Data Mining HW1

Team 5

陳世運 R06725024、陸艾眉 R06725030、劉庭維 R06725031、陳信豪 R06725048

一、作業概述：

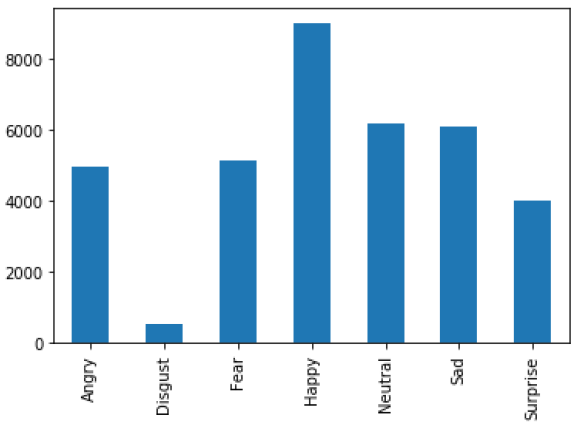
採用 PCA 及 NMF 對人臉資料做降維的前處理，比較兩者套用機器學習演算法後的效能與準確度。

二、資料描述：

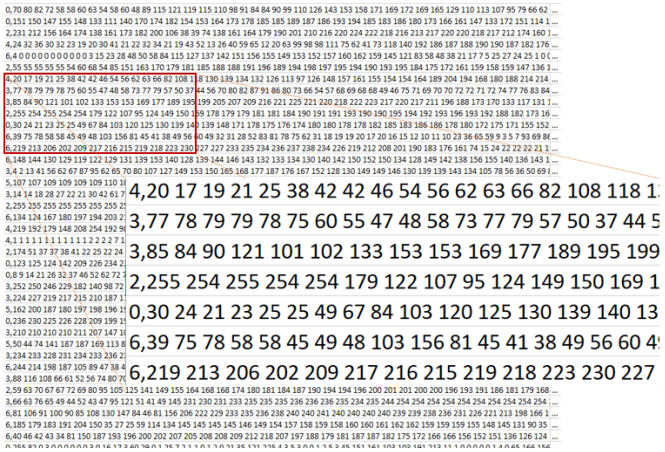
我們從 Kaggle 競賽網址中獲取資料集的 csv 檔案，資料集中共有 35887 筆資料，資料分布情形如圖一所示。每一橫列的一筆資料即代表一張圖片，每張圖片為 48*48 像素的灰階照片，可顯現人臉情緒表情。每筆資料以逗號分隔出兩欄位，第一欄為情緒，為該張圖片的情緒分類編號原檔案共歸類為七種主要情緒，包含：憤怒、噁心、害怕、開心、難過、驚訝、無特殊表情等七種，按照索引值零至六依序排列。第二欄則為像素資料，顯示其組成灰階圖片的像素結構，以空格區分出 2304 個（48*48 個像素）坐落在 0 至 255 間的值，顯示圖片自左上到右下各像素的灰階色彩。（如圖三）

採用之資料集來源：取自 2013 年 Kaggle 上人臉情緒辨識主題的競賽

（https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge）



(圖一)情緒資料分布圖



（圖二）資料集 csv 檔內部資料略圖

三、資料預處理

由於某些情緒在沒有準確演算法下支持，會造成比對結果較相近使資料模糊化，比方憤怒驚訝害怕、難過噁心和無特殊表情在比較時會容易造成混淆，因此我們過濾到只剩**開心與難過**的圖片數據以便對比。於是將數據做洗牌，打亂過後篩選出僅含這兩者各五千筆的數據。此處取出 64 筆部分數據描繪 64 張圖形形成示意圖（圖三），可看出確實僅含兩種表情。

（圖三）僅含開心難過數據之呈現圖

四、實作方法：

1. 套件使用：

   利用簡單高效的數據挖掘和數據分析工具，以及訓練速度快、準確率高的模型，作為這次作業主要使用的工具。

PKG	Version
Python	3.6.3
Scikit-Learn	0.19.1
XGBoost	0.7
Matplotlib	2.2.2

2. 作法採用：

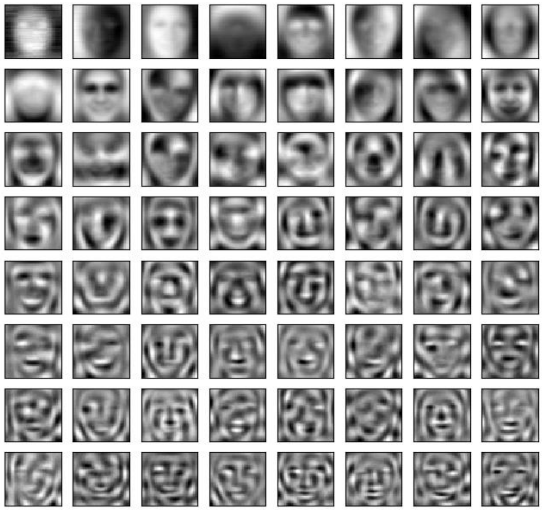
   我們選用了不同的分類模型，如 SVM、RandomForest、ExtraTrees、XGBoost，並且針對各個分類模型做 10-Fold 來檢驗 PCA 和 NMF 的降維效果。

