INFO 5602 Final Project Writeup

Lawrence J. Hessburg[*]

University of Colorado Boulder

Abstract

For the past 4 years, I have been using a service called Last.fm to record every song I listen to from my music streaming apps. In this time span, I have gathered roughly 45,000 individual listens (or ”scrobbles”). The Last.fm website automatically provides some basic statistics about this dataset, along with some limited browsing features, but I wanted to gain some deeper insight into my music listening trends, and correlate them with major life events. To do this, I scraped my data from the service using the Last.fm web API, and processed the resulting dataset to produce some basic visualizations about my music listening habits, and how they have been affected by major events in my life over the past few years.

1 Introduction

Last.fm is a music website that was originally introduced in 2002 as a music recommendation service. Using an open-source API, users were able to create applications which recorded individual listens (called ”scrobbles”) from their music-listening platforms (such as Spotify, Soundcloud, iTunes, online radio stations, and even offline apps like Windows Media Player) and uploaded them to the Last.fm database. By gathering a dataset of the music-listening habits of each individual user, Last.fm aimed to provide quality recommendations for new music to its users by linking users with similar taste.

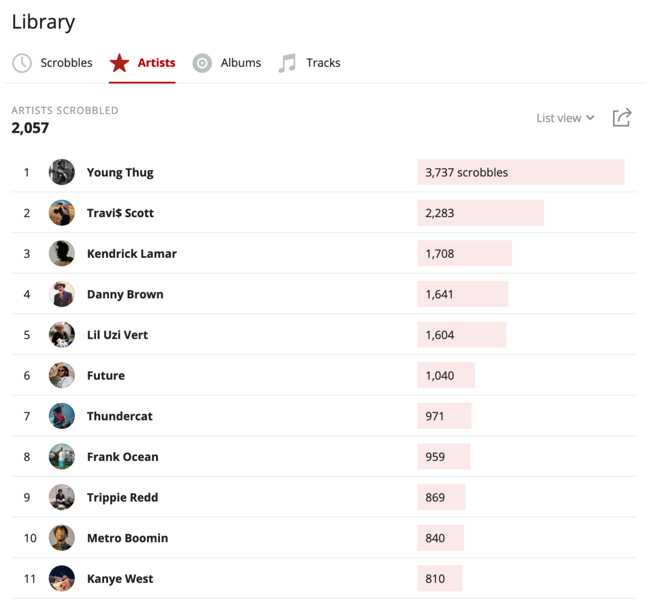
While the music-recommendation component of Last.fm’s service never really took off, and indeed the platform itself is still not widespread in the mainstream, there has been a strong, steady userbase, and a growing community of developers contributing to the service since it’s inception. As digital music consumption has moved from locally downloaded files to online streaming over the past decade, the community has continued to maintain support for recording scrobbles from modern streaming platforms. Most modern streaming platforms contain their own robust music-recommendation services that often use modern big-data and machine-learning techniques to provide useful recommendations, and in that regard Last.fm has fallen to the wayside. In recent years however, the platform has transitioned from being primarily a recommendation service to almost a sort of music-based social media service, driven by its users music listening data.

Within the last few years, Last.fm has implemented multiple ways to browse through your music-listening dataset through its web interface, and view users with tastes similar to yours. Additionally, the service has added a number of basic data visualizations to view your music listenind habits over time. Examples of some of these can be found below:

While browsing these pages is entertaining and somewhat insightful, the main focus of these visualizations is to associate and compare one’s music listening trend’s with other users on the platform, rather than providing deep insight into one’s individual habits and trends over time.

Since each ”scrobble” is annotated with the track name, album, artist, time and date of listen, and other useful metadata, the full set of one’s scrobbles provides an incredibly rich dataset that has a staggering number of exploration options. With this project, my goal was to harness my own scrobble dataset and associate it with various major life events over the course of the last few years, to see if any interesting patters seem to emerge.

[*]e-mail: lahe2512@colorado.edu



Library

ARTISTS SCROBBLED
2,057

1	Young Thug	3,737 scrobbles
2	Travi$ Scott	2,283
3	Kendrick Lamar	1,708
4	Danny Brown	1,641
5	Lil Uzi Vert	1,604
6	Future	1,040
7	Thundercat	971
8	Frank Ocean	959
9	Trippie Redd	869
10	Metro Boomin	840
11	Kanye West	810

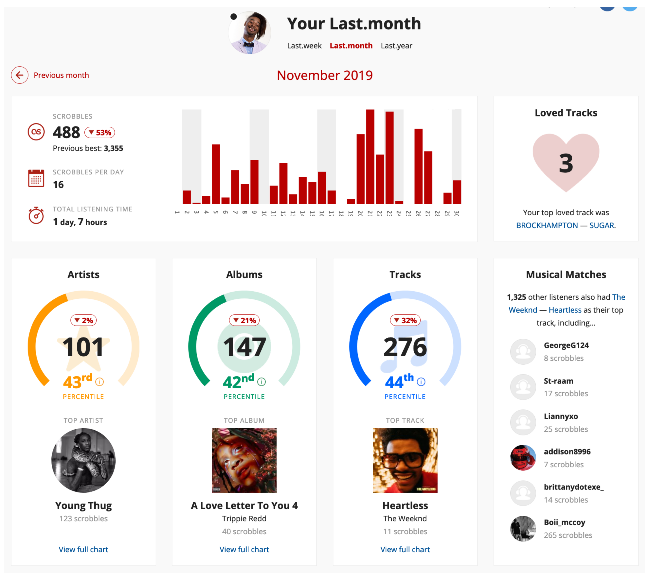
Figure 1: My top artists of all time as visualized by Last.fm

