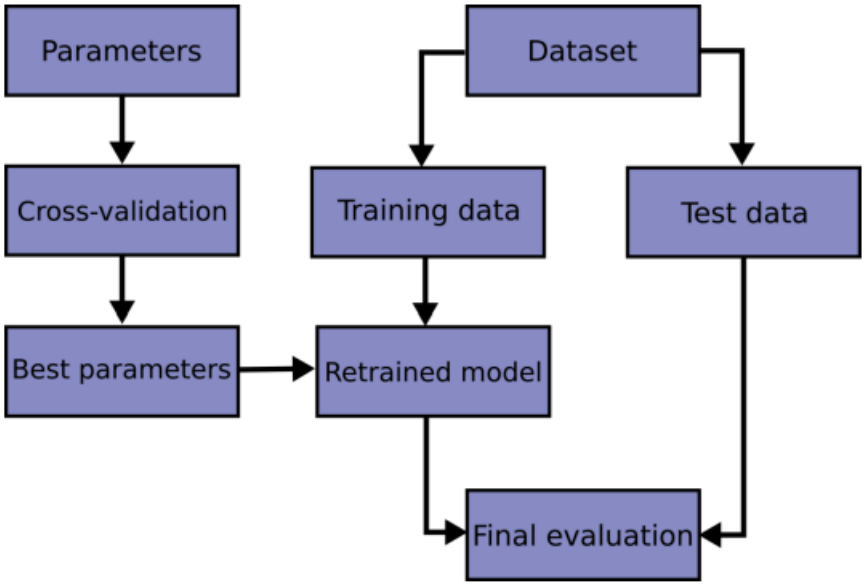
Lab3 109061217 林峻霆

Design Specification

在前幾個 Lab 中，我們介紹 MFCC 並理解它是一個不錯的語音資訊，也常用在語音辨識上。因此，在這個 Lab 中，我們將時做一個 ML 的模型，並運用 MFCC 當作語音特徵來進行語音辨識。

Design Implementation

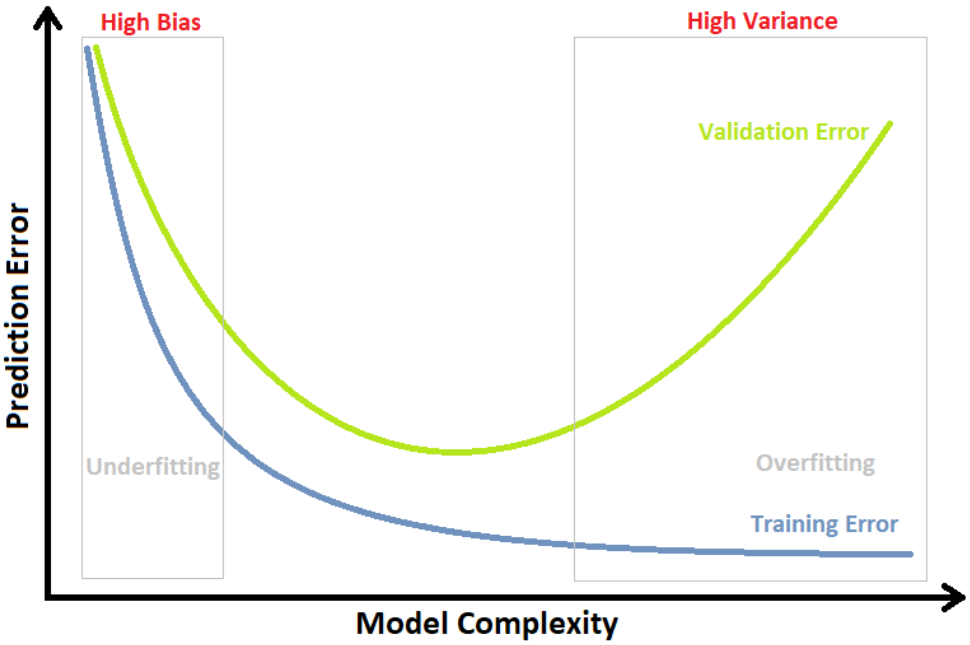
這個部份我會著重於講解整個 model 是如何建立的，以及各個步驟用了哪些 Machine Learning 的工具。我們所做的 ML 的流程圖如下：



