Import Package

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import spotipy
import spotipy.util as util
```

pip install spotipy

Training

使用csv來訓練情緒的分類

df_feature_selected = df.drop(['f_name', 'a_name', 'title', 'lyrics', 'spot_id', 'sr_json', 'tr_json', "mood"], axis=1)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	f_name	a_name	title	lyrics	spot_id	sr_json	tempo	energy	danceability	loudness	valence	acousticness	r_json	mood
2	angry_all/50	PJ Harvey	50 Ft Queen	Hey I'm one	3fJprjhRxTV	{ "tracks": {	126.366	0.667	0.382	-16.077	0.718	0.000652		1
3	angry_all/A	Keane	A Bad Drea	Why do I ha	17ScnUBsr3	{ "tracks": {	145.035	0.76	0.405	-4.852	0.316	0.00824		1
4	angry_all/Ag	The Faint	Agenda Suic	You could fc	4Mhj9IjSxTc	{ "tracks": {	144.585	0.703	0.574	-7.789	0.275	0.00658		1
5	angry_all/Al	Stiff Little F	Alternative U	There's nothi	0MDJjySh4y	{ "tracks": {	118.181	0.925	0.6	-9.148	0.382	0.00882		1
6	angry_all/An	The Cranber	Animal Insti	Suddenly sor	3J58Ccc5iTt	{ "tracks": {	132.145	0.823	0.622	-5.381	0.605	0.0946		1
7	angry_all/Bl	Living Sacrif	Bloodwork	A simple tes	4wmH0KK1	{ "tracks": {	86.383	0.879	0.344	-5.111	0.479	0.000427		1
8	angry_all/Bc	Pixies	Bone Machii	(This is a sor	58BsYHaWc	{ "tracks": {	115.906	0.678	0.63	-12.757	0.964	0.000677		1
9	angry_all/Bc	Ice Cube	Bop Gun	At these up	10oZiK5HCN	{ "tracks": {	103.05	0.837	0.891	-5.402	0.767	0.283		1
10	angry_all/Bc	Kaiser Chief	Boxing Chai	We went to t	1r7XNA9Cz	{ "tracks": {	119.674	0.553	0.494	-4.848	0.296	0.962		1
11	angry_all/Br	Recoil	Breath Contr	Who wouldr	4CSyaZkuA	{ "tracks": {	162.002	0.645	0.58	-9.784	0.546	0.0609		1
12	angry_all/Br	Evanescence	Bring Me Tc	How can you	0COqiPhxzc	{ "tracks": {	189.931	0.945	0.316	-3.169	0.32	0.00895		1
13	angry_all/Bu	Nine Inch N	Burn	This world re	0nCvBep0qe	{ "tracks": {	89.919	0.912	0.592	-6.81	0.647	0.00146		1
14	angry_all/Ca	Kelis	Caught Out	Yo, this song	1nZkrUFLq2	{ "tracks": {	92.996	0.691	0.848	-6.775	0.922	0.0512		1
15	angry_all/Ch	Ice Cube	Check Yo St	So come on	3NGT0Td7F	{ "tracks": {	101.368	0.735	0.934	-6.668	0.768	0.031		1

Fit to random forests

clf = RandomForestClassifier(min_samples_split=4, criterion="entropy")

features_train, features_test, labels_train, labels_test = train_test_split(features, labels, test_size=0.20, random_state=91)

Test Accuracy: 0.848314606741573

\---

Precision: 0.8506850837253214
Recall: 0.848314606741573
confusion matrix
[[63 9 3 4]
 [9 83 4 2]
 [4 1 69 4]
 [4 0 10 87]]

find the accuracy of the model

try other classifier
kNN、SVM、Decision Tree

Try other classifier

Random Forest Classifier ⭐

```
RandomForest Train Accuracy: 0.9823943661971831
RandomForest Test Accuracy: 0.8202247191011236
-------------------------------------------------
RandomForest Precision: 0.8210756861425652
RandomForest Recall: 0.8202247191011236
RandomForest confusion matrix
[[70  9 11  1]
 [10 82  2  6]
 [ 3  5 61  8]
 [ 2  0  7 79]]
```

Support Vector Machines, SVM

```
SVM Train Accuracy: 0.676056338028169
SVM Test Accuracy: 0.5056179775280899
-------------------------------------------------
SVM Precision: 0.4998380210144346
SVM Recall: 0.5056179775280899
SVM confusion matrix
[[34 20 20 17]
 [22 48 16 14]
 [12 17 32 16]
 [ 9  2 11 66]]
```

Decision Tree Classifier

```
DecisionTree Train Accuracy: 0.9929577464788732
DecisionTree Test Accuracy: 0.797752808988764
-------------------------------------------------
DecisionTree Precision: 0.8009008060227272
DecisionTree Recall: 0.797752808988764
DecisionTree confusion matrix
[[69 10  8  4]
 [ 9 72  9 10]
 [ 2  5 61  9]
 [ 3  0  3 82]]
```

K Nearest Neighbor, kNN

```
kNN  Train Accuracy: 0.652112676056338
kNN  Test Accuracy: 0.508426966292134
-------------------------------------------------
kNN Precision: 0.5111367785310117
kNN  Recall: 0.508426966292134
kNN  confusion matrix
[[43 23 13 12]
 [29 48 12 11]
 [19 14 32 12]
 [15  4 11 58]]
```

Try other classifier

AdaBoost Classifier

```
AdaBoost Train Accuracy: 0.5091549295774648
AdaBoost Test Accuracy: 0.46629213483146065
-----------------------------------------------
AdaBoost Precision: 0.4817275400870143
AdaBoost Recall: 0.46629213483146065
AdaBoost confusion matrix
[[38 15 18 20]
 [17 55 10 18]
 [ 7  7 37 26]
 [10 12 30 36]]
```

Predict New Music

code for spotify get meta

```
username = 'qnwv65t11cplaz4dikhl4mjgi'
CLIENT_ID = '07f9611e9b234caea4fcee288da82e61'
CLIENT_SECRET = '087b1a26a1294bc58a0a89d4a29463e4'
REDIRECT_URI = 'http://localhost/'
SCOPE = 'user-library-read'
```

Account overview

Profile

Username

qnwv65t11cplaz4dikhl4mjgi

Email

chiouchingyi@smail.nchu.edu.tw

Date of birth

1/21/96

Country

TW

My Spotify

An innovative Spotify integration that does creative things.

Client ID   07f9611e9b234caea4fcee288da82e61

Client Secret   087b1a26a1294bc58a0a89d4a29463e4 RESET

HIDE CLIENT SECRET

Predict New Music

Spotify Features