2 Related Work

As mentioned previously, Last.fm itself does provide a number of data visualization options. More specifically, it provides the ability to select any arbitrary range of dates, and view one’s artists, albums, and tracks in order of most scrobbles within that date range. An example of this is found in Figure 1. Additionally, the service provides the Last.week, Last.month and Last.year pages, which provide a summary of one’s scrobbling history over the past week, month and year respectively. These summaries include total number of artists, albums and tracks scrobbled in these timeframes as a percentile of the overall last.fm userbase. Also, they report histograms of scrobbling habits by time of day and day of week within the date ranges. Lastly, they provide a list of other Last.fm users with similar scrobbling reports within the date range.

Due to the robust and user-friendly web API provided by Last.fm, there are a plethora of user-developed visualizations of last.fm data. Generally, these visualizations provide time-series representations of specific aspects of the dataset such as stream graphs or bar charts, and other visualizations like word-clouds based on genre tags. An example of one such user-made vis can be found in Figure 3.

Aside from visualizations, there are also a multitude of user-made applications that interface with the Last.fm web API to either scrobble listens to the dataset, or download records of scrobbles by user. There are even apps such as VinylScrobbler which allow a



Your Last.month

Last.week Last.month Last.year

Previous month

November 2019

SCROBBLES
488 ▼ 53%
Previous best: 3,355

SCROBBLES PER DAY
16

TOTAL LISTENING TIME
1 day, 7 hours

Loved Tracks
3
Your top loved track was BROCKHAMPTON — SUGAR.

Artists
▼ 2%
101
43rd PERCENTILE
TOP ARTIST
Young Thug
123 scrobbles
View full chart

Albums
▼ 21%
147
42nd PERCENTILE
TOP ALBUM
A Love Letter To You 4
Trippie Redd
40 scrobbles
View full chart

Tracks
▼ 32%
276
44th PERCENTILE
TOP TRACK
Heartless
The Weeknd
11 scrobbles
View full chart

Musical Matches
1,325 other listeners also had The Weeknd — Heartless as their top track, including...
GeorgeG124 8 scrobbles
St-raam 17 scrobbles
Liannyxo 25 scrobbles
addison8996 7 scrobbles
brittanydotexe_ 14 scrobbles
Boii_mccoy 265 scrobbles

Figure 2: Some statistics about my last month of listening, along with similar users

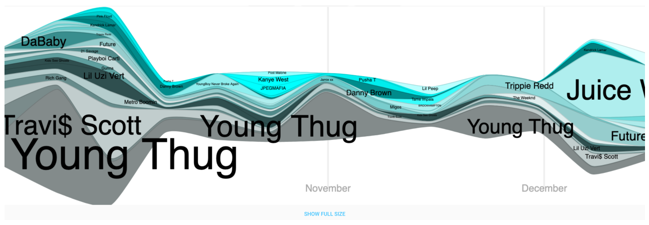


Figure 3: Stream graph of my top artists for the past 3 months as visualized by savas.ca/lastwave

user to simply enter the name of an album they are listening to in physical format, and scrobble said album's tracks to their Last.fm profile, timestamped appropriately for the length of said tracks.

Lastly, there are many more community-developed repositories that act as interfaces and/or wrappers between the Last.fm web API and common programming languages such as Python and Javascript.

To reduce repeated work, many of these projects were integrated into my own. Further details into the implementation can be found below.

3 Project Description

3.1 Data Acquisition

The first step in implementing my visualizations was collecting my scrobble data from the Last.fm database. I spent a significant amount of time researching various options. The first was to write a web scraper from scratch using my Last.fm account's API key. Since the Last.fm API conforms to the REST standard, this process is actually relatively straightforward (albeit time-consuming), and response payloads can be easily formatted and concatenated with common formats like XML and JSON. The second option was to use a pre-existing interface such as pylast to handle the API calls, and write the code for converting the response payloads into something more useable suh as a .csv file. Lastly, I discovered a user-made application called Last.fm to csv, which automates the entire process. This service only includes track, album, artist and time/date, excluding other metadata such as genre tags, which is sub optimal for my goal. However, due to time constraints I opted for this as my data acquisition method. Last.fm to csv provides a .csv file containing one line for every scrobble in my account's history, with the aforementioned parameters.

3.2 Data Preprocessing

The hardest and most time-consuming part of this project was the data preprocessing. I originally set out to visualize trends in properties like my top-n artists over time. However, converting a list of individual datetime-labeled scrobbles into a format usable for purposes like that turned out to be a very non-trivial data structures problem. This is due to the fact that I had to essentially move through time, keeping track of my top-n artists at the current time step in a separate data structure, along with other useful data I wanted to visualize. For all internal data representation and preprocessing, I used the common python libraries numpy and pandas.

3.3 Visualization Implementation

Once I had the data processing sorted out, I explored various ways to visualize said data. Because I had some experience with it from the other projects in this class, and becaus the end goal was to host the visualization on a web page, I opted to use the python library bokeh for my vis. Bokeh is a visualization library specifically targeted for web hosting, and provides a high degree of built-in interactivity that can be utilized. Additionally, it integrates quite seamlessly with other common python libraries such as numpy, pandas and matplotlib.

3.4 Visualization Features

4 Analysis

5 Future Work

6 Conclusion