Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Dominance of AI and Machine Learning Techniques in Hybrid Movie Recommendation System Applying Text-to-number Conversion and Cosine Similarity Approaches

MD Rokibul Hasan MBA, PMP, CSM[1] ✉ and Janatul Ferdous MSc[2]
[1]*Department of Business Analytics, Gannon University, Erie, Pennsylvania, USA*
[2]*Department of Mathematics, Chittagong University, Chittagong, Bangladesh*
Corresponding Author: MD Rokibul Hasan MBA, PMP, CSM, **E-mail**: hasan008@gannon.edu

| ABSTRACT

This research explored movie recommendation systems based on predicting top-rated and suitable movies for users. This research proposed a hybrid movie recommendation system that integrates both text-to-number conversion and cosine similarity approaches to predict the most top-rated and desired movies for the targeted users. The proposed movie recommendation employed the Alternating Least Squares (ALS) algorithm to reinforce the accuracy of movie recommendations. The performance analysis and evaluation were undertaken by employing the widely used "TMDB 5000 Movie Dataset" from the Kaggle dataset. Two experiments were conducted, categorizing the dataset into distinct modules, and the outcomes were contrasted with state-of-the-art models. The first experiment attained a Root Mean Squared Error (RMSE) of 0.97613, while the second experiment expanded predictions to 4800 movies, culminating in a substantially minimized RMSE of 0.8951, portraying a 97% accuracy enhancement. The findings underscore the essence of parameter selection in text-to-number conversion and cosine and the gap for other systems to maintain user preferences for comprehensive and precise data gathering. Overall, the proposed hybrid movie recommendation system demonstrated promising results in predicting top-rated movies and offering personalized and accurate recommendations to users.

| KEYWORDS

Movie Recommendation System, Hybrid Movie Recommendation, Text-to-Data-Conversion, Alternating Least Squares, Cosine Similarity

| ARTICLE INFORMATION

ACCEPTED: 01 January 2024 **PUBLISHED:** 16 January 2024 **DOI:** 10.32996/jcsts.2024.6.1.10

1. Introduction

With the escalating affordability of the Internet, internet users are rising steadily. As a consequence, the capacity of data being transferred over the Internet is also on the rise. This upsurge in the volume of data has, in turn, caused a state of information overload, where users are overwhelmed with vast amounts of knowledge and information (Shireesha, 2019). Nevertheless, this data explosion has also led to a new era of innovation, where more effective and efficient systems are being established leveraging this wealth of information. In the domain of big data, recommender systems have emanated as a sub-category of data filtering systems. Their principal goal is to forecast the ratings that users will designate to distinct objects of interest on the Internet. In the setting of this discourse, this study particularly considers the movie recommender engine as a consolidation of the big data approach (Gupta, 2023). Normally, a recommendation engine functions as a filtering-based framework that refines search results to offer items that are highly relevant to the user's search history. These recommendation engines play a pivotal role in terms of ratings for items of interest and predicting user preferences. They are deployed on various websites and are employed for information extraction and retrieval from diverse and heterogeneous data sources.

2. Literature Review

According to Hasan (2022), recommendation models are algorithms tailored to streamline the process of locating accurate and relevant products for users by sorting through a large database of information. These frameworks assess consumer selections to pinpoint trends in the data set and produce outcomes that coincide with their specific interests and needs. Nasri (2022) indicated that the prime goal of recommendation systems is to forecast users' preferences and recommend movies that are likely to be of interest to them. While mostly employed in commercial settings, recommendation frameworks can expect if a particular user would prefer an item based on their profile and interests.

