Personal Key Indicators of Heart Disease

心臟病關鍵指標

Data science final project study

指導老師 : 張家銘 老師

第三組 學生 :

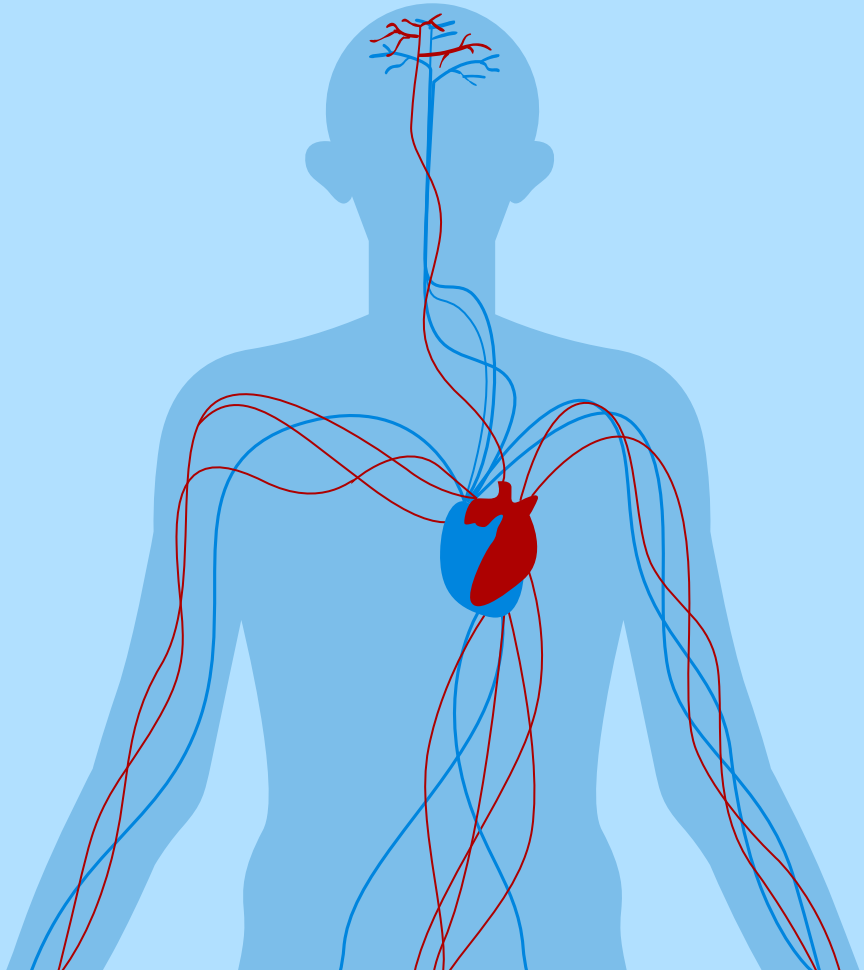
周正晏 111971003　施宗佑 111971005 楊昇豐 111971013

謝弘軒 111971022　郭書瑋 111971023 胡元亨 111971024

TABLE OF CONTENTS

01

ABOUT DATASET



About Dataset

Kaggle Project - 2020年的美國疾病控制與預防中心(CDC)對40萬名成年人進行的年度調查數據與其健康狀況相關。

根據CDC的數據，**心臟病是美國各種族人群**(非洲裔美國人、美洲原住民和阿拉斯加原住民以及白人)的**主要死因之一**。大約有一半的美國人(47%)至少有三個主要心臟病風險因素之一：高血壓、高膽固醇和吸煙。其他關鍵指標包括糖尿病狀態、肥胖(高BMI)、缺乏足夠的體育活動或過量飲酒。檢測和預防對心臟病產生最大影響的因素在醫療保健中非常重要。**目標使用機器學習方法，從檢測數據中預測患者狀況的"概況"。**

數據集來自CDC，是行為風險因素監測系統(BRFSS)的一個重要部分，該系統每年進行電話調 查，收集美國居民健康狀況的數據。BRFSS每年完成超過40萬次成人訪談，使其成為全球最大的持續進行的健康調 查系統。"最近的數據集(截至2022年2月15日)包含了2020年的數據。它由401,958行和279列組成。絕大多數列是對受訪者關於他們健康狀況的問題，例如"您是否在行走或爬樓梯時有嚴重困難？ "或"您是否在整個人生中至少吸過 100支香煙？[註：5包 = 100支香煙]"。在這個數據集中，有許多直接或間接影響心臟病的不同因素(問題)。

About Dataset

樣本總數： 共 319795 筆

目標欄位： 心臟病 Heart Disease (Yes or No)

特徵資料：共 18 列變量

數值型欄位資料：

BMI、PhysicalHealth、MentalHealth、AgeCategory、SleepTime

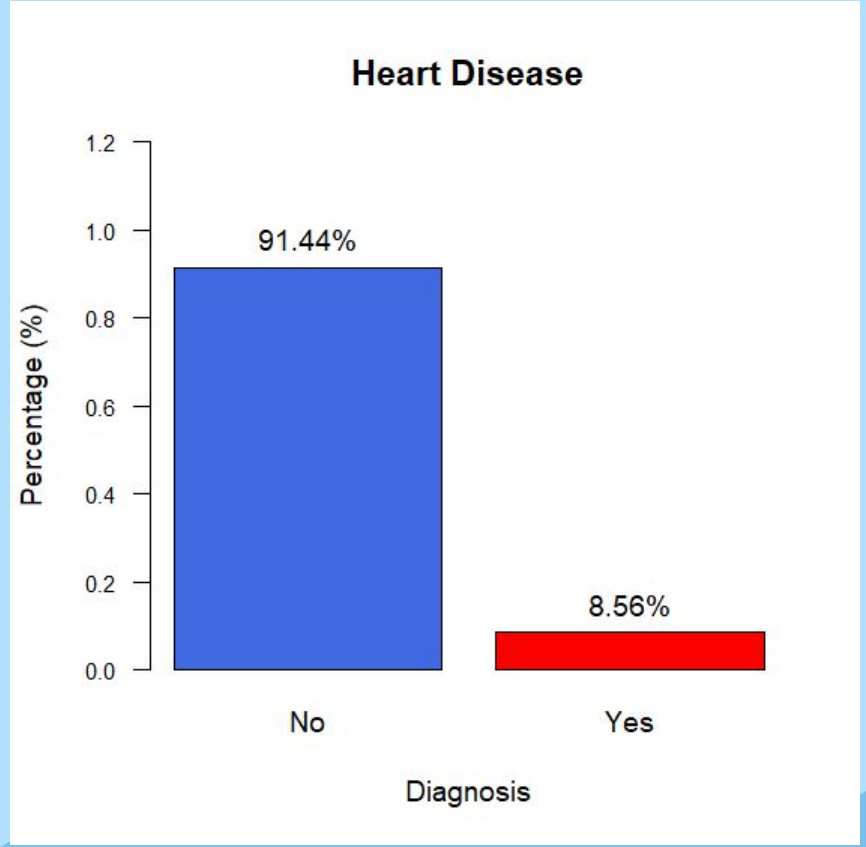
類別型欄位資料：

Smoking、AlcoholDrinking、Stroke、DiffWalking、Sex、Race、Diabetic、PhysicalActivity、GenHealth、Asthma、KidneyDisease、SkinCancer

