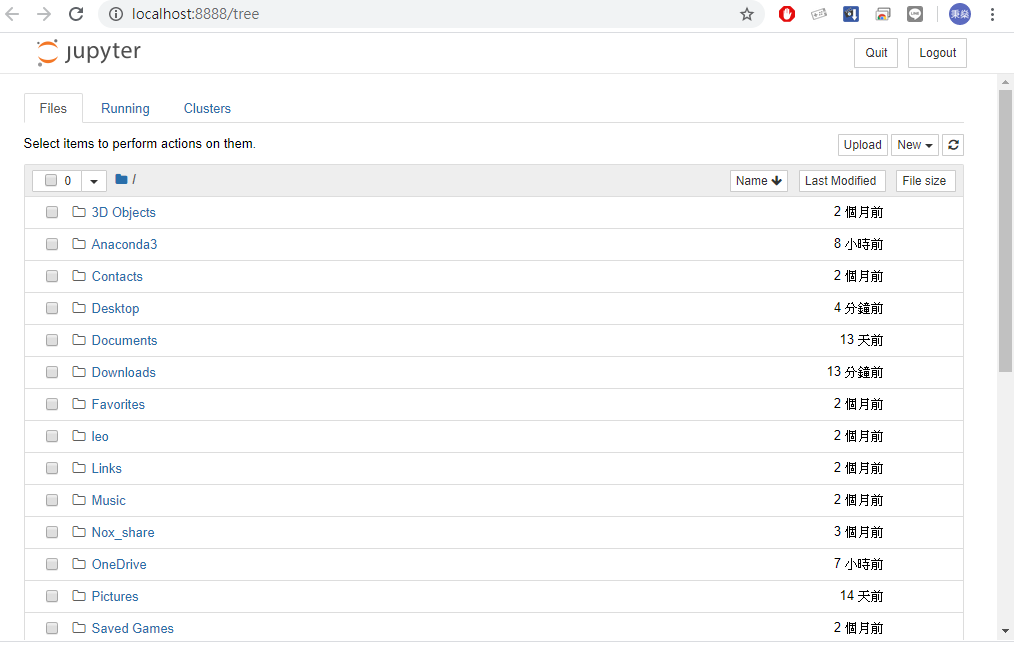
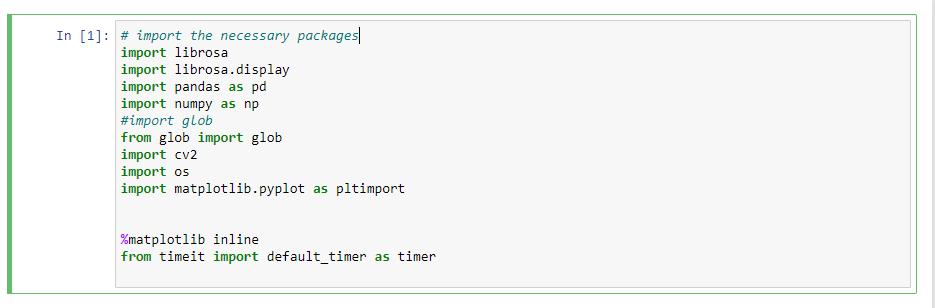
先開啟jupyter notebook,打開MLP.ipynb檔



