Sentiment Analysis on Top Billboard Songs

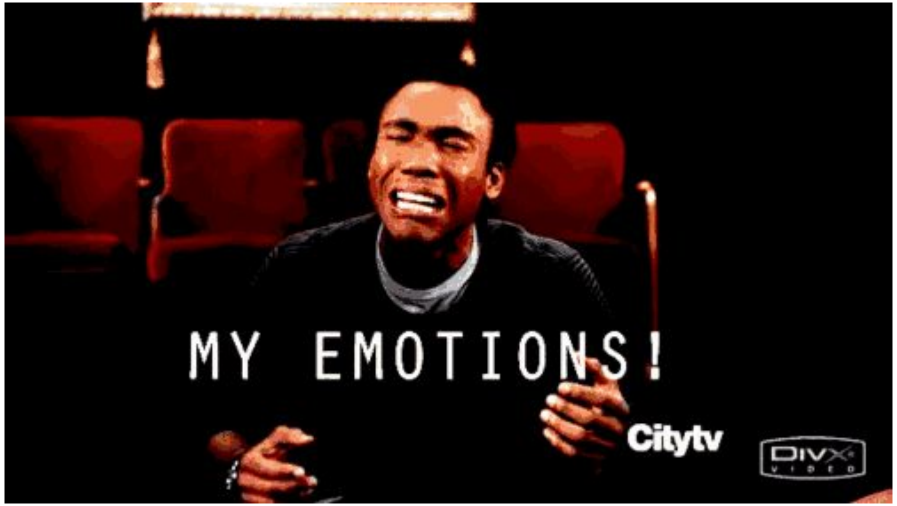
Maggie Tjia

Introduction

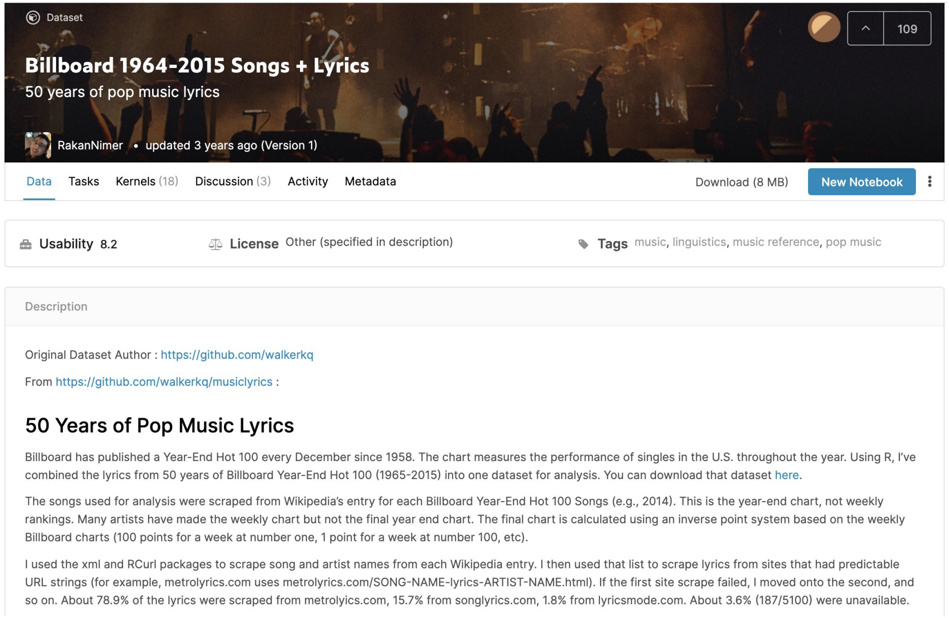
- Why did I pick this topic?

- Times are tough!

