Chest X-Ray Images (Pneumonia)

一、第 14 組組員

Student ID	Name	Work assignment
0853434	曾耀緯	組長、Bayesian Classifier
0853430	黃佳晨	SVM
0853431	范姜永岩	XGboost
0853439	游家權	製作投影片、口頭報告
0853423	羅紹華	製作 report
0853411	劉書維	資料前處理、CNN

二、研究目的與動機

武漢肺炎正讓全世界都飽受痛苦之中，目前也沒有很有效的解藥能夠醫治 武漢肺炎，因此如何能夠有效的判斷病人是否有肺炎是非常重要的事情。

三、資料及敘述

資料運用：Kaggle 中的 5863 張小孩胸腔 X 光照片。

Reference：

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Code：

https://drive.google.com/drive/u/3/folders/1efS6xv27GyEtt6dnuMoxtr1f3tXWbaWU

四、 分析工具

開發工具：Spyder

開發環境：Python 3.7

五、 資料前處理

資料分為 train、val、test 三種，分別為 5216、16、624 張胸部 X 光圖片。

Dataset	
train	5216
val	16
test	624

Step1: 利用 os 與 vc2 的套件依照路徑取得資料夾內所有圖片，並區分為肺炎(0)、正常(1)。

Step2: 將各種類的 feature 與 label 分為 x 與 y。

Step3: 將 x 的資料正規化到 0~1 之間 (也就是 /255)。

Step4: 將 x 的資料 resize 成(150, 150, 1)的大小(黑白圖片)。

Step5: 做 data augmentation，避免 overfitting 和資料 imbalance。

六、實作與評估方法

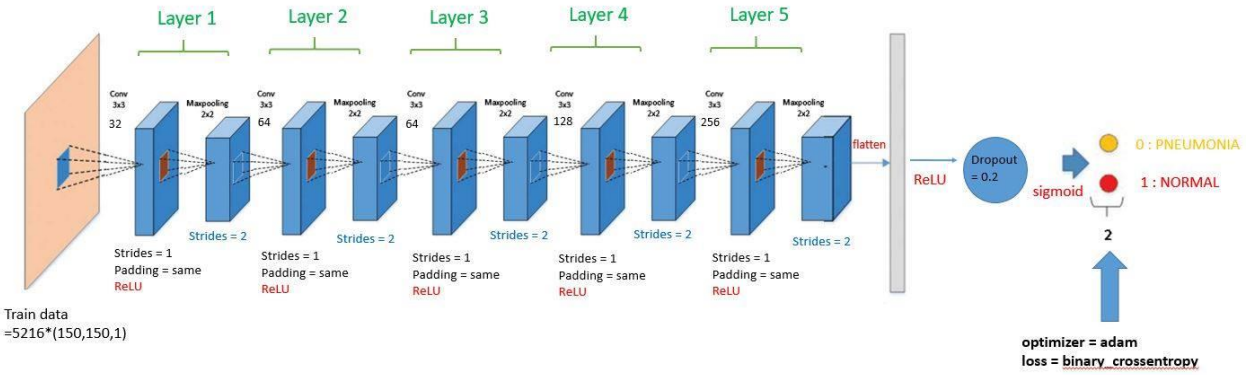
1. Convolution Neural Network (卷積神經網路):

特性之一是權值共享（weight sharing）網絡結構更類似於生物神經網路，降低了網絡模型的複雜度，減少了權值的數量。常應用於當前語音分析和影像識別領域的研究。

