MOVIE RECOMMENDATION SYSTEM USING COLLABORATIVEFILTERING

Dr SVG Reddy[1], Putchakayala Meher Sowjanya[2], Annem Pavan Kumar Reddy[3], Bavisetti Sai Saketh[4],Lekkala Yaswanth Kumar[5], Karri Viswa Abhiram Reddy[6]

1Associate Professor, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, AndhraPradesh, 530045, India.
2,3,4,5,6B. Tech Student, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, AndhraPradesh, 530045, India.

---***---

Abstract -

Nowadays, the recommendation system is crucial and is used by many significant applications. The proliferation of applications, the emergence of a global village, and the availability of a large amount of information are the results of the recommendation system. An overview of the approaches and techniques developed by this study introduces a recommender system based on collaborative filtering, which has evolved over time to include content-based and hybrid approaches. The study examines various techniques used for collaborative filtering, including matrix factorization, user- based and item-based recommendation. It also serves as a guide for future research in this field. By analyzing user preferences and eating habits, we extract aspect-based ratings from reviews and recommend reviews accordingly. Furthermore, the proposed movie recommendation system is tested against multiple evaluation criteria and performs better than existing approaches. Through extensive research and review of numerous papers, it was discovered that content-based filtering typically relies on a single technique for converting text into vectors for recommendations and a single approach to detect similarity between vectors. Think of it as a hybrid strategy that only uses a content-based filtering method. The way we search for exciting items has revolutionized today's recommendation systems. It is vital to advise mobile users to download OTT movie apps. To help you choose suitable movies, it thoroughly summarizes user preferences, reviews, and[SS1] feelings. It requires absolute accuracy and timeliness, yet this information filtering technique predicts user preferences. A recommender system is a system that aims to predict or filter selections according to user choices. OTT platforms, search engines, articles, music, videos, and other similar platforms are typical applications of recommender systems. In this work, we tend to provide a movie design system based on a common approach. It's a supported collaborative filtering strategy that takes user-provided knowledge, analyses it, and then suggests the most appropriate movie for the users.

INTRODUCTION

Because of the richness of knowledge amassed up to the twenty- first century and the increasing rate at which information is gushing over the internet, there is a great deal ofconfusion over what to consume and what not to ingest. Even on YouTube, there are always a ton of videos available if you want to see one about a particular concept. Since the results are suitably ranked, there might not be much of a problem right now, but what if they weren't? In this situation, we would undoubtedly spend a lot of time looking for the best movie thatfits us and meets our needs. When you look for something on a website, this is what happens the algorithm could be able to offer you recommendations the next time you visit a particular website without you even having to search. This feature is fascinating, isn't it? A recommender system's main responsibility is to present the user with the items that are the most pertinent. Recommendation engines are used by Amazon, Flipkart, Netflix, and YouTube to propose videos, items, movies, and other content. Regardless of what you do on these websites, a system is in place that tracks your activity and makes suggestions for activities or products that you are very likely to be interested in. This research paper addresses the logic behind movie suggestions, conventional movie recommendation systems, problems with conventional movie recommendation systems, and a suggested fix for an AI-based personalized movie recommendation system. There are already a lot of well-known movie recommendation datasets available on Kaggle and other sites. Movie lens, the TMDB Movie Dataset, and the Netflix dataset are a few of the well-known datasets. Websites like Netflix, Amazon Prime, and others employ movie recommendations to increase revenue or profits by eventually enhancing the user experience. In reality, in 2009 Netflix conducted a competition with a prize pool of about $1 million ($1M) for creating at least 10% upgrades to the current system. As was previously mentioned, we have access to a large amount of data, and since we are not interested in everything that is available to us, we must filter itin order to use it.

To filter the data, filtering tactics are needed. Different filtering methods or movie recommendation algorithms can be used to create a recommendation system. The main algorithms for filtering or suggesting movies are as follow

1.1 Content Based Filtering:

A content-based recommender system tries to infer user characteristics or behavior from the characteristics of an object to which the user responds favourably.