取自助教提供的講義

1. Cross Validation

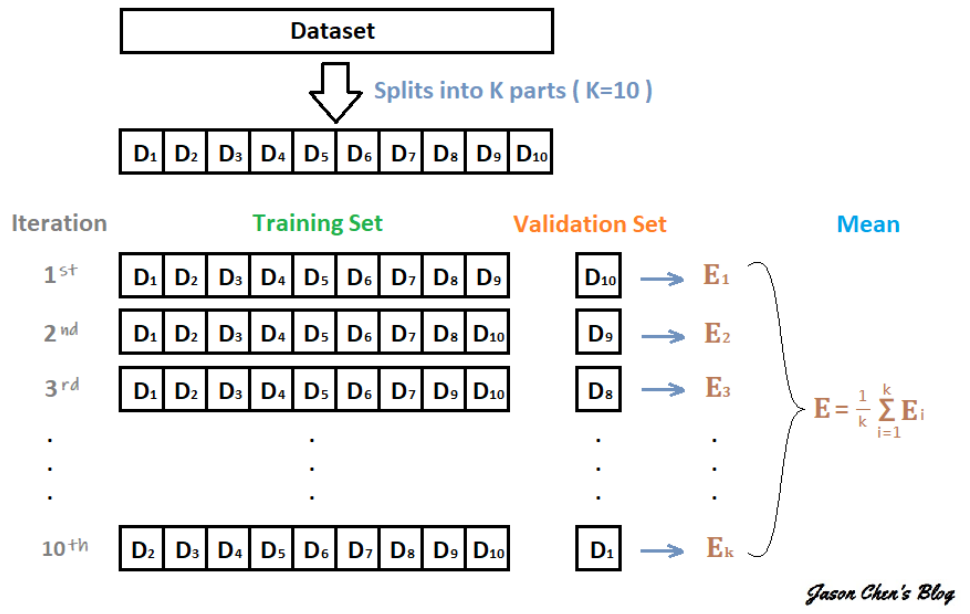
在訓練模型時，我們會習慣**將 Dataset 分割成訓練集(Training set)以及驗證集(Validation set)**。我們透過訓練集來訓練模型，驗證集來判斷模型的好壞。在 ML 中，我們希望我們的模型能達成 low Bias Error 和 low Variance Error，但同時我們可能會遇到 over-fitting 和 under-fitting 的問題，訓練集和驗證集就能很好的處理這個問題。



取自 Jason Chen’s Blog

一般來說，我們**通常取一個 DataSet 的 5%~20%來當作驗證集，但在某些狀況下(ex. 小樣本)，這種作法不夠 general (因為樣本數不夠，不符合大數法則，代表性不足)**。因此，這時我們有一個做法就是做交叉驗證(cross-validation)。

K-fold Cross-Validation 就是將整個 Dataset 切成 K 等分，每等分輪流當驗證集，其他當訓練集訓練 K 次，算出 K 個 Validation Error 做平均來判斷模型的好壞。

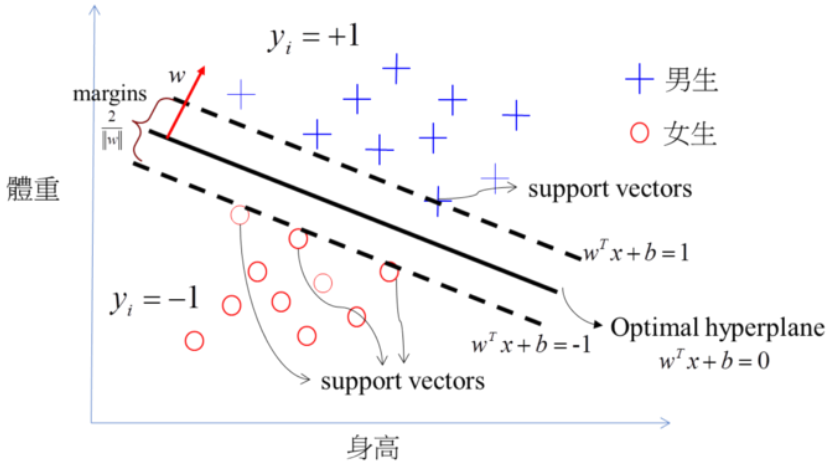


取自 Jason Chen’s Blog

2. Support Vector Machine(SVC)

假設我們想對兩種物品做區分，**SVM 就是在找一個 hyperplane(可能是一條線、一個平**

面或更高維的東西)將我們要區分的兩個東西分越開越好(與 hyperplane 的距離越大)。



這事實上就是一個最佳化的問題，以上面的圖為例，我們將男女分類為：

$$\begin{cases} 男生 = F \geq 1 \\ 女生 = F \leq -1 \end{cases} where\ F = y_i w^T x_i + b \quad \rightarrow \quad Condition\ 1$$