```
[{'danceability': 0.674,          # 跳舞性 (tempo, rhythm stability, beat strength, and overall regularity)
  'energy': 0.881,                # 強度
  'key': 9,                       # 調性 (0 = C、1 = C#/D♭、 2 = D …，沒偵測到=-1)
  'loudness': -2.853,             # 音軌的總響度(dB)
  'mode': 1,                      # 模式 (0=小調minor、1=大調major)
  'speechiness': 0.147,           # 音軌中存在的口語單詞
  'acousticness': 0.296,          # acoustic (0-1之間)
  'instrumentalness': 3.01e-06,   # 預測音軌包含人聲的程度
  'liveness': 0.0793,             # 檢測錄製中是否有觀眾 (值越高代表是live)
  'valence': 0.234,               # 音樂正向性 (值越高情緒越正面)
  'tempo': 98.994,                # bpm (每秒幾拍)
  'type': 'audio_features',
  'id': '5WHTFyqSii0lmT9R21abT8',       # 音樂ID
  'uri': 'spotify:track:5WHTFyqSii0lmT9R21abT8',    # 音樂uri
  'track_href': 'https://api.spotify.com/v1/tracks/5WHTFyqSii0lmT9R21abT8',
  'analysis_url': 'https://api.spotify.com/v1/audio-analysis/5WHTFyqSii0lmT9R21abT8',
  'duration_ms': 178480,          # 音軌時間長度(毫秒)
  'time_signature': 4}]           # 拍號 (每一小節有多少拍)
```

Predict New Music

singerdata.txt - 記...

檔案(F)　編輯(E)　格式(O)　檢視(V)　說明(H)

Camila Cabello
Lika Morgan
Maroon 5
Dean Lewis
Ed Sheeran
DJ Khaled
Clean Bandit
Logic
Kendrick Lamar
Mark Morrison
Keala Settle
Selena Gomez
Celine Dion

Windo　第 13　100%

songdata.txt - 記事本

檔案(F)　編輯(E)　格式(O)　檢視(V)　說明(H)

Havana
Give in To Me
Sugar
Be Alright
Shape of You
I'm the One
Symphony
1-800-273-8255
DNA.
Return Of The Mack - C & J Street Mix
This Is Me
Wolves
When I Fall In Love

Window　第 13 列　100%

Predict New Music

Table = [['Song Name', 'Singer', 'tempo', 'energy', 'loudness', 'danceability', 'valence', 'acousticness', 'happy', 'angry', 'sad', 'relax', 'Mood Class']]

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Havana	Camila Ca	104.988	0.523	-4.333	0.765	0.394	0.184	12	51.33333	36.66667	0	1
