# Movie Recommendation System using Term Frequency-Inverse Document Frequency and Cosine Similarity Method

**N. Muthurasu, Nandhini Rengaraj, Kavitha Conjeevaram Mohan**

***Abstract:*** ***Recommendation engines are trained to produce fast and coherent suggestions to users. This paper describes a hybrid video recommendation system using Term-frequency Inverse document frequency technique for vectorization. Cosine similarity method is used for similarity measure. The system is presented to the user through a web-hosted user-interface. The advantages of the system include efficient recommendations, correct suggestions even with a small data model. Future enhancements include user profiling and documentations, analytics reports for producers and users and data acquirement through web scraping.***

***Keywords: Movie Recommendation systems, Cosine Similarity, TF-IDF***

## I. INTRODUCTION

A Recommendation engine recommends products or items present in a group of data to the users according to their likes and preferences. It is a very popular and crucial part of any commercial website for suggesting products to the users of the website. It allows the user to have sufficient information about the contents of a website and allows them to make informed decisions. Producers of any e-commercial website use the recommendation engine meta-data and data analytics to track the performance, needs and interests of the user market There are various ways of implementing a recommendation engine. Limitations of the present day systems include the lack of data, cold start problems, changing data and user preferences. [1] Some examples of highly efficient recommendation systems include the engines used by multinational tech giants such as YouTube, Netflix, Amazon and Pandora. These systems process tremendous amounts of data at various stages including training stages. A big setback to these systems is the availability of big data, which leads to less efficiency with lesser data availability. Recommendation systems using two primary machine learning approaches- i) Collaborative filtering techniques and ii) Content-based filtering techniques. Some engines, depending on the requirement and operational data, utilize both the approaches and are commonly referred to as Hybrid systems. There are three kinds of recommendations that a system can make. They are- alternative, complementary and generic. Generic recommendations are made with little or no input from the user. Complementary recommendations suggest complementing or similarly related elements. Alternative recommendations provide other similar elements (alternative suggestions). Recommendation Engines, even the one's with extreme efficiency face certain limitations due to the availability of data. Recommendation engines designed for a very large group of customers' (orders of billions) might not accurate for small scale use-cases. As a result smaller e-commerce websites don't have adequate complexities and efficiency to develop a recommendation system. Our goal is to provide a recommendation system with increased accuracy and efficiency for less data and less computational complexity.

## II. RELATED WORK

As a ground work, we examined and studied about different kinds of existing recommendation engines and their uses and limitations. We have presented a concise version of our study here. Xiaokun Wu et al discussed in their paper about the QoS attribute value -based collaborative filtering service recommendation includes two important steps. One is the similarity computation, and the other is the prediction for the QoS attribute value, which the user has not experienced. [2]Xin Guan et al summarized in their paper about collaborative algorithms used to predict other items the current users might like based on the preferences of users for some item in the past. [3] Songtao Shanget al discussed about the slope one algorithm to recommend the items to users by comparing the items. [4] Gilda Moradi Dakhel, Mehregan Mahdavi suggested a recommendation system that uses K-means clustering algorithm to group user based on their interests/likes and suggest videos which are close to users preference. [5] Anant Gupta, Dr.B.K.Tripathy in their paper "A generic hybrid recommender system based on neural networks" describe a cognitive approach towards recommendation engines.

**Revised Manuscript Received on March 25, 2019.**

**N. Muthurasu,** Assistant Professor in Department of Computer Science SRM University, Vadapalani

**Nandhini Rengaraj,** Department of Computer Science SRM University, Vadapalani

**Kavitha Conjeevaram Mohan,** Department of Computer Science SRM University, Vadapalani

*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*

Their paper explains the recurrent neural network with a feed forward network which represents the user-item correlations by integrating, thereby increasing the prediction accuracy. [6]

## III.PROPOSED SYSTEM

Our proposed system is a hybrid movie recommendation system (MRS) that uses Term frequency-Inverse document frequency and cosine similarity algorithm for recommending movies. This system understands customers, their behaviours and activities and trains the system. The main advantage of this system is that, the algorithm is desgined to work efficiently even for a small set of data. The recommendations are based on the movie plot information, history, user profile information, likes and dislikes of the user. It allows the users to save time and future enhancements include a data analytics portal which allows the movie producers to analyze and monitor the performance and preferences 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.

