Music To Book Recommendation System

Kovida Samir Kshatri
kovida22106@iitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Mitali Chakrabarty
mitali22039@iiitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Smiti
smiti220726@iitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Smiti Singhal
smiti22073@iitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Swati Jha
swati22080@iitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

Mehak Bansal
mehak22111@iitd.ac.in
Indraprastha Institute of Information Technology
Delhi, India

ABSTRACT

The recommendation systems of today primarily focus on only one type of content, such as books, movies,music,news etc. However, there is a lack of platforms that can provide recommendations across domains, such as music to book and vice-versa. The cross-domain recommendations could potentially provide diverse recommendations, with an element of surprise in it. In this project, we aim to use various machine learning and natural language processing techniques to implement the music to book recommendation system. This could help new readers to find books that could match their tastes and preferences. Beyond this, we have also built a book to book recommendation, engine,which could help the book enthusiasts to further expand on their current interests. We have also built an interactive web interface for the easier accessibility of the users to our music to book recommendation engine.

KEYWORDS

Information Retrieval, Logistic Regression, Light GBM

ACM Reference Format:
Kovida Samir Kshatri, Mitali Chakrabarty, Smiti, Smiti Singhal, Swati Jha, and Mehak Bansal. 2023. Music To Book Recommendation System . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The ever-increasing growth of internet has affected all dimensions of our lives, and specially the way we consume content. Most importantly, the internet has taken the entertainment industry by storm, which we, can see from the rise of platforms such as Netflix, Spotify, Prime, Gaana.com,Amazon Kindle store etc. All these recommender systems essentially focus on the content consumption history of the users and accordingly recommend content to them. Mostly, all the recommender systems, across different domains, work essentially on the same type of algorithm. However, they focus on only one type of content. The advent of the cross-content recommendation systems could help to recommend content across various content. This could help improve the diversity and personalization component of the recommendations generated and help to increase the retention rate of the users. We specially focus to implement the music to book recommendation system in this project, by using machine learning algorithms such as Light GBM, logistic regression, and natural language processing techniques such as cosine similarity and word embedding.

2 PROBLEM STATEMENT

The project aims to develop a music to book and book to book recommendation system with an interactive web interface on a model trained on LightGBM for the music dataset and Logistic Regression and OneVsRestClassifier for the book dataset after performing Word Embedding and Cosine Similarity for genre classification. The system will work for both English and Hindi Music.

3 MOTIVATION

With this project we intend to present a cross - domain recommendations system that takes descriptions of music and books and recommends users on the basis of similarity by employing various data processing techniques and machine learning algorithms. The idea behind this is that people who have a particular taste in music are likely to enjoy books catering to similar interests. This can be an incredibly helpful tool for the non-book readers, who spend considerable time on music streaming platforms. Personalised book recommendations can help them to inculcate a habit of reading. For our bibliophile netizens, who also spend some time on these platforms, the diversified recommendations can help them to expand on, and beyond their interests. For the authors and publishers, this can be a way to better understand their audiences and create targeted marketing campaigns, which, in turn, would help the users to discover relevant recommendations, and increase their engagement with the platform.

4 LITERATURE REVIEW

Penha et al.[1] analysed the amount of factual knowledge that is already stored in the parameters of the heavily pre-trained BERT

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

transformer, in the domain of books, movies, and music recommendations. They studied how much a simple BERT, without any fine-tuning, knows about recommendation items. This was tested by using different probes which require different types of knowledge, i.e., context-based and collaborative-based. These two pieces of knowledge were then infused in BERT during the fine-tuning step to help develop a conversational recommendation system.

G. Ramakrishnan et al.[2] reviewed the collaborative filtering techniques for a book recommender system, wherein instead of using the user ratings explicitly, the idea was to incorporate implicit information in the user ratings and the similarities between the items rated. The user ratings were assigned as a weighted average of all the similar book ratings. For books with no other book of significant cosine similarity, a standard deviation with the average rating was used. Based on RMSE, the technique presented proves to be better than other methods.

