Career Point International Journal of Research(CPIJR)
©2022 CPIJR | Volume 4 | Issue 2 | ISSN : 2583-1895
December-2023 | DOI: https://doi.org/10.5281/zenodo.11291679

Cinematic Curation : Unveiling the Magic of ML-Based Movie Recommendations

Mahak Kaur Chhabra[1] , Rohit Maheshwari[2]

School of Engineering & Technology, Career Point University, Kota
Email: mchhabra1808@gmail.com
Assistant Professor, School of Engineering & Technology, Career Point University, Kota Email: rohit.maheshwari@cpur.edu.in

Abstract:
Recommender systems are designed to provide personalised suggestions to users, enhancing the overall user experience. This paper features a **content-based** recommender system, which recommends based on the similarity of content, utilising "tags". The main computational method harnessed is the cosine similarity function, sourced from the sci-kit-learn library.

Keywords: Content-Based Filtering, Recommender System, Machine Learning, Movie Recommendations, Data Analysis, EDA

I Introduction:

In the era of digital content consumption, the overwhelming abundance of movies poses a challenge for audiences to discover films that align with their preferences. Movie recommendation systems have emerged as indispensable tools, leveraging advanced algorithms and data analytics to assist users in navigating the vast cinematic landscape. This review paper immerses itself in the intricacies of a content-based recommendation system project, centred around a singular cosine similarity function designed to unearth tailored movie suggestions. The primary goal is to present an in-depth exploration of the project's methodology, research findings, and emerging trends in the realm of content-based movie recommendations.

The exponential growth of digital platforms, coupled with the diversification of user preferences, has spurred the evolution of recommendation systems. From early collaborative filtering approaches to sophisticated content-based methods, the landscape is marked by a rich tapestry of techniques employed to enhance the accuracy and effectiveness of movie recommendations. Understanding the historical progression and the intricate interplay between methodologies is pivotal in comprehending the current state of the field.

Amidst the vast array of recommendation systems, this focused review not only unravels the inner workings of the content-based model but also addresses its implications for user-centric movie discovery. By leveraging the cosine similarity function within this singular approach, we hope to illuminate the potential of such streamlined methods in delivering precise and relevant movie recommendations to users . Beyond the algorithms themselves, the paper explores the pivotal role of datasets in training and evaluating these systems, shedding light on the implications of data biases and the challenges associated with ensuring representative and diverse recommendations.

In the subsequent sections, we delve into the project's experimental setup, results, and critical analysis, offering insights into the implications of the chosen methodology. Through this exploration, we aim to contribute to the broader discourse on content-based recommendation systems and their role in shaping the future of personalized content discovery.

Career Point International Journal of Research(CPIJR)
©2022 CPIJR | Volume 4 | Issue 2 | ISSN : 2583-1895
December-2023 | DOI: https://doi.org/10.5281/zenodo.11291679

Conceptual Framework :

The recommendation process involves utilizing cosine similarity, [1] where 'A' represents the user vector and 'B' signifies an item vector. The resulting values in the cosine similarity matrix are sorted in descending order, and the top items are recommended for the user.

$$cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_i A_i B_i}{\sqrt{\sum_i A_i^2}\sqrt{\sum_i B_i^2}}$$

II Review of Literature:

Recommender systems fall into three main categories: content-based recommender systems, collaborative recommender systems, and hybrid recommender systems. Figure 1 illustrates these various types of recommender systems.

Content filtering [2] leverages item attributes [3] to suggest similar items based on user preferences. This method analyzes the likeness of user and item features, drawing insights from user information and interactions. For instance, if a user shows an interest in action-adventure books and sci-fi movies, a content filtering recommender might recommend a new release within the same genres, such as a popular book like "Dystopian Odyssey" or a movie like “Interstellar.”

In **collaborative filtering**, [4] recommendations are driven by user behavior and historical interactions. The user's past preferences and actions are instrumental in identifying patterns and similarities. For example, if User ‘A' has shown a liking for ‘BTS’, ‘TXT’, and ‘ENHYPHEN’, and User ‘B’ shares similar preferences by liking ‘BTS’, ‘TXT’ and ‘EXO’ there is a high likelihood that User 'A' might enjoy ‘EXO’, and User ‘B’ might appreciate ‘ENHYPHEN’. Collaborative filtering utilizes these shared preferences to provide personalized recommendations.

Hybrid recommender systems [5] integrate multiple recommendation strategies in diverse ways to leverage their complementary strengths. Many research studies often incorporate collaborative filtering with another technique, frequently employing a weighted approach.