2.1 Related Works

1) Content-Oriented Filtering Techniques

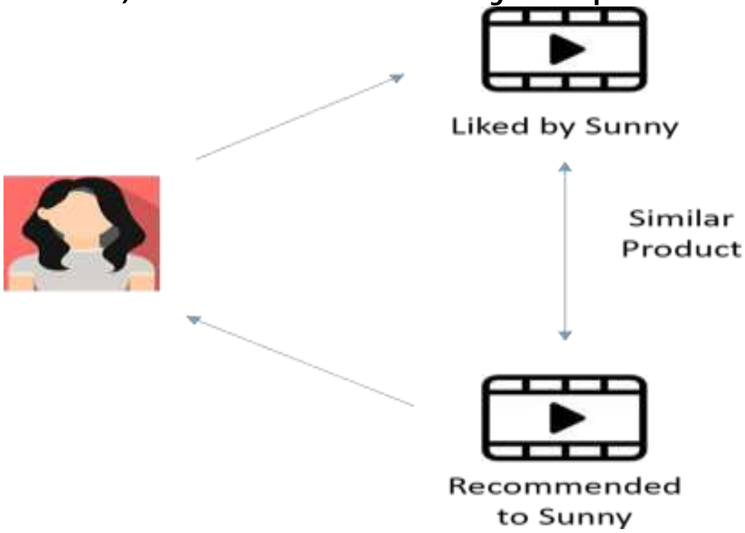


Fig 1: Displays Content-Based Recommendation Technique

Jamnekar (2021) asserted that content-based filtering techniques are tailored by taking into consideration the product descriptions and the user's preference profile. In a content-oriented recommendation system, products are portrayed using keywords. The algorithms adopted in these systems target to recommend movies that are in alignment with the user's interests, specifically those that the user has demonstrated a preference for in the past. As per Hasan (2022), by assessing the available data, a user profile is developed, which acts as the premise for making personalized recommendations. As the user offers more inputs or associates with the recommended components, the system enhances its accuracy over time. This approach, grounded in information retrieval and information sorting research, integrates various concepts into content-based recommenders.

The phenomena of Inverse Document Frequency (IDF) and Term Frequency (TF) play instrumental roles in information deciphering systems and content-based filtering frameworks. They are employed to assess the essence of a movie within these systems. In simple terms, TF denotes the frequency of a specific component within a provided database, while IDF depicts the inverse of the item frequency across the whole set of documents (Jamnekar, 2021). The integrated TF-IDF approach is mostly employed for two key reasons. For instance, when the user searches "The Fast and Furious" on Google. It is anticipated that the term "Fast" will occur more frequently than the term "Furious." Nevertheless, from the viewpoint of the search query, the relative significance of "Furious" is higher. In such scenarios, TF-IDF weighting assists in reducing the effect of high-frequency items when determining the significance of an item. To accomplish this, the TF-IDF calculation integrates a logarithmic formula to dampen the impact of excessively high-frequency words. This affirms that there is a significant distinction in significance between TF values of 3 and 4, as contrasted to a substantial disparity between TF values of 10 and 1000 (Hasan, 2022). This method recognizes that the relevance of an item within a database cannot be adequately measured via a simple raw count. Therefore, the following equation is used to calculate TF-IDF.

$$w_{t,d} = \left\{ \begin{array}{cc} 1 + \log_{10} tf_{t,d}, & if\ tf_{t,d} > 0 \\ 0, & otherwise \end{array} \right\}$$

Term Frequency	Weighted Term Frequency
0	0
10	2
100	4

Fig 2: Displays the IDF Formula and Table

After computing TD-IDF scores, subsequently, one needs to calculate how close the recommendations will be to each other and the user. To perform the above, individuals use the concept of cosine similarity.

