Deep Learning for Music Analysis and Generation

HW1

Singer classification

Yi-Hsuan Yang Ph.D.
yhyangtw@ntu.edu.tw

The Homework

- **Singer classification**
  - which is, well, related to timbre
- Submit your **codes**, **report**, and **prediction result** for the test set
- Done individually
- No cheating
- Evaluation
  - Report (50%), accuracy (50%)
  - Choose the **best three** for a quick oral presentation (5 mins each) during the class (bonus points offered)

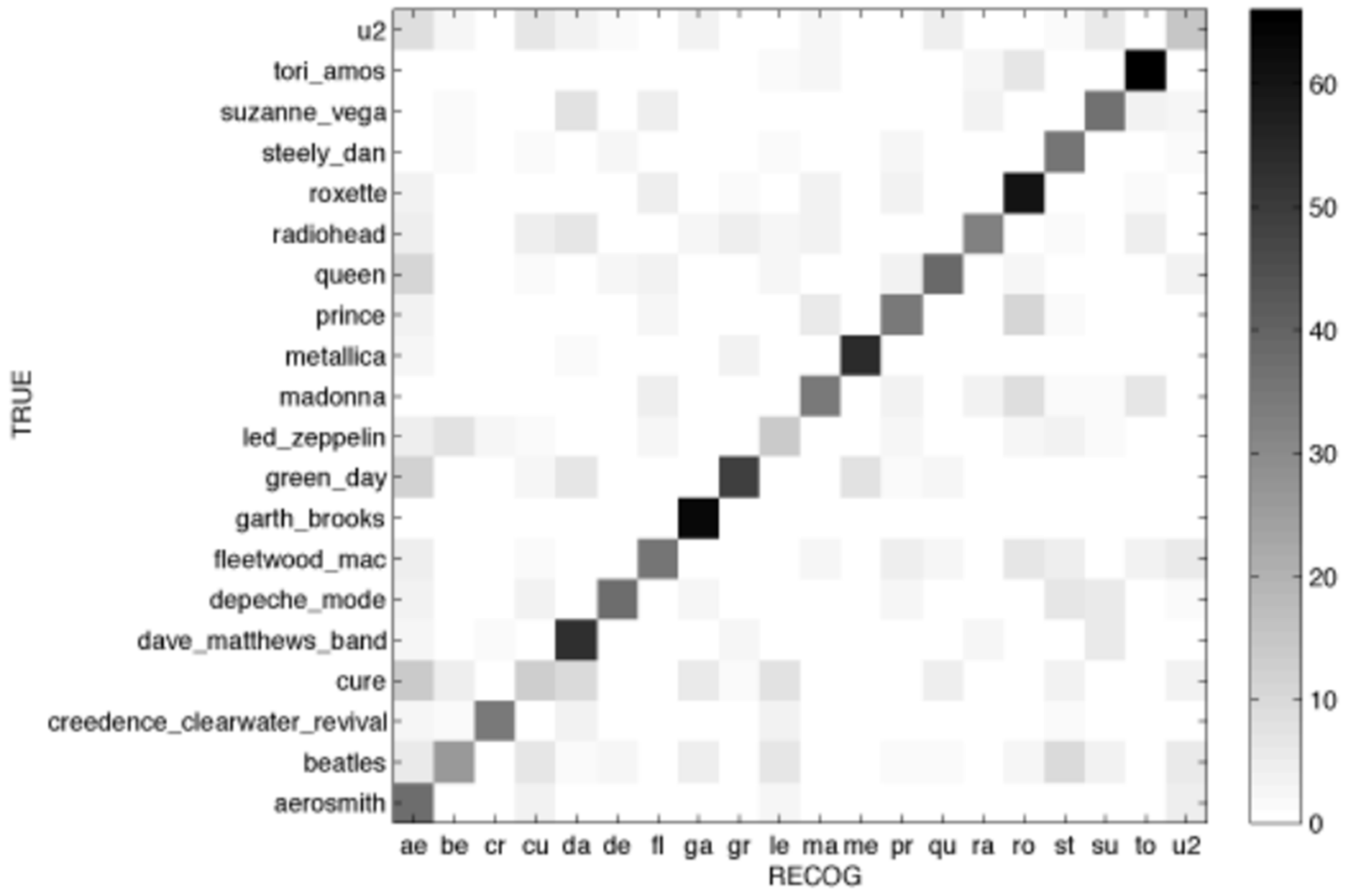
Timeline

- **W3 – 9/21** (Thursday): Announcement of HW1 & Release of training set
- **W4 – 9/30** (Saturday): Release of test set
- **W5 – 10/4** (Wednesday **1:00pm**)**: First deadline**
    - Optional; will announce the test result so that you can further improve your model
- **W6 – 10/11** (Wednesday **23:59pm**): **Final Deadline**
    - Late submission: 1 day (-20%), 2 days (-40%), after that (-60%)

- **W6 – 10/12** (Thursday): Announcement of HW2
- **W7 – 10/19** (Thursday): Oral presentation of HW1
    - I will be in touch with the three presenters two days ahead (i.e., on 10/17)
    - No need to prepare fancy slides

