Data Mining Project2

Classification Analysis

F74102030 資訊114 練智剛

Data

1.1 Problem Definition/Data Design

身處在充滿交通亂象的台南兩年半，對於交通十分有感，所以設計這份資料藉由機車騎士的各種特徵，包括離散、連續、布林值等資料，來評判發生車禍的風險高低，風險值一共分為三種，分別為low risk, medium risk, high risk。

1.2 Features

Categorical Discrete

1. 性別 (man, woman)
2. 排氣量 (50, 100, 125, 150 cc)
3. 機車款式 (DRG, JET, cygnus, GP, duke, Many, famous, Fiddle, Cuxi, Limi, axis Z, Swish, POG, FORCE, BWS)
4. 後座是否有載人(分為 never, seldom, sometimes, usually, always)
5. 安全帽(1西瓜皮, 2普通安全帽, 3全罩式)

Gussian Continuous

6. 男生身高(平均173 cm、標準差8)
7. 女生身高(平均158 cm、標準差8)
8. 男生體重(平均70 kg、標準差10)
9. 女生體重(平均50 kg、標準差10)
10. 年齡(平均23 歲、標準差15、最小值設定為18歲)

Discrete Numerical

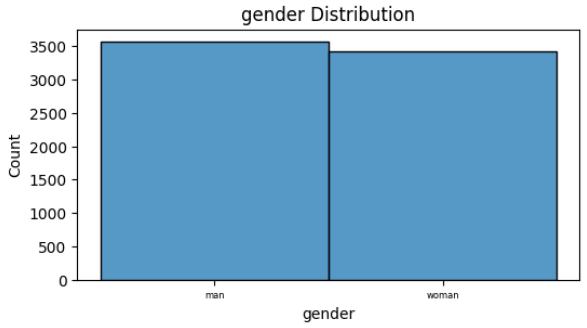
11. 購買機車價格(40k ~ 130k)

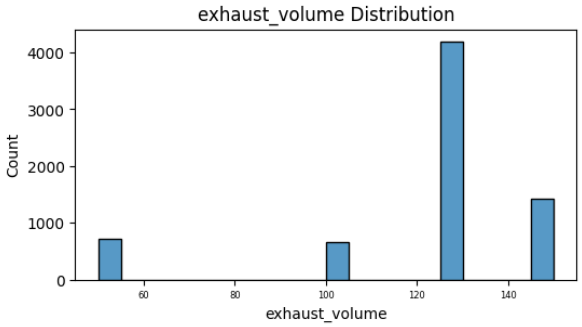
12. 每月里程數(0km ~ 1000km)
13. 收到交通罰單次數(0 ~ 10次)
14. 平時通常超過速限多少(0 ~ 30 km/hr)

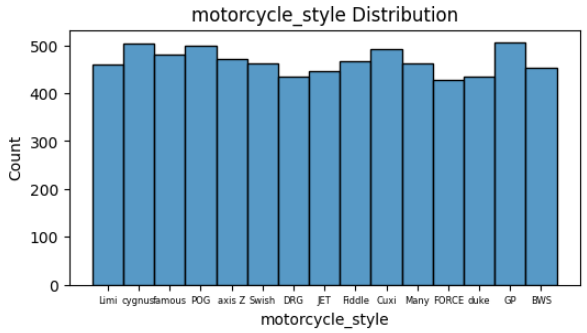
Discrete boolean

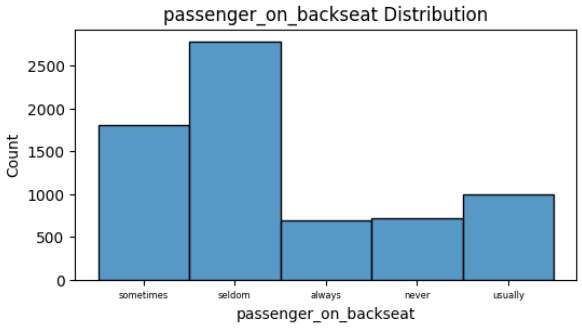
16. 駕照(0 沒有、1 有)
17. 改裝機車(0 沒有、1 有)(不包含排氣管)
18. 很吵的排氣管(0 沒有、1 有)
19. 跑山(0 沒有、1 有)
20. 定期機車保養(0 沒有、1 有)
21. 用機車通勤上班(0 沒有、1 有)