### A. Term Frequency-Inverse Document Frequency

The two basic concepts used in this recommendation system are Term frequency- Inverse Document frequency(TF-IDF) and Cosine Similarity. TF-IDF is used for the vectorization of the data and cosine similarity is used to compute the similarity measure between the vectors.

This method is commonly used as a part of content-based recommendations systems. It consists of two terms. They are Term Frequency(TF) and Inverse Document Frequency(IDF). Term Frequency deals with the frequency of interests and favorites in user profile. Whereas, Inverse document frequency deals with inverse of term frequency among the whole data provided by user profile. These two concepts are combined together in order to provide the recommendation for a user based on the data's provided by user profile. The main purpose of using this concept is to determine the weight-age of the effect of high frequency interests in determining the importance of a recommended video.

The TF-IDF algorithm operates by calculating the relative frequency of words in a specific document compared to the word over the entire document set. Eventually this determines how often the word is used in a particular document. Words that are common in a single or a small group of documents tend to have higher TF-IDF numbers than common words such as articles and prepositions [7] . The implementation procedure process of TF-IDF algorithm has small diversity over all its applications. There are some exceptional case for common words such as prepositions, articles, and pronouns which doesn't hold any appropriate meaning in a query. therefore, such words acquires a very low TF-IDF score, leading to negligence of such words during the search. It is stored as a vector of its attributes in n-dimensional space and angle between vectors is calculated to determine the similarities between the vectors. Assume that N is the total number of documents that are collected and that keyword $k_j$ appears in $n_i$ of them. Moreover, assume that $f_{i,j}$ is the number of times keyword $k_j$ appears in document $d_j$ . Then , the term frequency of keyword $k_j$ in document $d_j$ , is denoted as

$$\square\square_{\square,\square} = \frac{\square_{\square,\square}}{\square\square\square_{\square}\square_{\square,\square}} \qquad (1)$$

The inverse document frequency for word is commonly represented,

$$\square\square\square_{\square} = \log\frac{\square}{\square_{\square}} \;(2)$$

Hence, the TF-IDF weight for keyword in document can be written

$$\square_{\square,\square} = \square\square_{\square,\square} * \square\square\square_{\square} \;(3)$$

In case of synonyms, it is observed that TF-IDF algorithm does not consider the relationship between words which is present in the document. If the user wanted to find information about, say, the word "priest", TF-IDF would not consider documents that might be relevant to the query but instead use the word reverend.[7] In some rare cases, TF-IDF algorithm is unable to recognize the word with its plural form (example: if the searched word is work , then algorithm doesn't consider the plural of the word works) The advantage of this recommendation system is it analyse all the data provided by the user in their user profile and then it recommends the video based on their interests (i.e.,user independence) , no cold start for new item with not enough description or reviews , transparency which explains how the recommender system works, that is represented explicitly by listing features or descriptions. The drawbacks of the used recommendation system is limited content analysis which leads to less accuracy of the recommendation system , very poor at observing the complex behaviours of user based on their user profile , serendipity problem (mind cages for a particular set of users based on their interests) which is also known as over-specialization , new user who doesn't have enough no of ratings required on order to before a content-based recommender system can determine user preferences and provide accurate recommendations .

### B. 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 relevancy between query and document.

Similarity measurement between the user query and document should analyze the meaning of the term. Cosine similarity on the other hand still can't deal with the semantic meaning of the query very well. Semantic meaning problem does not meet the difference of syntax matching. Since Information Retrieval system may produce unstable result and it might lead to lack of its utmost performance. Research has been done on similar problem and it is found that WordNet is use. WordNet is most common method used because of utilization of lexical database as semantic network. This results in improvement of determining the cosine similarity along with semantic analysis between the two vectors. This application desires to increase the accuracy of the similarity value between two vectors. In document-query cases, a document can be represented as a term vector that the vector's dimensions refer to the terms available in the document.[8] Dimension's value is occurrence of term inside a document.[8] A document can be represented as a form of vector as:

$$\square = (\square_{\square 0}, \square_{\square 1}, \dots, \square_{\square\square}) \quad (4)$$

