MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

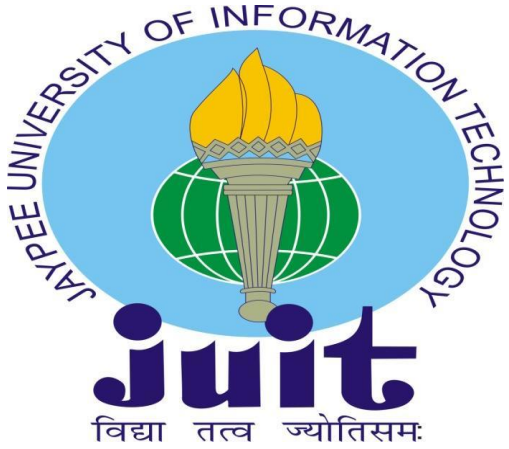
In

Computer Science and Engineering/Information Technology

By

KUNAL GARG(191445)

UNDER THE SUPERVISION OF
Dr. YUGAL KUMAR
to



Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE DECLARATION

I hereby declare that the work presented in this report entitled **"Movie Recommendation System using Machine Learning"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat, is an authentic record of my own work carried out over a period from August 2022 to December 2023 under the supervision of **Dr. Yugal Kumar** (Associate Professor in Computer Science and Engineering & Information Technology)

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Data Science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

KUNAL GARG (191445)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Dr. Yugal Kumar

Associate Professor

CSE&1T Department

Dated:

PLAGIARISM CERTIFICATE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: .....

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ **Department:** _____ **Enrolment No** _____

Contact No. _____ **E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><td>Copy Received on</td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="3"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

.....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

With the guidance and assistance of numerous well-wishers, an endeavor over a lengthy period of time can be effective. We would like to take this time to let everyone know how much we appreciate them.

In the beginning, I'd like to express our gratitude to our supervisor, Dr. Yugal Kumar, Associate Professor, Department of Computer Science & Engineering and Information Technology Jaypee University of Information Technology (JUIT), for his invaluable support and direction throughout the project's implementation.

We wish to express our sincere thanks and gratitude to our project guide, Dr. Yugal Kumar, Associate Professor, Department of Computer Science & Engineering and Information Technology at Jaypee University of Information Technology (JUIT), for the stimulating discussions, in analyzing problems associated with our project work, and for guiding us throughout the project. Project meetings were highly informative. We express our warm and sincere thanks for the encouragement, untiring guidance, and confidence she has shown in us. We are immensely indebted to her for her valuable guidance throughout our project.

KUNAL GARG (191445)

TABLE OF CONTENT

LIST OF ABBREVIATIONS

SHORT FORM		MEANINGS
TFIDF	=	Term Frequency - Inverse Document Frequency
SVM	=	Support Vector Machine
DT	=	Decision Tree
GBC	=	Gradient Boosting Classifier
LR	=	Logistic Regression
RFC	=	Random Forest Classifier
CV	=	Count Vectorizer
FIG	=	Figure

LIST OF FIGURES

LIST OF GRAPHS

LIST OF TABLES

ABSTRACT

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically. For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favourable movies online we can utilize movie recommendation systems, which are more reliable, since searching of preferred movies will require more and more time which one cannot afford to waste. In this paper, to improve the quality of a movie recommendation system, a Hybrid approach by combining content based filtering and collaborative filtering, using Support Vector Machine as a classifier and genetic algorithm is presented in the proposed methodology and comparative results have been shown which depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches in three different datasets. Hybrid approach helps to get the advantages from both the approaches as well as tries to eliminate the drawbacks of both methods.

CHAPTER-1
INTRODUCTION

1.1) Introduction

The study of statistical models and methods used by computers to do certain tasks devoid of explicit instructions and in favour of patterns and inference is known as machine learning (ML). It's thought to be a part of artificial intelligence. Without being explicitly told to do so, machine learning algorithms create a mathematical model from sample data, or "training data," in order to produce conclusions or predictions. There are many similarities between machine learning and computational statistics, which focuses on computer-aided prediction. Machine learning benefits from the methodologies, theories, and fields of application created through the study of mathematical optimisation.

A recommendation system, sometimes known as a recommendation engine, is a paradigm for information filtering that aims to anticipate user preferences and offer suggestions in accordance with these preferences. These technologies are now widely used in a variety of industries, including those that deal with utilities, books, music, movies, television, apparel, and restaurants. These systems gather data on a user's preferences and conduct, which they then employ to enhance their suggestions going forward.

Movies are a fundamental aspect of life. There are many various kinds of movies, such as those meant for amusement, those meant for teaching, children's animation movies, horror movies, and action movies. Movies' genres, such as comedy, thriller, animation, action, etc., make it simple to distinguish between them. Another approach to differentiate between movies is to look at their release year, language, director, etc. When watching films online, there are many to choose from in our list of top picks. We can use movie recommendation systems to find films based on favoured films among all of these other movie genres, saving us the hassle of having to spend a lot

of time looking for our favourite films. As a result, it is essential that the system for suggesting films to us is very trustworthy and gives us recommendations for the films that are either most similar to or identical to our tastes.

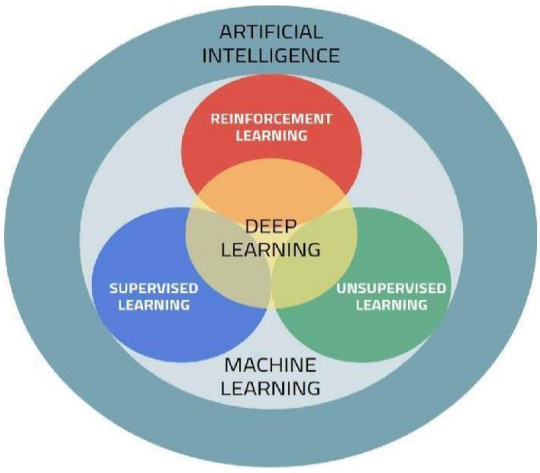


Figure:1.1

1.1.1) Natural Language Processing (NLP)

A branch of computer science and artificial intelligence known as "natural language processing," or NLP, studies how computers interact with human (natural) languages with the goal of effectively teaching computers to analyse massive volumes of natural language data. The study of how computers interact with human (natural) languages is known as natural language processing (NLP), and it is a branch of linguistics, computer science, information engineering, and artificial intelligence. Its primary goal is to train computer programmers to evaluate and interpret vast amounts of natural language.

.

1.1.2) Movie Recommendation

Recommendation systems are being used by a lot of businesses to improve customer interaction and the purchasing experience. The most significant advantages of recommendation systems are client happiness and income. A very effective and crucial mechanism is the movie recommendation system. However, because of the limitations with a pure collaborative method, scalability concerns and poor recommendation quality also affect movie recommendation systems.

This feature certainly intrigues me. As a result, the main duty of a recommender system is to provide the user with the most useful recommendations. While Amazon, Flipkart, and Netflix utilise recommendation algorithms for product recommendations, Amazon Prime and YouTube use them for movie recommendations.

Any action you undertake on these websites is being tracked by a system, which then makes suggestions for goods or products that you are very likely to find interesting. This study looks at movie recommendations, the logic behind them, as well as more traditional movie recommendation systems and a solution for an AI-based customised movie recommendation system.

From a business perspective, user engagement is higher the more relevant products they discover on the site. Increasing platform income is a common outcome of this. From a business perspective, user engagement is higher the more relevant products they discover on the site. Increasing platform income is a common outcome of this. Various sources claim that as much as 35–40% of the revenue of internet behemoths comes from only referrals.

1.2) Problem Statement

The goal is to develop a movie recommendation system that can provide users with tailored movie suggestions based on their tastes in films. Based on a user's historical movie ratings and preferences, as well as suggestions of comparable films seen by other users with similar likes, the system should be able to forecast with accuracy which films the user will likely appreciate. The system should also be able to scale easily and handle enormous volumes of data with efficiency. By suggesting films that the user is likely to enjoy, the system hopes to improve user experience and increase user engagement and retention.

1.3) Objective

Creating a personalised system that can make movie suggestions to users based on their prior movie choices is the main goal of a movie recommendation system project. Utilising machine learning algorithms, the system will analyse user data and produce recommendations that are relevant to their interests. The following objectives are the focus of the project:

- Providing accurate and customized movie recommendations to users based on their past behavior and preferences.
- Analyzing user data utilizing machine learning algorithms to produce movie recommendations.
- Considering multiple factors such as user movie ratings, movie genre, movie director, and similar movies watched by other users with similar preferences to produce recommendations.
- Designing a scalable and effective system capable of processing large amounts of data and delivering rapid recommendations.
- Enhancing user engagement and retention by providing a smooth and enjoyable movie recommendation experience

1.4) Methodology

1.4.1) Dataset

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question.To develop a movie recommendation system, you can follow these general steps:

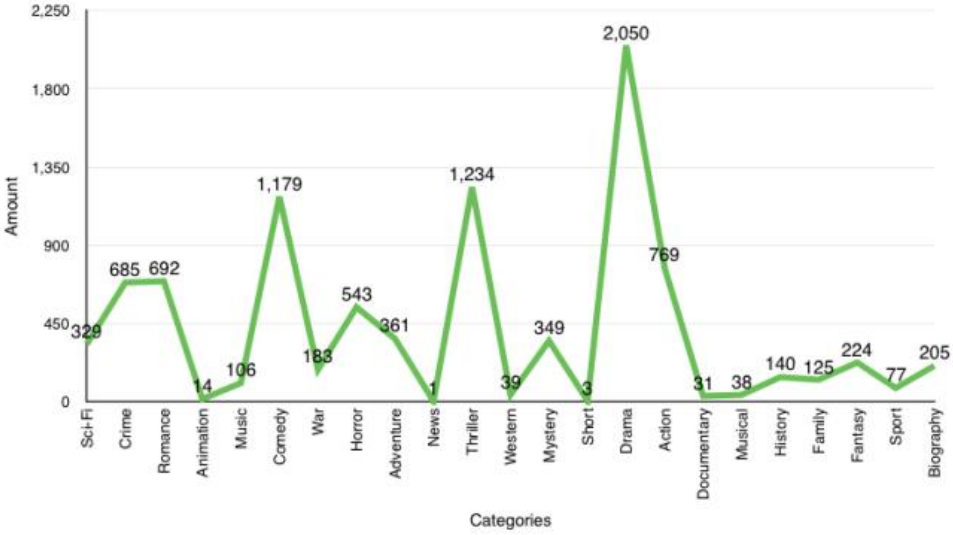
1. Collect data on movies, such as their title, genre, director, actors, release date, and ratings. You can use public datasets such as the IMDb dataset, or collect your own data through web scraping or API calls.

