A PROJECT REPORT

on

"MOVIE RECOMMENDATION"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE

BY

N Pranav Reddy	2105637
B Bhanu Prasanth	2105881
Manish Kumar Singh	2105890
N Kaghuvahan Reddy	21052774
Vemana Sumanth	21052803

UNDER THE GUIDANCE OF
DR. PRADEEP KANDULA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

May 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

“MOVIE RECOMMENDATION“

submitted by

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 12 /04/2024

DR. PRADEEP KANDULA
Project Guide

Acknowledgements

We are profoundly grateful to **DR. PRADEEP KANDULA** of **KIIT University** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion

N Pranav Reddy

B Bhanu

Prasanth

Manish Kumar

Singh

N Kaghuvahan

Reddy

Vemana

Sumanth

ABSTRACT

A typical recommendation system engine uses different algorithms to extract different types of data, examine the type of previous behavior of the user, and output the best matches of related items as needed. and recommends the most suitable items based on that. Something that is currently in high demand. Another main reason for this could be to maximize profits.

There are basically two ways he can recommend items to a user: Demographic filter that provides one or more general recommendations for the most popular movies based on their genre. Recommend movies to users if the demographics match. In general, popular movies are likely to be liked by more users. Content-based filtering is also used, which will take into account people's interests and suggest movies according to his/her preferences, and his third technique used here is to search for the same people with the same characteristic interests, or the same pattern. It is a collaborative filtering method where movies are grouped together.

Keywords: KNN , Cosine-Similarity, Collaborative Filtering, Content-Based Filtering, Efficiency, Scalability, Accuracy, User Demographics, pattern Regcognition.

Contents

Chapter 1

Introduction

Recommendation systems are essentially filtering systems that predict a user's decisions and suggest more accurate results based on the user's previous preferences. We have been using various applications of these recommendation systems for many years, and they are now used on various different online programs. Essentially, all of these sites' core content consists of several movie genres, including: B. A romantic suspense thriller or an online store: Social media networks that link to business websites such as LinkedIn. On Instagram, for instance, you can view past stories in the feeds of users you follow.

As you can see, Instagram tracks your conversations and previous actions with various users and only draws connections between stories about other accounts that have engaged in similar behaviors in the past or present. Based on the activities that a significant number of users have attempted through you, recommendation systems frequently enhance those users' activities. For instance, you are recommended to purchase a USB Type-C or Type-A adapter for your laptop and to use a tempered glass case for your phone if you purchase a laptop or mobile phone from Flipkart.

With steady improvements in recommendation systems, users will always receive better recommendations and will continue to improve throughout the 21st century to provide near-accurate solutions. If a dispute arises between electronic app music, music platforms, and educational institutions, you can simply refuse to use the app. Additionally, companies should focus on recommendation systems that are more complex than they appear. Whether you want to listen to music while gaming, traveling, running, or after a relationship argument, Different users have different preferences and choices depending on the type of activity and mood

Chapter 2

Basic Concepts/ Literature Review

2.1.1 Machine Learning Algorithms

K-Nearest Neighbors (KNN):

KNN is a versatile algorithm for classification and regression tasks. It predicts a data point's class or value based on the majority class or average value of its nearest neighbors. Key points:

Classification: Assigns the most common class label among the K nearest neighbors.

Regression: Predicts the average value of the target variable among the K nearest neighbors.

Distance Metrics: Relies on metrics like Euclidean, Manhattan, or Minkowski distance to measure similarity.

Parameter Selection: Choosing an appropriate K value is crucial and often involves cross-validation.

Cosine Similarity:

Cosine similarity measures similarity between two vectors in an n-dimensional space. It calculates the cosine of the angle between the vectors. Key aspects:

Vector Representation: Used in NLP and information retrieval, where documents are represented as vectors.

Normalized Dot Product: Measures the cosine of the angle between two vectors, ranging from -1 to 1.

Applications: Widely used in recommendation systems, document clustering, and text classification.

Understanding KNN and cosine similarity provides a solid foundation for their application in machine learning and data analysis.