Career Point International Journal of Research(CPIJR)
©2022 CPIJR | Volume 4 | Issue 2 | ISSN : 2583-1895
December-2023 | DOI: https://doi.org/10.5281/zenodo.11291679

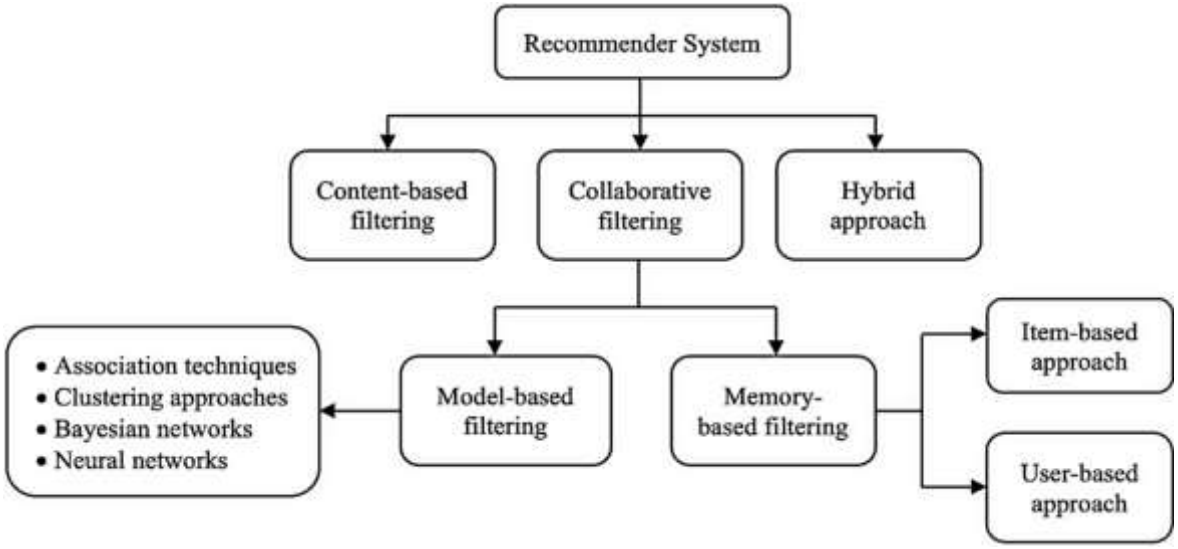

Fig. 1 : Types of recommender systems

III Methodology

The utilised dataset originates from the TMDB movie dataset obtained from Kaggle, consisting of two CSV files - one for movies and the other for credits. The 'movies' dataset encompasses 4803 records, each with 20 features, while the 'credits' dataset includes 4803 records with 4 features : 'movie_id', 'title', 'cast', and ‘crew.’

Following the merging of both datasets based on the 'title' feature, a consolidated dataset of dimensions (4809, 23) was obtained. To streamline model training, irrelevant features such as 'budget,' 'homepage,' and 'production_company' were excluded.

Given the nature of content-based recommender systems relying on tags, careful consideration was given to columns conducive to tag creation. The refined set of features includes 'movie_id,' 'title,' 'overview,' 'genres,' 'keywords,' 'cast,' and ‘crew.'

During **data preprocessing**, null values (3) were dropped, and duplicate records were removed. Additionally, columns were formatted correctly, and dictionary structures within the 'genres,' 'keywords,' and 'cast' columns were converted into lists using the 'ast' module's 'literal_eval()' function. For example, the transformation from [{“id” : 28, “name” : “Action”},{“id” : 12, “name” : “Adventure”}] to [Action, Adventure].

Further transformations were applied to the 'crew' column, isolating and updating it with only the director information. To address the string format problem of the 'overview' column, it was split into a list format. Additionally, whitespace removal transformations were executed on columns such as 'keywords,' 'genres,' 'cast,' and ‘crew.’ For instance, 'Science Fiction' is transformed into 'ScienceFiction' to prevent the recommendation system from treating 'Science' and 'Fiction' as distinct entities, potentially causing confusion and inaccurate predictions.

Career Point International Journal of Research(CPIJR)
©2022 CPIJR | Volume 4 | Issue 2 | ISSN : 2583-1895
December-2023 | DOI: https://doi.org/10.5281/zenodo.11291679

A new column named 'tags' was created, containing the concatenated data from the relevant columns. Subsequently, a new dataframe was constructed, focusing on 'movie_id,' 'title,' and 'tags' columns, and the list in the 'tags' column was converted to lowercase string.

