專題二
白酒品質
分類預測

095 顏钧洲
093 楊濰寧
086 顏玎窈

目錄

一、前言

二、專題介紹:背景、動機及目的

三、研究架構 Research Architecture

四、執行方法及步驟

五、成果分析

六、結論

七、參考文獻

前言

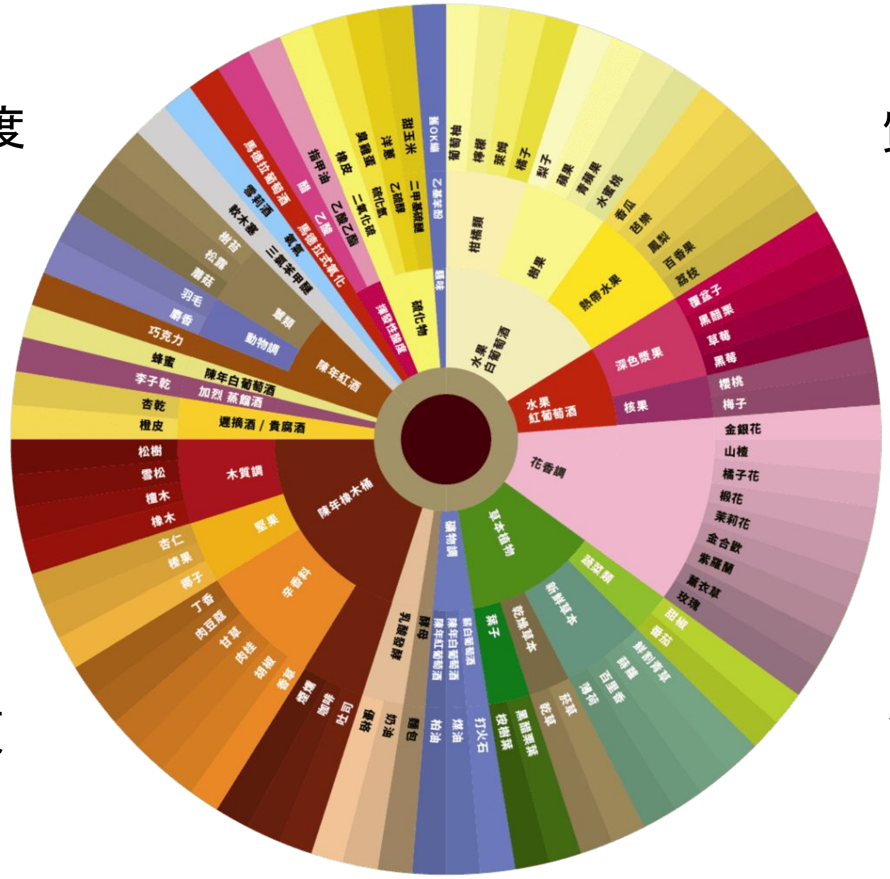
影響酒類的品質關鍵-

甜度

質地

酒精濃度

酸度



酯類

氣味

風味輪

品質

研究背景

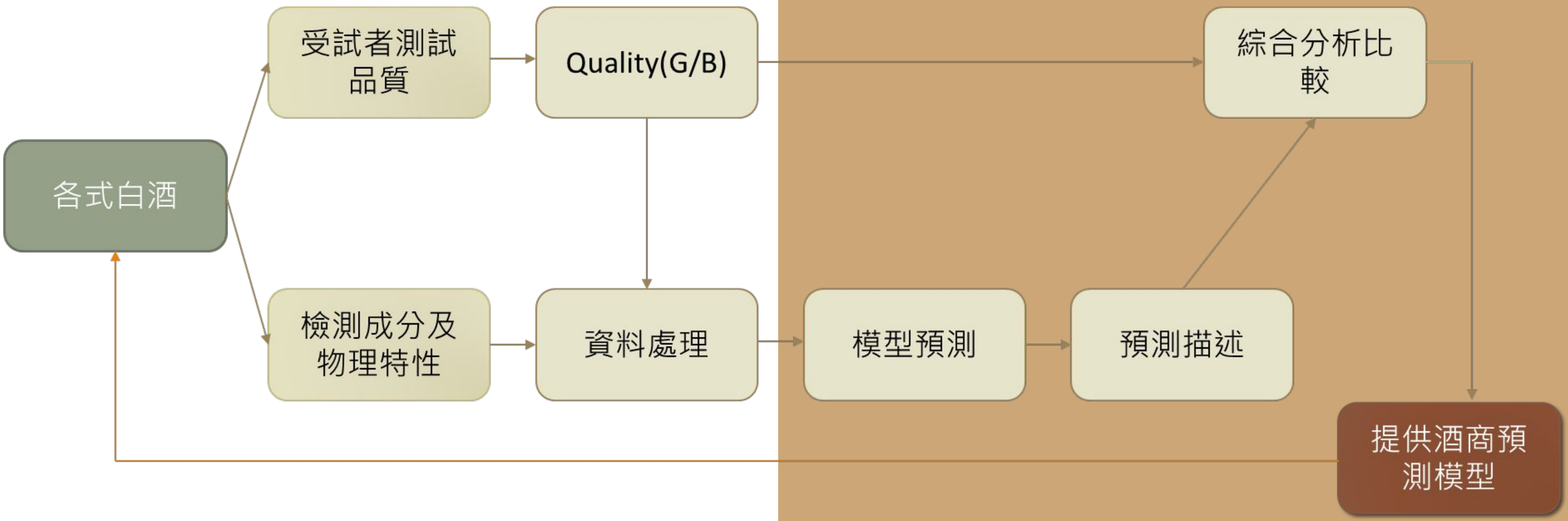
在酒類發酵過程中，雖然乙醇是最主要的產物。但酒類的香氣主要是來自於醇類、醛類、酸類、酯類以及雜醇油交互影響的結果。所以這些化合物，其成分含量及濃度多寡對於酒 的嗅覺及口感官能品評上是具有正面之意義。

研究動機及目的

因此我們希望建立一個以酒的質地、風味、氣味為基礎的模型，來預測酒類的品質，為釀酒廠商提供一套可預測客戶偏好之模型，幫助酒商研發更符合市場偏好的商品

研究架構

Research Architecture



白酒品質分類預測

- 資料集 : White Wine Quality
- 來源 : https://www.kaggle.com/datasets/piyushagni5/white-wine-quality
- 類別 : 分類
- 描述 :
紀錄樣本酒類物理及化學特性，並請受使者評分，以判斷酒類之物理化學特性是否影響風味。
- 輸入參數 : 見下頁
- 輸出參數 : 見下頁

Input variables
(based on physicochemical tests)

