Lab 3 – MATH 240 – Computational Statistics

Avery Johnson
Colgate University
Department of Mathematics
aqjohnson@colgate.edu

Abstract

This lab aimed to automate the creation of a batch file to process music tracks. The process involved building a batch file that generates commands for each song in the directory, allowing for automated analysis. This lab extended this work by processing all JSON files, compiling extracted data into a consolidated data frame, integrating LIWC data, and merging multiple data sources into a single dataset. The task provided practice with installing, loading, and learning to use libraries, working with character objects, coding for loops, and accessing elements of vectors and lists.

Keywords: libraries; character objects; for() loops; and vectors/lists.

1 Introduction

In 2018, the Front Bottoms, Manchester Orchestra, and All Get Out collaborated on a track called "Allentown." To explore which band contributed most to the song, a data analysis was conducted on their previous tracks. The goal was to analyze the sound features on each track using Essentia (Bogdanov et al., 2013), an open-source tool for music analysis. Given the large number of songs to process, the task was to automate the data collection process using R, enabling a more efficient workflow for handling 181 tracks. In this lab, we will be working with smaller set of .WAV files to complete this same task. This lab aims to build a batch file to facilitate the analysis of each track's audio data, automate the command line process for each song, clean the data, and integrate multiple data sources to create a comprehensive data set that will help us answer the question of which band contributed most to the song.

2 Methods

The data consists of audio files in the .WAV format stored in a nested directory structure. The first directory level represents artists, and the second represents albums. The analysis involves building a batch file and processing the JSON file in order to compile data.

2.1 Task 1: Build a Batch File

Task 1 focuses on creating a batch file that automates the execution of Essentia for each audio track. The key steps in this process include:

1. Extracting subdirectories for each album
2. Filtering and counting .WAV files in each album's subdirectory
3. Constructing commands that execute the Essentia program for each track
4. Writing these commands to a batch file for execution

We used the `stringr` package for R (Wickham, 2023) to manipulate and analyze file paths and the `list.files()` function to retrieve directory contents. The `for()` loop structure facilitated processing each track.

2.2 Task 2: Process JSON Output

Task 2 focuses on processing the JSON output. After the batch file runs and generates the JSON files, the next step is to process these JSON files and analyze the data to determine the musical contributions of each band. In this lab, we focused just on the .JSON output for the song Au Revoir (Adios) on the Talon of the Hawk album by The Front Bottoms. The steps involved in this task are:

1. Parsing the file names to extract the artist, album, and track information
2. Loading the JSON data into R
3. Extracting key audio features

We used the `stringr` package for R (Wickham, 2023) to handle string splitting in the extracting process. The `jsonlite` package for R (Ooms, 2014) was utilized, specifically the `fromJSON()` function to convert the JSON files into R list objects for easier extraction of the relevant features.

2.3 Task 3: Compile Data

To extend the previous analysis, lab 3 focuses on loading and cleaning the data given the 181 JSON files and the CSV file. We merge all of the desired data into one data frame so that we can more easily examine the musical contributions of each band. The steps involved int his task are:

1. Complete Task 2 for all JSON files
2. Loading and cleaning the data from the Essentia models and LIWC
3. Storing extracted data vectors rows in a data frame
4. Merging the data from the `streaming_music_extractor` calls, the Essentia models, and LIWC into one data frame
5. Creating training and testing sets

In addition to the `stringr` (Wickham, 2023) and `jsonlite` (Ooms, 2014) packages for R, we used `read.csv` to load the CSV files. The `rowMeans` function was useful in the data cleaning process, specifically to average the different extractors for each feature. The `merge` function was utilized to merge all of the data into one data frame, and finally, `write.csv` was used to write the training and testing CSV files.

3 Results

3.1 Task 1 Results - Lab 2

The R script successfully identified album subdirectories and filtered .WAV files. It then generated batch commands for each track, which were saved in a text file named `batfile.txt`. These commands generated the Essentia program for each track, saving the corresponding output as JSON files. The process was automated, enabling batch processing for all audio tracks.

3.2 Task 2 Results - Lab 2

Task 2 involved processing the JSON output for a single track. The artist, album, and track name were extracted from the file name, and relevant audio features were successfully extracted. Although only one track was processed, the methods are scalable for multiple files.

Feature	**Value**
Artist	The Front Bottoms
Album	Talon Of The Hawk
Track	Au Revoir (Adios)
Avg. Loudness	0.5450
Spectral Energy Mean	0.0218
Danceability	0.9749
BPM	140.88
Key	A
Key Scale	Major
Length	108.49

Table 1: Extracted Audio Features for "Au Revoir (Adios)"

3.3 Task 3 Results - Lab 3

Task 3 involved extracting and compiling data for all JSON files in the EssentiaOutput folder, loading and cleaning the model outputs, loading and cleaning the LIWC data, and merging the extracted data sets into a single, unified data frame. This information was successfully merged into a data frame titled `merged_df` with 181 rows (each different track) and 140 columns (representing each data point). A training data CSV file, titled `trainingdata.csv`, containing all the tracks except "Allentown" and a testing data CSV file, titled `testingdata.csv`, containing only the track "Allentown" were both successfully written, which will be helpful in determining which band contributed most to the song.

