iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** II **Month of publication:** February 2024

DOI: https://doi.org/10.22214/ijraset.2024.58529

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Movie Recommendation System Using Machine Learning

Prof. M. A. Parde[1], Rushikesh P. Shirke[2], Shreyas S. Patil[3], Prajwal S. Bundhade[4], Niraj R. Raut[5]
[1]*Assistant Professor,* [2, 3, 4, 5]*Student at P R Pote College of Engineering and Management Amravati*

Abstract: ***A recommendation engine utilizes various algorithms to filter information and suggest the most relevant items to users. It begins by analyzing a user's past behaviour and then recommends products based on that data. However, when a completely new user visits an e-commerce website, the site lacks any historical data on that user. In such cases, one approach could be to suggest popular products, those in high demand. Another approach might prioritize recommending products that would generate the most profit for the business. Three main approaches are commonly used in recommender systems: Demographic Filtering, Content-based Filtering, and Collaborative Filtering. Demographic Filtering provides generalized recommendations to users based on demographic features, such as movie quality or genre, assuming that users with similar demographics will have similar preferences. However, this approach is considered too simplistic since every user is unique. Content-based Filtering attempts to profile a user's interests using collected data and recommends items based on that profile. Collaborative Filtering groups similar users together and uses data about the group to make recommendations to individual users.***

Keywords: Movie Recommendation, Machine Learning, Prediction, Content Based Filtering, Cosine Similarity, Count Vectorizer, Cross Validation.

I. INTRODUCTION

In today's digital era, the internet plays a vital role in providing abundant choices and information. To tackle information overload, recommendation systems have been deployed by companies across various sectors, aiding users in decision-making processes, from choosing accommodations to investment options. These systems have notably enhanced the success of e-commerce platforms like Amazon and Netflix, as evidenced by statistics such as:

1) Netflix: Two-thirds of watched movies are recommended.
2) Google News: Recommendations result in 38% more click-throughs.
3) Amazon: 35% of sales are generated from recommendations.
4) Choicestream: 28% of individuals would purchase more music if they discovered what they liked.

Recommender systems are employed in various domains including movies, music, news, books, research articles, search queries, social tags, and products in general. A recommendation system is essentially a filtration program designed to predict items that a user would prefer within a specific domain. In our context, the domain-specific item is a movie, and the primary objective of our recommendation system is to filter and predict movies that a user would prefer based on information about the user. There are numerous approaches to building a movie recommendation system, but we have opted for a content-based recommender system to provide users with movies most similar to their interests. Our recommender system suggests the top five movies that are most similar to the user's selected movie.

II. LITERATURE REVIEW

Several studies have been conducted on the topic relevant to our project.

Alexandra Franca et. Al (2021) developed and compared multiple recommendation systems that utilize machine learning algorithms to provide item suggestions to users based on various data such as user preferences, item characteristics, and user-item interactions. They explore different approaches for training, evaluating, and comparing these recommendation systems to provide an accurate solution for ranking prediction.[1]

D. K. Yadav et al. (2015) introduced MOVREC, a movie recommendation system based on the cooperative filtering approach. Cooperative filtering utilizes user-provided data to recommend movies, prioritizing those with the highest ratings.[2]

Gupta et al. (2015) combined user-specific and item-specific data to form clusters using the Chameleon technique, an efficient method based on class-conscious clustering for recommender systems.

