Data Mining Project2

姓名 :電機所 卓冠廷
學號 : N26114277

一、 目的

本作業利用四個不同的分類器(Dicision Tree、Random Forest、KNN、K-means)做分類，並針對各個模型預測的結果作分析。此次預測的題目為「此作業是否為好作業?」。

二、 Input 設計

Attributes: 分四大類(實作的收穫、時間成本、感受、學生能力)

所有 Attributes 都寫成 dictionary 的形式，values 在-2 到 2 之間

- 實作的收穫
  - ✓ 'Reward from homework' ：做功課的收穫
  - ✓ 'Course relevance' ：課堂關聯性
  - ✓ 'Interdisciplinary learning' ：跨領域學習
  - ✓ 'Cooperation and Discussion' ：團隊合作與討論

以上四項屬性的 values 會相加算出 **reward** 值

- 時間成本
  - ✓ 'Homework numbers' ：功課數量
  - ✓ 'Scale' ：作業規模
  - ✓ 'Average time spent' ：作業平均花費時間
  - ✓ 'Homework deadlines' ：作業繳交期限

以上四項屬性的 values 會相加算出 **time_cost** 值

- 感受
  - ✓ 'Step by step' ：作業難度是否循序漸進
  - ✓ 'Feeling' ：做作業時的感受
  - ✓ 'Difficulty of implementation' ：作業實作的難度
  - ✓ 'Difficulty in understanding' ：作業理解的難度

以上四項屬性的 values 會相加算出 **feel** 值

- 學生能力
  - ✓ 'Student qualifications' ：學生的資質
  - ✓ 'Student concentration' ：學生的專注程度

- ✓ 'Student responsibility' ：學生的責任感
- ✓ 'Research relevance' ：學生的研究相關性

以上四項屬性的 values 會相加算出 **ability** 值

Redundant Attributes:

所有 Redundant Attribute 都寫成 dictionary 的形式，values 在-2 到 2 之間

- ✓ 'Student weight' ：學生的體重
- ✓ 'Student height' ：學生的身高

Absolutely Right Rules: (五個規則)

reward*2 (≤ -8：低 -8 到 8：中 ≥8：高)
time_cost、feel、ability (≤ -4：低 -4 到 4：中 ≥4：高)
這邊將 reward 乘上兩倍是因為個人認為實作的收穫相對於其他三者，在判斷是否為好作業時較為重要，因此將此分數的權重調高。

Rule1 = reward*2 ≥ 8 and time_cost ≤ -6
(時間成本低，收穫高)
Rule2 = reward*2 ≥ 8 and feel ≥ 6
(感受好，收穫高)
Rule3 = reward* ≥ 8 and ability ≥ 6
(學生能力高，收穫高)
Rule4 = reward*2 ≥ 8 and 4 ≥ time_cost ≥ 0 and 6 ≥ feel ≥ 2 and 6 ≥ ability ≥ 2
(時間成本適中，感受適中偏好，學生能力適中偏高，收穫高)
Rule5 = reward*2 ≥ 8 and time_cost ≥ 4 and 6 ≥ feel ≥ 2 and -2 ≥ ability ≥ -6
(時間成本高，感受適中偏好，學生能力適中偏低，收穫高)

資料量:總共生成 10000 筆資料

Rule1:1000 筆
Rule2:1000 筆
Rule3:1000 筆
Rule4:1000 筆
Rule5:1000 筆
隨機產生的資料:5000 筆

可以發現 Absolutely Right Rules 的總數大於 Good homeworks，是因為同一筆資料可能同時滿足多個 Rules，而 Bad homeworks 的資料數略小於 5000 筆，是因為隨機產生的資料(Bad homeworks)可能滿足 Rule1 到 Rule5，產

生結果如下:

```
(DM_hw2) C:\Users\poetr\OneDrive\桌面\hw2>python data_generator1.py
Rule1 numbers:  1060
Rule2 numbers:  1179
Rule3 numbers:  1110
Rule4 numbers:  1200
Rule5 numbers:  1026
Good homeworks numbers:  5001
Bad homeworks numbers:  4999
```