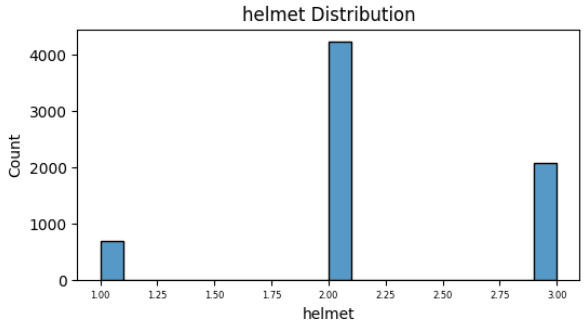
Bar Garph

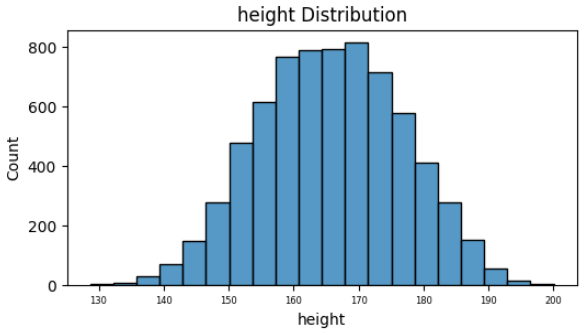


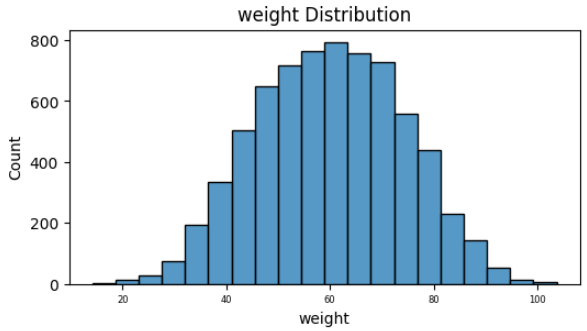


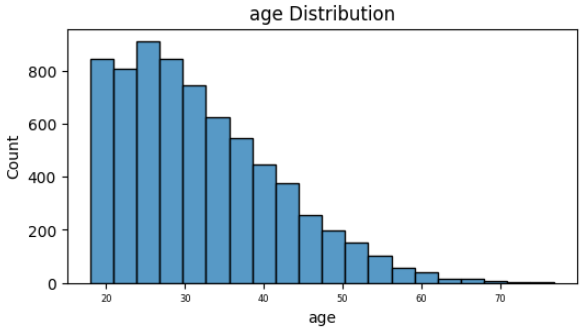


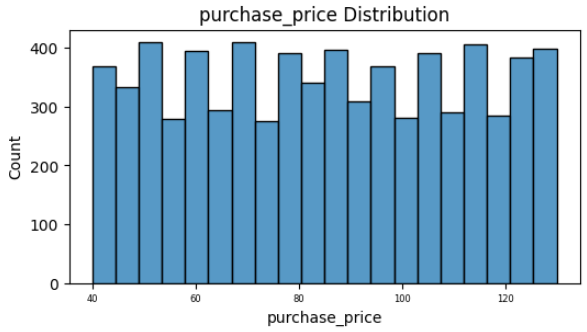


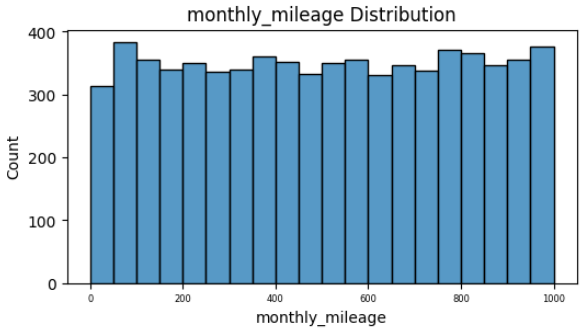

helmet Distribution
Count
helmet

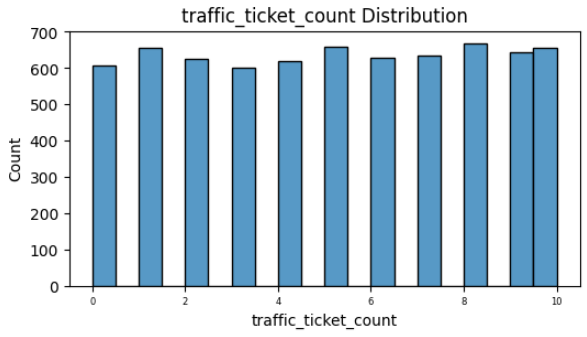

height Distribution
Count
height

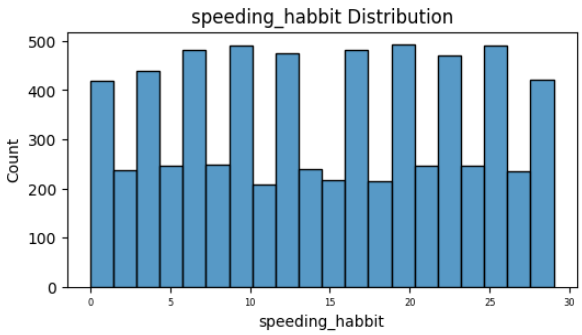

weight Distribution
Count
weight

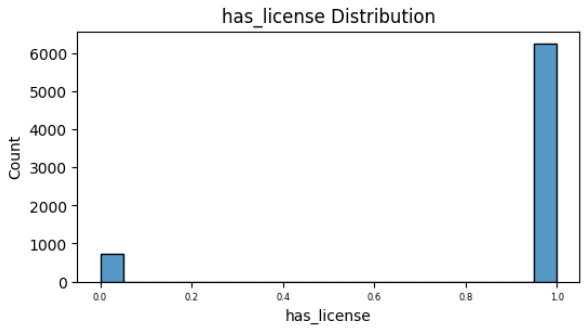

age Distribution
Count
age

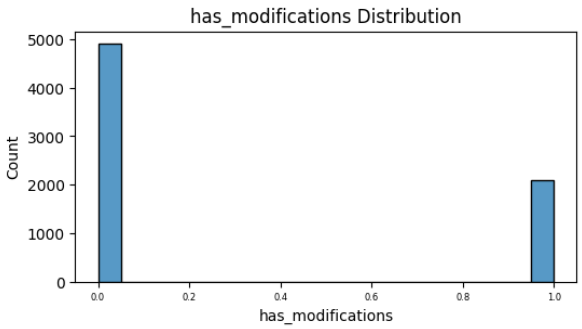

purchase_price Distribution
Count
purchase_price

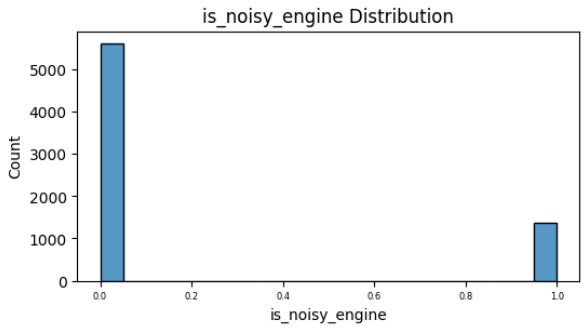

monthly_mileage Distribution
Count
monthly_mileage

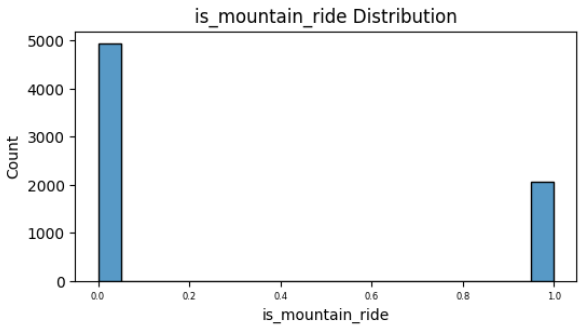

traffic_ticket_count Distribution
Count
traffic_ticket_count

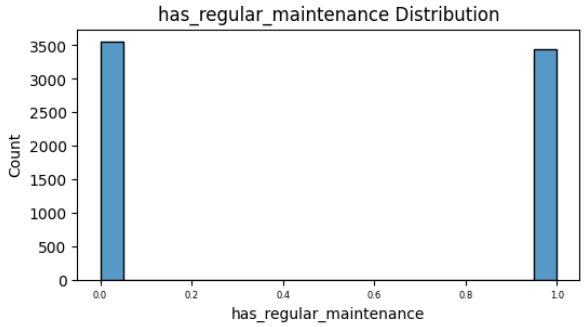

speeding_habbit Distribution
Count
speeding_habbit

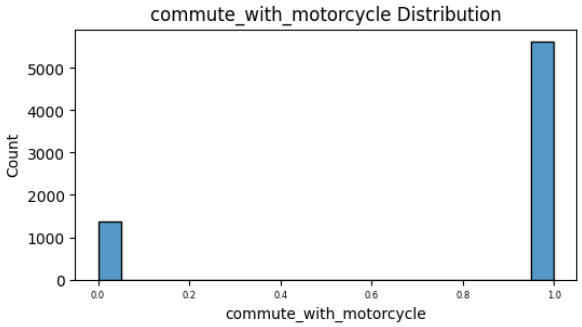












1.3 Absolutely-Right Rules

高風險群(high risk)

1. 第一種(山道猴子):

必要條件

- 男性
- 有駕照
- 排氣量 125 or 150

- 有改裝
- 會跑山

滿足必要條件且滿足下列其中一項

- 有很吵的排氣管
- 購買價錢 > 10k
