第六組_DM_final

組員:R07725049吳姿君、R08725008周若涓、R08725010陳亦珊、R08725030徐薇尹

## 1.0 研究動機與目的

觀察熱門的歌曲瞭解整體歌曲曲風的分類，再進一步透過分析各地區音樂排行榜，來瞭解並探討不同地區的人所喜歡的音樂類型差異，以及音樂曲風為何，並實作根據使用者歌單資料來分析該使用者的喜好較接近哪個地區，形成一個以歌單來識別使用者與哪個城市的歌單排行榜曲風相近的測驗遊戲。

## 2.0 資料描述

收集的資料分為 General data(2038筆) 與 Musical Cities data(3400筆)，以上兩張 table 收集的 attributes 包含 track_id, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo；除此之外，Musical Cities data 除了上述 attributes, 還會有 city label。

其中 General data 為了要讓它有 general 的特性，選出有超過 100 萬followers 的歌單，很多人聽代表是屬於 general 的資料，例如: Global Viral 50, Global Top 50, Today's Top Hits, Confidence Boost, Happy Beats, etc.

Musical Cities data 的部分，從 Musical map of the world 找出 city，以及每個 city 的 top list，Musical Cities data 共收集了 3400 筆，其中包含 33 個 cities，例如: Taipei, Tokyo, London, Wellington, Berlin, Madrid, Amsterdam 等。

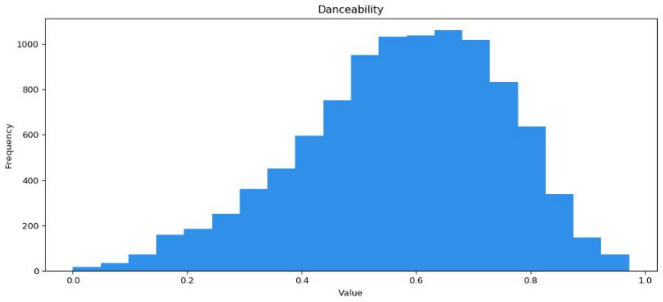
收集資料的方式就是複製各個 city 的 playlist_id(e.g. 2WxMWXBdooAvIG0w4LfrS8)，每個playlist 都有 100首歌，透過 Spotify 的 Get a Playlist API，得到 playlist 的詳細資料，如playlist 的 owner, followers, track。

利用 JSON Editor，萃取 track 的 100 個items，得到track 相關資料的table，需要的只有 track.id 欄位，再將各個國家的 playlist 的 track.id 欄位, 丟進 Spotify 的 Get Audio Features for Several Tracks API (這個 API 有一個限制，每次只能丟入 100 個 track.id)，得到音訊的 feature attributes，如 danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo 等，透過 API 得到 feature attribute 的 JSON file，再透過 tool 將 JSON file 轉成 csv file.

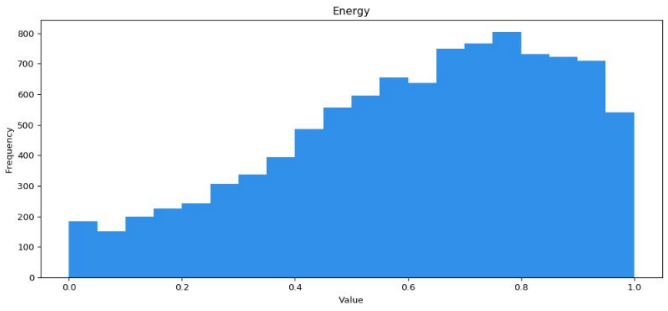
Attributes of Data 詳細內容如下:

- track_id: Spotify URI 的末尾有 base-62 的 identifier, 透過不同的 identifier 可以找到不同的藝術家、曲目、專輯、播放列表等資料類型。

- danceability: 舞蹈性是根據節奏、節奏穩定性、拍子強度和整體規律性等音樂元素的組合, 描述歌曲適合跳舞的方式。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:

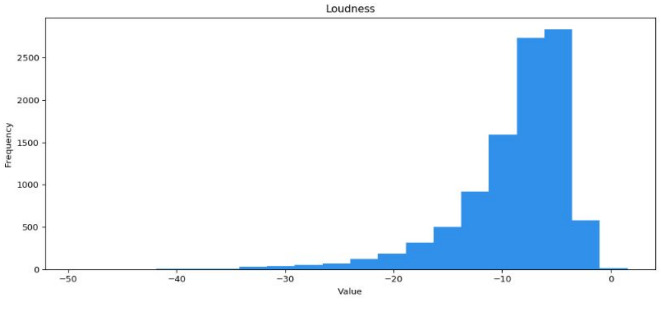



- energy: 能量代表強度和活動的量度, 充滿活力的曲目通常會感覺快速, 響亮且嘈雜。
    - 例如: 金屬樂具有較高的能量, 而巴哈前奏樂曲的得分則較低。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:

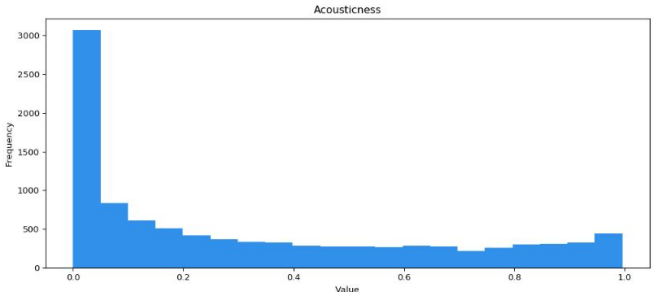



- key: 整體估計歌曲的調性, 資料型態為整數, 例如: 0 = C, 1 = C♯/ D♭, 2 = D, 依此類推。

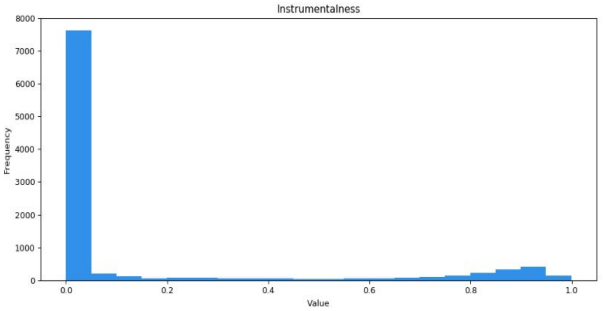
- loudness: 歌曲的的整體響度平均值, 以分貝（dB）為單位, 響度是聲音的質量, 與振幅有主要關聯。
    - 資料型態為 float, 值的通常介於 -60 至 0 db。
    - 值的分布如圖:



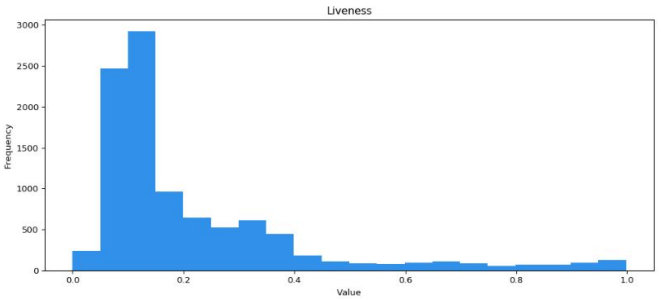
- mode: 模式為歌曲的形式, 可以知道其旋律內容的音階類型。
    - 主要以 1 或 0 表示。
- speechiness: 語音可以檢測歌曲目中是否存在人說話的聲音。
    - 說話成分越多（例如脫口秀、有聲讀物、詩歌）, 屬性值就越接近1.0。
    - 大於 0.66 的值就可以知道歌曲可能完全由口語組成。
    - 介於 0.33 到 0.66 之間的值表示可能同時包含音樂和語音的曲目, 無論是否為分段(包括 rap)。
    - 低於 0.33 的值最有可能代表音樂和其他非語音類曲目。
- acousticness: 表示歌曲有多少出色的聲音(原聲吉他和手鼓), 或是電子聲有多少(即合成器和鼓機), 當出色的聲音越多, 值越高, 近年流行的acousticness 都呈現低的狀態。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:



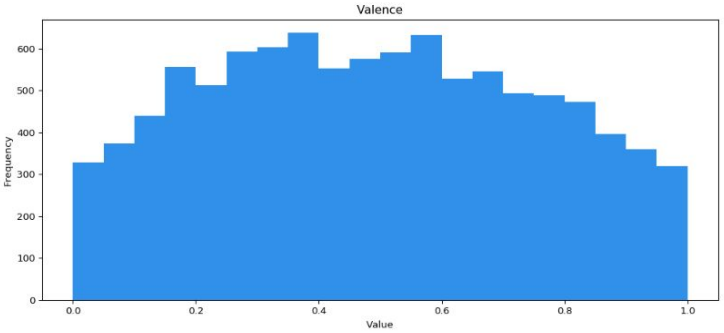

- instrumentalness: 歌曲不包含人聲、單純只有樂器的程度。
    - 在這種情況下, “Ooh” and “aah” 的聲音被視為樂器。
    - Rap 或說話的聲音顯然是“人聲”。
    - 樂器性值越接近 1.0, 則曲目中沒有人聲內容的可能性越大。
    - 高於 0.5 的值表示為樂器音樂。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:



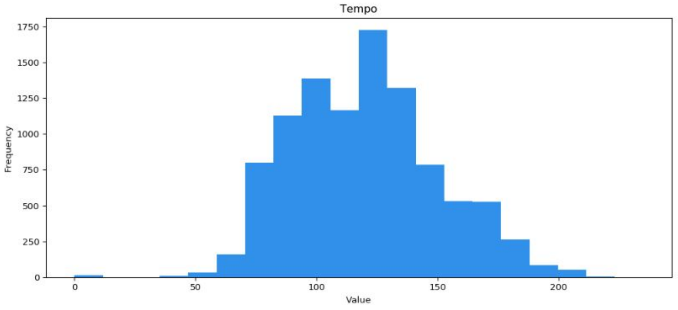
- liveness: 檢測歌曲中是否有觀眾的聲音。
    - 較高的活躍度值表示增加了現場歌曲的可能性, 大於0.8的值很可能會顯示該軌道處於活動狀態。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:




- valence: 描述歌曲傳達的音樂積極性。
    - 值高的歌曲聽起來更積極(例如快樂、開朗、欣喜), 而值低的歌曲聽起來更消極(例如悲傷、沮喪、憤怒)。
    - 資料型態為 float, 值介於 0 到 1 之間。
    - 值的分布如圖:




- tempo: 曲目的總體估計速度, 以每分鐘心跳數（BPM）為單位。
    - 用音樂術語來說, 節奏是指給定樂曲的速度或節奏, 它直接來自平均拍子持續時間。
    - 資料型態為 float。
    - 值的分布如圖:



## 3.0 實驗方法

(1) 曲風分類模型

(a) 概念

我們使用在全球排行榜的播放清單中的7,437筆歌曲資料作為訓練資料集。由於每筆資料都沒有曲風的ground truth label，所以我們決定使用Clustering分法來找出特徵向量比較相似的歌曲，並且以1000 iterations後的Clusters來作為不同曲風的劃分分界。而這些Clusters的centroids就會被記錄下來，之後新的input sample要分類時，就比較該sample在維度空間與每個centroids的尤氏距離，將sample判給離他最近的那群，將那群的label當作sample的曲風類型label。

(b) 資料預處理

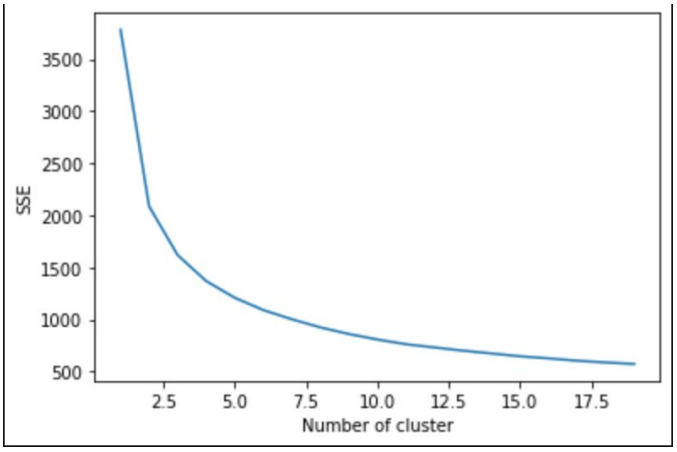
每筆資料有11個特徵，皆為數值資料(numerical data)，但是型態有些不同，如tempo音樂節拍這個特徵會介於60~200之間，但是acousticness這個特徵會介於0~1之間。所以我們對於資料，依照他們的最大最小值，做了標準化(normalization)的處理，使每個特徵都落在0~1的區間內。

(c) 資料降維

為了讓Clustering可以有比較好的結果，所以我們對資料做了降維，捨棄掉比較沒有影響力的特徵。我們使用PCA來做降維，從11個特徵中保留了5個特徵。這五個特徵的重要程度依序為：mode、acousticness、key、valence、liveness。