Movies	User 1	User 2	User 3	User 4	Action	Comedy
Item 1	1		4	5	Yes	No
Item 2	5	4	1	2	No	Yes
Item 3	4	4		3	Yes	Yes
Item 4	2	2	4	4	No	Yes

Fig.1 Content Based Filtering

The two columns after that both humor and action Describe the genres of the film. Now that we are aware of which individuals favor certain genres, we may find characteristics that are unique to that person based on how those individual responds to films in that genre.

Using the feature vector created to embed the user in an embedding space after we are informed of the user's preferences provide recommendations to the user based on those preferences. The feature vectors for the item and the user- selected feature vectors from previous records are used to generate the similarity metrics, which will be covered in more detail later, during recommendation. The best few are then suggested. Content- based filtering does not require other users' data during recommendations to one user.

1.2 Collaborative Filtering:

Collaboration does not need the provision of item characteristics. Each user and item are described by a feature vector or embedding.

Both users and things can be embedded by it. The same embedding space contains both users and items.

The responses of other users are taken into consideration whenproposing a certain individual. In order to suggest items to a particular user, it maintains track of their favourite items as wellas those of users who behave and have tastes similar to theirs. It uses customer feedback on a range of products to generatesuggestions.

As the name suggests, this filtering strategy is based on a combination of pertinent user and other user activities. The system compares and evaluates these acts in order to produce the optimal results. It is the result of the preferences and acts ofnumerous moviegoers.

This movie recommendation system and the ML algorithm itis based on are built on the history of each user in the database. The interaction of all system users with articles is absolutely necessary for collaborative filtering to work. Since content-based filtering only uses the user's input for modelling, each user of this ML-based recommender system has an impact on the final product. In collaborative filtering, we share a lot of similarities, some of them are as follows:

1.2.1Jaccard similarity:

A common proximity measurement called Jaccard Similarity is used to compare two objects, such two texts, to see how similar they are. The Jaccard similarity can be used to find similarities between two collections or two asymmetric binary vectors.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

J = Jaccard distance
A = set 1
B = set 2

1.2.2 Cosine similarity:

In an inner product space, two vectors are compared using cosine similarity. The cosine of the angle between two vectors can be used to calculate whether or not they are pointing in the same general direction.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

1.2.3 Pearson similarity:

The Pearson correlation coefficient assesses the strength of a linear relationship between two variables. Its value can be between -1 and 1, where -1 denotes a total linear correlation that is negative, 0 denotes a lack of relationship, and +1 denotes a total linear correlation that is positive..

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

Knowledge-based filtering:

If a recommender system provides recommendations based on the user's specific requests rather than the user's rating history, it is a knowledge base. The user can be required to provide a set of standards or specifications as well as an illustration of what the final product should look like. Following a search of the item database, the algorithm returnsresults that are comparable.

1.4 Deep Neural Networks Model:

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Like flat ANNs, DNNs can model complex nonlinear relationships.

Neural networks' main objective is to receive a variety of inputs, do more complex calculations, and produce outputs for practical problems like categorization. It solely supports transfers between neural networks.

Neural networks are frequently employed in supervised learning and reinforcement learning tasks. Layers are joined to one another to form these networks.

Deep learning has a potentially enormous number of hidden layers, most of which are nonlinear. Consider that there are 1000layers.

Performance-wise, DL models outperform conventional ML networks.

Loss function minimization and network optimization are the two main applications of gradient descent.

A crucial component of deep learning models are training datasets. In addition, backpropagation is the most common approach used to train DL models.

Large neural networks with intricate input/output transformations are trained using deep learning (DL).