3. 觀察資料

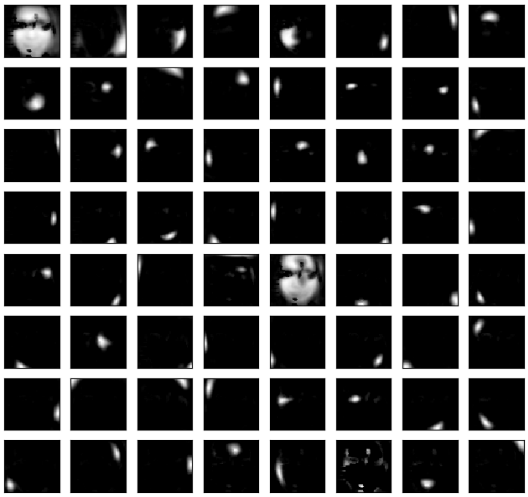
- 降維成像

  首先，我們分別用 PCA 和 NMF 進行維度的下降，從原本的 2304(即沒有經過降維的 48 * 48 的原始維度)以 2 的指數進行維度的調整，分別是 512、256、128、64、32、16、8、4、2 的維度選擇。

  這裡舉例觀察壓縮成 64 維度時，PCA 和 NMF 取出的 components，並做成下圖。



(圖四) PCA eigenface

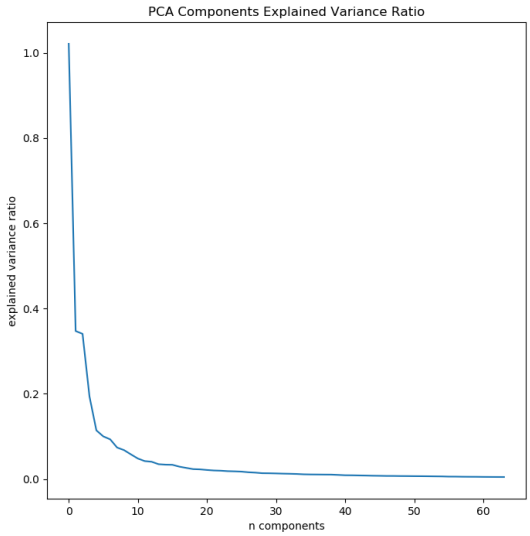


(圖五) NMF eigenface

1) 從 PCA 取出的成分， 確實看出「人臉」的模樣，愈往後的向量其影像愈模糊， 長相也略有不同。但相較於 NMF 的圖形， PCA 所取的成分更為人類能直觀所理解，也比較可以看出較明顯的人臉輪廓。NMF 則比較難以用肉眼進行辨識。
2) PCA 和 NMF 在成像上都是愈上層(即左上)的 eigenfaces 越有真實人臉的輪廓，而越往右下的成像則愈不明顯。

- 解釋變量

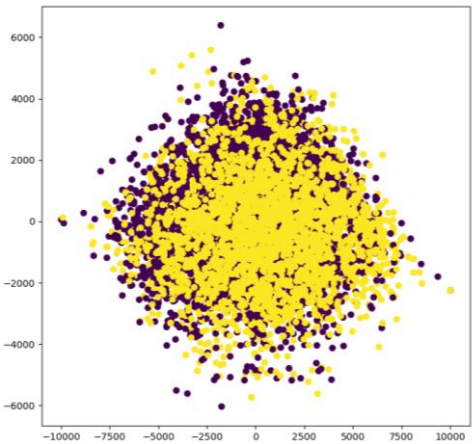
  再者，我們觀察 PCA 的解釋變量，並繪製如下圖。從圖可以發現越後方的 components 能解釋的變異量越少，為指指數衰減，正好可以呼應圖四的成像，從左上清晰的輪廓可以解釋的含量大到右下模糊的型態變化。