Overall Architecture



Dataset Description



Dataset

Billboard 1964-2015 Songs + Lyrics

50 years of pop music lyrics

RakanNimer • updated 3 years ago (Version 1)

109

Data Tasks Kernels (18) Discussion (3) Activity Metadata Download (8 MB) New Notebook

Usability 8.2 License Other (specified in description) Tags music, linguistics, music reference, pop music

Description

Original Dataset Author : https://github.com/walkerkq

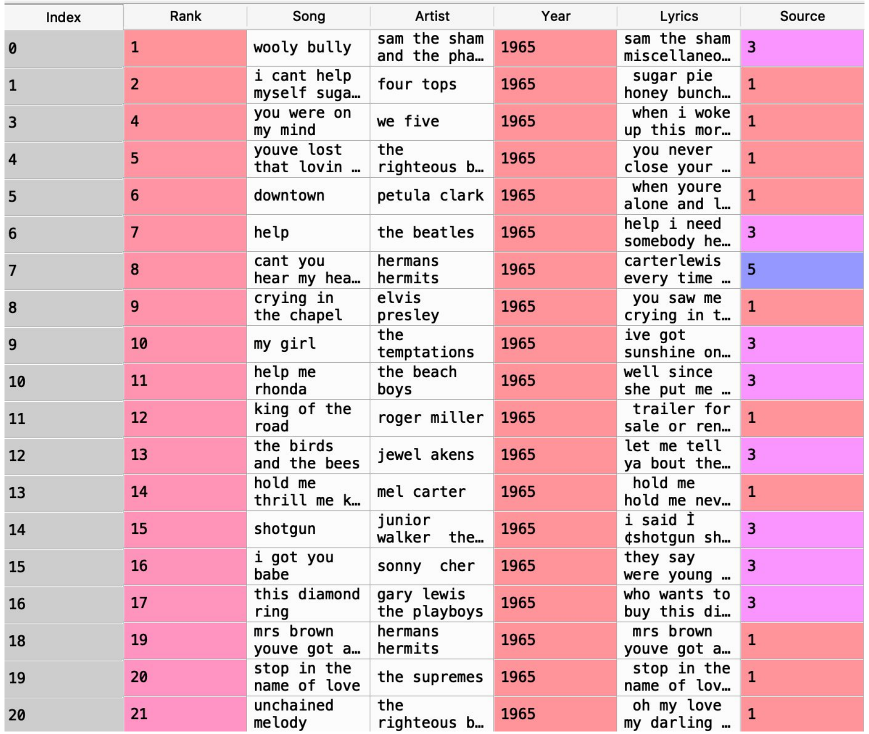
From https://github.com/walkerkq/musiclyrics :

50 Years of Pop Music Lyrics

Billboard has published a Year-End Hot 100 every December since 1958. The chart measures the performance of singles in the U.S. throughout the year. Using R, I've combined the lyrics from 50 years of Billboard Year-End Hot 100 (1965-2015) into one dataset for analysis. You can download that dataset here.