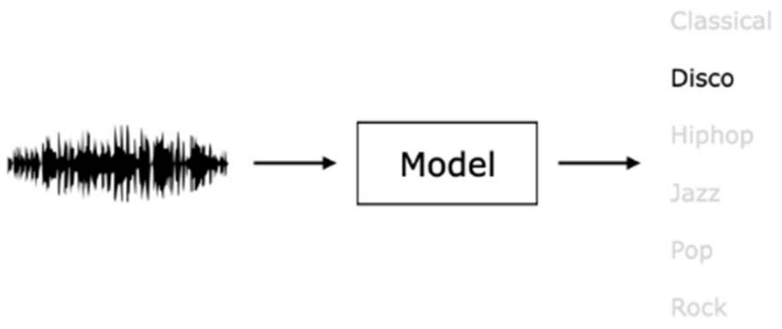
Dataset: artist20

http://labrosa.ee.columbia.edu/projects/artistid/

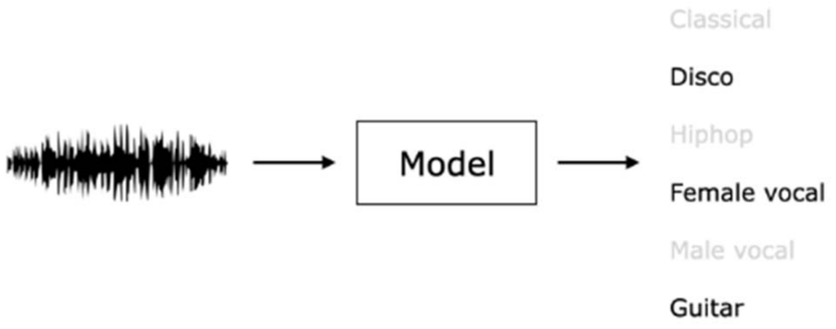


The Task: Single-label Classification

- An audio clip is associated with only one class



An example of single-label classification



An example of multi-label classification

Figures source: https://music-classification.github.io/tutorial/part1_intro/what-is-music-classification.html

Train/Validation/Test split

- Predefined by the Tas
  - Use **training** data for training, **validation** data for parameter tuning, and the **test** data for evaluation
  - **NOTE**: don’t use test data for model training and hyperparameter tuning
  - Training & validation sets: released on Day 1
  - Test set: released the next weekend
- **Album-level split**
  - The clips from the same **album** would be in the same split
  - To avoid confounds due to the “album effect”
    - “**song-split may leak production details associated with an album over the training and testing subsets**, giving an SID model additional clues for classification. Accordingly, the accuracy for song-split may be overly optimistic and tends to be higher than that of album-split”

Ref: Hsieh et al, “Addressing the confounds of accompaniments in singer identification,” ICASSP 2020

Segmenting Your Songs into Shorter Segments

- Segmentation is needed because your models won’t be able to take full-length songs as input
  - We likely train models for short-time segments (e.g., 5 seconds with or without some overlaps), and make predictions segment-by-segment
  - Song-level prediction result can be obtained by, for example, majority voting of the segment-level prediction results

- **Song-level split**: Segments from the same **song** should be in the *same* split
  - Which is guaranteed in our case because we use an even stricter “album-level split”
- Segment length: up to you

Approach 1

- Consider it as a general music classification problem
- https://github.com/minzwon/sota-music-tagging-models

Approach 2

- Resort to existing work on singer recognition
  - 19[ijcnn] Musical artist classification with convolutional recurrent neural networks https://github.com/ZainNasrullah/music-artist-classification-crnn
  - 20[icassp] Addressing the confounds of accompaniments in singer identification https://github.com/bill317996/Singer-identification-in-artist20
  - 21[aaai] Positions, channels, and layers fully generalized non-local network for singer https://github.com/ian-k-1217/Fully-Generalized-Non-Local-Network
  - 22[ijcnn] Singer identification for metaverse with timbral and middle-level perceptual features
  - 23[ismir] Singer identity representation learning using self-supervised techniques

Approach 3

- Seek help from the neighboring field of speaker recognition and verification
  - i-vector, x-vector
  - https://www.isca-speech.org/archive/interspeech_2020/chowdhury20b_interspeech.html
  - https://github.com/TaoRuijie/ECAPA-TDNN

Approach 4

- Use pre-trained models
  - 23[apsipa] Toward leveraging pre-trained self-supervised frontends for automatic singing voice understanding tasks- three case studies https://arxiv.org/abs/2306.12714
  - 23[ismir] On the effectiveness of speech self-supervised learning for music https://arxiv.org/abs/2307.05161

Data Pre-processing / Data Augmentation

- **Source separation** (*recommended*; more on this next week)
  - open-unmix
  - spleeter
  - demucs

