ISSN: 2584-0495 Vol. 3, Issue 2, pp. 1584-1593

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Maximizing User Satisfaction with Machine Learning-Powered Movie Recommender Systems

Sarita Kumari, Kusumika Krori Dutta, Harsha Karamchandani and Kehkeshan Jallal S

Cite as: Kumari, S., Dutta, K. K., Karamchandani, H., & S, K. J. (2025). Maximizing User Satisfaction with Machine Learning-Powered Movie Recommender Systems. International Journal of Microsystems and IoT, 3(2), 1584–1593. https://doi.org/10.5281/zenodo.18144674

Published online: 25 February 2025

https://doi.org/10.5281/zenodo.18144674

International Journal of Microsystems and IoT
Vol.3, Issue 2, pp.1584-1593; DOI: https://doi.org/10.5281/zenodo.18144674

Maximizing User Satisfaction with Machine Learning-Powered Movie Recommender Systems

Sarita Kumari[1], Kusumika Krori Dutta[2], Harsha Karamchandani[3] and Kehkeshan Jallal S[4]

[1]Department of ECE, Amity University Jharkhand, Ranchi, India
[2]Department of AIML, Dayananda Sagar College of Engineering, Bangalore, India
[3]Department of ECE, Nitte Meenakshi Institute of Technology, Bangalore, India
[4]Department of ECE, Presidency University, Bangalore, India

ABSTRACT

The goal of recommendation systems is to provide customers with practical and sensible recommendations for products or goods they might be interested in. A recommendation engine extracts data and, using a variety of techniques, suggests to customers the most important stuff. Content-driven filtering (CF), item-based collaborative filtering (IBCF), and the K-Nearest algorithm (KNN) are movie recommendation strategies used in this study. These methods make an effort to filter users' preferences using the data collected and present movies based on that profile. The MovieLens dataset is used by all three algorithms to produce the Cosine Similarity index. Both cold start capabilities and the issue of data sparsity are addressed. For a sample of 910 films, the actual and expected ratings are displayed using Tableau visualization tools. Further precision is calculated when the evaluation assessment using Root Mean Square Error (RMSE) is completed. According to the experimental findings, item-based collaborative filtering, out of the three algorithms, produces the best results with the least amount of mistake and the greatest degree of precision (84.9%).

KEYWORDS:
Recommendation system,
Content-based filtering,
Item-based collaborative filtering,
K-Nearest Neighbor Tableau.

1. Introduction

In today's digital age, the availability of a vast amount of movie content can make it overwhelming for users to decide what to watch. Movie recommendation systems powered by machine learning techniques have emerged as a valuable solution to address this challenge. These systems leverage the power of data analysis and predictive algorithms to provide personalized movie recommendations to users, enhancing their movie-watching experience. A movie recommendation system utilizes various machine learning approaches to analyze user preferences, movie attributes, and historical data to generate accurate and relevant recommendations. By understanding user behavior and movie characteristics, these systems aim to match users with movies that align with their tastes and preferences. The core objective of a movie recommendation system is to maximize user satisfaction by offering tailored suggestions that resonate with their unique preferences. Machine learning (ML) and Deep Learning (DL) [1][2] algorithms play a vital role in achieving this goal by learning from user interactions and patterns to generate recommendations that are highly likely to be well-received.

The success of a movie recommendation system relies on effective data collection, feature engineering, algorithm selection, and continuous learning and adaptation based on user feedback. It is crucial to consider factors like personalization, explain ability, diversity, and real-time updates to ensure that users receive recommendations that align with their evolving interests.

In summary, machine learning-powered movie recommendation systems leverage algorithms and data analysis to provide personalized movie suggestions to users.

By analyzing user preferences and movie attributes, these systems aim to maximize user satisfaction, enhance movie discovery, and create a more engaging and enjoyable movie-watching experience. Content-based filtering gives customers a set of predefined criteria based on their personal preferences, although product features aren't necessarily present in the dataset. Without prior knowledge of product features, the KNN approach can recommend related products, but it requires a high user similarity index. Item-based collaborative filtering, on the other hand, employs both product attributes and user preferences, but it necessitates a huge amount of data to train with a high number of parameters.

This paper focuses on ML based movie recommendation systems. Section 2 deals with literatures available followed by smart recommendation fundamentals which are discussed in section 3. Section 4 gives a brief of methodology used followed by section 5 and section 6 discusses the results and conclusion of the work.

2. Literature Survey

Recommendation system is categorized into three classes: Collaborative Filtering, Content-based, and hybrid-based Approach" [3]. Content-based recommendation mechanisms are limited to individuals; they do not prescribe stuff out of the box, thus restricting the ability to learn further [4]. The hybrid movie recommendation engine has solved this constraint of personality. A hybrid engine can suggest films to users according to their preferences and recommend films ranked by other users that are equivalent to the consumer [5].

In [8][11], the KNN algorithm is applied along with the cosine similarity theorem as it provides more precision than the other distance metrics, and the complexity is comparatively low. The authors of the paper [2] offer an insight into the common recommendation methods, and algorithms for optimization. The most relevant guidelines are the collective filtering strategy along with the time-varying multi-armed optimization algorithm. As per [9], the authors have used Apache Spark and Elastic search to enhance the processing and time calculation for a large dataset. The precision is measured between the two matrix factorization algorithms, Alternative Least Square and Singular Value Decomposition, for different research subset values. "The hybrid approach overcomes drawbacks of each algorithm and improves the performance of the system" [5].

Table 1: Comparison of recommendation and prediction algorithm implemented by various Authors.

