# Data Mining Hw2 Report

👧 P77111037 樊紹萱

## 1️⃣ 檔案結構與開發環境

1. tool: Pipenv
2. python version: python 3.8

```
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
pandas = "==1.2.0"
numpy = "==1.23.4"
tensorflow = "==2.10.0"
sklearn = "*"
scikit-learn = "*"
matplotlib = "*"

[dev-packages]

[requires]
python_version = "3.8"
```

## 2️⃣ Data Design

### 💡 Problem Definition: 決定柴犬是否可以超過10歲？

- 利用以下 11 種特徵和 5 個規則，來判定柴柴是否能活超過十歲
- file: `genrerate_data.py`
- 共生成 `5000` 筆資料
- inputs > data.csv

| Current Age | Weight | Height | Gender | Month Of Birthbsp; | Hair Color | Food | Feeding Method | Temper | Exercise Frequency | First Alphabet Of Name | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 7 | 30 | 0 | 1 | 4 | 2 | 0 | 0 | 4 | 6 | 1 |
| 3 | 9 | 33 | 0 | 1 | 1 | 1 | 2 | 0 | 7 | 4 | 1 |
| 4 | 12 | 34 | 0 | 1 | 2 | 2 | 0 | 0 | 7 | 21 | 0 |
| 4 | 6 | 43 | 0 | 1 | 4 | 3 | 0 | 1 | 3 | 14 | 1 |
| 7 | 8 | 41 | 0 | 1 | 4 | 0 | 2 | 0 | 4 | 1 | 1 |

### ⭐ Feature

1. Current age: 0 ~ 9
2. Weight (kg): 5~15
3. Height (cm): 30 ~ 45
4. Gender: `F` or `M`

5. Month of birth: 1 ~ 12
6. Hair Color: `yellow` or `black` or `white` or `black and tan`
7. Food: `feed` or `can` or `fresh`
8. Feeding Method: `stocking` or `indoor` or `indoor and yard`
9. Temper: `good` or `bad`
10. Exercise frequency per week: 1~7
11. First alphabet of name: A ~ Z

Reference

### ⭐ Rule

📝 若滿足以下**五個規則之一**，則判定該柴柴會活超過10歲

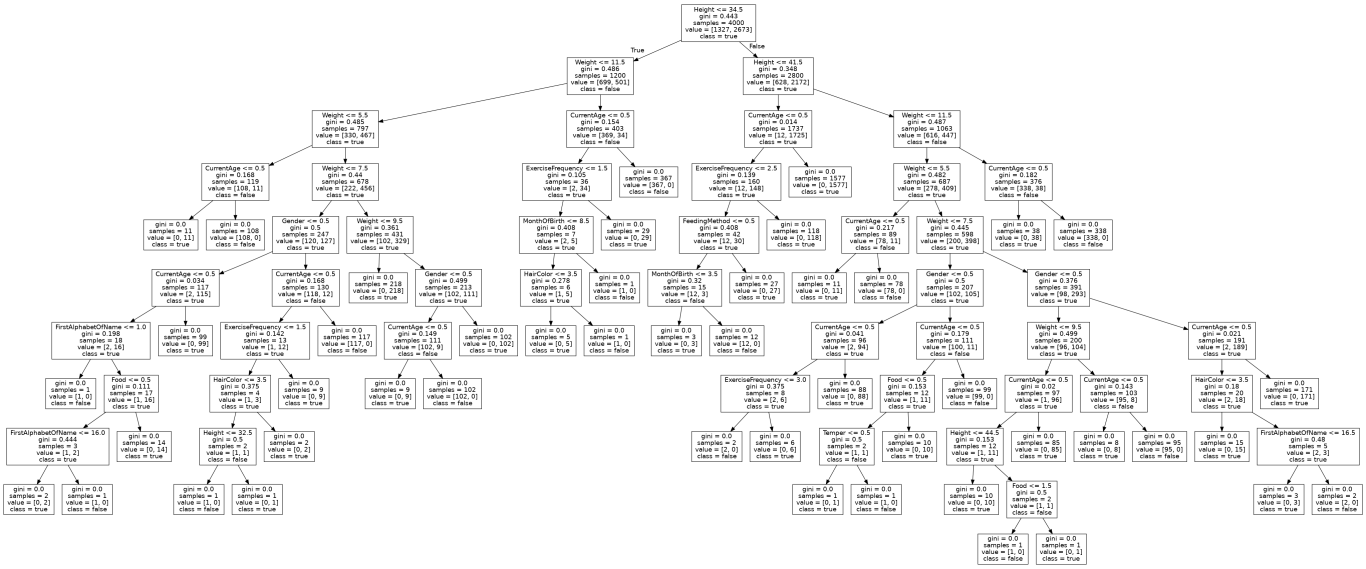
1. If current age >= 1:
    - Female: weight 👉 6 ~ 9
    - Male: weight 👉 8 ~ 11
2. If current age >= 1:
    - Height: 35 ~ 41
3. Month of birth (If is Fibonacci): 1,2,3,5,8
4. Feeding Method: `indoor` or `indoor and yard`
5. Exercise frequency per week: 3~7

