IPEC Journal of Science & Technology, Vol. 02 (02), Dec 2023
ISSN: 2583-3286(Online)

Design and development of Cordless electric vehicle charger

Deeksha Sharma, Umang Mishra, Palak Pathak, Khushi Goel
Department of Electronics and Communication
Inderprastha Engineering College
Ghaziabad, India

***Abstract*: Recommender systems have become an integral part of our digital lives, shaping the way we discover and consume information, products, and services. This review paper aims to provide a comprehensive overview of the recent advancements in recommender systems, highlighting the key algorithms, evaluation metrics, and emerging trends. We delve into the evolution of recommender systems, from traditional collaborative filtering and content-based methods to the latest developments in deep learning and hybrid approaches. Additionally, we discuss the challenges and ethical considerations associated with recommender systems, emphasizing the need for transparency and user privacy. By synthesizing insights from a wide range of research studies, this review aims to serve as a valuable resource for researchers, practitioners, and enthusiasts in the field of recommender systems. This review paper offers an in-depth exploration of recommender systems, encompassing traditional methods, contemporary algorithms, and emerging trends. From collaborative filtering and content-based filtering to the integration of deep learning and hybrid models, we scrutinize the intricacies of each approach.**

***Keywords*: Precision, Recall, F1-score, Bias and fairness, Privacy concerns Explainability, etc.**

I. INTRODUCTION

Recommender systems, also known as recommendation engines, play a pivotal role in personalized content delivery across various domains, including e-commerce, entertainment, and social media. This section provides an overview of the importance of recommender systems and their impact on user experience.
In the dynamic landscape of information and digital services, recommender systems stand as pivotal orchestrators, shaping the user experience by tailoring content recommendations to individual preferences. This technical review embarks on a journey to unravel the intricate machinery that propels the evolution of recommender systems. From foundational collaborative and content-based filtering techniques to the sophistication of hybrid models, and the transformative potential of deep learning paradigms, this review navigates the algorithmic terrain that underlies personalized recommendation systems. Beyond the algorithms, it scrutinizes the technical rigor of evaluation metrics, delving into precision, recall, and emerging metrics related to diversity and serendipity. As recommender systems continue to play a critical role in enhancing user engagement and satisfaction, this review is poised to provide a comprehensive technical exploration, offering insights and perspectives for researchers, practitioners, and technologists invested in the relentless pursuit of refining and advancing recommender system technologies.

Date of Submission: 1 Dec 2023
Date of Acceptance : 23 Dec 2023
Corresponding Author: Umang Mishra
(e-mail: umangmishra392@gmail.com)

II. TYPES OF RECOMMENDER SYSTEMS:

We categorize recommender systems into collaborative filtering, content-based filtering, and hybrid methods. We discuss the strengths and limitations of each approach, highlighting how hybrid models aim to leverage the advantages of multiple techniques to enhance recommendation accuracy.
Collaborative Filtering:
This section explores the evolution of collaborative filtering techniques, from user-based and item-based methods to matrix factorization and model-based collaborative filtering. We delve into the challenges of sparsity and scalability and discuss how novel algorithms address these issues.

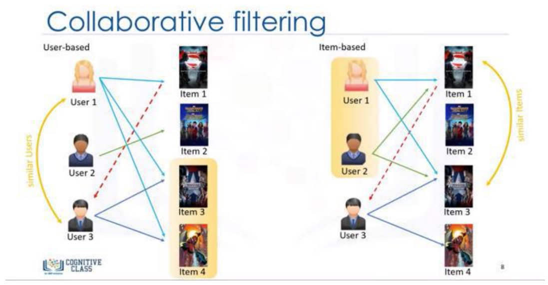

Figure 1: Collaborative Filtering

Content-Based Filtering:

Content-based filtering relies on the characteristics of items and users' preferences. We review the key concepts of content-based filtering, including feature extraction and similarity metrics. Additionally, we discuss the incorporation of natural language processing and image analysis in content-based recommendation.

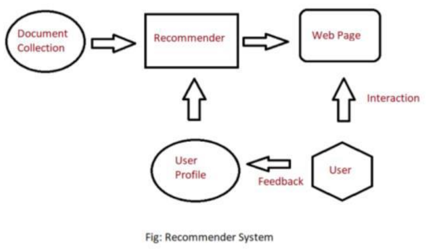

Figure 2: Content based Filtering

Hybrid Recommender Systems:

Hybrid recommender systems combine collaborative filtering and content-based filtering to mitigate the limitations of individual methods. We examine the different hybridization strategies, such as feature combination, cascade models, and meta-learning, providing insights into their effectiveness.