The subsequent step involved text vectorization to address the problem statement of returning the five most similar movies based on user input. The '**Bag of Words**' technique will be employed, wherein all the tags are combined, and the 5000 most common words are identified and extracted. The frequency of occurrence of each of these words in each movie's tags is then counted, resulting in a matrix of dimensions (5000, 5000). Each row represents a vector in a 5000-dimensional vector space. Notably, stop words (e.g., 'In,' 'of,' 'is,' etc.) will be disregarded among these 5000 common words. This is achieved through the use of the scikit-learn's CountVectorizer() class, with parameters 'max_features = 5000' and 'stop_words = 'English''.

Using the **Bag of Words** technique, tags were converted into vectors, and a multidimensional vector space was established. The process involved transforming tags into a numpy array, eliminating similar features through stemming, and repeating all vector conversion steps. This includes transforming variations like ['loved', 'loving', 'loves'] to ['love'] using the stemming technique, facilitated by the '**nltk**' library's '**PorterStemmer**' class, essentially obtaining the root word.

At this point, we have 4806 movies, each with 5000 dimensions. The next step involves calculating the distance between each movie and every other movie. It's important to note that distance is inversely proportional to similarity. Instead of calculating Euclidean Distance, which is not a reliable measure for high-dimensional data due to the 'curse of dimensionality,' we opt for cosine distance, representing the angle between vectors.

To compute cosine distance, we utilize the 'cosine_similarity()' function from the 'sklearn.metrics.pairwise' library. This function takes the vectors as input, resulting in a matrix of shape (4806, 4806).
The recommendation function sorted the vector of each movie in descending order, retaining the index through enumeration to fetch movie names based on similarity scores.

WEBSITE:

Developed using PyCharm, the website leverages the Streamlit library for the frontend. To showcase the list of movies on the website, we employ the pickle library to create and dump a pickle file containing the movie dictionary. This data is loaded for display, and a similarity matrix pickle file is also stored. Additionally, we design necessary functions to retrieve similar movies.

The subsequent phase involves presenting movie posters alongside their names. To achieve this, we utilize an API from TMDB's website, fetching the movie posters based on the respective 'movie_id'.

DEPLOYMENT:

For deployment, the project utilizes the Spaces platform offered by 'huggingface'. It provides a convenient and user-friendly environment for hosting and showcasing machine learning projects, enabling easy access for users to interact with the deployed application. In this context,

'huggingface' Spaces is employed as the hosting platform to make the movie recommender system accessible to users over the web.

A live demo of the project can be found **here**!!

Research Findings:
Calculating accuracy for a content-based recommender system typically involves evaluating how well the system's recommendations align with user preferences or actual user interactions. While traditional accuracy metrics like precision, recall, or F1 score are commonly used for collaborative filtering recommender systems, content-based systems might be assessed differently.

Here are some approaches to evaluate the accuracy of a content-based recommender system without explicitly splitting the dataset :

- User Feedback or Surveys:
  - Gather user feedback on recommendation relevance and satisfaction.
  - Utilize surveys, interviews, or ratings to gauge user opinions.
- Implicit Feedback:
  - Use implicit indicators like clicks, views, or watch time if explicit feedback is lacking.
  - Measure user interaction frequency with recommended movies.
- Domain-Specific Metrics:
  - Define metrics aligned with system goals (e.g., user engagement).
  - Assess metrics like time spent on the platform post-recommendation.
- Diversity and Novelty:
  - Evaluate diversity and novelty in recommendations.
  - Use metrics based on genres, actors, or other movie features..
- Comparison to Baseline:
  - Establish a baseline (e.g., rule-based or random recommendations).
  - Compare content-based system performance against the baseline.
- User Retention:
  - Monitor user retention and engagement post-implementation.
  - A successful system should encourage users to explore recommended content.

The evaluation of the content-based recommendation system was primarily conducted through user surveys, where users were asked to provide feedback on the relevance and accuracy of the suggested movie recommendations. Remarkably, the majority of predictions were deemed relevant and correct by users, with a success rate of approximately 80-90%. This high level of user satisfaction indicates the effectiveness of the content-based approach in delivering personalized movie suggestions tailored to individual preferences. However, it's worth noting that a small percentage of users, around 10-20%, reported instances where recommendations were not entirely aligned with their preferences, highlighting the ongoing challenges in achieving perfect personalization for all users.

Career Point International Journal of Research(CPIJR)
©2022 CPIJR | Volume 4 | Issue 2 | ISSN : 2583-1895
December-2023 | DOI: https://doi.org/10.5281/zenodo.11291679

IV Conclusion :

In conclusion, this paper delves into the development and evaluation of a content-based recommender system focused on movie recommendations. Leveraging the cosine similarity function, the system utilizes tags and advanced data analytics to enhance the user experience in navigating the vast array of digital content. The exploration of methodologies, research findings, and emerging trends sheds light on the intricacies of content-based recommendation systems.

