Personalization and Machine Learning
Mini-Project Report
By Marissa Beaty

A Content-Based Recommendation System For Receiving Song Recommendations From a Selected Book

Project Analysis

The goal of this project was to create a recommendation system that recommended songs based on a book, by using a dataset from Kaggle containing the book’s text and genres (Naren, 2019) and a self-created song playlist made on Spotify containing 78 total songs representing three songs from every major genre grouping (Wikipedia, 2023). The foundation for the project came out of the Personalization and Machine Learning Week 3.1 class. Due to the time available for this project, I limited the scope to obtaining song recommendations from one given book. The book selected is Jane Auston's "Persuasion." As a quick note, much of the determination for how these features were translated comes out of personal knowledge obtained through an undergraduate degree in English Literature. Though these translations are not a perfect replication of the feature's defined for a song, they introduce the types of thinking that could yield successful recommendations with this type of system.

The first task for this project was to determine the song features that will be translated to fit the structure and set-up of a novel and my book's dataset. I narrowed down my search to four features: Energy, Valence, Instrumentalness, and Speechiness. Speechiness (Feature 1) and Instrumentalness (Feature 4) were the first features I began translating.

Spotify defines speechiness as the frequency of words in a song. Rap songs, for example, have a high speechiness score, whereas orchestral songs have a low speechiness score. I translated this to mean wordiness in a text, which describes text that uses too many words for simple phrases. In doing so, a lot of the book’s words have no meaning. I first thought of defining wordiness by calculating a percentage of unique words against the total word count. This resulted in a 7% relationship. After additional thought, I realized this calculation did not quite fit the definition of wordiness. Wordiness does not consider unique words but rather the words that carry the most significance: words that removed from the text reduce its coherence. Instead, I decided to calculate the ratio of stop words to total words as stop words are exactly that: words that cannot be removed without losing meaning. I used the NLTK list of stopwords for this task but adjusted it to include alternatives to combat wordiness (Fordham, 2023). This new stopwords list allowed me to collect the words that always carry significance to the story and text. This ratio of stopwords to total words gave a percentage close to 50%. Though this is closer to what I expected, I used both to get recommendations to compare the results. Though I do think this second iteration of speechiness was a bit more accurate, I would be interested in furture iterations of this project to pursue alternatives to this process, such as calculating speechiness by way of how lyrical the prose. Highly lyrical prose could be like musical songs or rap songs, whereas non-lyrical prose could be like podcasts. I think this could be an interesting alternative to my algorithm, even though it not possible with my time or resources.

Whereas Speechiness defines how many words are in a song, Instrumentalness defines whether a song contains any words at all with a value inverse of that produced by Speechiness. In instrumentalness, a low score means the track contains many vocals, and a high score means the track contains no vocals. This interpretation for a book is tricky as all books - other than picture books - contain some form of 'vocals'. I interpreted I defined this as a ratio between the total number of characters in our book against the book with the longest number of characters (Guinness World Records, 2023). I selected this as it calculates a representation of all values in the text, like how instrumnetalness collects all forms of vocals. For my selected book, this ratio was 7%. I realized, however, that this definition of instrumentalness failed to account for the fact that instrumentalness recorded words and not just sounds. To accommodate this, I decided to instead calculate instrumentalness by a ratio of the texts total word count against the average book word count. To do this, I researched the average word count of various book types and used that to create a series of 12 buckets. These buckets ranged in values from 0 to 150,000, which is the largest recommended word count for a novel (Sambuchino, C., 2021). The value for instrumentalness was found by fitting our books word count into a bucket, finding the ratio between our book's word count and the maximum word count of that bucket, and multiplying it by a percentage of which bucket it is in. With our 12 buckets, this percentage ranged from 0, 1/11, 2/11, up to 10/11. Any word count in our last bucket would have a value of 1. I first calculated this using a range of values for each bucket defined in the same article by C. Sambuchino. This resulted in an instrumentalness score of 61%. Out of curiosity, I also calculated this using evenly distributed buckets. This results in an instrumentalness score of 52%. Both were used to acquire recommendations.

