A k-Armed Bandit with Thompson Sampling for Movie Recommendation

A PROJECT REPORT

Submitted by,

NETRA S BANKAR - 20201CAI0164

BHAVISH R -20201CAI0219

CHUKKA MANOJ KUMAR -20201CAI0199

ASHOK REDDY CHILAKA -20201CAI0203

Under the guidance of,

Dr. ALAMELU MANGAI JOTHIDORAI

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE

CERTIFICATE

This is to certify that the Project report "**A K-ARMED BANDIT WITH THOMPSON SAMPLING FOR MOVIE RECOMMENDATION"** being submitted by **"NETRA S BANKAR, BHAVISH R, CHUKKA MANOJ KUMAR REDDY, ASHOK REDDY CHILAKA** with roll numbers "**20201CAI0164, 20201CAI0219, 20201CAI0199, 20201CAI0203**"in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

11/1/24

Dr. ALAMELU MANGAI
Professor
School of CSE
Presidency University

11/01/24

Dr. ZAFAR ALI KHAN N
HoD
School of CSE
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. SHAKKEERA L
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING &

INFORMATION SCIENCE

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **“A K-ARMED BANDIT WITH THOMPSON SAMPLING FOR MOVIE RECOMMENDATION”** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **computer science and engineering,**

is a record of our own investigations carried under the guidance of **DR. ALAMELU MANGAI School of Computer Science Engineering , Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll_Number	Sign
NETRA S BANKAR	**20201CAI0164**	
BHAVISH R	**20201CAI0219**	
CHUKKA MANOJ KUMAR REDDY	**20201CAI0199**	
ASHOK REDDY CHILAKA	**20201CAI0203**	

ABSTRACT

With the ever-growing library of content on streaming platforms like Netflix, amazon providing users with personalized and engaging recommendations is crucial for enhancing their viewing experience. This paper explores the application of Multi-Armed Bandits (MAB) algorithms to optimize the recommendation system on Netflix. MAB, a branch of reinforcement learning, offers a dynamic approach to balancing exploration and exploitation, making it particularly well-suited for the evolving and diverse preferences of Netflix users.

The proposed system leverages MAB to dynamically allocate resources to different content recommendation strategies, continuously adapting to user feedback and evolving viewing habits. Traditional recommendation systems often struggle with the exploration-exploitation trade-off, either focusing too heavily on known preferences or exploring new content at the expense of user satisfaction. MAB algorithms address this challenge by intelligently allocating resources to explore new content and exploit successful recommendations based on real-time user interactions.

The study involves the development and integration of a MAB-based recommendation model within Netflix's existing recommendation infrastructure. Through experimentation and A/B testing, the performance of the MAB-enhanced system is evaluated against baseline recommendation methods. The metrics include user engagement, click-through rates, and user satisfaction, among others.

Preliminary results demonstrate the potential of the MAB approach to significantly improve the quality of recommendations on e-commerce in websites. By adapting in real-time to user behaviour, the MAB-based system achieves a more personalized and responsive recommendation experience. The findings suggest that MAB algorithms can enhance the efficiency and effectiveness of content recommendations on large-scale streaming platforms, ultimately leading to increased user satisfaction and prolonged engagement.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science and Engineering and School of Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C** and **Dr. Shakkeera L**, School of Computer Science and Engineering and School of Information Science, Presidency University and **Dr. Zafar Ali Khan**, Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr Alamelu Mangai**, Assistant Professor, School of Computer Science and Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and the department Project Coordinator **Dr Murali Parameswaran**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project

NETRA S BANKAR - 20201CAI0164

BHAVISH R - 20201CAI0219

CHUKKA MANOJ KUMAR - 20201CAI0199

ASHOK REDDY CHILAKA - 20201CAI0203

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Netflix uses collaborative filtering and content-based filtering to recommend content to users. Collaborative filtering analyzes user behavior, preferences and interactions to recommend content. It can detect patterns of users with similar tastes and recommend products based on similar users' interests. This method does not require deep knowledge of the project itself, but is based on the user's behavior. Content-based filtering focuses on the content of the project. By analyzing the features of the products the user has liked in the past, it recommends products similar to the products the user has liked in the past and recommends products with similar features; However, both methods have limitations. Collaborative filtering can suffer from a “cold start,” making it difficult to provide recommendations to new users or projects with insufficient data. Content filtering will have difficulty capturing preferences or needs that cannot be influenced by the user's design

1.1 Content-Based Filtering:

How it works: Content-based filters show products based on their features or functions. For example, if the user likes movies of a particular actor or director, this method will recommend videos with similar characteristics regardless of other users' preferences.

Disadvantages: It will be difficult to capture diversity or changing user tastes. This method will not change fast enough if the user's preferences change or they look for something different.

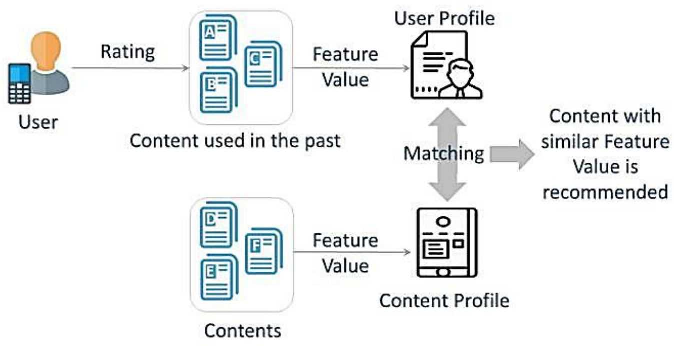


Fig. 1.1 Content based Filtering

Source:(**Machine Learning Model for Personalizing Online Arabic Journalism)**

Figure 1.1 shows how content-based filtering works. When the user reads the examples (A, B, and C), the model captures the main features of the selected items, which are the main points in the example, as well as the main points of the equipment. The user's score is automatically calculated based on the location of the article the user read and the time spent in each section. The model saves user preferences in the background. When a new article is published, it will be checked whether the model meets the customer's needs. If the user likes it, Recommend it to the user.

1.2 Collaborative Filtering:

How it works: Content-based filters show products based on their features or functions. For example, if the user likes movies of a particular actor or director, this method will recommend videos with similar characteristics regardless of other users' preferences.

Disadvantages: It will be difficult to capture diversity or changing user tastes. This method will not change fast enough if the user's preferences change or they look for something different

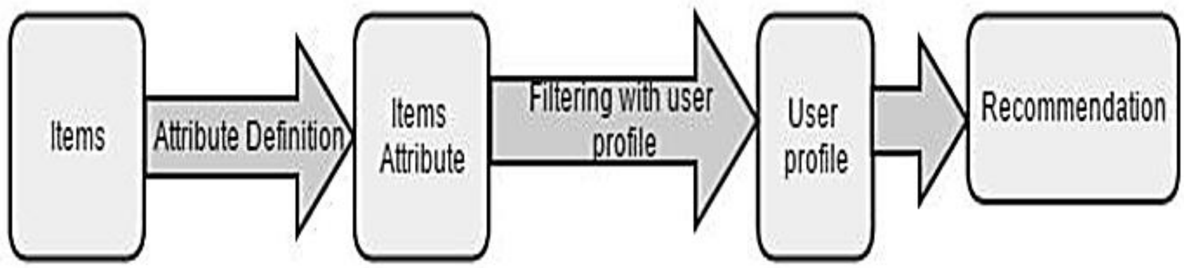


Figure 1.2. Collaborative filtering algorithm framework

Source:(**Machine Learning Model for Personalizing Online Arabic Journalism)**

In fig 1.2 shows how the collaborative filter works in recommendation. It calculates a user's rating for a product (phrase and keyword) and then predicts and recommends available products based on other users' products (phrases and keywords) (text and content) by presenting them measure by measure to existing users.

1.3 Collaborative Filtering vs Content-Based Filtering:

Table 1.1: Collaborative Filtering vs Content-Based Filtering

S.NO	Collaborative Filtering	Content Based Filtering
1	You must understand the features of the product.	Must have background or historical interest.
2	Requires very little start up input from the user, so there are no cold start issues.	New tasks cannot be created and cause cold boot problems.
3	Recommendations are given solely based on the user's current interest..	Help you find new interests.

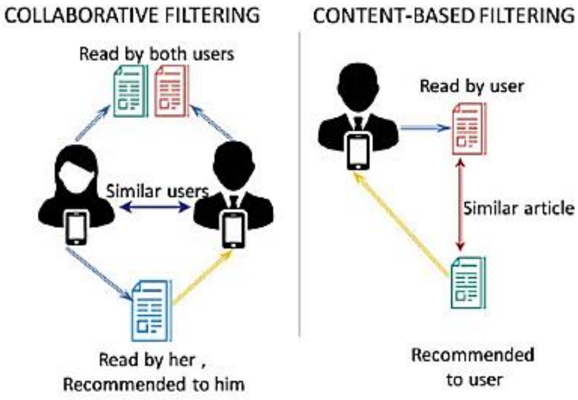


Fig. 1.3. Collaborative Vs. Content based Filtering

Source:(**Machine Learning Model for Personalizing Online Arabic Journalism)**

1.4 Multi Armed-Bandit:

Multidisciplinary forces are problem-solving models that use diverse fields such as machine learning, economics, and operations research. It involves deciding how to allocate resources in a way that balances exploration (trying different ways to gather information) and exploitation (using known information to make more money). The name "The Bandit" comes from the idea of a casino that introduces many slot machines (the one-armed bandit) and aims to maximize its winnings over time.

The advantages of using different types of soldiers are:

1. **Research and Budgeting**: It is equally helpful to research different options to gather more information and use the best options to get the best results.

2. Adaptability: As it learns from past behaviour, it changes its strategy over time to optimize performance, making it suitable for dynamic environments where outcomes will change.

3. Effective learning: Has the ability to make smart choices based on past performance, thus learning and making decisions quickly.

4. Resource Efficiency: Compared to traditional testing methods that would distribute resources evenly among options, the joint military approach can allocate resources more efficiently by favouring the better options.

5. Instant decision making: Suitable for situations that require immediate decision making or situations that require constant learning and adjustment.

6. Applications in various fields: This method has applications in various fields, such as online advertising, clinical trials, recommendations, networking, etc., and it shows that it has many things.

However, it is worth noting that military warfare algorithms have limitations and difficulties as well as their advantages. For example, sometimes they may not be able to adapt to the best solution in the world, and it can become difficult to effectively manage the balance between search and use.

CHAPTER-2

LITERATURE SURVEY

A Contextual Multi-Armed Bandit Approach Based on Implicit Feedback for Online Recommendation

The article[1] unified filter bandit algorithm is proposed for dynamic consensus fields. This algorithm involves collaboration between users and projects by grouping users and projects based on their behavior. Visual analysis shows a better predictive value compared to state-of-the-art methods. Performance improvement: Visual analysis shows that the algorithm outperforms the state-of-the-art method in predicting performance. The algorithm dynamically groups users and products based on their behavior to better adapt to changes in user preferences and similar products over time. Visual analytics is a multi-stakeholder approach, from implicit feedback to online consensus.

A Contextual Multi-armed Bandit

This paper[2] outlines various military methods for online confirmation based on implicit feedback. The author presents a method that combines the advantages of content with various military algorithms to improve agreement. They also discuss various existing algorithms and ideas in the field of multi-armed bandits and contextual bandits for the proposed systems. Multi-armed bandits can search and use different options, allowing for a more efficient operation. Both algorithms can be adjusted to changing customer preferences over time, increasing the accuracy and precision of recommendations. Contextual Bandit and Multi-Armed Bandit are scalable and can handle large-scale consultant systems with multiple projects and users. These algorithms can be used in real-time situations, making them suitable for traditional models. This article mainly discusses the process of content of various military and various algorithms currently on offer. It provides a survey of practical applications and highlights the importance of the content in the user's opinion. However, there is no clarity about some of the limitations or shortcomings of the said process or discussion algorithm

Adaptively Optimize Content Recommendation Using Multi Armed Bandit Algorithms in E-commerce

Article[3] discusses the problem of various military forces, especially in the context of supporting web searches. In this case, an online algorithm chooses the most profitable actions based on the given context (such as the user's search query). The author proposed a regret algorithm O(Ta + b + 1a + b + 2 + ε); where a and b are the program sizes of the query site and the broadcast site. They also prove that any algorithm has fewer regrets. This problem is related to search engines crawling the site, suitable for its purpose, suitable for its purpose, suitable for its purpose, suitable for its purpose, suitable for its purpose, suitable for its purpose, dealing with publications. Maximize click-through rate: This algorithm is designed to display ads relevant to user queries, increasing clicks and potentially increasing expected revenue. 2. Learning over time: The algorithm constantly learns and adapts based on user questions and social media, improving the accuracy and effectiveness of ads. 3. Search and efficiency: The algorithm optimizes the balance between immediate benefits and long-term learning by striking a balance between using existing ads and searching for relevant ads. One of the limitations mentioned, depending on the article you are reading, is that an algorithm has been proven to have lower regret. This means that no algorithm can achieve regret below the limits specified in the article. This limitation suggests that there may be limits to the effectiveness of the content of many rogue algorithms for optimizing the promotion of web searches.

Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System

This paper[4] discusses the use of multiple military algorithms for content recommendation in e-commerce. The authors analyze three competing military assets and measure their performance using simulated data and real-world A/B testing. They have also introduced a batch update process to resolve data latency issues and provide true online A/B performance testing. Adaptive Optimization: The MAB algorithm dynamically optimizes content recommendations based on user input to accommodate personalization and recommendations. Beyond the reward: Compared to traditional A/B testing, the MAB algorithm described in this article can be more profitable, especially when the success of campaign recommendations varies greatly. According to the article, one of the limitations mentioned is that the MAB algorithm faces difficulty for non-fixed users. User preferences and behaviors may change over time, and the MAB algorithm may have difficulty adapting to these rapid changes. This can lead to recommendations if the algorithm cannot search and use the most important content. To achieve optimal performance in an e-commerce environment, the MAB algorithm needs to be regularly monitored and tuned

Recommendation systems: Principles, methods and evaluation

This paper[5] provides an overview of proposals, including integration, content filtering, and a combination of the two. Key strategic issues such as data sparsity, scalability, diversity, and attack vulnerability are discussed. This article also describes memory-based and model-based joint filtering techniques. We also explore the concepts, benefits, and limitations of content filtering. Personalization. Recommendations can increase user satisfaction and engagement by providing personalized recommendations based on user preferences. Encourage user interaction. By providing recommendations, they can keep users interested and encourage them to explore more content or products. Recommendations can be difficult to provide recommendations to new users or projects with limited knowledge. This is called the cold start problem, where there are not enough users or data to make good recommendations.

Collaborative Filtering Bandits

This article[6] explores the principles, methods, and evaluation of consensus. He mentions that to solve problem of information overload, people must agree to filter and submit relevant information. This article reviews different prediction methods in the recommendation process and provides examples of collaborative filtering and content-based filtering. He also discusses the hybrid filtering process, which offers more options. Article finally discusses comparison of recommendation algorithms. Personalization: Recommendations can provide personalized recommendations based on user preferences and behavior and improve user experience by providing relevant content or products. Increased engagement: Providing recommendations through recommendations can increase user engagement and keep users coming back for more visits. Recommendations rely on user data interaction plans to generate recommendations. However, when there are interactions or ratings on certain products, inconsistent information can become a problem and customer preferences can become difficult to predict.

A brief overview of the multi armed bandit in RL

This article [7] gives a brief summary of the problems many soldiers have in advanced training. He defines two types of warfare (random and competitive) and discusses popular algorithms such as epsilon-greedy, UCB, and Thompson Sampling. This article also addresses the persistence of the multiple theft problem and concludes by highlighting the importance of this problem in real use. Flexibility: Many soldiers have the ability to control a chaotic and chaotic environment, making them suitable for many locations. There are many real world applications. Use of exploration and sampling methods: Methods such as Epsilon greedy, UCB and Thompson Sampling allow good decision making by allowing different selection of search methods and use of the best experience. Content understanding method: content-based Filters are based on determining the features of the product to be recommended. However, it may be difficult to understand the full meaning or context of the content, causing recommendations to be inaccurate or irrelevant. This limitation is particularly evident in complex or subjective domains where content analysis may not capture subtle preferences or user preferences regarding the activity being analyzed.

MAB for recommendations and A/B testing on amazon rating datasets

This article [8] explores the concept of Multi-Armed Bandits and their applications in the optimization process and A/B testing. It explains how Multi-Armed Bandits work by evaluating research and application, and provides examples of real-world situations using it. approach. This article also discusses the differences between various military sampling methods and introduces three popular sampling methods: Epsilon-Greedy, Upper Confidence Bound (UCB), and Thompson Sampling. He highlights the advantages of using Multi-Armed Bandits over traditional A/B testing and concludes that Thompson sampling is the most effective among the algorithms in question. The article also addresses problems in evaluating algorithms using historical data and offers a solution called offline replay evaluation. Advantages: Continuous Learning: Multi-Armed Bandits is constantly learning and updating its recommendations based on user feedback; This makes it suitable for the environment where users' Favorites may change over time. Efficient allocation of resources: The bandit algorithm allocates resources more efficiently by focusing on the best options, reducing the need to re-search and gather information about inferior options. Limitations: Many Body Bandits refer to computational complexity and program complexity. Implementing and optimizing the Multi-Armed Bandit algorithm can be difficult and requires expertise in algorithm design and implementation. The article also mentions the difficulty of using historical data to evaluate algorithms due to biases in the data. Offline retesting has been proposed as a solution to this limitation, but may still cause problems in evaluating the effectiveness of the Multi-Arm Bandit algorithm.

MAB in recommendation system A survey of the state-of-art and future direction

This paper [9] conducts a qualitative literature review on Multi-Armed Bandits (MAB) from a formal perspective. This review aims to provide an overview of research in this field by focusing on the most commonly used techniques and methods and evaluating the applicability of MAB-based methods. This review identifies several gaps in the existing literature and provides direction for future research. Advantages: This article gives a review of Multi-Armed Bandits (MAB) in the recommendations area. This review aims to provide an overview of research in this field by focusing on the most commonly used techniques and methods and evaluating the applicability of MAB-based methods. This review identifies several gaps in the existing literature and provides direction for future research. Limitations: There is no consensus on best practices for designing, implementing, and evaluating Field Multi-Armed Bandit (MAB) deployments. This article highlights the need for further research to address this gap and provides clear guidelines for the use of MAB-based recommendations.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

3.1 Research Questions in Focus

This section describes the 6 research questions (RQs) presented in Table 2, which are combined to provide detailed information on various aspects of the recommendations. The following subsections attempt to highlight these points in response to the RQ.

Table 3.1 : Research Questions

RQ#	**Research Question Statement**
RQ1	What are the pros and cons of the proposed RS internship?
RQ2	What issues and problems are encountered in RS deployment?
RQ3	What are the many applications and responses RS receives?
RQ4	What measures are used to measure the quality of RS?
RQ5	What is the difference in RS quality? Current RS research?
RQ6	What methods does RS use to generate recommendations?

Collaborative Filtering (CF) Technology Collaborative filtering technology recommends products to target users based on the past preferences of other similar users. CF helps users make choices based on the opinions of others. For this reason, it is also called verbal communication. Last.fm, Ringo and Video Recommender are examples of CF systems. CF technology falls into two groups: memory-based approach, which makes recommendations using all user data; and the model-based method, which first initializes a model of user data and then uses it to make predictions. Content-based filtering (CB) recommends products similar to ones the user has liked in the past. Create a profile for each user or project that describes their unique characteristics. For example, video profile elements include the type of video, director, actors, popularity, etc. it could be. User data may include product selections, ratings, demographic information and more. Profiles help recommendations connect users to the right products. Pandora Internet Radio is a great example of this approach. Social filtering

(SF), also known as RS in the community, recommends activities based on the interests of the user's friends. These RSs follow the phrase: "Tell me who your friends are and I'll tell you who you are." In general, users trust the recommendations of their friends more than the recommendations of strangers. This type of system collects users' social information and their friends' preferences from social networks and creates recommendations based on the ratings given by users' friends. Different proposals for different public roles. For example, some websites consider certain personalization methods based on demographic characteristics such as language, country, or age. Knowledge-based systems (KB) rely on domain knowledge to share projects with users. They share the project's features with the user's needs and preferences to determine whether the project will be beneficial to the user. These systems are called case-based systems; A similar function measures how well the user's needs correspond to the recommendations. Utility-based systems (UB), like cognitive systems, base their recommendations on a combined assessment of user needs and available options. These machines calculate the energy cost of each product and recommend it to the user. Hybrid RS follows a hybrid approach that encompasses all other avenues to achieve some synergy between them. This process of combining two technologies tries to solve the disadvantages of one technology by using the advantages of the other. For example, CF and CB methods can be used together to avoid a new project using CF technology. Many different methods have been proposed in the literature to create composite systems by combining two or more methods. In addition to the above methods, many other methods have been found in the literature, including information that can be used to improve RS performance; context-aware recommendations to users without explicit request; Semantic-based semantic methods are similar to determining the relationship between objects presented in the written ontology and user interests. From the data, it can be seen that CF and CB are the most popular methods that are recognized and widely used in the field of RS compared to other methods.

3.2 Strengths and weakness of RS approaches:

In this subsection, we will review an extensive scientific and technological literature to examine and identify the advantages and disadvantages of the proposed method. Each of these has its own advantages and disadvantages, which are summarized in Table 3 and asked in future questions. RQ2: What are the pros and cons of recommendations for implementation in SC? Updated January 2022 In my experience, Netflix recommendations, like many other recommendations, rely on collaboration, content filtering, and the hybrid process.

3.3 Issues and Challenges in Recommendation System

Here are some of the common findings and issues in the approval process as of my last update. Note that specific issues with Netflix's recommendations may have changed since then:

1. Cold start issue:

Cold start issue occurs when a new user or project does not have any interactive data. In this case, it is difficult for the approval process to make recommendations. Research may focus on developing ways to deal with cold start situations, such as adding publicly available information or content.

2. Context-Aware Recommendations:

Integrating contextual information, such as time, location, and user behavior, into recommendation algorithms remains a challenge. Improving the accuracy and relevance of recommendations by considering the context in which users consume content is an ongoing research are

3. Dynamic User Preferences:

User preferences can change over time, and existing recommendation systems may not effectively capture these changes. Research could explore adaptive algorithms that continuously update user profiles based on evolving preferences, ensuring that recommendations remain relevant.

4. Expandability and Interpretability:

Many recommendation systems, including Netflix's, are considered "blackbox" models, making it challenging for users to understand why specific recommendations are made. Research and development of more defined consensus models can increase user confidence and satisfaction.

5. Dealing with multiple approvals:

It is difficult to ensure that the approval process includes different customers and avoids overspecialization. Research could focus on ways to strike a balance between giving positive feedback and demonstrating diversity to avoid dissent.

6. Contains false feedback:

Most recommendations are based on clear user feedback (ratings, likes), but false feedback (search history, view time) is also important. Research into improving the integration of implicit feedback to achieve more positive feedback is of ongoing interest.

7. Scalability and Efficiency:

As the amount of data continues to grow, ensuring that recommendation systems are scalable and efficient is crucial. Research could focus on optimizing algorithms and infrastructure to handle large datasets and provide real-time recommendations.

8. Ethical Considerations and Fairness:

Ensuring fairness in recommendations, avoiding biases, and addressing ethical considerations are increasingly important. Research could focus on developing algorithms that are more transparent, fair, and unbiased in their recommendations.

To get the most update information on research gaps specific to Netflix's recommendation system, it's recommended to explore recent publications and academic papers in the field of recommender systems and machine learning. Conference proceedings such as ACM RecSys (Conference on Recommender Systems) and journals like the Journal of Machine Learning Research often feature the latest research in this area.

Limited Serendipity: Contentbased filtering relies on explicit features of items to make recommendations. This can lead to recommendations that are too similar to a user's past preferences, limiting the system's ability to introduce users to new and diverse content (lack of serendipity).

Feature Engineering Challenges: Contentbased systems require accurate and relevant features to describe items. Obtaining and maintaining these features can be challenging, especially for dynamic content libraries or when dealing with user-generated content.

OverSpecialization: Based filtering systems may suffer from overspecialization,where recommendations are overly tailored to a user's past behavior and may not capture evolving preferences or sudden changes in interests.