## 3️⃣ Classification Models

1. Decision Tree:
    - file: `decision_tree.py`
2. Any models of your preference 👉 KNN:
    - file: `knn.py`

## 4️⃣ Report

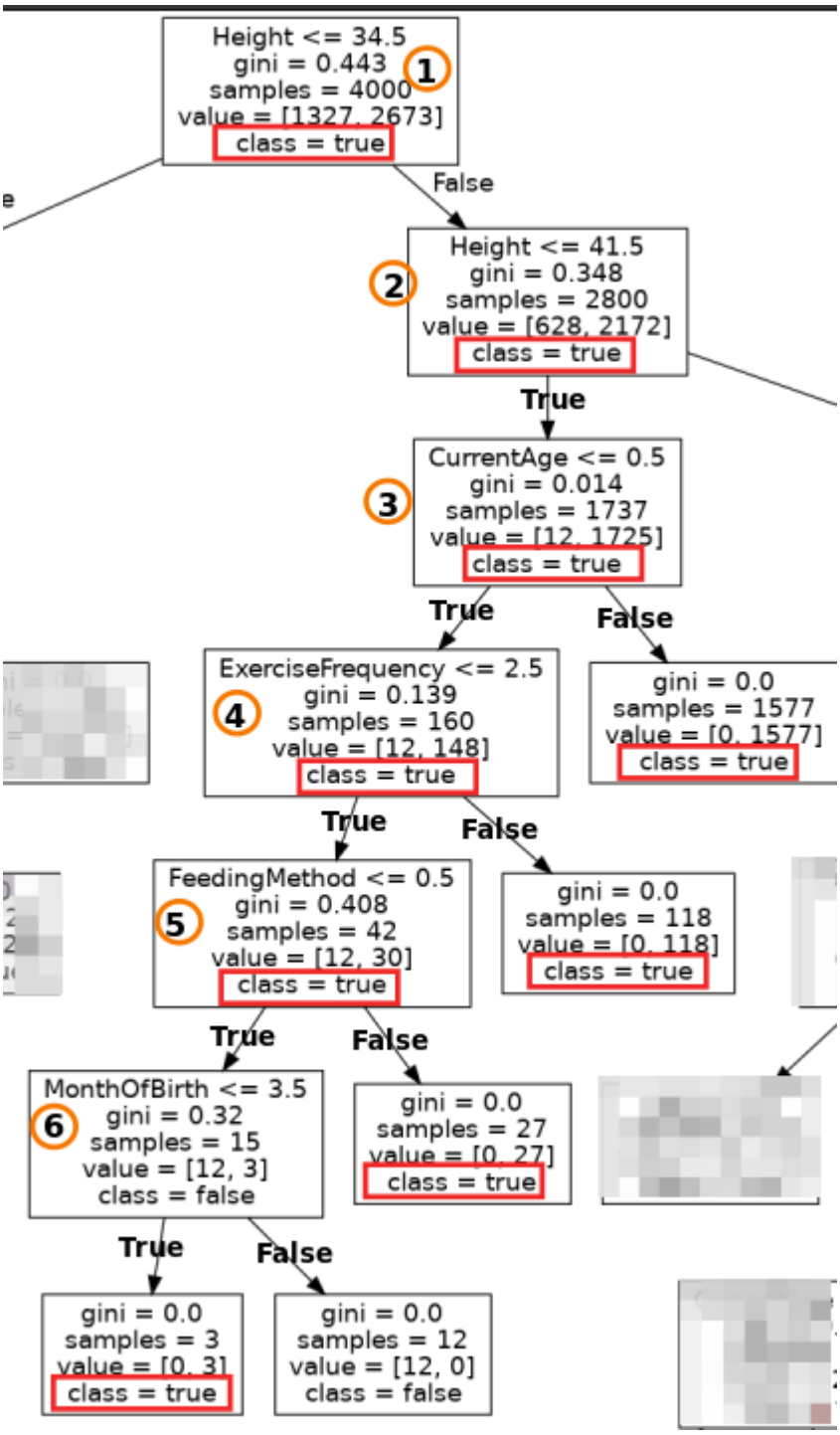
### 🌲 Decision Trees




## 🌲 Comparisons

Compare your absolutely right rules with the rules generated by the classification model(s).