Movie dataset has

- **Movie Id** – once the recommendation is done, we get a list of all similar movieId and get the title for each movie from this dataset.
- **Genres** – which is not required for this filtering approach.
- **Budget** – money spent in making money
- **Original language** – initial language in which movie is made
- **Production Companies**- it tells company that made the movie
- **Cast** – actors and actresses the acted in the movie
- **Keywords** – words that describe the movie and can be used to indentify the movie

Catergory Distribution:

It shows which category or genre movie is being seen maximum number of people



Graph:1.1

2. Clean and preprocess the data by removing duplicates, missing values, and irrelevant columns. You can also use feature engineering techniques to create new features that capture the characteristics of the movies. To remove duplicates in machine learning, the following steps can be taken:

- Locate and identify duplicate records in the dataset using a unique identifier or a combination of attributes.
- Choose a criterion for selecting one record from each group of duplicates, such as selecting the first occurrence or the one with the highest or lowest value of a particular attribute.

- Eliminate the duplicate records from the dataset, retaining only the chosen record for each group of duplicates.

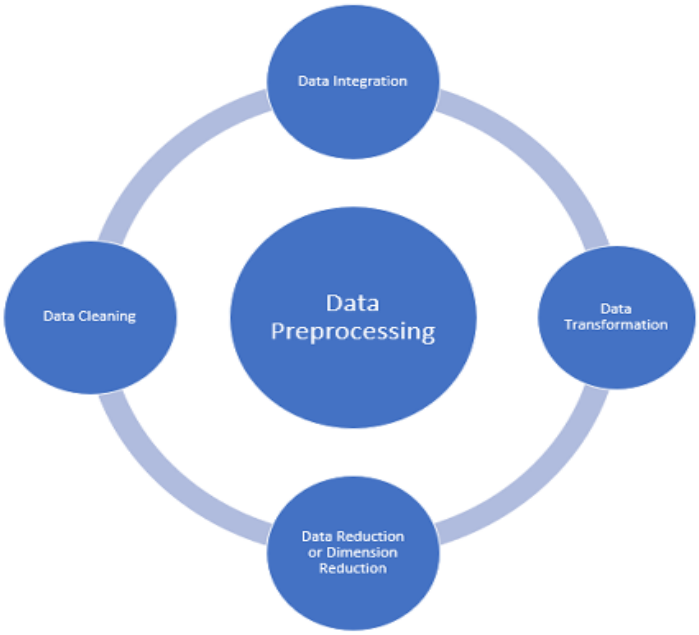
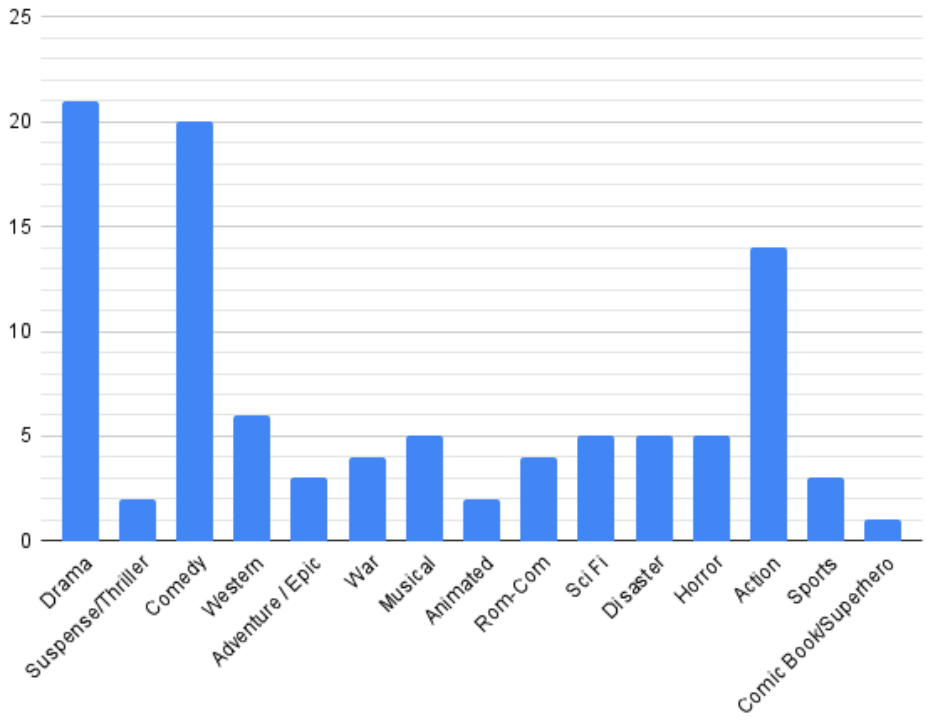


Figure:1.2

3. Perform exploratory data analysis to gain insights into the data, such as the distribution of ratings, the most popular genres, and the correlations between different features. Examining and interpreting data to derive important insights and conclusions is the process of data analysis. It entails analysing huge datasets using a variety of statistical and computational tools to find patterns, trends, and relationships.

   Using descriptive statistics, we may enumerate and explain the key characteristics of a dataset. These include measures of variability like standard deviation and range as well as measures of central tendency like mean, median, and mode.

Using inferential statistics, it is possible to predict and infer information about a broader population from a sample of data. Confidence intervals and hypothesis testing are some of these methods.



Graph:1.2

4. Choose a machine learning algorithm to build the recommendation system. Some popular algorithms for recommendation systems include collaborative filtering, content-based filtering, and hybrid filtering. Movie recommendation systems mainly use three types of algorithms to provide personalized recommendations to users:

- Content-Based Filtering: This algorithm analyzes a user's previous movie ratings or preferences and recommends new movies based on the content

of the movies. It identifies patterns and similarities between the movies and suggests new movies that have similar characteristics to the ones that the user has already watched and liked.

- Collaborative Filtering: This algorithm recommends movies based on the user's behavior and patterns in the past. It analyzes the user's movie ratings and preferences, as well as those of other users with similar tastes. Based on this analysis, it identifies movies that the user may be interested in and recommends them.

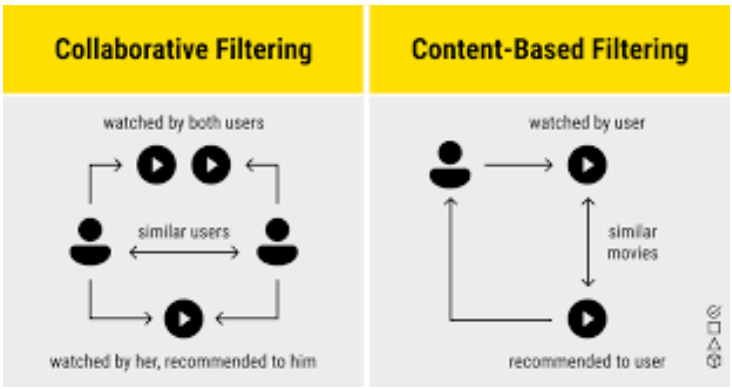


figure:1.3

- Hybrid Recommendation: This algorithm combines both content-based and collaborative filtering algorithms to provide a more accurate and personalized recommendation. It leverages the strengths of both algorithms to overcome their individual weaknesses and produce more relevant and effective recommendations.

- Overall, these three types of algorithms are widely used in the development of movie recommendation systems to enhance user engagement and satisfaction by providing tailored and relevant recommendations.

5. Train the model on the movie data, using techniques such as matrix factorization, deep learning, or clustering. Training a movie recommendation model involves feeding it with a dataset of movie ratings and other relevant information, such as movie genres, actors, directors, and release years.