1 - fixed acidity 非揮發性酸

2 - volatile acidity 揮發性酸

3 - citric acid 檸檬酸

4 - residual sugar 殘糖

5 - chlorides 氯化物

6 - free sulfur dioxide 游離二氧化硫

7 - total sulfur dioxide 總二氧化硫

8 - density 濃度

9 - pH 酸鹼值

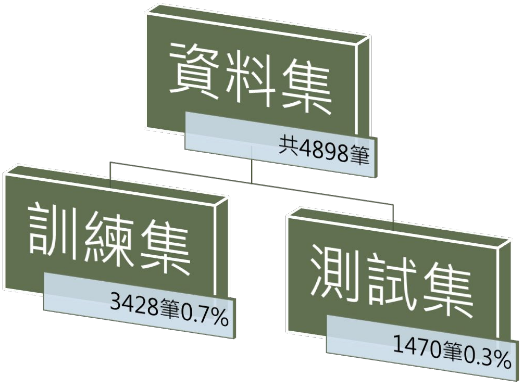
10 - sulphates 硫酸鹽

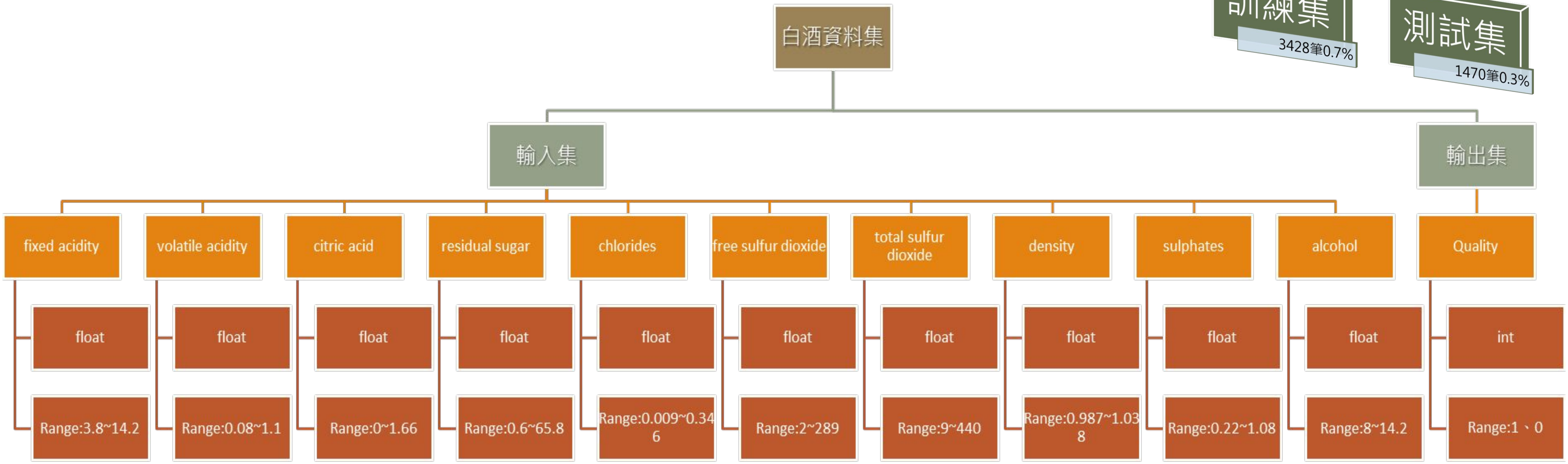
11 - alcohol 酒精濃度

Output variable (based on sensory data) :

12 - quality (score between 0 and 10)品質

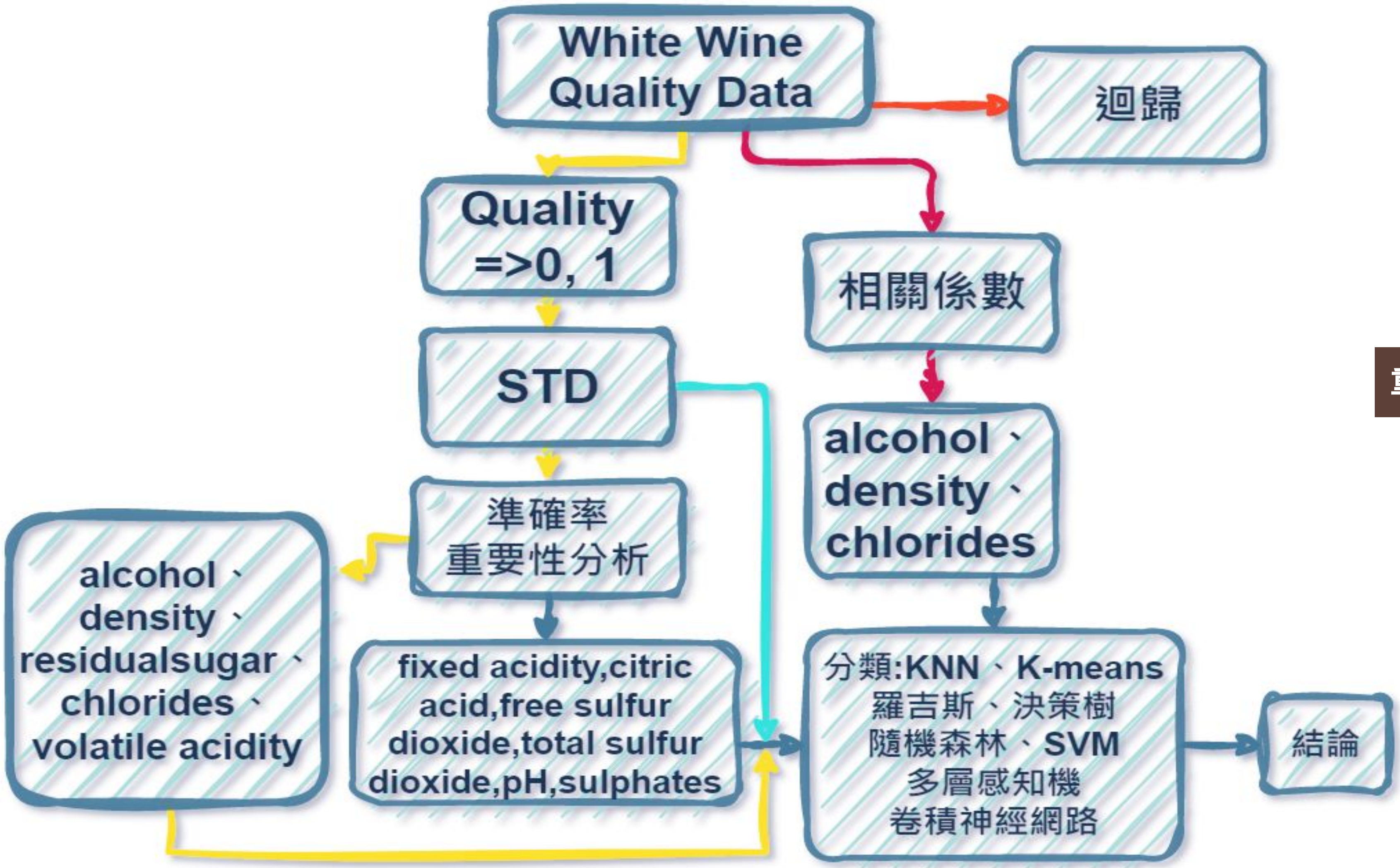
資料集設計
Design of Dataset





因數值差異大，本實驗將所有欄位的數值都先以標準化做處理再進行分析

執行方法及步驟



相關係數排序

	feature	importance
10	alcohol	0.320351
4	chlorides	0.128830
7	density	0.105121
6	total sulfur dioxide	0.083117
3	residual sugar	0.073098
5	free sulfur dioxide	0.071872
1	volatile acidity	0.062055
8	pH	0.046991
9	sulphates	0.043283
2	citric acid	0.037050
0	fixed acidity	0.028232