原始資料(data1)

加入干擾:

- 加入 Redundant Attributes 產生資料(data2)

```
(DM_hw2) C:\Users\poetr\OneDrive\桌面\hw2>python data_generator2.py
Rule1 numbers:  1050
Rule2 numbers:  1180
Rule3 numbers:  1100
Rule4 numbers:  1173
Rule5 numbers:  1021
Good homeworks numbers:  5000
Bad homeworks numbers:  5000
```

- 加入 Redundant Attributes 並將後 500 筆的 Labels 故意標錯(data3)
  (訓練出的模型去預測 data2)

```
(DM_hw2) C:\Users\poetr\OneDrive\桌面\hw2>python data_generator3.py
Rule1 numbers:  1066
Rule2 numbers:  1166
Rule3 numbers:  1095
Rule4 numbers:  1186
Rule5 numbers:  1026
Good homeworks numbers:  5500
Bad homeworks numbers:  4500
```

- 加入 Redundant Attributes 並放寬 Rules，將實作的收穫(reward)從高(reward ≥ 8)改為適中偏高(6 ≥ reward ≥ 2)，亦即調低 reward 的重要程度

```
(DM_hw2) C:\Users\poetr\OneDrive\桌面\hw2>python data_generator4.py
Rule1 numbers:  1100
Rule2 numbers:  1244
Rule3 numbers:  1147
Rule4 numbers:  1262
Rule5 numbers:  1037
Good homeworks numbers:  5191
Bad homeworks numbers:  4809
```

從上圖可見隨機產生的資料(Bad homeworks)有較多被轉換成 Good homeworks，這是因為每條 Rules 的標準降低所導致

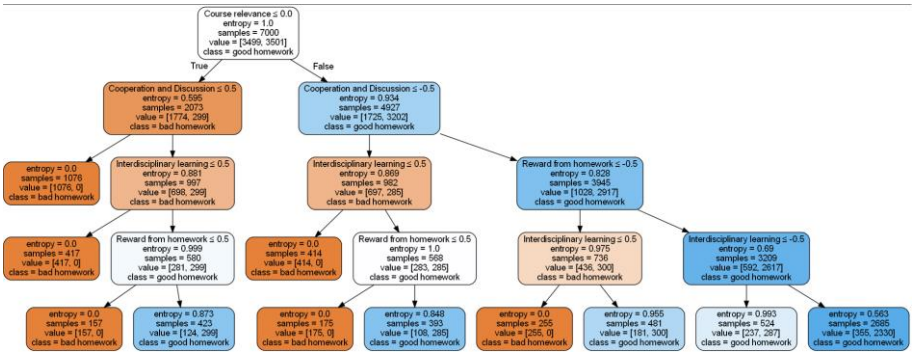
訓練集/測試集:

兩者的比例採 7:3 去做分割，以下為 data1/data2/data3/data4 的分割結果:

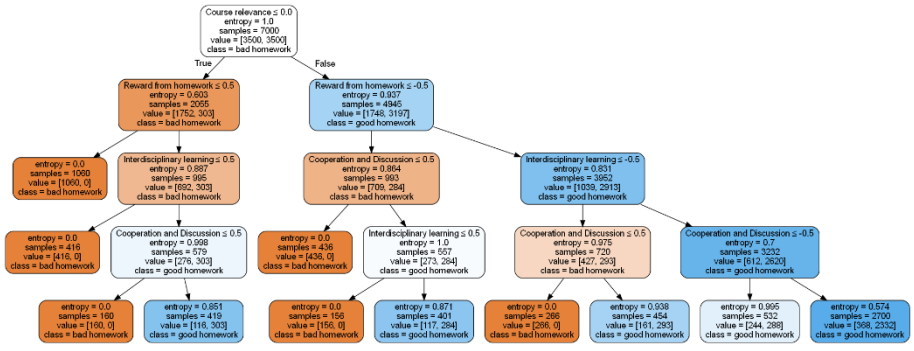
```
Shape of data1 x_train: (7000, 16)
Shape of data1 x_vad: (3000, 16)
Shape of data1 y_train: (7000,)
Shape of data1 y_vad: (3000,)
Shape of data2 x_train: (7000, 18)
Shape of data2 x_vad: (3000, 18)
Shape of data2 y_train: (7000,)
Shape of data2 y_vad: (3000,)
Shape of data3 x_train: (7000, 18)
Shape of data3 x_vad: (3000, 18)
Shape of data3 y_train: (7000,)
Shape of data3 y_vad: (3000,)
Shape of data4 x_train: (7000, 18)
Shape of data4 x_vad: (3000, 18)
Shape of data4 y_train: (7000,)
Shape of data4 y_vad: (3000,)
```

