

程式碼語言:python

內容:

1. 資料清理與視覺化圖表 (參考程式碼 Part 1)

- head()方法來檢查前幾項數據
- info() 檢查有沒有缺值與資料類別
- shape() 查看整筆數據欄位與行數
-

根據輸出結果如下，我們可以得到以下資訊：

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

(918, 12)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Age             918 non-null   int64
1   Sex             918 non-null   object
2   ChestPainType   918 non-null   object
3   RestingBP       918 non-null   int64
4   Cholesterol     918 non-null   int64
5   FastingBS       918 non-null   int64
6   RestingECG      918 non-null   object
7   MaxHR           918 non-null   int64
8   ExerciseAngina  918 non-null   object
9   Oldpeak         918 non-null   float64
10  ST_Slope        918 non-null   object
11  HeartDisease    918 non-null   int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

HeartDisease	
0	410
1	508

data.head() 顯示了資料框的前五行。每一列代表一個觀測值，每一欄代表一個特徵。

data.shape 顯示資料框的形狀，有 918 行和 12 列。

data.info() 提供了資料框的摘要資訊。每個欄位的名稱、非空值個數以及資料型態都被列出。可以由結果看出每一項特徵值皆為 918，代表沒有缺失值，因此不需要去除缺失值或是補值

data.groupby('HeartDisease') 是一個分組操作，根據 'HeartDisease' 欄位將資料分組，判斷分布

由於 csv 檔中的欄位包含文字種類而不是數值，你可以使用獨熱編碼 (One-Hot Encoding) 或者對文字種類進行數值編碼來處理。

在這裡使用的是 One-Hot Encoding 的方式

```
columns_to_encode = ['Sex', 'ChestPainType',  
'RestingECG', 'ExerciseAngina', 'ST_Slope'] # 指定要編碼的欄位名稱列表  
encoded_data = pd.get_dummies(data, columns=columns_to_encode)  
print(encoded_data.head())
```

可以從結果得出已從原本的 18 欄變成 21 欄:

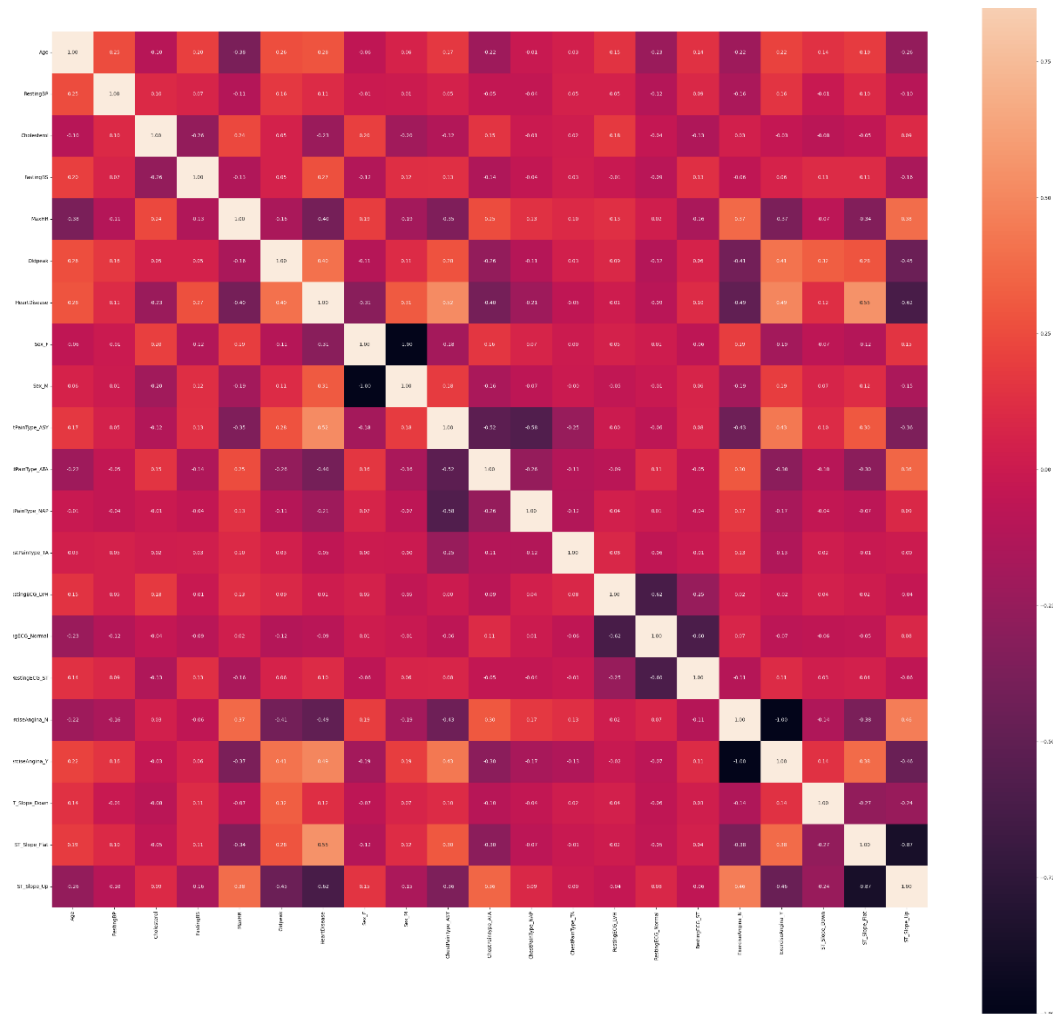
```
Age  RestingBP  Cholesterol  FastingBS  MaxHR  ...  ExerciseAngina_N  ExerciseAngina_Y  ST_Slope_Down  ST_Slope_Flat  ST_Slope_Up  
0    40         140         289         0    172  ...           1           0           0           0           1  
1    49         160         180         0    156  ...           1           0           0           1           0  
2    37         130         283         0     98  ...           1           0           0           0           1  
3    48         138         214         0    108  ...           0           1           0           1           0  
4    54         150         195         0    122  ...           1           0           0           0           1  
[5 rows x 21 columns]
```