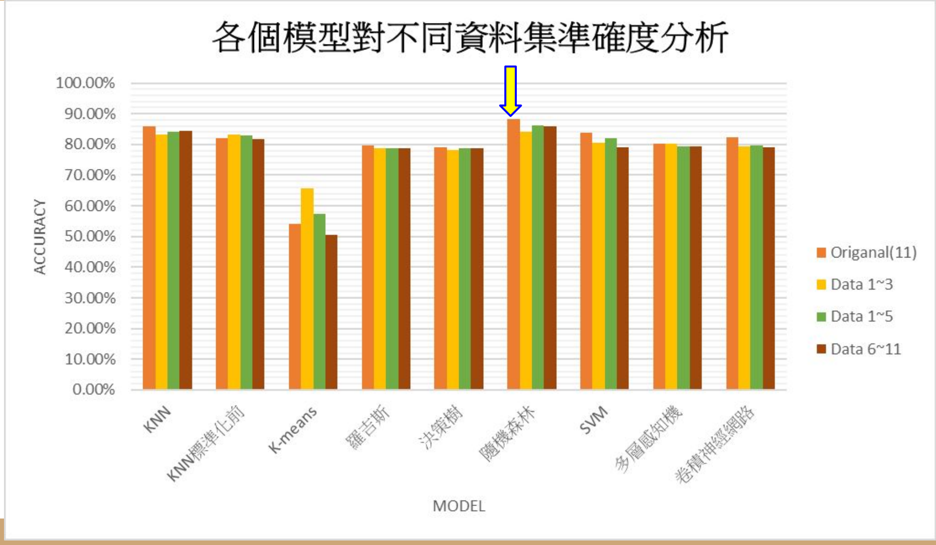
重要性排序

	feature	importance
10	alcohol	0.152134
7	density	0.119200
3	residual sugar	0.091456
4	chlorides	0.089534
1	volatile acidity	0.086117
5	free sulfur dioxide	0.084918
8	pH	0.082698
6	total sulfur dioxide	0.080799
9	sulphates	0.076855
2	citric acid	0.071235
0	fixed acidity	0.065055

成果分析

Accuracy	KNN	K-means	羅吉斯	決策樹	隨機森林	SVM	多層感知機	卷積神經網路
Origanal (11)	85.78%	54.12%	79.73%	79.05 %	***88.23%***	83.80%	80.40%	82.23%
Data 1-3 (3)	83.27%	65.78%	78.91%	78.23 %	84.08%	80.54%	80.13%	79.36%
Data 1-5 (5)	84.15%	57.21%	78.71%	78.64 %	86.26%	82.04%	79.45%	79.59%
Data 6-11 (6)	84.42%	50.65%	78.64%	78.71 %	85.85%	79.21%	79.31%	79.21%

- 準確度同時受特徵多寡及特徵的重要度影響
- 分類模型確實有強弱之分
- 原始資料集的訓練效果最好



觀察與結論

1.以KNN為例:

- 標準化前後特徵重要度會不一樣，準確率也會因此被影響
- 部分數據標準化後準確度會提高(eg. 82.17%->85.78%)

Accuracy	KNN	KNN標準化前
Origanal(11)	85.78%	82.17%
Data 1~3	83.27%	83.19%
Data 1~5	84.15%	83.06%
Data 6~11	84.42%	81.77%

2.測試不同的亂數種子可以提高模型準確度(更改Random_state)。

3.不同模型對於特徵多寡與特徵重要性的反饋都不一樣。

4.猜測KNN模型的輸入特徵越多，當K值遞增準確度遞減。

5.選擇神經網絡模型不能單看準確度，還要同時考慮模型是否過擬和

6.在模型的參數調整方面，各項參數不一定是越大越好，以隨機森林為例，深度超過20，準確度會隨之遞減。

7.隨機森林分類很強!

測試不同的亂數種子

```python
random_state_acc = {}
for r in range(100):
    XTrain, XTest, yTrain, yTest = train_test_split(X, y, test_size=0.3, random_state=r)

    # sc=StandardScaler()
    # sc.fit(XTrain)
    # X_train_sd=sc.transform(XTrain)
    # X_test_sd=sc.transform(XTest)
    # XTrain=X_train_sd
    # XTest=X_test_sd

    k=2
    knn = neighbors.KNeighborsClassifier(n_neighbors=k)
    knn.fit(XTrain, yTrain)
    random_state_acc[r] = knn.score(XTest, yTest)

max_val = max(random_state_acc, key=random_state_acc.get)
print(f"random_state:{max_val}\naccuracy:{random_state_acc[max_val]}")
```

取出準確率最高的亂數種子

```
random_state:5
accuracy:0.8217687074829932
```

```python
for k in range(1, 20):
    knn = neighbors.KNeighborsClassifier(n_neighbors=k)
    knn.fit(XTrain, yTrain)

    print(knn.score(XTest, yTest))
```

```
0.8537414965986394
0.8578231292517007
0.8448979591836735
0.8414965986394558
0.8299319727891157
0.8367346938775511
0.826530612244898
0.8306122448979592
0.8326530612244898
0.8312925170068027
0.826530612244898
0.8258503401360544
0.827891156462585
0.826530612244898
0.8285714285714286
0.8244897959183674
0.8258503401360544
0.8306122448979592
0.8319727891156462
```

Deep Learning has also been overhyped. Because neural networks are very technical and **hard to explain**, many of us used to explain it by drawing an analogy to the human brain. But we have pretty much no idea how the biological brain works.

