Volume: 9 Issue: XII Month of publication: December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39538

A Movie Recommendation System Based on A Convolutional Neural Network

Jyoti Kumari[1], Dr. Sanjiv Sharma[2]
[1, 2]Madhav Institute of Technology and Science, Gwalior(India)

***Abstract:** Due to its vast applications in several sectors, the recommender system has gotten a lot of interest and has been investigated by academics in recent years. The ability to comprehend and apply the context of recommendation requests is critical to the success of any current recommender system. Nowadays, the suggestion system makes it simple to locate the items we require. Movie recommendation systems are intended to assist movie fans by advising which movie to see without needing users to go through the time-consuming and complicated method of selecting a film from a large number of thousands or millions of options. The goal of this research is to reduce human effort by recommending movies based on the user's preferences. This paper introduces a method for a movie recommendation system based on a convolutional neural network with individual features layers of users and movies performed by analyzing user activity and proposing higher-rated films to them. The proposed CNN approach on the MovieLens-1m dataset outperforms the other conventional approaches and gives accurate recommendation results.*

***Keywords:** Recommender system, convolutional neural network, movielens-1m, cosine similarity, Collaborative filtering, content-based filtering.*

I. INTRODUCTION

Recommender systems (RS) have proven to be a successful technique for dealing with the ever-increasing volume of internet information. Given the widespread implementation in many web applications, as well as its ability to alleviate numerous problems linked to over-choice, the value of RS cannot be emphasized. Deep learning (DL) has evolved in several fields because of its exceptional performance and the enticing characteristic of learning feature representations from scratch. The impact of DL is also persistent, with new research showing its utility in information extraction and recommender systems. The field of DL in recommender systems is booming[1].

With the huge quantity of information available on the internet and the rise in online purchasing, the necessity for RS to lead people toward their preferences became clear. Users are the set of components of a system that interact with each other in RS, while items are indeed the set of entities that can be liked or viewed by users. When the number of consumers or items in a recommender system expands exponentially at an incredible rate, scalability becomes a concern, hence a recommendation approach should be quick and efficient on both small and large datasets. The cold-start problem, on either side, is caused by a lack of information about new users or objects, which affects prediction accuracy. Liben-Nowell and Kleinberg developed one of the ways to deal with this problem in 2007, which finds similarities between two entities in a network[2].

RSs predict users' preferences for things and make proactive recommendations to them. collaborative filtering, content-based, and hybrid recommender systems are the three types of recommendation models. When processing information for the suggestion, RSs utilizing a collaborative method consider the user data. For example, the RS can obtain all user data, including age, country, city, and songs purchased, by browsing profiles in a digital music store. The algorithm can utilize this data to identify users who have similar musical tastes and then recommend songs to them.

Content-based filtering RSs provide suggestions based on the existing items they have access to. Consider the case of a user who is searching for a different computer on the internet. When a user searches for a specific computer (item), the RS collects data about it and searches a database for computers with similar characteristics, including price, CPU speed, and memory capacity. The user is then given recommendations based on the results of the search[3].The RSs major goal is to deliver recommendations and suggestions that are based on the preferences and choices of our customers. Koren et al.[4] concentrated on recommendation accuracy. The author claims that when a great number of user reviews and ratings are evaluated to produce efficient and accurate suggestions, the quality of the suggestions becomes extremely crucial. Booking through online platforms and recommenders has grown in popularity in recent years, and global corporations are attempting to capitalize on this trend. Customers can benefit from an RS not just in terms of locating appropriate hotels, but also in terms of movies, books, and a variety of other products and items. RS can analyze a variety of data kinds, including hotels, movies, and music. Figure 1 depicts the generic recommender architecture.

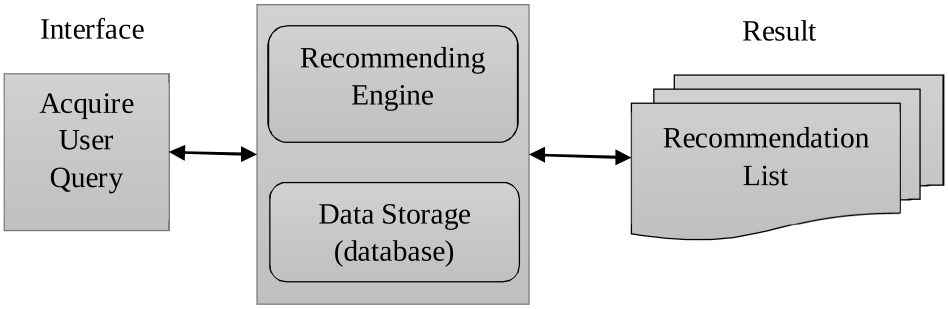


Figure 1- A general framework for recommendation system.

II. RELATED WORK

For utilizing user experiences on the Internet, RS is unavoidable. They employ a variety of methods to recommend the Top-K things to consumers based on their tastes. RSs have become one of the most essential components of large-scale mining strategies in recent years. A DL approach for a collaborative RS is proposed (DLCRS). They compared the suggested method to existing methods in their research. On 100k and 1M Movie Lens datasets, the model was verified. The outcomes show that the suggested approach outpaces other approaches, demonstrating that incorporating a DL methodology into a recommender system is a worthwhile endeavor[5]. GMF, which employs a linear kernel to represent latent feature interaction, and CNN, which utilizes a non-linear kernel to learn the interaction function from any input, were combined in this RS. On the MovieLens-1 m dataset, numerous experiments revealed that the suggested CNNCF model outperformed all state-of-the-art approaches in several situations, with the greatest result being 0.7068 for HR@10 and 0.4259 for NDCG@10 and with 3 layers of CNN[6]. This research presents a CNN-based RS for textile products that consider color compatibility. For training and testing the CNN model, they employed their pattern dataset of 12000 photos. The training of the RS is continued after receiving feedback over some time. The system's performance will improve as a result of the feedback. Overall accuracy was 82.08 percent, precision 82.00 percent, recall 83.50 percent, and f1-score 82.30 percent were obtained[7]. To produce more trustworthy recommendations that can manage cold start and overfitting concerns, the article presents a DL approach based on collaborative filtering and combining stochastic gradient optimization techniques. Multiple units are associated with highly identifiable items in user and item-based collaborative filtering. These items are used for model training that predicts user ratings on new products that make conclusions and recommendations. In terms of MAE and RMSE, the suggested model's experimental results were compared to those of state-of-the-art models. To overcome the cold-start problem for new items, the authors presented a Hybrid Movie RS that blends collaborative filtering and content-based filtering. HMRS-RA will reduce the cold start problem for new films based on their genre by taking into account relevant data such as genre. The suggested methodology (HMRS-RA) tackles the scalability problem by reducing the data's dimensionality by clustering. When contrasted to some conventional and recent work in RSs, they enhanced the suggestion performance by incorporating resource allocation as a weight for utilizing similar users in each cluster. The MAEs of the suggested algorithm are 0.23, 0.34, 0.23, and 0.34 for men, women, 20-39, respectively, according to the experimental data. As a result, HMRS-RA improved suggestion accuracy[2]. The author adapted and changed book RSs as part of the project. This Book Recommender system took into account several factors, including ratings, book title, cover image, and price. They used cosine similarity to successfully deploy and find related books on Amazon and Flipkart. In addition, the system was built utilizing a dataset of book covers and a CNN approach to identify the most similar book covers based on the input book cover[8].

III. PROPOSED METHOD

Neural Network (NN) consists of a series of interconnected nodes that represent the activity of the brain. Every node has a weighted association in neighbouring layers with several different nodes. The node uses the data from the respective nodes & the weights together with the basic ability to calculate output values. In numerous forms and models, neural systems come. Due to the complexity of the problem, the NN architecture must be determined by the user including several hidden layers, no. of nodes, and their connectivity in specific hidden layers. It is capable of learning and modelling nonlinear and dynamic relationships, this is very important because many input-output-to-output relations in real life are both nonlinear or complex. In the wake of gaining from the underlying information sources and their connections, it can surmise concealed connections on inconspicuous information also, along these lines influencing the model, to sum up, and anticipate concealed information[9].

CNN- CNN is a widely used paradigm for deep learning inspired by animals' visual cortex. Convents are like ordinary neural networks and may be viewed as an acyclic neuron array. A neural network varies primarily from it a neuron of the secret layer is linked only in the previous layer with a subset of neurons. Owing to this sparse connectivity, it can learn implicit characteristics[10].
Convolutional Layer-This layer constitutes a fundamental unit of a Conv Net, which includes calculations. It is a set of signature maps of neurons. The layer parameters are a series of filters or kernels that can be analysed. Such filters are complemented by the feature maps to generate a two dimensional activation map that shows the output volume as it is stacked along the depth axis. Weight (parameter sharing) of neurons in the same map decreases network size by the number of parameters. A hyperparameter called the receptive field is a spatial extension of sparse relation amid two-layer neurons[11].

The outcome $Y_i^{(l)}$ of layer 1 involves $m_1^{(l)}$feature maps of size $m_2^{(l)} \times m_3^{(l)}$. The ith feature map signified $Y_i^{(l)}$ (equation 1), is calculate as

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)} \qquad (1)$$

where $B_i^{(l)}$is a bias matrix & $K_{i,j}^{(l)}$ is the filter of size $2h_1^{(l)} \times 2h_1^{(l)}$connecting the jth feature map in layer (l−1) with the feature map in a layer.

Pooling Layer- Basic Conv Net architecture has alternate conv or group layers so these functions minimize (without loss of information) spatial dimensions of activation maps & no. of parameters in a network, thus minimizing total device complications [76]. This tests the overfitting issue. The bundling operations include max pooling, average pooling, stochastic pooling, spectral reselling, spatial pyramid bundling, or multi-scale order less pooling. Layer 1 has 2 hyperparameters, the spatial extent of the filter $F^{(l)}$ & the stride $S^{(l)}$. It takes an input of size $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$ & delivers an outcome of size $m_1^{(l)} \times m_2^{(1)} \times m_3^{(1)}$ is shown in equation (2).

where;

$$m_1^{(l)} = m_1^{(l-1)}$$

$$m_2^{(1)} = m_2^{(l-1)} - F^{(1)}/S^{(1)} + 1$$

$$m_3^{(1)} = m_3^{(l-1)} - F^{(1)}/S^{(1)} + 1 \qquad (2)$$

Fully Connected Layer- It's used to categorize things. For classification, it employs a softmax activation function. The term "Fully Connected" implies that each neuron in the previous layer is linked to each neuron in the subsequent layer. Higher-level features are represented by the outcomes from earlier layers. This layer's main goal is to classify the input using the higher-level features.
f(x)=max (0, x)
As a methodology of feature extraction, CNN employs learned convolutional kernels or filters. The overall aim is to use learned features as the feature representation for a recommendation rather than pre-designed features. Deep CNNs that have recently been developed combine numerous convolutional layers with fully connected layers. Higher-level features are frequently learned and discriminative elements are accentuated as the network's depth is increased.
Finally taking into account the study made on different research papers, we propose a movie recommendation system. The functional model is created with individual features layers of users, movies, CNN, and a fully connected layer. After training, we use cosine similarity measures to find similar movies related to the various user aspects. Once the user watches the required movie, the system recommends the movies similar to the users watching.

IV. PROPOSED ALGORITHM

1) Step 1. Start
2) Step 2. Collect the MovieLens-1m dataset in three files users.dat, movies.dat, and ratings.dat acquired by the Group Lens Research Project at the University of Minnesota.
3) Step 3. Performing EDA on the dataset, after analysis we found that user data and movies can be used for training and ratings can be used for predicting and as labels.
4) Step 4. Plotting graphs and charts to understand every bit of dataset like top ten movies rated by users, movies liked by users of age 25-35, etc

5) Step 5. Perform the data pre-processing to extract the relevant features by following steps-
a) Gender: change F to 0 and change M to 1.
b) Age: transform to continuous numbers from 0 to 7.
c) Genres: transform to digits.
d) Title: same as Genres.

6) Step 6. Create a functional model with individual features layers of users, movies, CNN, and a fully connected layer.
7) Step 7. Fit the model with the data and let it train, after train, we measure the accuracy of Training and testing both.
8) Step 8. End

V. PROPOSED FLOWCHART

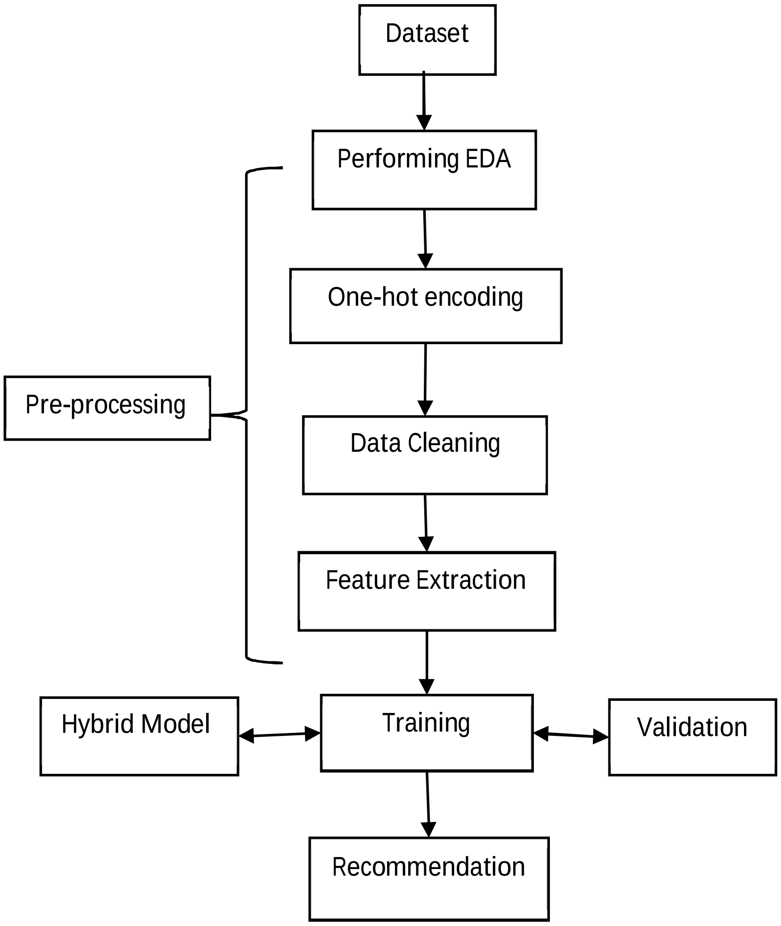


Figure 2- Flowchart of the proposed movie recommendation system.

As depicted in Figure 3, a CNN-based recommendation system recommends things for the goal user, consists primarily of the process below. The input layer first inputs data and preprocesses the qualities in the data. The embedded layer is then used to extract features from the pre-processed data and construct each characteristic feature vector. The full-connection layer then implements the operation after the embedding operation to connect attributes features and produce the user features and item features. The prediction rating is then calculated using the user and object features. Lastly, the Top-k products with the highest prediction rating but no user ratings are chosen for recommendation.

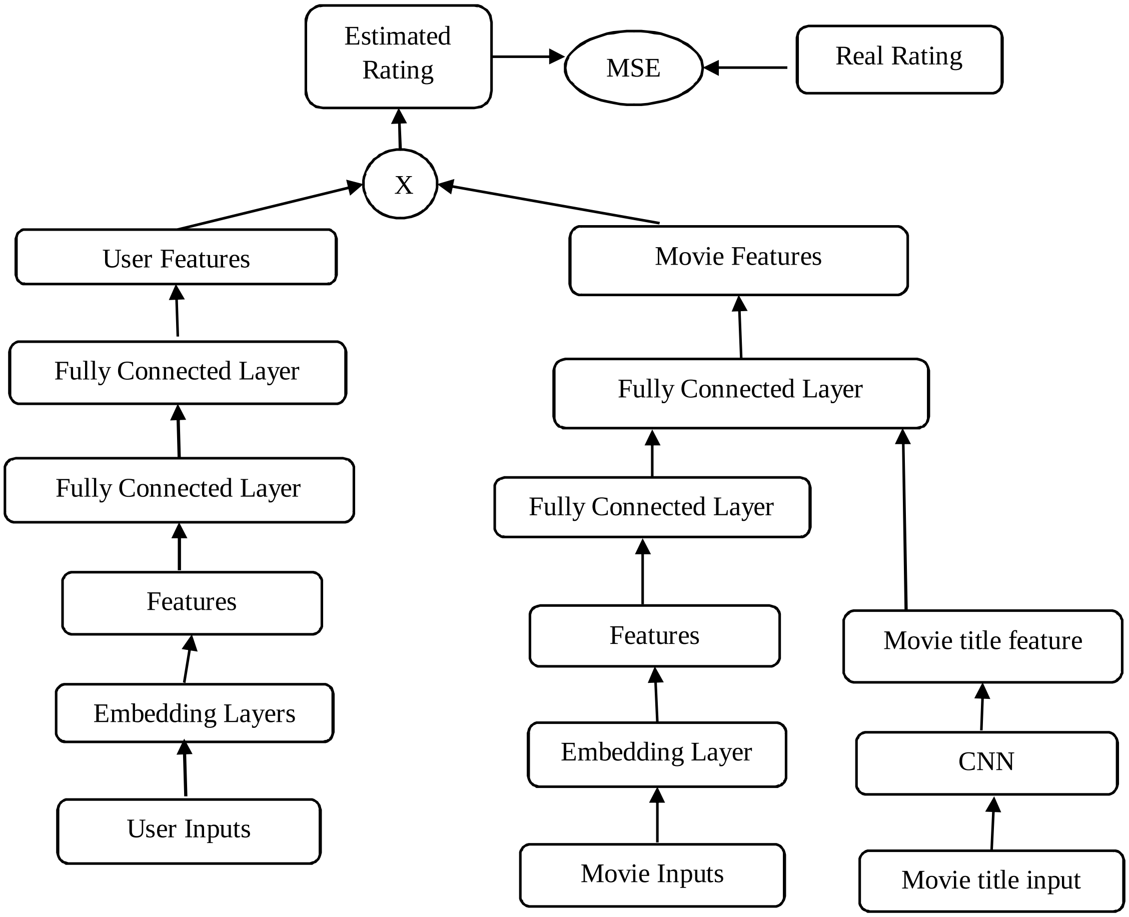


Figure 3- The proposed CNN model.

VI. EXPERIMENTAL ANALYSIS

The MovieLens-1m (https://grouplens.org/datasets/movielens/lm/) experiment dataset was taken for the developed framework, which contains 1000000 ratings from 6040 people on 3952 various films and was acquired by the GroupLens Project at the Minnesota University. Users.dat, movies.dat, and ratings.dat are the three files that contain the information. The exploratory dataset primarily contains the details, that include various properties, namely, UserID, Gender, Age, OccupationID, and Zip-code, where Gender is denoted by M and F, Age is separated into many age categories, and Zip-code is of no value. Figure 4(a) shows the user attributes, which include five attributes: UserID, Gender, age, occupationID, zip-code. Figure 4(b) depicts the movie statistics with three attributes: MovieID, Title, and Genre. Figure 4(c) shows the ratings attributes as userID, movieID, ratings, and timestamps. The users' evaluations range from 1 to 5, The higher the rating, the more enthusiastic the users are about the film. A movie, as indicated in Figure 4(c), might feature multiple genres at the same time, such as crime and adventure.

MovieId	Title	Genres

(a)

MovieId	Title	Genres

(b)

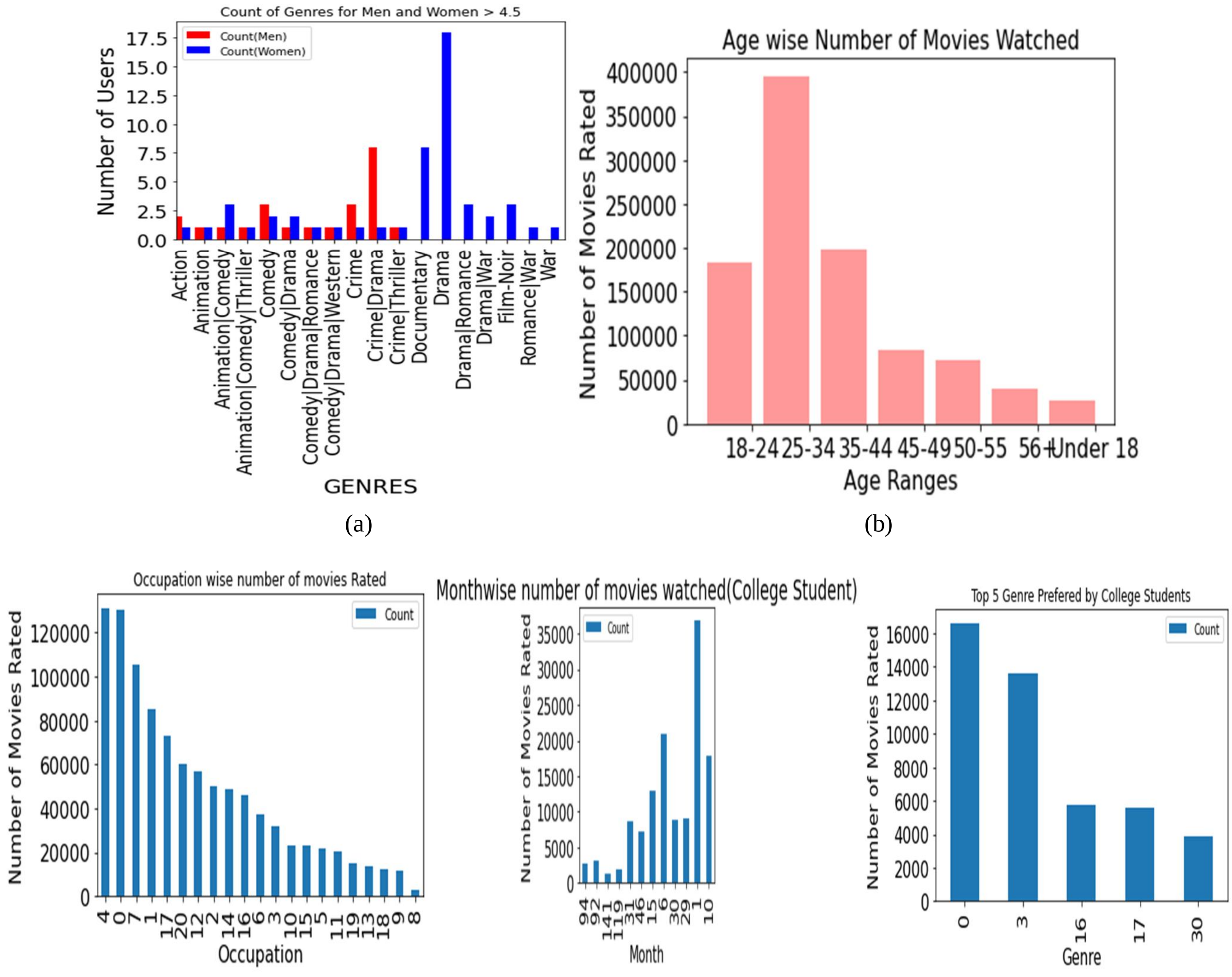
UserID	Gender	Age	OccupationID	Zip-code

(c)

Figure 4- Figure (a), (b), (c) shows the attributes of users, movies, and ratings.

1) *Data Pre-processing:* We initially pre-process the data to make the precise and accurate data and acceptable for operation, so that we can process it more rapidly and smoothly later.
2) *Performing EDA:* EDA (Exploratory Data Analysis) is a technique for highlighting the most important features of a dataset. It's used to understand data, put it in perspective, understand variables and their associations, and come up with ideas that could help with the construction of prediction models. It aids us in studying the entire dataset and summarising its most important features. After analyzing the dataset using EDA, we discovered that user data and movies may be utilized for training, while ratings can be used for prediction and as labels.

We're dealing with a user, a movie, and a user-item rating information table. We plot graphs and charts represented in figure 5 to understand every bit of the dataset like top ten movies rated by users, movies liked by users of different ages, occupation-wise movies watched by users, month-wise number of movies watched by students, etc are shown in the figure.

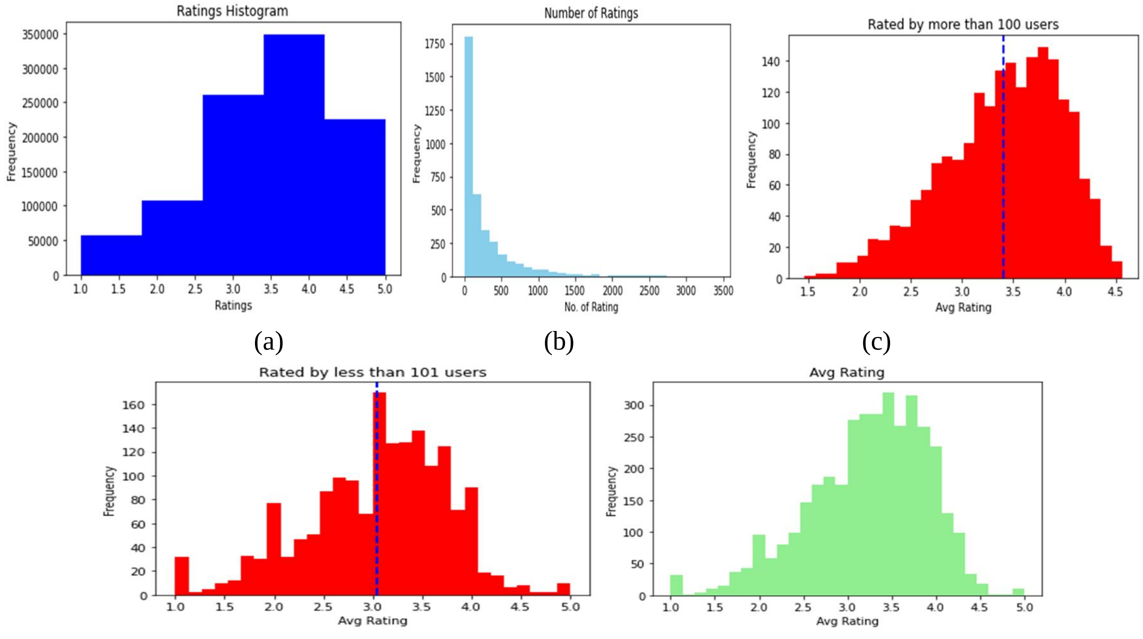


(a) (b)

(c) (d) (e)

Figure 5- Figure 5(a) shows the count of Genres for men and women for several users > 4.5. Figure 5(b) shows the movie ratings by different age groups. Figure 5(c) shows the movie ratings by users of different occupations. Figure 5(d) shows the movies ratings for month-wise movies watched by college students. Figure 5(e) shows the movie ratings for top genres preferred by college students.

After this, we plot the different histograms (shown in figure 6) ratings by users for checking the ratings, the number of ratings, rating graph for more than 100 users, rating graph for less than 101 user ratings, and average ratings.

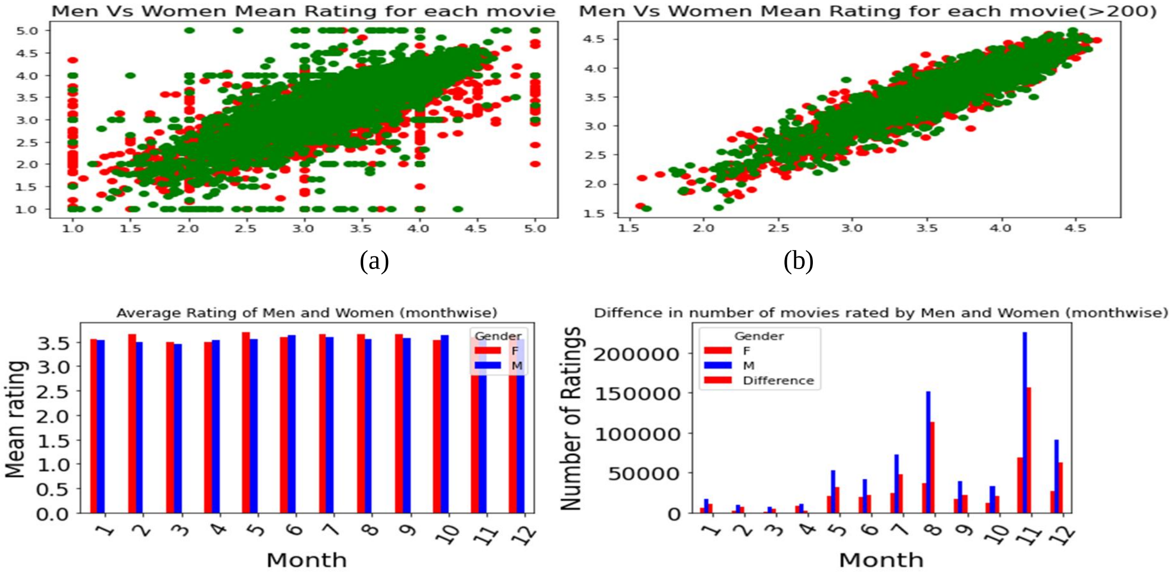


(a) (b) (c)

(d) (e)

Figure 6- Figure 6(a) shows the rating histogram. Figure 6(b) illustrates the number of rating histograms. Figure 6(c) shows the average ratings for more than 100 users. Figure 6(d) represents the average ratings for less than 101 users. Figure 6(e) shows the average ratings done by users.

After the histograms for checking different aspects of user ratings, we plot the charts and graphs represented in figure 7 by categorizing them Gender wise. The charts for men Vs women mean rating for each movie, men Vs women mean rating for each movie where the rating is greater than 200, the month-wise average rating of men and women, the month-wise difference in the number of movies rated by men and women are shown in figure 7.



(a) (b)

(c) (d)

Figure 7- Figure 7(a) shows the charts for men Vs women's mean rating for each movie. Figure 7(b) shows the men Vs women mean rating for each movie where the rating is greater than 200. Figure 7(c) shows the month-wise average rating of men and women. Figure 7(d) shows the month-wise difference in the number of movies rated by men and women.

- The data pre-processing algorithm consists mostly of three steps. For Userid, occupation, and movie, nothing needs to be changed.
- Gender: For the user information table's properties users.dat, In this case, 'F' must be changed to 0 and 'M' must be transformed to 1.
- Age: The data must be translated into seven successive digits ranging from 0 to 6.
- Genres: The data must be translated to a numerical value. Because some movies are a mixture of existing genres, the classes in genres are first transformed into strings and then into a number dictionary, and then the genres section of each movie is changed into a number list.
- Title- In the beginning, a text-to-numbers dictionary is created, which converts the title description into a series of numbers. The year should be omitted from the title.
- The length of genres and titles must be consistent for neural networks to comprehend them efficiently. The number corresponding to the blank part is filled in.

VII. MODEL EVALUATION

In the convolution layer of CNN, the features are extracted by moving the kernel function, multiplying pixels and the associated weights of the kernel, ultimately adding all the products to generate outcomes in the Convolution layers by using four Conv2d layers. The user's characteristics matrix is mostly created by incorporating user data into the network. UserId, MovieId, gender, age, and occupation information are all changed to number type during pre-processing, and this data is then used as the index of the embedding matrix. The embedding layer is utilized the movie title in the network's first layer to keep the data input dimension for user gender, user age, user job, movieid, and movie categories.

Subsampling and space pooling are all terms used to describe max pooling. It reduces the dimensionality of each characteristic map while retaining the most important data. Max-pooling can take many forms, including number, mean, and average. The most important component of the rectified function map of that window in the case of Max Pooling is a spatial neighborhood.

For classification tasks, the Fully Connected Layer is used for gender, age, job, movie id, and movie categories. For classification, it employs a softmax activation function. The term "Fully Connected" means that each neuron in the earlier layer is linked to each neuron in the subsequent layer. Higher-level features are represented by the output from preceding levels.

Passing this to inference and lambda dense layer. The dense layer is the layer of a basic neuron where each neuron takes inputs from all prior layer neurons. The output from the convolutional layers is utilized to classify the image using a dense layer.

Some key parameters for the algorithm are considered in this model during the training phase, and the concluding outcomes are shown in Table 1. For training, an ADAM optimizer is used for 20 epochs with k-fold.

Table 1- Table defines the parameters taken for model evaluation.

Parameters	Values
Epoch	20
Learning Rate	0.001
Optimizer	ADAM Optimizer
Activation Function	Relu
Output	Softmax Function

Evaluation Metrics- We evaluate the CNN model with MSE (Mean Square Error)and RMSE (Root Mean Square Error). The formulas in equations (3) & (4) are as follows-

MSE= $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$ (3)

RMSE= $\sqrt{MSE}$= $\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$ (4)

Where ŷ is the predicted value of y.

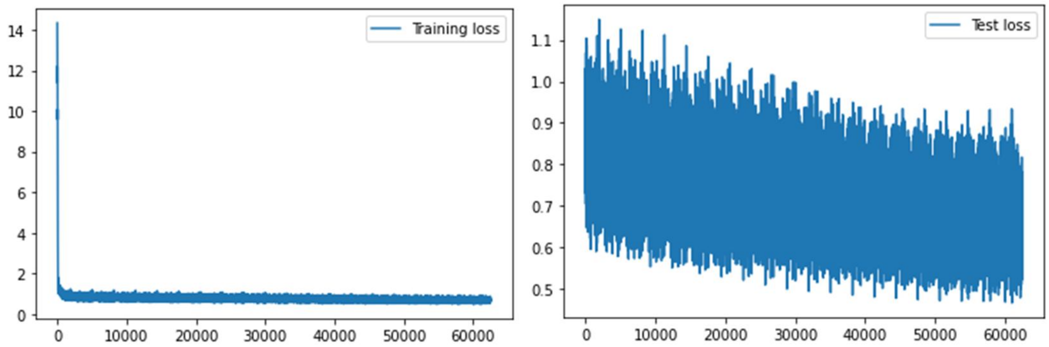


Figure 8- Charts showing the loss values of training and testing.

Table 2–Table representing the MSE and RMSE values of the proposed system.

Accuracy	MSE	RMSE
Base	0.76	0.87
Proposed	0.67	0.81

After training and testing, the training loss and test loss graphs are shown in figure 8. Table 2 shows the MSE and RMSE values for our proposed approach which determines that the suggested approach is better than the other traditional approaches and gives the results with less mean square error. The figure shows the training time of the model with 20 epochs and the test set of the model with an MSE of 0.67 and RMSE of 0.81.

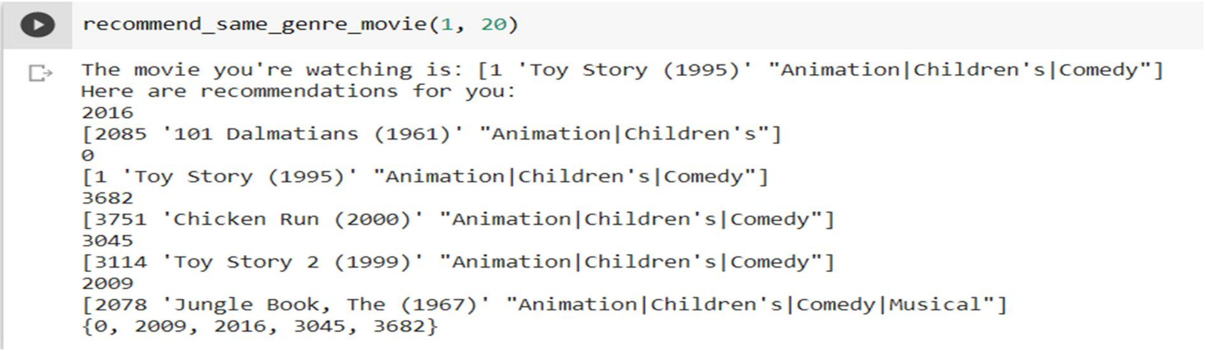
For Recommendations:

The cosine similarity is used in positive space. If the distance between two nonzero feature vectors is close to one, it signifies, they are highly comparable. When the distance between two nonzero feature vectors approaches 0, it indicates that they are not at all similar.

The cosine similarity of two vectors of qualities, X and Y, is stated using a dot product and magnitude (in equation (5)) as:

$$\text{Cos }(\theta)\frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \qquad (5)$$

- Calculate the cosine similarity between the specified movie and the feature matrix of all movies.
- Return the top k highest maximum similarity values.
- Choose at random to confirm that each recommendation is unique.

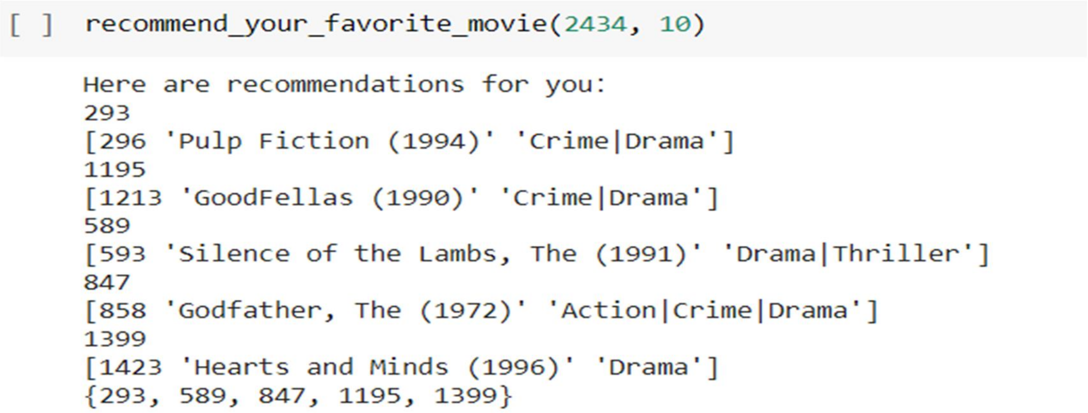


```
recommend_same_genre_movie(1, 20)
```

```
The movie you're watching is: [1 'Toy Story (1995)' "Animation|Children's|Comedy"]
Here are recommendations for you:
2016
[2085 '101 Dalmatians (1961)' "Animation|Children's"]
0
[1 'Toy Story (1995)' "Animation|Children's|Comedy"]
3682
[3751 'Chicken Run (2000)' "Animation|Children's|Comedy"]
3045
[3114 'Toy Story 2 (1999)' "Animation|Children's|Comedy"]
2009
[2078 'Jungle Book, The (1967)' "Animation|Children's|Comedy|Musical"]
{0, 2009, 2016, 3045, 3682}
```

Figure 9- Figure shows the movie recommender for the same Genre.

Figure 9 shows the movie recommender for the same Genre. For example, the movie which is watched is ‘Toy Story (1995)’ with Genres ‘Animation/Children's / Comedy’ and the first recommended movie is ‘101 Dalmatians (1961)’ with the same Genre.



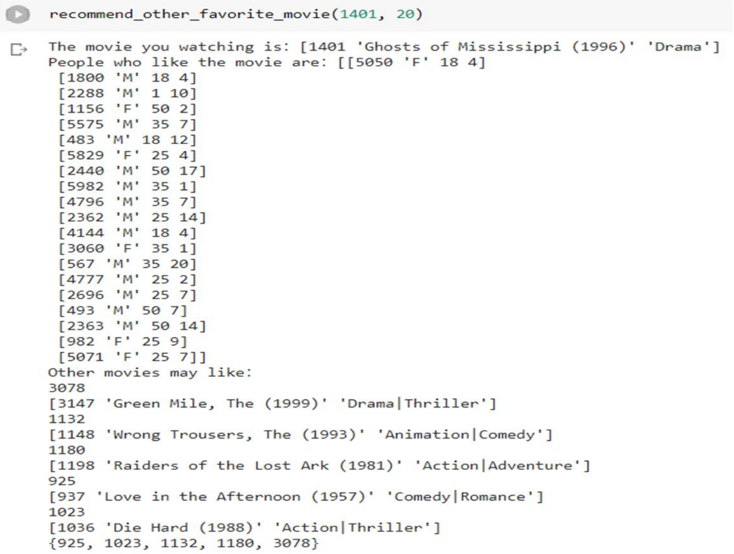
```
[ ]  recommend_your_favorite_movie(2434, 10)

     Here are recommendations for you:
     293
     [296 'Pulp Fiction (1994)' 'Crime|Drama']
     1195
     [1213 'GoodFellas (1990)' 'Crime|Drama']
     589
     [593 'Silence of the Lambs, The (1991)' 'Drama|Thriller']
     847
     [858 'Godfather, The (1972)' 'Action|Crime|Drama']
     1399
     [1423 'Hearts and Minds (1996)' 'Drama']
     {293, 589, 847, 1195, 1399}
```

Figure 10- Figure shows the movie recommender for the favorite movie.

Figure 10 shows the movie recommender for the favorite movie. For example, with the favorite genre Crime/ Drama, etc. the first recommended favorite movie is ‘Pulp Fiction (1994)’.

Recommend other favorite movies according to the given movie.
• To get their feature matrix, choose the top k users that like a supplied movie.
• Compute the average of all of these people's movie ratings.
• Recommend films with the best reviews.
• Choose movies with similar ratings at random.



```
recommend_other_favorite_movie(1401, 20)

The movie you watching is: [1401 'Ghosts of Mississippi (1996)' 'Drama']
People who like the movie are: [[5050 'F' 18 4]
 [1800 'M' 18 4]
 [2288 'M' 1 10]
 [1156 'F' 50 2]
 [5575 'M' 35 7]
 [483 'M' 18 12]
 [5829 'F' 25 4]
 [2440 'M' 50 17]
 [5982 'M' 35 1]
 [4796 'M' 35 7]
 [2362 'M' 25 14]
 [4144 'M' 18 4]
 [3060 'F' 35 1]
 [567 'M' 35 20]
 [4777 'M' 25 2]
 [2696 'M' 25 7]
 [493 'M' 50 7]
 [2363 'M' 50 14]
 [982 'F' 25 9]
 [5071 'F' 25 7]]
Other movies may like:
3078
[3147 'Green Mile, The (1999)' 'Drama|Thriller']
1132
[1148 'Wrong Trousers, The (1993)' 'Animation|Comedy']
1180
[1198 'Raiders of the Lost Ark (1981)' 'Action|Adventure']
925
[937 'Love in the Afternoon (1957)' 'Comedy|Romance']
1023
[1036 'Die Hard (1988)' 'Action|Thriller']
{925, 1023, 1132, 1180, 3078}
```

Figure 11- The figure shows the recommended favorite movie based on the movies watched by the user.

The figure shows the recommended favorite movie based on the movie you are watching and the people who like the movie.

VIII. CONCLUTION

The most crucial aspect of life is the recommendation system. Online purchasing has grown in popularity as a result of the suggestion system. The recommender system is capable of providing adequate services. As a result, recommender systems have proven to be a boon to society, assisting customers in making decisions among their options. The recommender system employs a variety of methods to provide recommendations. This paper suggested a functional model with individual features layers of users, movies, convolutional neural network, and fully connected layer. The MAE and RMSE outcomes show that the CNN model works well in recommending the movies and outperforms the other conventional models.

REFERENCES

[1] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Comput. Surv., vol. 52, no. 1, pp. 1–35, 2019, doi: 10.1145/3285029.

[2] M. Khalaji, C. Dadkhah, and J. Gharibshah, “Hybrid Movie Recommender System based on Resource Allocation,” vol. 17, no. 2, pp. 17–25, 2020.

[3] I. Portugal, P. Alencar, and D. Cowan, “The use of machine learning algorithms in recommender systems: A systematic review,” Expert Syst. Appl., vol. 97, pp. 205–227, 2018, doi: 10.1016/j.eswa.2017.12.020.

[4] M. Hayakawa, “MF Techniques,” Earthq. Predict. with Radio Tech., pp. 199–207, 2015, doi: 10.1002/9781118770368.ch6.

[5] M. F. Aljunid and M. Dh, “An Efficient Deep Learning Approach for Collaborative Filtering Recommender System,” Procedia Comput. Sci., vol. 171, no. 2019, pp. 829–836, 2020, doi: 10.1016/j.procs.2020.04.090.

[6] Y. H. Low, W. S. Yap, and Y. K. Tee, Convolutional Neural Network-Based Collaborative Filtering for Recommendation Systems, vol. 1015. Springer Singapore, 2019.

[7] U. Turkut, A. Tuncer, H. Savran, and S. Yilmaz, “An Online Recommendation System Using Deep Learning for Textile Products,” HORA 2020 - 2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc., pp. 6–9, 2020, doi: 10.1109/HORA49412.2020.9152875.

[8] D. Wadikar, N. Kumari, R. Bhat, and V. Shirodkar, “Book Recommendation Platform using Deep Learning Book Recommendation Platform using Deep Learning,” Int. Res. J. Eng. Technol., vol. 07, no. July, pp. 6764--6770, 2020.

[9] V. Aggarwal and G. Kaur, “A review:deep learning technique for image classification,” Accent. Trans. Image Process. Comput. Vis., vol. 4, no. 11, pp. 21–25, 2018, doi: 10.19101/tipcv.2018.411003.

[10] L. Zhang, D. Li, and Q. Guo, “Deep Learning from Spatio-Temporal Data Using Orthogonal Regularizaion Residual CNN for Air Prediction,” IEEE Access, vol. 8, pp. 66037–66047, 2020, doi: 10.1109/ACCESS.2020.2985657.

[11] H. Wang, N. Lou, and Z. Chao, “A Personalized Movie Recommendation System based on LSTM-CNN,” Proc. - 2020 2nd Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2020, pp. 485–490, 2020, doi: 10.1109/MLBDBI51377.2020.00102.