Machine Learning for COVID-19 Detection Based on Routine Blood Tests

邱承漢
M104020029
謝博丞
M104020054

國立中山大學資訊管理所 – 資料探勘 課程 - 第十組

Midterm Report

參考論文：

Development, evaluation, and validation of machine learning models for COVID-19 detection based on routine blood tests
https://www.degruyter.com/document/doi/10.1515/cclm-2020-1294/html

目錄

一、論文介紹：

雖然 rRT-PC 核酸檢測是目前檢測 Covid-19 最有效的方法，但也有以下已知的缺點，包括測試週期較長、可能會有試劑短缺的情況、有 15%~20%的假陰性機率、重點是 PCR 的設備成本昂貴，所以這篇論文的作者認為，利用血液常規檢查來檢測 covid-19 是一種更快更便宜的替代方案。

二、資料集介紹:

這次使用的資料集是 OSR(San Raphael Hospital)資料集，收集了 1737 筆常規抽血的測試結果資料，每筆資料由 34 個 featrue 組成，預測目標的兩種類別數量平衡，約有 52%的 COVID-19 陽性資料，48%的 COVID-19 陰性資料。

資料集的 feature 大概可以分成 5 種類別，分別是血液學的資料(紅白血球數量、淋巴細胞的數量、血小板的數量等…)、跟凝血有關的資料(凝血時間、纖維蛋白酶)、跟生物學有關的(膽紅素、鉀離子)、Rapidpoint 500 System 量測值(血氧、酸鹼值)、最後是一些額外資訊、有 Age、Sex 等患者的資本資訊，以及 Suspect，紀錄的是病患在看診的時候是否出現 covid-19 的疑似病癥、例如發燒、乾咳。如果有值就是 1、沒有就是 0。

Table 1: Complete list of the analyzed features in the *OSR dataset*.

Category	Instrument/sample	Parameter	Acronym	Unit of measure	COVID-specific features	CBC features	Missing rate, %
Hematological	Sysmex XE 2100/ whole blood	White blood cells	WBC	10^9/L	X	X	2.4
		Red blood cells	RBC	10^{12}/L	X	X	3.6
		Hemoglobin	HGB	g/dL	X	X	2.4
		Hematocrit	HCT	%	X	X	2.4
		Mean corpuscular volume	MCV	fL	X	X	3.6
		Mean corpuscolar hemoglobin	MCH	pg/Cell	X	X	3.6
		Mean corpuscolar hemoglobin concentration	MCHC	g Hb/dL	X	X	2.4
		Erythrocyte distribution width	RDW	CV%	X	X	3.7
		Platelets	PLT	10^9/L	X	X	3.6
		Mean platelet volume	MPV	fL	X	X	5.9
		Neutrophils count, (%)	NE	%	X	X	18.9
		Lymphocytes count, (%)	LY	%	X	X	15.2
		Monocytes count, (%)	MO	%	X	X	15.2
		Eosinophils count, (%)	EO	%	X	X	15.2
		Basophils count, (%)	BA	%	X	X	15.2
		Neutrophils count	NET	10^9/L	X	X	15.2
		Lymphocytes count	LYT	10^9/L	X	X	15.2
		Monocytes count	MOT	10^9/L	X	X	18.9
		Eosinophils count	EOT	10^9/L	X	X	15.2
		Basophils count	BAT	10^9/L	X	X	18.9
Coagulation	STA-R MAX/Plasma sample	Prothrombin time (INR)	PTINR	INR			31.0
		Activated partial thrombo-plastin time (R)	PPTR	Ratio			31.5
		Fibrinogen	FG	mg/dL			70.2
		D-dimer	XDP	µg/mL			70.4
Biochemical	Cobas 6000 Roche/ serum sample	Glucose	GLU	mg/dL	X		3.4
		Creatinine	CREA	mg/dL	X		2.4
		Urea	UREA	mg/dL	X		37.0
		Direct bilirubin	BILD	mg/dL			23.3
		Indirect bilirubin	BILIN	mg/dL			23.3
		Total bilirubin	BILT	mg/dL			25.3
		Alanine aminotransferase	ALT	U/L	X		3.1
		Aspartate aminotransferase	AST	U/L	X		3.2
		Alkaline phosphatase	ALP	U/L	X		23.7
		Gamma glutamyltransferase	GGT	U/L	X		24.5
		Lactate dehydrogenase	LDH	U/L	X		13.2
		Creatine kinase	CK	U/L	X		60.3
		Sodium	NA	mmol/L	X		3.9
		Potassium	K	mmol/L	X		2.7
		Calcium	CA	mmol/L	X		3.8
		C-reactive protein	CRP	mg/L	X		5.5
		NT-proB-type natriuretic peptide	PROBNP	pg/mL			91.1
		Troponin T	TROPOT	ng/L			62.8
		Interleukin 6	IL6	pg/mL			92.2
Rapidpoint 500 (Siemens Healthcare)	Hemogasanalysis, venous blood gas	pH	PHPOC	U			18.5
		Carbonic anhydride (pCO_2)	CO2POC	mmHg			22.4
		Oxygen (pO_2)	PO2POC	mmHg			22.4
		Bicarbonates	BICPOC	mmol/L			18.7
		Standard calculated bicarbonates	BISPOC	mmol/L			23.0
		Base excess	BEPOC	mmol/L			22.8
		Actual base excess	BEEPOC	mmol/L			18.9

Table 1: (continued)