(d) 訓練階段：Clustering

我們使用K-Means來實現Clustering。而K(群個數)的選定，是由SSE的變化程度來選擇的。



如上圖可以看到在k=10時，SSE開始進行收斂，且群個數不會太多、將sample分得太細碎，所以我們認為k=10是一個合理的選擇，並且將k設為10。Clustering分出來這10群分別有[832, 706, 793, 672, 458, 824, 749, 1079, 605, 719]個sample。

(e) 應用

要對於一個新的sample分類曲風時，會先對該sample做標準化的處理，所有特徵數值轉為[0,1]區間，再來做PCA降維保留上述五個特徵，再分別與10個centroids的特徵向量比較尤氏距離，最後將曲風類別判給最近的那個群，曲風label即為該群編號。

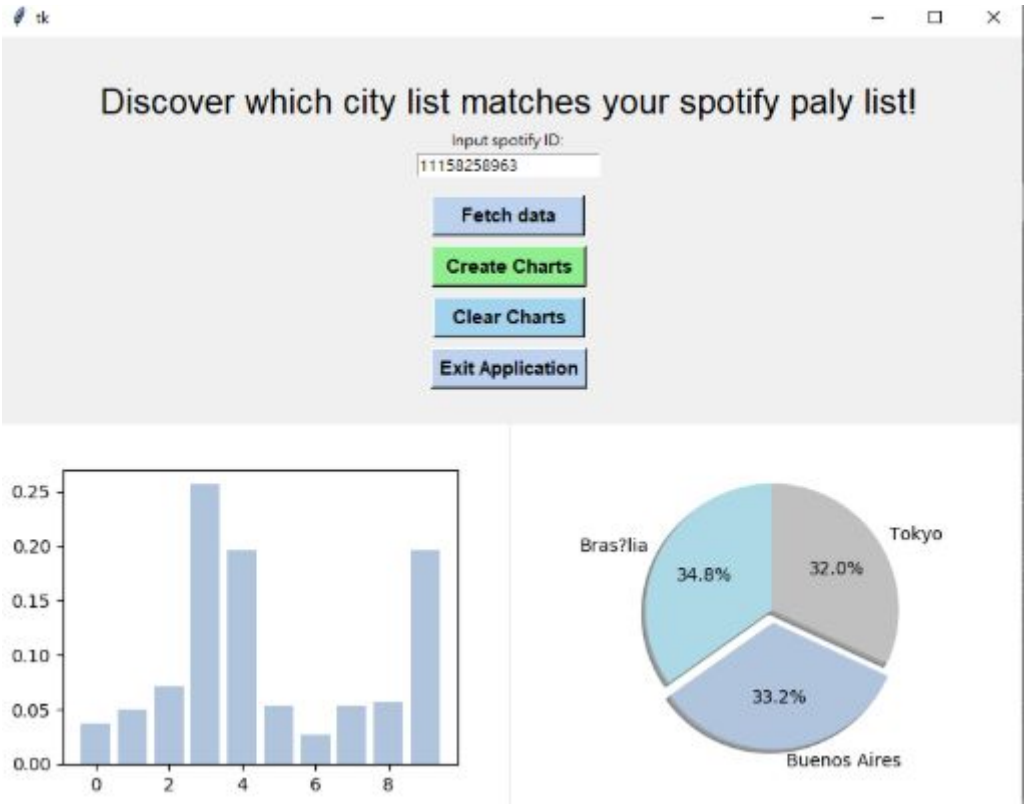
(2) 計算各城市 Top List 分群向量

(a) 將前面搜集的 Musical Cities data 共計 3400 筆、34個國家的歌曲透過 clustering model 進行歌曲分群，得到各群的結果為 [427, 363, 324, 236, 179, 356, 414, 544, 284, 272]。

