Lab4

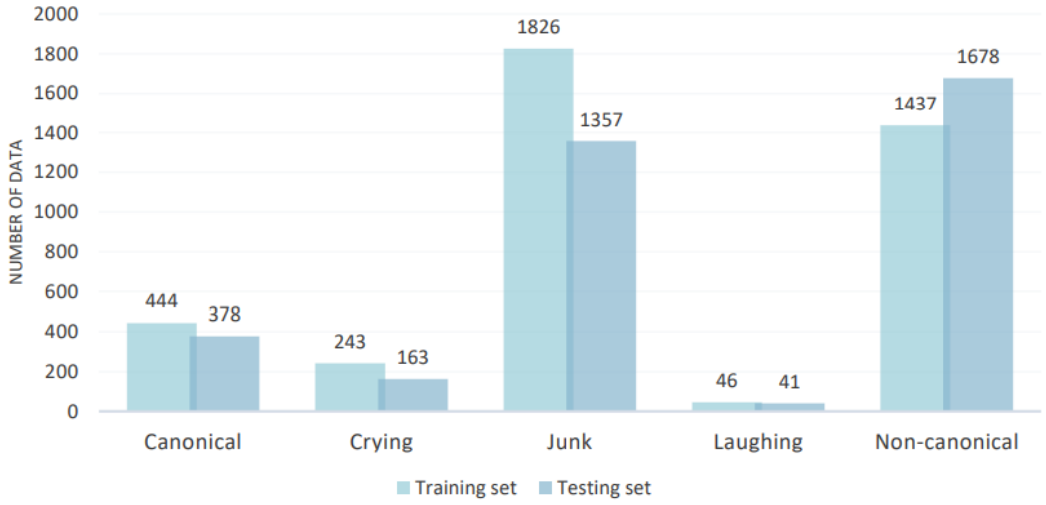
109061217 林峻霆

Abstract

綜合前幾個 Lab 所學的知識(MFCC, Cross-Validation and SVC 等)去對嬰兒的各種聲音進行辨識，並比較各種 model 下的結果差異。

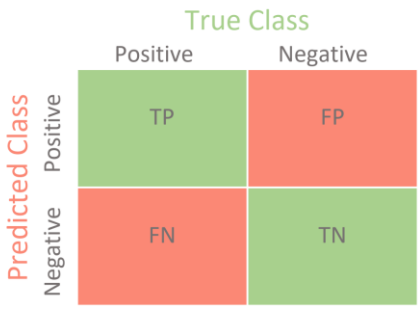
Lab objective

這個 Lab 用的 dataset 是 The INTERSPEECH 2019 Computational Paralinguistics Challenge，他主要包含嬰兒會發出的五種聲音，這五種聲音在 train set 和 test set 的分布如下圖。可以發現，這是一個相當不平衡的 dataset。



取自助教提供的講義

我們的目標在於對這五種聲音進行區分，而我們對 model 的評分標準來自於 F1 score，其計算方式如下圖所示，F1 score 不僅考慮 accuracy，也要注意 precision 和 recall。



F1 is calculated as follows:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

where:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

In "macro" F1 a separate F1 score is calculated for each `species` value and then averaged.

Method

這個部分介紹我有嘗試那些 feature, classifier 以及 preprocessing 的方法，簡單講解他們背後的原理。整個 model training 的部分會留到下一個部分做說明：

A. 我有嘗試的 feature 如下：

1. Root-mean-square
2. Spectral Centroid
3. Spectral Bandwidth
4. Zero-crossing rate
5. Spectral roll-off
6. Mean of MFCC
7. STD of MFCC
8. Δ and Δ^2 of MFCC

簡單介紹幾個名字比較沒那麼直觀的 feature：

2. Spectral Centroid

類似於物體重心的概念，但是是在頻譜上做計算，計算方法如下：

$$\text{Centroid} = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

透過 FFT 我們將頻譜切成 N 等分， x(n)代表第 N 等分的 magnitude，而 f(n)則是第 N 等分代表的 frequency。Spectral Centroid 跟聲音的音色有很大的相關性。

5. Spectral roll-off

整個 frame 達到特定百分比能量的頻率，以 95%為例：

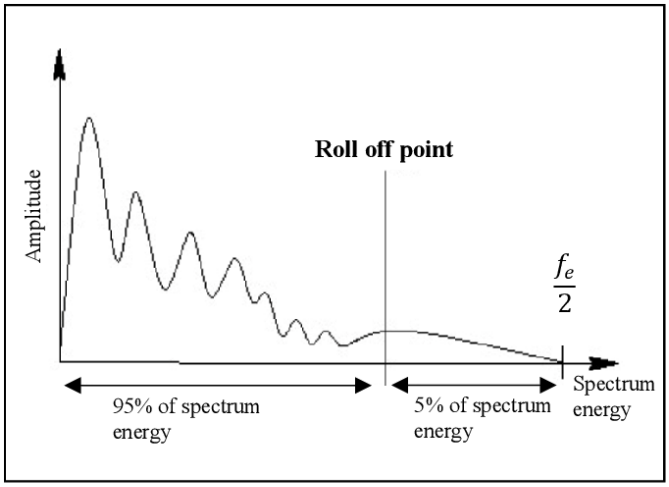


Fig. 2. Spectral Roll off Point [16]

8. Δ and Δ^2 of MFCC

能顯示 MFCC 在時間上的變化程度。

B. 而我有去嘗試的 classifier 如下：

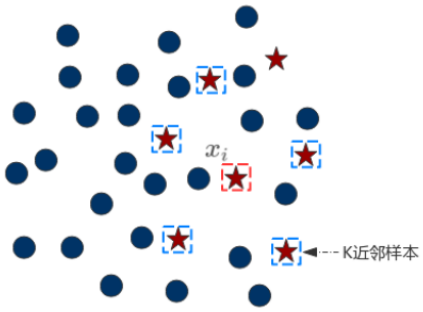
1. Linear SVM
2. SVM
3. Logistic Regression
4. Decision Tree
5. KNN

另外，因為這次的 dataset 各類別的資料量是不平衡的，所以我有嘗試一些額外的資料預處理，分別是 over_sampling(SMOTE, BorderLine SMOTE, ADASYN) 和 under_sampling(ENN, Tomek link)。這邊解釋一下什麼是 over_sampling 和 under_sampling，這兩個方法常運用在 imbalanced dataset 中，over_sampling 就是透

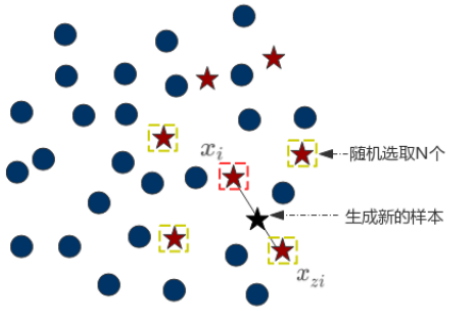
過人造的方式將數量較少的類別的 data 增多；under_sampling 則是將數量較多的類別的 data 減少，兩者的目的都在於將模型的區分能力更加提升，下面解釋他們的原理：

1. **SMOTE (Synthesized Minority Oversampling Technique)**

   對少數樣本的數據點取其 K 個 neighbor，並透過這些點生成新的數據點



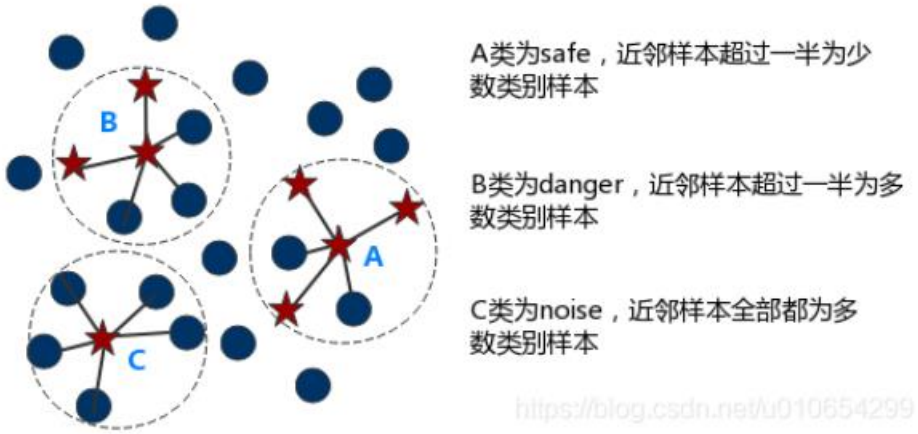
图a 计算Xi的K近邻样本



图b 合成新样本

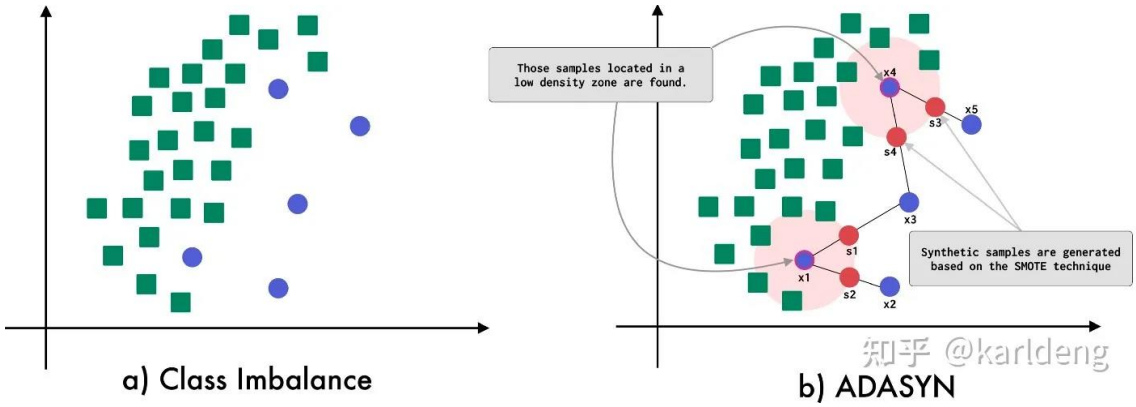
2. **Borderline SMOTE**

   SMOTE 的變化版本，顧名思義我們只考慮邊界上的少數類別數據點來合成新數據點，具體如下圖。我們將少數類別分成三類：safe、danger、noise，然後只對 danger 的少數類別作 SMOTE。



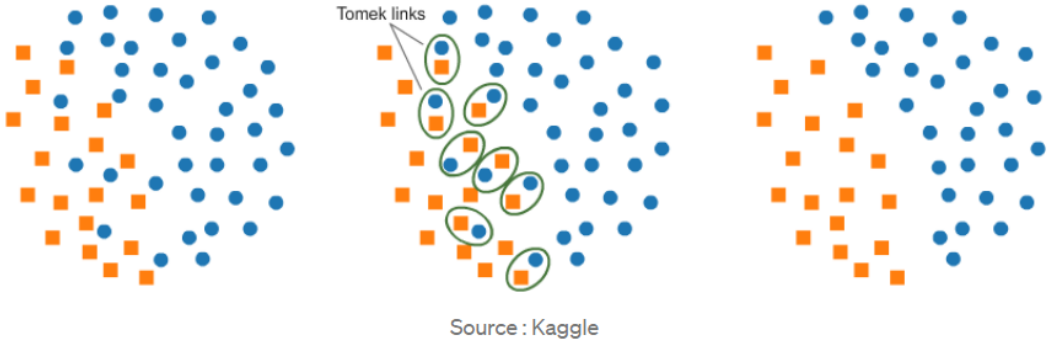
3. **ADASYN (Adaptive Synthetic Sampling)**

   ADASYN 會根據數據集的分佈情況，自動判斷每個少數類別樣本需要合成多少新的樣本，而不是像 SMOTE 那樣每個少數類樣別本合成的樣本數量都一樣。少數類周圍的多數類樣本越多，則其權重也就越大，生成的點就越多。



4. **Tomek Links**

   Tomek Link 表示不同類別之間距離最近的一對樣本，Tomek Link 中的兩個樣本可能其中一個是噪音或雜訊，也可能是兩個樣本都在邊界附近。這個方法的想法在於，剔除邊界那些鑑別度不高的樣本。



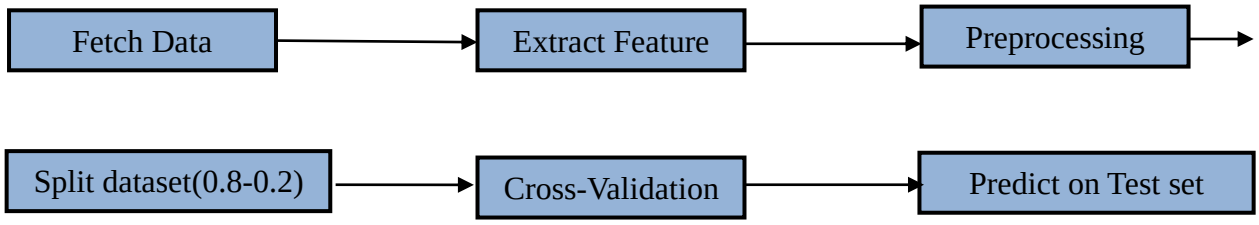
5. **ENN (Edited Nearest Neighbor)**

   與 Tomek Links 觀念相同，但改為對多數類樣本點尋找其 K neighbor，若有某個 threshold 之上的 neighbor 不屬於多數類樣本，就把這個樣本點剃除。

一般來說，我們都會結合 over_sampling 和 under_sampling，over_sampling 提高資訊量、under_sampling 消除雜訊。在這次的 Lab 中，我有嘗試單獨用 SMOTE、SMOTE + ENN 和 SMOTE + Tomek Links。

- ## Flow Chart

訓練所使用的流程圖如下，首先我們先 extract feature，並做 preprocessing 如上一部分所述，接著我們將資料集按照 0.8 : 0.2 的比例分成 train set 和 validation set，這樣做的原因在於我們並沒有我們最終要做 predict 的 testcase 的正確答案，若直接對整個 dataset 做 cross-validation 會沒辦法發現有 over-fitting 的問題，因此切 dataset 的一部分來當作 validation set 來檢驗我們做 cross-validation 的結果。



- ## Result and Discussion

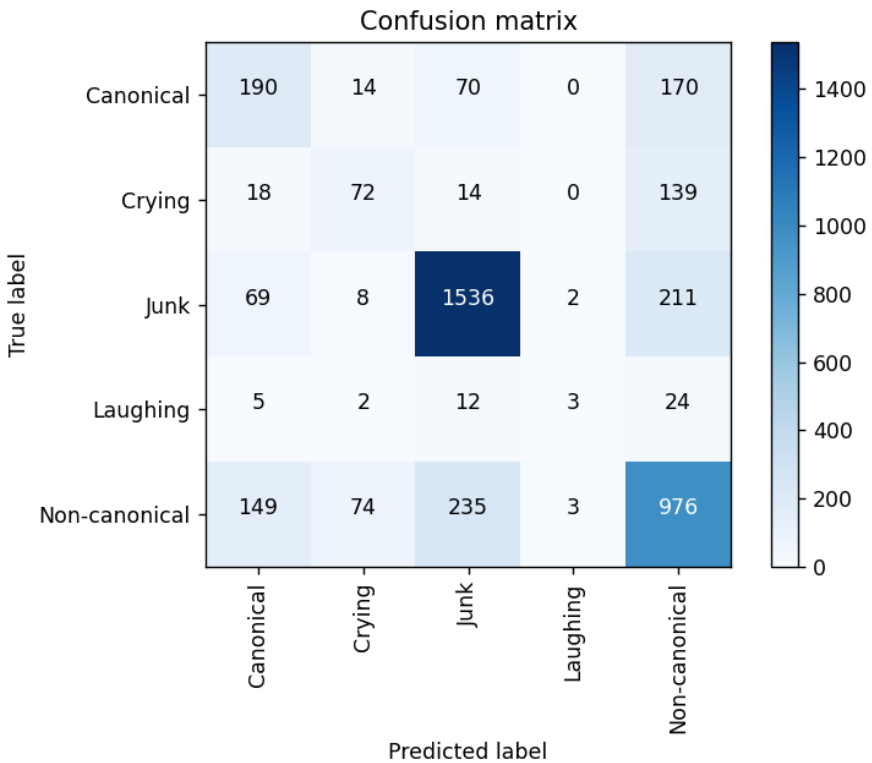
A. Best Result

Feature : 40 MFCC pick 20 to calculate MEAN, 40 to calculate STD → (MEAN, STD)

Classifier : SVC (C = 1.5, kernel = rbf)

10-fold Cross-Validation

Result : Cross-Validation = 68.8 % / Test Set = 67.4%



Canonical	Crying	Junk	Laughing	Non-Canonical	Mean F1 score
43.40%	34.90%	83.20%	11%	66.00%	68.80%

B. Different Cases

Baby Sound (F1 score %)							
	Canonical	Crying	Junk	Laughing	Non-Canonical	Mean F1 score	Test set
LogisticRegression	37.80%	30.80%	80.30%	0%	62.60%	65.30%	66.00%
LinearSVC	23.50%	16.70%	79.70%	0%	64.80%	63.40%	60.20%
SVC (rbf)	43.40%	34.90%	83.20%	11%	66.00%	68.80%	67.40%
SVC (linear)	35.60%	25.60%	80.70%	18.20%	62.40%	65.00%	x
SVC (poly)	45.80%	34.60%	79.80%	0%	62.70%	66.20%	x
SVC (sigmoid)	28.20%	27.80%	72.90%	0%	54.30%	57.60%	x
SMOTE + SVC (rbf)	43.80%	45.90%	75.30%	0%	60.50%	63.80%	58.10%
SMOTE + Tomeklink + SVC (rbf)	43.10%	47.70%	76.30%	0%	61.70%	64.70%	66.40%
SMOTE + ENN + SVC (rbf)	32.60%	37.00%	50.80%	9.10%	32.40%	40.80%	x
ADASYN + TomekLinks + SVC (rbf)	50.00%	33.30%	80.70%	40.00%	66.60%	66.60%	62.20%
Decision Tree	27.70%	24.70%	74.10%	0%	54.40%	58.00%	x
KNN	39.80%	27.80%	80.70%	0%	63.40%	66.30%	60.60%

上面是我有嘗試的各種 classifier 加上 preprocessing 的組合，並透過 cross-validation 調整參數以及採用的 feature，找到的最好結果。下面對這個結果做一些分析：

1. SVC, Linear SVC, Logistic Regression > KNN > Decision Tree。在 performance 上 SVC 和 Logistic Regression 的表現相較於其他的 classifier 好。
2. resample 後的結果跟預期上的效果有差異，並沒有很好的效果，雖然確實讓少數類的 performance 提高，但同時也降低多數類的 performance，導致最後的 mean F1 Score 相較於沒有 resample 的結果沒有 improvement，甚至還有些退步 (66.4% < 67.4%)。
3. SVC 中不同的 kernel 對於各類別的區分能力不同，可以發現 linear 時對 Laughing 的區分能力較好，但其對 Canonical 的 performance 就比較差；poly 則對 Canonical 的區分能力較好，但其他項目則較差；而 rbf 則比較趨近於兩者之間，他對 Junk 和 Non-canonical 的區分能力也是所有 kernel 中最好的，這也是其 mean F1 score 最高的原因。實作上應該能針對各種不同的 case 去選擇 model 來達到較好的成效。
4. ADASYN 對 laughing 的 improvement 是可觀的，從原先只有 10%~20%之間的 performance 提升到 40%。或許可作為未來進一步實作的經驗。

C. Solution for imbalanced dataset

如前面所述，我有嘗試去 resample dataset，總共嘗試三種組合：

1. SMOTE
2. SMOTE + Tomek Link
3. SMOTE + ENN
4. ADASYN + Tomek Link

其中，SMOTE + ENN 的效果不太好，僅有 40%的 F1 score，我認為原因來自於 ENN 剔除太多多數類樣本點，可以看到 SMOTE + ENN 在 Junk 和 Non-Canonical 的 performance 都低上許多，但其特殊的點在於相較於大多數的 case，他能區分出少量

的 Laughing。而另外三個 case 雖然結果還可以，ADASYN 對於 laughing 的 improvement 是巨大的，但是相較於 best result(沒做 resample)的整體 mean F1 score 仍然比較低。我認為原因來自下面三點：

(i) SMOTE、ADASYN 再增加樣本點的同時也會增加 over-fitting 的可能性，且會增加樣本中的雜訊，影響到多數類的 performance，可以發現 Junk 和 Non-canonical 的 performance 都下降了。但另一方面也可以發現 Crying 的 performance 明顯上升了。

(ii) 原先的少數類樣本點本身就分布位置就有很大的變異，並非集中在一起，導致 SMOTE、ADASYN 生出的點大多是雜訊，對訓練模型不僅無效，甚至可能造成反效果。

(iii) 少數類別因為數量較少，對 mean F1 Score 的影響不大，若以提升 mean F1 score 為目的應把目標放在提升多數類的區分能力上。

D. Other Problem or Discovery

Feature 的選擇是這次遇到的一個問題。起初，我只嘗試 MFCC 的 Mean 和 STD，但由於換各種 classifier 和參數對 performance 的 improvement 不大，因此我才考慮要不要嘗試其他 feature。然而，要使用哪些 feature 成為一個問題。由於繳交次數有上限，因此只能使用 cross-validation 多次嘗試。而在這個過程我發現，MFCC 的 STD 是一個對 performance 相當重要的一個 feature，沒有採用這個 feature 的話 performance 往往連 60%都不到。另外，我也發現各種 classifier 對不同維度的 feature 處理能力不同，SVM 對高維度(採用較多 feature)處理能力相較於其他的 classifier 還要出色。

● Conclusion

這個 Lab 中我們實作一個 ML 的模型，提取嬰兒語音的特性(ex. MFCC, roll-off, rms 等)，並嘗試各種 classifier 去做語音辨識。最後我們用 MFCC 搭配 SVC 在 Kaggle 上達到 67.4%的 Mean F1 score。

這次的 Lab 讓我實際體驗整個 ML 的過程，從資料的特徵選擇、預處理到模型的建立和訓練，透過嘗試各種可能在 Kaggle 上提高 performance 也相當有趣，只可惜有時候 test set 上的結果不如預期、performance 停滯不前加上本身背景知識不夠，感覺怎麼調整方法都沒辦法有更大的 improvement。此外，我認為我對聲音的認識也不夠多，performance 要做得好的條件也包含 feature 要挑的好，然而我對我所採用的 feature 並沒有很深的了解，只能隨便嘗試各種組合。希望未來在對 ML 和 feature engineering 有更多的了解後能重新挑戰這個問題。

- Reference
  - ML/DL 資料前處理 | MaDi's Blog (dysonma.github.io)
  - 資料科學(一)：處理不平衡資料幾種方法. 在經典的分類問題中(classification… | by LUFOR129 | Medium
  - SMOTE + ENN : 解決數據不平衡建模的採樣方法. SMOTE + ENN : A sampling method that… | by Edward Tung | 數學、人工智慧與蟒蛇 | Medium
  - imbalanced-learn documentation — Version 0.9.1
  - iT邦幫忙 SVC, KNN, Decision Tree, Logistic Regression
  - 教授與助教的講義