They utilized legal systems to predict item ratings, resulting in lower error rates and better clusters of similar items.[3]
Weng, Lin, and Chen (2007) conducted an evaluation study demonstrating that incorporating multidimensional analysis and additional customer profiles enhances recommendation quality. They utilized the MD recommendation model proposed by Tuzhilin and Adomavicius (2001), showing that multidimensional recommendation models improve recommendation quality.[4]
Lawrence et al. (2001) developed a recommender system based on customer purchase behaviour and history, focusing on suggesting new products in the market. They refined recommendations using both collaborative and content-based filtering approaches, leveraging ratings from previous users to predict and recommend items to potential customers.[5]

III. DATA MINING PROCESS

The fundamental structure of machine learning can be outlined as follows:

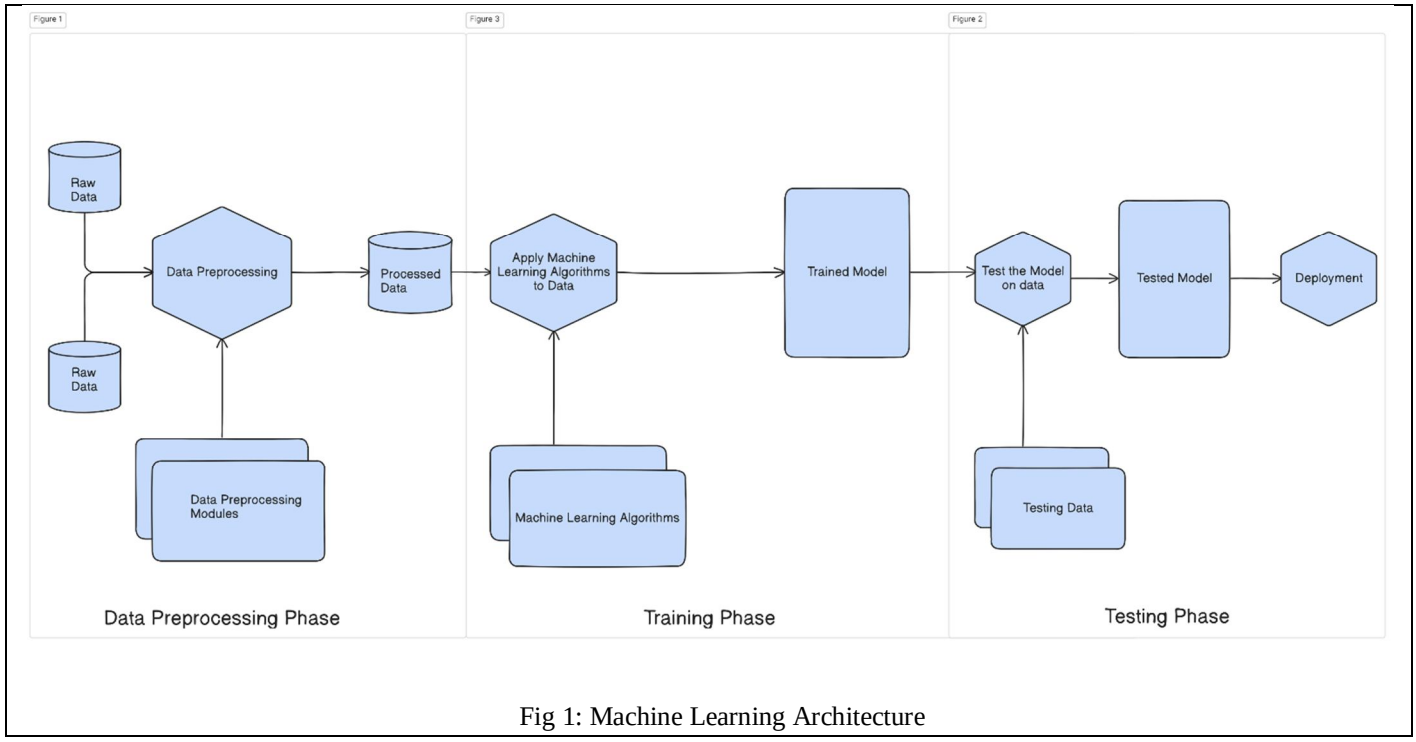


Fig 1: Machine Learning Architecture

The methodology of the proposed system encompasses a series of sequential steps:

- Data-set Collection.
- Pre-Processing.
- Training and Validating.
- Prediction.
- Result.