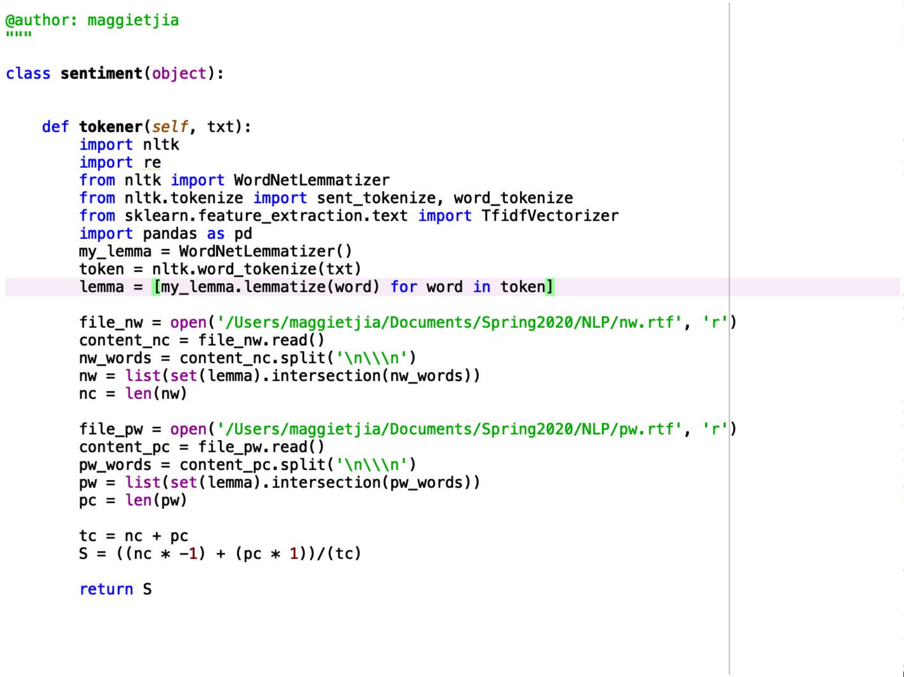
The songs used for analysis were scraped from Wikipedia's entry for each Billboard Year-End Hot 100 Songs (e.g., 2014). This is the year-end chart, not weekly rankings. Many artists have made the weekly chart but not the final year end chart. The final chart is calculated using an inverse point system based on the weekly Billboard charts (100 points for a week at number one, 1 point for a week at number 100, etc).

I used the xml and RCurl packages to scrape song and artist names from each Wikipedia entry. I then used that list to scrape lyrics from sites that had predictable URL strings (for example, metrolyrics.com uses metrolyrics.com/SONG-NAME-lyrics-ARTIST-NAME.html). If the first site scrape failed, I moved onto the second, and so on. About 78.9% of the lyrics were scraped from metrolyics.com, 15.7% from songlyrics.com, 1.8% from lyricsmode.com. About 3.6% (187/5100) were unavailable.