**步驟一:載入套件**



導入套件,若是沒有安裝這些套件,程式則無法執行,若是缺少套件要自行前往收尋python XXXX 安裝。

**步驟二:建立CSV檔-事前要先把所有資料分成訓練跟測試,約7:3,CSV做成下面圖片務必是隨機排列,因為隨機排列的成果會遠優於排列完整的情形。**

write\_to\_csv-定義函式名

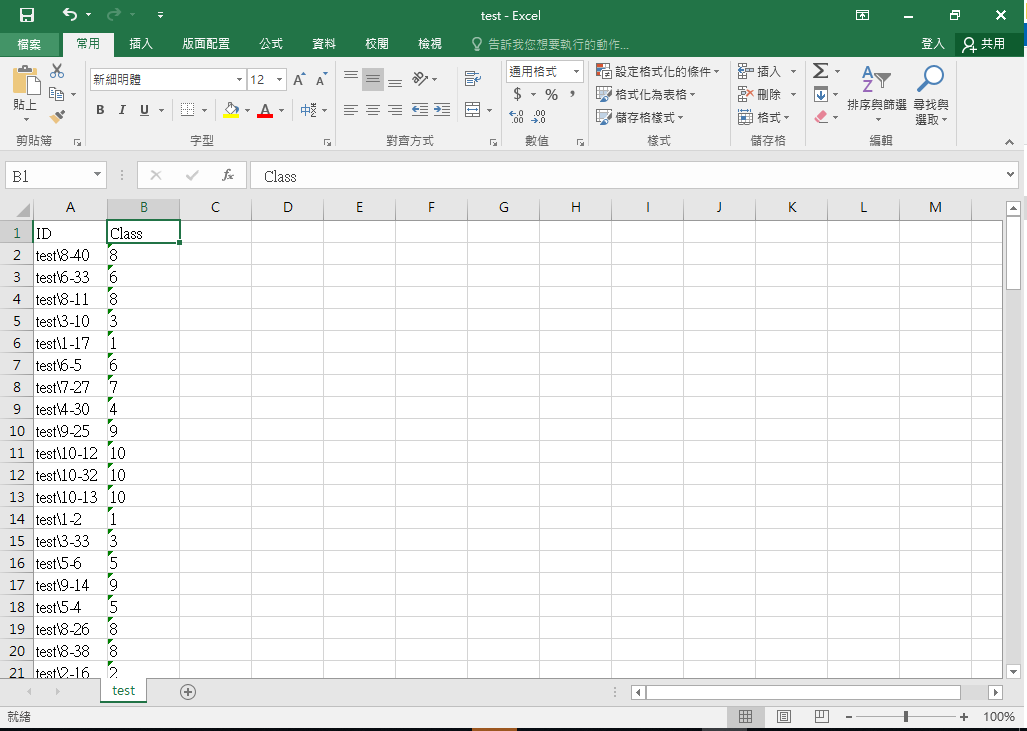
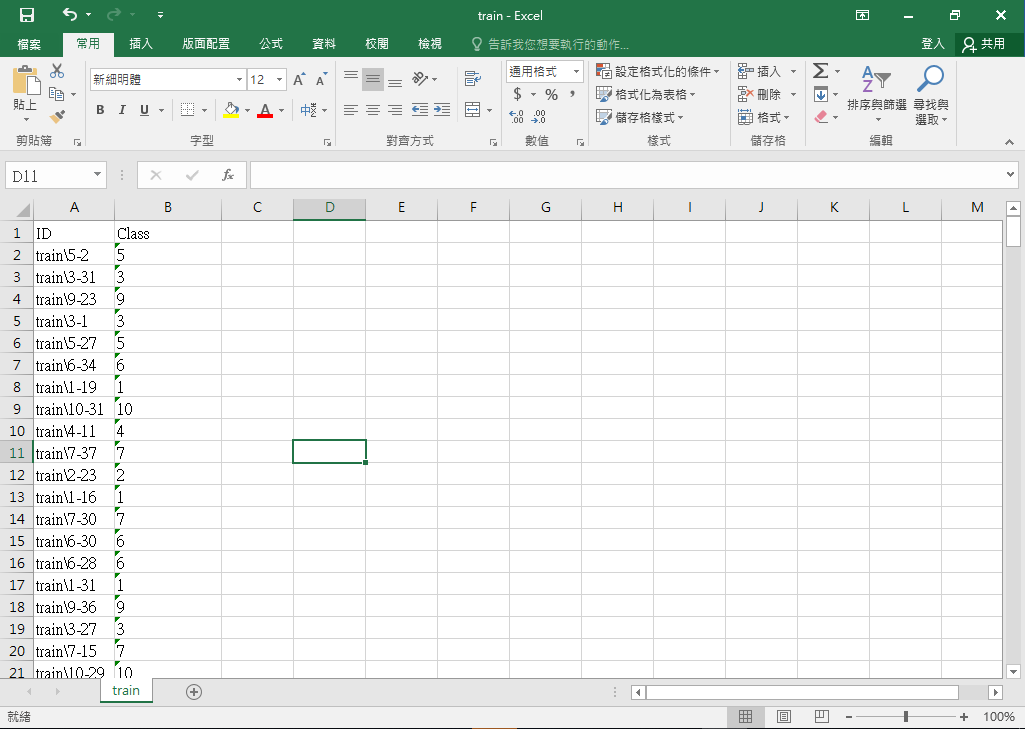
csvName, dataDir-變數名