(b) 將各城市的結果分別加總，並分別計算各分群結果所佔比例，整理成表格如下：

| City | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Taipei | 0.06 | 0.07 | 0.09 | 0.12 | 0.14 | 0.11 | 0.09 | 0.1 | 0.1 | 0.12 |
| Taichung | 0.05 | 0.06 | 0.13 | 0.17 | 0.1 | 0.1 | 0.04 | 0.1 | 0.06 | 0.19 |
| Tainan | 0.02 | 0.01 | 0.1 | 0.39 | 0.07 | 0.08 | 0.02 | 0.02 | 0.05 | 0.24 |
| Tokyo | 0.2 | 0.16 | 0.11 | 0 | 0.01 | 0.09 | 0.09 | 0.2 | 0.12 | 0.02 |
| Osaka | 0.17 | 0.07 | 0.15 | 0.07 | 0.03 | 0.09 | 0.06 | 0.24 | 0.04 | 0.08 |
| Manila | 0.08 | 0.07 | 0.13 | 0.14 | 0.07 | 0.11 | 0.04 | 0.11 | 0 | 0.25 |
| Kuala Lumpur | 0.04 | 0.13 | 0.12 | 0.05 | 0.17 | 0.08 | 0.03 | 0.07 | 0.25 | 0.06 |
| Jakarta | 0.15 | 0.04 | 0.1 | 0.12 | 0.02 | 0.14 | 0.05 | 0.16 | 0.05 | 0.17 |
| Canberra | 0.09 | 0.06 | 0.18 | 0.14 | 0.03 | 0.14 | 0.04 | 0.11 | 0.1 | 0.11 |
| Wellington | 0.14 | 0.04 | 0.12 | 0.04 | 0.05 | 0.12 | 0.18 | 0.19 | 0.04 | 0.08 |
| London | 0.11 | 0.17 | 0.09 | 0 | 0.05 | 0.13 | 0.13 | 0.2 | 0.1 | 0.02 |
| Madrid | 0.21 | 0.08 | 0.11 | 0.06 | 0.07 | 0.10 | 0.13 | 0.10 | 0.08 | 0.08 |
| Berlin | 0.13 | 0.07 | 0.12 | 0.01 | 0.07 | 0.13 | 0.11 | 0.16 | 0.1 | 0.1 |
| Paris | 0.15 | 0.19 | 0.08 | 0 | 0.04 | 0.07 | 0.24 | 0.15 | 0.04 | 0.04 |
| New York | 0.18 | 0.12 | 0.05 | 0.03 | 0.02 | 0.16 | 0.22 | 0.11 | 0.11 | 0 |
| Chicago | 0.08 | 0.09 | 0.09 | 0.02 | 0.01 | 0.19 | 0.16 | 0.15 | 0.18 | 0.03 |
| Ottawa | 0.12 | 0.14 | 0.05 | 0.04 | 0.08 | 0.12 | 0.12 | 0.17 | 0.11 | 0.05 |
| Bras?lia | 0.22 | 0.12 | 0.03 | 0.01 | 0.02 | 0.06 | 0.11 | 0.33 | 0.06 | 0.04 |
| Buenos Aires | 0.14 | 0.17 | 0.02 | 0.01 | 0.03 | 0.04 | 0.18 | 0.3 | 0.05 | 0.06 |
| Berkeley California | 0.06 | 0.09 | 0.09 | 0.19 | 0.09 | 0.13 | 0.05 | 0.12 | 0.04 | 0.14 |
| Rome | 0.13 | 0.13 | 0.11 | 0.05 | 0.11 | 0.14 | 0.11 | 0.11 | 0.07 | 0.04 |
| Milan | 0.12 | 0.16 | 0.07 | 0.01 | 0 | 0.11 | 0.17 | 0.16 | 0.19 | 0.01 |
| Amsterdam | 0.05 | 0.19 | 0.11 | 0.03 | 0.09 | 0.07 | 0.2 | 0.14 | 0.08 | 0.04 |
| Cambridge | 0.17 | 0.12 | 0.1 | 0.07 | 0.01 | 0.1 | 0.13 | 0.15 | 0.07 | 0.08 |
| Birmingham | 0.12 | 0.09 | 0.08 | 0.01 | 0.02 | 0.09 | 0.2 | 0.1 | 0.26 | 0.03 |
| Atlanta Georgia | 0.14 | 0.09 | 0.1 | 0.04 | 0.02 | 0.14 | 0.14 | 0.13 | 0.16 | 0.04 |
| Las Vegas Nevada | 0.16 | 0.11 | 0.08 | 0.06 | 0.02 | 0.07 | 0.13 | 0.2 | 0.07 | 0.1 |
| Boulder Colorado | 0.06 | 0.06 | 0.19 | 0.16 | 0.07 | 0.16 | 0.04 | 0.09 | 0.04 | 0.13 |
| Calgary | 0.14 | 0.08 | 0.12 | 0.01 | 0.02 | 0.12 | 0.13 | 0.33 | 0.03 | 0.02 |
| Le?n | 0.12 | 0.09 | 0.08 | 0.14 | 0.05 | 0.11 | 0.07 | 0.14 | 0.06 | 0.14 |
| Cali | 0.13 | 0.22 | 0.02 | 0.07 | 0.06 | 0.07 | 0.23 | 0.14 | 0.02 | 0.04 |
| Rio Branco | 0.07 | 0.15 | 0.11 | 0.04 | 0.03 | 0.08 | 0.18 | 0.22 | 0.03 | 0.09 |
| Santiago | 0.25 | 0.11 | 0.01 | 0 | 0.05 | 0.02 | 0.2 | 0.35 | 0.01 | 0 |

