Reinforcement Learning based Recommender Systems for Web Applications: scenarios of Radio and Game aggregators

Universidade Fernando Pessoa

André Batista

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

A thesis submitted for the degree of

Master of Science

2023

Abstract

With the progression of time and the continuous evolution of digital entertainment services such as YouTube, Netflix, Spotify, and online gaming platforms, recommendation systems have become an essential daily tool for users. These systems save users time by analyzing various content, facilitating searches, and suggesting relevant content in a personalized manner. However, the same level of personalization is not consistently found across all media domains, particularly within the radio streaming and gaming sectors.

For radio streaming, users must currently search explicitly for a specific internet radio station's name either through a search engine or a radio aggregator like myTuner. This process can lead to significant time consumption and potential loss of user interest, especially if users are unsure of the type of radio they wish to listen to. A similar challenge is faced in the gaming industry, where an overwhelming array of choices can lead to difficulty in discovery and decision-making for players.

Furthermore, even within aggregators that offer some form of recommendation, a convergence is often observed where the most popular items dominate the top spots. This dynamic makes it challenging to discover lesser-known radio stations or games, resulting in a homogenized user experience.

In response to these challenges, this thesis presents the design, implementation, and empirical evaluation of a recommendation system, specialized in the aforementioned domains. Utilizing machine learning and emphasizing deep reinforcement learning techniques, the system optimizes content suggestions, considering variables such as language, region, and user history, fostering personalized recommendations.

The system has been deployed in two distinct production scenarios, demonstrating promising preliminary results. It exhibits consistent improvement and adaptability over time, reinforcing its practical applicability and effectiveness.

Resumo

Com o progresso do tempo e a contínua evolução dos serviços digitais de entretenimento como o YouTube, a Netflix, o Spotify e plataformas de jogos online, os sistemas de recomendação tornaram-se uma ferramenta diária essencial para os utilizadores. Estes sistemas poupam tempo aos utilizadores, analisando vários conteúdos, facilitando as pesquisas e sugerindo conteúdo relevante de forma personalizada. No entanto, o mesmo nível de personalização não é consistentemente encontrado em todos os domínios de mídia, particularmente nos setores de transmissão de rádio e jogos.

Para a transmissão de rádio, os utilizadores devem atualmente procurar explicitamente pelo nome de uma estação de rádio na Internet, seja através de um motor de busca ou de um agregador de rádio como o myTuner. Este processo pode levar a um consumo significativo de tempo e potencial perda de interesse do utilizador, especialmente se os utilizadores não têm a certeza do tipo de rádio que desejam ouvir. Um desafio semelhante é enfrentado na indústria de jogos, onde uma vasta gama de escolhas pode levar a dificuldades na descoberta e tomada de decisão para os jogadores.

Além disso, mesmo dentro de agregadores que oferecem alguma forma de recomendação, observa-se frequentemente uma convergência onde os itens mais populares dominam os primeiros lugares. Esta dinâmica torna desafiante a descoberta de estações de rádio ou jogos menos conhecidos, resultando numa experiência de utilizador homogeneizada.

Em resposta a esses desafios, esta tese apresenta o design, implementação e avaliação empírica de um sistema de recomendação, especializado nos domínios mencionados. Utilizando machine learning e enfatizando técnicas de deep reinforcement learning, o sistema otimiza sugestões de conteúdo, considerando variáveis como língua, região e histórico do utilizador, promovendo recomendações personalizadas.

O sistema foi implementado em dois cenários de produção distintos, demonstrando resultados preliminares promissores. Exibe uma melhoria e adaptabilidade consistentes ao longo do tempo, reforçando sua aplicabilidade prática e eficácia.

It is with heartfelt gratitude that I dedicate this dissertation to my beloved parents, family and girlfriend. Their constant affection and motivation have been invaluable throughout the duration of this academic journey.

Acknowledgements

I offer my profound gratitude towards Professors José Torres and Pedro Sobral for their crucial role in guiding me through the entire process of working on this dissertation successfully. Notably as well are all other computer engineering course instructors whose teachings filled my mind with knowledge and creativity essential in achieving this research feat as well as those who extended their unwavering support to me, including my family, friends and colleagues.

In addition, I want to show my gratitude to Eduardo Carqueja from AppGeneration for presenting me with this demanding and innovative project.

Lastly, this research work was developed under a collaboration between AppGeneration | digitally powered and Fundação Ensino e Cultura Fernando Pessoa.

Contents

List of Figures

List of Tables

Acronyms

AI *Artificial Intelligence*

ATF *Above-The-Fold*

CBF *Colaborative Base Filtering*

CF *Collaborative Filtering*

CNN *Convolutional Neural Network*

DDPG *Deep Deterministic Policy Gradient*

DNN *Deep Neural Network*

DNNS *Deep Neural Networks*

DQL *Deep Q-Learning*

DQN *Deep Q Network*

DQNS *Deep Q Networks*

DRL *Deep Reinforcement Learning*

GAN *Generative Adversarial Network*

ILS *Intra-List Similarity*

LDA *Latent Dirichlet Allocation*

MAE *Mean Absolute Error*

MDP *Markov Decision Problem*

MF *Matrix factorization*

MSE *Mean Square Error*

PCT *Plausibly Complete Time*

RA *Recommender Agent*

RL *Reinforcement Learning*

RLRS *Reinforcement Learning based Recommender System*

RMSE *Root Mean Square Error*

RNN *Recurrent Neural Network*

RS *Recommender System*

R-CNN *Region - Convolutional Neural Network*

SARSA *State-Action-Reward-State-Action*

SVD *singular Value Decomposition*

Chapter 1

Introduction

Many people might not realize it, but the technology behind recommendation systems plays a big role in our everyday experiences. For example, consider a user exploring a game aggregator platform or a radio aggregator platform. The user's preferences are analyzed and used to recommend games or radio stations that align with their interests, thereby enhancing their user experience. The game aggregator platform might propose new games based on previous preferences and gaming behavior, while the radio aggregator could suggest stations based on past listening patterns and preferred music genres. By 2025, as projected by Statista report from 2021, recommender systems are anticipated to govern a significant portion of online and offline activities, making their prevalence unquestionable.

Laying the groundwork for personalized, intelligent information access was the core objective of recommender systems (Ricci et al., 2011). By designing a framework, rather than a specific technology, almost two decades after its inception, the definition of recommender systems is still dynamic, with an expanding array of technological subsets converging under the term. From collaborative filtering to reinforcement learning, numerous domains have been moulding recommender systems, consequently facilitating their rapid growth.

However, the rapid growth of recommender systems also introduces challenges to existing infrastructures and regulations. From the additional computational load resulting from the analysis of vast amounts of user data, to the rigorous privacy regulations, difficult to enforce in a data-intensive environment, recommender systems still have many hurdles to overcome to continue their evolution. In (Ricci et al., 2011), the authors compiled a list of challenges faced by recommender systems, among which we underscore:

- Personalization: Personalizing recommendations to meet individual preferences and tastes poses a significant technological challenge. Future research should focus on refining personalization techniques to improve the relevance of recommendations.

- Scalability: With the increasing amount of data available, recommender systems can potentially overwhelm existing computational resources, thus raising concerns regarding the scalability of the current cloud computing paradigm for recommender systems.

- Cold Start: The "cold start" problem, where insufficient data about new users or items makes it hard to provide accurate recommendations, continues to be a significant issue for recommender systems. New methods to address this issue are needed.

Most of the processing load and latency in recommender systems stem from the necessity of handling and analyzing vast amounts of data, and then generating personalized recommendations in real-time. The challenge of scalability arises due to the growing volume of user data and the number of items for recommendation.

In the context of reinforcement learning, the balance between exploration (trying out new, potentially rewarding actions) and exploitation (relying on known, rewarding actions) becomes an additional challenge. This balance becomes even more complex as the volume of users and items increase, leading to a more dynamic and volatile environment.

Furthermore, the "cold start" problem — where there's insufficient data about new users or items to provide accurate recommendations — remains a significant issue. New strategies are required to handle this challenge, such as introducing effective initial policies or incorporating additional information beyond user-item interactions.

Given these challenges, we need to investigate new paradigms for the design and implementation of recommender systems. The focus in the following chapters will be on the use of reinforcement learning as a way to dynamically adapt to users' changing preferences and to manage the trade-off between exploration and exploitation effectively.

This thesis will investigate these issues in the context of two real-world recommender systems:

- A radio aggregator platform, where the system should recommend a list of radio stations to a certain country, based on its listenners past listening patterns and preferred music genres. The system should provide these recommendations in real-time.

- A game aggregator platform, aiming to personalize game recommendations based on the user's preferences and gameplay history.

1.1 Problem

Within the domain of digital entertainment, recommendation systems are critical to fostering individualized user experiences. However, these systems confront distinct challenges

when applied across varied sectors, notably within the contexts of radio streaming and gaming.

In the sphere of radio streaming platforms, recommendation systems encounter significant challenges related to the acquisition of both precise and exhaustive data concerning users, alongside the vast repertoire of songs and radio programs. Employing machine learning algorithms, these systems are responsible for interpreting user preferences, aiming to predict their subsequent selections. However, the inherent volatility of musical inclinations and the expansive heterogeneity of content often culminate in recommendations that fall short of optimal. The problem is accentuated by the presence of incomplete or fragmented data, which can potentially lead to reduced platform engagement among users. Compounding this, the capricious nature of musical affinities necessitates these algorithmic models to be inherently adaptive, capable of rapid recalibrations in response to evolving tastes.

Simultaneously, game recommender systems are confronted with a disparate matrix of complexities. The multifaceted domain of gaming, characterized by the abundance of genres, mechanics, and narratives, mandates a comprehensive cognizance of player behaviors and preferences. Beyond the rudimentary task of associating a player with a specific genre, the recommendation paradigm must assimilate multifactorial elements encompassing play style, gameplay duration, and player skill proficiency. A suboptimal game recommendation carries considerable ramifications, potentially undermining user trust and platform engagement. Analogous to the music domain, this challenge is amplified by the dynamic nature of gaming trends, necessitating agile adaptation to a spectrum of gaming behaviors.

Across both these realms, the implications of inaccurate recommendations are profound. Such misalignments can precipitate a decline in user trust and platform loyalty, highlighting the imperative for state-of-the-art, malleable, and precise recommendation systems within the field of digital entertainment.

1.2 Motivation

The motivation behind this theme arises from both personal curiosity and professional aspirations. From a personal perspective, I am intrigued by technologies and areas that are rapidly evolving and gaining prominence, notably artificial intelligence and machine learning. The significance of these technologies in my daily life, from streaming platforms to e-commerce, navigation system suggestions, and even ad placements, only heightens my interest. Additionally, this research direction affords me an opportunity to delve into reinforcement learning, a subject I touched upon minimally during my undergraduate studies, thereby expanding my knowledge base for future projects.

On the professional front, an exciting opportunity emerged to integrate cutting-edge

technologies, specifically recommendation systems and reinforcement learning, into the core product of my current company, Country Radios Website. This integration aims to enhance the quality of content presented to users, thereby boosting user retention. Such improvements are vital to ensuring the project remains competitive and relevant in the market.

1.3 Objectives

The objectives of this dissertation reside in the following topics:

- To study and develop state-of-the-art knowledge in machine learning and reinforcement learning, particularly focusing on recommender systems.
- Develop an innovative and efficient recommendation system, empowered by reinforcement learning, for two distinct applications: a game aggregator website and a radio aggregator website. This system should be capable of providing personalized game and radio station suggestions, adapting to users preferences, interests, as well as changes in consumption patterns over time.
- Integrate the developed recommender system into the existing platforms for both applications. The implementation should ensure that the system is user-friendly, visually appealing, functional, and contributes to an enriched and efficient user experience.

1.4 Contributions

Recommender systems, while being an integral part of the online experience, are still very much under constant development, especially with the addition of reinforcement learning mechanisms. With the introduction of this learning approach, few systems have been fully implemented and evaluated in real-world environments, thus offering a wide area for exploration and contribution.

To add to this development, this dissertation provides a comprehensive exploration of a recommender system architecture focused in reinforcement learning. A detailed architecture of the recommender system is presented, adopting a top-down approach, specifying every layer involved. Two distinct real-world applications are implemented to evaluate the efficacy of the presented architecture, namely a game aggregator website and a radio aggregator website.

The methodology, the datasets used, and the reinforcement learning algorithm implemented is thoroughly documented, enabling future research in this area. Therefore, the main contributions of this dissertation are distinguishable in the following areas:

- Specification of a detailed recommender system architecture, enhanced using deep reinforcement learning, that can be applied to distinct application scenarios.
- Implementation and evaluation of the proposed solution to one of the core products of AppGeneration company, the regional radio websites app myTuner Radio, with the objective of recommending only less popular radios, by considering the top played radios as context.
- Implementation and evaluation of *Reinforcement Learning based Recommender System* (RLRS) in a game aggregator website, taking as context the previous games played by our users, to bolster user retention.

1.5 Document Structure

This document is divided into six chapters; Introduction, State of the Art, Architecture Specification, Application Scenarios, System Evaluation and finally Conclusions.

The first chapter introduces the topic of the thesis, along with the reasons behind conducting this research and a detailed explanation of the problem being addressed. The next chapter presents a review of work on recommender systems with a specific focus on those that have integrated reinforcement learning techniques.

Chapter three describes an architecture for a recommender system that incorporates reinforcement learning to meet requirements. The fourth chapter delves into an in depth exploration of two real world application scenarios: a game aggregator website and a radio aggregator website.

The fifth chapter covers how the recommender system was implemented in these scenarios and concludes by evaluating its performance. Lastly in the final chapter conclusions are drawn from the research conducted and possibilities for advancements, in reinforcement learning based recommender systems are discussed.

Chapter 2

State of the art on recommender systems

2.1 Recommender Systems

Over time, we have witnessed an unprecedented boom in internet usage and its integral role in our quotidian. This surge has been accompanied by an overwhelming volume of data being generated continuously, leading to an ever-expanding diversity of information. Consequently, users often find themselves inundated and overwhelmed, struggling to sift through and select information that best aligns with their needs.

Recommender systems have emerged as a beacon in this sea of information. They are designed to analyze and filter content, molding it to fit individual user preferences. In doing so, these systems craft a more personalized digital journey for users. They act as a bridge, connecting large corporations with vast user bases to individual users, ensuring a tailored and enhanced user experience (Adomavicius and Tuzhilin, 2005). This personalization ensures that users are presented with custom content that aligns with their preferences and needs.

As defined by (Jannach et al., 2010) and (Ricci et al., 2011), a recommender system is a software tool adept at suggesting items of interest to its users, refining its recommendations as it learns from their preferences over time.

Think of recommender systems as digital compasses, guiding users through vast and intricate information landscapes. These systems, as emphasized by (Konstan and Riedl, 2012), serve as precise filters. They meticulously sieve through extensive datasets, extracting pieces of information that align seamlessly with a user's profile. Their dynamic nature allows them to adapt based on a user's behavior, interests, and preferences (Pan and Li, 2010). The true strength of these systems lies in their predictive capabilities: they can astutely anticipate a user's preferences, forecasting their likely inclinations towards specific items.

2.1.1 Recommendation Process

The recommendation process, as detailed by (Isinkaye et al., 2015), is a structured journey comprising three distinct phases:

- Information Gathering Phase
- Learning Phase
- Prediction/Recommendation Phase

Initially, the system focuses on acquiring user-centric data, constructing a comprehensive user model, and leveraging diverse feedback channels. This foundation paves the way for the learning phase, where algorithms discern patterns in user behavior. Finally, the system delves into pattern exploration, crafting tailored recommendations and anticipating user preferences for future interactions. The success of a recommender system is deeply rooted in its design and understanding of user behavior, with each phase playing a pivotal role in shaping user perceptions and enhancing the overall experience.