- 收到罰單次數 ≥ 5
- 機車款式DRG JET FORCE CYGNUS BWS

2. 第二種(在地無照老人三寶):

必要條件

- 年齡 > 50
- 無駕照
- 排氣量 50、100、125
- 安全帽 1 or 2
- 交通違規次數 ≥ 3

3. 第三種(市區飄車仔):

必要條件

- 安全帽 1
- 超速習慣 15km/hr
- 有改裝

滿足所有必要條件且滿足下列其中一個

- 有很吵的排氣管
- 交通罰單 ≥ 3
- 機車款式 Many or Cuxi

中風險群(medium risk)

1. 第一種(普通機車通勤族)

必要條件

- 排氣量 100 or 125
- 安全帽 2
- 使用機車通勤

2. 第二種(車子快拋錨騎很慢的人)

必要條件

- 年齡 > 40
- 沒有定期保養
- 超速習慣 = 0

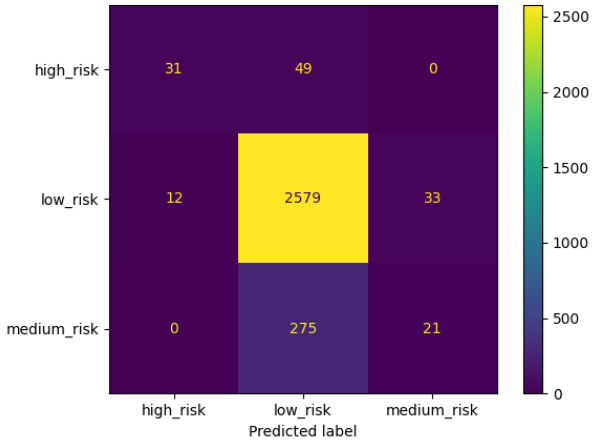
Classification Models

2.1 Decision Tree

資料設定

總共生成了兩組資料，一組有7000筆資料當作訓練資料`dataset1-7000.csv` 和另一組有3000筆資料當作測試資料 `dataset2-3000.csv`

