LYRICS DATABASE TOOL FOR EXPLORATION AND DISCOVERY

ABSTRACT

The purpose of this tool is to allow end users to explore song and lyrics, of top 100 billboard songs from 1965 to 2017. The data set itself contains more than 5,200 songs 100 a year for the past 52 years. The initial idea was to develop a tool that recommends song/s to the end user based on certain defined criteria's by the end user. This we learned in Natural Language Processing is call Collaborative Filtering, which involves user input. Using analytics the more repetitive, common words used for different genres from 1965 to 2017 are fetched to recommend the suitable genre of songs. The recommender system is based on certain criteria the user defines, can be used to identify children friendly songs based on the search with words like “love”, “she” ,“happy” to appear on the lyrics and the tool can give an end user a recommendation of which songs from the billboard top 100 historical to listen to.

Team decided to build the tool based on lyrics because of interest in music. We got insight into music world while building the tool itself and believe will help other music lovers also.

Keywords

billboard, lyrics dashboard, R, tableau.

1. OVERVIEW

A standalone tool like this one might be unique. A tool like this one will best to integrate with music streaming services such as iTunes, Pandora, etc. Full integration tools will be more appreciated by typical end users because if all the tool does is to recommend them a song that has the lyric criteria they have selected but then don’t offer the end user to listen to the song itself, the end user might have to go to a different platform just to type the song name in on the search bar. That is a multiple step process rather verses were a fully integrated system can offer both. Recommender algorithms already exist on our every tools, sites and apps. This tool simply offers an end user filters and criteria they can use to recommend them songs they might enjoy with lyrics that suits their criteria selections

2. RELATED WORK

Various related work were referenced during the project**.** Research articles:
https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf

3. Overview of Functions

We chose to use a R based lyrics scrapper script because it is efficient way to access lyrics database websites. Following three websites sources were used for lyrics scrapping:

- metorlyics.com
- songlyrics.com
- lyricsmode.com

Lyrics scrapper provided necessary data points like song titles, year, artist, lyrics, etc., ETL processes were used to clean and load data for presentation layer. Lemur stop words library and Porter2 stem-words library were used in the process.

We decided to write the Tableau's drag-n-drop interface to build many visuals in the project. Tableau is world's leading business intelligence tool. Recommendation system for lyrics database and 1965 - 2017 Billboard Word Cloud Dashboard were built using tableau.

4. Implementation (Extract Transform and Load (ETL processes) Overview)

A.) songscrape2.R – Data Extraction Processes (https://github.com/jjasinski66/CS410FinalProject)

This script will automatically get the Billboard Top 100 for any year or sequence of years the user specifies.

- The script takes two command line arguments
  - beginning, The First year to start searching
  - ending, The last year to search
- This script requires R and utilizes the libraries (RCurl, XML, stringer, optparse)

Example command: Rscript songscrape2.R --beginning=1984 --ending=1984

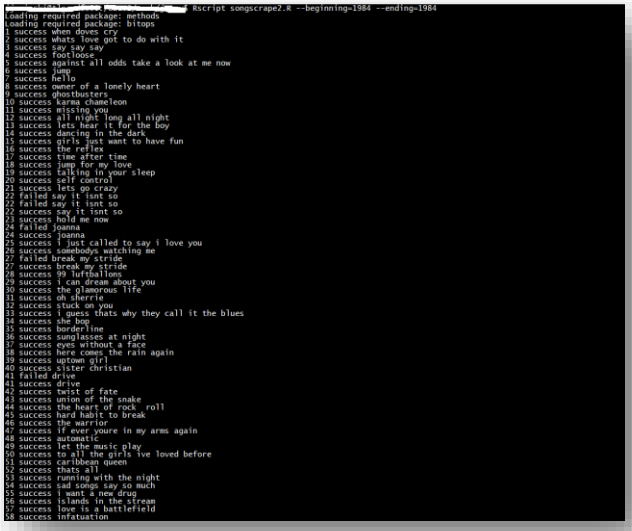
The Script acts in three different parts.