Category	Instrument/sample	Parameter	Acronym	Unit of measure	COVID-specific features	CBC features	Missing rate, %
	CO-oxymetry	Hematocrit (POC)	HCTPOC	%			22.7
		Total oxyhemoglobin	THBPOC	g/dL			22.8
		O_2 saturation	SO2POC	%			18.3
		Oxyhemoglobin/Total hemoglobin	FO2POC	%			18.6
		Carboxyhemoglobin	FCOPOC	%			18.8
		Methemoglobin	METPOC	%			22.5
		Deoxyhemoglobin	HHBPOC	%			18.8
	Oxygenation	Bound O_2 maximum concentration	BO2POC	mL/dL			23.8
		Total oxygen	CTOPOC	mL/dL			20.9
		Inspired oxygen fraction	FIOPOC	mL/dL			67.4
		Inspired O_2/O_2 ratio	OFIPOC	Ratio			64.0
	Electrolytes POC	Sodium (POC)	NAPOC	mmol/L			22.5
		Potassium (POC)	KPOC	mmol/L			22.4
		Chloride (POC)	CLPOC	mmol/L			22.7
		Ionized calcium (POC)	CAPOC	mmol/L			23.1
		Standard ionized calcium (POC)	CASPOC	mmol/L			23.2
		Anion gap	ANGPOC	mmol/L			19.6
		Glucose blood gas	GLUEMO	mg/dL			18.6
		Lactate (POC)	LATPOC	mmol/L			18.5
Additional information		Age	Age	Years	X	X	0
		Gender	Sex	Male/ Female	X	X	0
		COVID-19 suspect (patient suffers from COVID-19 specific symptoms at triage)	Suspect	Yes/No	X	X	0
Target		COVID-19 positivity	Target	Positive/ Negative	X	X	0

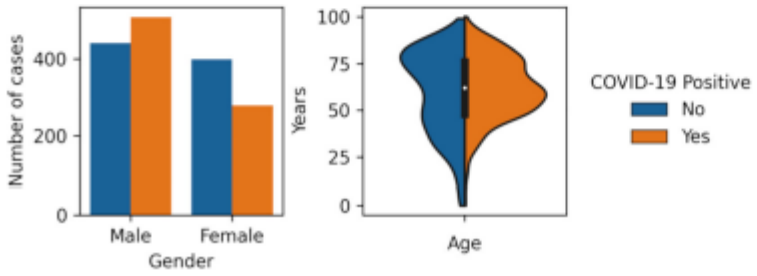


Figure 1: Demographic feature (gender and age) distributions for positive and negative cases.
The blue and orange areas correspond to negative and positive cases, respectively.

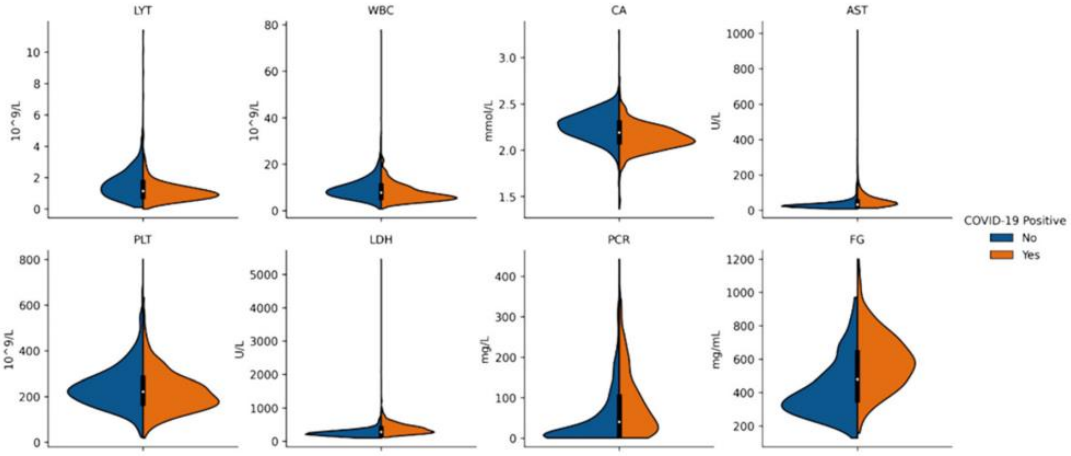
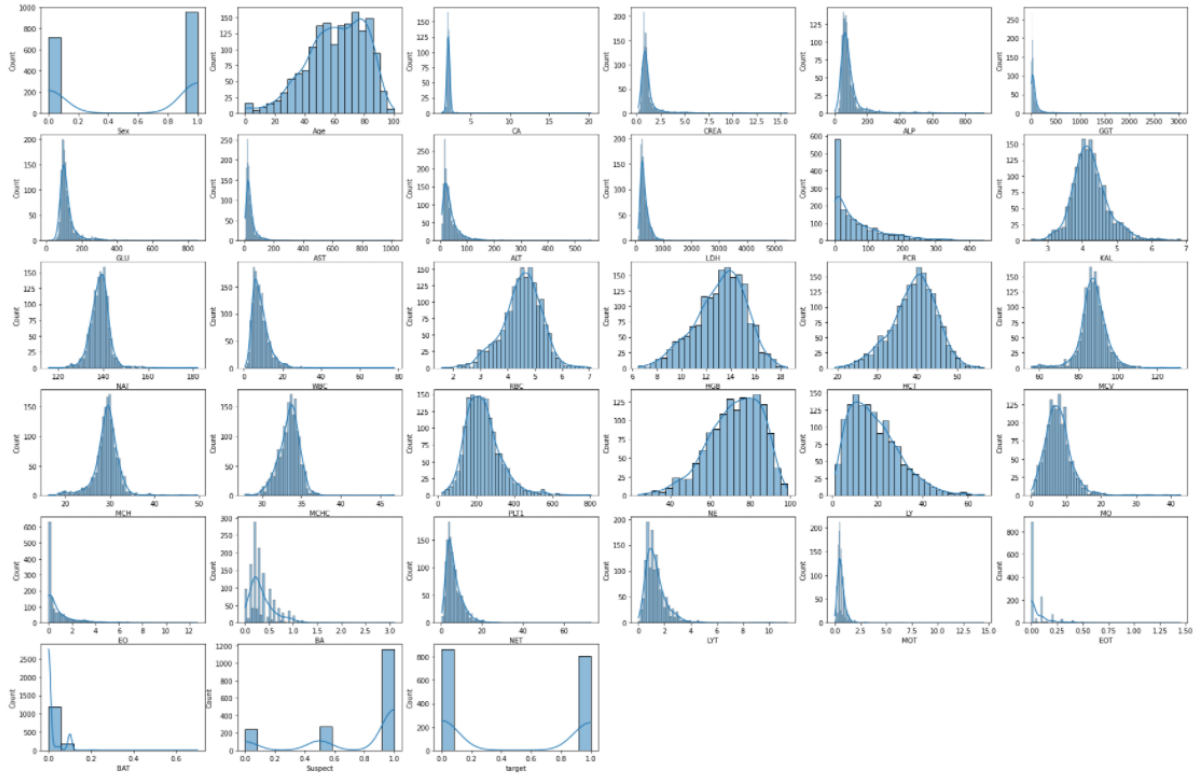