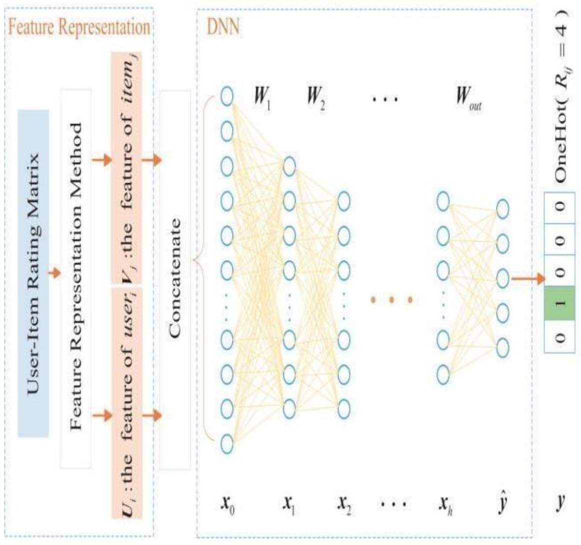

Fig.2 Deep Neural Networks

1.4.1 Sources of user-item interactions:

Implicit Feedback: Based on the user's clicks, searches, and purchases, the preferences and dislikes of the user are monitoredand recorded. There are plenty of them, but no one objects.

Explicit Feedback: When a person reacts to or rates a product,they are expressing their preferences and dislike Although thereare fewer of them, it has both positive and negative inputs.

1.5 Hybrid Filtering:

A hybrid recommendation system blends collaborative filtering methods with content-based filtering approaches. Mixing collaborative and content-based filtering could help us get around the problems we run into when employing them independently, and in some situations, it might even be more effective. There are many ways to put into practise hybrid recommender system techniques, such as employing content and collaborative-based methods to produce predictions separately, then combining the predictions, or simply improving a content- based approach with collaborative-based method capabilities (and vice versa).Numerous research evaluates the effectiveness of hybrid approaches to conventional procedures and come to the conclusion that using hybrid methods yields more accurate suggestions.

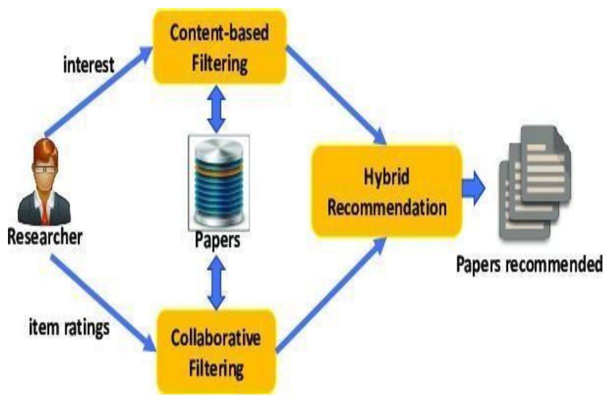

Fig.3 Hybrid Recommender System

1.5.1 Types of Data for Generating a RecommendationSystem:

The approaches allow us to categories the data into two categories from which we can create a recommendation system:

-

• **Explicit Feedback:** data that explicitly includes user feedback. Explicit feedback is a sort of user evaluation that expresses the user's feelings towards the product, such as whether he enjoyedit or not.

Implicit Feedback: This data may contain information like c licks, movies watched, music played, and other activities ratherthan the user's rating or score.

We'll discuss implicit feedback's significance because, in this piece, we'll develop a recommendation system based on it. We attempted to define both explicitly in the previous section. andimplicit feedback. So let's talk about a recommender that is based on explicit input and makes suggestions based on ratings, like making book recommendations based on user reviews. The recommender is now focusing on the rating; however, it does not take into account the book a user ultimately decides to read. If ratings are not available, this could result in the recommendation being in a state of incomplete information.

Information on which books have received the most or least votes, or which books have been selected by the majority of people, can be a useful source of information for a recommendation system to make the most of itself..

Because the user chooses which objects to rate and the left receives a blank rating, it is extremely clear that the missing ratings are more likely to be bad. Alternatively, we can say things that we don't think others will agree with. Without ranking them, we left them.

These findings inspired us to develop a model that can function with implicit feedback. Additionally, Light FM, a library, can assist us in creating a recommendation system onimplicit feedback.