準確率 87.70%

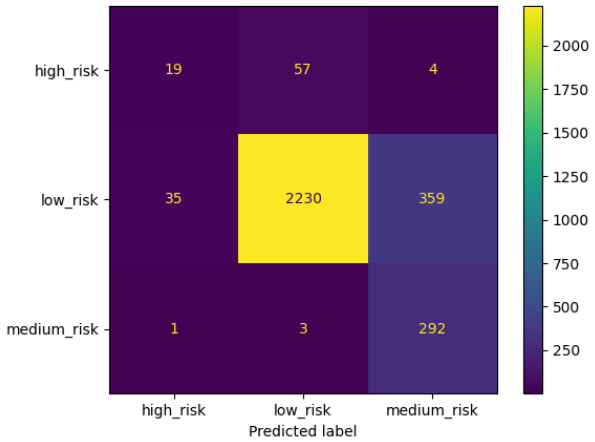


2.2 Other models

總共使用了Decision tree, Naive_bayes, SVM, KNN

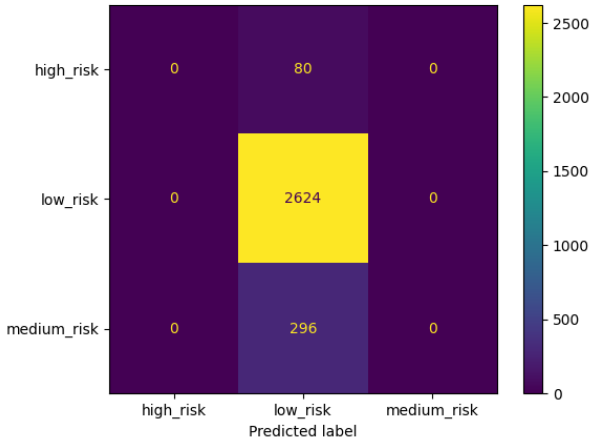
- **Naive Bayes**

  準確率 84.70%



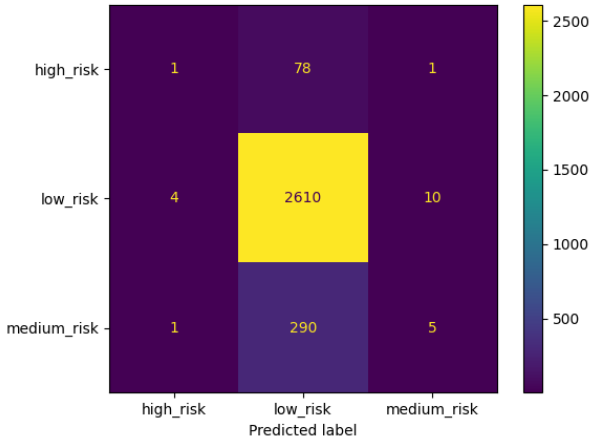
- **SVM**

  準確率 87.47%



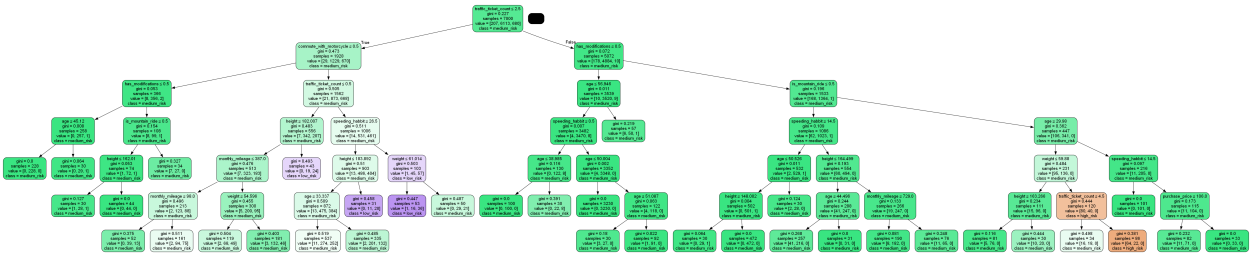
- **KNN**

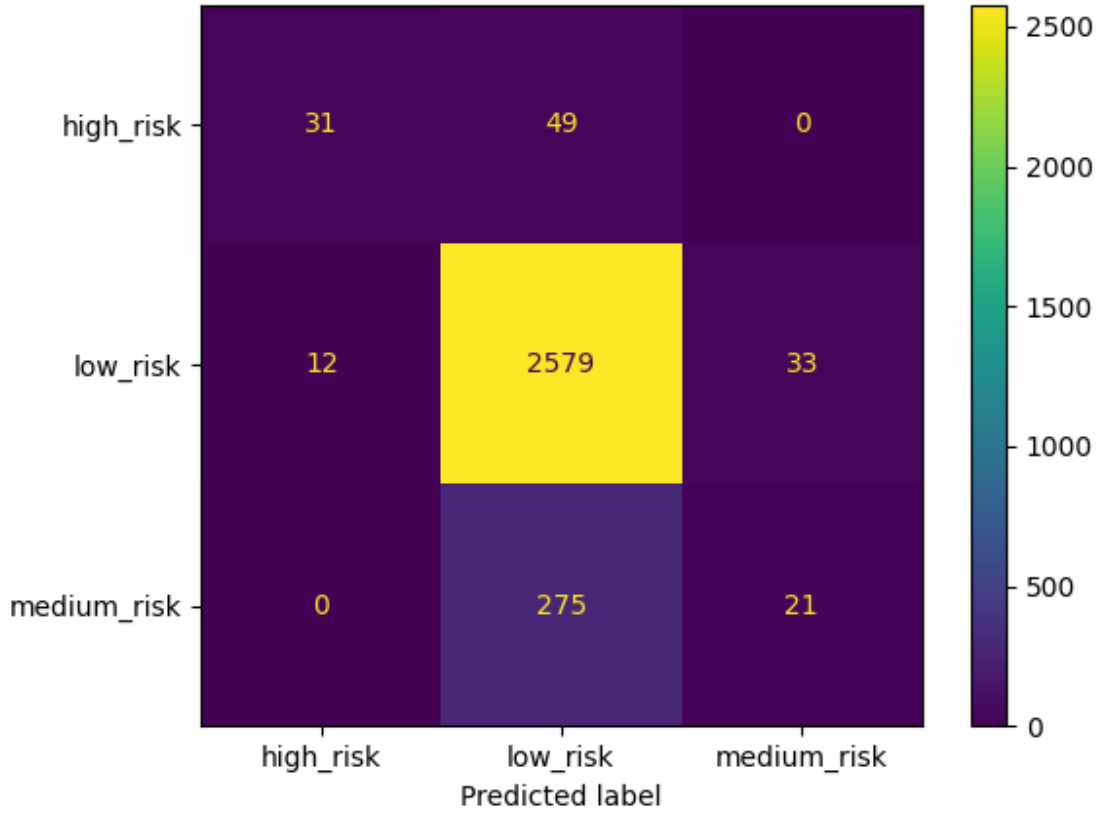
  準確率 87.20%



Analysis

3.1 Decision Trees





在生成decision tree 的過程中，原本因為在判斷high_risk和medium_risk的條件寫得太多，又沒有設定decision tree的max_depth，導致跑出來的結果很雜亂。後來將max_depth 設為7，結果跑出來沒有high_risk的leaf，所以我將一些原本設定的必要條件移動至附加條件上，反覆修改了幾次條件之後，才有比較好的結果。