2.1.2 Recommendation Filtering Techniques

In the domain of recommender systems, threre are several methods to address the intricacies of information or content filtering. (Aggarwal et al., 2016) have thoroughly examined approaches such as Collaborative, Content-Based, Hybrid, Knowledge-Based, and Demographic techniques. This study will concentrate predominantly on three of these methods, which have garnered substantial attention due to their widespread application and recent advancements, as underscored by (Liu et al., 2019).

Collaborative Filtering

Collaborative filtering is a technique that recommends items to users based on the preferences and behaviors of similar users. The underlying principle is that users who have agreed in the past tend to agree again in the future about the preference for certain items (Kumar and Thakur, 2018). This method is particularly effective for recommending items that are hard to describe with content attributes, such as movies or music (Isinkaye et al., 2015). Collaborative filtering can be categorized into:

Item-based: This approach identifies items that are similar to those the user liked, based on past user-item interactions. The similarity between items is often computed using methods like cosine similarity or Pearson correlation (Sarwar et al., 2001).

User-based: User-based collaborative filtering predicts a test user's interest in a test item based on rating information from similar user profiles ((Breese et al., 1998), (Herlocker et al., 1999)). The main challenge here is dealing with scalability and the dynamic nature of users (Koohi and Kiani, 2016).

Content-based Filtering

Content-based filtering selects items based on the correlation between the content of the items and the user's preferences ((Van Meteren and Van Someren, 2000), (Burke, 2002)). The underlying principle is that if a user expressed interest in a particular item, they are likely to appreciate similar items. Unlike collaborative filtering, content-based filtering operates independently of other users' profiles, focusing solely on the individual user's preferences and past interactions, this allows it to overcome collaborative filtering challenges (Isinkaye et al., 2015).

Hybrid Filtering

Hybrid filtering methods integrate various recommendation techniques with the aim of enhancing system optimization and mitigating the shortcomings inherent to standalone recommendation systems (Burke, 2002). By leveraging the strengths of both approaches, hybrid methods can overcome the limitations inherent to each individual method.

2.1.3 Recommender Systems Evaluation Metrics

Prediction Accuracy

The metrics of predictive accuracy serve as a quantitative measure to assess the exactitude of forecasted ratings in comparison to the authentic ratings assigned by users (Ziegler et al., 2005). These metrics are instrumental in evaluating the fidelity and reliability of the predictions made by recommendation systems in replicating the actual user ratings.

MAE (Mean Absolute Error)

The *Mean Absolute Error* (MAE) is a fundamental metric utilized to assess the quality of recommendations in a recommendation system. It quantifies the average absolute discrepancy between the predicted ratings and the actual ratings conferred by users. Let (p) represent the predicted rating and (r) represent the actual rating, the MAE is calculated as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n}$$

An MAE of zero indicates impeccable accuracy, signifying no deviation between the predicted outcomes and the users' actual ratings. Consequently, a lower MAE value denotes a higher accuracy of the recommendation system's predictions.

MSE (Mean Squared Error)

Mean Square Error (MSE) is another pivotal metric for evaluating the performance of recommendation systems. It is discussed in detail by (Katarya and Verma, 2017), where various algorithms, including matrix factorization, are evaluated based on MSE among other metrics. Instead of utilizing the absolute error, the MSE employs the squared difference to mitigate the impact of negative signs. The MSE is computed using the following formula, where (p) represents the predicted rating and (r) represents the actual rating:

$$\text{MSE} = \frac{\sum_{i=1}^{n}(p_i - r_i)^2}{n}$$

A lower MSE value is indicative of predictions that are closely aligned with the actual values, signifying an enhanced performance and accuracy of the recommendation system.

RMSE (Root Mean Squared Error)

Root Mean Square Error (RMSE) is a prevalent metric utilized in evaluating the efficacy of algorithms. The RMSE is particularly suitable for prediction tasks as it comprehensively measures inaccuracies across all ratings, irrespective of whether they are positive or negative, thereby emphasizing larger errors more significantly. RMSE can be given by the square of subtraction with the predicted and identified results summation up to n times. Then dividing the result and then raised to power half. The RMSE is also defined as the square root of the mean squared error (Katarya and Verma, 2017). Given (pi,j) as the predicted rating for user (i) regarding item (j), and (ri,j) as the actual rating, with (K) representing the set of hidden user item ratings, the RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum i = 1^{n}(pi, j - r_{i,j})^2}{n}}$$

A lower RMSE value is indicative of a recommendation system that is more adept and accurate in mirroring the actual user ratings.

Coverage

Coverage is an essential metric in recommender systems, emphasizing the breadth of recommendations. Jesse, Bauer, and Jannach (2022) discuss the significance of coverage in the interactive evaluation of recommender systems, highlighting its role in assessing recommendation quality (Šafařík et al., 2022). The Coverage metric, often referred to as the non-accuracy metric, gauges the proportion of data the model leverages relative to the entire dataset that could have been employed (Ziegler et al., 2005).

ILS (Intra-List Similarity)

Intra-List Similarity (ILS) is a metric employed to quantify diversity within recommendation lists. Jesse, Bauer, and Jannach (2022) emphasize the importance of the details in the implementation of ILS, suggesting that it can be a good proxy for human perceptions of diversity when appropriately configured (Jesse et al., 2022). The ILS is computed using the following formula:

$$\text{ILS} = \frac{\sum k \in L \sum j \in L, j \neq k c(k, j)}{2}$$

where (c(k,j)) denotes the degree of similarity between two items in the list, ranging between 0 and 1, based on specific predefined characteristics. In particular instances, cosine similarity has been utilized to calculate this similarity.

The intra-list diversity is then determined as follows:

$$D = 1 - \left(\frac{\text{ILS}}{N}\right)$$

Category Diversity

Metrics that measure precision although important often fail to capture the aspects of user satisfaction in recommender systems. They primarily focus on the accuracy of the recommended item list disregarding other factors that contribute to user contentment. To address this limitation, a nuanced approach known as category diversification aims to improve the usefulness and relevance of recommendation lists by offering a range of options. By doing so, it aims to transcend mere accuracy, fostering a more comprehensive and satisfying user experience (Ziegler et al., 2005). This method emphasizes the significance of diversifying recommendations to cater to users multifaceted interests and preferences ultimately enhancing the effectiveness and appeal of *Recommender System* (RS).

While the techniques mentioned above have proven effective in situations there is a growing interest in exploring dynamic and adaptive methods. Among these approaches *Reinforcement Learning* (RL) stands out as a promising approach. In RL the system learns how to make recommendations by interacting with its environment and receiving feedback, this enables more adaptive and personalized suggestions. Delving deeper into the recommendation processes, the next section will focus on the integration of reinforcement learning strategies, highlighting their benefits and challenges within recommendation contexts.

2.2 Reinforcement Learning

Initially introduced by Sutton and Barto in 1998, RL has seen extensive application and remarkable success across a broad range of areas and uses. Unlike supervised learning, which works with pre-labeled datasets and scheduled training to extrapolate or generalize reactions to unforeseen environmental changes, a Reinforcement Learning algorithm takes a more vibrant approach. Actively interacting with its environment, it accumulates data which subsequently informs its decision-making procedures and forecasts (Sutton and Barto, 2018).

The inherent aim of this technique, echoing human learning methods, is to optimize rewards while minimizing risk. This resemblance has further boosted Reinforcement Learning's increasing prominence (O'Doherty et al., 2015). The methodology uses a trial-and-error process, with rewards and penalties shaping behavior, akin to human learning aspects.

The ultimate goal of Reinforcement Learning is to increase an agent's rewards by executing a series of actions in response to a dynamically changing environment. As Figure 2.1 illustrates, it involves a complex interaction between Five key elements:

- **Agent:** The agent is the learner or decision-maker that interacts with the environment, continuously making decisions to achieve the best possible outcome. It learns and adapts based on the consequences of its actions, with the ultimate goal being to maximize the cumulative reward over time.

- **Environment:** This represents the external context or system with which the agent interacts. The environment responds to the agent's actions by presenting a new state and issuing a reward or penalty, guiding the agent's learning and adaptation process.

- **State:** The state represents the current situation or context in which the agent operates. It provides the agent with the necessary information to make informed decisions and select an appropriate action.

- **Reward:** The reward is the feedback received by the agent following each action. It serves as a scalar signal that indicates how well the agent is performing, influencing the agent's future decisions.

- **Action:** Actions are decisions made by the agent that influence the environment. They are the means through which the agent interacts with and influences its surroundings. The collective set of all possible actions available to the agent is known as the action space, and it can consist of discrete or continuous options.

Therefore, Reinforcement Learning distinguishes itself through its proactive, interactive, and adaptive approach to learning in unfamiliar environments.

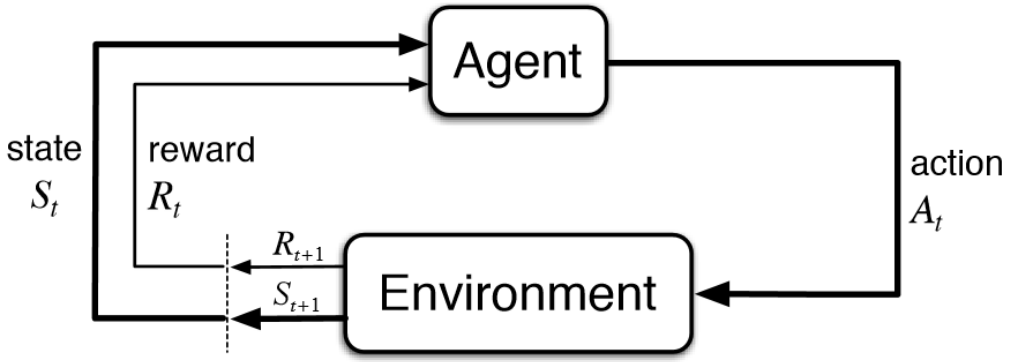


Figure 2.1: Reinforcement Learning Diagram

In the context of recommendation systems, Reinforcement Learning has the potential to enhance the recommendations offered to users. Achieving this could lead to increased revenue, not only by fostering user engagement but also by extending the user interaction duration on the page, potentially leading to increased ad revenue.

Numerous recommendation systems follow a fixed strategy, ignoring the ongoing interaction between the recommendation systems and the users. They also often focus on immediate feedback, neglecting long-term rewards (Liu et al., 2019). This is where RL can make significant enhancements to recommendation systems. It's worth noting that Reinforcement Learning scenarios do not always assure positive rewards. For example, an agent traversing a maze may encounter negative rewards at every stage, thereby necessitating an expedient exit strategy. Several other practical applications demonstrate similar contexts, including self-driving cars, recommendation systems, ad placements on web pages, or directing an image classification system focus (Géron, 2019).

Incorporating research like that of (Zheng et al., 2018), which considers user return or retention as supplement for reward metrics, RL can not only refine strategies through iterative user interactions (Zhao et al., 2019b) but can also recognize shifts in user preferences and identify tactics that optimize future benefits. This consideration of user retention significantly enhances the capability of recommendation systems to achieve more long-term user engagement and benefits.

Various algorithms fall under the domain of Reinforcement Learning, including but not limited to Q-Learning, *State-Action-Reward-State-Action* (SARSA), *Deep Q Network* (DQN), and *Deep Deterministic Policy Gradient* (DDPG)).

2.2.1 Q-Learning

Q Learning, as described by Sutton and Barto (2018) is an off-policy Reinforcement Learning algorithm that allows agents to achieve performance, by exploring and learning from the consequences of their actions, without requiring them to construct domain

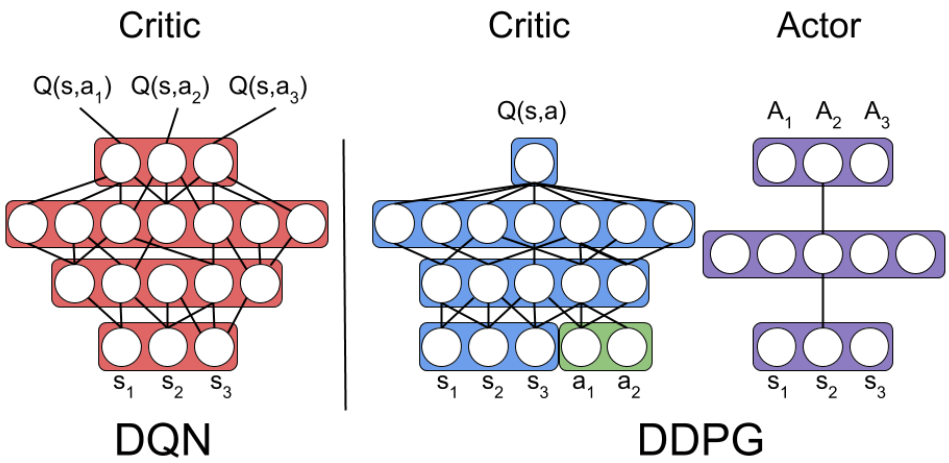


Figure 2.2: DQN and DDPG Architecture Comparison. Image source: `https://www.linkedin.com/pulse/ddpg-dqn-which-use-ridhwanul-haque/`

maps. Q Learning is widely used because it can evaluate the expected usefulness of actions without relying on a model of the environment. This algorithm allows the discovery of an optimal action for any terminal state of the *Markov Decision Problem* (MDP) (Mnih et al., 2013).

Reinforcement Learning framework consists of an agent, a set of states and a set of actions for each state. Q learning utilizes a Q matrix, where rows correspond to the set of potential states and columns correspond to the set of possible actions. When the agent takes an action it transitions from one state to another. The agent then receives a reward based on how that action performed in that specific state.

Updating the values in the Q matrix involves the use of Bellman equation (Puterman, 1994) which considers two inputs: state (s) and action(a):

$$Q_{new}(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \tag{2.1}$$

In this equation:

- s_t represents the current state and s_{t+1} signifies the subsequent state.
- a_t is the current action and a_{t+1} is the next action.
- α is the learning rate, defined between 0 and 1.
- r is the reward obtained when the agent moves from state s_t to state s_{t+1}.
- γ is the discount factor, also defined between 0 and 1.

The discount factor γ establishes the importance given to future rewards. A smaller discount factor de-emphasizes future rewards, focusing more on the present, while a larger discount factor accords more significance to future rewards.

The learning rate α signifies to what extent the new information replaces the old. A lower rate means the agent learns more slowly, while a rate closer to one denotes faster learning.

At each time-step t, when an agent selects an action a_t, it receives a reward r and enters a new state s_{t+1}, leading to the updating of Q. This process continues until learning is ceased. Consequently, the Q-Matrix gets updated, and the Q-value function is maximized. It's crucial to note that Q-learning is an iterative algorithm and thus assumes an initial condition before the first update.

2.2.2 DQN

One of the challenges that arise with Q Learning is its ability to scale when applied to Markov Decision Processes (MDPs) that have a vast number of states and actions.

To address the scalability and approximation challenges in Q-Learning, researchers have proposed using function approximation techniques. The goal is to find a function, denoted as $Q_\theta(s,a)$, which acts as an approximation for Q-Value of any state action pair (s, a). This approach is known as Approximate Q-Learning (Singh and Sutton, 1996). n the past it was practice to use combinations of manually designed features extracted from the state in order to estimate Q Values. However, in (2013), DeepMind demonstrated that *Deep Neural Networks* (DNNS) offer improvements for complex problem domains without requiring explicit feature engineering. By harnessing the representational power of DNNS, *Deep Q Networks* (DQNS) can effectively estimate Q-Values without relying on manual feature engineering, making them highly suitable for addressing complex RL problems (van Hasselt et al., 2016).

Recently, the DQN algorithm has gained recognition for its ability to address challenges posed by high-dimensional, large-scale, and continuous state-space problems. One key advantage of this algorithm is its utilization of neural networks to approximate the Q-value function, resulting in a reduction in the dimensionality of the input data (Lillicrap et al., 2019).

