Taylor Lundeen
December 11th, 2017
Dr. Teplovs
SI 330

SI 330 Final Report*: Analyzing the Polarity of Popular Music Over the Last 50 Years*

1) **Motivation**-

The correlation between music and society is something that has always been very interesting and important to me. In planning for this project, I knew I wanted to find a way to analyze a dataset that somehow involved music. I was lucky to find the billboard_lyrics_1964-2015.csv file on Kaggle. This CSV file contains all of the Billboard Top 100 Songs and their corresponding rankings, artists, and lyrics from the charts of 1965-2015. Once I found this dataset, I was really intrigued by doing some sort of text processing on the lyrics to spot any trends in song lyrics within the same decade. After researching the various methods by which I could analyze text, I decided analyzing the sentiment of the lyrics could provide insight into the overall tone of the music of each of the last five decades. With this project, I tried to assess the polarity of popular music over the last fifty years. Answering the question: *What is the polarity of the lyrics of the top 1000 songs from the past five decades, and what trends can be spotted in lyric polarity over time?*

2) **Data Sources-**

To answer the above question via a Python script and a dataset, I employed the lboard_lyrics_1964-2015.csv file and the TextBlob Python Library.

- CSV File (https://www.kaggle.com/rakannimer/billboard-lyrics) -

As mentioned earlier, the CSV file contains data regarding the Billboard Top 100 charts for the years 1965-2015. I found and downloaded this dataset via Kaggle: (https://www.kaggle.com/rakannimer/billboard-lyrics) This file is of size 2 MB. This file contains the following variables for each song- Rank(integer representing the songs ranking on the charts for a given year), Song(string containing the song title), Artist (string representing the artist who sings the song), Year(integer representing the year the song was on the chart), and Lyrics (string representing the lyrics of the song). Overall, this file contains 5,101 records, each representing a song, and all of these records were used in calculating the overall polarity of the music of each decade.

- *TextBlob Python Library* (http://textblob.readthedocs.io/en/dev/quickstart.html)-

To analyze the sentiment of the strings representing lyrics from each decade, I employed TextBlob. TextBlob is a Python library that contains an API that can be used for various natural language processing methods. For the

purposes of my script, I used TextBlob's, textblob.sentiments module to return float values representing the calculated polarity score and subjectivity of the input strings. For reference, the polarity score is ranked as a float within the range [-1.0, 1.0], and subjectivity is ranked as a float on a scale of [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective. This Python library was used to process five strings, each representing a decade (or 1000 songs) worth of lyrics.

3) Data Manipulation Methods/Analysis Techniques

Tools Used:

- TextBlob Python Library
- Petl Python Library
- SQLite3

Overall Workflow & Manipulation Methods of the Script:

Manipulating the CSV file:

To begin manipulating the CSV file, I first wanted to export the records from the CSV file to a SQL database. I did so because I knew I wanted a way to easily query the data and eventually build a new table within an existing database to hold the results of the sentiment analysis on the lyrics. To create the database, I used the SQLite3 module to build a connection to a new database, 'billboard_top100.db', and created a new table titled Songs. This table holds a SQL generated primary key for each song, a string representing a given song's rank in the charts, a string representing each song's title, a string representing the corresponding artist of each song, a string representing the year the song was in the charts, and a string representing the lyrics of the song. Then, to export the records from the CSV file, I used the Python petl library to read the CSV file and write the records to a table, while also encoding the records for compatibility. I then used petl's cut function to cut out the fields I wanted from the table (Rank, Song, Artist, Year, and Lyrics). I then used petl to export the records from the table to the Songs table in the 'billboard_top100.db' database. I then used SQLite3 to select just the Song titles and their corresponding lyrics from the database for use in building a dictionary. I built lyrics_dict to associate each year (the keys) with a list of all the strings representing song lyrics for each of the top 100 songs of the year. I iterated through the return object (a list of tuples) of my SQL select statement to build this dictionary. I knew I then wanted to group the lyrics not only by year, but also by decade. I manually created lists of strings representing each year in each of the five decades of music I analyzed, to use as an input to the function analyze_sentiment.

Using TextBlob:

The analyze_sentiment function takes the decade_list input, and iterates through lyrics_dict to identify all of the lists of song lyrics associated with a given decade and join them into one large string. The function then creates a TextBlob object- using the TextBlob library- and calls the textblob.sentiments module to

return a tuple containing each decade of lyrics' polarity and subjectivity scores. I then created a dictionary, sentiments_dict, whose keys are strings representing the range of years within each decade, and whose values are the tuple of polarity and subjectivity scores returned by the analyze_sentiment function.

Other manipulation measures:

After creating the sentiments_dict, I again used SQLite3 to build another table, Sentiments, within the 'billboard_top100.db' database. This table has the following fields: decade (a string representing the range of years within each decade), polarity (an integer representing the polarity score of the lyrics of the corresponding decade), and subjectivity (an integer representing the subjectivity score of the lyrics of the corresponding decade). To insert values into this table, I iterated through the sentiment_dict and executed the insert statement on each key and value pair.