Method	**Remarks**	**Applications**
Collaborative Filtering [6][11][15][16]	prowess CF algorithm - user-based approach and Item-based approaches.	environmental sensing, movie suggestion, financial services, marketing
Hybrid [15]	System efficiency is increased.	Movie recommendation system (MRS), Books recommendation.
ALS algorithm [16]	Personalized movie recommendation using feedback	Used by Google news, Facebook, Netflix etc.
RNN method [17] Ante RNN[28]	LSTM model, recommendation using text and vision data	Stock market prediction
Star rating based [4]	LSTM model The proposed STAR model outperforms many of the existing models.	movie reviews, product reviews, teaching reviews, hotel reviews, etc.
Hidden Markov Model (HMM) [7]	95.6 % accuracy for depression detection	Depression detection
Integrated automated Review mining System (IRSC) in Cloud environment [16]	Efficient review mining and sentiment analysis	Market research, training data for various models, product analysis, and review analysis.
Docker technology [17]	sentiment analysis over Cloud	Online review sites, twitter, Facebook.
Content based filtering [10]	Doc2Vec and Tf-Idf are used in hybrid mode.	Mobiles, Books, T.V. Shows prior to its release in the market
Deep Learning with collaborative filtering [22]	The accuracy of the test set is 95.89% for the Seq2Seq system based on the LSTM .	Recommendation based on social media
K-means algorithm using cuckoo search algorithm [23]	The algorithm gives 0.68 Mean Absolute Error (MAE)	E-Commerce websites
Natural language processing (NLP) tool [24]	SVM method for sentiment analysis	Analyzing the movie reviews
Naïve Bayes and support vector machine (SVM) [25]	NB accuracy is 97.33% while SVM is 98.63%	Sentiment analysis
Clustering with Missing Value prediction[26]	Recommendation generation time is reduced	Missing Values in matrix are predicted
Random forest based recommendation tion [27]	High prediction accuracy, and exclusive music for particular users	Personalized music recommendation

Missing values of the rating matrix can be predicted by integrated algorithm [26]. ML based random forest algorithm recommends personalized music with high accuracy for particular users [27]. Ante RNN provides a more transparent recommendation system. Ante RNN or Attentive RNN is the dynamic recommendations system both text an visual fusion [28]. AutoML for deep recommender systems (AutoRecSys) produces well-performing deep recommender systems in a data-oriented and task-specific manner as opposed to conventional recommender systems, which require experts to develop a specific model [26]. The comparison of different recommendation systems are explained in Table 1. It was reported that AutoML can be implemented for various applications such as on-device recommender systems, social Recommendation [31, 32], sequential recommendation Tasks [33, 34], GNNs-based Recommendations [33] etc. There are many new recommendations apps such as StreamRec[35] which supports stream processing system and produces real-time recommendations. A Movie Recommender System: MOVREC allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes and using K-means algorithm [36]. In order to protect the privacy of the user federated learning is applied to the movie recommendation systems. The common privacy mechanisms and privacy protection techniques can be used for movie recommendation[37]

3. Recommendation System Technique

There are several recommendation system techniques commonly used in machine learning. Here are three popular approaches: i) content based filter, ii) collaborative filter and iii) hybrid method. The choice of recommendation technique depends on factors such as the available data, system requirements, and the nature of the problem being solved. It is also worth considering factors like scalability, interpretability, and the ability to handle cold-start scenarios (where there is limited user data). Experimentation and evaluation of different techniques can help identify the most effective approach for a specific application.

3.1 Content –Based filtering

Content-based filtering recommends items to users based on the characteristics or features of the items themselves. In the case of movies, these characteristics can include genre, actors, directors, plot keywords, or even user-generated tags. The system builds a user profile based on their previous preferences, and then recommends items that have similar features to the ones the user has previously liked. Content-based filtering is particularly useful when there is limited or no user history available. It extracts the similarity of the item and implements the TF-IDF (Term Frequency-Inverse Document Frequency) [8] [10].

$$Tf(t) = \frac{Frequency\ occurance\ of\ term\ t\ in\ the\ document}{Total\ number\ of\ terms\ in\ document} \quad (1)$$

$$If(t) = \log 10\ \frac{Total\ number\ of\ document}{Number\ of\ documents\ containing\ term\ t} \quad (2)$$

Three methods that can compute the similarities between vectors are Cosine similarity, Euclidian distance and Pearson's correlation [5-8].

3.2 Collaborative filtering

Collaborative filtering is based on the idea that users who have similar preferences in the past are likely to have similar preferences in the future. This technique analyzes user behavior, such as movie ratings or purchase history, and identifies patterns and similarities among users. It then recommends items that users with similar tastes have liked or consumed. Collaborative filtering can be further divided into two subtypes:

3.2.1 Item- Based Collaborative Filtering

This approach focuses on the similarity between items rather than users. It identifies items that are similar to the ones the target user has liked and recommends those similar items. In this system, the similarity is measured on an item that the user's rate. The following is the recommended procedure: Assume there may be a list of z users as shown in Figure 1.
$P = (P_1, P_2, P_3 \ldots P_z)$ as well as a list of n items:
$Q = (Q_1, Q_2, Q_3 \ldots Q_n)$ and also the user-item rating matrix R_{zxn} is :

$$R_{zxn} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & R_{1j} & R_{1n} \\ R_{31} & R_{32} & R_{33} & R_{3j} & R_{3n} \\ R_{i1} & R_{i2} & R_{i3} & R_{ij} & R_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_{z1} & R_{z2} & R_{z3} & R_{zj} & R_{zn} \end{bmatrix}$$

Figure 1: Rating matrix R_{mxn}

Where R_{ij} represents the user *I* rating of the item *j* and represents the user *I* preference for the item *j*. By considering each item as a vector and determining the cosine of the angle created by the vectors, the similarity of two items may be calculated. P and Q are vectors and *n* is their size; In vector cosine similarity formula between P and Q, the vectors P and Q are more similar when sim_{cos} (P, Q) is larger sim_{cos} (P, Q) = 0 indicates that the two vectors are completely different, where as sim_{cos} (P, Q) = 1 indicates that they are fully similar.

3.2.2 User-based Collaborative filtering

This approach finds users who are similar to the target user based on their ratings and recommends items that those similar users have liked.

3.3 Hybrid Methods

Hybrid recommendation systems combine multiple techniques to provide more accurate and diverse recommendations. For example, a hybrid system can integrate collaborative filtering and content-based filtering to leverage the advantages of both approaches. It can also incorporate other techniques such as matrix factorization, deep learning models, or reinforcement learning to enhance recommendation accuracy.

3.4 Evaluation criteria

Two metrics are used to evaluate the recommended results: Root mean square Error (RMSE) and Precision. The nature of the research data in this study requires the use of a method that is sensitive to error performance. As a result, the root mean square error (RMSE) analysis will be adopted in statistical accuracy evaluation. RMSE may better reflect the accuracy of experimental data measurement. RMSE evaluates the accuracy of a system by comparing the numerical recommendation scores against the actual [20].

$$\text{RMSE} = \sqrt{\frac{(PU1-P1)^2+(PU2-P2)^2+\cdots+(PU-Pn)^2}{n}} \quad (3)$$

$$= \sqrt{\frac{\sum_{i=1}^{n}(PUi-Ui)^2}{n}}$$; where PUi is the predicted score and Ui is the observed score.