A. Data-set Collection

The initial phase involves the acquisition of a comprehensive dataset, which serves as the foundational element of the machine learning process. In this context, our dataset comprises an extensive collection of 5000 Hollywood movies, each accompanied by a diverse array of informational attributes.

B. Pre-processing

Within the realm of machine learning, pre-processing assumes a pivotal role in refining and structuring the raw data for subsequent analysis. This task is facilitated through the utilization of specialized libraries, namely "pandas" and "numpy". NumPy, an abbreviation for 'Numerical Python', functions as an open-source module within Python, enabling swift and efficient mathematical computations on arrays and matrices.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue II Feb 2024- Available at www.ijraset.com

Conversely, Pandas stands out as a cornerstone Python library in the field of data science, offering a versatile suite of tools for data manipulation and analysis.

These libraries empower practitioners to perform a multitude of tasks, ranging from the creation of pivot tables to the generation of insightful data visualizations. Moreover, the integration of machine learning techniques, facilitated by Scikit Learn and NLTK (Natural Language Toolkit), further enriches the pre-processing phase. Scikit-learn, renowned as a robust and versatile library for machine learning in Python, furnishes practitioners with an extensive repertoire of tools encompassing classification, regression, clustering, and dimensionality reduction. NLTK, on the other hand, serves as a comprehensive suite of libraries and programs tailored for statistical natural language processing. Notably, stemming and lemmatization techniques, integral to text normalization in language processing, are harnessed to enhance the efficacy of the pre-processing pipeline.

C. Training and Validation

Subsequently, the pre-processed data is serialized into a ".csv" file format for future reference and analysis. This dataset is then partitioned into two distinct subsets: a training set, comprising 70% of the original data, and a testing set, encompassing the remaining 30%. The training phase entails the deployment of various classification models, which are rigorously trained and evaluated to ascertain their predictive accuracy. Validation procedures are subsequently executed to fine-tune the performance of these models, thereby enhancing the overall efficacy of the project.

D. Prediction

The culmination of these efforts manifests in the presentation of algorithmic recommendations, predicated on a meticulous analysis of accuracy and performance metrics. These recommendations are tailored to align with the unique preferences and interests of users, thereby facilitating informed decision-making in the realm of movie selection.

E. Result

Ultimately, the output of the system crystallizes in the form of personalized movie recommendations, reflecting a synthesis of machine learning algorithms and user-centric preferences.

IV. ALGORITHMS USED

A. Count Vectorizer

Count Vectorizer is utilized to convert text data into a numerical representation for predictive modeling. This involves tokenization, where words are parsed and encoded as integers or floating-point values for use in machine learning algorithms. Scikit-learn's Count Vectorizer facilitates this process by converting a collection of text documents into a vector of term/token counts. It also allows for text preprocessing before generating the vector representation, making it a versatile feature extraction module for text data. The Bag of Words technique is often used in conjunction with Count Vectorizer to convert text to vectors.

B. Cosine Similarity

Cosine Similarity is a measure of similarity between two non-zero vectors, typically used in recommendation systems. It calculates the cosine of the angle between the vectors, producing a value ranging from -1 to 1. A value of -1 indicates dissimilarity, 0 represents orthogonality (perpendicularity), and 1 signifies total similarity. Cosine Similarity operates within the positive space, between 0 and 1, and is not affected by variations in magnitude (length) but only represents similarities in orientation. It is often used to calculate similarity between sets of data in recommendation systems.

C. Cross Validation

Cross-validation is a technique used in machine learning to assess the performance of a model and ensure its generalizability to unseen data. It involves splitting the dataset into training and testing subsets, training the model on the training subset, and evaluating its performance on the testing subset. The three steps involved in cross-validation are:

Reserve some portion of the sample dataset.

Train the model using the remaining dataset.

Test the model's performance using the reserved portion of the dataset.

Cross-validation helps ensure that the model learns patterns from the data effectively and does not overfit or generalize poorly to new data. It is crucial for validating the robustness and reliability of machine learning models.

