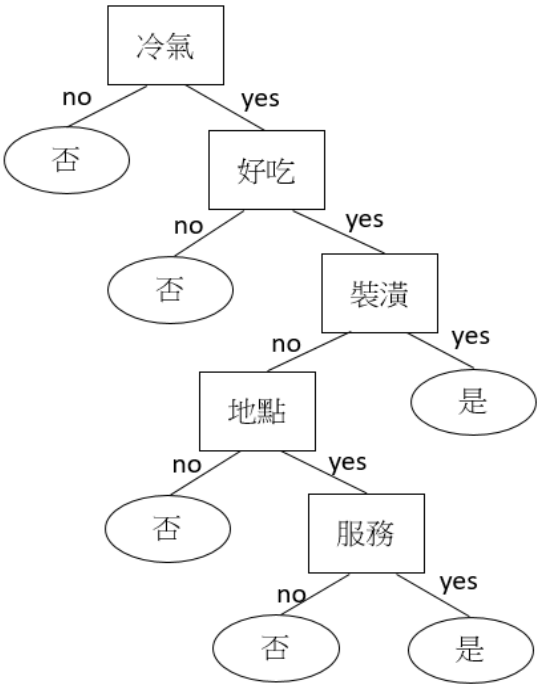
P78071549 陳珞安

Project2：餐廳是否受歡迎?

I. Decision Tree

My _Model:



Data:

1	AC	delicious	decorate	place	service	popular
2	0	1	0	1	1	0
3	0	1	1	0	1	0
4	1	0	1	1	1	0
5	1	1	1	0	0	1
6	1	1	0	0	1	0
7	1	1	0	1	0	0
8	1	1	0	1	1	1
9	1	1	1	1	1	1
10	1	1	1	0	1	1
11	0	0	0	1	1	0
12	0	0	1	1	1	0
13	1	0	1	0	1	0
14	1	1	0	0	1	0

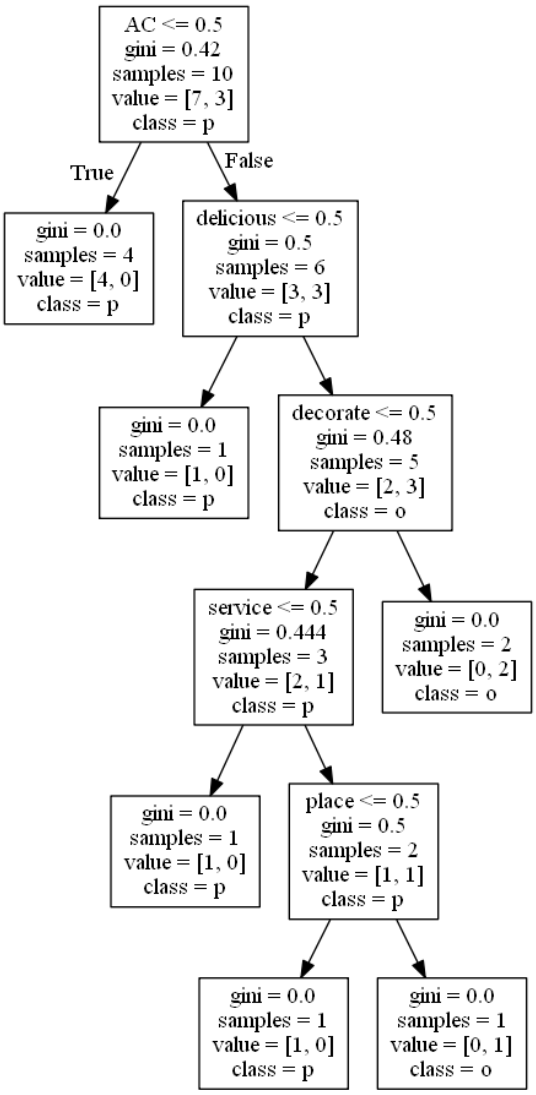
k(# of features) = 5 , M(# of datas) = 13

Result:

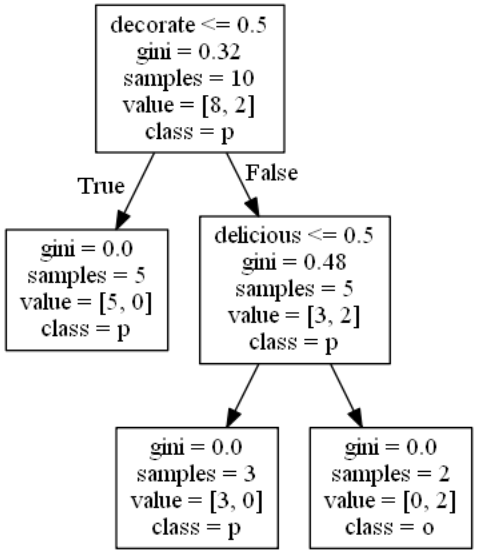
執行方式： python3 decision.py

執行結果:

因為程式是將 80%的 data 作為 training data，剩下的 20%作為 testing data(random)，由於自己產生的 data 量不足，所以只要沒有 train 到某幾條關鍵 data，gini index 算出來就會不一樣，跑出來的結果就會相差很多，圖(一)便是完全預測的，但圖(二)準確率卻只有 0.3333，甚至也有機會跑出準確率為 0 的時候。



圖(一)準確率為 1 所畫出的 decision tree



圖(二)準確率為 0.333 所畫出的 decision tree

因此，便將 data 數(M)提升至 30 筆，這個時候大多的準確率都會落在 0.8~1 之間，我覺得原因是因為這個 data 的 feature 不多，而且都是二元的分類(是 or 否)，所以當正確的例子越多的時候，也代表相同情形的例子出現次數越多，這樣一來便比較不容易出現與預想相差極大的分類方式。

問題

這樣用 random 的方式所產生出的 train_set 以及 test_set 的優點是比較不會因為選擇某筆固定的 data 造成之後實驗的 overfit 或者是 underfit，但缺點是在這邊無法固定 train_set，所以無法與其他分類的演算法做比較。

因此後來將資料做分割的時候，統一固定使用 30 筆的資料，並挑出其中的 8、9、15、17、23、27 作為 test_set，其他則做為 train_set 來做實驗。

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

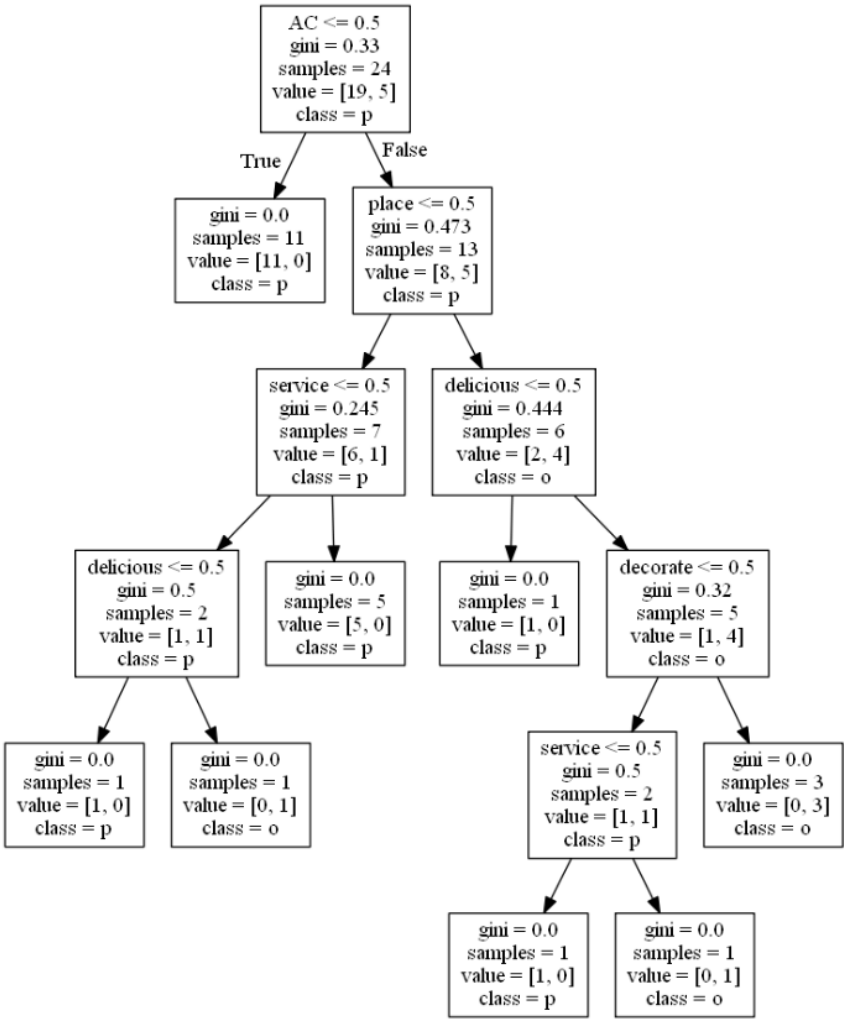
※統一 random_state(亂樹種子)，以達到之後都挑出同樣的 train_set