可以發現 data2/data3/data4 的維度多 2 維，因為 Redundant Attributes 的加入

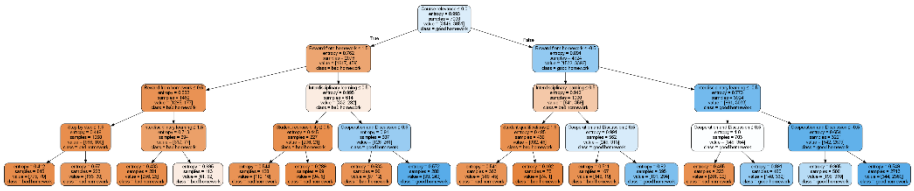
三、 Decision Tree 分析



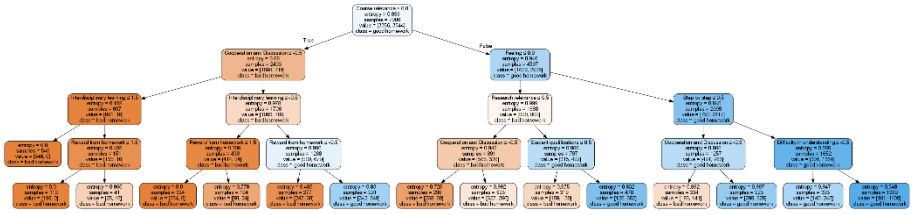
decision_tree_data1_ max_depth=4.png(清晰圖在 hw2 內)



decision_tree_data2_ max_depth=4.png(清晰圖在 hw2 內)



decision_tree_data3_ max_depth=4.png(清晰圖在 hw2 內)

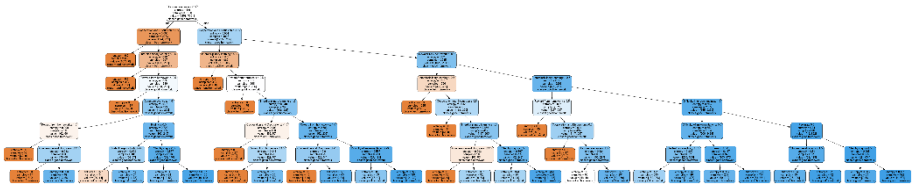


decision_tree_data4_ max_depth=4.png(清晰圖在 hw2 內)

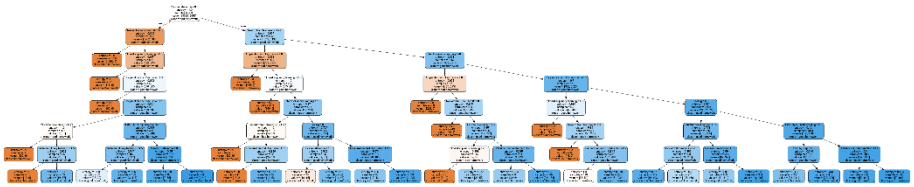
```
score of decision_tree_data1 model is :  0.8556666666666667
score of decision_tree_data2 model is :  0.8526666666666667
score of decision_tree_data3 model is :  0.833
score of decision_tree_data4 model is :  0.756
```

