2022 Data Mining Project 2

P76114511 資訊所一 楊晴雯
Topic: Classification Analysis

I. 資料設計（Data Design）

這是一個動漫資料集之設計，每一筆資料代表一部動漫作品。給定一組離散（類別或數值）或連續數值的屬性資料，該任務為一三分類任務，三個分類為：鉅作（masterpiece）、小品（refreshing）、平淡（bland），欲判斷「該部動漫為三分類的哪一類」。

(I) 特徵/屬性（**Features/Attributes**）

以如下11個屬性輸入作為判斷。

1. 屬性簡介

Discrete, numerical 離散數值型

`STUDIO_STAFF_NUM` ：動畫工作室（Anime Studio）員工數。可能人數為50-300人（整數）。
`SCREENPLAY_WRITER_NUM` ：劇作家人數。可能人數為1-6人（整數）。
`SEASONS` ：季數。可能範圍為1-5季（整數）。

Discrete, numerical 離散數值型：**Boolean** 布林變數

`IS_AIRING` ：是否仍在連載播出。0/1變數。
`IN_SEIYUU_RANK` ：主要角色聲優是否有十年內日本聲優大賞。0/1變數。

Discrete, categorical 離散類別型

`AUTHOR` ：漫畫原作作作者。10位作者（無名字，以數字代替）。
`THEME_SINGER` ：主題曲歌手。7位： `'Lisa', 'Sawano Hiroyuki', 'Hoshino Gen', 'Kenshi Yonezu', 'Mukai Taichi', 'Eve', 'Chico with HoneyWorks'` 。
`STYLE` ：風格分類。7種： `'運動', '戰鬥', '致鬱', '奇幻', '搞笑', '懸疑', '戀愛'` 。

Continuous 連續型

`AVG_RATING` ：平均得分（0-5星）。模擬某個論壇（eg.巴哈姆特）上的評分制度。
`AVG_EP_LENGTH` ：每集平均時長（分）。
`CHARACTER_NUM` ：角色群人數。

2. 資料產生設定
   `STUDIO_STAFF_NUM` ：50-300人（整數）。抽樣分佈為 uniform distribution。
   `SCREENPLAY_WRITER_NUM` ：1-6人（整數）。抽樣分佈為 uniform distribution。
   `SEASONS` ：1-5季（整數）。抽樣分佈為 uniform distribution。
   `IS_AIRING` ：0/1變數，抽樣分佈為 uniform distribution，即機率各佔一半。
   `IN_SEIYUU_RANK` ：同上。
   `AUTHOR` ：抽樣分佈為 uniform distribution。
   `THEME_SINGER` ：抽樣機率為事先設定的機率值 `[0.3, 0.3, 0.1, 0.1, 0.1, 0.05, 0.05]` 。其意義為 Lisa 和 Sawano Hiroyuki 被抽選的機率各為0.3，Hoshino Gen 等人的機率為0.1，Eve 和 Chico With Honeyworks 的機率則為0.05。
   `STYLE` ：抽選機率為 `[0.3, 0.3, 0.1, 0.1, 0.1, 0.05, 0.05]` 。同上類推。
   `AVG_RATING` ：抽樣自 $N(3.5,0.5^2)$，並clip範圍於0-5之間。
   `AVG_EP_LENGTH` ：抽樣自 $N(25,5^2)$。
   `CHARACTER_NUM` ：抽樣自 $N(30,5^2)$，隨後再取floor。

註：分類時無法以字串形式輸入，因此會將類別編碼為數字後才放入分類模型內。順序如上所示，第一個元素編號為0，第二個元素為1，以此類推。

3. 資料噪音

(II) 規則（**Rules**）

基於幾項事實如
(1) Lisa 配樂的戰鬥番（鬼滅之刃、FateZero）都滿有名。
(2) 在 Sawano Hiroyuki 的神曲加持後，戰鬥番都會變特別好看。
(3) 排球少年、黑子籃球等運動番等都頗負盛名。
定義鉅作（masterpiece）為

