Data mining HW2

NE6101034

AI 碩一 柳譯筑

因為我的圖片 export 成 pdf 有點跑掉，所以如果助教方便的話可以到下面的網址看沒有跑掉的 👀

https://cubic-cycle-30f.notion.site/Data-mining-HW2-3043f5239d5e49339682bd3e0a024804

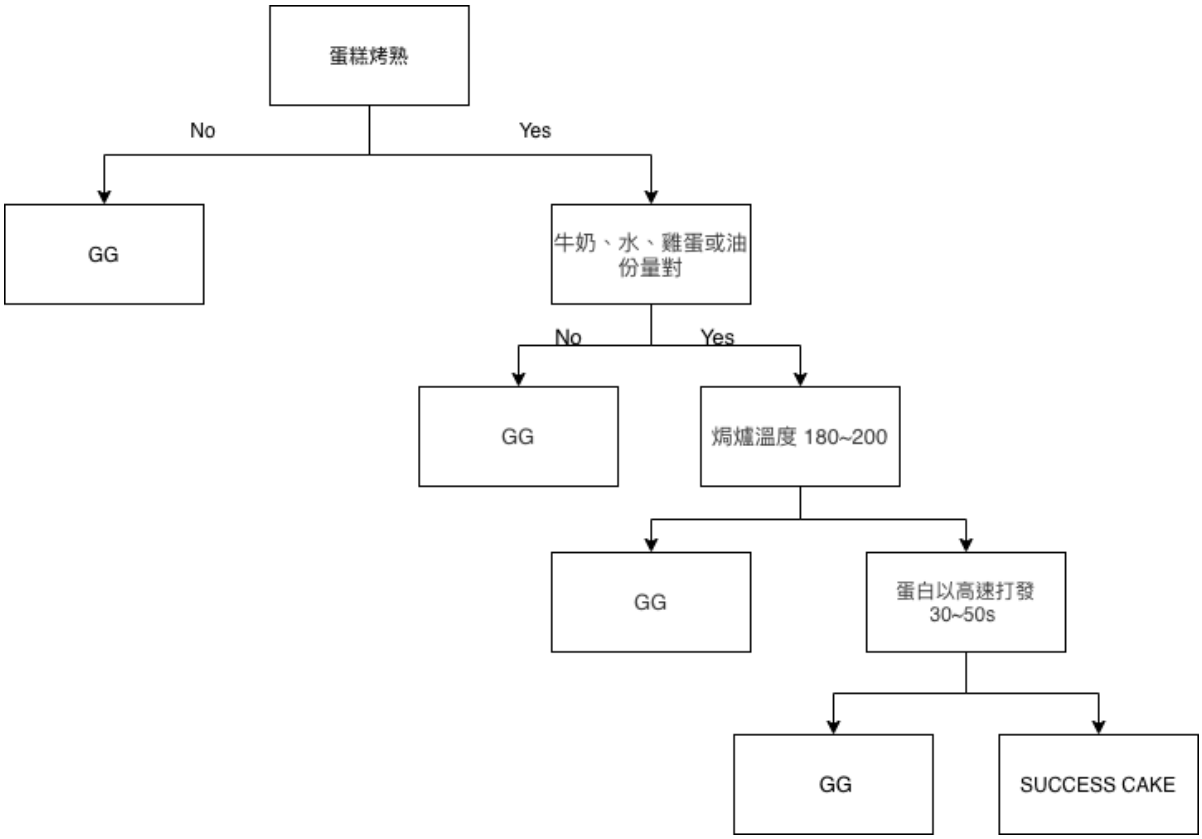
題目

Step 1: Design a set of rules to classify data, e.g., classify students with good performance.
¤ You should design k features / attributes for your problems first.
¤ Use ‘absolutely right’ rules to generate your positive and negative data (the number of data = M)

1. 第一個設定的 Absolute right rule （共 4 features）

   Boolean: 蛋糕烤熟、食材份量對不對

   離散值：溫度（150~230）、打發速度（30~55）



2. redundant attributes（共 3 attributes）

   蛋糕的顏色（3種）、口味（4種）、尺寸（6種）

- 總共先生成 500 筆資料

```
def good_cake(baked,weight,temperature,speed,color):
    if baked and weight and temperature<=200 and temperature>=180 and speed>=30 and speed<=50  :
        return True
    return False
```

Step 2: Use the data generated in Step 1 to construct your classification model ¤ Decision tree is basic requirement, you can add more classification models.