- Singing voice detection
  https://github.com/keums/melodyExtraction_JDC

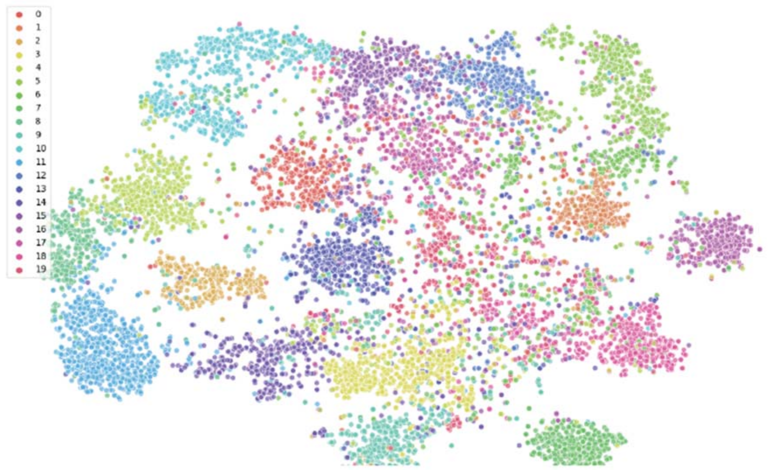
- Data augmentation for robustness to recording noises
  https://github.com/sevagh/audio-degradation-toolbox

Evaluation

- Report the
  - Top-1 accuracy
  - Top-3 accuracy
  - Confusion matrix

Bonus

- Record **your own** singing voice and use that as input to the classifier you built; discuss the result
- Use a recording of **a singer you like** as input to the classifier; discuss the result
- Use t-SNE to visualize the learned the embeddings; use colors to mark different singers



Ref: Hsieh et al., "Addressing the confounds of accompaniments in singer identification," ICASSP 2020

The Report

- Format: **slides**
  - Cover page: your name, student ID etc
  - **Novelty highlight** (one page; *optional*): what's special about your work?
  - **Methodology highlight** (one page): how did you make it?
    - Or, list the *attempts* you have made
  - **Result highlight** (one page): result on your validation set
  - **Findings highlight** (one page): main takeaways of your study
  - **Details of your approach** (multi-pages)
    - If you use open source code, you may want to read some of the associated paper(s) and summarize your understanding of the paper(s) (e.g., why it works)
  - **Result analysis & discussion** (multi-pages)

The Report: Result analysis & Discussion

- Discuss the result
  - Exemplar topics:
    - Does the model overfit?
    - Does the confusion table looks right?
    - Can the model distinguish between male and female singers?
    - Can the model distinguish between singers with different vocal range?
- Document your findings

Rules about HW Submission

- By the TAs

HW1 scoring and submission

HW1 TA: 葉軒瑜(Fischer Yeh)

Office hour: Thu.6 13:20 ~ 14:05 @BL505

When you encounter problem:

1. Check out all course materials and announcement documents

2. Use the power of the internet and AI

3. Use “Discussions” on NTU COOL

4. Email me fish90510@gmail.com or come to office hour

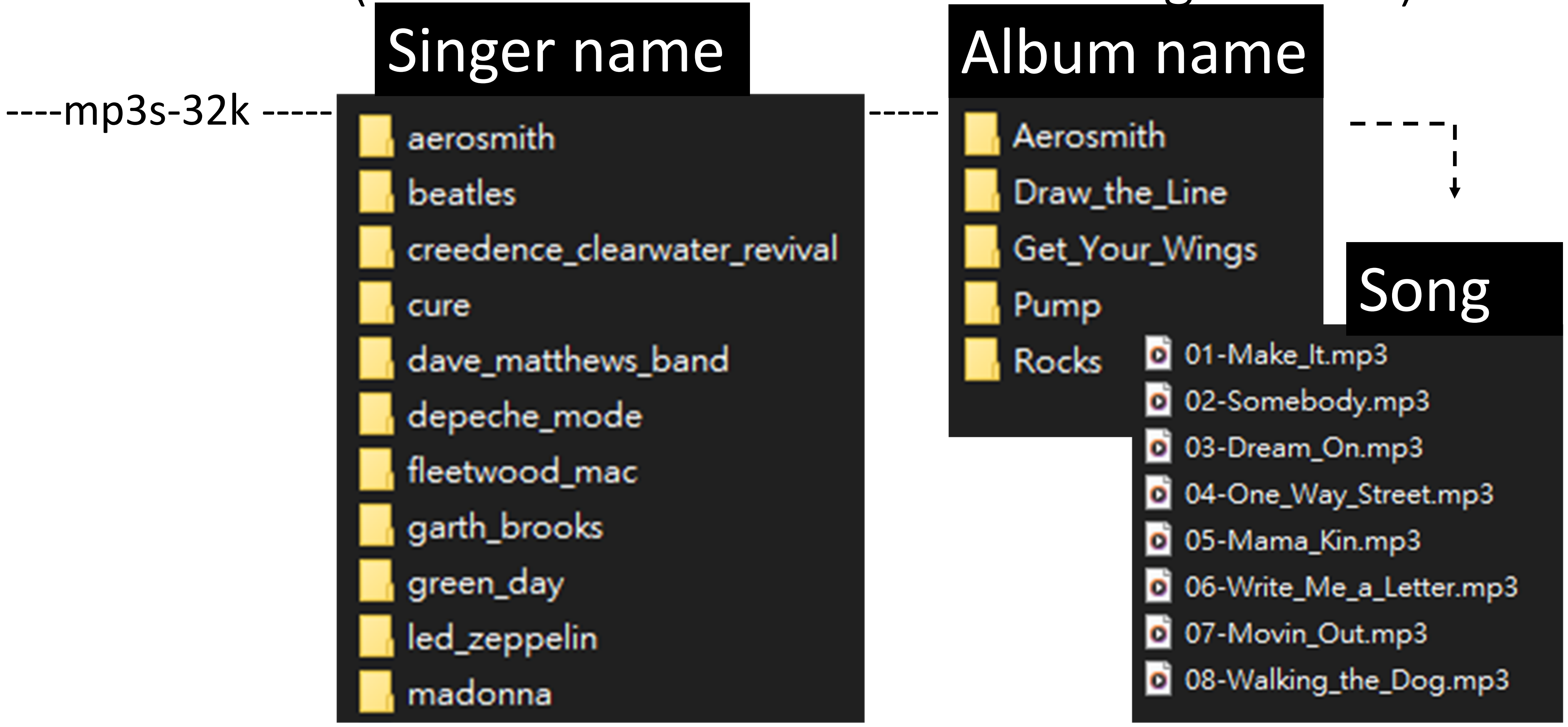
Rules

- Don’t cheat

- Use DL methods

- Can use public codes with cite in report
- Can use pretrained model
- Can add extra data except artist20

Artist20 (data release on NTU Cool assignments)



----mp3s-32k -----

Singer name

aerosmith
beatles
creedence_clearwater_revival
cure
dave_matthews_band
depeche_mode
fleetwood_mac
garth_brooks
green_day
led_zeppelin
madonna

Album name

Aerosmith
Draw_the_Line
Get_Your_Wings
Pump
Rocks

Song

01-Make_It.mp3
02-Somebody.mp3
03-Dream_On.mp3
04-One_Way_Street.mp3
05-Mama_Kin.mp3
06-Write_Me_a_Letter.mp3
07-Movin_Out.mp3
08-Walking_the_Dog.mp3

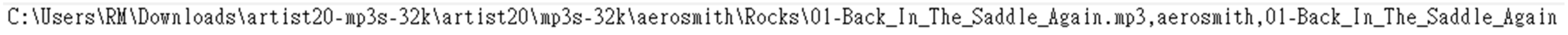
Artist20

Provided	Testing Released on September 30th.
• Sample rate: 16000Hz • Ext: mp3 • Channel: Mono • Length: full song	• Sample rate: 16000Hz • Ext: mp3 and wav • Channel: Mono • Length: full song, 30s, 60s, 90s

Data Split

- Within each artist, select the album title that comes last in alphabetical order as the validation data.
- Sample code provided (you can adjust the sample code to match your desired output format)

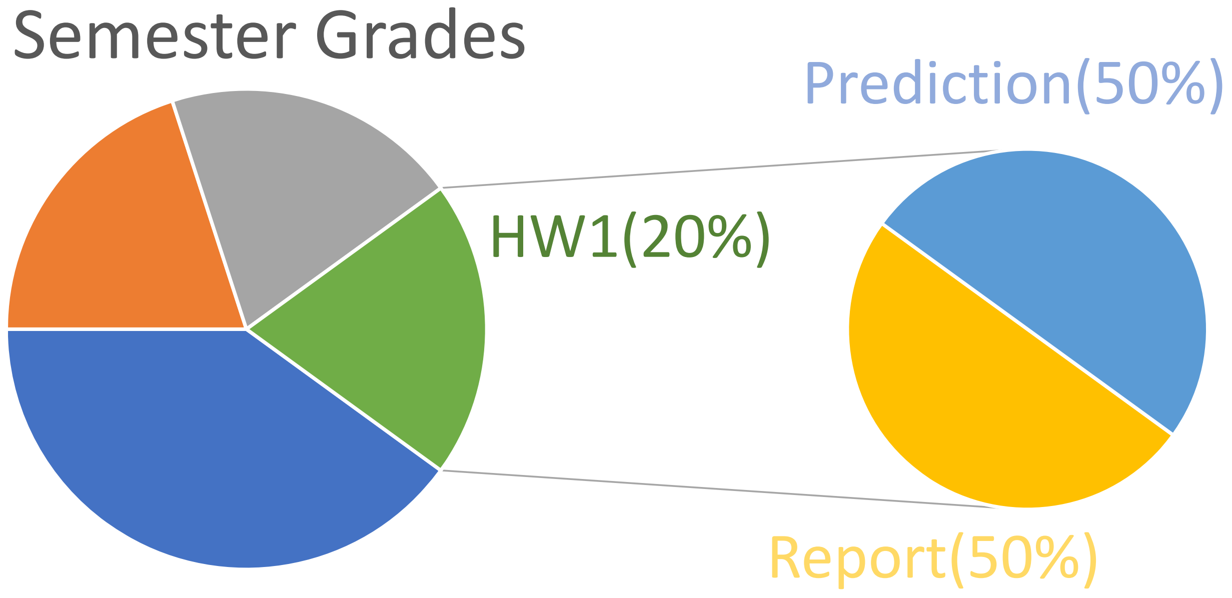
- Output: train.txt & validation.txt
- Audio_path,singer_name,song_title



```
C:\Users\RM\Downloads\artist20-mp3s-32k\artist20\mp3s-32k\aerosmith\Rocks\01-Back_In_The_Saddle_Again.mp3,aerosmith,01-Back_In_The_Saddle_Again
```