About Dataset

心臟疾病標籤分析

- 有心臟病資料僅佔總資料8.56%
- 在樣本比例失衡下，需要做資料整理來平衡樣本

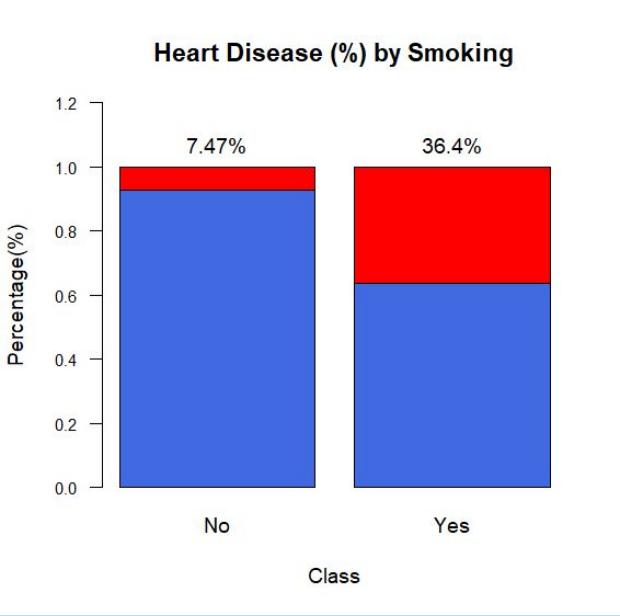


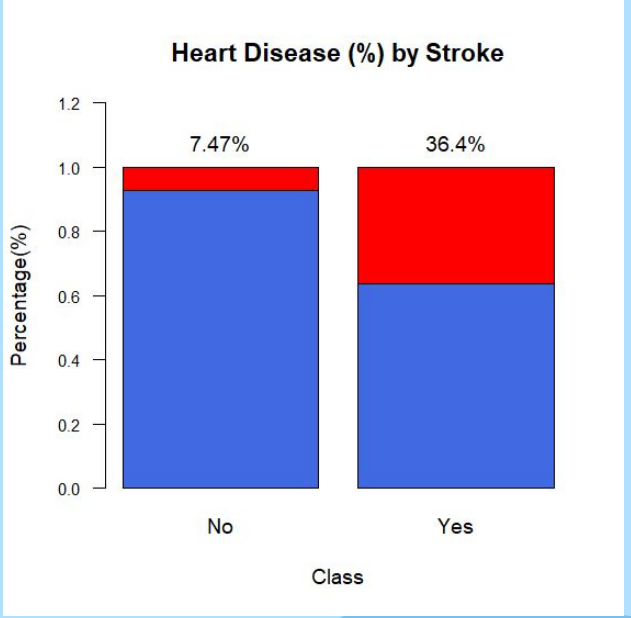
原始資料集觀察 – 罹病相關項目

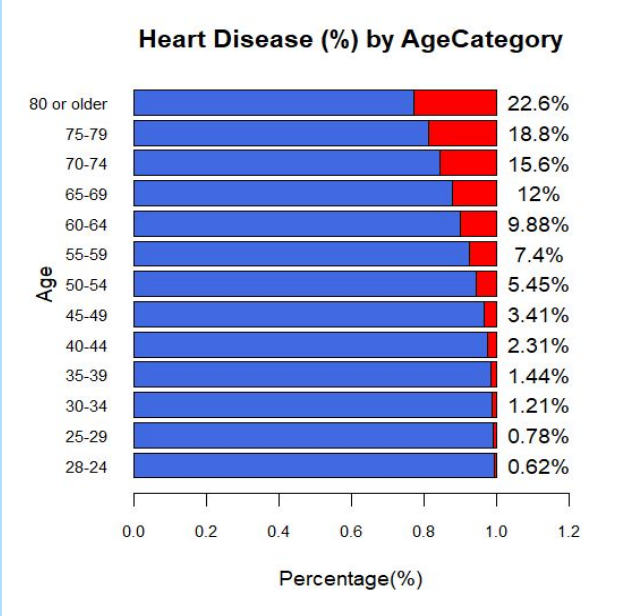
吸菸者比例分析：
Yes 高於 No 約 2 倍

中風者分析：
Yes 高於 No 約 4.8倍

年齡分布比例分析：
50 歲以後罹病比例遽增



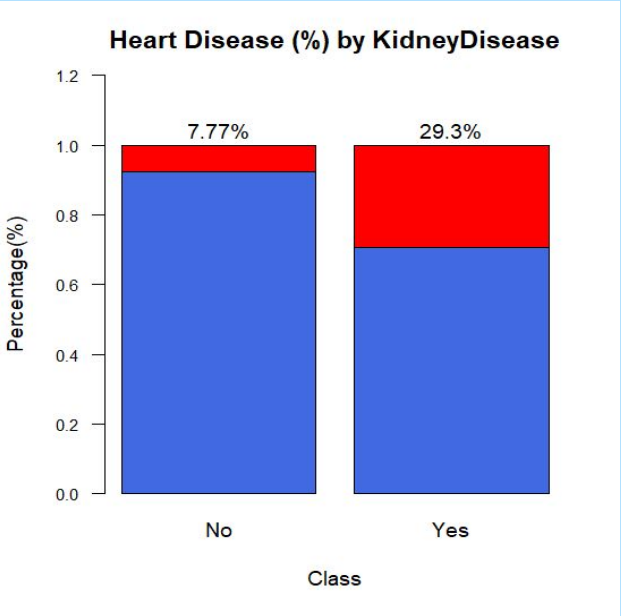




原始資料集觀察 – 罹病相關項目

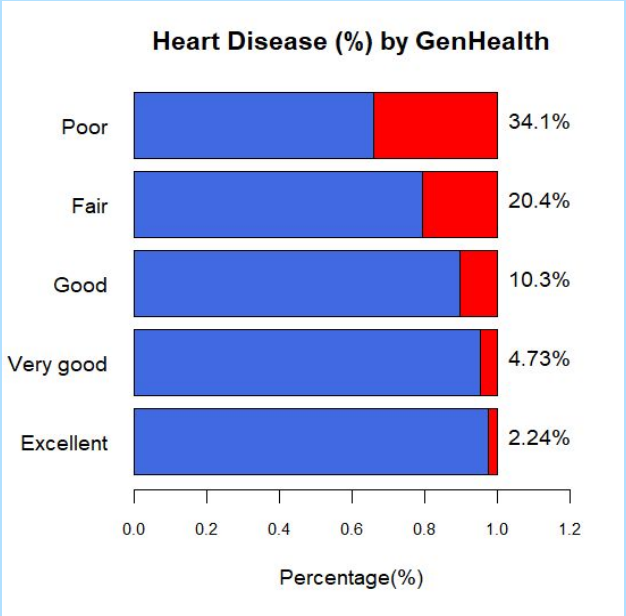
腎臟病患者比例分析：

Yes 高於 No 約 3.8 倍



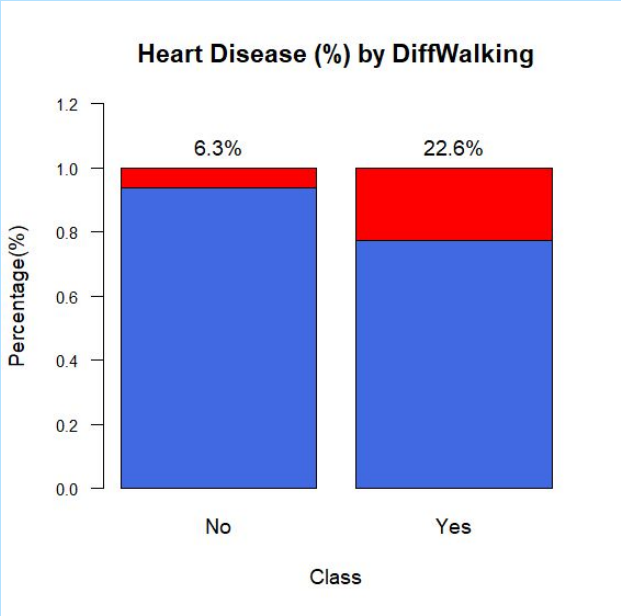
健康狀況比例分析：

心臟病患者隨健康不良程度增高



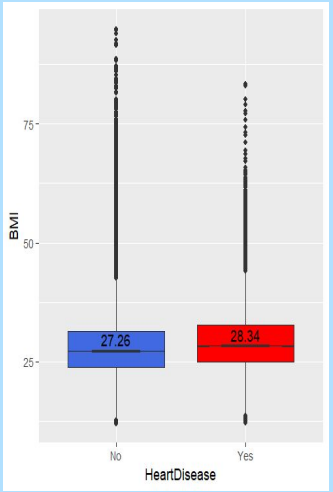
不良於行者(行動不便者?)比例分析：

Yes 高於 No 約 3.5 倍

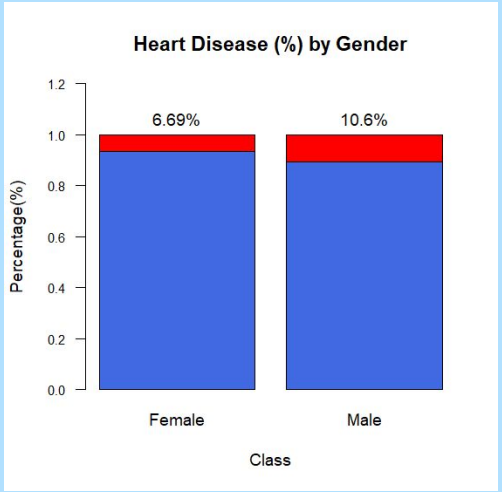


原始資料集觀察無明顯相關但分析後其實重要
透過VIF分析察覺潛藏重要欄位?

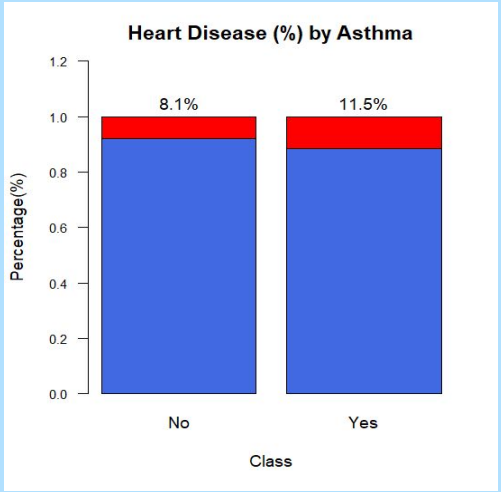
BMI



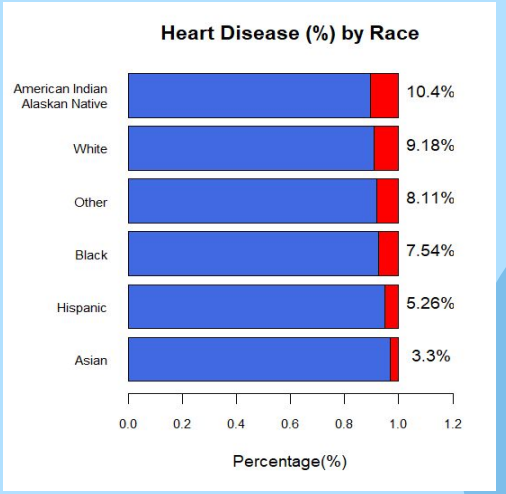
性別



哮喘患者



種族分布比例



上述變量初步觀察雖無明顯相關，但經資料前處理 VIF (Variance Inflation Factor) 確認其共線性低，可有效作為Model 訓練觀察變量

02

資料前處理

原始資料集的分析問題：NA、重複、特徵類型、不平衡

```
> str(dataset)
'data.frame':  319,795 obs. of  18 variables:
$ HeartDisease    : chr  "No" "No" "No" "No" ...      ⇒ target column, binary
$ BMI             : num  16.6 20.3 26.6 24.2 23.7 ... ⇒ numeric；4 個
$ Smoking         : chr  "Yes" "No" "Yes" "No" ...
$ AlcoholDrinking : chr  "No" "No" "No" "No" ...     ⇒ binary；9 個
$ Stroke          : chr  "No" "Yes" "No" "No" ...
$ PhysicalHealth  : num  3 0 20 0 28 6 15 5 0 0 ...
$ MentalHealth    : num  30 0 30 0 0 0 0 0 0 0 ...
$ DiffWalking     : chr  "No" "No" "No" "No" ...
$ Sex             : chr  "Female" "Female" "Male" "Female" ...
$ AgeCategory     : chr  "55-59" "80 or older" "65-69" "75-79" ...⇒ category；4 個
$ Race            : chr  "White" "White" "White" "White" ...
$ Diabetic        : chr  "Yes" "No" "Yes" "No" ...
$ PhysicalActivity: chr  "Yes" "Yes" "Yes" "No" ...
$ GenHealth       : chr  "Very good" "Very good" "Fair" "Good" ...
$ SleepTime       : num  5 7 8 6 8 12 4 9 5 10 ...
$ Asthma          : chr  "Yes" "No" "Yes" "No" ...
$ KidneyDisease   : chr  "No" "No" "No" "No" ...
$ SkinCancer      : chr  "Yes" "No" "No" "Yes" ...
```

```
summary(duplicated(dataset))
Mode   FALSE    TRUE
logical  301,717   18,078

table(dataset$HeartDisease)
    No     Yes
292,422  27,373
```