—前百度首席科學家、史丹佛大學副教授吳恩達

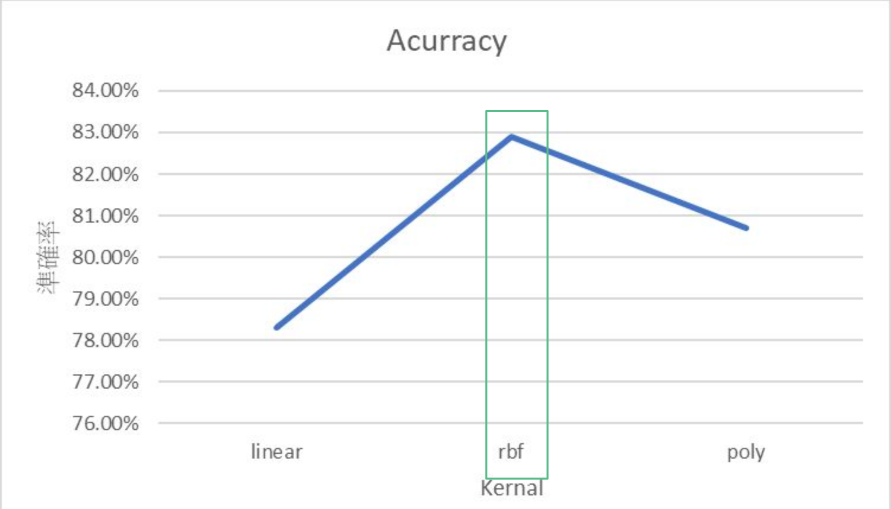
參考文獻

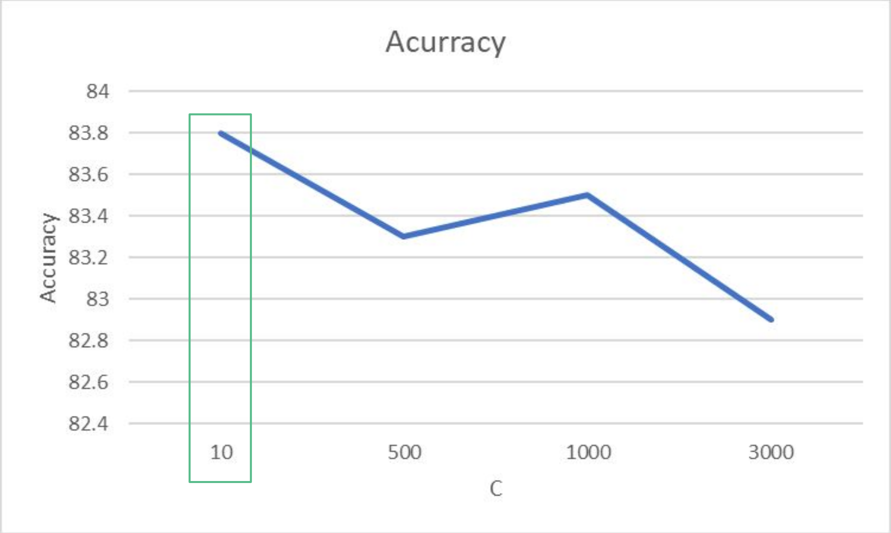
- https://cdrfoodlab.drins.com.tw/%E9%A3%B2%E6%96%99/%E8%98%8B%E6%9E%9C%E6%B0%B4%E6%9E%9C-%E9%85%92
- https://www.kaggle.com/datasets/piyushagni5/white-wine-quality
- https://www.quora.com/What-does-Andrew-Ng-think-about-Deep-Learning

BACKUP

結果(Result)-支援向量機

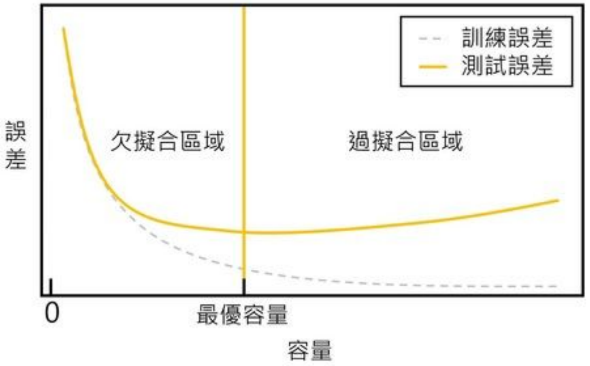
SVM	kernel	gamma	C	tolerance	Acurracy(%)
Test1	linear	auto	1	0.1	78.3/78.3
Test2	rbf	auto	1	0.1	83.3/82.9
Test3	poly	auto	1	0.1	81/80.7
Test4	rbf	auto	10	0.1	87.6/83.8
Test5	rbf	auto	10	1	87.8/83.6
Test6	rbf	auto	500	0.1	96.3/83.3
Test7	rbf	auto	1000	0.1	97.5/83.5
Test8	rbf	auto	3000	0.1	98.9/82.9
Test9(F6-11)	rbf	auto	10	0.1	82.03/81.22
Test10(F1-3)	rbf	auto	10	0.1	79.98/80.54
Test11(F1-5)	rbf	auto	10	0.1	82.20/82.04

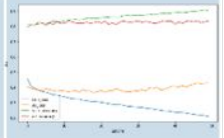
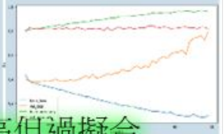
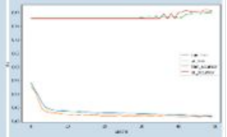
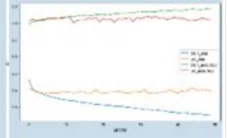
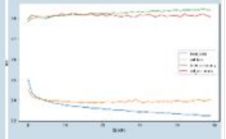




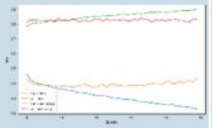
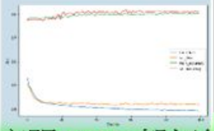
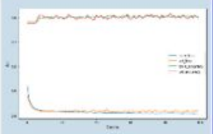
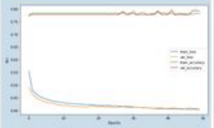
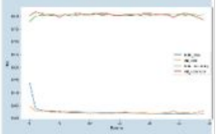
訓練集/測試集 準確度

結果(Result)-多層感知機



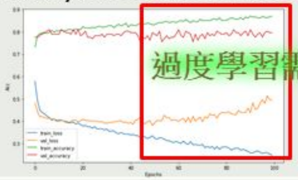
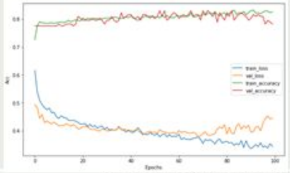
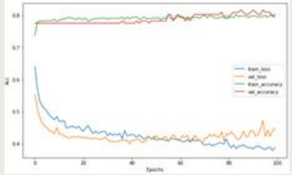
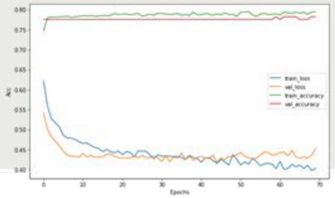
	Input	Hidden1	Hidden2	Output	epochs	Accuracy	Results
Test1	18	64	32	2	50	84.60%	
Test2	128	64	32	2	50	86.50%	
Test3	2	64	32	2	50	78.23%	
Test4	18	32	32	2	50	82.17%	
Test5	8	32	32	2	50	81.08%	

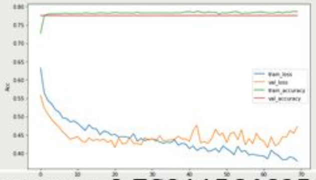
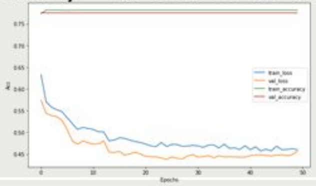
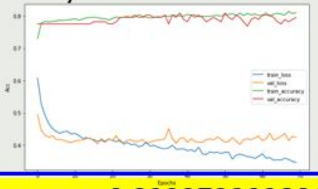
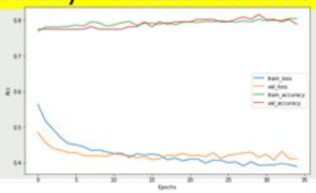
精準度高但過擬合

	Input	Hidden1	Hidden2	Hidden3	Hidden4	Output	epochs	Accuracy	Results
Test6	2	128	64	32		2	50	83.60%	
Test7	2	64	32			2	100	76.59%	
Test8	2	32	16	8		2	100	80.40%	
Test9	2	32	16	8	8	2	50	75.91%	
Test10	2	32	16	8	8	2	30	79.59%	