Avi Rana and K. Deeba[3] presented an online book recommendation system that uses collaborative filtering with Jaccard similarity (measures similarity between the preferences of different users in the collaborative filtering algorithm, i.e., the ratio of the size of intersection of the two sets of books to the size of union of two sets). The system collects user's ratings and preferences to books, then uses Jaccard similarity to evaluate the similarity between users in this system, based on which, a list of books is generated by the system which may be of interest to that user whose Jaccard similarity was just evaluated.

Cho et al.[4] compared 3 approaches for creating an efficient book recommendation system: (i) content-based, (ii) collaborative filtering, and (iii) hybrid approach. The hybrid system used the confidence levels of the results predicted by both the former approaches and performed their weighted sum to arrive at a result. They used reviews from various platforms to curate their dataset. The final results indicated that the hybrid model gave better results as it reduced the issues that both the approaches would otherwise face individually.

Kaminskas et al.[5] discusses various tools and techniques to address the research challenges posed by context-aware music retrieval and recommendation. It covers topics from classical music information retrieval and recommender system techniques to contextual music retrieval, emotion recognition in music, and social computing. It also considers user-related context such as activity, demographics, and emotional state. The approaches that relate music to context are data driven and are dependent on multiple fields.

N. Pelchat et al.[6] covered how various NNs can be used to classify songs according to their genres. They improvised upon a CNN for the same. The input to the CNN was short time segments of the spectrograms of the songs.It had six convolutional layers followed by a fully connected layer and then a softmax function. The results obtained were 85

Singh et al.[7] released a new dataset comprising different Indian Song genres, namely, Bollywood Rap, Bollywood Romantic, Ghazal, Folk(Garhwali), Sufi, Bhojpuri and Bhajan. The authors proposed an automatic genre classification model for Indian songs using machine learning. They compared the performance of different ML algorithms and concluded that the Light GBM classifier gives the best accuracy of 77.2

Hungund [8] proposed a machine learning model for genre classification of books by using multi label binarizer and Logistic Regression. The dataset used had 227 different types of genres.

5 NOVELTY

The content based recommender systems developed so far tend to have a single type of content as their cynosure, for example, movies, music, books, etc. This points to an apparent disconnect between various types of recommendation systems on the basis of content. The novelty of this system lies in its ability to connect two very different types of content: music and books and then utilize a person's musical preferences and interests and translate it into relevant book recommendations that could interest the user, thereby, enabling cross-domain recommendation from music to book. This idea of using music as the basis for recommending books is born out of the concept that oftentimes, it has been seen that a person tends to have a strong emotional and narrative connection with his/her musical taste. Hence, the system takes advantage of this fact and employs it to recommend books that can resonate with someone's preferences and inclination.

Beyond this, the system is an end-to-end platform that has been built to cater to both types of audiences: the book enthusiasts, as well as the new readers.It achieves this by integrating a music-to-book recommender as well as a book-to-book recommender.

6 DATABASE

We have used the following 3 datasets for our music to book recommendations systems

6.1 Music dataset

We have used the following two datasets to train our music recommender

- **Hindi Music Dataset** : This is a new dataset which consists of 7 different genres from Indian regional music : Bollywood Rap, Bollywood Romantic, Ghazal, Folk(Garhwali), Sufi, Bhojpuri and Bhajan. Each genre has around 100 different full length tracks.
- **English Music Dataset**|: We have taken the GTZAN dataset which consists of 10 different genres: Blues, Rock, Classical, Reggae, Disco, Pop, Hiphop, Metal and Jazz.Each genre has 100 different tracks of 30 seconds each.

6.2 Book dataset

We have used the CMU book summary dataset for this project. The dataset contains book summaries of more than 16000 books along with its metadata which includes details such as wikipedia ID, author, title, genres,publication date,freebase ID, etc.

7 METHODOLOGY

7.1 Music to Book recommendation

The following sections explain how the musical interests of the user are exploited to recommend books that might match his taste.