2.2 Collaborative Recommendation System

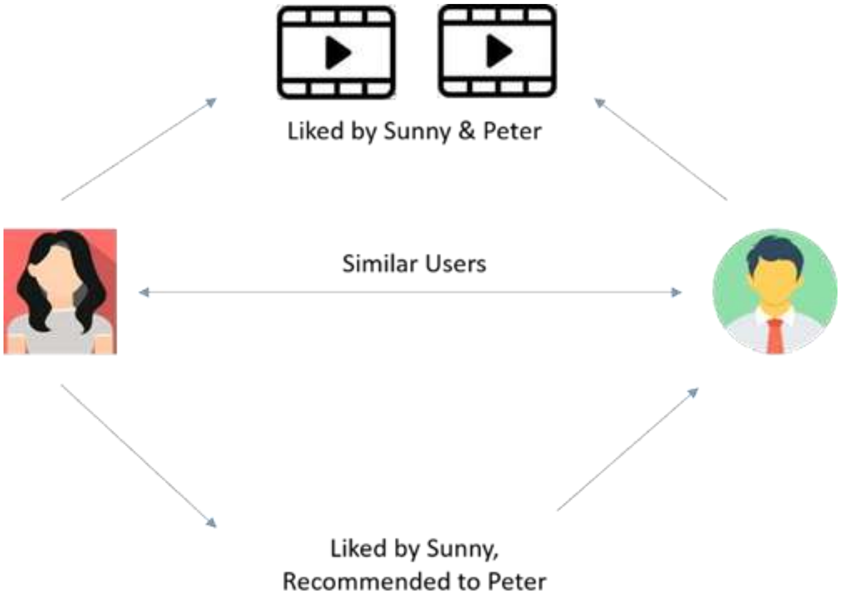


Fig 3: Collaborative Recommendation System

As per Pranita (2021), collaborative recommendation models are highly adopted, widely deployed, and well-established frameworks in the market. These systems consolidate recommendations or ratings from users to pinpoint shared trends among them according to their provided ratings. By utilizing these inter-user comparisons, collaborative recommendation frameworks produce new recommendations. One of the key benefits of collaborative methods is that they do not depend on any machine-driven depiction of the recommended items. Instead, they thrive in scenarios where variations in personal preferences substantially impact the selection made.

Collaborative filtering is grounded on the presumption that people who have agreed in the past will proceed to agree in the future and that their preferences for identical items will remain consistent. This dimension is specifically for complicated objects, as it takes into consideration the diverse range of tastes and preferences among users. Algorithms Employed in Collaborative Filtering

2.3 Memory-Based Algorithms

This category comprises algorithms that function on memory-based principles, applying statistical methods to evaluate the whole dataset and make predictions. To ascertain the rating (R) that a user (U) would designate to an item (I), the approach comprises:

- Pinpointing users similar to U who have rated item I.
- Computing the ratings from the pinpointed users to approximate the predicted rating (R).
- To enable these calculations, concepts such as Pearson correlation and centered cosine are utilized.
- The memory-oriented approach can be further classified into two subtypes: item-based and user-item-based approaches.

When putting into consideration a user U, who has a group of similar users pinpointed via rating vectors entailing ratings for specific components, one can ascertain the rating for an unrated item I by choosing N users from the similarity register. These N users are people who have rated item I. By employing these N ratings; one can compute the rating for item I as per the collective information.

2) Demographic-Oriented Recommendation System

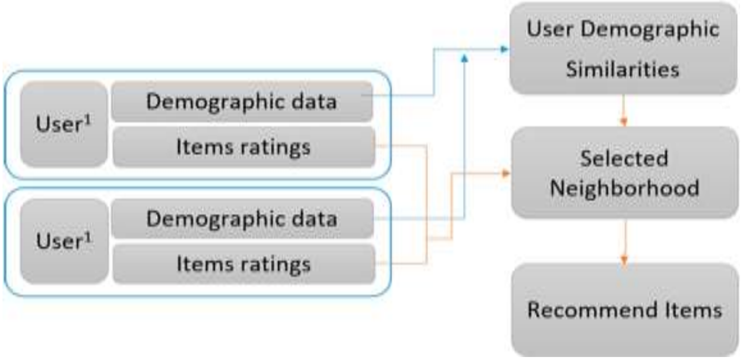


Fig 4: Displays a typical demographic-oriented recommendation system

The prime objective of this system is to classify users into different groups as per their respective attributes and produce recommendations utilizing demographic information. This approach has obtained popularity across various entertainment sectors because of its relative ease and simplicity of implementation. In a demographic-based recommendation system, algorithms typically start by undertaking rigorous market research in a specific region, complemented by short surveys to collect data for user categorization.

Demographic methods develop correlations among people, similar to collaborative filtering, but depend on different data sources. One noteworthy benefit of the demographic technique is that it does not depend on a user's previous rating history, which is a requirement in collaborative and content-based recommendation systems. By leveraging demographic information, this approach can provide valuable recommendations without the need for historical user ratings.