Chapter 3

Problem Statement:

Developing an efficient movie recommendation system poses a challenge in catering to individual preferences while navigating vast datasets. The objective is to create a system that accurately predicts user tastes based on past interactions, demographic information, and genre preferences. This entails designing an algorithm that can sift through extensive data, curate a personalized list of 2000 movies, and recommend selections using various methodologies. Achieving scalability, efficiency, and accuracy amidst diverse user profiles and preferences is crucial. The goal is to deliver a tailored entertainment experience that enhances user satisfaction and engagement.

3.1 Project Planning

3.1.1 Existing movie recommendation systems may lack the specialized knowledge required for providing medical advice.

3.1.2 There is a growing demand for a dedicated Movie Recommendation System that offers personalized and accurate movie suggestions.

3.1.3 The challenge involves refining existing recommendation algorithms with specialized movie datasets and adapting safety protocols to maintain user privacy and data security.

3.1.4 Integrating features such as user feedback mechanisms can enhance the accuracy and effectiveness of the recommendation system.

3.2 Project Analysis

3.2.1 Evaluate the comprehensiveness of the movie ratings,Imdb top 250 dataset.

3.2.2 Analyze the base model's capabilities and limitations in understanding the Recommendation and context.

3.2.3 Ensure the project aligns with Movie U/A regulations and ethical standards

3.3 System Design:

3.3.1 Software Requirements:

- Leveraged Recommendation requirements such as jupyter notebook for sophisticated movie understanding and processing.

3.3.2 Hardware Requirements:

- Ensured sufficient hardware resources to support seamless processing and quick response times.

3.3.3 Machine Learning Libraries:

- Integrated machine learning libraries like pandas , numpy , sk_learn , natural language tool kit, KNN and Cosine Similarity for model training.

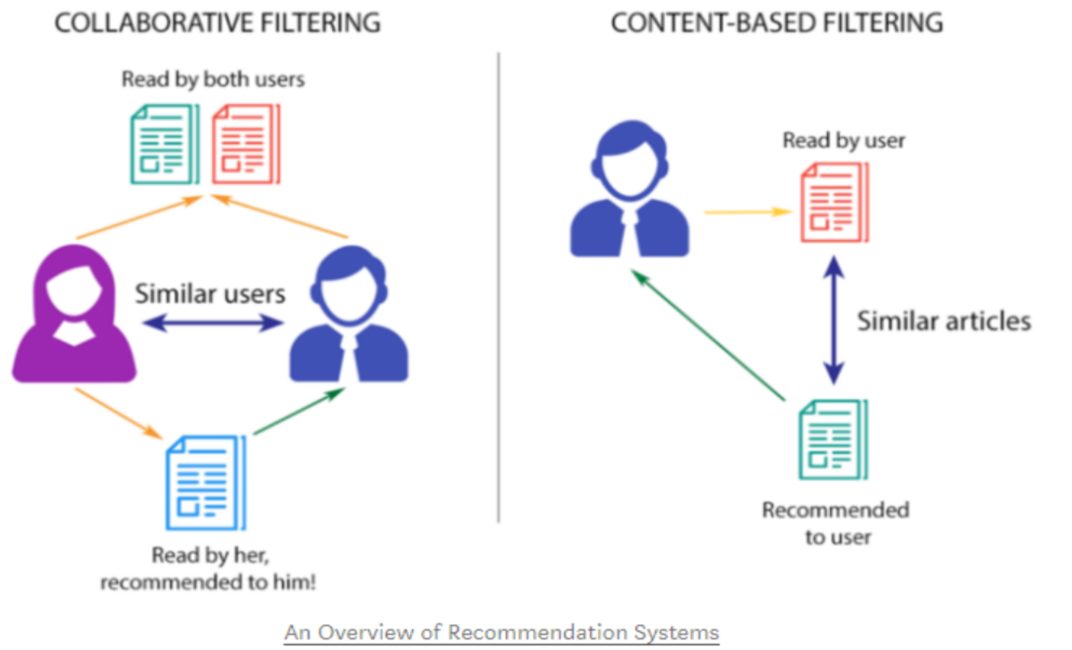
Movie Recommendation

Chapter 4

Implementation

In this section, present the implementation done by you during the project development.