沒有NA；有18078列重複；沒病：有病約10：1(不平衡資料集)

為了建模需要，進行如何的前處理

1. **特徵屬性有numeric、binary、category；**

   應對方式：

特徵屬性	作法	說明
numeric	--	–
binary	轉成numeric	1. Yes->1 2. No->0
category	轉成numeric 擴展成dummy col	1. 特徵AgeCategory:chr "55-59" -> 中位數57 2. 特徵Race-> Race_White、Race_Black、... a. 移除特徵Race

2. **有18,078列重複：**
   應對方式：去重，319,795 - 18,078 = 301,717
3. **不平衡資料：**
   應對方式：對去重後的訓練集資料取平衡，再建立模型；

資料集比例	沒病	有病
未取平衡後的比例	10	1
取平衡後的比例	1.3~4	1
備註	21萬取平衡，占比 1.3； 10萬取平衡，占比 4	--

切3個fold， 使用決策樹	取平衡的 訓練集	未取平衡的 測試集	未取平衡的 驗證集
建立模型	V	--	--
模型預測	V	V	V

train/test，使用隨機 森林、XGBoost	取平衡的 訓練集	未取平衡的 測試集
建立模型	V	--
模型預測	V	V

未處理、前處理後的資料集差異對照:

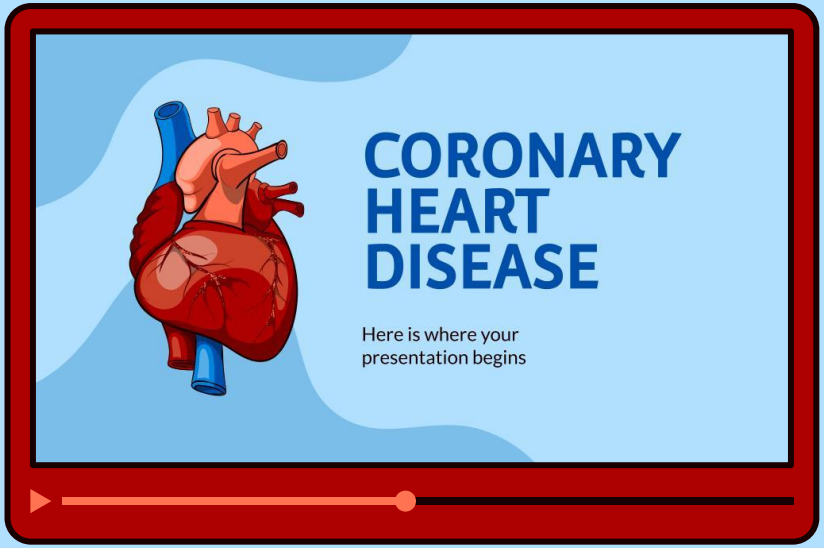
	未處理	前處理後
資料筆數	319,795	301,717
不平衡的狀況	沒病　有病 292,422 27,373	沒病　有病 274,456 27,261
不平衡的比例	10:1	10:1
取平衡後的比例	--	訓練集資料呈現1.3~4:1
欄位數量(扣除target)	17	29
特徵屬性轉換示意		

```
$ HeartDisease    : chr  "No" "No" "No" "No" ...
$ BMI             : num  16.6 20.3 26.6 24.2 23.7 ...
$ Smoking         : chr  "Yes" "No" "Yes" "No" …
$ AgeCategory     : chr  "55-59" "80 or older" "65-69" "75-79" ...
$ Race            : chr  "White" "White" "White" "White" ...
```

```
$ HeartDisease                    : chr  "No" "No" "No" "No" ...
$ BMI                             : num  16.6 20.3 26.6 24.2 23.7 ...
$ Smoking                         : num  1 0 1 0 0 1 0 1 0 0 ...
$ AgeCategory                     : num  57 80 67 77 42 77 72 80 80 67 ...
$ Race_American_Indian_Alaskan_Native: int  0 0 0 0 0 0 0 0 0 0 ...
$ Race_Asian                      : int  0 0 0 0 0 0 0 0 0 0 ...
$ Race_Black                      : int  0 0 0 0 0 1 0 0 0 0 ...
$ Race_Hispanic                   : int  0 0 0 0 0 0 0 0 0 0 ...
$ Race_Other                      : int  0 0 0 0 0 0 0 0 0 0 ...
$ Race_White                      : int  1 1 1 1 1 0 1 1 1 1 ...
```

03

各模型的執行狀況


CORONARY HEART DISEASE
Here is where your presentation begins

各模型執行情況_1

model	決策樹	隨機森林	XGBoost
苦主	昇豐	元亨/書瑋	宗佑/正晏
k-fold/ training+test	k-fold, k = 3	training+test 70%+30%	
smote(處理不平衡)	smote(trainset=100,000) 沒病 : 有病 = 4 : 1	smote(trainset=210,000) 沒病 : 有病 = 1.3 : 1	
threshold(閥值)	--	--	基於F1得出 29 -> 0.4 13 -> 0.35
model 參數	minsplit = 5, minbucket = 5, cp = 0.001	ntree=100, mtry=3	objective = binary:logistic, eval_metric = error, max_depth = 6, eta = 0.3
feature selection	前處理後, 產生29 feature。 基於glm, 得出13 feature。		

各模型執行情況_2

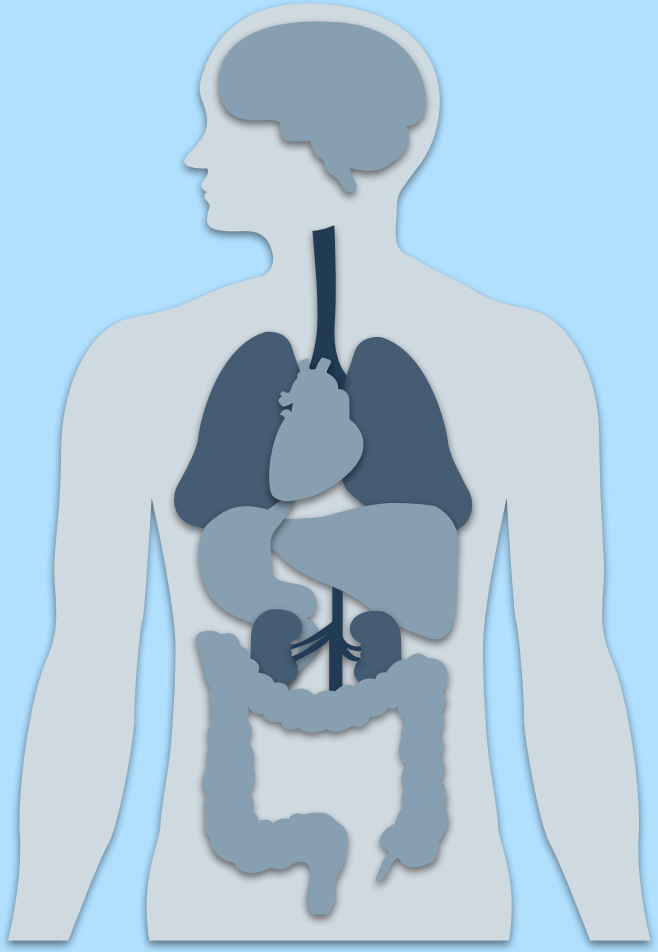
model	決策樹		隨機森林		XGBoost	
Accuracy	29 -> 0.89	13 -> 0.89	29 -> 0.82	13 -> 0.77	29 -> 0.857	13 -> 0.834
Precision	0.38	0.38	0.27	0.23	0.314	0.288
Recall= Sensitivity= True positive rate	0.34	0.33	0.65	0.74	0.506	0.576
Specificity= True negative rate	0.95	0.95	0.83	0.77	0.891	0.860
F1	0.35	0.35	0.38	0.36	0.388	0.384
AUC	0.64024	0.6357	0.74	0.76	0.833	0.830

參考家銘老師的建議，以AUC決定模型優劣，建議使用隨機森林、XGBoost處理。

理由：資料集是不平衡(沒病：有病約10：1)，直觀預測沒病也有10/11的準確率，造成選用AUC以外的指標較無意義。

04

介紹最終建議方案



(4-1) 特徵工程 - chi-squre檢視欄位跟target HeartDisease的顯著性:

未處理, 17個欄位

col_order	col_name	X-squared	df	p-value
2	BMI	6724.613578	3603	2.49E-192
3	Smoking	3713.815575	1	0
4	AlcoholDrinking	329.1041963	1	1.51E-73
5	Stroke	12390.18061	1	0
6	PhysicalHealth	9735.616088	30	0
7	MentalHealth	971.4189962	30	5.50E-185
8	DiffWalking	12953.23319	1	0
9	Sex	1568.808198	1	0
10	AgeCategory	19299.92039	12	0
11	Race	844.314886	5	2.99E-180
12	Diabetic	10959.86128	3	0
13	PhysicalActivity	3199.864826	1	0
14	GenHealth	21542.17736	4	0
15	SleepTime	2303.946242	23	0
16	Asthma	549.2855397	1	1.80E-121
17	KidneyDisease	6741.981347	1	0
18	SkinCancer	2784.787544	1	0

前處理後, 29個欄位

col_order	col_name	X-squared	df	p-value
2	BMI	6210.709	3603	7.40E-143
3	Smoking	3296.317	1	0
4	AlcoholDrinking	397.3195	1	2.11E-88
5	Stroke	11433.4	1	0
6	PhysicalHealth	8597.503	30	0
7	MentalHealth	1065.589	30	7.13E-205
8	DiffWalking	11640.48	1	0
9	Sex	1671.667	1	0
10	AgeCategory	18912.37	12	0
11	PhysicalActivity	2643.152	1	0
12	SleepTime	1901.084	23	0
13	Asthma	386.3422	1	5.18E-86
14	KidneyDisease	6141.505	1	0
15	SkinCancer	2479.035	1	0
16	Race_American_Indian_Alaskan_Native	12.66774	1	0.000372019
17	Race_Asian	325.408	1	9.62E-73
18	Race_Black	63.58335	1	1.54E-15
19	Race_Hispanic	499.2888	1	1.36E-110
20	Race_Other	11.13859	1	0.0008455
21	Race_White	721.2408	1	7.19E-159
22	Diabetic_No	8310.73	1	0
23	Diabetic_No_borderline_diabetes	57.39683	1	3.56E-14
24	Diabetic_Yes	9658.298	1	0
25	Diabetic_Yes_during_pregnancy	72.56368	1	1.62E-17
26	GenHealth_Excellent	3867.477	1	0
27	GenHealth_Fair	6192.675	1	0
28	GenHealth_Good	304.1095	1	4.19E-68
29	GenHealth_Poor	8971.392	1	0
30	GenHealth_Very_Good	3049.853	1	0

- pseudo code : chisq.test(dataset$HeartDisease, dataset[,i], correct=FALSE)
- **p-value < 0.05, reject H0**, 所有欄位都與HeartDisease有相關(dependent)

(4–1) 特徵工程 – 借助glm，得出欄位重要性、共線性：

pseudo code：

(1) preprocess(dataset)。

(2) suffle(dataset)。

(3) 取平衡(dataset[100000,])，去除na，

得出45,000筆，36000沒病：9000有病。

(4) model <- glm(HeartDisease~.), family="binomial"

, data=dataset)

欄位重要性，取 >4，13個欄位

> caret::varImp(model)

	Overall
BMI	**4.4294187**
Smoking	**13.0590904**
AlcoholDrinking	1.9100625
Stroke	**23.3059047**
PhysicalHealth	1.5127184
MentalHealth	2.1510181
DiffWalking	**6.5815495**
Sex	**25.4474306**
AgeCategory	**47.6255661**
PhysicalActivity	1.3458511
SleepTime	2.1629561
Asthma	**7.2590399**
KidneyDisease	**9.7363807**
SkinCancer	1.7886560
Race_American_Indian_Alaskan_Native	0.4889626
Race_Asian	3.5278293
Race_Black	3.8891932
Race_Hispanic	**4.0550250**
Race_Other	1.3882326
Diabetic_No	1.3356281
Diabetic_No_borderline_diabetes	1.0085721
Diabetic_Yes	1.2487449
GenHealth_Excellent	**7.0720475**
GenHealth_Fair	**21.7952554**
GenHealth_Good	**14.6508521**
GenHealth_Poor	**20.6333198**

13個欄位的VIF*數值皆小*，表示無共線性。

因此，**決定選取此13個欄位**。

> car::vif(model)

BMI	Smoking	Stroke	DiffWalking
1.108250	1.031146	1.021945	1.297409
Sex	**AgeCategory**	**Asthma**	**KidneyDisease**
1.059703	1.104021	1.053452	1.036308
Race_Hispanic	**GenHealth_Excellent**	**GenHealth_Fair**	**GenHealth_Good**
1.024754	1.229081	1.623939	1.585377
GenHealth_Poor			
1.406402			

(4–2) 建議模型 – XGBoost_1_參數與演算法選擇

(1) 參數說明

1. max_depth：樹的最大深度，使用max_depth=6。
   目的：數值愈大模型擬合度越高。
2. eta：又稱為learning_rate，使用預設值0.3。
   目的：此參數用於防止over fitting。
3. eval_metric = error： 此為二進制分類錯誤率，預設使用0.5來判斷。
   目的：評估每回迭代的分類效果，公式「#(wrong cases) 除以 #(all cases)」

(2) 演算法選擇

1. binary:logistic： 羅吉斯回歸，model對每一筆預測資料輸出機率

   考量實務面，選用logistic。理由：使用閥值(threshod)、產生ROC圖。

2. binary:hinge: 二元分類中使用hinge loss作為loss function進行分類，
   model對每一筆預測資料輸出2元結果(1, 0)

(4–2) 建議模型 – XGBoost_2_程式碼與使用流程

資料前處理
one-hot encoding
Smote轉換

選擇13 Feature
轉換dataset成
DMatrix格式

XGBoost Model

train_matrix_selected	External pointer of class 'xgb.DMatrix'

```
> print(train_matrix_selected)
xgb.DMatrix  dim: 95855 x 13  info: label  colnames: yes
```

DMatrix 是 XGBoost 使用的內部資料結構，它會優化內存效率和訓練速度。

```r
# select 13 features
# "BMI+Smoking+Stroke+DiffWalking+Sex+AgeCategory+Asthma+KidneyDisease+Race_Hispanic+GenHealth_Excellent+GenHealth_Fair
# +GenHealth_Good+GenHealth_Poor"      You, 1 second ago • Uncommitted changes
features_selected <- c("BMI","Smoking","Stroke","DiffWalking","Sex","AgeCategory","Asthma",
"KidneyDisease","Race_Hispanic","GenHealth_Excellent","GenHealth_Fair","GenHealth_Good","GenHealth_Poor")
# features_selected <- c(1, 2, 4, 7, 8, 9, )
train_features_selected <- train_df[, features_selected]
test_features_selected <- test_df[, features_selected]
train_features <- as.matrix(train_features_selected)
train_labels <- as.matrix(factor(train_df[, 1]))

test_features <- as.matrix(test_features_selected)
test_labels <- as.matrix(test_df[, 1])

train_matrix_selected <- xgb.DMatrix(data = train_features, label = train_labels)
test_matrix_selected <- xgb.DMatrix(data = test_features)
```

```r
params <- list(
  "objective" = "binary:logistic",
  "eval_metric" = "error",
  "max_depth" = 6,
  "eta" = 0.3,
  "nthread" = 4
)

model_xgboost_selected <- xgb.train(params = params, data = train_matrix_selected, nrounds = 100)
```

(4-2) 建議模型 – XGBoost_3_threshold(閥值)選取方式

	threshold	sensitivity	specificity	F1
1	0.00	1.000000000	0.0000000	0.163362620
2	0.05	0.960004968	0.4110278	0.240221293
3	0.10	0.916532108	0.5503129	0.281035172
4	0.15	0.852937523	0.6491317	0.313189820
5	0.20	0.788970314	0.7198414	0.338719138
6	0.25	0.719165321	0.7789823	0.361118907
7	0.30	0.639920507	0.8252450	0.373144057
8	0.35	0.566761893	0.8646682	0.383864726
9	0.40	0.480934045	0.8972521	0.379682291
10	0.45	0.411377469	0.9216749	0.371675457
11	0.50	0.341696684	0.9423506	0.353689895
12	0.55	0.269531735	0.9600432	0.321100917
13	0.60	0.212520184	0.9721333	0.283747927
14	0.65	0.156253882	0.9815071	0.232231863
15	0.70	0.109054776	0.9886253	0.177966961
16	0.75	0.073903863	0.9934881	0.129587281
17	0.80	0.043721277	0.9966531	0.081115336
18	0.85	0.022233263	0.9985569	0.042879387
19	0.90	0.007949323	0.9995634	0.015703595
20	0.95	0.001490498	0.9999272	0.002974346

[1] "Max F1 0.383864726171448 at threshold 0.35"

1. 採用13個Feature，基於binary:logistic搭配不同threshold，產生三種評估指標。
2. 已知資料集是不平衡(沒病:有病約10:1)，**成功檢測出患者有病才是重要的。**
3. 若一味降低threshold提高Precision，會擴大沒病者判定為有病的情況，此是不樂見的。