As same as the document, the user query's term can be represented as a form of vector as:

$$\square = (\square_{\square 0}, \square_{\square 1}, \dots, \square_{\square\square}) \quad (5)$$

Where $w_{di}$ and $w_{qi}$ are float numbers denotes the frequency of each term present inside a document, while each vector's dimension corresponds to a term available in the document.[8][9]

Based on vector similarity, similarity between two vectors can be represented as:

$$\sin(\square, \square) = \frac{\square \cdot \square}{|\square||\square|} = \frac{\sum_{\square=1}^{\square} \square_{\square\square} \cdot \square_{\square\square}}{\sqrt{\sum_{\square=1}^{\square}(\square_{\square\square})^2} \cdot \sqrt{\sum_{\square=1}^{\square}(\square_{\square\square})^2}} \quad (6)$$

Null distribution is one of the error cases in case of cosine similarity. While determining the similarity value between two vectors may result in negative as well as positive. Therefore cosine similarity of null distribution is distribution of dot product of two independent vectors. This distribution has a mean of zero and a variance of 1/n (where n is the number of dimensions), and although the distribution is bounded between -1 and +1, as n grows large the distribution is increasingly well-approximated by the normal distribution [9][10]. For other types of data, such as bit streams (taking values of 0 or 1 only), the null distribution will take a different form, and may have a nonzero mean. [11]

### C. Architecture

The architecture diagram of a system explains explicitly all the modules and process flow implied in the process. It provides an overview of what is to be done, in order to complete the process and helps in distributing each module among the group. The architecture diagram provides with the information of how its organised. The behaviour of the process can be predicted through the architecture diagram provided by the developer. This section explains briefly the modules used in our proposed system.

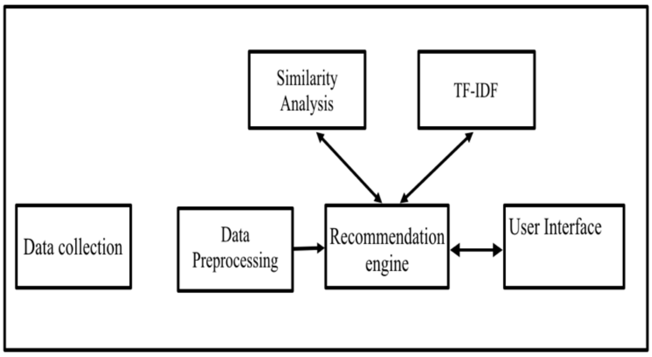

Figure 1: Architecture diagram

- **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.

- **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 number-MovieID. 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.

- **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.

The engine is trained to produce similarity measures using the training data. The backend is scripted using Python language along with jQuery for handling Http requests.

- **User Interface**

The user interface is be designed in such a way that it relates to the real world environment, making it unnecessary for the user to remember any particular command or an input. The web interface which is designed for this system created using a micro-framework in python called flask. Flask works on Jinja template engine with the help of Werkzeug toolkit. It has full unicode support, an optional integrated sandboxed execution environment, widely used and BSD licensed. [12] The web-page for user interface is constructed using HTML, css, Jquery and mySQL.

## IV.RESULTS

Though the Movie recommendation system constructed by other developers have used any one of the filtering techniques, they had faced drawbacks which were little disturbing. In our project we had implemented both content-based and collaborative-based filtering, which makes our system as a hybrid system. By implementing both in a single system we had overcome drawbacks from both types of filtering techniques. The movie recommendations system requires a very large amount of data to work efficiently, which is lacking for many small scale commercial websites. But our system does overcome this problem too, as it works efficiently when provided with small amount of data which is a boon for small scale commercial websites. The advantage of this recommendation system is it analyze all the data provided by the user in their user profile and then it recommends the video based on their interests (i.e., user independence) , no cold start for new item with not enough description or reviews , transparency which explains how the recommender system works, that is represented explicitly by listing features or descriptions.

## V.LIMITATIONS

The drawbacks of the used recommendation system is limited content analysis which leads to less accuracy of the recommendation system , very poor at observing the complex behaviours of user based on their user profile , serendipity problem (mind cages for a particular set of users based on their interests) which is also known as over-specialization , new user who doesn't have enough no of ratings required on order to before a content-based recommender system can determine user preferences and provide accurate recommendations .