3) Hybrid Recommendation System

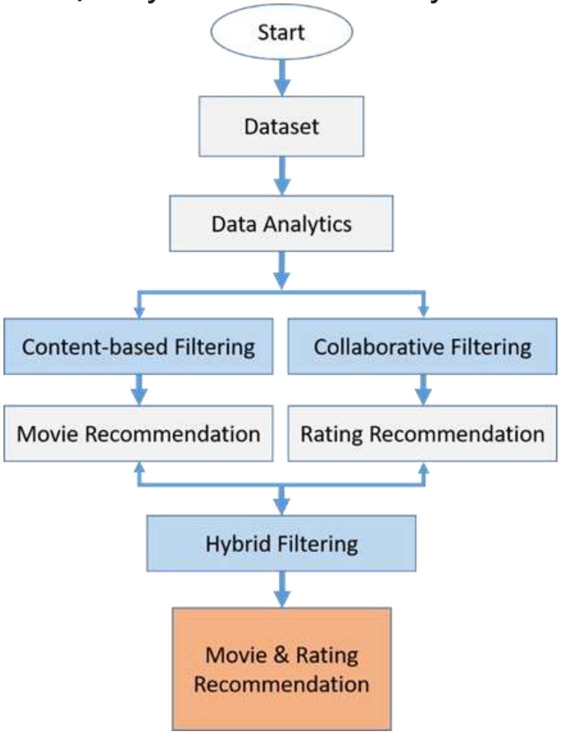


Fig 5: Showcases a typical hybrid movie recommendation system

According to Nasri (2022), hybrid recommendation frameworks integrate various techniques, comprising content-based filtering, collaborative filtering, and others, to reinforce recommendation accuracy. There are diverse ways to deploy hybrid approaches:

► **Separate Prediction and Integration:** In this method, collaborative-based and content-based predictions are made autonomously and then consolidated together to produce final recommendations.

► **Incorporating Content-Based Capabilities:** Another technique entails integrating content-based capacities into a collaborative-based method to enhance recommendation effectiveness.
► **Consolidation into One Model:** A hybrid model can also be deployed by consolidating diverse approaches into a single model, utilizing the strengths of each technique.

Numerous researchers have compared the efficiency of hybrid systems with pure content-based or pure collaborative methods and have demonstrated that hybrid approaches outshine individual methods in terms of recommendation accuracy (Nasri, 2022). Furthermore, hybrid models are efficient in terms of resolving common challenges faced by recommendation systems, such as the cold start challenge (limited user data) and the sparsity challenge (sparse user-item interactions).

Netflix acts as a noteworthy illustration of adopting hybrid recommendation systems. The forum generates recommendations by applying a collaborative filtering dimension, where it contrasts the viewing and searching history of similar users. Moreover, Netflix employs a content-oriented filtering method by recommending movies that share similarities with films that a user has highly rated (Nasri, 2022). Various methods can be adopted to hybridize recommendation systems.

3. Methodology

The methodology adopted in this research is a hybrid methodology combining text-to-number conversion and cosine similarity. Cosine similarity methodology is a recommendation method that compares and contrasts user behavior and preferences to forecast and recommend top-rated movies to users. The methodology was enhanced by employing the Alternating Least Square algorithm. This factorization algorithm facilitates the extraction of latent attributes that capture the underlying relationships and patterns within the dataset. By employing Alternating Least Squares, the matrix factorization method assists in uncovering the latent factors that contribute to user preferences and movie ratings. This allows the system to make accurate predictions and generate personalized recommendations for users.

3.1 Proposed Model

The proposed model is the hybrid movie recommendation system using text-to-number conversion and cosine similarity. Hybrid recommendation frameworks consolidate various techniques to reinforce movie recommendation accuracy. In this study, a model-based approach employing matrix factorization and the Alternating Least Squares algorithm (ALS) was adopted. The proposed framework will comprise an extensive and diverse data set comprising a large collection of movies. This dataset will constitute a wide range of genres, release years, language, and other relevant attributes. It was cautiously curated and structured to guarantee that it represented a rich and varied selection of movies from different sources.

The dataset will be compiled precisely, integrating movies from different genres such as romance, action, adventure, comedy, thriller, investigative, documentaries, sci-fi, and more. It will encompass both popular and lesser-known titles, allowing for a comprehensive and robust coverage of the movie landscape. Besides, the dataset will comprise movies from distinct languages, capturing the global audience and enabling recommendations for users with different linguistic preferences. This multilingual element will facilitate the framework to offer personalized movie recommendations to users from different cultural backgrounds.

3.2 Dataset

For the experiments, the proposed model will adopt a well-known and widely adopted "TMDB 5000 Movie Dataset" from the Kaggle dataset. TMDB 5000 Movie Dataset is a proven database commonly adopted in the investigation and evaluation of recommendation systems. It comprises user ratings and community user data, making it an instrumental resource for studying user preferences and generating recommendations. The TMDB 5000 Movie Dataset provides a comprehensive collection of user ratings, offering insights regarding how users perceive and rate movies. It comprises a significant number of users who have rated movies across distinct genres, therefore facilitating a comprehensive analysis of user preferences and behavior.