. Losses used by Recommendation Systems:

There are two ways we can create recommendation systems, each using a different loss strategy:

When the user interacts favorably with the data and the ROC AUC needs to be optimized, we can utilize the Bayesian Personalized Ranking (BPR) pairwise loss approach. Using the pairwise loss, we seek to maximise the difference between predictions for positive and randomly chosen negative feedback.

Weighted Approximate-Rank Pairwise (WARP) loss: this is crucial when there is positive feedback interaction and we need to optimize particular top recommendations. This strategy maximizes the rank of positive input by continually sampling the negative feedback until it finds the one that is defying the rank.

2.LITERATURE REVIEW

Based on "Content-based recommendation systems," M. J. Pazzani and D. Billsus, published in The Adaptive Web, Springer, Berlin, Germany, pages (325–341). In order to provide items to consumers, content-based recommendation systems take advantage of commonalities between items. We have reviewed and rated movies, and this recommender system makes suggestions based on their description or features.

The topic of "Probabilistic matrix factorization," by A. Mnih and R. R. Salakhutdinov, in Proc.Adv. Neural Inf. Process. Syst., 2008, pp. (1257–1264,) is merging latent features of the individual and properties of the object to fit the appropriate rating score. when the square error of the loss function is minimized.

For the improvement and performance of hybrid and content based X. Wang and Y. WangThe paper titled "Enhancing Content-Based and Hybrid Music Recommendation using Deep Learning" was presented at the 22nd ACM International Conference on Multimedia in 2014, with a focus on utilizing deep neural network models to improve the accuracy of music recommendation systems. The aim of this approach is to increase the precision of predictions.

E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, “Deep neural network language models,” in Proc. NAACL-HLT Workshop, Will Ever Really Replace N-Gram Model? 2012, pp. (20–28). is based upon Using the forward propagation approach in a deep neural network model, which uses the latent properties of users and things as inputs to predict the score. Here we used the DNN model and ReLU is activation function and entropy for evaluating differences to get an equation.

In thED.Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in

Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2012, pp. (3642–3649) based upon Our deep artificial neural network designs, which are biologically plausible, can. Convolutional winner-take-all neurons with small receptive fields produce dense networks with nearly the same number of sparsely linked neural layers.The user profile is used to generate a recommendation. Arecommendation is made by content-based filtering based solely on one user's X profile. Based on the user &X's prior activity, the system tries to suggest articles that are comparable to this article.

3.PROBLEM IDENTIFICATION & OBJECTIVES

3.1 PROBLEM STATEMENT

The main issues with the method for suggesting movies include false user reviews, an exaggerated rating, and irate consumers. As an output, take into account not just a rating choice but also an "explanation" of why the user would enjoy the movie.

A recommendation system's objective is to offer users the best possible solutions.

Choose the movies that consumers wish to see based on theirpost-viewing ratings.

Alternatives include:

Create a list of movies that at least one user should startwatching as their next choice.

3.2 OBJECTIVES

To increase the impact of movie review endorsement. To make it easier for people to managetheir time.

To provide consumers with a pleasant user experience that enables them to use flexible.

4.SYSTEM DESIGN

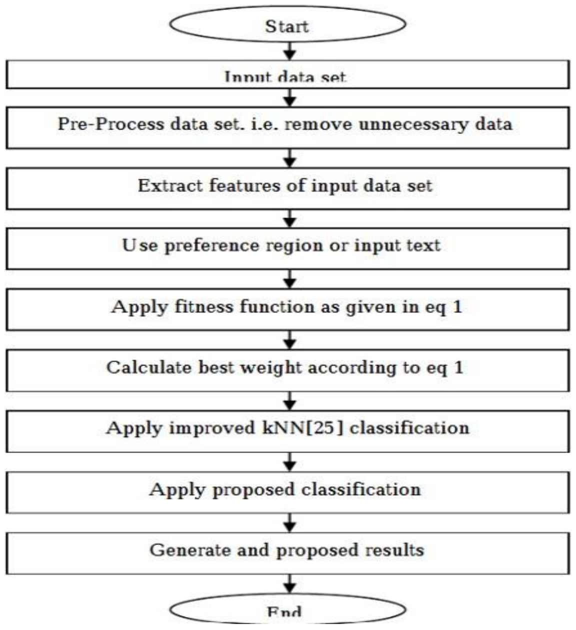
Start
Input data set
Pre-Process data set. i.e. remove unnecessary data
Extract features of input data set
Use preference region or input text
Apply fitness function as given in eq 1
Calculate best weight according to eq 1
Apply improved kNN[25] classification
Apply proposed classification
Generate and proposed results
End