> `IN_SEIYUU_RANK` = 1（True） and `STUDIO_STAFF_NUM` >= 120，且滿足(a), (b)中至少一個條件：
> (a)
> `AVG_RATING` > 4.0 AND `CHARACTER_NUM` > 35 AND `SCREENPLAY_WRITER_NUM` > 3 AND `SEASONS` > 2
> (b)
> `THEME_SINGER` = ‘Lisa’ or ‘Sawano Hiroyuki’ AND
> `STYLE` = ‘運動’ OR ‘戰鬥’ AND
> `AVG_RATING` > 3.8

定義小品（refreshing）作品為

> 不符合鉅作規定者中，
> `IN_SEIYUU_RANK` = True, `STUDIO_STAFF_NUM` >= 50，且滿足(a), (b)中至少一個條件：
> (a)
> `AVG_RATING` > 3.5 AND `CHARACTER_NUM` > 20 AND `SEASONS` > 1 AND (`AUTHOR` = 1 OR 2)
> (b)
> `THEME_SINGER` 為 ‘Hoshino Gen’, ‘Kenshi Yonezu’, ‘Mukai Taichi’, ‘Chico with HoneyWorks’ 其中之一 AND
> `STYLE` 為 ‘致鬱’, ‘奇幻’, ‘搞笑’, ‘懸疑’, ‘戀愛’ 其中之一 AND
> `AVG_RATING` > 3.8 AND
> `AVG_EP_LENGTH` > 23

3. 資料集數據
   共產出兩組資料，其中
   `anime_dataset_10000-0.csv` 包含10,000筆資料，完全按照規則分類，
   `anime_dataset_10000-0.05.csv` 則是另產生一包含10,000筆的資料檔案，並隨機修改(tweak) 5% 的資料類別
   （屬性欄位不變，將類別隨機改成規則類別之外的其中一個類別），總共修改500筆。

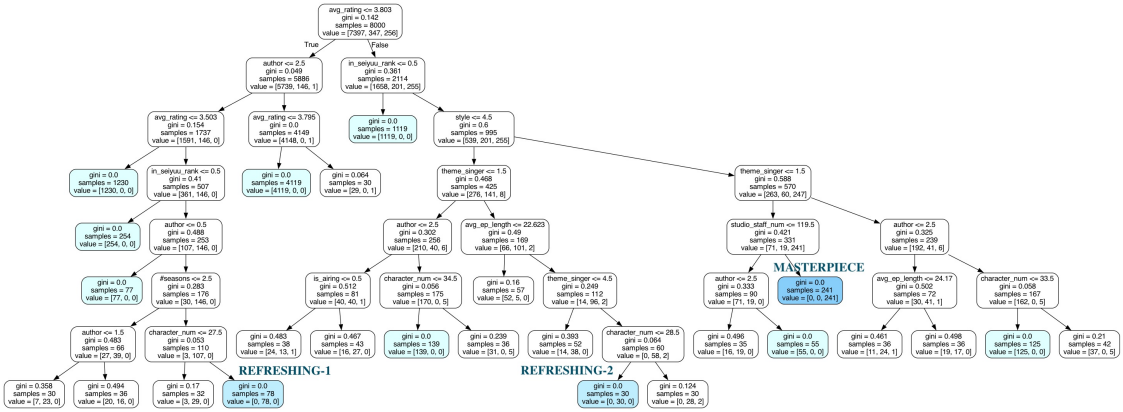
Dataset	Masterpiece	Refreshing	Bland
Absolute Rules (0)	328	446	9,226
Tweaked (0.05)	559	652	8,789

