LyricCheck

Technical Report

PREPARED BY
Hassan Khan
Stacy Jong
Jack Si
Kenji Nakachi
Christopher Carrasco

Motivation

This website is one that consolidates data on top artists (monthly listeners), charting songs, artist bios, song lyrics, and a profanity score. Using Perspective's NLP model to grade song lyrics, our project will also provide insight on the profanity preferences over time, based on location, and based on artists. It will additionally have a Spotify playlist analyzer that will grade the profanity level of Spotify playlists, allowing for a bulk solution to children-safe songs. There are sites that tackle one or even part of one responsibility that we hope our project will accomplish, but there are severe patches of information missing from each. By also making use of a profanity grade, our project can provide valuable insight into the worrying NSFW trend of modern day pop culture.

User Stories

Phase 1

- List Countries on Artist Pages

  "I was using your site when I found an artist I hadn't heard of before. I wanted to find out what country they were from, but it wasn't listed on the artist's page. I would like a link to the country page(s) that each artist is from."

  Status: Not implemented.
- Lyric Format

  "Hello, I am a user who really enjoys reading lyric lines, but it is slightly difficult to do so on your website. I think it would be great if they were spaced line by line. This would really help me be able to follow along with the lyrics easily."

  Status: Not implemented.
- Artist Images on Top Songs

  "Hi, I'm a user who likes visuals and tries to recognize artists by their faces. I noticed that under Charting, the Top Artists had images of the artists whereas Top Songs didn't. I think if the Top Songs could also include artist images, then it would look more consistent across the board and appealing."

  Status: Not implemented.
- Profanity Metric Explanation

  "Hello, I am a user who wants to look at lyrics before I let my child listen to them. I think it would be great if the profanity average had an explanation. This would make it much easier than having to read all the lyrics."

  Status: Not implemented.

- Similar Songs Attribute

  “Hello, I am a user who is actively looking for new music. I like that you have a "similar artists" attribute on the Top Artists models. However, I think that I would also benefit from a "similar songs" attribute for the on the Top Songs models!”

  Status: Not implemented.

Comments: Although some of the features in the user stories are within the scope of Phase I, we were not able to get to them as the user stories came in close to the Phase I submission date, leaving insufficient time to implement.

RESTful API’s

Perspective https://perspectiveapi.com/

- We use Perspective API to provide a “profanity score.” A song’s lyrics are ran through the API and provides a score based on profanity, severe toxicity, sexual explicitness, and identity attacks. These metrics are returned along with an average.
- Using this, we can give an artist, or album, a profanity score by averaging the scores of their songs. Likewise, we will be able to use the user's playlist using the profanity scores of the songs in the playlist. We also give a profanity score for a country by averaging the profanity score of the current top 50 songs in the country.

Spotify https://developer.spotify.com/documentation/web-api/

- Spotify does not provide all the artist and song information we would like, but we are using Spotify to provide information on users’ playlists. We will hopefully be able to score a playlist on profanity using an average of the songs inside the playlist. And we may sort and list the songs in order of decreasing profanity.

Last.fm https://www.last.fm/api/intro

- Last.fm provides most of the information concerning an artist, album, or track such as the description information, the top albums/tracks for an artist, along with some information on the current top charts.

REST Countries https://restcountries.com/

- We use REST countries to get a list of all the countries and non-music related data about them. Out of the API, we’re able to get names, location information, language, map urls, and flag urls images.

MusixMatch https://developer.musixmatch.com/

- MusixMatch lets us search songs and get the lyrics for the results. We feed the list of songs we get from Last.me into MusixMatch to get their lyrics, then we can feed the lyrics into Perspective to get their profanity score.

Models

Each of the three models are accessible through the navbar where songs and artists are together under *Charting* as *Top Songs* and *Top Artists*, and countries through *Top Profane Countries*.

Country

- Each country page has a display of each country's name and a picture of their national flag. In addition to their common name, an official name and country code is provided. In the future, these country codes will be used to be able to look up countries more efficiently without having to write out their whole name. After their geographical location (given by continent and relative location), the profanity score of each country will be displayed. These scores consist of songs that originate from their country and their average amount of toxicity, profanity, sexually explicit and identity-attacking lyrics, and average overall. These scores are derived from the Perspective API, which produces these scores for each country it is given. Below that is a reactive Google Maps window positioned on the country. The languages spoken in the country are listed below that. Finally, a list of top songs and top artists of that country is shown.

Artist

- Each artist page displays their portrait on the upper-right side of the screen of them or their preferred picture visible on Spotify. Below that, running down the right side of the page, is their social media. Below that is their profanity score. And finally below that is their stats on Spotify. To the left of all that information begins their biography. Then, below that is their albums and songs are listed in decreasing order of popularity. The album/song's covers, along with the text below saying its name, will be linked to that album/song's page.

Song

- Each song page begins by displaying the song title and the artist that wrote it. Afterwards, a view count derived from the last.fm is shown, allowing users to see how popular a certain song is compared to others. We are able to sort the top songs by this statistic, as well as song name, genre, and profanity score. The profanity score, derived from last.fm, provides a score for the song's lyrics, including profanity, toxicity, and sexually explicit and identity-attacking lyrics. The lyrics of the song are also displayed on the page for the user to read for themselves.

Tools

Frontend

- ReactJS - used for web application development and interfacing with backend.
- React Bootstrap - CSS framework used for splash, model, about pages, and navigation bar

Backend

- Namecheap - domain name registrar used to obtain lyriccheck.me.
- AWS Amplify - cloud platform used to host the React app

DevOps

- GitLab - DevOps software used for source code management and Continuous Integration.

API Documentation

- Postman - used for testing and building APIs
    - https://documenter.getpostman.com/view/19683184/UVksKDAj

Style

- ESLint - linter used to statically analyze our code to quickly find problems.
- Prettier - an opinionated code formatter

Hosting

Amazon Web Services (AWS)

- We obtained our URL “lyriccheck.me” from the domain name registrar Namecheap. We chose to use AWS Amplify to deploy our React application. Following the setup guide provided (https://github.com/forbesye/cs373/blob/main/React_AWS_Setup.md), we set the host to our Gitlab repository. We then connected our domain name and received an SSL certificate to enable https. There was a bit of difficulty setting this part up as we had previously connected our domain name to a hosting zone/buckets, and registered it with NameCheap for our static HTML website, but it did not take too long to sort out. We then added CNAME and ALIAS records to our registrar and waited for the changes to propagate. Our site is now accessible from lyriccheck.me, www.lyriccheck.me, https://lyriccheck.me, and https://www.lyriccheck.me.