The study emphasizes the evolution of recommendation systems, from collaborative filtering to sophisticated content-based methods, and underscores the importance of understanding historical progression and methodological interplay. By leveraging the cosine similarity function, the paper illuminates the potential of streamlined methods in delivering precise and relevant movie recommendations. Additionally, the role of datasets in training and evaluating systems is examined, addressing implications related to data biases and the challenges of ensuring diverse recommendations.

Through experimental setups, results, and critical analyses, the paper contributes valuable insights to the discourse on content-based recommendation systems. The exploration of research methodologies, encompassing data preprocessing, text vectorization, and recommendation functions, further enriches the understanding of system intricacies.

The deployment of the recommender system through the 'huggingface' Spaces platform exemplifies a user-friendly approach, providing a convenient environment for interaction. In the realm of research findings, the evaluation of the content-based recommendation system relies on user surveys, revealing a notable success rate of 80-90%. While the majority of users found the recommendations relevant and accurate, a small percentage highlighted challenges, emphasizing the ongoing quest for perfect personalization. This study contributes valuable insights to the dynamic landscape of personalized content discovery.

Future Scope :

- Explore integrating additional features for enhanced recommendations.
- Consider extending the recommender system to diverse content domains.
- Conduct a comparative analysis with other recommendation approaches.
- Evaluate system robustness with larger datasets.
- Implement real-time user feedback mechanisms.
- Compare diverse recommendation strategies for user engagement.
- Discuss and justify chosen metrics for system evaluation.
- Explore alternative metrics for assessing accuracy.

References:

1. Alzubaidi, L., Zhang, J., Humaidi, A. J., Duan, Y., Santamaría, J., Fadhel, M. A., & Farhan, L.

(2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 1-74. https://doi.org/10.1186/s40537-021-00444-8

2. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8
3. Yu, J., & De Antonio, A. (2022). Deep Learning (CNN, RNN) Applications for Smart Homes: A Systematic Review. *Computers*, *11*(2), 26. https://doi.org/10.3390/computers11020026
4. Khaki, S., Wang, L., & Archontoulis, S. V. (2020). A CNN-RNN Framework for Crop Yield Prediction. *Frontiers in Plant Science*, *10*, 492736. https://doi.org/10.3389/fpls.2019.01750
5. Kotappa, Y. G., Krushika, M., Ravichandra, M., & Pranitha, Mrs. (2022). A Review Paper on Computer Vision and Image Processing. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(2), 68.
6. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306
7. P. Garg and A. Sharma, "A distributed algorithm for local decision of cluster heads in wireless sensor networks," *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, India, 2017, pp. 2411-2415, doi: 10.1109/ICPCSI.2017.8392150.
8. A. Sharma and A. Sharma, "KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density based clustering," *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Kerala, India, 2017, pp. 787-792, doi: 10.1109/ICICICT1.2017.8342664.
9. Sherstinsky, A. (2018). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *ArXiv*. https://doi.org/10.1016/j.physd.2019.132306
10. Collins, C., Dennehy, D., Conboy, K., & Mikalef, P. (2021). Artificial intelligence in information systems research: A systematic literature review and research agenda. *International Journal of Information Management*, *60*, 102383. https://doi.org/10.1016/j.ijinfomgt.2021.102383
11. Kaur, P. (2023). Artificial Intelligence. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 11(X), 597. https://doi.org/10.22214/ijraset.2022.44306

12. Dr. Amit Sharma. 4g wireless technology and its standards taking consideration evolution of 4g technology. National Journal of Multidisciplinary Research and Development, Volume 3, Issue 1, 2018, Pages 1102-1105

13. Dr. Amit Sharma. Development of android application services at Arokia and its architecture. National Journal of Multidisciplinary Research and Development, Volume 3, Issue 1, 2018, Pages 1072-1075

14. Vijay Malav, Dr. Amit Sharma. Effect and benefits of deploying Hadoop in private cloud. National Journal of Multidisciplinary Research and Development, Volume 3, Issue 1, 2018, Pages 1057-1062

15. Dr. Amit Sharma. Implementing the design of service oriented architecture. National Journal of Multidisciplinary Research and Development, Volume 3, Issue 1, 2018, Pages 1027-1030

16. Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, *16*(1), 1-27. https://doi.org/10.1186/s41239-019-0171-0
17. Çano, Erion. (2017). Hybrid Recommender Systems: A Systematic Literature Review. Intelligent Data Analysis. 21. 1487-1524. 10.3233/IDA-163209.
18. F. Ricci, L. Rokach, B. Shapira, Recommender Systems Handbook, Springer US, Boston, MA, 2011, Ch. Intro-duction to Recommender Systems Handbook, pp. 1–35. doi:10.1007/978-0-387-85820-3_1.