

資料分析期末報告
資科二B 陳妙音 11173228

主題：對於中文流行樂的資料進行分析

1. 用 selenium 爬取 google 顯示的歌詞，並用 **文字雲** 呈現
2. 用 librosa 偵測匯入的 mp3 檔案，並 **偵測節拍**
3. 用 librosa 產生該 mp3 檔案的 **聲音波形圖**
(2 和 3 都需要先下載音檔，但是無法從正規方法下載 YT 音樂，所以我只能先用螢幕錄影的方式轉換成 mp3，所以只有 4 首歌可以執行偵測節拍和產生波形圖的動作)
4. 把上面 3 個放到 **customtkinter** 上
5. (原本想要利用 librosa 的 mfcc 再加上 **偵測唱歌的歌手是誰**，但是我的資料量太少，訓練模型準確度太低，上不了臺面，但仍然提供我的程式碼，我只成功辨別到韋禮安的聲音而已，程式碼在最後面)

成果：



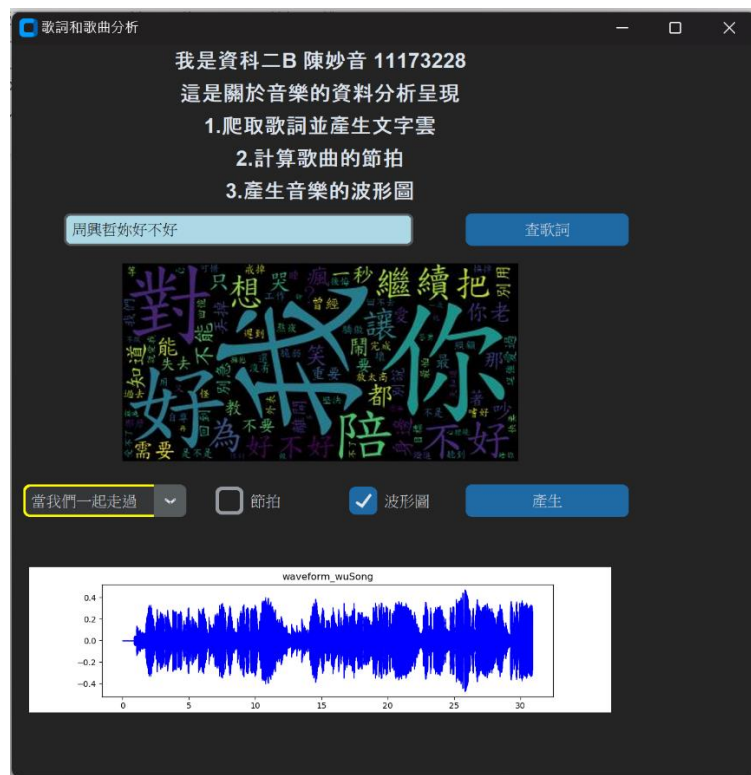
下圖：你好不好的文字雲&如果可以的節拍



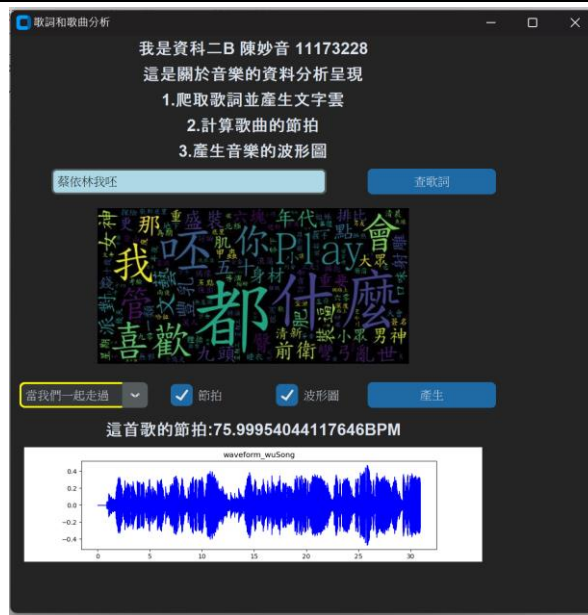
如果可以 - 電影"月老"主題曲 is a **moody** song by WeiBird with a tempo of **75 BPM**. It can also be used **double-time** at **150 BPM**. The track runs 4

根據網路上的資料的確是 151BPM 左右，準確度還不錯

下圖:妳好不不好的文字雲&當我們一起走過的波形圖



下圖:我呸的文字雲&當我們一起走過的節拍+波形圖



SongBPM

<https://songbpm.com> › dang-wo-men-yi-qi-zou-guo

Tempo for 當我們一起走過

當我們一起走過 is a moody song by sodagreen with a tempo of 77 BPM. It can also be used double-time at 154 BPM. The track runs 4 minutes and 45 seconds long ...