Figure 2: Violin plots depicting the distributions of eight relevant features in the *OSR dataset* (selected for their predictivity toward COVID-19). The blue and orange areas correspond to negative and positive cases, respectively.

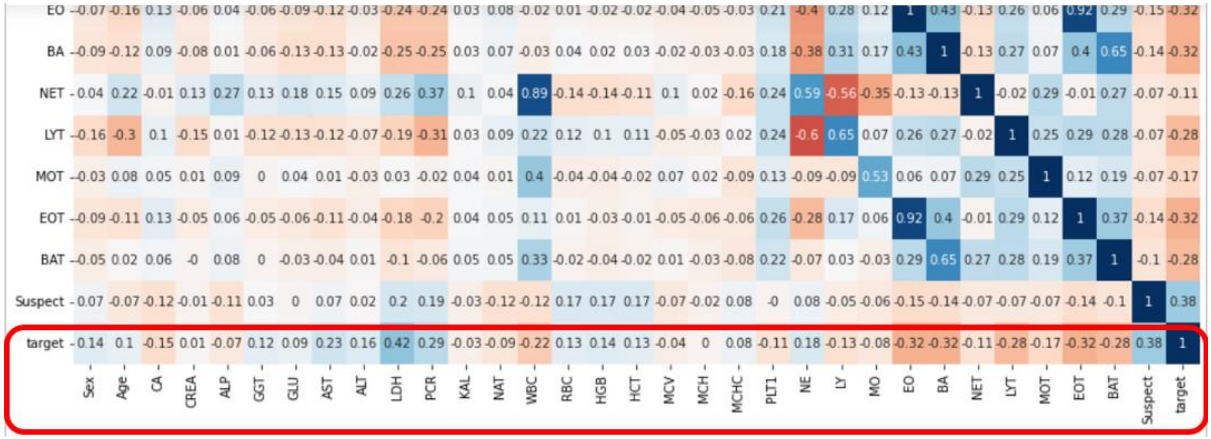
三、實作：

Exploratory Data Analysis (EDA)

大概看一下每個欄位資料值的數量分布情形，可以看到很多欄位的值，數量的分布並不平均，但是可以看到我們的 target 數量是蠻平衡的，不會有某個類別數量特別多的情況。



將每個特徵間的 Correlation 以熱度圖的方式畫出來，顏色越深代表關聯性越高，藍色代表正相關，橘色代表負相關。看已看出 LDH(乳酸脫氫酶)與 target 關聯性最高，有 0.42，其他超過 0.3 的值有 EO、BA、EOT 等欄位。



Drop Column & Row

我們把每個欄位的資料缺失比例算出來後，可以發現 CK 這個欄位的缺失比例是很嚴重的，超過一半的資料這個值都是缺失的，我們把缺值率超過 30%的 featrure 移除。

	Missing Rate
CK	58.00
UREA	36.75
ALP	24.70
GGT	22.43
EOT	18.02
MO	18.02
EO	18.02
BA	18.02
NET	18.02
LYT	18.02
MOT	18.02
BAT	18.02
LY	18.02
NE	18.02
LDH	14.50

我們也將每筆資料的缺少欄位的比例算出來，發現有 60 筆資料是除了性別、年齡，其他所有特徵都是缺失的，這些資料對這次任務是毫無意義的，所以將這些資料也移除。

											...									
1530	PSMAY0116_2020-05-04	0	41.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1544	PSMAY0061_2020-05-03	1	54.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1547	PSMAY0015_2020-05-05	0	61.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1
1549	PSMAY0027_2020-05-04	1	46.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1
1554	PSMAY0092_2020-05-18	0	93.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1560	PSMAY0209_2020-05-30	0	66.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1565	PSMAY0164_2020-05-11	0	35.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1569	PSMAY0005_2020-05-05	0	45.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1
1575	PSMAY0159_2020-05-22	1	60.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1598	PSMAY0202_2020-05-01	1	78.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1600	PSMAY0017_2020-05-04	1	51.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	1
1620	PSMAY0036_2020-05-19	1	47.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1688	6	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0
1711	29	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0

60 rows × 36 columns

Train-Test Split:

這篇論文原本有用兩個其他的資料集來做外部驗證，測試他們模型的好壞。但是我們找不到論文說的其他兩個資料集，論文底下提供的連結就只有 OSR 資料集，我們只能把這個資料集以比例以 8:2 的方式切成訓練集跟測試集，切的時候指定要照著原本資料集 target 的比例去切，切完後的訓練集跟測試集的陰陽性比例跟原本的資料集相同。

```
raw data percentage :
0    51.73031
1    48.26969
Name: target, dtype: float64

train percentage :
0    51.716418
1    48.283582
Name: target, dtype: float64

test percentage :
0    51.785714
1    48.214286
Name: target, dtype: float64
```

KNN Imputation

我們需要對缺值的欄位進行補值，我們嘗試過使用最基本的中位數補值、或是用平均值補，但是其實效果都差不多，所以我們最後使用跟論文一樣方法，用 KNN 的方式補值。簡單來說就是根據每筆資料的距離，將缺值以離目標最近的附近點的同欄位補值。k 的值設定為 5。

Min-Max Normalization

由於我們的特徵間的值域並不相同，所以我們在訓練之前將資料的特徵做標準化，將所有特徵的值域轉換到 0 到 1 之間。可以看到轉換完後的每個 feature 最大值都是 1 最小值都是 0。由於原始論文沒有明確地講使用的方法，所以我們無法從這點進行比較。

	count	mean	std	min	25%	50%	75%	max
Sex	1340.0	0.57	0.50	0.0	0.00	1.00	1.00	1.0
Age	1340.0	0.61	0.19	0.0	0.48	0.62	0.77	1.0
CA	1340.0	0.05	0.03	0.0	0.04	0.04	0.05	1.0
CREA	1340.0	0.06	0.06	0.0	0.04	0.05	0.06	1.0
ALP	1340.0	0.11	0.08	0.0	0.07	0.09	0.12	1.0
GGT	1340.0	0.05	0.08	0.0	0.02	0.03	0.06	1.0
GLU	1340.0	0.13	0.07	0.0	0.09	0.11	0.13	1.0
AST	1340.0	0.04	0.05	0.0	0.01	0.02	0.04	1.0
ALT	1340.0	0.06	0.08	0.0	0.02	0.04	0.07	1.0
LDH	1340.0	0.14	0.09	0.0	0.08	0.11	0.18	1.0
PCR	1340.0	0.15	0.18	0.0	0.01	0.09	0.23	1.0
KAL	1340.0	0.40	0.12	0.0	0.32	0.39	0.46	1.0
NAT	1340.0	0.34	0.07	0.0	0.30	0.35	0.38	1.0
WBC	1340.0	0.10	0.06	0.0	0.06	0.09	0.13	1.0
RBC	1340.0	0.54	0.13	0.0	0.46	0.55	0.62	1.0
HGB	1340.0	0.57	0.18	0.0	0.46	0.59	0.70	1.0
HCT	1340.0	0.56	0.16	0.0	0.47	0.58	0.67	1.0
MCV	1340.0	0.41	0.09	0.0	0.37	0.41	0.46	1.0

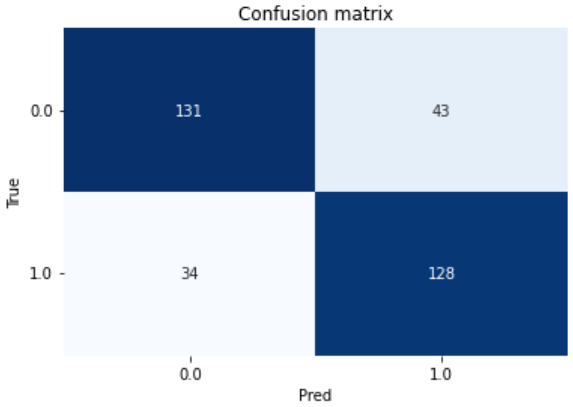
Model Construction

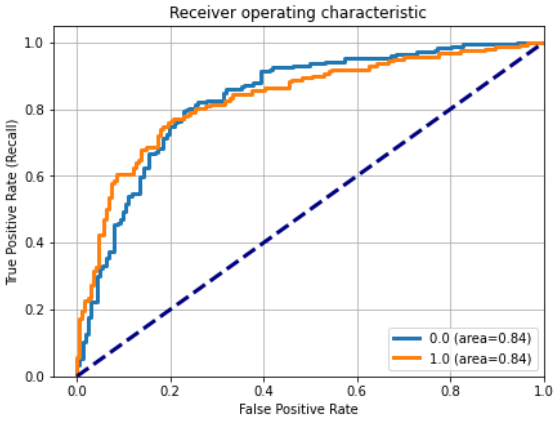
- SVM Classifier

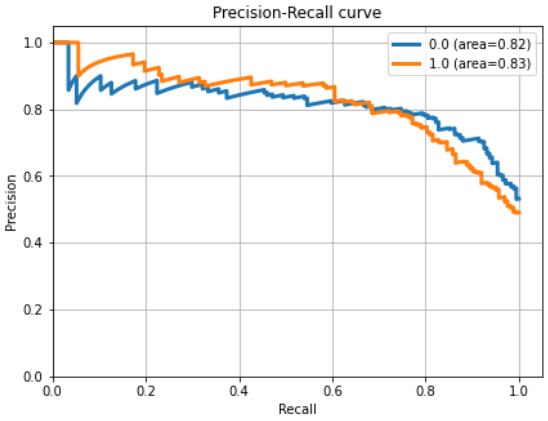
```
model type: SVM
time costing: 0.15199995040893555
Accuracy: 0.77
Auc: 0.84
Detail:
              precision    recall  f1-score   support

         0.0       0.79      0.75      0.77       174
         1.0       0.75      0.79      0.77       162

    accuracy                           0.77       336
   macro avg       0.77      0.77      0.77       336
weighted avg       0.77      0.77      0.77       336
```





