# Films And Series Recommendation System

**Project Report**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**Computer Science (Mobile Computing) Submitted**

**by:**

**BISWAJEET MUDULI**

**19BCS4323**

**HARDIK CHAUHAN**

**19BCS4325**

**JACOB VARGHESE**

**19BCS4331**

**Under the Supervision of:**

**YOGIRAJ ANIL BHALE**


**CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB**

# ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of CHANDIGARH UNIVERSITY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Mr. Yogiraj Anil Bhale **M.E., Professor**, Minor Project for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project colleagues**, Jacob and Biswajeet** for their valuable support, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

# ABSTRACT

A recommendation system plays a crucial role in tailoring content suggestions to a user's preferences, finding applications in various domains like movies, music, news, and e- commerce. By analyzing user data and item relationships, these systems aim to predict what a user might like, thus enhancing their experience. For instance, Amazon and Netflix use recommendation systems to suggest products and movies.

Popular music streaming platforms like Spotify and Deezer also employ recommendation systems to offer personalized music recommendations that evolve over time. These systems often use collaborative filtering algorithms, focusing on user-item relationships, to make these recommendations. They are typically developed using Python and data mapping and correlation techniques, ensuring users receive content tailored to their preferences based on past interactions.

In this context, the recommendation system relies on accumulating a dataset containing user identifiers, ratings, item identification, and time spent. By employing mapping techniques and correlation concepts, it creates a linkage between user IDs and their respective ratings. For instance, if a user rates a series of movies positively, the system takes these ratings into account to suggest new movies that align with the user's established preferences. Over time, the system adapts to the evolving tastes of the user, providing a dynamic and personalized experience.
These recommendation systems, powered by collaborative filtering, represent a synergy of user interactions and data analysis, making them a fundamental part of modern content delivery platforms.

## TABLE OF CONTENTS

Chapter 1

# INTRODUCTION

Recommendation systems are tools used to engage customers by understanding their preferences. These systems have gained widespread recognition due to their ability to deliver personalized content that aligns with a customer's interests. In today's world, online retail platforms offer a vast array of products, making it challenging for customers to make informed choices. Recommendation systems step in to rapidly suggest the most suitable items, assisting users in finding and selecting products, such as books, movies, or restaurants, from the overwhelming online options.

Imagine these recommendation systems as your personal shopping assistant, sifting through an extensive product catalog to offer you a curated selection based on your stated preferences. Specifically, in the case of a movie recommendation system, it enhances convenience and personalization, helping users engage with the system and discover films tailored to their tastes. Our decision to focus on a movie recommendation system for our BE Project primarily stems from our desire to provide users with a comfortable and personalized movie-watching experience.

The primary objective of our system is to recommend movies to users based on their viewing history and ratings they've provided. Additionally, our system can facilitate collaborations with various e-commerce companies interested in targeting specific customer segments based on their movie genre preferences. Personalized recommendation engines play a crucial role in helping a diverse audience navigate the vast world of potential movie choices to find those that resonate with their unique tastes. Two prominent methods employed for offering recommendations are collaborative filtering and content-based filtering, each having its strengths and weaknesses. In our paper, we propose an integrated approach that combines both methods to enhance the overall performance and accuracy of our recommendation system.

### 1.1 RELATED WORK

In recent years, film recommendation systems have been a subject of considerable focus, with various techniques coming into play. These methods encompass systems employing the ALS algorithm, coefficient-based recommendations, and collaborative filtering based on item similarity. These approaches rely heavily on prior user-generated movie ratings for assessment. However, it's worth noting that these systems still exhibit room for improvement, and ongoing research aims to enhance their overall performance.

An example of such an initiative is the design and implementation of a collaborative filtering

approach utilizing the K-nearest neighbors (KNN) method, as detailed by Cui, Bei-Bei [2]. This system makes movie recommendations by assessing ratings and similarities between two users. It then further divides the movie dataset into rated and unrated test sets using the KNN model. This method not only suggests films to users based on their browsing history but also generates novel and unconventional movie recommendations, taking into account a movie's history and rating. The dataset used in this approach is housed within a MYSQL database.

The user's interaction with the system captures their external and internal behavior traits, which are stored in the user database through a login module. The diagram below, Figure 1, outlines the effective methodology for collaborative filtering using KNN. Comparing this approach to other algorithms, Goutham Miryala conducted a similar ALS-based analysis [4]. Interestingly, the results demonstrated that using a more extensive training dataset split of 80% for training and 20% for testing led to a model with a lower Root Mean Square Error (RMSE) when compared to the 60-40 dataset split. These findings imply that higher regularization parameters increase RMSE, while the reverse is true for ALS. Additionally, when ALS was compared to other algorithms such as Singular Value Decomposition (SVD), KNN, and Normal Indicator, it consistently outperformed them in the context of recommendation systems.

### 1.1 EXISTING SYSTEM

The two most prominent types of recommendation systems are content-based and collaborative filtering systems. Collaborative filtering, in particular, relies on the collective behavior of a group of users to provide recommendations to other users. This approach leverages the preferences of multiple users to make suggestions. For instance, a simple example might involve recommending a movie to a user because their friend enjoyed it. Collaborative filtering can be categorized into two main models: Memory-based and Model-based techniques.

Memory-based methods have the advantage of being straightforward to implement, and their recommendations are typically easy to explain. They come in two forms: User-based collaborative filtering, where items are recommended to a user based on the preferences of users similar to them, and Item-based collaborative filtering, which identifies similar items based on past user ratings. For example, if Derrick and Dennis have a similar taste in movies and Derrick liked a certain film, the system might recommend that film to Dennis because their preferences align.

Model-based approaches, on the other hand, are rooted in Matrix Factorization and are better suited to handling data sparsity. These systems employ data mining and AI algorithms to predict how users would rate unrated items. Techniques like matrix factorization and dimensionality reduction are employed to enhance accuracy. Model-based methods encompass decision trees, rule-based models, Bayesian models, and latent factor models.

Content-based systems recommend items based on their attributes, such as genre, director, actor, or artist. For example, if a user enjoyed a movie featuring Vin Diesel, a content-based system might suggest "Fast & Furious" because of Vin Diesel's involvement. Similarly, for music recommendations, the system may suggest songs by artists the user has previously shown interest in. Content-based systems operate under the assumption that if you liked one specific item, you're likely to enjoy something similar.

### DISADVANTAGES

- It does not work for one more shopper UN agency has not appraised any issue nevertheless as enough appraisals square measure needed substance based mostly recommendation assesses the shopper inclinations and provides actual proposals. Complex interface
- No suggestion of lucky things.
- Limited Content Analysis-The recommendation does not work if the framework neglects to acknowledge the items cap a shopper likes from the items that he does not look after.

## 1.2 PROPOSED SYSTEM

Collaborative filtering (CF) emerges as a dominant and effective recommendation strategy, surpassing many content-based methods dependent on user and item attributes. Unlike its counterparts, CF techniques derive predictions solely from user-item interactions, excelling at uncovering latent connections between users and items. This not only facilitates unexpected recommendations but also contributes to diversifying the overall recommendation landscape. In our contemporary data-rich landscape, recommendation systems play an indispensable role, particularly on the internet, where an abundance of content is accessible. These systems, employing knowledge discovery techniques, extend beyond traditional methods, incorporating sentiment analysis to provide personalized recommendations. By gauging user sentiments through their interactions, these systems offer a holistic understanding of user preferences, guiding them through vast collections of articles, movies, music, web pages, and more. Noteworthy instances include Amazon's adept product recommendations, Last.fm's nuanced music suggestions, and MovieLens' insightful movie recommendations, showcasing the synergy of collaborative filtering and sentiment analysis in refining the user experience.

**Sentiment Analysis of IMDb Reviews:**

In the core of our Films and Series Recommendation System lies a sophisticated sentiment analysis module that delves into IMDb reviews, extracting valuable insights from user sentiments. Leveraging Natural Language Processing (NLP) techniques, the system systematically evaluates the textual content of reviews, discerning sentiments as positive, negative, or neutral. This process allows for a nuanced understanding of audience reactions, enabling the system to recommend content that aligns with individual preferences. By pinpointing the sentiment behind each review, the system not only refines its recommendation accuracy but also captures the diverse spectrum of user opinions. The integration of sentiment analysis serves as a powerful tool, shedding light on the overall reception of films and series within the IMDb community. This comprehensive approach not only enhances the personalization of recommendations but also offers a deeper understanding of the dynamic interplay between user sentiment and content preference.

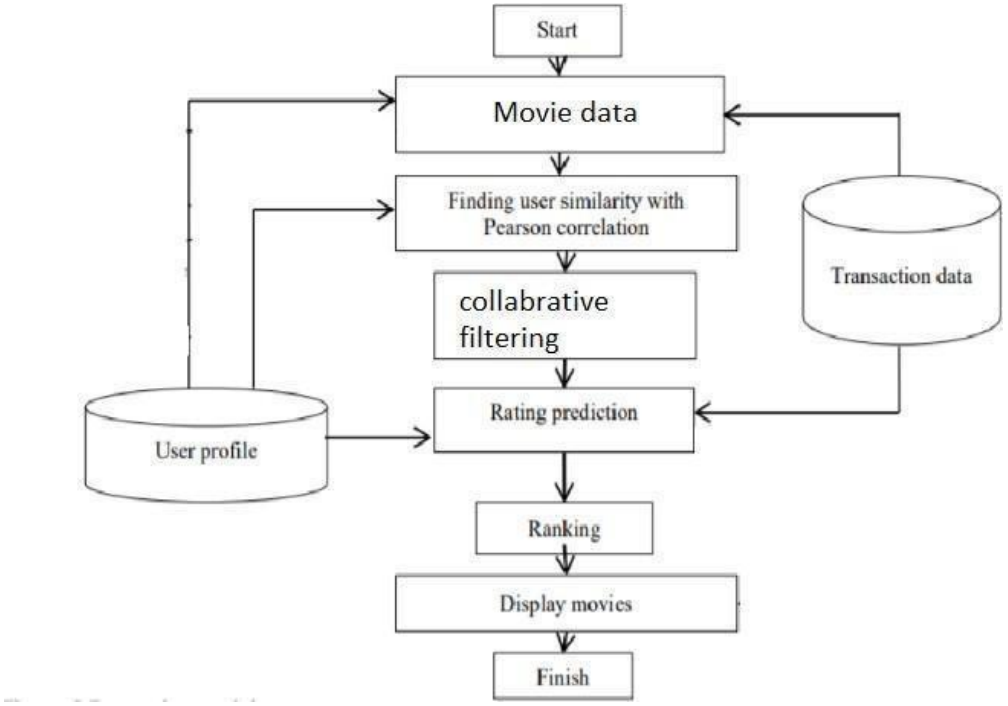



**FIG 1 SYSTEM ARCHITECTURE**

## ADVANTAGES OF THE PROPOSED SYSTEM

- It is subject to the association between shoppers that suggests that it's content autonomous. Scalableclient administrations.
- CF recommendation frameworks will propose lucky things by noticing comparative leaning individuals' conduct.
- They will create real quality analysis of things by considering completely different folks teams insight.

## CHAPTER 2

## LITERATURE SURVEY

The movie recommendation system employs a collaborative filtering approach, leveraging user-provided information to suggest movies. These recommendations are based on the analysis of user data, prioritizing movies with the highest ratings. Luis M. Capos et al. conducted an analysis of two traditional recommendation systems: content-based filtering and collaborative filtering, each having its own limitations. In response, they proposed a novel system that combines Bayesian networks with collaborative filtering to overcome these drawbacks. Harpreet Kaur et al. introduced a hybrid system that incorporates both content and collaborative filtering algorithms, considering the context of the movies during recommendations. They take into account user-user and user-item relationships in their recommendation process.

Utkarsh Gupta et al. employ a technique using chameleon to cluster user-specific and item-specific information efficiently, relying on hierarchical clustering for recommendation systems. A voting system is used to predict item ratings, resulting in lower error rates and improved clustering of similar items. Urszula Kużelewska et al. suggested using clustering as an approach to address recommendation systems. They presented and evaluated two methods for computing cluster representatives, enhancing recommendation accuracy compared to traditional centroid-based methods. Costin-Gabriel Chiru et al. introduced a movie recommendation system that utilizes user information to provide movie suggestions. This system tackles the challenge of generating unique recommendations by considering the user's psychological profile, viewing history, and movie scores from other websites. It combines content-based filtering and collaborative filtering, employing an aggregate similarity calculation.

For predicting the difficulty level of each case for trainees, Hongli Lin et al. proposed a method called content-boosted collaborative filtering (CBCF). This algorithm consists of two stages: content-based filtering to improve trainee case ratings data and collaborative filtering to provide final predictions. CBCF combines the strengths of both CBF and CF while addressing their individual limitations..

# CHAPTER 3

# METHODOLOGY

## 3.1 AIM OF THE PROJECT

To implement a recommendation for movies, based on the content of providing the most relevant information to a user by discovering patterns in a dataset. The algorithm rates the items and shows the userthe items that they would rate highly.

## 3.2 SYSTEM REQUIREMENTS

### 3.2.1 SOFTWARE REQUIREMENTS

- Operating system : Windows 7 and above (64-bit).
- Python : 3.6

### 3.2.2 HARDWARE REQUIREMENTS

- Hard disk : 80GB or more
- Ram : 70Mb or more
- Processor : : Intel Core Duo 2.0 GHz or more

## 3.3 OVERVIEW OF THE PLATFORM

### 3.3.1 Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed foremphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

**What can Python do?**

- Python can be used on a server to create web applications.

- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.

- Python can be used for rapid prototyping, or for production-ready software development.

**Why Python?**

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written.This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way

**Good to know**

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is stillquite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which containeda few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updatesfor both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of 9 January 1, 2020 has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is

still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

- It is possible to write Python in an Integrated Development Environment,such as Thonny, Pycharm,Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

**Python Syntax compared to other programming languages**

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation,using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose. Python is Interpreted Many languages are compiled, meaning the source code you create needs to be translated into machinecode, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, withoutthe intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiledinto the native language of the computer processor tend to run more quickly than interpreted programs. For some 10 applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediencyof coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming,and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability wellbeyond what is built into the language, such as database manipulation or GUI programming.

- Python accomplishes what many programming languages don't: the language itself is simplydesigned, but it is very versatile in terms of what you can accomplish with it.

### 3.3.2 Collaborative Filtering

Collaborative filtering, a fundamental technique in recommendation systems, can be understood in both a narrow and a broader context.

In its more specific interpretation, collaborative filtering refers to the method of automatically predicting a user's preferences by gathering information on their tastes from a collaborative group of users. The underlying assumption here is that if person A shares similar opinions with person B on one topic, it is likely they will have comparable preferences on other topics when compared to a randomly chosen individual. For example, in a television recommendation system based on collaborative filtering, predictions are made about which TV shows a user might enjoy, using data from a group of users. These predictions are personalized but draw insights from the collective user data. This approach differs from a simpler method of assigning generic scores to items of interest based on factors like the number of votes they receive.

In the broader sense, collaborative filtering involves the process of filtering information or identifying patterns through cooperation among multiple agents, perspectives, and data sources. Collaborative filtering applications typically deal with extensive datasets, spanning various domains, including data from sensing and monitoring activities (e.g., mineral exploration, large-scale environmental sensing), financial data (such as integrating diverse financial sources), and user-related data in electronic commerce and web applications. This discussion primarily focuses on collaborative filtering in the context of user data, although some methods and principles may also be applicable to other domains.

The rapid expansion of the internet has made it increasingly challenging to efficiently extract valuable insights from the vast pool of online information. The sheer volume of data necessitates effective mechanisms for information filtering, and collaborative filtering is one of the essential techniques employed to address this challenge.

Collaborative filtering is primarily motivated by the belief that people often receive the most relevant recommendations from individuals with similar tastes. It encompasses a range of methods aimed at matching users with akin interests, forming the basis for making personalized recommendations.

The successful implementation of collaborative filtering algorithms generally hinges on several factors:
(1) active user engagement
(2) user interest representation in an accessible format, and
(3) the deployment of algorithms capable of identifying individuals with shared interests.

Typically, the workflow of a collaborative filtering system is:

1. A user expresses his or her preferences by rating items (e.g. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item).

A key problem of collaborative filtering is how to combine and weight the preferences of user neighbors. Sometimes, users can immediately rate the recommended items. As a result, the system gains an increasingly accurate representation of user preferences over time.

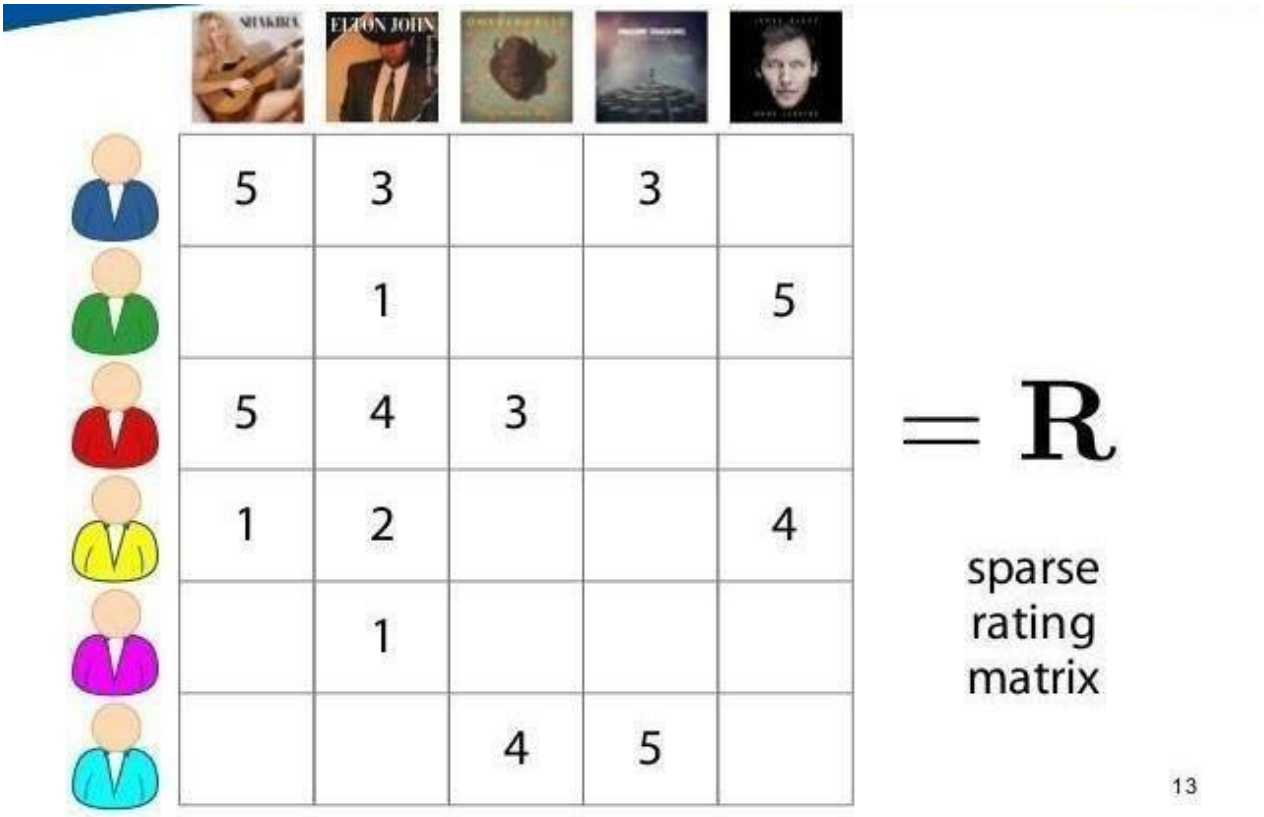



FIG 2 COLLABORATIVE FILTERING (CF)

### 3.3.3 USER BASED FILTERING

Imagine we want to suggest a movie to our friend Stanley. Our approach is based on the idea that people who are alike tend to have similar preferences. If Stanley and I have watched many of the same movies and rated them almost identically, but he hasn't seen 'The Godfather: Part II' while I have and loved it, it's reasonable to assume he might also enjoy it. In essence, we've generated a simulated rating for Stanley based on our shared tastes.

User-Based Collaborative Filtering (UB-CF) operates on this principle. It offers recommendations by identifying users who are similar to the one we want to suggest a movie to. A specific application of this is the user-based nearest neighbor algorithm, which involves two key tasks:

In essence, we construct a User-Item Matrix and predict how the active user might rate

movies they haven't watched yet, relying on the preferences of other similar users. This method is memory-based in nature.

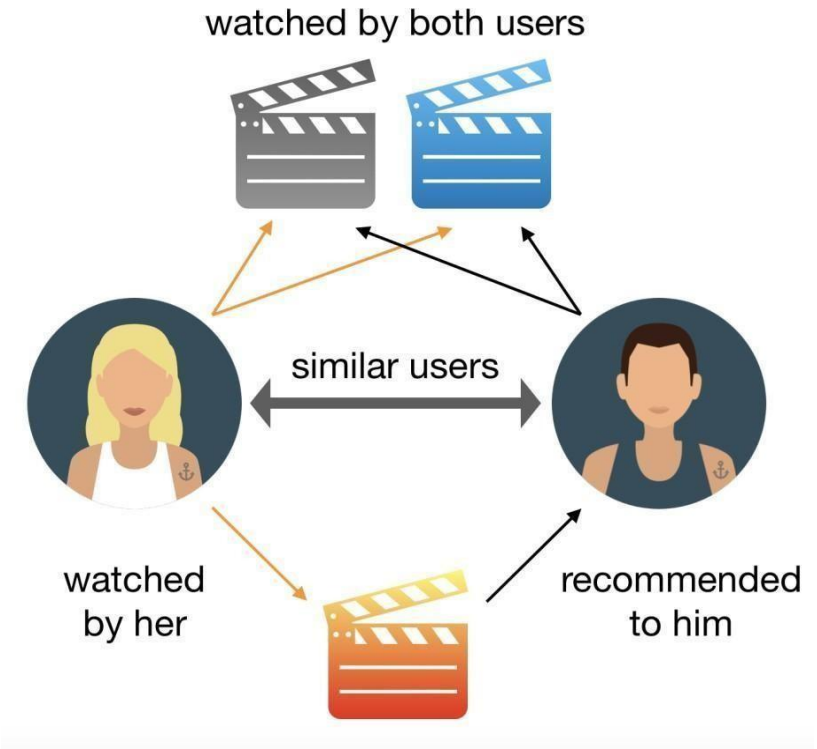



**FIG 3 USER BASED FILTERING**

### 3.3.4 KNN ALGORITHM

The ***K*-nearest neighbors algorithm** (***K*-NN**) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover.It is used for Classification and regression. In both cases, the input consists of the *k* closest training examples in data set. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a pluralityvote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In *k-NN regression*, the output is the property value for the object. This value is the averageof the values of *k* nearest neighbors.

*K-NN* is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data canimprove its accuracy dramatically.

In both classification and regression tasks, a valuable approach is to apply weights to the inputs from neighboring data points, with closer neighbors having a stronger influence on the outcome than those farther away. Typically, a common approach is to assign a weight of 1 divided by the distance (1/d) to each neighbor.

These neighbors are selected from a pool of objects for which we already have knowledge of their class (in the case of k-NN classification) or the property value of interest (for k-NN regression). This set can be viewed as the algorithm's training data, although there's no distinct training phase involved.

1. Find the K-nearest neighbors (KNN) to the user ***a,*** using a similarity function ***w*** to measure thedistance between each pair of users:

$$Similarity(a, i) = w(a, i), i \in K$$

2. Predict the rating that user ***a*** will give to all items the ***k*** neighbors have consumed but ***a*** has not. Welook for the item ***j*** with the best predicted rating.

# CHAPTER 4

# MODULE DESCRIPTION

## 4.1 SYSTEM STUDY

A recommendation engine is a system that suggests products, services, information to users based onanalysis of data. Notwithstanding, the recommendation can derive from a variety of factors such as the history of the user and the behaviour of similar users.

Recommendation systems are quickly becoming the primary way for users to expose to the whole digital world through the lens of their experiences, behaviours, preferences and interests. And in a world of information density and product overload, a recommendation engine provides an efficient way for companiesto provide consumers with personalised information and solutions.

### 4.1.1 BENEFITS

A recommendation engine can significantly boost revenues, Click-Through Rates (CTRs), conversions, andother essential metrics. It can have positive effects on the user experience, thus translating to higher customer satisfaction and retention.

Let’s take Netflix as an example. Instead of having to browse through thousands of box sets and movie titles, Netflix presents you with a much narrower selection of items that you are likely to enjoy. This capability savesyou time and delivers a better user experience. With this function, Netflix achieved lower cancellation rates, saving the company around a billion dollars a year.

Although recommendation systems have been used for almost 20 years by companies like Amazon, it hasbeen proliferated to other industries such as finance and travel during the last few years.

### 4.1.2 DIFFERENT TYPES

The most common types of recommendation systems are CONTENT-BASED and COLLABORATIVE FILTERING recommendation systems. In collaborative filtering, the behavior of a group of users is used to make recommendations to other users. The recommendation is based on the preference of other users. A simple example would be recommending a movie to a user based on the fact that their friend liked the movie.There are two types of collaborative models MEMORY-BASED methods and MODEL-BASED methods. The advantage of memory-based techniques is that they are simple to implement and the resulting recommendations are often easy to explain. They are divided into two:

- **User-based collaborative filtering**: In this model, products are recommended to a user based on the fact that the products have been liked by users similar to the user. For example, if Derrick and Dennis like the same movies and a new movie come out that Derick like, then we can recommend that movie to Dennis because Derrick and Dennis seem to like the samemovies.

- **Item-based collaborative filtering**: These systems identify similar items based on users' previous ratings. For example, if users A, B, and C gave a 5-star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A, B, and C.

Model-based methods are based on Matrix Factorization and are better at dealing with sparsity. They are developed using data mining, machine learning algorithms to predict users' rating of unrated items. In this approach techniques such as dimensionality reduction are used to improve accuracy. Examples of such model-based methods include Decision trees, Rule-based Model, Bayesian Model, and latent factor models.

- **Content-based systems** use metadata such as genre, producer, actor, musician to recommend items say movies or music. Such a recommendation would be for instance recommending Infinity War that featured Vin Diesel because someone watched and liked The Fate of the Furious. Similarly, you can get music recommendations from certain artists because you liked their music. Content-based systems are based on the idea that if you liked a certain item you are most likely to like something that is similar to it.

### 4.1.3 CHALLENGES A RECOMMENDATION SYSTEM FACE

1. Sparsity of data. Data sets filled with rows and rows of values that contain blanks or zero values. So finding ways to use denser parts of the data set and those with information is critical.

2. Latent association. Labelling is imperfect. Same products with different labelling can be ignored or incorrectly consumed, meaning that the information does not get incorporated correctly.

3. Scalability. The traditional approach has become overwhelmed by the multiplicity of products and clients. This becomes a challenge as data sets widen and can lead to performance reduction.

## 4.2 DATA PRE-PROCESSING

For kNN based model, the underlying dataset ml-100k from the Surprise Python sci-unit was used. Shock may be a tight call in any case, to search out out regarding recommendation frameworks. It's acceptable for building and examining recommendation frameworks that manage unequivocal rating data.

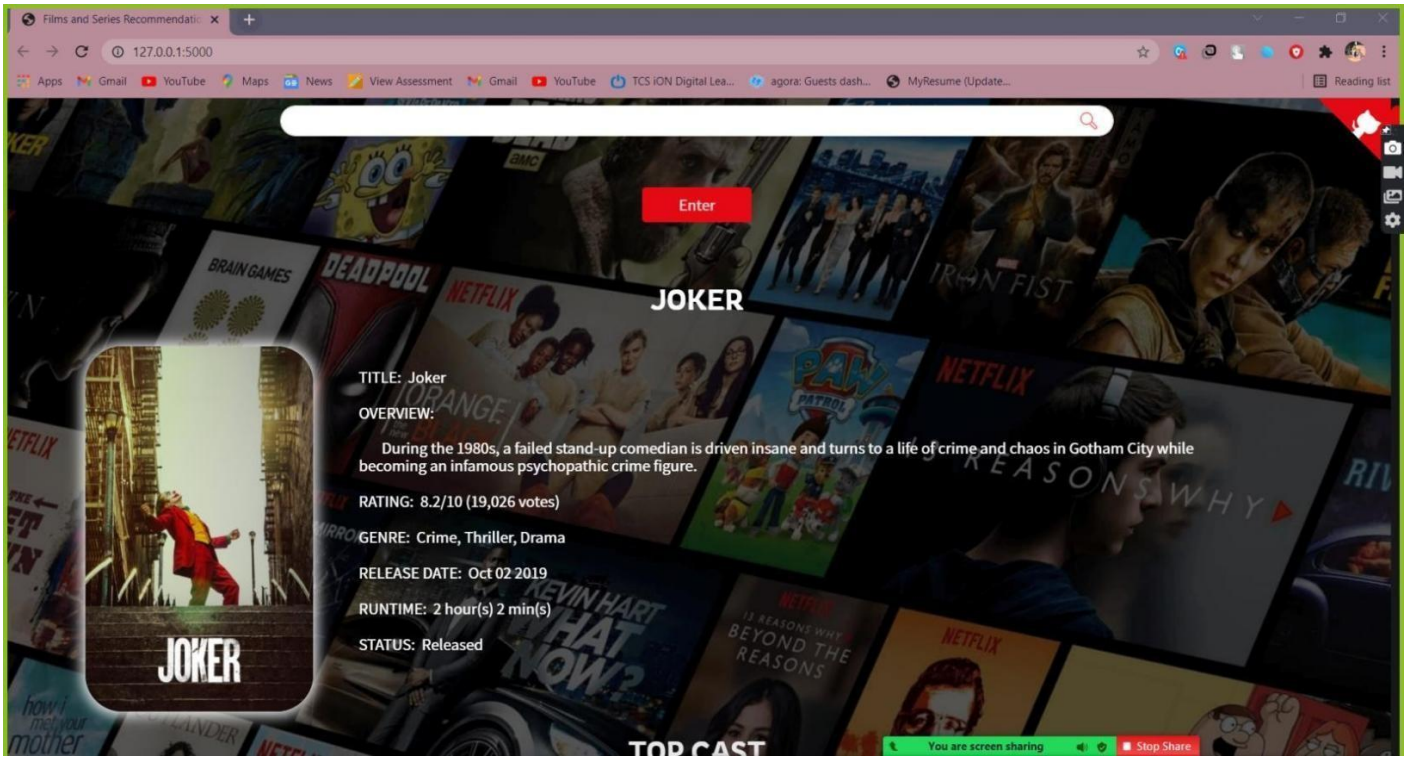
## 4.3 MODEL BUILDING

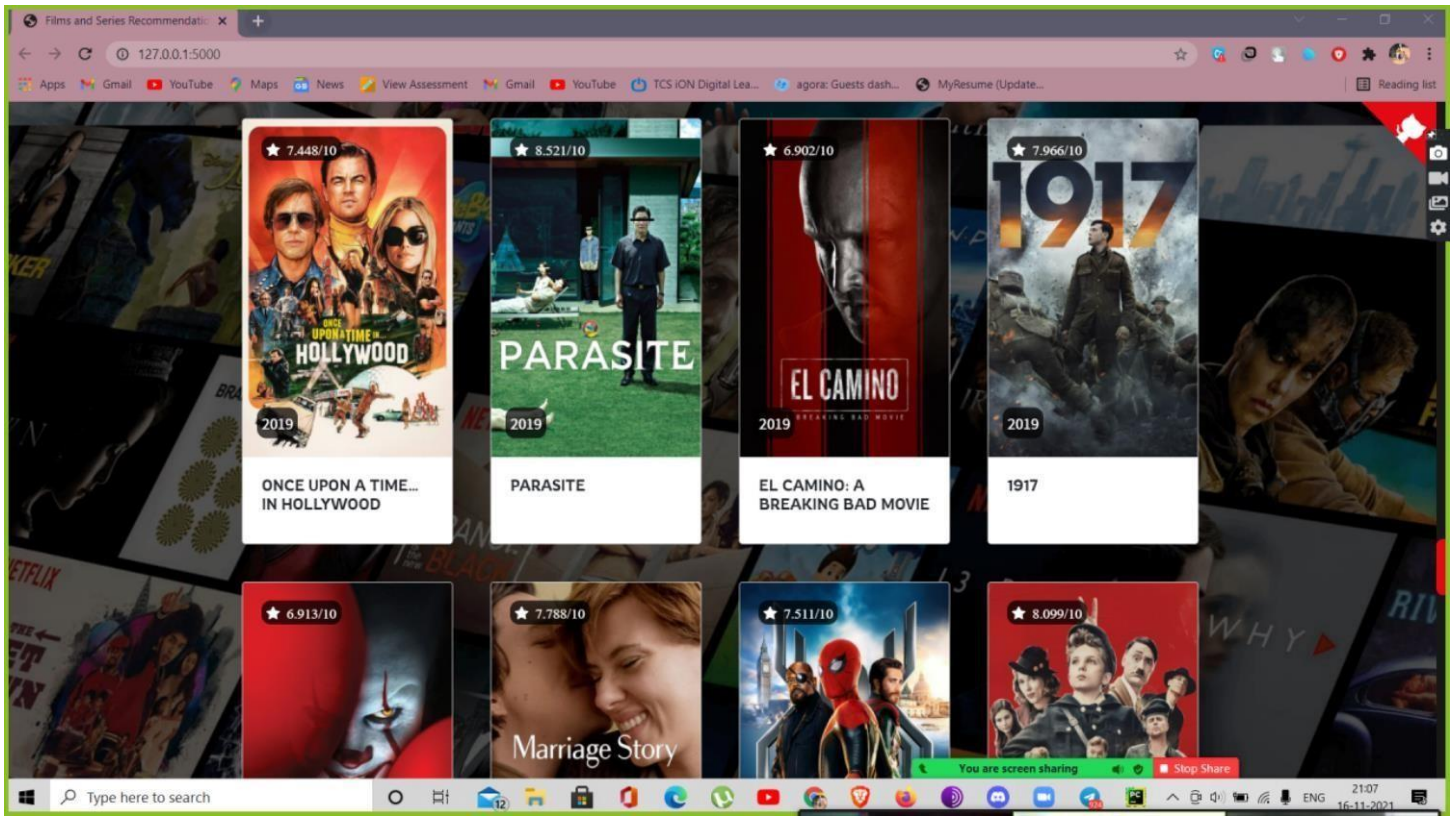
Information is an element into a seventy fifth train take a look at and twenty fifth holdout take a look at. Grid Search CV completed over five - overlap, is employed to find the most effective arrangement of closeness live setup (sim_options) for the forecast calculation. It utilizes the truth measurements because the premise to get completely different mixes of sim options, over a cross-approval system.

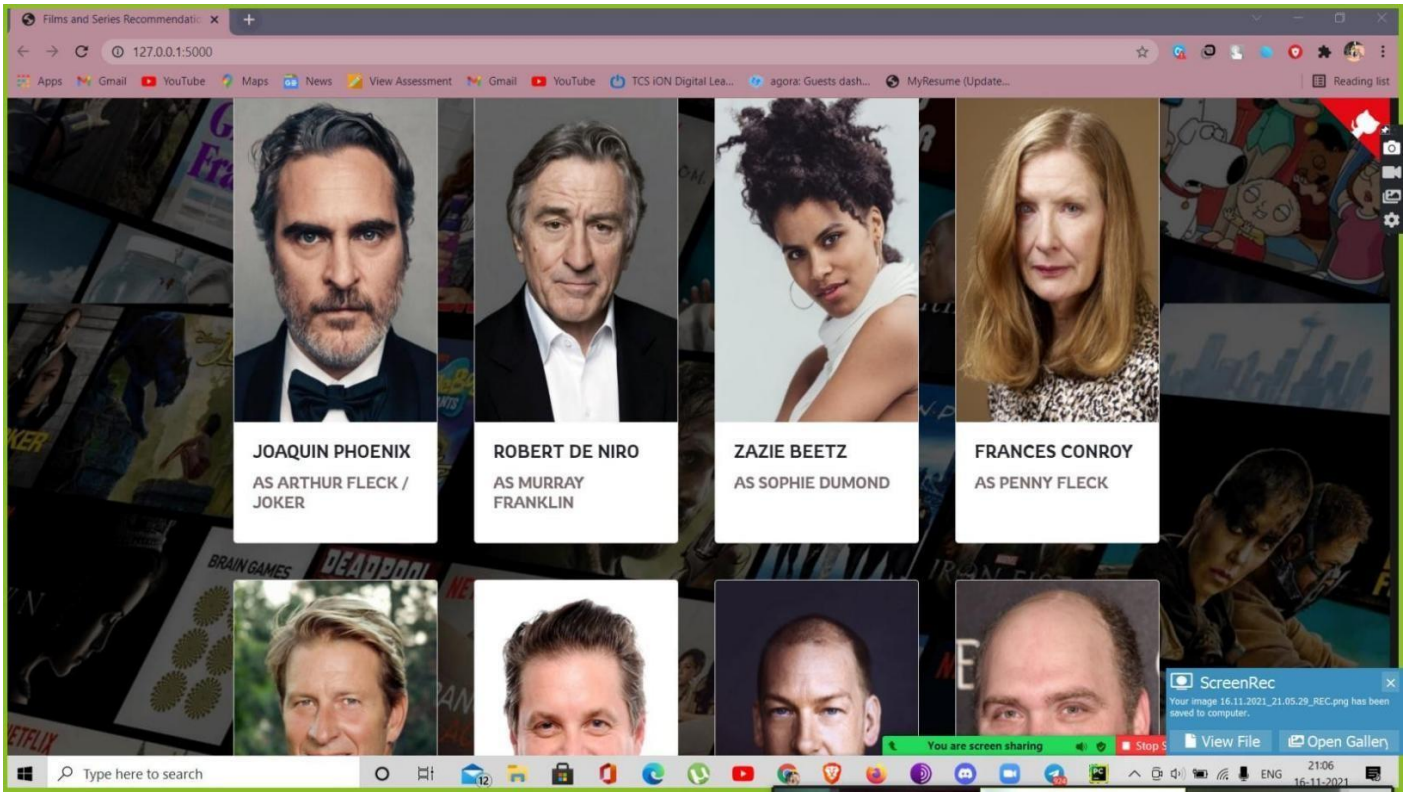
## 4.4 DATA SET USED:

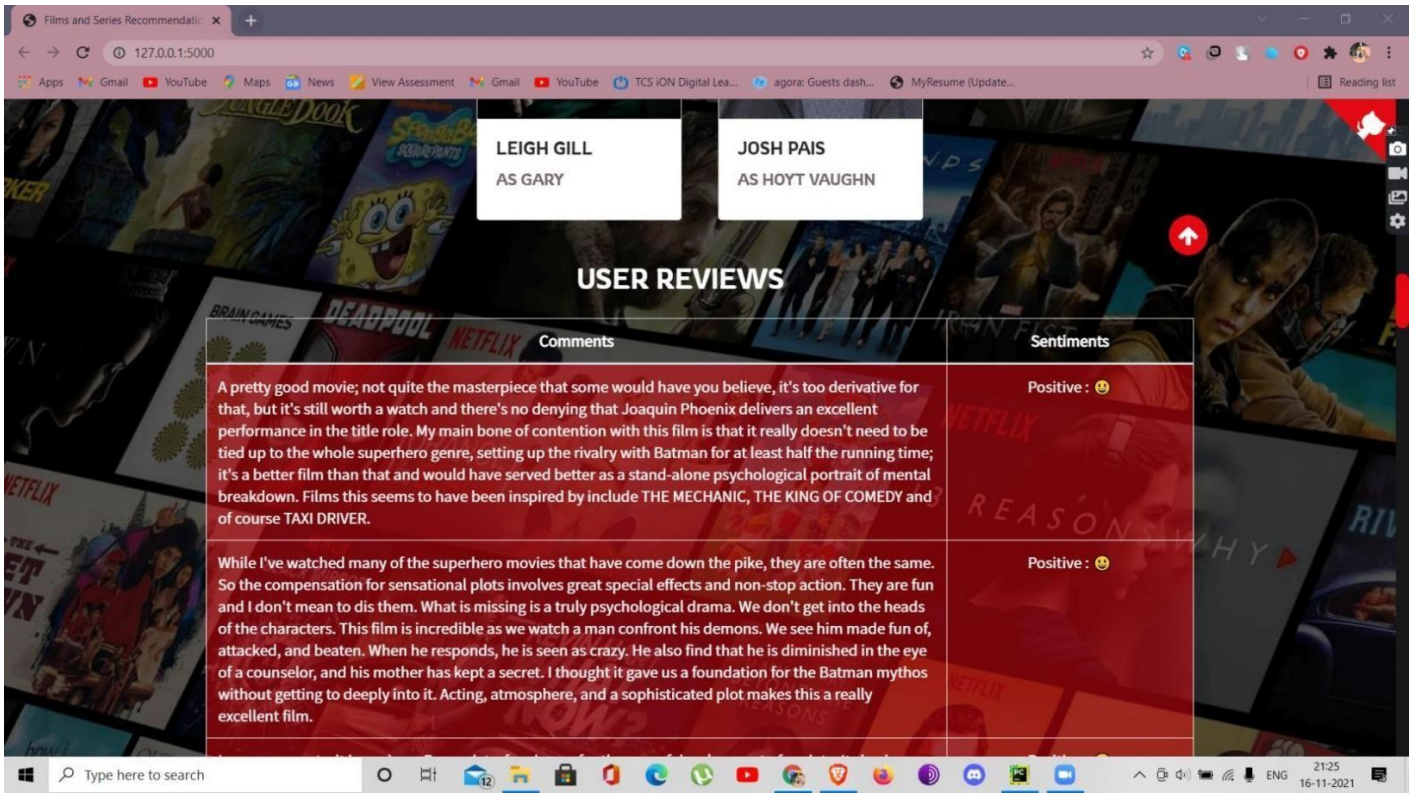
we are using the Movie Lens Data Set. This dataset was put together by the Group lens research group at the University of Minnesota. It contains 1, 10, and 20 million ratings. Movie lens also has a website where you can sign up, contribute reviews and get movie recommendations.

## 4.5 RECOMMENDATION VISUALIZATION:




Films and Series Recommendati
127.0.0.1:5000
7.448/10
2019
ONCE UPON A TIME... IN HOLLYWOOD
8.521/10
PARASITE
2019
PARASITE
6.902/10
EL CAMINO
2019
EL CAMINO: A BREAKING BAD MOVIE
7.966/10
1917
2019
1917
6.913/10
7.788/10
Marriage Story
7.511/10
8.099/10
You are screen sharing
Stop Share
Type here to search
ENG
21:07
16-11-2021


Films and Series Recommendati
127.0.0.1:5000
JOAQUIN PHOENIX
AS ARTHUR FLECK / JOKER
ROBERT DE NIRO
AS MURRAY FRANKLIN
ZAZIE BEETZ
AS SOPHIE DUMOND
FRANCES CONROY
AS PENNY FLECK
ScreenRec
Your image 16.11.2021_21.05.29_REC.png has been saved to computer.
View File
Open Gallery
You are screen sharing
Type here to search
ENG
21:06
16-11-2021


Films and Series Recommendatio
127.0.0.1:5000
LEIGH GILL
AS GARY
JOSH PAIS
AS HOYT VAUGHN
USER REVIEWS
Comments
Sentiments
A pretty good movie; not quite the masterpiece that some would have you believe, it's too derivative for that, but it's still worth a watch and there's no denying that Joaquin Phoenix delivers an excellent performance in the title role. My main bone of contention with this film is that it really doesn't need to be tied up to the whole superhero genre, setting up the rivalry with Batman for at least half the running time; it's a better film than that and would have served better as a stand-alone psychological portrait of mental breakdown. Films this seems to have been inspired by include THE MECHANIC, THE KING OF COMEDY and of course TAXI DRIVER.
Positive :
While I've watched many of the superhero movies that have come down the pike, they are often the same. So the compensation for sensational plots involves great special effects and non-stop action. They are fun and I don't mean to dis them. What is missing is a truly psychological drama. We don't get into the heads of the characters. This film is incredible as we watch a man confront his demons. We see him made fun of, attacked, and beaten. When he responds, he is seen as crazy. He also find that he is diminished in the eye of a counselor, and his mother has kept a secret. I thought it gave us a foundation for the Batman mythos without getting to deeply into it. Acting, atmosphere, and a sophisticated plot makes this a really excellent film.
Positive :
Type here to search
ENG
21:25
16-11-2021

# CHAPTER 5

# RESULT CONCLUSION AND DISCUSSION

## 5.1 CONCLUSION

In the last few decades, recommendation systems have been used, among the many available solutions, in order to mitigate information and cognitive overload problem by suggesting related and relevant items to the users. In this regards, numerous advances have been made to get a high-quality and fine-tuned recommendation system.

Nevertheless, designers face several prominent issues and challenges. Although, researchers have been working to cope with these issues and have devised solutions that somehow and up to some extent try to resolve these issues, however we need much to do in order to get to the desired goal. In this research article, we focused on these prominent issues and challenges, discussed what has been done to mitigate these issues, and what needs to be done in the form of different research opportunities and guidelines that can be followed in coping with at least problems like latency, sparsity, context-awareness, grey sheep and cold-start problem.

## CHAPTER 6

## REFERENCES


[1] https://www.geeksforgeeks.org/python-implementation-of-movie-recommender-system/

[2] https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/

[3] https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/

[4] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009

[5] A Nearest Neighbor Approach using Clustering on the Netflix Prize Data

[6] https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/

[7] Kharita, M. K., Kumar, A., & Singh, P. (2018). Item Based Collaborative Filtering in Movie Recommendation in Real-time. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). DOI:10.1109/icsccc.2018.8703362