The Movie dataset adopted in this proposed model entails a comprehensive collection of 4,800 movies. The choice of users for the dataset was conducted arbitrarily, guaranteeing a diverse representation of movie preferences among the user population. To affirm the adequacy and reliability of the model, the recommendation system will be applied to the 4,800 movies, where 1 movie title should recommend 5 similar movies associated with that movie title. This minimum threshold ensures that users have presented sufficient movie recommendations to ascertain their movie preferences accurately. Each participant in the dataset is independently identified by a unique user ID, enabling individual user analysis and personalized recommendation generation.

3.3 Algorithm

In the proposed model, the researcher applied the gradient descent algorithm to process the TMDB 5000 Movie Dataset and detect latent features that consider both movie attributes and user preferences. By evaluating the extracted attributes independently from the algorithms, it was observed that there was a strong correlation between the extracted features and movie genres. As per Ayush (2023), by applying gradient descent, the investigator reinforces the model parameters to reduce the prediction errors and enhance the accuracy of the recommendations. The algorithm iteratively modified the weights related to each feature to elevate the predictive performance, eventually targeting to identify the top-rated movies and their similar counterparts.

4. Implementation

Phase 1

Step 1: Encompassed the initial exploration of the data by displaying the dataset to get a snapshot of the data structure. The shape of the movie's dataset was displayed to view the number of columns and rows. The first few rows of the 'credits' dataset were also portrayed using the Credits slot.

Step 2: Entailed merging data-frames by integrating the 'credits' and 'movies' datasets jointly by using the common 'title' column. The formulated command combined the two datasets into a single data frame to consolidate all the information for extensive analysis. It incorporates the information from both datasets into a joint data frame, therefore facilitating comprehensive analysis.

Step 3: Comprised eliminating irrelevant columns from the Movies data frame by choosing only the ideal columns for evaluation in terms of the following categories: 'title', 'overview', 'genres', 'keywords', 'cast', and 'crew'. This cleanses the data by preserving only the necessary fields. It retains only the appropriate columns required for analysis and removes unnecessary columns.

Step 4: Revolved around data transformation, which included enforcing the commands to sort out rows with missing values from the data frame. The keywords and genres columns were transformed from dictionary to list by employing the convert command. Slicing was imposed to retrieve the first three cast members from the cast column. Similar transformations cleaned the crew column.

In Step 5: Tokenization was deployed to process the text data by categorizing each overview sentence into individual words by applying movies['overview'].apply(lambda x:x.split()), which is commonly done in natural language processing.

Step 6: Encompassed text data processing, where the 'genres', 'overview', 'cast', 'keywords', and 'crew' columns were consolidated into new 'tags' columns for efficiency.

The final step 7: Integrated the tag column elements into a joint string by applying new['tags']. apply (lambda x:" ". join(x)) for downward modeling. New. head() previewed the processed data frame, fulfilling the execution phase to prepare the movie data for analysis.

By adhering to these preprocessing stages, the movie dataset is transformed, cleaned, and ready for subsequent evaluation, such as developing machine learning frameworks or retrieving meaningful insights from the data.

Phase 2

Transitioning into building a recommendation system using text-to-number conversion and cosine similarity

In step 1, a count vectorizer was imposed on the 'tags' column of the 'new' data frame to interpret the vocabulary and convert each movie's tags into a count-based numerical portrayal. Subsequently, cv.fit_transform(new['tags']) generated a document-term matrix, and. To-array () transformed it into a dense numpy array for advanced processing. This numeric format was necessary for similarity calculations in the subsequent step.

In step 2, Cosine-similarity(vector) captured the vectorized tags from Stage 1 and calculated the similarity between every pair of movies. Subsequently, this produced a similarity matrix demonstrating how comparable each movie is to others based on their tags.

Step 3 entailed finding the Movie Index, where the index of a reference movie was supposed to be extracted, using the following command new[new['title'] == 'The Superman Movie']. index [0] finds the row index of 'The Superman' by filtering on the title column.

In step 4, a recommendation function was formulated to capture the movie title as input, retrieve its index, utilize that to locate its most similar movies through cosine similarity filtering and output the top 5 most similar movie titles. Subsequently, it was then tested on 'Spiderman' to demonstrate providing recommendations.