Fig.4 Flowchart of the collaborative Filtering Approach

5.IMPLEMENTATION

In the first step we import the python libraries pandas and Numpy to perform cosine similarity vectorization considerably more quickly and effectively,. To reach the highest achievable time complexity, we shall make the best use of the libraries as they are made available.

Following that, we will download two Kaggle data sets containing the information required to make suggestions. We combine two data sets into a single Data Set to improve performance.

We will next read the Data Sets and Apply Data Preprocessing operations on them. Data preprocessing functions include data cleaning, removing unwanted data rows, unidentified data columns, and modifying data as Now we change the datatypes for the columns and define a function to analyse the data, which allows us to understand what the data indicates and how to use it. The null values are then removed from the database.

Finally, we create the tag attribute by combining all of the columns such as overview, genres, keywords, cast, crew, and so on.

The primary code is simply made up of the Pickle Files that wereceived after running the Google Collab extensively to obtainthose pickle files and streamlit.

As we know, these Pickle files include data that has been processed and vectorized in order for us to make decisions inthe final section. At the same time, we added a Deep Neural Network Code by modifying the main code.

Here is where we create a website that loads other URLs, suchas one that retrieves movie posters from an external link.

When a user inputs the proper information to receive recommendations, the website will function according to the Main code. The user is aware that they won't be able to access the pickle files; instead, all they have access to is this website, which is essentially made possible by the main code.

This code serves as an intermediary between the user and the pickle files, data, and, ultimately, the recommendation system.

6.TECHNOLOGIES

The technologies used for the implementation are: Python libraries:

Pandas: In order to work best with relational or labelled datasets, Pandas is a well-known open-source Python library for data analysis and manipulation. It provides a variety of data structures and procedures for working with matrices and numerical data. The following are some of the main characteristics of pandas.:

A two-dimensional table with rows and columns is known as a data frame and is the main data structure used by Pandas. Many sorts of data can be stored and handled using dataframes..

Pandas include powerful data manipulation tools such as filtering, sorting, grouping, merging, and reshaping.

Pandas can read and write data from and to a wide range of file formats, such asExcel, CSV, SQL databases, and others.

NumPy: The name NumPy refers to a free and open-source Python package for scientific computing. Multidimensional array objects and processing routines are also included in this package. Arrays can be used by NumPy to execute logical and mathematical calculations. The following list of NumPy's important features:

A potent N-dimensional array object that can be used to represent vectors, matrices, and higher-dimensional arrays is included in the NumPy programming language. This makes applying mathematical operations to data arrays simple.

For performing common mathematical operations like addition, multiplication, and trigonometric functions, NumPy provides optimised numerical functions. These operations are considerably quicker than comparable Python operations and are designed to work with NumPy's N-dimensional arrays .

NumPy includes a powerful linear algebra module that includes functions for performing matrix operations such as multiplication, inversion, and decomposition.

Pickle: Pickle is commonly used to serialize and deserialize a Python object structure. A Python object must be transformed into a byte stream before it can be saved to a file or database, maintained across sessions, or sent over a network.

Streamlit:A Python framework called Streamlit makes it simple to create interactive web applications for data science and machine learning. You don't need to write HTML, CSS, or JavaScript code to design bespoke user interfaces, dashboards, and visualizations with Streamlit.

The following are the Key characteristics of Streamlit include: Automatic responsiveness: Whenever you make changes to yourapp's code or data, Streamlit instantly re-renders it.

