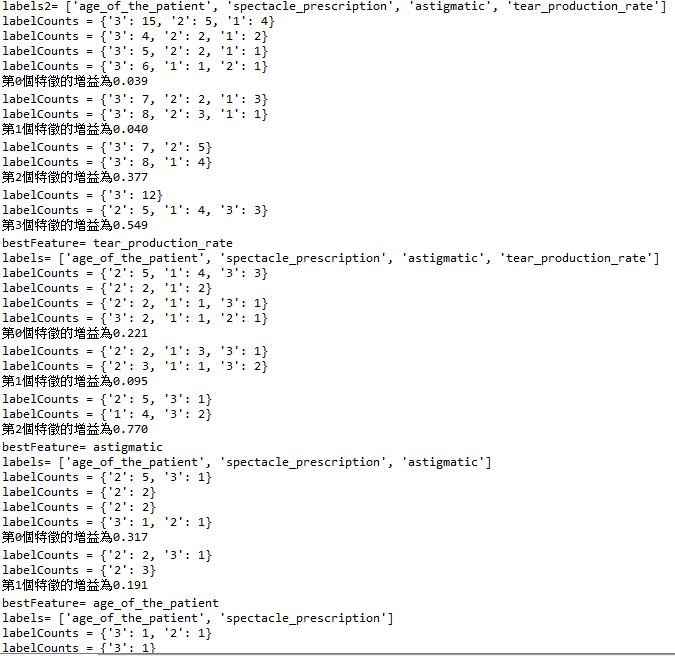
第二份資料為Kaggle中的Mushroom Classification資料集(https://www.kaggle.com/uciml/mushroom-classification/)，此資料集紀錄蘑菇的各種特徵，以及該蘑菇是否可食用，總共有8124筆觀察值，22種特徵值，和2種分類屬性(可食用、不可食用)，下方直接引用網站上的文字作為各項特徵值的說明:



1. 決策樹的實作、驗證與使用
2. 使用隱形眼鏡資料集建立決策樹

為了能夠驗證實作出的決策樹之正確性，因此選擇使用資料量較少的資料集來進行建構。

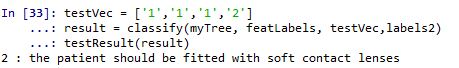
首先確認程式能夠計算Entropy和information gain並正確得執行ID3演算法如下圖所示:



再來確認程式能夠正確地完成決策樹的構建如下圖所示:



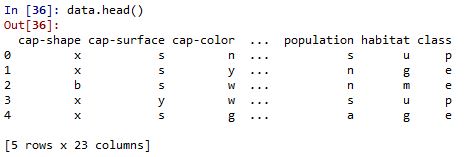
最後輸入一筆測試資料以確認預測分類功能能夠正常執行



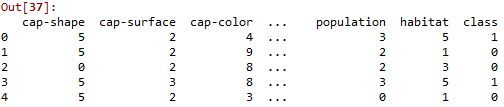
確認了決策樹的構建與分類功能的正確性之後，便能將決策樹套用到Mushroom Classification資料集中。

1. 使用決策樹進行預測

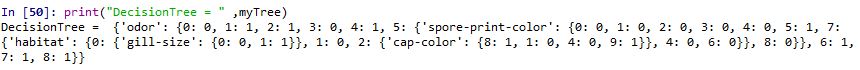
原先的資料集如下所示:



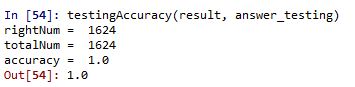
可以看出這是一個分類變量的資料集，為了建模的方便性，因次我使用sklearn套件中的LabelEncoder將資料內容轉為數值變量，如下所示:



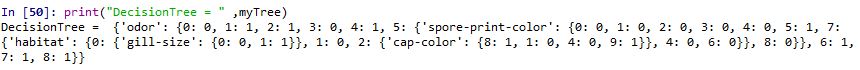
接著便將資料以80%:20%的比例區分為訓練集與測試集，並將訓練集套用到決策樹模型中以完成Mushroom Classification資料集的決策樹，產生的決策樹如下所示:



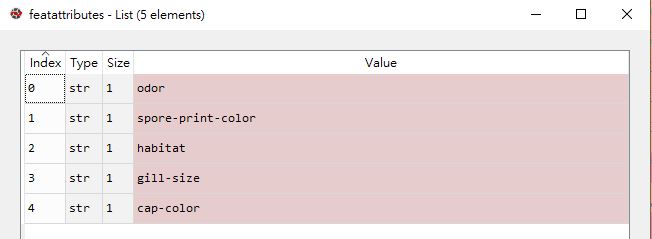
最後再將測試集資料丟入完成訓練的決策樹中，進行分類預測，最後的準確率達到100%:



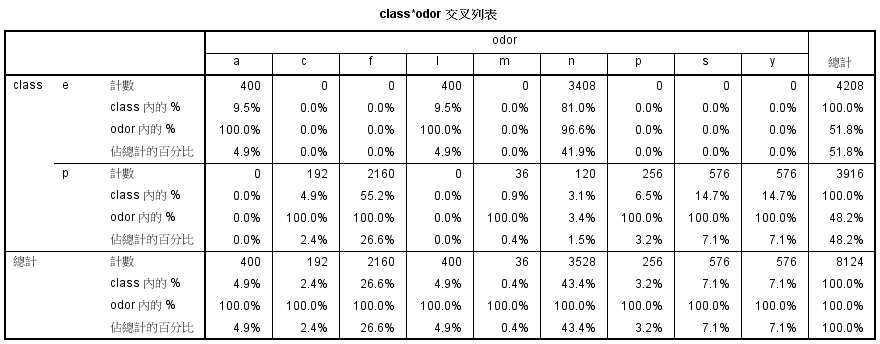
1. 結果分析



仔細觀察生成的決策樹後，可以發現生成決策樹需用到的判斷分類結點只有5個，我建立了一個list來依序儲存這些結點對應的特徵值如下所示:

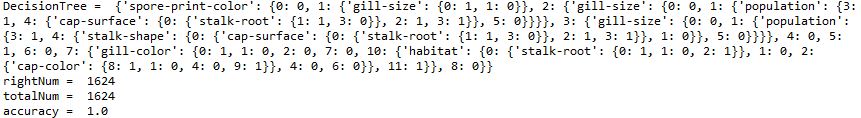


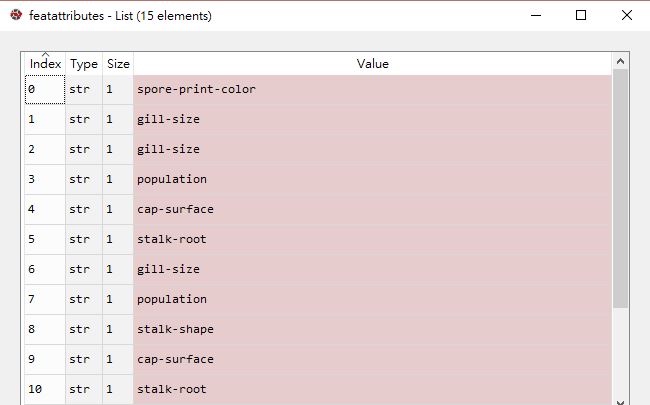
觀察後發現，特徵值”odor”幾乎已經可以完成絕大多數的分類，因此我使用SPSS的交叉表來檢查odor與class之間的關係:



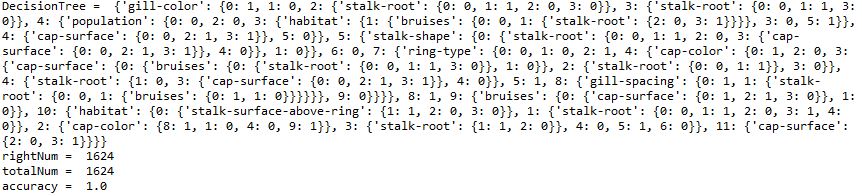
透過表格可以清楚的發現，當odor為a、i時，其所對應的class均為”e”(可食用的蘑菇)，當odor為c、f、m、p、s、y時，其所對應的class均為”p”(有毒的蘑菇)，此外當odor為n時，也有超過9成的可能性是可食用的蘑菇。由於此特徵值的特性，使得決策樹的生成十分簡單。透過這張交叉表，我們看到許多可以供判斷的數值，但因為無法斷定這資料集中資料是否包含全部母體，因此雖然像表格中odor為c全部都是有毒的蘑菇，我們仍然不能直接肯定的說odor為c就是有毒的蘑菇，但可以將此做為一個參考依據。

由於odor作為特徵值的分類判斷能力太卓越，因此我決定將odor特徵值從列表中移除，並在重新生成決策樹，此次決策樹預測準確率仍為100%:





可以發現，在移除了odor後，決策樹的複雜度提高，構建出完整的決策樹需要15個判斷結點，不過也可以發現，儘管複雜度提高，但在全部22種特徵值中，作為判斷結點的特徵值只有其中的9種。於是我再次刪除特徵值(此次刪除spore-print-color以及gill-size)，並重新生成決策樹，此次生成決策樹的分類預測率仍為100%，但決策樹的複雜度變得更高，需要30個判斷結點才能完成分類:



不過仔細觀察可以發現，仍有一些特徵值並沒有被作為判斷結點使用，於是我再次透過SPSS建立了各特徵值的次數分配表以進行觀察，結果發現，有許多特徵值的某一屬性出現率超過90%，舉例而言veil-type(菌幕類別)的資料全部都是p(partial)，而如此高的出現率勢必無法作為判斷該蘑菇是否可食用的依據，因此決策樹不以這些特徵值作為判斷結點也是十分合理的。以下列出某屬性出現率超過90%的特徵值:

|  |  |  |
| --- | --- | --- |
| 屬性名稱 | 資料值 | 出現比率 |
| gillattachment | f | 97.4% |
| veil-type | p | 100% |
| veilcolor | w | 97.5% |
| ringnumber | o | 92.2% |

1. 結語

透過這一次的作業，我對決策樹有了更深一層的認知。決策樹是一種過程直覺單純、執行效率也相當高的supervised機器學習模型，跟其他機器學習模型比起來，執行速度是他的一大優勢。此外，決策樹的每一個決策階段都非常清楚且明確(yes or no)，若通過視覺化，其過程更是能讓人一目瞭然。

在這邊稍微整理一下決策樹進行資料分類的優缺點，我認為決策樹不用懂複雜的理論和技術，就可以理解其運作原理是此分類法最大的優勢，此外決策樹所提取並建立的分類規則，也就是機器學習在建立的規則，可以說是十分直觀。不過決策樹在處理連續型或不相互獨立的資料時，決策樹需要做很多預先處理的工作，且當要分析的類別(class)太多時，錯誤率可能會明顯的升高，這是此分類法的缺點。

這一次的作業，仍存在著很多可以改善跟延伸的地方，若有機會深入此次的作業，我會著重在以下幾點上:

1. 本次作業所選的資料集結構上較簡單，並沒有太多不同的類別需要進行分類(僅分為可食用與不可食用2類別)，因此預測準確率可以輕易的達到100%，如此一來無法看出和其他分類模型的不同之處，因此下次蒐集的資料及應該要擁有更多的分類類別。
2. 這一次我並沒有完成決策樹的可視化，也因此沒有辦法很輕易得判讀決策樹的判斷過程，這也是可以進步的部分。
3. 本次作業並沒有對決策樹做限制，例如:資料數目不得小於多少才能在產生新結點、限制決策樹的深度最多為幾層、要成為最終結點(葉結點)最少需要多少資料等限制，因此可能會產生overfitting的狀況，若資料量更龐大更複雜，可能會導致模型太過於複雜。
4. 由多個決策樹組成的Random Forest可以顯著的提升單個決策樹的效能，也是近年相當熱門的機器學習演算法，未來若有機會希望可以針對Random Forest進行研究。