Despite its effectiveness, the DQN algorithm exhibits two main limitations:

- **Slow convergence speed:** This limitation has been well-acknowledged by researchers as a significant area needing improvement (Schulman et al., 2017). The slow convergence hinders the algorithm's ability to reach an optimal solution quickly and efficiently.

- **Inconsistent decision-making process:** This issue often leads the algorithm to make suboptimal recommendations. To overcome this problem, various exploration-exploitation strategies have been explored and analyzed (Fortunato et al., 2018).

2.2.3 DDPG

Deep Deterministic Policy Gradient is an influential reinforcement learning algorithm for continuous control tasks (Lillicrap et al., 2019). It combines insights from earlier work on deterministic policy gradients (Silver et al., 2014) and deep Q-learning (Mnih et al., 2013) to enable model-free learning in high-dimensional action spaces.

At a high level, DDPG learns a policy function and a Q-function critic simultaneously. The policy maps states to actions while the critic evaluates the policy's actions. Key to DDPG´s approach is the use of deep neural networks as function approximators for both the policy and critic. This structure and the interactions between the policy and the critic can be visualized in the block diagram shown in Figure 2.3. This allows DDPG to scale to problems not tractable for traditional reinforcement learning methods.

Several innovations enable stable deep reinforcement learning in DDPG. Target networks, periodically copied from live networks, provide consistent targets during Q-learning (Mnih et al., 2013). Experience replay breaks correlation in the training data and smooths out the training distribution. These mechanisms improve stability compared to standard policy gradient methods.

Empirically, DDPG has achieved state-of-the-art results on challenging physical control tasks. In simulated robotic arm manipulation, DDPG outperformed prior gradient-based policy search and dynamic programming techniques (Lillicrap et al., 2019). However, sample efficiency and tuning remain challenges. Recent work has focused on enhancing DDPG´s stability and data efficiency through additional algorithmic modifications (Fujimoto et al., 2018).

As mencioned by (Tan, 2021), DDPG involves two key innovations adapted from DQN:

- First, it employs an experience replay mechanism, which involves storing a number of recent experiences in terms of state transitions and rewards. These experiences are then randomly sampled in mini-batches to update the action-value function periodically. This mechanism helps to decorrelate the observations, thereby removing correlations in the data and smoothing over changes in the data distribution.

- Second, DDPG uses a target Q network to provide consistent targets during temporal difference backups, this consists in a reinforcement learning method where the value of a state or action is updated based on the difference between its current estimated value and the value of the next state or action, combined with the reward received for taking that action. this approach allows the algorithm to adjust its understanding of the value of an action based on new experiences. By using the target Q network, the DDPG algorithm can provide a stable target for these updates. This network is updated periodically, ensuring a more stable training target. The com-

bination of these two methodologies allows DDPG to learn value functions using large, non-linear function approximators in a stable and robust manner.

The Actor function operates by taking the current state as input and outputs the parameters of a state-specific scoring function. It can be represented in the form:

$$a = \mu(s|\theta^{\mu})$$

Where:

- s denotes the current state
- a represents the current action
- θ^{μ} is the parameter of the actor function

The *Recommender Agent* (RA) then scores all items based on the Actor's output and selects an item with the highest score. The Critic function learns a value function (Q-value), which acts as a judge of how suitable the chosen action is given the current state. The Critic shares the same architecture with the DQN and can be represented as:

$$Q(s,a|\theta^{Q}) = r + \gamma * Q'(s', \mu'(s'|\theta^{\mu'})|\theta^{Q'})$$

Where:

- s represents the current state and s' the next state
- a is the current action
- r is the reward received when the agent moves from s to s'
- γ is the discount factor, defined between 0 and 1
- θ^{Q} and θ^{μ} are the parameters of the critic and actor functions, respectively
- The prime notation (') indicates the parameters of the target networks.

The Critic's assessment informs the Actor's policy parameter updates, guiding the algorithm towards performance improvement in subsequent iterations. This mechanism reduces the redundant computation and is suitable for scenarios with a large action space.

In summary, DDPG pioneered deep reinforcement learning for continuous control. Its modular actor-critic framework and use of insights like target networks and experience replay enabled scalable learning on hard control problems. Despite its many advantages, DDPG does have limitations, including the large number of training steps required to find solutions, which can lead to significant time complexity.

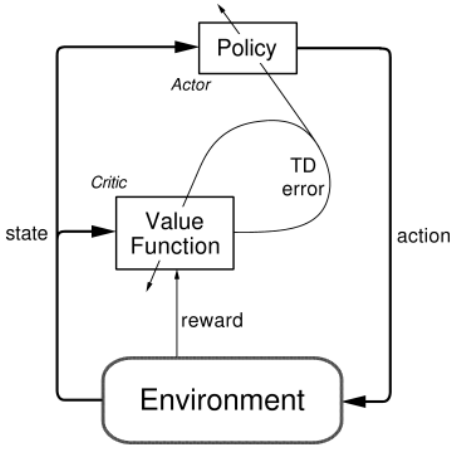


Figure 2.3: DDPG Block Diagram. Image source: `https://github.com/msinto93/DDPG`

2.3 Related Work on Reinforcement Learning

The field of recommender systems has experienced a considerable paradigm shift in recent times due to an increased utilization of reinforcement learning techniques for improving recommendation capabilities. In its 2019 AI research report OpenAI recognizes this development as a crucial trend while highlighting how *Reinforcement Learning* can transform this field completely by adapting and evolving from user interactions. With the help of *Reinforcement Learning*, recommender systems can now provide personalized recommendations that cater to every individuals unique taste, which changes over time. However impressive it may sound *Reinforcement Learning* based recommender systems do incur some challenges regarding computational intensity and data sparsity - hindering their full fledged implementation.

Reinforcement learning has been increasingly explored in recommender systems with the aim of providing personalized and adaptive recommendations (Zhao et al., 2019a). A comprehensive overview of Reinforcement Learning fundamentals was presented by Sutton and Barto in (2018), which highlighted the diverse possibilities of Reinforcement Learning applications. This was further exemplified by (Huang et al., 2021), who introduced a novel end-to-end neural network-based *Collaborative Filtering* (CF) model designed to address long-term interactive top-N recommendations. This model effectively handles the cold-start problem and enhances recommendation accuracy by incorporating additional historical information for the warm-start scenario.

Simultaneously, Deep Learning has emerged as a critical tool in the RS domain due to its efficacy in identifying intricate and nonlinear correlations between users and items (Afsar et al., 2022). Liu et al in (2019) exposed the limitations of traditional methods like collaborative filtering and content-based filtering, and proposed a novel approach to recommendation systems that utilizes *Deep Reinforcement Learning* (DRL). This technique

treats recommendation as a process of sequential decision-making and employs a reinforcement learning method to model the interactions between users and recommendation systems. This approach is capable of considering both dynamic adaptation and long-term rewards, which aligns with the objectives of (Huang et al., 2021) model.

Zhao et al in (2018b) also recognized the constraints of conventional methods and presented a novel recommendation system with the ability to consistently enhance its strategies during interactions with users. They represented the sequential interactions between users and a recommendation system as a MDP, and used reinforcement learning to automatically discover optimal strategies through a system of trial-and-error suggestion of items, and the augmentation of these items via user feedback.

Chen et al in (2023) provided an exhaustive overview of the recent developments in the application of deep reinforcement learning in recommender systems. They delineated the motivation behind applying Deep Reinforcement Learning to recommender systems and classified the current DRL-based recommender systems, summarizing prevalent methods. This survey serves as a crucial resource for scholars and industry professionals, highlighting significant opportunities for future research. It also aligns with the work of (Mooney and Roy, 2000), who designed a book recommendation system rooted in content, which employs information extraction techniques and a machine learning algorithm for text classification. Their work suggests that such a strategy can generate highly accurate recommendations, further emphasizing the potential of reinforcement learning and deep learning in the field of recommender systems.

In (Zhao et al., 2019b), the authors propose a novel recommender system that continuously improves its strategies during interactions with users. The system models sequential interactions as a Markov Decision Process and uses Reinforcement Learning to learn optimal strategies. The authors also introduce an online user-agent interacting environment simulator and validate the importance of list-wise recommendations. The proposed framework, LIRD, is tested on a real-world e-commerce dataset and found to be effective.

Zhao et al in (2018a) discusses the challenges of page-wise recommendations in e-commerce, where users are recommended a page of items and provide feedback, and the system recommends a new page of items. The authors propose a novel page-wise recommendation framework based on deep reinforcement learning, DeepPage, which can optimize a page of items with proper display based on real-time feedback from users. The experimental results based on a real-world e-commerce dataset demonstrate the effectiveness of the proposed framework.

Zheng et al in (2018) proposes a deep reinforcement learning framework for news recommendation. The authors highlight the need for a system that can adapt to dynamic changes in news features and user preferences. They propose a model that explicitly considers future rewards and uses a multi-layer Deep Q-Network to predict rewards based on four types of features extracted from news and users. The model is updated online as the

recommendation agent interacts with users. The authors also propose using the Dueling Bandit Gradient Descent exploration strategy to improve recommendation diversity and avoid harm to recommendation accuracy induced by classical exploration strategies.

Munemasa et al in (2018) presents a novel approach to optimizing store recommendation functions using deep reinforcement learning. The authors propose a method to handle the "curse of dimensionality" problem, which arises due to the enormous number of stores, by expressing actions as continuous values. They utilize *Latent Dirichlet Allocation* (LDA) to convert store information into a distributed representation. The state, created based on browsing history and area information, is also treated as a continuous value. The authors employ the DDPG, known for its high performance with continuous value output, for deep reinforcement learning. The proposed method is compared with bandit algorithms using the delivery record of Ekiten, demonstrating its superiority.

Rojanavasu et al in (2005) presented a distinct approach to web recommendation systems. He proposes a system that leverages reinforcement learning, specifically the ε-greedy policy combined with the SARSA prediction method, to enhance the recommendation process. The system is designed to provide users with opportunities to explore new pages or products that may not be popular but could align with their interests. It comprises two models:

- A global model that captures the behavior of all customers to discern system trends.
- A local model that records individual user browsing history to make personalized recommendations.

The authors conducted experimental studies to evaluate the click rate of the recommendation list, which serves as a measure of the system's effectiveness. The results indicate that the balance between exploration and exploitation is crucial in recommendation systems. Too much exploration might lead to recommendations that do not match the customer's interests, while excessive exploitation could result in users missing out on discovering other products. He suggest that future work could focus on reducing the space required to store global and local states of all users and studying the effect of the weight parameter 'w' to optimize the total Q-value.

In the realm of recommendation systems, a noteworthy approach has been proposed by (Chen et al., 2019) , which leverages a *Generative Adversarial Network* (GAN) to simulate user behavior dynamics and deduce their reward function. This approach is part of a model-based RL framework designed specifically for recommendation systems. To manage the vast array of potential items efficiently, the authors employ the Cascading DQN algorithm, which facilitates the development of a combinatorial recommendation policy. Empirical evidence suggests that this GAN-based user model provides a more accurate representation of user behavior compared to other models. Furthermore, the

RL policy derived from this model appears to yield superior long-term rewards for users and enhances the system's click rate. However, the authors also highlight the challenges associated with applying RL to recommendation systems, such as the indeterminacy of the reward function and the impracticality of model-free RL .

It is crucial to mention (Wang et al., 2023) that introduced RL4RS, an open-source, real-world dataset specifically designed for reinforcement learning-based recommender systems (RL-based RS). This dataset was developed to address the limitations of artificial and semi-simulated datasets previously used in the field. The study also proposed a comprehensive evaluation framework for RL-based RS, which includes environment simulation evaluation, environments evaluation, counterfactual policy evaluation, and test set-based environment evaluation. The RL4RS suite encompasses two real-world datasets, data understanding tools, finely tuned simulation environments, advanced RL baselines, batch RL baselines, and counterfactual policy evaluation algorithms. This resource is anticipated to significantly contribute to the research in applied reinforcement learning and various recommendation scenarios, particularly in the e-commerce industry.

Table 2.1 lays out a qualitative comparison analysis between the related work described in section 2.3. Three distinct parameters were considered. First, area of use relates to the specific application of the recommender system. Secondly, methods used focus on the techniques and algorithms employed by the system. Lastly, scalability relates to the potential of the solution to handle larger and more complex datasets.

In conclusion, recommender systems are a critical component of e-commerce platforms, helping to personalize the user experience and increase sales. Both papers reviewed in this section propose innovative methods using reinforcement learning to improve the effectiveness of these systems. However, further investigation would be needed to fully understand their scalability potential.

Table 2.1: Comparison of related work on Recommender Systems

Paper	Problem Domain	Methods Used	Key Results
(Zhao et al., 2019b)	E-commerce	Reinforcement Learning, Markov Decision Process	Effective on a real-world e-commerce dataset
(Zhao et al., 2018a)	E-commerce	Deep Reinforcement Learning	Effective on a real-world e-commerce dataset
(Zheng et al., 2018)	News Recommendation	Deep Reinforcement Learning, Deep Q-Network, Deep Q-Learning	Success in user retention, tested in a real world product
(Liu et al., 2019)	Generic Recommendation Systems	Deep Reinforcement Learning	Identified drawbacks in traditional methodologies
(Munemasa et al., 2018)	Store Recommendation	Deep Reinforcement Learning, Latent Dirichlet Allocation, Deep Deterministic Policy Gradient	Superior to bandit algorithms in store recommendations
(Rojanavasu et al., 2005)	Web Recommendation	Reinforcement Learning, ε-greedy policy, SARSA	Enhances recommendation process
(Chen et al., 2019)	Generic Recommendation Systems	Generative Adversarial Network, Reinforcement Learning, Cascading Deep Q-Network	Enhances system's click rate
(Wang et al., 2023)	E-commerce	Dataset for Reinforcement Learning-based Recommender Systems	Introduced RL4RS, a real-world dataset for RL-based RS

Chapter 3

System Specification

In the digital age, there has been a significant surge in consumer participation across various platforms, leading to an overwhelming amount of available information. Given the limited time consumers have, it becomes imperative to present products or services tailored to their unique profiles. Traditional recommendation systems, while prevalent in digital environments, often fall short as they perceive consumers as entities with static preferences. This perspective is far from the dynamic nature of real-life consumer preferences.

The primary objective of this research is to address this gap by developing a recommendation system that filters through the information overload and presents consumers with products or services aligned with their evolving interests. Users today are not just seeking options; they desire suggestions that resonate with their unique profiles. Furthermore, to validate the efficacy and practicality of the proposed recommendation system, it is crucial to test it with real users in varied contexts, such as radio and gaming.

This chapter will delve into the methodologies adopted in this research, highlighting the strategies and frameworks that enabled the realization of these objectives. The discussions will encompass the cold start issue in recommendation systems, the architecture of the proposed system, including the DDPG algorithm and features extraction, and the reward design integral to reinforcement learning.

3.1 Proposed Solution

After defining our problem for this study, it was necessary to create a system that would allow us to overcome the defined problems. To this end, an analysis was made of the various existing and explored methods that proved to be successful in these areas. After this analysis, an approach based on deep reinforcement learning, more specifically DDPG, was selected due to its ability to address problems with high-dimensional state spaces and continuous, low-dimensional action spaces (Tan, 2021).

After identifying the problem under study, it was essential to develop a system aimed at solving it. Initially, various methods that could potentially contribute to the study of recommendation systems were analyzed.

After conducting this initial analysis, we ended up adopting a DRL approach, specifically employing the DDPG algorithm. This algorithm was selected due to its proven effectiveness in scenários involving high dimension state spaces and continuous action spaces, as discussed in (Tan, 2021).

DDPG, an actor-critic algorithm, has been particularly successful in various applications, demonstrating its ability to learn policies effectively, making it suitable for our study. The algorithm operates by utilizing two neural networks, one for the actor and another for the critic, which work together to optimize the policy and value functions (Lillicrap et al., 2019).