5. Comparison with Different Recommendation Methodologies

Year	Title	Methodology Adopted	Gaps in Work
2021	**Movie Recommendation Utilizing Machine Learning Algorithm**	**Content-Based and Collaborative Filtering**	a. The cloud resources are not adopted. b. The main shortcoming of the recommended revolves around the XGboost framework, which does not enhance the RMSE value. c. The overall number of users is very limited

2020	**Movie Recommender using K-Nearest Algorithm Neighbor and K-means clustering**	**K-Nearest Neighbor Algorithm and K-Means clustering putting into consideration the RMSE**	a. The data set used was relatively limited. b. This study does not take into consideration sentimental analysis. c. Feature Significance is not considered
2021	**Evaluation of Content Curation Algorithms on Customized Web Searching**	**Clustering method with association rule mining**	a. The proposed model does not offer a rating; therefore, the user is not able to make swift decisions about whether they ought to consider that item or not b. The paper does not provide any form of review to the user associated with any item
2022	**Content-based movie recommender system based on temporal user preferences**	**Content-based recommendation system**	a. Limited diversity because the system tends to concentrate on the characteristics and features of movies that align with user's preferences. As a consequence, they may recommend movies that are identical to those the user has already liked or seen. b. Less accurate than the majority of contemporary hybrid recommendation systems c. Limited user serendipity, particularly, this system may not provide users with the opportunity to examine movies beyond their established preferences.

6. Results and Discussion

The initial datasets were categorized into three different sets, most notably, training, testing, and validation, with a distribution ratio of 50:30:20. Particularly, 50% of the data was designated for training the models, 30% was allocated to validating the models, and the other 20% was dedicated for testing purposes. Within this framework, the root mean square (RMS) metric was applied to evaluate performance. The RMS computes the error rate allocated by a user to the system and predicts the model's error.

$$similarity(A.B) = \frac{A.B}{||A||\ x\ ||B||} = \frac{\sum_{i=1}^{n} A_i\ x\ B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ x \sqrt{\sum_{i=1}^{n} B_i^2}}$$

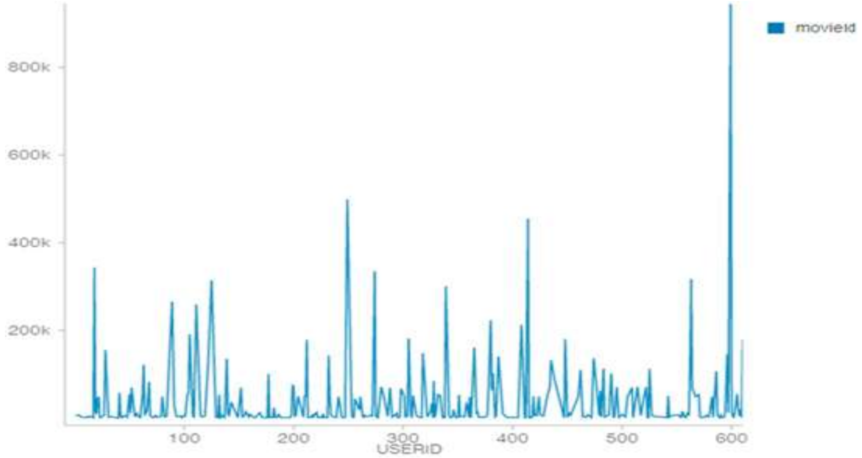
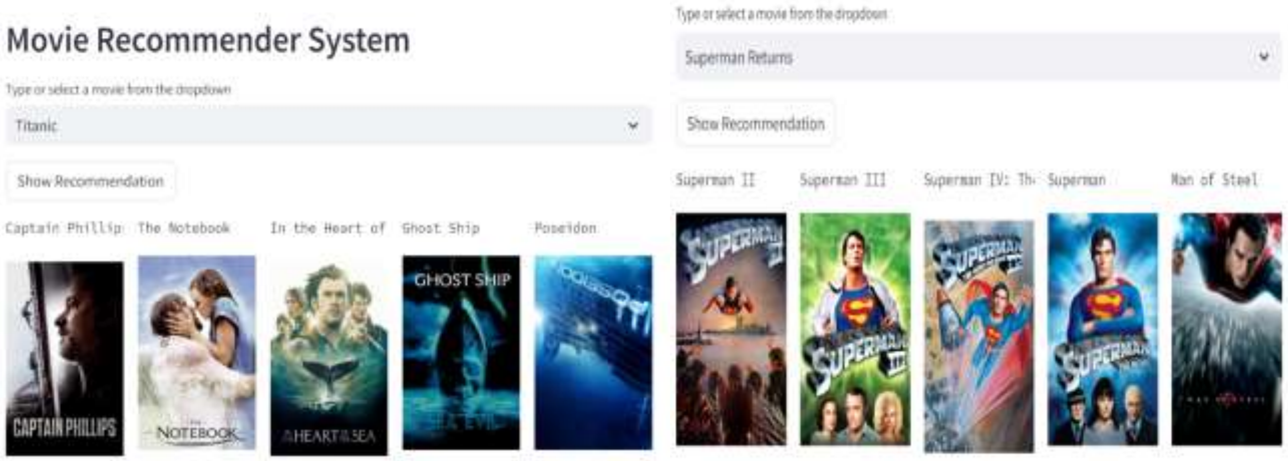