data1/ data2/ data3/data4 辨識準確度

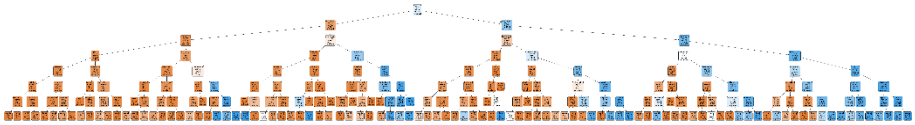
在 max_depth=4 的情況下，data1 與 data2 影響好壞功課的 Attributes 皆出自於實作的收穫(reward)，而 data3 與 data4 的 Decision Tree 因為有錯誤的 Labels 和放寬 Rules 的影響，考量了一些實作的收穫(reward)外的 Attributes，像是 Step by step、Student qualifications、Student responsibility。



decision_tree_data1_ max_depth=7.png(清晰圖在 hw2 內)



decision_tree_data2_ max_depth=7.png(清晰圖在 hw2 內)



decision_tree_data3_ max_depth=7.png(清晰圖在 hw2 內)

decision_tree_data4_ max_depth=7.png(清晰圖在 hw2 內)

```
score of decision_tree_data1 model is :  0.9026666666666666
score of decision_tree_data2 model is :  0.9023333333333333
score of decision_tree_data3 model is :  0.8743333333333333
score of decision_tree_data4 model is :  0.8063333333333333
```

data1/ data2/ data3/data4 辨識準確度

在 max_depth=7 的情況下，影響好壞功課的 Attributes 並非完全屬於實作的收穫(reward)，從 data1 與 data2 來看，實作的收穫(reward)考慮了 4 項

Attributes，感受(feel)考慮了 3 項 Attributes，學生能力(ability)考慮了 2 項 Attributes，時間成本(time_cost)考慮了 1 項 Attributes，沒有考慮任何 Redundant Attributes，而 data3 和 data4 的 Decision Treeur 皆考慮了所有的 Attributes(16+2)。

深度對 Decision Tree 的影響:

比較兩種深度的結果跟 Absolutely Right Rules 的關聯，前四層出現的 Attributes 大部分都屬於實作的收穫(reward)，因為 5 條 Rules 皆跟 reward 相關，且權重還乘 2 倍；感受(feel)在第 5 層後開始出現，因為 3 條 Rules 跟 feel 相關，且皆希望感受要好或偏好；學生能力(ability)在第 6 層後開始出現，因為 3 條 Rules 跟 ability 相關，分別是學生能力偏低、偏高、高；時間成本(time_cost)在第 7 層才出現，因為 3 條 Rules 跟 time_cost 相關，分別是時間成本低、適中、高，範圍最廣，因此在深度比較深的位置才會影響決策，深度較淺的決策則受到權重較重或範圍較窄的 Attributes 影響，而隨深度的上升，準確度也會上升。

Redundant Attributes 對 Decision Tree 的影響:

比較兩種深度 data3 的 Decision Tree，會發現深度較淺的情況下，雖有錯誤的 Labels，但不會把 Redundant Attributes 考慮進去，僅考量到較重要的 Attributes 去做決策，而隨著深度的提升，Decision Tree 必須考量到較次要的 Attributes，此時 Redundant Attributes 自然有機會浮現，準確率也會下降較多(max_depth=4 下降約 2%，max_depth=7 下降約 3%)

放寬 Rules 對 Decision Tree 的影響:

比較 data4 和 data2 的 Decision Tree，會發現在不同深淺的情況下，data4 考慮的 Attributes 皆比 data2 來的多，放寬 Rules，亦即將實作的收穫(reward)的 Attributes 影響調降，其他的 Attributes 被納入考量，而當有較多 Attributes 需考量時，準確率自然跟著下降。