In the context of recommendation systems, the application of DDPG presents a novel approach. Traditional recommendation systems, such as those based on Q-Learning with tabular Markov chains, have their limitations, such as the inability to handle large state and action spaces effectively. DDPG, with its continuous action space handling (Dulac-Arnold et al., 2015), offers a promising alternative, allowing for more nuanced recommendations.

The DDPG-based model was developed in a controlled environment, enabling a proof of concept regarding its utility in recommendation systems.

3.2 Developed System

The sequence diagram provided in 3.1 illustrates the integration of a recommender system into a web application. The process begins when a user accesses the web application and requests a page. Upon receiving the request, the web server checks for any cached recommendation corresponding to the current state, if none are found, the recommendation system generates a new list of items, these items will then be retrieved and presented to the user through the web application. Each interaction a user has with the recommended items is captured through a post request sent to the server, where it is stored in the database for future reference and analysis. Unlike conventional RLRSs, which typically provide immediate recommendations in response to a user's interaction, our system introduces a strategic delay in this process, This delay varies based on the specific application scenario.

By accumulating more user interactions before generating recommendations, our system allows for a more nuanced and informed evaluation of user preferences and behaviors in different contexts. This approach facilitates the making of recommendations that are more tailored, thoughtful, and relevant to the user's actual needs and interactions under various scenarios.

This architecture allows for seamless integration of the recommender system into the

web application, providing users with personalized recommendations based on their previous interactions and preferences. The use of a database to store user interactions and recommendations allows for continuous improvement of the recommendation algorithm, resulting in more accurate and relevant recommendations over time.

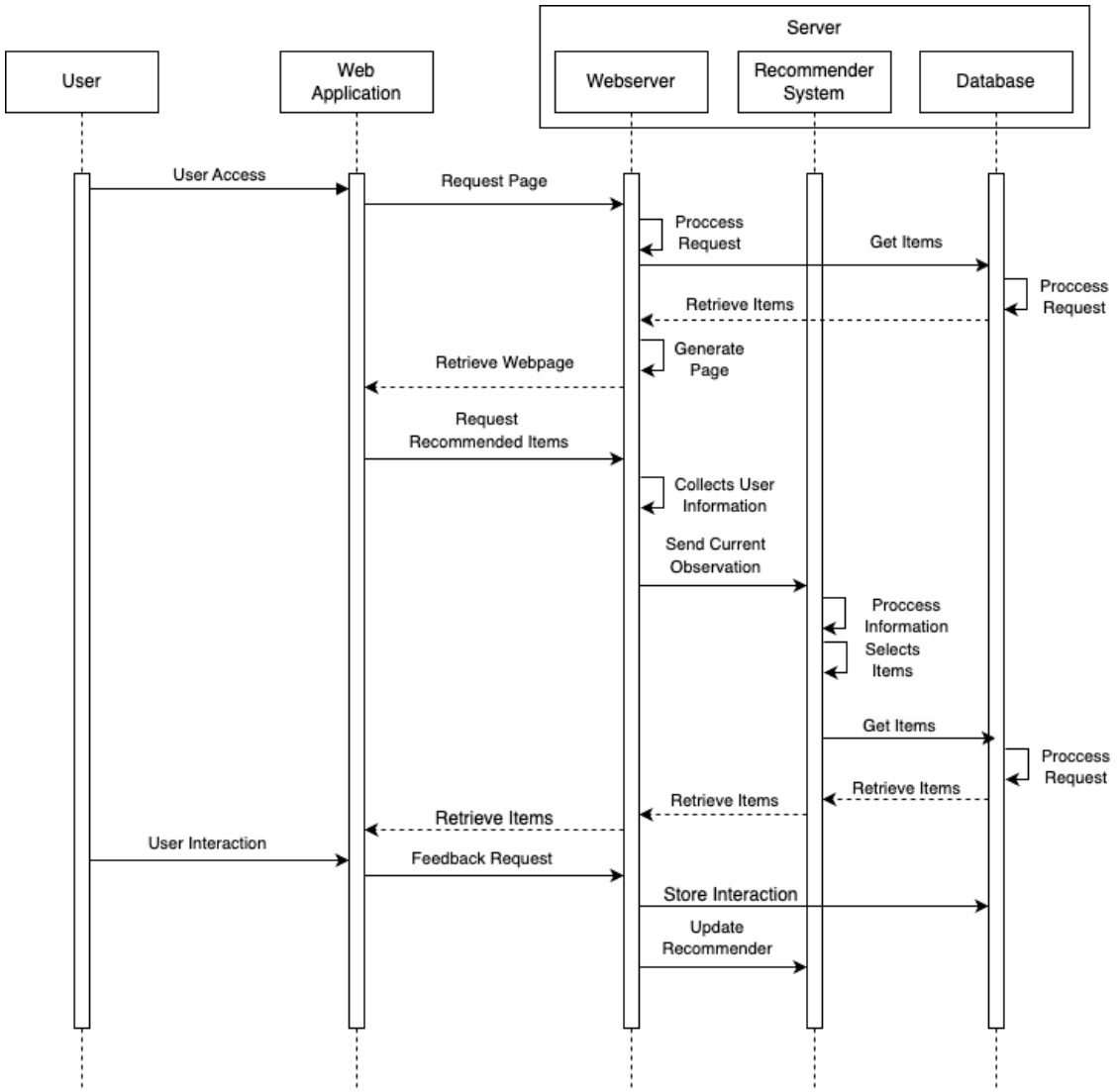


Figure 3.1: Recommendation Process

Recommender System

This subsection provides a detailed exploration of the "Recommender System" block depicted in the sequence diagram (Figure 3.1), this block is represented in detail in (Figure 3.2). The Recommender System is central to our architecture, responsible for generating personalized recommendations for users based on their historical interactions and preferences.

Modeling the Recommendation Task: The recommendation task is modeled as a

Markov Decision Process (MDP), where a Recommender Agent (RA) interacts with the environment (users) over sequential time steps. The objective is to maximize cumulative rewards by making strategic item recommendations that resonate with users' preferences and browsing history.

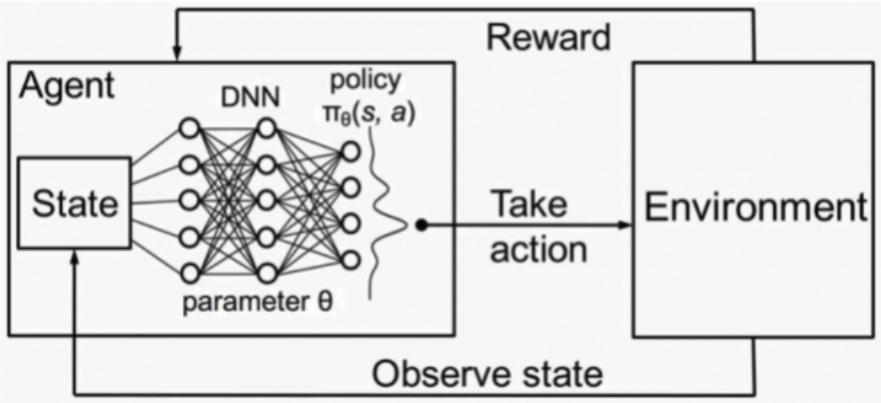


Figure 3.2: DDPG Interaction Flow. Image source: `https://www.mql5.com/pt/articles/3856`

Components of the MDP:

- State space S: A state $s_t = \{s_t^1, s_t^2, ..., s_t^n\} \in S$ is defined as the interactions history of a user, i.e., previous N items that a user browsed before time t. The items in s_t are sorted in chronological order.

- Action space A: An action $a_t = \{a_t^1, a_t^2, ..., a_t^n\} \in A$ is to recommend a list of items to a user at time t based on current state s_t, where K is the number of items the RA recommends to user each time.

- Reward R: After the recommender agent takes an action a_t at the state s_t, i.e., recommending a list of items to a user, the user then browses these items and provides his feedback. He can then skip (not click), click, or play these items, and the agent receives immediate reward $r(s_t, a_t)$ according to the user's feedback.

- Transition probability P: Transition probability $p(s_{t+1}|s_t, a_t)$ defines the probability of state transition from s_t to s_{t+1} when RA takes action a_t. We assume that the MDP satisfies $p(s_{t+1}|s_t, a_t, ..., s_1, a_1) = p(s_{t+1}|s_t, a_t)$. If user skips all the recommended items, then the next state $s_{t+1} = s_t$; while if the user clicks/orders part of items, then the next state s_{t+1} updates.

- Discount factor γ: $\gamma \in [0, 1]$ defines the discount factor when we measure the present value of future reward. In particular, when $\gamma = 0$, RA only considers the immediate

reward. In other words, when $\gamma = 1$, all future rewards can be counted fully into that of the current action.

Deep Reinforcement Learning (DRL) Approach: Utilizing the Deep Deterministic Policy Gradient (DDPG) algorithm, the RA learns to make recommendations by interacting with the environment and receiving feedback in the form of rewards. DDPG, an actor-critic algorithm, is adept at handling continuous action spaces, making it suitable for this recommendation task.

Continuous Improvement: The architecture allows for the continuous enhancement of the recommendation algorithm by learning from user interactions, ensuring that the recommendations remain aligned with evolving user preferences.

3.3 Features Extraction and selection

In recommender systems, using only discrete item identifiers is insufficient for modeling item relationships and properties. An alternative is to construct item embeddings using auxiliary information such as attributes, text, or historical interactions (Wang et al., 2019). In this work, we adopted the latter approach - exploiting user-item interaction history to generate embeddings.

The methodology treats items analogously to words, with a user's sequential interaction history analogized to a textual sentence. Latent semantic connections between items can thus be extracted by applying word embedding techniques on the corpus of historical sessions. As demonstrated by (Wang et al., 2019), this allows learning of dense, low-dimensional vector representations which encode meaningful item properties based on usage patterns.

Specifically, we have employed a pipeline which engineers features like popularity, encodes categorical metadata via one-hot encoding, and extracts semantic vectors from title text via TF-IDF and SVD. The concatenated output is a rich item embedding integrating metadata, text, and collaborative signals. Dimensionality reduction improves efficiency for downstream tasks. This representation learning approach induces embeddings which capture not only attributes but also latent relationships between items emerging from community usage patterns.

In this way, we move beyond simple indexes to a learned semantic vector space capturing nuanced connections between items. Our usage-based technique provides a complementary learning signal to attributes or descriptions. The resulting embeddings can augment existing recommender approaches, providing useful features to better discern and recommend relevant items.

3.4 Cold Start

The cold start problem is a significant challenge in recommendation algorithms. According to (Lika et al., 2014), there are three distinct types of cold start scenarios that can be identified:

- New Users (a): The algorithm lacks historical data or preferences to make accurate recommendations for new users.
- New Items (b): The system doesn't have sufficient user interactions or feedback to accurately gauge the relevance of new items.
- Both Users and Items are New (c): This represents the most complex situation. The algorithm, faces a double challenge since it must navigate the absence of historical data for both users and items, complicating the recommendation process.

To tackle the cold start problem, a multi-faceted approach is proposed. To address recommendations for new users, we recommend items based on their popularity among existing users, leveraging the current trends in the system. As for new items, we devise item embeddings that utilize item metadata, such as descriptions, categories, and sub-categories. By leveraging these metadata-based embeddings, we can establish connections between recently added items and previously existing items, facilitating more informed recommendations. This approach enables the system to make intelligent inferences and mitigate the impact of the cold start problem in recommendation algorithms.

3.5 Reward Design

At the heart of reinforcement learning lies the concept of rewards, which provide a numerical measure of the quality or desirability of an agent's action in a particular state. The agent's objective is to learn a policy that maximizes the expected cumulative reward over interactions. In light of this, the design of the reward system has a big influence the agent's behavior and directing it toward desired results.

Objective

The primary objective in the context of our recommendation system is to guide the recommender agent (RA) to suggest items that the users would find relevant and engaging. Consequently, the reward function should be designed to reflect the quality of recommendations made by the RA.

Components of Reward

The reward function is meticulously designed to encompass various facets of user interactions and engagement, directly tied to the effectiveness of the recommendations. Two pivotal components constitute the reward function:

1. **Click Distribution (CD)**: This component gauges the distribution of user clicks across the array of recommended items. It plays a vital role in ensuring the diversity and balance within the recommended content. A more evenly spread click distribution signifies that users find a broader array of recommendations appealing and engaging, enhancing the overall utility and attractiveness of the recommendation system. This aspect is particularly crucial for ensuring users satisfaction, therefore increasing the probability of return.

2. **Dwell Time (DT)**: This is a measure of the time users spend interacting with a recommended item. A more extended dwell time is indicative of the user finding the recommended content engaging and captivating, which is a positive outcome from a recommendation quality perspective. In the context of web applications reliant on ad revenue, a lengthier dwell time enhances the likelihood of users viewing and engaging with more advertisements, thus boosting potential ad revenue.

Formulation of Reward

Given these crucial components, the reward function $R(s,a)$ for a state s and an action a is strategically crafted as follows:

$$R(s,a) = w_1 \times \mathrm{DT} + w_2 \times \mathrm{CD}$$

In this formulation, w_1, w_2 are weights assigned to calibrate the significance of each component within the reward function. The careful design of this reward function allows a good evaluation of the recommendation system's performance, ensuring that it is not just theoretically sound but also practically aligned with real-world business objectives.

Given the recommendation system described earlier, it's crucial to illustrate its application within different areas - one being a game aggregator and the other being a radio aggregator. Both areas, while distinct in their offerings, provide a unique perspective on the functionality and adaptability of our algorithm. Consequently, this will help us understand the system's versatility and its ability to generate meaningful recommendations in disparate domains. In light of these considerations, a more detailed account of these

applications, their specific features, the challenges encountered, and the ensuing solutions will be exhaustively discussed in 4.

Chapter 4

Application Scenarios

The ongoing attempts to fine-tune definitions, establish patterns, and delineate boundaries, primarily in the theoretical domain, have yet to fully materialize in real-world applications, with only a few systems implementing true reinforcement learning recommender systems (2.3).

Aiming to showcase the key benefits of the reinforcement learning recommender system approach, two distinct application scenarios have been specified:

- Radio Aggregator - A radio station recommendation system that uses reinforcement learning to deliver personalized radio station recommendations. This system takes into account the user's listening history, and geolocation context to provide recommendations that continually evolve to meet the user's changing preferences.
- Game Aggregator - A reinforcement learning enabled, all-in-one gaming platform that recommends games to users based on their past preferences and interactions. This RL-based system utilizes real data of user preferences and game metadata to create dynamic and personalized game recommendations.

These scenarios were designed not just to explore the potential of reinforcement learning in recommendation systems, but also to assess its effectiveness and efficiency in real world products needs and restrictions.

4.1 Self-improving Lists for Radios

In the evolving digital landscape, radio broadcasting continues to hold a significant place despite the proliferation of various music and information platforms. Presently, many radio aggregators, such as myTuner, base their recommendations on simple metrics like location or station popularity. myTuner is a notable platform that aggregates radio stations from all over the world, providing users with a wide array of listening options. It provides

stations recommendations based on regional divisions, such as countries, to tailor the listening experience to localized tastes and preferences.