- **Decision tree**
  - 接著我拿60% (300筆) 的資料訓練 40%（200筆）的資料做測試，用decision tree 做出來的結果 accuracy 幾乎都是 1.0
  - accuracy 1.0
  - predict 結果:

goodcake

	baked	weight	speed	temperature	size	color	flavor	Y	predict
195	1	1	54	201	1	1	2	FALSE	FALSE
410	0	0	39	216	5	3	3	FALSE	FALSE
258	1	0	36	202	1	1	1	FALSE	FALSE
475	1	0	44	216	1	2	2	FALSE	FALSE
391	1	0	36	183	2	3	1	FALSE	FALSE
297	0	1	55	214	6	1	2	FALSE	FALSE
51	1	0	50	150	4	3	2	FALSE	FALSE
383	1	0	55	210	1	3	1	FALSE	FALSE
108	0	0	38	225	5	3	2	FALSE	FALSE
432	0	1	34	175	3	2	2	FALSE	FALSE
177	0	1	33	178	4	1	4	FALSE	FALSE
325	0	0	31	212	2	2	4	FALSE	FALSE
247	1	0	45	204	6	2	1	FALSE	FALSE
175	0	1	44	150	4	3	4	FALSE	FALSE
234	1	0	48	151	1	1	2	FALSE	FALSE
346	0	0	33	216	3	3	4	FALSE	FALSE
46	0	0	49	161	1	3	4	FALSE	FALSE
109	0	1	40	206	5	1	1	FALSE	FALSE
214	1	1	47	193	1	3	2	TRUE	TRUE
499	0	1	43	200	4	2	3	FALSE	FALSE
315	0	0	37	228	6	3	2	FALSE	FALSE
453	0	1	49	156	3	2	4	FALSE	FALSE
390	0	1	53	190	4	3	4	FALSE	FALSE
43	1	0	44	179	5	2	4	FALSE	FALSE
264	1	1	31	203	1	1	3	FALSE	FALSE
268	1	1	52	230	2	2	1	FALSE	FALSE
269	1	0	54	215	6	1	4	FALSE	FALSE

- **RandomForest**
  - n_estimators = 100, 60% (300筆) 的資料訓練 40%（200筆）的資料做測試
  - accuracy : 1.0
  - predict 結果:

goodcake_randomforest

	baked	weight	speed	temperature	size	color	flavor	Y	predict
165	0	0	47	158	6	3	4	FALSE	FALSE
29	0	1	33	219	3	1	1	FALSE	FALSE
196	0	0	37	186	3	1	1	FALSE	FALSE
283	0	0	34	210	3	1	2	FALSE	FALSE
389	0	0	31	169	1	3	4	FALSE	FALSE
442	0	1	43	183	1	3	3	FALSE	FALSE
210	0	1	43	214	6	3	4	FALSE	FALSE
448	1	1	31	166	1	1	3	FALSE	FALSE
376	1	1	43	157	5	2	3	FALSE	FALSE
36	0	1	41	204	1	3	1	FALSE	FALSE
711	1	1	33	194	4	1	4	TRUE	TRUE
23	1	1	38	195	2	2	3	TRUE	TRUE
157	0	1	42	189	2	1	1	FALSE	FALSE
775	1	1	35	183	2	3	1	FALSE	FALSE
479	1	1	52	205	6	1	4	FALSE	FALSE
208	1	1	48	155	6	3	2	FALSE	FALSE
796	1	1	34	184	4	1	4	TRUE	TRUE
347	1	1	40	179	3	2	2	FALSE	FALSE
31	0	1	52	167	2	3	3	FALSE	FALSE
196	0	0	37	186	3	1	1	FALSE	FALSE
10	0	0	53	191	6	1	2	FALSE	FALSE
150	1	1	51	171	3	1	1	FALSE	FALSE
674	1	1	39	190	4	1	1	TRUE	TRUE
925	1	1	39	197	3	3	2	FALSE	FALSE