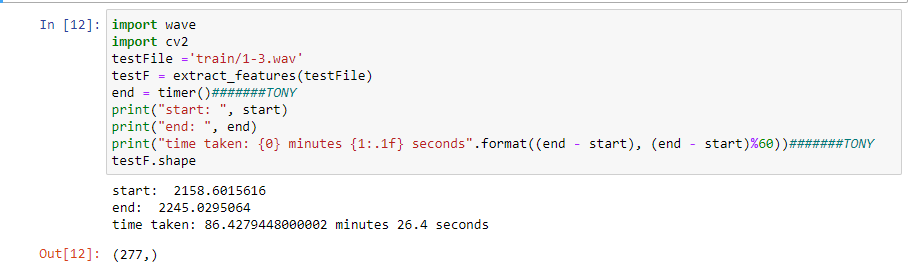
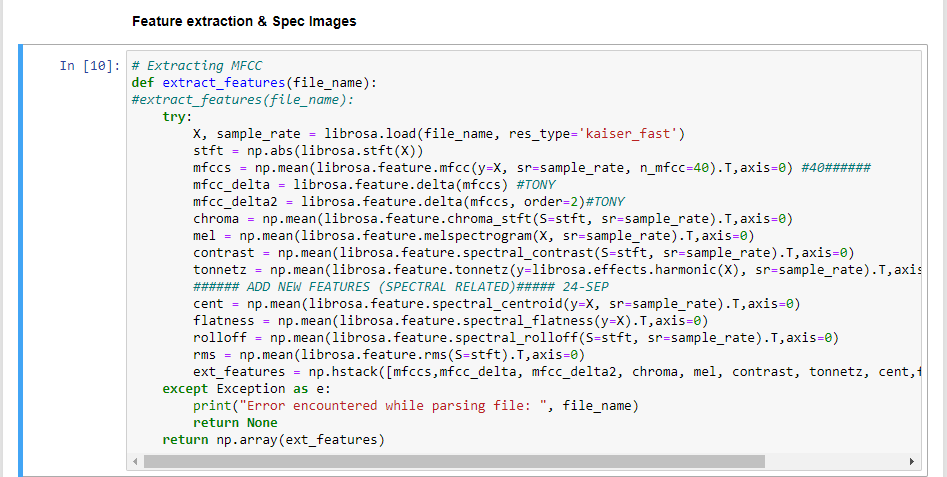
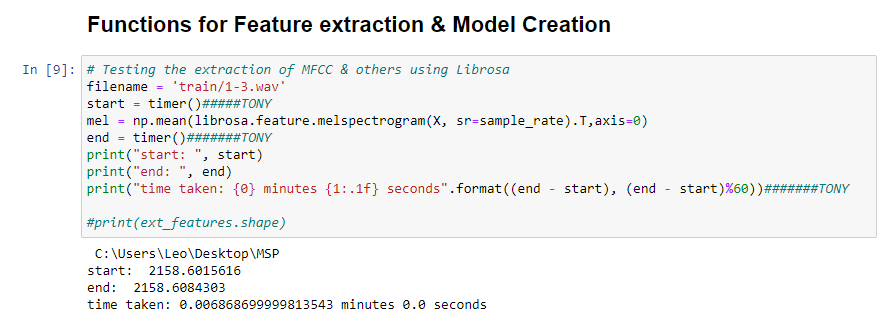
fileID-檔案名稱