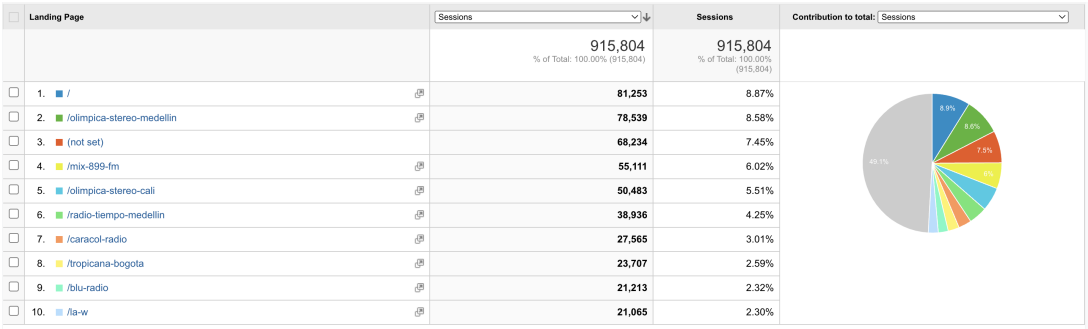
Taking Colombia as example, this regional radio website results from the divion of myTunner, in order to cater to the diverse musical tastes and cultural nuances of its population. For the experiments Colombia was chosen due to the rich diversity in user tastes, offering a broad spectrum of preferences that can be analyzed and catered to. This regional division allows for a more nuanced and tailored recommendation system, promoting a variety of lesser-known broadcasters and fostering a diverse radio ecosystem.

The goal here was to produce a list of radios for each regional website that could improve itself based on user interactions and listen time. The premise is to display lesser-known items to users, kind of a discovery list. This is challenging because we are attempting a one-size-fits-all approach, trying to recommend a list that caters to a large number of people, ensuring exposure to novel and diverse content.

Notably, top-ranking stations or those belonging to large radio groups will be excluded from the recommendation pool to prevent oversaturation. In such a digital ecosystem, a combination of inherent browser features and the broader Matthew Effect inadvertently narrows listeners' choices. This highlights the critical need for innovative recommendation mechanisms that counteract these forces, ensuring a diverse radio landscape for listeners. As radio remains a key medium for music consumption (LUMINATE, 2022), efforts should be directed towards broadening listeners' horizons.

Problem

Listeners exhibit a distinct navigation pattern in this domain. As evidenced in Figure 4.1, approximately 91% of listeners bypass the homepage entirely, directly accessing specific radio URLs. Web browsers autocomplete features, which prioritize previously visited or frequently accessed URLs, can partly explain this behavior. Designed for user convenience, these features inadvertently direct listeners to familiar stations, thus limiting the discovery of new or lesser-known stations on the homepage. Supporting this observation, a study by (Hackett and Parmanto, 2009) highlights that the homepage isn't always the primary entry point for users. Of the 8.87% who do access the homepage, a mere fraction explore beyond the initial listings. This constrains their exposure to a broader array of radio options. This trend aligns with the "Matthew Effect" or the "Rich Get Richer" principle (Merton, 1968), where already popular or frequently accessed radio stations continue to dominate, overshadowing lesser-known stations. The challenge lies in maintaining recommendation relevance over time, especially with the station pool's constraints, a manual curation process would be labor-intensive and not cost-effective due to the fact the human curator has to constantly adapt to regional preferences and tastes.



Landing Page	Sessions	Sessions	Contribution to total: Sessions
	915,804 % of Total: 100.00% (915,804)	915,804 % of Total: 100.00% (915,804)	
1. /	81,253	8.87%	
2. /olimpica-stereo-medellin	78,539	8.58%	
3. (not set)	68,234	7.45%	
4. /mix-899-fm	55,111	6.02%	
5. /olimpica-stereo-cali	50,483	5.51%	
6. /radio-tiempo-medellin	38,936	4.25%	
7. /caracol-radio	27,565	3.01%	
8. /tropicana-bogota	23,707	2.59%	
9. /blu-radio	21,213	2.32%	
10. /la-w	21,065	2.30%	

Figure 4.1: Colombian Users Landing Page Distribution

Objectives

Developing self-improving lists from a set of less used radios that adapt using metadata from user interactions and preferences, taking into account regional nuances. By learning from listener behavior and regional music preferences, the system can tailor recommendations for a region like Colombia. This ensures exposure to a diverse content range while supporting smaller broadcasters, with the system's accuracy and relevance evolving as more data is assimilated. The objective is to enhance listener engagement and satisfaction by presenting them to this curated list of lesser-known stations. Drawing inspiration from the motivational power of games (Zichermann and Cunningham, 2011), the discovery process can be envisioned as a form of exploration game, making the act of finding new stations more engaging.

4.1.1 Radio Database Selection

During our research, we sourced data from an authentic database comprising various radio stations. Following a meticulous selection process, we narrowed down our dataset to a total of n radio stations, where $n = 367$ based on the subsequent criteria:

- The radio station does not belong to a mainstream group. In this context, a "mainstream group" refers to radio stations owned by large corporations that hold multiple stations, often benefiting from broader reach and popularity.
- The radio station is not part of the top k in the country under study, where $k = 60$.
- The radio station has been played successfully for more than t hours within the last d days, where $t = 2$, $d = 15$.
- The inclusion of 10 custom radios, specifically curated and owned by AppGeneration.

Figure 4.2 illustrates that the distribution of these radio stations across different genres is not uniform. This uneven genre distribution, rather than being a limitation, reflects the

inherent diversity and variability within radio station genres. Such diversity introduces added complexity to our study, offering a robust platform to test the proposed recommender system's performance in realistic scenarios.

The selection criteria were meticulously designed to align with our research objectives, ensuring a dataset that is both comprehensive and diverse, making it well-suited for the proposed recommender system.

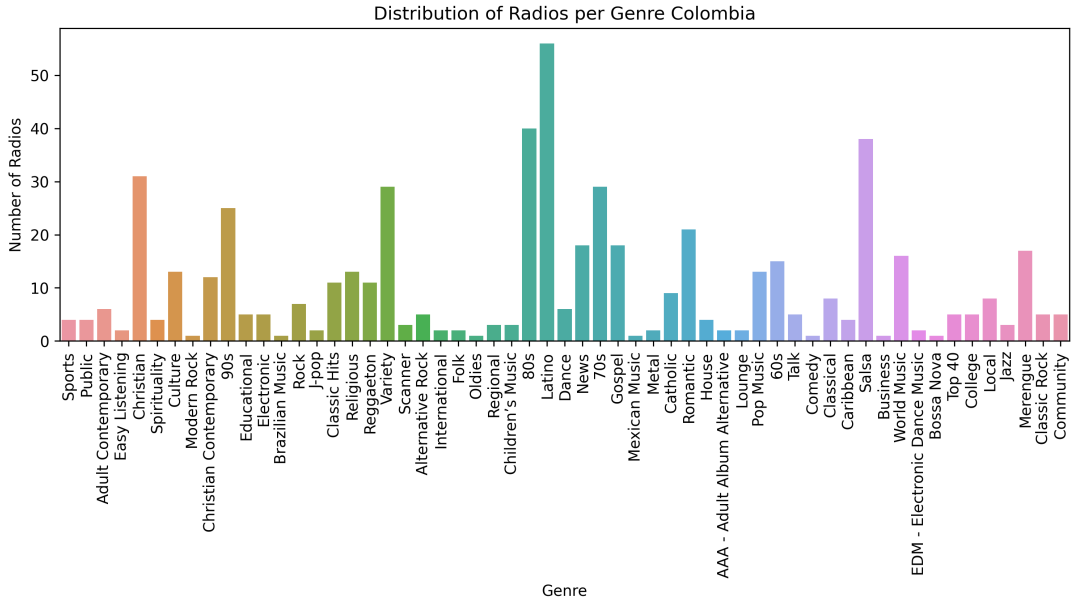


Figure 4.2: Radios Distribution Per Genre

4.1.2 System Design

In this study we are not customizing recommendations, for users. Instead our aim is to create a playlist that evolves and caters to countries. To achieve this we have designed our model to undergo updates every t, where $t = 24$ hours. This timeframe allows us to gather feedback from a range of users ensuring that our playlist updates reflect the diverse musical preferences within each country. By leveraging the intelligence of our user base we continuously enhance the quality of our recommendations. Improve the overall user experience.

While the previous architecture described in chapter 3 is highly adaptable certain adjustments are necessary for each application scenario due to problem characteristics. For instance when it comes to creating self improved radio lists we curate collections that align with the tastes of individuals, in each country. To meet this objective we have made some modifications in how we approach and define the problem formulation:

- State space S: A state $s_t = \{s_t^1, s_t^2, ..., s_t^n\} \in S$ is defined as the top played radios during the previous 24 hours. The items in s_t are sorted by the total time played.

- Action space A: An action $a_t = \{a_t^1, a_t^2, ..., a_t^n\} \in A$ is to recommend a list of radios to a user at time t based on current state s_t, where K is the number of items the RA recommends to user each time.

- Reward R: After the recommender agent takes an action a_t at the state s_t, i.e., recommending a list of items to a user, the user browses these items and provides his feedback. He can then skip (not click), click, or play these items, and the agent receives immediate reward $r(s_t, a_t)$ according to the user's feedback, since our target is to increase play time and time spent in the page we give a bigger weight to that parameter.

4.1.3 Design and Organization of User Interface

The strategic placement of the recommender system is a factor, in utilizing its potential. One key aspect of this strategy involves ensuring that the recommendations are prominently displayed in the area of the screen when the page initially loads, as shown in Figure 4.3. This is important because users tend to engage more with content that's immediately visible without requiring scrolling or navigation to access it.

By keeping the recommender system above the fold its visibility and accessibility are maximized, increasing the likelihood of user interaction with recommendations. This becomes considering that users often have limited attention spans when browsing digital platforms. If recommendations are not immediately visible there is a risk that users may overlook this feature entirely resulting in missed opportunities for engagement and personalization.

This principle has been extensively discussed in web design and user experience literature. For example a study titled "A Sign of Things to Come; Predicting the Perception of Above the Fold Time, in Web Browsing" conducted by (Jahromi et al., 2021) highlights the significance of *Above-The-Fold* (ATF) time – referring to how long it takes for a web pages visible parts to be rendered and displayed within a browser.

The study presents a metric called *Plausibly Complete Time* (PCT) which estimates the time it takes for users to perceive a websites ATF content. It emphasizes the importance of ATF content, in shaping a users satisfaction with the website. Our objective is to integrate a recommender system into the existing layout ensuring it is prominently displayed at the top of the page. This approach aims to strike a balance between user familiarity with the platforms interface and introducing functionality. This equilibrium is crucial for maintaining user engagement and satisfaction in recommender systems where personalized and captivating browsing experiences are paramount.

To summarize successfully integrating and positioning a recommender system within an existing user interface plays a role in this field. Achieving harmony, between user familiarity and improved functionality can result in heightened user engagement making

the platform more effective and satisfying for its users.

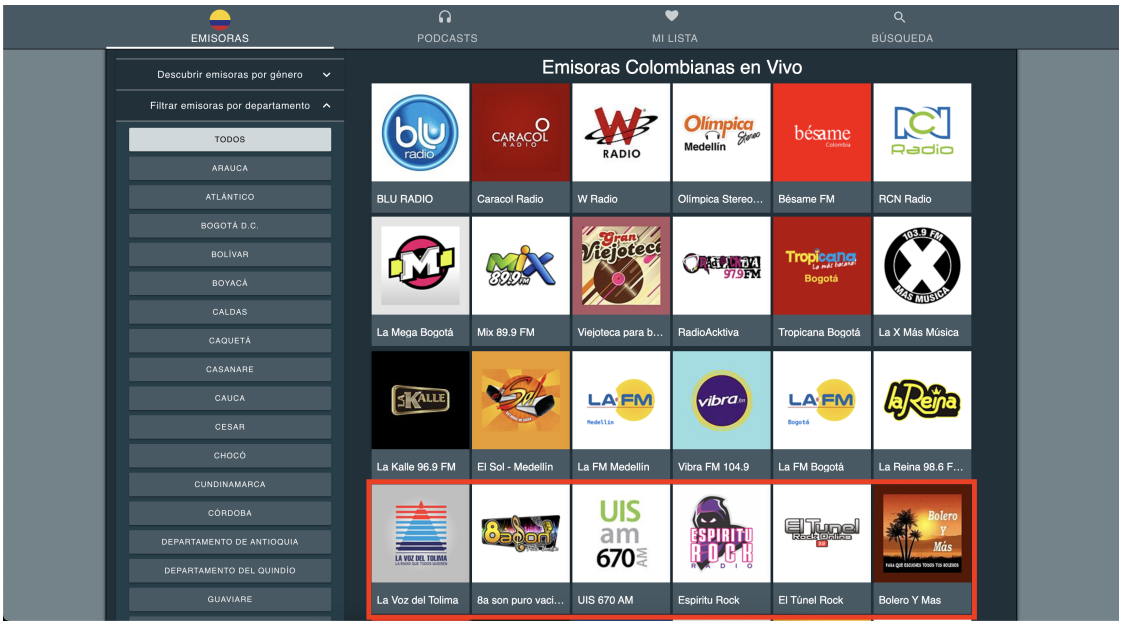


Figure 4.3: Recommended Radios User Interface

4.2 Game Recommender System

In the second scenario selected, our recommender system is deployed in a game aggregator website, reludi. Reludi is a game aggregator website that consolidates a variety of games, making them accessible to users, similar to what the Play Store offers in the Android ecosystem. However, in our specific case, most games are sourced from various developers and platforms, limiting our access to data primarily to categorical information and time spent on the page, as we do not manage the game implementations and events.

Objectives

The primary goal is to utilize available data and historical user interactions, like the duration spent on each game, despite their constraints. This aims to enhance the user experience by providing tailored game recommendations.

The architecture of the system is crafted to be flexible and responsive, centering on the player's progressive gaming pathway and shifting interests. This strategy is intended to ensure the recommendation system maintains its pertinence and reactivity, perpetually modifying in response to alterations in user inclinations and actions, thereby offering a captivating and absorbing user interaction.

By doing so, the recommender system seek to facilitate a continuous discovery process for the users, encouraging them to explore and engage with a diverse array of games

that cooperate with their tastes and preferences. This user centred approach aims to curate a recommendation environment where users are introduced to new and relevant games, enhancing their overall gaming experience and satisfaction.

Problem of scarce information about users

A significant challenge faced in this scenario is the limited availability of comprehensive user data due to the lack of user account creation. This limitation restricts our ability to fully understand and track user preferences and behaviors. To address this, a system has been implemented that assigns a unique ID to users, stored in local storage, allowing for some level of user tracking and behavior analysis. However, this approach still provides limited insights, as it is not as robust as having full user account information. This scarcity of data hampers the recommender system's ability to deliver highly accurate and personalized game recommendations, and poses challenges in effectively assessing user engagement and satisfaction levels, thereby limiting the system's capacity to evolve and refine its recommendation strategies over time.

4.2.1 Game Database Selection

The efficacy of any recommender system is deeply anchored in its foundational data. In our game recommender system, we harness a real-world game database, presenting a genuine reflection of available games and their categorizations. This extensive dataset encompasses 809 games, distributed across 70 distinct categories, capturing the breadth and diversity of the modern gaming landscape.

Our database is methodically structured, categorizing games into defined genres, this vital characteristic enhances the recommendation process. The distribution of games among these categories is visually represented in Figure 4.4.

This graphic depiction of "Distribution of Games per Category" underscores the diversity of content available, facilitating a deeper understanding of potential recommendation paths. The inherent complexities of this real-world data enable the system to discern and predict user preferences with heightened precision. Through the integration of this data, the recommender system is equipped to deliver informed game suggestions.