SVM 在找的 Optimal hyperplane 就是希望區隔兩類之間的邊界(2 / |w|)可以越大越好

$$\max_w \{2/\|w\|\} \rightarrow \min_w \frac{1}{2} w^T w$$

$$subject\ to \quad y_i(w^T \boldsymbol{x}_i + b) \geq 1, \qquad \forall i = 1, \ldots, n$$

然而，在**真實情況上這個 hyperplane 可能不會這麼好找，因此我們可以允許以下這個情況：某些的 data point 可以掉在 margin 內部，我們稱這些點為 support vector。**可以發現這些 support vector 就不會滿足我們原先的 condition 1，因此我們得對 Condition 1 做一些修改，額外加入一個容忍值(slack variable ζ，ζ值越大代表我們容忍的範圍越大，support vector 與 hyperplane 的距離可以越大)。原先的 Condition 1 變成：

$$\begin{cases} 男生 = F \geq 1 \\ 女生 = F \leq -1 \end{cases} where\ F = y_i w^T x_i + b + \xi_i \quad \rightarrow \quad Condition\ 1$$

同時我們的 optimal hyperplane 也變成下圖所示。其中，**C 被稱作權重/懲罰參數，能夠幫助來調整 slack variable，進而改變 support vector，找出一個較好的 hyperplane。**C 越大，在訓練樣本中的準確率會越好，越貼合 Dataset，但這樣就可能發生 over-fitting 的問題；相反的，C 小的時候，我們能獲的一個比較 general 的 model，能容忍多一些的 error。

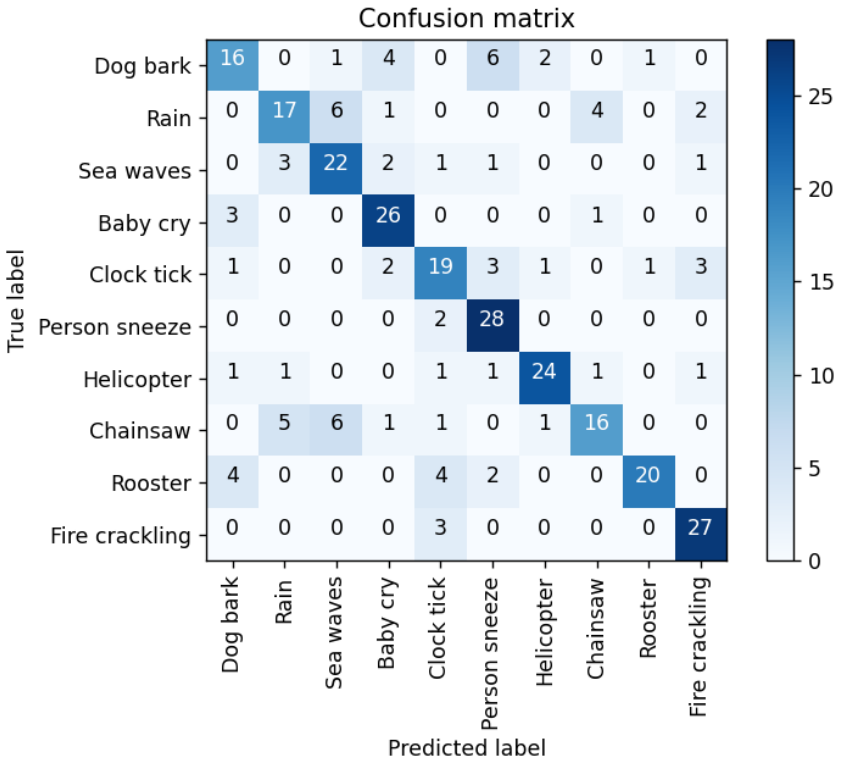
$$\min_{w,\xi_i} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$$

$$subject\ to \quad y_i(w^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, \ldots, n$$