優點：在網路的輸入是多維影像時表現的更為明顯，使影像特徵可以直接作為網絡的輸入，避免了傳統識別演算法中複雜的特徵提取和資料重建過程。

缺點：全連結 DNN 的結構裡下層神經元和所有上層神經元都能夠形成連線，帶來了引數數量的膨脹問題。

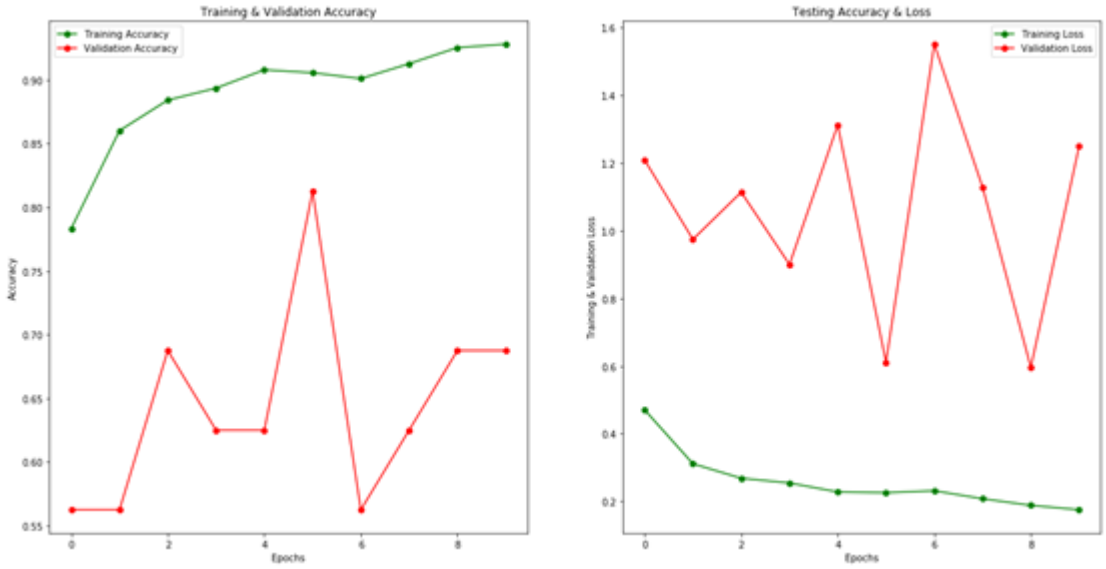
我們利用 keras 建構 CNN 模型，每一層都有 conv 3*3 （ReLU）+ maxpooling 2*2，共有五層，之後 flatten 後進行 ReLU、dropout、Sigmoid，就輸出 0 和 1 ，每個模型跑 10 個 epochs，結構如下圖所示：



將 CNN 分為 2 layers ~ 5 layers 來比較結果。結果如下：

CNN two layers：

Train and Val data train accuracy：0.92 val accuracy：0.81



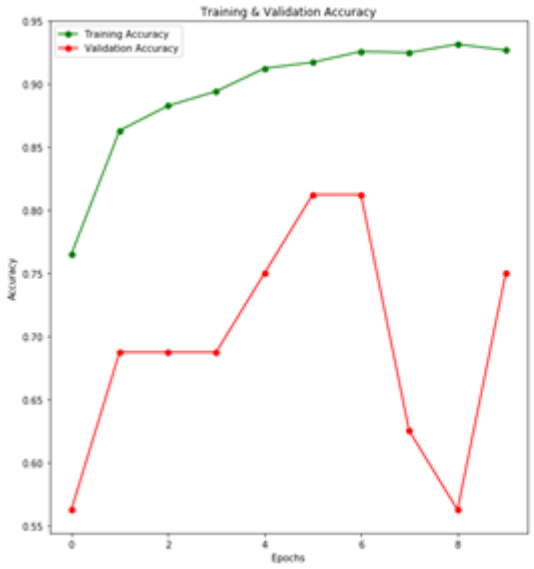
Test data test accuracy：0.91

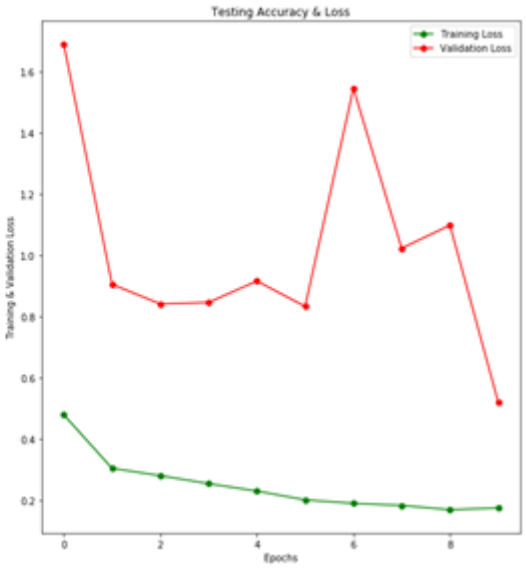
	precision	recall	f1-score	support
Pneumonia (Class 0)	0.89	0.97	0.93	390
Normal (Class 1)	0.95	0.80	0.87	234
accuracy			0.91	624
macro avg	0.92	0.89	0.90	624
weighted avg	0.91	0.91	0.91	624

```
[[380  10]
 [ 47 187]]
```