4.2.2 User Profiling Design

In this particular study, the emphasis is placed on delivering customized game suggestions for each user. The objective is to formulate a vibrant, self-enhancing roster of proposed games that accommodates individual tastes. A mechanism for regular updates has been incorporated to maintain the relevance of the list. A temporal parameter, denoted as t, is

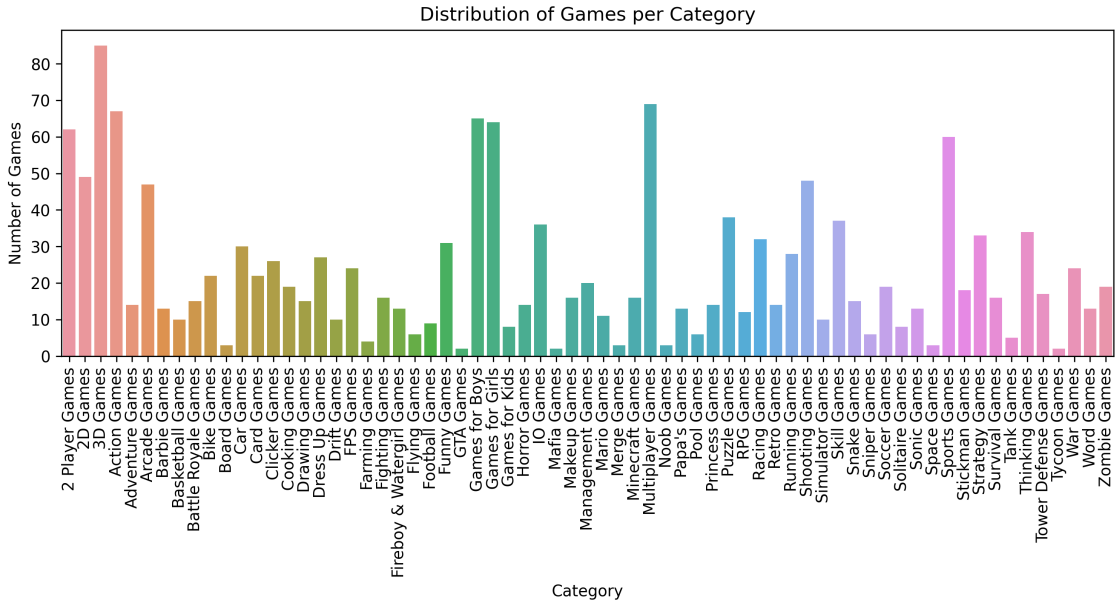


Figure 4.4: Games Distribution By Category

utilized to establish the frequency of recommendation refreshments. Consequent to each t interval, the system undertakes a reassessment of user data to refine the game proposals.

The philosophy guiding this configuration is to present suggestions that embody a blend of thoroughness and contemporaneity, mirroring the transitional preferences of our users. Through the routine revitalization of the list, an agile adaptation to shifts in user tendencies and likings is achieved.

Leveraging the aggregated wisdom of our user ensemble, the aspiration is to perpetually amplify the caliber of our game suggestions. This not only augments the user interaction but also aspires to fortify user loyalty and extend the duration of engagement.

In this distinct instance, adjustments have been executed to conform to the specialized circumstances. The foundational structure is an extrapolation of the schema delineated in Chapter 3, accompanied by the subsequent alterations:

- State space S: A state $s_t = \{s_t^1, s_t^2, ..., s_t^n\} \in S$ is defined as the last 10 games played by the user. The items in s_t are sorted by the chronological order.

- Action space A: An action $a_t = \{a_t^1, a_t^2, ..., a_t^n\} \in A$ is to recommend a list of games to a user at time t based on current state s_t, where K is the number of items the RA recommends to user each time.

- Reward R: After the recommender agent takes an action a_t at the state s_t, i.e., recommending a list of items to a user, the user browses these items and provides his feedback. He can then skip (not click), click, or play these items, and the agent receives immediate reward $r(s_t, a_t)$ according to the user's feedback, since our target

is to increase play time and time spent in the page we give a bigger weight to that parameter.

4.2.3 User Interface

In the development of our game recommendation system, we had to adhere to specific design choices mandated by the project owner. One such significant decision was the placement of our recommendations on the website. Positioned well below the fold, as shown in Figure 4.5, this location does not ideally support our data gathering efforts. However, given the project's innovative nature and its beta status, these compromises were necessary. They were made to align with the broader vision set by the project owner and to ensure the system's smooth integration without disrupting the overall user experience.

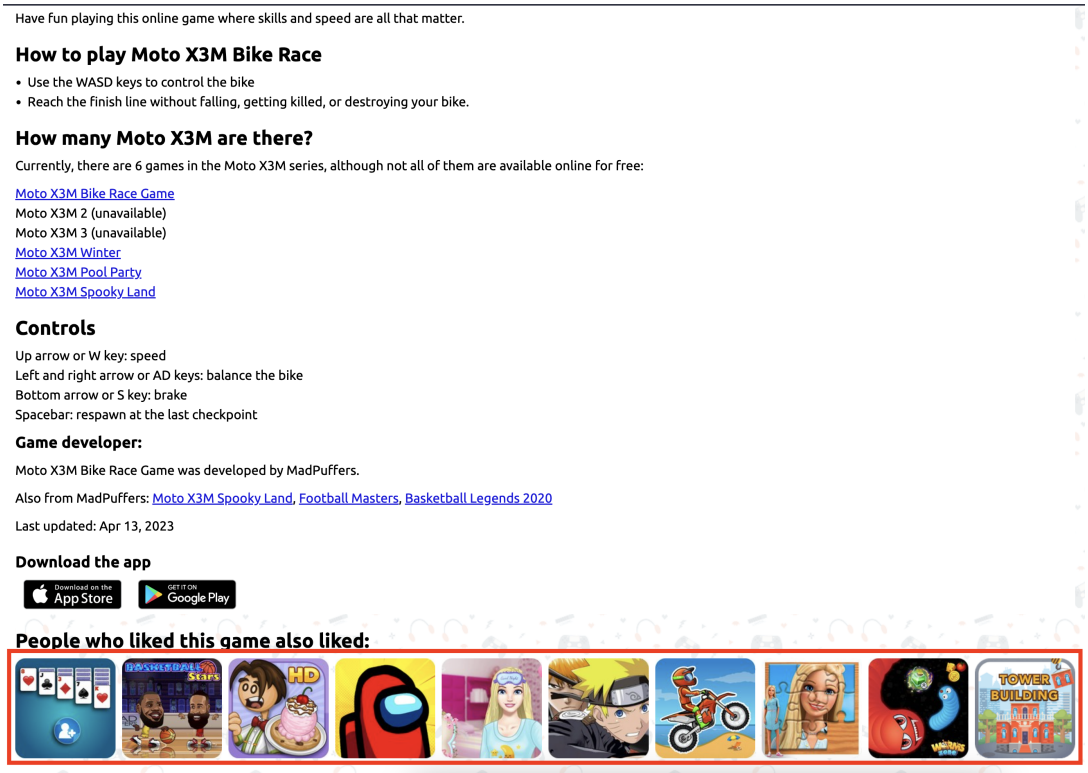


Figure 4.5: Recommended Games User Interface Position In The Page

The recommended games are showcased in a grid layout as it can be seen in 4.6, comprising n items. For design coherence and to maintain page alignment, n is set to 10, ensuring the entire page width is utilized. However, this grid is situated "below the fold," requiring users to scroll further down to access these recommendations. This design choice is suboptimal, as our website analytics reveal that a mere 7% of users navigate to this section. The retention metrics are even more concerning, with a minimal number of users revisiting this section. This placement adversely affects our recommendation engine's performance, given the reduced user interaction with the suggested games.

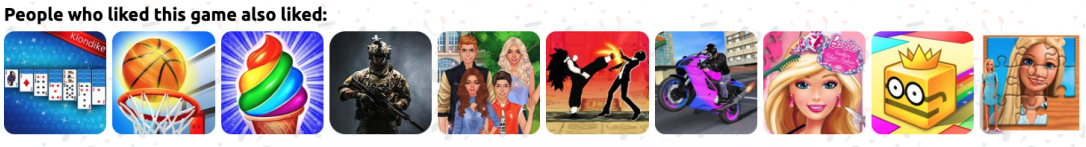


Figure 4.6: Recommended Games User Interface

Our game recommendations are thoughtfully integrated into two primary sections of the website: the individual game page and the category page. This integration guarantees that users who venture below the fold are presented with personalized game suggestions in various contexts. However, the predominant challenge is enhancing the initial visibility and accessibility of these recommendations to a broader user base.

This chapter discusses the application of reinforcement learning in recommender systems, illustrated through two distinct scenarios: a Radio Aggregator and a Game Aggregator. The Radio Aggregator aims to foster a diverse radio ecosystem by recommending lesser-known stations based on users' listening history and geolocation, overcoming challenges such as users bypassing the homepage and the dominance of popular stations. The Game Aggregator, despite limited user data, focus on enhance user experience by offering dynamic and personalized game recommendations, adapting to users' evolving gaming interests. Both systems emphasize a self-improving mechanism, where recommendations are periodically updated to reflect users' changing preferences and behaviors. The design considerations, including the user interface and system architecture, are meticulously detailed, underscoring the strategic efforts to optimize user engagement and satisfaction in the face of various challenges such as data scarcity and interface constraints.

Chapter 5

Evaluation

In this segment, we will commence by scrutinizing the accumulated data to ascertain a comprehensive understanding of the subject matters. Subsequently, the outcomes derived from the algorithm will be deliberated, emphasizing the operational patterns of the recommendations in both online and offline modes.

Pertaining to our investigative endeavor, two separate application contexts are being examined. Notwithstanding the disparities in the functional landscapes, a uniform methodological strategy will be sustained throughout the evaluative proceedings.

In each instance, predominant methodologies discerned from our literature exploration will be employed, accompanied by a detailed elucidation of the pertinent theoretical paradigms, as articulated in Subsection 2.1.3.

5.1 Self-improving Lists for Radios

5.1.1 Data Collected for Evaluation

Before diving into into the evaluation of our self-improving system, it's pivotal to lay out the context and nature of the data collected. This foundational understanding will facilitate comprehension of the subsequent analyses and discussions.

As previously mentioned, our system underwent testing in a real-world, live application setting with genuine users. To initiate this, we primed our algorithm using data amassed over the preceding three months. Following this preparatory phase, we deployed the system on Colombia's regional website. Over the ensuing two months, the algorithm operated in this live environment, autonomously gathering and processing user interaction data. Our evaluation, as detailed in this chapter, is centered on analyzing the outcomes and insights derived from this two-month operational period, during which our algorithm produced 60 lists.

5.1.2 Analysis of the Results of the Algorithm

To effectively assess the progression and performance of our algorithm, we will employ two pivotal metrics. The first, the cumulative reward, offers insight into the total benefits the algorithm accumulates over time. The second metric, regret, provides a comparative perspective, highlighting the gap between the outcomes achieved by our algorithm and the optimal possible outcomes. By jointly analyzing these metrics, we can obtain a holistic view of the algorithm's efficiency and areas for potential improvement.

Table 5.1: Self Improved Radio List Performance:Reward and Regret Metrics over time

	First Month	**Second Month**
Average Reward	20	30
Average Regret	80	70

While our regret values might initially appear high, it is pivotal to understand the nuances behind these figures to fully appreciate the context. Our reward design, composed of two main components, significantly contributes to these metrics:

- **Played Time Rate:** Representing 70% of our reward calculation, this metric measures the ratio of the total playtime generated from our recommendation list to the overall playtime on the site.
- **Click Distribution:** Capturing the spread of user interactions with our recommendations, this metric occupies the remaining 30% of our reward.

Considering these intricacies, the observed regret values reflect a combination of our reward design and distinct user navigation patterns (as outlined in Section 4). As users frequently opt for familiar content, a significant portion of recommendations may remain unexplored, thus heightening the regret values.

However, it's essential to view these regret values as more than mere numbers. They offer a practical evaluation of the algorithm's real-world efficiency. Encouraging trends, such as the uptick in average reward and reduction in average regret, suggest our algorithm's commendable performance in bolstering user interactions. Furthermore, we believe that as we refine our algorithm by harnessing user behavior insights, these metrics will see even greater improvements.

Coverage

In the specific case under study, the coverage is evaluated based on the selection of radio stations in Colombia. Over a period of two months, the system consistently exhibited a coverage value of approximately 25%. This was determined by the selection of 92 radios

out of a possible 367, illustrating the system's ability to access and utilize a diverse range of radio options.

The concept of coverage in this study is crucial as it reflects the system's capacity to recommend a broad spectrum of radios, ensuring that users are provided with varied options. Maintaining a consistent and comprehensive coverage allows for a more robust evaluation, ensuring that the analysis is not influenced by fluctuations or biases in the selection process. This approach aims to uphold a level of uniformity across all user scenarios and recommendation systems, fostering a more reliable and equitable analytical framework.

Intra-List Similarity (ILS)

The ILS metric is employed to gauge the variety of radio suggestions offered by the system. The ILS is pivotal as it sheds light on the breadth and depth of the recommendations presented to the audience. A table delineating the diversity outcomes over two sequential months is detailed below:

Table 5.2: Diversity Results For Radio Recommendations

	First Month	**Second Month**
Diversity	0.64223	0.62104

- **First Month:** During the starting month, the system showcased a diversity value of 0.64223. This value denotes a balanced degree of diversity in the radio suggestions, reflecting a commendable range of choices extended to the users.
- **Second Month:** In the month thereafter, there was a minor dip in diversity, with the value dropping to 0.62104. This points to a slight narrowing in the suggestions, hinting that the radio options put forth were marginally more alike or had less variation than the prior month.

The discerned pattern, a gentle reduction in diversity across the two months, could be shaped by multiple determinants such as patterns of user engagement, tweaks in the algorithm, or shifts in radio trends and accessibility. While an elevated diversity value is typically sought-after to guarantee a wide array of suggestions, striking a harmonious equilibrium is pivotal. Overemphasizing diversity might inadvertently yield suggestions that resonate less with the user's preferences.

Category Diversity

Category diversity has an essential role in assessing the richness and variety of the recommendations made. It ensures that users are exposed to a diverse array of items across

multiple categories, thereby reducing monotony and enhancing user engagement. Take this example, a user who frequently listens to news radio may also have an interest in other genres, such as jazz. Thus, providing recommendations that solely align with the user's explicit preferences, like news, may not always yield the most engaging and satisfying user experience.

Table 5.3: Category Diversity Results For Radio Lists

	First Month	**Second Month**
Category Diversity	3	3

As illustrated in Table 5.3, the category diversity score was found to be 3, this value results from the average rounded to the units of the number of categories addressed for each list made by our system. Given that each recommendation list comprised six items, it signifies that each list, on average, contained approximately two items from each category. This degree of diversity ensures a balanced and varied set of recommendations, preventing an over-concentration in specific categories and fostering a more engaging user experience.

Popularity

Popularity trends serves as a cornerstone for understanding the robustness and adaptability of our recommendation algorithm in the realm of radio broadcasting.

From the data illustrated in Figure 5.1, and further corroborated by the provided dataset, we discerned noteworthy patterns during two distinct intervals: June-July and July-August. The data shows a clear trend where the algorithm tends to favor a specific set of radios that appear most relevant to listeners over time.

However, a closer inspection of the dataset reveals granular insights:

- There are radios, such as *Bolero Y Mas*, that witnessed a substantial decline in their popularity rankings. This could be attributed to various factors, including changes in listener preferences or algorithmic adjustments.
- Conversely, radios like *Don Tamalio Online* experienced a surge in their popularity metrics, indicating a potential increase in their relevance or listener engagement.
- A subset of radios, including *Espiritu Rock* and *Machín Estéreo*, displayed remarkable stability, indicating a consistent listener base or algorithmic confidence in their relevance.

While these insights are enlightening, it's crucial to emphasize the exploratory nature of this phase. Given the embryonic stage of our algorithm's learning curve, such trends, although promising, mandate continuous oversight and iterative refinements. This careful

approach ensures our recommendation system's resilience, ensuring it remains responsive to the ever-changing preferences of its listeners.