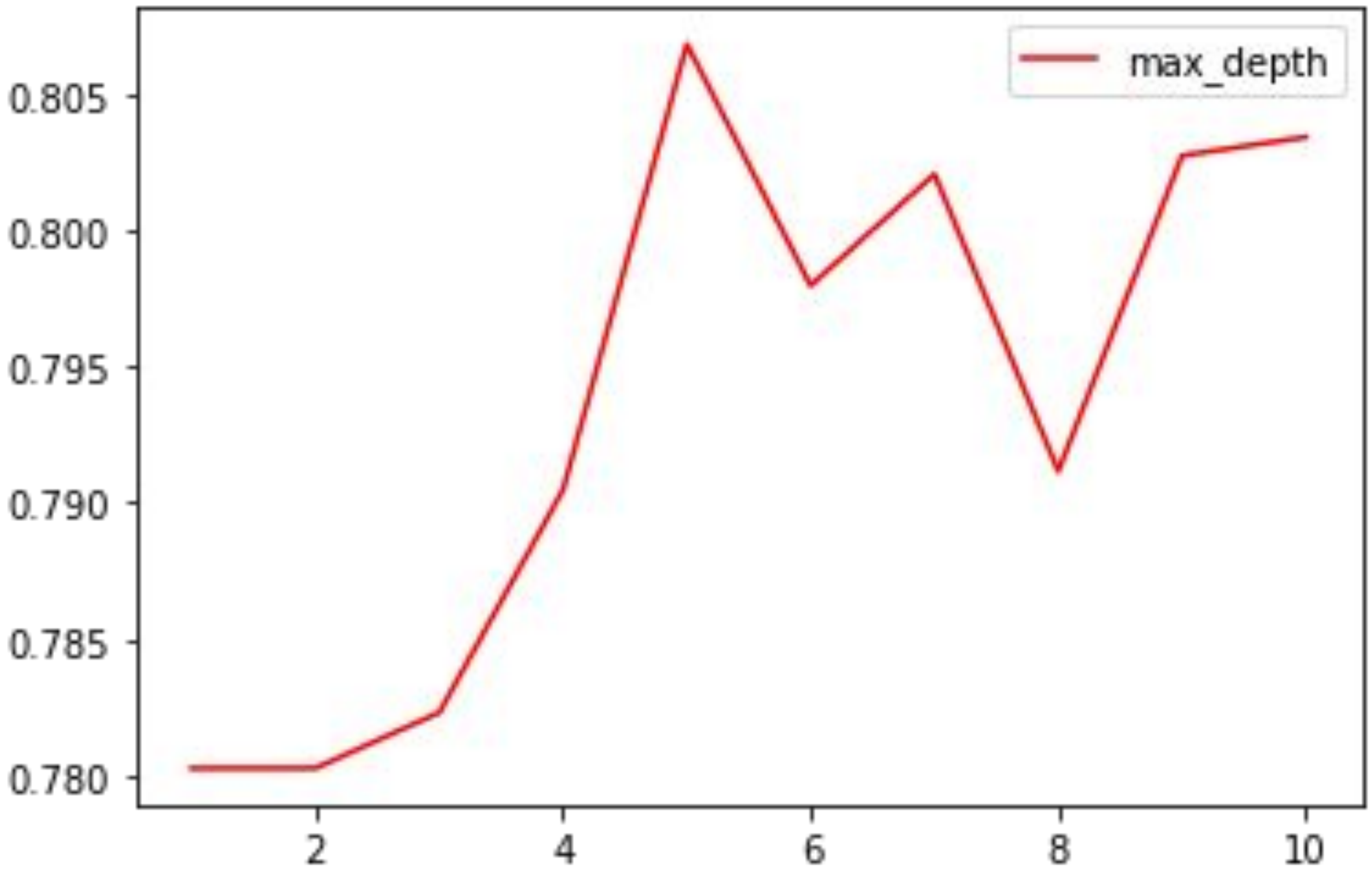
精準度跟fitting都好的最佳解

結果(Result)-卷機神機網絡

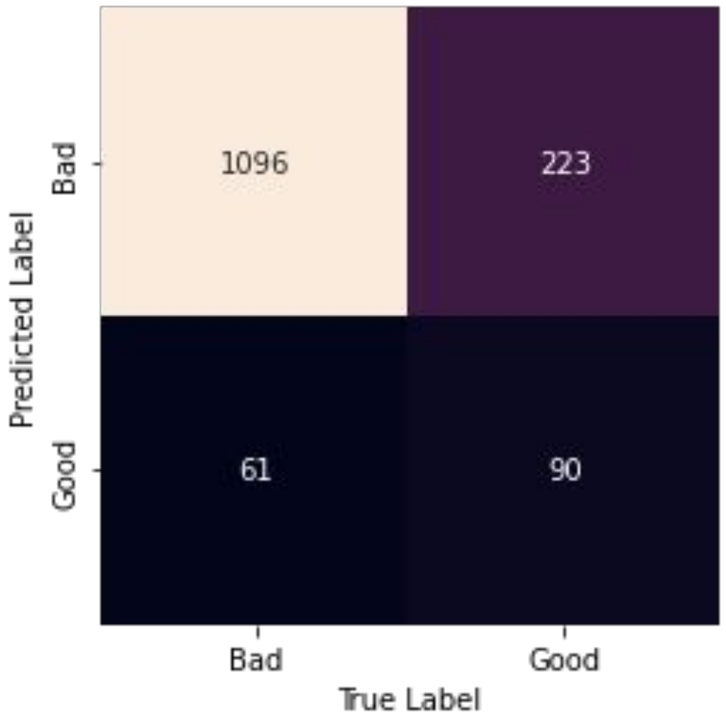
		Input	Convolution1	Convolution2	Output	Epoch	Results
Test1	Dense	32	16		2	100	Accuracy: 0.8510959939531368 
	Drop	0.5	0.5				
Test2	Dense	32	16		2	100	Accuracy: 0.8450491307634165 
	Drop	0.5	0.8				
Test3	Dense	32	16		2	100	Accuracy = 0.8057445200302343 
	Drop	0.8	0.8				
Test4	Dense	32	16		2	70	Accuracy = 0.7928949357520786 
	Drop	0.8	0.8				

		Input	Convolution1	Convolution2	Output	Epoch	Results
Test5	Dense	32	16	16	2	70	Accuracy = 0.7891156462585034 
	Drop	0.5	0.8	0.8			
Test6	Dense	2	32		2	50	Accuracy = 0.7891156462585034 
	Drop	0.5	0.9				
Test7	Dense	32	16		2	70	Accuracy = 0.8337112622826909 
	Drop	0.5	0.8				
Test8	Dense	32	16		2	35	最佳解 Accuracy = 0.8223733938019653 
	Drop	0.5	0.8				