Dependency on Feature Quality: The effectiveness of contentbased filtering is heavily depen dent on the quality and relevance of item features. If features don't accurately represent user preferences, the system's recommendations may be suboptimal.Netflix, like many recommen dation systems, often employs a hybrid approach that combines collaborative filtering and co ntentbased filtering to mitigate some of these drawbacks and provide more accurate and dive rse recommendations. Additionally, exploring innovative approaches such as MultiArmed B andits (MAB) can help address some of the limitations inherent in collaborative and content based methods.

Netflix, like many recommendation systems, often employs a hybrid approach that combines collaborative filtering and contentbased filtering to mitigate some of these drawbacks and pr ovide more accurate and diverse recommendations. Additionally, exploring innovative appro aches such as MultiArmed Bandits (MAB) can help address some of the limitations inherent in collaborative and content-based methods.

3.4 Existing Solutions

Netflix's recommendation system is based on a combination ofvarious algorithms, each of which results in a different recommendation process. Here are some of the important methods and techniques used in Netflix offering:1. Matrix Factorization: Collaborative Filtering: Matrix factorization techniques such as Singular Value Decomposition (SVD) or Alternative Least Squares (ALS) help identify latent factors in the customer interaction matrix to make predictions about users based on user preferences. 2. Content-based filtering: Natural Language Processing (NLP): Analyzes metadata and descriptions of movies or TV shows to recommend content based on similar text or topics. Genre, Actor and Director preferences: Use features such as genre, actor, director or publication year to suggest content similar to content the user has previously liked. 3. Hybrid Algorithm: Integrated approach: Combine various recognition technologies (such as collaborative filtering and content-based filtering) to increase accuracy and provide diversity > and personalization is recommended. Parse Machine: Use this technique to capture the interaction between features (such as customer preferences and products) in a more complex model. 4. Deep Learning: Deep Neural Networks: Applying neural network architectures to understand complex patterns in user behavior, preferences, and content features for more accurate predictions. Autoencoders: Unsupervised learning techniques used to learn efficient representations of input data, often applied in feature learning or dimensionality reduction for content or user profiles.5. Bandit Algorithms: Contextual Bandits: Exploring the use of bandit algorithms to dynamically optimize recommendations based on contextual information, adjusting recommendations in real-time based on user interactions.6. Reinforcement Learning: Interactive Reinforcement Learning: Experimenting with reinforcement learning techniques to optimize recommendations by learning from user interactions and feedback in a sequential manner.7. A/B Testing and Personalization: A/B Testing Frameworks: Continuously testing new algorithms or features with a subset of users to evaluate performance and effectiveness before deploying them to the wider user base. Personalized Ranking Systems: Tailoring recommendations based on individual preferences, using user-specific features and feedback to rank items uniquely for each user. Netflix constantly innovates and evolves its recommendation system, experimenting with new algorithms and approaches to improve accuracy, relevance, and user satisfaction. The blend of these algorithms helps create a robust and dynamic system that caters to the diverse tastes and preferences of its global user base.

CHAPTER-4

OBJECTIVES

Objectives Suggested objectives of the Netflix video when using the Multi-Armed Bandit (MAB) search-exploitation strategy:

1. Personalized Recommendations: The main aim is to provide movie recommendations to each user based on their personal preferences, viewing history and feedback.

2. Maximize user engagement: Recommendations maximize user engagement by recommending videos that users like most, thus maximizing time to watch and enjoy. 3. Balance of search and spending: The system aims to strike a balance between searching for new videos to understand user preferences and using the best video movie recommendations to make users happy.

4. Real-time learning and updating: The system strives to constantly update with user interactions and instantly adjusts its recommendations to reflect changes in user preferences and behavior.

5. Performance metrics: The system is designed to improve performance metrics such as click-through rate, view time, user retention, and overall user satisfaction using the MAB search implementation strategy.

6. Efficient allocation of resources: Using the MAB algorithm, the system aims to allocate resources efficiently by focusing on recommending videos that users are most likely to benefit from.

7. Improve content discovery: The goal is to help users explore a variety of video content, including popular content and specific videos, while ensuring that the message agrees to accommodate the user's preferences. Netflix movie recommendations aim to achieve the goals of offering users the opportunity to watch personalized and interesting movies by using the MAB search-consumption strategy.

CHAPTER-5

PROPOSED METHODOLOGY

4.1 Problem statement

Collaborative filtering and content-based filtering are two methods commonly used in recommendations, including the method used by Netflix. While these techniques have proven effective in many cases,

they also have some disadvantages: Disadvantages of integrated filters:

• Cold start problem:

Collaborative filtering relies on user interaction history to make recommendations. As a result, new users or projects with limited engagement history may experience difficulties receiving positive and relevant feedback, leading to cold start issues.

• Data sparsity:

The efficiency of the system in collaborative filtering depends on the availability of large and dense data. Inconsistent data (users only interested in a small number of available products) can affect the accuracy of recommendations.

• Scalability issues:

As the number of users and system layout increases, collaborative filtering will face scalability issues. As datasets grow, computation and storage requirements increase, potentially impacting performance.

• Low interpretability:

Integration filter models are often uninterpretable. While they can make accurate predictions based on user behavior, they may not provide insights into why a particular recommendation was made, making it challenging to explain recommendations to users.

Drawbacks of Content-Based Filtering:

• Limited Serendipity:

Content-based filtering relies on explicit features of items to make recommendations. This can lead to recommendations that are too similar to a user's past preferences, limiting the system's ability to introduce users to new and diverse content (lack of serendipity).

• Feature Engineering Challenges:

Content-based systems require accurate and relevant features to describe items. Obtaining and maintaining these features can be challenging, especially for dynamic content libraries or when dealing with user-generated content.

• Over-Specialization:

Content-based filtering systems may suffer from over-specialization, where recommendations are overly tailored to a user's past behavior and may not capture evolving preferences or sudden changes in interests.

• Dependency on Feature Quality:

The effectiveness of content-based filtering is heavily dependent on the quality and relevance of item features. If features do not accurately represent user preferences, the system's recommendations may be suboptimal.Netflix, like many recommendation systems, often employs a hybrid approach that combines collaborative filtering and content-based filtering to mitigate some of these drawbacks and provide more accurate and diverse recommendations. Additionally, exploring innovative approaches such as Multi-Armed Bandits (MAB) can help address some of the limitations inherent in collaborative and content-based methods.

4.2 Proposed Solution:

The current agreement comes as streaming platforms like Netflix face persistent problems, especially with collaboration and content-based filtering processes. Collaborative filtering relies on user interaction history and causes cold start issues for new users and problems processing disparate data. In contrast, content-based filtering relies on intangible product characteristics and may be limited to capturing user preferences. To solve these problems, this study attempts to reduce the limitations of the current consensus by proposing new ideas.

The power of the Multi-Armed Bandit (MAB) algorithm. The MAB framework supports discovery and change and creates strategies to improve the approval process through continuous learning and real-time updates of user preferences.

The main objectives of this research are:

• Dynamic Adaptation: Creating a system where recommendations are adapted according to user preferences using the MAB algorithm. Based on the difference between static collaboration and content-based approach, the aim is for the system to continuously learn and update its suggestions based on user interaction and feedback. • Reducing the cold start issue: Discover and approve content for new users without history restrictions by solving the cold start issue associated with collaborative filtering. Even with inconsistent data, the MAB algorithm should provide accurate recommendations, thus reducing the cold start problem.

• Improved personalization: Overcome the limitations of content filtering by better understanding user preferences. MAB-based recommendation systems aim to establish a balance between searching for different content and using recommendations, enabling the personal development of many users.

• Real-time optimization: Real-time employee approval time to enable rapid changes in user behaviour and preferences. The MAB framework allows for instant changes, providing more responsive and user-friendly solutions. This research aims to demonstrate the ability of the MAB algorithm to transform Netflix recommendations by achieving these goals. The solutions not only combine the advantages of collaboration and content-based approaches, but also reduce cold start issues, making content more flexible and useful for new and existing users of negotiated mainstream media.

4.3 Methodology

Preprocessing:

After the data have been acquired, the next step is to preprocess the data in order to prepare them for further analysis. This comprises cleaning and altering the data in order to prepare it for analysis in the most effective manner possible. When cleaning your data, you should remove duplicates, record previous experiences, and eliminate possible objections. The process of modifying the profile requires, among other things, measuring features, coding categorical variables, and creating new features (if necessary).

Feature Selection:

The next step is to identify the main features of the model, which brings us to the next step, feature selection. At this stage, we decide the quality of the model. The feature selection process involves determining which features are most important and have the greatest impact on the variables under study. This is an important step in the feature selection process. Methods such as the chi-square test, correlation analysis, and elimination of duplicates are just a few examples of the many techniques that can be used to complement the brand.

Model Selection Method:

After determining which features are important for the current situation, the next stage is to select the model that best meets the criterion need. This comes after deciding what is important to the case. In this case, Thomson Sampling, epsilon greedy, UCB-1 (confidence limit above -1), SoftMax search, etc. We will analyze the effectiveness of several different learning systems such as Model evaluation: The Next Step is to evaluate the effectiveness of the selected model.

1. Precision@K

This is similar to normal precision, but calculates the precision of the first k items while sorting the way you want. This allows you to change the K and see how the score changes

$$\text{Precision@K} = \frac{\text{Number of relevant items in top} - \text{k}}{\text{K}}$$

It has many uses in the healing world. A very common use is to measure search engine performance based on the top 10 results for a query.

2. Recall@K

It is similar to the normal return metric and similar to the Precision@K metric above. There is only a minor change in the model compared to the model above.

$$\text{Recall@K} = \frac{\text{Number of relevant items in top} - \text{k}}{\text{K}}$$

It is very important in cases where there are only a few products and we really want them to rank high.

3. MAP@K

MAP@K or Average Precision @ K is the professional version of Precision@K. It is good to measure the accuracy of the truth rather than just measuring the 1 value of K. Let's look at the average truth @K.

$$AP@K = \frac{1}{N}\Sigma(precision@i) * rel(i)$$

Here, N denotes the total relevant items within top-K items. rel(i) is the relevance of an item at the i'th position. For a simple case, it is either 1 (relevant) or 0 (not relevant). MAP@K is just the mean of all the AP@K over all the queries that you may have. It is a better alternative to both the above metrics.

4. MRR

MRR or Average Reciprocal Ranking is the average of the joint rankings of the first priority for each question. Formally written

$$MRR = \frac{1}{Q}\Sigma\frac{1}{rank_i}$$

Here, rank_i represents the rank of the first element of the ith query. This is a very simple guide. It can be very easy to spread. However, it is best for a quote with 1 perfect/correct answer. It may not be a good idea for a scenario with multiple relevant items like e-commerce..