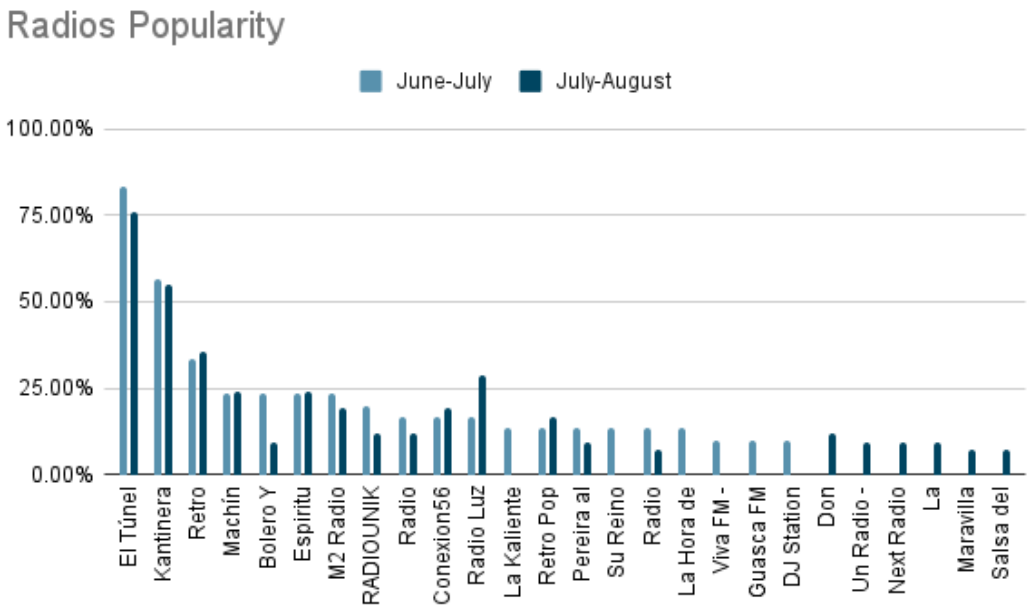


Figure 5.1: Comparative Popularity Analysis of Radios: June-July vs. July-August

5.1.3 Online Metrics Evaluation

Numerical insights, although valuable, may not always portray the full story. To offer a more comprehensive view, Figure 5.2 provides a visual representation of our system's progression in the Colombian market from June 21 to August 21.

This figure clearly indicates a general upward trend, underscoring an enhancing alignment between user inclinations and our recommendations. However, it also highlights a noteworthy dip between weeks 29 and 31. Such occasional deviations can be attributed to the explorative nature of reinforcement learning algorithms, which sometimes take seemingly sub-optimal actions that are essential for broader, long-term learning.

Importantly, while the preliminary results are indeed promising, we must underscore that our system is still in its developmental phase. So far, our algorithm has explored a fraction of the potential candidate space, this indicates a vast realm of possibilities yet to be tapped, which bodes well for the future.

Delving deeper into the deviations observed between weeks 29 and 31, Figure 5.3 draws attention to an intriguing pattern.

This figure accentuates the correlation between radios that failed to playback and the dip in our performance metrics. Specifically, during weeks 30 and 31, a concerning average of 2.9 out of the 6 radios didnt did not play. While this might seem minimal in

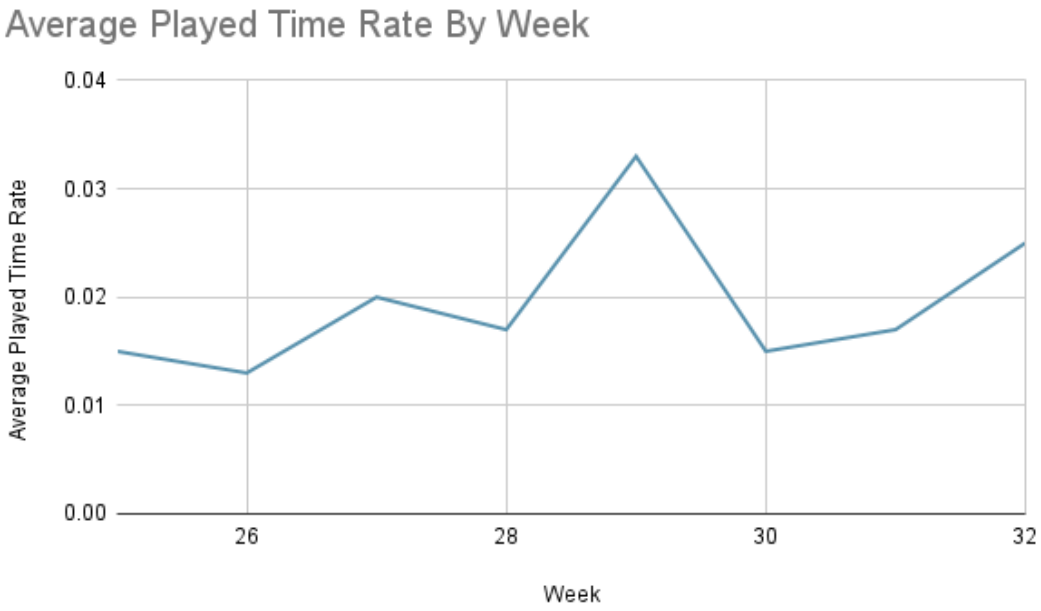


Figure 5.2: Recommendation Evolution over Weeks

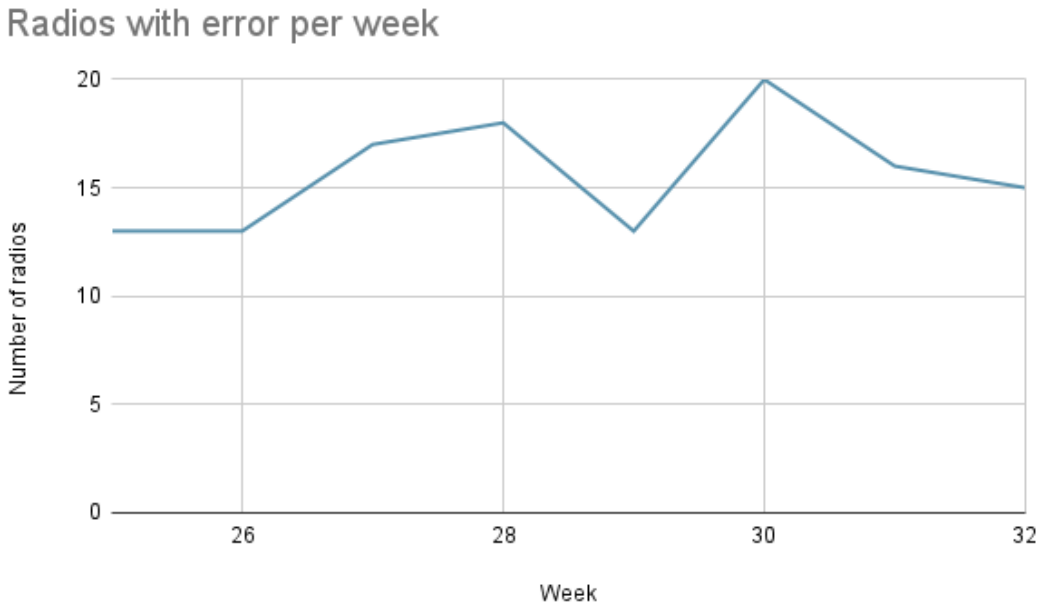


Figure 5.3: Playback Errors by Week

isolation, compared to the 2.1 average of other weeks, the difference becomes significant. Given our lists contain six items, and on average, half of them fail to playback due to errors, this limitation becomes glaring and demands immediate attention.

To summarize, although the results obtained so far are promising, the early stage of our algorithm and the challenges we've observed suggest there's a vast scope for growth and optimization.

5.2 Game Recommender System

5.2.1 Data Collected for Evaluation

In our evaluation process, we collected data from Reludi, our game aggregator web application. The dataset contained chronological user interaction metrics such as clicks and playtime, accumulated over a span of approximately one month. This sampled data was then used for the initial training of our recommendation system, enabling the model to harness valuable insights and patterns from user interactions.

We began by sorting our data in chronological order and then divided it into two parts: training and testing. To do this, we used the first three weeks of collected data for training and allocated the remaining week for testing. This systematic organization of data was crucial to preserve the sequential integrity of user interactions, facilitating the extraction of meaningful temporal insights.

Following the offline evaluation, the pre-trained model was deployed into a live production environment. Transitioning to an online setting, the model utilized the knowledge acquired from offline training to adapt to real-world user interactions and feedback. This phase was conducted over a period of six months. This thoughtful approach and structured implementation of the model in a dynamic environment allowed for refined tuning and optimization, enhancing its performance and recommendation capabilities based on actual user engagement and behaviors.

5.2.2 Analysis of the Results of the Algorithm

In the detailed examination of this scenario, a consistent methodological approach will be preserved to ensure the reliability of the evaluation results. Since the basic structure of our system remains unchanged, it is essential to use the same evaluation measures as previously outlined.

Reviewing Table 5.4, it is clear that the offline setting provides a superior average reward and a reduced average regret compared to the online setting.

In the offline environment, the algorithm shows satisfactory performance, evident from the average rewards and regrets. This performance is due to the stable nature of

Table 5.4: Self Improved Game List Performance:Reward and Regret Metrics in Offline and Online Environments

	Offline Environment	Online Environment
Average Reward	40	20
Average Regret	60	80

the environment where the algorithm interacts with fixed, unchanging data. This stability allows for a more controlled and predictable evaluation process, contributing to the consistent performance of the algorithm.

Conversely, the online setting is characterized by a more dynamic and changeable operational environment. Here, the algorithm faces the challenges of fluctuating real-time user interactions and preferences. A significant limitation in this setting is the less optimal visibility and placement of content; as noted earlier, our lists are positioned below the main viewing area. This positioning inherently limits the ease of content discovery, complicating the task of effectively optimizing user engagement.

Coverage

In the evaluation of this application scenario within the domain of video games, a distinct operational challenge was encountered. Over a period of six months, games were periodically introduced into the system by the content team, a dynamic that the system had to adapt to and manage efficiently. Despite the fluctuating game pool, the system displayed a remarkable adaptability, maintaining a coverage metric of approximately 53.15% by recommending 430 games out of an evolving pool that reached up to 809 games.

This incremental content introduction likely influenced the system's recommendation diversity and adaptability, urging it to continuously evolve and reassess the suitability of its recommendations. The periodic addition of new games suggests that the system's algorithms were responsive and capable of integrating fresh content effectively, ensuring that the users received up-to-date and varied recommendations.

Intra-List Similarity (ILS)

In this segment of our evaluation, the Intra-List Similarity (ILS) is examined to assess the diversity of game recommendations proffered by the system. As previously explained, is instrumental, furnishing insights into the assortment and depth of the game titles recommended to users. It acts as a pivotal gauge that elucidates the extent of variety encapsulated within the recommendation lists, ensuring that users are exposed to a multifaceted array of games that span across various genres and styles. In table 5.5 we can compare the ILS measures obtained in both environments during the experiment period

Table 5.5: Diversity Results for Game Recommendations

	Offline Environment	Online Environment
Diversity	0.54331	0.55076

- **Offline Environment:** In the offline setting, the system displayed a diversity score of 0.54331. This score signifies a balanced diversity in the game recommendations, ensuring that users receive a varied array of gaming options, which is neither too narrow nor excessively broad.
- **Online Environment:** Transitioning to the online environment, a slight improvement in diversity was noted, with the score marginally increasing to 0.55076. This subtle enhancement indicates a broader array of game recommendations, providing users with a slightly more varied selection of gaming genres and styles.

As earlier mentioned in the previous application scenario its important to keep a balance between high diversity and over saturation of games present to the users in order to not compromise the recommendations.

Category Diversity

The concept of category diversity is essential when evaluating recommendations. It ensures that users are presented with game titles spanning various genres, thereby enriching their gaming experience and preventing a sense of redundancy. As previously referred in the radios application scenario a similar effect can be seen in this domain. For example, a user who predominantly plays strategy games might also be intrigued by adventure or action games. Solely recommending games based on a user's evident preferences, such as strategy, might not encapsulate the full spectrum of their potential interests.

Table 5.6: Category Diversity Results For Game Lists

	Offline Environment	Online Environment
Category Diversity	4	5

As illustrated in Table 5.6, the category diversity score for game recommendations was observed to be 4 in the offline environment and 5 in the online environment. This score is derived from the rounded average of the number of genres represented in each recommendation list generated by our system. Considering that each list consists of ten game, it implies that, on average, each list encompasses games from four to five different genres. Such a level of diversity ensures that users are not pigeonholed into a single genre, promoting a richer and more immersive user experience.

Popularity

Has previously mentioned popularity can help us track the robustness and adaptability of a recommender systems, so we are using the same metric in the realm of our game aggregator recommender system.

The Plot 5.4 was created using our production data, in the y axis we can see the percentage of the number of time a game was recommended taking in account the number of recommendations, and in the x axis we can see the game name.

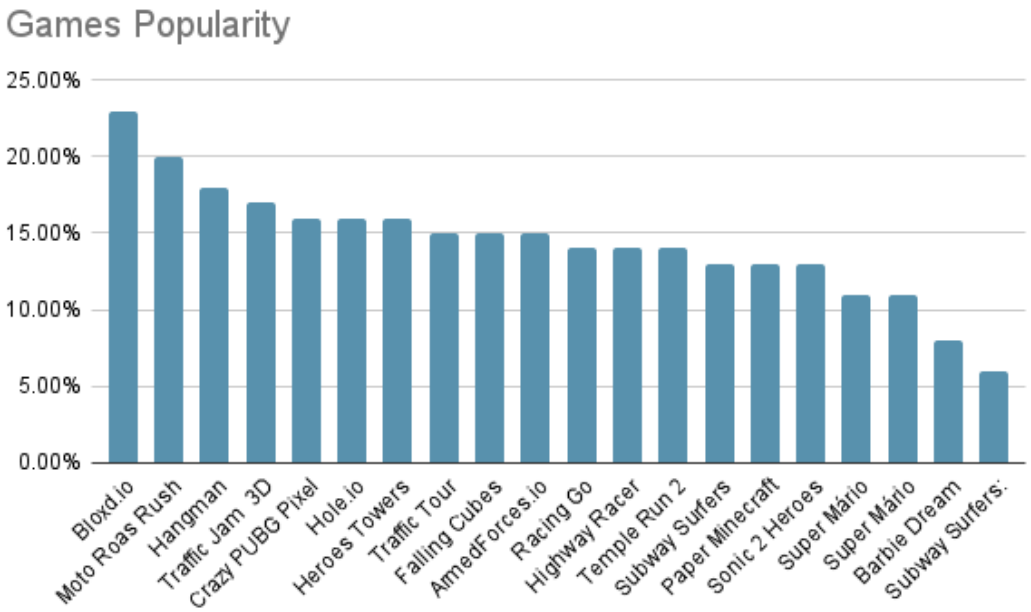


Figure 5.4: Games Popularity Analysis

5.2.3 Online Metrics Evaluation

In Figure 5.5, a comprehensive evolutionary trajectory of our recommender system is portrayed through users' interactions. The visualization demonstrates two essential metrics: the total duration in seconds and the total clicks, illustrated on the y-axis, while the x-axis presents the number of interactions. Within the illustration, individual lines represent unique users, whereas a more pronounced red line demarcates the average user in both plots, establishing a comparative benchmark across the evaluated metrics.

An insightful observation that emerges from the collected data during this period is the limited user interaction with the recommender system. Only 7% of users navigating through the website manage to engage with our recommender system. This phenomenon can be attributed the fact that our system is being positioned below the fold. This becomes even more significant when considering that the retention of users interacting more than

twice with the system is quite scarce, corroborating the anticipated user behavior based on the system's positioning and visibility.