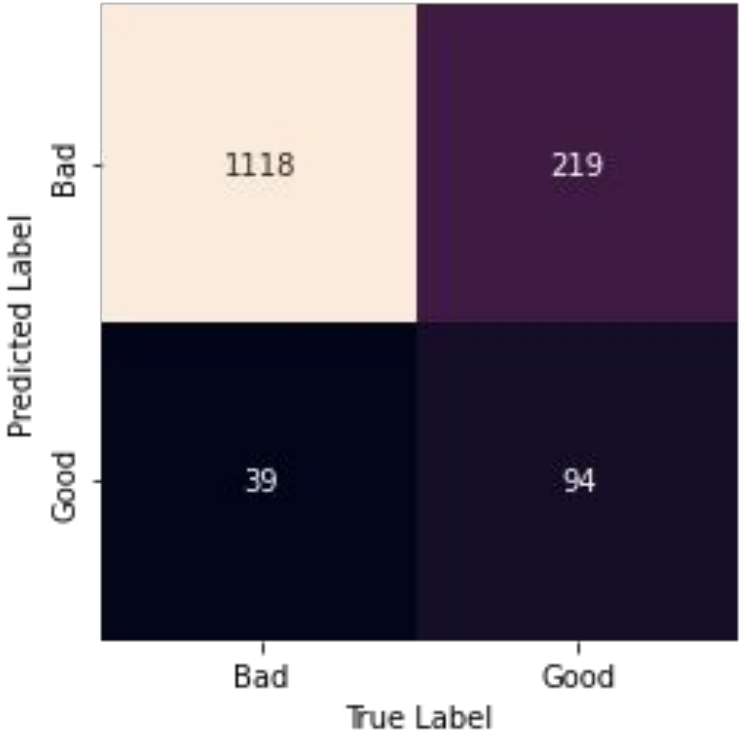
決策樹
最大深度參數之正確率



Accuracy =
80.68027210884354 %

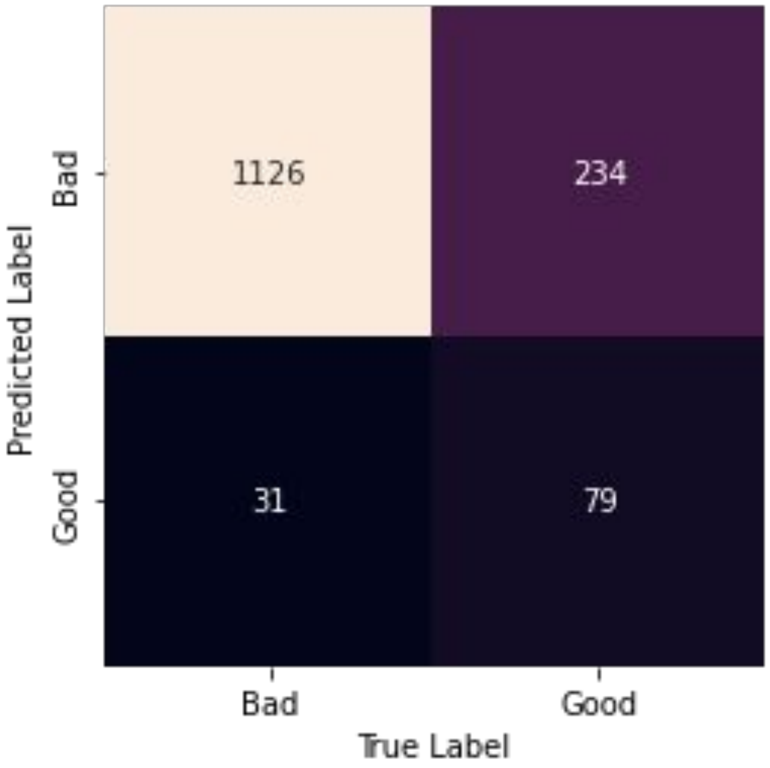


變數:總樹量
最佳參數:10、100



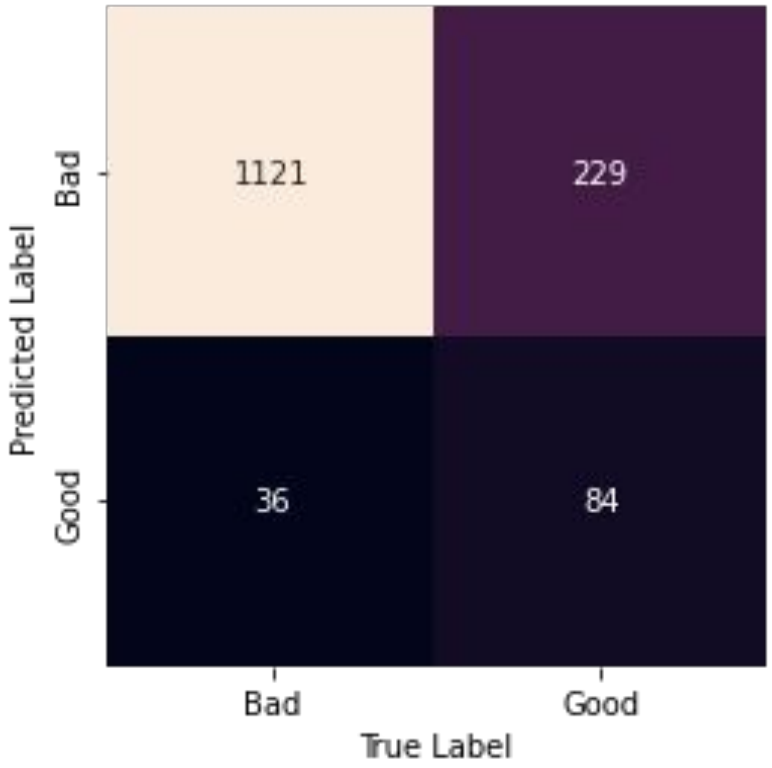
隨機森林
參數:(gini,10,5,10)
Accuracy = 82.44897959183673 %

	precision	recall	f1-score	support
Bad	0.84	0.97	0.90	1157
Good	0.71	0.30	0.42	313
accuracy			0.82	1470
macro avg	0.77	0.63	0.66	1470
weighted avg	0.81	0.82	0.80	1470



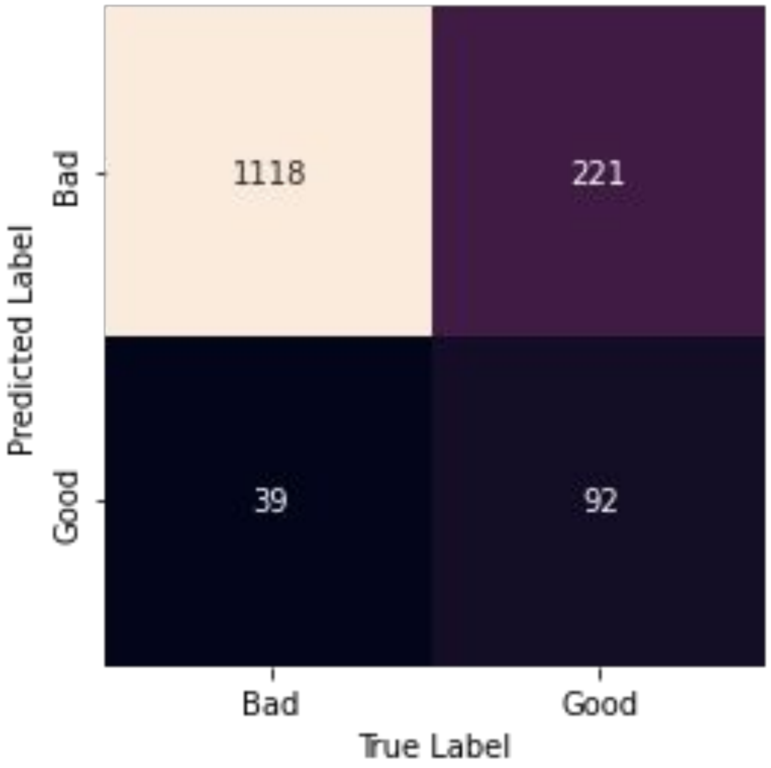
隨機森林
參數:(gini,20,5,10)
Accuracy = 81.97278911564626 %

	precision	recall	f1-score	support
Bad	0.83	0.97	0.89	1157
Good	0.72	0.25	0.37	313
accuracy			0.82	1470
macro avg	0.77	0.61	0.63	1470
weighted avg	0.80	0.82	0.78	1470