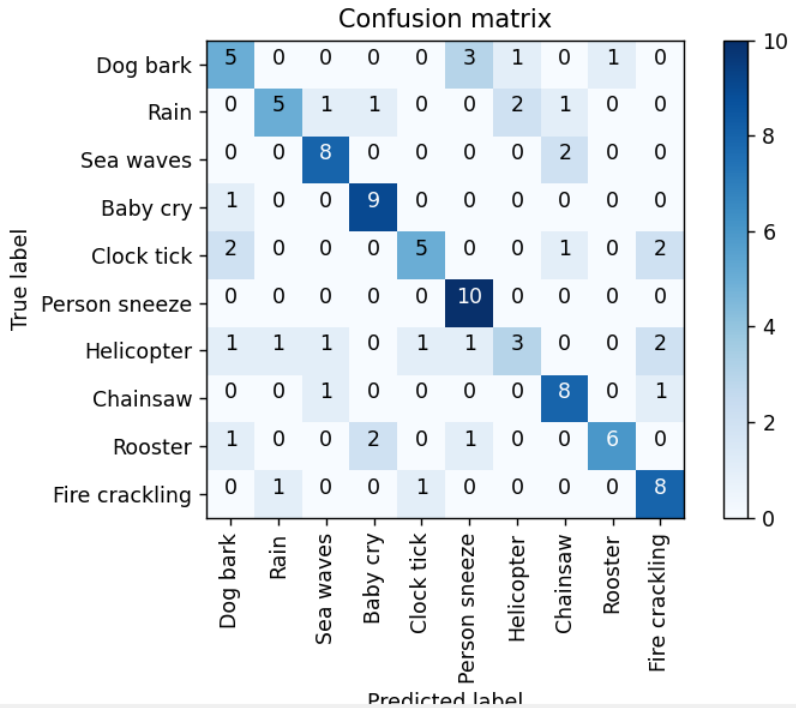
- ## Discussion

接下來我們展示採用各種 features, statistics function 以及 classifier 做出的結果。對於每個 model 我會根據他們在 testset 上的表現作一些討論