CNN three layers：

Train and Val data train accuracy：0.93 val accuracy：0.825





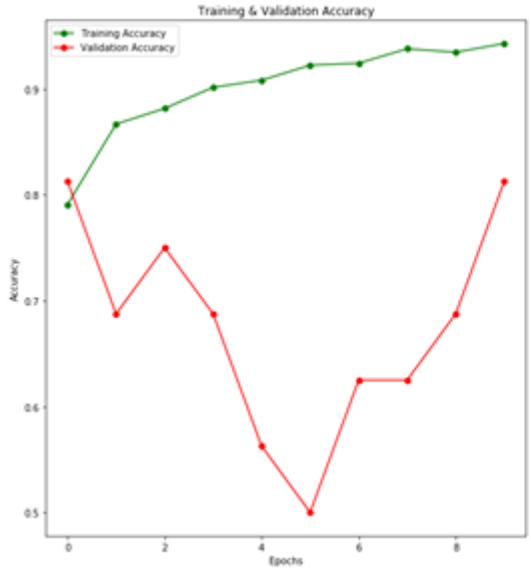
Test data test accuracy：0.93

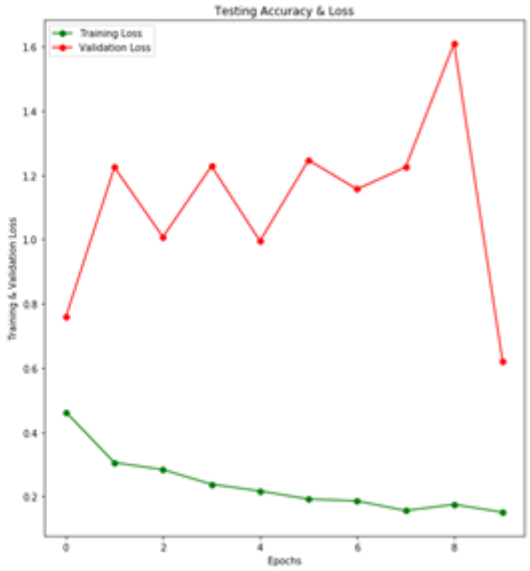
	precision	recall	f1-score	support
Pneumonia (Class 0)	0.92	0.97	0.95	390
Normal (Class 1)	0.94	0.86	0.90	234
accuracy			0.93	624
macro avg	0.93	0.92	0.92	624
weighted avg	0.93	0.93	0.93	624

```
[[378  12]
 [ 32 202]]
```