- Part 1: Get the song list from the Wikipedia page for the Billboard Top 100 for any year.
  - The script will retrieve and parse the wiki page for the specified year and create a dataframe with the Rank, Title, Artist, and Year of the song.
  - Each Years hits are pasted onto the data frame using the rbind command.
- Part 2: Use the Artist and Song information to lookup the song from one of three open websites.
  - note: I wanted to use this method for two reasons.
    - The Lack of required Hash key or login for any other API's makes this more user friendly and anonymous
    - Open websites make the dataset easily retrievable by anyone.
    - The XML used by these websites is consistent, and the tags necessary to find the lyrics are simpler to program.
  - The three websites used, metorlyics.com, songlyrics.com, lyricsmode.com have a huge repository of song lyrics. Between the three of them, the chance of finding a song's lyrics is greatly increased.
  - The script then loops over the data frame using the artist and song title to create a suspected URL of the song in question.
  - The script they tries to retrieve the lyrics from a site, if it fails, it goes onto the next site, and so on.
  - Once a lyric is found, the lyrics are added to the data frame.
- Part 3: Text processing on the lyrics.
  - Since the lyrics themselves are scraped from a web html, There are many non word characters that need to be cleaned.
  - Also, special characters need to be converted or eliminated.
  - End lines, and tab characters are replaced with empty strings.
  - All characters are converted to lowercase and normalized to UTF-8.
  - The error messages from websites for copyrighted, or missing lyrics are replaced with NA.
  - Lastly Instrumental hits have no lyrics, and are therefore converted to NA as well.
  - Finally the entire data frame is written to a csv for more convenient transportation.

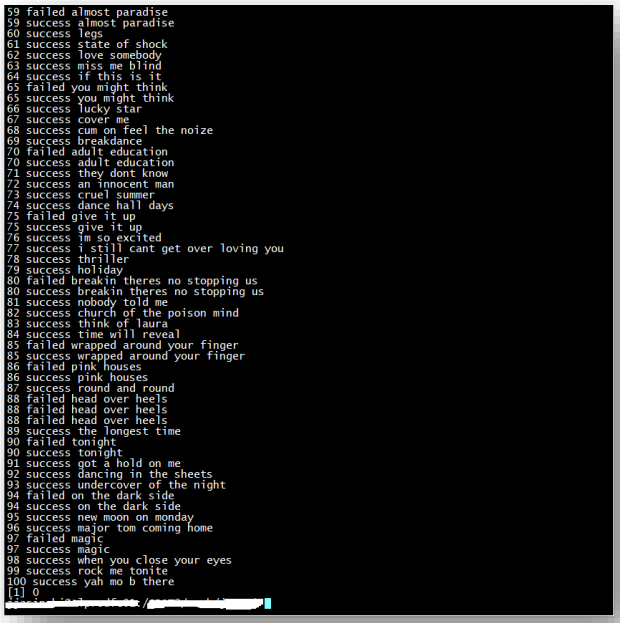
- note: The script will also display the percentage of songs that failed to get lyrics from any of the three sites.
- Future iterations of this scraper can merge lyrics from multiple sites for a more consistent data set.

Included in this repo is the set from 1965 to 2017

R Script Figure A:



R Script Figure B:



B.) Data Transformation processes:

- Since the data is captured on csv for easy transportation. To clean the data for the presentation layer it needs to be massaged to suit this layer
  - First things first you can open the file with any text editor, for example we can use Excel here as the csv editor.
  - The data without any type of massaging is in it's rawest form.
  - immediately after opening the csv file we see that artists are named in a variety of ways for example, we have Usher, Usher Featuring Ashanti, Usher Featuring P Diddy and all kinds of variations.
  - Thus this created a lot of redundancy and receptiveness on the artist column. Since featuring is just an addendum, the song itself is still entitled to the Original Artist. For example here, any songs from Usher featuring other artist is still entitled to Usher as the lead singer.
  - This is an important part of transforming the data because once we do a summation by the artist, the data aggregation layer needs to understand that the other songs featuring belongs to it's Original Artist, not by the featurettes for example.
  - There are 52 years within this period of evaluation. Having 52 selections for end users is a bit overwhelming, therefore the years were converted to decades. Decades that the songs fell into. With this we only have 6 different decades, it will be much easier to digest as an end user.
  - The data set entirely was also in lowered cases which is not quite presentable to end users when it is no properly formatted.
  - Therefore the next step is to use a Proper function to format all words with a upper case letter.
  - We will need 2 libraries for the next part
    - For Stop Words you can download the file from the repository or simply use the WGET function on python:

      wget -nc https://raw.githubusercontent.com/meta-toolkit/meta/master/data/lemur-stopwords.txt
    - For Stem words also call a lemmatizer, you can get this file from the repository or also by utilizing the WGET function

      wget -nc http://snowball.tartarus.org/algorithms/english/diffs.txt

- The purpose of the two libraries utilized is to inflect different forms of a word all reduce to the same representation. For example words such as lover, lovers, loving, will be reduced to it's root word Love instead of having many variations of the same representation of the word. Sort of like what we did with the Original Artists. Same idea
- The next steps without going into too much depth is basically a word count function to count which individual words within the lyrics occurred the most in all of 1965 – 2017 lyrics.
- By utilizing the two libraries mentioned above, we got rid of stop words such as he/she/it that occurred the most among common words and we converted all the words to it's stem words. We have picked among the top 20 of the unique words that occurred the most frequent within the lyrics, words include love, baby, girl etc seems to occurred the most in all lyrics.
- Now because of time constraints the team have decided to pick only the top 20 words that occurred the most within the lyrics after cleaning. This can be expanded further into top 50, top 100, or even top 500 words that occurred in history but anymore than that it will cloud the data set to make it massive that it will not be presentable to mainstream audiences.
- With the top 20 words selected. We then added new columns for each individual words to count for example to count the word love we basically utilized a formula such as *(=SUMPRODUCT((LEN("Lyrics")-LEN(SUBSTITUTE((UPPER("Lyrics")),UPPER("Love"),""))))/LEN("Love")))*
To explain it in words, did a sum products of the words in each individual lyric of the word love within the passage. If the word "Love" occurred 10 times within the lyrics the output of the formula will give a count of 10 for example. WE did this for all 20 individual top words.
- With the data cleansed. It is ready to do the development piece within a leading business intelligence tool call Tableau.

C.) Loading into Tableau Development Processes:

- The next phase is the development phase. Taking the cleansed data and making it presentable to mainstream audiences. This part does not involve heavy coding because Tableau is a business intelligence front end tool that is very user friendly. It offers drag and drop functions. The hardest part about using tableaus might be the strategic portion on how do you want to present your data and how to filter the data to make it presentable that the mainstream audiences can just pick it up without having any sort of training for dashboard/scorecard

functionalities. How to incorporate common filtering and slice & dice functionality that an end user can just define our tools as a user-friendly tool.

- Now the strategy the team have come up with is to separate the tool into two development portions. There is the dashboard portion and then the recommender system portion.
- On the dashboard potion, the general idea was to create individual visualizations in a worksheet. A visualization of all of the Original Artists within the data, sorted by A-Z so the end users can view by artist names along with the song titles that belonged to those artists. Then a word cloud of the top 20 unique words that occurred the most within the lyrics, this part we know we can add onto or enhanced in the future from expanding from 20 to few hundred words. A word cloud is a nice way to visualize this data set because the size and color of the words are associated with their total score. What total score is, it is simply a count of the individual words within history of the Billboard dataset from 1956 - 2017 if no other filters are selected. Since each individual word/s have a total score, Original artists can also be ranked and scored. This we can use the summation of total scores on artists as well if no other filters are selected. For example Keith Sweat have contributed a total score of 338 among the top 20 unique words and thus within his tree map he will have the biggest square box on the street map because his total score when distributed among other artists is the highest and vice versa. Now the next piece we want to think about are the individual filters. What do we want the end user to be able to filter on, for this portion we have added decades and Original Artist names. An end user can choose a decade and this will reduce the original artists to only the decade they occurred in. You can also choose an individual artist, this will narrow down the dataset to only show songs by this artist.

- The second portion we want to implement next is a recommend system promised. How can the team execute on something like a recommend system within Tableau. I mean Tableaus is not D3.js where it can be programmed as desire. Tableau is a drag and drop user friendly tool that offers workarounds. Workarounds is what we have to utilized here on Tableau for a recommended system. We know that to implement filters on Tableau is very straight forward. User chooses a decade and the dashboard will show only artists of that decade. Then the end user can choose an artist within that decade and the top lyrical word that the user wants to occur within the songs that belonged to the original artists. This seems very straight forward when explained in words that it could be just a three level filtering systems. Well this recommender system took some efforts. It wasn't as straight forward as the team thought it

was. At first we ran into multiple road blocks were the measures such as the 20 top words could not be used as a word cloud filter. Then we got rid of the word cloud selection because only dimensions could be filters. However, with further luck and research the team found another way to implement the word cloud filters into a parameter system. First we had to create all 20 parameters on tableau. This will the end user a selection box of the top 20 words. Then add a calculated field such that whenever a selection is made, please sum the total of the word count of that word selected. This has allowed the team to fix the word cloud selection issue.

- The formula for the paramters were basically an IF-Else statement that says if user chooses the word “Love” simply sum the total occurrences of the word “Love”

```
IF ATTR([Choose Word Cloud])="Away" Then Sum([Away])
ELSEIF ATTR([Choose Word Cloud])="Baby" Then Sum([Baby])
ELSEIF ATTR([Choose Word Cloud])="Girl" Then Sum([Girl])
ELSEIF ATTR([Choose Word Cloud])="Video" Then Sum([Video])
ELSEIF ATTR([Choose Word Cloud])="Photos" Then Sum([Photos])
ELSEIF ATTR([Choose Word Cloud])="Songs" Then Sum([Songs])
ELSEIF ATTR([Choose Word Cloud])="Music" Then Sum([Music])
ELSEIF ATTR([Choose Word Cloud])="Life" Then Sum([Life])
ELSEIF ATTR([Choose Word Cloud])="Heart" Then Sum([Heart])
ELSEIF ATTR([Choose Word Cloud])="Love" Then Sum([Love])
ELSEIF ATTR([Choose Word Cloud])="Light" Then Sum([Light])
ELSEIF ATTR([Choose Word Cloud])="Eyes" Then Sum([Eyes])
ELSEIF ATTR([Choose Word Cloud])="Call" Then Sum([Call])
ELSEIF ATTR([Choose Word Cloud])="Believe" Then Sum([Believe])
ELSEIF ATTR([Choose Word Cloud])="Emotional" Then Sum([Emotional])
ELSEIF ATTR([Choose Word Cloud])="Friends" Then Sum([Friends])
ELSEIF ATTR([Choose Word Cloud])="Time" Then Sum([Time])
ELSEIF ATTR([Choose Word Cloud])="Feel" Then Sum([Feel])
ELSEIF ATTR([Choose Word Cloud])="Night" Then Sum([Night])
ELSEIF ATTR([Choose Word Cloud])="World" Then Sum([World])
ELSEIF ATTR([Choose Word Cloud])="Total Score" Then Sum([Total Score])
ELSEIF ATTR([Choose Word Cloud])="Word Selected" Then Sum([Total Score])
END
```

- Now with multi-layers of filtering that acts as prompts for the end user for the recommender system, the system itself can do it's magic and narrow down to only a few songs from that decade, from the user selected, and that it also contained those individual words within that lyrics selected. This after solving the multiple roadblocks ladies and gentlemen's we have a tool that worked exactly like a recommender system to offer the end user a couple of songs based on their selections to listen to. Now this is not deep-learning or machine learning, it is not a model that learns user behaviors or trains the model using historical data that we know about the end user. We simply do not possess a household file that we knows the preferences of each individual household in the states. This simply is a three level filtering system. This as we all learned is class is collaborative filtering because we want the end user to give us the answers, it required interactions from the end users in order to choose the right songs for them.

- The last and final process in the implementation phase is to link the two tools together. We simply can do this by adding a hyperlink of the two tools that were posted on Tableau public. This is a great straightforward process from Tableau that it has a public server were registered users can post their work and offers public view to any body within the community, sort of like YouTube for dashboards and scorecards. The tools can be ran on any browser on the market. Google Chrome is highly recommended because it has been tested well with the tool. Tableau runs on javascript so any browser that has javascript enabled can run the tools without any issues.

5. **Usage Documentation : Tutorial Details on** https://youtu.be/OM_GdRCHrvY

Google Chrome browser is recommended for the tableau application developed for the project.

Tool #1: Song recommendation application works with user selecting the favorite music decade. Check figure 1 for decade selection.
https://public.tableau.com/profile/ronald.xu#!/vizhome/Zyrus/Sel_Decades_DB

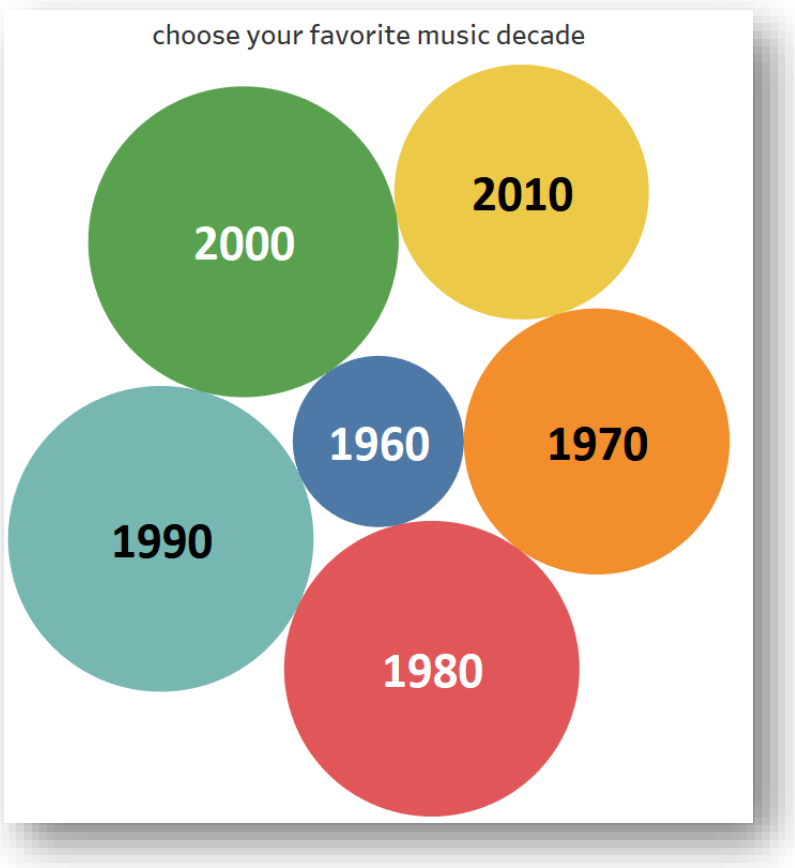


Figure A: Zyrus Song recommendation application

After selecting the decade user gets to see the top 20 words used in the lyrics during that period. The screen also shows the artist and number of time artist has used the word selected. Check Figure B.

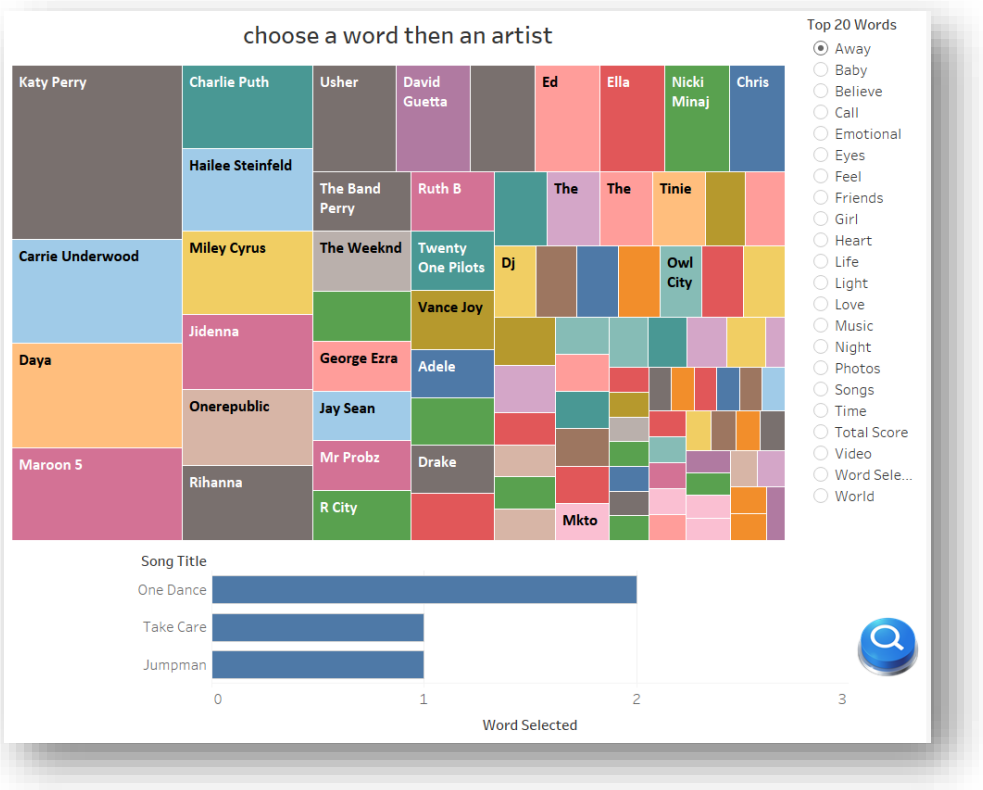


Figure B: word and artist selection then the system generates recommended songs based on user criteria defined.

Tool #2: Dashboard for 1965 - 2017 Billboard Word Cloud

https://public.tableau.com/profile/ronald.xu#!/vizhome/BillboardSongsDashboard/Dashboard

Word count dashboard is exploratory dashboard for any music lover. User gets options to select the singer and the lyrics, score of singer for most frequent words used in the period 1965-2017. Filtering is also available to select the decade and the artist.

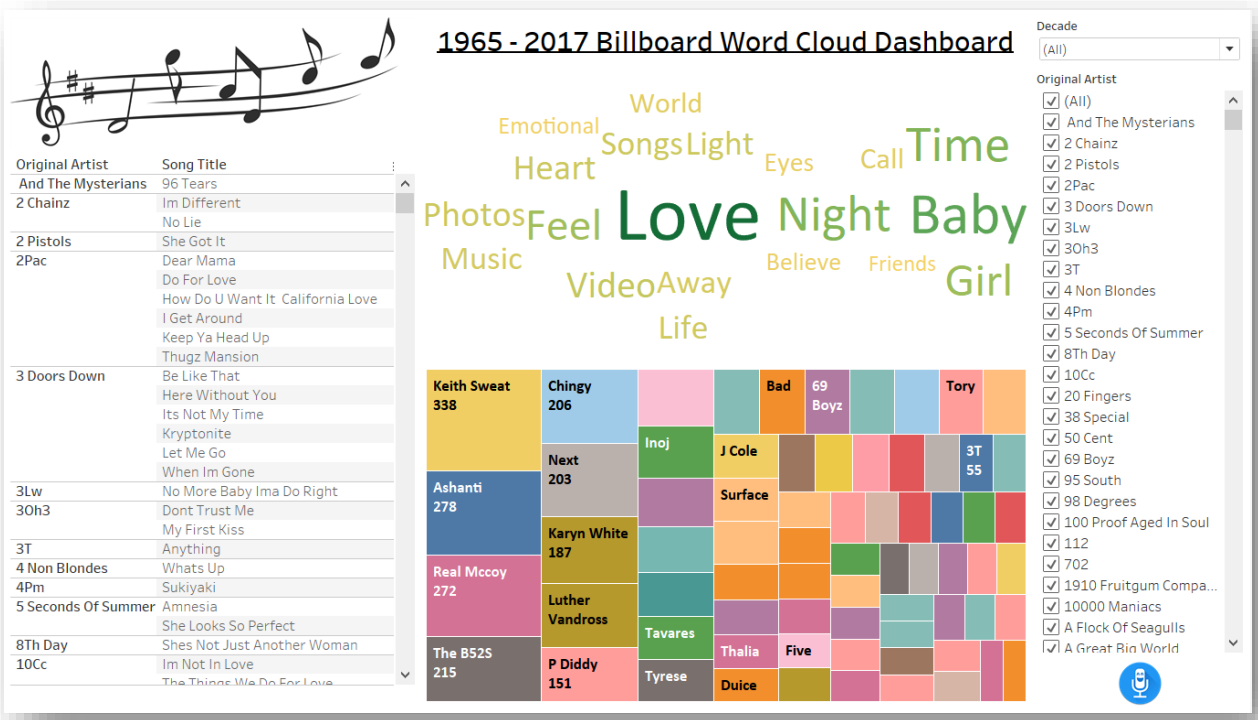


Figure C: Billboard Word Cloud Dashboard for Song Exploratory purposes

5. Common FAQs:

Who will benefit from such a tool?

All end users, people that love music or simply want to explore the lyrics can benefit from this tool. The tool is not limited to any sort of end users simply those who enjoy exploring and being adventurous.

What techniques/algorithms will you use to develop the tool?

- utilizing R to do data scraping from 3 major sources to gather all the necessary data like song titles, year, artist, lyrics and such
- using knowledge gained from MP1-MP3 and ETL processes to transform the data and load into a front-end BI tool to further clean the data for presentation layer
    - Lemur-stopwords library
    - Porter2 Stemwords library
- using tableau to do development needs and filtering for recommendations and dashboard abilities.
    - Any recommendation is better than none

Appendix:

Related works: https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf

R script and Documentations on GItHUB: https://github.com/jjasinski66/CS410FinalProject

Stop Words Repository: https://raw.githubusercontent.com/meta-toolkit/meta/master/data/lemur-stopwords.txt

Stem Words Repository: http://snowball.tartarus.org/algorithms/english/diffs.txt

Project Tutorial Videos: https://youtu.be/OM_GdRCHrvY

Song Recommender System:
https://public.tableau.com/profile/ronald.xu#!/vizhome/Zyrus/Sel_Decades_DB

1965 – 2017 Word Cloud Dashboard for Exploratory:
https://public.tableau.com/profile/ronald.xu#!/vizhome/BillboardSongsDashboard/Dashboard