5. R^2 Score

We now turn to a slightly different area of the list that we did not measure directly. Direct evaluation of the list sometimes causes us to miss information such as how close (in terms of confidence) the first and second items of the list are. The coefficient of determination, also known as R^2, is a simple and useful metric for regression-type problems.

$$R^2 = 1 - \frac{\sum_i^N (y_i - \widehat{y_i})^2}{\sum_i^N (y_i - \overline{y_i})^2}$$

Here number represents the number of residuals (correct prediction) and number represents the number of squares (N* variations). R^2 can provide information about how well the model's recommendations match the actual situation or the recommendations of other models.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

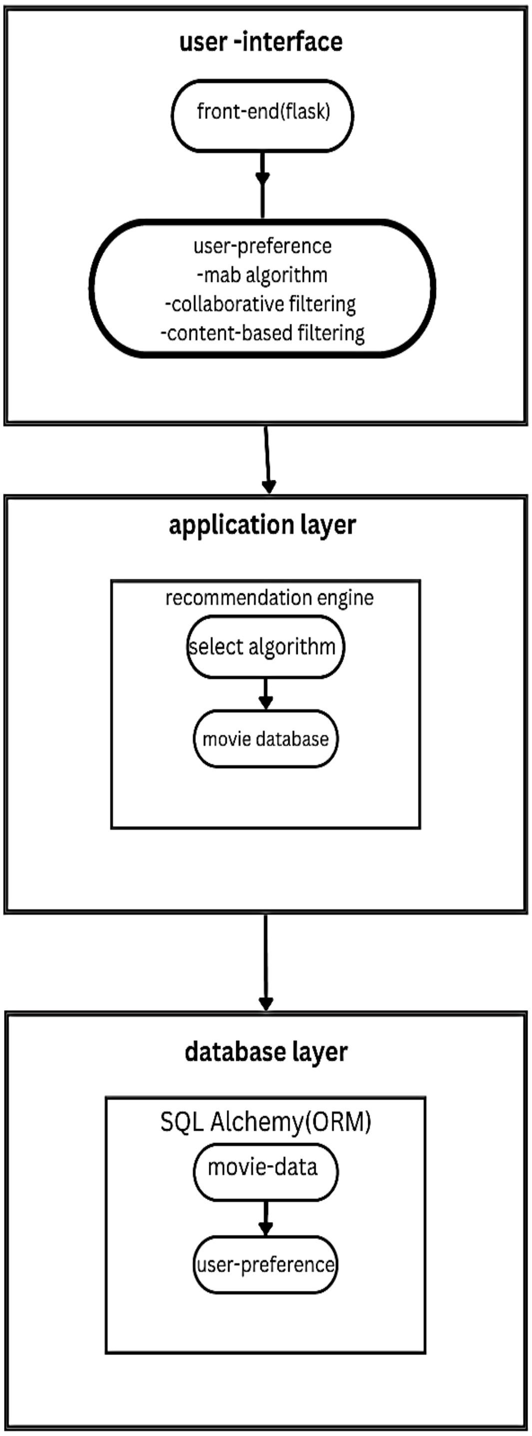


Fig. 6.1 System Design and Implementation

1. **User interface :-** Displays User preferences and the choice of the recommendation algorithm are captured through the user interface. The user can choose between different algorithms like MAB, Collaborative Filtering, and Content-Based Filtering.

2. **Recommendation -engine:** core module that generates recommendations Consists of following components
   1.1 **Bandit algorithms :-** determine the exploration and exploration trade-off and selects which items to recommend
   1.2 **Contextual information:-** Input features that help personalize based on user context
   1.3 **Arms:-** Represents the items and actions that the system can recommend

3. **Database:-** stores historical data, feedback, and item information. User for training and updating the recommendation model
4. **Model-training :-** Periodically updates the recommendation model using historical data stored in the database.

5. **System Configuration:-** Parameters and settings for the overall system, including the bandit algorithm and other components

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

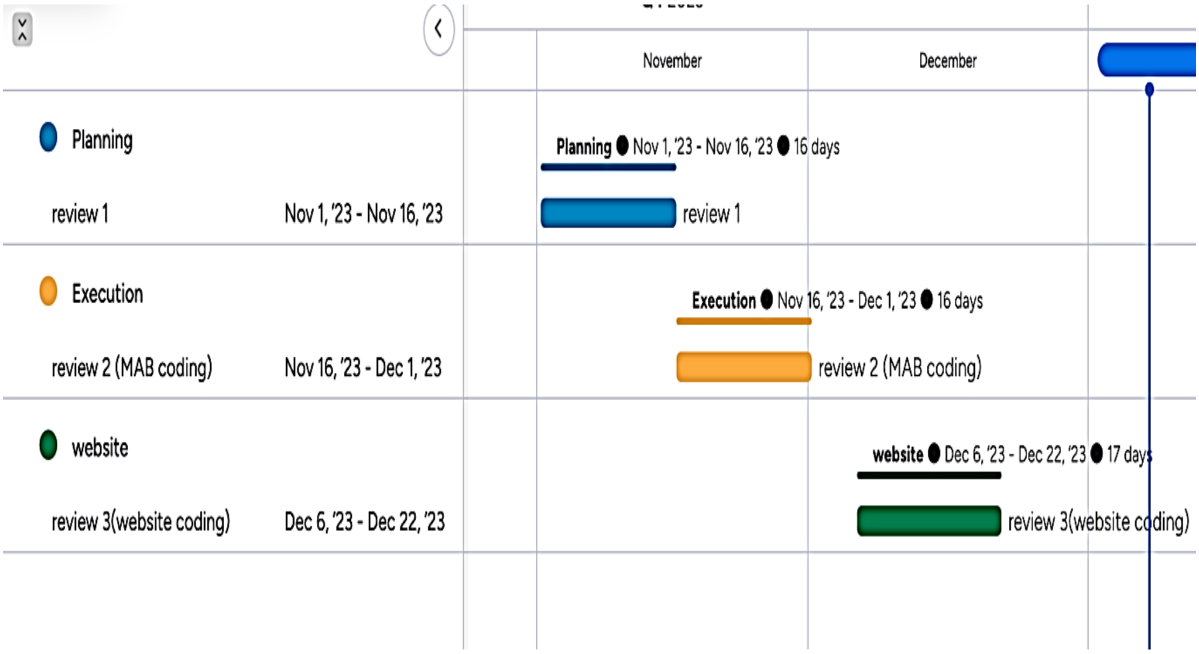


Fig. 7.1 Gantt Chart

CHAPTER-8

OUTCOMES

8.1 Collaborative Filtering:

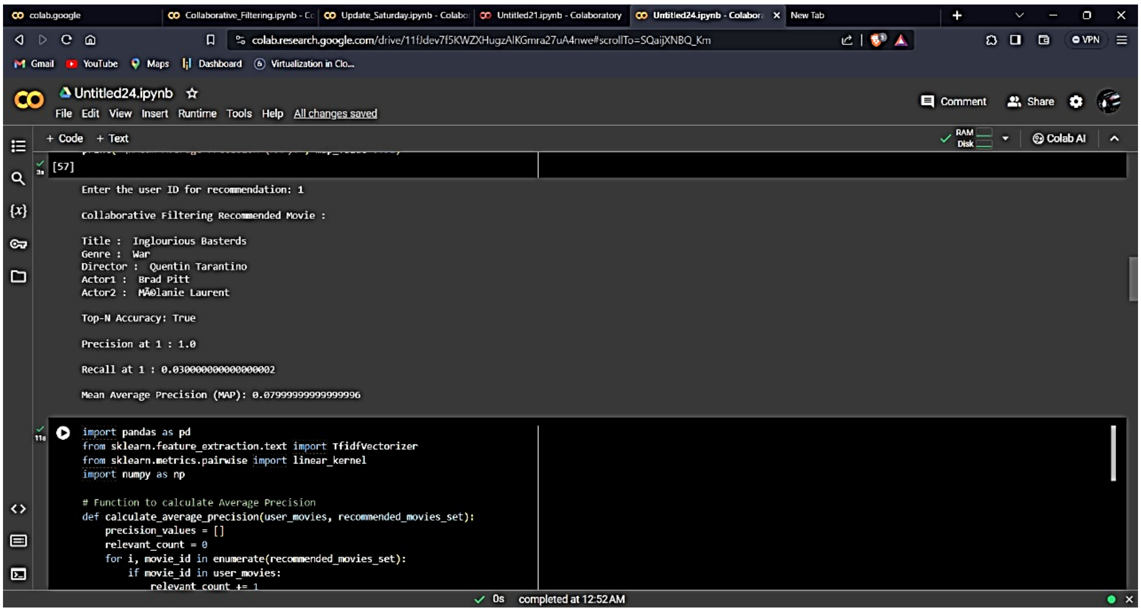


Fig. 8.1 Outcome of Collaborative Filtering

8.2 Content Based Filtering:

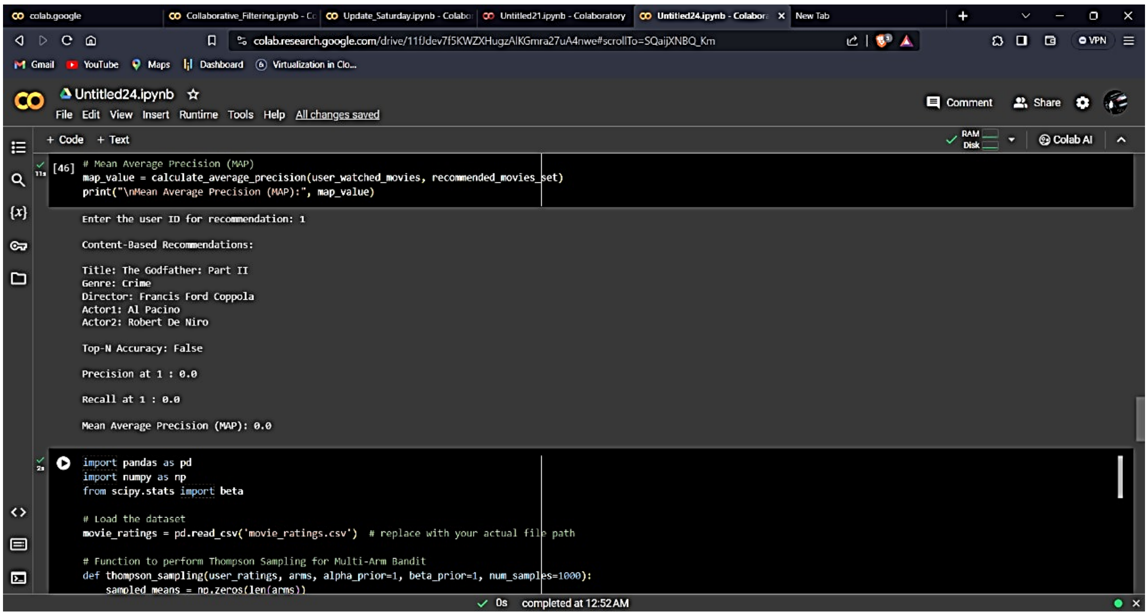


Fig. 8.2 Outcome of Content Based Filtering

8.3 Multi Arm Bandits

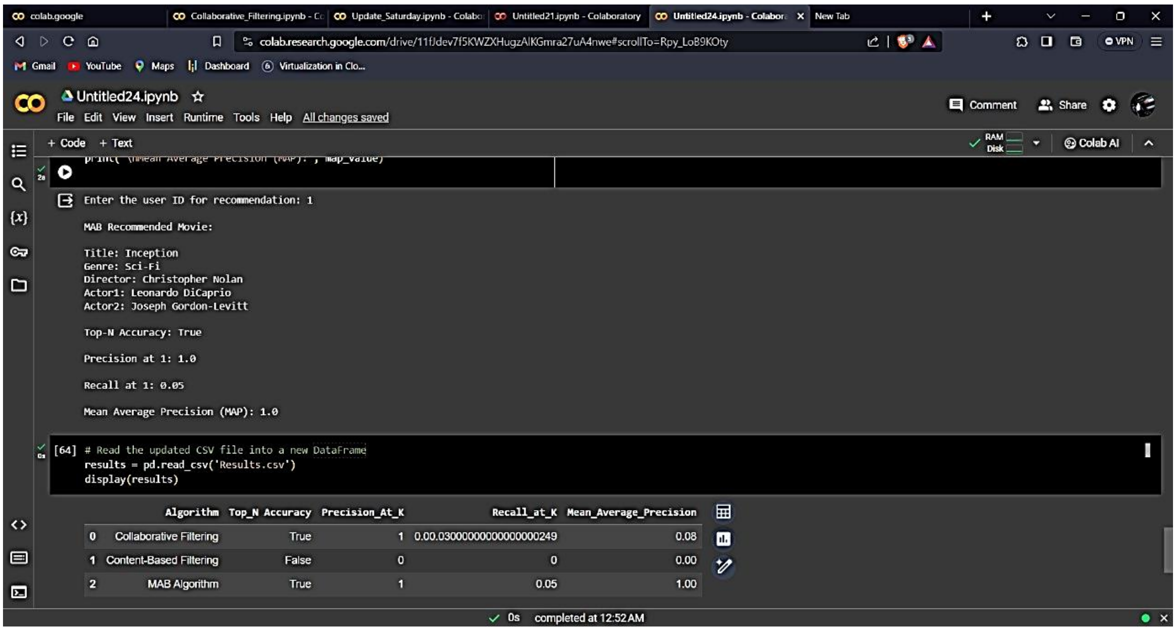


Fig. 8.3 Outcome of MAB Based Filtering

CHAPTER-9

RESULTS AND DISCUSSIONS

	User_Id	Movie_Id	Rating	Movie_Title	Genres	Release_Year	Director	Actor_1	Actor_2	Keyword_1	Keyword_2	Keyword_
0	1	101	4	Inception	Action	2010	Christopher Nolan	Leonardo DiCaprio	Joseph Gordon-Levitt	dream	heist	paralle worl
1	4	102	5	The Shawshank Redemption	Drama	1994	Frank Darabont	Tim Robbins	Morgan Freeman	prison	friendship	redemptio
2	1	103	3	The Dark Knight	Action	2008	Christopher Nolan	Christian Bale	Heath Ledger	vigilante	joker	crim
3	3	104	2	Forrest Gump	Drama	1994	Robert Zemeckis	Tom Hanks	Robin Wright	life-events	love	wa
4	4	105	5	Pulp Fiction	Crime	1994	Quentin Tarantino	John Travolta	Uma Thurman	crime	drugs	gangste