4.1 Methodology OR Proposal



Content Based Filtering system:

It utilizes content-based sifting strategies to contrast various articles with clients' advantage profiles. Basically, a client profile contains content reasonable for use as usefulness.

This takes into account previous actions or feedback, and typically takes into account descriptions of content edited by users of various selections. Suppose someone buys their favorite item "M" but it's sold out, so they need to buy item "N" on someone's recommendation. Because "N" has the same match characteristics as the first item. So this is basically satisfied based on separation as displayed underneath.
Here, the cosine closeness between two classifications of movies is used to decide how comparable they are and to decide a score. The size of the score got from cosine closeness can be immediately determined.

The following are the actions needed to obtain a movie recommendation:

using the title to access the film's index. Determine the cosine similarity ratings for every film. putting the scores in ascending order, descending from the highest priority. after which the list is whittled down using the similarity scores. Get the first ten items on the list, omitting the first one since it is the title of the film itself. Obtaining the most important components

By rehashing the over steps, you will discover the best motion pictures that can get you the best conceivable suggestion based on separate.Ultimately, the recommender shows users movies that the general user base might like, and uses the information gathered through various activities to try to find users with different interests in other ways. Indian uses collaborative filtering to test all users with the same interests and get a final set of movies recommended individually to the user. Cosine similarity accounts for the dot product space, which can be defined as the scalar product of two vectors split by the product of Euclidean quantities, is the angle formed by two vectors when they are nonzero. Cosine similarity is typically employed to provide user-friendly recommendations.

This technique essentially utilizes the cosine distance between vectors, utilizes likeness to work out a score, and afterward computes the client's inclination. For instance, a film where entertainers characterize the quantity of preferences of clients and just few entertainers loathe a gathering of clients. In this manner, we expect that a decent sign point is drawn between the client vector and the film vector. This for the most part brings about an enormous positive fragmentary point near nothing, and there is a little cosine distance between the two vectors.

In some sense, the metric is better because the distance between the film and the cosine is large, and the similarity of the cosine is lost. In this case, a new method called decision tree is used to improve the recommendation system. This strategy as a rule includes layers. Child can apply a few conditions to its classification approach to progress its proposal framework to decide which motion pictures clients need to observe or not at all.

Advantages :

●Fresh Finds: Content-based filtering excels at recommending new items (like unrated movies) because it analyzes the item's features itself, not relying on other users' ratings.

●Personalized Picks: It personalized recommendations based on a user's past preferences (like ratings). If a user enjoys comedies with specific actors, the system can suggest similar movies.

Disadvantages:

●Echo Chamber Effect: This approach can get stuck in an echo chamber, recommending items very similar to what the user already likes. It might miss out on suggesting new genres or movies the user might enjoy but hasn't interacted with yet.
●Limited Exploration: Since recommendations focus on similar items, users might not be exposed to diverse options that could broaden their interests. This can hinder business growth if users stick to familiar categories and don't explore new products.

Collaborative based Filtering:

Content-based sifting has a few impediments as it can as it were to recommend motion pictures that the client has a inclination for and cannot suggest sorts. In any case, frameworks based on collaborative sifting make it exceptionally complex to discover datasets between users' similitudes and their peers with comparative interface. Cosine likeness or Pearson relationship is utilized to degree the closeness of client sees. For illustration, in the taking after lattice, there is a client in each push of the column that compares to motion pictures with the same similitude. There are too client evaluations of different motion pictures. Each motion picture has a target client.

For user-based sifting, all collaborative sifting is straightforward, but it too has downsides. The primary challenge is that client choices alter over time. Framework precomputation illuminated the execution corruption issue. Subsequently, item-based collaborative sifting can be utilized. This basically considers things based on their likeness to the thing and recommends that the same closeness coefficient can be utilized with Pearson relationship or cosine similitude when finding comparable matches with the target client. Component based agreeable filtering is the most inactive in nature. For illustration, he is the as it were client who has related both The Network and Titanic, so there is as it were one closeness between them. These two distinctive motion pictures can too be exceptionally comparable since they have millions of clients and these two diverse motion pictures rank the same for clients who have evaluated both.