Result_2:

執行方式： python3 decision.py

執行結果:

```
預測結果：
[0 1 0 0 0 0]
標準答案：
    popular
27        0
15        0
23        1
17        0
8         1
9         0
準確率: 0.5
```



II. KNN

k(# of features) = 5 , M(# of datas) = 30

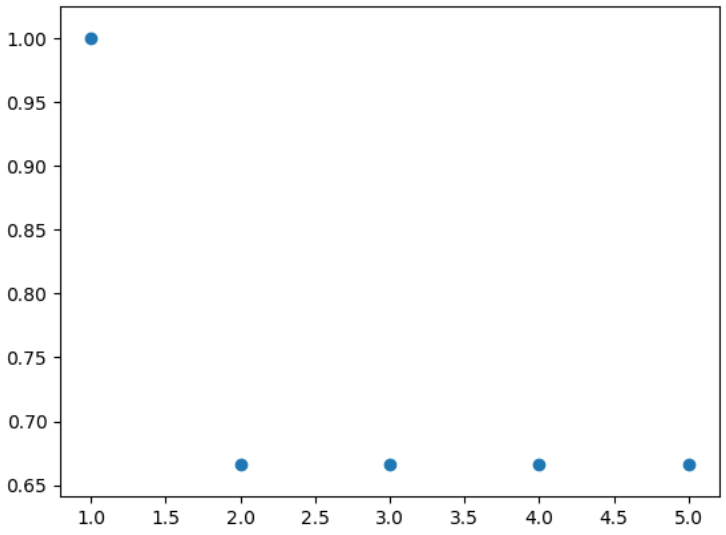
Result:

執行方式: python3 KNN.py

執行結果:

可以從下圖看出，在 k=1 的時候，準確率是 1，但在之後卻都呈現 0.68 左右的準確率，可見在這個情況下 K=1 是準確率最高的。

(準確率)



(K 值)

III. Gaussian Naïve Bayes

k(# of features) = 5 , M(# of datas) = 30

Result:

執行方式: python3 bayes.py

執行結果:

準確率: 0.833333333

IV. Neural Network

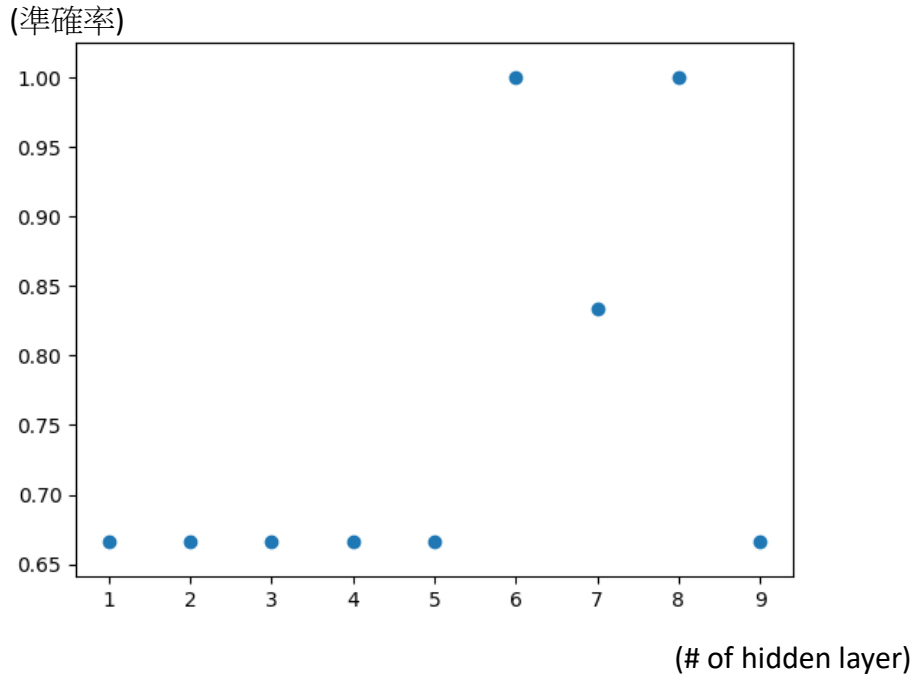
k(# of features) = 5 , M(# of datas) = 30