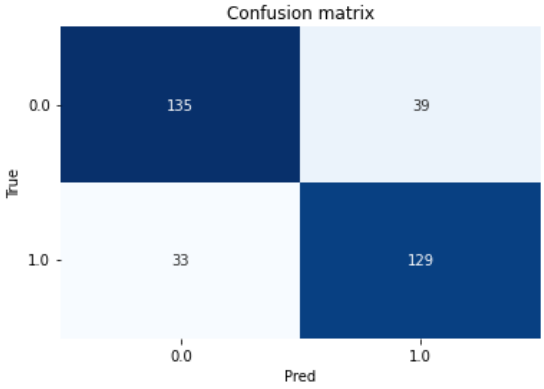


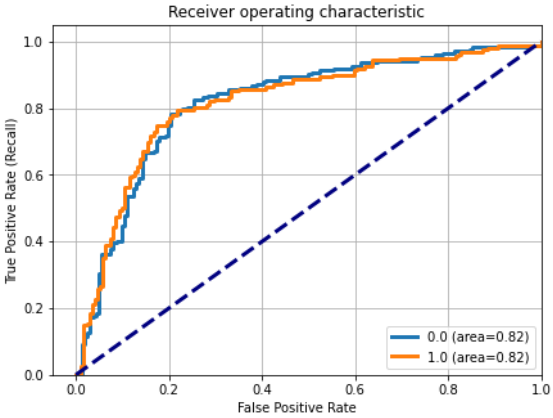
- Logistic Regression

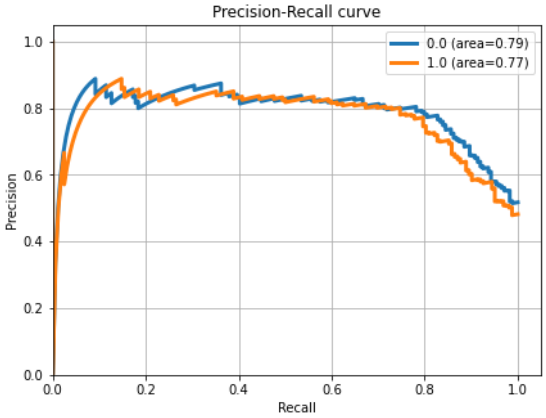
```
model type: Logistic Regression
time costing: 0.021999597549438477
Accuracy: 0.79
Auc: 0.82
Detail:
              precision    recall  f1-score   support

         0.0       0.80      0.78      0.79       174
         1.0       0.77      0.80      0.78       162

    accuracy                           0.79       336
   macro avg       0.79      0.79      0.79       336
weighted avg       0.79      0.79      0.79       336
```





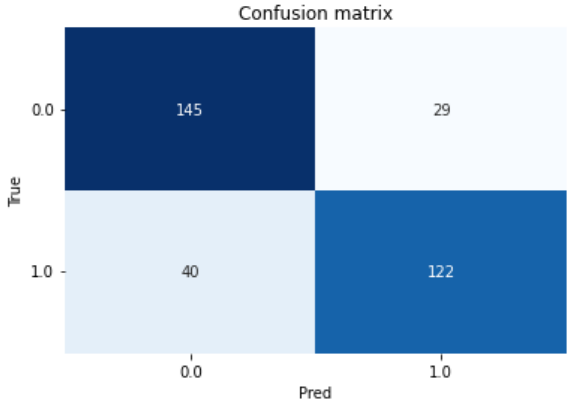


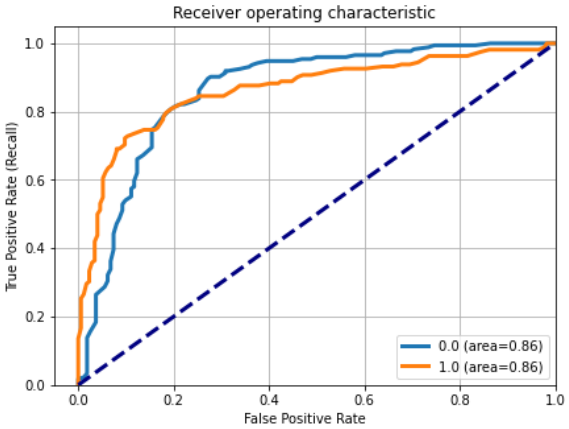
- Random Forest

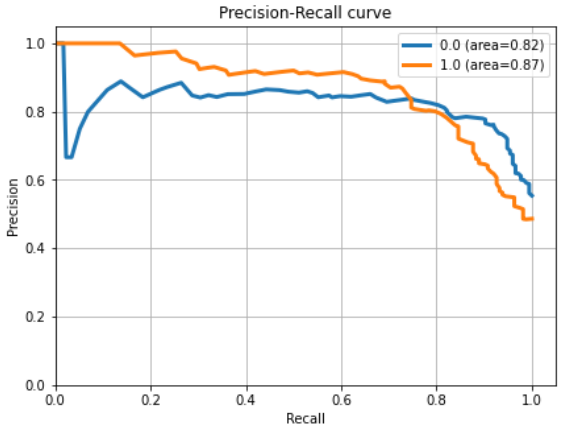
```
model type: Random Forest
time costing: 0.34414243698120117
Accuracy: 0.79
Auc: 0.86
Detail:
              precision    recall  f1-score   support

         0.0       0.78      0.83      0.81       174
         1.0       0.81      0.75      0.78       162

    accuracy                           0.79       336
   macro avg       0.80      0.79      0.79       336
weighted avg       0.80      0.79      0.79       336
```