**Compare (以最終類別是 ⭕ 的情況來說)**




| Index | DecisionTreeClassifier() rules | My rules |
|---|---|---|
| 1 | [class 👉 True] Height >=34.5 | 📌 If CurrentAge >= 1 **&** Height: 35 ~ 41 👉 True<br>✏️ 發現其實這邊蠻像的，決策樹的規則是 Height >=34.5 且 <=41.5 (可對比我的 Height 需介於 35 ~ 41)<br>✏️ 決策樹的規則是以 CurrentAge <=0.5 來做決策，而我是用 CurrentAge <=1 |
| 2 | [class 👉 True] Height <=41.5 | ^ 同上 |

| Index | DecisionTreeClassifier() rules | My rules |
|---|---|---|
| 3 | [class 👉 True]<br>CurrentAge <=0.5 | ^ 同上 |
| 4 | [class 👉 True]]<br>ExerciseFrequency <=2.5 | 📌 Exercise frequency per week: 3~7 👉 True<br>✏️ 決策樹是以 ExerciseFrequency 2.5 來做分界，而我自訂的規則是 3。<br>✏️ 綜合上一個 CurrentAge ， 決策樹這邊若 CurrentAge <= 0.5 時，即使 ExerciseFrequency <= 2.5 ，則最後的結果還是判定為 True。因為我的規則裡沒有去限制 CurrentAge <1 的情況，所以是合裡的。 |
| 5 | [class 👉 True]]<br>FeedingMethod <=0.5 | 📌 Feeding Method: `indoor` (對應到的值：1) or `indoor and yard` (對應到的值：2)<br>✏️ 決策樹是定義 >= 0.5 時則結果判斷為 True，可對應到我的規則，當 Feeding Method = 1 or 2 時會判定結果為 True |
| 5 | [class 👉 True]]<br>MonthOfBirth <=3.5 | 📌 Month of birth (If is Fibonacci): 1,2,3,5,8<br>❗ **決策樹是定義 <= 3.5** 時則結果判斷為 True，跟我的規則稍稍有衝突，因為**我是定義 MonthOfBirth in [1,2,3,5,8]** 時會判定結果為 True，我自己的想法是認為因為 3.5 差不多是等於 [1,2,3,5,8] 的平均，決策樹才會這樣子判斷的 |

## 🌲 & **KNN** Discussion

**更動原本的規則**

👉 把 CurrentAge 改為 >=0.8 （原本是 >=1） 👉 生成第二組資料集:

- 共生成 `5000` 筆資料
- inputs > data-2.csv

| Current Age | Weight | Height | Gender | Month Of Birth | Hair Color | Food | Feeding Method | Temper | Exercise Frequency | First Alphabet Of Namesp | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 8 | 36 | 1 | 1 | 0 | 0 | 2 | 0 | 4 | 25 | |
| 8 | 9 | 35 | 1 | 3 | 4 | 2 | 1 | 0 | 6 | 25 | |
| 6 | 9 | 41 | 0 | 7 | 1 | 2 | 1 | 0 | 1 | 25 | |
| 5 | 11 | 34 | 0 | 5 | 4 | 3 | 0 | 0 | 7 | 25 | |
| 5 | 12 | 32 | 0 | 2 | 2 | 1 | 2 | 0 | 2 | 25 | |