Considering all the advantages our system has, there are even certain potential limitations which are needed to by overcome by our system in future. One of those limitations are such that our system doesn't take users or clients reviews and likes (likes similar to that are present in Facebook) about the movie into account while training the data to recommend the movies to users. This is not yet implemented in our system due to certain problems like sarcasm and homophones, which are needed to be taken care for. This requires a lot of data in order to work efficiently. Another limitation we face is that, it only works efficiently in small amount of data. This is because, a system to handle a very large amount of data requires a very complex system, which should even analyse data present in the video too. One more limitation we face in our system is that changing user's preferences. A user's interest and preferences may change from time to time. Understanding such behaviour and recommending customised movies according to their mood and preferences are difficult. This is one of our future implementation we had planned for to overcome.

## VI.FUTURE ENHANCEMENTS

The user interface through the website can be made easier to access through user profiling. The user's previous like, dislikes etc. can be stored. Data regarding the movies for which the recommendations are requested can also be gathered from several websites using web scraping or web data extraction tools after obtaining legal permissions. This can also be developed as a standalone application (engine) which can be used by small e-commerce site vendors to acquire and attach to their sites. For processing large scale data, the application can be integrated along with intelligent data analyses using big data techniques to provide authentic and accurate analytics. Sentiment analysis can also be applied to "comments" information to identify the emotion behind the comments (positive, negative or neutral) to recommend movies appropriately.

## VII.CONCLUSION

Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations. Vast amounts of data required to produce accurate recommendations open scope to other applications such as incorporation of big data tools and efficient data processing methodologies. Another challenge in the changing data. Data in the system is nowhere constant and is subjected to continuous fluctuations as a result of varying user behaviours and preferences. Recommendations in line with these interests are a lot more intricate to generate than simpler generic recommendations. Every system is weighted with its advantages and limitations. The key to finding the perfect recommendation system for the user's needs lies in identifying the correct algorithm for the data to be processed and balancing the outcomes with a combination of similarity algorithms

## REFERENCES

1. Richar Macmanus. Redwrite.com 5 Problems of Recommender Sys-tems. Web Source. Published on Jan 28, 2009 in WEB.
2. Xiaokun Wu, Bo Cheng, and Junliang Chen. Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method. IEEE Transactions on Services Computing( Volume: 10, Issue: 3, May-June 1 2017 ). 352 – 365

3. Xin Guan, Chang-tsun Li, and Yu Guan. Matrix Factorization With Rating Completion : An Enhanced SVD Model for Collaborative Filtering Recommender Systems. IEEE Access (Volume: 5 ). 24 November 2017. 27668 – 27678
4. Songtao Shang, Minyoung Shi, Wenqian Shng, Zhiguo Hong. A Micro-video Recommendation System Based On Big Data. Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Con-ference. 29-26 June 2016
5. Gilda Moradi Dakhel, Mehregan Mahdavi. A New Collaborative Fil-tering Algorithm Using K-means Clustering and Neighbor's Voting. International Conference on Hybrid Intelligent Systems. 2011. 179-184
6. Anant Gupta, Dr.B.K.Tripathy .A generic hybrid recommender system based on neural networks. Advance Computing Conference (IACC), 2014 IEEE International. 21-22 February 2014.
7. Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855
8. Semantic Cosine Similarity Faisal Rahutomo*, Teruaki Kitasuka, and Masayoshi Aritsugi Graduate School of Science and Technology, Ku-mamoto University
9. Spruill, Marcus C (2007). "Asymptotic distribution of coordinates on high dimensional spheres". Electronic communications in probability.
10. CrossValidated: Distribution of dot products between two random unit vectors in RD
11. Graham L. Giller (2012). "The Statistical Properties of Random Bit-streams and the Sampling Distribution of Cosine Similarity". Giller Investments Research Notes (20121024/1).
12. https://en.wikipedia.org/. Flask Web Framework. Web Source. Wikipedia.
13. Nandhini Rengaraj. C.M.Kavitha. .Sabitha Loganathan. N.Muthurasu. A study of existing systems of recommendation engines. National Conference of Emerging Computing Technologies and Applications. Vel Tech University. Avadi, Chennai. 05-06 April 2018.