Proposal for Analysis of Pitchfork Album Reviews

J.T. Liso, Caleb Fabian, and Richard Riedel

***Abstract*— Pitchfork is the premier website for album reviews of all genres of music. Critics commonly claim that the reviewers are biased toward specific artists or genres. Is this an accurate sentiment? This project seeks to analyze the patterns within album reviews on Pitchfork. We will collect all album reviews from Pitchfork's conception in 1999 and perform textual and numerical analyses on the data. Afterwards, The data will be compared to other information metrics such as Spotify streams and YouTube views. Our end goal is to explore the habits of specific reviewers, find overrated or underrated albums, artists, or genres, and determine if specific times or seasons lead to better reviews.**

I. MOTIVATION

As online music review sites become more prevalent, critics claim that certain sites have certain biases toward various genres and artists. People rely on these websites for recommendations, so the accuracy of the content is vital for rising artists' success. Since Pitchfork.com is the premier mainstream media source for music reviews and news, we have decided to explore their reviews for inaccuracies and biases.

To do this, we have decided to collect all reviews saved on Pitchfork's album reviews endpoint[1], which dates back to 1999. We will compare the ratings and textual content of the reviews with other online information such as Spotify and YouTube streams to compare the review score to the popularity of the music. Additionally, we will sort the data based on reviewer, genre, month of review, and artist. With this sorting, we will attempt to discover additional biases.

II. DATA COLLECTION

Data collection will be divided into two separate parts. First, web crawler will need to be designed and created to scrape the Pitchfork album reviews endpoint while only collecting and sorting the relevant data. In addition to the Pitchfork scraper, scrapers will be needed to access and organize the YouTube and Spotify stream metrics

A. Pitchfork Reviews

Pitchfork contains an archive of every album the site has ever reviewed since January 1999, totaling to over 18,000 different reviews. Each review contains a header containing the album name, album cover, the artist, and the score given to the album by the reviewer (on a scale from 0 - 10 in .1 increments). Following the header, information about the reviewer, the date of the review, and the textual review of the album in about 1000 words are listed. The review also declares other information such as if the album is a "Reissue" or considered "Best New Music" (given to new albums with 8.0+ scores that are especially good).

We intend to create a database with the following 6 fields of the review:

1) Album Name
2) Artist Name
3) Album Review Score
4) Album Genre
5) Album Review Content
6) Reviewer Name

The numerical values of the album review score will be analyzed over different time series such as comparing by month, year, day of the week, and time of the day. The scores will also be analyzed for potential normal distribution and perhaps predicting certain scores using the model. Additionally, the scores will be divided by reviewer to analyze patterns within each different reviewer.

We will use an undetermined textual analysis method to parse the album review content and determine how closely related certain words are to high or low album review scores. However, as storing the reviews will be quite data-intensive, we intend to store the URL of each review. This way, when we need to access the review, we can just query the webpage of the review using the requests framework in Python (as demonstrated in Mini-Project1). This will prevent loading, storing and querying a large database each time we do an analysis. Yet this does come with a speed tradeoff as reading from disk is much faster than querying a webserver each time. To alleviate this, we will create a memorization cache of some number of the most recent queries.

B. YouTube and Spotify Streams

Ideally, we would compare the Pitchfork reviews to album sales, but since the music industry has turned to a stream-based industry, we decided to collect information from two of the most-used music streaming services, YouTube and Spotify.

We will implement two different web scrapers that crawl YouTube and Spotify, respectively, for the total number of streams for each album that is reviewed on Pitchfork. Because this step is dependent on the names of the albums and artists that appear on Pitchfork, this step cannot proceed until the data collection of the Pitchfork Reviews is completed.

One perceived problem is that YouTube allows users to post videos containing music that they do not have the rights to. Due to this, there may be multiple videos for a single album. To account for this we will either have to determine which posting of the music is official (posted by the record label or the band itself) or just use the posting with the most

[1] https://pitchfork.com/reviews/albums/

views. Alternatively we could use all of the postings of an album and combine (or average) the view counts. Another alternative it to use YouTubes music rights section within the video metadata to state that the video belongs to a certain song and sum the views of all videos together, and use that sum as the total number of views. Another problem is that there may be relevant videos with the same title, but they are not the song, thus we must make sure that the search queries we provide are as exact as possible.

Spotify, on the other hand, only has postings from official users, i.e. the band/artist or record label. This allows us to pull the information metrics without potentially invalid data. One potential issue is that Spotify tracks stream counts by songs, and not by album. Consequently, we will calculate the average stream count for each album by finding the mean number of streams for each song on the album. This calculation will give us a better idea of how many people listened to an entire album rather than just a single song within the album.

This stream count combined with the Pitchfork metadata described previously will be stored in one cohesive database. This make the various analyses much easier as we can just poll the database for specific artists, reviewers, and albums to get the corresponding data.

III. TENTATIVE TIMELINE

The following is a tentative timeline for this project. As ideas get added or changed, the timeline will be modified accordingly.

- **October 2** - Determine what APIs we need to use and what languages support them (Completed)
- **October 13** - Functioning Pitchfork web scraper should be finished, and needs to begin running (in case the scraping takes longer than expected). Determine what type of textual analysis to use for the written portion of the reviews.
- **October 20** - Functioning code for textual analysis of reviews
- **October 27** - Use the results from the Pitchfork web scraper to scrape the number of streams from Spotify and/or YouTube for each album that is released on the sites
- **November 3** - Determine what graphs and charts we want to generate based on the preliminary results
- **November 17** - Completed code for data analysis including how we will generate graphs and charts
- **December 1** - Finish up any lingering data analysis questions and prepare for final presentation

IV. EXPECTED OUTCOMES

From this analysis, we hope to conclude whether Pitchfork reviews contain biases. From our own experiences with the website, we believe that we will find that certain artists will be unreasonably favored over others of the same genre by certain reviewers. However, we do think there will be some sort of normal distribution of review scores for most artists. We also believe to find that certain genres get higher reviews in certain times of the year, as well as different reviewers rating the same artists quite differently. With this, we hope to guess the musical tastes of each reviewer and determine his/her favorite genres and/or artists.

Moreover, this project will serve as a beneficial use of the skills gained from the Fundamentals of Digital Archaeology course. This project will use the textual and numerical analysis skills demonstrated in Mini-Project 1 as well as the web-scraping and database management skills from Mini-Project 2. We are considering the potential of using R for some of the data analytics as that is another component of this course (although our team has limited experience in R).