4 Discussion

Lab 2 successfully automated the process of generating batch file commands and processing JSON output to extract audio features for analysis. Task 1 demonstrated the efficiency of automating file handling and command generation, while Task 2 allowed for the extraction of key features from JSON files. In lab 3, we were successfully able to complete task 2 for a much larger data set. We extracted all the key features of each song and wrote the desired information into a CSV file that we can later use to analyze which band contributed most to the song.

References

Bogdanov, D., Wack, N., Gómez Gutiérrez, E., Gulati, S., Boyer, H., Mayor, O., Roma Trepat, G., Salamon, J., Zapata González, J. R., and Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In Britto, A., Gouyon, F., and Dixon, S., editors, *14th Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 493–498, Curitiba, Brazil. International Society for Music Information Retrieval (ISMIR).

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects. *arXiv:1403.2805 [stat.CO]*.

Wickham, H. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.1.

5 Appendix

	artist	description	feature
1	All Get Out	Outlying	spectral_skewness
2	Manchester Orchestra	Within Range	spectral_skewness
3	The Front Bottoms	Out of Range	spectral_skewness
4	All Get Out	Out of Range	spectral_rolloff
5	Manchester Orchestra	Within Range	spectral_rolloff
6	The Front Bottoms	Out of Range	spectral_rolloff
7	All Get Out	Outlying	spectral_kurtosis
8	Manchester Orchestra	Within Range	spectral_kurtosis
9	The Front Bottoms	Out of Range	spectral_kurtosis
10	All Get Out	Outlying	spectral_entropy
11	Manchester Orchestra	Within Range	spectral_entropy
12	The Front Bottoms	Out of Range	spectral_entropy
13	All Get Out	Out of Range	spectral_energyband_middle_high
14	Manchester Orchestra	Within Range	spectral_energyband_middle_high
15	The Front Bottoms	Out of Range	spectral_energyband_middle_high
16	All Get Out	Out of Range	spectral_complexity
17	Manchester Orchestra	Within Range	spectral_complexity
18	The Front Bottoms	Out of Range	spectral_complexity
19	All Get Out	Out of Range	spectral_centroid
20	Manchester Orchestra	Within Range	spectral_centroid
21	The Front Bottoms	Out of Range	spectral_centroid
22	All Get Out	Out of Range	melbands_spread
23	Manchester Orchestra	Within Range	melbands_spread
24	The Front Bottoms	Out of Range	melbands_spread
25	All Get Out	Out of Range	melbands_flatness_db
26	Manchester Orchestra	Within Range	melbands_flatness_db
27	The Front Bottoms	Out of Range	melbands_flatness_db
28	All Get Out	Out of Range	erbbands_skewness
29	Manchester Orchestra	Within Range	erbbands_skewness
30	The Front Bottoms	Out of Range	erbbands_skewness
31	All Get Out	Outlying	erbbands_flatness_db
32	Manchester Orchestra	Within Range	erbbands_flatness_db
33	The Front Bottoms	Out of Range	erbbands_flatness_db
34	All Get Out	Outlying	dissonance
35	Manchester Orchestra	Within Range	dissonance
36	The Front Bottoms	Out of Range	dissonance
37	All Get Out	Out of Range	barkbands_skewness
38	Manchester Orchestra	Within Range	barkbands_skewness
39	The Front Bottoms	Out of Range	barkbands_skewness
40	All Get Out	Outlying	barkbands_flatness_db
41	Manchester Orchestra	Within Range	barkbands_flatness_db
42	The Front Bottoms	Out of Range	barkbands_flatness_db
43	All Get Out	Outlying	average_loudness
44	Manchester Orchestra	Within Range	average_loudness
45	The Front Bottoms	Outlying	average_loudness
46	All Get Out	Outlying	chords_strength
47	Manchester Orchestra	Within Range	chords_strength
48	The Front Bottoms	Out of Range	chords_strength
49	All Get Out	Out of Range	conj
50	Manchester Orchestra	Outlying	conj
51	The Front Bottoms	Within Range	conj
52	All Get Out	Out of Range	Perception
53	Manchester Orchestra	Within Range	Perception
54	The Front Bottoms	Out of Range	Perception
55	All Get Out	Outlying	OtherP
56	Manchester Orchestra	Outlying	OtherP
57	The Front Bottoms	Within Range	OtherP
58	All Get Out	Outlying	positivewords
59	Manchester Orchestra	Outlying	positivewords
60	The Front Bottoms	Within Range	positivewords

Table 2: Summary of Features Identifying Influencing Band