The smaller the RMSE, the better the recommender system predicts data since the deviation between actual and predicted data is smaller [20].

Precision is the fraction of relevant recommendations to the total Recommendations [19]. *Precision is calculated as:*

$$\text{Precision} = \frac{\text{Relevantrecommendations}}{\text{TotalRecommendations}} \quad (4)$$

4 METHODOLOGY

The following methodology is adopted in this research work.

4.1 Dataset

100,000 ratings (1-5) from 943 people were collected in this data collection for 1682 films. A minimum of 20 films have been rated by each user. Age, gender, occupation, and zip code are just a few examples of the demographic data used to determine users [12].

4.2 Data preprocessing

Only the columns u.data, u.item, and u.user from the pool of datasets have been utilised. The majority of the other sections were removed because they weren't needed for these tests. The dataset that needed to be resolved had numerous duplicate and blank values [12].

The complete U.Data set has 100000 ratings from 943 people on 1682 items. At least 20 films have been rated by each user. u.item - Information about the items (movies); this is a tab-separated list containing movie id, movie title, release date, and video release date. Users and items are numbered consecutively starting with 1.

u.user is a tab that contains demographic data about the users. User ID, age, gender, occupation, and zip code are listed separately. In the u.data data, the user ids are employed in Figure 2.

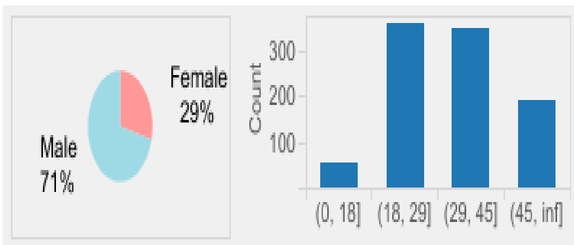


Figure 2: Gender and age group

4.3 Proposed solution

In this paper an online dashboard where users could choose a movie and view the suggestions has been proposed. The three approaches we preferred were content based filtering (CF), item-based collaborative filtering and KNN nearest neighbor algorithm. Tableau is used to develop an assessment metric RMSE error and Precision and a statistical analysis of actual and projected ratings to determine the effectiveness of three algorithms. Using the Association matrix, An exploration to choose several movies is conducted. The cosine similarity technique for recommenders with this data set is used, because it contains more movies and ratings. A converse sparsity formula is used to solve the issue of sparsity. The cosine similarity is favorable because, even if the Euclidean distance separates two comparable data items due to their size, they may have a shorter angle between them. The greater the similarity is the smaller the angle between them. When displayed on an optimization graph, cosine similarity captures the orientation angle of the data items rather than the magnitude. After obtaining the lowest RMSE error item-based collaborative filtering gives much accurate results when compared to the other two approaches. The flow of the proposed method is illustrated in Figure 3.

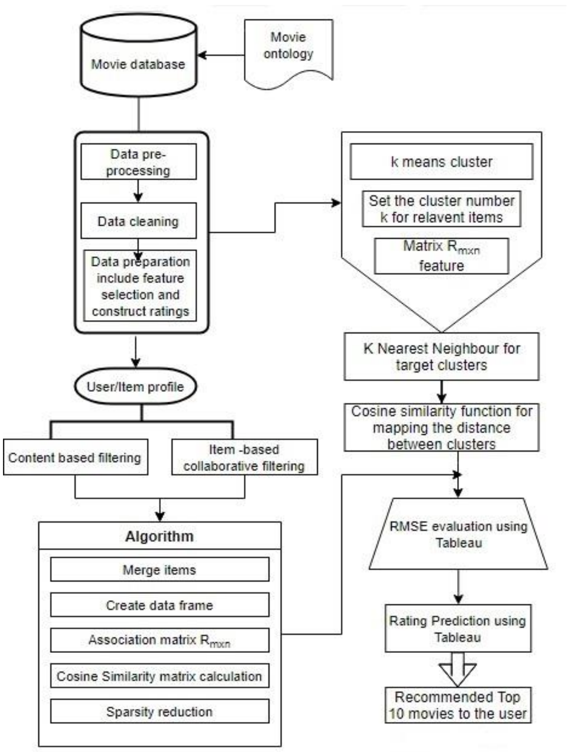


Figure 3: Proposed Smart recommender system for movie selection application.

4.4 K- Nearest Neighbor Algorithm

K-Nearest Neighbor is one of the simplest supervised learning technique-based Machine Learning algorithms. The K-NN algorithm stores all data available and classifies, based on similarities, a new data point. This suggests that it can be conveniently grouped into a well-suited class using the K-NN algorithm as new data emerges. The K-NN algorithm can be used both for regression and classification, but it is mainly used for problems of classification.

Assuming D is data available a, D= $\{(x_1, y_1)\ldots\ldots(x_n,y_n)\}$; where each of n is some part of data .$x_i \epsilon R^d$ and $y_i \epsilon \{0,1\}$ where d is the real dimensional space and y_i is the binary classification. If there is a new point x and to classify the y of this point then we need to classify the k nearest point of that new data. In the case of k=1,

Distance matrix $d(x_i,x_j)$; where $x_i,x_j \epsilon R^d$

$$d(x_i, x_j)=\|x_i - x_j\|^2=\sum_{k=1}^{d}\left(x_i k - x_j k\right)^2 \quad (5)$$

where; $x_i= (x_i1, x_i2, x_i3, \ldots.. x_id)$

Probabilistic interpretation when k is fixed. Random variable Y~P where P(y) = fraction of points x_i in $N_k(x)$ such that y_i= y. Where; $N_k(x)$ is the k nearest point to x

Conditional probability is P (y|x, D) for which

$$\hat{y} = \underset{y}{avgmax}\, P(y|x, D) \quad (6)$$

where $\hat{y}$ is to predict the value of y

For example, in Figure 4, if there is a movie that is similar to the category of movies A and B, but it has to be categorized as, movie A or B. This KNN model will find the similar features of the new data set to the movie A and movie B and based on the most similar feature it will put it in either movie A or B category. In this proposed model, Cosine distance to get nearest neighbor has been used.