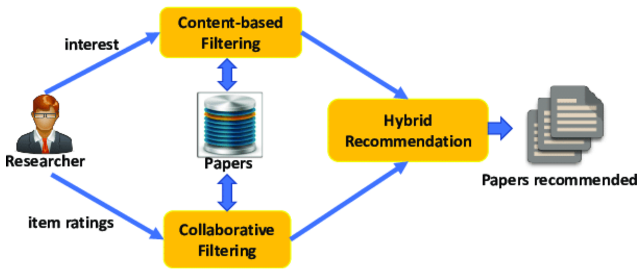

Figure 3: Hybrid recommender systems

III. DATA COLLECTION:

Data collection is the process of collecting data related to our requirements. Here the data required by our video recommendation system are videos and the information related to it such as name of the video, plot of video and genre the video belongs to. The data used in this project consists of approximately 5043 records of data, each describing a single movie with keys such as poster, genres, director, number of likes, number of dislikes, plot of the movie, movie title, actor names etc. This data is fed as a csv file which is then parsed in the data processing stage.

IV. DATA PROCESSING

Data processing is the initial steps in the recommendation system. The data collected is imported as a record and it split and flattened using keys into lists of categories such as movie titles, movie category and movie plot summary. Each movie is identified through an unique identification numberMovieID. The movie ID or the name provided as input by the user is validated and then passed to the engine for generating recommendations. The validation process checks the length of the input, verifies the input range and ensures that the input provided by the user is a valid alphanumeric character.

V. DEEP LEARNING IN RECOMMENDER SYSTEMS:

The integration of deep learning has revolutionized recommender systems by capturing intricate patterns in user behavior and item characteristics. We review deep learning architectures like neural collaborative filtering, autoencoders, and recurrent neural networks, emphasizing their ability to model complex relationships in large-scale datasets.

Recommendation Engine:

Term-Frequency Inverse Document frequency is used to provide recommendations to the user’s preferences. Each data record is converted into a vector by using the TF-IDF vectorization algorithm described previously. For each vector, a similarity measure is computed using the cosine similarity method. When a user requests for certain number of recommendations for a particular movie, the correlation coefficients are generated for the movies with respect to that movie. Each similar movie selected will have a certain score of how similar it is to the denoted movie, which is sorted into descending order, in order to list the movies with high to low similarity. According to the number of recommendations requested by the user, the indices of those movies are collected and displayed to the user as a list of movies. The recommendations generated by the engine are displayed through a user interface to the user.

V. PROPOSED SYSTEM

Our proposed system is a content based movie recommendation system (MRS) that uses string similarity document frequency and cosine similarity algorithm for recommending movies. This system understands customers, their behaviors and activities and trains the system. The main advantage of this system is that, the algorithm is designed to work efficiently even for a small set of data. The recommendations are based on the movie plot information, tags, description of movie, genres and casting of movies. It allows the users to save time and future enhancements include a data analytics portal which allows the movie fans

ISSN: 2583-3286(Online)

to find of the user to a particular genre/video. Better and more efficient recommendation systems also increase market reach and create a flux of recurring customers for the site.

Cosine Similarity:

Cosine Similarity is a complex concept which has been widely discussed in information retrieval. This algorithm converts a text document as a vector of terms. By this model, the similarity between two dataset can be found by determining cosine value between two vectors. Application of this algorithm can be performed on any two texts such as documents, sentence or paragraph. Sometimes during the similarity measurement between the vectors yields unstable results. In case of search engines, the similarity value between user query and documents are determined and then it is categorized from highest to lowest one. Higher the similarity score between the user query vector and document vector means more relevancies between query and document.

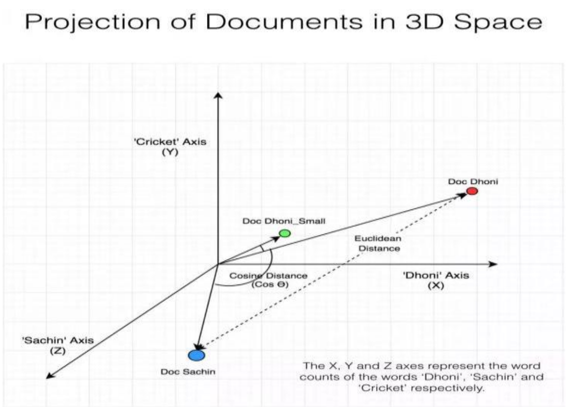

Figure 4: Projection in 3D Space

Text Vectorization:

Text vectorization in Python is a crucial step in natural language processing (NLP) tasks, transforming textual data into numerical representations. One widely-used approach is the Bag of Words (BoW) model. Python libraries like scikit-learn provide efficient tools for this. The TfidfVectorizer, for instance, not only captures term frequencies but also considers the importance of terms in the entire corpus.

Another powerful technique is Word Embeddings, where words are mapped to dense vectors, capturing semantic relationships. The popular Word2Vec and GloVe models have Python implementations. The gensim library simplifies training and using Word2Vec models.

Recent advancements include transformers like BERT, facilitating contextual embeddings. The Hugging Face Transformers library in Python provides pre-trained models for various NLP tasks, making it easier to leverage state-of-the-art approaches.