- **Naive Bayes Classification**

- accuracy 0.84 左右

goodcake_Bayes

	baked	weight	speed	temperature	size	color	flavor	Y	predict
183	0	0	43	212	2	2	4	FALSE	FALSE
21	0	1	47	170	2	2	1	FALSE	FALSE
799	1	1	49	184	1	3	3	TRUE	TRUE
273	1	0	37	183	1	3	4	FALSE	FALSE
199	0	1	53	197	4	1	3	FALSE	FALSE
740	1	1	51	188	5	1	1	FALSE	TRUE
37	0	0	38	181	2	1	2	FALSE	FALSE
626	1	1	49	199	3	3	2	TRUE	TRUE
321	1	0	46	196	4	1	4	FALSE	FALSE
843	1	1	39	188	2	2	2	TRUE	TRUE
623	1	1	49	186	4	3	2	TRUE	TRUE
625	1	1	43	194	5	1	1	TRUE	TRUE
326	1	1	34	202	4	3	1	FALSE	TRUE
468	1	1	50	159	1	3	1	FALSE	TRUE
258	0	1	30	177	2	3	1	FALSE	FALSE
287	1	0	47	189	1	2	1	FALSE	FALSE
790	1	1	47	193	2	1	4	TRUE	TRUE
856	1	1	45	199	4	1	3	TRUE	TRUE
385	0	1	39	169	5	1	4	FALSE	FALSE
471	1	1	36	201	1	1	2	FALSE	TRUE
94	1	0	35	204	5	1	4	FALSE	FALSE
904	1	1	54	187	2	3	3	FALSE	TRUE
281	0	0	51	153	6	3	2	FALSE	FALSE
822	1	1	44	200	5	2	3	TRUE	TRUE
435	0	0	47	192	3	1	2	FALSE	FALSE
32	0	0	32	214	2	3	3	FALSE	FALSE
70	1	0	35	180	2	1	1	FALSE	FALSE
286	1	0	31	221	5	3	4	FALSE	FALSE
300	0	1	47	180	6	1	2	FALSE	FALSE
366	0	0	34	219	3	2	4	FALSE	FALSE
891	1	1	33	186	5	3	1	TRUE	TRUE
829	1	1	41	181	5	2	3	TRUE	TRUE
885	1	1	50	200	5	1	4	TRUE	TRUE
6	1	1	38	177	5	2	2	FALSE	TRUE