classID-類別

split-分隔符號

glob("train/\*")-檔案位置



**步驟三:提取特徵值與創建模型-基本上只有filename要更動**

filename-檔案名稱

librosa.load(filename)-讀取資料

librosa.feature.melspectrogram-計算梅爾頻譜圖

mel-計算梅爾頻譜圖

chroma-根據波形或功率譜圖計算色譜圖。

Contrast-計算光譜對比度

tonnetz- Computes the tonal centroid features

cent-計算頻譜質心

flatness-計算光譜平坦度

rolloff-計算滾降頻率

rms-根據音頻樣本或聲譜圖計算均方根

ext\_features-將所有特徵矩陣堆疊起來

shape-維

步驟四:創建MLP模組-可自行改參數輸出結果會有變化

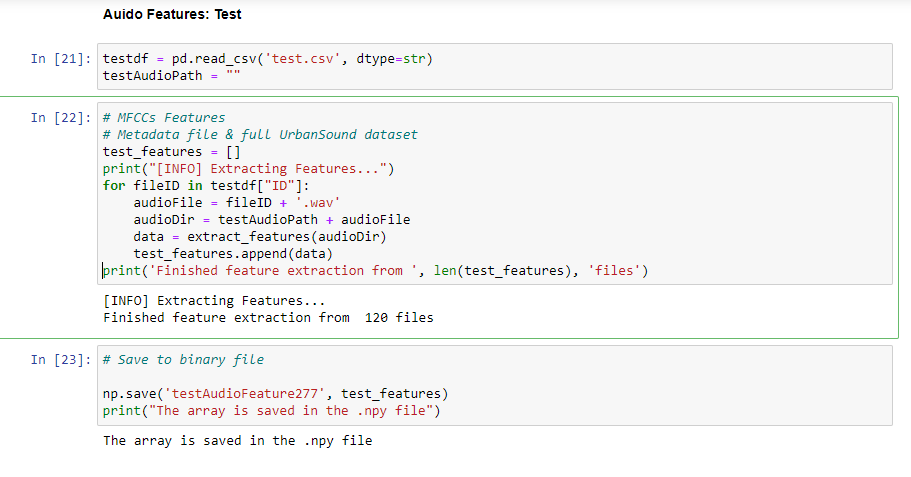
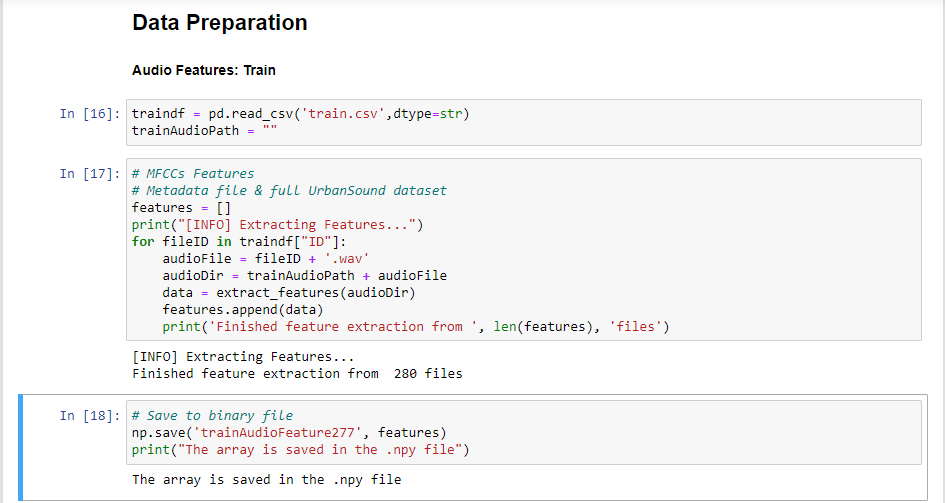