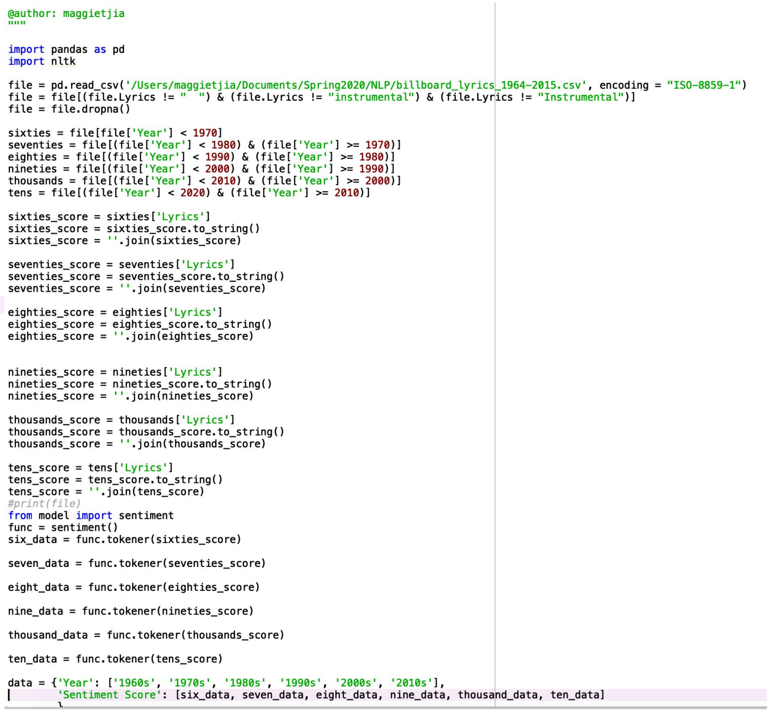


Index	Rank	Song	Artist	Year	Lyrics	Source
0	1	wooly bully	sam the sham and the pha…	1965	sam the sham miscellaneo…	3
1	2	i cant help myself suga…	four tops	1965	sugar pie honey bunch…	1
3	4	you were on my mind	we five	1965	when i woke up this mor…	1
4	5	youve lost that lovin …	the righteous b…	1965	you never close your …	1
5	6	downtown	petula clark	1965	when youre alone and l…	1
6	7	help	the beatles	1965	help i need somebody he…	3
7	8	cant you hear my hea…	hermans hermits	1965	carterlewis every time …	5
8	9	crying in the chapel	elvis presley	1965	you saw me crying in t…	1
9	10	my girl	the temptations	1965	ive got sunshine on…	3
10	11	help me rhonda	the beach boys	1965	well since she put me …	3
11	12	king of the road	roger miller	1965	trailer for sale or ren…	1
12	13	the birds and the bees	jewel akens	1965	let me tell ya bout the…	3
13	14	hold me thrill me k…	mel carter	1965	hold me hold me nev…	1
14	15	shotgun	junior walker the…	1965	i said Ì ¢shotgun sh…	3
15	16	i got you babe	sonny cher	1965	they say were young …	3
16	17	this diamond ring	gary lewis the playboys	1965	who wants to buy this di…	3
18	19	mrs brown youve got a…	hermans hermits	1965	mrs brown youve got a…	1
19	20	stop in the name of love	the supremes	1965	stop in the name of lov…	1
20	21	unchained melody	the righteous b…	1965	oh my love my darling …	1

Methodology



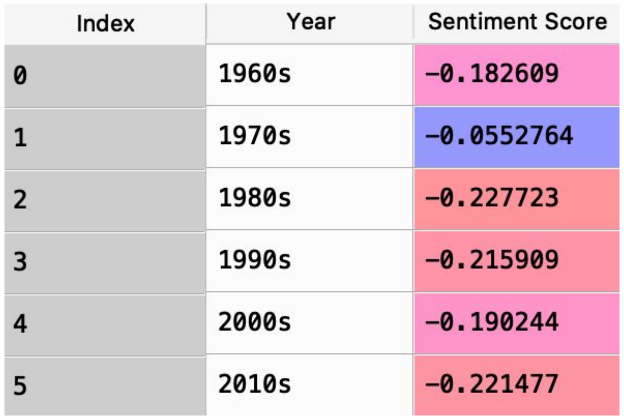
```
@author: maggietjia
"""

class sentiment(object):


    def tokener(self, txt):
        import nltk
        import re
        from nltk import WordNetLemmatizer
        from nltk.tokenize import sent_tokenize, word_tokenize
        from sklearn.feature_extraction.text import TfidfVectorizer
        import pandas as pd
        my_lemma = WordNetLemmatizer()
        token = nltk.word_tokenize(txt)
        lemma = [my_lemma.lemmatize(word) for word in token]

        file_nw = open('/Users/maggietjia/Documents/Spring2020/NLP/nw.rtf', 'r')
        content_nc = file_nw.read()
        nw_words = content_nc.split('\n\\\n')
        nw = list(set(lemma).intersection(nw_words))
        nc = len(nw)

        file_pw = open('/Users/maggietjia/Documents/Spring2020/NLP/pw.rtf', 'r')
        content_pc = file_pw.read()
        pw_words = content_pc.split('\n\\\n')
        pw = list(set(lemma).intersection(pw_words))
        pc = len(pw)

        tc = nc + pc
        S = ((nc * -1) + (pc * 1))/(tc)

        return S
```



```
@author: maggietjia
"""

import pandas as pd
import nltk

file = pd.read_csv('/Users/maggietjia/Documents/Spring2020/NLP/billboard_lyrics_1964-2015.csv', encoding = "ISO-8859-1")
file = file[(file.Lyrics != "  ") & (file.Lyrics != "instrumental") & (file.Lyrics != "Instrumental")]
file = file.dropna()

sixties = file[file['Year'] < 1970]
seventies = file[(file['Year'] < 1980) & (file['Year'] >= 1970)]
eighties = file[(file['Year'] < 1990) & (file['Year'] >= 1980)]
nineties = file[(file['Year'] < 2000) & (file['Year'] >= 1990)]
thousands = file[(file['Year'] < 2010) & (file['Year'] >= 2000)]
tens = file[(file['Year'] < 2020) & (file['Year'] >= 2010)]

sixties_score = sixties['Lyrics']
sixties_score = sixties_score.to_string()
sixties_score = ''.join(sixties_score)

seventies_score = seventies['Lyrics']
seventies_score = seventies_score.to_string()
seventies_score = ''.join(seventies_score)

eighties_score = eighties['Lyrics']
eighties_score = eighties_score.to_string()
eighties_score = ''.join(eighties_score)


nineties_score = nineties['Lyrics']
nineties_score = nineties_score.to_string()
nineties_score = ''.join(nineties_score)

thousands_score = thousands['Lyrics']
thousands_score = thousands_score.to_string()
thousands_score = ''.join(thousands_score)

tens_score = tens['Lyrics']
tens_score = tens_score.to_string()
tens_score = ''.join(tens_score)
#print(file)
from model import sentiment
func = sentiment()
six_data = func.tokener(sixties_score)

seven_data = func.tokener(seventies_score)

eight_data = func.tokener(eighties_score)

nine_data = func.tokener(nineties_score)

thousand_data = func.tokener(thousands_score)

ten_data = func.tokener(tens_score)

data = {'Year': ['1960s', '1970s', '1980s', '1990s', '2000s', '2010s'],
        'Sentiment Score': [six_data, seven_data, eight_data, nine_data, thousand_data, ten_data]
```