CNN four layers：

Train and Val data train accuracy：0.925 val accuracy：0.81





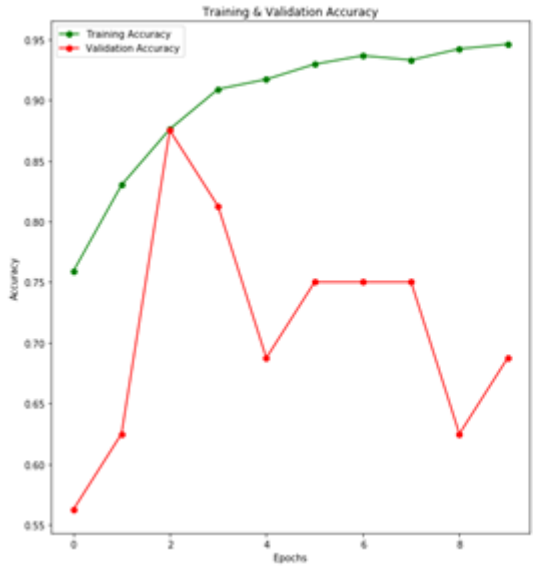
Test data test accuracy：0.91

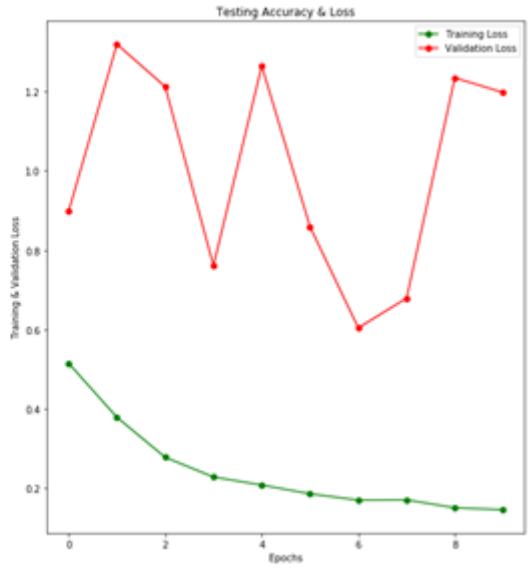
	precision	recall	f1-score	support
Pneumonia (Class 0)	0.93	0.92	0.93	390
Normal (Class 1)	0.87	0.88	0.88	234
accuracy			0.91	624
macro avg	0.90	0.90	0.90	624
weighted avg	0.91	0.91	0.91	624

```
[[360  30]
 [ 28 206]]
```

CNN five layers：

Train and Val data train accuracy：0.95 val accuracy：0.875





Test data test accuracy：0.91

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.90	0.97	0.93	390
Normal (Class 1)	0.95	0.81	0.87	234
accuracy			0.91	624
macro avg	0.92	0.89	0.90	624
weighted avg	0.91	0.91	0.91	624

```
[[379  11]
 [ 44 190]]
```

可以從上表看出是 five-layer model 較為準確，但是在第 2 個 epoch 時就已經對 val data 有 overfitting 的狀況，且 loss 也有上升，我們推測可能是 val 中的 data 過少，所以容易造成很大的變動，所以應該增加一些資料，來做出各有統計意義的判斷。

2. Support Vector Machine（SVM）:

SVM 是一種監督式的學習方法，用統計風險最小化的原則來估計一個分類的超平面（hyperplane），並找到一個決策邊界（decision boundary）讓兩類之間的邊界（margins）最大化，使其可以完美區隔開來。

在用 SVM 處理問題時，如果資料線性不可分，希望通過將輸入空間內線性不可分的資料 對映到一個高維的特徵空間內，使資料在特徵空間內是線性可分的，這個對映記作 $\phi(x)$，之後優化問題中就會有內積 $\phi i \cdot \phi j$，這個內積的計算維度會非常大，因此引入了核函式（kernel）可以幫我們很快地做一些計算，否則將需要在高維空間中進行計算。

我們使用 3 種不同的 kernel 來比較其結果好壞:

Gaussian radial basis function (RBF)：

通用於各種應用中，線性不可分時，特徵維數少 樣本數量正常時，在沒有先驗知識時用，取值在[0,1]。

Val data val accuracy：0.75

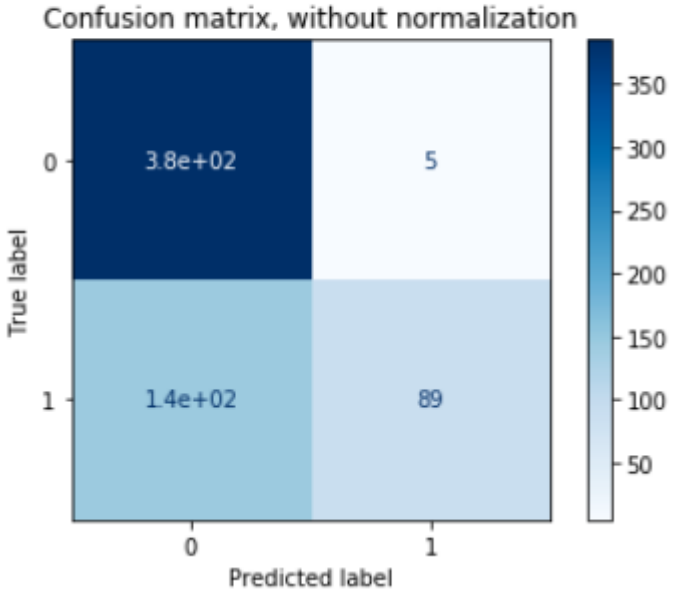
```
# predict validation set
pred_x_val = svm.predict(x_val)
accuracy_score(pred_x_val, y_val)
```

```
0.75
```

Test data test accuracy：0.759

	precision	recall	f1-score	support
0	0.73	0.99	0.84	390
1	0.95	0.38	0.54	234
accuracy			0.76	624
macro avg	0.84	0.68	0.69	624
weighted avg	0.81	0.76	0.73	624

```
Confusion matrix, without normalization
[[385   5]
 [145  89]]
```



Polynomial kernel：

常用於 image processing，引數比 RBF 多，取值範圍是(0,inf)

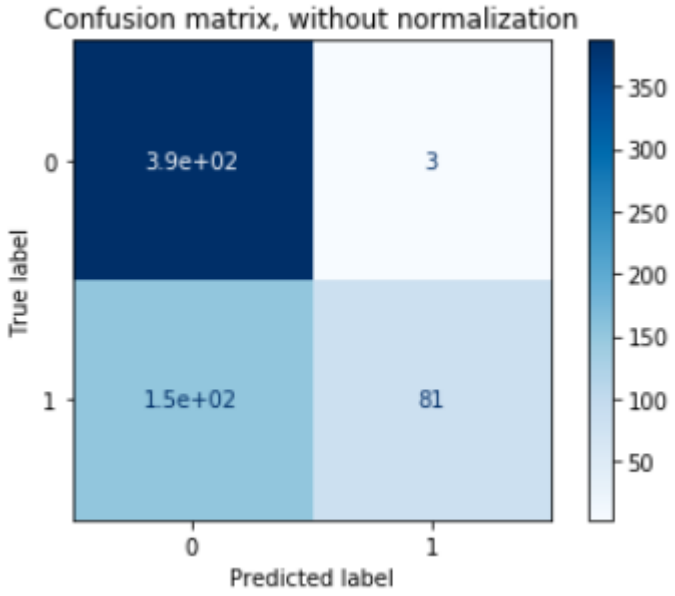
Val data val accuracy：0.75

```
# predict validation set
pred_x_val = svm.predict(x_val)
accuracy_score(pred_x_val, y_val)
```

```
0.75
```

Test data test accuracy：0.75

	precision	recall	f1-score	support
0	0.72	0.99	0.83	390
1	0.96	0.35	0.51	234
accuracy			0.75	624
macro avg	0.84	0.67	0.67	624
weighted avg	0.81	0.75	0.71	624



Linear kernel：

線性可分時，特徵數量多時，樣本數量多再補充一些特徵時，linear kernel

可以是 RBF kernel 的特殊情況。

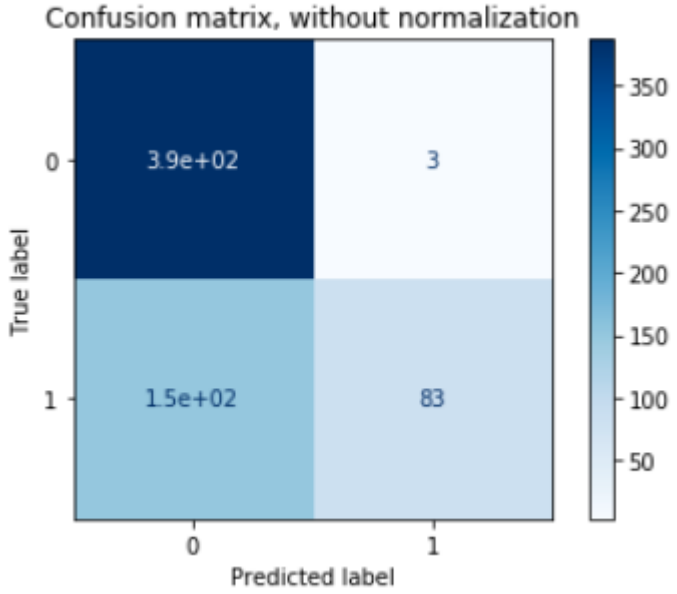
Val data val accuracy：0.8125

```
# predict validation set
pred_x_val = svm.predict(x_val)
accuracy_score(pred_x_val, y_val)
```

```
0.8125
```

Test data test accuracy：0.753

	precision	recall	f1-score	support
0	0.72	0.99	0.83	390
1	0.97	0.35	0.52	234
accuracy			0.75	624
macro avg	0.84	0.67	0.68	624
weighted avg	0.81	0.75	0.72	624



3. XGBoost

XGBoost 實現的是一種通用的 Tree Boosting 演算法。分類方法就是不斷地添加樹，不斷地進行特徵分裂來生長一棵樹，每次添加一個樹，其實是學習一個新函數，去擬合上次預測的殘差。當我們訓練完成得到 k 棵樹，我們要預測一個樣本的分數，其實就是根據這個樣本的特徵，在每棵樹中會落到對

應的一個葉子節點，每個葉子節點就對應一個分數，最後只需要將每棵樹對

應的分數加起來就是該樣本的預測值。

Val data val accuracy：0.5625

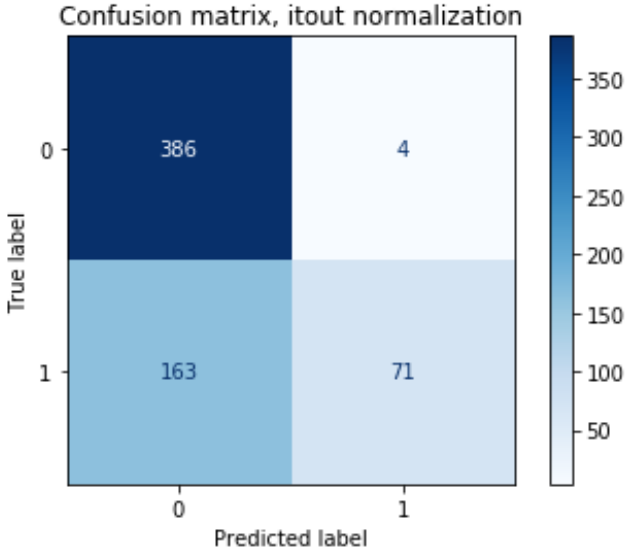
```
#predict validation
x_val = x_val.reshape(len(x_val),img_size*img_size)
print("val_accuracy",xgbc.score(x_val,y_val))
```

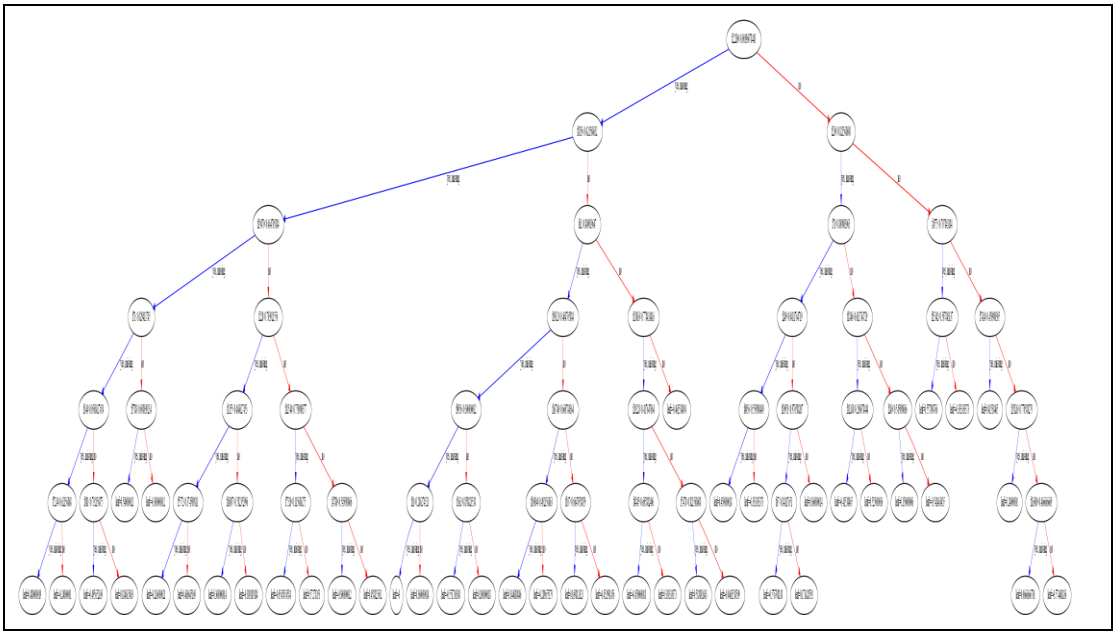
```
val_accuracy 0.5625
```

Test data test accuracy：0.732

	precision	recall	f1-score	support
0	0.70	0.99	0.82	390
1	0.95	0.30	0.46	234
accuracy			0.73	624
macro avg	0.82	0.65	0.64	624
weighted avg	0.79	0.73	0.69	624

```
Confusion matrix, itout normalization
[[386   4]
 [163  71]]
```





4. Bayesian Classifier(貝氏分類器):

單純貝氏是一種構建分類器的簡單方法。該分類器模型會給問題實例分配用特徵值表示的類標籤，類標籤取自有限集合。它不是訓練這種分類器的單一演算法，而是一系列基於相同原理的演算法：所有單純貝氏分類器都假定樣本每個特徵與其他特徵都不相關。

我們使用 3 種不同的 Bayesian Classifier 來比較其結果好壞:

高斯貝氏：

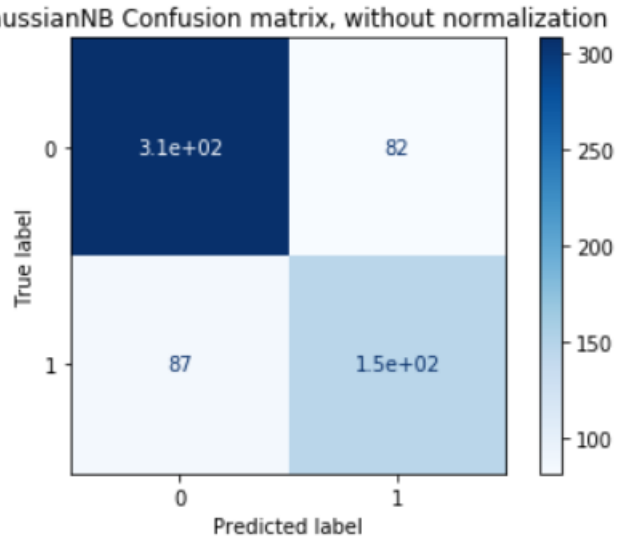
Val data val accuracy：0.6875

```
高斯貝氏分類器
test_accuracy 0.7291666666666666
val_accuracy 0.6875
```

Test data test accuracy：0.729

	precision	recall	f1-score	support
0	0.78	0.79	0.78	390
1	0.64	0.63	0.63	234
accuracy			0.73	624
macro avg	0.71	0.71	0.71	624
weighted avg	0.73	0.73	0.73	624

```
GaussianNB Confusion matrix, without normalization
[[308  82]
 [ 87 147]]
```