#	Column	Non-Null Count	Dtype
0	Age	918 non-null	int64
1	RestingBP	918 non-null	int64
2	Cholesterol	918 non-null	int64
3	FastingBS	918 non-null	int64
4	MaxHR	918 non-null	int64
5	Oldpeak	918 non-null	float64
6	HeartDisease	918 non-null	int64
7	Sex_F	918 non-null	uint8
8	Sex_M	918 non-null	uint8
9	ChestPainType_ASY	918 non-null	uint8
10	ChestPainType_ATA	918 non-null	uint8
11	ChestPainType_NAP	918 non-null	uint8
12	ChestPainType_TA	918 non-null	uint8
13	RestingECG_LVH	918 non-null	uint8
14	RestingECG_Normal	918 non-null	uint8
15	RestingECG_ST	918 non-null	uint8
16	ExerciseAngina_N	918 non-null	uint8
17	ExerciseAngina_Y	918 non-null	uint8
18	ST_Slope_Down	918 non-null	uint8
19	ST_Slope_Flat	918 non-null	uint8
20	ST_Slope_Up	918 non-null	uint8

2. 敘述性統計分析(參考程式碼 Part 2)

```
3. # 使用 describe() 方法計算統計摘要  
4. statistics = encoded_data.describe()  
5.  
6. # 輸出統計摘要  
7. print(statistics)
```

3 特徵相關性分析 (參考程式碼 Part 3)



發現有正相關的欄位有:

1. ST_Slope_Flat(0.55):心電圖最高 S-T 段的斜率越大越亦有心臟病
2. ExerciseAngina(0.49): 運動誘發心絞痛越高較亦有心臟病
3. ChestPainType_ASY(0.52): 當病患為無症狀 (Asymptomatic) 胸痛類型時，心臟疾病的發生可能性較高。
4. Oldpeak (0.40): 表示當 ST 段壓低 (Oldpeak) 較高時，心臟疾病的發生可能性較高。

從以上的結果來看 ST_Slope、ExerciseAngina、ChestPainType、 Oldpeak 對心臟病的機率: (參考程式碼 Part 3 中的整理機率)

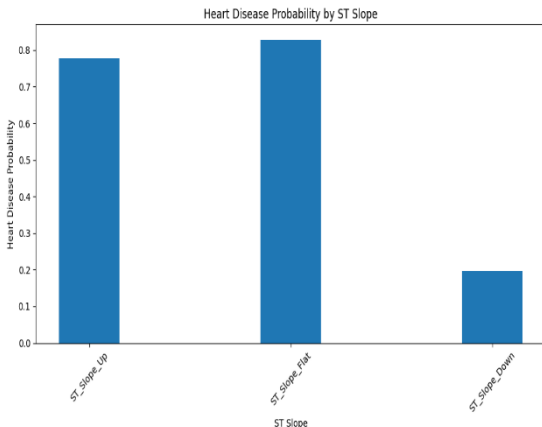
```
Oldpeak 的心臟病比率:
Oldpeak HeartDiseaseRatio
0      -2.6      1.000000
1      -2.0      1.000000
2      -1.5      1.000000
3      -1.1      0.000000
4      -1.0      1.000000
```