7.1.1 *Feature Extraction*

. After ingesting the audio files from the English and Hindi music dataset, we extract 12 essential features from each of the audio

files. These include: onset strength, zero crossing rate, chroma stft, tonnetz, chroma cqt, spectral rolloff, chroma cens, spectral contrast, melspectrogram, spectral bandwidth, mfcc and spectral centroid. These features can be extracted from shorter frames of the audio tracks and can further be integrated to analyse the temporal evolution of these features over the course of the track. The following features: spectral centroid, spectral roll off, Mel frequency cepstral coefficient (mfcc), zero crossing rate, can be broadly categorised as timbral features, which have been extracted from the frequency of the audio signal. Further, features such as chroma stft and chroma cens are extracted by utilising the pitch content of the music. Then, there are other features like tonnetz which helps to better analyse the relationship between musical pitches, and onset strength which helps to understand the strengths of the onsets in music , i,e., the points in the audio track where a new musical event occurs, such as, start or end of a note.

7.1.2 *Dataset preparation*

. The feature extraction renders us the numerical representation of each of the audio tracks which are stored in csv files as per their genre. These files are then concatenated to prepare the final database. This database would finally contain the 12 features for each song in the dataset and a label for each song, stating the genre to which the song belongs. After splitting the features and labels into X and y, this file acts as the training data for our classification model.

7.1.3 *Model training for music genre classification*

. Using the previous work of other people in this domain as our basis [7] and comparing the accuracy values given by various models, we select Lightgbm as our classification model. The model results in 64% accuracy on the test dataset, obtained from splitting the above-mentioned dataset into training and testing data.

LightGBM stands for Light Gradient Boosting Machine. It is an open-source framework for the purpose of gradient boosting. It employs tree-based decision algorithms. The model is scalable, flexible and achieves a higher accuracy than the other GBMs. Beyond this, the model is lighter, faster, possesses a wide variety of parameters and supports parallel learning.

7.1.4 *Mapping the genre of music to book*

. The next step is to map the genre of music to the book genres. We achieved this objective by using cosine similarity metric. Cosine similarity helps us to measure the similarity quotient between two nonzero vectors by considering them in the high-dimensional space. It measures the cosine of the angle between the vectors and ascertains if the vectors point approximately towards the same direction. It is an extremely helpful metric in text analysis and is often used for the purpose of measuring similarity within documents.

However, for using cosine similarity, we first needed to convert the genres into vectors. For this purpose, we have used the technique of word embedding. Word embedding is a NLP technique that helps to represent words as a vector of numerical values. They are especially helpful since they allow the words that are synonymous, to have similar representations. We have used the GoogleNews-vectors-negative300 word embedding model to convert the genres of music and book to word embedded representations. It is a pre-trained model that has been trained on a huge corpus which contained news articles. The Word2Vec algorithm was used to train this model. It has a vocabulary of 3 million words and phrases, and the dimension of each word vector in this model is 300.

After using the word embedding representation and applying cosine similarity between all possible pairs of vectors of music and books, we extract the top genres of book that share the maximum similarity with a particular genre of music.

7.1.5 *Processing the book dataset*

. For recommending the appropriate books, we would now ingest the book summaries dataset. Post this, we perform some essential data preprocessing. Firstly, we extract the following attributes from the dataset: Wikipedia ID, book title, book summary and its genres. Then, we perform some data cleaning by removing stop words and punctuations from the "book summary" and "genre" columns.