I selected this somewhat complicated process for Instrumentalness as I believe these sticks closest to calculating a good ratio of total words in "Persuasion" against total words expected in other books, thus, giving as close to a book understanding of instrumentalness as I could define. I think this understanding of instrumentalness could vary and thus vary the results provided. I would be interested to see how others re-creating this project alter their variations and how those results in different recommendations.

The next feature I defined is Valence (Feature 2). Valence was quite easy to interpret as it identifies the feeling of the song. Songs with a low valence are interpreted as sad. Songs with a high valence are happy. To determine this in our book, I used a pre-trained sentiment analysis called Flair. Flair produced a sentiment value of .626 or 62%. Having read "Persuasion" several times, this value seemed accurate and was used as our valence value for our recommendations.

The last feature I determined was Energy (Feature 3). This value was the most difficult to determine. Energy for a song is defined by a song's intensity. If I had more time, I would determine energy by developing a machine learning algorithm that defined energy by scanning the book's text and determining points of plot change, moments of intensity or excitement, or dialogue that includes terms of action. As you can imagine, this is worthy of a final project in and of itself. Because I lack the time and resources to do this, I devised an alternative that calculated the cosine similarity of the word vectors of the book's genres to the word vector for energy. I first calculated the cosine similarity for all genres listed in our book's dataset so I could pull out the genres for "Persuasion" more easily. Doing this required extensive cleaning of the data as several genres were not in the correct format to determine their word vectors. Once the data was cleaned, I calculated the average 'energy' for "Persuasion's" genres by finding the mean of each genre's cosine similarity value. This resulted in an 'energy' value of 70%. Though this does seem high given my familiarity with the novel, this was the best result I could determine given the algorithm I defined.

After I had my values, I compared my book to the features pulled out of the songs in my playlist. I created four book variations, accounting for the various options produced for

Speechiness and Instrumentalness. The recommendations for all four books are in the last part of the notebook. As you can see, the results are quite interesting when purely looking at the names of the songs. Our original "Persuasion" recommendations had values such as A$AP Rocky's "Lost and Found" and Blackstreet's "No Diggity." These are not what I think of when I think of 19th century literature. When looking at their data, however, the recommendations appear spot-on with the values for each book. For example, when we alter our Speechiness score to get "Persuasion 1" similarities, all recommended songs have a low value of speechiness.

The same is true when we altered our instrumentalness score. All recommended songs now had similar instrumentalness values to our books'. What this tells us, is that the recommendation system itself is not inaccurate, rather the values we found for each feature for our book is. The alternative interpretation, however, is that our understanding of book and song associations is highly subjective to how that book makes us feel (a feeling not even sentiment analysis can determine). When we think of Jane Auston, we think of period-pieces, ballrooms, classical instruments, and dress. Many of these things cannot be defined by a song feature. In this case, it is not that our song recommendations are wrong, but our perception of them is clouded by our pre-conceived notions of what a Jane Auston novel is and encompasses.

It is known that Spotify uses all features of a song to make the most accurate recommendations. Though not every feature can be translated to fit a book, I would be curious to attempt this project again defining a few more features and comparing the results to what I produced with this project. I believe that more features, in addition to more complex definitions for features like 'Energy' would create different results and may potentially provide results closer to those expected.

Despite these potential alterations to the project, I produced the recommendation system I set out to do. For that, I am quite satisfied with the projects results. I set out to create a recommendation system that would recommend songs from a book. While I do believe there could have been some fine-tuning to increase the accuracy of the song recommendations, this project proves that it is possible to draw a link between the two and confronts us with our expectations and pre-conceived notions of a book. The results are not perfect, but they are interesting and thought=provoking. That is all I need to determine this project a success.