2	Give in To	Lika Morg	124.071	0.871	-4.069	0.769	0.672	0.0873	45	10	35	10	0
3	Sugar	Maroon 5	120.076	0.788	-7.055	0.748	0.884	0.0591	20	16	34	30	2
4	Be Alright	Dean Lew	126.684	0.586	-6.319	0.553	0.443	0.697	23.33333	22.66667	22.5	31.5	3
5	Shape of Y	Ed Sheera	95.977	0.652	-3.183	0.825	0.931	0.581	20	40	30	10	1
6	"Im the O	DJ Khaled	80.924	0.668	-4.284	0.609	0.811	0.0552	5.714286	26	58.28571	10	2
7	Symphony	Clean Ban	122.863	0.629	-4.581	0.707	0.457	0.259	5.714286	40	34.28571	20	1
8	1-800-273	Logic	100.021	0.574	-7.788	0.62	0.352	0.57	15	41.66667	20	23.33333	1
9	DNA.	Kendrick	139.913	0.523	-6.664	0.638	0.422	0.00454	6	14	60	20	2
10	Return Of	Mark Mor	95.487	0.833	-5.379	0.715	0.612	0.00631	25.71429	6	38.28571	30	2
11	This Is Me	Keala Sett	191.702	0.704	-7.276	0.284	0.1	0.00583	10	6	64	20	2
12	Wolves	Selena Go	124.946	0.807	-4.59	0.72	0.305	0.129	15.71429	10	44.28571	30	2
13	When I Fa	Celine Di	68.353	0.241	-13.251	0.19	0.107	0.577	56.66667	0	43.33333	0	0

Song Name
Singer

6 features

percent of 4 emotion

emotion result

Problems

yourOutputTest.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Havana	Camila Ca	104.988	0.523	-4.333	0.765	0.394	0.184	12	51.33333	36.66667	0	1
2	Give in To	Lika Morg	124.071	0.871	-4.069	0.769	0.672	0.0873	45	10	35	10	0
3	Sugar	Maroon 5	120.076	0.788	-7.055	0.748	0.884	0.0591	20	16	34	30	2
4	Be Alright	Dean Lew	126.684	0.586	-6.319	0.553	0.443	0.697	23.33333	22.66667	22.5	31.5	3
5	Shape of Y	Ed Sheeran	95.977	0.652	-3.183	0.825	0.931	0.581	20	40	30	10	1
6	"Im the O	DJ Khaled	80.924	0.668	-4.284	0.609	0.811	0.0552	5.714286	26	58.28571	10	2
7	Symphony	Clean Ban	122.863	0.629	-4.581	0.707	0.457	0.259	5.714286	40	34.28571	20	1
8	1-800-273	Logic	100.021	0.574	-7.788	0.62	0.352	0.57	15	41.66667	20	23.33333	1
9	DNA.	Kendrick l	139.913	0.523	-6.664	0.638	0.422	0.00454	6	14	60	20	2