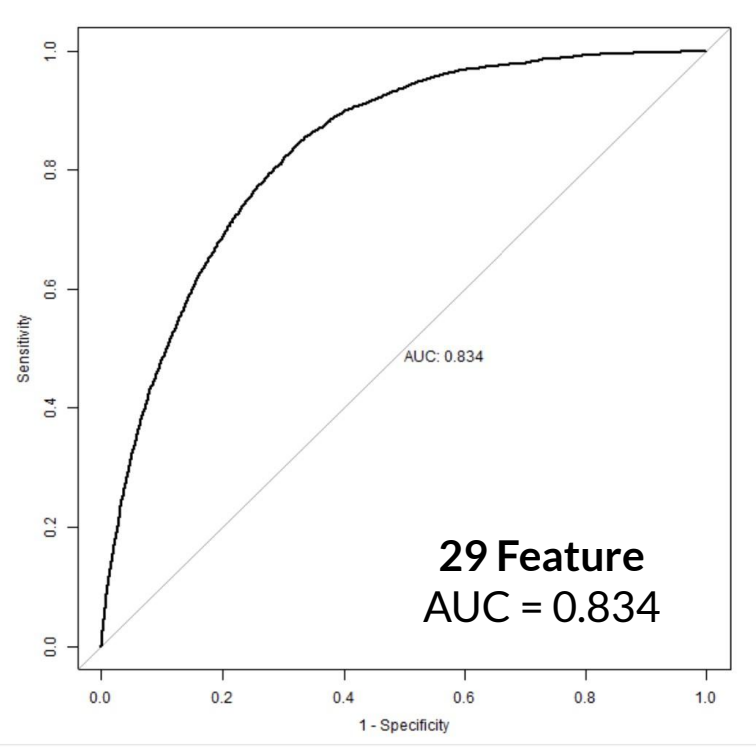
因此以綜合考量Precision and Recall的F1，

在F1有最大值 = 0.3838，採取threshold = 0.35進行建模。

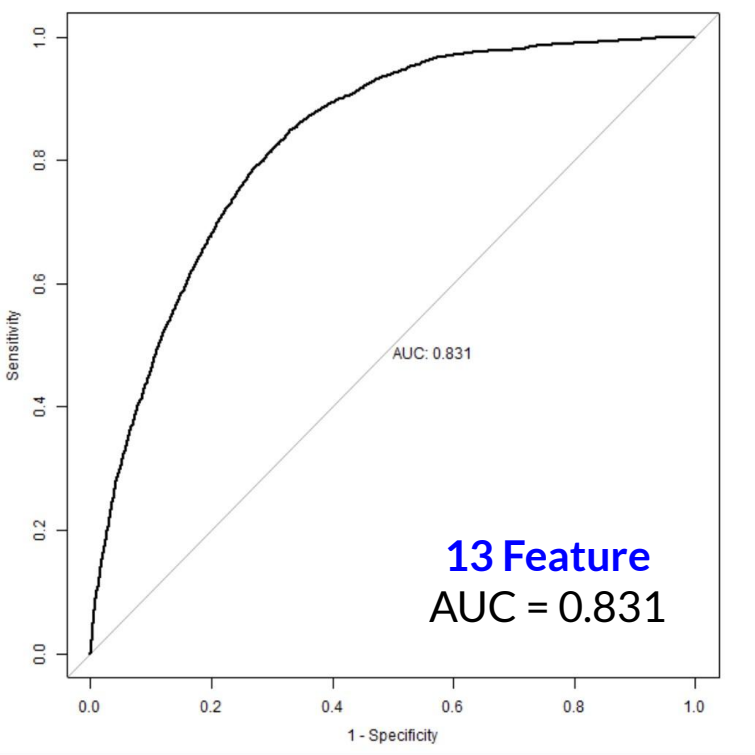
The F1 score

$$\frac{2}{\frac{1}{precision}+\frac{1}{recall}} = \frac{2*precision*recall}{precision+recall} = \frac{2*TP}{2*TP+FP+FN}$$

(4–2) 建議模型 – XGBoost_4_建模後對test資料集的預測結果



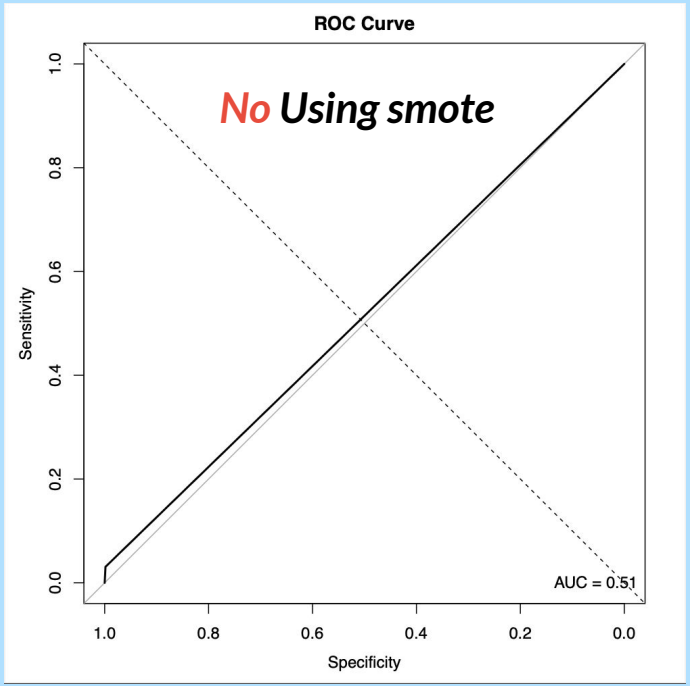
29 Feature
AUC = 0.834



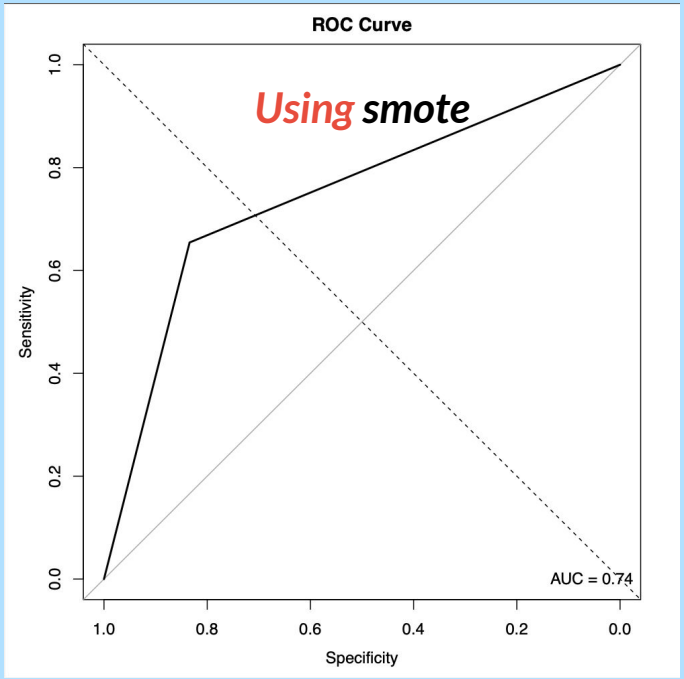
13 Feature
AUC = 0.831

test資料集是不平衡(沒病:有病約10:1)

(4–2) 建議模型 – 隨機森林_1_AUC變化_
使用smote處理不平衡資料集的必要性



*train 資料集是**不平衡**(沒病:有病約10:1)*



*train 資料集是**平衡**(沒病:有病約1.3:1)*

29 Feature
AUC = 0.51

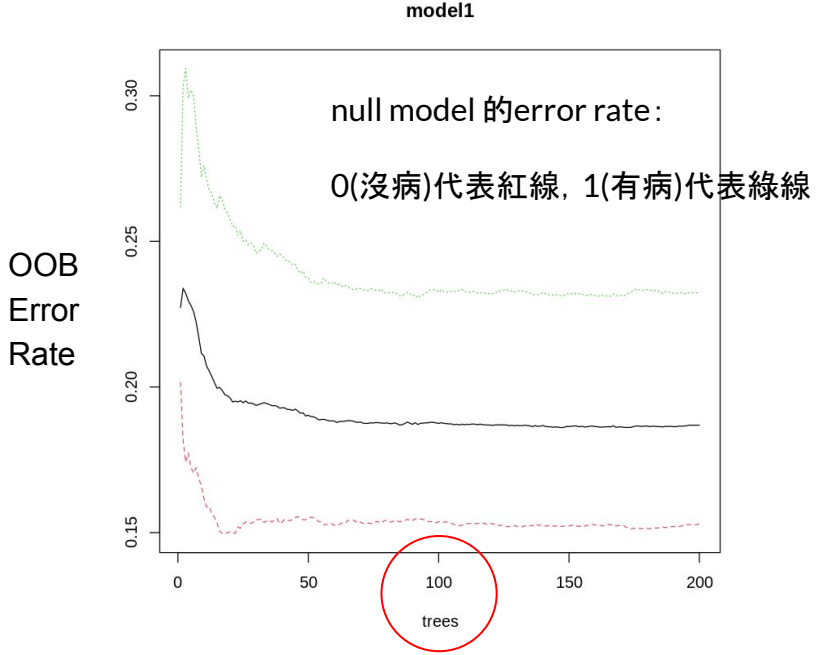
→

29 Feature
AUC = 0.74

(4–2) 建議模型 – 隨機森林_2_超參數數值設定_1_圖示

ntree：樹的數量

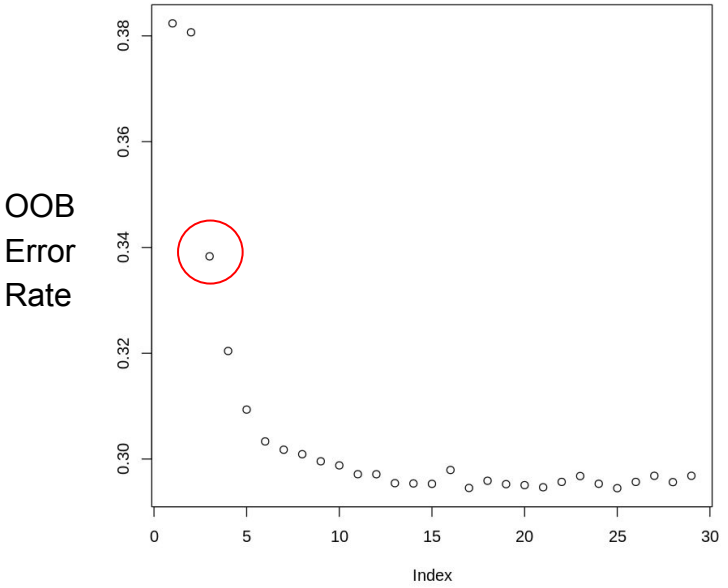
參數預設是500,，數據觀察在100左右處於穩定



mtry：每棵樹使用的特徵數量

參數預設為sqrt(**29 Feature**)，也就是5左右；

等於3時，OOB Error Rate**降幅**最大，因此設定為 3。



(out of bag)OOB：random forest 採樣時，沒被採樣到的資料。

利用沒被選到的資料進行模型的 **驗證評估**，得出OOB error rate = 1 - (OOB data猜對數量 / (OOB data全部數量))