下圖:匯入韋禮安唱歌的音檔，成功偵測出韋禮安的聲音

```
#測試模型
def predict(model, audio_file):

    #匯入要偵測是誰唱的音檔
    y, sr = librosa.load(audio_file)
    mfcc = librosa.feature.mfcc(y=y, sr=sr)

    #進行多類別分類預測
    label = model.predict(mfcc.T)
    proba = model.predict_proba(mfcc.T)

    #獲取概率最大的類別標籤(最像的聲音)
    max_prob_idx = np.argmax(proba[0])
    max_prob_label = label[max_prob_idx]

    return max_prob_label

#辨別韋禮安唱歌"如果可以"的音檔
label = predict(model, "C:/Users/miao/wei_test.mp3")

if label==0.0:
    singer='吳青峰'
elif label==1.0:
    singer='韋禮安'
elif label==2.0:
    singer='楊丞琳'
elif label==3.0:
    singer='蔡依林'
elif label==4.0:
    singer='黃偉晉'
elif label==5.0:
    singer='小賴'
elif label==6.0:
    singer='9m88'
elif label==7.0:
    singer='Joyce'
elif label==8.0:
    singer='阿信'
print('這是', singer, '的聲音')
```

這是 韋禮安 的聲音

文字雲

```
Sentence = jieba.cut_for_search(lyrics_text)

hash = {}

for item in Sentence:

    if item in stopwords: #將不要的字詞排除在雲之外
        continue

    if item in hash:
        hash[item] += 1 # hash[item] = hash[item] + 1
    else:
        hash[item] = 1 #沒有放在dict的給值1

wc = WordCloud(font_path="TW-Kai-98.1.ttf", #設置字體
               background_color="black", #背景顏色
               max_words = 1000, #文字雲顯示最大詞數
               stopwords=stopwords) #停用字詞

# 使用dictionary的內容產生文字雲
wc.generate_from_frequencies(hash)

# 視覺化呈現
plt.imshow(wc)
plt.axis("off")
plt.figure(figsize=(50, 50), dpi = 300)

wc.to_file('wordcloud.jpg')

#放在ctk
wordIMG = ctk.CTkImage(light_image=Image.open('wordcloud.jpg'),
                       dark_image=Image.open('wordcloud.jpg'),
                       size=(340,170))
wordLabel.configure(image=wordIMG)
```

Librosa 偵測節拍和波形圖

```

import librosa
import matplotlib.pyplot as plt
import librosa.display
import matplotlib.backends.backend_tkagg as tkagg

def create_bpm_waveform():
    global canvas_widget

    #選擇哪一首歌
    song_combo=combo.get()
    if song_combo=='如果可以':
        audio_path="C:/Users/miao2/wei_test2.mp3"
        name='weiSong1'
    elif song_combo=='當我們一起走過':
        audio_path="C:/Users/miao2/wu_test.mp3"
        name='wuSong'
    elif song_combo=='第一個想到你':
        audio_path="C:/Users/miao2/wei_test3.mp3"
        name='weiSong2'
    else:
        audio_path="C:/Users/miao2/yang_test.mp3"
        name='yangSong'

    y, sr = librosa.load(audio_path)

    #節拍顯示
    if check_var_bpm.get() == 'on':
        #節拍偵測
        tempo, beat_frames = librosa.beat.beat_track(y=y, sr=sr)
        label7.configure(text='這首歌的節拍:'+str(tempo)+'BPM')
    else:
        label7.configure(text='')

    #波形圖顯示
    if check_var_wave.get() == 'on1':

        #繪製波形圖
        plt.figure(figsize=(10, 2.5))
        librosa.display.waveshow(y, sr=sr, color="blue")
        plt.title('waveform_'+name)

    # 將Matplotlib圖形嵌入Tkinter
    canvas = tkagg.FigureCanvasTkAgg(plt.gcf(), master=bg)
    canvas_widget = canvas.get_tk_widget()
    canvas_widget.grid(column=0, row=9, columnspan=4, pady=10, padx=20)

    else:
        canvas_widget.destroy()

plt.close()

