CMSC 12200 Project

**Team “Maroon 4”:**
Will Hwang <heeseung@uchicago.edu>
Yongju Kim <yongjukim0128@uchicago.edu>
Jeong Whan Lee <zycon@uchicago.edu>
H.I. Park <hyunin@uchicago.edu>

**Goal:** Use historical Billboard Top 100 songs to compare lyrics to those of different artists

**Description:** As fervent listeners to pop music, some of our team members have been writing lyrics in their free time. We have been interested in creating a dynamic application that allows us to analyze different aspects of the written lyrics - and possibly compare them to established artists. To create this application, we decided to gather lyrics from historical Billboard Hot 100 chart, and then use the dataset to analyze the similar artist, period and genre of the input lyrics.

**User Input:**

Any lyric that is longer than a certain threshold (to be comparable to other lyrics)

**Software Output:**

1. Interactive lyrics analysis - Take the user’s lyric as input, and the software analyzes it to find artist / genre / era of which lyric style the input most resembles

2. Lyrics data summarization across genre / era - leveraging the collected song / lyrics data, we will also summarize trends of frequently used vocabulary in lyrics across genre and era. This is planned to be provided to users through interactive graphs.
   (This may involve implementing R within Python code)

**Data for Analysis:** Lyrics of “popular” songs.
To collect the data, we first collected the data of the Billboard weekly top 100 from 1958 to 2019.
This data was collected by data.world contributor Sean Miller.
<https://data.world/kcmillersean/billboard-hot-100-1958-2017>

The data set has several problems:
1. There are redundant songs due to the fact that many songs last for multiple weeks on the chart - The redundant data can simply be removed

2. The song itself might include many artists such as featuring artists - we will simply be regarding the lyrics of the featuring artists as the lyrics of the main artist.

3. No lyrics are included in the dataset - we will be crawling azlyrics.com to obtain the lyrics. For the lyrics not available in the site, we will assign “N/A” to the songs. At the moment we believe there are only a few N/As, thus we will either 1) use another crawling source for “N/A songs” or 2) ignore them if the number is significantly small.

**Process:**

A. **Data Gathering & Processing - Preliminary Task**

- Inspect Billboard dataset to find and handle defective data rows (e.g. N/A for certain variables) as well as redundancy (i.e. same song on the chart for several weeks)
- Extract artist names and song titles; process them into a usable form for website crawling
- Match lyrics data gathered from web crawling to the songs data and store them in an appropriate format for the main task
- At this step, we are considering creating a class for song with attributes such as title, artist, lyrics, genre, week, etc.

B. **Text Similarity - Main task**

- Packages to consider - Word2Vec, Doc2Vec, Sentence2Vec, NLTK, Plagiarism
- Create a "similarity" score calculator (i.e. Text similarity metrics) that quantifies the association of two strings in terms of word frequency, grammatical structure, sentiment, etc.
- Calculate the similarity score of user input to the lyrics/songs of existing artists
- Return the name of the artist with the highest similarity score to the user input
- Return the genre/time period with the highest similarity score to the user input
- Inputs to consider for similarity score - word frequency, grammatical structure, sentiment. We will weight each category differently.

C. **Additional Analysis on Word Frequency - Side task**

- Examine the change in word usage across time
- Compare word frequency across different genres/time periods