- 再測試了 **knn classifier**

  n_neighbors 從 2~6 accuracy 差不多

  accuracy : 0.89

goodcake_knn

	baked	weight	speed	temperature	size	color	flavor	Y	predict
870	1	1	34	183	4	2	3	TRUE	TRUE
432	1	0	52	206	5	1	3	FALSE	FALSE
741	1	1	32	198	3	2	2	TRUE	TRUE
254	1	0	38	159	4	1	4	FALSE	FALSE
139	1	0	32	188	6	2	3	FALSE	TRUE
336	0	1	55	204	4	1	1	FALSE	FALSE
618	1	1	54	181	6	2	1	FALSE	FALSE
127	0	1	36	167	2	1	1	FALSE	FALSE
106	1	1	36	220	3	3	2	FALSE	FALSE
86	0	0	40	168	5	2	4	FALSE	FALSE
607	1	1	46	199	4	3	1	TRUE	TRUE
179	1	0	48	220	6	1	3	FALSE	FALSE
941	1	1	42	185	5	3	3	TRUE	TRUE
415	0	0	30	185	5	1	1	FALSE	TRUE
634	1	1	48	197	2	1	1	TRUE	TRUE
36	0	1	37	200	6	1	1	FALSE	TRUE
12	0	1	53	160	6	1	3	FALSE	FALSE
368	1	1	36	213	6	3	1	FALSE	FALSE
430	1	0	54	158	5	2	4	FALSE	FALSE
659	1	1	55	183	2	3	1	FALSE	FALSE
498	1	0	37	174	3	2	4	FALSE	FALSE
629	1	1	35	191	2	2	4	TRUE	FALSE
688	1	1	44	193	3	3	3	TRUE	TRUE
477	0	0	31	159	4	2	4	FALSE	FALSE
183	0	1	43	203	1	2	4	FALSE	FALSE
586	1	1	38	197	6	2	3	TRUE	FALSE
972	1	1	33	200	6	3	2	TRUE	TRUE
218	0	0	52	173	6	3	4	FALSE	FALSE
854	1	1	35	183	4	1	1	TRUE	TRUE
114	1	0	45	160	2	3	4	FALSE	FALSE
165	0	1	49	161	1	2	3	FALSE	FALSE
512	1	1	55	180	6	2	2	FALSE	FALSE
143	0	1	36	192	1	3	1	FALSE	TRUE

Step 3: Compare the rules in the decision tree from Step 2 and the rules you used to generate your ‘right’ data , Step 4: Discuss anything you can

Visualizing Decision tree

```
Decision tree
(feature_0 蛋糕烤熟, feature_1 食材份量對不對, feature_2 speed, feature_3 temperature)
|--- feature_1 <= 0.50
|   |--- class: False
|--- feature_1 >  0.50
|   |--- feature_0 <= 0.50
|   |   |--- class: False
|   |--- feature_0 >  0.50
|   |   |--- feature_3 <= 200.50
|   |   |   |--- feature_3 <= 179.00
|   |   |   |   |--- class: False
|   |   |   |--- feature_3 >  179.00
|   |   |   |   |--- feature_2 <= 49.50
|   |   |   |   |   |--- feature_2 <= 30.50
|   |   |   |   |   |   |--- class: False
|   |   |   |   |   |--- feature_2 >  30.50
|   |   |   |   |   |   |--- class: True
|   |   |   |   |--- feature_2 >  49.50
|   |   |   |   |   |--- class: False
|   |   |--- feature_3 >  200.50
|   |   |   |--- class: False
```

Decision tree 基本上跟我的 absolute right rule是一樣的，但判斷條件數據還是有一點點誤差，像是我的條件是 temperature ≥180 true ，他條件給 temperature >179 true

所以我額外加入了一筆 temperature 179.5，結果 rule 的數據改成 >179.75 true

```
Decision tree accuracy 1.0
|--- feature_1 <= 0.50
|   |--- class: False
|--- feature_1 >  0.50
|   |--- feature_0 <= 0.50
|   |   |--- class: False
|   |--- feature_0 >  0.50
|   |   |--- feature_2 <= 50.50
|   |   |   |--- feature_3 <= 200.50
|   |   |   |   |--- feature_3 <= 179.75
|   |   |   |   |   |--- class: False
|   |   |   |   |--- feature_3 >  179.75
|   |   |   |   |   |--- class: True
```

```
|   |   |   |--- feature_3 >  200.50
|   |   |   |   |--- class: False
|   |   |--- feature_2 >  50.50
|   |   |   |--- class: False
```

不過多執行幾次，跑出來的 rule 會不太一樣，下面也是有那筆另外多加的 data，在 temperature 的數據又不一樣了。

```
accuracy 1.0
|--- feature_0 <= 0.50
|   |--- class: False
|--- feature_0 >  0.50
|   |--- feature_1 <= 0.50
|   |   |--- class: False
|   |--- feature_1 >  0.50
|   |   |--- feature_2 <= 50.50
|   |   |   |--- feature_3 <= 179.50
|   |   |   |   |--- class: False
|   |   |   |--- feature_3 >  179.50
|   |   |   |   |--- feature_3 <= 200.50
|   |   |   |   |   |--- class: True
|   |   |   |   |--- feature_3 >  200.50
|   |   |   |   |   |--- class: False
|   |   |--- feature_2 >  50.50
|   |   |   |--- class: False
```

由於準確率都超高（基本上 100% 準確），因為一開始我設的兩個 Boolean 用 random 在前兩個條件就會篩成約剩下 1/4 True, 再加上後面的條件會導致 True 太少分類結果太不平均，所以我寫死了 baked = True, weight = True，讓兩個類別 (true / false) 的數量平均一點。

現在總共是1000筆資料

結果還是可以100%準確，應該不是 negative 太多的問題，感覺單純就是我設的條件 Decision tree 可以完全學到（畢竟 if else 基本上就是 decision tree）。

- 我接著稍微調整了原本的 absolute right rule

```
# rule2
if color == 1 or color==2: #顏色是1或2的成功條件
        if baked and weight and temperature<=200 and temperature>=180 and speed>=30 and speed<=50  :
            return True
        else:
            return False
    elif color == 4: #顏色是4的成功條件
        if baked and weight and temperature<=200 and temperature>=190 and speed>=35 and speed<=50  :
            return True
        else:
            return False
    return False #顏色如果是3就不會成功
```

這是第二個 rule 的 Decision Tree：

```
|--- feature_5 <= 2.50
|   |--- feature_1 <= 0.50
|   |   |--- class: False
|   |--- feature_1 >  0.50
|   |   |--- feature_0 <= 0.50
|   |   |   |--- class: False
|   |   |--- feature_0 >  0.50
|   |   |   |--- feature_2 <= 50.50
|   |   |   |   |--- feature_3 <= 200.50
|   |   |   |   |   |--- feature_3 <= 178.00
|   |   |   |   |   |   |--- class: False
|   |   |   |   |   |--- feature_3 >  178.00
|   |   |   |   |   |   |--- class: True
|   |   |   |   |--- feature_3 >  200.50
|   |   |   |   |   |--- class: False
|   |   |   |--- feature_2 >  50.50
|   |   |   |   |--- class: False
|--- feature_5 >  2.50
|   |--- class: False
```

也是跟我的 rule 完全一樣 100%準確，feature 5 就是顏色的判斷

所以我做了第三種 rule - 把不同的attribute去做了關聯，也就是用到oblique decision tree的概念

[Note] About *oblique decision tree*:

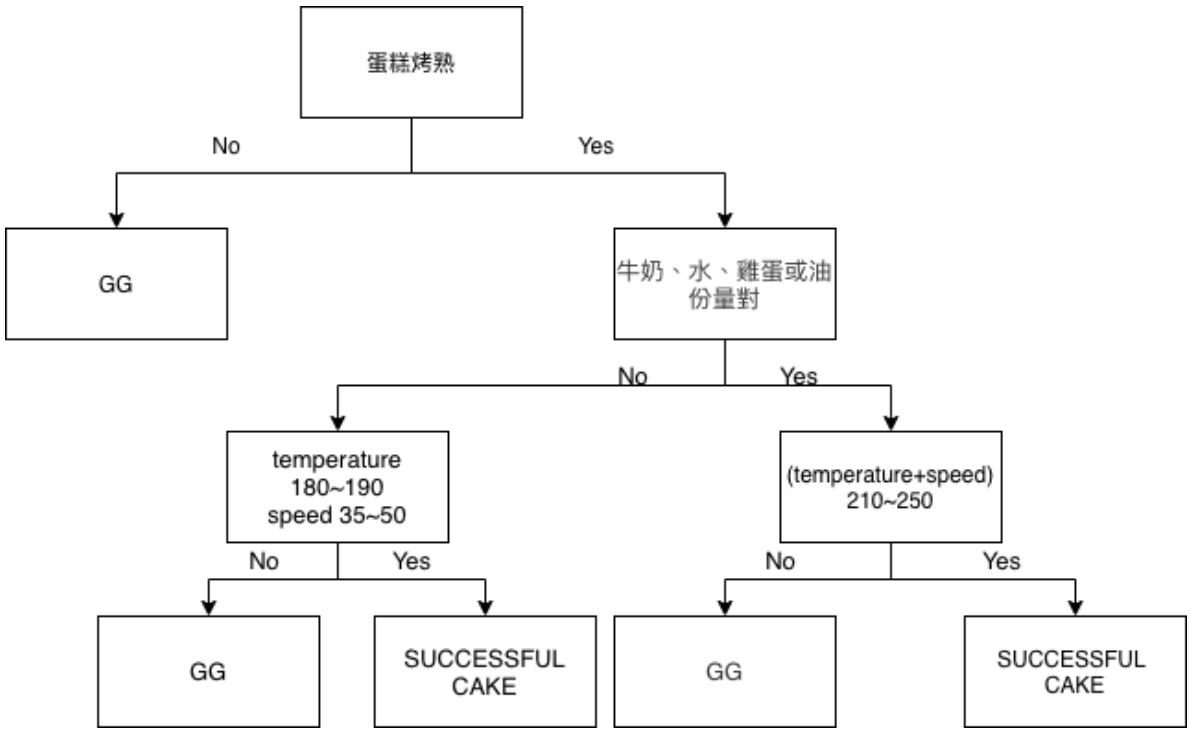
(1) test condition may involve multiple attributes

(2) more expressive representation

(3) finding optimal test condition is computationally expensive

Rule 3

```
# rule3
if baked and weight:
    if (temperature+speed)>=210 and (temperature+speed)<=250:
      return True
    elif temperature<=190 and temperature>=180 and speed>=35 and speed<=50:
      return True
  return False
```



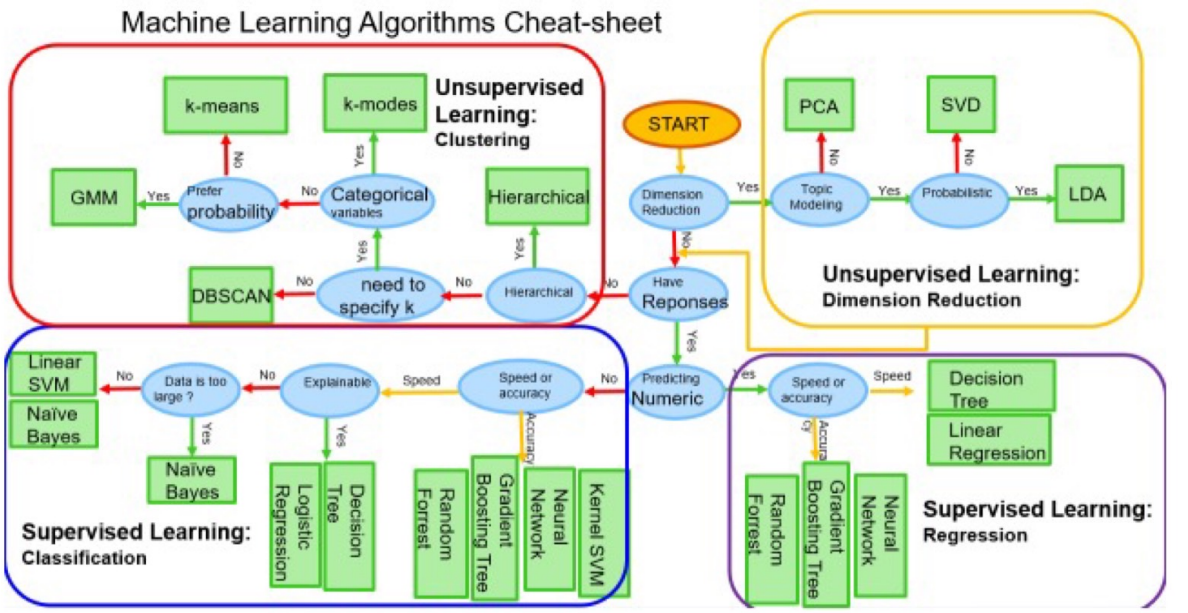

```
[1500 rows x 8 columns]
accuracy 0.9916666666666667
|--- feature_0 <= 0.50
|   |--- class: False
|--- feature_0 >  0.50
|   |--- feature_1 <= 0.50
|   |   |--- class: False
|   |--- feature_1 >  0.50
|   |   |--- feature_3 <= 204.50
|   |   |   |--- feature_3 <= 169.00
|   |   |   |   |--- feature_3 <= 165.50
|   |   |   |   |   |--- class: False
|   |   |   |   |--- feature_3 >  165.50
|   |   |   |   |   |--- feature_2 <= 40.00
|   |   |   |   |   |   |--- class: False
|   |   |   |   |   |--- feature_2 >  40.00
|   |   |   |   |   |   |--- class: True
|   |   |   |--- feature_3 >  169.00
|   |   |   |   |--- feature_2 <= 50.50
|   |   |   |   |   |--- feature_3 <= 177.00
|   |   |   |   |   |   |--- feature_2 <= 35.00
|   |   |   |   |   |   |   |--- class: False
|   |   |   |   |   |   |--- feature_2 >  35.00
|   |   |   |   |   |   |   |--- feature_5 <= 1.50
|   |   |   |   |   |   |   |   |--- class: False
|   |   |   |   |   |   |   |--- feature_5 >  1.50
|   |   |   |   |   |   |   |   |--- class: True
|   |   |   |   |   |--- feature_3 >  177.00
|   |   |   |   |   |   |--- class: True
|   |   |   |   |--- feature_2 >  50.50
|   |   |   |   |   |--- feature_3 <= 195.50
```

```