The most important step performed during the processing of the book dataset was to group the similar genres. The genre column is a multi-valued attribute and the dataset originally has 227 unique genres. It contained genres such as, Bangsian fantasy, Biopunk, Edisonade,etc. which gave a more granular description of the book genre, however they were irrelevant from the standpoint of the music to book recommender system, as we needed generic and broader genres that could be mapped to the music genre relatively easily and accurately. So, in order to remove the irrelevant genres and keep broader genres, we first performed a manual grouping of some of the genres. For instance, "Bangsian Fantasy" was replaced with "Afterlife Fantasy", "Biopunk" was replaced with "Biotechnology fiction" "Edisonade" was replaced with "Inventor fiction",etc. After this we mapped the remaining genres to the genre list that we were originally using when we were mapping the music genres to the book genres list. The book genres list contained the following genres: History, Poetry, Autobiography, Romance,Biography, Literary, Spirituality, Western fiction, Comedy, Adventure,Dark, Sociology, Comic, Anthology, Drama, Thriller, Action, Conspiracy, Catastrophic, Young adult,Fantasy, Travel, Science, Speculative, Horror, Mystery, Non fiction. As we can infer, this list contains broader categories and more conventional names of genres. So, now the book genres in the book summary dataset are mapped to these broader genres list , by using cosine similarity.

7.1.6 *Generating book recommendations basis the mapped genre*

. Finally, this is the step where we generate book recommendations. For a given audio file (it can be in any format, ex: mp3, opus, etc.), we would first predict the genre of the music. After this, using the music to book mapping created in 1.3, we would extract the corresponding book genres. Then, we would use the preprocessed book summary dataset, and render the name of those books that have the same genre as the ones rendered from music to book mapping.

7.2 Book- to -Book recommendation

We have built another model that would help us to recommend books on the basis of the user's existing reading pattern.

7.2.1 *Feature encoding*

. Since every book in the dataset could belong to multiple genres,

therefore, we have used the MultiLabelBinarizer that would convert the list of multiple labels to a binary label indicator matrix.

7.2.2 *Data preprocessing*

. We particularly focus on our "book summary" column within the dataset. We would extract this column from the original dataset, remove the stop words and punctuations, and would use it to train our model.

8 CODE

We have primarily used the following technology stack to build our application: python, html,css and flask.

8.1 Music to book recommendation

The following section covers the implementation details of the music to book recommender system. We have primarily used Python and its associated packages to develop the music to book recommendation engine.

```
fn_list_i = [
    librosa.onset.onset_strength,
    feature.chroma_stft,
    feature.chroma_cqt,
    feature.chroma_cens,
    feature.melspectrogram,
    feature.mfcc,
    feature.spectral_centroid,
    feature.spectral_bandwidth,
    feature.spectral_contrast,
    feature.spectral_rolloff,
    feature.tonnetz
]

fn_list_ii = [
    feature.zero_crossing_rate
]

def get_feature_vector(y,sr):
   feat_vect_i = [ np.mean(funct(y,sr)) for funct in fn_list_i]
   feat_vect_ii = [ np.mean(funct(y)) for funct in fn_list_ii]
   feature_vector = feat_vect_i + feat_vect_ii
   return feature_vector
```

Figure 1: Code snippet for feature extraction

8.1.1 *Feature extraction*

. We have used the librosa library from python to extract the 12 essential features mentioned in the previous section , from our dataset of audio files. The librosa library is a python package which has been built with the purpose of music and audio analysis.

As we can see from figure 1, we use the get_feature_vector() function to extract features for each audio file. The two lists defined above: fn_list_i and fn_list_ii, contain the various features such as onset strength, chroma stft, zero crossing rate, which have been imported using the librosa library. The get_feature_vector takes as input the audio file and the sampling rate , and then uses them to calculate the mean value of each feature.

8.1.2 *Dataset preparation and model training*

. After the feature extraction step, we have one csv file corresponding to each genre Then, we concatenate all these files to form a single dataset that contains the feature vectors of all the audio files across all genres present in the dataset. We have extensively used the pandas library provided by python for processing the csv files. Before concatenating, we add the label column, to determine the genre of the audio files. The genre labels are represented as numbers, for instance, 0, represents Bollywood rap, 1 represents, 1 represents Ghazal and so on. Then, we split the dataset into train and test in the ratio of 80:20

Further, we have used the LightGBM model for the purpose of music genre classification. Python has a library called lightgbm, which was used for this purpose. The training parameters such as learning rate, minimum child samples on the leaf nodes of each tree, and boosting type, were chosen accordingly to achieve higher accuracy.

8.1.3 *Mapping the genre from Music to book*

. For computing similarity between the music and book genres, we first define two lists: one list contains all the music genres that are present in the dataset, and the other list contains the book genres that broadly encompasses all book categories that are available. Then , we used the pre-trained word embedding model called GoogleNews-vectors-negative-300 to convert the genres present in these two lists to suitable word embedded representations. The GoogleNews-vectors-negative-300 model was trained using a large corpus of news articles by employing the Word2vec algorithm. It is considered as one of the most accurate pre-trained word embedding models and has been widely used for other tasks such as sentiment analysis, text classification, etc. This model was loaded and processed with the help of the Gensim library in our code. Gensim library is a popular python package used for various natural language processing tasks, such as analysis of similarity of documents, topic modelling, etc. We use the "KeyedVectors" class of the Gensim library to load the model which has been originally stored in binary format. After loading the model, we clean the name of the genres present in both the lists by removing punctuations and then converting the words present in the genre to tokens. Then, we compute the mean vector for each genre by using the vector representation of the words present in the name of the genre.

	genre	m-genre
0	rap	[[Sociology, Young adult, Comic, Comedy, Poetry]]
1	ghazal	[[Western fiction, Romance, Anthology, Literar...
2	folk	[[Comedy, Literary, Western fiction, Spiritual...
3	bhajan	[[Comic, Comedy, Literary, Spirituality, Poetry]]
4	romantic	[[Western fiction, Spirituality, Comedy, Adven...
5	sufi	[[Romance, Western fiction, Literary, Spiritua...

Figure 2: The mapping between music and book genre

After the conversion of genres into word embedding representations, we use the cosine similarity metric to compute the similarity

ratio from each pair of genres. We have used the sklearn library , which possesses the cosine similarity function. More specifically, the cosine similarity function is present in the metrics.pairwise package of the sklearn module.

As shown in figure 2, column 1 called [genre] contains the music genres, and column 2 called [m-genre], represents the mapped book genres corresponding to each genre of music which has been generated by using cosine similarity.

8.1.4 *Processing the book dataset*

. Processing the book dataset primarily involved extracting the relevant columns, cleaning the data and then grouping the similar genres, and converting them to the book genres that we have defined in the previous step, so that we could use the music to book genre mapping created effortlessly.

Firstly, we load and process the dataset using the csv library of python. Then, we extract the following 4 columns: Id (Wikipedia ID) , Name, Description (Book summary) , and the genre. Then, we start with the process of data cleaning. The first step performed for this is cleaning the genre column. The genre column was originally stored in json format. For instance, the genre of the book titled "Transfer of Power" was stored as follows: "/m/01jfsb": "Thriller", "/m/02xlf": "Fiction". So, we had to extract the genre from this data and then store it in the column. For this, we employed the json library of python to extract the genres (which were stored as values of the key-value) from each row in the genre column. Then, we removed all those entries where the genre column was empty. Next, we removed the stop words and punctuations from the genre and description columns.

The last major step in the processing of the book dataset was to group the similar genres. Firstly, we manually grouped some of the genres to a broader category, on the basis of evident similarity between the names. For instance, "Postcyberpunk" was replaced with "cyberpunk" and "Biopunk" was replaced with "Biotechnology fiction". Next, we used cosine similarity to measure the similarity between the genres present in the book dataset and the genres that we had defined in the music to book mapping step. After this step, we found a relevant genre (relevant as per the music to book mapping) for each and every book genre originally present in the dataset.

In figure 3, column 1 called [genre], represents the original genres present in the book dataset, while column 2 called [m-genre] represents the mapped genre for the given genre. After the similarity grouping of genres ,we were able to reduce the genre set from 227 to 27 genres.