而結果跑出來也蠻符合我原本想要的預期，就是`high_risk` (橘色)和`low_risk` (紫色)各在tree最一開始分支的地方就分開了，各自在左右邊，而`medium_risk` (綠色)遍布所有的leaf，但是令我比較意外的是low_risk的leaf非常的集中，而從confusion matrix也可以看出low_risk的預測結果相當準確。

3.2 Comparisons

Decision Tree

觀察decision tree所產生出來的結果，可以發現high_risk的決策過程為 `traffic_ticket_count>2.5 → has modified → is_mountain_ride → age < 29.98`，此結果符合我在high_risk的判斷條件的第一條和第三條，但第二個條件比較沒有被顯現出來，我推測原因是原本high_risk的資料數就不多，而第一和第三條的規則又有比較多重複的地方，所以能夠被decision tree 正確預測。

而low_risk大部分的決策過程為 `not commute_with_motorcycle → traffic_ticket_count > 1` 此結果的commute_with_motorcycle符合我當初設定，不需要使用通勤上班的人，騎機車的頻率較低，也不會在尖峰時段在路上騎車，發生車禍的機率就相對低，但有趣的是竟然跑出一個traffic ticket couunt，我想應該是我沒有在這個輸入做太多的設定，導致隨機影響到了結果。