隨機森林
參數:(gini,50,5,10)
Accuracy = 81.97278911564626 %

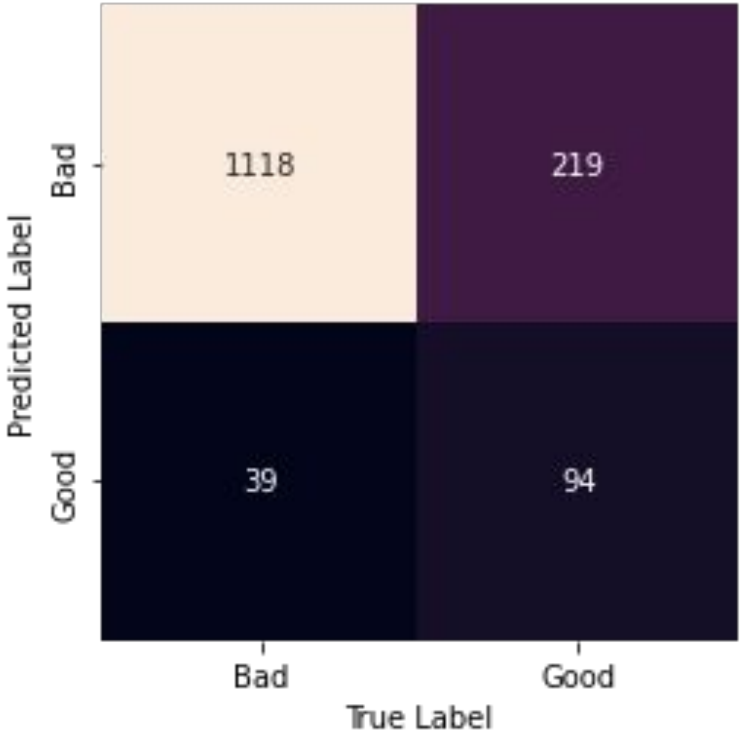
	precision	recall	f1-score	support
Bad	0.83	0.97	0.89	1157
Good	0.70	0.27	0.39	313
accuracy			0.82	1470
macro avg	0.77	0.62	0.64	1470
weighted avg	0.80	0.82	0.79	1470



隨機森林
參數:(gini,100,5,10)
Accuracy = 82.31292517006803 %

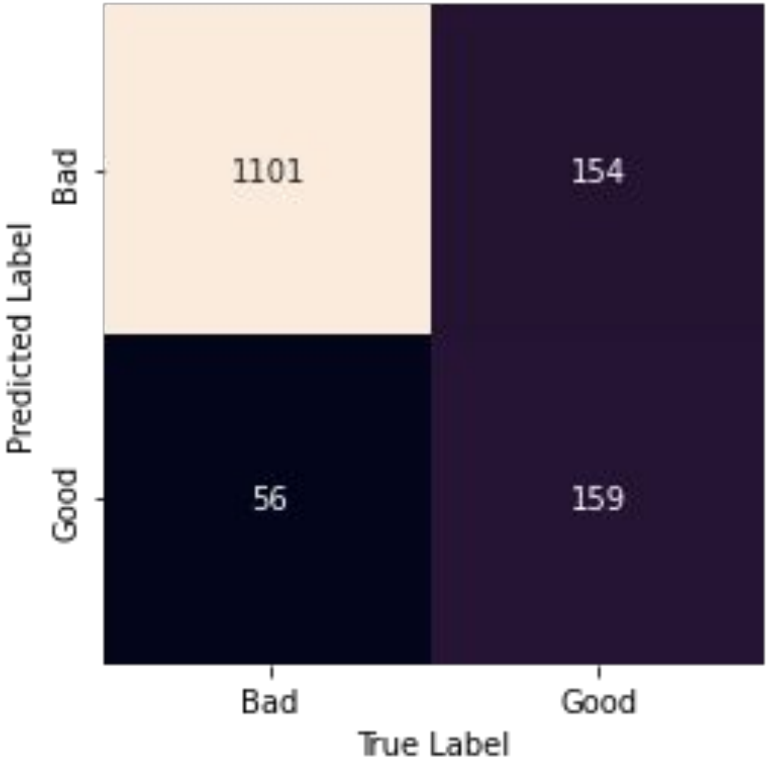
	precision	recall	f1-score	support
Bad	0.83	0.97	0.90	1157
Good	0.70	0.29	0.41	313
accuracy			0.82	1470
macro avg	0.77	0.63	0.66	1470
weighted avg	0.81	0.82	0.79	1470

變數:最大深度
最佳參數:15、10



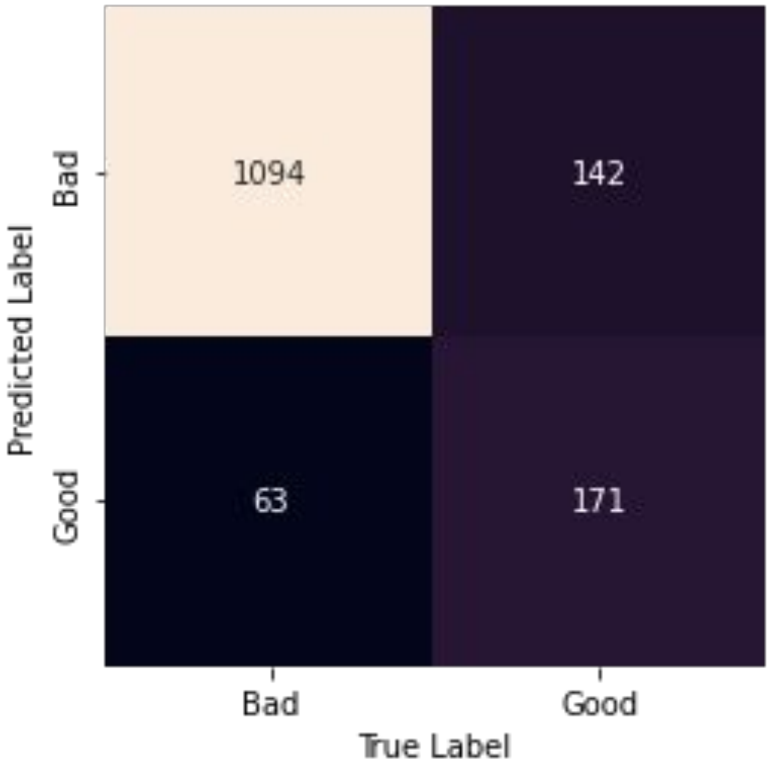
隨機森林
參數:(gini,10,5,10)
Accuracy = 82.44897959183673 %

	precision	recall	f1-score	support
Bad	0.84	0.97	0.90	1157
Good	0.71	0.30	0.42	313
accuracy			0.82	1470
macro avg	0.77	0.63	0.66	1470
weighted avg	0.81	0.82	0.80	1470



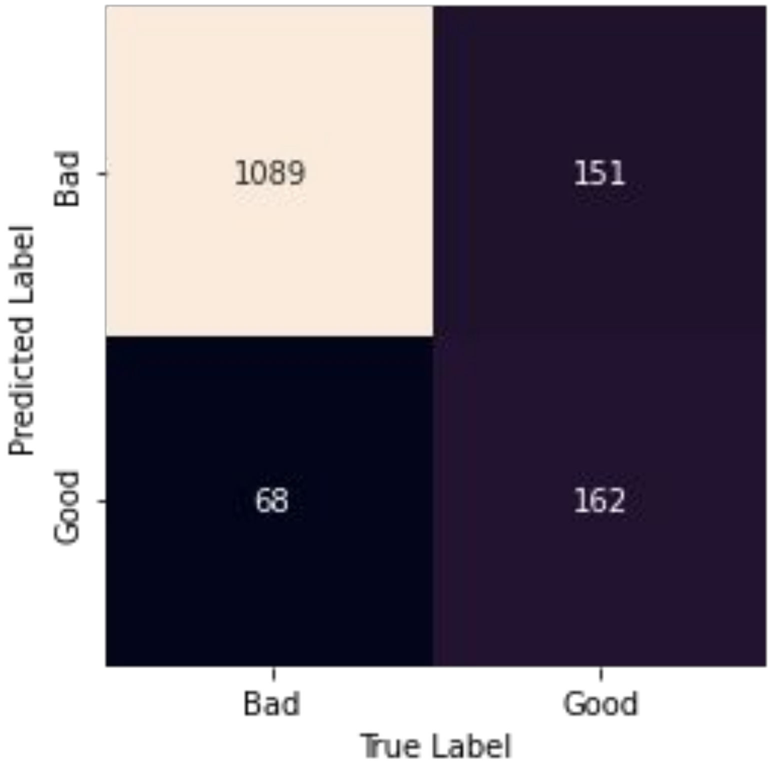
隨機森林
參數:(gini,10,10,10)
Accuracy = 85.71428571428571 %

	precision	recall	f1-score	support
Bad	0.88	0.95	0.91	1157
Good	0.74	0.51	0.60	313
accuracy			0.86	1470
macro avg	0.81	0.73	0.76	1470
weighted avg	0.85	0.86	0.85	1470