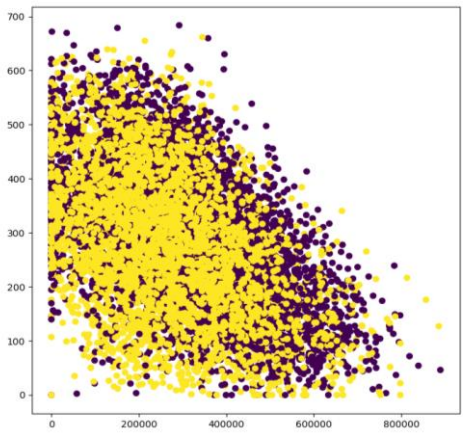
(圖六) PCA components 解釋的變異量

- Scatter plot 分布

  透過 PCA 和 NMF 將資料下降成二維並做 scatter plot，雖然成效不是很明顯，但仍可發現兩種情緒在分佈會有稍微的不同。在 PCA 的分布中，紫色的點略偏於左；在 NMF 的分布中，紫色的點則略偏向於右上。因此，我們可以知道維度下降幅度太大，即使資料的分布還是會有些微不同，但是得到的資訊太少也容易造成資料難以進行準確的判斷。



(圖七)PCA scatter plot



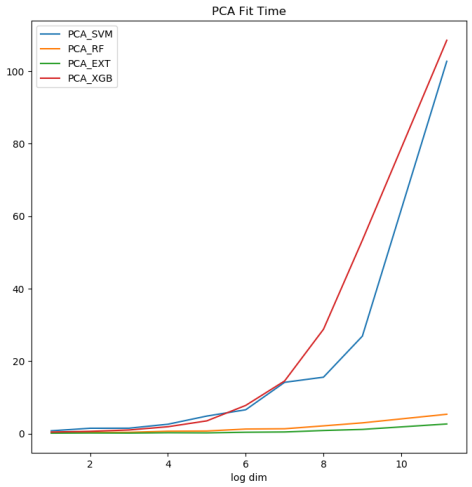
(圖八)NMF scatter plot

4. 分類器檢定

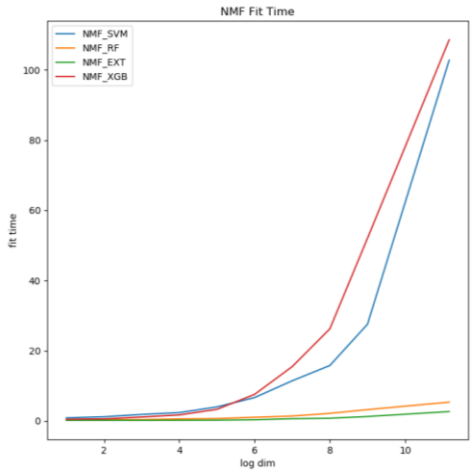
我們利用 SVM、RandomForest、ExtraTrees、XGBoost 分別檢驗 2304, 512, 256, 128, 64, 32, 16, 8, 4, 2 維度。

- Fit time

我們發現，在降維後 SVM 和 XGBoost 大幅度減少了 model 訓練時間，而 RandomForest 和 ExtraTrees 也有明顯下降，證明降維度可以讓我們加快我們的學習的速度，有效地提升訓練效率。