model = Sequential()-建立線性堆疊模型

model.add(Dense(256, input\_shape=(277,)))-256為神經元個數,277為輸入維數

Activation('relu')-定義激活函數為relu

Dropout(0.5)-每次訓練迭代時,隨機在隱藏層中放棄50%神經元

**步驟五:抓取特徵並且儲存成.npy檔**

traindf-讀取訓練CSV檔

testdf-讀取測試CSV檔

trainAudioPath-檔案位置

trainAudioFeature277-訓練特徵儲存之檔名

testAudioFeature277-測試特徵儲存之檔名

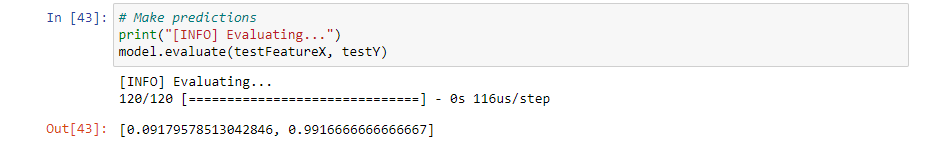
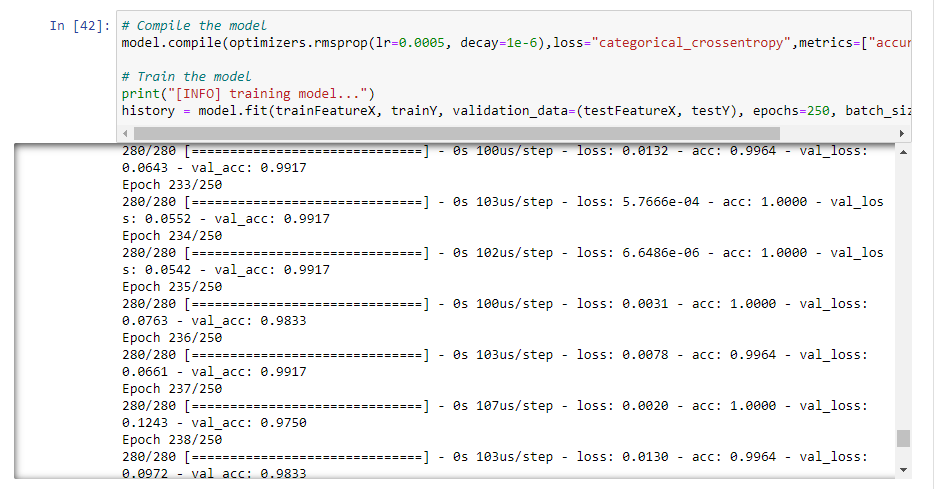
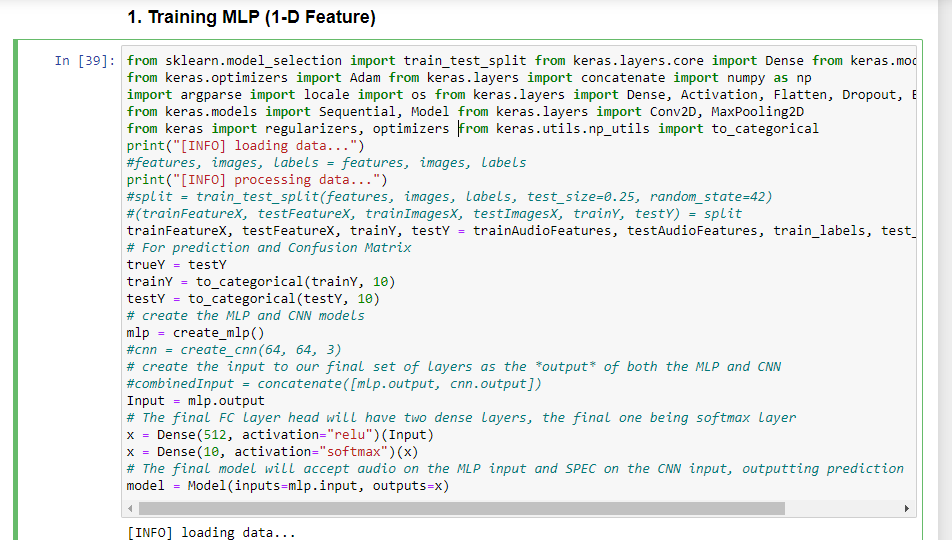
步驟六:讀取特徵與真實值

trainAudioFeatures = np.load('trainAudioFeature277.npy', allow\_pickle=True)-載入先前做好的測試特徵值

train\_labels = traindf["Class"]-訓練的label為,CSV檔案中class那行

test\_labels = testdf["Class"]-測試的label為,CSV檔案中class那行

le.transform-轉換成整數

**步驟七:**MLP訓練測試

trainY = to\_categorical(trainY, 10)-10為有10種可能分別是1-10號語者

testY = to\_categorical(testY, 10) -10為有10種可能分別是1-10號語者

mlp = create\_mlp()-呼叫先前的create\_mlp()

x = Dense(10, activation="softmax")(x)-使用softmax激活函數進行轉換,10為對應到結果有1-10共10位語者

optimizers-設定訓練時的最優化方法

loss-設定損失函數

epochs=250-執行250次訓練週期

batch\_size=32-每一次32筆資料