10	Return Of	Mark Mor	95.487	0.833	-5.379	0.715	0.612	0.00631	25.71429	6	38.28571	30	2
11	This Is Me	Keala Sett	191.702	0.704	-7.276	0.284	0.1	0.00583	10	6	64	20	2
12	Wolves	Selena Go	124.946	0.807	-4.59	0.72	0.305	0.129	15.71429	10	44.28571	30	2
13	When I Fa	Celine Dic	68.353	0.241	-13.251	0.19	0.107	0.577	56.66667	0	43.33333	0	0

yourOutputTest2.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Havana	Camila Ca	104.988	0.523	-4.333	0.765	0.394	0.184	35	40	2.5	22.5	1
2	Give in To	Lika Morg	124.071	0.871	-4.069	0.769	0.672	0.0873	25	45	12.5	17.5	1
3	Sugar	Maroon 5	120.076	0.788	-7.055	0.748	0.884	0.0591	23.33333	20	26.66667	30	3
4	Be Alright	Dean Lew	126.684	0.586	-6.319	0.553	0.443	0.697	56	4	20	20	0
5	Shape of Y	Ed Sheeran	95.977	0.652	-3.183	0.825	0.931	0.581	30.83333	26.66667	32.5	10	2
6	"Im the O	DJ Khaled	80.924	0.668	-4.284	0.609	0.811	0.0552	20	68	12	0	1
7	Symphony	Clean Ban	122.863	0.629	-4.581	0.707	0.457	0.259	25	50	25	0	1
8	1-800-273	Logic	100.021	0.574	-7.788	0.62	0.352	0.57	49.33333	14	33.33333	3.333333	0
9	DNA.	Kendrick l	139.913	0.523	-6.664	0.638	0.422	0.00454	15	33	32	20	1
10	Return Of	Mark Mor	95.487	0.833	-5.379	0.715	0.612	0.00631	33.33333	20	36.66667	10	2
11	This Is Me	Keala Sett	191.702	0.704	-7.276	0.284	0.1	0.00583	30	0	50	20	2
12	Wolves	Selena Go	124.946	0.807	-4.59	0.72	0.305	0.129	40	50	10	0	1
13	When I Fa	Celine Dic	68.353	0.241	-13.251	0.19	0.107	0.577	15	0	55	30	2

解決：np.random.seed(1)
每一次跑出來都一樣

Music Dataset
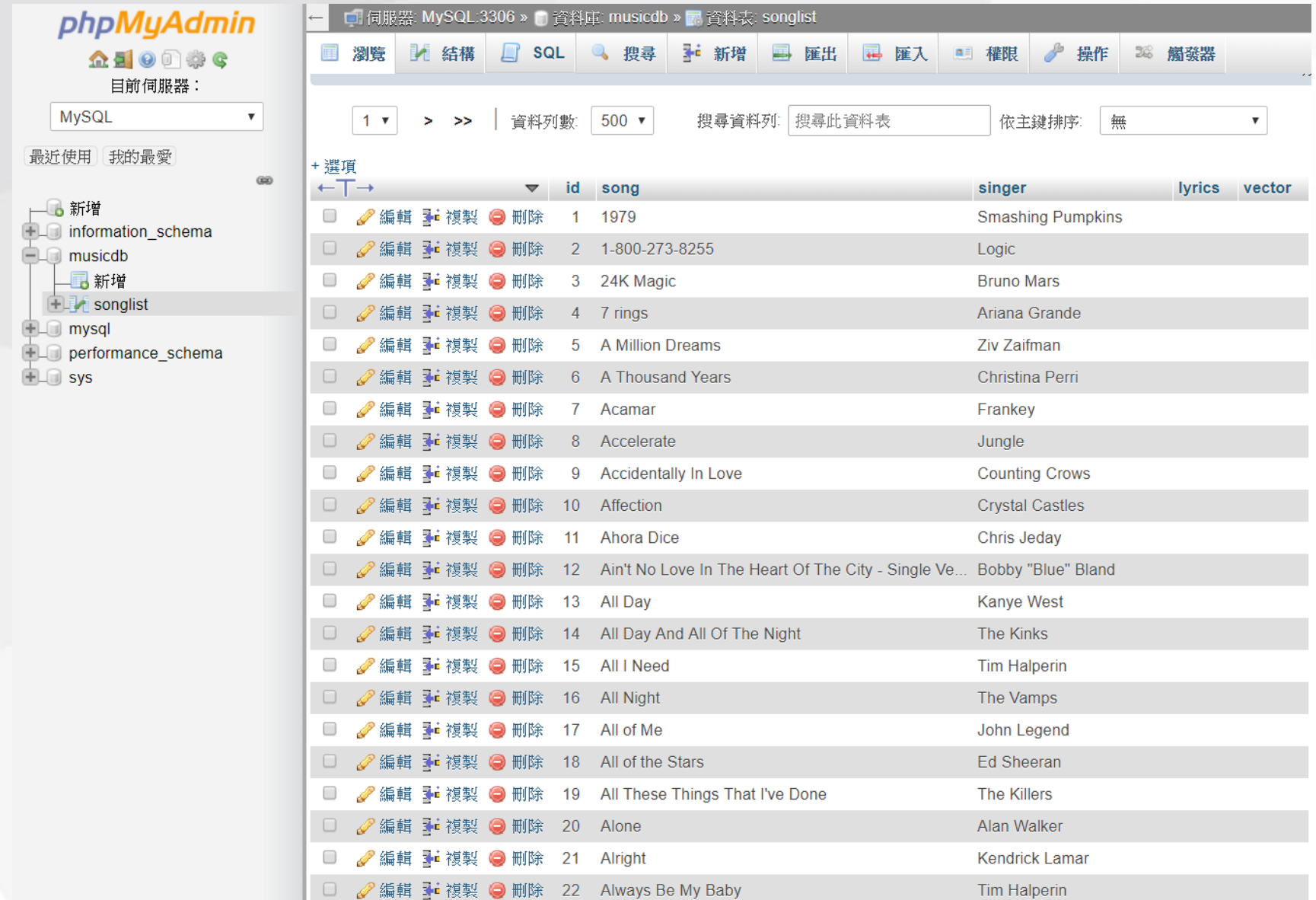
Happy : 117

Angry : 134

Sad : 177

Relax : 122

Total : 550

yourOutput_0316.csv

Music Dataset

☐	✏ 編輯	⤵ 複製	⊖ 刪除	1	1979	Smashing Pumpkins
☐	✏ 編輯	⤵ 複製	⊖ 刪除	2	1-800-273-8255	Logic
☐	✏ 編輯	⤵ 複製	⊖ 刪除	3	24K Magic	Bruno Mars
☐	✏ 編輯	⤵ 複製	⊖ 刪除	4	7 rings	Ariana Grande
☐	✏ 編輯	⤵ 複製	⊖ 刪除	5	A Million Dreams	Ziv Zaifman