Scoring

- HW1 accounts for <u>20%</u> of the total grade.
- In HW1, Report (50%), prediction accuracy (50%)



Report

- Write with PPT or PPT-like format (16:9)
- Upload studentID_report.pdf (ex: r12345678_report.pdf)

- Please create a report that is clear and can be understood without the need for oral explanations.
  There is no specific length requirement, but it should clearly communicate the experiments conducted and their results.
  Approximately 10 pages is a suggested standard, but not a strict limitation.

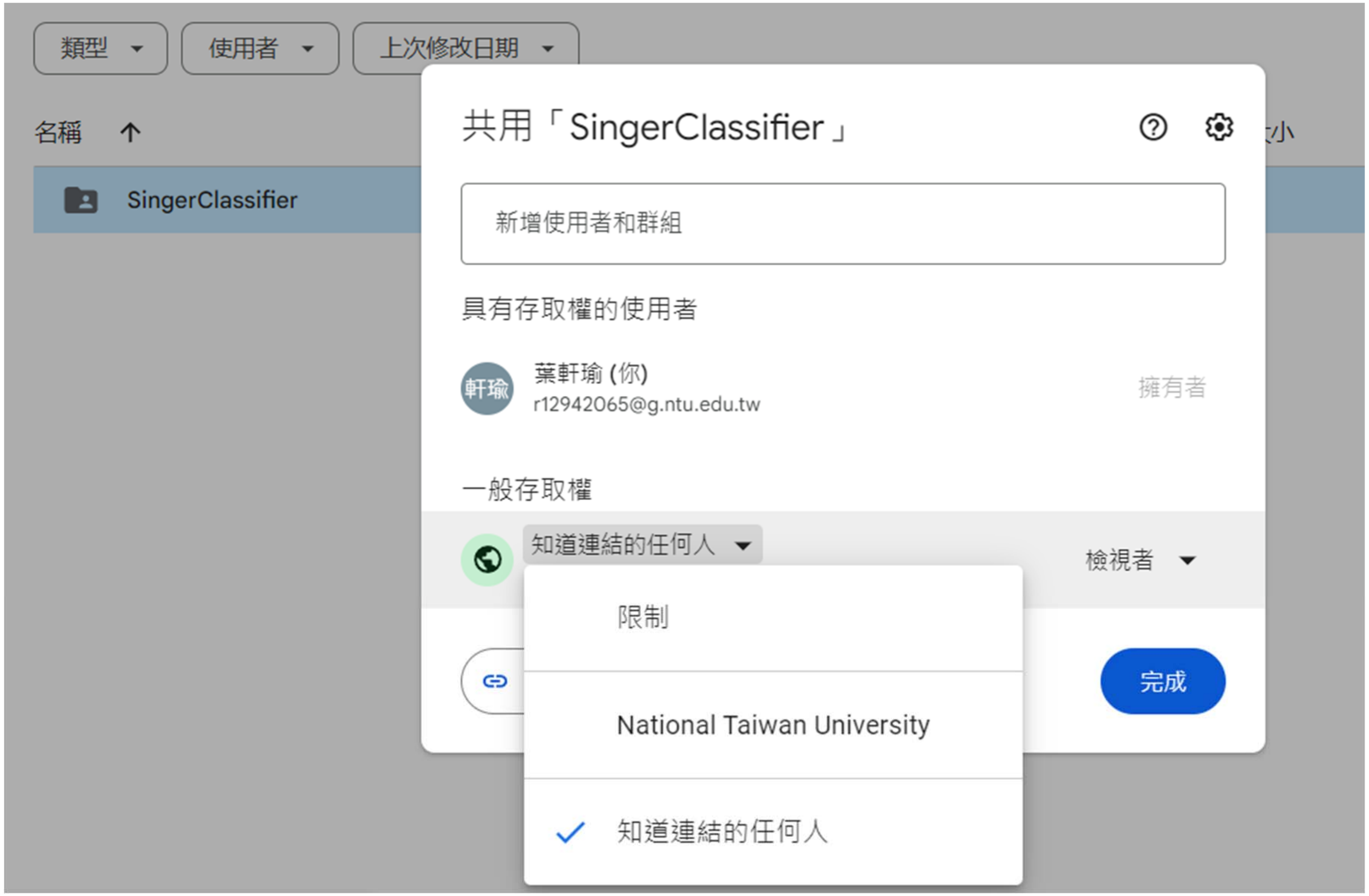
Code

- Upload all your source code to a cloud drive, open access permissions, and then upload the link to the NTU Cool assignment HW1_report in comments, as well as include it on the first page of the report.

- We will randomly select several classmates' code to run inference on your model, so please ensure that the files you upload include trained model and can successfully execute the entire inference process.