|   |   |   |   |   |   |   |--- class: True
|   |   |   |   |   |   |--- feature_3 >  195.50
|   |   |   |   |   |   |   |--- feature_3 <= 198.50
|   |   |   |   |   |   |   |   |--- feature_2 <= 53.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 4.50
|   |   |   |   |   |   |   |   |   |   |--- class: True
|   |   |   |   |   |   |   |   |   |--- feature_4 >  4.50
|   |   |   |   |   |   |   |   |   |   |--- class: False
|   |   |   |   |   |   |   |   |--- feature_2 >  53.50
|   |   |   |   |   |   |   |   |   |--- class: False
|   |   |   |   |   |   |   |--- feature_3 >  198.50
|   |   |   |   |   |   |   |   |--- class: False
|   |   |   |--- feature_3 >  204.50
|   |   |   |   |--- feature_2 <= 36.50
|   |   |   |   |   |--- feature_3 <= 217.50
|   |   |   |   |   |   |--- class: True
|   |   |   |   |   |--- feature_3 >  217.50
|   |   |   |   |   |   |--- class: False
|   |   |   |   |--- feature_2 >  36.50
|   |   |   |   |   |--- class: False
```

結果讓兩個 attribute 互相關聯相較於可以 100% 正確的獨立 attribute ，他的 accuracy 雖然也接近完全正確，但至少會錯個兩三筆。

其他心得

- 對於 decision tree, random forest model 的結論是只要把條件設得複雜一點， decision model 就可能會學不到你的 absolute right rule，另外 oblique decision tree 的 computation 也更複雜，他不會學到你包含兩個 attribute 的 rule，而是每個節點都會去判斷一次其他獨立的 feature。
- 由於生成資料的原理比較是 base on decision tree 這種 if else的感覺，其他 Model 像是 Naive Bayes, KNN model 的 accuracy 就比較低
- 關於選擇 model



Rule base dataset 顯然的是適合用 decision tree or Random forest，這次測試Naive Bayes 的結果特別差，在 explainable 那個條件我覺得是關鍵，另外 Naive Bayes 也比較適合 independent attribute。

以上是我在這個作業所觀察到的簡易分類器（主要是Decision Tree） 所得到的特性分析，感謝助教耐心批閱🍇