Universitá della Svizzera italiana

Information Retrieval

Course Project N.19: A Music Search Engine

Academic Year 2018-19

Satish Kumar

supervised by
Prof. Fabio Crestani

December 6, 2018

Introduction

In this project, I created a music search engine. Which help user to search in a repository of music and songs like SoundCloud or Spotify etc. He can submit some keywords to search for a music/song. User can search for songs/ musics for based on their textual description.

Additional feature is automatic recommendation. When user will select any song than it will return a list of songs which is similar to current user selected song based on say, genre, title, mood etc.

Another feature, Users can use the clustering like looking for songs title, album, artists or Genres/mood.

Design

My application is called Search and it is a search engine for music. It collects data about songs form Spotify which is the worlds most popular and authoritative source for songs content. The application is composed of two main parts: search bar, results.

Search bar:

The search bar is accessible from the top of the page, it is very simple and its functionality is very clear. Its pretty big with some clear placeholder text 'Search...' to hint the user. You need to press a search button. And when nothing is typed in the search bar all songs are shown, so if someone does not know what title to look for but he only wants to browse songs in general he can.

Results:

The results section is where we the results from the search are presented. When we want we can select a title, album, artist, genre to show all the music from the current query contained in that title, album, artist, genre . The songs are presented with some clean cards that contain all information needed for a preview. They are subdivided by four per row.
The card for each songs contains the poster image, the title, the album, the genres, and the artists.
There are another page also when we click on any song it will show a music

player near to search bar and all another are same but now songs in the result fields are not according to search. It is similar to the user selection song with album, artist, genres.

Implementation

Requirement

- Spotify API
- Python
- Solr
- Flask
- Requests
- urllib
- simplejson
- HTML
- AJAX
- javascript

Crawling

I am adding one txt file datas.txt, which contain the data. Than you do not crawl again. It is time consuming. I crawled 10004 songs.
We crawl one website: Spotify. I used the Spotify api for crawling because we got difficulties to crawl Spotify with Nutch. I tried many websites with Nutch but i did not get any success.

And we save the following properties:

- Title
- Album
- Artist
- Image

- GenresSpotify

After discussion with TA, he suggested me, we can use the Spotify API for get the songs metadata. But for the Spotify api we have to need register our app on Spotify and get the client id and client secret code. And after getting this data, I saved it in the txt file which is data.txt.

How to crawl

- cd Spotify-master
- open songs.py
- enter your client id and client secret id
- python songs.py
- it will create a data.txt file
- you can add manually this file to solr

Now, it will save all the songs in data.txt file. And we upload the data manually on line solr.

Now everything is ready to start the queries, Solr handles the queries too. We can search by field and it will return 200 results It has to display. I created songs, album, artists, genres fields so that we can search directly in every field, with text in search field. This way we can search songs by title, genre etc. The application can access very easily to the queries with a get request.

User interface

For the User Interfiac, we used HTML, Ajax, javascript, Flask, requests, css. Flask is web framework for python. We send ajex request to send the data and click to Flask to response to users.

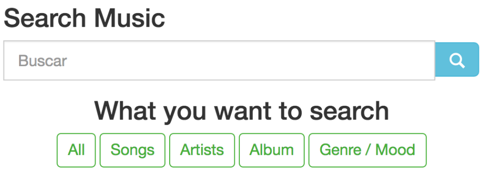
Start the application

I am sending my solr folder also because i made some changes in solr like More like this functionality. It is better if you use my solr database.
I am sending sportify master folder becasue if you want to crawl. My implementation is in app folder.
After finishing the crawling and put the data on solr:

- download solr from this link solr
- cd solr-6.5.1/bin
- ./solr start
- check solr start with port:8983
- cd ..
- cd ..
- cd spotify-master
- cd app
- python hello.py
- open webpage http://127.0.0.1:5000