- Don’t upload: training data, testing data, preprocessed data, others model, cache file

Code



類型 使用者 上次修改日期

名稱 ↑

SingerClassifier

共用「SingerClassifier」

新增使用者和群組

具有存取權的使用者

葉軒瑜 (你)
r12942065@g.ntu.edu.tw
擁有者

一般存取權

知道連結的任何人 ▾　檢視者 ▾

限制

National Taiwan University

✓ 知道連結的任何人

完成

Code

- You will need to upload requirements.txt

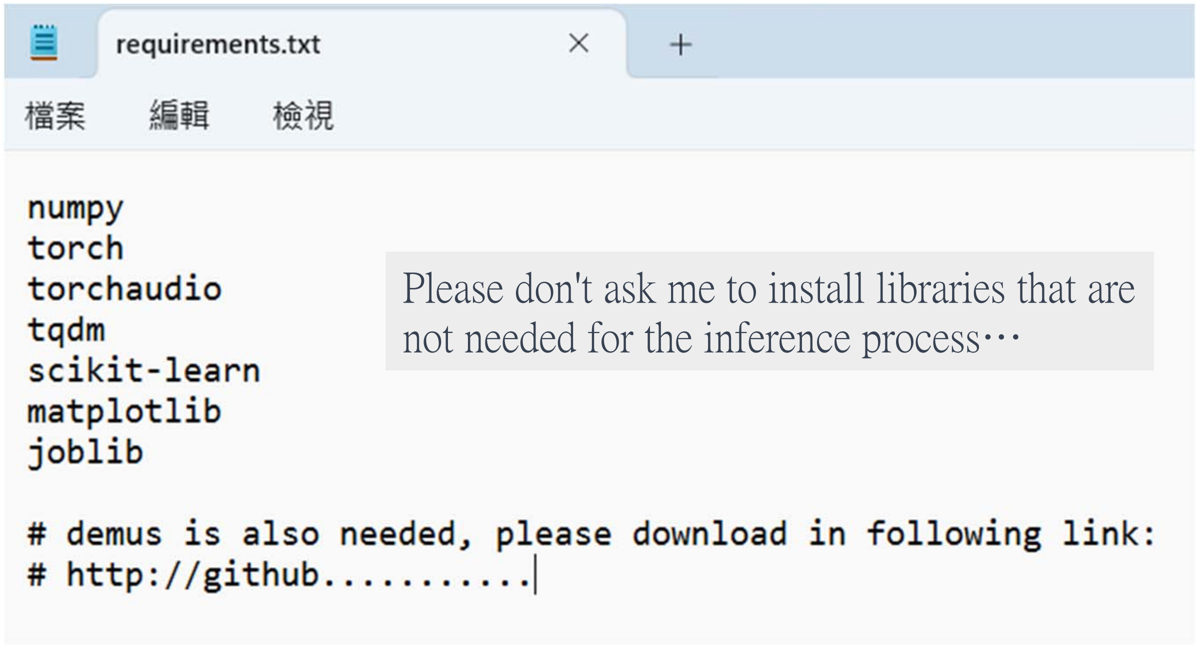
I will run:

```
pip install -r requirements.txt
```

to install all library you need to run inference.

If you have used third-party programs that cannot be installed directly via 'pip install,' please use '#' to write the installation URL and method in a text file.

Example



```
numpy
torch
torchaudio
tqdm
scikit-learn
matplotlib
joblib

# demus is also needed, please download in following link:
# http://github...........
```

Please don't ask me to install libraries that are not needed for the inference process…

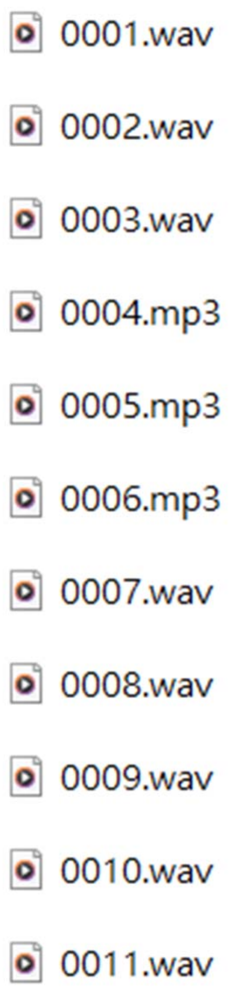
Code

- You will also need to upload README.txt or README.pdf to guide me on how to perform inference on your model. I will use data in the same format as the test data, and the results must match exactly with the predictions you uploaded.

- Your inference process should allow me to choose the test data folder path as well as specify the filename and path for the output prediction.csv.