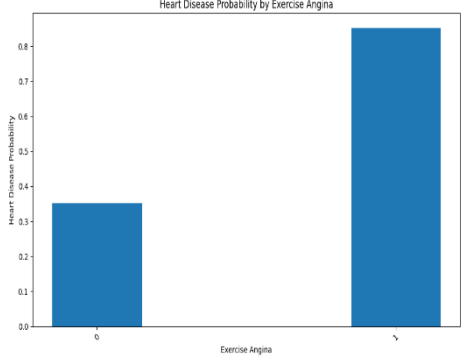
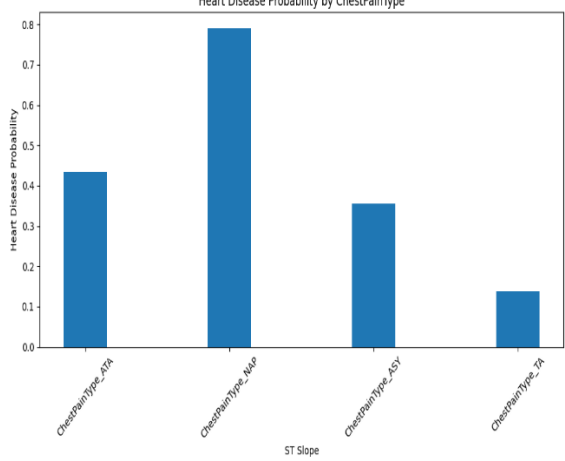
```

不同 ChestPainType 的心臟病比率:
ChestPainType_ATA ChestPainType_NAP ChestPainType_ASY ChestPainType_TA HeartDiseaseRatio
0 0 0 0 1 0.434783
1 0 0 1 0 0.790323
2 0 1 0 0 0.354680
3 1 0 0 0 0.138728
有沒有 ExerciseAngina 的心臟病比率:
ExerciseAngina_Y ExerciseAngina_N HeartDiseaseRatio
0 0 1 0.351005
1 1 0 0.851752
不同 ST_Slope 的心臟病比率:
ST_Slope_Up ST_Slope_Flat ST_Slope_Down HeartDiseaseRatio
0 0 0 1 0.777778
1 0 1 0 0.828261
2 1 0 0 0.197468
PS C:\Users\Fiona\Desktop\機器學習\HW2> & C:/Users/Fiona/AppData/Local/Programs/Python/Python37/p
習/HW2/HW2
不同 ChestPainType 的心臟病比率:
ChestPainType_ATA ChestPainType_NAP ChestPainType_ASY ChestPainType_TA HeartDiseaseRatio
0 0 0 0 1 0.434783
1 0 0 1 0 0.790323
2 0 1 0 0 0.354680
3 1 0 0 0 0.138728
有沒有 ExerciseAngina 的心臟病比率:
ExerciseAngina_Y ExerciseAngina_N HeartDiseaseRatio
0 0 1 0.351005
1 1 0 0.851752
不同 ST_Slope 的心臟病比率:
ST_Slope_Up ST_Slope_Flat ST_Slope_Down HeartDiseaseRatio
0 0 0 1 0.777778
1 0 1 0 0.828261
2 1 0 0 0.197468

```

用表格整理出來: (參考程式碼 Part 3 中的畫長條圖)