Result:

執行方式： python3 ANN.py

執行結果:

可以從下圖看出，在 hidden layer 數目為 6 以及 8 的時候準確率為百分之百，並且大致上維持的趨勢是 hidden layer 的數目越多，準確率通常越高。



V. Support Vector Machines

k(# of features) = 5 , M(# of datas) = 30

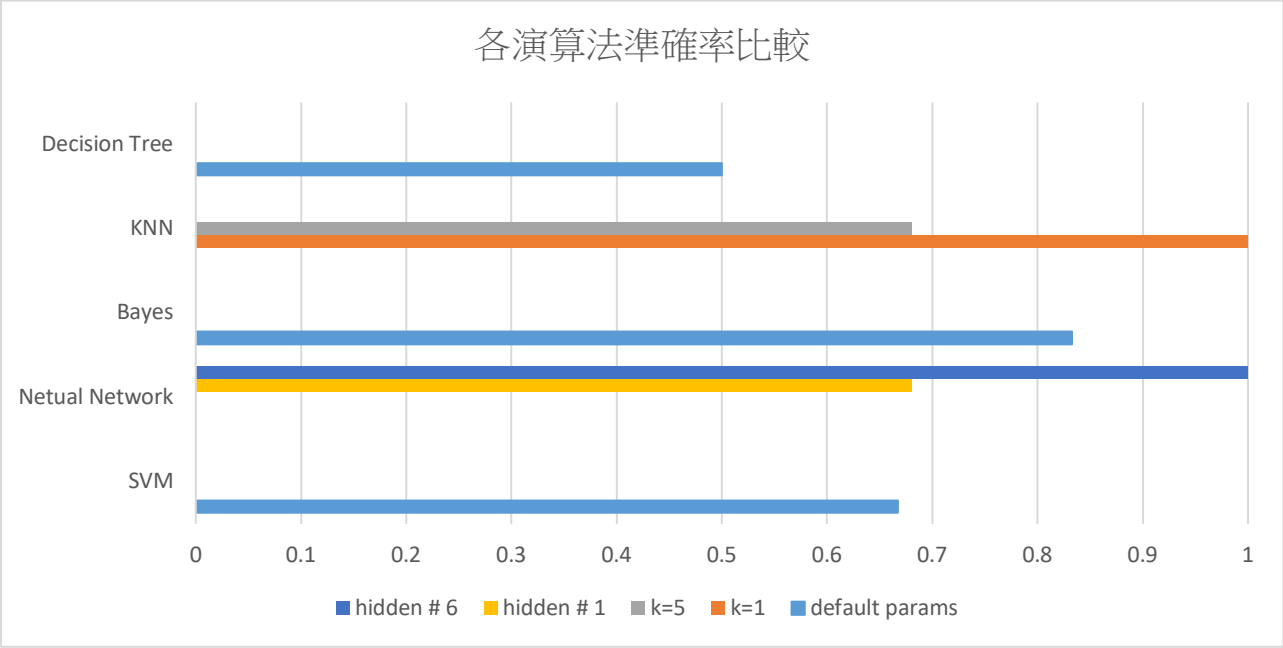
Result:

執行方式： python3 SVM.py

執行結果:

準確率：0.6666666667

比較&結論:



在 k=5,M=30 以及 train/test set 都相同的情況下，準確率是 ANN = KNN > Bayes > SVM > Decision Tree，雖然可能因為係數的調整而導致結果會大不相同，但基本上可以看出 decision tree 在分類的時候，優點是結構簡單並且沒有參數，但缺點是由於 Generalization 的能力太差，所以很難處理沒有 train 到的或者沒有出現過的值，也因此實際預測效果並不如其他的方法。

至於其他的方法如 KNN、ANN……等等，因為大多有參數上的調整差距，往往需要根據每個不同的問題來調整最佳的參數，所以並不能一概的拿來比較準確率，但也多少能從個別演算法的參數調整來看出一些趨勢，例如類神經網路的黑盒運算，其 hidden layers 數目在多的時候通常預測結果會比少的時候還好。