Widgets that are simple to use: You can easily add widgets to your app to make it responsive and interactive.

Easy integration with popular Python libraries like Pandas, Matplotlib, and Scikit-learn: Streamlit integrates with these libraries seamlessly.

HTML: Web pages are typically created using the markup language HTML (Hypertext Markup Language). Using numerous elements and tags, it is a text-based language that is used to define the organisation and content of a web page. Web browsers parse HTML code, converting it into a visual representation of the online page.

There are many different components in HTML, and each onehas a specific function.

Headings, paragraphs, lists, links, pictures, and tables are a few typical HTML elements. These components are used to organise the information on a web page and offer supplementary details, such as formatting and semantic meaning.

Usually, a text editor or an integrated development environment are used to build HTML documents (IDE). as soon as the HTML **CSS**: The visual styling of HTML and XML documents is described using the language CSS (Cascading Style Sheets). It enables web designers to isolate a document's display from its information, making it simpler to produce appealing and consistent designs across numerous pages and websites.

CSS specifies font size, colour, placement, and layout by applying rules to elements inside a document. The style> tag can be used to incorporate these rules directly into an HTML document or in a separate CSS file.

Javascript: High-level programming languages like JavaScript are frequently used to develop interactive online apps, games,and dynamic web content. It is a client-side scripting language, which means that rather than being performed on a server, it does it on the user's web browser.

Because it can be used for a variety of tasks, such as producing animations, modifying HTML and CSS, verifying form input, and managing user interactions, JavaScript is renowned for its flexibility. It serves as the foundation for a number of well- known web frameworks and libraries, including React, Angular, and Vue. **Google colab:**A cloud-based development environment called Google Colab (short for "Collaboratory") enables you to create and run Jupyter Notebook documents, which let you write and run code in a web browser. Google's Colab service, which is available for free, gives customers access to powerful computing tools including CPUs, GPUs, and TPUs aswell as well-known machine learning frameworks and libraries like TensorFlow, Keras, PyTorch, and OpenCV.

Working on Python projects, data analysis, machine learning, deep learning, and other tasks is possible with Colab. Also, you can quickly share your notebooks and work in real-time collaboration with others.

One of the key benefits of using Colab is that everything is managed for you, so there's no need to set up and maintain your own development environment.

RESULT

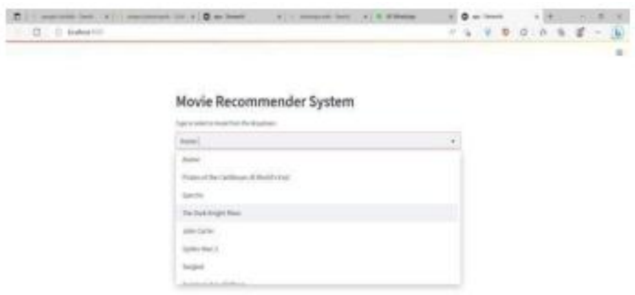

Fig.5 Searching Top 5 Movie Names By Entering 1 Movie

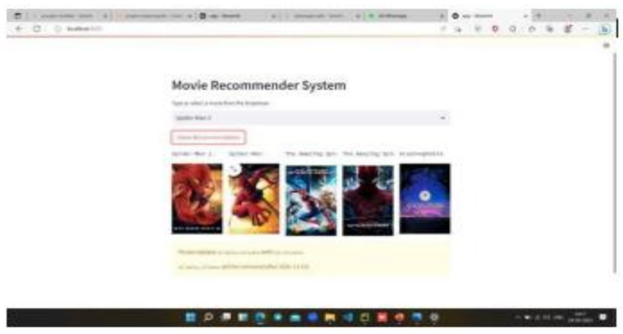

Fig.6 Final Output

7.CONCLUSION

A number of datasets can be used to create a movie recommendation system. However, we will use a dataset containing the movie's metadata for this endeavor (cast, crew, budget, etc...). This project involved creating and implementing an algorithm for a collaborative filtering recommendation system for a movie recommendation system. In this customized recommendation system, the top n movies are suggested to the active User using the singular value decomposition approach and the User-based cosine similarity algorithm.