When performing collaborative sifting, attempt to discover out which clients are interested in names and comparable settings. In this case, or maybe than utilizing thing characteristics to make suggestions, we classify clients into clusters of comparable characteristics & part each cluster in arrange of client inclinations. You can too utilize cosine remove here. This considers clients with the same intrigued, which is more prominent than a little cosine point between two clients. Here you can essentially utilize a utility network and relegate invalid values to the meager columns to make the calculation less demanding. Object-based collaborative sifting is by and large favored since it considers motion pictures or maybe than the number of clients, assist streamlining the classification of motion pictures and clients.

Hence, user-based collaborative sifting is not prescribed as it essentially takes into account the clients and overlooks the inadequate values that cause issues in highlighting the recommender system's execution.

Presently I would like to change over the suggestion issue into an optimization issue. The most favored and common metric is the root cruel square blunder (RMSE). The way better the execution, the lower the RMSE esteem.

Preferences of collaborative sifting based frameworks are:

. It is basically substance dependent

. It regularly peruses the intellect of individuals having same preferences

. Make genuine quality appraisal of things.

The disadvantages of collaborative filter systems are:

The early rating problem is the most common problem where collaborative filtering techniques fail to provide ratings for movies without the user having to wait. . Sparsity issues are more common with this type of welding method. There are so many zero values that it is difficult to find an element that is valued by the majority of people.
Algorithms Used

Cosine Similarity
Cosine closeness accounts for the point between two vectors when the vectors are nonzero, and the speck item space is portrayed as the scalar item of two vectors isolated by the item of Euclidean amounts. The littler the point, the more prominent the likeness. Cosine likeness is much better than adjusted remove since the point is littler in cosine closeness.

RMSE (Root Mean Square Error)

RMSE is basically fair the standard deviation of the expectation mistake. Remaining mistake. The relapse degree where the information focuses are found. Be that as it may, it moreover appears the dispersion of residuals inside the information focuses and decides the best fit to the information. They are moreover utilized in forecast and relapse investigation to discover exploratory residuals and approved comes about. The superior the execution, the lower the RMSE esteem.

$$\mathbf{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}}$$

$\mathbf{RMSD}$ = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

4.2 Testing OR Verification

Test ID	Test Case Title (Movie title)	Test Condition (checking weather input is string or not)	System Behavior	Expected Result
T01	The avengers	yes	predicted successfully	Guardians of the Galaxy Vol. 2 Aliens Guardians of the Galaxy The Martian Interstellar Blade Runner Kill Bill: Vol. 1 The Thing Spider-Man: Homecoming The Terminator
T02	Gladiator	yes	predicted successfully	'Ben-Hur', 'The Sting', 'On the Waterfront', 'Deadpool', 'Star Wars: Episode VI - Return of the Jedi', 'Roman Holiday', 'Léon: The Professional', 'The Martian', 'Kill Bill: Vol. 1', 'The Godfather: Part II'
T03	The Shawshank Redemption	yes	predicted successfully	Pulp Fiction Se7en

				Rope Goodfellas Hachi: A Dog's Tale The Green Mile The Great Escape Million Dollar Baby Beauty and the Beast Unforgiven

4.3 Result Analysis OR Screenshots

```
recommend_movies('The Avengers')

['Guardians of the Galaxy Vol. 2',
 'Aliens',
 'Guardians of the Galaxy',
 'The Martian',
 'Interstellar',
 'Blade Runner',
 'Kill Bill: Vol. 1',
 'The Thing',
 'Spider-Man: Homecoming',
 'The Terminator']
```

Chapter 5

Conclusion

Hence, when executing a cross breed substance and collaborative sifting procedure, consider a half breed approach to make strides the in general framework execution and suggest motion pictures agreeing to the user's determination in a much way better way than the other two suggestions. frameworks. To do. Diminishing the normal blunder encourage makes strides the exactness of the recommender framework, making it way better usable in future applications.