Prediction

- There are 970 testing audios.
- File name: ID.mp3 or ID.wav
- The ID numbers range from 0001 to 1000, but there may be missing number in between.
- Example_submission.csv



0001.wav
0002.wav
0003.wav
0004.mp3
0005.mp3
0006.mp3
0007.wav
0008.wav
0009.wav
0010.wav
0011.wav

Prediction

- Format:

ID,No.1,No.2,No.3

ID,No.1,No.2,No.3

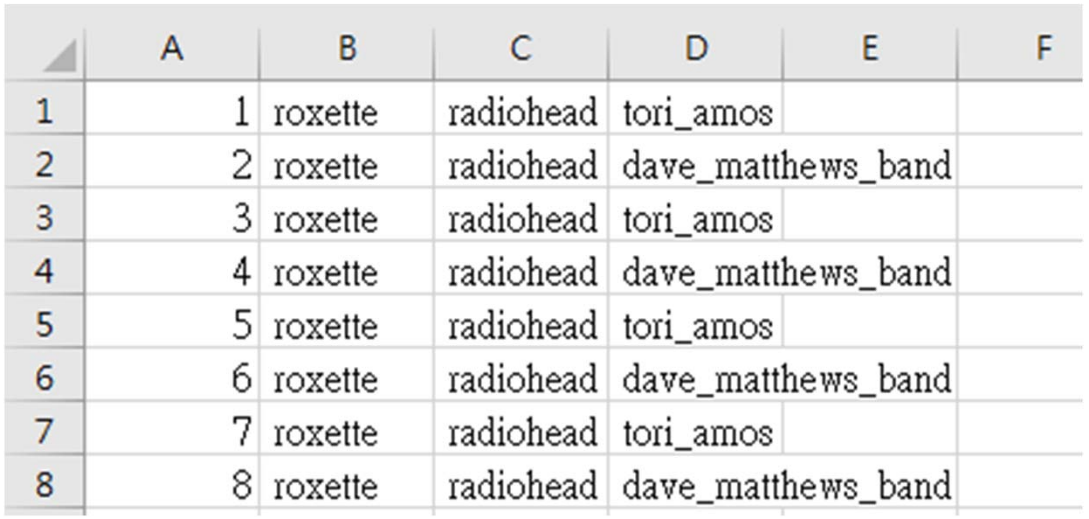
…

(Please sort the IDs in ascending order.)

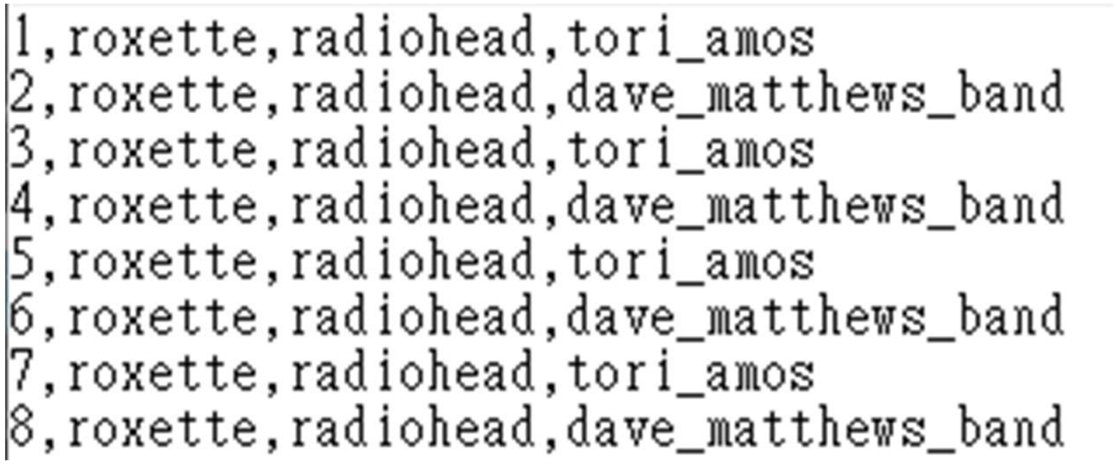
- File name and type

studentID.csv

ex: r12345678.csv



	A	B	C	D	E	F
1	1	roxette	radiohead	tori_amos		
2	2	roxette	radiohead	dave_matthews_band		
3	3	roxette	radiohead	tori_amos		
4	4	roxette	radiohead	dave_matthews_band		
5	5	roxette	radiohead	tori_amos		
6	6	roxette	radiohead	dave_matthews_band		
7	7	roxette	radiohead	tori_amos		
8	8	roxette	radiohead	dave_matthews_band		



```
1,roxette,radiohead,tori_amos
2,roxette,radiohead,dave_matthews_band
3,roxette,radiohead,tori_amos
4,roxette,radiohead,dave_matthews_band
5,roxette,radiohead,tori_amos
6,roxette,radiohead,dave_matthews_band
7,roxette,radiohead,tori_amos
8,roxette,radiohead,dave_matthews_band
```

Prediction score

- 10/4 13:00

  First test prediction deadline, the result will announce before 10/4 21:00 as possible.
- 10/11 23:59

  Homework deadline.

Rank with your best score

Score = Top1 acc + 0.5 * Top3 acc

Please follow the csv format or my code will show no mercy to you!

Coding advise

- Avoid excessive use of global variables. Use

```
if __name__ == ‘__main__‘:
   main()
```

- Make comments in your program
- Allow your program to dynamically adjust multiple parameters based on user requirements.
- Define variables with meaningful names.
- Break your program multiple functional blocks.