1. 首先是要 demo 的部分，這裡我們**採用的 MFCC 延時間軸的 mean 來當作 feature 送入 model 做訓練：**

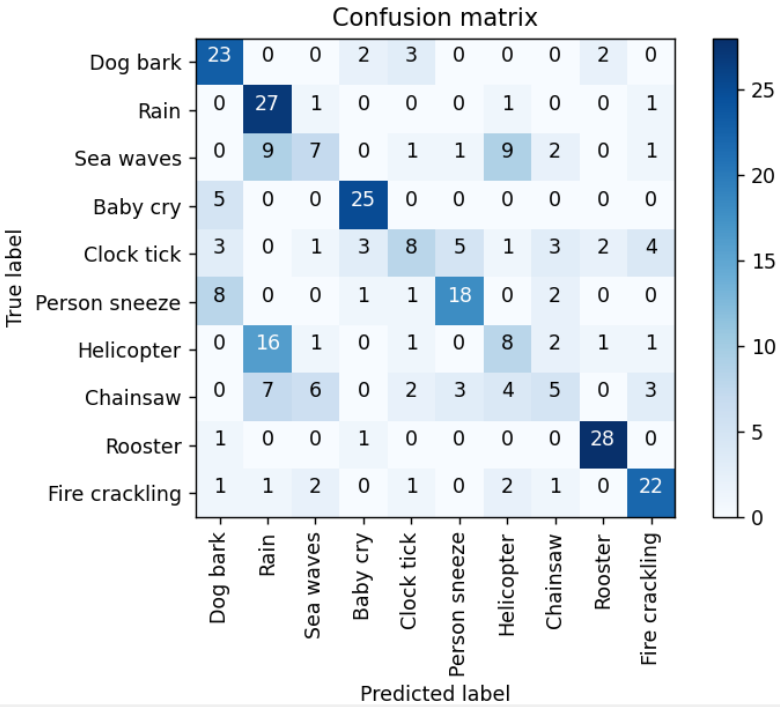


ACC = 71.7%

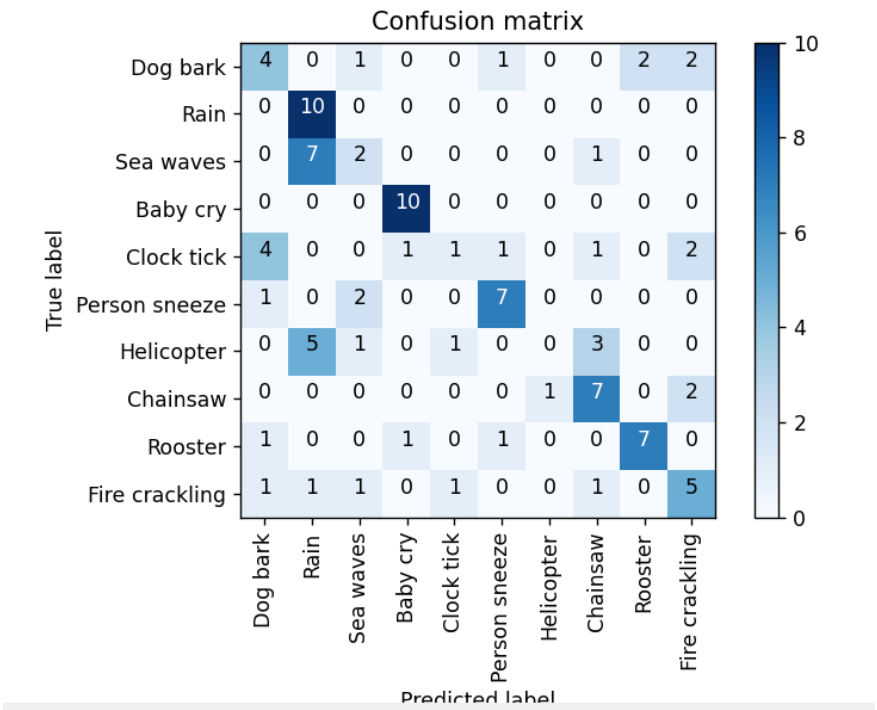


ACC = 67%

接著我們嘗試**用 MFCC 延時間軸的 standard deviation 當作 feature，結果如下：**



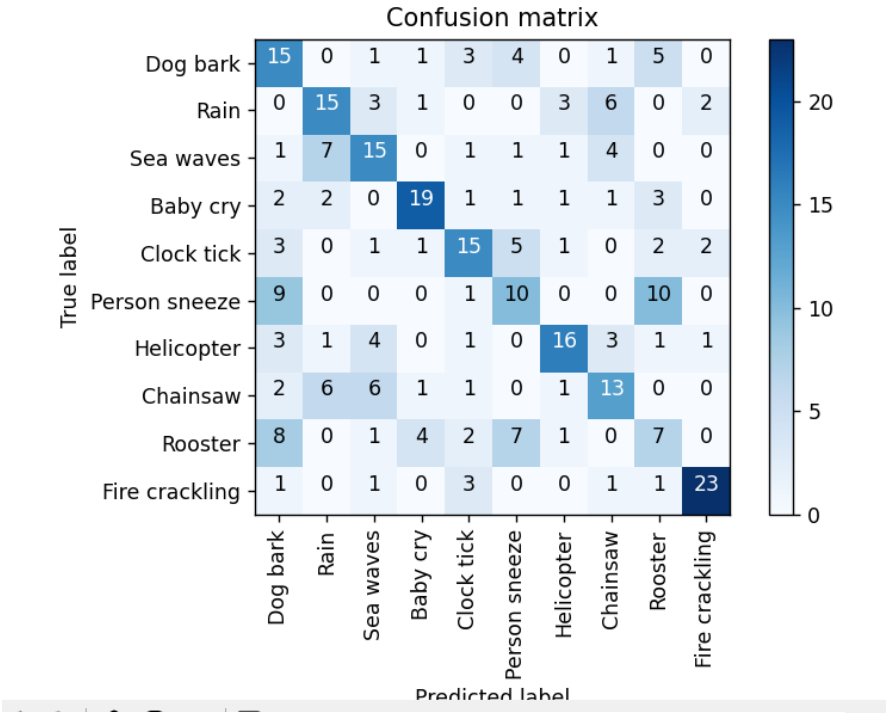