四、 Random Forest 分析

此模型是基於 Decision Tree 去開發的，由多棵 Decision Trees 去做多數投票機制進行預測，優點在於每棵樹會用到哪些訓練資料及特徵都是隨機決定，且每一棵樹都是獨立的。這邊設定樹的數目為 100 棵，深度一樣以 4 和 7 去做討論

	precision	recall	f1-score	support
Bad homeworks	0.99	0.81	0.89	1500
Good homeworks	0.84	0.99	0.91	1500
accuracy			0.90	3000
macro avg	0.91	0.90	0.90	3000
weighted avg	0.91	0.90	0.90	3000

data1_ classification_report_max_depth=4

	precision	recall	f1-score	support
Bad homeworks	0.96	0.82	0.89	1500
Good homeworks	0.84	0.97	0.90	1500
accuracy			0.90	3000
macro avg	0.90	0.90	0.89	3000
weighted avg	0.90	0.90	0.89	3000

data2_ classification_report_max_depth=4

	precision	recall	f1-score	support
Bad homeworks	0.99	0.77	0.86	1500
Good homeworks	0.81	0.99	0.89	1500
accuracy			0.88	3000
macro avg	0.90	0.88	0.88	3000
weighted avg	0.90	0.88	0.88	3000

data3_ classification_report_max_depth=4

	precision	recall	f1-score	support
Bad homeworks	0.89	0.72	0.79	1453
Good homeworks	0.78	0.92	0.84	1547
accuracy			0.82	3000
macro avg	0.83	0.82	0.82	3000
weighted avg	0.83	0.82	0.82	3000

data4_ classification_report_max_depth=4

考慮到不同類別的樣本不均衡，這邊看的是 weighted avg 做比較

data1 和 data2 的結果比較

發現 precision 的部分 data1 高了 1%，說明 Redundant Attributes 有稍微影響到 precision，跟 Decision Tree 不一樣，因為 Random forest 的訓練資料及特徵是隨機決定，因此在投票表決階段受較多 Attributes 影響，recall 的部分一樣，表示為真的情況下，正確判斷出來的把握度差不多。

data2 和 data3 的結果比較

發現 recall 的部分 data3 低了 2%，表示錯誤的 Labels 使為真的情況下，正確判斷的把握度下降，但因 Random forest 是經每個 Decision Tree 投票表決，precision 並未因此受到影響。

data2 和 data4 的結果比較

發現 precision 和 recall 分別降低了 7% 和 8%，放寬 Rules 代表單一 Decision Tree 參考的 Attributes 提升，但因為深度太淺分不太出來，使精準度下降，且表示為真的情況下，正確判斷的把握度也下降。

	precision	recall	f1-score	support
Bad homeworks	1.00	0.85	0.92	1500
Good homeworks	0.87	1.00	0.93	1500
accuracy			0.92	3000
macro avg	0.93	0.92	0.92	3000
weighted avg	0.93	0.92	0.92	3000

data1_ classification_report_max_depth=7

	precision	recall	f1-score	support
Bad homeworks	1.00	0.85	0.92	1500
Good homeworks	0.87	1.00	0.93	1500
accuracy			0.92	3000
macro avg	0.93	0.92	0.92	3000
weighted avg	0.93	0.92	0.92	3000

data2_ classification_report_max_depth=7

	precision	recall	f1-score	support
Bad homeworks	1.00	0.83	0.90	1500
Good homeworks	0.85	1.00	0.92	1500
accuracy			0.91	3000
macro avg	0.93	0.91	0.91	3000
weighted avg	0.93	0.91	0.91	3000

data3_ classification_report_max_depth=7

	precision	recall	f1-score	support
Bad homeworks	0.94	0.80	0.86	1453
Good homeworks	0.83	0.95	0.89	1547
accuracy			0.88	3000
macro avg	0.89	0.87	0.88	3000
weighted avg	0.89	0.88	0.88	3000

data4_ classification_report_max_depth=7

考慮到不同類別的樣本不均衡，這邊看的是 weighted avg 做比較

data1 和 data2 的結果比較

發現 precision 和 recall 都一樣，說明 Redundant Attributes 在深度較高的情況下對精確度和召回率影響不大，可能是深度高時考量的 Attributes 本身就較多，Redundant Attributes 的影響被淡化導致。

data2 和 data3 的結果比較

與淺層結果差不多，故不多做討論。

data2 和 data4 的結果比較

與淺層相比，precision 和 recall 下降程度較小，單一 Decision Tree 參考的 Attributes 雖提升，但因為深度提升，分類效果較好，使得 precision 和 recall 下降程度減少。

