PROJECT REPORT

EE 541 – Computational Introduction to Deep Learning

CineLit Recommender: Elevating Entertainment Discovery

Nissanth Neelakandan Abirami
University of Southern California
nissanth@usc.edu

Sam Devavaram Jebaraj
University of Southern California
devavara@usc.ed

Abstract

In an era with cinephiles and bibliophiles, there are vast repositories of cinematic and literary content. In order to strike a balance for customers, we suggest creating CineLit, a recommendation system that will revolutionize content discovery by symbiotically uniting the domains of literature and film.

CineLit also recognizes the necessity for a comprehensive recommendation system that takes user preferences into account. By utilizing content-based filtering, the system will take into account the inherent qualities of films and books, making recommendations that are in line with users' own preferences for certain genres and subjects.

Collaborative filtering enhances CineLit's capabilities by analyzing user behavior and preferences, enabling the system to identify like-minded individuals and present recommendations based on collective tastes. Therefore, this collaborative method fosters a sense of community and a common appreciation for literary and cinematic excellence.

Keywords: Cinelit, Cinematic and Literary excellence, Collaborative filtering, Content based filtering, Recommendation system.

1. Introduction

Our project seeks to deliver a sophisticated recommendation engine as a customized solution for content discovery, specifically designed to address the growing difficulty of traversing the huge and constantly developing landscape of literary and cinematic content. By making movies and books more closely match personal tastes and inclinations, this effort aims to improve the user experience.

This study is important because it addresses the problems that users now encounter in the face of an excessive amount of entertainment content. As a result of the constant overabundance of books and movies, users are faced with an overwhelming amount of information, making a personalized content discovery solution necessary.

Online streaming services and digital libraries should also prioritize this improved user experience as it supports their primary business goal of increasing user interaction. For example, in the domain of Amazon Prime Video, where a multitude of movies are accessible, our recommendation engine may easily spread its impact to the Kindle website, increasing the channels through which users can acquire books according to their tastes.

The recommendation system that has been suggested has the potential to revolutionize the dynamics of user interaction in the entertainment sector. Through the provision of tailored recommendations based on user preferences, the system seeks to greatly extend user engagement and increase retention rates. Our platform stands apart from competitors in the highly competitive entertainment industry thanks to the integration of cutting-edge techniques like collaborative filtering and content-based filtering. This tactical advantage gives the company a clear competitive advantage while also improving the user experience. Furthermore, the system's ability to reliably provide accurate and engaging content recommendations acts as a stimulant to draw in

new users and keep hold of current ones. Good user experiences should naturally lead to word-of-mouth recommendations, which will increase the subscription base. By using feed-forward networks, the system's dynamic flexibility guarantees that it will remain relevant over time by keeping up with changing user preferences and industry content trends. To put it simply, the recommendation system not only becomes a source of increased engagement but also a means of strategic differentiation in the market, allowing for consistent increases in subscriber base and user retention.

In this project, collaborative filtering, feed-forward networks, matrix factorization perspectives and correction mechanisms into learning into recommendation systems are all thoroughly explored.

2. Problem statement

In today's environment inundated with an overwhelming array of cinematic and literary content, users face a substantial challenge in discovering content that resonates with their unique preferences. The inadequacies of current recommendation systems in offering a comprehensive resolution to this problem have highlighted the necessity for a novel and revolutionary platform.

Acknowledging this necessity, we suggest creating CineLit, a recommendation engine aimed at transforming the content discovery industry. CineLit aims to solve the drawbacks of conventional recommendation systems by fusing the worlds of literature and film.

Through the strategic application of cutting-edge methods such as content-based filtering and collaborative filtering. CineLit aims to offer a more sophisticated and practical answer to the complex problem of tailored content discovery.

Figure 1: CineLit recommending the first Harry Potter book after the completion of the first movie.Additionally, the second movie is recommended following the order of the movie series.

3. Mini Literature Review

[1] Previous research in the field of university library book recommendation systems has extensively investigated collaborative filtering, content-based filtering, and hybrid models. According to the findings of these studies, combining content-based filtering with Course Syllabus information, in addition to collaborative filtering, is effective in overcoming the limits of individual recommendation algorithms. Evaluation measures such as RSME and K-Fold Cross Validation have proved critical in measuring the performance of hybrid systems. The current research follows suit by providing a hybrid recommendation system distinguished by a novel weighting mechanism for increased resilience. The emphasis on tackling the cold start problem and boosting accuracy demonstrates the field's consistency of aims, emphasizing a consistent commitment to developing recommendation system approaches.