特徵值和與其有高度正 相關欄位間係數		說明	視覺化 0: 無心臟病 · 1: 有心臟病																
<table><tr><th></th><th>HeartDisease</th></tr><tr><td>ST_Slope_Up</td><td>-0.622164</td></tr><tr><td>ST_Slope_Flat</td><td>0.554134</td></tr><tr><td>ST_Slope_Down</td><td>0.122527</td></tr></table>		HeartDisease	ST_Slope_Up	-0.622164	ST_Slope_Flat	0.554134	ST_Slope_Down	0.122527	最高 S-T 段的斜率	<ul style="list-style-type: none">最高 S-T 段的斜率中途形 向下有心臟病機率為 77.77%。最高 S-T 段的斜率中途形 向上有心臟病機率為 19.74%。最高 S-T 段的斜率中途形 平坦有心臟病機率為 82.82%。	 <table><caption>Heart Disease Probability by ST Slope</caption><thead><tr><th>ST Slope</th><th>Heart Disease Probability</th></tr></thead><tbody><tr><td>ST_Slope_Up</td><td>0.7777</td></tr><tr><td>ST_Slope_Flat</td><td>0.8282</td></tr><tr><td>ST_Slope_Down</td><td>0.1974</td></tr></tbody></table>	ST Slope	Heart Disease Probability	ST_Slope_Up	0.7777	ST_Slope_Flat	0.8282	ST_Slope_Down	0.1974
	HeartDisease																		
ST_Slope_Up	-0.622164																		
ST_Slope_Flat	0.554134																		
ST_Slope_Down	0.122527																		
ST Slope	Heart Disease Probability																		
ST_Slope_Up	0.7777																		
ST_Slope_Flat	0.8282																		
ST_Slope_Down	0.1974																		

運動誘發性心絞痛	HeartDisease		◦ 有運動誘發性心絞痛的人有心臟病的機率為 85.17%
	ExerciseAngina_N	-0.494282	
	ExerciseAngina_Y	0.494282	◦ 有運動誘發性心絞痛的人有心臟病的機率為 35.10%
			
胸部疼痛類型	HeartDisease		◦ 非典型心絞痛具有心臟病的機率為 13.8%。
	ChestPainType_ATA	-0.401924	◦ 非心絞痛具有心臟病的機率為 35.46%。
	ChestPainType_NAP	-0.212964	◦ 無症狀具有心絞痛的機率為 79.03%。
	ChestPainType_ASY	0.516716	◦ 典型心絞痛具有心臟病的機率為 43.47%。
	ChestPainType_TA	-0.054790	
			

標準化數據(參考程式碼 Part 3 中的標準化數據)

以 sklearn 的資料前處理提供的 StandardScaler 進行資料標準化。為了避免後續模型訓練時特徵值大的資料欄位影響其他特徵值，將所有資料標準化；使其變異數為 0，標準差為 1。

4.資料分割與建置 4 個分類模型(1. Logistic regression、2. SVM、3. Random forest、4. KNN) (參考程式碼 Part 4)

資料分割: 資料分割，將讀出來的資料切成訓練集、驗證集與測試集劃分比例為 6:1:3

1.Logistic regression (參考程式碼 Part 4 中的 Logistic regression)

使用了一些超參數來防止擬和，比較有用跟沒有使用超參數在精準度上的不同使用的最佳超參數: {'C': 0.01, 'class_weight': None, 'solver': 'lbfgs'}
分別代表:

- C: 數值越大對 weight 的控制力越弱，預設為 1。

- solver: 優化器的選擇。newton-cg,lbfgs,liblinear,sag,saga。預設為 liblinear。
- class_weight: 若遇資料不平衡問題可以設定 balance，預設=None。

測試結果:

最佳超參數組合: {'C': 0.01, 'class_weight': None, 'solver': 'lbfgs'}

訓練集準確率: 0.8693284936479129

測試集準確率: 0.8581818181818182

沒有使用超參數訓練集準確率: 0.8729582577132486

沒有使用超參數測試集準確率: 0.8545454545454545

2.SVM(參考程式碼 Part 4 中的 SVM)

SVM 需要選擇合適的參數，如 C 值、核函數等，以避免過擬合或欠擬合。

測試結果:

訓練集準確率: 0.9038112522686026

測試集準確率: 0.9018181818181819

最佳超參數組合: {'C': 0.1, 'kernel': 'rbf'}

有參數訓練集準確率: 0.8638838475499092

有參數測試集準確率: 0.8763636363636363

3. Random forest

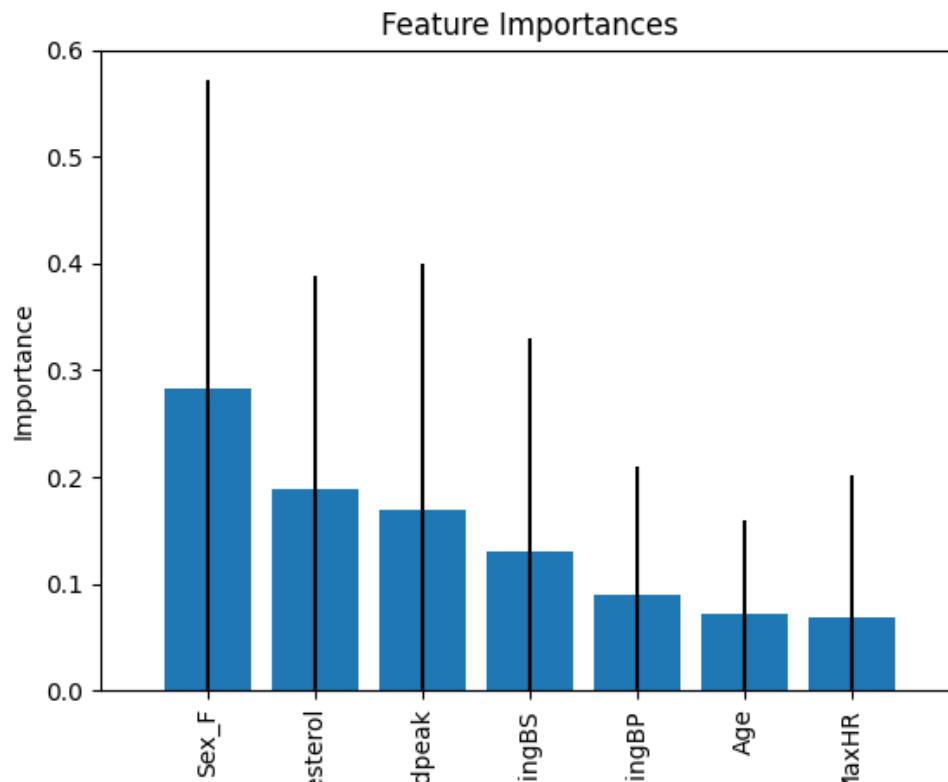
輸出結果:

使用所有特徵的分類報告：

	precision	recall	f1-score	support
-1	0.86	0.78	0.82	117
0	0.85	0.91	0.87	158
accuracy			0.85	275
macro avg	0.85	0.84	0.85	275
weighted avg	0.85	0.85	0.85	275
門檻值 = 0.05				
特徵遮罩：	[False	False	False	False
	False	False	False	True
	False	False	True	False
	True	True	False	True
	True	True	True	True

使用特徵選擇後的分類報告：

	precision	recall	f1-score	support
-1	0.82	0.71	0.76	117
0	0.80	0.89	0.84	158
accuracy			0.81	275
macro avg	0.81	0.80	0.80	275
weighted avg	0.81	0.81	0.81	275



4.KNN(參考程式碼 Part 4 中的 KNN)