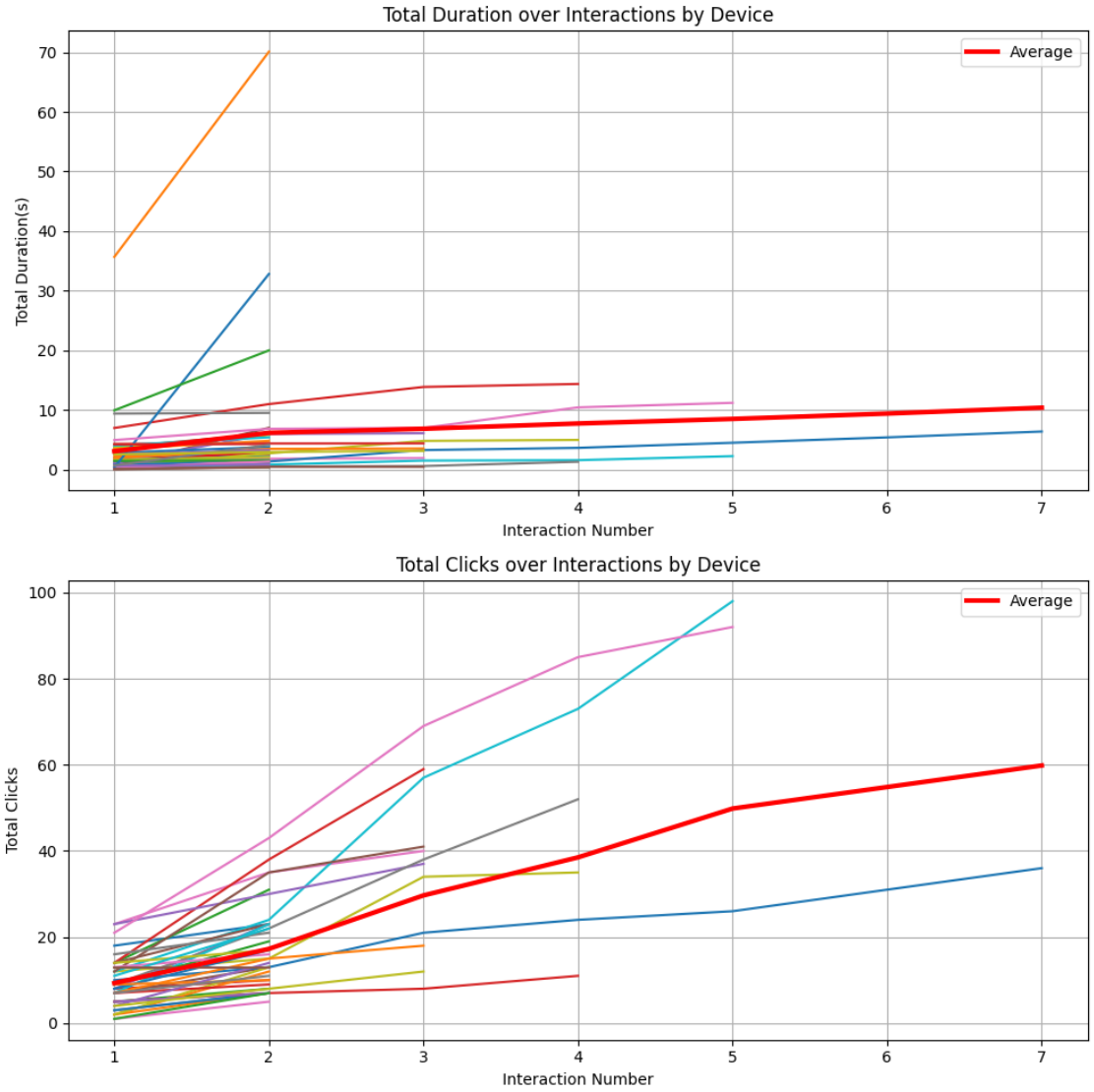


Figure 5.5: Game Recommender Evolution Over Slates

Delving deeper into the analysis, it is noteworthy that our algorithm manifests a dynamic improvement trajectory. A meticulous examination of Figure 5.5 unveils a discernible enhancement pattern in algorithm performance corresponding with increased user interactions. This is substantiated by the cumulative metrics of clicks and playtime, where a progressive refinement in response to user feedback is evident.

In conclusion, despite the modest user interaction levels, our recommender system exhibits a good capacity to leverage user interactions for iterative improvement and refinement. The observable upward trend in user clicks and playtime signals the system's ability to adapt and refine in order to keep up with with user preferences and behaviors.

Chapter 6

Conclusion

In the course of this research, we successfully designed and implemented a recommendation system. This system, built on the foundations of the DDPG algorithm and it's methodology, has been purposefully integrated into two distinct applications: a game aggregation platform and a radio station aggregation platform. The system successfully personalizes game and radio station recommendations by adapting to user preferences and their evolving consumption patterns.

The user interface of the recommender system is not merely visually captivating and user-friendly but also harmoniously melds with the pre-existing design ethos of the platforms in both application scenarios. Such a fusion is strategized to mitigate disorientation among habitual users, fostering a unified user interaction landscape. Consequently, the system operates with efficacy, augmenting the user engagement quotient, thereby realizing one of our pivotal objectives.

However, certain challenges were encountered during the implementation of this system. For instance, the substantial volume of pre-test data required could lead to increased processing time for recommendations as the databases of items expand. Despite this, our primary aim of demonstrating the practicality and usefulness of a RL based recommendation system has been satisfactorily achieved.

Examination outcomes unveiled a progressive enhancement in the recommendation system's proficiency, adeptly modulating in resonance with the fluctuating user consumption paradigms. The recommendations proffered were pertinent and variegated.

To provide a substantive contribution to the current body of research, our system was implemented in two real-world scenarios and remains operational at present. This approach served to anchor our results and evaluations in actual practice, thereby enhancing the authenticity and reliability of our study.

In encapsulation, the revelations of this dissertation accentuate the promise harbored by the evolved recommendation system in delivering precise and individualized user rec-

ommendations. Moreover, the system's recommendations exhibit a capacity for temporal evolution, synchronizing with user predilections, thereby amplifying the comprehensiveness of their engagement experience.

6.1 Future Work

Both application scenarios presented in this study, concerning self-improving radio lists and self-improving game lists for users, reflect ongoing work with refined recommendation models as the ultimate goal. Although successfully validated, there is room for future enhancements in both domains.

In the context of the game list recommendation, future work is targeted at improving the presentation of the recommended games. The effectiveness of recommendations could potentially be improved if the list of suggested games were presented in a more prominent section of the site. This would increase the visibility of the recommendations and potentially lead to better user engagement. Therefore, more research into effective user interface design and its impact on recommendation systems would be beneficial.

On the other hand, in the context of radio recommendations, the encouraging results obtained could be further enhanced by collecting more metadata about the current song, music history, and live feed. At the time of this study, such data were not available. However, obtaining this data could lead to more precise and personalized recommendations that better match the user's preferences. Future research could focus on techniques for obtaining and utilizing such data.

Moreover, this dissertation has presented the specifications of a generic architecture implementing a reinforcement learning paradigm for recommender systems. This presents a foundation on which future recommender systems applications can rely. It is hoped that this will inspire other researchers to further develop these concepts and contribute to the evolution of recommender systems in various application domains.

Bibliography

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17:734–749, 6 2005. URL https://pages.stern.nyu.edu/~atuzhili/pdf/TKDE-Paper-as-Printed.pdf. 6

M. Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys (CSUR)*, pages 1–37, 12 2022. ISSN 0360-0300. doi: 10.1145/3543846. URL https://dl.acm.org/doi/10.1145/3543846. 17

Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016. 7

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, page 43–52, 1998. 7

Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370, 2002. 8

Xiaocong Chen, Lina Yao, Julian McAuley, Guanglin Zhou, and Xianzhi Wang. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowledge-Based Systems*, 264:110335, 3 2023. ISSN 09507051. doi: 10.1016/j.knosys.2023.110335. 18

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. Generative adversarial user model for reinforcement learning based recommendation system. In *Proceedings of the36thInternational Conference on MachineLearning*, pages 1–10, 2019. 19, 21

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *CoRR*, pages 1–11, 12 2015. URL http://arxiv.org/abs/1512.07679. 23

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *ICLR 2018*, pages 1–21, 6 2018. URL http://arxiv.org/abs/1706.10295. 14

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Addressing Function Approximation Error in Actor-Critic Methods*, pages 1–15, 2 2018. URL http://arxiv.org/abs/1802.09477. 15

Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019. URL http://14.139.161.31/OddSem-0822-1122/Hands-On_Machine_Learning_with_Scikit-Learn-Keras-and-TensorFlow-2nd-Edition-Aurelien-Geron.pdf. 12

Stephanie Hackett and B. Parmanto. Homepage not enough when evaluating web site accessibility. *Internet Res.*, 19:78–87, 2009. doi: 10.1108/10662240910927830. 31

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 230–237, 1999. 7

Liwei Huang, Mingsheng Fu, Fan Li, Hong Qu, Yangjun Liu, and Wenyu Chen. A deep reinforcement learning based long-term recommender system. *Knowledge-Based Systems*, 213, 2 2021. ISSN 09507051. doi: 10.1016/j.knosys.2020.106706. 17, 18

Folasade Isinkaye, Yetunde Folajimi, and Bolanle Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16, 08 2015. doi: 10.1016/j.eij.2015.06.005. 7, 8

Hamed Z. Jahromi, Declan Delaney, and Andrew Hines. A sign of things to come: Predicting the perception of above-the-fold time in web browsing. *Future Internet*, 13(2), 2021. doi: 10.3390/fi13020050. 34

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: An introduction. *Recommender Systems: An Introduction*, 9780521493369:1–335, 1 2010. doi: 10.1017/CBO9780511763113. URL https://www.cambridge.org/core/books/recommender-systems/C6471B59388D8A9F684C49C198691B53. 6

Mathias Jesse, Christine Bauer, and Dietmar Jannach. Intra-list similarity and human diversity perceptions of recommendations: the details matter. *User Modeling and User-Adapted Interaction*, 33, 2022. 10

Rahul Katarya and Om Prakash Verma. An effective collaborative movie recommender system with cuckoo search. *Egyptian Informatics Journal*, 18:105–112, 2017. 9

Joseph A. Konstan and John Riedl. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22:101–123, 4 2012. doi: 10.1007/S11257-011-9112-X/METRICS. 6

Hamidreza Koohi and Kourosh Kiani. User based collaborative filtering using fuzzy c-means. *Measurement*, 91:134–139, 2016. doi: https://doi.org/10.1016/j.measurement.2016.05.058. URL https://www.sciencedirect.com/science/article/pii/S0263224116302159. 7

Pushpendra Kumar and Ramjeevan Singh Thakur. Recommendation system techniques and related issues: a survey. *International Journal of Information Technology (Singapore)*, 10:495–501, 12 2018. doi: 10.1007/S41870-018-0138-8/METRICS. URL https://link.springer.com/article/10.1007/s41870-018-0138-8. 7

Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41:2065–2073, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2013.09.005. 27

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. 14, 15, 23

Feng Liu, Ruiming Tang, Xutao Li, Weinan Zhang, Yunming Ye, Haokun Chen, Huifeng Guo, and Yuzhou Zhang. Deep reinforcement learning based recommendation with explicit user-item interactions modeling. *arXiv preprint arXiv:1810.12027*, pages 1–11, 2019. 7, 12, 17, 21

LUMINATE. Luminate Year-End Music Report. Technical report, LUMINATE, United States, 2022. 31

Robert K. Merton. The matthew effect in science. *Science*, 159(3810):56–63, 1968. doi: 10.1126/science.159.3810.56. URL https://www.science.org/doi/abs/10.1126/science.159.3810.56. 31

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *DeepMind Technologies*, 1:1–9, 12 2013. URL http://arxiv.org/abs/1312.5602. 13, 14, 15

Raymond J Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 195–204, 2000. 18

Isshu Munemasa, Kunioki Hayashi, Yuta Tomomatsu, and Tomohiro Takagi. Deep reinfocement learning for recommender systems. In *2018 International Conference on Information and Communications Technology*, pages 226–233, 2018. ISBN 9781538609545. 19, 21

John P. O'Doherty, Sang Wan Lee, and Daniel McNamee. The structure of reinforcement-learning mechanisms in the human brain. *Current Opinion in Behavioral Sciences*, 1: 94–100, 2 2015. ISSN 2352-1546. doi: 10.1016/J.COBEHA.2014.10.004. 11

Chenguang Pan and Wenxin Li. Research paper recommendation with topic analysis. In *2010 International Conference On Computer Design and Applications*, volume 4, pages V4–264–V4–268, 2010. doi: 10.1109/ICCDA.2010.5541170. 6

Martin L. Puterman. *Markov Decision Processes*. John Wiley & Sons, Inc, 1994. 13

Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. *Recommender Systems Handbook*, pages 1–35, 2011. doi: 10.1007/978-0-387-85820-3_1. URL `https://link.springer.com/chapter/10.1007/978-0-387-85820-3_1`. 1, 6

Pornthep Rojanavasu, Phaitoon Srinil, and Ouen Pinngern. New recommendation system using reinforcement learning. In *Proceedings of the Fourth International Conference on eBusiness*, pages 23–27, 2005. URL `https://www.researchgate.net/publication/252413398`. 19, 21

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, page 285–295, 2001. doi: 10.1145/371920.372071. 7

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 31 st International Conference on Machine Learning, Lille, France, 2015.*, pages 1–17, 2 2017. URL `http://arxiv.org/abs/1502.05477`. 14

David Silver, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. *31st International Conference on Machine Learning, ICML 2014*, 1:1–9, 6 2014. 15

Satinder Singh and Richard Sutton. Reinforcement learning with replacing eligibility traces computer go view project reinforcement learning models view project. *Machine*

Learning, 22:123–158, 11 1996. doi: 10.1023/A:1018012322525. URL https://www.researchgate.net/publication/2453976. 14

Richard S Sutton and Andrew G Barto. *Reinforcement Learning An Introduction second edition*. Bradford Book, 2018. 11, 12, 17

Haining Tan. Reinforcement learning with deep deterministic policy gradient. In *Proceedings - 2021 International Conference on Artificial Intelligence, Big Data and Algorithms, CAIBDA 2021*, pages 82–85. Institute of Electrical and Electronics Engineers Inc., 5 2021. ISBN 9781665424905. doi: 10.1109/CAIBDA53561.2021.00025. 15, 22, 23

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2094–2100, 2016. URL www.aaai.org. 14

Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop*, volume 30, pages 47–56. Barcelona, 2000. 8

Jan Šafařík, Vojtěch Vančura, and Pavel Kordík. Repsys: Framework for interactive evaluation of recommender systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*, page 636–639. Association for Computing Machinery, 2022. 9

Kai Wang, Zhene Zou, Minghao Zhao, Qilin Deng, Yue Shang, Yile Liang, Runze Wu, Xudong Shen, Tangjie Lyu, and Changjie Fan. Rl4rs: A real-world dataset for reinforcement learning based recommender system, 2023. 20, 21

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. Explainable reasoning over knowledge graphs for recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/view/4470. 26

Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for page-wise recommendations. In *RecSys 2018 - 12th ACM Conference on Recommender Systems*, pages 95–103. Association for Computing Machinery, Inc, 9 2018a. ISBN 9781450359016. doi: 10.1145/3240323.3240374. 18, 21

Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1040–1048, New York, NY, USA, 2018b.

Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219886. URL https://doi.org/10.1145/3219819.3219886. 18

Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. Deep reinforcement learning for search, recommendation, and online advertising: a survey. *SIGWEB*, pages 1–15, 12 2019a. doi: 10.1145/3320496.3320500. URL http://arxiv.org/abs/1812.07127http://dx.doi.org/10.1145/3320496.3320500. 17

Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Dawei Yin, Yihong Zhao, and Jiliang Tang. Deep reinforcement learning for list-wise recommendations. In *Deep Reinforcement Learning for List-wise Recommendations*, volume 1828, pages 89–93, 12 2019b. doi: 10.48550/arxiv.1801.00209. URL https://arxiv.org/abs/1801.00209v3. 12, 18, 21

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *the 2018 World Wide Web Conference*, 2018. 12, 18, 21

Gabe Zichermann and E Christopher Cunningham. *Gamification by Design*. O'Reilly Media, 2011. 32

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, page 22–32. Association for Computing Machinery, 2005. 8, 9, 10