ACC = 57%



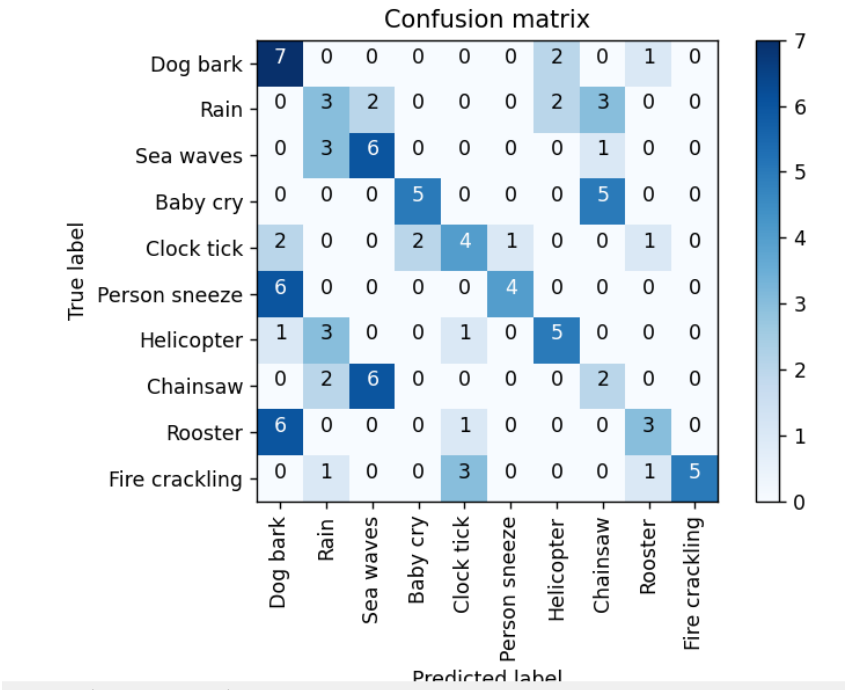
ACC = 53%

比較兩者的結果可以發現以 mean 當 feature 訓練出來的 model 效果比較好。但兩者仍然與我們這次的目標(ACC > 75%)仍有些許的差距。mean 的 model 在直升機的聲音方面表現相當差。而 **standard deviation 的 model 雖然 accuracy 低，它在 baby cry 方面卻有不錯的辨識能力**，此外它對於雨聲的 Type II Error 也相對多上許多。