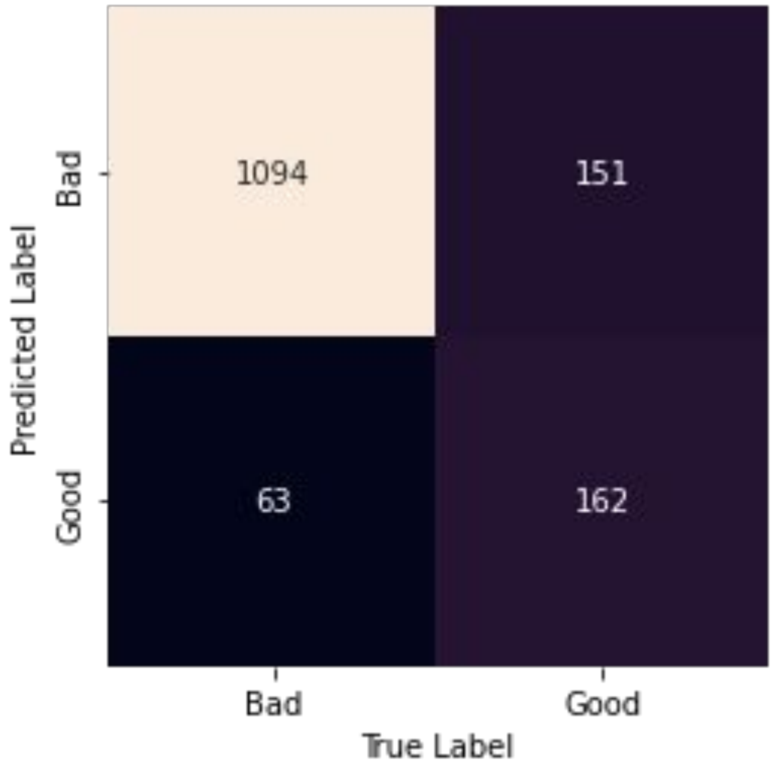
隨機森林
參數:(gini,10,15,10)
Accuracy = 86.05442176870748 %

	precision	recall	f1-score	support
Bad	0.89	0.95	0.91	1157
Good	0.73	0.55	0.63	313
accuracy			0.86	1470
macro avg	0.81	0.75	0.77	1470
weighted avg	0.85	0.86	0.85	1470



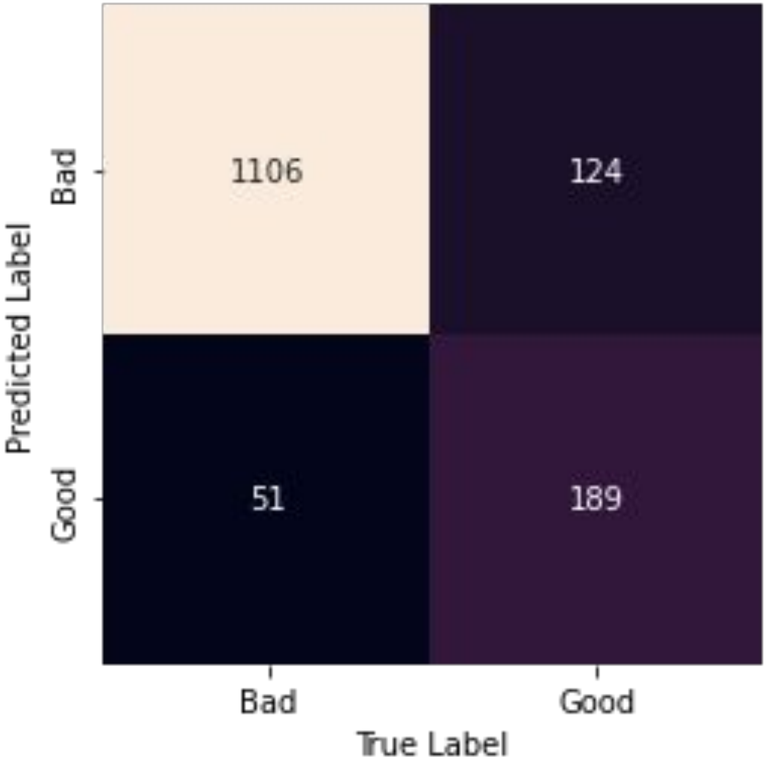
隨機森林
參數:(gini,10,20,10)
Accuracy = 85.10204081632654 %

	precision	recall	f1-score	support
Bad	0.88	0.94	0.91	1157
Good	0.70	0.52	0.60	313
accuracy			0.85	1470
macro avg	0.79	0.73	0.75	1470
weighted avg	0.84	0.85	0.84	1470



隨機森林
參數:(gini,10,50,10)
Accuracy = 85.4421768707483 %

	precision	recall	f1-score	support
Bad	0.88	0.95	0.91	1157
Good	0.72	0.52	0.60	313
accuracy			0.85	1470
macro avg	0.80	0.73	0.76	1470
weighted avg	0.84	0.85	0.85	1470



隨機森林
參數:(gini,100,50,10)
Accuracy = 88.09523809523809 %

	precision	recall	f1-score	support
Bad	0.90	0.96	0.93	1157
Good	0.79	0.60	0.68	313
accuracy			0.88	1470
macro avg	0.84	0.78	0.81	1470
weighted avg	0.88	0.88	0.87	1470

隨機森林參數組合之
準確率比較

最大深度\總樹量	10	20	50	100
5	82.45%	81.97%	81.97%	82.31%
10	85.71%	86.05%	86.46%	86.53%
15	86.05%	87.21%	87.48%	87.62%
20	85.10%	86.19%	87.62%	**88.23%**
50	85.44%	86.67%	87.41%	88.10%
100	85.44%	86.67%	87.41%	88.10%

羅吉斯回歸

準確率

參數(tol=0.1, C=1, max_iter=5)

未經標準化：78.70748299319727 %

標準化：79.72789115646258 %

正規化(1)：78.50340136054422 %

正規化(2)：77.89115646258503 %

參數調整(tol=4, C=0.1, max_iter=10)：79.25170068027211 %

	全	前三	前五	後六	參數
決策樹	79.05 %	78.23 %	78.64 %	78.71 %	criterion=gini max_depth=4 splitter=random
隨機森林	88.23% 87.95%	84.08 % 84.21%	86.26% 86.25%	85.85% 85.99%	criterion='gini' n_estimators=100 max_depth=20 random_state=10
羅吉斯	79.73%	78.91%	78.71%	78.64%	tol=0.1 C=1 max_iter=5