(圖九)PCA Fit Time



(圖十)NMF Fit Time

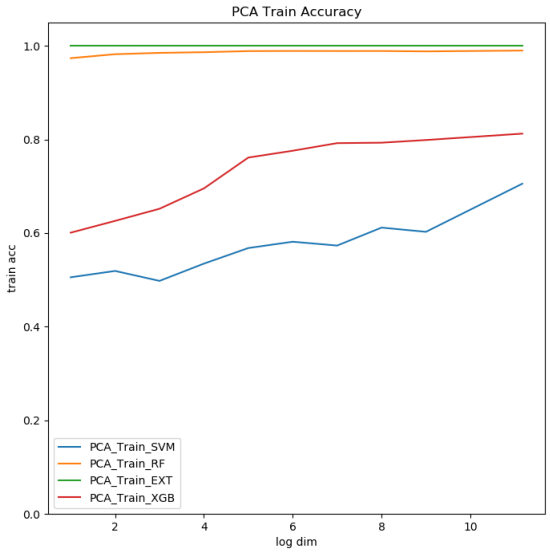
- Accuracy

執行降維後在 training 的 ACC 準確率。我們發現三個現象：

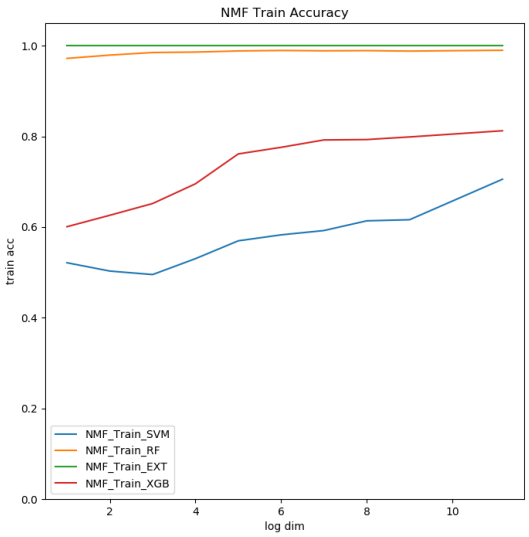
第一是 ExtraTrees 分類器效果最好，RandomForest 僅略低一點，兩者效果都趨近於 1.0。而 XGBoost 可以得到 0.6~0.8 的準確率；較差的是 SVM，可以得到 0.5~0.7 的準確率。

第二是在 PCA 和 NMF 上的數據曲線，4 個分類器的呈現趨勢相同；而在 ExtraTrees、RandomForest、XGBoost 方面，因為三種 model 是 boosting，所以在呈現上不論是 PCA 還是 NMF 準確率都相較於穩定。而在 SVM 分類器之下，因為不是 boosting 模型，所以會有 PCA 的準確率會比較曲折，過程中會有下降的現象；而在 NMF 中只有較小維度略有下降，其餘是穩定上升的。

第三是在整體而言，可以發現維度降得越低，XGBoost 和 SVM 的 trainACC 也會跟著越低，而在跟 test ACC 比較後發現 overfitting 的 ExtraTrees 和 RandomForest 就比較沒有受影響，準確率維持在一定水平。



(圖十一) PCA Train Accuracy

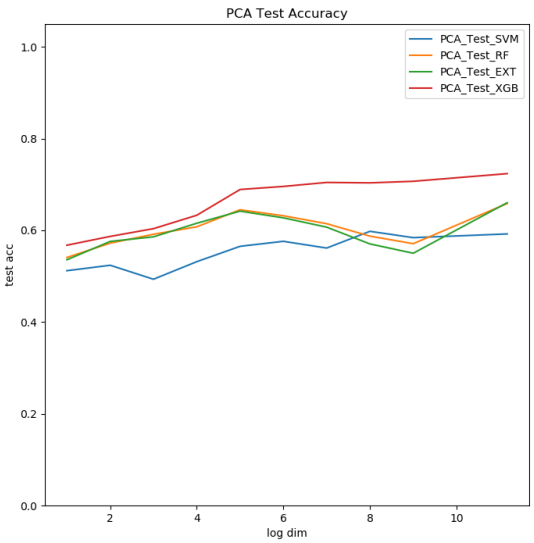


(圖十二)NMF Train Accuracy

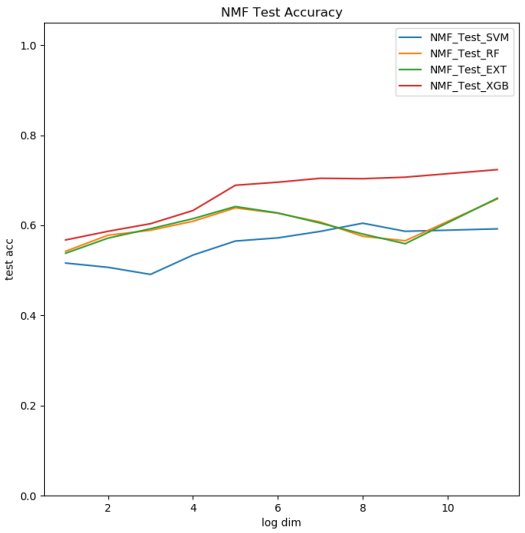
執行降維後在 testing 的 ACC 準確率。我們發現兩個現象：

第一是 XGBoos 分類器效果最好，約 0.6~0.7；ExtraTrees 和 RandomForest 略低一點，兩者效果約在 0.6 上下；較差的是 SVM，準確率大約在 0.5~0.6。