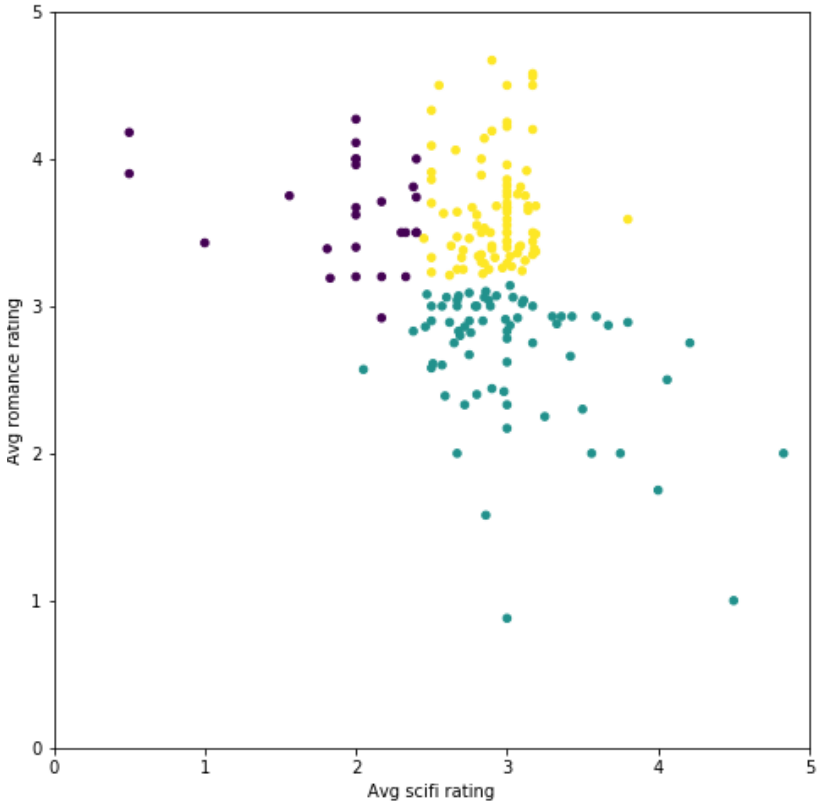


Figure:1.4

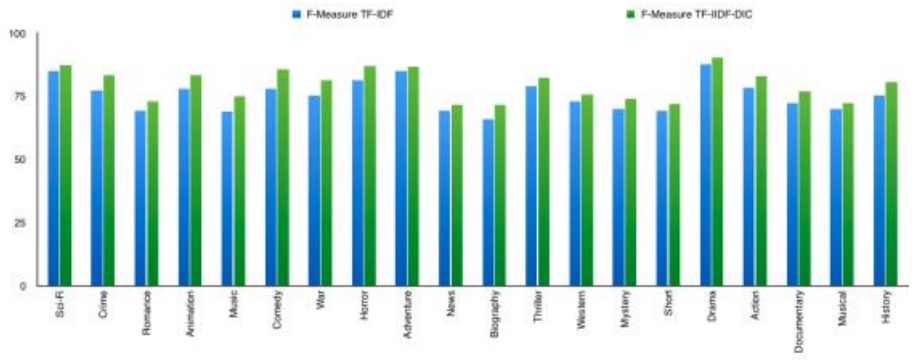
6. Evaluate the performance of the model using metrics such as precision, recall, and F1 score. You can also use techniques such as A/B testing or user studies to evaluate the user satisfaction with the recommendations. The evaluation of a movie recommendation system's performance is essential to ensure that it is delivering accurate and relevant recommendations to its users. Below are some

of the metrics typically used to evaluate the performance of a recommendation system:

- **Mean Absolute Error (MAE**): This metric measures the average absolute difference between the predicted and actual ratings. The lower the MAE, the better the system's performance.

- **Root Mean Square Error (RMSE):** This metric measures the square root of the average squared difference between the predicted and actual ratings. As with MAE, a lower RMSE indicates better performance.

- **Precision and Recall:** These are classification metrics used to evaluate the accuracy of binary recommendation systems that predict whether a user will like a movie or not. Precision measures the proportion of recommended movies that the user actually liked, while recall measures the proportion of movies that the user liked that were recommended by the system.



graph:1.3

7. Deploy the recommendation system as a web application or API, where users can input their preferences and receive personalized movie recommendations. Model deployment is the action of implementing machine learning models. This makes the model's predictions accessible to users, developers, or systems, allowing them to interact with their application (such as identify a face in an image) or make business decisions based on data.

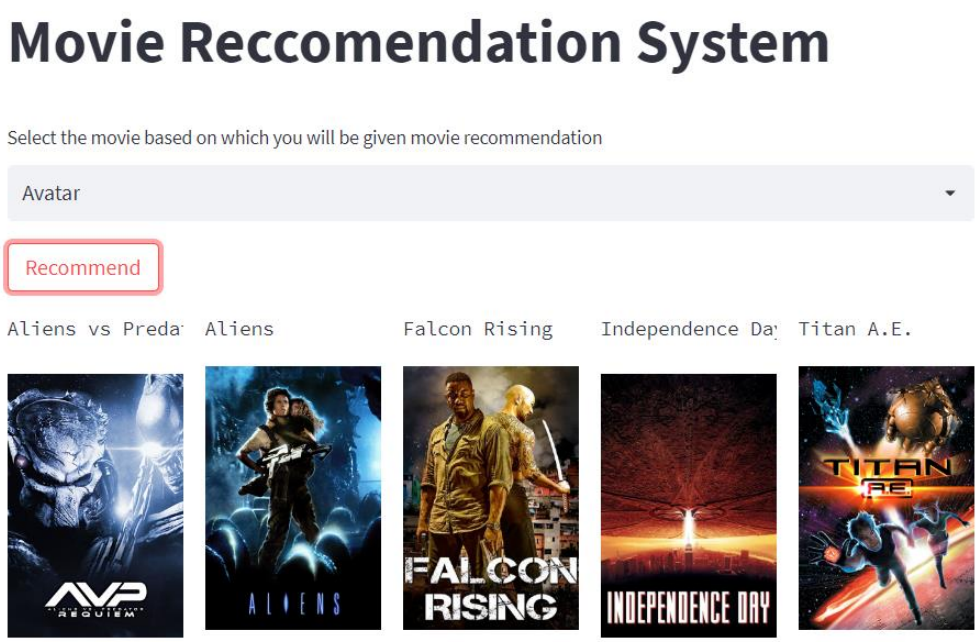


figure:1.5

When developing a methodology for a movie recommendation system, it's important to consider factors such as user demographics, user feedback, and the diversity of the recommended movies. Additionally, it's crucial to ensure that the system is transparent, interpretable, and respects user privacy

1.4.2) Flowchart:

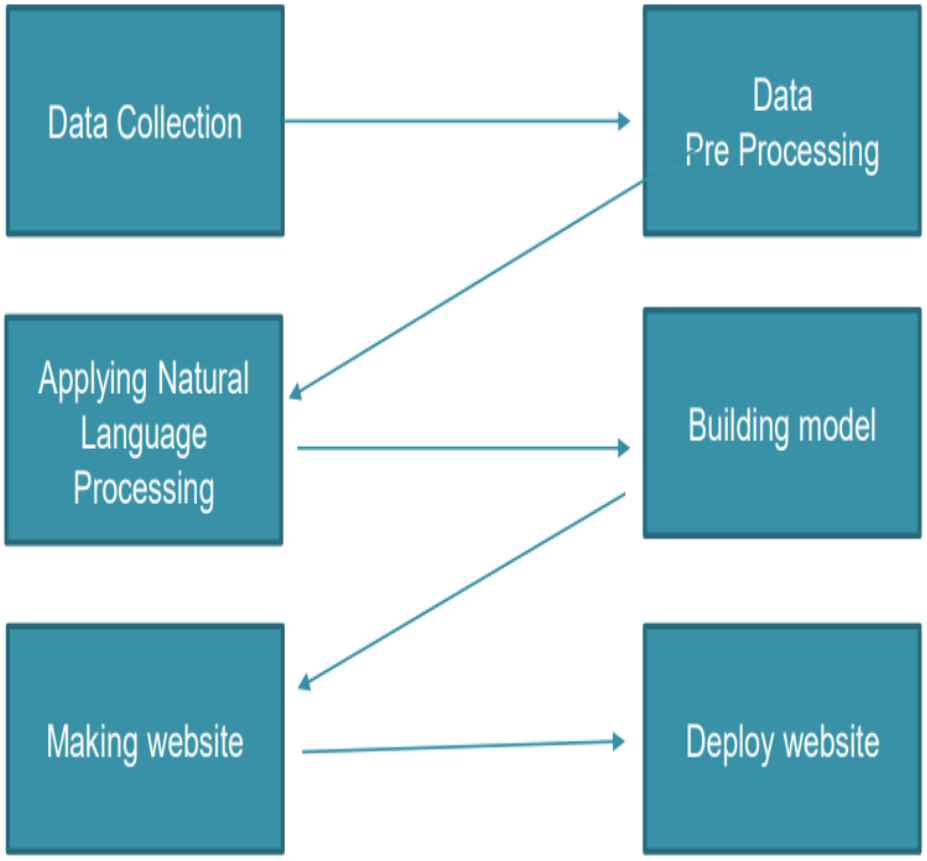


Figure:1.6

1.4.3) Algorithm for The Proposed System

Step 1: Pre-processing

- Removing the repeated attributes data.
- Clean the text by removing punctuation, stopwords, and lowercasing the text
- Split the dataset into training and testing sets

Step 2: Count Vectorization

- Use Count Vectorizer from sklearn library to convert the text data into numerical data
- Create a document-term matrix that represents the frequency of each word in each document
- Use the training set to fit the Count Vectorizer and transform the data
- Use the testing set to transform the data

Step 3: TFIDF Vectorization

- Use Tfidf Vectorizer from sklearn library to convert the text data into numerical data
- Create a document-term matrix that represents the importance of each word in each document
- Use the training set to fit the Tfidf Vectorizer and transform the data
- Use the testing set to transform the data

Step 4: Training the Models

- Use the transformed data from Count Vectorizer and Tfidf Vectorizer to train different models such as Naive Bayes, Logistic Regression, Support Vector Machines (SVM), Random Forest, etc.