(3) 使用者互動應用介面 - 判斷spotify user的曲風喜好和哪些城市相似

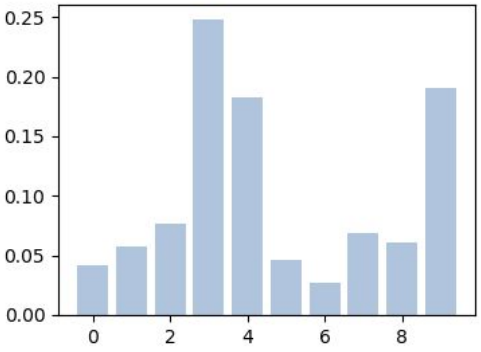


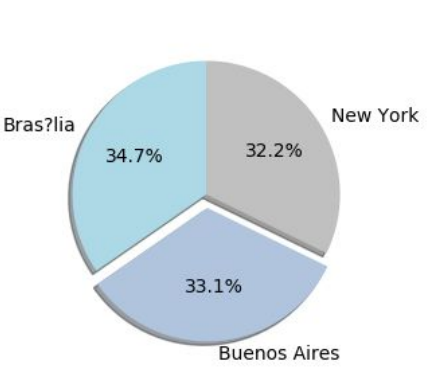

使用者輸入spotify ID，按下fetch data，系統會使用spotify api抓取該帳號所儲存的所有playlist，並取得所有playlist下的歌曲。接著對每一首歌曲進行(1)(e)步驟，分類完該帳號的所有歌曲後，計算所有曲風比例，對城市曲風比例的向量做歐式距離比較。最後使用者按下create，將相似度最高(距離最短)的三個城市以圓餅圖顯示於UI(右圖)，帳號曲風分布也一併顯示(左圖)。

## 4.0 實驗結果與分析

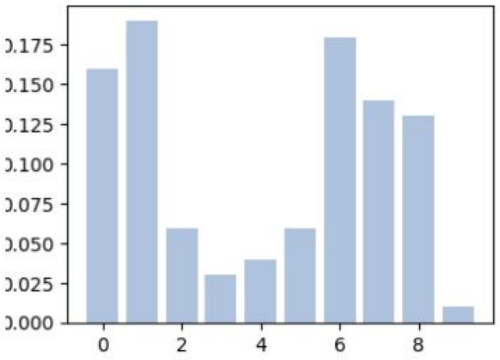
以下為不同spotify user結果：

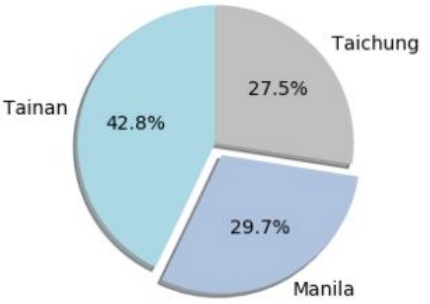
(1) User 1




(2) User 2




根據本結果可以看出單一user的音樂偏好與哪些城市的使用者相似，像是user1跟Brasilia, Buenos Aires, New York的使用者有相似偏好。user2則是跟Tainan, Manila, Taichung的使用者有相似偏好。另外曲風並沒有受到語言干擾，例如播放清單為亞洲地區，城市結果並非全部屬於亞洲。系統產出的偏好配對預測結果不僅可以提供使用者自我認識的樂趣，也可以因為音樂的連結，讓使用者對於不同城市產生興趣，甚至納入未來的造訪城市清單。在商業運用上，也可以藉此了解不同user的偏好相似度，並且開發以音樂播放清單配對為基礎的社交平台。

## 5.0 參考資料


- Get Audio Features for a Track
    - https://developer.spotify.com/documentation/web-api/reference/t

racks/get-audio-features/

- Discovering similarities across my Spotify music using data, clustering and visualization
    - https://towardsdatascience.com/discovering-similarities-across-my-spotify-music-using-data-clustering-and-visualization-52b58e6f547b
- Musical Map of the World
    - https://spotifymaps.github.io/musicalcities/