V. PROPOSED METHODOLOGY

The project methodology is structured into six steps:

1) Installation of Python and Jupyter Notebook platform: This step involves installing the Python programming language and the Jupyter Notebook platform.
2) Loading the dataset: The dataset for movie recommendations is required to be imported in ".csv" format.
3) Summarizing the dataset: Sorting and cleaning the dataset is crucial to improve project efficiency. Techniques such as data imputation using the "imputer" function can be employed to handle missing data.
4) Visualizing the dataset: The datasets, "tmdb_5000_movies.csv" and "tmdb_5000_credits," can be visualized through platforms like Kaggle.com. Preprocessing steps are performed on these datasets.
5) Evaluating algorithms: After visualizing the dataset, the project moves to the training and testing phase. The data is divided into a 70:30 ratio for training and testing, respectively. Suitable models such as Count Vectorizer and Cosine Similarity are chosen and trained to determine the accuracy of predictions.
6) Making predictions: The final stage involves making predictions based on user input. Users manually provide their preferences, and the system recommends movies accordingly.

Additionally, for the content-based recommender system, efforts are made to enhance accuracy by finding new methods to represent movies and suggest the top five similar movies based on user interests.

Furthermore, to streamline the project, a frontend website is designed with specific functions for recommendation and display.

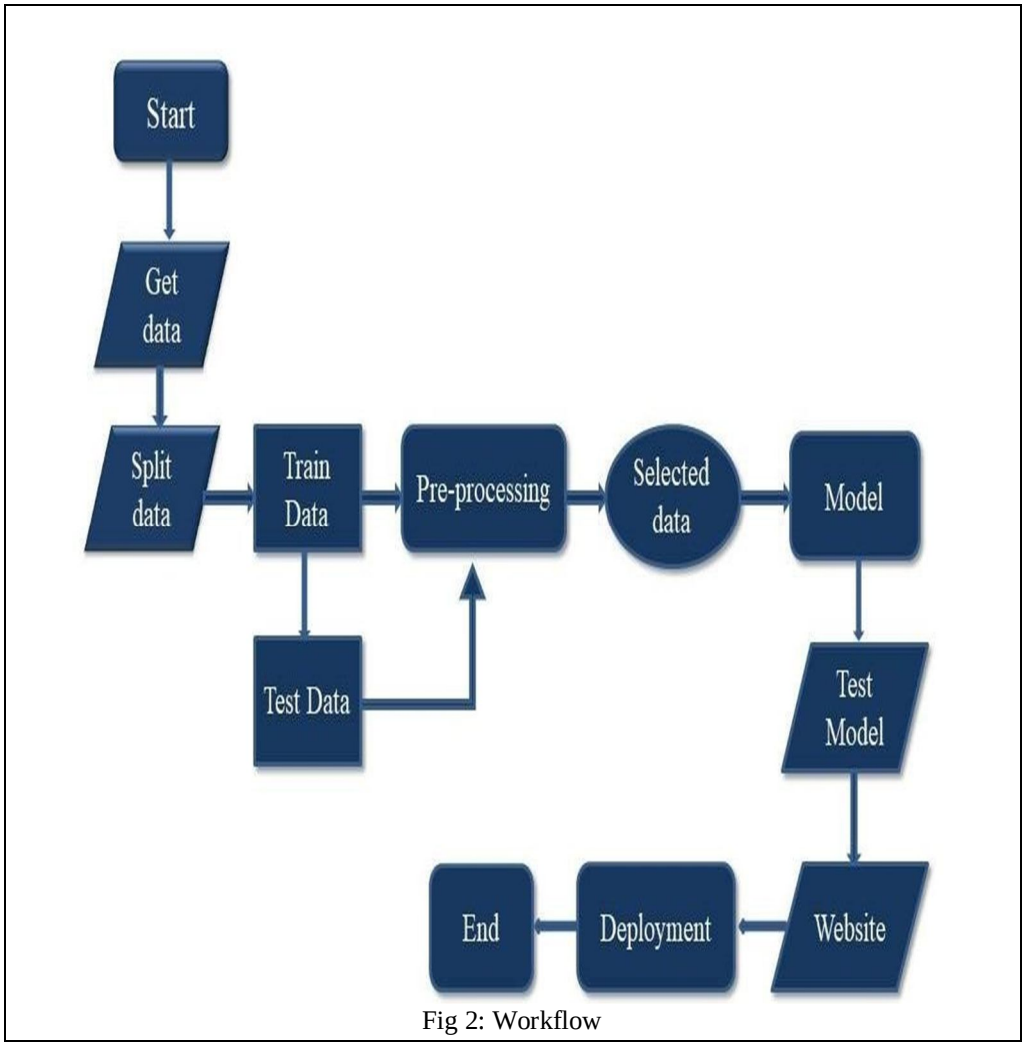


Fig 2: Workflow

VI. RESULT ANALYSIS

After training the model, it is essential to test its performance to ensure its effectiveness in real-world scenarios. This involves using a portion of the dataset specifically reserved for analysis to evaluate the model's proficiency. By testing the model on unseen data, it can be assessed whether it can handle situations that were not part of its training data. In essence, the analysis phase involves using the tested data and the trained model to verify whether the model is functioning properly.

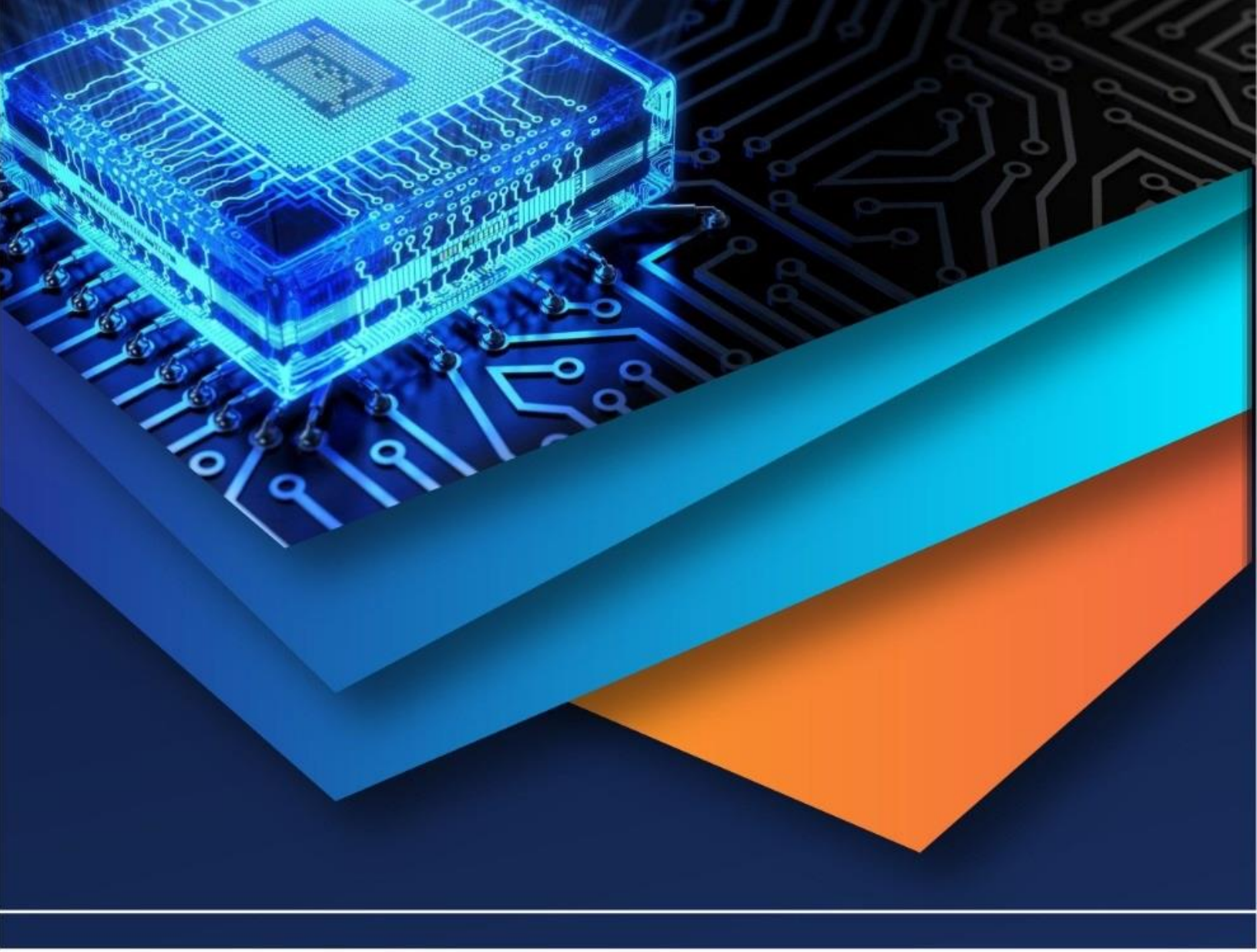
Machine learning is fundamentally about using data to answer questions or make recommendations. Therefore, recommendation or inference is the crucial step where the value of machine learning is realized. This is the culmination of the entire process, where the model is finally put into action to predict whether a similar movie should be recommended to the user based on their interests, leveraging the similarity of movies.