☐	✏ 編輯	⤵ 複製	⊖ 刪除	6	A Thousand Years	Christina Perri
☐	✏ 編輯	⤵ 複製	⊖ 刪除	7	Acamar	Frankey
☐	✏ 編輯	⤵ 複製	⊖ 刪除	8	Accelerate	Jungle
☐	✏ 編輯	⤵ 複製	⊖ 刪除	9	Accidentally In Love	Counting Crows
☐	✏ 編輯	⤵ 複製	⊖ 刪除	10	Affection	Crystal Castles
☐	✏ 編輯	⤵ 複製	⊖ 刪除	11	Ahora Dice	Chris Jeday
☐	✏ 編輯	⤵ 複製	⊖ 刪除	12	Ain't No Love In The Heart Of The City	Bobby "Blue" Bland
☐	✏ 編輯	⤵ 複製	⊖ 刪除	13	All Day	Kanye West
☐	✏ 編輯	⤵ 複製	⊖ 刪除	14	All Day And All Of The Night	The Kinks
☐	✏ 編輯	⤵ 複製	⊖ 刪除	15	All I Need	Tim Halperin
☐	✏ 編輯	⤵ 複製	⊖ 刪除	16	All Night	The Vamps
☐	✏ 編輯	⤵ 複製	⊖ 刪除	17	All of Me	John Legend
☐	✏ 編輯	⤵ 複製	⊖ 刪除	18	All of the Stars	Ed Sheeran
☐	✏ 編輯	⤵ 複製	⊖ 刪除	19	All These Things That I've Done	The Killers
☐	✏ 編輯	⤵ 複製	⊖ 刪除	20	Alone	Alan Walker
☐	✏ 編輯	⤵ 複製	⊖ 刪除	21	Alright	Kendrick Lamar
☐	✏ 編輯	⤵ 複製	⊖ 刪除	22	Always Be My Baby	Tim Halperin
☐	✏ 編輯	⤵ 複製	⊖ 刪除	23	Always Remember Us This Way	Lady Gaga
☐	✏ 編輯	⤵ 複製	⊖ 刪除	24	Amazing	Westlife
☐	✏ 編輯	⤵ 複製	⊖ 刪除	25	Amsterdam	Coldplay

songdata7_0408.txt

singerdata7_0408.txt

移除歌名有feat等等

yourOutput_0408.csv

Features Analyze
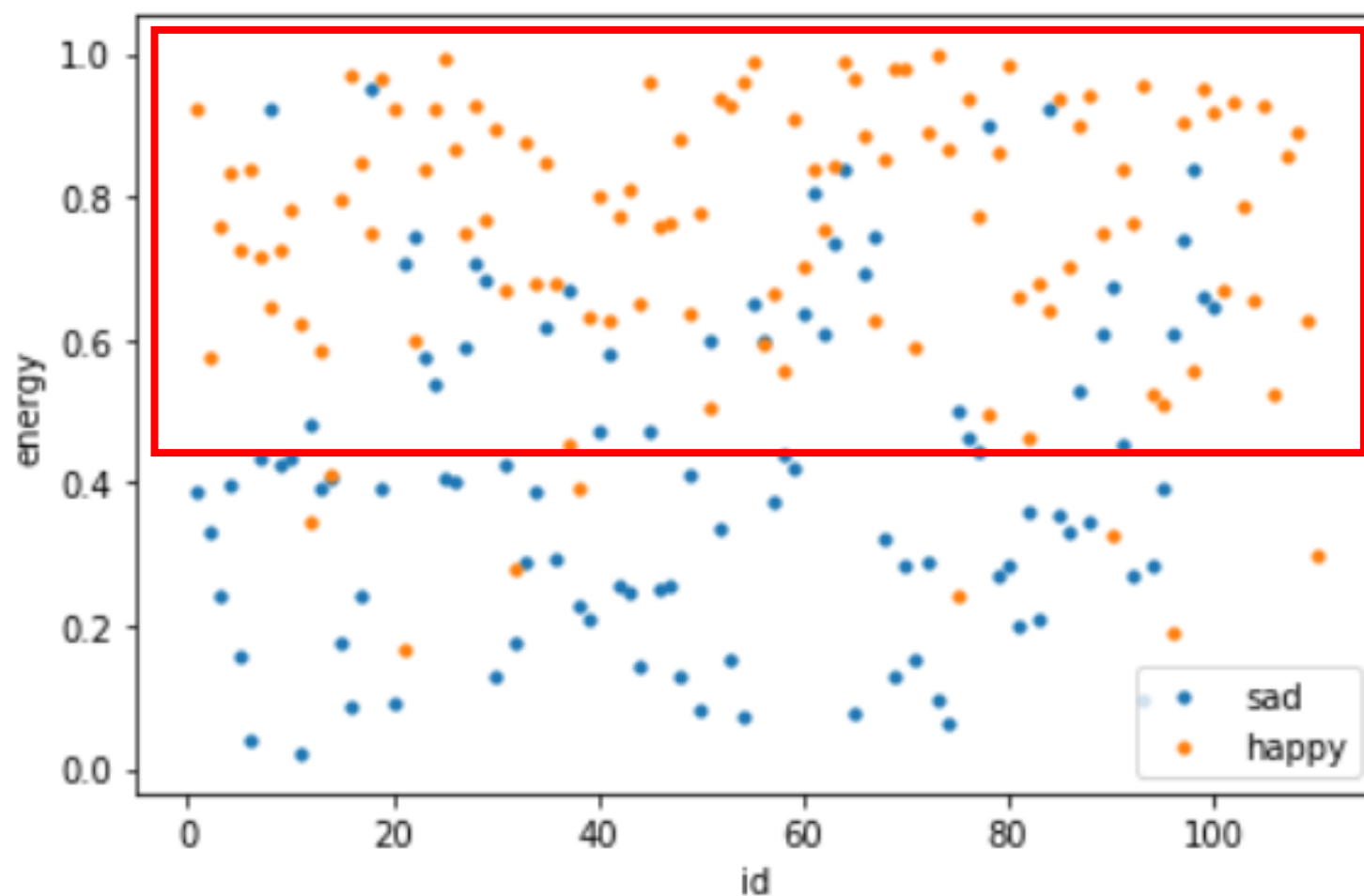
happy → energy高 / ~~sad → energy低~~

Features Analyze
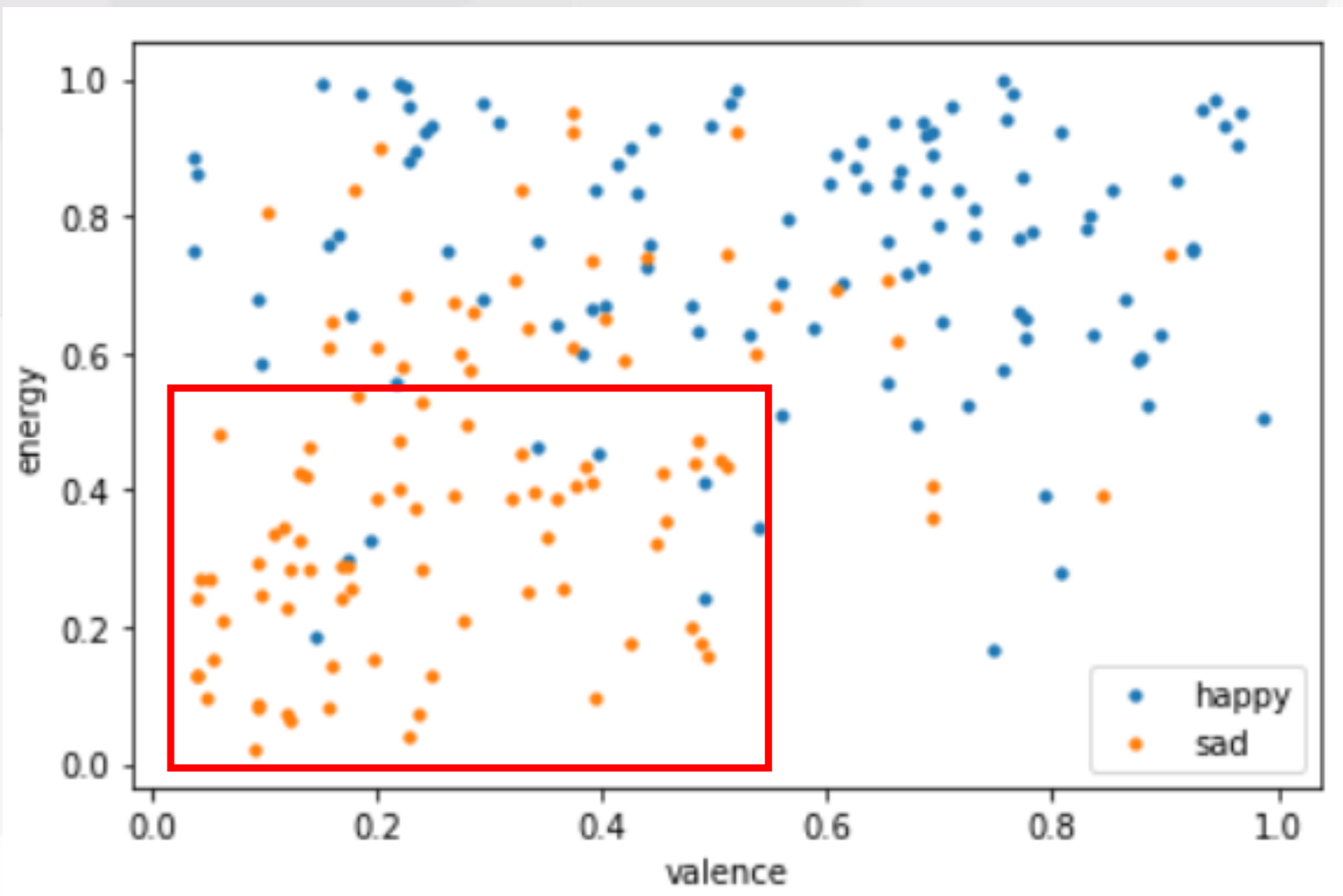
sad → valence低、energy低

Features Analyze
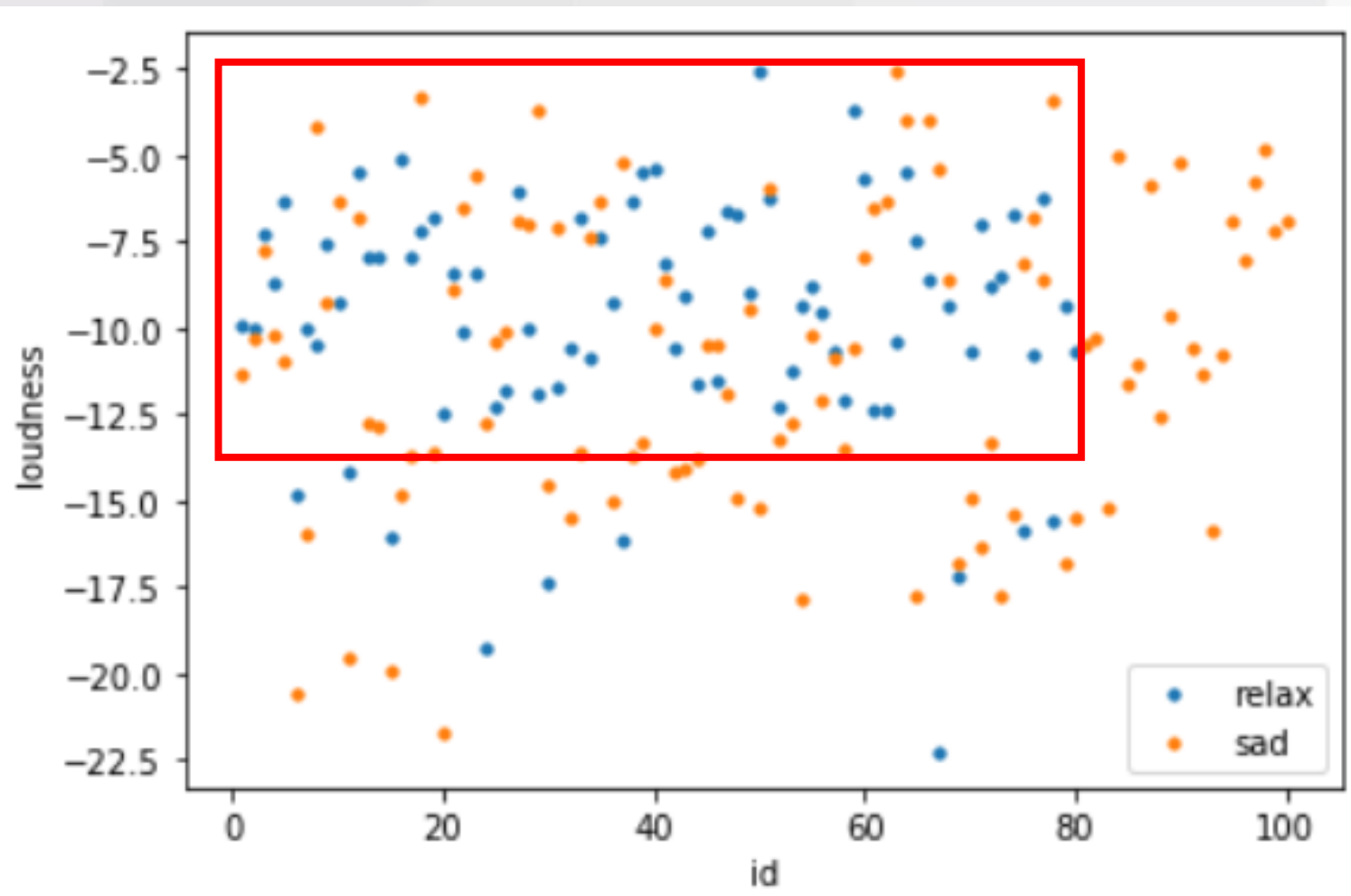
relax → loudness大

Features Analyze
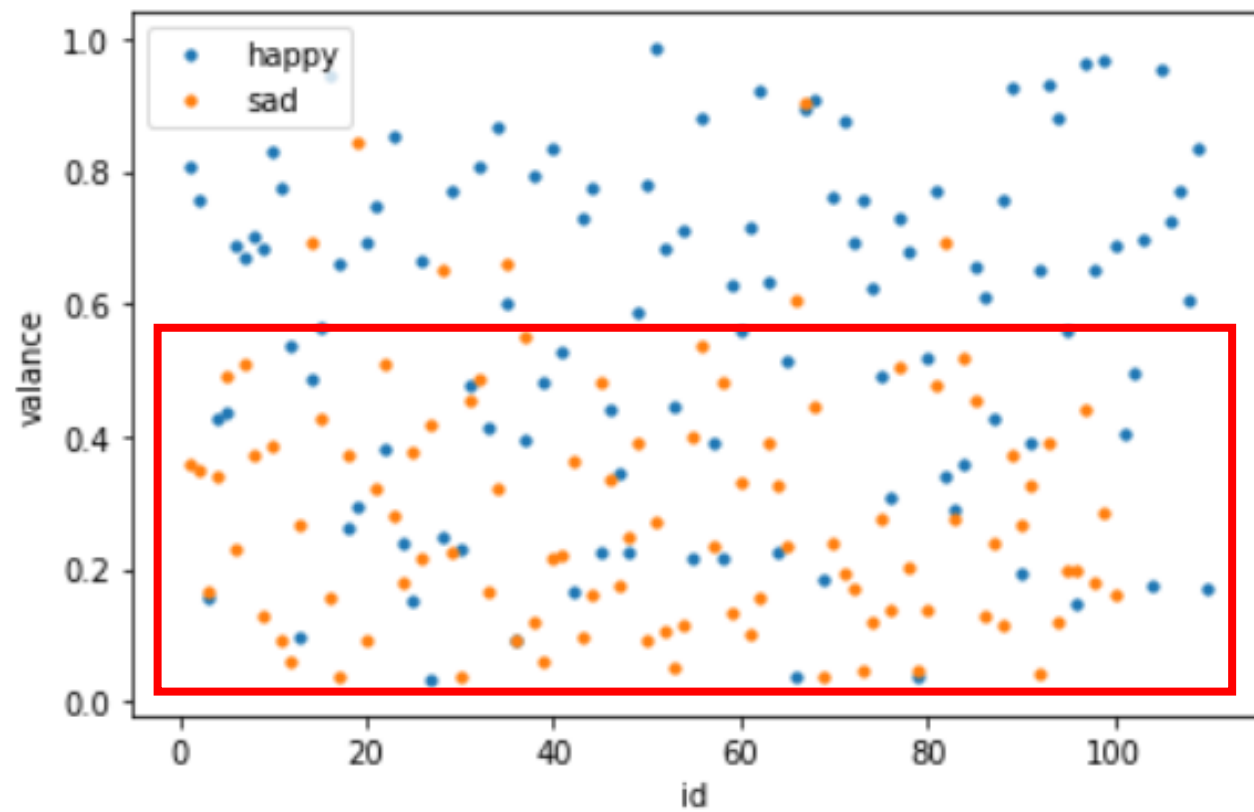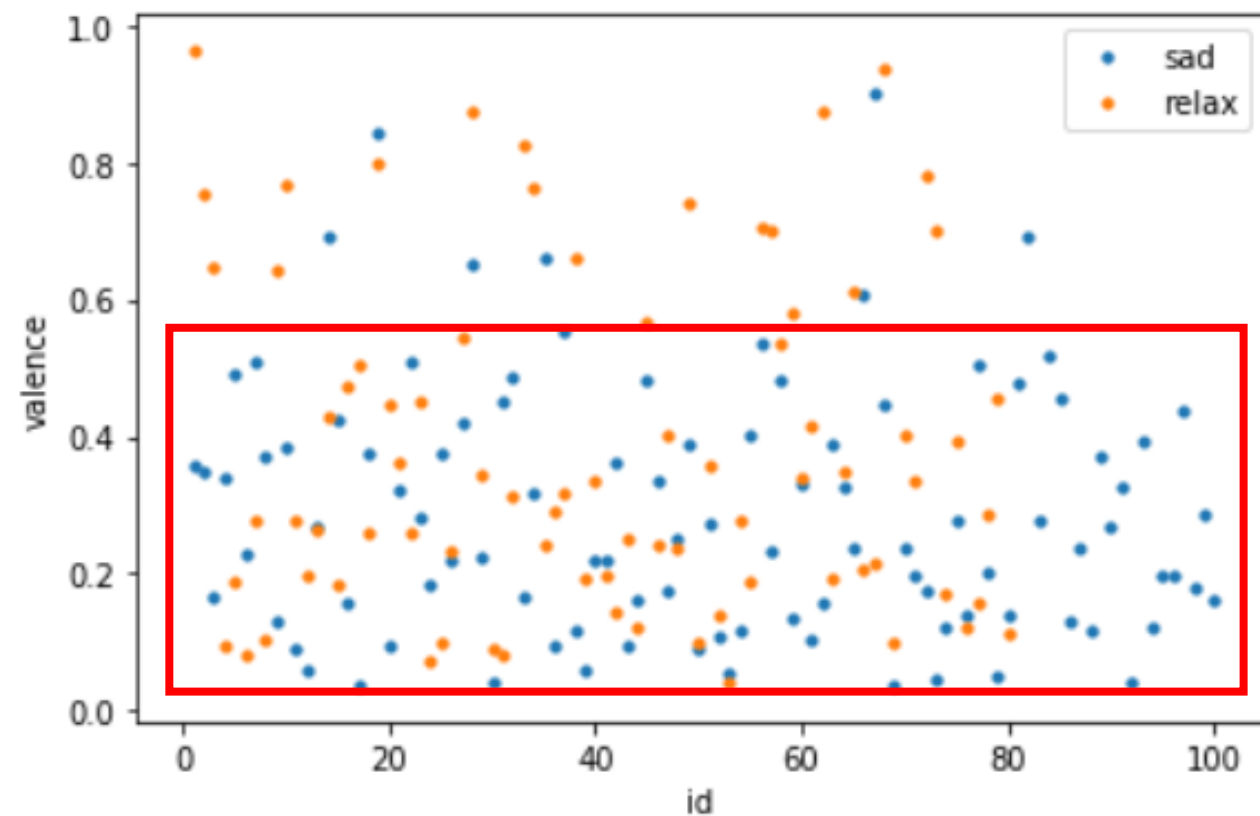
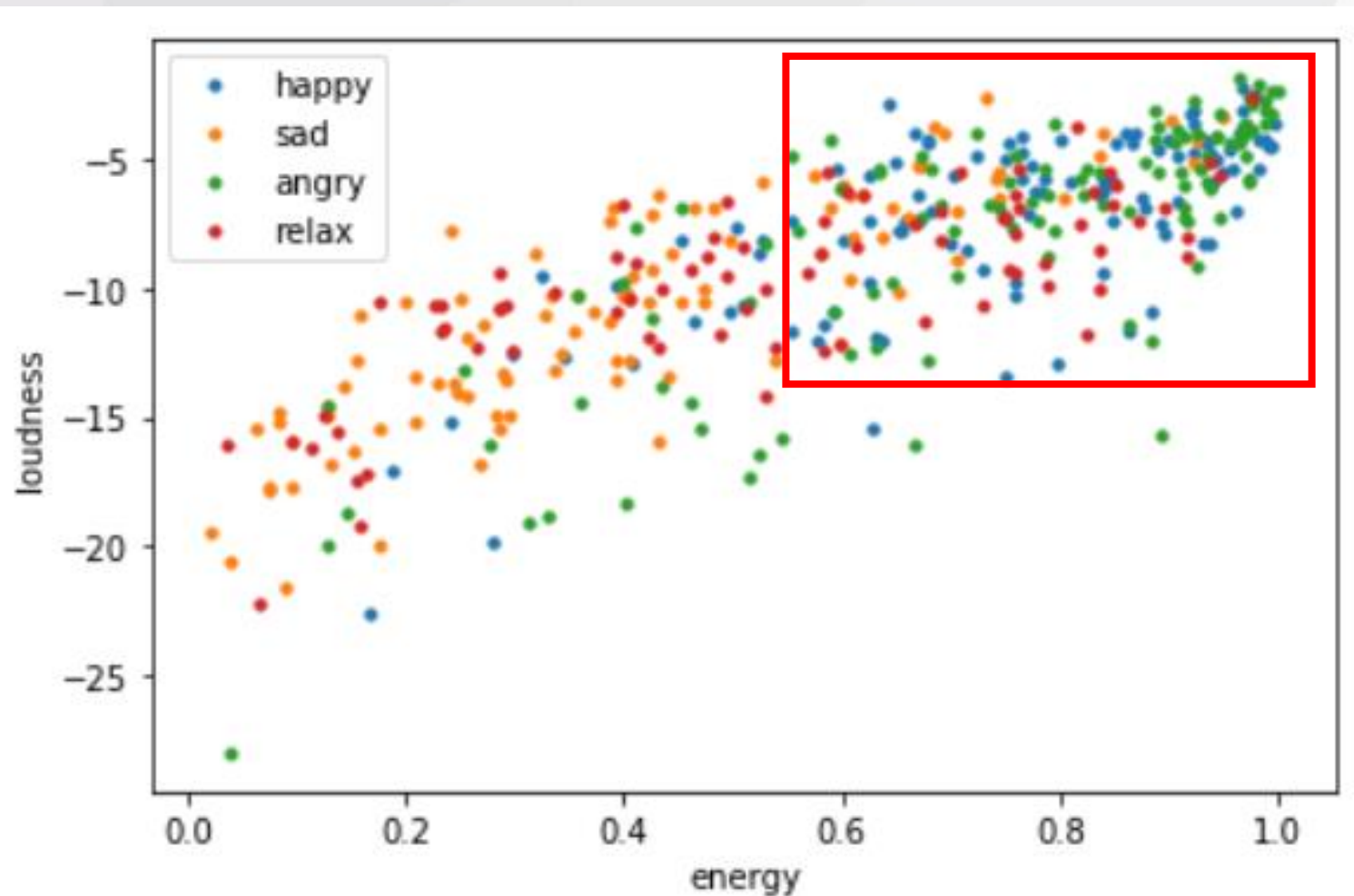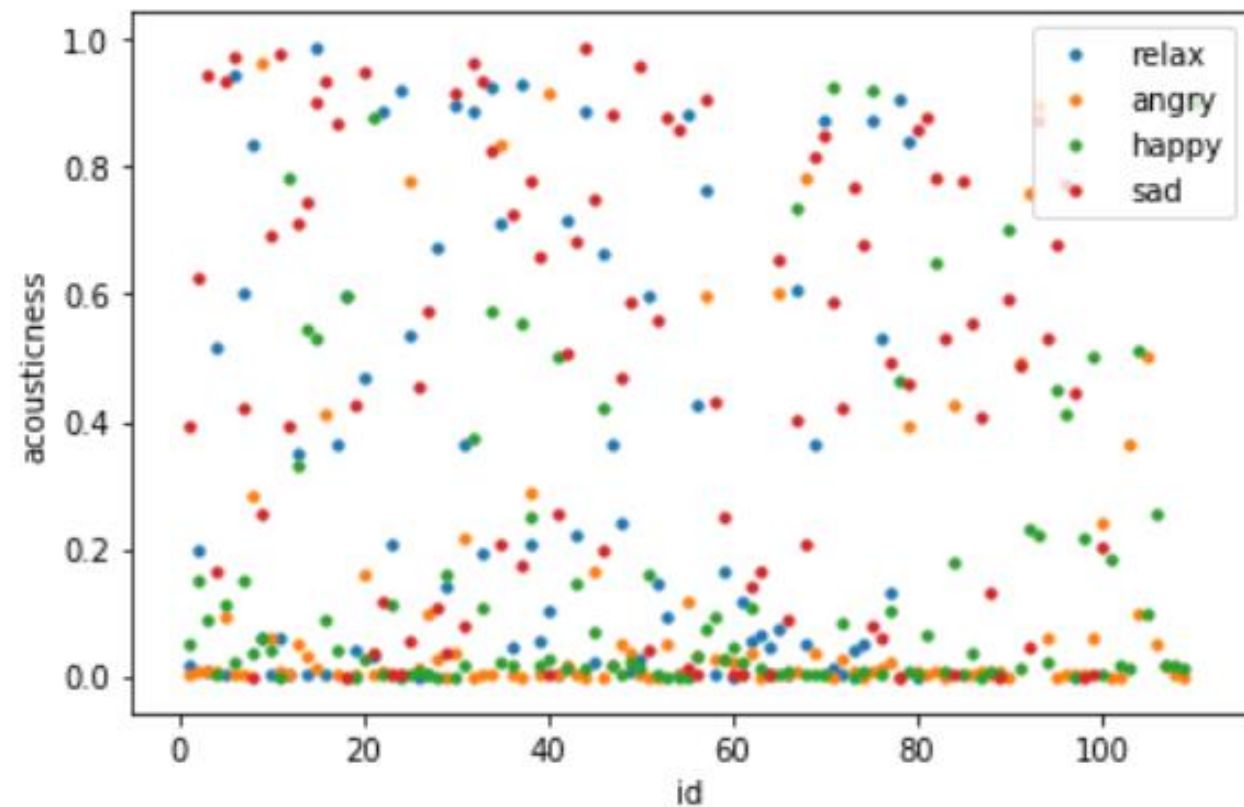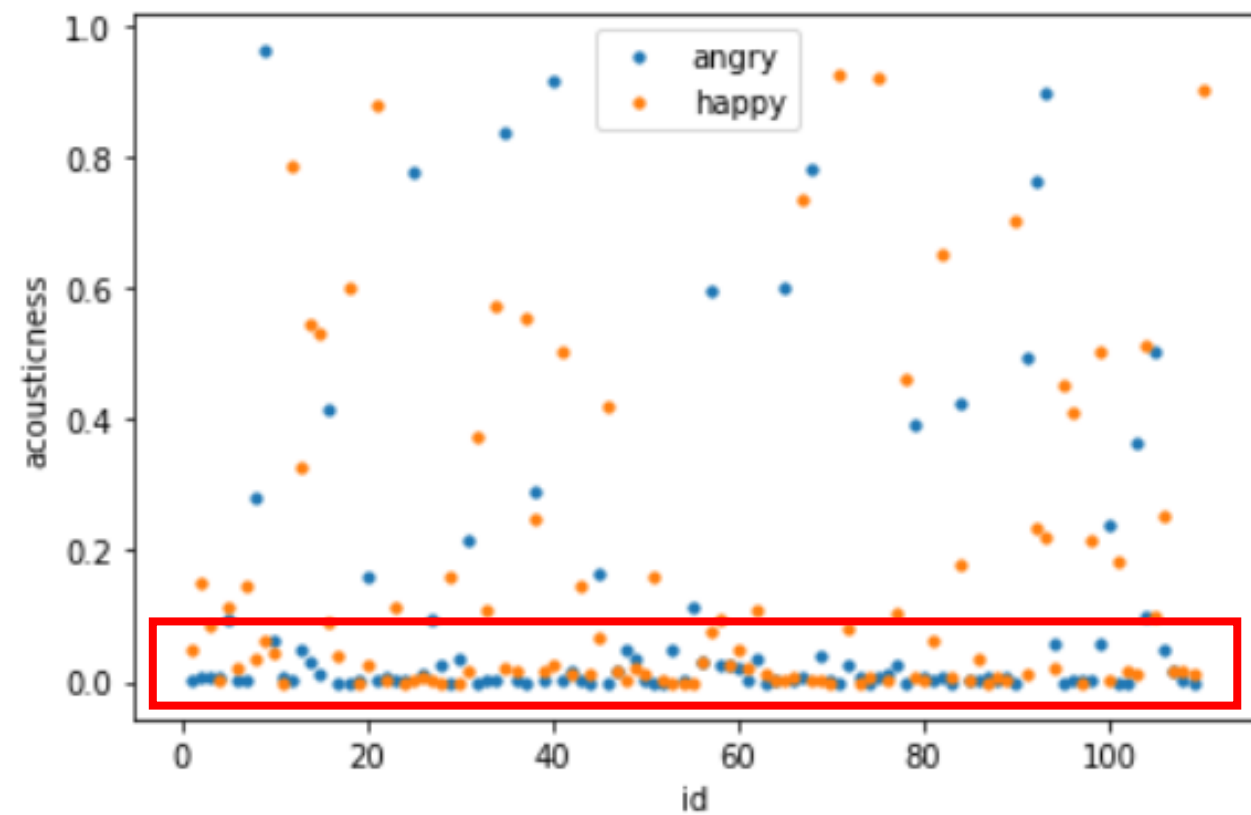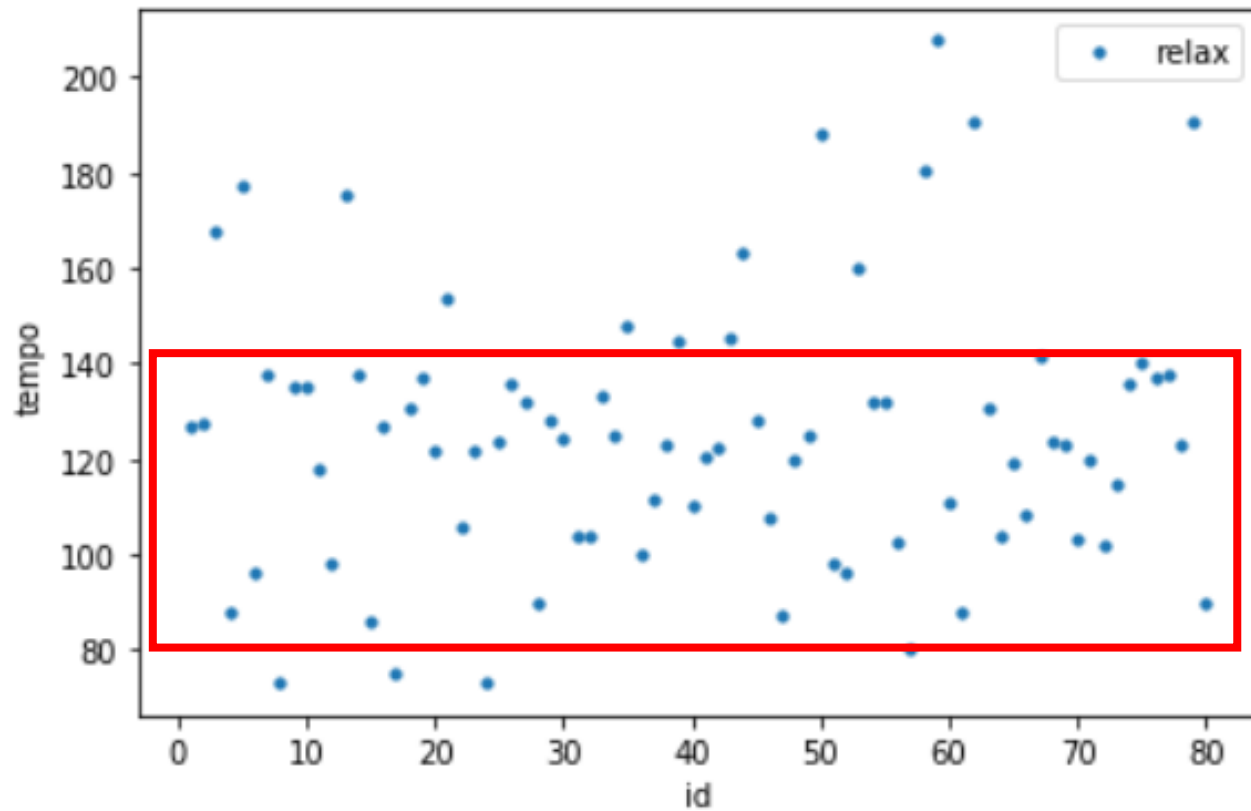sad → valence低   sad、relax → valence低

Features Analyze

happy、angry → energy高、loudness大
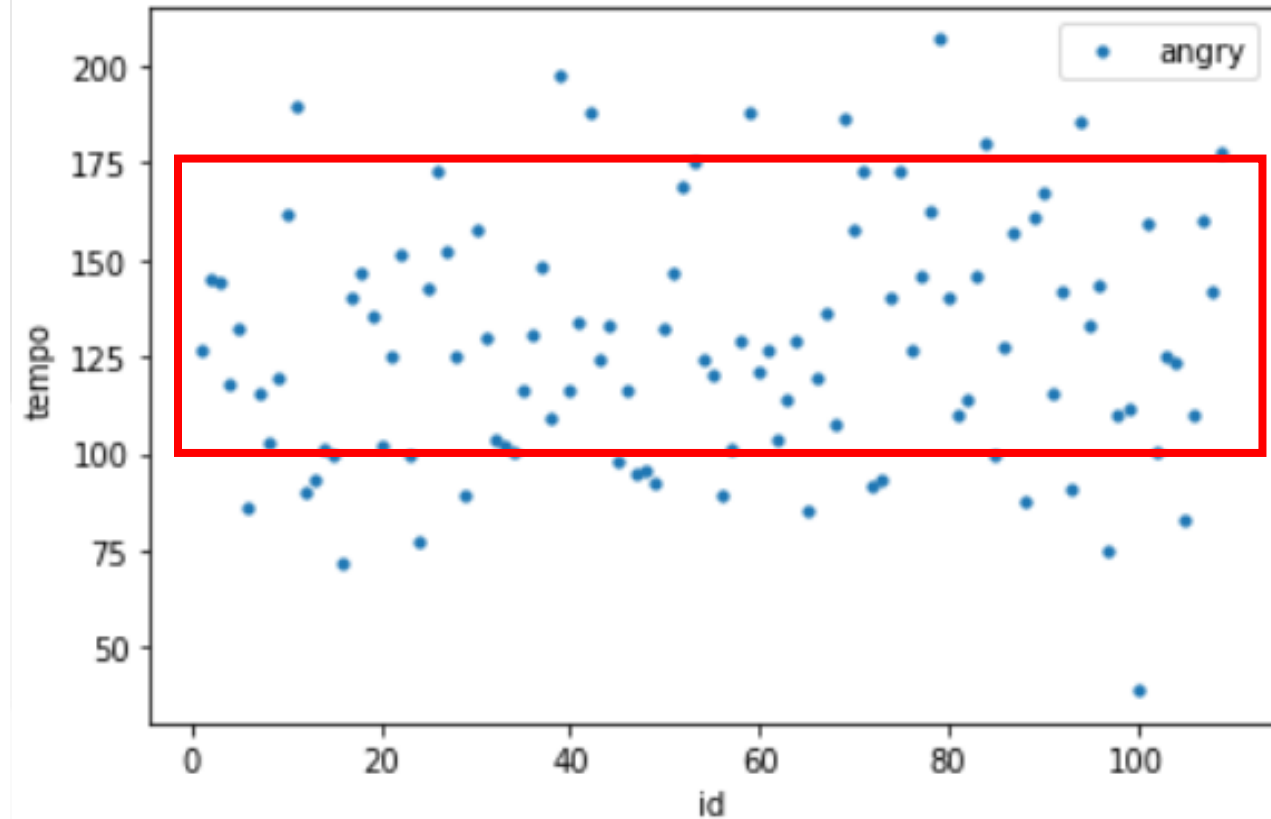
Features Analyze

angry、happy → acousticness低

Features Analyze

relax → tempo(80-140)

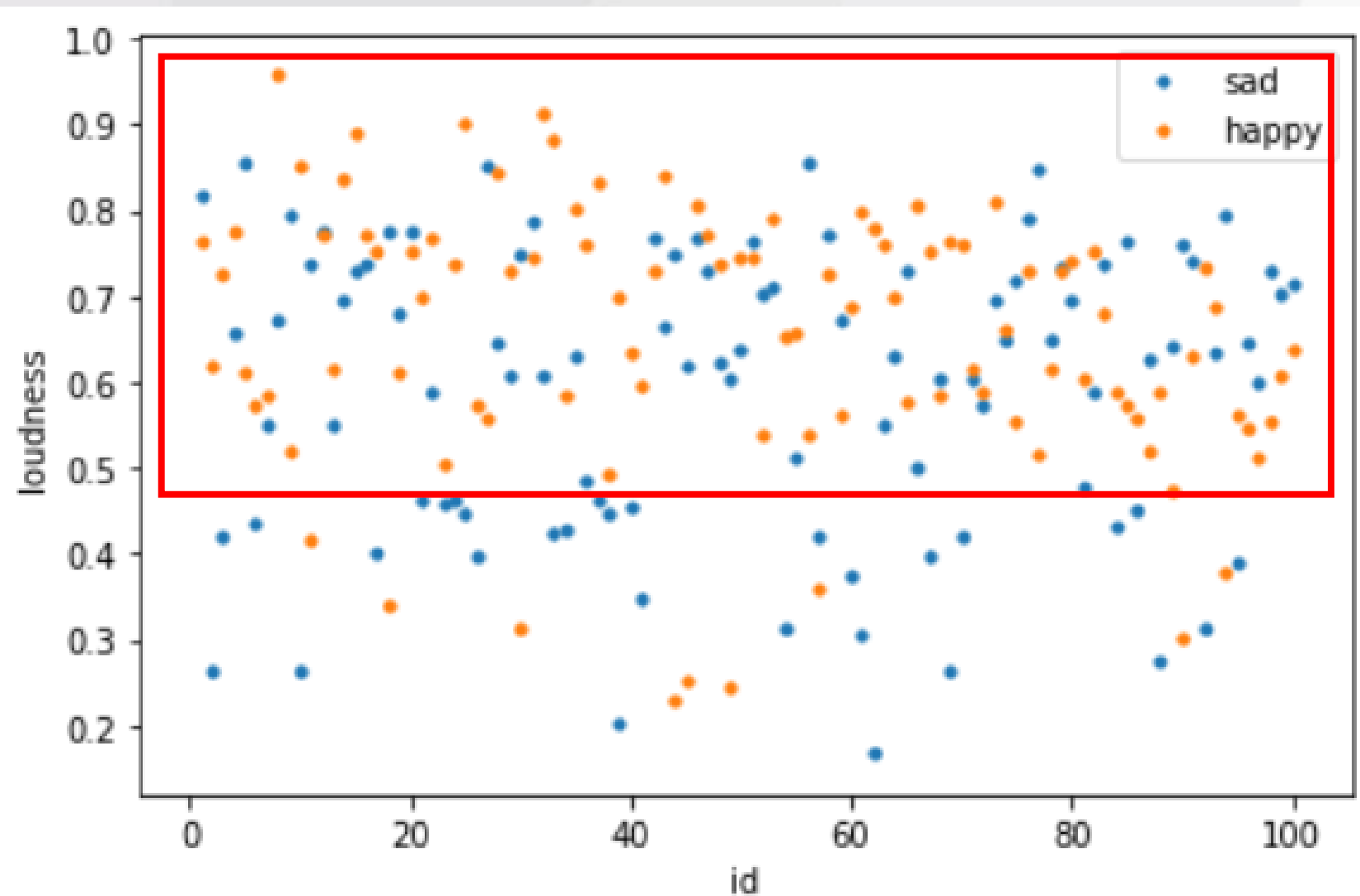angry → tempo(100-175)