2. 我們嘗試使用其他的 statistic function(ex. Median)，結果如下：



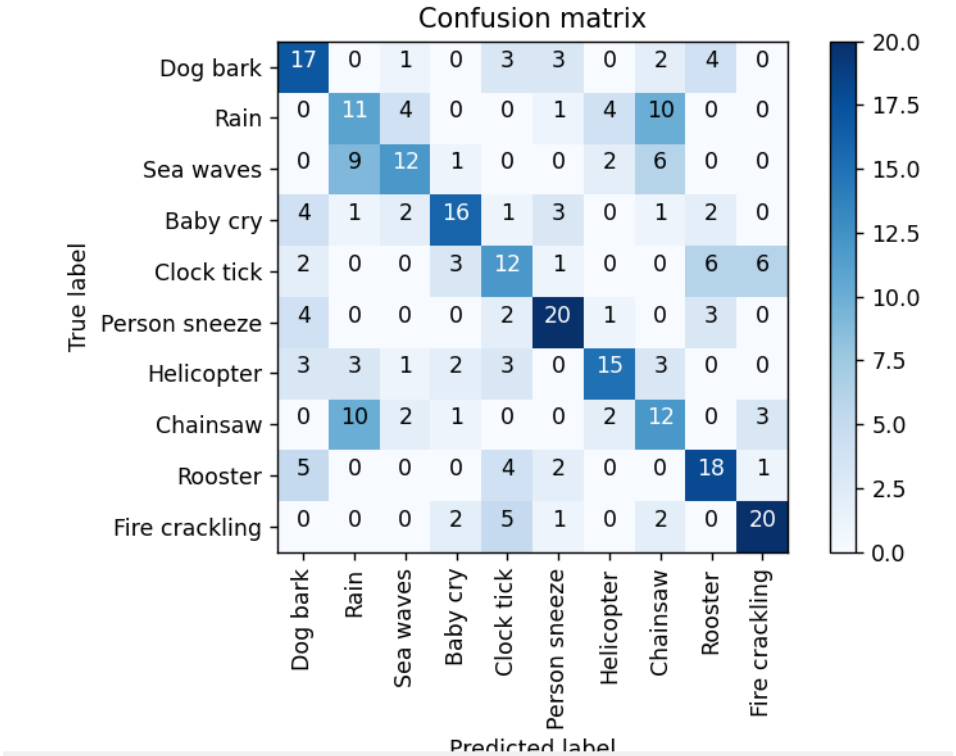
ACC = 49%



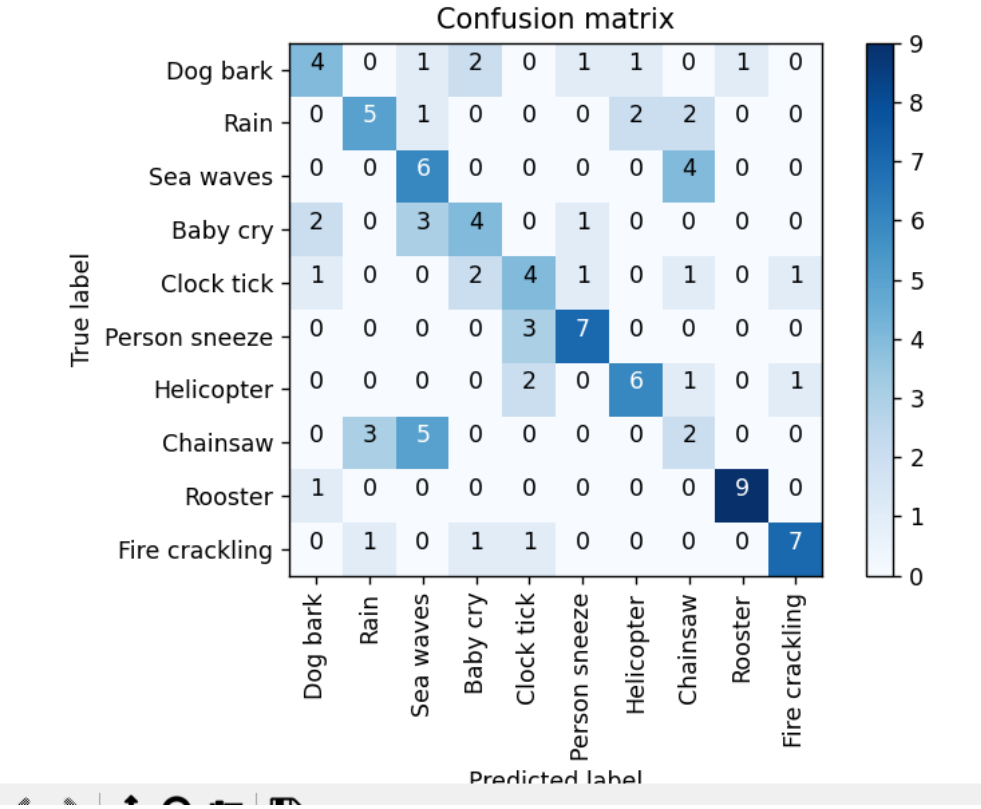
ACC = 44%

這個 model **對於所有 case 的 performance 都相當不好**，他的 Type I Error 相當多，我認為這來自於中位數本身就不是一個好描述樣本的統計特性，尤其是樣本小的時候。

我們也可以嘗試另一種 classifier(ex. Decision Tree)，我們直接呼叫 scikit learn 內建的 DecisionTreeClassifier，並把 mean 當作 feature，結果如下：