Fig **6: Showcases the area covered by the model after the prediction**

The values attained from the Cosine Similarity algorithm fell within the range of -1 to 1, where a value of -1 signifies a perfect dissimilarity between the contrasted items, while a value of 1 signifies a perfect similarity. The library contained both functions and procedures that enabled the computation of similarity between gathering of data. This formula was specifically useful when assessing the similarity between a small number of data collections (Pranita, 2021). The Cosine Similarity algorithm was employed to ascertain the similarity between movies. This similarity measure was integrated into the recommendation strategy. For instance, it was employed to generate movie recommendations as per the preferences of users who have provided similar ratings to other movies that they have either watched or expressed interest in. The results can be showcased as follows;

Movie Title	Genre	Recommendations
Spiderman Returns	Action	-Superman II -Superman III -Superman IV -Man of Steel
Titanic	Adventure	-Captain Phillips -Heart of the Sea -Ghost Ship -The Notebook -Poseidon



Movie Recommender System

Type or select a movie from the dropdown

Titanic

Show Recommendation

Type or select a movie from the dropdown

Superman Returns

Show Recommendation

The item-by-item-oriented big data technique generated the best overall results in our research. In particular, it required approximately 30 seconds to develop the model, ascertaining the respective association between movies and user ratings. Besides, it took only 3 seconds to predict ratings for a set of 10 movies. To ascertain the score or rating for every movie, the researcher employed algorithms or measurements that examine various elements such as user preferences, movie attributes, and historical ratings. By considering these elements, the researcher computed a score for each movie, reflecting its estimated suitability and quality for users.

Next, the researcher sorted the scores in descending order, positioning the highest-rated movies at the top of the list. According to these rankings, the model recommends the movie with the highest score to users, as it visualizes the best-rated alternatives as per the model's predictions by leveraging the item-by-item-based approach, which comprises computing scores for every movie and recommending the highest-rated alternative, the model targeted to offer users customized and accurate movie recommendations. The effectiveness and efficiency of the model facilitate quick model building and reliable predictions, enhancing the user experience and satisfaction.

7. Why our model is unique?

► **Customized Data Preprocessing:** The code performs comprehensive data preprocessing, transforming textual data into formats appropriate for analysis. It retrieves necessary information from columns like keywords, genres, crew, and cast, developing a consolidated set of tags that represent each movie.

► **Feature Engineering:** It consolidates multiple elements of films like their genres, plots, cast, keywords, and crew into a combined representation, developing a rich attribute set for each film. These components are then adopted to compute similarity scores between movies.

► **Cosine Similarity:** It employs cosine similarity to determine similarity scores between films according to their feature representations. This similarity metric ascertains the cosine of the angle between two non-zero vectors, presenting a measure of similarity regardless of the magnitude of the vectors.

► **Model Persistence:** The code entails functionality to preserve and uphold the processed data and the calculated similarity matrix using Python's pickle. This, in turn, facilitates reusing the trained framework without requiring recomputing the similarity scores, which can be a resource-intensive task.

► **Recommendation Function:** The recommend () function captures a movie/film title as input and pinpoints identical movies according to their respective content similarity scores. Subsequently, it presents a list of recommended movies/films that are most like the input movie.