In conclusion, Python offers a diverse range of tools for text vectorization, enabling practitioners to choose the method that best suits their specific NLP goals, from traditional BoW models to cutting-edge transformer-based embeddings.

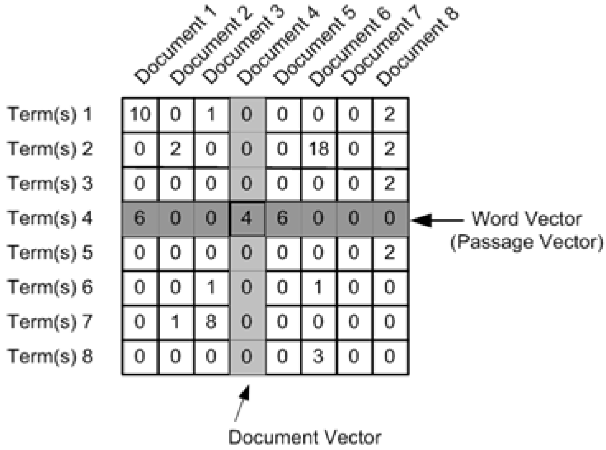

Figure 5: Vector representation

Evaluation Metrics:

Assessing the performance of recommender systems is crucial for ensuring their effectiveness. We discuss commonly used evaluation metrics, such as precision, recall, F1-score, and Mean Average Precision (MAP), shedding light on their suitability for different recommendation scenarios.

VI. CHALLENGES AND ETHICAL CONSIDERATIONS

As recommender systems become more pervasive, issues related to bias, fairness, and user privacy come to the forefront. We examine the challenges associated with biased recommendations and the ethical considerations that researchers and practitioners must address to build trustworthy and inclusive systems.

Emerging Trends:

The final section explores emerging trends in recommender systems, including the incorporation of context-aware recommendation, reinforcement learning, and the use of explainable AI to enhance user understanding and trust.

User interface (react):

Project installation: Set up a new React project using Create a React App or one of the other recommended methods. Organize your project structure using components, styles, and material folders. Parts of the user interface:

Design and implement UI components such as homepage, recommendation screen, user profile and navigation bar.

Use React Router to navigate between different pages. API communication:

Create functions or hooks to interact with Django's backend

API using a search API or library like Axios. Enable the feature to get user information, product recommendations and other relevant information. State administration:
Use React state or a state management library like Redux to manage your application and state. Store user information, recommended products and other related information in the application state. User verification:
Enable user authentication features using JWT (JSON Web Tokens) or another authentication method. Allows users to log in and out and view their profiles. Responsive Design:
Ensure the user interface is responsive and works well across devices using media queries and responsive design principles. Handling errors and suggestions:
Enable error handling for API requests and provide meaningful feedback to users. Use load wheels or other indicators to display data.

VII. CONCLUSION

In conclusion, this review paper provides a comprehensive overview of recommender systems, covering their evolution, types, evaluation metrics, and emerging trends. By understanding the landscape of recommender systems, researchers and practitioners can make informed decisions to advance the field and address its challenges.
Recapitulation
- Summarizing key technical insights from the comprehensive review.
- Reiterating the significance of technical advancements in shaping the future of recommender systems.
Future Directions
- Discussing potential technical research directions and areas for future exploration.
- Highlighting the evolving landscape of recommender systems and the role of technology in driving innovation.
This technical review endeavors to provide a meticulous examination of the algorithms, models, and methodologies that underpin recommender systems. By delving into the technical intricacies, challenges, and emerging trends, this review aims to be a valuable resource for researchers, practitioners, and technologists engaged in advancing the field of recommender systems. Recommender systems emerged from practical requirements because personalized electronic services are needed in many application areas, but existing recommender system research mostly focuses on recommendation theories and approaches. This paper mapped recommender systems from the requirements of each application area, complementing existing recommender system research and providing a useful guide for industry practitioners and researchers; this article comprehensively and illustratively summarizes the research achievements of recommender systems from the "system" perspective, and strategically groups the applications of recommender systems into eight application areas, forming a framework for the development of recommender systems; for each application domain, it carefully analyzes typical recommendation system frameworks and effectively identifies the specific requirements of recommender techniques in that domain. It directly encourages and supports researchers and practitioners to promote the popularization and implementation of recommendation systems in various fields; it highlights several very new recommendation techniques, such as social network-based and context-aware recommendation techniques, and reveals their successful application areas.

REFERENCES

[1] S. Boccaletti, Complex networks: structure and dynamics Physics Report(2006)
[2] S.-H. Min,Detection of the customer time-variant pattern for improving recommender systems Expert Systems with Applications(2005)
[3]C.-J. Zhang, Behavior patterns of online users and the effect on information filtering Physica A(2012)
[4] D.J. Watts, A twenty-first century science Nature (2007)
[5] J. Bobadilla, Recommender systems survey Knowledge-Based Systems (2013)