Features Analyze

sad → valence低、~~energy低~~   angry、happy 區分？

relax → loudness大、tempo(80-140)

happy、angry → energy高、loudness大、 acousticness低

angry → tempo(100-175)

Features Analyze

happy → loudness大

Features Analyze
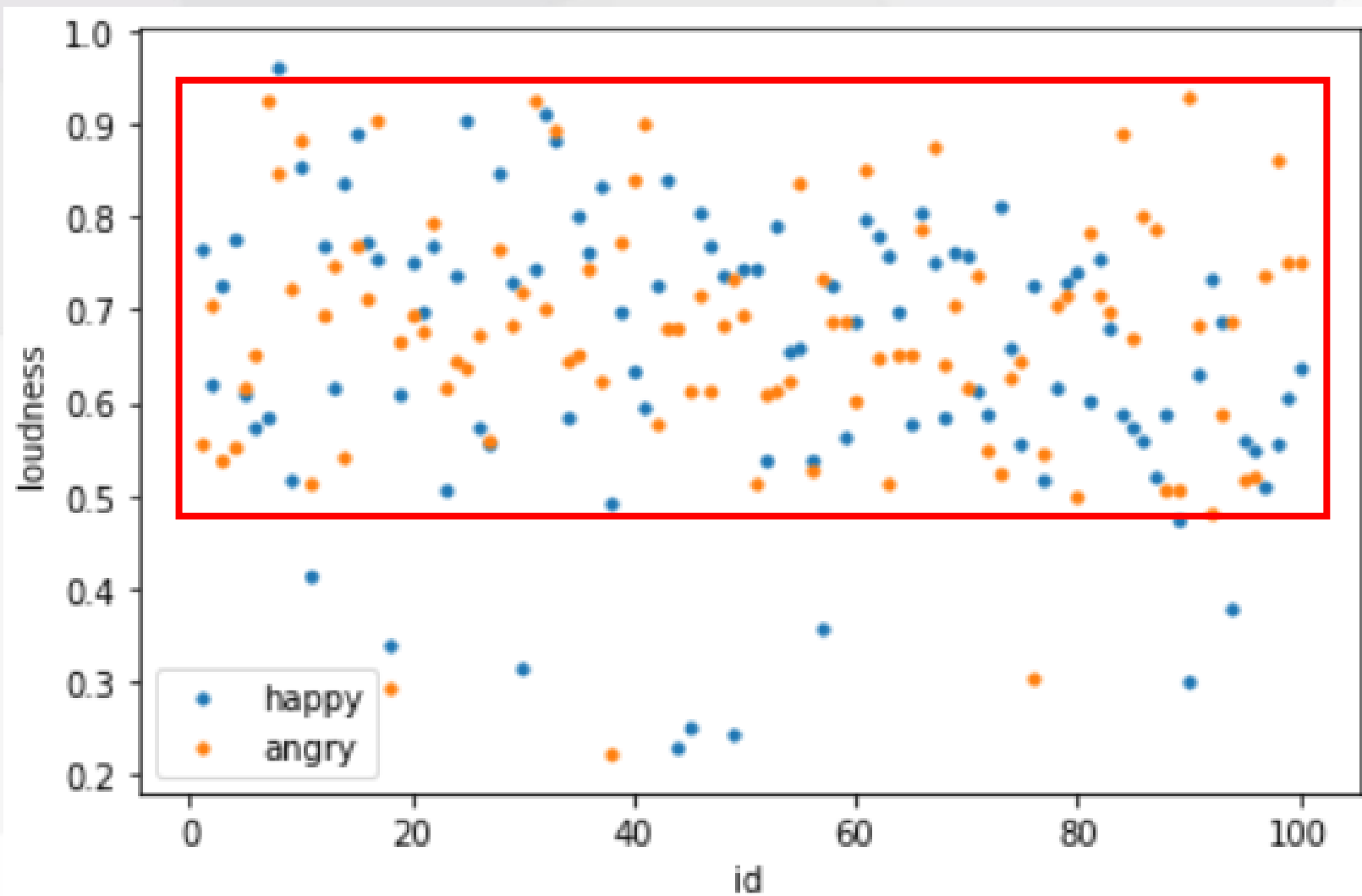
happy、angry → loudness大

Features Analyze
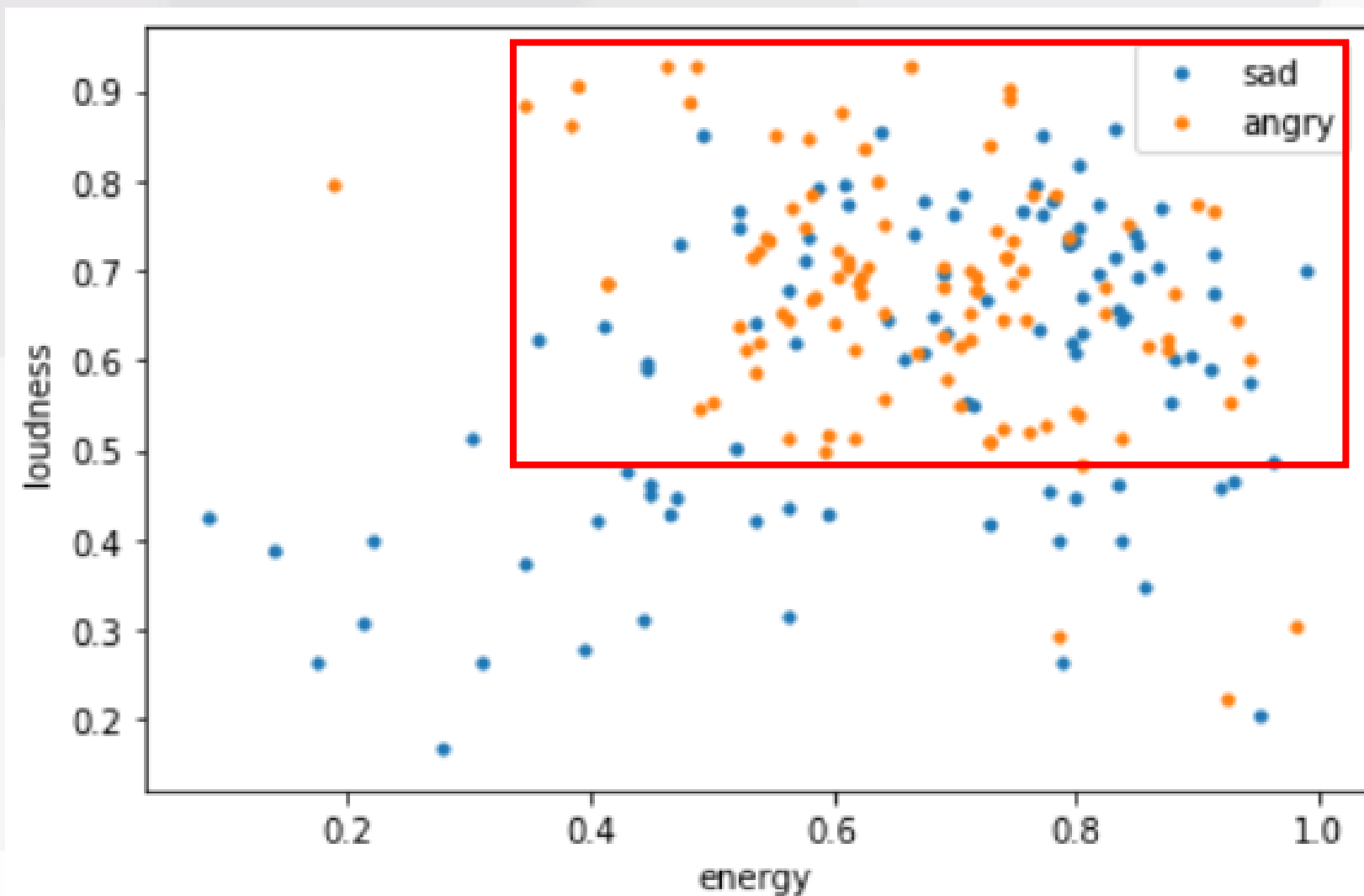
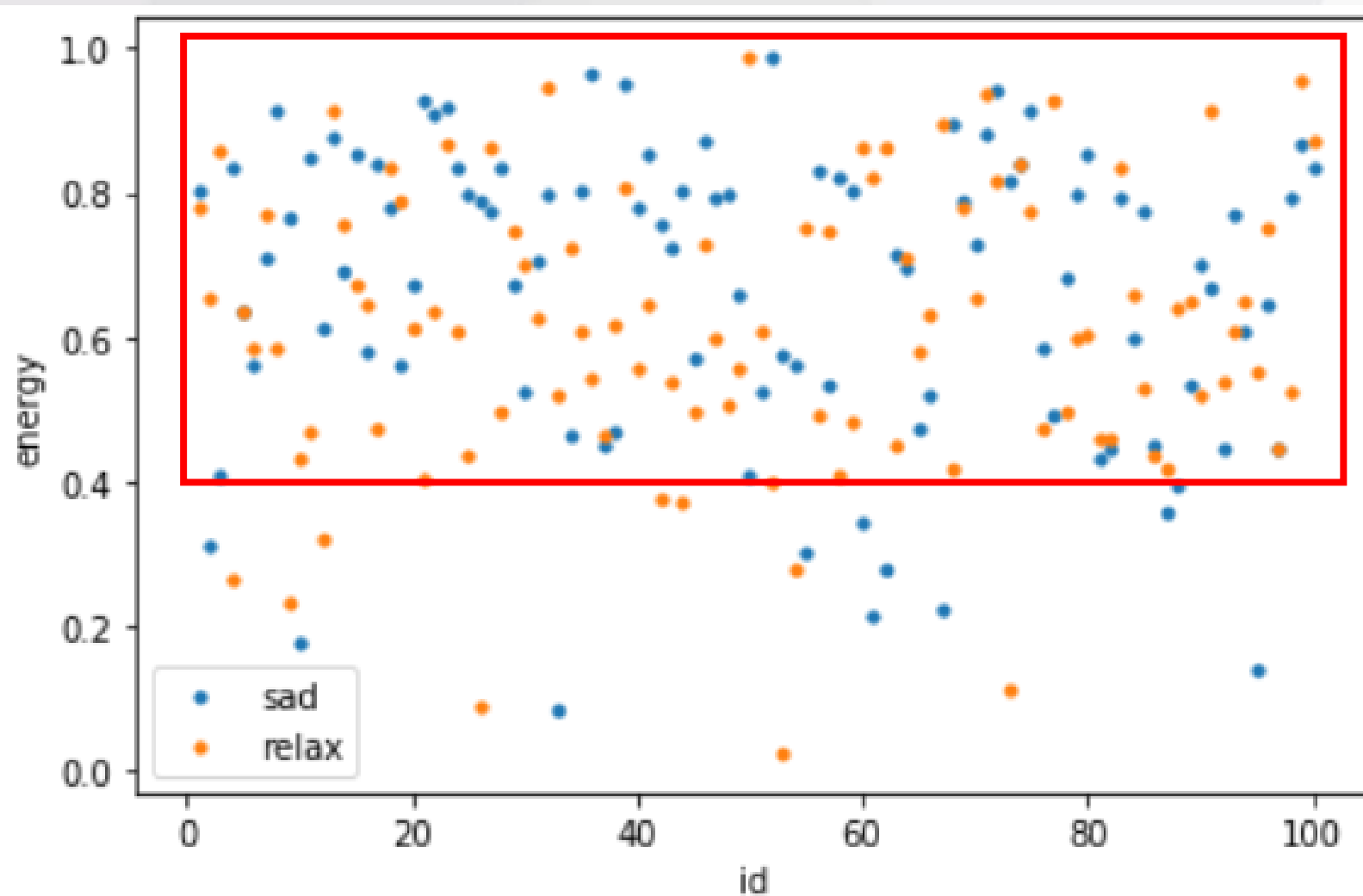angry → energy高、loudness大
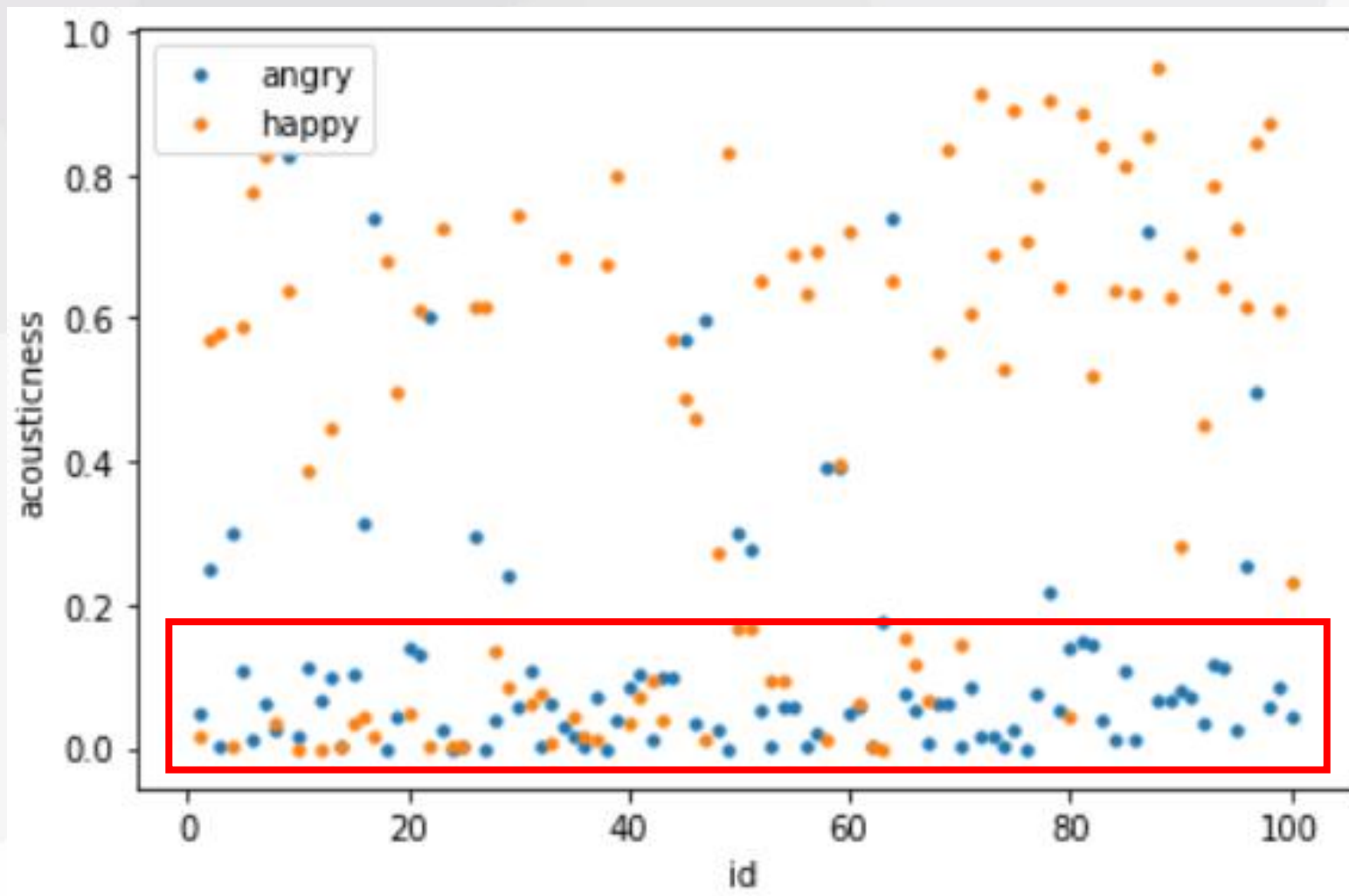
Features Analyze

angry → acousticness低
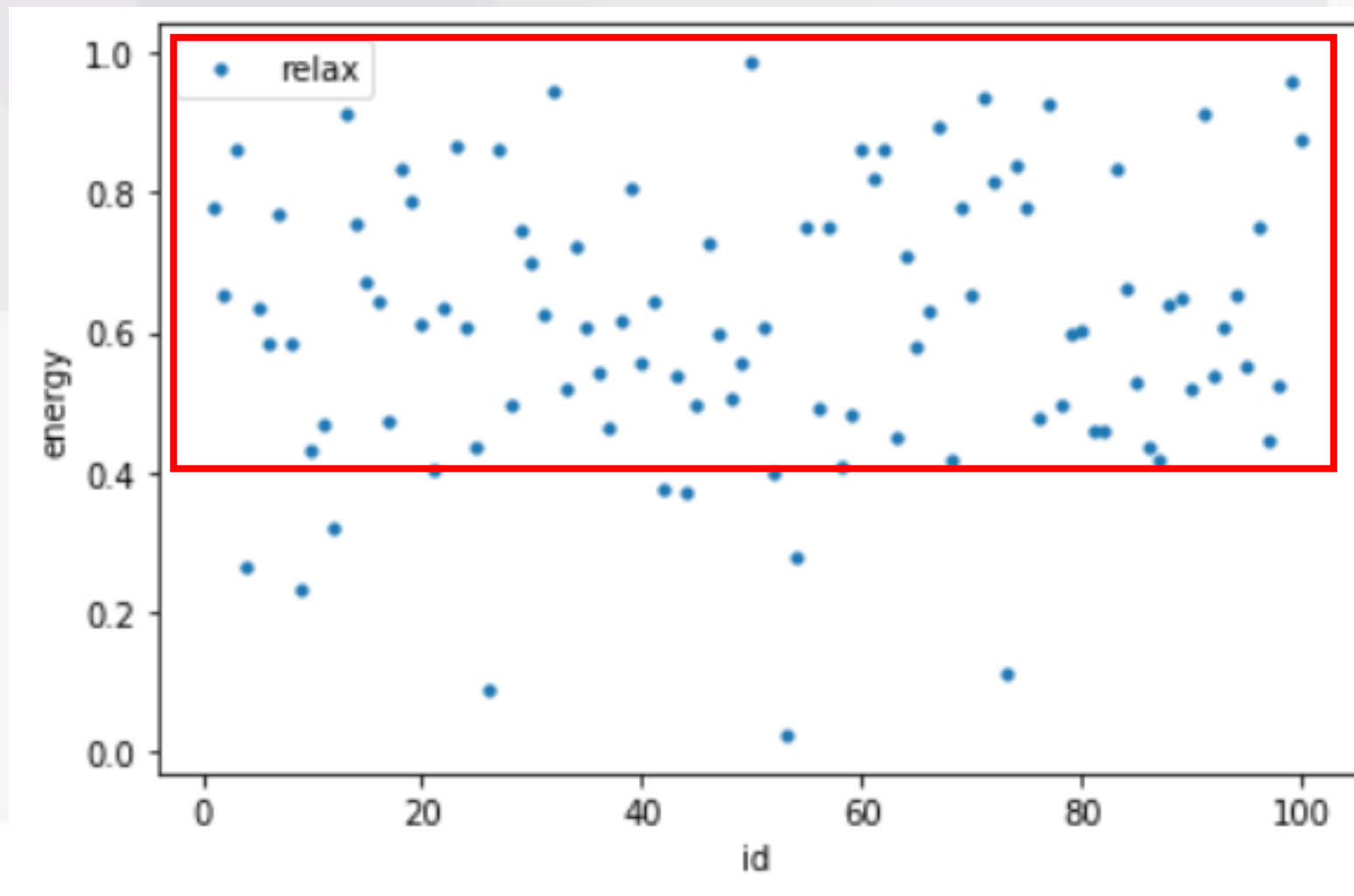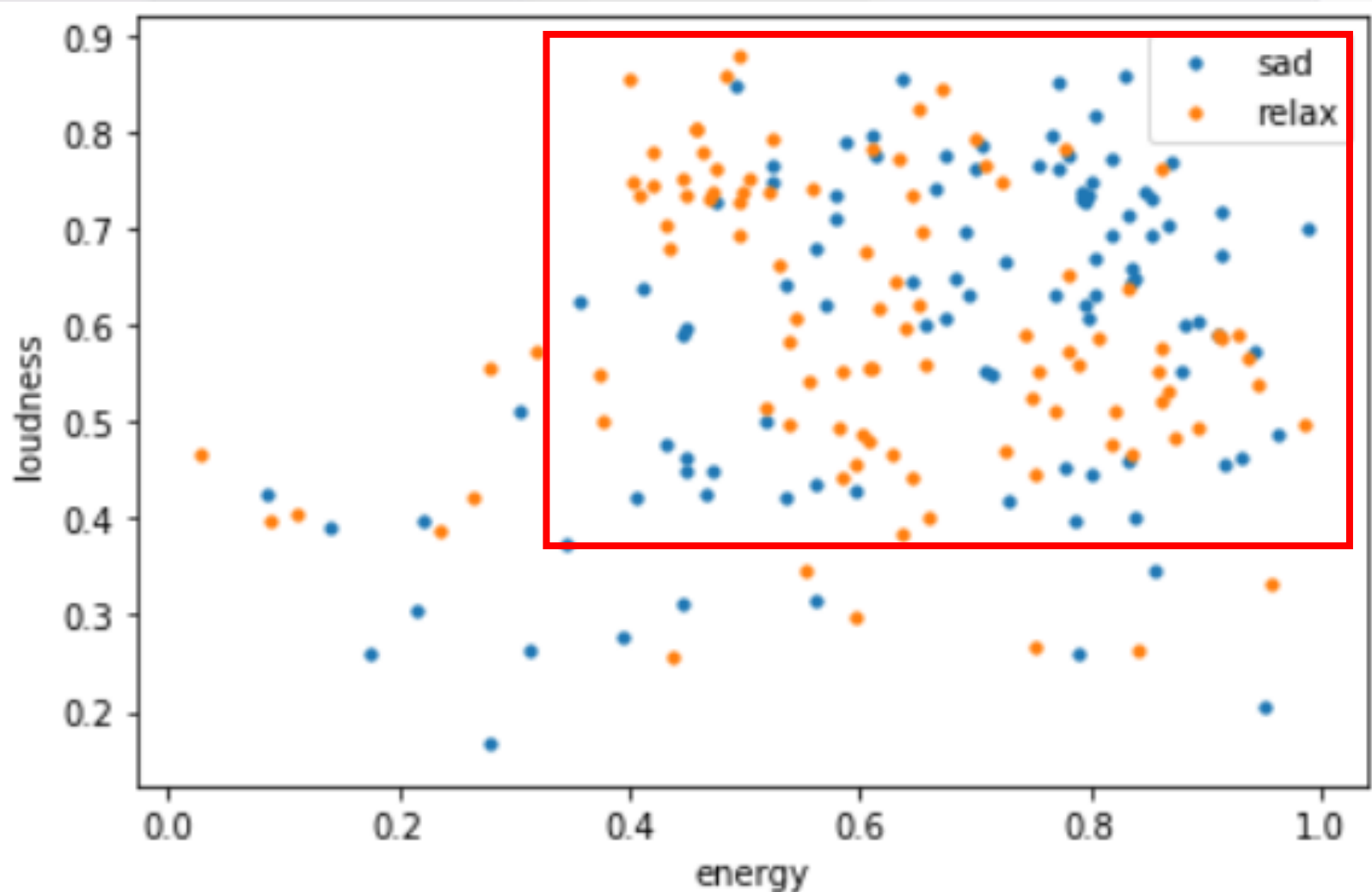
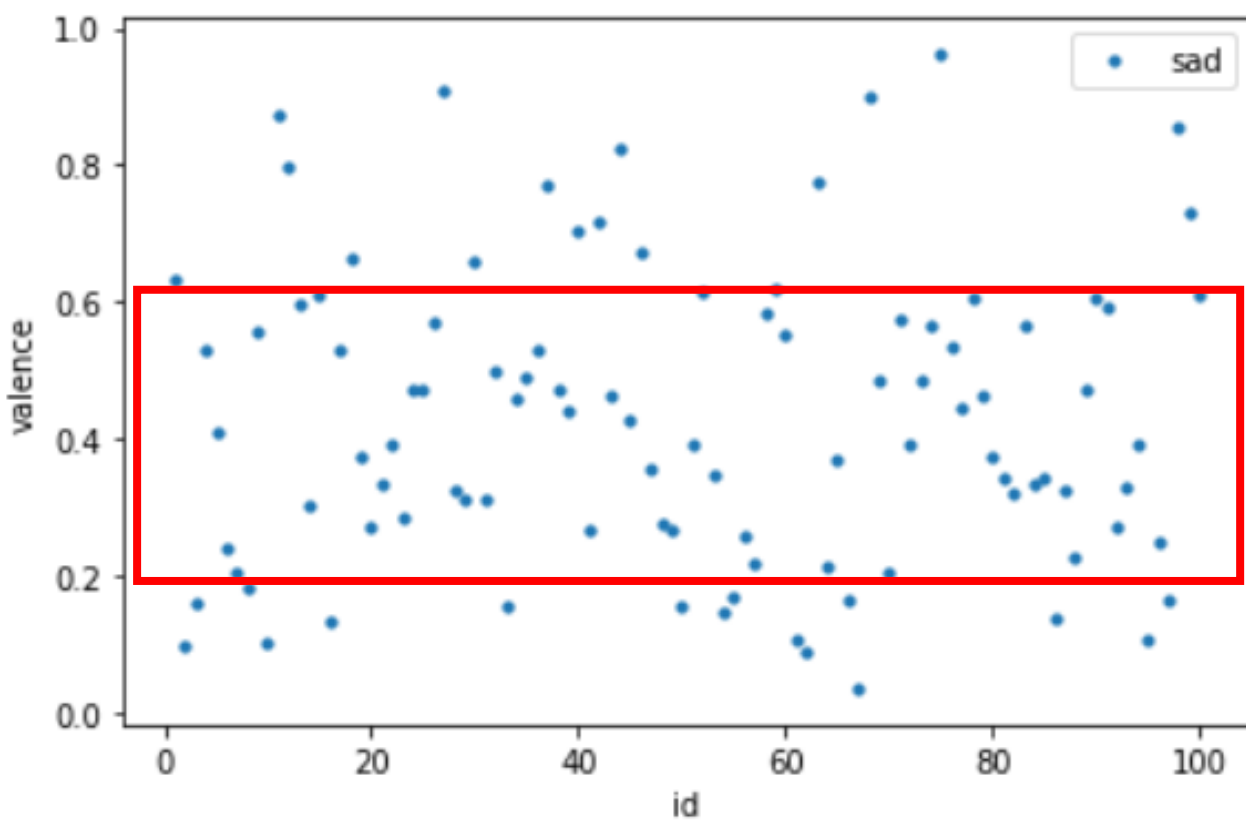# Features Analyze

relax → energy高

Features Analyze

sad、relax → energy高

Features Analyze
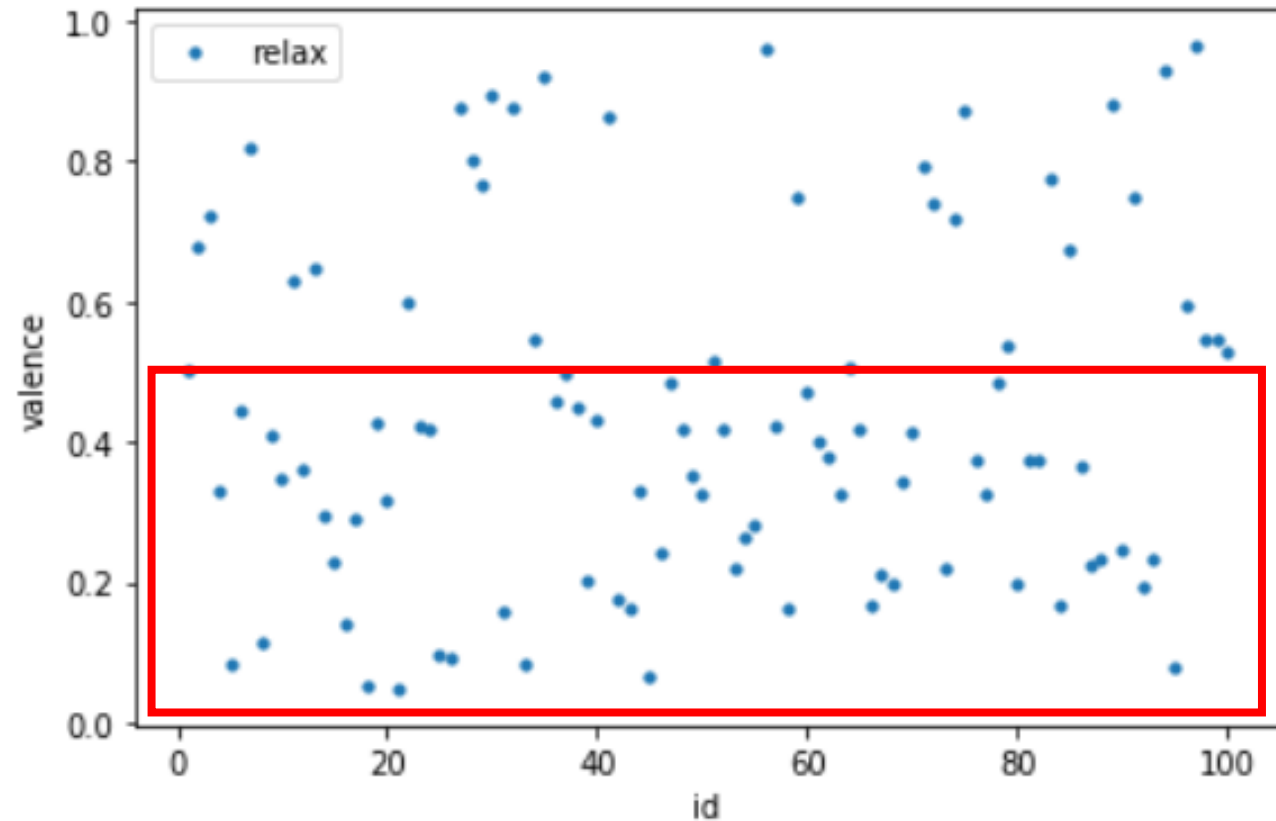

sad 、 relax → loudness大 、 energy高

Features Analyze

sad → valence (0.2~0.6)

relax → valence (0.0~0.5)

Features Analyze
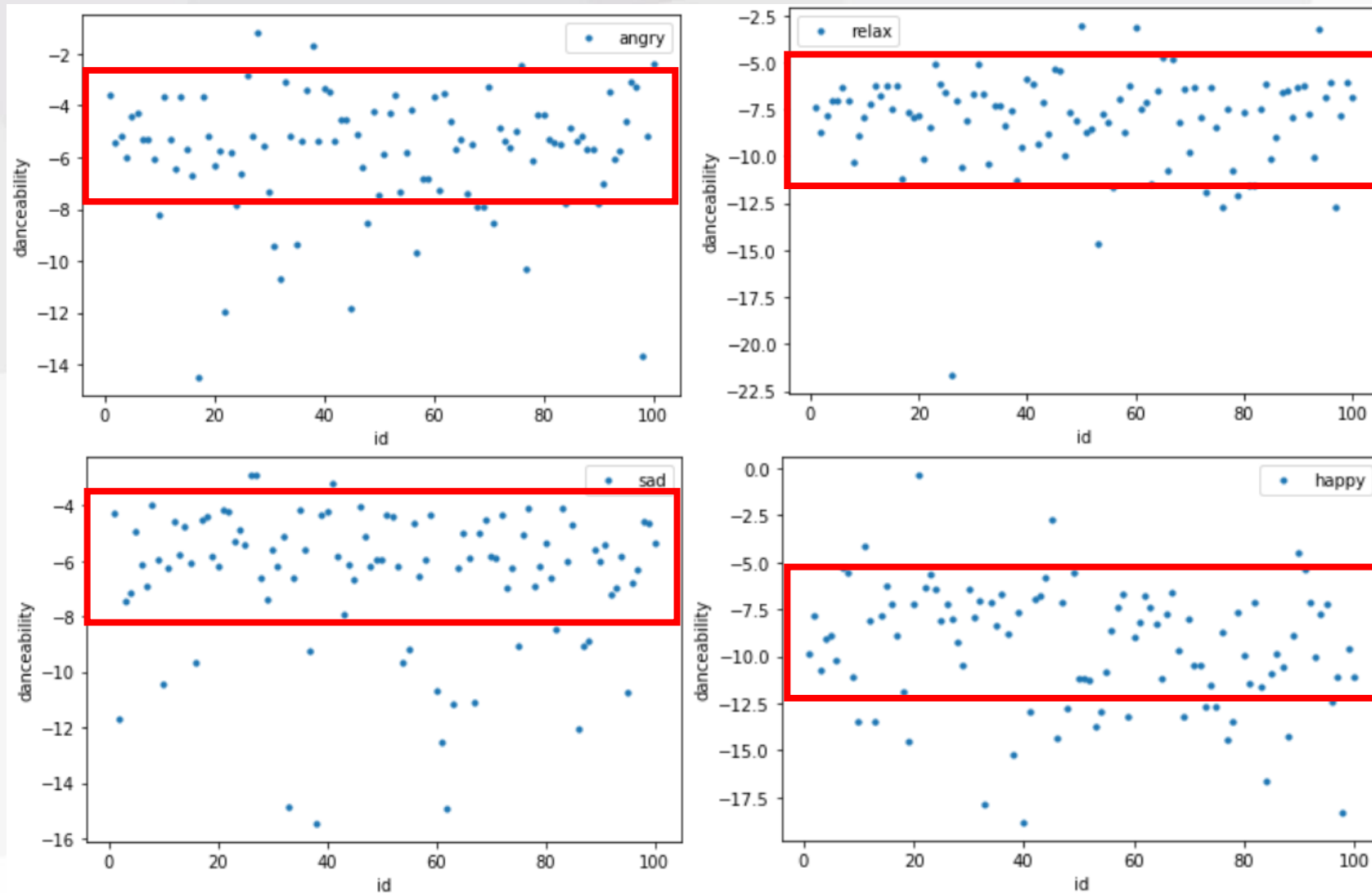
angry → danceability(-3.5~-8) /   sad → danceability(-4~-8)

relax → danceability(-5~-11)   /   happy → danceability(-5~-15)

Features Analyze
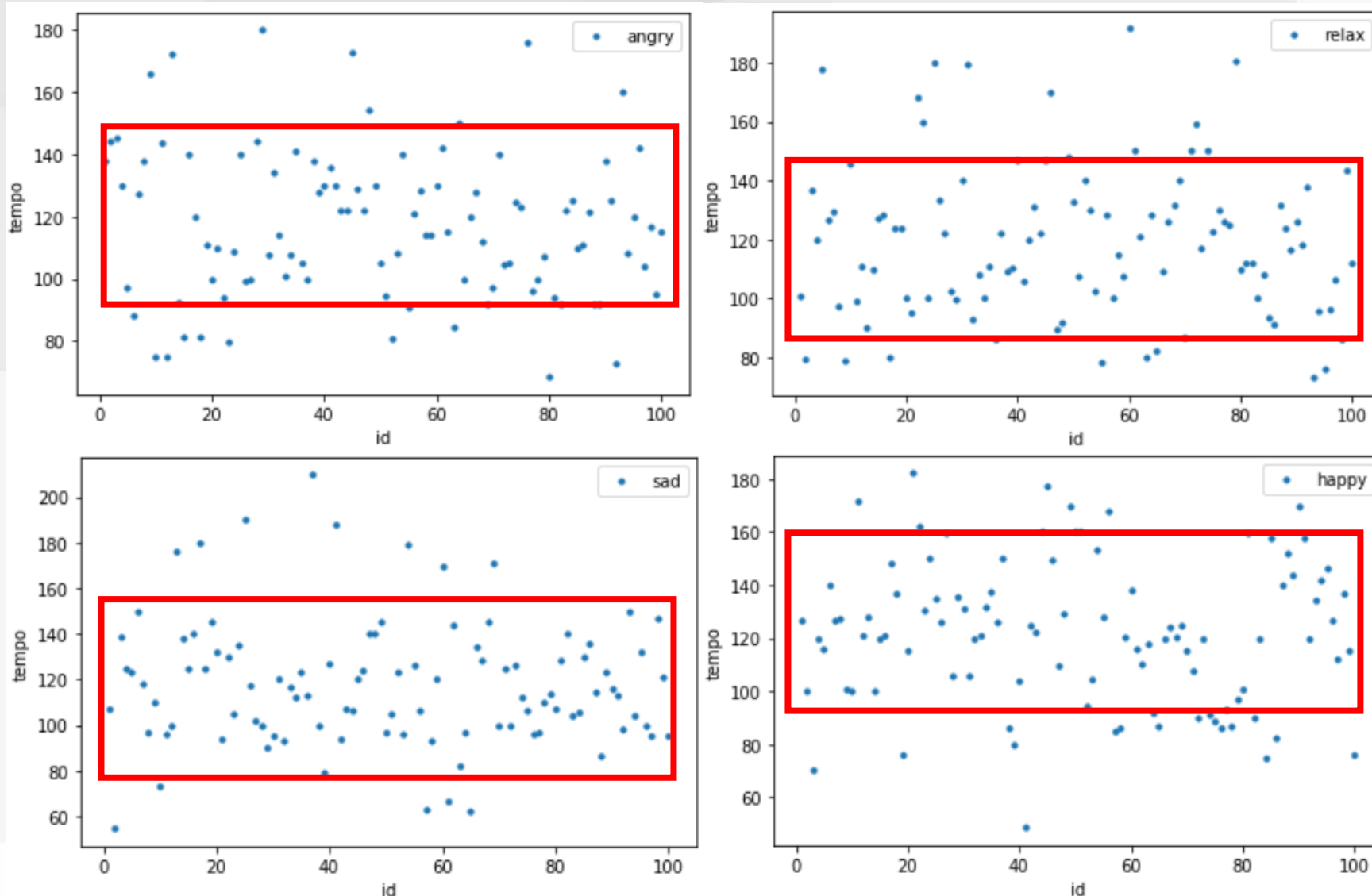
angry → tempo(90-150) / sad → tempo(80-150) / relax → tempo(90-150) / happy → tempo(100-160)

Features Analyze
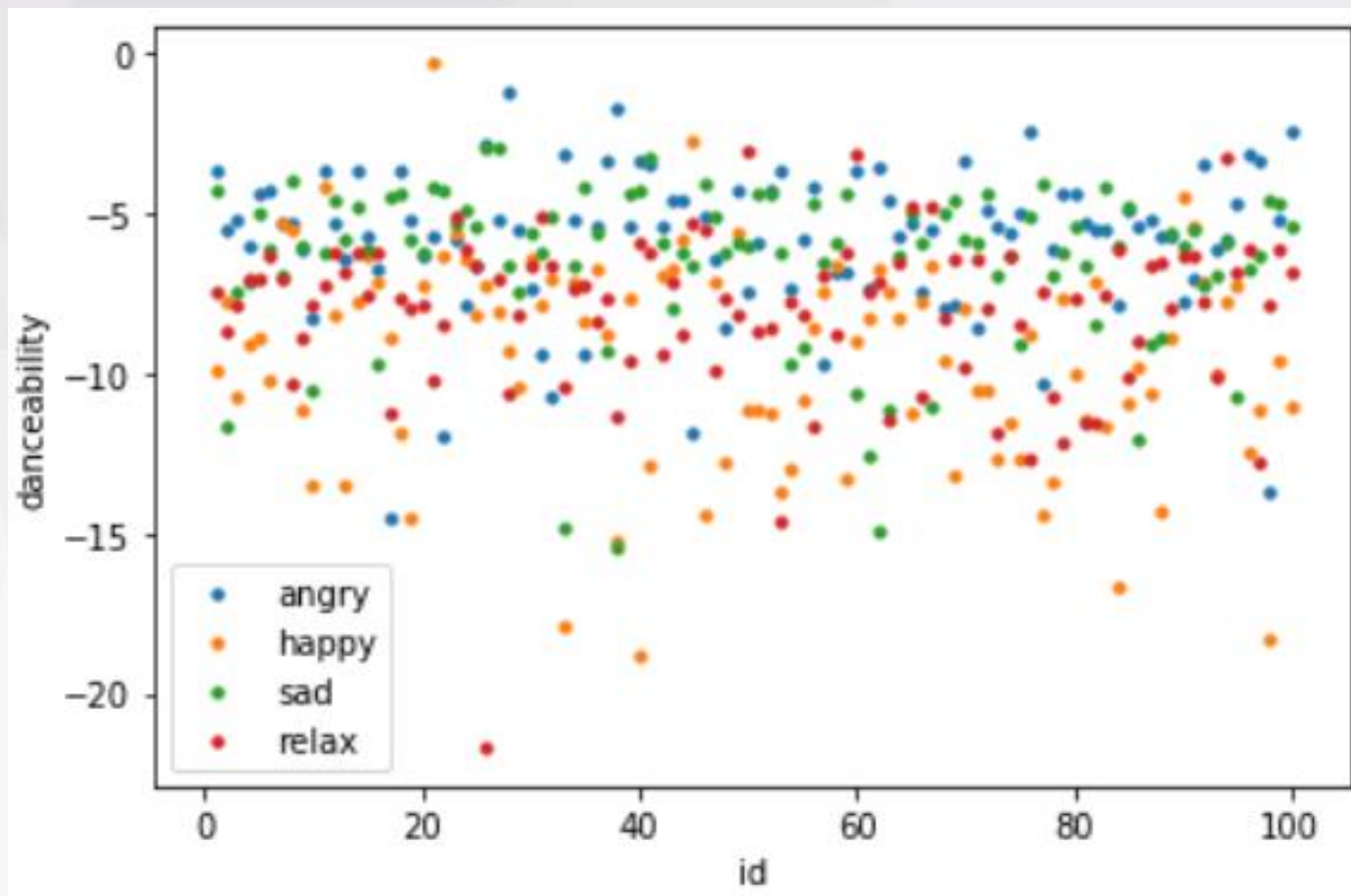
Features Analyze

happy、angry → loudness大 **(✔)**

angry → energy高、loudness大、acousticness低 **(✔)**

sad、relax → loudness大 **(✔)**、energy高

sad → valence (0.2~0.6)

relax → valence (0.0~0.5)

sad → valence低

relax → loudness大、tempo(80-140)

happy、angry → energy高、loudness大、acousticness低

angry → tempo(100-175)

Spotify Audio Features

Learn more about the audio properties of your favourite tracks, including detailed rhythmic information.

To get these values, we use the Spotify API's Get Audio Analysis for a Track endpoint.

Let's search for a track:

| And Now The Day Is Done | SUBMIT |