Figures

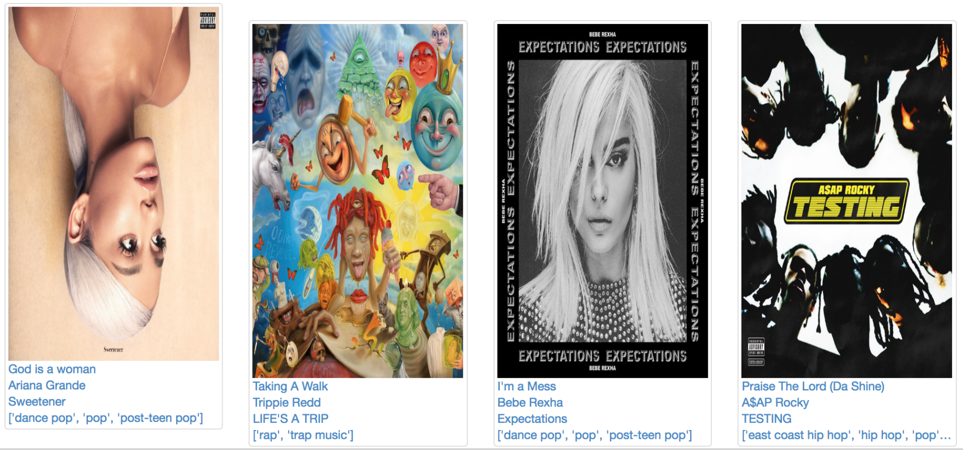
This is the main page:



Search Music

Buscar

What you want to search



God is a woman
Ariana Grande
Sweetener
['dance pop', 'pop', 'post-teen pop']

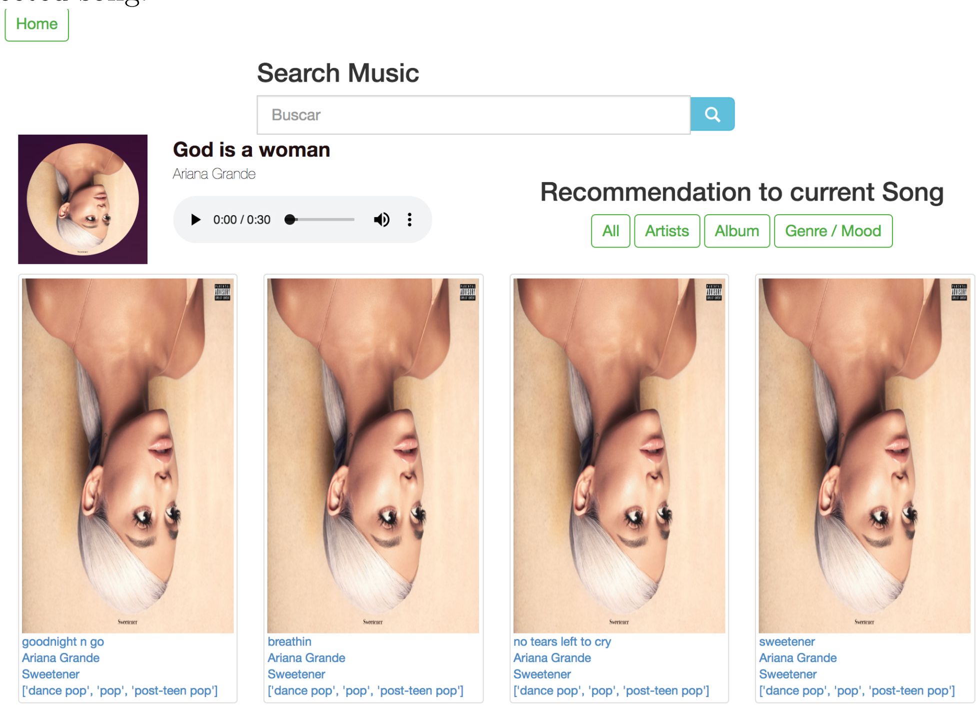
Taking A Walk
Trippie Redd
LIFE'S A TRIP
['rap', 'trap music']

I'm a Mess
Bebe Rexha
Expectations
['dance pop', 'pop', 'post-teen pop']

Praise The Lord (Da Shine)
A$AP Rocky
TESTING
['east coast hip hop', 'hip hop', 'pop'...

This is song selected by user
It use the more like this functionality. It show all the songs which is similar

to current selected song.



Search Music

God is a woman
Ariana Grande

Recommendation to current Song

goodnight n go
Ariana Grande
Sweetener
['dance pop', 'pop', 'post-teen pop']

breathin
Ariana Grande
Sweetener
['dance pop', 'pop', 'post-teen pop']

no tears left to cry
Ariana Grande
Sweetener
['dance pop', 'pop', 'post-teen pop']

sweetener
Ariana Grande
Sweetener
['dance pop', 'pop', 'post-teen pop']

Evaluation

User Name : Bojken Sina

Feedback : The website is fine, and its functionality is fine. Need more CSS. It work fine in term of Information retrival.

User Name : Luka Volk Feedback : The website works perfectly and easy to use.