8.1.5 *Generating recommendation*

. For generating the final book recommendations, firstly, we ingest the audio files. Then, we extract the feature vector of these audio files and then using the lightGBM model, we predict the genre of the music. Then we use the mapping created between music and book genres to extract the book genres corresponding to the predicted music genre.. Then , we iterate through each row of the dataset, to find those rows of the genre column, which contains the relevant genres, and extract the corresponding titles of the book. If the number of options for book recommendation during this process exceeds 5, we randomly select any 5 books from these options and then recommend the user.

	genre	m-genre
0	School story	Drama
1	Prose poetry	Poetry
2	Suspense	Drama
3	Economics	Sociology
4	High fantasy	Fantasy

Figure 3: The genre mapping for some of the genres in the book dataset.

8.2 Book- to -Book recommendation

For the book recommendation, we have firstly used the MultiLabelBinarizer from the SKLearn library which helped us to encode the multiple labels present in the genre column for each book. We have majorly considered the [Description] column in the dataset, column to train our model. The Description column contains the book summaries which were cleaned by remobving stopwords and punctuation. We have employed the nltk library to implement the removal of stop words, punctuation, tokenization,etc. RThen we have used the Tfidf vectorisation for the words present in the summary of each book. Then, we have utilized the logistic regression coupled with the OnevsRest classification strategy to perform multiclass classification. We again took the help of sklearn to implement this.

9 EVALUATION

We have applied subjective reasoning to evaluate our system. For baseline results, individual accuracies for book genre classification and music genre classification had been calculated. As for the final system i.e. music to book, there is no publicly available dataset combining both music and book domains, we do not ground truth labels available to us. This results in us resorting to human evaluation for the evaluation of our system's results using cosine similarity to join both the domains. Accuracy for Book to Book - 11.7%; Music to Book - 64%; Hamming Loss for Book to Book is 0.07%. Information about state of the art (SOTA) technologies is not available because of Research Gap.

10 CONCLUSION

We were able to successfully implement the music to book and book to book recommendation system and also the web interface for accessing them. The user can now upload a song as per his/her preference on the interface and the the system generates the recommendations for books that would match the user's preferences.Besides, the book to book recommendation engine could also be used to

generate relevant book recommendations for the user by uploading the preferred books of the user.

11 REFERENCES

[1] Penha, G., & Hauff, C. (2020, September). What does bert know about books, movies and music? probing bert for conversational recommendation. In Proceedings of the 14th ACM Conference on Recommender Systems (pp. 388-397). https://dl.acm.org/doi/pdf/10.1145/3383313.3412249

[2] Ramakrishnan, G., Saicharan, V., Chandrasekaran, K., Rathnamma, M. V., & Ramana, V. V. (2020). Collaborative filtering for book recommendation system. In Soft Computing for Problem Solving: SocProS 2018, Volume 2 (pp. 325-338). Springer Singapore.https://doi.org/10.1007/978-981-15-0184-5_29

[3] Rana, A., & Deeba, K. (2019, November). Online book recommendation system using collaborative filtering (with Jaccard similarity). In Journal of Physics: Conference Series (Vol. 1362, No. 1, p. 012130). IOP Publishing. https://iopscience.iop.org/article/10.1088/1742-6596/1362/1/012130/pdf

[4] Cho, J., Gorey, R., Serrano, S., Wang, S., & Watanabe-Inouye, J. (2017). Book recommendation system (Doctoral dissertation, Bachelor's thesis. Carleton College). https://cs.carleton.edu/cs_comps/1617/book_rec/final-results/paper.pdf

[5] Kaminskas, M., & Ricci, F. (2012). Contextual music information retrieval and recommendation: State of the art and challenges. Computer Science Review, 6(2-3), 89-119. https://www.sciencedirect.com/science/article/abs/pii/S1574013712000135

[6] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," in Canadian Journal of Electrical and Computer Engineering, vol. 43, no. 3, pp. 170-173, Summer 2020, doi:10.1109/CJECE.2020.2970144.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9165253

[7] Music genre classification reference:
https://github.com/ujjwalll/GACMIS/blob/master/Final{%20}Manuscript/ML___GACMIS___Version___1%20(5).pdf

[8] Book genre classification reference:
https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/
https://www.kaggle.com/code/iamhungundji/book-summary-genre-prediction/notebook