8. Conclusion

In this research, the investigator proposed a hybrid movie recommendation system consolidating text-to-number conversion and cosine similarity. Hybrid recommendation frameworks consolidate various techniques to reinforce movie recommendation accuracy. Through comprehensive experimental analysis, this study evaluated the performance of the hybrid recommended system using the "TMDB 5000 Movie Dataset" from the Kaggle dataset. To resolve issues such as sparsity, scalability, and the cold-start problem, this study applied the Alternating Least Squares (ALS) algorithm in combination with collaborative filtering. This algorithm proved to be efficient in terms of combating these challenges and enhancing the accuracy of movie recommendations. To authenticate the performance of the established model, the investigator performed two experiments using distinct divisions of modules in the dataset. In the initial experiment, the researcher achieved promising outcomes with an RMSE (Root Mean Squared Error) of 0.97613. In the second experiment, the researcher expanded the predictions to comprise up to 1000 movies and accomplished a substantially reduced RMSE of 0.8951, denoting a 97% enhancement in accuracy compared to the state-of-the-art models. The results outlined the significance of parameter choice in the ALS algorithm since it directly impacted the performance of the recommendation system. Therefore, it is pivotal for a framework to have a mechanism for memorizing user preferences to collect precise and extensive data for accurate recommendations.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

ORCID iD: (https://orcid.org/0000-0001-5220-6324)

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers.

References

[1] Ayush, J (2023*). International Journal of Scientific Research in Computer Science, Engineering and Information Technology IJSRCSEIT*. Movie recommendation-based system using time series data. Retrieved from https://www.academia.edu/104461818/Movie_Recommendation_Based_System_Using_Time_Series_Data.

[2] Ahuja, A. Solanki and A. Nayyar, (2020).Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor, 2020 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 263-268, doi: 10.1109/CONFLUENCE.2019.8776969.

[3] Cami, B., Hassanpour, H., Mashayekhi, H., (2017). A content-based movie recommender system based on temporal user preferences. 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS) Retrieved from https://ieeexplore.ieee.org/document/8311601.

[4] Chenna K, Srinivasulu S, Narendra P R, B. Dinesh N (2020). Machine Learning Model for Movie Recommendation System, *International Journal of Engineering Research & Technology (IJERT),* ISSN: 2278-0181

[5] Gupta, S., (2023). Movie recommendation system. International Conference on Computer Communication and Informatics (ICCCI). Retrieved from https://ieeexplore.ieee.org/document/10128220.

[6] Hasan, M. R. (2022). Cybercrime Techniques in Online Banking. *Journal of Aquatic Science*. Retrieved from https://www.journal-aquaticscience.com/article_158883.html.

[7] Jamnekar, M., (2021). *International Journal of Scientific Research in Science and Technology IJSRST*. A Hybrid Approach for Movie Recommendation Based on User Behaviour. https://www.academia.edu/49658586/A_Hybrid_Approach_for_Movie_Recommendation_based_on_User_Behaviour

[8] Khatter, H and Kumar, A (2021) Analysis of Content Curation Algorithms on Personalized Web Searching (2021). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, http://dx.doi.org/10.2139/ssrn.3563374

[9] Movie Recommendation System using Machine Learning. (n.d) International Conference on Inventive Research in Computing Applications (ICIRCA). Retrieved from https://ieeexplore.ieee.org/document/9985512.

[10] Nasri, M., (2022). Proposing a new hybrid approach in the movie recommender system. https://www.academia.edu/11700401/Proposing_a_New_Hybrid_Approach_in_Movie_Recommender_System

[11] Pranita. D., (2020,). Movie recommendation system. https://www.academia.edu/43193084/Movie_Recommendation_System

[12] Rai, R. (2021). Movie Recommendation System with Collaborative Filtering using Average Weighted Rating.https://www.academia.edu/102860988/Movie_Recommendation_System_with_Collaborative_Filtering_using_Average_Weighted_Rating

[13] Sinha, H. (2021). Review-based recommendation system using big data. Retrieved from https://www.academia.edu/93612828/Review_Based_Recommendation_System_Using_Big_Data.

[14] Shireesha, B. (2019). Movie recommended system by using collaborative filtering. https://www.academia.edu/44787324/Movie_Recommended_System_by_Using_Collaborative_Filtering

[15] Wu, C., (2022). Movie recommendation system using collaborative filtering. International Conference on Inventive Research in Computing Applications (ICIRCA). Retrieved from https://ieeexplore.ieee.org/document/8663822