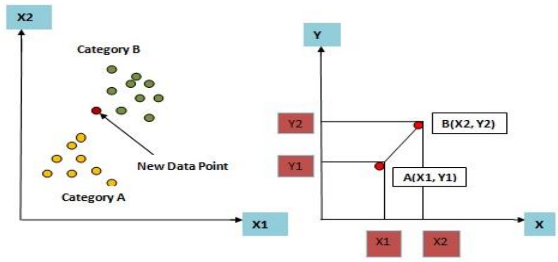


Figure 4: Working of K-NN model

4.5 Cosine Similarity Computation

The estimation of the items rated by two users from Table 2 is used to determine the similarity between them. To compute the similarity of v1 and v3, first figure out the set of movies that they all scored as {m1, m2, m4, m5} and comparative scores of these movies. v1's score vector is {1, 4, 3, 2} while v3's score vector is {2, 3, 1, 5}. The similarity formula computes the similarity of v1 and v3. The similarity between v and v' is indicated by sim (v, v'), and the most popular technique of measuring user similarity is Cosine Similarity. Cosine similarity is a method used regardless of their size to determine how identical the records are. Numerically, the angle of the cosine between two vectors represented in a multidimensional feature space is determined. The lower the angle is, the greater the relation to cosine.

Table 2: Computing Similarity between users

V/M	m1	m2	m3	m4	m5
v1	1	4	4	3	2
v2	3	2	4		
v3	2	3		1	5
v4	2		2		

$$sim_{cos}(v, v') \text{ or } (P, Q) = \frac{P.Q}{\|P\|\times\|Q\|} = \frac{\sum_{i=1}^{n} P_i \times Q_i}{\sqrt{\sum_{i=1}^{n} P_i^2}\times\sqrt{\sum_{i=1}^{n} Q_i^2}} \quad (6)$$

This method has been implemented to calculate how similar movies are dependent on the various properties of their similarities. Statistically, the angle of two vectors drawn in multidimensional space is shown by the cosine. The resemblance of cosine is quite useful because it helps to find identical objects [8].
The cosine angle will evaluate the relation between the two films. The θ has a spectrum of 0-1. If another value of θ falls close to 1, it is more comparable, and if it is closer to 0, it's also least comparable. If it is identical to 1, the film would be suggested, and then there will be no similarity among them. In Python, the cosine has been computed using the Scikit - learn library. The pairwise distances produce is *1-cosine similarity*. Cosine Similarity is the number between 0 and 1.

4.6 Algorithm Approach

Approach 1: Content-based filtering-
This approach filters user profile characteristics such as age, gender, ethnicity, demographic details, and many more to improve the recommendation rate [2].This technique separates the products based on the user's likings. Input 1 contains all the parameters required for content-based filtering.
*Approach 2: Item-based Collaborative filtering-*Each item in this paper is represented by a movie. Vectors are constructed using all of the reviews for every movie and then the cosine similarity of the vectors is computed. When a user did not evaluate a movie, the vector contains many 0 values to fill in the null values, the cosine similarity between all of the movies was computed. Input 2 consists of all the labels for user ratings.

Input 1: User profile such as user ID, age, gender, occupation, zip code, movie genre (Action, Adventure, Animation, Children, Comedy, Crime, Documentary, and drama, Fantasy, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western) of MovieLens100K.
Input 2: User profile such as movie ID, item ID, ratings, timestamp, video release date, IMDb URL of MovieLens100K.
Output: Recommends set of 10 Movies

Proposed algorithm for both the approaches:

Approach 1 and 2: Content-based and Item-based Collaborative filtering Algorithm

a) Load three datasets*u.user, u.data, u.item*
b) Description of each movie set *user. describe ( )*
c) Merge the *item_id* and *movie_ID* data dataset
d) Create a Dataframe to make *movie_ID, movie_ title, user _ID, item_id ratings and timestamp* in one frame.
e) Create and apply the pivoted table for it in terms of a matrix (learning Matrix) R_{mxn}
f) LearningMatrix = ratings.pivot_table (index=['item_id'], columns=['user_id'],values='rating').reset index(drop=True learningMatrix.fillna (0, inplace=True)
g) learningMatrix.head (20)
h) Association Matrix R_{mxn} Pivot table is created
i) Apply Cosine similarity formula on association matrix between the two objects
j) Pairwise distance is calculated for the matrix.
k) movie similarity =1 - pairwise_distances (learningMatrix, metric="cosine")
l) np.fill_diagonal (movie_similarity, 0)
m) Reduce the sparsity using csr_matrix function
n) csr_data=csr_matrix(dataset_final.values)
o) Display Recommendation for the given movie.

Approach 3: KNN nearest Neighbor - following the computation of user similarity as sim (v, v'), the algorithm picks a number of users with the highest similarity as V's neighbor, indicated as v'. Set a preset value K for neighbor determination, and choose just the neighbors with the highest K similarity, giving little attention to user estimation of neighbor comparability. After deciding the users' neighbor, the score of the item's neighbor can be used to predict the score.

4.7 Model Building

To create a recommender framework, the MovieLens 100k dataset is used. The first step is to import pandas and Numpy and then using pandas' read _csv () utility, dataset was loaded. The dataframewas created in order to see that the data frame is in terms of user ID or not. Now once the table is being projected and the Cosine similarity formula is applied in the Association matrix. Cosine Similarity is used to find the distance, to know how the data is correlated in each data frame. The cosine similarity for k=11 and n=5 is shown in Table 3.

Table 3 Calculated Cosine similarity

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.000000	0.402382	0.330245	0.454938	0.286714	0.116344	0.620979	0.481114	0.496288	0.273935	0.468291	0.460392
1	0.402382	0.000000	0.273069	0.502571	0.318836	0.083563	0.383403	0.337002	0.255252	0.171082	0.468506	0.459946
2	0.330245	0.273069	0.000000	0.324866	0.212957	0.106722	0.372921	0.200794	0.273669	0.158104	0.361165	0.319295
3	0.454938	0.502571	0.324866	0.000000	0.334239	0.090308	0.489283	0.490236	0.419044	0.252561	0.588337	0.584884
4	0.286714	0.318836	0.212957	0.334239	0.000000	0.037299	0.334769	0.259161	0.272448	0.055453	0.375809	0.373824
5	0.116344	0.083563	0.106722	0.090308	0.037299	0.000000	0.139617	0.083876	0.151064	0.203097	0.063987	0.144471

5 Result and Discussion

5.1 Data Analysis

Using Python's Matplotlib tools and Tableau, the dataset is examined to acquire insight into the movie dataset that may help in the design of the proposed system. As shown in Table 4, trends are found, such as the most rated movies, users' descriptions, and the movie's title in each rating group as shown in Figure 5 and Figure 6.

Table 4. Movie rating based on ID

User ID	Movie ID	Rating
19	4	4
475	50	5
571	64	4

100,000 records x 3 fields

The KNN algorithm is explained below.

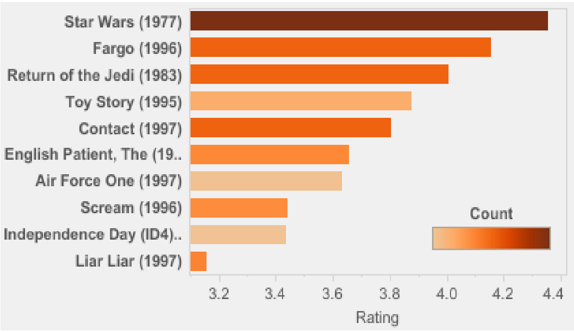


Figure 5: Top rated movies with average ratings.

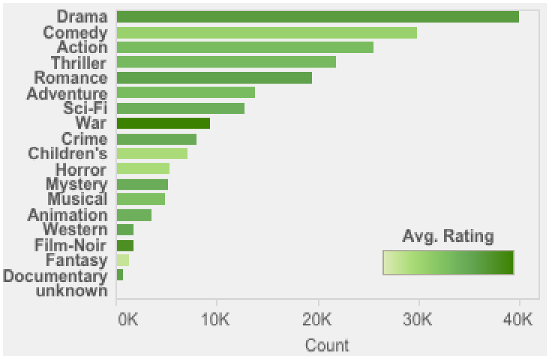


Figure 6: Popular genre of all gender and age group

Table 5: Head of the user data set

User Id	Age	Gender	Occupation	Zip code
1	24	M	Technician	85711
2	53	F	Other	94043
3	23	M	Writer	32067
4	24	M	Technician	43537
5	33	F	Other	15213

An online dashboard is created using three approaches where users may choose a movie and analyze the recommendations. Analyzing the Table 5, it is found that the reference movie list contains the 10 most similar movies found by the content-based filtering, Item based collaborative filtering and K-NN model using the relevance similarity scores. The conclusions seem fair. The most related movies to Dead Man Talking, for instance, are Leaving Las Vegas, which is also released in the same year (1995) and with the same IMDb ratings. A movie that this system recommends is on the basis of the similarity index of every data set in the database. This is because both of these films have identical scores of significances that are high for tags relevant to subjects such as ethnicity, time of release, IMDB ratings, age group user choice, Zip code, and profession, which is entirely dependent on the three different proposed approaches.

Table 6 shows the reference movie title based on which recommendations are to be made: Dead Man Walking (1995) recommended movies based on the choice of dead man walking (1995).

Table 6: Movie recommendation of Dead man walking (1995)

Approach 3: K Nearest Neighbor Algorithm

a) Create a two-dimensional scoring matrix R_{mxn} based on the user and the items.
b) Calculate user similarity between matrix using
c) *sim (v, v')*
d) Based on Step 2 assign the neighbors' number K
e) The cosine distance of K is determined as the number of neighbors
f) Take the K closest neighbors based on the calculated cosine distance. knn=NearestNeighbors(metric='cosine',algorithm 'knn', n_neighbours=10)
g) knn.fit (csr_data)
h) Count the number of data points in each group among these k neighbors.
i) Add new data points to the segment for which the neighbor's maximum number is set.
j) Compute the estimation

Movie_id		Movie_title	Similarity
275	276	Leaving Las Vegas (1995)	0. 590753
126	127	Godfather, The (1972)	0.529239
6	7	Twelve Monkeys (1995)	0.527462
236	237	Jerry Maguire (1996)	0.527137
507	508	People vs. Larry Flynt, The (1996)	0.509791
49	50	Star Wars (1977)	0.509013
123	124	Lone Star (1996)	0.505800
0	1	Toy Story (1995)	0.496288
97	98	Silence of the Lambs, The (1991)	0.494959

5.2 Comparison of different Root Mean Square Error (RMSE)

For all the three models, Root mean square error (RMSE) is calculated in Tableau and then compared. The best results are shown bold in Table 7.

Table 7: Comparisons of different RMSE value

Proposed Technique	Estimated Regularization parameter	RMSE
Approach 1	λ=1	0.75
Approach 2	λ=10	0.54
Approach 3	k=5	0.65

The optimization value of regularization parameter for the three approaches is calculated as shown in Figure 7. It is observed from the figure that each of the proposed model gives lowest Error at λ= 1, λ=10, k=5, of the Linear regression y~βX

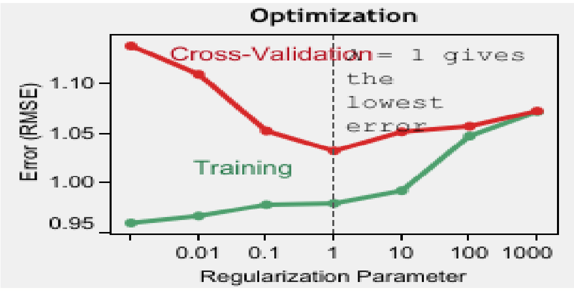
where, Response variable y = rating mean normalized by age group and gender. Independent variable X=movie genres. Fitting parameter β= user preference towards rating and genres.

The costs function of Content-based filtering:

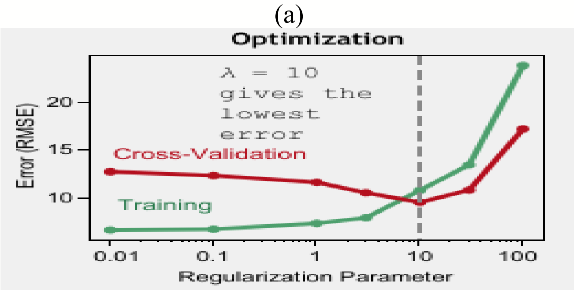
$$J(\beta, \lambda) = |y - \beta X|^2 + \lambda|\beta|^2 \qquad (7)$$

The cost function of Collaborative based filtering:

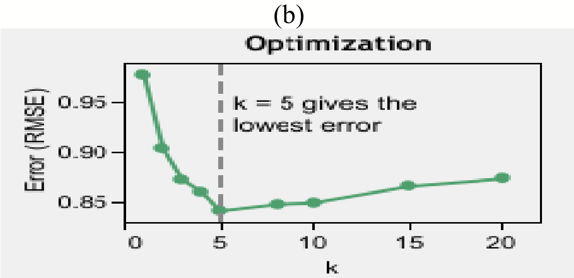
$$J(\beta, X, \lambda) = |y - \beta X|^2 + \lambda|\beta|^2 + \lambda|X|^2 \qquad (8)$$



(a)



(b)



(c)

Figure 7: (a) Error (RMSE) vs. Regularization Parameter (λ) for approach 1

(b) Error (RMSE) vs. Regularization Parameter (λ) for approach 2

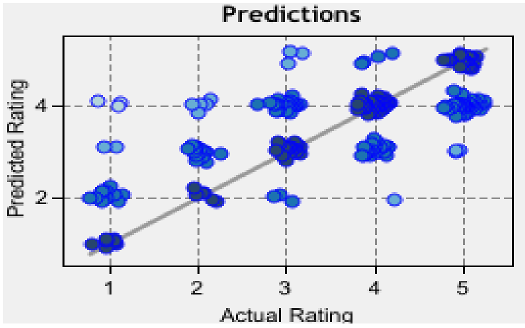
(c) Error (RMSE) vs. Regularization Parameter (k) for approach 3.

https://public.tableau.com/views/MovieRecommendation [Accessed on 15 October, 2021]

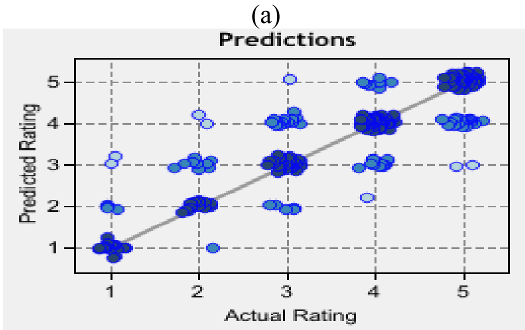
However it is observed that whenever the value of "λ" is increased the error also increases in Approach 1 and Approach 2 and when the value is kept very low as 1e-03 still the error increases in Approach 1 and approach 2 whereas in Approach 3 error increases when the value "k" decreases. Approach 2 gives minimum RMSE errors out of the three approaches.

6.3 Comparison of the different algorithms on predicted and actual ratings for Precision

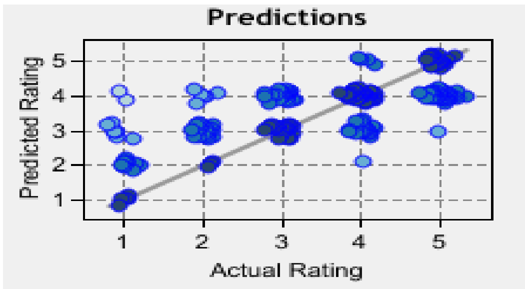
After obtaining the lowest RMSE error, it can be observed that the item-based collaborative filtering gives accurate results when compared to Approach 1 and Approach 3.Due to the sparsity of dataset only ratings of similar movies was considered, as a result there weren't many such movies for many of the users, for rating. To overcome this issue, a straightforward solution could not be obtained hence visualization of the actual and predicted rating was done. This technique indicated the best algorithm that can predict ratings by uploading the dataset in tableau.



(a)



(b)



(c)

Figure 8: (a) Rating prediction graph for approach 1 (b) Rating prediction graph for approach 2 (c) Rating prediction graphs for approach 3.
https://public.tableau.com/views/MovieRecommendation [Accessed on 15 October, 2021]

Figure 8 illustrates the rating prediction graph for all the three approaches. The straight line (m=100) in the above graph is the standard parameter for comparing the actual and predicted ratings. When precision at m=100, in approach 1 and approach 3 the clusters are sparsely distributed away from the Standard line of observance. When precision at m=100, in approach 2, most of the clusters form around the standard line. As a result, the predicted rating of approach 2 matches with the actual ratings and precision of Item-based Collaborative filtering is better than the Content-based approach and KNN approach.

Table 8: Precision of the proposed recommender System

Sample size(s) = 910 movies

Approach	Actual predicting ratings	Actual predicting ratings	Precision
1.Content based filtering	586	324	64.3%
2.Item-based collaborative filtering	773	137	84.9%
3.K-NN	650	260	71.4%

The precision of the recommender system is shown in the Table 8 for Content-based, Item-based Collaborative Filtering, and K-NN techniques. According to the proposed model, the Item-based Collaborative filtering approach has a higher precision than the Content-based and KNN techniques. As the number of users increases, the item-based collaborative filtering approach outperforms the other two, overcoming the limitations of both techniques. Figure 9 shows the Precision-based comparative study of recommendation approaches.

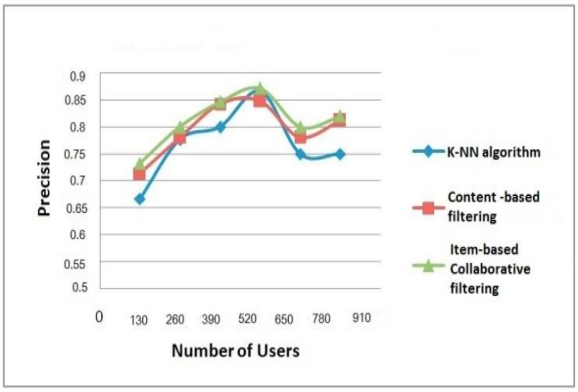


Figure 9: Precision of the recommendation technique

The proposed model gives the top 10 recommended movie list after calculating cosine similarity. This research paper addresses the problem emphasized by the authors of the paper [1] [8] [13] [20] [21]. Two main problems were isolated, mainly due to the sparsity of the data: The first problem that was encountered, was when the items with only one common user were compared to those with two common users. Since the user rating is the most important variable in the calculation, the items with only one common user got the highest adjusted-cosine similarity values. The second challenge was that only 10 similar movies were stored for each test user-movie pair. Since user viewed movies were considered, it resulted in unsatisfied overall predictions for large test sets. After obtaining the minimum RMSE error of 0.54, item-based collaborative filtering gives much more accurate results when compared to content-based and KNN techniques. Hence, the performance of Item-based collaborative filtering gives higher accuracy and precision around 84.9%, which increases the performance of the Smart recommendation system for movie suggestions.

6 CONCLUSION AND FUTURE WORK

The proposed smart recommender model for movie selection makes recommendations based on the user's previous behavior and that of similar users. These findings help to better understand the similarities and differences between the three main recommendation systems. Based on the results in the dataset, it can be inferred that the item-based Collaborative filtering method has an exciting feature: its predictions are mostly significant in smaller recommendation lists than those performed by CF and KNN. Furthermore, this approach can be extended to analyze qualitative information about each film, such as its decade, major actor, director, length, and cost, to improve the recommendation. The proposed system makes it very simple to implement, easy, and precise suggestions. The proposed work may be advantageous to many over the top (OTT) media platforms to indulge in advertising campaigns by helping customers to judge upcoming movies with the suggestion. Soon, authors will be work on improving the user-based recommendations by providing plenty of reviews and designing an algorithm to predict ratings.

ACKNOWLEDGEMENT

In this research work, Movie Lens Data Set was used. MovieLens data sets were collected by the Group Lens Research Project at the University of Minnesota.

REFERENCES

[1] K Indira, KK Dutta, S Poornima, SA Swamy Bellary, book chapter titled "Deep Learning Methods for Data Science", ch7 of book titled "Advanced Analytics and Deep Learning Models", May2022, Publisher John Wiley & Sons, Inc.PP 149-179. DOI: 10.1002/9781119792437.ch7

[2] KK Dutta, S Poornima, R Sharma, D Nair, PG Ploeger, book chapter titled, "Applications of Recurrent Neural Network: Overview and Case Studies", book titled," Recurrent Neural Networks", June 2022, CRC press pp 23-41.eBook ISBN9781003307822

[3] Pooja and Mr. Bhupender Sharma, "Movie Recommendation System : A Review Report," *J. Res.*, vol. 04, no. 01, pp. 53–58, 2018, [Online]. Available:http://www.journal4research.org/articles/J4RV4I1015.pdf.

[4] F. Furtado and A. Singh, "Movie recommendation system using machine learning," *Int. J. Res. Ind. Eng.*, vol. 9, no. 1, pp. 84–98, 2020, doi: 10.22105/riej.2020.226178.1128.

[5] A. Vijayvergia and K. Kumar, "STAR: rating of reviewS by exploiting variation in emoTions using trAnsferleaRning framework," 2018 Conference on Information and Communication Technology (CICT), 2018, pp. 1-6, doi: 10.1109/INFOCOMTECH.2018.8722356.

[6] C. M. Wu, D. Garg and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.

[7] Babu, Sonal& P, Madhu. (2020). A Survey on Various Techniques for Personal Movie Recommendation. International Journal of Scientific Research in Science, Engineering and Technology. 156-163. 10.32628/IJSRSET207143.

[8] G. Arora and A. Kumar, "Movie Recommendation System Based on Users'Similarity," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 4, pp. 765–770, 2014, [Online]. Available: http://ijcsmc.com/docs/papers/April2014/V3I4201494.pdf.

[9] H. Ansari, A. Vijayvergia and K. Kumar, "DCR-HMM: Depression detection based on Content Rating using Hidden Markov Model," 2018 Conference on Information and Communication Technology (CICT), 2018, pp. 1-6, doi: 10.1109/INFOCOMTECH.2018.8722410.

[10] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, "Movie Recommendation System using Cosine Similarity and KNN," *Int. J. Eng.*

Adv. Technol., vol. 9, no. 5, pp. 556–559, 2020, doi: 10.35940/ijeat.e9666.069520

[11] R. Mhetre and D. P. G, “Movie Recommendation Engine using Collaborative Filtering with Alternative Least Square and Singular Value Decomposition Algorithms,” *Ijarcce*, vol. 8, no. 2, pp. 88–92, 2019, doi: 10.17148/ijarcce.2019.8216

[12] R. Singla, S. Gupta, A. Gupta and D. K. Vishwakarma, "FLEX: A Content Based Movie Recommender," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154163.

[13] D. K. Behera and M. Das, “A Research on Collaborative Filtering Based Movie Recommendations: From Neighborhood to Deep Learning Based System,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 10809–10814, 2019, doi: 10.35940/ijrte.d4362.118419.

[14] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (IIS) 5, 4, Article 19 (December 2015), 19 pages. DOI-http://dx.doi.org/10.1145/2827872

[15] Thakker, U., Patel, R. & Shah, M. A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimed Tools Appl* 80, 28647–28672 (2021). https://doi.org/10.1007/s11042-021-10965-2.

[16] Miryala, Goutham & Gomes, Rahul & Dayananda, Karanam. (2017). COMPARATIVE ANALYSIS OF MOVIE RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING IN SPARK ENGINE. Journal of Global Research in Computer Science. 8. 10-14.

[17] Adil Moghar, MhamedHamiche, Stock Market Prediction Using LSTM Recurrent Neural Network, Procedia Computer Science, Volume 170, 2020, Pages 1168-1173, ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.03.049.

[18] S. Kumar and K. Kumar, "IRSC: Integrated automated Review mining System using Virtual Machines in Cloud environment," 2018 Conference on Information and Communication Technology (CICT), 2018, pp. 1-6, doi: 10.1109/INFOCOMTECH.2018.8722387.

[19] K. Kumar and M. Kurhekar, "Sentimentalizer: Docker container utility over Cloud," 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR), 2017, pp. 1-6, doi: 10.1109/ICAPR.2017.8593104.

[20] K. Kumar, R. Bamrara, P. Gupta, and N. Singh, “M2P2: Movie’s Trailer Reviews Based Movie Popularity Prediction System,” in *Soft Computing: Theories and Applications*, 2020, pp. 671–681.

[21] M. Chandak, S. Girase, and D. Mukhopadhyay, “Introducing hybrid technique for optimization of book recommender system,” *Procedia Comput. Sci.*, vol. 45, no. C, pp. 23–31, 2015, doi: 10.1016/j.procs.2015.03.075.

[22] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, “Personalized real-time movie recommendation system: Practical prototype and evaluation,” *Tsinghua Sci. Technol.*, vol. 25, no. 2, pp. 180–191, 2020, doi: 10.26599/TST.2018.9010118.