- Use the training set to fit the models
- Use the testing set to predict the labels of the news articles
- Calculate the accuracy score of each model using the predicted labels and the actual labels

Step 5: Confusion Matrix

- Create a confusion matrix for each model to evaluate its performance

- The confusion matrix shows the number of true positives, true negatives, false positives, and false negatives
- Use the confusion matrix to calculate metrics such as precision, recall, and F1-score

Step 6: Accuracy

- Calculate the accuracy of each model using the predicted labels and the actual labels
- Accuracy is how close a given set of measurements are to their true value,

CHAPTER-2
LITERATURE SURVEY

Over the years, many recommendation systems have been developed using either collaborative, content based or hybrid filtering methods. These systems have been implemented using various big data and machine learning algorithms.

[1]Unnathi Bhandari, D. Garg, M. A. Maarof, and R. A. Rashid's paper ": A survey" is a survey; it contains no experiments or findings. Instead, the study presents a thorough analysis of the pros and cons of the many movie recommendation techniques suggested in the literature, as well as the datasets that were utilised to test them. The methods utilised by various research in terms of feature selection, feature extraction, classification algorithms, and assessment metrics are analysed and compared by the authors. They also emphasise the difficulties and potential avenues for future study in the area of movie recommendation systems.

[2] In the paper, MovieREC—a recommender system for movies—is introduced. It enables a user to choose from a predetermined set of criteria and then suggests a list of films for him based on the cumulative weight of the various attributes and the K-means algorithm. Author selects K initial centroids in the K-means clustering algorithm, where K is the required number of clusters. Each point is subsequently assigned to the cluster's centroid, which has the closest mean. Then, using the points allocated to each cluster, we update the centroid of each cluster.

[3] The author use content-based filtering, which is determined by the item's description and the user's preference profile. In CBF, we employ keywords in place of the user's profile to represent an item's preferred likes and dislikes. In other words, CBF algorithms promote products that were previously liked or products that are related to those products. It looks at previously rated things and suggests the best item that matches.

[4] The author contrasts different methods for creating a movie recommendation system. Hybrid recommender systems frequently combine these methods. An earlier study by Eyjolfsdottir et al. for the suggestion of films through MOVIEGEN had some shortcomings, including the time-consuming set of questions it asks consumers. However, it was not user-friendly due to the fact that it turned out to be somewhat stressful.

[5] With these drawbacks in mind, authors created MovieREC, a movie recommendation system that makes movie suggestions to consumers based on the data they submit. In the current study, a user has the ability to choose from a variety of variables, such as actor, director, genre, year, and rating, among others. Based on the preferences of users' prior visited histories, we forecast the users' selections. The system was created in PHP and at the moment only offers a straightforward console-based user interface.

[6] Author selects K initial centroids in the K-means clustering algorithm, where K is the required number of clusters. Each point is subsequently assigned to the cluster's centroid, which has the closest mean. After that, based on the points assigned to each cluster, the author updates the centroid of each cluster. Once the cluster centre (centroid) had not changed, the procedure was repeated. Last but not least, the objective of this algorithm is to minimise an objective function, in this case a squared error function.

[Ref. No.]	Author(s)	Published By (IEEE,Elsevier,Springer)	Pros and cons
[1]	Deepati Garg, Unnati Bhandari, Ching Sen	Movie Recommendation System Using Collaborative Filtering.	The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users. Accuracy is low as compared to other.
[2]	Nitasha Soni, Krishan Kumar, Ashish Sharma, Aman Yadav	Machine Learning Based Movie Recommendation System	The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in. It was trained only one model with accuracy of 80.035
[3]	Narendra Kumar Rao, Nagendra Pannini Challa	Movie Recommendation System using Machine Learning	Content-based filtering uses similarities in products, services, or content features, as well as information accumulated about the user to make recommendations.
[4]	P. Karthinkey, C. Tejaswani	Review of Movie Recommendation System	Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
[5]	Shourya Chawla, Sumita Gupta. Rana Majumdar	Machine Recommendation Models using Mchine Learning	The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

[6]	SKevin Andrews, KLakshmi Narayanan,K Balasubadra,M S Josephine	Web based movie recommendation system using content based filtering	Collaborative filtering relies on the preferences of similar users to offer recommendations to a particular user.

Table:2.1

CHAPTER-3
SYSTEM DEVELOPMENT

3.1) System Configuration

Common hardware can be used to run this project. We used an Intel I5 CPU with 8 GB of RAM, a 2 GB Nvidia graphics processor, and 2 cores with respective clock speeds of 1.7 GHz and 2.1 GHz to complete the project. Predictions may be made and accuracy can be assessed in a couple of seconds during the test phase, which follows the training phase and lasts for approximately 10-15 minutes.

3.2) Software Requirements

Distribution of anacondas:

Python is a free and open-source programming language that may be used for scientific computing (data science, machine learning), and Anaconda is a distribution of it that aims to simplify the package management system and deployment (for things like apps, big data processing, predictive analytics, etc.). Package versioning is managed by a system called Conda. The Anaconda distribution comes includes data science packages that work with Windows, Linux, and MacOS.3

3.2.1)Python Libraries

Scikit-Learn (SKlearn): The classification, regression, and clustering techniques in this collection include support vector machines, random forests, gradient boosting, k-means, and DBSCAN, among others. It is made to

function perfectly with Python's NumPy and SciPy scientific and numerical libraries.

NumPy: A well-liked all-purpose array processing package is NumPy. In addition to tools for working with these arrays, it offers a high-performance multidimensional array object. It is a foundational Python package for scientific computing.

Pandas: Another popular Python package in data science is Pandas. It offers user-friendly, high-performance structures and tools for data analysis. A DataFrame is a 2D table object that may be stored in memory in Pandas, as opposed to NumPy, which offers objects for multidimensional arrays.

Flask: The WSGI (Web Server Gateway Interface) web application framework Flask is compact. With the potential to scale up to complicated applications, it is made to set up quickly and effortlessly. It started out as a straightforward wrapper for Werkzeug and Jinja but has since grown to be one of the most well-liked Python web application frameworks.

Matplotlib: A plotting library for the Python programming language is called Matplotlib. It offers a selection of static, animated, and interactive Python visualisations.

In a movie recommendation system project, these libraries are essential for carrying out a number of tasks, including data processing, data visualisation, machine learning, and web application development. Developers can create effective and precise movie recommendation systems that offer customers personalised movie recommendations by utilising these libraries.

3.3) System Analysis & Design

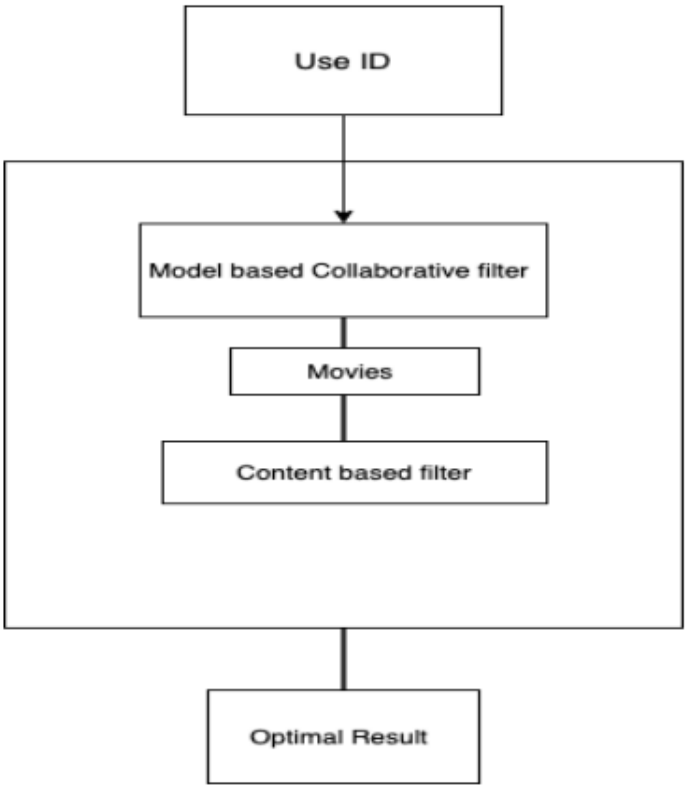


figure:3.1

System architecture for proposed model

A common strategy in recommender systems is content-based recommendation, which aims to suggest products that are comparable to those that a consumer has previously shown an interest in. A content-based recommender system's primary objective is to determine how similar products are to one another. There are several ways to model items, with the Vector Space Model being one of the most widely used.

The TF-IDF is used to extract keywords from items and determine their weights in the Vector Space Model. Let ki be the ith keyword and wij be the

weight of ki for the provided item, dj. So, a series of weights can be used to indicate the content of dj: Content(dj) is equal to "w1j, w2j,..."

$$Content(d_j) = \{w_{1j}, w_{2j}, ...\}$$

equation:1

Based on their history of liked items, a user's preference vector, Content-Based Profile(u), can be constructed to model their preferences. The following definition of Content-Based Profile(u) can be used if N(u) is the collection of items that user u has liked:

$$ContentBasedProfile(u) = \frac{1}{|N(u)|} sum_{d \in N(u)} Content(d)$$

equation:2

This makes it possible for the content-based recommender system to provide the user with recommendations based on their preferences and similarities among things. N(u) is the previous user that u loved. Given each user u and an item d, the similarity between the content vector Content(.) and the content preference vector Content Based Profile of all users indicates how the user feels about the item:

$$p(u, d) = sim(ContentBasedProfile(u), Content(d))$$

equation:3

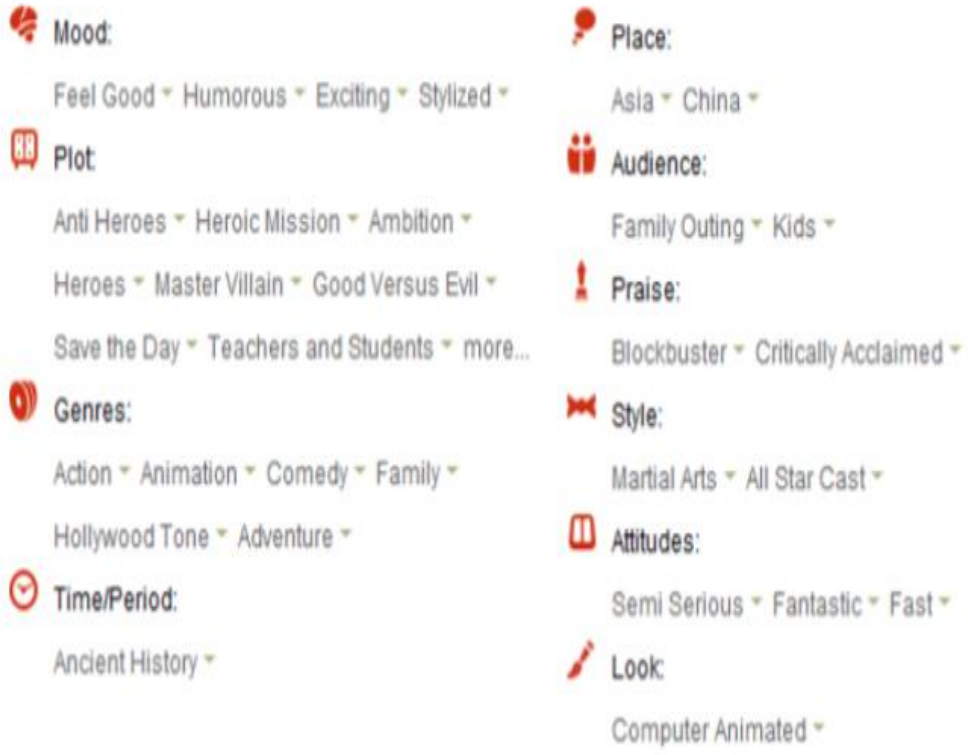


figure:3.2

Tags for movie Avatar

3.4) Activity Diagram

Activity diagrams are visual depictions of workflows with choice, iteration, and concurrency supported by activities and actions. Similar to the other four diagrams, activity diagrams serve similar fundamental goals. It captures the system's dynamic behaviour. The message flow from one item to another is depicted using the other four diagrams, whereas the message flow from one activity to another is depicted using the activity diagram.

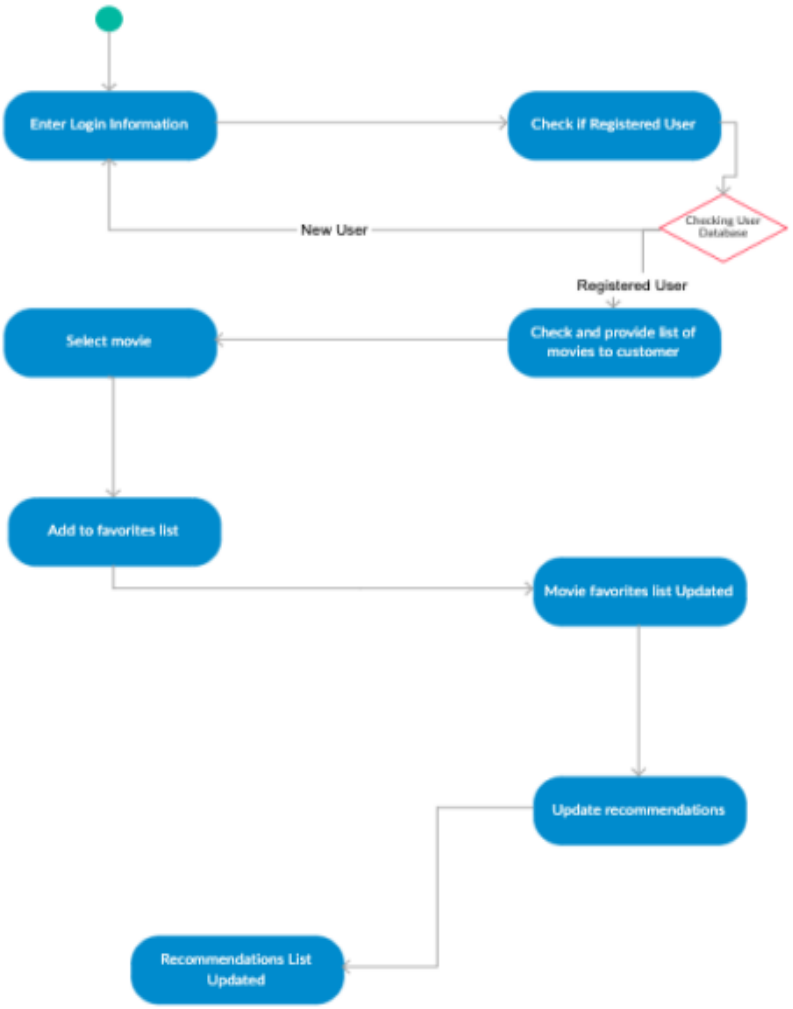


Figure:3.3

The user is given a list of recommended films after logging in using the user id, which is available in the csv file and ranges from 1-5000. After that, each movie in the test set is classified, which in our case involves assigning a genre to each movie. Since we now know the appropriate movie genre, the following part will look at the appropriate and erroneous categorizations and utilise metrics to judge the advancement.

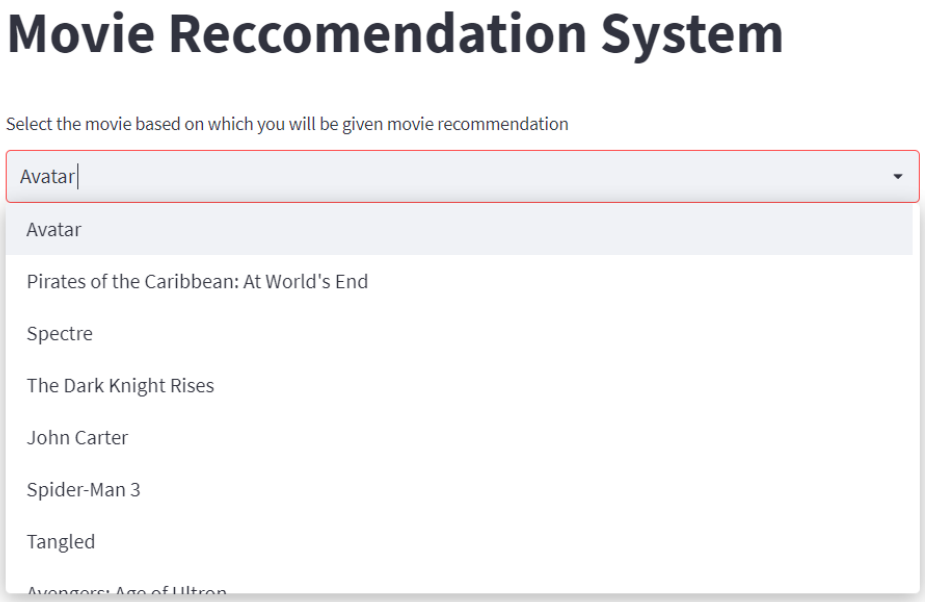


figure:3.4

3.5)Data Flow Diagram

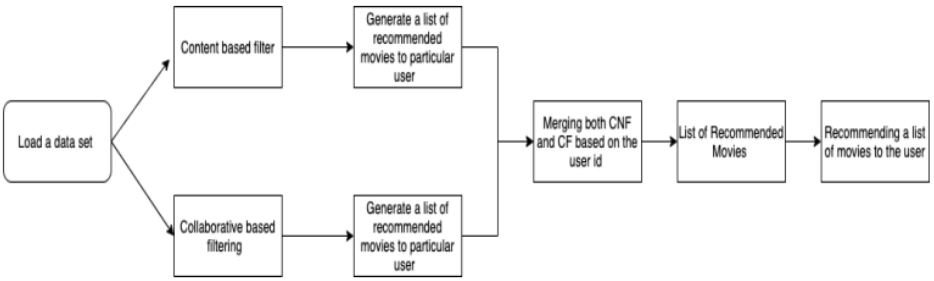


figure:3.5

Initially, it is best to load the data sets required to build a model.This project requires the use of the files movies.csv, rats.csv, and users.csv. Each data set can be found on the Kaggle.com website. This project's material essentially creates two models.

CHAPTER-4
EXPERIMENT AND RESULT ANALYSIS

4.1) EXPERIMENT

Initially, it is best to load the data sets required to build a model.This project requires the use of the files movies.csv, rats.csv, and users.csv. Each data set can be found on the Kaggle.com website. This project's material essentially creates two models.

4.1.1) Data Set

From the perspective of a recommender system, a movie can be described by a number of characteristics, including genres, actors, directors, and so on.

- **Director**: IMDb was used to obtain the director's information; while most films only have one director, some do have two or more.

- **Actors**: Large casts are common in films, however the vast majority of them are detrimental to the recommender system and have negative effects. As a result, there are just three notable actors in the film. They come from IMDb as well.

- **Keyword**: We use LSI to extract keywords from the Wikipedia plot with help from our pals at VionLabs.

- **Release Year**: The information is from IMDb and represents the year the movie was released.

4.1.2) Category

For a movie, we'll divide the films into 23 categories based on the common genres. Here is a list of the categories we used. Each movie-based document in the case is represented by one of the eight features specified in Section. The video is represented using a vector space model, and each feature of the document includes the word "movie."

Sci-Fi	Crime	Romance	Animation	Music
Comedy	War	Horror	Adventure	News
Biography	Thriller	Western	Mystery	Short
Drama	Action	Documentary	Musical	History
Family	Fantasy	Sport		

Table:4.1

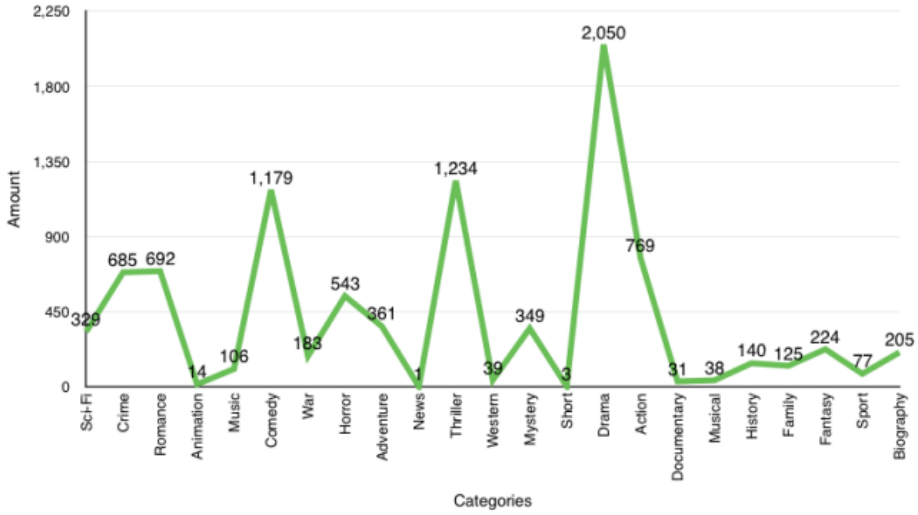
The genre is often used as a vector in other content-based recommender systems to estimate how similar two things are. This is only one element of the movie; there are many others, such as the actor and the setting. As a result, we add new features, some of which are really distinctive because they were found via our own research.

The reasons, though, are why we didn't just add features together to calculate TF-IDF. A natural trait that can be used to classify something is the genre.

4.1.3) Document

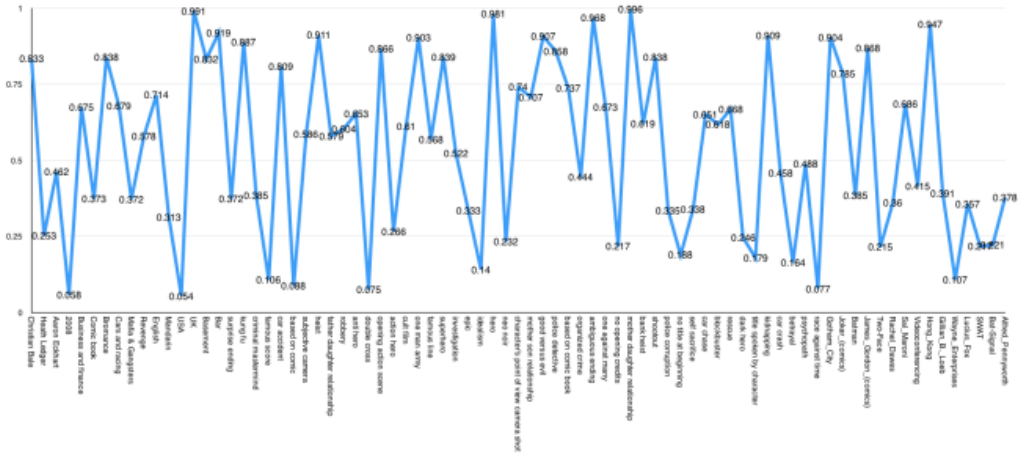
As we previously explained, the document in this instance is a video that has a lot of features. In the experiment, a vector space model will represent the movie. We described the characteristics that the movie is modelled on. The format of the vector space model is as follows:

Movie Model = **[Writers, Performers, Keywords, Year of Release, Vionel Themes, Languages, Places, Vionel Scenes]**



graph:4.1

The vector is typically quite long because there can be numerous directors and actors in a single movie. Here, we use the movie Avatar to show how the model works. Our TF-IIDF-DC calculated that The Dark Knight has 80 distinct features; this figure highlights the importance of each feature. From this vantage point, it is clear that a similar distribution of a film's features denotes a film's similarity to another



Graph:4.2

Directors	Christopher Nolan
Actors	Christian Bale Heath Ledger Aaron Eckhart
Keywords	Keywords
Release Year	2008
Vionel Themes	Business and finance Comic book Bromance Cars and racing Mafia & Gangsters Revenge
Language	English, Mandarin
Location	USA, UK
Vionel Scene	Basement, Bar

table:4.2

4.1.4) Result

Feature to cinema in this context refers to the phrase document. The video format of the vector space model, which may be used to assess similarity, can be easily converted. Thanks to preceding calculations, each movie in the database may be represented by a vector. The cosine similarity approach was then used to calculate how similar one movie is to the others.

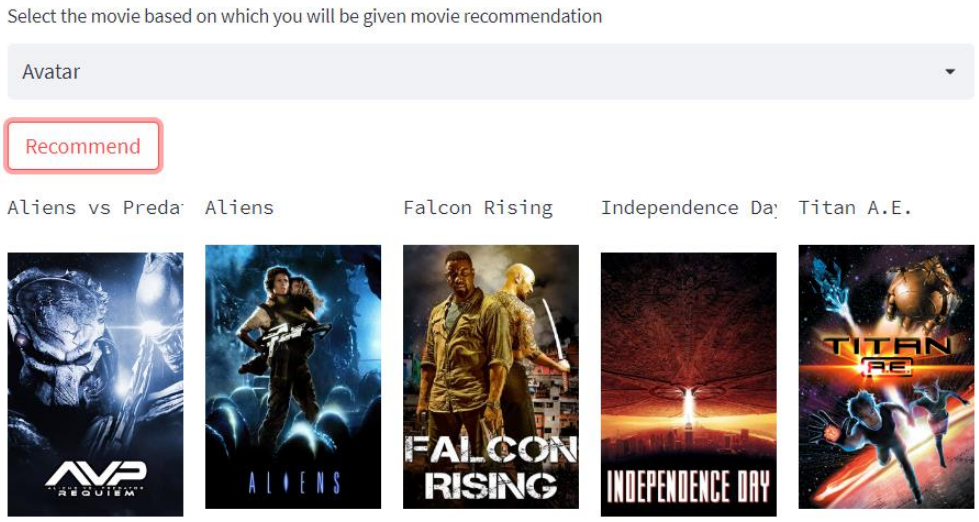


figure:4.1

4.2) Implementation

The System Make Use Different Algorithms and Methods for the implementation of Content Approach

4.2.1) Cosine Similarity:

The similarity of two non-zero vectors in an inner product space is measured by the cosine of the angle between them. Cosine similarity is a statistic that is

used to assess how similar two documents are, regardless of the size of the documents. The cosine of the angle made by two vectors projected onto a multidimensional space is computed. Even if the two comparable documents are separated by a significant Euclidean distance due to the size of the documents, the cosine similarity is advantageous since it enhances the possibility that the two comparable documents will be oriented closer together. As the angle gets smaller, the cosine similarity gets stronger.

$$Cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \ \sqrt{\sum_1^n b_i^2}}$$

equation:4

4.2.2) Singular Value Decomposition (SVD):

With singular vectors v1, v2,..., vr and corresponding singular values 1, 2,..., r, let A be a n*d matrix. The left singular vectors are then ui = (1/i)Avi, where i = 1, 2,..., r, and according to Theorem 1.5, A may be broken down into a sum of rank one matrices.

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u_i} \mathbf{v_i}^T.$$

equation:5

First, we provide a straightforward lemma that states two matrices A and B are equivalent if Av = Bv for all v. The lemma argues that a matrix A can be thought of as a transformation that translates vector v onto Av in the abstract.

4.2.3) Manhattan Distance:

The Manhattan distance metric measures the distance between two points as the sum of the absolute differences between their Cartesian coordinates. The sum of the discrepancies between the x- and y-coordinates can be used to express it.

When p1 and p2 are situated in a plane at (x1, y1) and (x2, y2), respectively,

$$|x_1 - x_2| + |y_1 - y_2|$$

equation:6

4.2.4) Euclidean Distance:

The Euclidean distance between two points in either flat or three-dimensional space determines the length of a segment connecting them. It is the method that best demonstrates a distance between two locations. How far apart two points are can be calculated using the Pythagorean Theorem.

Formula: The euclidean distance in two dimensions between the points (x1,y1) (x1,y1) and (x2,y2) (x2,y2).

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

equation:7

4.2.5) Jaccard Similarity:

The Jaccard index, sometimes referred to as Intersection over Union and the Jaccard similarity coefficient, is a statistic for evaluating the similarity and variety of sample sets. The Jaccard coefficient measures the similarity between finite sample sets by dividing the size of the intersection by the size of the union of the sample sets. The Jaccard index, often known as the Jaccard similarity coefficient, is a statistic for assessing the diversity and similarity of sample sets.

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

equation:8

Comparison Between Cosine and Manhattan distance

Cosine similarity or Euclidean distance can be used to calculate the distance between two vectors in a vector space. Because it's not always evident what we mean when we talk about the distance between two vectors, as we'll see in a moment, it's imperative that we be explicit about what we mean.
In a 2D vector space, three distinct points—blue, red, and green—are located. We could ask ourselves which pair or pairs of points are closer to one another. As we go, we expect that the answer will include a specific pair (or pairs) of points:

If just one combination is the closest, the answer can either be (red, green), (blue, red), or (blue, green). If two pairs are the closest, then three sets are possible; these sets match all two-element combinations of the three pairs.

There is only one set that could possibly contain all three couples if they are all equally near.This indicates that the closest set One of the seven potential sets is a pair or pairs of points. Then, how can we decide which of the seven potential solutions is correct? To do this, we must first choose a technique for gauging distances. Using a ruler and two points, we may measure the reading to determine which response is correct.If we do this for all possible pairs, we can generate a list of measurements for pair-wise distances. The table can then be sorted ascendingly to reveal the pairwise pairing of points with the least distances.

In this instance, the pair (red, green) makes up the set with the shortest distance. Thus, we can claim that the shortest Euclidean distance between the red and green points in our collection is the distance measured by a ruler between them. We can also use an entirely different but equally acceptable method to get the distances between the identical places. Let's imagine that we are looking at the points from within the plane, specifically from its inception, as opposed to the top of the plane or from a bird's eye view. This allows us to depict with an arrow the direction that we consider when analysing each point: Regardless of how far apart the points. From our perspective point, it doesn't really matter how far the points are from the origin. Actually, without leaving the plane and entering the third dimension, we are unable to understand that. When viewed from the origin, all of the points appear to be on the same horizon; their only difference is the path they take in relation to a reference axis.

Choosing the metric to employ relies on the specific activity that needs to be completed:

Both metrics are helpful for various tasks, such preliminary data analysis, because they each make it possible to glean particular insights about the structure of the data. Euclidean distances usually function better when applied to others, such as text classification.

The retrieval of the texts that are most similar to a given document is one example of a more comprehensive application where cosine similarity performs better. The challenge is in comprehending all methods and learning the heuristics associated with their use, as is frequently the case with machine learning. One discovers this by trial and error.

4.3) Method Analysis

Recommender systems have grown in popularity as a research topic in recent years. Many academics have offered a variety of different recommendation tactics. The most well-known classification of these tactics is

- Content-based recommendations
- Collaborative filtering is advised.
- Recommendation-for hybrids.

4.3.1) Content Based Recommendation System

Based on parameters for movies like genre, director, description, actors, etc., it provides recommendations for users. A user might love a movie or television show similar to one they already enjoyed, according to the logic behind this type of suggestion system.

Many recommender systems begin by modelling the item with keywords. But extracting keywords from a piece of content can be difficult, especially in the media sector where it can be difficult to extract text keywords from videos. There are primarily two methods for resolving this kind of problem. In the first, users can tag the items, while in the second, experts are involved. Jinni and Pandora, respectively, are the exemplary experttagged systems for music

and movies. As an example, consider Jinni, whose researchers identified over 900 tags as "movie genes" and permitted movie industry pros to generate tags for them. A number of criteria, including "movie genre," "story," "time," "place," and "cast," apply to these keywords.

ADVANTAGES	DISADVANTAGES
• Since the recommendations are particular to this person, the model doesn't require any information about other users. This makes scaling to a huge user base simpler.	• This technique needs a lot of domain knowledge because the feature representation of the items is somewhat hand-engineered. The quality of the model is therefore limited to the hand-engineered elements.
• The model may identify a user's precise preferences and recommend specialised products that only a small number of other users are likely to be interested in..	• Only recommendations based on the user's current interests can be made by the model. In other words, the model has little capacity to further develop the interests of the users..

Table:4.3

Figure:4.2

4.3.2) Collaborative Recommendation system

It matches people with similar interests and gives recommendations based on their preferences. Sam and Robin, two examples, who favour the movie A, B, C, and D, respectively. Sam would recommend the films A and B to Robin because C and D are also favourites of Sam's. Collaborative filtering does not use metadata to produce suggestions.

I will be focusing on content-based recommendation systems for this project since I think that using metadata like "genres," "actor," and "overview/plot" will provide us a lot of insight on understanding users' interests and help recommend films or TV episodes in line with this. It facilitates the discovery of user preferences.

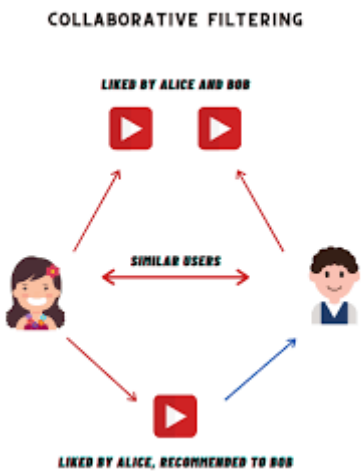


Figure:4.3

4.3.2.1) User-based collaborative filtering:

It is presumed that a user will like things that other users with similar tastes will also like. Thus, the first step in user-based collaborative filtering is identifying users with similar likes. When users favour related things, this is referred to as collaborative filtering. In other words, given user u and user v, N(u) and N(v) are the things set liked by u and v, respectively. As a result, it is simple to determine how similar u and v are:

User/Item	Item A	Item B	Item C
User A	✓		✓
User B	✓	✓	✓
User C	✓		recommend

table:4.4

Imagine that we wish to give our friend Stanley a movie recommendation. We could infer that people who are similar will have similar tastes.

4.3.2.2) Item-**based collaborative filtering**:

It is presumed that a user will like things that other users with similar tastes will also like. Thus, the first step in user-based collaborative filtering is identifying users with similar likes. When users favour related things, this is referred to as collaborative filtering. In other words, given user u and user v, N(u) and N(v) are the things set liked by u and v, respectively. Thus, it is simple to summarise how similar u and v are.

An illustration of an item-based CF recommendation is the table. We may infer that Item A and Item C are comparable since those who like Item A also like Item C, according to the interest history of all the users for Item A. Since User C enjoys Item A, it stands to reason that she could also enjoy Item C.

User/Item	Item A	Item B	Item C
User A	✓		✓
User B	✓	✓	✓
User C	✓		recommend

Table:4.5

Collaborative filtering with user and item based criteria was seen. In the first, the emphasis is on populating a user-item matrix and making recommendations based on users who are more like the active user. IB-CF, on the other hand, fills out a matrix of related objects and makes recommendations.

Although it is challenging to cover all of these topics succinctly, doing so is the first step in learning more about **RecSys**

Advantages	Disadvantages
Due to the fact that the embeddings are automatically taught, we don't require domain expertise.	The prediction of the model for a certain (user, item) pair is represented by the dot product of the related embeddings. As a result, the system cannot embed or use an item to query the model if it is not detected during training.
The model's users might discover new hobbies. The machine learning algorithm may not be aware of the user's interest in a certain item in a catalogue, but the model may nevertheless recommend it because there may be other users who have the same interest.	By averaging the embeddings of objects from the same category, from the same uploader (in YouTube), and so forth, the system can approximate its embedding if it lacks interactions.
The system may, to a certain extent, train a matrix factorization model solely on the feedback matrix. The system doesn't specifically need contextual characteristics. Actually, it is possible to use any of a number of candidate generators.	Side features are any features that extend past the query or item ID. The user's age or country may be side variables for movie suggestions. The model's quality increases when accessible side features are added.

Table:4.6

4.3.3) Hybrid Recommender System

The popularity of hybrid recommender systems is growing right now. According to recent studies, collaborative filtering and content-based filtering can work better together. Hybrid recommender systems can be implemented in a variety of ways: easily add CF capability to a CB technique and aggregate the results of CF and CB recommendations.

There are seven ways to hybridise:

- **Weighted** : Add the results of the various recommender component scores.
- **Switching** : Select ways by alternating between various recommender components.
- **Mixed** : Display the outcomes of multiple systems' recommendations.
- **Combining features** : Taking features from several sources and combining them into one input.
- **Feature Augmentation** : Compute features using a single recommender and go on to the next stage using the results.
- **Cascade** : Use a recommender technique to generate a rough result, then recommend it on top of the prior result.
- **Meta-level** : Input another recommender approach with the model produced by one recommender.

Each approach has advantages and disadvantages, and the results change based on the dataset. The approach might not be suitable for all problems because of the algorithm's inherent constraints. For instance, it is difficult to automate feature extraction from media data using a content-based filtering strategy. Additionally, the diversity is not as good because the recommendation only contains products that the customer has already selected.

Recommending to users who never make decisions is exceedingly difficult. Collaborative filtering techniques somewhat minimise the previously mentioned drawback.

However, because CF relies so much on past data, there are issues with cold starts and sparsity. Due to cold start challenges that involve both new user

and new item issues, collaborative filtering, which is based on the similarity between the things selected by users, finds it challenging to recommend a new item that has never been recommended before.