[2] This study addresses the problem of improving movie recommendations by combining Content-based and Collaborative-based filtering algorithms. Content-based techniques focus on previous data and preferences of users, whereas collaborative approaches take insights from comparable users' behaviors. The study compares the effectiveness of User-based, Item-based, SVD, and SVD++ algorithms in Collaborative filtering. Notable is the introduction of a revolutionary hybrid recommendation engine that combines Content-based and SVD models for maximum performance. In contrast to previous studies, this

research coincides with current issues in recommendation systems but distinguishes itself by applying specialized algorithms, such as SVD and SVD++. The unique hybrid approach, which combines SVD and Content-based filtering, is a pioneering effort to integrate the capabilities of both techniques in order to overcome restrictions and demonstrate possible gains in both performance and efficiency.

[3] This study describes a unique unsupervised learning-based hybrid recommender system that incorporates Collaborative Filtering, a Content-Based Approach, and a Self-Organizing Map. The system surpasses state-of-the-art approaches in Collaborative Filtering when tested on a subset of the Movies Database, displaying higher accuracy, precision, and efficiency. The hybrid model combines three recommendation methodologies, combining collaborative filtering with a self-organizing map and age demographic attribute consideration. Despite a lengthier recommendation time, the system outperforms in terms of precision and performance improvement. The combination of Self-Organizing Map with collaborative filtering decreases RMSE in clusters, outperforming K-means clustering. Based on real user data, the Precision-Recall assessment highlights the system's capacity to improve movie suggestions.

4. Dealing with data

4.1 Dataset Description

CineLit's recommendation algorithm leverages an extensive Kaggle Movie Metadata dataset from three key files: movies_metadata.csv, keywords.csv, and credits.csv. With details on 45,000 films, movies_metadata.csv is crucial for content-based filtering. The keywords.csv file encapsulates movie plot keywords in a JSON object, providing insights into film topics. Similarly, credits.csv explores movie credits using JSON-structured objects for a comprehensive understanding of film attributes. Links.csv provides essential identifiers like IMDB and TMDB IDs, facilitating data augmentation and integration with external datasets. For optimal resource utilization, Links_small.csv offers a subset of identifiers for 9,000 films. Preliminary testing is supported by Ratings_small.csv, containing 100,000 user ratings.

Shifting to the Books Dataset from Kaggle's Good Books-2k, ratings.csv encompasses reviews for 2,000 books. The dataset, including metadata from books.csv—featuring average ratings, authors, Goodreads IDs, and titles—proves valuable. To_read.csv marks user-intended "to read" books, while tags.csv converts tag IDs to names, serving as a reference in book_tags.csv for user-assigned tags, shelves, and genres. This robust dataset enhances the CineLit recommendation algorithm's comprehensiveness.

4.1 Cleaning and Managing Missing Values:

In the initial data processing phase, the focus is on creating a robust and tidy dataset by importing information from CSV files, specifically "credits," "keywords," and "movies_metadata." This involves eliminating redundant columns and handling missing values, with special attention given to columns like "belongs_to_collection," "homepage," and others. Simultaneously, the 'id' column is transformed to the 'int64' data type for compatibility. This meticulous dataset preparation lays the foundation for streamlined analysis and feature extraction, crucial for subsequent stages in the movie recommendation system.

In the process of managing missing values, the central goal is to preserve dataset integrity. This involves a dual approach: systematically filling missing values in key columns like "original_language" and "runtime" with appropriate defaults, ensuring crucial details are retained. Additionally, any remaining missing values are carefully addressed by removing the corresponding rows. This comprehensive strategy for managing missing values is essential for maintaining overall accuracy and dependability, establishing a strong basis for subsequent modeling and analysis within the movie recommendation system.