[23] Mishra, Nitin & Chaturvedi, Saumya & Vij, Aanchal & Tripathi, Sunita. (2021). Research Problems in Recommender systems. Journal of Physics: Conference Series. 1717. 012002. doi: 10.1088/1742-6596/1717/1/012002.

[24] Liu, J., Choi, W. H., & Liu, J. (2021). Personalized movie recommendation method based on deep learning. *Mathematical Problems in Engineering*,volume *2021*,https://doi.org/10.1155/2021/6694237

[25] Rahul Katarya, Om Prakash Verma,An effective collaborative movie recommender system with cuckoo search,

[26] C. Xiong, Y. Liu, Z. Yang and E. Liu, "Improving Recommendation Performance with Clustering and Missing Value Prediction," 2020 54th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 2020, pp. 1-6, doi: 10.1109/CISS48834.2020.1570617370.

[27] Z. Shu, Q. Shen and T. Zeng, "Research and Application of Music Personalized Recommendation System Based on Random Forest Algorithm," 2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2023, pp. 1387-1391, doi: 10.1109/ITNEC56291.2023.10082503.

[28] Peng Liu, Lemei Zhang, Jon Atle Gulla, “Dynamic attention-based explainable recommendation with textual and visual fusion” , Information Processing & Management, Volume 57, Issue 6, 2020, ISSN 0306-4573

[29] Egyptian Informatics Journal,Volume 18, Issue 2,2017,Pages 105-112,ISSN 1110-8665,https://doi.org/10.1016/j.eij.2016.10.002

[30] N. Kapoor, S. Vishal and K. K. S., "Movie Recommendation System Using NLP Tools," 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 883-888, doi: 10.1109/ICCES48766.2020.9137993.

[31] N Pavitha, Vithika Pungliya, Ankur Raut, Roshita Bhonsle, Atharva Purohit, Aayushi Patel, R Shashidhar,Movie recommendation and sentiment analysis using machine learning,Global Transitions Proceedings,Volume 3, Issue 1,2022,Pages 279-284,ISSN 2666-285X,https://doi.org/10.1016/j.gltp.2022.03.012.

[32] Zheng, Ruiqi, Liang Qu, Bin Cui, Yuhui Shi, and Hongzhi Yin. "AutoML for Deep Recommender Systems: A Survey." ACM Transactions on Information Systems (2023).

[33] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, Jiali Yang, and Xiaofang Zhou. 2019. Social Influence-Based Group Representation Learning for Group Recommendation. In 2019 IEEE 35th International Conference on Data Engineering (ICDE). 566–577. https://doi.org/10.1109/ICDE.2019.00057

[34] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. SelfSupervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In Proceedings of the Web Conference 2021 (Ljubljana, Slovenia) (WWW ’21). Association for Computing Machinery, New York, NY, USA, 413–424. https://doi.org/10.1145/3442381.3449844

[35] Chandramouli, Badrish and Levandoski, Justin J. and Eldawy, Ahmed and Mokbel, Mohamed F.. “StreamRec: a real-time recommender system” Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, 2011, Association for Computing Machinery, Athens, Greece,1243–1246,https://doi.org/10.1145/1989323.1989465

[36] Manoj Kumar, Dharmendra Kumar Yadav, Ankur Singh, Vijay Kr, “A Movie Recommender System: MOVREC”, International Journal of Computer Applications, International Journal of Computer Applications, Volume 124 – No.3, August 2015

[37] Zehua Sun, Yonghui Xu, Yong Liu, Wei He, Lanju Kong, Fangzhao Wu, Yali Jiang, Lizhen Cui, “A Survey on Federated Recommendation Systems”, IEEE Transactions on Neural Networks and Learning Systems, VOL. 14, NO. 8, December 2022

[38] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: a domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. arXiv preprint arXiv:2105.03300 (2021)

[39] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Guandong Xu, and Quoc Viet Hung Nguyen. 2022. On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (Madrid, Spain) (SIGIR ’22). Association for Computing Machinery, New York, NY, USA, 546–555. https://doi.org/10.1145/3477495.3531775

AUTHORS

Sarita Kumari has been Assistant Professor in the Department of Electronics and Communication Engineering at Amity University Jharkhand since September 2018. She completed her B.E. degree in Instrumentation Engineering from SLIET (Longowal), M.E. degree in control and Instrumentation Engineering from VMU (Salem) and Ph. D. in Engineering from Birla Institute of Technology (Mesra) in 2005, 2009 and 2019 respectively. She has 19+ years of teaching and research experience. Her research interests include sensor design, magneto-optic sensor, biomedical instrumentation. She has published research papers in SCI and Scopus indexed journals and book chapters.

Corresponding Author Email: gs.sarita@gmail.com

Kusumika Krori Dutta received the B.E. degree in electrical engineering from Jorhat Engineering College, Assam, India, in 2001 and the M.Sc.(Engg) by research degrees in Super conducting fault current limiter from VTU, Karnataka in 2011. PhD in epilepsy diagnostic using machine learning from VTU. Currently, she is an Associate Professor at the Department of AI&ML, Dayananda Sagar College of Engineering, Bangalore. She has authored or coauthored more than42 refereed journal and conference papers, 10 book chapters, with Elsevier and Springer and 11 patent filed out of which 5 got grant and 2 published. Her research interests include digital signal processing, digital image processing, biomedical signal processing, machine learning, deep learning, artificial neural networks, artificial intelligence applied electrical system.

Email: kusumika@msrit.edu.

Harsha Karamachandani is having 15 years of rich teaching experience in Electronics & Communication Engineering, the author has published 18 papers in international journals and has presented 17 papers in national and international conferences. She is currently working as an Assistant professor in Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore, INDIA.

Email: harsha.mkaramchandani@nmit.ac.in

Kehkeshan Jalall S is currently pursuing her Doctoral degree at Presidency University, Bangalore, India. She is working at Presidency university as faculty of engineering since February 2020. She holds a master's degree in VLSI Design and Embedded System from Visvesvaraya Technological University, Belgaum, in 2010, and Bachelor's degree in 2000 from Bangalore University, India. She has 19+ years of experience in the area of teaching and research. Her areas of area include Image Processing, VLSI Design and Machine Learning.

Email: kehkeshan@presidencyuniversity.in