伯努力貝氏：

Test data test accuracy：0.7868

Val data val accuracy：0.6875

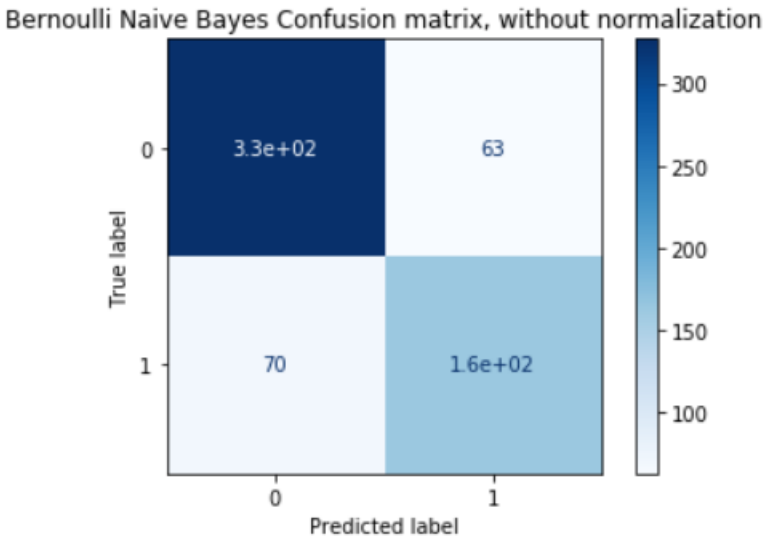
```
伯努力貝氏分類器
test_accuracy 0.7868589743589743
val_accuracy 0.6875

              precision    recall  f1-score   support

           0       0.82      0.84      0.83       390
           1       0.72      0.70      0.71       234

    accuracy                           0.79       624
   macro avg       0.77      0.77      0.77       624
weighted avg       0.79      0.79      0.79       624

Bernoulli Naive Bayes Confusion matrix, without normalization
[[327  63]
 [ 70 164]]
```



多項氏貝氏：該模型常用於文本分類。

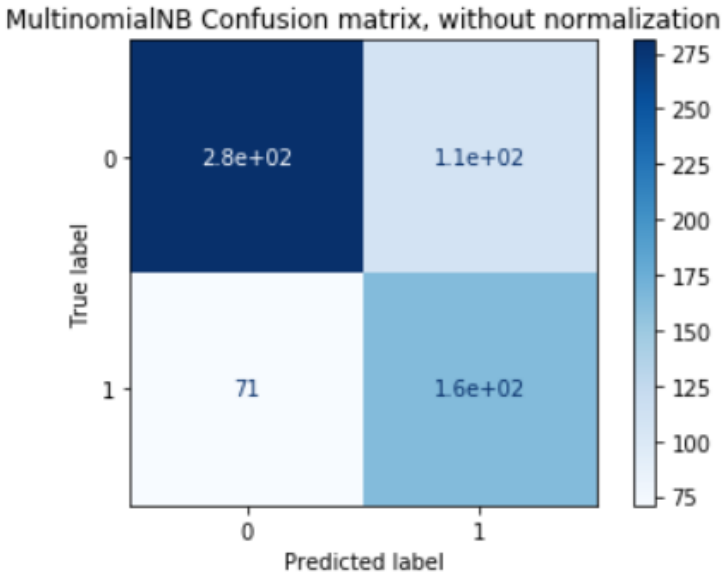
Val data val accuracy：0.75

```
多項式貝氏分類器
test_accuracy 0.7115384615384616
val_accuracy 0.75
```

Test data test accuracy：0.7115

	precision	recall	f1-score	support
0	0.80	0.72	0.76	390
1	0.60	0.70	0.64	234
accuracy			0.71	624
macro avg	0.70	0.71	0.70	624
weighted avg	0.72	0.71	0.71	624

```
MultinomialNB Confusion matrix, without normalization
[[281 109]
 [ 71 163]]
```



七、結果分析與未來展望：

我們以各種主要方法中的最高的 test accuracy 來做綜合比較：

CNN	SVM	XGBoost	Bayesian
five layers	Gaussian radial basis function (RBF)		伯努力貝氏
0.91	0.7596	0.7323	0.7868

可以由此表中看出是 CNN 5-layer 模型準確率最高，所以 CNN 在此種圖形分類與判別上真的有其精準度，另外比較令人驚訝的是伯努力貝氏分類器居然在 test accuacy 在準確率居然比 SVM 和 XGBoost 來的高，有時較簡單的模型也可能有意想不到的功效，但是其他種類的貝氏模型確實準確率偏低，可以知道，了解資料的分布類型對於資料分析也是很重要的一環。

此次期末專題也有許多可以做的方向，像是可以加入 random forest 或其他機器學習的方法來比較彼此間的差異，另外也可以加入 covid-19（俗稱武漢肺炎）的 X-ray 來新增一個肺炎的類別，做更多有關醫療的判別方法，也可以針對 CNN 嘗試不同的激勵函數與損失函數，來比較結果的差異。