訓練集精確度: 0.8929219600725953

測試集準確率: 0.8436363636363636

5.綜合比較 4 個模型的分類結果與分析討論

混淆矩陣模型評估 評估訓練出來的模型成效好不好，我們使用混淆矩陣，分辨

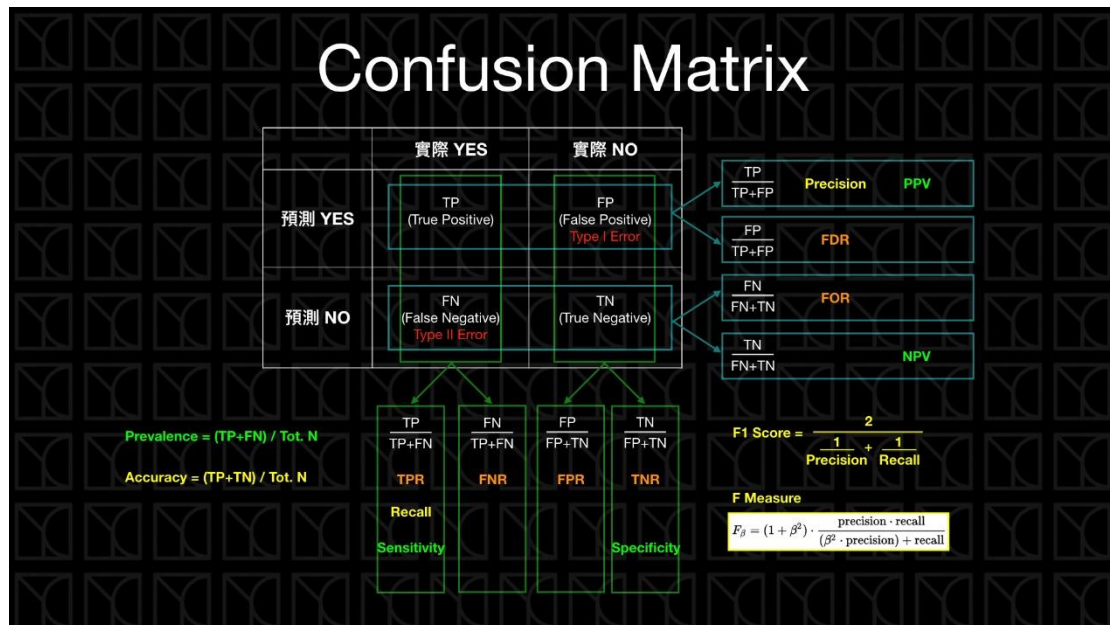
模型在分類上的準確率，有四個分類指標。

- True Positive (TP)「真陽性」:真實情況是「有」，模型說「有」的個數。
- True Negative(TN)「真陰性」:真實情況是「沒有」，模型說「沒有」的個

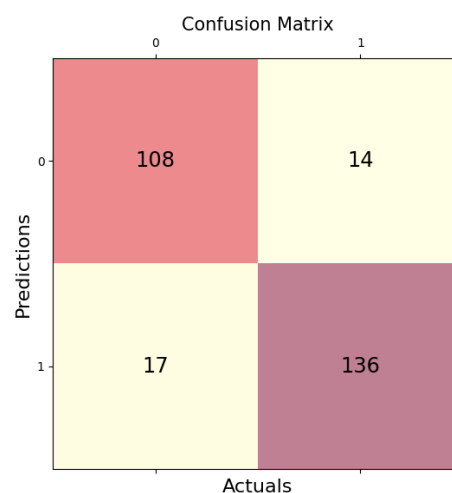
數。

◦ False Positive (FP) 「偽陽性」:真實情況是「沒有」, 模型說「有」的個數。

◦ False Negative(FN) 「偽陰性」:真實情況是「有」, 模型說「沒有」的個數。



1.Logistic regression (參考程式碼 Part 4 中的 Logistic regression 的混淆矩陣)



從這張圖我們可以看出他的 TP=109, FP=14, FN=17, TM=136, 總共得到 275

筆資料, 和實際分得的 train_data 數量一致。比較重要的是對角線的兩個數

值, 只要他有得病就要被抓出來, 因此準確率要高。

精確度 : $109 / (109 + 14) \approx 0.886$

召回率 : $109 / (109 + 17) \approx 0.865$

準確率 : $(109 + 136) / (109 + 14 + 17 + 136) \approx 0.891$

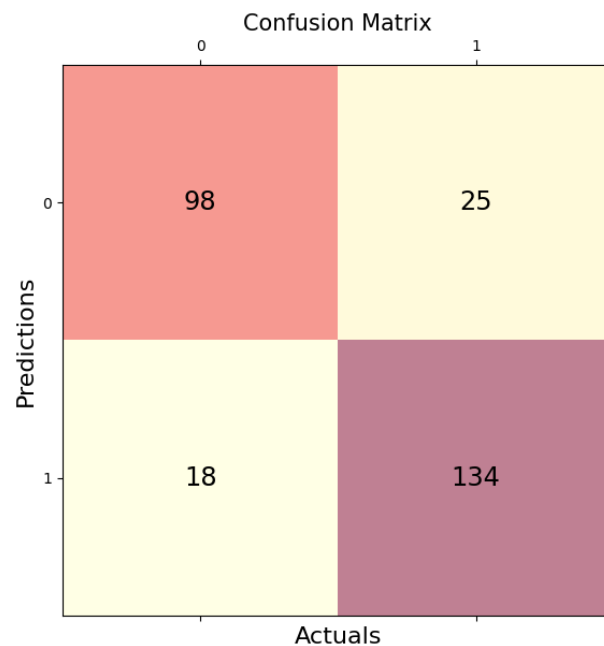
F1 值：綜合考慮精確度和召回率的指標， $2 * (0.886 * 0.865) / (0.886 + 0.865) \approx 0.875$

模型在檢測實際為心臟病的樣本方面表現良好，具有相對高的真陽性率。

模型在檢測實際為沒有心臟病的樣本方面也具有一定的能力，呈現較高的特異度。

然而，模型存在一定的風險，將一些實際上沒有心臟病的樣本誤判為有心臟病，呈現較高的假陽性率。

2. SVM (參考程式碼 Part 4 中的 SVM 的混淆矩陣)



由上圖可知:

精確率 (Precision) = $TP / (TP + FP) = 98 / (98 + 25) \approx 0.796$ ，表示模型預測

為有心臟病的樣本中約有 79.6% 是正確的。

準確性: $(98 + 134) / (98 + 25 + 18 + 134) \approx 0.84$ ，表示模型在預測心臟病的

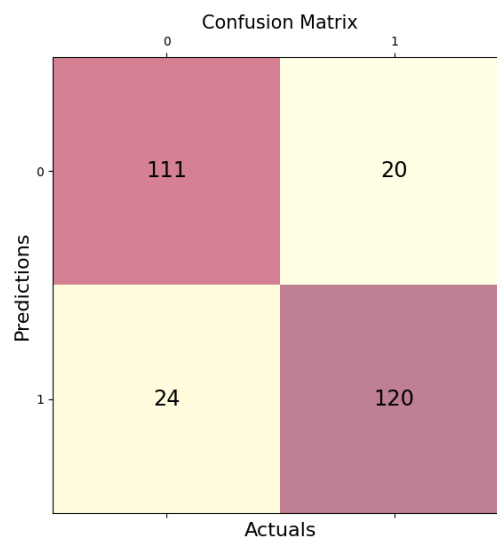
準確性為 84%。

召回率為 $98 / (98 + 18) \approx 0.845$ ，表示模型能夠檢測出 84.5% 的實際心臟病

樣本。

F1 值：綜合考慮精確度和召回率的指標， $2 * (0.796 * 0.845) / (0.796 + 0.845) \approx 0.820$ 。

3. SVM (參考程式碼 Part 4 中的 RandomForestRegressor 的混淆矩陣)



從上圖可以看出:

精確度： $111 / (111 + 20) \approx 0.847$ ，表示模型預測為有心臟病的樣本中約有 84.7% 是正確的。

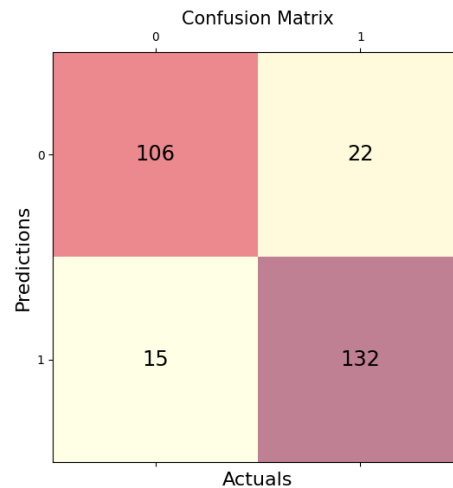
召回率： $111 / (111 + 124) \approx 0.472$ ，表示模型能夠捕捉到約 47.2% 的實際有心臟病樣本。

準確度為 $(111 + 120) / (111 + 20 + 124 + 120) \approx 0.547$ ，表示模型對於所有樣本的預測正確率約為 54.7%。

F1 值：綜合考慮精確度和召回率的指標， $2 * (0.847 * 0.472) / (0.847 + 0.472) \approx$

0.611。

4.KNN(參考程式碼 Part 4 中的 RandomForestRegressor 的混淆矩陣)



由上圖:

準確度 $= (106 + 132) / (106 + 22 + 15 + 132) \approx 0.8686$

精準度 $= 106 / (106 + 22) \approx 0.8281$ (約為 0.8281)

召回率 $= 106 / (106 + 15) \approx 0.8760$ (約為 0.8760)

F1 值 $= 2 * (0.8281 * 0.8760) / (0.8281 + 0.8760) \approx 0.8512$ (約為 0.8512)

總結:

Logistic Regression 模型在精確度、召回率、準確度和 F1 值方面都表現良好，具有較高的綜合性能。

SVM 模型在精確度和召回率方面表現較低，準確度和 F1 值也較低，相對於其他模型，它的表現較差。

Random Forest 模型在精確度方面表現較好，但召回率較低，準確度和 F1 值也相對較低。

KNN 模型在精確度、召回率、準確度和 F1 值方面都表現良好，但相對於 Logistic Regression 模型，其精確度稍低一些。

綜合考慮模型的各項指標，Logistic Regression 模型可能是最佳選擇，具有較高的綜合性能。