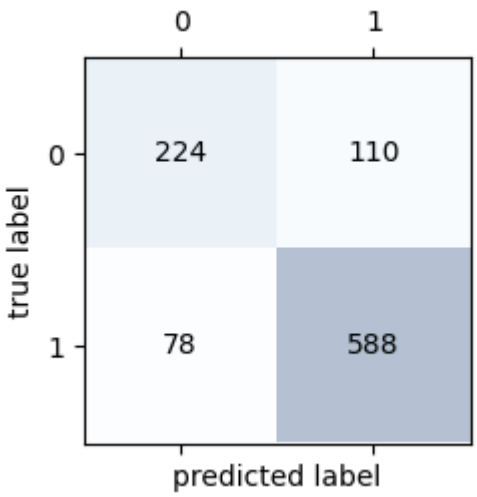
**KNN 影響 (以 confusion matrix 來解釋)**

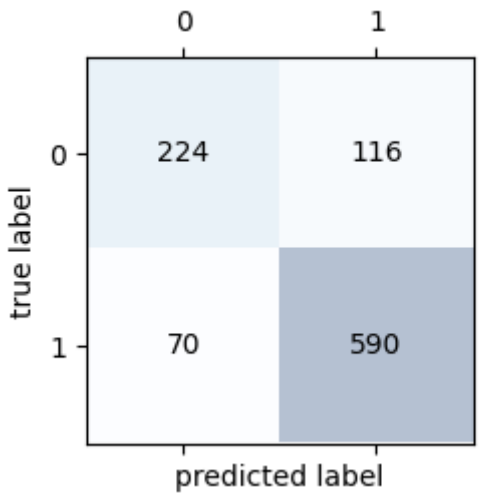
```
資料量： 5000 * 0.2 = 1000
neighbor: 5 ( default is 5)
algorithm: auto, ball_tree, kd_tree, brute (default is auto)
```