Fig 9.1 Sample Dataset

Dataset has following attributes

'User_Id', 'Movie_Id', 'Rating', 'Movie_Title', 'Genres','Release_Year', 'Director', 'Actor_1', 'Actor_2', 'Keyword_1','Keyword_2', 'Keyword_3']

There are 4 users with user_id 's and 30 movies with movie ID's which are rated at scale of 1-5. For more specific recommendation we have included Genres, Release year, main lead actors , key words to search from the SEO's(Seach engine optimization)

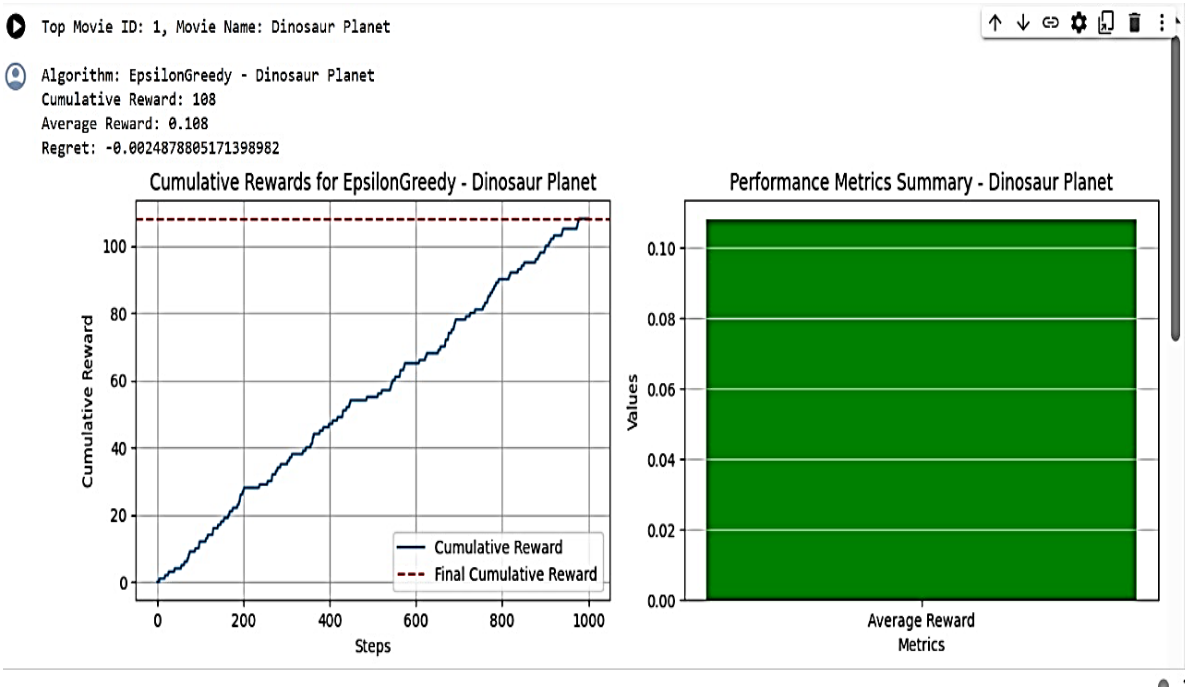


Fig 9.2 Epsilon greedy performance metrics

The image appears to be a line chart showing the cumulative rewards and average rewards for the Epsilon Greedy algorithm applied to the movie "Dinosaur Planet." The chart also includes performance metrics summary for the movie. The x-axis represents the steps or time periods, while the y-axis represents the cumulative and average rewards.

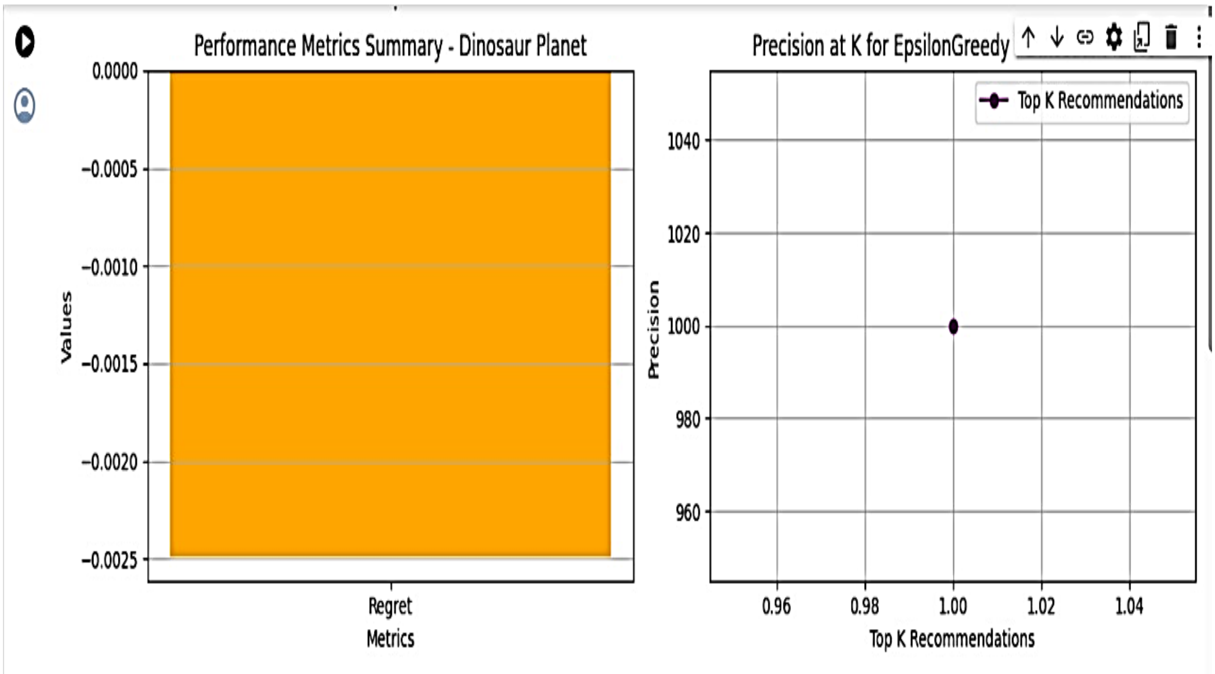


Fig 9.3 Regret and precision for epsilon greedy

The left side of the graph shows a scale with the title "Precision," while the right side of the graph shows a scale with the title "Primary." The graph is primarily orange, with a few black lines and numbers scattered throughout the image.

There are two distinct sets of numbers on the graph, one on the left side and the other on the right side. The left side of the graph has a set of numbers ranging from 0.0 to 0.99, while the right side of the graph has a set of numbers ranging from 0.0 to 1.0. The graph appears to be a representation of precision and primary measurements, possibly related to scientific or technical data

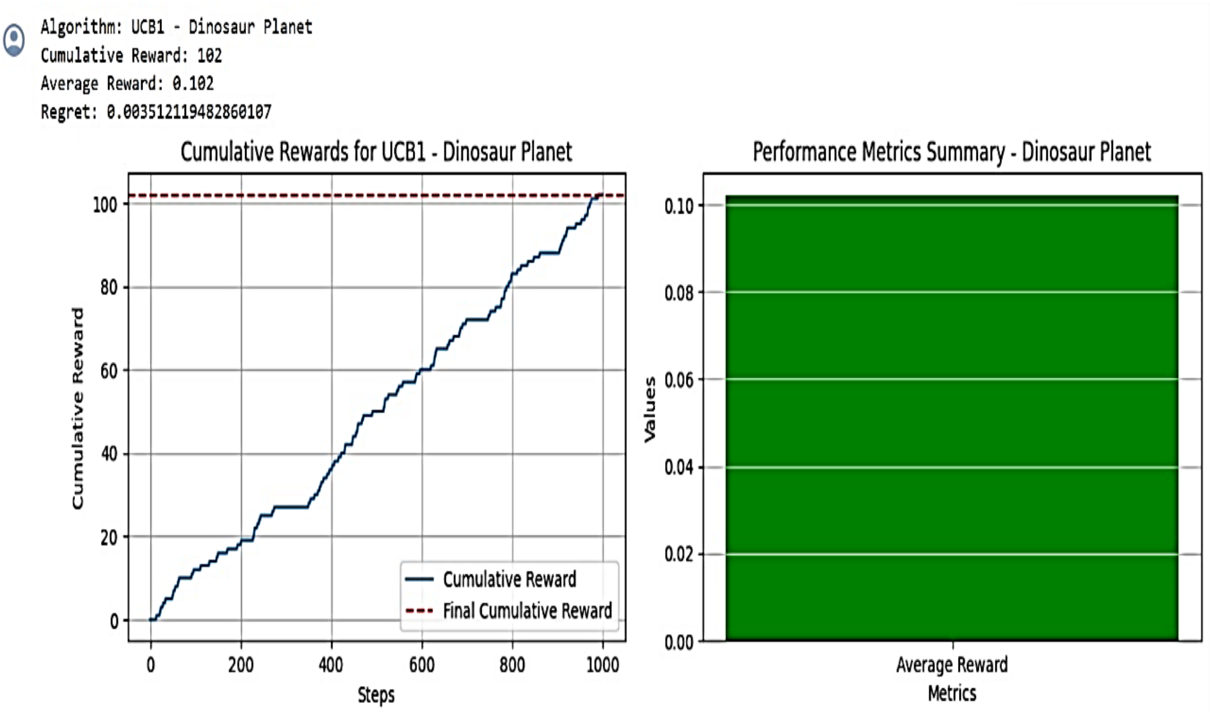


Fig 9.4 UCB1 performance metrics

The image represents a graph with two different lines displayed on it. The first line is a blue line that shows the performance of a UCb1 Dinosaur Planet, while the second line is a green line representing the cumulative reward. The graph is divided into two sections, one for the performance and the other for the reward.

The performance line starts at a low point and gradually increases, while the cumulative reward line starts at zero and gradually rises, indicating that the reward is increasing over time. The graph provides a visual representation of the relationship between the performance and the reward for the UCb1 Dinosaur Planet.