In spite of the fact that there are moreover efficient computational impediments and confinements for applying recommender frameworks to huge datasets, we have made a great exertion here to recognize between different recommender frameworks and the last has a half breed recommender framework on beat of everything.

From this, we can conclude that hybrid-based sifting makes a difference to make the framework fracture more effective and make strides the generally framework exactness. And blending both substance in a collaborative sifting strategy guarantees that indeed if one strategy falls flat, the other strategy keeps up the in general exactness of the framework and basically moves forward the by and large execution. There is no question that it implies.

Future Scope

1.) In case of content based filtering method we can look up on the cast and crew also where we have only considered the genre and also we can see at the movies are compatible or no t.

2.) Comparison of collaborative filtering based approaches and different kind of similarity measurements would be a good one for the recommender system

3.) We can use matrix factorization for calculating the number of factors involved.

4.) We can also apply deep learning techniques to for the the enhance the recommender system and optimising the efficiency of the system.

5.) We can work on different areas such as video some books aur even recommending some songs to the users of the mobile phones based on the platforms of the different apps available on the Play Store

6.) Various techniques such as clustering classification can be used to get the better version of our recommender system which for the enhance the accuracy of the overall model.

References

[1] Hirdesh Shivhare, Anshul Gupta and Shalki Sharma, "Recommender system using fuzzy cmeans clustering and genetic algorithm based weighted similarity measure" Communication and Control. IEEE International Conference on Computer, 2015.
[2] Manoj Kumar, D.K. Yadav, Ankur Singh and Vijay Kr. Gupta. "A Movie Recommender System: MOVREC, International Journal of Computer Applications (0975-8887) Volume 124-No.3, 2015.
[3] RyuRi Kim, Ye Jeong Kwak, HyeonJeong Mo, Mucheol Kim, Seungmin Rho,Ka Lok Man, Woon Kian Chong "Trustworthy Movie Recommender System with Correct Assessment and Emotion Evaluation". Proceedings of the International MultiConference of Engineers and Computer Scientists Vol II, 2015
[4] Zan Wang, Xue Yu*, Nan Feng, Zhenhua Wang, "An Improved Collaborative Movie Recommendation System using Computational Intelligence" Journal of Visual Languages & Computing. Volume 25, Issue 6, 2014
[5] Debadrita Roy, Arnab Kundu, "Design of Movie Recommendation System by Means of Collaborative Filtering". International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4, 2013.
[6] E. Amolochitis, Algorithms and applications for academic search, recommendation and quantitative association rule mining. Gistrup, Denmark: River Publishers, 2018.
[7] T. Brodt, Collaborative Filtering. Saarbrücken: VDM, Müller, 2010.
[8] M. Jalali, H. Gholizadeh and S. Hashemi Golpayegani, "An improved hybrid recommender system based on collaborative filtering, content based, and demographic filtering", International Journal of Academic Research, vol. 6, no. 6, pp. 22-28, 2014. 37
[9] C. Li and K. He, "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis", Concurrency and Computation: Practice and Experience, vol. 29, no. 10, p. e4092, 2017.

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION

NARREDDHULA PRANAV REDDY
2105637

Abstract: The recommendation system engine utilizes demographic filtering, content-based filtering, and collaborative filtering techniques to enhance recommendation accuracy. Demographic filtering suggests popular movies based on genre and user demographics. Content-based filtering tailors recommendations to individual interests, while collaborative filtering identifies users with similar interests. By employing these methods, the system maximizes user satisfaction and potential profits by offering personalized, in-demand movie selections.