1️⃣ algorithm = `auto`

✏️ 左邊是原本的，右邊是 CurrentAge 改為 >=0.8

✏️ 可以看到調整 rule 後， **預測錯誤的數量減少了**。188 👉 186

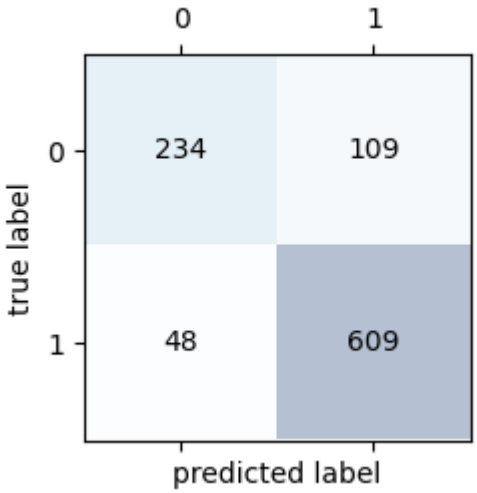


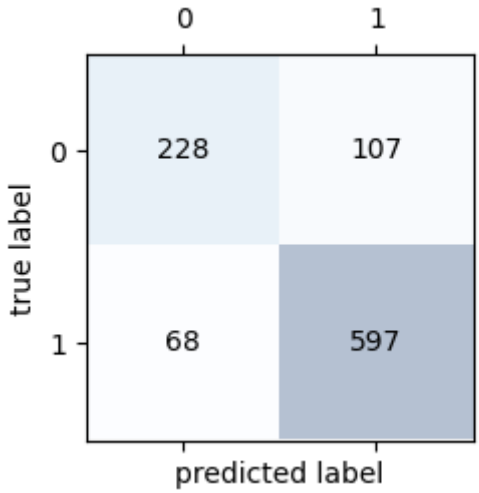




2️⃣ algorithm = `ball_tree`

✏️ 左邊是原本的，右邊是 CurrentAge 改為 >=0.8

✏️ 可以看到調整 rule 後， **預測錯誤的數量增加了**。157 👉 175

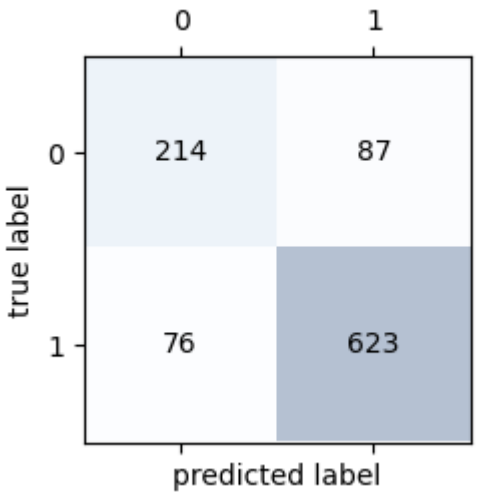


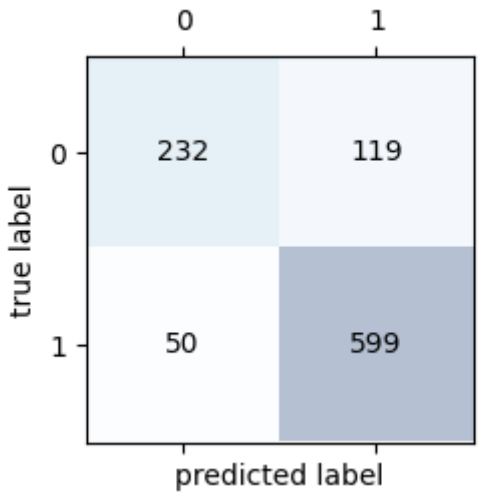




3️⃣ algorithm = `kd_tree`

✏️ 左邊是原本的，右邊是 CurrentAge 改為 >=0.8

✏️ 可以看到調整 rule 後， **預測錯誤的數量增加了**。163 👉 169

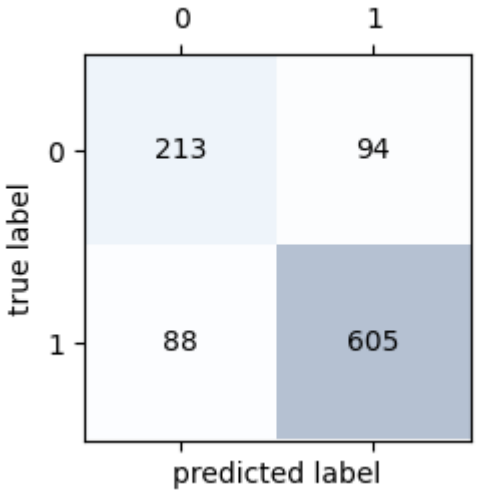


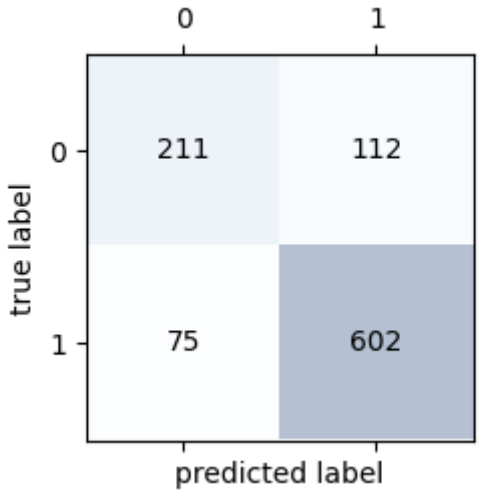




3️⃣ algorithm = `brute`

✏️ 左邊是原本的，右邊是 CurrentAge 改為 >=0.8

✏️ 可以看到調整 rule 後， **預測錯誤的數量增加了**。`182` 👉 `187`



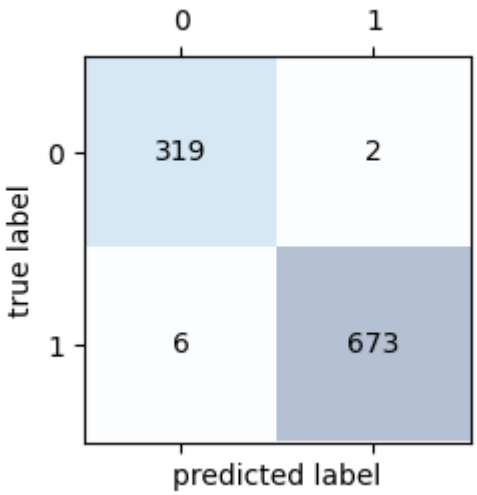

4️⃣ 結論

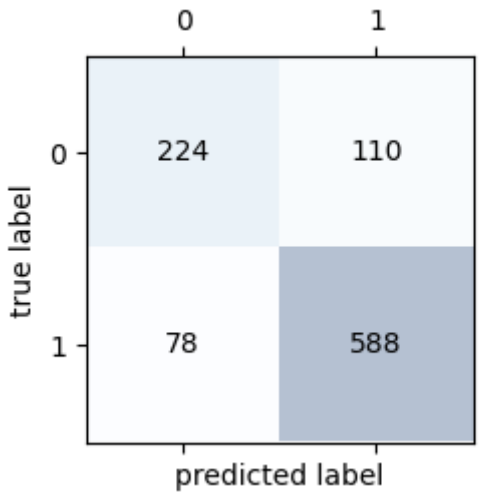
更改某一個規則後 (`CurrentAge 改為 >=0.8`)，只有 algorithm="auto" 的時候，預測的結果較好

5️⃣ 比較 Decision tree & KNN confusion matrix

```
資料量： 5000 * 0.2 = 1000
KNN:
   neighbor: 5 ( default is 5)
   algorithm: auto
```

✏️ 左邊是 Decision Tree，右邊是 KNN (algorithm=auto)


0
1
0
1
true label
319
2
6
673
predicted label


0
1
0
1
true label
224
110
78
588
predicted label