(4–2) 建議模型 – 隨機森林_2_超參數數值設定_2_數據佐證

ntree：樹的數量

0、1 欄位為null model 的error rate，

0代表上一張投影片的紅線，1代表綠線

1	ntree	OOB	0	1
90	89	0.1875289	0.1541984	0.2319695
91	90	0.187171	0.1540679	0.2313085
92	91	0.1877302	0.1547594	0.2316912
93	92	0.1871263	0.1545637	0.2305431
94	93	0.1875065	0.1546811	0.2312737
95	94	0.1875885	0.1545246	0.2316738
96	95	0.1876705	0.1542897	0.2321783
97	96	0.1876854	0.1537287	0.2329611
98	97	0.1878942	0.1537548	0.2334134
99	98	0.1878569	0.1537157	0.2333786
100	99	0.1876631	0.1533765	0.2333786
101	100	0.1874842	0.1537287	0.2324914
102	101	0.1877227	0.1534548	0.2334134
103	102	0.187514	0.1536244	0.2327001
104	103	0.1874171	0.1536374	0.2324566
105	104	0.1873201	0.1532721	0.2327175
106	105	0.1873127	0.1531677	0.2328393
107	106	0.1870219	0.1528024	0.2326479
108	107	0.1872083	0.1527241	0.2331872
109	108	0.1869399	0.1523327	0.2330828
110	109	0.1872381	0.1525806	0.2334481

mtry：每棵樹使用的特徵數量

2	mtry	error rate
12	1	0.3823684
13	2	0.3806774
14	3	0.3383198
15	4	0.3204217
16	5	0.3093427
17	6	0.3033233
18	7	0.301757
19	8	0.3009073
20	9	0.2995667
21	10	0.2987895
22	11	0.2971263
23	12	0.2971186
24	13	0.2954294
25	14	0.2953758
26	15	0.2953032
27	16	0.2979234
28	17	0.2945221

2	mtry	error rate
29	18	0.2958868
30	19	0.2952444
31	20	0.2950603
32	21	0.2946573
33	22	0.2956707
34	23	0.2967743
35	24	0.2953179
36	25	0.2944851
37	26	0.2956685

虛擬碼：

```
model <- randomForest(as.factor(HeartDisease)~.
,data=train_data, ntree=100, mtry=i)
```

更多數值 https://docs.google.com/spreadsheets/d/1IM2-4hOuFyP3FJt-YIv7QdDRmdFKmODZmsDmDpPo974/edit#gid=1549750515

(4–2) 建議模型 – 隨機森林_3_程式碼

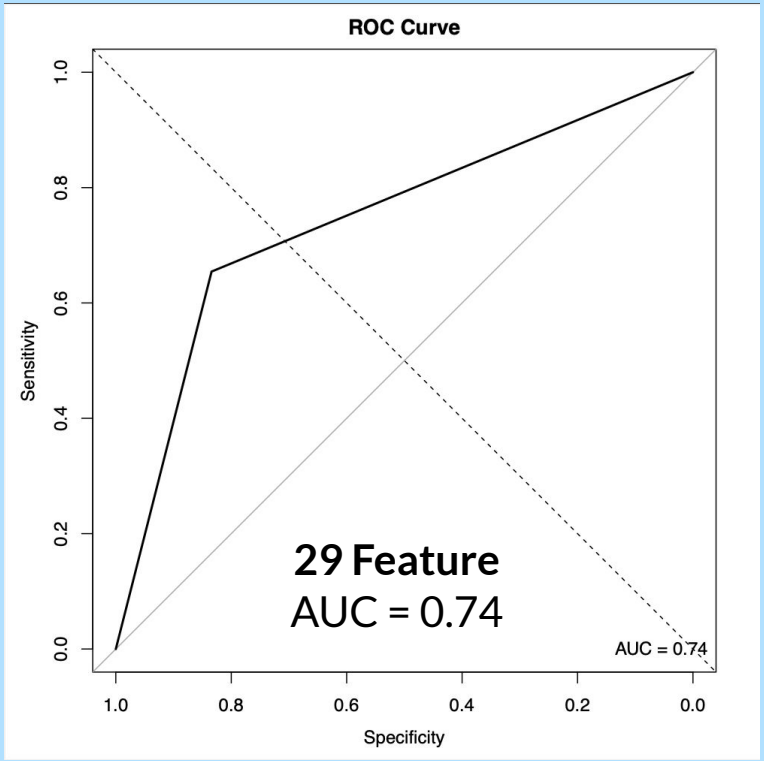
選用13個特徵:

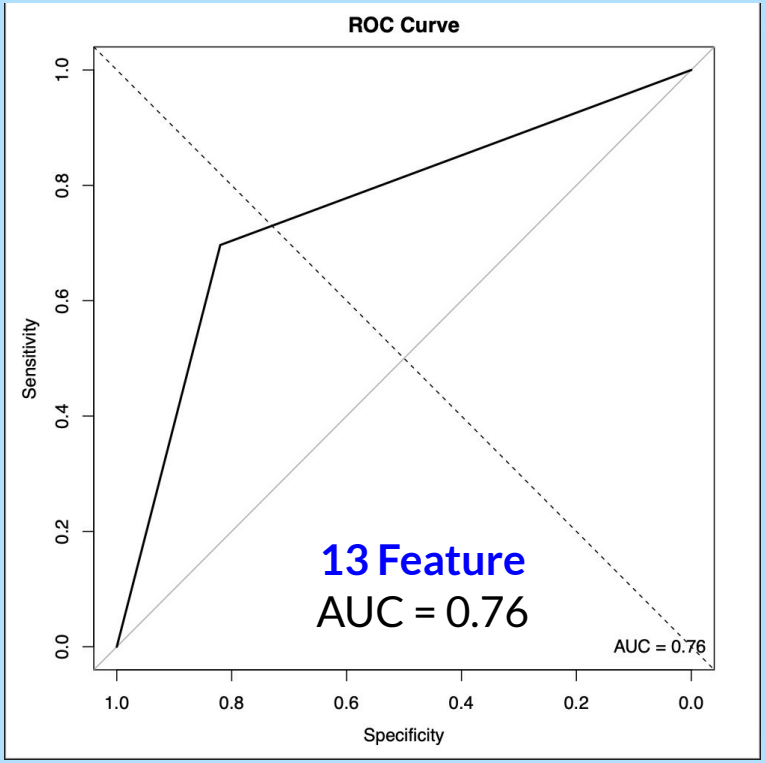
```
train.index <- createDataPartition(df$HeartDisease, p = .7, list = FALSE)
train_df <- df[train.index,]
train_df <- smote(HeartDisease ~ ., data = train_df)
train_df <- na.omit(train_df)
selected_features = c('HeartDisease', 'BMI', 'Smoking', 'Stroke',
 'DiffWalking', 'Sex', 'AgeCategory', 'Asthma', 'KidneyDisease',
 'Race_Hispanic', 'GenHealth_Excellent', 'GenHealth_Good', 'GenHealth_Fair', 'GenHealth_Poor')
train_df <- train_df[, selected_features]
train_df$HeartDisease <- as.factor(train_df$HeartDisease)
```

建模:

```
model <- randomForest(HeartDisease ~ ., data=train_df, ntree=100, mtry=3, importance=TRUE)
```

(4–2) 建議模型 – 隨機森林_4_建模後對test資料集的預測結果





test資料集是不平衡(沒病:有病約10:1)

(4–3) 再次重申評估與結論

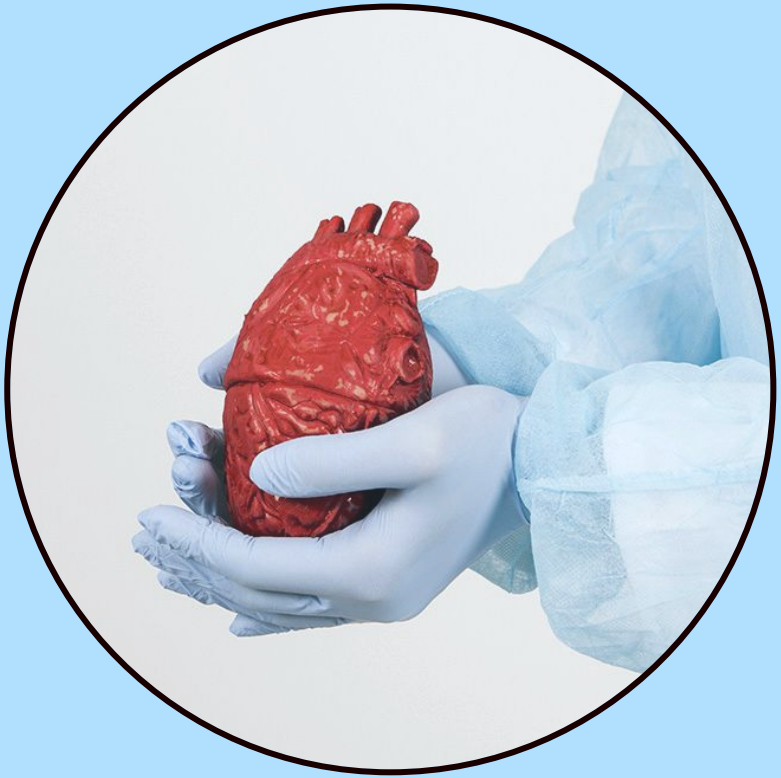
透過兩者的比對，得出分析價值是：(1) 確定**無相關**的特徵**其實是重要的**；
(2)得出不具有共線性效果的特徵，既**維持效能**又**提升**建立模型的**效率**。