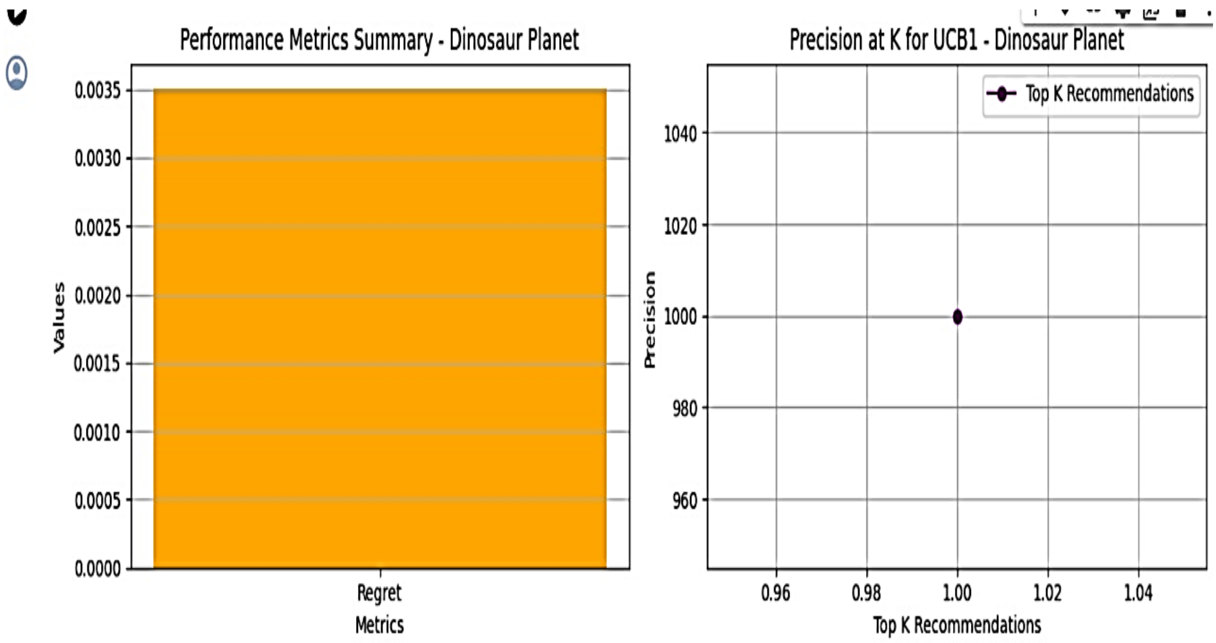


Fig 9.5 Regret and precision for UCB1

The image represents a graph with two different scales, one on the left and the other on the right. The left scale is labeled "Precision" and the right scale is labeled "Recall." The graph displays a series of measurements, with the highest point on the left side and the lowest point on the right side. The measurements are arranged in a linear fashion, with the values gradually decreasing from the top to the bottom.

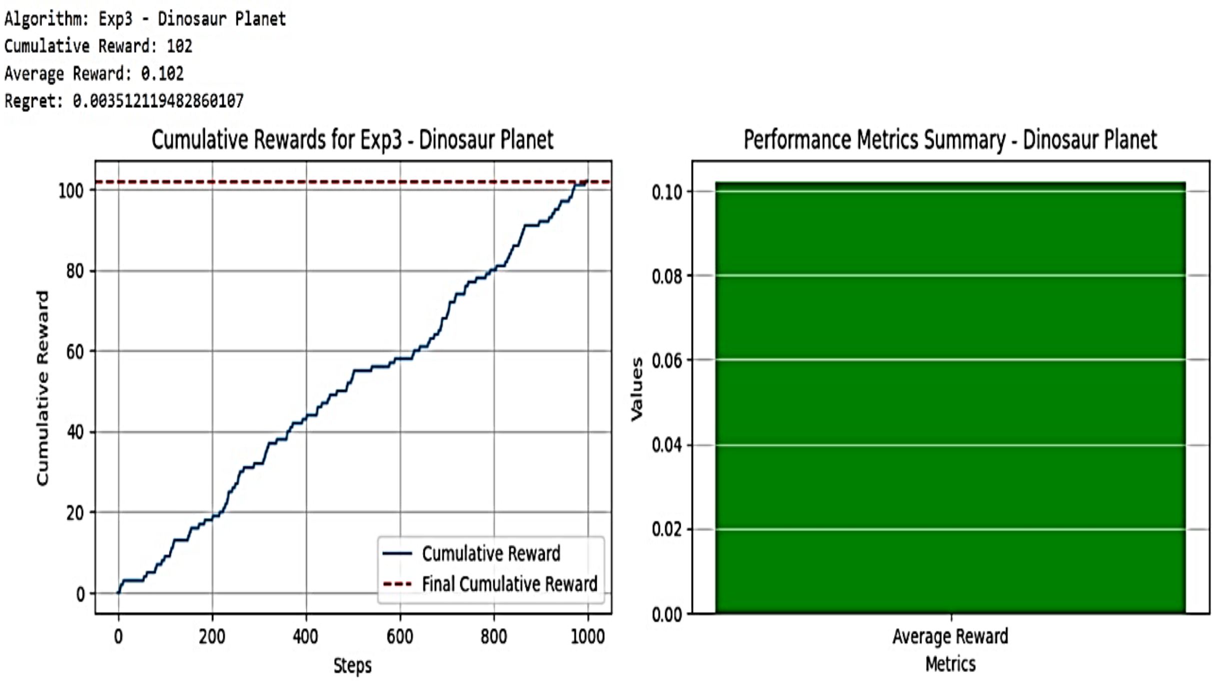


Fig 9.6 Exp3 performance metrics

The figure represents a graph displaying the performance of a dinosaur planet game. The graph shows the total reward earned by the players, with the reward increasing over time. The graph is divided into two parts, with one part showing the cumulative reward and the other part showing the final cumulative reward. There are two lines on the graph, one representing the total reward and the other representing the final cumulative reward. The graph is set against a green background, which adds a vibrant touch to the scene.

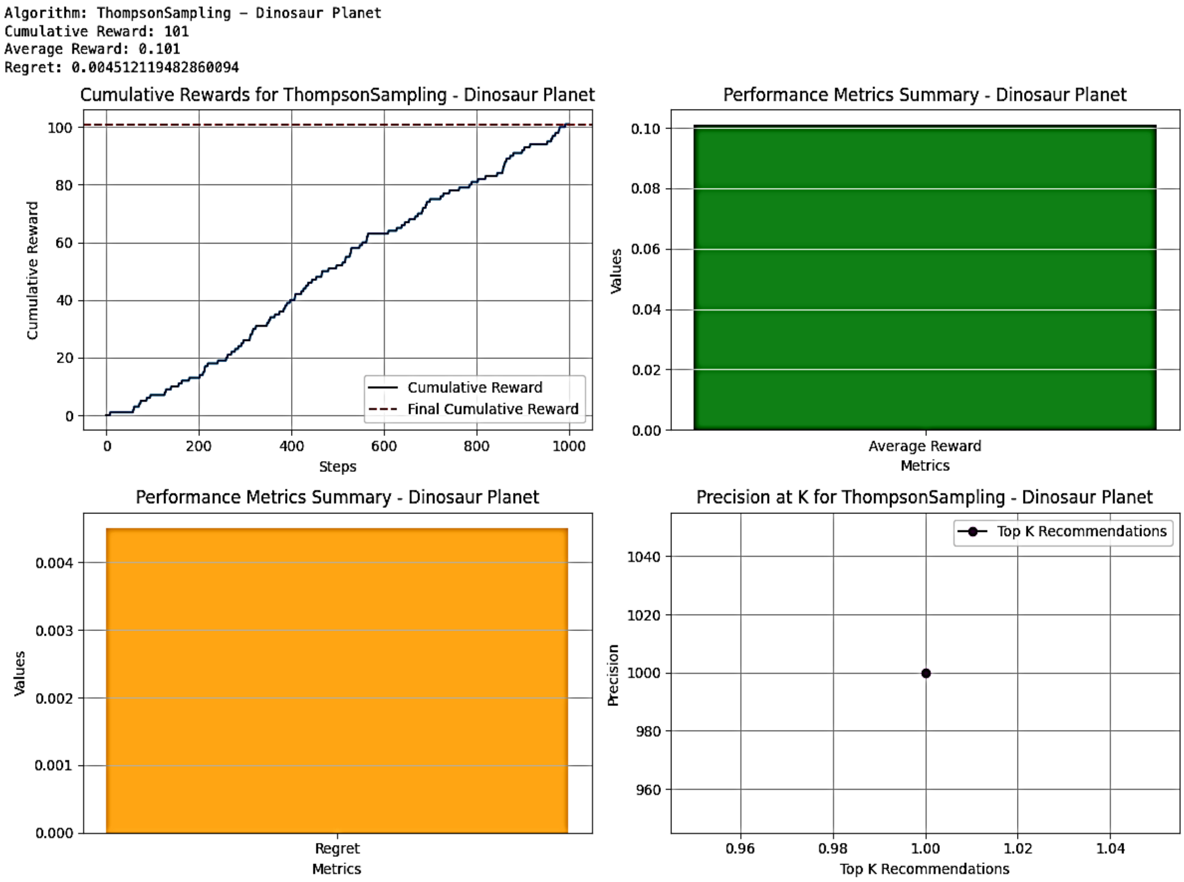


Fig 9.6 Thomson sampling performance metrics

The figure represents a graph displaying the performance of a dinosaur planet game using Thomson Sampling. The graph shows the total reward earned by the players, with the reward increasing over time. The graph is divided into two parts, with one part showing the cumulative reward and the other part showing the final cumulative reward. There are two lines on the graph, one representing the total reward and the other representing the final cumulative reward. The graph is set against a green background, which adds a vibrant touch to the scene. Here we are reading regret and Top K recommendations.

CHAPTER-10

CONCLUSION

Netflix's recommendations use a combination of the search-business complex, including Epsilon Greedy, Softmax Analysis, Thompson Sampling, and Upper Confidence Bound (UCB), which together solve technical problems. Each algorithm has unique strengths, resulting in a combination that attracts and delights users. The combination of these algorithms allows Netflix to strike a balance between discovering new content and using user-generated preferences. Epsilon Greedy's flexibility stabilizes the platform by allocating resources for exploration while adhering to preferences. Softmax Research provides an effective way to show users engaging content by switching between searching for new options and using existing options based on predictions. Yes. At the same time, Thompson Sampling is good at dealing with uncertainty and quickly adapting to uncertainty or unknown user preferences. UCB evaluates uncertainty by favoring options with greater potential while managing risk during the discovery process. This combination allows Netflix to continue to adapt to changing behaviors and attract users. The intelligent system develops real-time recommendations by learning from user interactions to deliver differentiated and personalized content. Netflix leverages these algorithms to create an ever-improving recommendation engine that increases user engagement. The choice of these algorithms depends on many factors such as user interaction goals, flexibility, complexity and platform risk during the research. This combination of algorithms forms the basis of Netflix's recommendations and ensures a good and personalized experience for users who are equally experienced and innovative in their recommendations. Finally, Netflix's innovative use of this process supports an evolving platform that meets the diverse interests of its users worldwide and ensures high user satisfaction and engagement.

REFERENCES

[1] B.Thorat, P., Goudar, R. M., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, *110*(4), 31. https://www.academia.edu/39606345/Survey_on_Collaborative_Filtering_Content_based_Filtering_and_Hybrid_Recommendation_System

[2] Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Lacerda, A.

[3](2017). Multi-Objective Ranked Bandits for Recommender Systems. *Neurocomputing*, *246*, 12–24. https://doi.org/10.1016/j.neucom.2016.12.076

[4] Li, S., Karatzoglou, A., & Gentile, C. (2016, May 31). *Collaborative Filtering Bandits*. ArXiv.org. https://doi.org/10.48550/arXiv.1502.03473

Lu, T., Pál, D., & Pál, M. (n.d.). *Contextual Multi-Armed Bandits*. Retrieved October 1, 2023, from http://proceedings.mlr.press/v9/lu10a/lu10a.pdf

[5] Maheshwarappa, A. (2020, December 18). *Multi-Arm Bandits for recommendations and A/B testing on Amazon ratings data set*. Medium. https://abhishek-maheshwarappa.medium.com/multi-arm-bandits-for-recommendations-and-a-b-testing-on-amazon-ratings-data-set-9f802f2c4073

[6] Mohan, S. (2023, March 23). *A brief overview of the Multi-Armed Bandit in Reinforcement Learning*. Analytics Vidhya. https://medium.com/analytics-vidhya/a-brief-overview-of-the-multi-armed-bandit-in-reinforcement-learning-d086853dc90a

[7] Silva, N., Werneck, H., Silva, T., Pereira, A. C. M., & Rocha, L. (2022). Multi-Armed Bandits in Recommendation Systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications*, *197*, 116669. https://doi.org/10.1016/j.eswa.2022.11666

[8]Xiang, D., West, B., Wang, J., Cui, X., & Huang, J. (2021, August 19). *Adaptively Optimize Content Recommendation Using Multi Armed Bandit Algorithms in E-commerce*. Arxiv.org. https://arxiv.org/pdf/2108.01440.pdf

APPENDIX-A

PSUEDOCODE

1.Epsilon-greedy:

The epsilon-greedy algorithm is a popular approach used in reinforcement learning and multi-armed bandit problems. It balances the exploration of different options with the exploitation of the best-known option. Here's a pseudocode representation of the epsilon-greedy algorithm:

Psuedo code:

```
Initialize action-values Q(a) for all actions a

Initialize the number of times each action was chosen N(a) for all actions a

Initialize the value of epsilon (exploration rate) between 0 and 1


For each time step t:

    With probability epsilon:

        Choose a random action

    Otherwise:

        Choose the action with the highest estimated action-value: argmax_a(Q(a))


    Take action A_t and observe reward R_t

    Update N(A_t) = N(A_t) + 1

    Update Q(A_t) = Q(A_t) + (1/N(A_t)) * (R_t - Q(A_t))
```

In this pseudocode:

- Q(a) represents the estimated value of taking action "a."