The findings demonstrate that the combined recommendations outperform the separate deep neural network algorithm recommendations.

Future work will focus on investigating the performance of recommendation systems using alternative deep learning techniques, such as the convolutional neural network approach.

9. REFERENCES

1. D. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of- the-art and possible extensions,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, Jun. 2005.

2. Z. Huang, D. Zeng, and H. Chen, “A comparison of collaborative- filtering recommendation algorithms for e- commerce,” IEEE Intell. Syst., vol. 22, no. 5, pp. 68–78, Sep./Oct. 2007.

3. X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Adv. Artif. Intell., vol. 2009, Aug. 2009, Art. no. 421425.

4. D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2012, pp. 3642– 3649.

5. F. Richardson, D. Reynolds, and N. Dehak, “Deep neural network approaches to speaker and language recognition,” IEEE Signal Process. Lett., vol. 22, no. 10, pp. 1671–1675, Oct. 2015.

6. E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, “Deep neural network language models,” in Proc. NAACL-HLT Workshop, Will Ever Really Replace N-Gram Model? 2012, pp.20–28.

7. J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in Proc. 14th Conf. Uncertainty Artif. Intell., 1998, pp. 43–52.

8. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 1999, pp. 230–237.

9. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 285–295.

10. T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in Proc. IJCAI, vol. 99, no. 1999, pp. 1– 6 1999.

11. K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in Proc. Topics Artif. Intell. (PRICAI), 2000, pp. 679–689.

12. S. Vucetic and Z. Obradovic, "Collaborative filtering using a regression based approach," Knowl. Inf. Syst., vol. 7, no. 1, pp.1– 22, 2005.

13. J. Liu, Y. Jiang, Z. Li, X. Zhang, and H. Lu, "Domain-sensitive recommendation with user-item subgroup analysis," IEEE Trans. Knowl. Data Eng., vol. 28, no. 4, pp. 939–950, Apr. 2016.

14. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30–37, 2009.

15. K. Yu, S. Zhu, J. Lafferty, and Y. Gong, "Fast nonparametric matrix factorization for large-scale collaborative filtering," in Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2009, pp. 211–218.

16. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—A case study," Dept. Comput. Sci. Eng. Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. TR-00-043, 2000.

17. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in Proc. Adv. Neural Inf. Process. Syst., 2008, pp.1257–1264.

18. R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann pp. 791–798.Georgiev and P. Nakov, "A non-IID framework for laborative filtering with restricted boltzmann machines," in oc. Int. Conf. Mach. Learn., 2013, pp. 1148–1156.

BIOGRAPHIES

Dr. S.V.G. REDDY completed M- Tech (CST) from CST) from Andhra University and has obtained a PhD in Computer Science and Engineering from JNTU Kakinada. Heis an Associate Professor, the Department of CSE, GIT, GITAM University. His area of researchwork is data mining, machine learning and deep neural networks. He has guided various B. Tech and M. Tech projects and has publications in several journals. His areas of interest are drug discovery, computer vision, brain-computer interface, climate change and waste management.

Putchakayala Meher Sowjanya currently pursuing B.Tech-(CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. Her areas of research work are machine learning and deep learning. Her areas of interest are Recommended Systems and data science.

Pavan Kumar Reddy currently pursuing B.Tech-(CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. His areas of research work are machine learning and data Science. His areas of interest are Deep Learning and Recommended Systems.

Bavisetti Sai Saketh currently pursuing B.Tech-(CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. His areas of research work are machine learning and data science. His areas of interest are deep learning and data science.

Lekkala Yaswanth Kumar currently pursuing B.Tech-(CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. His areas of research work are machine learning and artifitial intelligence. His areas of interest are Recommended Systems and neural networks.

Karri Viswa Abhiram Reddy currently pursuing B.Tech-(CSE) from GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam. His areas of research work are web development and machine learning. His areas of interest are artifitial intelligence and Recommended Systems