深度對分類結果的影響:

由以上結果可發現，隨深度提高，不同 data 的 precision 和 recall 皆隨之提高，模型對不同干擾的抵抗能力也都有明顯的提升，錯誤的 Labels 對 recall 的影響較高，放寬 Rules 則會對同時對 precision 和 recall 造成影響。

五、 K Nearest Neighbor 分析

此模型會在訓練資料集中尋找 k 個與 Input 向量 x 最近的向量的集合，然後把 x 的類別歸類為這 k 個樣本中類別數最多的那一類，也因為最後有類似投票的環節產生，為了避免平手的狀況，二元分類時，k

通常會選奇數個

	precision	recall	f1-score	support
Bad homeworks	0.98	0.83	0.90	1500
Good homeworks	0.85	0.98	0.91	1500
accuracy			0.90	3000
macro avg	0.91	0.90	0.90	3000
weighted avg	0.91	0.90	0.90	3000

data1_ classification_report_ n_neighbors = 3

	precision	recall	f1-score	support
Bad homeworks	0.97	0.81	0.88	1500
Good homeworks	0.83	0.98	0.90	1500
accuracy			0.89	3000
macro avg	0.90	0.89	0.89	3000
weighted avg	0.90	0.89	0.89	3000

data2_ classification_report_ n_neighbors = 3

	precision	recall	f1-score	support
Bad homeworks	0.98	0.76	0.86	1500
Good homeworks	0.80	0.98	0.88	1500
accuracy			0.87	3000
macro avg	0.89	0.87	0.87	3000
weighted avg	0.89	0.87	0.87	3000

data3_ classification_report_ n_neighbors = 3

	precision	recall	f1-score	support
Bad homeworks	0.93	0.77	0.84	1453
Good homeworks	0.81	0.94	0.87	1547
accuracy			0.86	3000
macro avg	0.87	0.85	0.86	3000
weighted avg	0.87	0.86	0.86	3000

data4_ classification_report_ n_neighbors = 3

考慮到不同類別的樣本不均衡，這邊看的是 weighted avg 做比較

data1 和 data2 的結果比較

發現 precision 和 recall 的部分 data1 高了 1%，說明 Redundant Attributes 對原始資料產生了干擾，這跟 k 值太小有關，只要有兩個 neighbors 選某個 Labels 就會被分類成此 Labels。

data2 和 data3 的結果比較

錯誤的 Labels 使 precision 和 recall 皆下降，尤其在 recall 的部分下降比較多，因此可推論錯誤的 Labels 對於 recall 的影響較大。

data2 和 data4 的結果比較

發現 precision 和 recall 都降低了 3%，表示放寬 Rules 會同時對 precision 和 recall 造成影響。

	precision	recall	f1-score	support
Bad homeworks	1.00	0.79	0.88	1500
Good homeworks	0.83	1.00	0.91	1500
accuracy			0.90	3000
macro avg	0.91	0.90	0.89	3000
weighted avg	0.91	0.90	0.89	3000

data1_ classification_report_ n_neighbors = 43

	precision	recall	f1-score	support
Bad homeworks	1.00	0.79	0.88	1500
Good homeworks	0.83	1.00	0.91	1500
accuracy			0.90	3000
macro avg	0.91	0.90	0.89	3000
weighted avg	0.91	0.90	0.89	3000

data2_ classification_report_ n_neighbors = 43

	precision	recall	f1-score	support
Bad homeworks	1.00	0.74	0.85	1500
Good homeworks	0.79	1.00	0.89	1500
accuracy			0.87	3000
macro avg	0.90	0.87	0.87	3000
weighted avg	0.90	0.87	0.87	3000

data3_ classification_report_ n_neighbors = 43

	precision	recall	f1-score	support
Bad homeworks	1.00	0.73	0.85	1453
Good homeworks	0.80	1.00	0.89	1547
accuracy			0.87	3000
macro avg	0.90	0.87	0.87	3000
weighted avg	0.90	0.87	0.87	3000

data4_ classification_report_ n_neighbors = 43

考慮到不同類別的樣本不均衡，這邊看的是 weighted avg 做比較

data1 和 data2 的結果比較

發現 precision 和 recall 的分數都一樣，說明 Redundant Attributes 對原始資料沒影響，k 值提高，經分類的結果較不容易受干擾。

data2 和 data3 的結果比較

與 n_neighbors = 3 結果差不多，故不多做討論

data2 和 data4 的結果比較

	precision	recall	f1-score	support
Bad homeworks	1.00	0.73	0.84	1453
Good homeworks	0.79	1.00	0.89	1547
accuracy			0.87	3000
macro avg	0.90	0.86	0.86	3000
weighted avg	0.89	0.87	0.86	3000

data4_ classification_report_ n_neighbors = 53

從上圖可以發現，在放寬 Rules 的情形下，過高的 k 值(n_neighbors = 53)反而 f1-score 分數較低(降低 1%)，這是因為 k 值太大可能會將不相干的樣本點考慮進來。

k 值對分類結果的影響:

由以上結果可發現，k 值上升，可降低干擾的影響，但 k 值過大，在放寬 Rules 的 data4 卻呈現反效果，因為考慮了不相干的樣本點，但就整體而言，k 值上升，不同 data 的 precision 和 recall 皆隨之提高，模型對不

同干擾的抵抗能力皆會提升。

六、 K-means 分析

會隨機設定 k 個群心，以 2 元分類來說，k=2，接著計算樣本到群心的距離，將樣本分配到距離最近的群心，再從各個分群中隨機選出新的群心，重複此步驟值到群心不變動。以下設定 max_iter=100，亦即執行一次 K-means 的最大疊代次數為 100 次，n_init=10，用不同的初始化群心運行的次數。由于 K-Means 若初始設定的群心不優，容易陷入局部最佳解，因此需要多跑幾次選擇較好的聚類結果。

	precision	recall	f1-score	support
Bad homeworks	0.21	0.19	0.20	1500
Good homeworks	0.25	0.27	0.26	1500
accuracy			0.23	3000
macro avg	0.23	0.23	0.23	3000
weighted avg	0.23	0.23	0.23	3000

data1_ classification_report

	precision	recall	f1-score	support
Bad homeworks	0.78	0.80	0.79	1500
Good homeworks	0.79	0.78	0.79	1500
accuracy			0.79	3000
macro avg	0.79	0.79	0.79	3000
weighted avg	0.79	0.79	0.79	3000

data2_ classification_report

	precision	recall	f1-score	support
Bad homeworks	0.78	0.80	0.79	1500
Good homeworks	0.79	0.78	0.79	1500
accuracy			0.79	3000
macro avg	0.79	0.79	0.79	3000
weighted avg	0.79	0.79	0.79	3000

data3_ classification_report

	precision	recall	f1-score	support
Bad homeworks	0.68	0.72	0.70	1453
Good homeworks	0.72	0.68	0.70	1547
accuracy			0.70	3000
macro avg	0.70	0.70	0.70	3000
weighted avg	0.70	0.70	0.70	3000

data4_ classification_report

考慮到不同類別的樣本不均衡，這邊看的是 weighted avg 做比較

data1 和 data2 的結果比較

從結果來看，加入 Redundant Attributes 對 data 聚群分類的效果較佳，這可能跟 Rules 訂定的嚴厲程度有關。Rules 越嚴厲，代表個別 Attributes 的資料會分較開，較容易分群，但若只能劃分為兩類，同時又有多個 Attributes 資料分很開，想精準的切成兩群難度就很高了，而 Redundant Attributes 的加入(資料是隨機產生，呈高斯分布)，使高維度資料的離散程度相對減少，因此分群的效果提升。

data2 和 data3 的結果比較

錯誤的 Labels 在此模型中影響不大，分類效果差不多。

data2 和 data4 的結果比較

發現 precision 和 recall 都降低了 2%，表示放寬 Rules 使得個別 Attributes 的資料較為考攏，不易分群，雖有 Redundant Attributes 的幫助，但分群的效果相比 data2 來的差