- N(a) represents the number of times action "a" has been chosen.

- epsilon is the exploration rate, which determines the probability of choosing a random action.

This pseudocode outlines the basic steps of the epsilon-greedy algorithm, where the agent balances exploration and exploitation by choosing between random actions and actions with the highest estimated value.

2.Softmax Function:

The softmax function is commonly used in machine learning and reinforcement learning to convert a vector of arbitrary real values into a probability distribution. Here's a pseudocode representation of the softmax function:

Psuedocode:

```
Define the input vector z
Define the temperature parameter tau (optional, default value is 1)


Compute the unnormalized probabilities:
for each element i in z:
    unnormalized_prob[i] = exp(z[i] / tau)


Compute the sum of the unnormalized probabilities:
sum_unnormalized_prob = sum of unnormalized_prob


Compute the normalized probabilities:
for each element i in z:
    normalized_prob[i] = unnormalized_prob[i] / sum_unnormalized_prob
```

In this pseudocode:

- The input vector "z" contains the real values that need to be converted into a probability distribution.

- The temperature parameter "tau" is an optional parameter that can be used to control the "smoothness" of the probability distribution. It defaults to 1 if not specified.

The softmax function computes the unnormalized probabilities by exponentiating each element of "z" and then normalizing these values to obtain a valid probability distribution. The result is a vector of normalized probabilities that sum to 1.

This pseudocode outlines the basic steps of the softmax function, which is widely used in various machine learning applications.

3.Upper Confidence Bound (UCB):

The Upper Confidence Bound (UCB) algorithm is commonly used in the context of multi-armed bandit problems and reinforcement learning. It balances exploration and exploitation by selecting actions based on their estimated value and the uncertainty around those estimates. Here's a pseudocode representation of the UCB algorithm:

Pseudocode:

```
Initialize action-values Q(a) for all actions a

Initialize the number of times each action was chosen N(a) for all actions a

Initialize the exploration parameter c (positive constant)


For each time step t:

  For each action a:

    Compute the upper confidence bound UCB(a):

    UCB(a) = Q(a) + c * sqrt(log(t) / N(a))


  Choose the action with the highest UCB value: argmax_a(UCB(a))
```

Take action A_t and observe reward R_t

Update N(A_t) = N(A_t) + 1

Update Q(A_t) = Q(A_t) + (1/N(A_t)) * (R_t - Q(A_t))

```

In this pseudocode:

- Q(a) represents the estimated value of taking action "a."
- N(a) represents the number of times action "a" has been chosen.
- c is a positive constant that determines the balance between exploration and exploitation.
- UCB(a) represents the upper confidence bound for action "a," which includes both the estimated value and a measure of uncertainty.

This pseudocode outlines the basic steps of the UCB algorithm, where the agent selects actions based on their estimated value and the uncertainty around those estimates, allowing for effective exploration and exploitation.

## 4.Thompson Sampling:

Thompson Sampling is a popular algorithm used in the context of multi-armed bandit problems and Bayesian reinforcement learning. It balances exploration and exploitation by sampling from the posterior distribution of action values. Here's a pseudocode representation of the Thompson Sampling algorithm:

**Thompson Sampling:**

Initialize the prior distribution for each action's value (e.g., Beta distribution for binary rewards, Gaussian distribution for continuous rewards)

For each time step t:

For each action a:

Sample a value from the posterior distribution of action value for action a
```

```
        Choose the action with the highest sampled value


    Take action A_t and observe reward R_t
    Update the posterior distribution for action A_t based on the observed reward R_t
```

In this pseudocode:

- The prior distribution for each action's value represents the initial beliefs about the action values before any observations are made. This can be represented using different distributions based on the nature of the rewards (e.g., Beta distribution for binary rewards, Gaussian distribution for continuous rewards).

- Sampling a value from the posterior distribution involves drawing a sample from the distribution that represents the updated beliefs about the action values after observing the rewards.

- Choosing the action with the highest sampled value balances exploration and exploitation based on the sampled values.

This pseudocode outlines the basic steps of the Thompson Sampling algorithm, where the agent selects actions based on sampled values from the posterior distribution, allowing for effective exploration and exploitation while incorporating uncertainty in the action values.