4.4) Output at various stages

- We obtained our dataset from Kaggle, which also contains the 5000 films listed on IMDb and IMDb. I've shown the dataset we're utilising in its initial form in the image below. I demonstrated it using Python's pandas package.From the perspective of a recommender system, a movie can be described by a number of characteristics, including genres, actors, directors, and so on.
- **Director** : The director's information is taken from IMDb; most films only have one director, but others have two or more.
- **Actors/cast**: Large casts are common in films, however the vast majority of them are detrimental to the recommender system and have negative effects. As a result, there are just three notable actors in the film..
- **Keyword**: We use LSI to extract keywords from the Wikipedia plot with help from our pals at VionLabs.
- **Release Year**: This is the film's release year, and the information comes from IMDb.
- **Genres** : This shows the type of film comedy , thriller , keywords etc.

```
In [4]: movies = pd.read_csv("tmdb_5000_movies.csv")
        credits = pd.read_csv("tmdb_5000_credits.csv")

In [5]: data = movies.merge(credits,on='title')

In [6]: data.head()
        data2 = data[['movie_id','title','overview','genres','keywords','cast','crew']]

In [7]: data2.head()
```

Out[7]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

```
In [8]: data2.isnull().sum()
```

Out[8]:

```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
```

Figure:4.4

- **Category :**

For a movie, we'll divide the films into 23 categories based on the common genres. Here is a list of the categories we used. Each movie-based document in the case is represented by one of the eight features specified in Section. The video is represented using a vector space model, and each feature of the document includes the word "movie."

Sci-Fi	Crime	Romance	Animation	Music
Comedy	War	Horror	Adventure	News
Biography	Thriller	Western	Mystery	Short
Drama	Action	Documentary	Musical	History
Family	Fantasy	Sport		

Table:4.7

In [22]:
```
data2['tags'] = data2['overview']+data2['genres']+data2['keywords']+data2['cast']+data2['crew']
```

```
C:\Users\raggh\AppData\Local\Temp/ipykernel_21344/2983703482.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data2['tags'] = data2['overview']+data2['genres']+data2['keywords']+data2['cast']+data2['crew']
```

In [23]:
```
data2.head()
```

Out[23]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

In [24]:
```
data3 = data2[['movie_id','title','tags']]
```

In [25]:
```
data3.head()
```

Out[25]:

	movie_id	title	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...

In [26]:
```
data3['tags'] = data3['tags'].apply(lambda x: [" ".join(x)])
```

```
C:\Users\raggh\AppData\Local\Temp/ipykernel_21344/630522310.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data3['tags'] = data3['tags'].apply(lambda x: [" ".join(x)])
```

In [27]:
```
data3['tags'][0]
```

Out[27]:
```
['In the 22nd century, a paraplegic Marine is dispatched to the moon Pandora on a unique mission, but becomes torn between following orders and protecting an alien civilization. Action Adventure Fantasy ScienceFiction cultureclash future spacewar spacecolony society spacetravel futuristic romance space alien tribe alienplanet cgi marine soldier battle loveaffair antiwar powerrelations mindandsoul 3d SamWorthington ZoeSaldana SigourneyWeaver JamesCameron']
```

Figure:4.5

Figure:4.6

```
In [35]: vectors.shape
Out[35]: (4806, 5000)

In [36]: from sklearn.metrics.pairwise import cosine_similarity

In [37]: similarity = cosine_similarity(vectors)

In [38]: def recommend(name):
             index = data3[data3['title']==name].index[0]
             similrty = similarity[index]
             movie_list = sorted(list(enumerate(similrty)),reverse=True,key=lambda x:x[1])[1:6]
             for i in movie_list:
                 print(data3.iloc[i[0]].title)
             return

In [39]: recommend('The Dark Knight')
         The Dark Knight Rises
         Batman Begins
         Batman Returns
         Batman Forever
         Batman

In [40]: import pickle

In [43]: pickle.dump(data3.to_dict(),open("data3.pkl","wb"))
         pickle.dump(similarity,open("similarity.pkl","wb"))

In [ ]:
```

Figure:4.7

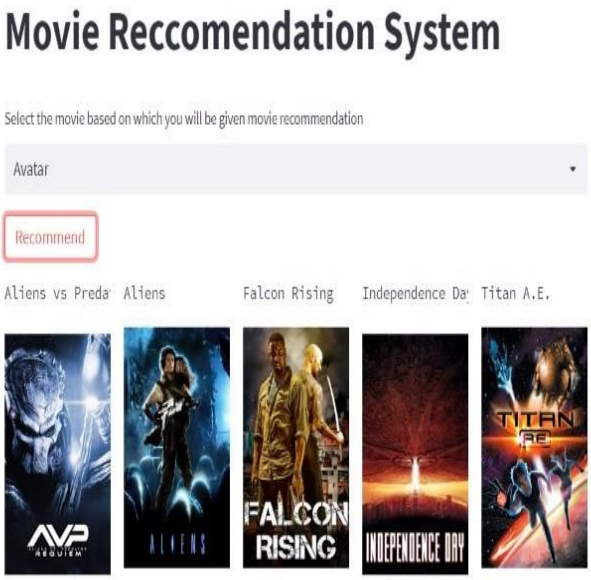


Figure:4.8

CHAPTER-5
CONCLUSIONS

5.1) Conclusions

In this project, content-based filtering and collaborative filtering are combined using Singular Value Decomposition (SVD) as a classifier and Cosine Similarity as the recommended approach. This tactic seeks to improve the effectiveness, superiority, and scalability of movie recommendation systems. On three separate movie datasets, existing pure algorithms and the hybrid technique are applied, and the outcomes are compared. Comparative results demonstrate that the suggested approach outperforms pure alternatives in terms of the accuracy, quality, and scalability of the movie recommendation system. The proposed solution is quicker when comparing the computation times of the three pure techniques.

These results suggest that a range of classifiers may be used with comparable success rates and that machine learning techniques can be highly effective at spotting bogus news. The models must be evaluated using different metrics, such as precision, recall, and F1-score, in addition to factors like interpretability, scalability, and computing requirements. It is important to keep in mind that accuracy is only one statistic. It may also be useful to look into different feature extraction and selection methods, classifier types, and ensemble techniques to see if even better results can be obtained.

In order to address the concerns we mentioned at the onset, we first use a content-based recommender algorithm, thus there is no cold start issue. all of the functions of our recommendation engine. Some of them are more

diversified and precise than others because they originate from various research departments within the organisation. Then, the cosine similarity was introduced, which is frequently used in industry. To improve the movie's representation for the weight of features, we introduced TF-IIDF-DC.

5.2) Future Scope

Movie genres have been included in the suggested strategy, but in the future, we need also consider user age because movie preferences fluctuate with age. For example, while we are young, we often like animated films over other genres. Future versions of the proposed solution should have lower memory requirements. Here, only different movie datasets have been used to apply the suggested methodology. The performance can be calculated in the future and applied to the Netflix and Film Affinity databases.

.

I will consider the following aspects in future work:

1. **Use collaborative filtering to make recommendations.**
   Once there is enough user data, recommendations for collaborative filtering will be introduced. As we discussed in, collaborative filtering is dependent on user social information, and future studies will analyse this data.

2. **Include more pertinent and accurate movie features.**
   Common collaborative filtering recommendations replace object features with the rating. In the future, we should extract information from films that can provide a more accurate description of the film, such as colour and subtitles.

3. **Present the user's dislike list of films.**

   User data is always useful for recommender systems. We'll keep compiling user data and add a list of films that people don't like. We will also enter a list of films we detest into the recommender system in order to create scores that will be added to the prior result. By doing this, we can improve the recommender system's functionality.

4. **Explain machine learning**

   Recommender systems with dynamic parameters will be studied later. Machine learning will be used to choose the best weights and automatically change the weights of each feature.

5. **Make the recommender system a part of the company.**

   In the future, the recommender system won't be a test-only external website. We'll develop a programmers-only internal API. Some movie listings on the internet will be sorted according to user reviews.

5.3) Application

Filtering and predicting only the movies that a matching user is most likely to wish to see is the main objective of movie recommendation systems. The user information from the system's database is used by the ML algorithms for these recommendation systems.

Since the recommendations are particular to this person, the model doesn't require any information about other users. This makes scaling to a huge user base simpler. The programme may identify a user's individual preferences and offer specialised products in which only a small percentage of other users are also interested.

According to the user's past behaviour or explicit feedback, content-based filtering uses item features to suggest additional goods that are similar to what they already enjoy.

hence, the applications of movie recommendation systems is to identify, filter, and forecast the movies that a given user is most likely to find interesting. The user information from the system's database is used by the ML algorithms for these recommendation systems.

REFERENCES

[1] Hirdesh Shivhare, Anshul Gupta and Shalki Sharma (2015), "Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure", IEEE International Conference on Computer, Communication and Control.

[2] Manoj Kumar, D.K. Yadav, Ankur Singh and Vijay Kr. Gupta (2015), "A Movie Recommender System: MOVREC", International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3.

[3] RyuRi Kim, Ye Jeong Kwak, HyeonJeong Mo, Mucheol Kim, Seungmin Rho,Ka Lok Man, Woon Kian Chong (2015),"Trustworthy Movie Recommender System with Correct Assessment and Emotion Evaluation", Proceedings of the International MultiConference of Engineers and Computer Scientists Vol II

[4] Zan Wang, Xue Yu*, Nan Feng, Zhenhua Wang (2014), "An Improved Collaborative Movie Recommendation System using Computational Intelligence",Journal of Visual Languages & Computing,Volume 25, Issue 6.

[5] Debadrita Roy, Arnab Kundu, (2013), "Design of Movie Recommendation System by Means of Collaborative Filtering", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4.