4.2 Scaling and Weighted Scoring:

In the data normalization phase, the focus is on achieving consistency in the treatment of numerical features. The Min-Max scaling algorithm is applied to key metrics such as "weighted_average" and "popularity." Scaling ensures uniformity in assessing numerical properties, reducing the potential for bias in subsequent analyses. Simultaneously, the feature combination and weighted scoring processes come into play. Through an adept combination of scaled features, a weighted score is generated, serving as a

critical parameter in the movie recommendation system. This score, derived from both popularity and weighted average vote, provides a reliable mechanism for prioritizing and sorting recommended movies.

$$\text{Weighted Average} = \frac{(\text{Vote Avg} * \text{Vote count}) + (\text{Mean (Vote Avg)} * 0.8\ \text{quant}}{\text{Vote count} + 0.8\ \text{quant (Vote\_Count)}}$$

$$\text{Calculated score} = \text{Weighted avg} * 0.4 + \text{Popularity} * 0.6$$

4.3 Feature Cleaning and Extraction:

In the text data domain, the focus is on preparing textual information for in-depth analysis. Employing text data learning techniques involves systematically removing digital characters, punctuation, and unnecessary elements from columns like "overview," "tagline," and "keywords." This meticulous cleaning enhances the streamlined nature of textual data, paving the way for more precise analysis. Subsequently, text feature extraction takes center stage, with the strategic creation of a bag-of-words representation. This format not only simplifies analysis but also forms the foundational element for later stages of the movie recommendation system, where textual insights play a crucial role in generating tailored recommendations.

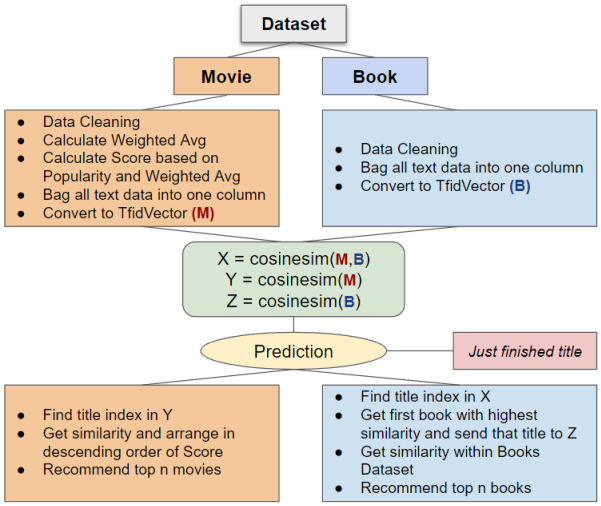


Figure 2. Flow Chart of the Hybrid Recommendation System

5. Solutions and implementations

Our main goal during the first stage of importing and cleaning movie data is to create a clean, well-structured dataset that will be the basis for further analysis. In order to do this, we first import the necessary data from CSV files, such as movies_metadata.csv, credits.csv, and keywords.csv, and arrange them into Pandas Data frames (credits, movies, and keywords). A thorough cleaning procedure is used in the movie DataFrame to guarantee that it is devoid of unnecessary information and that any missing data are properly addressed. In particular, columns that are not related to our research are removed, and the important identifier 'id' is transformed to 'int64' data type to ensure smooth operation compatibility.

Subsequently, these separate datasets are combined using the 'id' column to create a single Data Frame (df) that contains all of the movie-related information. This stage is essential for assembling information from multiple sources into a cohesive dataset that offers a comprehensive picture of every film.

Taking care of missing values is essential to maintaining data integrity. We employ a two-pronged strategy, methodically adding suitable default values to any missing data in columns like "original_language," "runtime," and "tagline." Concurrently, any remaining rows that have missing values are carefully eliminated from the dataset, guaranteeing that the final dataset is trustworthy and comprehensive for analysis.

This complex relationship is carefully calculated between movies and books using the cosine similarity lens.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The collaborative filtering process is based on the similarity scores that are obtained, which are the basis for the recommendation of books that are considered to be in line with a particular cinematic classic, such as the renowned "The Lord of the Rings." Collaborative filtering is stunning because it can uncover the hidden relationships between users' literary selections and the diverse range of movies they've seen.