Individual contribution and findings: Project's data preprocessing phase, I focused on integrating the two key datasets: "movies.csv" and "ratings.csv". Initially, I merged these datasets, combining movie information with corresponding user ratings. This step ensured that we had a comprehensive dataset to work with. Subsequently, I calculated the total count of ratings for each movie, providing valuable insights into their popularity and user engagement levels. Additionally, I created a Pivot Table using the processed data, enabling us to efficiently summarize and analyze the vast amount of information available. This Pivot Table serves as a valuable resource for the recommendation model, allowing for quick and interactive data summarization. Overall, my efforts in data preprocessing laid the groundwork for the effective implementation of the recommendation system, ensuring that it can provide accurate and relevant movie recommendations based on user preferences and engagement metrics.

Individual contribution to project report preparation: my individual contribution centered on crafting Chapter 1, encompassing the abstract, keywords, and introduction sections. In the abstract, I succinctly summarized the project's objectives, methodologies, and key findings, providing a concise overview to readers.

Individual contribution for project presentation and demonstration:my individual contribution focused on delivering the introduction and explaining the two main types of filtering: content-based and collaborative-based. I provided an overview of each filtering method, highlighting their respective approaches to recommending movies based on user preferences and behavior. Furthermore, I presented our data preprocessing steps, illustrating how we prepared the datasets for analysis by cleaning and formatting the data.

Full Signature of Supervisor: Full signature of the student:

……………………………

N. Pranav Reddy

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION

B Bhanu Prasanth
2105881

Abstract: The recommendation system engine utilizes demographic filtering, content-based filtering, and collaborative filtering techniques to enhance recommendation accuracy. Demographic filtering suggests popular movies based on genre and user demographics. Content-based filtering tailors recommendations to individual interests, while collaborative filtering identifies users with similar interests. By employing these methods, the system maximizes user satisfaction and potential profits by offering personalized, in-demand movie selections.

Individual contribution and findings:
For data preprocessing, I utilized the "IMDB_Top250Engmovies2_OMDB_Detailed.csv" dataset comprising 250 movies and their associated attributes: 'Plot', 'Genre', 'Actors', and 'Director'. The processing steps involved: Loading the dataset; Converting data to lowercase and removing numbers, punctuation, and spaces, resulting in clean dataset.Finally, merging the cleaned data from all columns into clean_input. This preprocessing ensured standardized and uniform data for subsequent analysis.

Individual contribution to project report preparation: In Chapter 3, Implementation, I contributed to the section outlining the Project planning ,Project analysis system design.

Individual contribution for project presentation and demonstration: Data Preprocessing for content based filtering using Regular language and natural language tool kit libraries

Full Signature of Supervisor:
…………………………….

Full signature of the student:
B Bhanu Prasanth

Movie Recommendation

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION

MANISH KUMAR SINGH
2105899

Abstract: The recommendation system engine utilizes demographic filtering, content-based filtering, and collaborative filtering techniques to enhance recommendation accuracy. Demographic filtering suggests popular movies based on genre and user demographics. Content-based filtering tailors recommendations to individual interests, while collaborative filtering identifies users with similar interests. By employing these methods, the system maximizes user satisfaction and potential profits by offering personalized, in-demand movie selections.

Individual contribution and findings:

Setup and Initialization:

I import necessary libraries including numpy, scipy, and pandas for data manipulation, as well as scikit-learn's NearestNeighbors for implementing the recommendation system. I define a custom class called NearestNeighbors which implements the k-nearest neighbor's algorithm. The class constructor initializes parameters such as the number of neighbors (default is 5), the distance metric (default is cosine), and the algorithm for finding neighbors (default is brute force).

Training the Model:

The fit method of the NearestNeighbors class is used to train the model. It takes the feature matrix of movies as input and stores it as the training data.

Generating Recommendations: To generate recommendations for a given movie, we randomly select a movie index from the dataset. Using the kneighbors method of the NearestNeighbors class, I compute the k- nearest neighbors of the selected movie. I then iterate through the distances and indices of the nearest neighbors and print the recommendations along with their distances from the selected movie. Recommendations are based on the movies with the shortest distances to the selected movie, indicating their similarity

Individual contribution to project report preparation: In Chapter 4.1, Implementation, I contributed to the section outlining the recommendation generation process based on the K-nearest neighbor's method, focusing on Content-Based Filtering system and Collaborative Filtering techniques.