Bibliography

@GrabNGoInfo, A. (2023) 'Sentiment Analysis: Hugging Face Zero-shot Model vs Flair Pre-Trained Model'. *Medium.com.* Available at: https://medium.com/@AmyGrabNGoInfo/sentiment-analysis-hugging-face-zero-shot-model-vs-flair-pre-trained-model-57047452225d#:~:text=The%20Flair%20pre%2Dtrained%20sentiment,quite%20different%20from%20IMDB%20data (Accessed: 23 May 2023).

Akbik, A. (2018) 'A very simple framework for state-of-the-art Natural Language Processing (NLP)'. *Github.com*. Available at: https://github.com/flairNLP/flair (Accessed: 23 May 2023).
Author Learning Center. (2020) 'Word Count by Genre: How Long Should a Book Be?' *authorlearningcenter.com.* Available at: https://www.authorlearningcenter.com/writing/managing-your-writing-life/w/goal-setting-and-process/7102/word-count-by-genre-how-long-should-a-book-be (Accessed: 28 May 2023).

Beaty, M. (2023) ‘All The Sweet Spots’. *Spotify Playlist*. Available at: https://open.spotify.com/playlist/6RPqf05XTHdRlTtkEc0edA?si=eda15523beb74a41 (Accessed: 1 June 2023).

Beaty, M. (2023) ‘All The Sweet Spots Shortened’. *Spotify Playlist.* Available at: https://open.spotify.com/playlist/4eIxXnvi5KeTMPAzcgwSud?si=d057c60b2e6e4c4d (Accessed: 12 June 2023).

Fordham University. (2023) ‘Wordiness’. *Logic and Rhetoric*. Available at: https://www.fordham.edu/academics/academic-resources/writing-center/writing-resources/logic-and-rhetoric/wordiness/#:~:text=Wordiness%20most%20often%20occurs%20when,tightened%20without%20loss%20of%20meaning (Accessed: 5 June 2023).

Guinness World Records (2023). ‘Longest novel’. *Guinnessworldrecords.com.* Available at: https://www.guinnessworldrecords.com/world-records/longest-novel#:~:text=A (Accessed: 24 May 2023).

Naren. (2019) ‘Goodreads’ Best Books Ever’. *Kaggle Dataset.* Available at: https://www.kaggle.com/datasets/meetnaren/goodreads-best-books (Accessed: 17 April 2023).

Numpy Library. (2023) ‘numpy.add’. *numpy.org*. Available at: https://numpy.org/doc/stable/reference/generated/numpy.add.html (Accessed: 19 May 2023).

Pandas Library. (2023) ‘pandas.DataFrame.’ *pandas.pydata.org*. Available at: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html (Accessed: 20 April 2023).

Sambuchino, C. (2021) ‘Word Count for Novels and Children’s Books: The Definitive Post | How Long Should a Book Be?’ *Writer’s Digest*. Available at: https://www.writersdigest.com/whats-new/word-count-for-novels-and-childrens-books-the-definitive-post (Accessed: 28 May 2023)

Scipy Library. (2023) ‘scipy.spatial.distance.cosine’. *dosc.scipy.org*. Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html (Accessed: 27 May 2023).

Spotify. (2023)’Get Track’s Audio Features’. *Spotify for Developers.* Available at: https://developer.spotify.com/documentation/web-api/reference/get-audio-features (Accessed: 20 March 2023).

Stack Overflow. (2019) ‘Filter dataframe matching column values with list values in python [duplicate]’. *Stackoverflow.com.* Available at: https://stackoverflow.com/questions/53082014/filter-dataframe-matching-column-values-with-list-values-in-python (Accessed: 15 May 2023).

Wikipedia. (2023) ‘List of Music Genres and Styles’. *Wikipedia.com*. Available at: https://en.wikipedia.org/wiki/List_of_music_genres_and_styles. (Accessed: 20 May 2023).