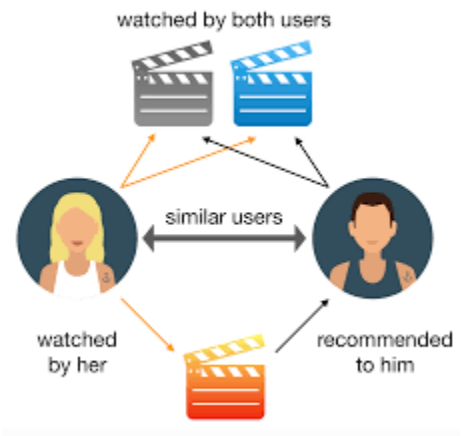


Figure 3. Collaborative Filtering Recommender System

In a parallel trend, the creation of a strong TF-IDF matrix for both movies and books propels content-based filtering to prominence. This matrix captures the linguistic and semantic subtleties, and the cosine similarity calculation that follows takes the system to the next level: it suggests movies that are consistent with the theme of a particular film.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

This hybrid paradigm creates a recommendation experience that goes above the norm by balancing content-based elegance with participatory filtering.

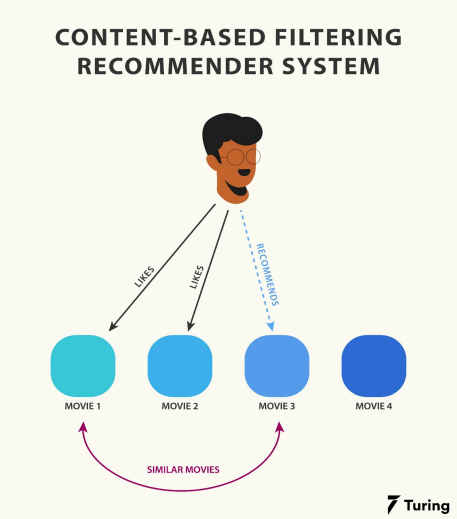


Figure 4. Content Based Filtering Recommender system

Finding distinct movie titles and user IDs turns into a crucial first step in building collaborative filtering models that are expertly created using TensorFlow's computational prowess.

The final score in the Hybrid Recommendation System is a weighted combination of content-based similarity scores (similarity) and collaborative filtering scores (score):

$$score=(1-similarity\_w) \times score + similarity\_w \times similarity$$

The parameter that regulates the impact of content-based similarity on the final score in this case is called similarity_weight. This final score is used to rank and suggest movies to users in the 'predict' function.

Figure 5. Hybrid Recommender system

Finally, a Deep learning model was implemented to combat the irregular outputs given by the former approach. We chose to settle with Tensorflow as its recommenders' library was much flexible to extract slices of TfidVectors.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_movie_model (Seque (None, 128)              0
_________________________________________________________________
sequential_user_model (Sequen (None, 128)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 129
=================================================================
Total params: 230,401
```

Figure 6. Hybrid Recommender system

6. Results and Discussion

The hybrid recommendation system with Movies and Books returned a good amount of books as well as watch-next movies that were visibly positive..

```
just_finished = 'The Lord of the Rings'
predict_movie(just_finished, similarity_weight=0.7, top_n=10)
```

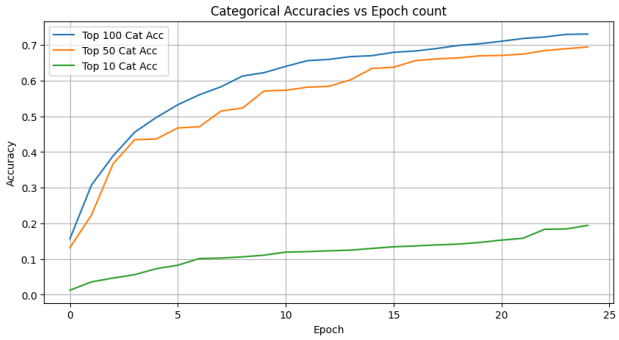
original_title	similarity
The Lord of the Rings	1.000000
The Lord of the Rings: The Fellowship of the Ring	0.388274
The Lord of the Rings: The Two Towers	0.376160
The Lord of the Rings: The Return of the King	0.280603
The Hobbit: An Unexpected Journey	0.293470
The Hobbit: The Battle of the Five Armies	0.249623
Minions	0.007537
The Hobbit	0.265034
The Hobbit: The Desolation of Smaug	0.211754
Big Hero 6	0.002285