II. 分類模型分析

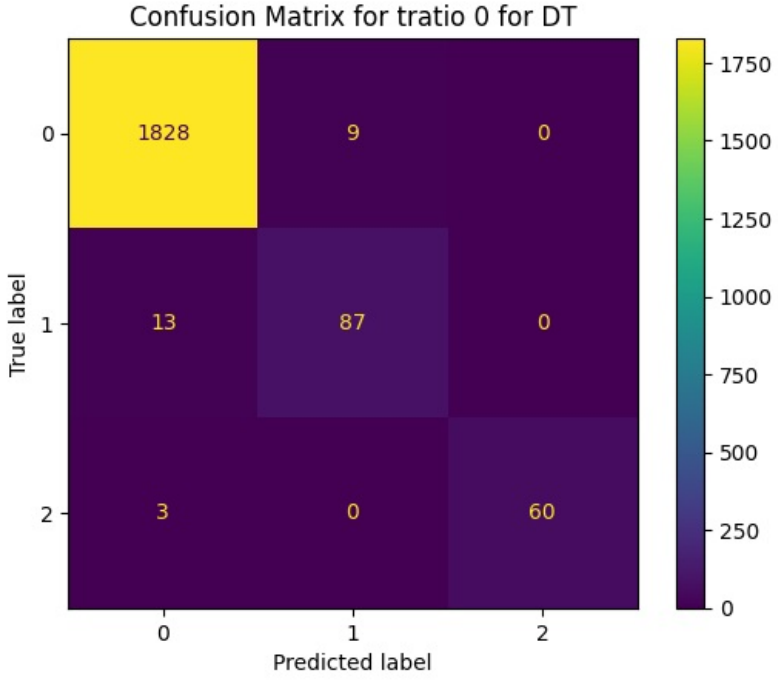
(I) 測試集正確率（**Testing Accuracy**）

tweak_ratio\model	Decision Tree	Naive Bayes Gaussian NB
0	0.988	0.743
0.05	0.935	0.888

(II) 絕對規則資料集（**Absolute-Rule Dataset**）

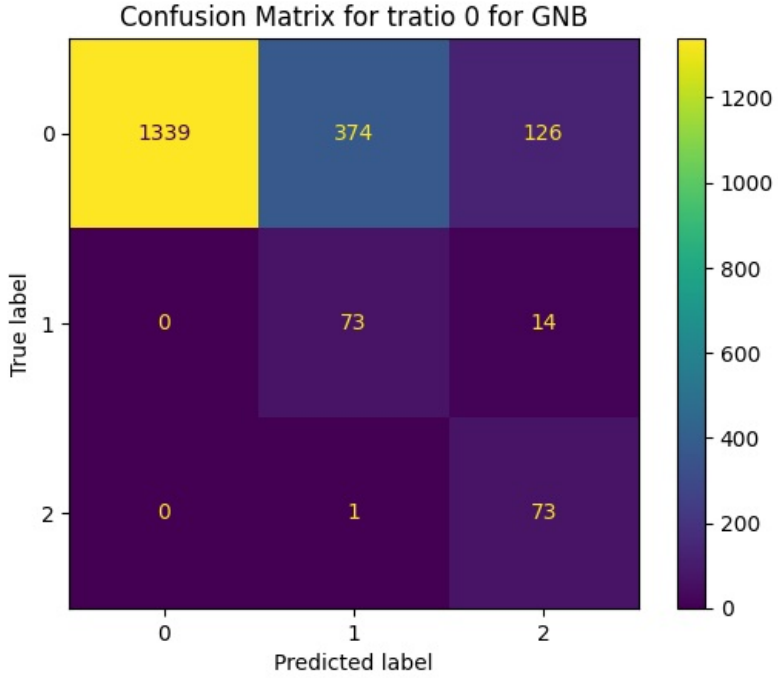


1. 決策樹（Decision Tree）
   設定決策樹超參數 `MIN_SAMEPLES_LEAF` = 30 與 `MAX_DEPTH` = 7，最深藍色者為鉅作，較淺者為小品，最淺者為普通，mixed nodes則為白底。
   其中241筆的鉅作動漫資料落在
   AVG_RATING > 3.8 → IN_SEIYUU_RANK = True → THEME_SINGER 為 Sawano 和 Lisa → STUDIO_STAFF_NUM > 120 的決策樹路徑上。可以看出這部分的動漫資料應該都是符合鉅作底下之條件(b)，因其中的主題曲歌手（Theme Singer）的機率經特地設計，該兩位歌手出現的機率也大幅提升，因此條件(b)發生機率較(a)高。該路徑也抓到了鉅作的必要條件：IN_SEIYUU_RANK = True （第二層） and STUDIO_STAFF_NUM >= 120 （第五層），而第一層的node之屬性為AVG_RATING，從資料設計規則來看也可以得到合理解釋：因從普通作品中要區別出鉅作和小品，AVG_RATING高於3.5是必要條件。
   而小品動漫資料在這棵決策樹中則落在兩個nodes上。
   'REFRESHING-1'點共78筆資料，其路徑上的幾個attributes （不按照路徑順序）為：3.5 <= AVG_RATING < 3.8, IN_SEIYUU_RANK = True, #SEASONS > 3，AUTHOR編號0或1, CHARACTER_NUM >= 27。
   這和資料規則中小品的條件(a)大致相同，唯#SEASONS和CHARACTER_NUM的要求比規則設計時更為嚴格一些。
   'REFRESHING-2'點則有30筆資料，其路徑上的幾個attributes為 AVG_RATING > 3.8, IN_SEIYUU_RANK = True, STYLE < 4.5 （意即類別屬於'搞笑'、'戀愛'、'致鬱'、'懸疑'、'奇幻'），AVG_EP_LENGTH > 22.6 CHARACTER_NUM <= 28.5。這和條件(b)大致相同，唯THEME_SINGER的條件未能夠列入該樹前7層。



2. 貝氏分類器（Naive Bayes）
   Naive Bayes 的前提假設之一為屬性間互相獨立，屬性中如有連續變數則假設其服從高斯分佈（這點和我資料設計的做法相同）。
   可以看到因為資料設計規則都是多個屬性要同時符合某項條件才能被劃分為某一類，因此屬性間的關聯性高，故 Naive Bayes 表現較差。從 confusion matrix 可以看出其常常將bland誤判為

refreshing。代表 refreshing 類別與 bland 類別的決策邊界對NB來說非常難以fit。
註：confusion matrix 中的類別0, 1, 2分別對應bland, refreshing和masterpiece。

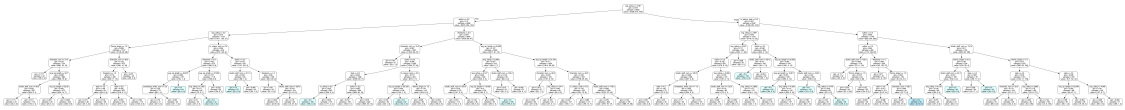


3. 特徵分析（Feature Analysis）
   我們使用 permutation importance 來評估哪些特徵對分類成效有顯著影響。一個特徵的 permutation importance (score) 定義為「當任意shuffle該特徵時，模型分數衰減的程度」（the decrease in a model score when a single feature is randomly shuffled）。故其值越大代表該特徵越顯著。以下挑出$R^2>0$的特徵並列出其 平均 ± 標準差 的形式。
   Decision Tree中以AVG_RATING影響最大，其次是IN_SEIYUU_RANK，STUDIO_STAFF_NUM需要超過某一閾值雖然也是必要的條件，但可能是因為該條件比較容易達成，因此該特徵的 permutation importance 計算起來反而不高。而Naive Bayes 中也是以這兩個特徵最為顯著（雖然順序調換）。

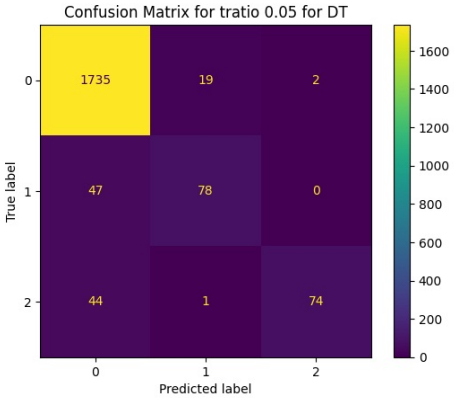
```
═══════ Decision Tree ═══════
Reading anime_dataset_10000-0.csv ...
class counts:bland          9226
refreshing      446
masterpiece     328
Name: class, dtype: int64
Score: 0.988
avg_rating      0.094 +/- 0.005
in_seiyuu_rank  0.077 +/- 0.004
author          0.044 +/- 0.003
style           0.036 +/- 0.003
theme_singer    0.034 +/- 0.003
studio_staff_num        0.016 +/- 0.002
#seasons        0.009 +/- 0.001
avg_ep_length   0.007 +/- 0.001
 ═══════ Naive Bayes ═══════
 Reading anime_dataset_10000-0.csv ...
 class counts:bland          9226
 refreshing      446
 masterpiece     328
 Name: class, dtype: int64
 Score: 0.736
 Macro F1: 0.544
 in_seiyuu_rank  0.080 +/- 0.007
 avg_rating      0.042 +/- 0.006
 theme_singer    0.019 +/- 0.004
 style           0.017 +/- 0.004
 studio_staff_num        0.005 +/- 0.002
```

(III) 修改資料集（**Tweaked Dataset**）

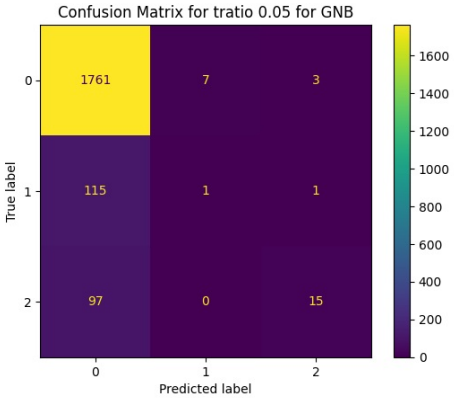
1. 決策樹（Decision Tree）
   加入了噪音後的資料集對Decision Tree造成4%左右的退步幅度。畢竟絕對規則如果被打破後對Decision Tree來說就會造成混淆，使得某些絕對規則可能不再適用。但因為 `tweak_ratio` 設定的不大，該模型的accuracy仍然維持得頗高。 至於樹的形狀則分枝非常嚴重，基本上permutation importance 為正的特徵都參與了這幾組分枝，使得樹的 leaf nodes 大概只有一半屬於 pure class，純屬於鉅作系列的node在 `MAX_DEPTH = 7` 的情況甚至還無法被區分出來。



（check `tree_0.05.png` for clearer image）



2. 貝氏分類器（Naive Bayes）
   神奇的是在混淆後，貝氏分類器的正確率竟然大幅上升至88%，然而只要細看 confusion matrix 就會發現該分類器將所有的資料幾乎都預測為bland（第0類）。故此時光看accuracy這個metric並不準確。改採macro f1可能是較好的衡量方法。



3. 觀察
   從下表可以看出在混淆部分資料後決策樹還能夠維持高達 0.77 的 macro f1（類別f1的平均），而NB則滑落至 0.395。因此使用了人為規則設計後的這份資料顯然比較適合以Decision Tree做預測。

```
========== Decision Tree ==========
Reading anime_dataset_10000-0.05.csv ...
class counts:bland          8789
refreshing      652
masterpiece     559
Name: class, dtype: int64
Macro F1: 0.770
Score: 0.935
avg_rating      0.088 +/- 0.005
in_seiyuu_rank  0.066 +/- 0.004
author          0.042 +/- 0.004
theme_singer    0.038 +/- 0.003
style           0.032 +/- 0.003
studio_staff_num        0.013 +/- 0.002
#seasons        0.008 +/- 0.001
avg_ep_length   0.005 +/- 0.001
========== Naive Bayes ==========
Reading anime_dataset_10000-0.05.csv ...
class counts:bland          8789
refreshing      652
masterpiece     559
Name: class, dtype: int64
Score: 0.888
Macro F1: 0.395
avg_rating      0.009 +/- 0.003
in_seiyuu_rank  0.007 +/- 0.002
studio_staff_num        0.004 +/- 0.001
```