Naive Bayes

因為navie bayes的假設是特徵之間是彼此獨立的，但我在設定特徵的時候，是基於現實情況去做absolutely rule的設定，所以有些條件是會一起出現的，像是收到罰單次數和有改裝的條件，兩者有高度相關，所以Naive Bayes在一些情況下判斷的結果並不好。從confusion matrix來看，可以發現在對low_risk進行預測的時候，常常預測成medium_risk，說明了誤判的可能性較大。

SVM

在做SVM的時候，我是將參數設定成linear，但有趣的是SVM做出來的結果模型完全沒有預測low_risk和high_risk，全部都是預測成medium_risk。合理的解釋是absolutely rule的條件設定裡，很多條件都是重疊到的，對於這種分群不明顯的點分布，SVM無法有效的切出分割線，所以預測結果全部都是同一種。

KNN

KNN所要找的結果是鄰近相似的值，但從Decision Tree很明顯看出來，medium_risk的leaf混和在大部分的條件下，尤其是在medium_risk和low_risk混和的情況更明顯，所以KNN常常將low_medium的值預測成medium_risk，無法利用KNN很難準確地找出鄰近的鄰居來預測新的數件點的類別，

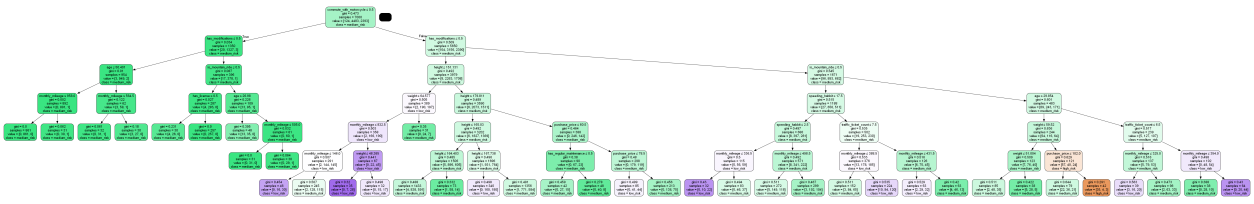
3.3 Discussion

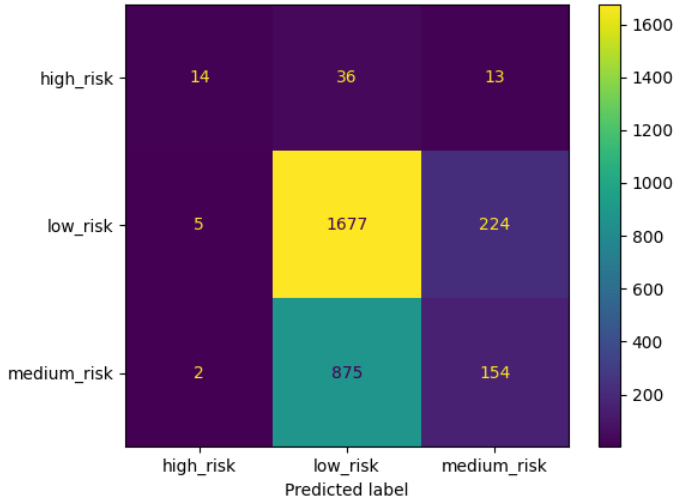
從原本Decision Tree的結果來看，第一個分支的判斷條件是`traffic_ticket_count`，我十分好奇將這個條件從absolutely right的條件裡拿掉會怎樣，於是我就將`traffic_ticket_count`從五個條件裡全部拿掉，判斷結果的準確率大幅的下降

```
========= Decision Tree =========
train data: dataset1-altered-7000.csv
test data: dataset2-altered-3000.csv
Accuracy with Decidion Tree: 61.50%
Success.
========= Naive Bayes =========
train data: dataset1-altered-7000.csv
test data: dataset2-altered-3000.csv
Accuracy with Gaussian Naive Bayes: 54.07%
Success.
============ KNN ===========
train data: dataset1-altered-7000.csv
test data: dataset2-altered-3000.csv
Accuracy on test data with KNN: 56.20%
Success.
============ SVM =============
train data: dataset1-altered-7000.csv
test data: dataset2-altered-3000.csv
Accuracy on test data with SVM: 63.53%
Success.
```