Individual contribution for project presentation and demonstration: I took the lead in preparing the presentation slides for the K-nearest neighbors (KNN) implementation and provided a detailed explanation of how the KNN algorithm works and its application in generating movie recommendations.

Full Signature of Supervisor: Full signature of the student:

…………………………… Manish Kumar Singh

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION

NUKALA KAGHUVAHAN REDDY

Abstract: The recommendation system engine utilizes demographic filtering, content-based filtering, and collaborative filtering techniques to enhance recommendation accuracy. Demographic filtering suggests popular movies based on genre and user demographics. Content-based filtering tailor's recommendations to individual interests, while collaborative filtering identifies users with similar interests. By employing these methods, the system maximizes user satisfaction and potential profits by offering personalized, in-demand movie selections.

Individual contribution and findings: In this project, my contribution involved collecting and analyzing data for two distinct models used in the movie recommendation system. For the collaborative filtering model, we utilized the Movie.csv dataset obtained from Kaggle, consisting of 9,743 movies and 100,147 reviews. Each movie and user was assigned a unique ID for identification. Meanwhile, for the content-based model, we employed the IMDB dataset, focusing on attributes like actors, plot, director, and genre for 250 movies. Following data collection, I conducted a comprehensive model study to assess user preferences and the effectiveness of each recommendation approach. The study revealed that most users preferred movies recommended by the content-based model, which operates by filtering movies based on their attributes. This model detects similarities between movies using contextual information, considering a user's past history to recommend similar movies. In comparison, the collaborative model suggests movies based on a user's past preferences and those of other users. Our survey results confirmed that the content-based model provided recommendations closer to user preferences, making it a more efficient choice for movie recommendations.

Individual contribution to project report preparation: my individual contribution focused on writing Chapter 5, which encompasses the conclusion, future scope, and references sections. In the conclusion, I summarized the key findings and insights obtained from the project, highlighting the significance of the research and its implications. I also reflected on the effectiveness of the implemented recommendation system and its contributions to the field.

Individual contribution for project presentation and demonstration: my individual contribution involved discussing the process of dataset collection and comparison between the two models. I outlined how we collected data from two distinct datasets. Additionally, I presented the results of our comparison study, highlighting the differences and similarities between the two recommendation models. Furthermore, I shared insights from a survey conducted by us to gauge user preferences regarding recommendations from both models.

Full Signature of Supervisor: Full signature of the student:

…………………………… Nukala Kaghuvahan Reddy

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION

VEMANA SUMANTH
21052803

Abstract: The recommendation system engine utilizes demographic filtering, content-based filtering, and collaborative filtering techniques to enhance recommendation accuracy. Demographic filtering suggests popular movies based on genre and user demographics. Content-based filtering tailors' recommendations to individual interests, while collaborative filtering identifies users with similar interests. By employing these methods, the system maximizes user satisfaction and potential profits by offering personalized, in-demand movie selections.

Individual contribution and findings:

- Implemented feature extraction to converts unprocessed text data into numerical vectors using TF - IDF (Term Frequency - Inverse Document Frequency).
- Implemented cosine similarity to find the similarity between two non-zero vectors in an inner product space.
- Define a function called recommend movies(arg1) that takes a movie title as input and gives output top ten movies related to the input.
- Additionally, we attempt other film titles like 'The Shawshank Reclamation' or

  'The Justice fighters' to get proposals.

Individual contribution to project report preparation: Wrote and researched chapter 2 and chapter 4.1, 4.2 in the report which explains about the basic concepts/ literature review and RMSE and testing and verification required for the project.

Individual contribution for project presentation and demonstration:
Wrote and explained about the implementation of feature extraction and cosine similarity

Full Signature of Supervisor: ……………………………

Full signature of the student: vemana sumanth

MOVIE RECOMMENDATION

ORIGINALITY REPORT

6%	6%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	5%
2	www.worldleadershipacademy.live Internet Source	1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		