```

customtkinter 介面

```

import tkinter
import customtkinter as ctk
from tkinter import ttk
from PIL import Image
import os

#-----customtkinter介面-----
ctk.set_appearance_mode("dark") # 設定背景顏色
ctk.set_default_color_theme("blue") # 設定物件色系
bg = ctk.CTk() # 設置畫布
bg.geometry("640x630") # 決定視窗大小
bg.title("歌詞和歌曲分析") # 給定視窗標題名稱

label1 = ctk.CTkLabel(bg, text = "我是資科二B 陳妙音 11173228",font=('微軟雅黑',18,'bold')).grid(column=0, row=0, columnspan=4)
label2 = ctk.CTkLabel(bg, text = "這是關於音樂的資料分析呈現",font=('微軟雅黑',18,'bold')).grid(column=0, row=1, columnspan=4)
label3 = ctk.CTkLabel(bg, text = "1.爬取歌詞並產生文字雲",font=('微軟雅黑',18,'bold')).grid(column=0, row=2, columnspan=4)
label4 = ctk.CTkLabel(bg, text = "2.計算歌曲的節拍",font=('微軟雅黑',18,'bold')).grid(column=0, row=3, columnspan=4)
label5 = ctk.CTkLabel(bg, text = "3.產生音樂的波形圖",font=('微軟雅黑',18,'bold')).grid(column=0, row=4, columnspan=4)

song_input = ctk.CTkEntry(bg, placeholder_text='Singer and name of song',
                        width=300,
                        text_color='black',
                        placeholder_text_color='grey',
                        fg_color=('blue','lightblue')
                        )
song_input.grid(column=0, row=5, columnspan=3, pady=5)

lyricsBT = ctk.CTkButton(bg, text='查歌詞', command=creat_wordcloud)
lyricsBT.grid(column=3, row=5)

# wordIMG = ctk.CTkImage(light_image=Image.open('wordCloud.jpg'),
#                        dark_image=Image.open('wordCloud.jpg'),
#                        size=(340,170))
wordLabel = ctk.CTkLabel(bg, text='', image=None)
wordLabel.grid(column=0, row=6, columnspan=4, pady=10)

songs=['如果可以','當我們一起走過','第一個想到你','青春住了誰']
combo = ctk.CTkComboBox(bg, values=songs,
                        border_width=2,
                        border_color='yellow'
                        )

```

```

combo.grid(column=0, row=7,pady=10,padx=10)

check_var_bpm = ctk.StringVar(value = "off")
bpm_check = ctk.CTkCheckBox(bg, text="節拍", variable=check_var_bpm, onvalue='on', offvalue='off')
bpm_check.grid(column=1, row=7,pady=10,padx=15)

check_var_wave = ctk.StringVar(value = "off1")
wave_check = ctk.CTkCheckBox(bg, text='波形圖', variable=check_var_wave, onvalue='on1', offvalue='off1')
wave_check.grid(column=2, row=7,pady=10)

enterBT = ctk.CTkButton(bg, text='產生', command=create_bpm_waveform)
enterBT.grid(column=3, row=7)

label7 = ctk.CTkLabel(bg, text = '',font=('微軟雅黑',18,'bold'))
label7.grid(column=0, row=8, columns=4)

# waveIMG = ctk.CTkImage(light_image=Image.open('waveform.jpg'),
#                         dark_image=Image.open('waveform.jpg'),
#                         size=(600,150))
#
waveLabel = ctk.CTkLabel(bg, text='', image=None)
waveLabel.grid(column=0, row=9, columns=4, pady=10, padx=20)
|
bg.mainloop()

```

訓練模型，用 librosa 的 mfcc 偵測是誰的音色

```

#聲音辨識
import librosa
import numpy as np
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
import IPython.display as ipd
#from sklearn.svm import SVC

#匯入聲音數據
def load_data():
    #以下為藝人在直播中說話的音檔(train)
    wu, sr1 = librosa.load("C:/Users/miao2/wu_train.mp3")          #吳青峰wu
    wei, sr2 = librosa.load("C:/Users/miao2/wei_train.mp3")       #韋禮安wei
    yang, sr3 = librosa.load("C:/Users/miao2/yang_train.mp3")     #楊丞琳yang
    jolin, sr4 = librosa.load("C:/Users/miao2/jolin_train.mp3")   #蔡依林jolin
    huang, sr5 = librosa.load("C:/Users/miao2/huang_train.mp3")   #黃偉進huang
    lai, sr6 = librosa.load("C:/Users/miao2/lai_train.mp3")       #小賴lai
    m88, sr7 = librosa.load("C:/Users/miao2/m88_train.mp3")       #m88
    joyce, sr8 = librosa.load("C:/Users/miao2/joyce_train.mp3")   #joyce
    shin, sr9 = librosa.load("C:/Users/miao2/shin_train.mp3")     #阿信

#提取MFCC特徵( 辨別音色)
wu_mfcc = librosa.feature.mfcc(y=wu, sr=sr1)
wei_mfcc = librosa.feature.mfcc(y=wei, sr=sr2)
yang_mfcc = librosa.feature.mfcc(y=yang, sr=sr3)
jolin_mfcc = librosa.feature.mfcc(y=jolin, sr=sr4)
huang_mfcc = librosa.feature.mfcc(y=huang, sr=sr5)
lai_mfcc = librosa.feature.mfcc(y=lai, sr=sr6)
m88_mfcc = librosa.feature.mfcc(y=m88, sr=sr7)
joyce_mfcc = librosa.feature.mfcc(y=joyce, sr=sr8)
shin_mfcc = librosa.feature.mfcc(y=shin, sr=sr9)

#將九個人的mfcc合併成一個數據集
X = np.concatenate((wu_mfcc.T, wei_mfcc.T, yang_mfcc.T,
                    jolin_mfcc.T, huang_mfcc.T, lai_mfcc.T,
                    m88_mfcc.T, joyce_mfcc.T, shin_mfcc.T), axis=0)

#y軸向量
y = np.concatenate((np.zeros(len(wu_mfcc.T)), np.ones(len(wei_mfcc.T)), 2*np.ones(len(yang_mfcc.T)),
                    3*np.ones(len(jolin_mfcc.T)), 4*np.ones(len(huang_mfcc.T)), 5*np.ones(len(lai_mfcc.T)),
                    6*np.ones(len(m88_mfcc.T)), 7*np.ones(len(joyce_mfcc.T)), 8*np.ones(len(shin_mfcc.T))))

return X, y

X, y = load_data()

#訓練模型
def train(X, y):

    #將數據集分成訓練&測試
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

    #使用邏輯回歸算法進行多類別分類
    model = LogisticRegression(multi_class='ovr',max_iter=1000)

    #訓練模型
    model.fit(X_train, y_train)

    return model

model = train(X,y)