(functions, files, cells or even separate with spaces)

Useful library

- Torchaudio: including loading audio, Resample, MelSpectrogram function

https://pytorch.org/audio/stable/index.html

- Sklearn: LabelEncoder, evaluation matrix

https://scikit-learn.org/stable/modules/classes.html

- Tqdm: Loop progress bar

https://github.com/tqdm/tqdm

- Librosa: audio processing, MelSpectrogram

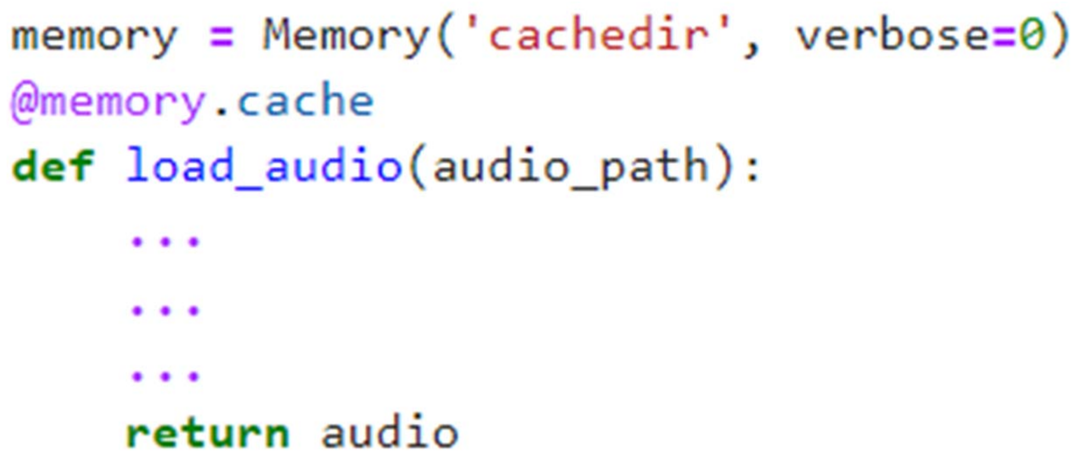
https://librosa.org/doc/latest/index.html

Useful library

- Joblib:
- Use disk to save cache to save execution time

(It may consume a significant amount of disk space, so use it according to your own circumstances.)

- Example usage:

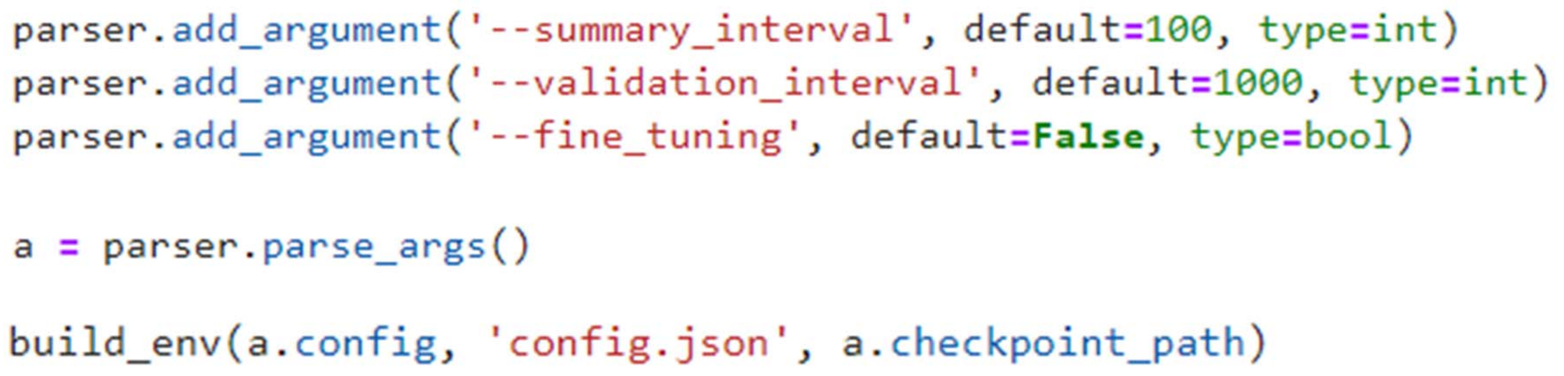


```python
memory = Memory('cachedir', verbose=0)
@memory.cache
def load_audio(audio_path):
    ...
    ...
    ...
    return audio
```

Useful library

- argparse:
- Making it easier and more structured to develop command-line tools

- Example usage:



```
parser.add_argument('--summary_interval', default=100, type=int)
parser.add_argument('--validation_interval', default=1000, type=int)
parser.add_argument('--fine_tuning', default=False, type=bool)

a = parser.parse_args()

build_env(a.config, 'config.json', a.checkpoint_path)
```

Code Ref: Hifi-Gan https://github.com/jik876/hifi-gan

ALL things you need to do before 10/11 23:59

- HW1_report
  - StudentID_report.pdf
  - Cloud drive link
    - README.txt or README.pdf
    - Requirements.txt
    - Codes and model to run inference
    - Others codes
- HW1_prediction
  - StudentID.csv