- Now the Day is Done - Frank Howard
- And Now The Day Is Done - Ron Sexsmith

Spotify Audio Features

Learn more about the audio properties of your favourite tracks, including detailed rhythmic information.

To get these values, we use the Spotify API's Get Audio Analysis for a Track endpoint.

Let's search for a track:

| Almost Lover | | SUBMIT |

- Almost Lover - A Fine Frenzy
- Almost Lover - Jasmine Thompson

Sections

Bars

Beats

Segments

Tatums

Spotify切割區段 - Audio Analysis Object

KEY	VALUE TYPE	VALUE DESCRIPTION
bars	an array of time interval objects	The time intervals of the bars throughout the track. A bar (or measure) is a segment of time defined as a given number of beats. Bar offsets also indicate downbeats, the first beat of the measure.
beats	an array of time interval objects	The time intervals of beats throughout the track. A beat is the basic time unit of a piece of music; for example, each tick of a metronome. Beats are typically multiples of tatums.
sections	an array of section objects	Sections are defined by large variations in rhythm or timbre, e.g. chorus, verse, bridge, guitar solo, etc. Each section contains its own descriptions of tempo, key, mode, time_signature, and loudness.
segments	an array of segment objects	Audio segments attempts to subdivide a song into many segments, with each segment containing a roughly consitent sound throughout its duration.
tatums	an array of time interval objects	A tatum represents the lowest regular pulse train that a listener intuitively infers from the timing of perceived musical events (segments). For more information about tatums.

Spotify Accounts Authentication

download node.js

`λ npm install`

authorization_code
client_credentials
implicit_grant
node_modules
.gitignore
LICENSE
package.json
package-lock.json
README.md

