Song Analysis Pseudocode

Pseudocode Details

Please refer to these hints when attempting to implement the functions compute_idf, compute_tf, compute_tf_idf, compute_corpus_tf_idf, and nearest_neighbor. The idea behind having you think about implementing these functions on your own first is to deeper your understanding of how tf, idf, and nearest neighbor work in unison in order to create a tool that can compare songs by genre. Thinking about how to implement a function based on an algorithm is an important skill that you will continue to build as you continue on your journey as an ethical computer scientist!

Pseudocode: compute_tf

In order to calculate the Term Frequency,

```
Loop through each word of a song
    if the word has not shown up before
        add word to the data structure storing
        the song's words and their tf values
    Increment the word occurence by one
return the data structure storing the song's term frequencies
```

Pseudocode: compute_idf

In order to calculate the Inverse Document Frequency for each song in your corpus,

```
create a df data structure
Loop through every song in the corpus
    Get the lyrics for the song
    for every word in the song
        If the word has not been seen before
            store a new set for the word
            in your df data structure
        If the song has not exist in the set
            add the song to the set
For every word in your df data structure
    calculate the idf for the word
    and store it in your idf data structure
```

Pseudocode: compute_tf_idf

In order to calculate the Term Frequency - Inverse Document Frequency,

```
Compute the term frequency of a song
For each word in that song, multiply by the idf
Store the result where each word has its tf-idf value
```

Pseudocode: nearest_neighbor

In order to calculate the nearest neighbor of a novel song,

```
Keep track of the highest similarity and best song
Verify new song lyrics contain only valid characters
Calculate the tf-idf for the song
Loop through the corpus
    If the similarity of a song is higher than the current highest,
    update similarity and song
return the best song
```

Testing accuracy

In order to check the accuracy of the program, here are some lyrics to try to compare your classifier to ours:

small_songdata:

- Song(id=14881, title='meet-me-on-the-equinox', year='2009', artist='death-cab-for-cutie', genre='Rock')
- Song(id=51228, title='i-wish', year='2007', artist='atl', genre='R&B')
- Song(id=15212, title='piano-bar', year='1997', artist='billy-joel', genre='Rock')
- Song(id=15643, title='the-family-album', year='2007', artist='comecon', genre='Metal')
- Song(id=17690, title='high-addal-vs-mida', year='2016', artist='addal', genre='Other')
- Song(id=27958, title='nothing', year='2006', artist='flipper', genre='Rock')
- Song(id=20188, title='gotta-boyfriend', year='2009', artist='frankmusik', genre='Electronic')
- Song(id=43253, title='somebody-special', year='2006', artist='fish', genre='Rock')
include results from using small and full csv

full_songdata:

- Song(id=206413, title='ydioe-aneea-iiinodu', year='2006', artist='edaiaoidee', genre='Rock')
- Song(id=115640, title='i-wish', year='2008', artist='dmx', genre='Hip-Hop')
- Song(id=238141, title='art-of-the-blacksmith', year='2007', artist='enid', genre='Metal')
- Song(id=226545, title='the-star', year='2006', artist='america', genre='Rock')
- Song(id=126503, title='when-the-going-gets-tough', year='2006', artist='boyzone', genre='Pop')
- Song(id=328911, title='baby-baby-baby-baby-baby-baby-baby-i-love-you', year='2006', artist='brainpool', genre='Pop')
- Song(id=310763, title='frankie', year='2007', artist='betty-blowtorch', genre='Rock')
- Song(id=43253, title='somebody-special', year='2006', artist='fish', genre='Rock')