- XGBoost

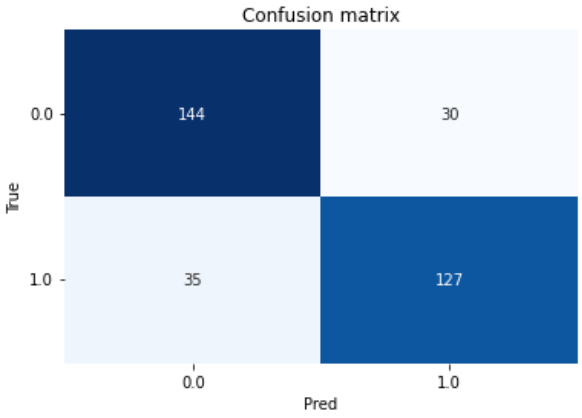
```
clf_xgb = XGBClassifier(num_iterations=1000, subsample=0.5,sampling_method='uniform', object='binary:logistic',
                        val_metric='auc', eta=0.01, gamma=0.3, reg_alpha=0.3, reg_lambda=0.7,
                        use_label_encoder=False, verbosity=0, # to ignore userwaning
                        )
```

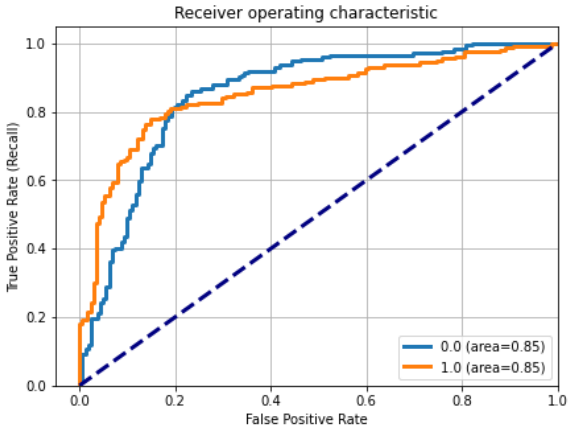
我們這組額外測的是 Boosting 的方法，不同於之前的模型都是一次將模型建好，Boosting 的演算法會根據目前模型的分類結果，修正改變下一輪模型抽樣的機率，以達到類似不斷改進學習的效果。

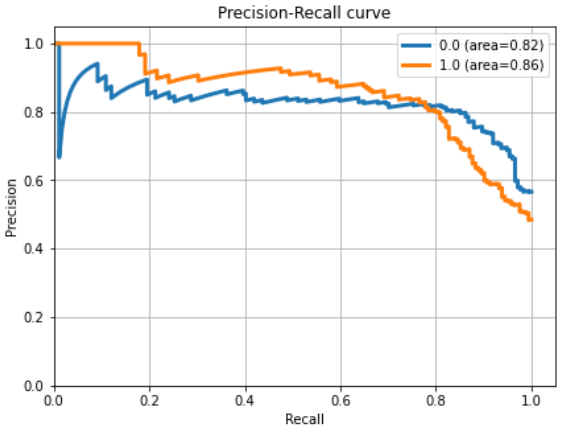
```
model type: XGBoost
time costing: 0.2906224727630615
Accuracy: 0.81
Auc: 0.85
Detail:
              precision    recall  f1-score   support

         0.0       0.80      0.83      0.82       174
         1.0       0.81      0.78      0.80       162

    accuracy                           0.81       336
   macro avg       0.81      0.81      0.81       336
weighted avg       0.81      0.81      0.81       336
```







- Multi-layer Perceptron(MLP)

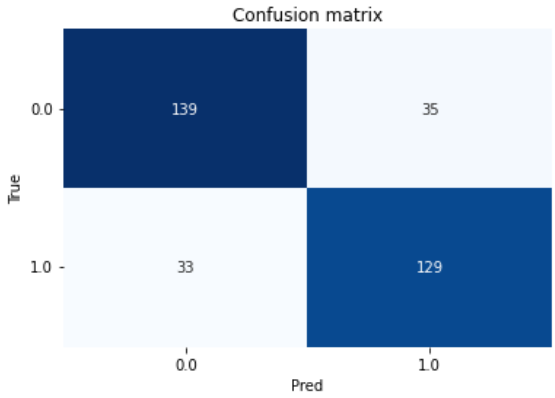
```
clf_mlp = MLPClassifier(hidden_layer_sizes=(128,), activation='relu', solver='adam',
                        alpha=1e-4, max_iter=10000,
                        early_stopping=False, shuffle=True, verbose=False)
```

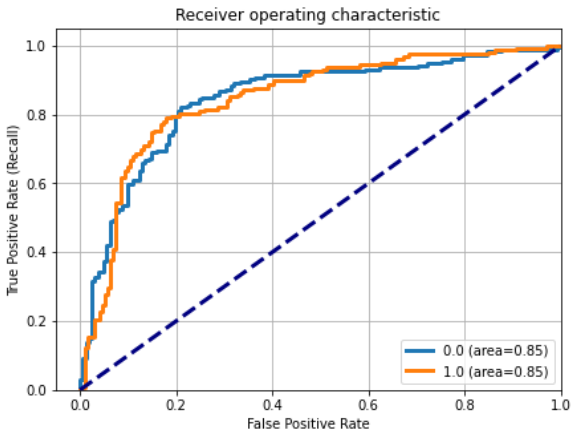
我們也加入了神經網路來測試，使用的是多層感知器 MLP 進行建模，神經網路的優點就是可以藉由疊加層數增加模型對資料特徵的抽象能力，缺點就是他難以被解釋，而且計算成本較高。在測試多次後，我們發現在這份資料上用太複雜的模型反而效果不好，所以我們只有一層 128 個神經元的隱藏層，總共三層的神經網路、activation function 用的是 ReLU、優化器用 adam。

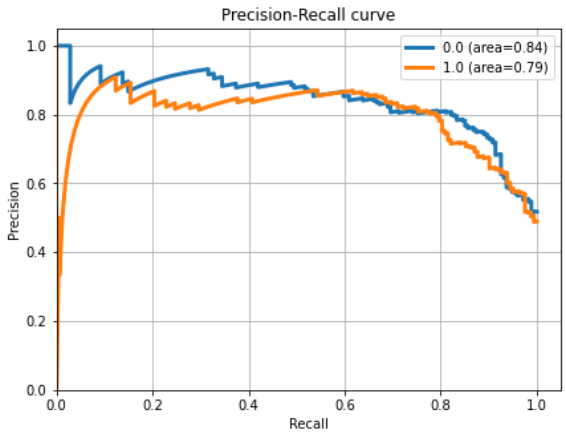
```
model type: MLP
time costing: 6.2444047927856445
Accuracy: 0.8
Auc: 0.85
Detail:
              precision    recall  f1-score   support

         0.0       0.81      0.80      0.80       174
         1.0       0.79      0.80      0.79       162

    accuracy                           0.80       336
   macro avg       0.80      0.80      0.80       336
weighted avg       0.80      0.80      0.80       336
```

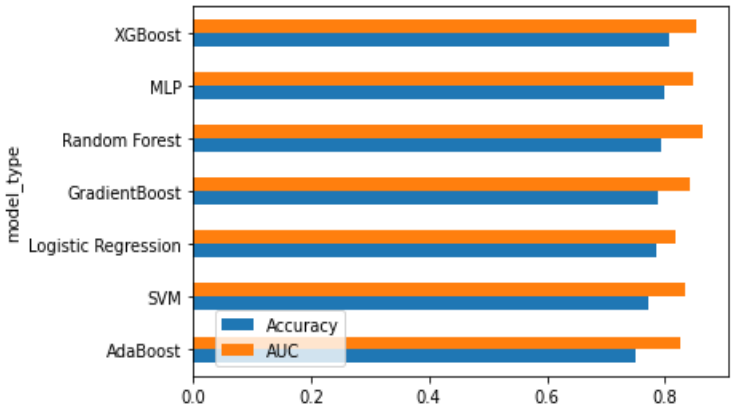


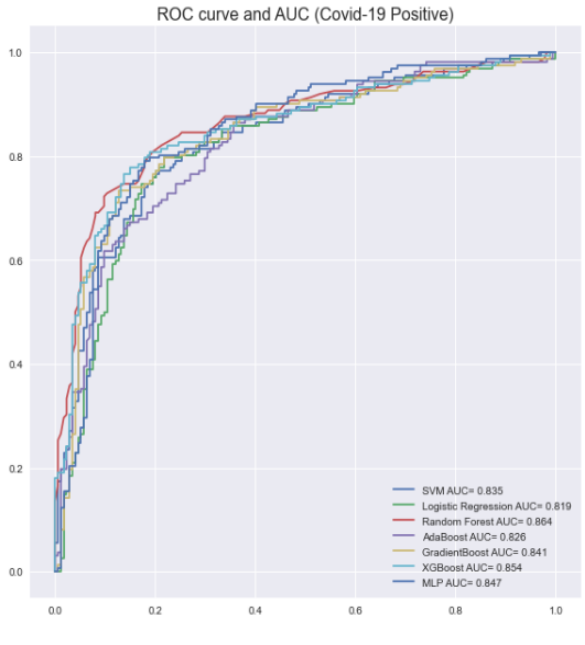


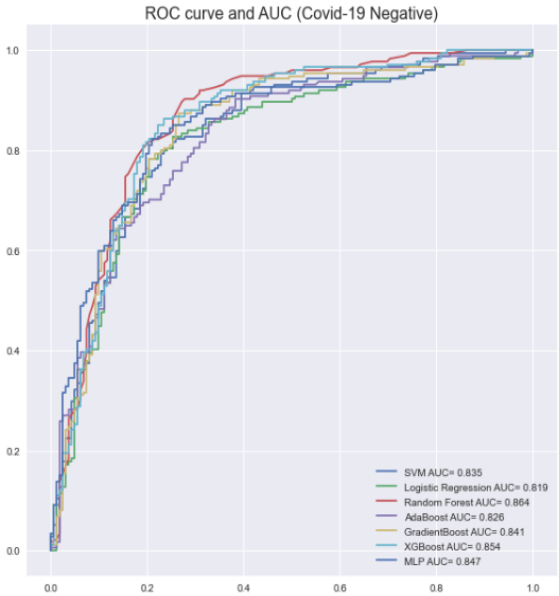


Result

	model_type	Accuracy	AUC
0	XGBoost	0.806548	0.853519
0	MLP	0.797619	0.84653
0	Random Forest	0.794643	0.863523
0	GradientBoost	0.78869	0.841032
0	Logistic Regression	0.785714	0.819107
0	SVM	0.770833	0.835001
0	AdaBoost	0.75	0.825617







所有模型準確率大概落在 0.75 至 0.8 成間，AUC 值在 0.8~0.87 間，綜合性能來說 XGBoost 跟 MLP 表現最好，Random Forest 表現也相當不錯

Recursive Feature Elimination(RFE)

論文使用了一種叫做 RFE 的 feature selection 方法，簡單來說就是根據訓練完模型後，模型算出來的特徵重要值，去篩選特徵，每一輪會刪除掉重要值最低的特徵。右邊這裡就是 XGBoost 訓練完後，模型對特徵的重要性排序，可以看到 EOT 跟 LDH 對於判斷結果來說是相當重要的。

```
model type: XGBoost
    Features  importances
29       EOT     0.455840
9        LDH     0.310221
13       WBC     0.169918
2         CA     0.024573
15       HGB     0.020856
8        ALT     0.018592
```

Principal Component Analysis(PCA)

但是 RFE 的方法有些限制，因為有些模型其實是不會對特徵的重要性做排序的，例如 KNN、或是神經網路這種較難解釋判斷標準的模型。所以我們另外嘗試了另一種無監督式的 feature selection 方法，使用 PCA 的好處是，不會有模型的限制，但缺點就是他把一些貢獻類似的特徵合在一起後，也會變得難以解釋。而且在我們實際測試後，PCA 降維後會對所有模型的表現造成些微的下降。

Grid Search

最後我們使用 Grid Search 的方式找出模型最佳化的參數。

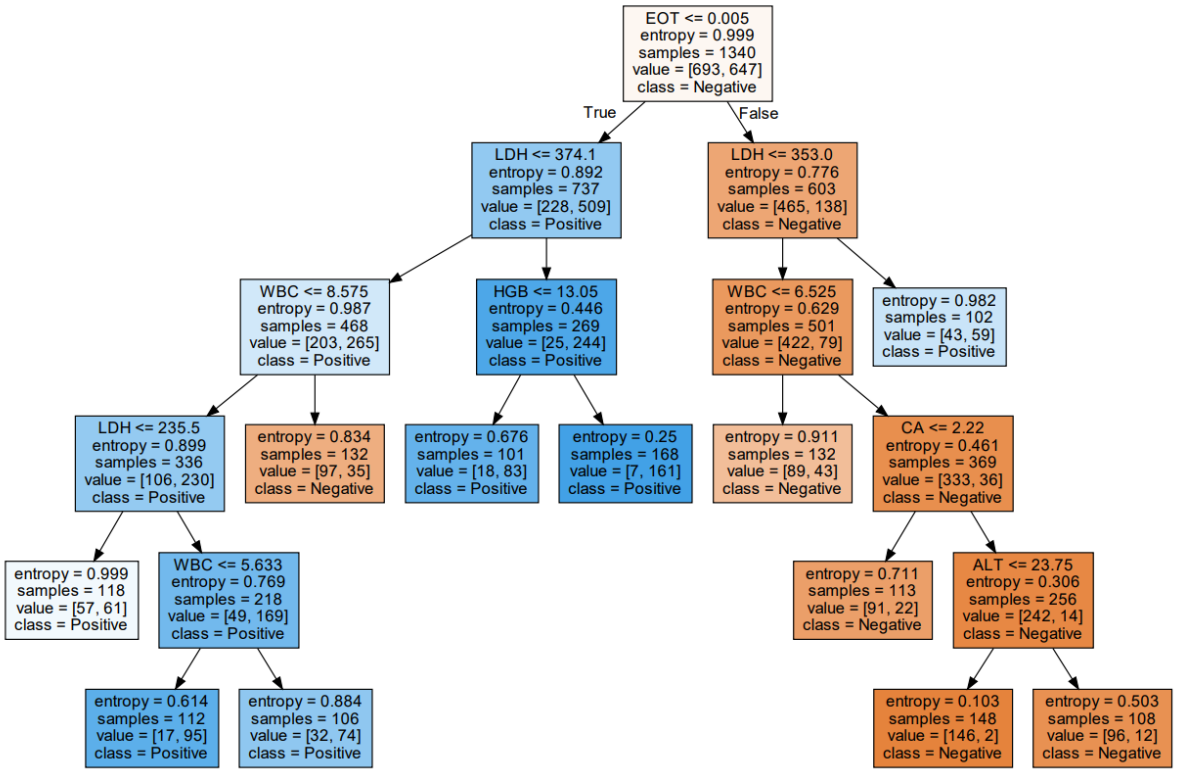
Random Forest

```
{'max_depth': 14, 'n_estimators': 42}
0.8395522388059702
```

XGBoost

```
{'n_estimators': 400, 'max_depth': 30, 'learning_rate': 0.09, 'colsample_bytree': 0.4}
0.8395522388059702
```

最後為了方便解釋，我們建了一顆決策數大概看一下這個資料集的判斷情形，可以看到一開始用 EOT(嗜酸性粒细胞數)判斷，單位是血常規，如果 EOT<=0.005 而且 LDH(乳酸脱氢酶) <=374.1 U/L 的話基本上就會被判斷為確診了。



四、心得：

不同於以往使用練習用資料集進行分析，網路上有許多參考的做法可以學習，這次報告則是使用正式研究用途的資料集進行分析，參考資料為正式的論文。在尋找相關論文的過程中，我們也可以從一些角度發現論文的不足之處，並進行改進。除了需要對資料進行更深入的探討，並透過自己嘗試不同的做法，比較不同方法的優劣，也藉由許多論文學習到許多課堂內不會提到的實作技巧，並結合課堂所學實際操作，將這些經驗累積成自己的知識。最後也透過課堂報告，參考他人的作法，同時思考自己作法的不足之處。就像助教上課所講，資料探勘沒有捷徑，最好的學習方式就是透過不斷的練習。

References：

https://www.degruyter.com/document/doi/10.1515/cclm-2020-1294/html

https://zenodo.org/record/4081318#.YkwDQS1BxPa

https://scikit-learn.org/stable/index.html