```

```

#測試模型
def predict(model, audio_file):

    #匯入要偵測是誰唱的音檔
    y, sr = librosa.load(audio_file)
    mfcc = librosa.feature.mfcc(y=y, sr=sr)

    #進行多類別分類預測
    label = model.predict(mfcc.T)
    proba = model.predict_proba(mfcc.T)

    #獲取概率最大的類別標籤(最像的聲音)
    max_prob_idx = np.argmax(proba[0])
    max_prob_label = label[max_prob_idx]

    return max_prob_label

#辨別韋禮安唱歌"如果可以"的音檔
label = predict(model, "C:/Users/miao2/wei_test.mp3")

if label==0.0:
    singer='吳青峰'
elif label==1.0:
    singer='韋禮安'
elif label==2.0:
    singer='楊丞琳'
elif label==3.0:
    singer='蔡依林'
elif label==4.0:
    singer='黃偉晉'
elif label==5.0:
    singer='小賴'
elif label==6.0:
    singer='9m88'
elif label==7.0:
    singer='Joyce'
elif label==8.0:
    singer='阿信'
print('這是', singer, '的聲音')

```

這是 韋禮安 的聲音

韋禮安是正確的，但匯入其他藝人唱歌的音檔，就會偵測成錯誤的人
 所以我又試了 SVM 模型，但一樣準確度很低

```

from sklearn.svm import SVC

def svm_model(X,y):
    # 假設X_train是音樂特徵，y_train是對應的歌手標籤
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # 使用SVM模型
    svm_model = SVC(kernel='linear', C=1, probability=True)
    svm_model.fit(X_train, y_train)

    # 預測
    y_pred = svm_model.predict(X_test)

    # 計算準確率
    accuracy = accuracy_score(y_test, y_pred)
    print(f'Accuracy: {accuracy}')

    return svm_model

svm_test = svm_model(X,y)

```

```

#測試模型
def predict(svm_model, audio_file):
    #匯入要偵測是誰唱的音檔
    y, sr = librosa.load(audio_file)
    mfcc = librosa.feature.mfcc(y=y, sr=sr)

    # 使用訓練好的 SVM 模型進行預測
    label = svm_model.predict(mfcc.T)
    proba = svm_model.predict_proba(mfcc.T)

    #獲取概率最大的類別標籤(最像的聲音)
    max_prob_idx = np.argmax(proba[0])
    max_prob_label = label[max_prob_idx]

    return max_prob_label

label = predict(svm_test, "C:/Users/miao2/wei_test2.mp3")

if label==0.0:
    singer='吳青峰'
elif label==1.0:
    singer='韋禮安'
elif label==2.0:
    singer='楊丞琳'
elif label==3.0:
    singer='蔡依林'
elif label==4.0:
    singer='黃偉晉'
elif label==5.0:
    singer='小賴'
elif label==6.0:
    singer='9m88'
elif label==7.0:
    singer='Joyce'
elif label==8.0:
    singer='阿信'
print('這是', singer, '的聲音')

```

這是 韋禮安 的聲音

用其他首韋禮安唱歌的音檔，也成功得到是韋禮安的音色，但若改成其他歌手就會回答不正確