APPENDIX-B

SCREENSHOTS

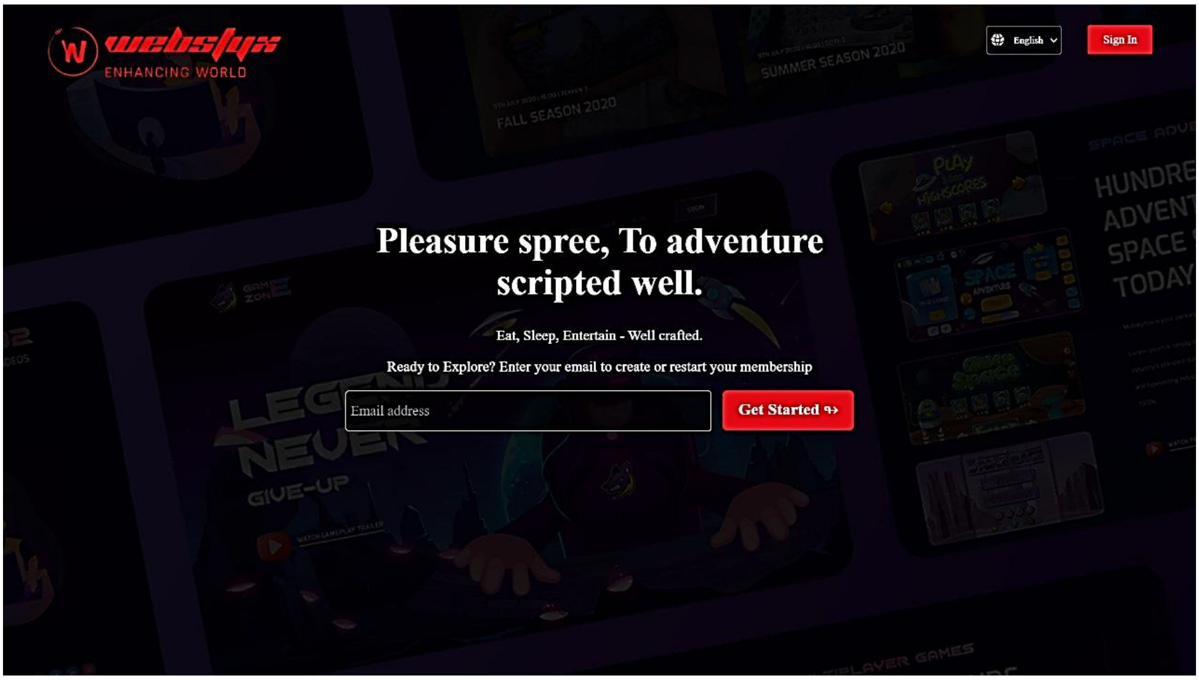


Fig 10.1 Outcome of Frontend - Start

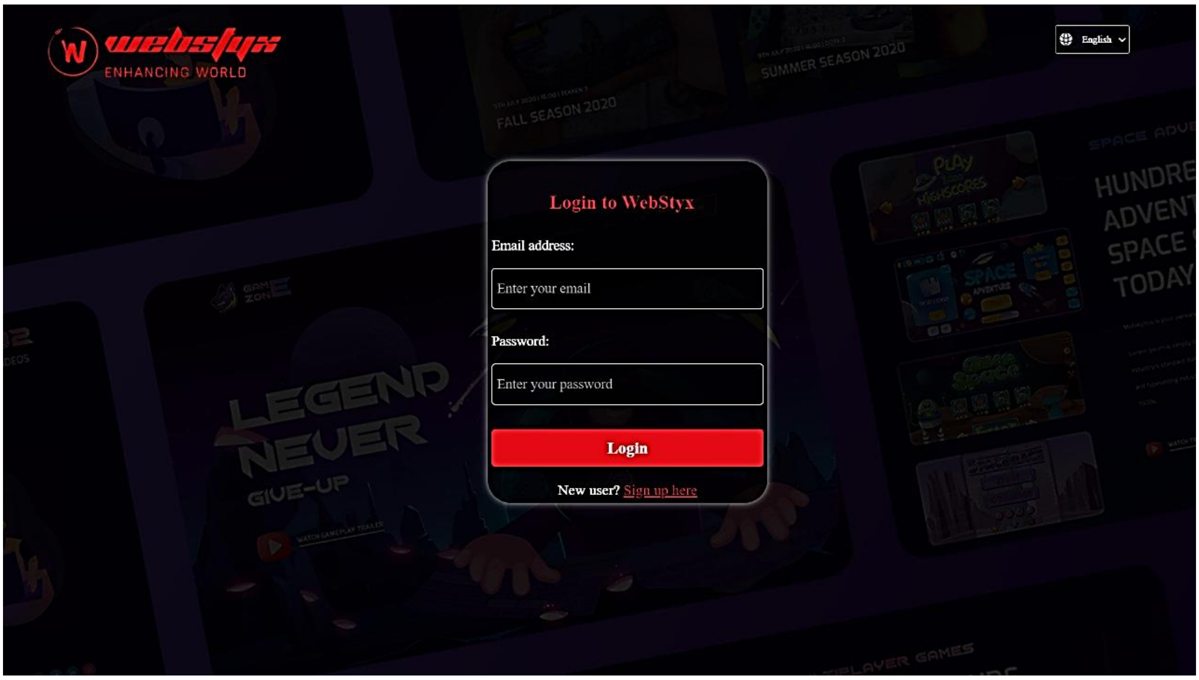


Fig 10.2 Outcome of Frontend - Login

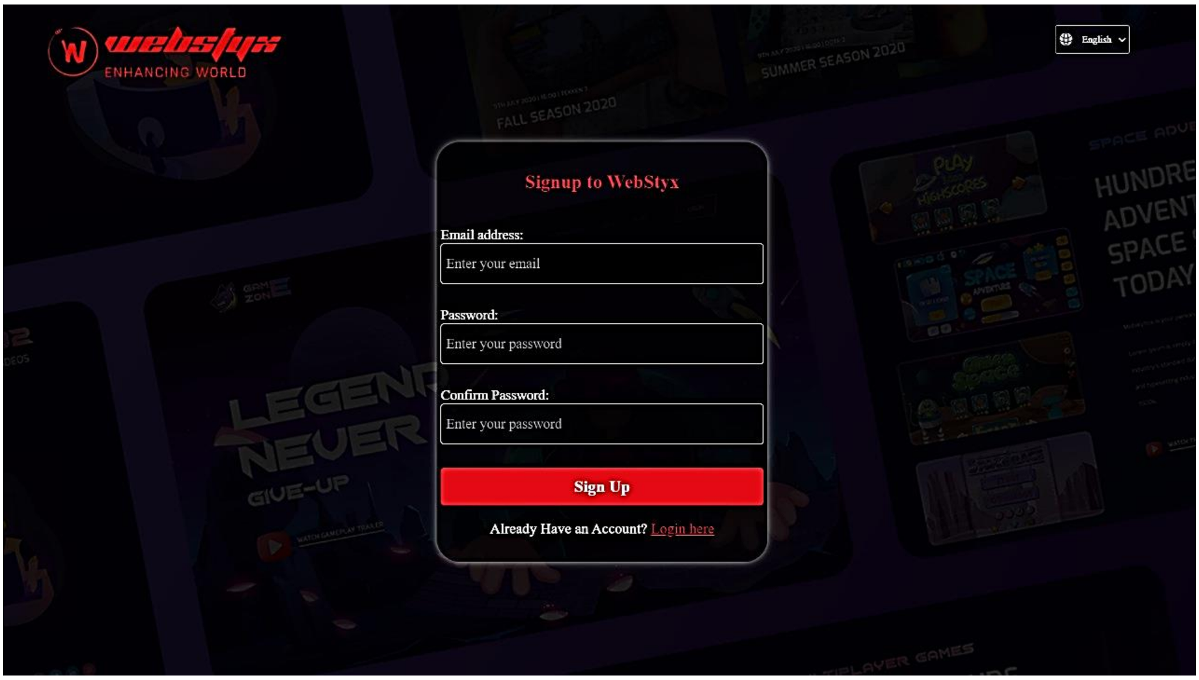


Fig 10.3 Outcome of Frontend – Signup

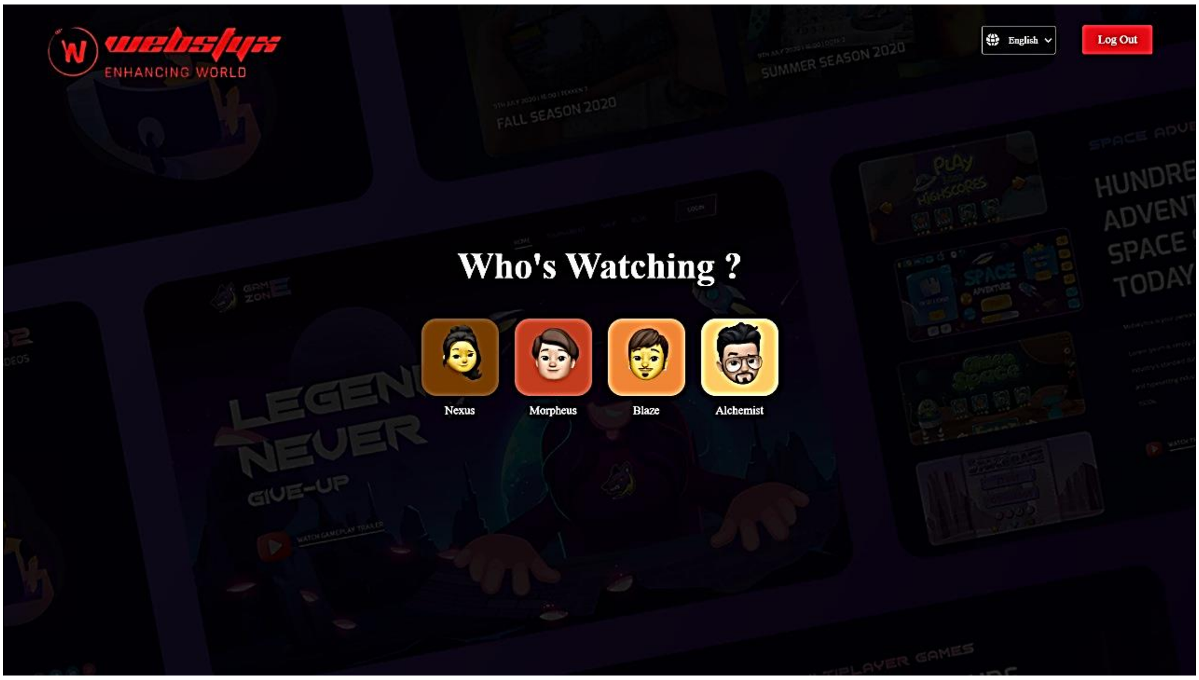


Fig 10.4 Outcome of Frontend – User Choice

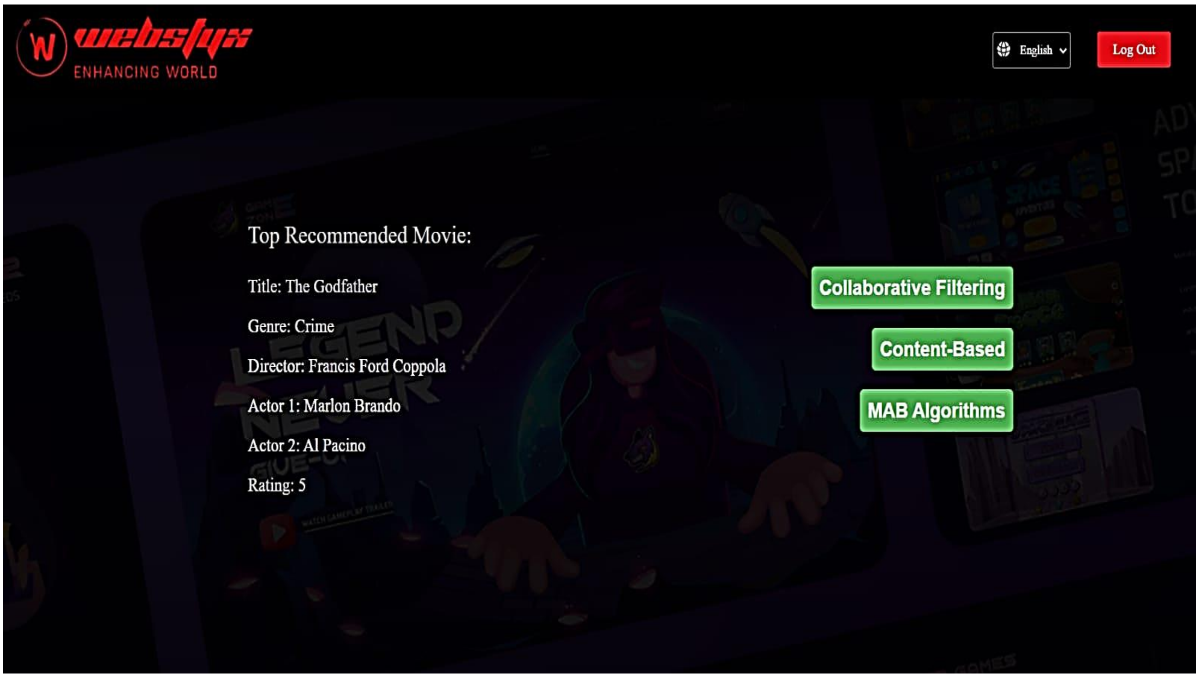
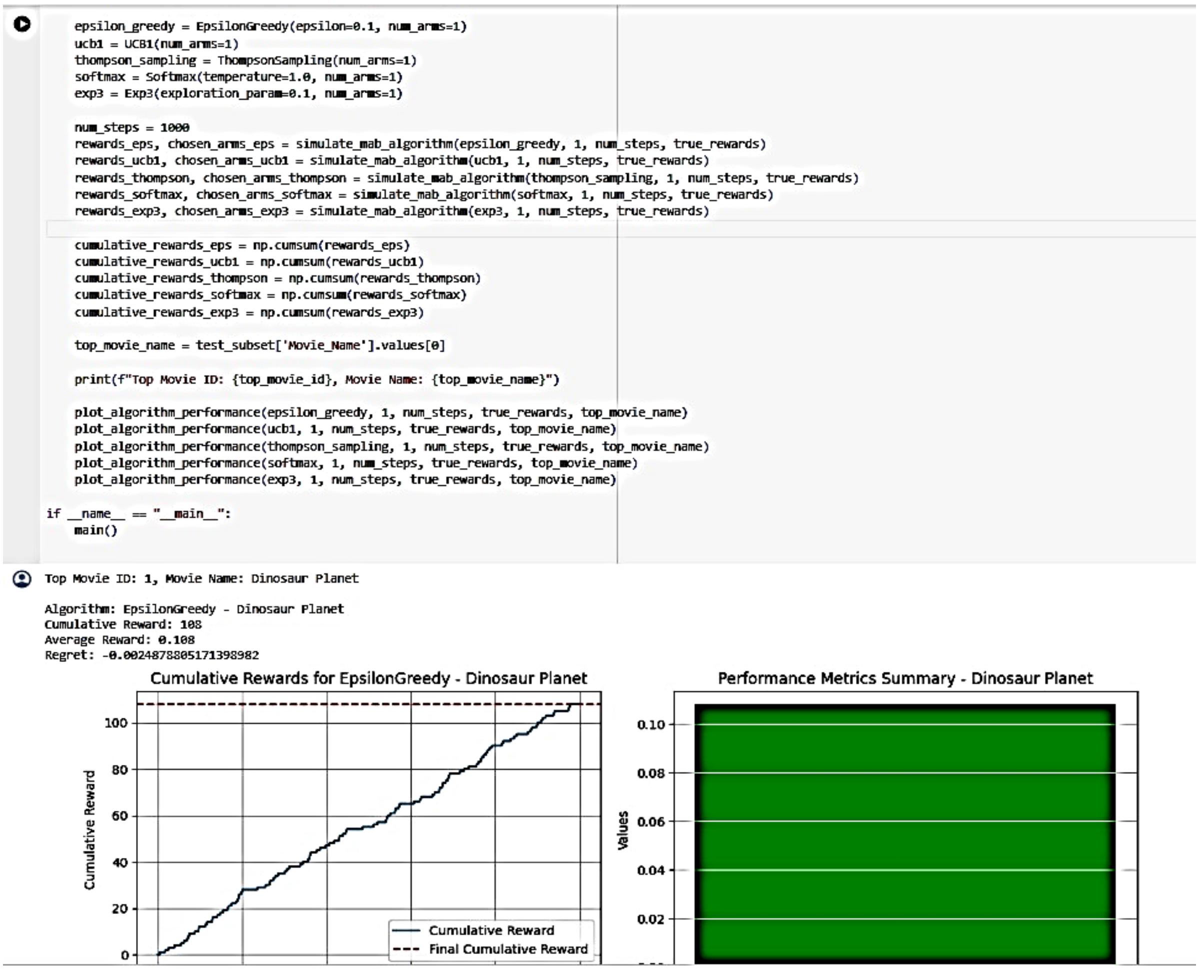


Fig 10.5 Outcome of Frontend – Movie Recommendation



```
    epsilon_greedy = EpsilonGreedy(epsilon=0.1, num_arms=1)
    ucb1 = UCB1(num_arms=1)
    thompson_sampling = ThompsonSampling(num_arms=1)
    softmax = Softmax(temperature=1.0, num_arms=1)
    exp3 = Exp3(exploration_param=0.1, num_arms=1)

    num_steps = 1000
    rewards_eps, chosen_arms_eps = simulate_mab_algorithm(epsilon_greedy, 1, num_steps, true_rewards)
    rewards_ucb1, chosen_arms_ucb1 = simulate_mab_algorithm(ucb1, 1, num_steps, true_rewards)
    rewards_thompson, chosen_arms_thompson = simulate_mab_algorithm(thompson_sampling, 1, num_steps, true_rewards)
    rewards_softmax, chosen_arms_softmax = simulate_mab_algorithm(softmax, 1, num_steps, true_rewards)
    rewards_exp3, chosen_arms_exp3 = simulate_mab_algorithm(exp3, 1, num_steps, true_rewards)

    cumulative_rewards_eps = np.cumsum(rewards_eps)
    cumulative_rewards_ucb1 = np.cumsum(rewards_ucb1)
    cumulative_rewards_thompson = np.cumsum(rewards_thompson)
    cumulative_rewards_softmax = np.cumsum(rewards_softmax)
    cumulative_rewards_exp3 = np.cumsum(rewards_exp3)

    top_movie_name = test_subset['Movie_Name'].values[0]

    print(f"Top Movie ID: {top_movie_id}, Movie Name: {top_movie_name}")

    plot_algorithm_performance(epsilon_greedy, 1, num_steps, true_rewards, top_movie_name)
    plot_algorithm_performance(ucb1, 1, num_steps, true_rewards, top_movie_name)
    plot_algorithm_performance(thompson_sampling, 1, num_steps, true_rewards, top_movie_name)
    plot_algorithm_performance(softmax, 1, num_steps, true_rewards, top_movie_name)
    plot_algorithm_performance(exp3, 1, num_steps, true_rewards, top_movie_name)

if __name__ == "__main__":
    main()
```

```
Top Movie ID: 1, Movie Name: Dinosaur Planet

Algorithm: EpsilonGreedy - Dinosaur Planet
Cumulative Reward: 108
Average Reward: 0.108
Regret: -0.0024878805171398982
```


Fig 10.6 Code and Result

PLAGARISM CHECK

Recommendation System using MAB

ORIGINALITY REPORT

18%	11%	12%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

5%

★ presidencyuniversity.in

Internet Source

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		