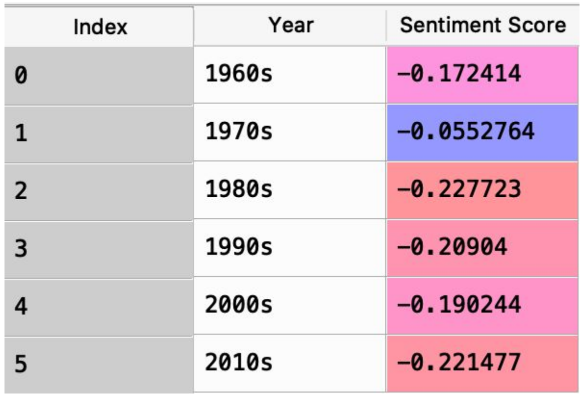
Results

Tokenize only



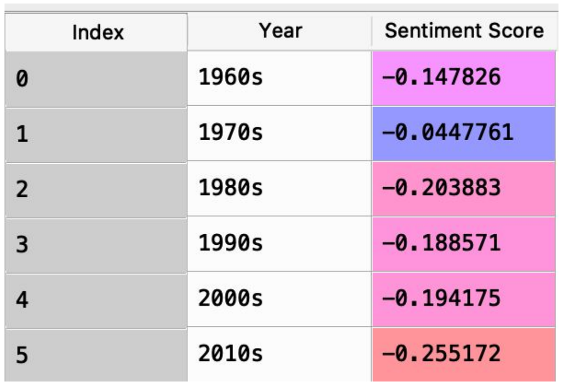
Index	Year	Sentiment Score
0	1960s	-0.182609
1	1970s	-0.0552764
2	1980s	-0.227723
3	1990s	-0.215909
4	2000s	-0.190244
5	2010s	-0.221477

Tokenize + Lemmatize



Index	Year	Sentiment Score
0	1960s	-0.172414
1	1970s	-0.0552764
2	1980s	-0.227723
3	1990s	-0.20904
4	2000s	-0.190244
5	2010s	-0.221477

Tokenize + Lemmatize + Remove Stopwords



Index	Year	Sentiment Score
0	1960s	-0.147826
1	1970s	-0.0447761
2	1980s	-0.203883
3	1990s	-0.188571
4	2000s	-0.194175
5	2010s	-0.255172

Likes vs Dislikes

- Likes
  - Music
  - Decade difference

- Dislikes
  - Limited years
  - Not consistent in the number of songs for each decade

Conclusion