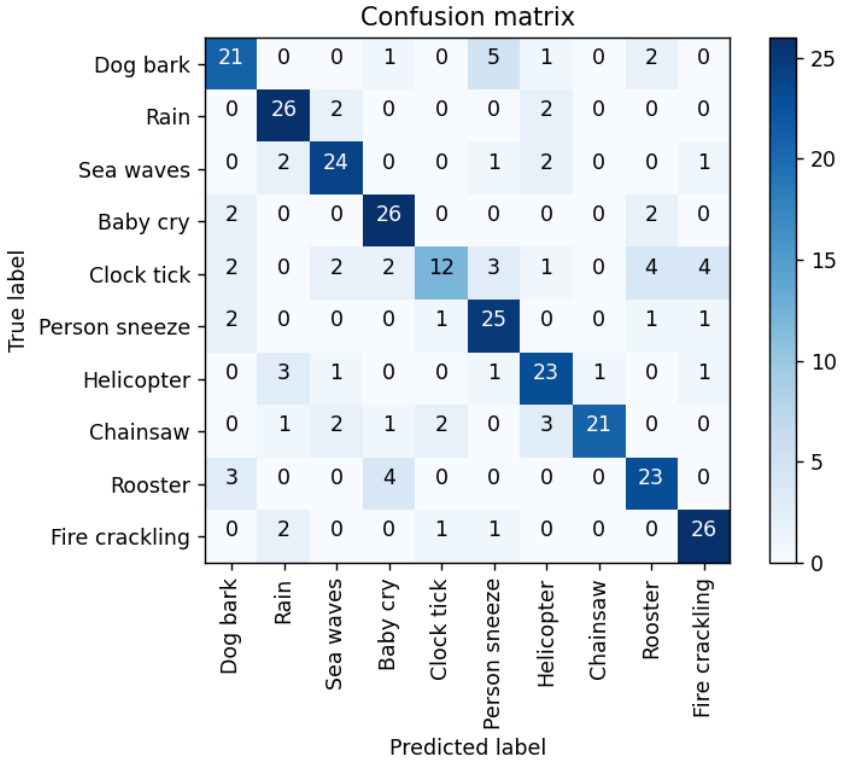
ACC = 51%



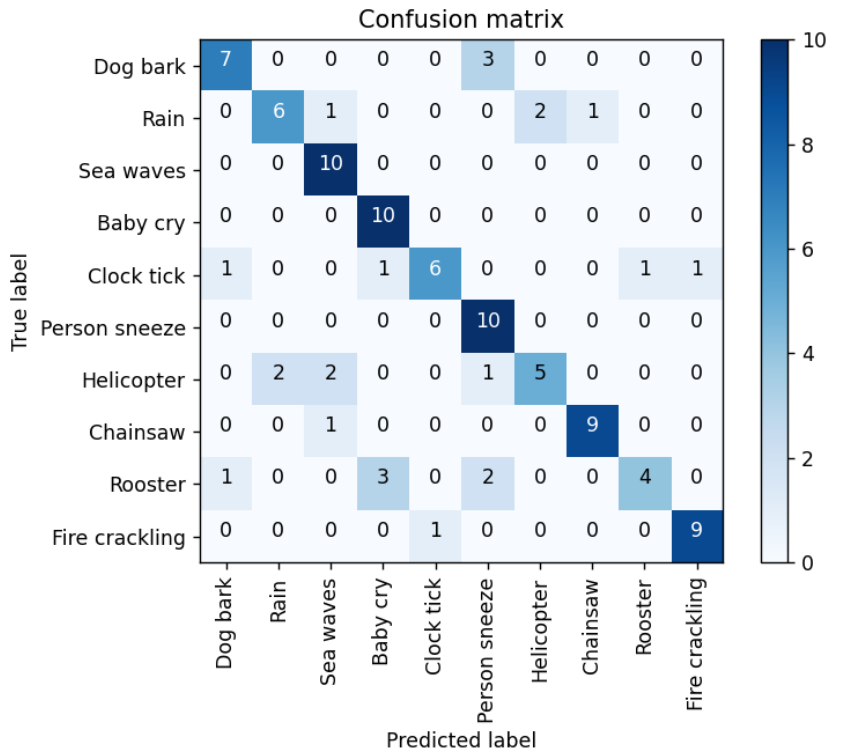
ACC = 54%

這個 model **對於公雞的叫聲有不錯的辨識率**，Type I Error(FP)和 Type II Error(FN)都相當少。但其在時鐘指針、海浪聲音方面則相當差。

3. 最後是我目前做出最好的結果：



ACC = 75.7%



ACC = 76%

兩張分別是在 Validation 和 Testing 中所得到的 confusion matrix，他們分別的 Accuracy 為 75.7%和 76%。而我的作法如下：

1. Number of MFCC 從原先預設的 20 改為 64
2. 取前 9 個 MFCC，對時間軸取平均，得到 9 個 feature
3. 將所有的 MFCC 沿著時間軸找 standard deviation，得到 64 個 feature

4. SVC 中的參數 C 設為 0.75，gamma 採用 auto

這些**參數選取的原因大致上是由 trial and error 以及前幾個 Lab 對 MFCC 特性的理解找出來的**。

◆ Conclusion

這個 Lab 讓我們用前面所學習過的語音特性 MFCC 來當作 feature 去做語音辨識，並嘗試透過設計訓練模型、調整參數以及預處理等步驟來提升 Accuracy。過程相當有趣，只可惜我在 ML 方面的知識尚少(還沒修過教授開的 ML 導論)，做出來的結果我認為也沒有到很好，希望未來再有充足的 domain knowledge，能再次重新挑戰這個問題。

◆ References

- Jason Chen’ s Blog https://jason-chen-1992.weebly.com/
- Tommy Huang’ s Medium https://chih-sheng-huang821.medium.com/
  上面兩個提供Cross Validation和SVM的基本觀念推導
- Scikit-learn https://scikit-learn.org/stable/index.html
  了解Python ML相關function的使用
- 教授與助教的講義