```
predict_book(just_finished)
```

title	similarity
The Two Towers (The Lord of the Rings, #2)	1.000000
The Return of the King (The Lord of the Rings, #3)	0.349337

Figure 6: Both book and movies being recommended from the system for the movie "The Lord of the Rings"

Over to the deep learning model, the model that was trained with a word embedding of 128 and a Keras framework achieved 70% accuracy and saturated. The accuracy were calculated with respect to Top 100, Top50 and Top10 and Root Mean Square Error was chosen as the best option.



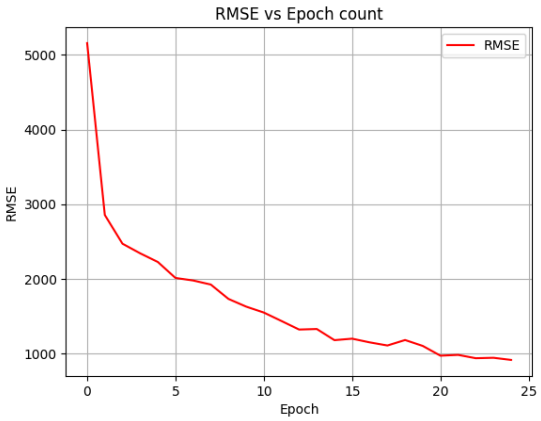


Figure 7: Accuracy and cumulative loss curve obtained from the model

A noticeable observation is that there are few junk recommendations visible in both approaches with a lower similarity score in the below the recommended table.

```
predict_movie(1, 5)
```

```
Top 5 recommendations for user 1:

1. American Pie
2. My Tutor
3. Greed
4. Vivement dimanche!
5. Rocky III
```

```
predict_rating(1,'My Tutor')
```

```
Predicted rating for My Tutor: 2.0572562217712402
```

Figure 8: User 1 here is a vivid fan of Comedy and Drama mixture and the model predicts the viewer's top 5 picks and estimated rating for one of the movie

7. Conclusion and Future Work

By expanding the capabilities of CineLit, our recommendation engine, we hope to create a

future in which cross-domain suggestions are a distinguishing feature of user interaction. We hope to investigate connections between CineLit and other fields, like music, art, and gaming, by expanding its impact beyond the boundaries of literature and film.

. Our goal is to improve the system's ability to comprehend complex user choices. Sentiment analysis must be included, user behavior patterns must be carefully examined, and contextual data must be included. By exploring particular moods, themes, and even historical records this creative method creates an environment where users actively influence the direction of content discovery. Together, these efforts can redefine and improve the user experience as CineLit develops, making content discovery a multifaceted, collaborative process.

References

[1]. Pitiwat Arunruviwat, Veera Muangsin, “A Hybrid Book Recommendation System for University Library” 2022 26th International Computer Science and Engineering Conference (ICSEC) | IEEE | DOI: 10.1109/ICSEC56337.2022.10049318.

[2]. Sakina Salmani, Sarvesh Kulkarni, “Hybrid Movie Recommendation System Using Machine Learning”, 2021 International Conference on Communication Information and Computing Technology (ICCICT) | IEEE | DOI: 10.1109/ICCICT50803.2021.9510058

[3]. Yassine Afoudi, Mohamed Lazaar, Mohammed Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” Elsevier, Simulation Modelling Practice and Theory, Volume 113, December 2021, doi: 10.1016/102375.

[4]. N. Ifada, T. F. Rahman and M. K. Sophan, ”Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation,” 2020 6th Information Technology International Seminar (ITIS), 2020, pp. 219-223, doi: 10.1109/ITIS50118.2020.9321014.

[5]. Y. Xiong and H. Li, ”Collaborative Filtering Algorithm in Pictures Recommendation Based on SVD,” 2018 International Conference on Robots and Intelligent System (ICRIS), 2018, pp. 262-265, doi: 10.1109/ICRIS.2018.00074.

[6]. S. Agrawal and P. Jain, ”An improved approach for movie recommendation system,” 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 336-342, doi: 10.1109/ISMAC. 2017.8058367.

[7]. Blog Post, “https://medium.com/analytics-vidhya/building-a-movie-recommendation-engine-in-python-53fb47547ace"

.