原始資料集觀察			
統計直觀	Feature	No, 有病%	Yes, 有病%
有相關	**Smoking** **Stroke** **AgeCategory** **KidneyDisease** **GenHeath** **DiffWalking**	7.47 7.47 (50-54)5.45 7.77 (Excellent)2.24 6.3	36.4 36.4 22.6(80 up) 29.3 34.1(Poor) 22.6
無相關	**BMI** **Sex(Gender)** **Asthma** **Race**	27.26 10.6(Male) 8.1 3.3(Asian)	28.34 6.69(Female) 11.5 10.4(Indian)

特徵工程		
Feature	caret::varImp (glm_model)	car::vif (glm_model)
Race_Hispanic	4.0550250	1.024754
BMI	4.4294187	1.108250
DiffWalking	6.5815495	1.297409
GenHealth_Excellent	7.0720475	1.229081
Asthma	7.2590399	1.053452
KidneyDisease	9.7363807	1.036308
Smoking	13.0590904	1.031146
GenHealth_Good	14.6508521	1.585377
GenHealth_Poor	20.6333198	1.406402
GenHealth_Fair	21.7952554	1.623939
Stroke	23.3059047	1.021945
Sex(Gender)	25.4474306	1.059703
AgeCategory	47.6255661	1.104021
備註：glm的欄位重要性，取>4，13個欄位；VIF值皆 < 2		

05

專案艱辛之處



專案艱辛之處

昇豐:從**「思想的巨人，行動的侏儒」中脫離，進行行動力變革**。讀過不少ML、DS、統計的書，但工作既不是此領域、大學的統計也沒學好，因此這段期末報告的準備過程真的是練習對抗的過程;所幸考試後的課程會提點分析工具選用的訣竅外，老師也願意給予意見。組員也願意與信任地按照我仿照課堂教科書訂定的報告綱要準備。

書瑋:**第一次接觸資料分析，也是第一次接觸R**，都是透過上課及作業學到的東西，加上不斷地Google及詢問ChatGPT，慢慢拼湊出最後的結果。

元亨:(程式的互相合作)我們總共採用了3種(決策樹、隨機森林、Xgboost)模型來分析資料，實際寫了5份建模程式及1份視覺化程式，在這些Code中，每個人的Code pattern, 前處理, 套件選用等都不盡相同，**讓我們在溝通、整合程式時費了一番功夫**。

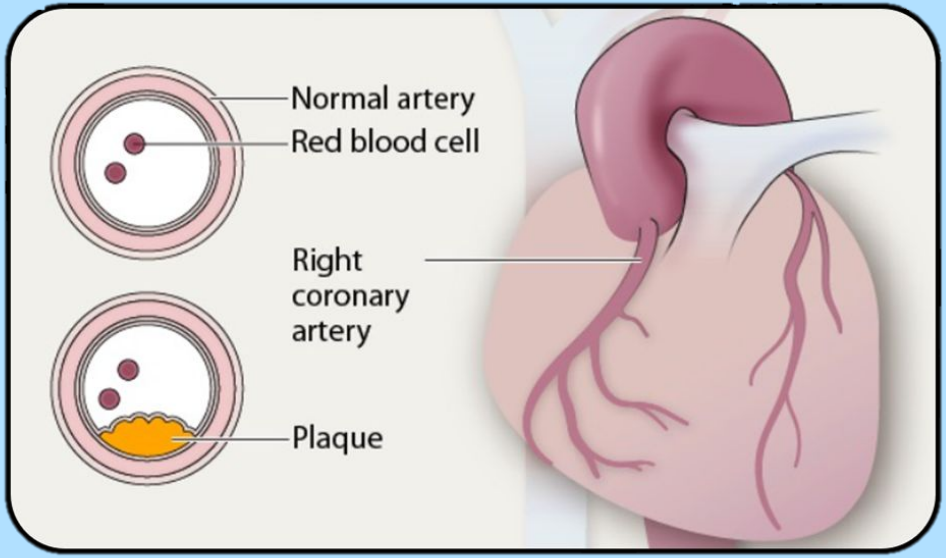
正晏:(對於不熟悉領域的探索)平時在於工作中並未使用到與Data Science相關的技能，本次專案算是一個新領域的探索，在專案建立的過程中，除了需要時不時複習老師上課的內容之外，還需要不斷地去學習，在資料集確定後就遇到了第一個難關，在不同的模型上，會針對其所需要的輸入去調整Attribute，在建模的過程中，**發現若不了解模型的原理，在參數的值上也難以去調整**。

宗佑:(資料集不平衡)本主題資料集相當不平均，患者與非患者為1比10，基本上使用null model全部猜測患者無病，準確性高達九成多。然而這樣的猜測沒有太大意義，因此我們**針對不平衡的資料瀏覽許多資料，最後透過mote，降低樣本比例至1比4，提高sensitivity**。

弘軒:同學間的默契在磨合後增長，討論凝聚眾人智慧，分派任務考驗彼此信賴;專案合作的經驗啟發團隊合作，善用工具實現共同目標迎接各種挑戰;至此**深刻體會到老師設計本門課程的價值與其重要性**。

06

DEMO



DEMO 說明

(1) Rscript語法

0. cd to code folder.
1. 產生資料分析圖示

Rscript data_plot.R

2. 執行主程式

Rscript main.R --d 2 --m 2

Rscript main.R --d 2 --m 3

(2)參數說明

--data_source or --d

1：讀取原資料，會處理dummy和smote

2：讀取已處理資料

--model or --m

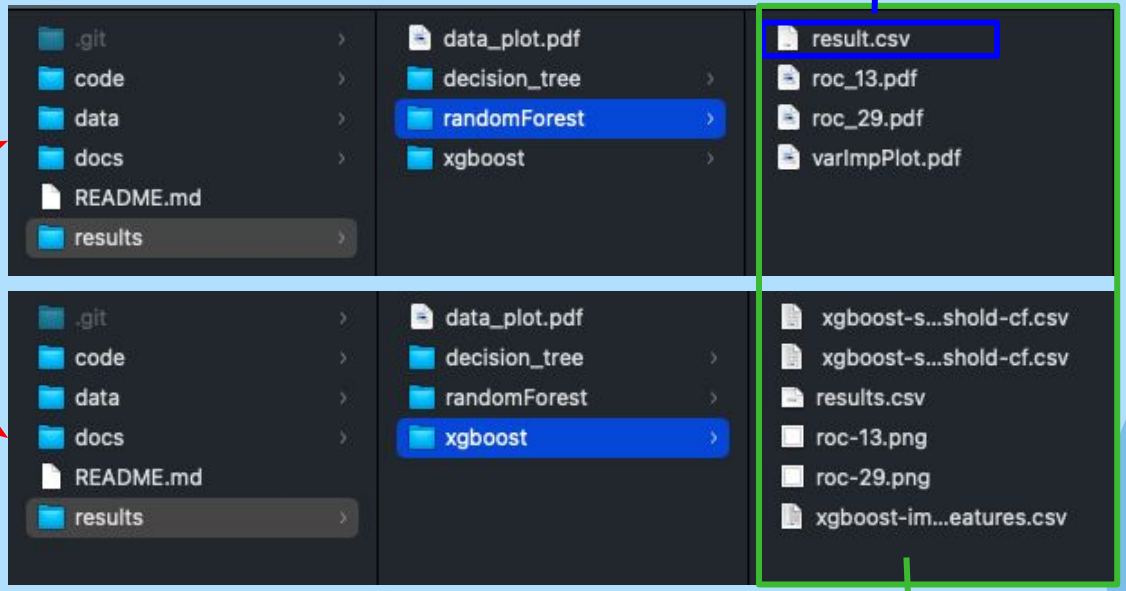
1：decision tree

(不支援data_source = 2，且會耗時3分鐘以上)

2：random forest

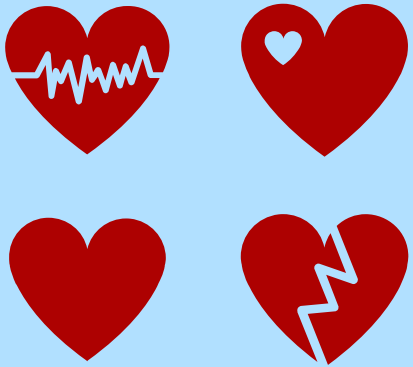
3：xgboost

Feature	Accuracy	Precision	Sensitivity	Specificity	F1.Score	AUC
29Features	0.9	0.41	0.24	0.97	0.31	0.6
13Features	0.9	0.41	0.28	0.96	0.33	0.62