Preparation for Visualization:

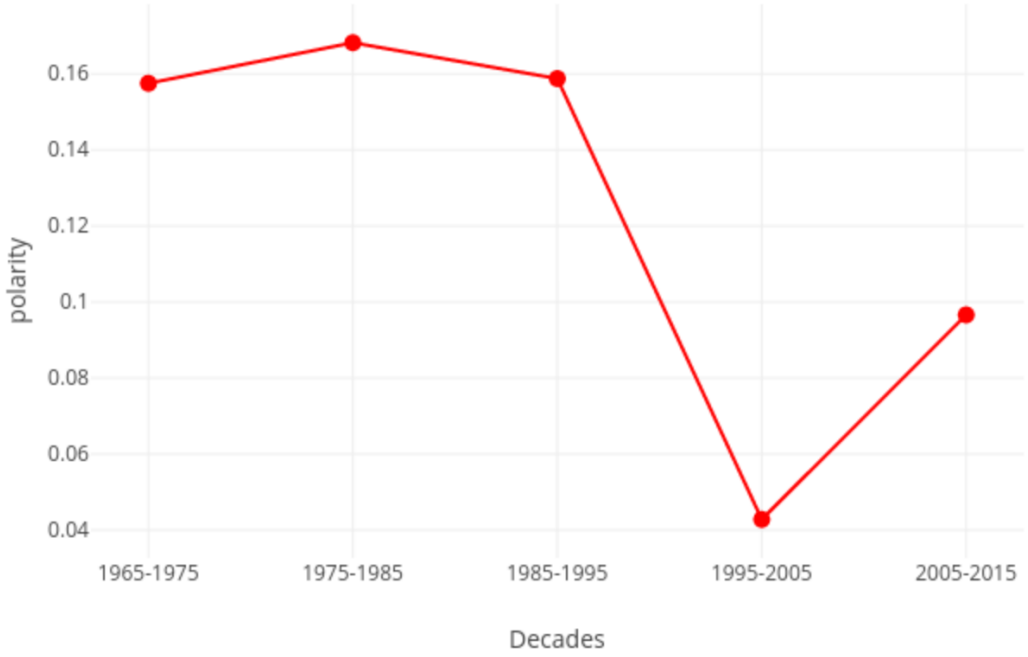
Finally, in preparation for using the Plotly API to create a plot visualizing my findings, I queried the SQL 'billboard_top100.db' database to select and sort the polarity scores for each decade by year, as well as selecting the decade ranges.

What challenges did I encounter and how did I solve them?

Originally, I had planned to use IBM Watson's sentiment analysis API to perform the analysis on the lyrics. In attempting to do so, I ran into major encoding issues involving the transformation of a text file into JSON formatting. I tried to remedy this issue a number of ways with no success. In the interest of time, I decided to take a different approach and began searching for other sentiment analysis tools for Python. While TextBlob allowed me to accomplish the goal of this project, its analysis focused on polarity and subjectivity, rather than the emotions detected in the tone of the text, which I had originally planned on analyzing using the IBM Watson API.

4) **Analysis and Visualization**

As mentioned in the workflow description above, I created a SQL database and tables to query and create one resource to compare the song lyrics of five decades and their polarity and subjectivity scores. As most music is subjective in nature, I decided that the polarity scores were much more indicative of the tone of the music and the times. To show the correlation between the polarity of music and the time frame within which it was made, I used the Plotly API to plot a line graph of the polarity associated with each decade. Here is the plot:



As you can see in the above plot, the polarity of music between the 1960's and 1990's have high polarity scores, then we see a dip in the mid 1990's with a slight increase around 2005. With the definition of polarity being "the presence or manifestation of two opposite or contrasting principles or tendencies," (Dictionary.com) it is interesting to see high polarity scores in decades that did indeed see a lot of contrasting views. For example, much of the music produced throughout the 60's and 70's either reflected sentiments about critical events like the Vietnam War while lighter tones came out of the psychedelic and disco oriented artists and music.

Furthering this project/What didn't work:

If I were to go further with this project, I would really like to explore the correlation between a decade's tone in lyrics and the major events, movements, and overall sentiments of the decade outside of music. While it is interesting to assess the polarity of music as it can tell us a lot about the similarity or dissimilarity in the perspectives of artists during a certain time period, I would like to go further by analyzing the multitude of tones and emotions that can be found within lyrics. This was my initial goal, but due to the issues I ran into with the IBM Watson API, I unfortunately could not analyze the data in this way. However, I had a lot of fun with this project and I will continue to use this dataset to make insights into how popular music can reflect the overall emotions that encompass a time period and its major events.