第二是維度降得越低，test ACC 也跟著下降。



(圖十三) PCA Test Accuracy

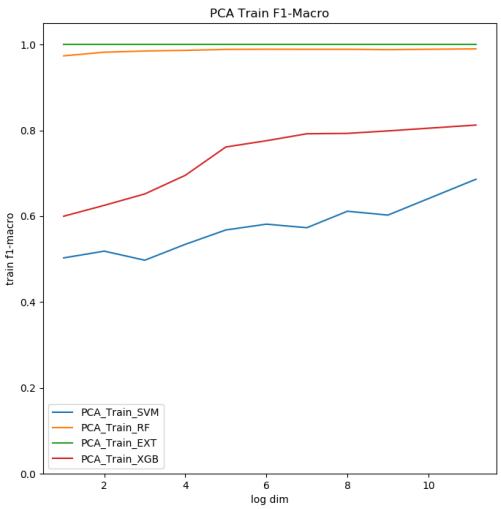


(圖十四)NMF Test Accuracy

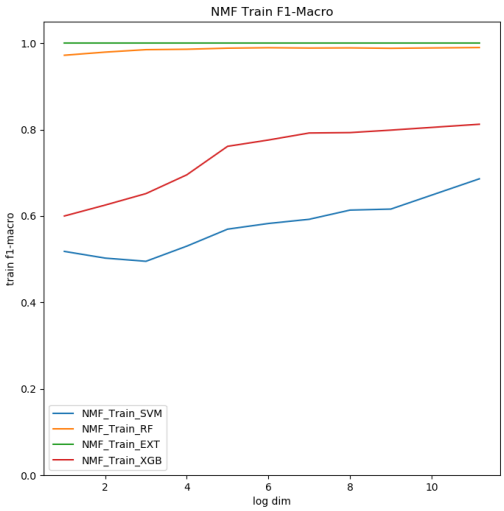
- F1-Score

  這裡比較各分類器在 training 和 testing 之下 F1-macro 的效果。

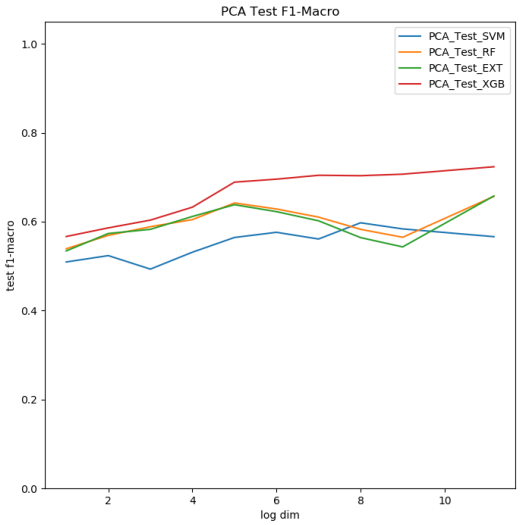
  在一開始取資料時，分別取出開心和難過的表情各 5000 筆資料，因此不論是 training 或是 testing 在各分類器下的結果和 ACC 的表現差不多，在這裡就不多加贅述。



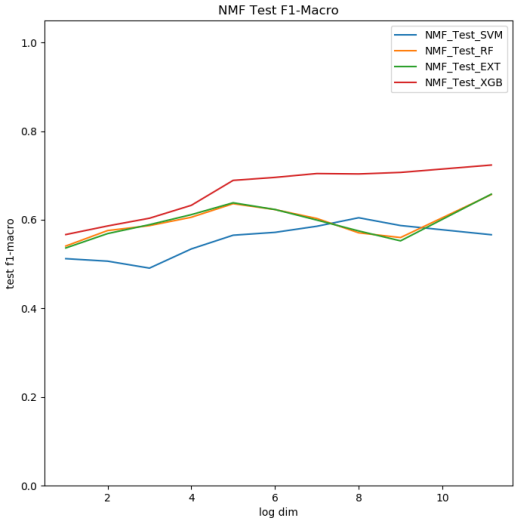
(圖十五) PCA Train F1-Macro



(圖十六)NMF Train F1-Macro



(圖十七) PCA Test F1-Macro



(圖十八) NMF Test F1-Macro

- 降維時間

  列舉出在 PCA 和 NMF 中下降維度所需的時間。由表可以看出，PCA 降維速度比 NMF 快很多。而基本上，降的維度愈低，所需收斂時間愈短。

Decomposition Dim	PCA	NMF
512	13.5825	1205.2244
256	6.4271	438.1547
128	11.0945	200.6971
64	3.1308	69.4797
32	4.0512	39.9522
16	2.0576	26.4505
8	1.6477	19.1031
4	1.8589	18.4982
2	1.5582	20.4613

五、結論

透過 PCA 和 NMF 降維，可以降低訓練時間（特別是對 XGB 和 SVM），而 trade-off 則是準確率也會下降。因此，在做機器學習時，如果今天的 feature 數量非常多，（如 image pixels、text tfidf、超商中顧客對產品的購買資訊 … 等等）我們就可以考慮使用 PCA 或 NMF 來進行降維，並在盡量不失去準確度的情況下，選擇最合適的維度來減少 model 所需的 training 時間。