Interview Process
Task for Data Engineer role

Task

Build a basic audio data processing pipeline that does the following:

- Load audio files from a dataset directory
- Extract audio features from each audio file
- Normalize the features
- Save features as numpy arrays to a file
- + describe how to save features in a database or data lake

Input Dataset

Use the GTZAN dataset. Download the dataset here. If you have problems downloading the dataset from the previous link, please use this link.

Assume the dataset with audio files organized like this:

```
/dataset
        /genre1
                file1.wav
        /genre2
                file2.wav
```

Output

Save the features as numpy arrays to .npy files (details below).

Structure the output the following way:

- in the output folder, replicate the folder structure of the dataset folder.

- for the filenames use: “originalfilename_” + feature type + “_” + “normalization strategy” + “.npy”

e.g.

/output_folder
    /genre1
        file1_mfccs_minmaxnormalizer.npy
        file1_mfccs_standardcaler.npy
        file1_feature-type2_minmaxnormalizer.npy
        file1_feature-type2_standardscaler.npy
        file2_mfccs_minmaxnormalizer.npy
        file2_feature-type2_minmaxnormalizer.npy
        ...
    /genre2
        file3_mfccs_minmaxnormalizer.npy
        ...

Features

The pipeline should be able to extract **2 different types of audio features using a feature extractor**.

Choose the type of library to extract audio features and the features you want to extract.

These are only loose example suggestions:

- E.g. Librosa library
- Please decide on the concrete features to extract (e.g. Mel-Spectrograms or MFCCs)
- Also decide whether you extract the features for the entire audio file, or segment-wise with subsequent aggregation.

Normalization

Implement 2 types of normalization per feature attribute:

- min max normalization
- standard scaling

Again you may use the library of your choice for this, or implement it yourself.

Feature Storage

A) To Files

Use the numpy library to save the normalized features to .npy files.
1 output file per input file and feature type:
filename pattern: “originalfilename” + “_” + “feature type” + “_” + “normalization strategy” + “.npy”

For folder organization see above.

B) To a Database or Datalake

Here we don’t ask you for the implementation.

Instead, please describe what type of database, or data lake, you would use to store these features, how you would store the data and what design and architecture considerations you have to make this database / data lake scalable.
If you have experience with cloud platforms (AWS, GCP) please mention considerations that you may have regarding available cloud technologies.

Include this in the Summary Report we mention below.

Configuration

The extractor and normalization parameters as well as input and output paths should be configurable (e.g., window size, hop length, num frequency bands, num mfccs, ...).

The yaml config file should provide details regarding two aspects of the pipeline (see Entry Point section below):

- Which feature extractor / normalizer to use
- Values to use for the parameters of each component of the pipeline (e.g., save folder for data saver, dataset folder for data loading, number of bands (Mels) for (Mel-)Spectrogram extractor)

Implementation details

- Use Python 3
- Use 3rd party libraries you see fit to carry out the different tasks
- Implement the pipeline using Object Oriented Programming
- **Write clean code**
- **Write unit tests**
- Have an entry point script that can trigger the pipeline from bash
- Pass a configuration yaml file to the script to set up the pipeline
    - e.g.: `$ extractor config.yml`
- Package the pipeline and make it installable via a setup.py script

Summary Report

Please write a 1 to 2 page summary of notes including the following:

- comment your choice of algorithms, libraries and implementation design / code organization
- describe your choice of database for feature storage that you would envision
- mention any obstacles, difficulties or further suggestions you had/have regarding this task

Include the report in the submitted repository as either .md, .txt, or .pdf file named “summary_report.*”.

Submission

- Store code and summary report on a private GitHub repo and provide the link to us
- If this is not possible, send us a zip file with the content including the summary report

Task deadline

1 week after you received the task
(please let us know if you need more)

Contact

If you have any doubt or questions regarding this task, please contact: Valerio Velardo <valerio.velardo@utopiamusic.com>