隨機森林、XGBoost產生檔案交集：

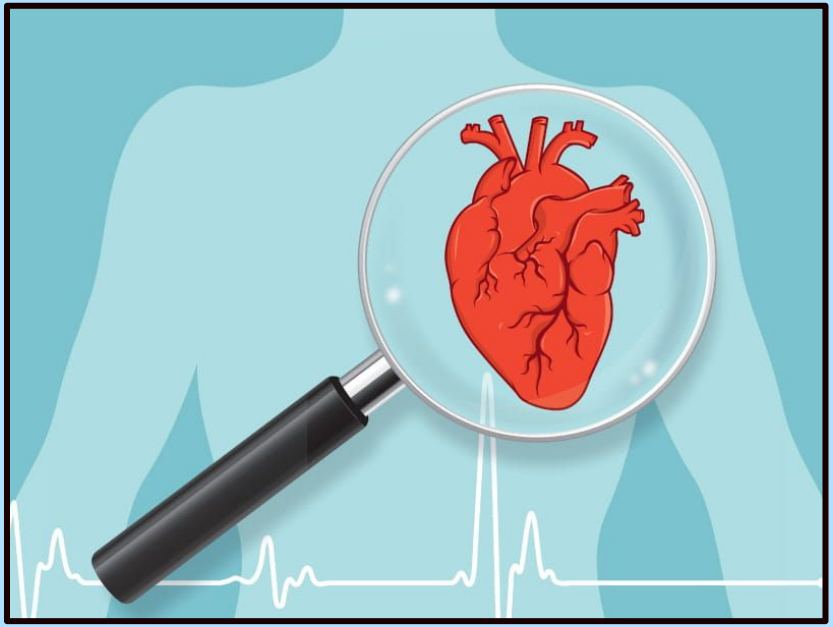
1. result.csv：模型預測評估結果。
2. roc-13、roc-29：不同Freature的AUC表現。
3. ImportantFeature：兩個模型產生的欄位重要性。



Thank you

07

專案引用的套件與外部資料



引用套件、外部資料套件

引用套件:

#packages
#1. rpart / rpart.plot : decision tress
#2. corrplot : correlation among cols
#3. performanceEstimation : smote to tackle unbalanced dataset
#4. dplyr : tackele dataframe
#5. ROCR : roc curve and aus
#6. caret::varImp : the important order of cols for target y
#7. car::vif : check collinearity in cols

```
# ------------installed packages---------------------
# install.packages('rpart.plot')
# install.packages('ggplot2')
# install.packages('png')
# install.packages('xgboost')
# install.packages('lightgbm')
```

外部資料:

kaggle dataset

(example)Hux.DA5030.Project | Kaggle
(example)STAT 451 Project | Kaggle
(example_explaindata)Heart Disease Scoring : Who is dangerous 🤔? | Kaggle
(example_explaindata)Heart Disease Prediction | Kaggle

smote

(performanceEstimation)imbalanced data - package to do SMOTE in R - Stack Overflow

VIF

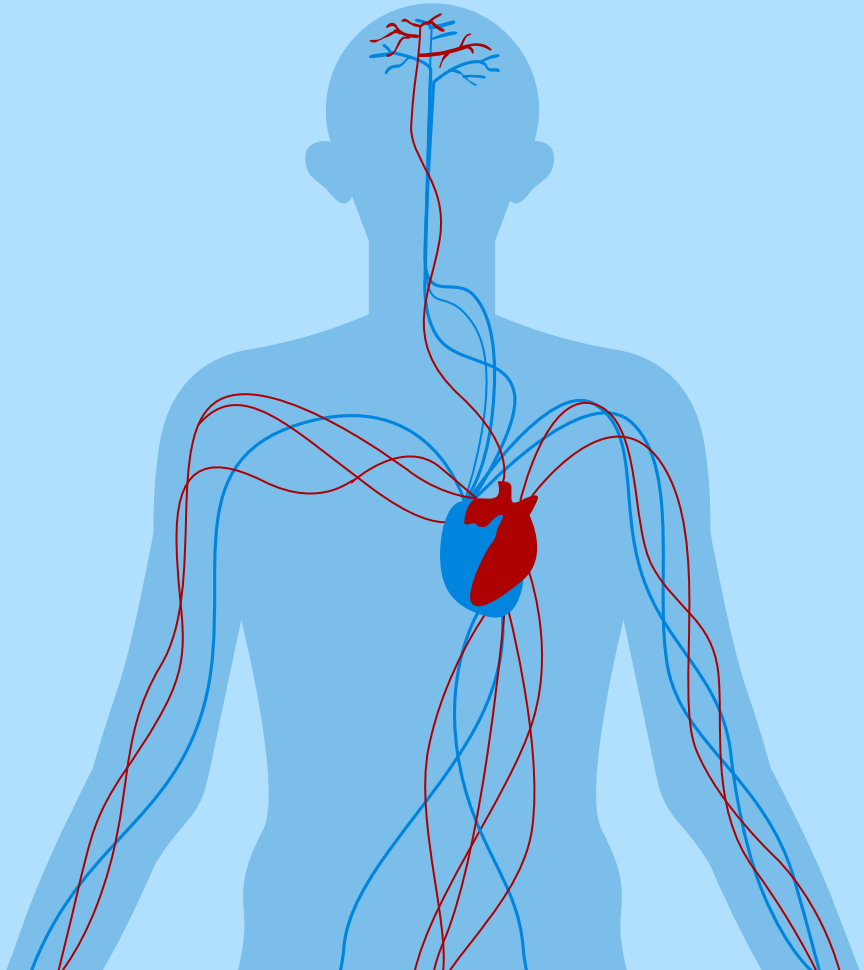
(glm_Variable Importance_VIF)How to Perform Logistic Regression in R (Step-by-Step) - Statology
(multicollinearity)How to Fix in R: there are aliased coefficients in the model - Statology

AUC

Interpreting ROC Curve and ROC AUC for Classification Evaluation | by Vinícius Trevisan | Towards Data Science

08

其他附件



專案艱辛之處 - 昇豐

(1) 範例：**專案甘苦談與專案引用的套件與外部資料** 階層式綱要如下：

1. 專案甘苦談

(1) 從「思想的巨人，行動的侏儒」中脫離，進行行動力變革。讀過不少ML、DS、統計的書，但工作既不是此領域、大學的統計也沒學好，因此這段期末報告的準備過程真的是練習對抗的過程；所幸考試後的課程會提點分析工具選用的訣竅外，老師也願意給予意見。組員也願意與信任地按照我仿照課堂教科書訂定的報告綱要準備。

(2) 期末主題是連續5個禮拜與老師討論後才決定。除了了解自己對資料分析的選題眼界不成熟之外，也理解DS專家看待資料集值得分析的考量角度。

(3) 會議討論都記載於google doc。雖然讓繁忙的組員感到煩，但卻也是累積、連結討論成果最有效的方式。從2023/04/21寄信組隊、4/27第一次討論開始，google doc、google ppt就是匯集共識、討論程式效果的地方。

專案艱辛之處 - 元亨

1. 資料的不平衡:
   我們的資料集總共有30萬筆資料，包含約21萬筆健康人資料和2萬筆病患資料，在這種不平衡的情況下，Null Model傾向將資料分類為健康，導致準確率很高，但Specificity(True Negitive Rage)和Sensitivity(True Positive Rate)卻極低。因此我們使用了smote採樣平衡了健康人及病患的資料。一開始，我們也不知道該先切分Training/Testing還是先smote，後來請教了老師後，才得知有很多人都在這步驟錯誤地先smote才切資料集。
2. 程式的互相合作:
   我們總共採用了3種(決策數、隨機森林、Xgboost)模型來分析資料，實際寫了5份建模程式及1份視覺化程式，在這些Code中，每個人的Code pattern, 前處理, 套件選用等都不盡相同，讓我們在溝通、整合程式時費了一番功夫。

引用資料:

[RandomForest in R](https://ithelp.ithome.com.tw/articles/10303882?sc=iThelpR)

專案艱辛之處 - 書瑋

1. 第一次接觸資料分析，也是第一次接觸R，都是透過上課及作業學到的東西，加上不斷地Google及詢問ChatGPT，慢慢拼湊出最後的結果。
2. 在對不平橫資料集的處理上，以及重要特徵的挑選上，因為不熟悉的關係，導致花了非常多的時間去嘗試。

引用資料

https://rpubs.com/jiankaiwang/rf
https://blog.csdn.net/ybdesire/article/details/120375089
https://towardsdatascience.com/what-is-out-of-bag-oob-score-in-random-forest-a7fa23d710

專案艱辛之處 - 書瑋

由於小組成員並無人從事資料科學相關行業，前期只能由小組成員一步一步針對老師上課內容互相討論。

本課題遇到最大兩個困難處為

1.資料集不平衡

本主題資料集相當不平均，患者與非患者為1比13，基本上使用null model全部猜測患者無病，準確性即高達九成多。然而這樣的猜測沒有太大意義，因此我們針對不平衡的資料瀏覽許多資料，最後透過smote，降低樣本比例至1比4，提高sensitivity。

2.特徵挑選

原本資料集有18個欄位，但有些特徵的數值為種類，例如膚色.....，因此先進行轉換，轉換後欄位高達29個。為此我們思考是否有些欄位重要較低，因此使用不同方式進行評估，最後挑選了13個欄位，其最終auc結果與29個欄位相差不多。

資料前處理完成後，彼此分工針對不同模型進行測試，由於每個人撰寫程式碼的習慣不同，因此整合階段也額外花費許多時間。分組報告並非單獨個體，必須透由成員彼此討論、分工合作，才能有效解決問題。如果無法融入群體，只會拖累彼此，阻礙進度。所幸本小組最後還是非常感謝大家積極參與、熱烈討論，順利完成本次課題，令我不只在學科上獲得知識，也在人際互動受益良多。