```
app.js   index.html

9
10   var express = require('express'); // Express web server framework
11   var request = require('request'); // "Request" library
12   var cors = require('cors');
13   var querystring = require('querystring');
14   var cookieParser = require('cookie-parser');   Using your own credentials
15
16   var client_id = '07f9611e9b234caea4fcee288da82e61'; // Your client id
17   var client_secret = '087b1a26a1294bc58a0a89d4a29463e4'; // Your secret
18   var redirect_uri = 'http://localhost:8888/callback'; // Your redirect uri
19
20   /**
21    * Generates a random string containing numbers and letters
22    * @param  {number} length The length of the string
23    * @return {string} The generated string
24    */
25   var generateRandomString = function(length) {
26     var text = '';
27     var possible = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
28
29     for (var i = 0; i < length; i++) {
30       text += possible.charAt(Math.floor(Math.random() * possible.length));
31     }
32     return text;
33   };
34
35   var stateKey = 'spotify_auth_state';
36
37   var app = express();
38
39   app.use(express.static(__dirname + '/public'))
40     .use(cors())
41     .use(cookieParser());
42
43   app.get('/login', function(req, res) {
44
45     var state = generateRandomString(16);
46     res.cookie(stateKey, state);
47
48     // your application requests authorization
49     var scope = 'user-library-read user-read-private user-read-email';
50     res.redirect('https://accounts.spotify.com/authorize?' +
51       querystring.stringify({
52         response_type: 'code',
53         client_id: client_id,
54         scope: scope,
```

Spotify Accounts Authentication

λ cd D:\中興資管所\7 實驗進度\音頻情緒分類\web-api-auth-examples-master\authorization_code

λ node app.js

current_user_saved_tracks()

print the track that the user saved

```python
# 歌單中的歌名以及歌手名稱(最愛的歌曲)
sp = spotipy.Spotify(auth=token)
results = sp.current_user_saved_tracks()
print("最愛的歌曲：")
for item in results['items']:
    track = item['track']
    print ('| - ' + track['name'] + ' | ' + track['artists'][0]['name'])
```

lyrics emotion

LogisticRegression方法

LogisticRegression方法 Testing accuracy score: 0.7584269662921348
LogisticRegression方法 Traing accuracy score: 0.9316901408450704
LogisticRegression方法 confusion matrix
[[71 17 3 2]
 [2 83 2 0]
 [11 26 48 6]
 [5 9 3 68]]

LinearSVC方法

LinearSVC方法 Testing accuracy score: 0.6769662921348315
LinearSVC方法 Traing accuracy score: 0.9316901408450704
LinearSVC方法 confusion matrix
[[71 17 3 2]
 [2 83 2 0]
 [11 26 48 6]
 [5 9 3 68]]

lyrics emotion

LogisticRegression方法

```
In [11]:  1  pred=model.predict(X_test)
          2  print("Testing accuracy score: " + str(model.score(X_test, y_test)))

Testing accuracy score: 0.3281786941580756
```
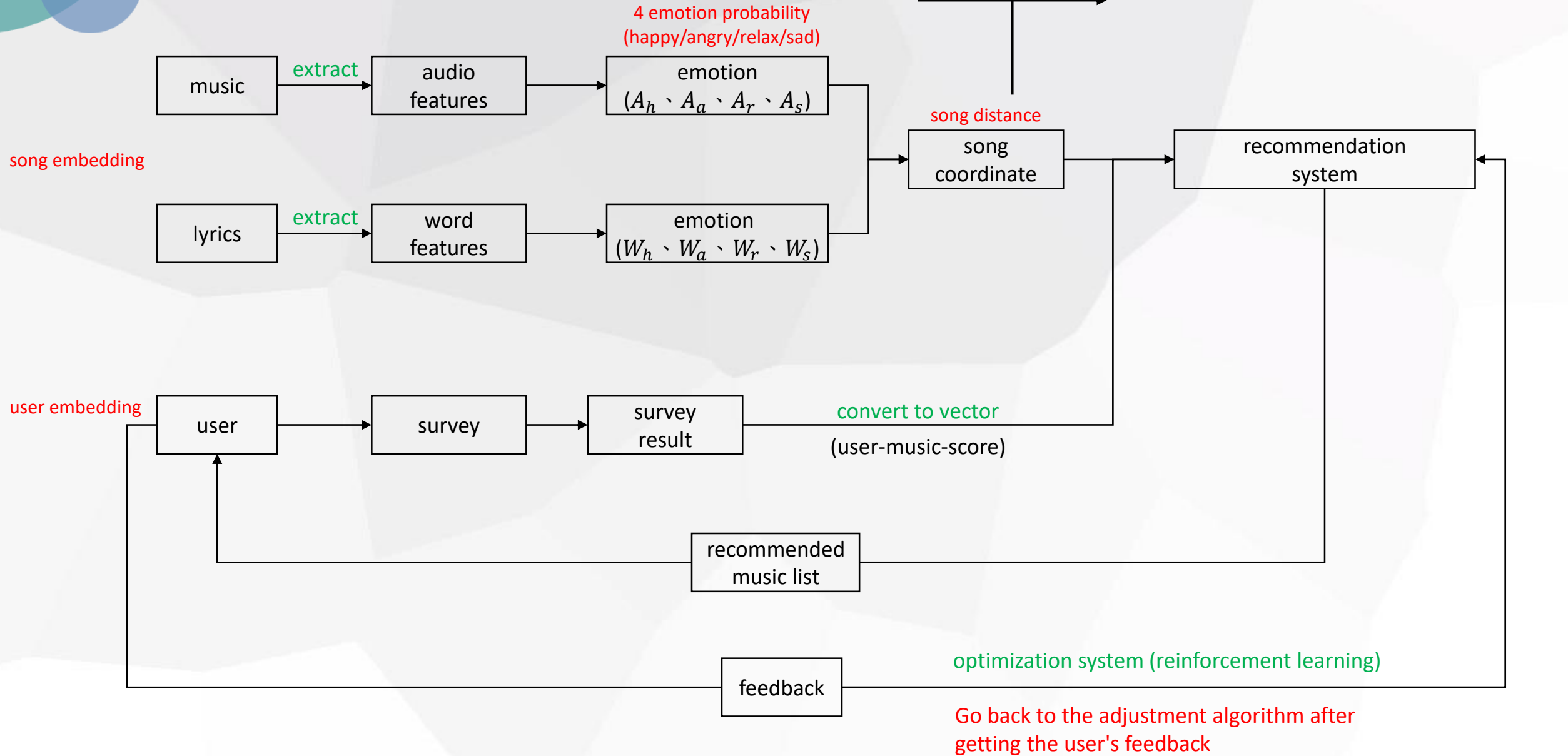
LinearSVC方法

```
In [12]:  1  #LinearSVC方法
          2  linear_svc = LinearSVC(C=1.0, penalty='l1', max_iter=3000, dual=False)
          3  model2 = linear_svc.fit(X_train,y_train)
          4  pred = model2.predict(X_test)
          5  print("LinearSVC Testing accuracy score: " + str(model2.score(X_test, y_test)))

LinearSVC Testing accuracy score: 0.32989690721649484
```

lyrics0503(1-2000)_d.csv

Framework

● $[A_h, A_a, A_r, A_s, W_h, W_a, W_r, W_s, \ldots other\ features\ldots]$

feature selection

4 emotion probability
(happy/angry/relax/sad)

| music | --extract--> | audio features | --> | emotion $(A_h \cdot A_a \cdot A_r \cdot A_s)$ |

song distance

| song coordinate | --> | recommendation system |

| lyrics | --extract--> | word features | --> | emotion $(W_h \cdot W_a \cdot W_r \cdot W_s)$ |

song embedding

user embedding

| user | --> | survey | --> | survey result |

convert to vector

(user-music-score)

| recommended music list |

| feedback |

optimization system (reinforcement learning)

Go back to the adjustment algorithm after getting the user's feedback

Survey Page

Enter your information

Username

Gender ☐ **Male** ☐ **Female**

Age

NEXT

Survey Page

Score the following songs according to your preferences

play for 30s

Music 1

Music 2

Music 3

Music 4

Music 5

Music 6

Music 7

Music 8

1 ⭐   5 ⭐

O K

Preferences List

⭐ 5   Music 5

⭐ 4   Music 3、Music 6、Music 8

⭐ 3   Music 2、Music 7

⭐ 2   Music 4

⭐ 1   Music 1

data : user-music-score

THANKS!

chiouchingyi@smail.nchu.edu.tw