K-means 分類效果討論:

由以述的結果比較可總結，想提升 K-means 分類效果可從兩方面下手，一是提升 Rules 的嚴厲程度，使個別 Attributes 的資料分較開，易於分群，二是加入呈高斯分布的 Redundant Attributes，使高維度資料離散程度相對減少，下方是**不加 Redundant Attributes，但放寬 Rules 的條件 (data5)**經 K-means 跑出的分類結果。

	precision	recall	f1-score	support
Bad homeworks	0.68	0.72	0.70	1453
Good homeworks	0.72	0.68	0.70	1547
accuracy			0.70	3000
macro avg	0.70	0.70	0.70	3000
weighted avg	0.70	0.70	0.70	3000

data5_ classification_report

比較 data1 和 data5，會發現雖然沒有 Redundant Attributes，僅放寬 Rules，但分類效果相較 data1 好很多(23%提升到 70%)

七、 綜合模型評估與討論

模型分類效果評估

根據上述四種模型的分類效果進行排序，可得以下結果:

data1: Random Forest > Decision Tree ≒ KNN > K-means

data2: Random Forest > Decision Tree ≒ KNN > K-means

data3: Random Forest > Decision Tree ≒ KNN > K-means

data4: Random Forest > Decision Tree ≒ KNN > K-means

Random Forest > Decision Tree

前者跟後者的差異在於引進了隨機取樣和表決的機制，同時也是後者的延伸運用，因此分類效果較佳

KNN > K-means

兩者都用到 K 值作為模型的重要參數，同時也須計算空間中點的距離，差別就在於前者的最終目的是做分類，屬於監督式學習的一種，但沒有 Loss function 的存在，僅與旁鄰中出現最多的分類標籤去經多數決進行預測，而後者是希望聚類，將空間中距離近的點分配在同一群，尋找最終的群心，

屬於非監督式學習的一種。從學習方式上來看，監督式學習有參考的目標，自然分類的效果會較非監督式學習來的好。

不同干擾對模型的影響

- **加入 Redundant Attributes**

  對於 Decision Tree、Random Forest、KNN 的 precision 和 recall 影響較小，但在 K-means 影響很大

- **錯誤的 Labels**

  對於 Decision Tree、Random Forest、KNN 的 precision 和 recall 都有影響，其中 recall 影響較大，但對 K-means 的影響較小

- **放寬 Rules**

- 對於 Decision Tree、Random Forest、KNN 的 precision 和 recall 都有影響，此干擾比錯誤的 Labels 影響還大，對 K-means 的影響也很大

各模型面對不同干擾的處理方式

- **加入 Redundant Attributes**

  在 Random Forest 和 Decision Tree 模型中，透過將深度調低，能減少 Redundant Attributes 對模型效果的影響。

  在 KNN 模型中，則是通過調高 k 值，來減少此干擾。

  在 K-means 模型中，適當加入此干擾，有機會提升模型分類效果。

- **錯誤的 Labels**

  只有 Random Forest 的隨機取樣和表決的機制比較有機會規避掉此問題。

- **放寬 Rules**

  在 Random Forest 和 Decision Tree 模型中，透過將深度調高，使考慮的 Attributes 增加，能提升模型分類效果。

  在 KNN 模型中，也可以通過調高 k 值，來減少此干擾，但 k 值也不能過大，怕考慮了不相干的樣本點導致反效果。

  在 K-means 模型中，適當加入此干擾，有機會提升模型分類效果。