- **Decision Tree**

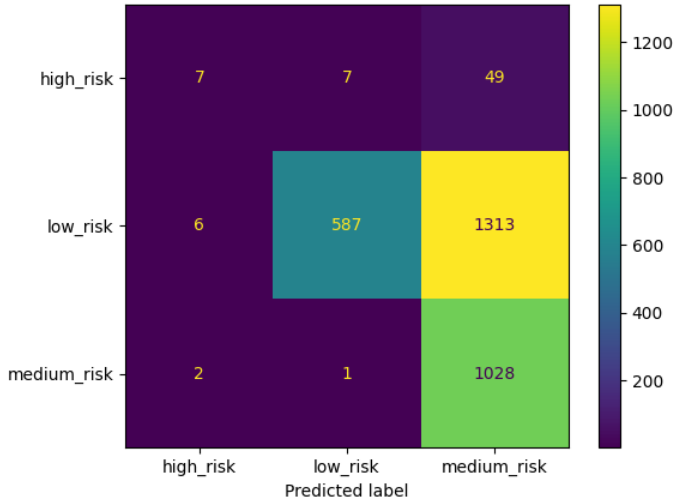
  可以發現分類變得很混亂，low_risk和high_risk常常混在一起，而且也不是在一開始的分支就分開來了



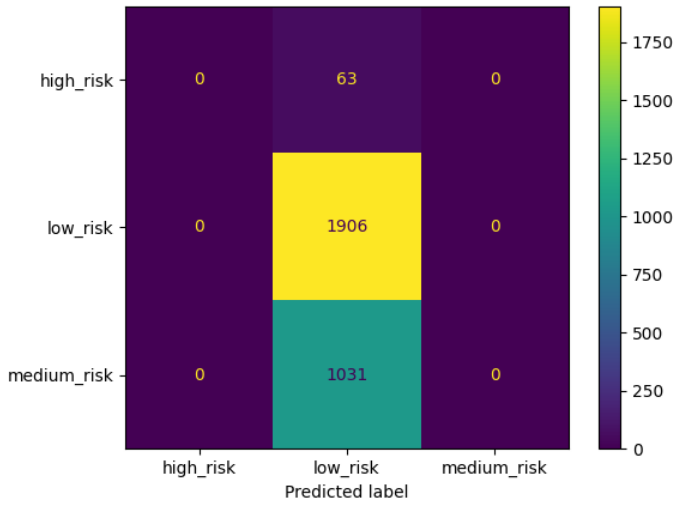


- **Naive Bayes**

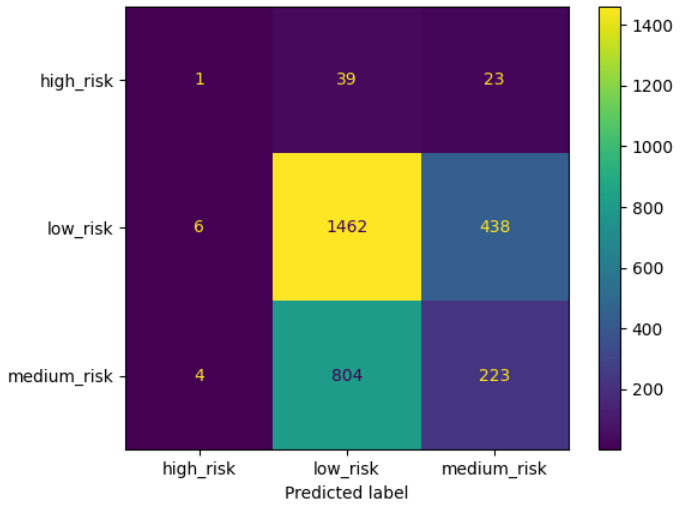
  Naive的誤判狀況更為嚴重



- **SVM**


high_risk
low_risk
medium_risk
0
63
0
0
1906
0
0
1031
0
high_risk
low_risk
medium_risk
Predicted label
1750
1500
1250
1000
750
500
250
0

- **KNN**


high_risk
low_risk
medium_risk
1
39
23
6
1462
438
4
804
223
high_risk
low_risk
medium_risk
Predicted label
1400
1200
1000
800
600
400
200