VII. CONCLUSION

The movie recommendation system project represents a significant advancement in the field of personalized movie recommendations. By employing content-based and collaborative filtering techniques, the system successfully provides users with tailored movie suggestions. The project's methodology, involving data collection, pre-processing, model training, and user-friendly interfaces, is a robust foundation for building efficient recommendation systems. While challenges in evaluating the system persist due to the subjective nature of movie preferences, the positive feedback received during initial testing underscores the project's potential. Future work, including the incorporation of larger datasets and diverse algorithms, promises to further enhance the system's accuracy and usability, ultimately delivering a more personalized and enjoyable movie-watching experience for users.

REFERENCES

[1] Manoj Kumar, D.K. Yadav, Ankur Singh, Vijay Kr. Gupta, A Movie Recommender System: MOVREC. International Journal of Computer Applications. 124, 3 (August 2015), 7-11. DOI=10.5120/ijca2015904111.
[2] Recommender system based on Hierarchical Clustering algorithm Chameleon, July 2015, DOI:10.1109/IADCC.2015.7154856, Authors: Utkarsh Gupta, Nagamma Patil
[3] Shreya Agrawal, Pooja Jain et. al. An improved approach for movie recommendation system, Published in: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), DOI: 10.1109/I-SMAC.2017.8058367.
[4] Urszula Ku zelewska. Recommendation system engines. Iranian Journal of Energy and Environment, 2019.
[5] Jing Yu, Jinaiing Shi et. al. Collaborative Filtering Recommendation with Fluctuations of User' Preference, Published in: 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), DOI: 10.1109/ICICSE52190.2021.9404120.
[6] Shweta Sinha et. al. Content-Based Movie Recommendation System: An Enhanced Approach to Personalized Movie Recommendations, May 2023, DOI:10.55524/ijircst.2023.11.3.12.
[7] G.C. Capelleveen, C. Amrit, D.M. Yazan, W.H.M. Zijm, "The recommender canvas: A model for developing and documenting recommender system design". Expert systems with applications, pp.97-117, 2019.
[8] F. Ricci, L. Rokach, B. Shapira, "Recommender Systems: Introduction and Challenges". In: Ricci F., Rokach L., Shapira B. (eds) Recommender Systems Handbook. Springer, Boston, MA, 2015.
[9] G.S. Milovanovic, "Hybrid content-based and collaborative filtering recommendations with {ordinal} logistic regression (1): Feature engineering", Data Science Central, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY