UNIVERSIDADE FEDERAL DO ABC

CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS APLICADAS

ENGENHARIA DE INFORMAÇÃO

MATHEUS RAMOS RIBEIRO

# CONTEXTUAL BANDITS

## Uma possível aplicação em Sistemas de Recomendação

SANTO ANDRÉ

2024

Matheus Ramos Ribeiro

# CONTEXTUAL BANDITS

## Uma possível aplicação em Sistemas de Recomendação

Trabalho de Graduação apresentado ao concluir a Graduação em Engenharia de Informação, como parte dos requisitos necessários para a obtenção do Título Bacharel em Engenharia de Informação.

Universidade Federal do ABC

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas

Engenharia de Informação

Orientador: Prof. Dr. Ricardo Suyama

SANTO ANDRÉ

2024

# Dedicatória

Dedico este trabalho a todas as pessoas que foram fundamentais em minha jornada acadêmica e pessoal.

Ao meu orientador Ricardo Suyama, pelos ensinamentos, orientações valiosas e pela confiança depositada em mim ao longo deste percurso acadêmico.

Aos meus pais, Eduardo e Rita, cujo amor, apoio e sacrifícios fizeram com que eu alcançasse este momento tão importante em minha vida.

E, especialmente, à minha amada esposa Sheila Cristina, pelo constante incentivo, compreensão e paciência durante os momentos de estudo e dedicação a este trabalho, por ser meu pilar de força, companheira incansável e fonte inesgotável de amor e inspiração. Sem você, nada disso seria possível. Este trabalho é dedicado especialmente a você, com todo o meu amor e gratidão.

*“Transmita o que aprendeu. Força, mestria. Mas fraqueza, insensatez, fracasso também. Sim, fracasso acima de tudo. O maior professor, o fracasso é. Luke, nós somos o que eles crescem além. Esse é o verdadeiro fardo de todos os mestres.”*

Mestre Yoda

## Resumo

Este Trabalho de Graduação investigou a aplicação de técnicas de Aprendizado por Reforço, especificamente *Contextual Bandits*, em sistemas de recomendação. Utilizando conjuntos de dados do MovieLens e IMDb, o estudo comparou o desempenho dessas técnicas com abordagens tradicionais de filtragem colaborativa e recomendação aleatória. A escolha da Regressão Linear como modelo base permitiu uma análise focada nas políticas e parâmetros de controle de exploração e exploitação, sem a complexidade de modelos mais sofisticados. As políticas *ε-Greedy* e *UCB* foram testadas em diferentes cenários, utilizando tanto a matriz de avaliações quanto informações contextuais dos filmes. Os resultados, considerando uma base de dados reduzida contendo os 0,2% dos avaliadores e filmes mais avaliados, indicaram que, em geral, as abordagens baseadas em *Contextual Bandits*, especialmente aquelas usando a política *ε-Greedy*, superaram as técnicas de filtragem colaborativa e recomendação aleatória. Destaca-se que a utilização do contexto dos filmes como base de treinamento mostrou-se mais eficaz do que a simples matriz de avaliações. Além disso, abordagens que favoreciam uma exploração moderada em relação à exploração excessiva tendiam a ter um desempenho superior. No entanto, alguns desafios foram identificados, especialmente relacionados à avaliação de técnicas de aprendizado por reforço. Este estudo contribui para a compreensão do papel das *Contextual Bandits* em sistemas de recomendação e sugere direções futuras de pesquisa para superar esses desafios e melhorar ainda mais o desempenho dessas abordagens.

**Palavras-chave**: Aprendizado por Reforço, Contextual Bandits, Sistemas de Recomendação, Filtragem Colaborativa, ε-Greedy, Upper Confidence Bound (UCB), MovieLens, IMDb.

## Abstract

This undergraduate thesis investigated the application of Reinforcement Learning techniques, specifically Contextual Bandits, in recommendation systems. Using datasets from MovieLens and IMDb, the study compared the performance of these techniques with traditional approaches such as collaborative filtering and random recommendation. The choice of Linear Regression as the base model allowed for an analysis focused on exploration-exploitation control policies and parameters, without the complexity of more sophisticated models. The ε-Greedy and UCB policies were tested in different scenarios, using both the ratings matrix and contextual information of the movies. The results, considering a reduced database containing the top 0.2% of reviewers and most rated movies, indicated that, overall, Contextual Bandits-based approaches, especially those using the ε-Greedy policy, outperformed collaborative filtering and random recommendation techniques. It is noteworthy that the use of movie context as training base proved to be more effective than the simple ratings matrix. Additionally, approaches that favored moderate exploration over excessive exploration tended to have superior performance. However, some challenges were identified, especially related to the evaluation of reinforcement learning techniques. This study contributes to understanding the role of Contextual Bandits in recommendation systems and suggests future research directions to overcome these challenges and further improve the performance of these approaches.


**Keywords**: Reinforcement Learning, Contextual Bandits, Recommendation Systems, Collaborative Filtering, ε-Greedy, Upper Confidence Bound (UCB), MovieLens, IMDb.

# Lista de Ilustrações

# Lista de Tabelas

## Sumário

# 1. Introdução

Com o crescente volume de dados disponíveis na era digital, os sistemas de recomendação desempenham um papel fundamental na personalização da experiência do usuário em diversas plataformas, como por exemplo serviços de streaming. Esses sistemas utilizam algoritmos sofisticados para prever as preferências dos usuários e sugerir itens relevantes que possam atender às suas necessidades e interesses.

No contexto específico de recomendação de filmes, o desafio é ainda maior devido à grande diversidade de títulos disponíveis e à subjetividade das preferências individuais dos espectadores. Tradicionalmente, abordagens baseadas em filtragem colaborativa e filtragem baseada em conteúdo têm sido amplamente empregadas. No entanto, com o advento do aprendizado por reforço, surge uma nova abordagem promissora que permite aos sistemas aprenderem a recomendar itens interagindo diretamente com o ambiente e recebendo feedbacks em tempo real.

A análise comparativa dessas diferentes técnicas, juntamente com a variação das políticas e dos parâmetros de controle de *exploration-exploitation*, proporcionará insights valiosos sobre o comportamento dessas variadas abordagens na recomendação de filmes. Além disso, a comparação entre a utilização da matriz de avaliações e do contexto dos filmes como entrada para o modelo permitirá avaliar a influência dessas informações na qualidade das recomendações.

# 2. Objetivos

O objetivo deste trabalho é explorar e comparar modelos de recomendação de filmes utilizando técnicas de aprendizado por reforço, especialmente focando na abordagem conhecida como *Contextual Bandits* (uma versão do *Multi-Armed Bandits*, porém com adicionando contexto para auxiliar na tomada de decisão). Com a utilização dos dados da base de dados do MovieLens e do IMDB para treinar e avaliar esses modelos. O objetivo central é empregar políticas de seleção de ações, como *ε-Greedy* e *Upper Confidence Bound* (UCB), em conjunto com a Regressão Linear, a fim de sugerir filmes aos usuários com base em suas características individuais e histórico de avaliações. Além disso, será realizada uma comparação com técnicas de Filtragem Colaborativa, amplamente utilizadas na recomendação de filmes, para determinar a eficácia relativa das abordagens baseadas em aprendizado por reforço em relação a essas técnicas tradicionais.

## 2.1. Revisão Bibliográfica

Nesta seção serão abordados alguns artigos importantes para definição e construção desse trabalho, desde artigos que serviram como inspiração para a temática e aplicações, como também influenciaram em diferentes etapas de estudo para finalização deste trabalho de graduação. Além disso, contribuíram para compreensão da importância do desenvolvimento de novas abordagens para contribuição no campo de recomendação personalizada.

Nessa Revisão Bibliográfica estão apenas alguns dos artigos mais importantes para construção desse trabalho, porém dezenas de outros foram estudados durante o desenvolvimento deste trabalho, cada um com sua devida contribuição, então os artigos que não estiverem presentes nessa revisão estão listados nas Referências deste trabalho.

### 2.1.1. Simulated Contextual Bandits for Personalization Tasks from Recommendation Datasets [1]

Este artigo se propõe a enfrentar uma das maiores dificuldades em aplicações de técnicas de recomendação personalizada, os dados, para isso é proposto um método para transformar as bases de dados de sistemas de recomendação tradicionais em bases de dados com informações suficientes para que se possa aplicar técnicas de maior personalização, como por exemplo, *Contextual Bandits* [1].

A base de dados utilizada como exemplo para aplicação são as bases do MovieLens e do IMDb, além de um ambiente de *Contextual Bandits*. Para o espaço de estados $s \in S$, são consideradas todas as informações disponíveis sobre o usuário, para o espaço de ações $a \in A$ são considerados todas as possíveis recomendações feitas pelo algoritmo, e o espaço de recompensas $r \in R$ estão os feedbacks providos pelos usuários (como por exemplo uma nota, avaliação ou apenas um ‘gostei’/’não gostei’). As abordagens e políticas estão dentro do universo de *Contextual Bandits.* [1].

O artigo não propõe uma solução de aplicação de aprendizado por reforço para esses cenários, apenas um método para construção desses ambientes que podem ser utilizados em treinamentos de abordagens de aprendizado por reforço, e consequentemente, *Contextual Bandits*. O cenário proposto de exemplo, da base de Dados do IMDb e do MovieLens considera os avaliadores do filmes, após uma

normalização do seu comportamento, como o espaço de estados $s \in S$, considera os filmes como sendo o espaço de possíveis ações $a \in A$, ou seja, as recomendações que seriam feitas, e considera as avaliações, que vão de 0,5 à 5,0, em intervalos de 0,5 no MovieLens e vão em um intervalo de 0 a 10, em unidades inteiras no caso do IMDb como sendo o espaço de recompensas $r \in R$ [1].

### 2.1.2. Ensemble Contextual Bandits for Personalized Recommendation [2]

Este artigo se propõe a testar uma abordagem de utilização conjunta (essa combinação de modelos é conhecida como *Ensemble*) de diferentes políticas para solucionar um problema de recomendação personalizada, ou seja, são construídos diversos modelos de recomendação, cada um com uma política diferente, com isso, é feito uma votação entre as recomendações de cada modelo para decidir qual item recomendar ao usuário, essa proposta confere três contribuições aos sistemas de recomendação, o primeiro é uma maior estabilidade nas recomendações, pois como um modelo pode errar há outros modelos para tentar contornar esse erro, a segunda vantagem é para lidar com o problema de *Cold-Start* (Um problema comum dos sistemas de recomendação, onde não há dados iniciais sobre nenhum usuário, o que impossibilita qualquer recomendação personalizada), e o terceiro é uma aplicação em uma base de dados real onde é comparado o desempenho com técnicas já estabelecidas [2].

O ambiente que o artigo propõe a abordagem é de páginas na internet, as recompensas são valores binários de 0 (não clicou no item recomendado na página) e 1 (clicou no item recomendado na página). A criação do ensemble consiste na criação de diversos modelos com diferentes configurações de *Contextual Bandits*, sendo que cada modelo possui uma política diferente do outro, ou seja, são diferentes estratégias e controle do dilema de exploração e exploitação. São propostos dois diferentes algoritmos de ensemble para lidar com o problema de Cold-Start em dados reais, cada um deles com suas próprias configurações de tomada de decisão (a partir do resultado dos diversos modelos integrados no *Ensemble* de cada algoritmo). São utilizadas duas bases de dados, uma de acessos à página de notícias do Yahoo (Yahoo! Today News) e outra do KDD Cup Online Advertising [2].

Foram avaliadas as taxas de cliques a partir de cada abordagem e técnica proposta, além da comparação com as decisões das políticas isoladas e de um tomador de decisão aleatório, onde os algoritmos de ensemble obtiveram melhor resultado, contribuindo assim, para novas abordagens em problemas de sistemas de recomendação e solução de Cold-Start [2].

### 2.1.3. Contextual-Bandit Based Personalized Recommendation with Time-Varying User Interests [3]

Este artigo aborda os problemas de recomendação como sistemas dinâmicos, não estáticos, ou seja, sistemas que mudam com o tempo, de forma a trazer as aplicações de aprendizado por reforço, que muitas vezes são construídos em ambientes estáticos para ambientes mais próximos da vida real, onde os usuários tem seus gostos e interesses alterados pelo tempo, seja pelo passar do tempo natural da vida da pessoa (por exemplo, adolescência para vida adulta), novos conhecimentos adquiridos, e quaisquer outros motivos que fazem com que as pessoas mudem seus interesses [3].

São propostas duas abordagens para lidar com esse novo sistema, no primeiro deles, chamado de ‘modelo de payoff disjunto’, é utilizado uma política UCB para fazer as recomendações a partir dos dados do passado, porém o algoritmo também faz uma detecção de mudança de interesse, e caso seja detectado essa possível mudança as recomendações são recalculadas. A segunda proposta de abordagem é uma adição à primeira abordagem, onde é nomeada de ‘payoff híbrido’, nessa técnica é adicionado um coeficiente invariante no tempo, que considera todas as ações possíveis [3].

Ambas as técnicas propostas obtiveram melhor desempenho do que técnicas já conhecidas pela literatura e do que uma abordagem totalmente randômica, sendo que, para quaisquer configurações de políticas, aquelas que utilizaram a *Híbrida* foram melhores do que suas equivalentes que utilizaram apenas a *Disjunta*, e consequentemente, muito melhores que as técnicas de comparação. Esses resultados abrem caminhos para novas abordagens e técnicas que lidem com variações temporais e assim, possam aproximar os algoritmos de recomendação personalizada de comportamentos humanos que são, naturalmente variantes no tempo [3].

### 2.1.4. Contextual Bandit Approach-based Recommendation System for Personalized Web-based Services [4]

Este artigo investiga a performance de 3 diferentes algoritmos de recomendação personalizada baseada em *Contextual Bandits* em diversas bases de dados diferentes, desde bases com dados sintéticos, criado com base em estratégias aleatórias, até bases com dados reais de larga escala, como o Yahoo! Today module, uma base de dados de acessos em portais de notícia do Yahoo, LastFM, uma base de dados de um serviço de streaming de músicas e por fim, o MovieLens20M, uma base de dados com avaliações de filmes, sendo este uma versão anterior da base de dados utilizada neste projeto [4].

As simulações foram realizadas limitando os usuários e o número de itens em quantidades pré-fixadas, sendo que para cada base de dados foram estipulados números diferentes de usuários e itens, e realizadas as simulações para todas as combinações possíveis de técnica, base de dados e quantidade de usuários e itens. Os resultados deste artigo sugerem que os algoritmos de recomendação baseados em *Contextual Bandits* não foram comparados com técnicas já estabelecidas no mercado, mas sugerem que há ótimas oportunidades de uso de *Contextual Bandits* para sistemas de recomendação personalizados, além da possibilidade de avanços nessa área [4].

# 3. Fundamentação Teórica

## 3.1.1. Sistemas de Recomendação

### 3.1.1.1. Visão geral

Um sistema de recomendação é uma abordagem que visa, através de diferentes técnicas e ferramentas, encontrar o melhor item para um determinando usuário (Há vários exemplos em diversos contextos, pode ser um filme, como exemplo de um serviço de Streaming, ou produto de consumo, como exemplo de alguma loja online). Um ‘item’ é um termo genérico utilizado para qualquer coisa que será recomendada a um usuário através do sistema de recomendação, no cenário que será retratado nesse trabalho o ‘item’ é referente ao filme a ser recomendado. O ‘usuário’ é a pessoa a quem será recomendado o ‘item’, então neste trabalho, será feita a recomendação de um filme a um usuário [5].

Há diversas formas de se construir sistemas de recomendação, seja pelo histórico do usuário (por exemplo, quais filmes já assistiu em uma determinada plataforma), pelo perfil do usuário (informações demográficas, perfil social, preferências declaradas ou quaisquer outras informações individuais sobre o respectivo usuário) ou pelas próprias informações do item a ser recomendado (por exemplo, em filmes há informações de duração do filme, gênero, ano de lançamento, etc), e é possível fazer combinações para dessas diferentes abordagens para construção do sistema de recomendação.

Em relação ao formato dos dados de entrada de um sistema de recomendação, normalmente é utilizado uma matriz onde um dos eixos são os itens a serem recomendados e o outro eixo são os usuários, sendo que os valores preenchidos em cada célula dependem do objetivo do sistema de recomendação, e essa matriz simplificada pode ser visualizado na Figura 1, onde o eixo y possui os diferentes usuários do sistema e o eixo x possui os itens (filmes) disponíveis nesse mesmo sistema, e cada célula está preenchida com um símbolo que representa se aquele item foi consumido pelo respectivo usuário.


*Figura 1: Exemplo de matriz de recomendações*

Sobre os objetivos do sistema de recomendação eles podem ser simples, como por exemplo, um sistema que gera uma lista de itens a serem recomendados ao usuário baseado apenas no fator de ‘consumiu’ ou ‘não consumiu’ o item, mas também podem ser mais complexos, onde é possível realizar as recomendações de forma a maximizar a experiência do usuário, então por exemplo, em um sistema onde a entrada são as notas dadas pelos usuários aos itens, é possível construir um sistema que visa maximizar essas notas, então o objetivo aumenta sua complexidade de ‘recomendar um item’ para ‘recomendar o melhor item’ [5].

Os sistemas de recomendação podem ser gerados por diferentes estratégias e técnicas, uma das mais comuns é a *Filtragem Colaborativa*, uma técnica que visa comparar usuários e itens parecidos a fim de realizar as próximas recomendações e essa técnica será utilizada como referência para comparação com as técnicas de *Bandidos Contextuais*.

### 3.1.1.2. Filtragem colaborativa

A filtragem colaborativa é uma das técnicas mais comuns e eficazes utilizadas em sistemas de recomendação. Ela se baseia na ideia de que as preferências de um usuário podem ser preditas com base nas preferências de outros usuários semelhantes, através de uma matriz de usuário-item. Em outras palavras, o sistema de recomendação busca recomendar itens para um usuário com base nas opiniões e

comportamentos de usuários semelhantes, com a premissa de que, se diferentes usuários concordaram no passado, provavelmente concordarão no futuro [6].

A filtragem colaborativa é eficaz porque não requer informações detalhadas sobre os itens sendo recomendados. Em vez disso, ela se baseia nas opiniões e comportamentos dos próprios usuários para fazer previsões sobre as preferências de um usuário. No entanto, ela pode enfrentar desafios como a grande esparsidade dos dados (quando há poucas avaliações disponíveis) e o problema do *ColdStart* (quando há poucas informações sobre novos usuários ou itens) [6]. Existem dois principais tipos de filtragem colaborativa, a baseada em usuário e a baseada em item, elas podem ser melhor visualizadas na Figura 2.

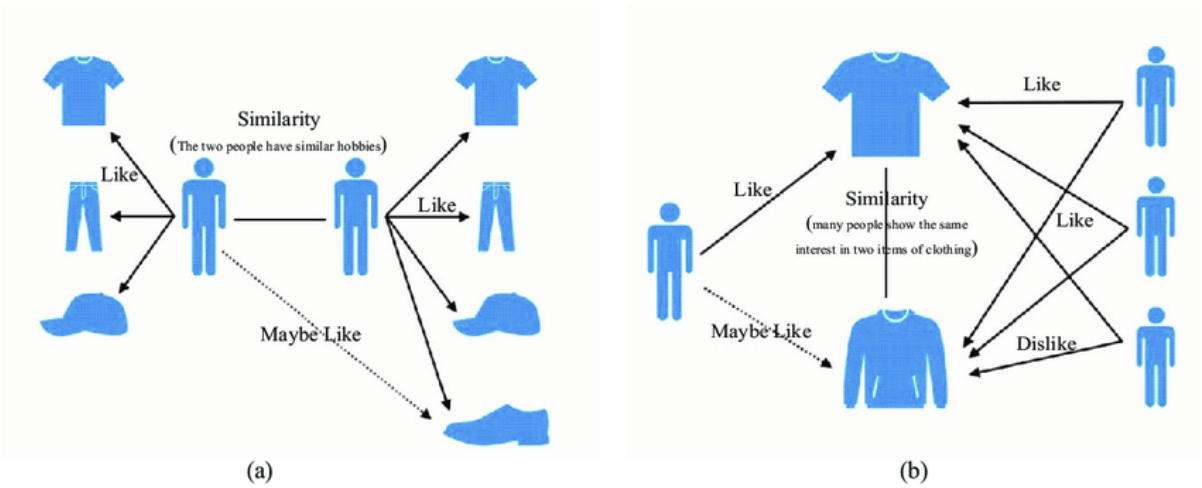


*Figura 2: Filtragem Colaborativa. (a)Baseada em usuário. (b)Baseada em item. [7]*

### 3.1.1.3. Filtragem colaborativa baseada em usuário

Nesta abordagem, presente na Figura 2 (a), o sistema de recomendação compara o perfil de um usuário com o perfil de outros usuários e identifica usuários com gostos semelhantes. Com base nessas semelhanças, o sistema recomenda itens que foram preteridos por outros usuários semelhantes [8].

No exemplo presente na Figura 2 (a), ambos os usuários gostam do mesmo tipo de roupa (camiseta de manga curta, calça e boné), porém, o usuário da direita também gosta de um sapato, o qual, não temos informação sobre a relação do usuário da esquerda com o sapato, porém, por possuírem outros gostos em comum, o sistema recomenda o sapato para o usuário que o desconhece.

### 3.1.1.4. Filtragem colaborativa baseada em item

Nesta abordagem presente na Figura 2 (b), o sistema de recomendação identifica itens semelhantes com base nas avaliações dos usuários. Se um usuário gostou de um determinado item, é provável que ele também goste de itens semelhantes. O sistema usa essas informações para recomendar itens similares aos que o usuário já gostou [8].

No exemplo da Figura 2 (b), os usuários da direita realizaram avaliações sobre a camiseta de manga curta, como essa camiseta se assemelha com a blusa de frio, então para aqueles que gostaram da camiseta de manga curta, será recomendado a blusa de frio pois provavelmente esse usuário iria gostar também.

#### 3.1.1.5. Desafios

Há diversos desafios ao lidar com sistemas de recomendação, dentre eles há o problema de dados muito esparsos, ou seja, os conjuntos de dados de interação usuário-item geralmente não são preenchidos, o que significa que os usuários podem ter interagido apenas com alguns itens, gerando matrizes muito grandes (N usuários vezes M filmes) com poucos valores de fato preenchidos, tornando difícil fazer recomendações precisas. Outro problema comum é o ColdStart, esse problema se dá quando é adicionado um novo usuário ou item ao sistema, pois em ambos os cenários essa nova coluna/linha entrará totalmente vazia pela falta de histórico, o que também dificulta em novas recomendações para esse usuário/item. Uma outra questão também muito relevante nos sistemas de recomendação é a escalabilidade, pois quanto mais cresce a matriz mais recursos computacionais são necessários para calcular de forma eficiente e rápida as próximas recomendações (independente da técnica ou abordagem utilizada, matrizes esparsas muito grandes continuam sendo problemas computacionais a serem otimizados).

Além dos desafios mais comuns de um sistema de recomendação há também outros desafios que são mais facilmente solucionados com técnicas mais robustas, mas também exigem um certo grau de atenção, como a ultra especialização, onde o sistema de recomendação pode ficar enviesado em alguns poucos itens, recomendado apenas os mesmos para todos os usuários. Um outro desafio que também possui uma solução na implementação, mas não deve ser ignorado, são as criações de bolhas, então o sistema encontra pessoas parecidas com gostos similares e recomenda e as recomendações dos usuários ficam presas em alguns itens que

estão contidos na bolha, impedindo assim que outros itens sejam recomendados para o usuário.

### 3.1.2. Aprendizado de Máquina

Aprendizado de máquina faz parte do grande campo da Inteligência Artificial onde o objetivo é construir modelos matemáticos, com as mais diversas complexidades, de forma automatizada, sendo necessário apenas o fornecimento dos dados para o modelo, cujo fim é encontrar a melhor função que explique as alterações do valor de uma variável (conhecida como “variável alvo” ou “variável resposta”) em resposta à outras variáveis (conhecidas como “variáveis explicativas”).

Dentro do aprendizado de máquina, há duas abordagens mais populares, a aprendizagem supervisionada e não-supervisionada, na primeira abordagem o treinamento é realizado a partir de dados já rotulados, onde sabemos a resposta previamente e pretendemos construir um modelo capaz de rotular novos dados não rotulados, assim, já possuímos algumas um conjunto de treino que pode ser modelado, na segunda abordagem os dados não possuem rótulos, o objetivo desse tipo de abordagem é, através de suas variáveis explicativas, encontrar semelhanças e/ou padrões entre as observações, assim, todas as observações são utilizadas para construir o modelo, e a validação desse modelo, é sobre o próprio dado de treino.

Além dessas duas abordagens mais populares, há também uma terceira abordagem, sendo ela conhecida como aprendizado por reforço, e esse subcampo do aprendizado de máquina tem ganhado mais popularidade nos últimos anos, principalmente à diversas empresas que tem popularizado a utilização de suas técnicas para solução de jogos e aplicações do cotidiano.

### 3.1.3. Aprendizado Supervisionado

O aprendizado supervisionado pode ser dividido em dois tipos de problemas, classificação e regressão. Problemas de classificação envolvem uma variável resposta discreta, onde os valores normalmente são binários e sua solução é para um problema dicotômico (Por exemplo, 1 ou 0). Já problemas de regressão são problemas onde os valores da variável resposta são contínuos, mas não necessariamente infinitos, e seus valores possuem uma ordenação. Para este projeto será utilizado o modelo de Regressão Linear.

#### 3.1.3.1. Regressão Linear

A regressão linear é um tipo de modelo estatístico que tenta modelar uma função linear que explique a relação entre as variáveis independentes (variáveis explicativas) e a variável dependente (variável resposta), esse modelo pode ser expressado da seguinte forma:

$$y = \beta_0 + \beta_1 * X1 + \beta_2 * X2 + \cdots + \beta_n * Xn$$

Onde y é a variável dependente (variável resposta), $\beta_i$ são os coeficientes que são encontrados pelo modelo, e $X_i$ são as variáveis independentes (variáveis explicativas). [9]

Para estimar os coeficientes normalmente é utilizado o método dos mínimos quadrados, que é um método que busca reduzir a soma do quadrado dos resíduos (a diferença entre o valor real e o valor previsto pelo modelo) [9].

## 3.2. Aprendizado por Reforço

### 3.2.1. Visão Geral

O Aprendizado por Reforço, diferentemente do aprendizado supervisionado e não-supervisionado, foca em maximizar a recompensa total de um agente quando este interage com um ambiente, além disso, é através de tentativa e erro e recompensas a longo prazo que fazem com que esse tipo de aprendizado seja tão particular.

A arquitetura básica do aprendizado por reforço está representada na Figura 3, na qual consiste em um ambiente (do inglês, *Environment*), um agente (do inglês, *Agent*), uma função de recompensa (do inglês, *Reward*), uma função de estado (do inglês, *State*) e uma função de ação (do inglês, *Action*). Sendo que, o *agente* é o tomador de decisões, é o componente que, estando em um respectivo *estado* do *ambiente*, realizará uma *ação* através de uma determinada estratégia, o *ambiente* é o espaço na qual o *agente* irá interagir, todos os *estados* são possíveis caminhos que o *agente* pode interagir dentro desse *ambiente*, as *recompensas* são uma forma quantitativa de avaliação da decisão tomada pelo *agente*, onde maiores *recompensas* implicam em melhores *ações* tomadas para um determinado *espaço* [10].

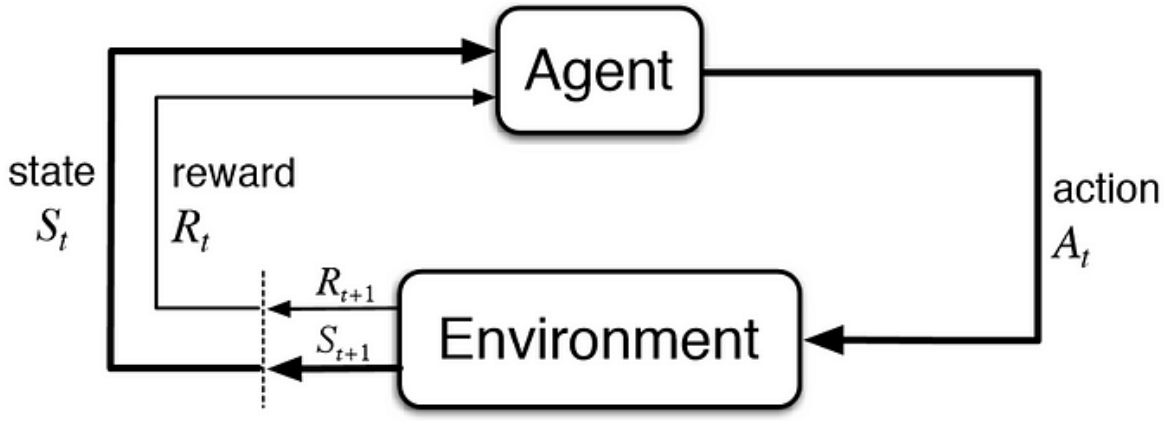


*Figura 3: Esquema básico de Aprendizado por Reforço. [10]*

No aprendizado por reforço, o agente busca aprender uma política ótima, que maximize a recompensa total através da tomada de decisão no ambiente que ele interage, para isso é necessário formular matematicamente o processo de aprendizagem e tomada de decisão, e a forma mais comum de se fazer isso é através do Processo de Decisão de Markov (Também conhecido como MDP, do inglês, *Markov Decision Process*). O MDP, pode ser definido pela tupla $(S, A, P, R, \gamma)$, onde $S$ representa os possíveis estados do ambiente, $A$ denota as possíveis ações que o agente pode tomar, $P$ representa as probabilidades de transição de um estado para outro, $R$ indica a função recompensa e $\gamma$ é um fator de desconto para recompensas.

O aprendizado funciona da seguinte forma, para cada estado *t*, o agente, estando em um estado $S_t \in S$ e tomando uma ação $A_t \in A$, irá receber uma recompensa $R_{t+1} \in R$ e terminará a rodada no estado $S_{t+1} \in S$. Dessa forma, o MDP pode ser representado por $\tau$, e a sequência de ações pode ser representada pela sequência abaixo [11].

$$\tau = \{S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_t\}$$

Onde T é o último estado possível do MDP. Como o objetivo do agente é maximizar a recompensa total, ele tenta tomar as ações que garantam uma maior recompensa acumulada, para isso, é preciso compreender o desconto, que é um peso dado para recompensas anteriores de forma a minimizar o seu valor conforme o respectivo momento onde essa recompensa foi adquirida se afasta do momento atual, e essa recompensa acumulada com desconto pode ser definida pela equação abaixo [11].

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{i=t+1}^{T} \gamma^{i-t-1} R_i$$

Ou seja, caso $\gamma = 0$ o agente irá maximizar somente a recompensa mais atual possível, e quanto mais próximo de 1 $\gamma$ se aproximar o agente irá priorizar mais recompensas futuras [11].

Dentro do universo de Aprendizado por Reforço, há duas principais abordagens, *baseada em modelo* (do inglês, *Model-based*) e *sem modelo* (do inglês, *Model-Free*), no *Model-based* o agente precisa de um modelo do ambiente construído, onde o agente pode prever consequências e planejar ações futuras, já o *Model-Free* não possui um modelo, então o agente precisa aprender diretamente com suas ações, através de diferentes interações ele vai estimando as recompensas a serem recebidas, pelo fato do *Model-Free* não necessitar da construção de um modelo do ambiente, ele acaba sendo mais simples e mais utilizado.

## 3.2.2. Multi-Armed Bandits

### 3.2.2.1. Conceito

Primeiramente é necessário entender o nome *Multi-Armed Bandits* para melhor compreender seu conceito, que está representado graficamente na Figura 5.

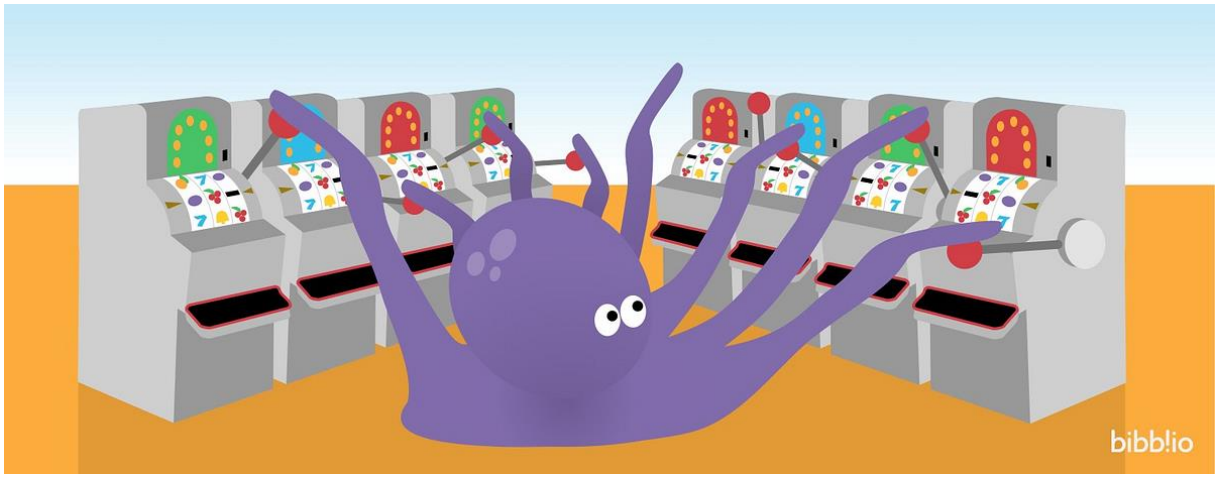


*Figura 4: Representação gráfica do Multi-Armed Bandits. [12]*

O termo *Bandits* remete à caça-níqueis, um termo utilizado para resumir a função dessas máquinas, *roubar* o dinheiro de seus jogadores, a escolha desse termo é devido ao fato de que, ao jogarmos nessa máquina, não sabemos a probabilidade de ganhar e nem o valor real do prêmio, além disso, uma mesma máquina pode ter mais de uma alavanca para puxar, que nesse contexto é chamado *Armed*, onde cada alavanca possui uma probabilidade diferente de recompensa, essa explicação está representada na Figura 4, devido à essa máquina com várias possíveis ações com

diferentes recompensas foi dado o nome de “Bandido de vários braços” (do inglês, *Multi-Armed Bandits*) para esse tipo de problema, onde é desejado estimar as possíveis recompensas para diferentes ações, de forma a maximizar a recompensa total adquirida.

A solução genérica do problema de *Multi-Armed Bandits* consiste em escolher a ação que maximiza a recompensa total dentre as possíveis ações disponíveis $a \in A$, onde cada ação tem uma recompensa $R$ com probabilidade $P$, porém, na maioria dos casos, essas probabilidades e recompensas não são conhecidas, por isso é necessário, através de diferentes estratégias, estimar quais são seus valores reais, a fim de maximizar essa recompensa, essa solução genérica consiste na tentativa de estimar as possíveis recompensas e probabilidades de cada escolha, ou seja, para uma ação a tomada no instante $t$ é retornado uma recompensa com uma certa probabilidade, para isso, constrói-se uma função de valor esperado de recompensa para cada ação, como a equação abaixo demonstra. Onde é desejável realizar as ações com o maior valor esperado possível.

$$q(a) = \mathbb{E}[R_t | A_t = a]$$

As estratégias para maximizar o $q(a)$ são diversas, mas elas são basicamente diferentes configurações de ações baseadas em tentativa e erro, e como realizar essas tentativas de estimação da forma mais inteligente possível, sempre visando a maximização da recompensa, e essas estratégias serão melhor descritas em 3.2.43.2.4.

Uma das formas de estimar o *Valor Esperado* de uma ação é através de um método conhecido como *Ação-Valor*, onde é calculado a soma total de recompensa obtida por uma ação sobre o total de vezes que aquela ação foi tomada, que pode ser explicitada matematicamente da seguinte forma,

$$Q_t(a) = \frac{Soma\ das\ recompensas\ obtidas\ na\ ação\ "a"\ até\ o\ momento\ t}{Total\ de\ vezes\ que\ ação\ "a"\ foi\ tomada\ até\ o\ momento\ t} = \frac{\sum^t R_a}{\sum^t a}$$

Com essa equação, é possível estimar os *Valores Esperados* de recompensa para cada ação do sistema, sendo assim, a ação a ser tomada até o momento $t$ ($A_t$) é aquela que maximiza essa equação, podendo ser definida como

$$A_t = argmax(Q_t(a))$$

Dessa forma, é possível maximizar as recompensas obtidas, porém, ainda é necessário realizar ações seguindo alguma política de tomada de decisão, e essas políticas serão explicadas em outro tópico.

#### 3.2.2.2. Variações

Há várias versões diferentes desse mesmo problema, na versão mais simples do *Multi-Armed Bandits* os valores de recompensa para cada ação é o mesmo para todos, porém as probabilidades variam entre cada possível ação, nesse tipo de problema, as estratégias consistem em basicamente estimar apenas as probabilidades de recompensa para cada ação, dado que os valores de recompensa são os mesmos. Por exemplo, para duas diferentes ações, uma ação “A” com probabilidade estimada de 50% de recompensa e outra ação “B” com probabilidade estimada de recompensa de 10%, a ação A possui um valor esperado de recompensa de $\frac{R}{2}$ e a ação B possui um valor esperado de $\frac{R}{10}$, portanto temos que,

$$q(A) = \frac{R}{2} > \frac{R}{10} = q(B)$$

$$q(A) > q(B)$$

Sendo assim, para esse cenário do exemplo, a melhor ação a tomar é ação “A”.

Uma outra versão desse problema tem as recompensas para cada ação diferentes, além das probabilidades, assim, as estratégias para solucionar esse tipo de variante devem levar em conta que, mesmo probabilidades muito baixas podem levar a recompensas maior, fazendo sentido então, estratégias que busquem fazer mais testes em ações com recompensas imediatas baixas, apenas para aumentar a confiança nessas ações. Por exemplo, para duas diferentes ações, com suas recompensas e probabilidades reais desconhecidas a priori, uma ação “A” com recompensa de $100R$ e uma probabilidade de recompensa de 1% e uma ação “B” com recompensa de $1R$ e probabilidade de recompensa de 50%, uma estratégia que faz poucos testes pode acabar estimando que o *Valor Esperado* da ação “A” é menor que o da ação “B”, pois sua probabilidade de obter uma recompensa é menor, porém os valores esperados são

$$q(A) = \frac{100R}{100} > \frac{1R}{2} = q(B)$$

$$q(A) = \frac{1R}{1} > \frac{1R}{2} = q(B)$$

$$q(A) > q(B)$$

Ou seja, o *Valor Esperado* da ação “A” é maior, mesmo que possua uma probabilidade menor, então utilizar uma estratégia que faça poucos testes e conclua que a recompensa esperada é 0, pode levar a uma ação que não é a que maximiza a recompensa total.

E por fim, há diversas outras variações, como por exemplo uma variante onde as recompensas ou as probabilidades podem mudar com o tempo, sendo necessários estratégias que estejam constantemente revendo as ações que já foram tomadas anteriormente para garantir que não mudaram seu valor esperado, e caso tenha mudado, fazer novas estimativas para garantir que a melhor ação continua sendo tomada. Variações onde são introduzidas novas possíveis ações ou ações ótimas deixam de existir, sendo assim necessário também contar com estratégias que possam lidar com essas adversidades. Existem uma infinidade de variações desse problema, mas todos precisam lidar com a mesma questão, que é construir um modelo que maximize a recompensa total ao realizar cada ação, para isso, é preciso equilibrar entre fazer as escolhas que já se tem um certo conhecimento, e que possuem uma boa recompensa, ou explorar novas ações onde não se há uma certeza sobre a verdadeira possibilidade de ganho e assim, acabar encontrando um caminho com um retorno maior ainda do que o já conhecido, esse dilema é conhecido como *Dilema de Exploração e Exploitação* (do inglês, *Exploration-Exploitation Trade-off*), esse dilema e suas possíveis soluções serão mais explicadas em outro tópico desse trabalho, além das possíveis estratégias para solução do *Multi-Armed Bandits*.

## 3.2.3. Dilema de Exploração e Exploitação

### 3.2.3.1. Conceito

Há um dilema interessante que deve ser resolvido no contexto de Multi-Armed Bandits, esse dilema é baseado no equilíbrio de exploração e exploitação (do inglês, *Exploration and Exploitation Trade-off*).

Esse dilema da exploração e exploitação está baseado na ideia de intercalarmos, de acordo com alguma política de tomada de decisão, entre a escolha de uma ação em

que a recompensa esperada é a melhor possível e já possuímos um certo grau de confiança nesse valor de recompensa e a escolha de uma ação em que não possuímos uma confiança tão grande sobre a recompensa real, ou seja, a recompensa esperada pode ser melhor do que a recompensa esperada do estado/ação que já possuímos maior confiança. Exploitação é quando a escolha da ação é tomada baseada na melhor recompensa disponível (baseado no conhecimento e confiança que possuímos nas possíveis ações), Exploração é quando a escolha da ação é tomada baseada na possibilidade de uma recompensa maior em uma ação menos conhecida, com menor confiança de sua recompensa real. [13]

#### 3.2.3.2. Exemplo

Um exemplo desse dilema pode ser visto na Figura 5, onde um robô precisa tomar a decisão de ir a um lugar que ele já conhece (restaurante da esquerda, com o nome “The Usual Place”) ou ir a um lugar que ele ainda não conhece (restaurante da direita, com o aviso de “Grand Opening!”). Caso o robô escolha pela Exploitação, irá para o restaurante da esquerda, onde já conhece a recompensa esperada com uma certa confiança, caso ele opte pela Exploração, irá para o restaurante da direita, onde não há conhecimento suficiente sobre a recompensa esperada, mas pode ter uma recompensa maior que a recompensa do lugar já conhecido.

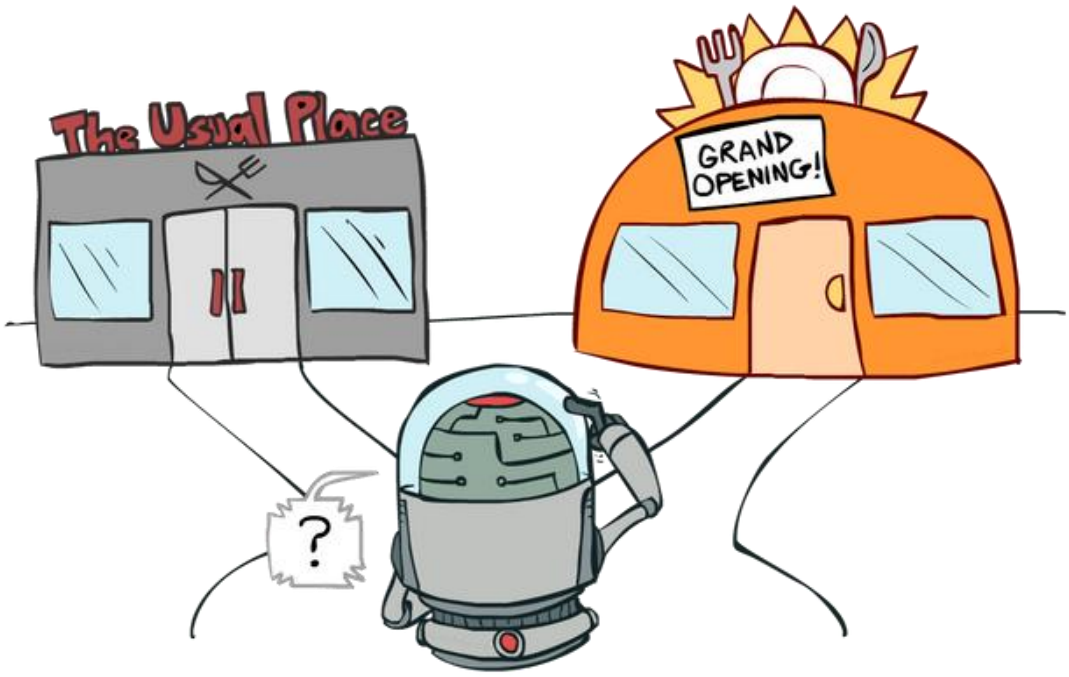


*Figura 5: Dilema Exploração e Exploitação. [14]*

Há diversas *políticas* para lidar com o equilíbrio desse dilema de Exploração e Exploitação e otimizar a tomada de decisões, dentre eles os mais famosos são o ε-Greedy e o Upper Confidence Bound (UCB) e nesse trabalho serão explorados mais a fundo as duas primeiras políticas em um outro tópico.

## 3.2.4. Políticas

### 3.2.4.1. ε-Greedy:

A política ε-Greedy (Ou *Epsilon-Greedy*, em sua escrita por extenso) consiste em escolher entre a melhor ação, ou seja, a ação que até o momento possui a melhor recompensa $\mathrm{Q_t}$ (também chamada de *Greedy*), e uma escolha aleatória com probabilidade ε, sendo representada matematicamente da seguinte forma:

$$A_t \begin{cases} Max\ Q_t & com\ probabilidade\ 1-\varepsilon \\ Ação\ aleatória & com\ probabilidade\ \varepsilon \end{cases}$$

Essa política é uma das mais simples e de fácil compreensão do seu funcionamento, apesar disso, é preciso ter cuidado na escolha do valor ε, pois caso ε seja muito grande, a estratégia mais favorecida será de *Exploração*, pois as decisões favorecerão o ganho de maior confiança em todas ações e não garantir uma maior recompensa a curto prazo, já para o caso de um ε pequeno, as escolhas serão principalmente na ação que possui a melhor recompensa conhecida com um certo grau de confiança, aumentando ainda mais a confiança naquela ação e sua recompensa esperada, porém, sem a possibilidade de explorar ações que possam ter uma recompensa ainda maior.

$$\varepsilon > 0.5 \qquad Maior\ confiança\ em\ todas\ a \in A; Maior\ exploração.$$

$$\varepsilon < 0.5 \qquad Maior\ confiança\ na\ MaxQ_t;\ Maior\ exploitação.$$

$$\varepsilon = 0.5 \qquad Equilibrio\ entre\ exploração\ e\ exploitação.$$

Há também uma variante do ***ε-Greedy*** onde o ε pode alterar o seu valor conforme o tempo passa, dessa forma, é possível combinar diferentes estratégias, podem ser uma *Exploração* bem grande no começo (com ε grande), aumentando a confiança na recompensa esperada de todas as ações e, posteriormente, uma *Exploitação* dos estados com maior recompensa esperada (com ε decrescendo com o tempo), resolvendo de forma otimizada esse dilema [13].

#### 3.2.4.2. Upper Confidence Bound

Na política de *Upper Confidence Bound*, o que é levado em consideração são as possibilidades de ganhos superiores, ao invés do ganho esperado que temos conhecimento, o Limite de Confiança Superior (do inglês, *Upper Confidence Bound*) pode ser descrito, em sua forma genérica e mais simples, dessa maneira:

$$UCB = Qt(a) + C * \sqrt{\frac{ln\ (t)}{Nt(a)}}$$

Onde o termo *Qt(a)* é a recompensa esperada na escolha da ação *a*, na rodada *t*, ou seja, é o próprio valor esperado de acordo com algum modelo que esteja prevendo esse valor, a variável *C* controla o quanto exploramos e exploitamos, sendo que quanto maior o seu valor, maior o poder de explorarmos novas possibilidades e ações que temos menor confiança de seu real valor. O termo *t* se refere à quantidade de jogadas, então o termo *ln(t)* é o logaritmo natural e *t* é o total de jogadas, e o termo *Nt(a)* é o total de escolha daquela ação *a*, ou seja, quanto menos vezes escolhemos a ação *a*, maior será seu valor a direita, por consequência, maior será seu o termo da direita da equação e maior será o *limite de confiança superior* [13].

Para escolha de ações utilizando essa política, é escolhido sempre o maior valor de UCB, por isso é muito importante encontrar um equilíbrio no dilema de exploração e exploitação.

### 3.2.5. Contextual Bandits

Diferente da solução proposta pelo *Multi-Armed Bandits*, onde cada ação é tomada com base apenas na busca pela recompensa acumulada a partir de uma certa política, quando entramos no universo de *Contextual Bandits* é preciso lidar também com o contexto (variáveis com informações sobre o usuário ou item) antes de tomar uma decisão (ação), além do fato desse contexto não ser imutável pelo tempo, podendo sofrer alterações no decorrer do tempo [13].

O “contexto” do *Contextual Bandits* seria análogo às variáveis explicativas de um modelo supervisionado, caracterizando aquele usuário e diferenciando de outros usuários, de forma a individualizar as recomendações e torná-las mais personalizadas à cada usuário. O contexto de um usuário seriam as suas características, como por exemplo ‘idade’, ‘renda’, ‘estado civil’, ‘quantidade de filhos’, entre outros, já o contexto

de um filme, em um cenário de recomendação de filmes também são suas características, por exemplo ‘duração’, ano de lançamento’, ‘faixa etária’, ‘orçamento’, etc. O contexto pode ser generalizado para quaisquer variáveis explicativas que podem ser utilizados para ajudar um modelo de *Contextual Bandits* a tomar decisão sobre qual ação será tomada [13].

O esquema de funcionamento do *Contextual Bandits* pode ser visualizado, de forma simplificada e análoga à estrutura padrão de aprendizado por reforço, no diagrama da Figura 6.

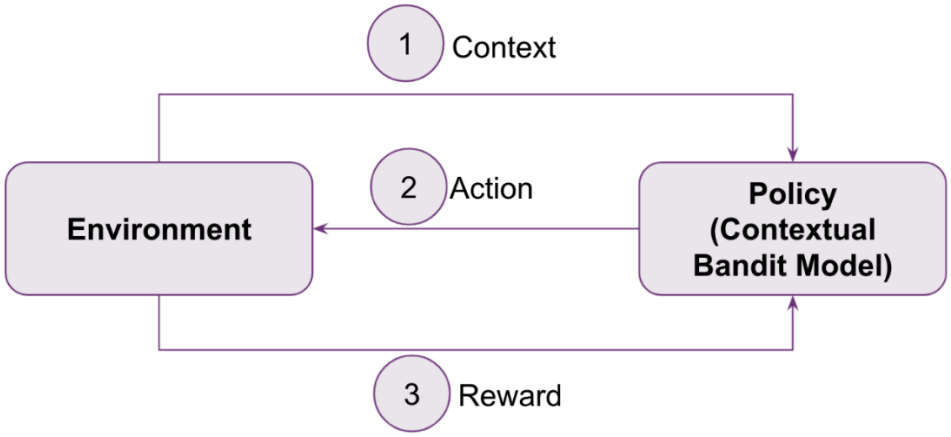


*Figura 6: Diagrama simplificado de Contextual Bandits. [15]*

Nesse diagrama, primeiro é dado um contexto, ou seja, as informações que o modelo pode usar para ajudar a tomar a decisão da melhor ação a ser tomada e esse contexto está representado em *(1) Context*, com o contexto em mãos, a partir de uma certa política já determinada, representada pelo retângulo na direita *Policy (Contextual Bandit Model)* é tomada uma decisão, e essa ação, representada pelo *(2) Action,* interage com o ambiente, representado pelo retângulo à esquerda *Environment*, esse ambiente então retorna qual a recompensa por essa ação tomada, representado por *(3) Reward*, assim, o modelo começa modelar a recompensa esperada para cada ação que possa tomar, porém esse valor esperado é construído com base nas informações de contexto de cada usuário, e não simplesmente em cada ação individualmente, como era feito no *Multi-Armed Bandits.*

#### 3.2.5.1. Aplicação em Sistemas de Recomendação

Considerando a função dos algoritmos de *Contextual Bandits*, que é selecionar a ação mais adequada para uma situação específica, podemos traçar um paralelo com os

sistemas de recomendação abordados neste trabalho. Tomando como exemplo um sistema de recomendação de filmes, a implementação de *Contextual Bandits* visa identificar a melhor opção de recomendação, ou seja, sugerir um filme que otimize a experiência do usuário. Este processo leva em conta o contexto atual, representado pela matriz de relação item-usuário, que compila os filmes já vistos e avaliados. O objetivo é determinar qual filme recomendar aos usuários de modo a maximizar a probabilidade de uma avaliação positiva para o filme sugerido.

A adaptação de técnicas de *Contextual Bandits* para solucionar problemas de sistemas de recomendação começa em definir o cenário e os parâmetros. O cenário a ser exemplificado é um cenário de recomendação de filmes, onde cada usuário avalia numericamente o filme assistido após sua respectiva recomendação pelo algoritmo, ou seja, quanto maior o valor numérico da avaliação, melhor foi essa recomendação (mais o usuário gostou do filme).

O ambiente é a própria matriz de avaliações, onde um dos eixos é de usuários e o outro eixo é de filmes, e cada célula dessa matriz esparsa representa a nota dada pelo usuário ao filme, como pode ser visualizado no exemplo abaixo, com 3 usuários e 4 filmes.

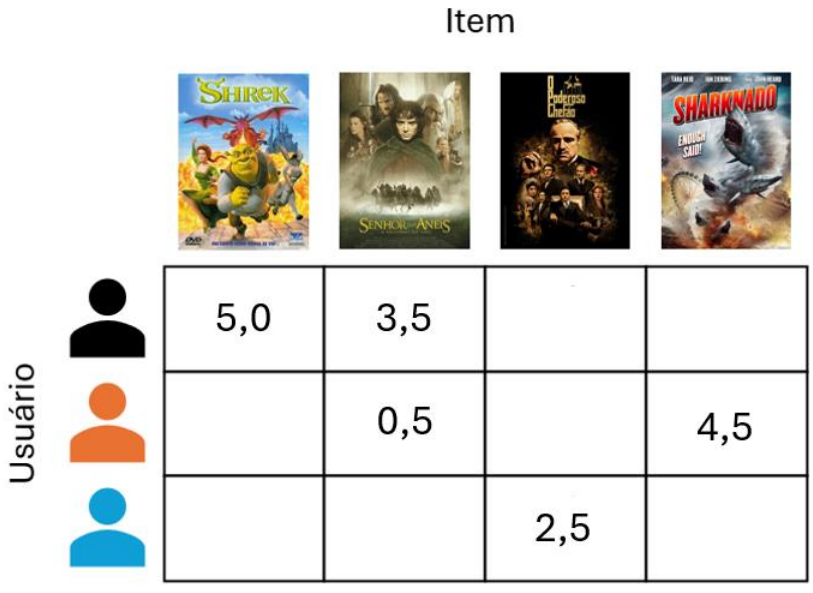


*Figura 7: Exemplo de matriz com avaliações numéricas*

Esse ambiente é apenas para exemplificar, pois ambientes de recomendações de filmes costumas possuir milhares ou até milhões de usuários e itens, sendo que apenas uma fração muito pequena está preenchida com alguma avaliação.

O estado é a própria matriz com suas respectivas avaliações já realizadas, ou seja, para cada nova avaliação é avançado do estado $S_t$ para o estado $S_{t+1}$, onde $S_t \in S$.

As ações são as próprias recomendações a serem feitas, sendo que o objetivo é maximizar a recompensa de cada ação, ou seja, recomendar o melhor filme possível, porém, diferentes estratégias e técnicas visam aumentar a recompensa a longo prazo, então nem sempre são esperadas recompensas altas a curto prazo.

O funcionamento genérico de um algoritmo de *Contextual Bandits* em um sistema de recomendação começa por observar o estado atual, ou seja, quais filmes já foram avaliados, em seguida, é feito através de uma combinação de algum algoritmo específico com alguma estratégia também já escolhida, o cálculo da probabilidade de melhores avaliações de cada filme caso seja recomendada a cada usuário, assim, é possível construir uma lista de melhores ações a serem tomadas (melhores filmes a serem recomendados a cada usuário), em seguida é feito a recomendação do filme ao usuário e obtido o feedback instantaneamente, dessa forma, a matriz é atualizada com a nova avaliação, as métricas são calculadas e o processo reinicia, como é exemplificada na Figura 8.

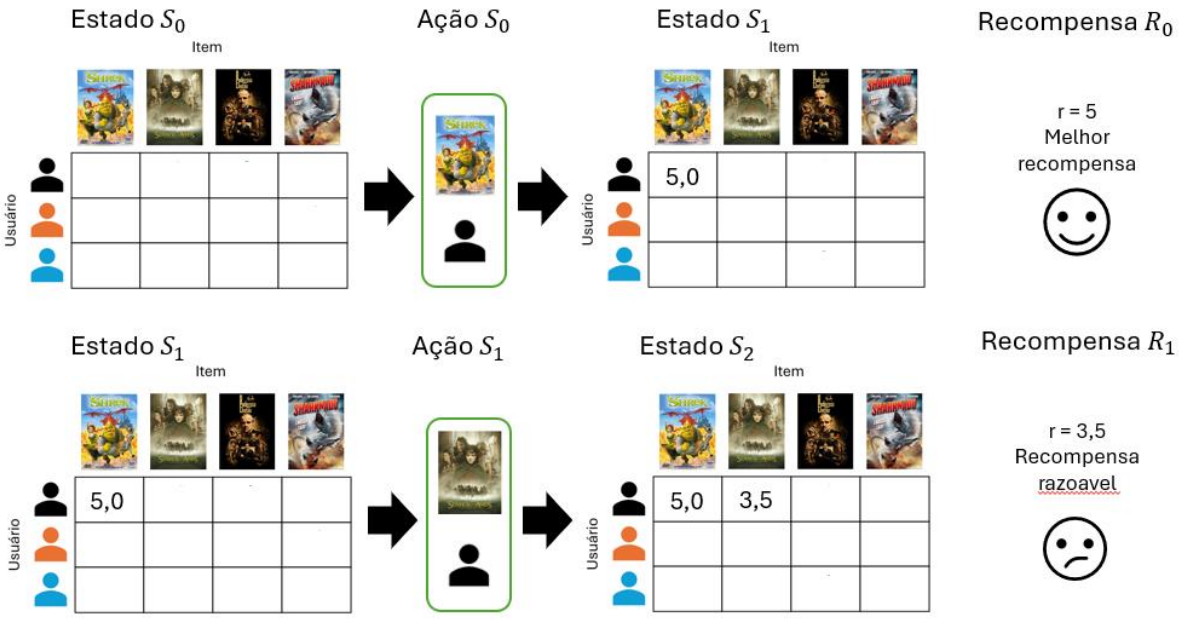


*Figura 8: Exemplo da aplicação de Contextual Bandits em Sistemas de Recomendação*

### 3.2.6. Desafios

O universo de aprendizado por reforço abarca uma infinidade de dificuldades, dentre os principais desafios está na melhor forma de avaliação dessas técnicas, que podem

ser quebrados em mais desafios, como por exemplo, o equilíbrio entre recompensas imediatas ou de longo prazo, por exemplo, recomendar apenas filmes com notas altas de avaliação retornam recompensas imediatas muito boas, dado que provavelmente a maioria das pessoas também iriam gostar do filme, porém essas informações agregam pouco conhecimento para o modelo, que apenas reforçou que aquele filme que tem uma nota boa, realmente tem uma nota boa, e não conseguiu expandir o conhecimento para nenhum filme novo e nem agregou conhecimento sobre outros gostos dos usuários, por outro lado, recomendar apenas filmes que possuem poucas ou nenhuma avaliação, ou recomendar estilos de filmes que um usuário nunca viu agrega muito conhecimento sobre os filmes e sobre os usuários para o sistema, aumentando o conhecimento para diferentes recomendações futuras, porém a recompensa a curto prazo tende a ser menor, dado que a chance de recomendar filmes que não agradem tanto os usuários é bem maior.

# 4. Experimentação

## 4.1. Sobre a base de dados

Foram utilizados dois conjuntos de dados para formar a base de dados final utilizado, a base de dados de avaliações do MovieLens, contendo 25 milhões de avaliações [16] e a base de dados de características sobre os filmes avaliados no IMDb [17] sendo o primeiro utilizado para criação da matriz de avaliações e o segundo como contexto para utilização dos modelos apresentados nesse trabalho.

O formato de recomendação foi considerando as observações como sendo os filmes, portanto, as variáveis de contexto utilizadas são sobre eles, já as colunas são referentes aos avaliadores, os valores da tabela são as respectivas notas dadas pelos avaliadores aos filmes e os valores faltantes são de avaliações não feitas (os valores, quando preenchidos, estão entre 0,5 e 5,0), como está na tabela abaixo.

| movieId | 187 | 548 | 626 | 757 | 803 | 847 | 997 | 1203 | 1228 | 1401 | ... | 161383 | 161544 | 161586 | 161826 | 161928 | 162047 | 162271 | 162495 | 162508 | 162516 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.5 | 4.5 | 4.5 | 3.0 | 5.0 | 4.0 | 4.5 | 3.0 | 5.0 | 4.5 | ... | 4.0 | 2.5 | 3.0 | 3.0 | 3.0 | 3.0 | 4.0 | 3.0 | 4.5 | 4.5 |
| 2 | 3.5 | 4.0 | 4.0 | 3.0 | NaN | NaN | 3.5 | 3.5 | NaN | 3.0 | ... | 4.0 | 1.5 | 1.5 | 1.0 | 4.0 | 2.5 | 1.0 | 3.0 | NaN | 2.5 |
| 3 | 3.0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | 4.0 | NaN | NaN | NaN | NaN | 1.5 | NaN | 3.5 | NaN | 0.5 |
| 5 | NaN | NaN | NaN | NaN | NaN | NaN | 3.0 | NaN | NaN | NaN | ... | 3.0 | NaN | NaN | NaN | NaN | NaN | NaN | 2.0 | NaN | NaN |
| 6 | NaN | 4.0 | NaN | NaN | NaN | 4.5 | 4.5 | 1.0 | 5.0 | NaN | ... | NaN | 3.5 | 2.0 | 4.0 | 4.0 | 4.5 | 3.5 | NaN | 3.0 | 4.5 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 122886 | 5.0 | NaN | NaN | NaN | NaN | 3.0 | 4.5 | NaN | 3.0 | 2.0 | ... | NaN | NaN | NaN | NaN | NaN | NaN | 2.0 | NaN | NaN | 4.0 |
| 122904 | NaN | NaN | NaN | NaN | NaN | NaN | 5.0 | NaN | 4.0 | 3.0 | ... | NaN | NaN | NaN | NaN | 4.0 | NaN | 3.0 | NaN | NaN | 4.0 |
| 134130 | 4.5 | NaN | NaN | NaN | NaN | 3.0 | 4.0 | NaN | 4.5 | 4.0 | ... | NaN | 3.5 | NaN | NaN | NaN | 4.0 | 3.0 | NaN | NaN | 4.5 |
| 134853 | NaN | NaN | NaN | NaN | NaN | 4.0 | 4.0 | NaN | NaN | 3.0 | ... | NaN | NaN | NaN | NaN | 3.0 | 3.0 | 3.0 | NaN | NaN | 4.0 |
| 164179 | NaN | NaN | NaN | NaN | NaN | 4.5 | 5.0 | NaN | 5.0 | 3.0 | ... | NaN | 4.0 | NaN | NaN | NaN | 3.5 | 3.5 | NaN | NaN | 4.5 |

591 rows × 1624 columns

*Tabela 1: Matriz de avaliações de filmes*

A base de dados com as variáveis explicativas era composta pelas variáveis de “isAdult”, indicando se a classificação etária era para maiores de idade, “startYear”, indicando o ano de lançamento do filme, “runtimeMinutes”, representando a quantidade de tempo do filme em minutos, e as outras variáveis eram binárias e representam se o filme estava dentro de um certo gênero, como por exemplo, “Action” para filmes de ação, ou “Adventure” para filmes de aventura. Um exemplo pode ser visto na tabela abaixo.

| movieId | isAdult | startYear | runtimeMinutes | Action | Adventure | Animation | Children | Comedy | Crime | Documentary | ... | Film-Noir | Horror | IMAX | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1995 | 81 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1995 | 104 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1995 | 101 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1995 | 106 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 1995 | 170 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 122886 | 0 | 2015 | 138 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 122904 | 0 | 2016 | 108 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 134130 | 0 | 2015 | 144 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 134853 | 0 | 2015 | 95 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 164179 | 0 | 2016 | 116 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

591 rows × 22 columns

*Tabela 2: Base de dados de Contexto*

A base de dados possui 162.541 avaliadores e 59.047 filmes avaliados, porém, a maioria dos avaliadores não avaliou a maioria dos filmes, por isso, foi realizado alguns filtros para escolha da quantidade de avaliadores e filmes, de forma a garantir uma quantidade suficiente de avaliadores e filmes para o treinamento e uma completude de matriz (quanto da matriz de filmes está preenchida com alguma avaliação, percentualmente) alta o suficiente para realização do experimento. Para construção dessa tabela foi ordenado os avaliadores e filmes por quantidade de avaliações e realizado filtros para escolha do melhor público. Essas análises e filtros de teste se encontram na Tabela 3:

| Quantidade de avaliadores | Quantidade de filmes | Quantidade de avaliações realizadas (mil) | Completude da Matriz |
|---|---|---|---|
| 162.541 (100%) | 59.047 (100%) | 25.000 (100%) | 0,2% |
| 8.122 (5%) | 2953 (5%) | 5.872 (23%) | 24,5% |
| 1.624 (1%) | 591 (1%) | 639 (2,5%) | 66,6% |
| 326 (0,2%) | 119 (0,2%) | 34 (0,1%) | 90,0% |

*Tabela 3: Análise de Completude da Matriz de Avaliações*

A coluna de *Quantidade de Avaliadores* representa a quantidade de diferentes avaliadores presentes na matriz das avaliações, o número entre parênteses representa a porcentagem dos avaliadores sobre o total. A coluna *Quantidade de filmes*, é análoga à coluna anterior, porém, sobre os filmes a serem avaliados. A coluna *Quantidade de avaliações realizadas (mil)*, indica a quantidade de filmes com alguma nota na matriz de avaliações, a ordem de grandeza é da casa do milhar (x 1.000), o número entre parênteses representa a porcentagem de filmes avaliados

sobre o total. E por fim, a coluna *Completude da Matriz*, indica, percentualmente, quanto da matriz está preenchida com alguma avaliação.

Portanto, a base completa, sem nenhum filtro, representada pela primeira linha da Tabela 3, não foi considerada adequada para uso pois, apesar de possuir 162 mil avaliadores (que representam 100% de todos avaliadores disponíveis), 59 mil filmes (que representam 100% de todos os filmes disponíveis) e 25 milhões de avaliações (que representa100% de todas as avaliações realizadas) essa base completa possui apenas 0,2% de toda matriz preenchida com algum valor, o que dificulta a realização das simulações, porém, foram feitos mais filtros, gerando outras 3 bases a fim de encontrar a base com quase toda a matriz preenchida. Nas duas bases da segunda e terceira linha da Tabela 3, foi reduzido consideravelmente a quantidade de avaliadores e filmes para 5% e 1% do total, respectivamente, porém, ambas as bases ainda possuíam menos de 2/3 da matriz de avaliações preenchidas (24,5% e 66,6%, respectivamente). Por fim, foi escolhido um filtro para utilização de apenas 0,2% dos avaliadores que mais avaliaram filmes e dos 0,2% de filmes com mais avaliações, essa escolha se baseou na última linha da Tabela 3. Com essa escolha, foi garantido uma completude de 90% das avaliações (aproximadamente 34 mil avaliações), com 326 avaliadores e 119 filmes.

## 4.2. Abordagem

O modelo utilizado foi apenas a Regressão Linear sem regularização, na sua forma padrão. A utilização de apenas um modelo simples se deu pelo fato do objetivo do trabalho ser da aplicação e comparação das técnicas de aprendizado por reforço em um sistema de recomendação com algumas abordagens populares e bem simples. Além disso, a escolha de um modelo simples colabora para um maior foco nas políticas e parâmetros de controle de exploração e exploitação, pois conforme aumenta a complexidade do modelo reduz as interpretações e aumenta também a quantidade de hiper parâmetros que precisam de otimização para um bom funcionamento.

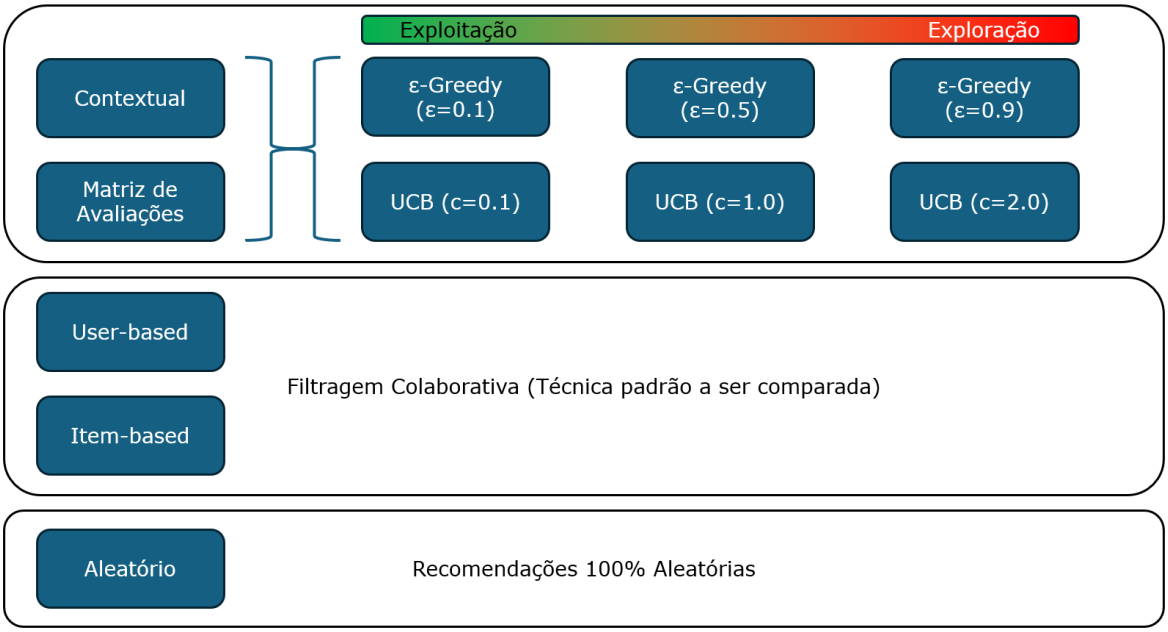


*Figura 9: Abordagens utilizadas*

Foram utilizadas duas diferentes políticas do universo de *Contextual Bandits*, sendo elas, ε-Greedy e UCB (*Upper Confidence Bound*), ambas foram comparadas na utilização da matriz de avaliações como entrada e na utilização do contexto (conteúdo dos filmes).

Foram realizadas 15 abordagens diferentes, 1 abordagem totalmente aleatória, a fim de avaliar um sistema totalmente sem inteligência para recomendação, 2 abordagens utilizando filtragem colaborativa, que são técnicas que podem ser encontradas em sistemas de recomendação mais simples no mercado, e 12 diferentes abordagens utilizando *Contextual Bandits*

A abordagem de *Contextual Bandits* consistiu em verificar as duas políticas, além de variar as variáveis controladoras do trade-off de exploração e exploitação, para as duas diferentes bases de dados, uma base com a matriz de avaliações e outra com as informações de contexto dos filmes, por fim, todas as abordagens utilizaram Regressão Linear, o que acabou resultando em 12 combinações diferentes.

Além das combinações utilizando técnicas de *Contextual Bandits*, também foi verificado o desempenho de duas abordagens mais simples para recomendação, sendo elas uma de filtragem colaborativa, uma baseada em usuário e outra baseada em item, a fim de comparar com as técnicas propostas nesse trabalho.

Por fim, também foi avaliado um sistema de recomendação aleatório, onde seriam recomendados filmes aleatórios à usuários aleatórios, a fim de medir como seria um desempenho em um cenário sem nenhum tipo de inteligência na recomendação.

## 4.3. Métricas

Para o cálculo das métricas foi comparado o valor real da avaliação com o valor de máxima avaliação, isso é, a nota 5, dado que o objetivo é recomendar filmes que os usuários iriam gostar. Porém, como o modelo possui ignorância sobre os valores reais, foi construída uma matriz de avaliações iniciais, sem conhecimento prévio, essa matriz possui o mesmo tamanho da matriz original de avaliações, porém é completamente preenchida pela nota 2,75, sendo essa a nota mediana entre os valores máximos e mínimos possíveis para avaliação (mínimo de 0,5 e máximo de 5,0), e sua escolha se deu pela premissa de total desconhecimento inicial sobre os gostos dos usuários, e essa tabela pode ser visualizada na Figura 10

```
array([[2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75],
       [2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75],
       [2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75],
       ...,
       [2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75],
       [2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75],
       [2.75, 2.75, 2.75, ..., 2.75, 2.75, 2.75]])
```

*Figura 10: Print da tabela utilizada como avaliações iniciais*

Para medir quantitativamente as diferentes abordagens e combinações foram utilizadas duas métricas, uma considerando a matriz completa e outra apenas o item recomendado.

A métrica da matriz completa, também conhecida como *Soma do Valor Absoluto dos Resíduos*, nesse projeto é chamada de $Metrica_{Full}$, foi utilizada para avaliar o quanto a matriz de avaliações estava distante da matriz real de avaliações, sendo que quanto menor o seu valor, mais próximo estava dos valores reais, e é calculada de acordo com a Equação abaixo:

$$Metrica_{Full} = \sum_{i=0}^{m} abs(y_{true} - y_{pred})$$

Onde $y_{true}$ são os valores reais das avaliações dadas pelos avaliadores aos filmes, e o $y_{pred}$ são os valores das avaliações atualizados após cada rodada de treinamento da respectiva abordagem.

Um grande problema dessa métrica de avaliação da matriz completa é que, conforme a quantidade de rodadas se aproxima da quantidade de avaliações disponíveis, o valor da métrica tende a ser o mesmo para todas as abordagens, dado que, após cada avaliação, a nota correta dada pelo usuário substitui a matriz de conhecimento do modelo.

A segunda métrica utilizada, é a métrica de avaliação individual, que é uma métrica de *Erro Quadrado*, nesse trabalho é chamada de $Metrica_{Item}$, sua utilização se deu para avaliar o quão bem a abordagem foi em cada rodada, sendo que quanto menor o valor, mais próximo do valor real, e é calculada da seguinte forma:

$$Metrica_{Item} = (y_{true} - y_{pred})^2$$

Essa métrica adquiriu uma importância maior na avaliação das diferentes abordagens, pois com ela foi possível medir se o desempenho das diferentes abordagens estava piorando, melhorando ou se mantendo constante conforme avançava a quantidade de rodadas.

## 4.4. Experimentos

Esse trabalho se propôs a avaliar o desempenho de diferentes abordagens para um sistema de recomendações utilizando métricas de longo e curto prazo (*MetricaFull* e *MetricaItem*), algumas abordagens utilizando técnicas de *Contextual Bandits* (Políticas *ε-Greedy* e *UCB*), algumas usando técnicas de sistemas de recomendação tradicionais (Filtragem colaborativa baseada em item e usuário) e uma abordagem totalmente aleatória.

Para cada abordagem foram realizadas 10 simulações, onde cada simulação avançou até 10 mil rodadas (10 mil recomendações feitas), a partir da média das simulações foi calculada ambas as métricas para cada diferente abordagem, a fim de evitar quaisquer vieses para alguma inicialização diferente.

Apesar das diferenças de cada abordagem, as etapas seguidas nas simulações foram as mesmas, como é possível ver, de forma simplificada, na Figura 11:

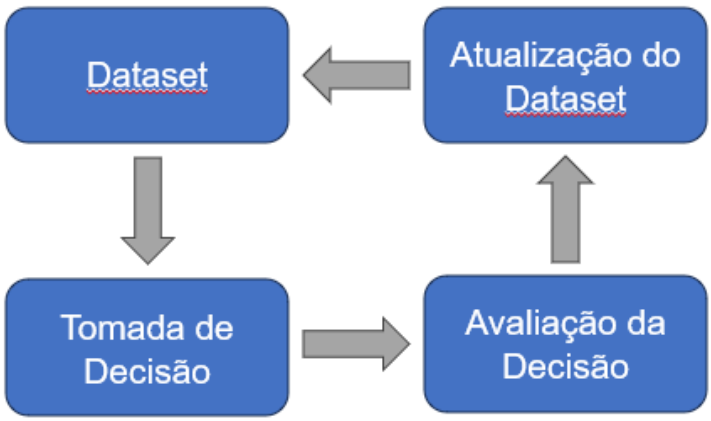


*Figura 11: Etapas realizadas em cada abordagem*

Portanto, a construção de cada simulação se deu da mesma forma, primeiro era avaliado o "Dataset", sendo este o *estado* atual da matriz de avaliações, no respectivo *estado* da simulação, estava preenchido com as avaliações reais apenas o conjunto item-usuário que já havia sido recomendado, para todos os outros conjuntos, a nota considerada era o valor intermediário entre a nota máxima e a nota mínima. Além da matriz de avaliações, para as abordagens baseadas em *Contextual Bandits* também foi utilizada nessa etapa a base de contexto contendo informações sobre os filmes.

Na segunda etapa, de "Tomada de Decisão", ocorria de fato, as diferentes abordagens entravam em ação para definir qual o melhor filme a ser recomendado a fim de maximizar as avaliações. Para ambas as abordagens foi calculada as estimativas de avaliações para cada item-usuário, ou seja, para cada usuário foi estimado a nota que aquele usuário daria para cada filme recomendado, em seguida foi feito um ranking ordenando quais os melhores filmes a serem recomendados a cada usuário (os que possuíam a maior estimativa de nota), na sequência. Como foram utilizadas diferentes abordagens, para àquelas baseadas em *Contextual Bandits*, foi utilizada a base de contexto para estimar as avaliações, para as abordagens baseadas na Matriz de Avaliações, foi estimada baseada na própria matriz com as avaliações já realizadas e as médias (desconhecidas) e para as abordagens baseadas em Filtragem Colaborativa foi feito da mesma forma, ordenando as melhores estimativas, já na abordagem aleatória, não foi feita nenhuma ordenação e o filme recomendado foi aleatório à um usuário aleatório.

Na etapa seguinte “Avaliação da Decisão” foi avaliado se a recomendação foi uma boa recomendação, para isso foi calculada as métricas propostas, ou seja, caso o filme recomendado tenha obtido uma nota 5 (máxima) foi obtida a melhor métrica e para um filme recomendado com nota 0,5 (mínima) foi obtida a pior métrica possível.

Por fim, na “Atualização do Dataset” foi atualizada a Matriz de Avaliações com a nota real dada pelo usuário ao filme recomendado, assim a nova matriz passa a possuir mais uma avaliação atualizada e pode ter o ciclo reiniciado, voltando a etapa do “Dataset”.

Todo esse processo foi repetido em 10 simulações, onde cada simulação iniciou com a matriz totalmente ignorante (sem nenhuma avaliação real feita) e progrediu até atingir 10.000 (dez mil) avaliações feitas.

# 5. Resultados e Discussão

## 5.1. Resultados das combinações na $Metrica_{Full}$

| Combinação | | | *MetricaFull* (Rodadas) | | | |
|---|---|---|---|---|---|---|
| **Base de Treino** | **Política** | **Parâmetro** | **1** | **100** | **1.000** | **10.000** |
| Contextual | e-Greedy | 0,5 | 43.278 | 43.147 | 41.926 | 29.816 |
| Contextual | e-Greedy | 0,1 | 43.277 | 43.149 | 41.876 | 29.837 |
| Matriz de avaliações | e-Greedy | 0,5 | 43.277 | 43.148 | 41.930 | 30.010 |
| Matriz de avaliações | e-Greedy | 0,1 | 43.277 | 43.155 | 41.968 | 30.029 |
| Contextual | e-Greedy | 0,9 | 43.277 | 43.154 | 42.030 | 30.755 |
| | Filtragem Colaborativa Item-Based | | 43.277 | 43.147 | 42.048 | 30.774 |
| Matriz de avaliações | e-Greedy | 0,9 | 43.278 | 43.154 | 42.050 | 30.851 |
| Contextual | UCB | 0,1 | 43.279 | 43.147 | 42.087 | 31.101 |
| Contextual | UCB | 2 | 43.278 | 43.150 | 42.073 | 31.102 |
| Contextual | UCB | 1 | 43.278 | 43.158 | 42.046 | 31.123 |
| | Aleatório | | 43.278 | 43.157 | 42.076 | 31.173 |
| | Filtragem Colaborativa User-Based | | 43.277 | 43.143 | 42.095 | 31.441 |
| Matriz de avaliações | UCB | 2 | 43.277 | 43.167 | 42.178 | 33.395 |
| Matriz de avaliações | UCB | 0,1 | 43.278 | 43.178 | 42.413 | 33.649 |
| Matriz de avaliações | UCB | 1 | 43.279 | 43.180 | 42.767 | 40.789 |

*Tabela 4: Resultados do Experimento pela MétricaFull*

Os resultados dos experimentos são exibidos na Tabela 4, onde estão ordenados em ordem decrescente da melhor combinação para a pior, de acordo com o resultado

após 10 mil rodadas, apesar disso, também estão na tabela a $Metrica_{Full}$ para 1 rodada, 100 rodadas e 1 mil rodadas. As cores representam o quão bom foi a métrica, sendo que quanto mais verde a cor, melhor foi a métrica, quanto mais vermelho, pior foi a métrica.

A abordagem que utiliza escolhas aleatórias serviu como uma forma de comparar as outras abordagens ao uso de recomendações sem nenhum tipo de inteligência. Além da abordagem aleatória, há também as duas abordagens mais simples que utilizam filtragem colaborativa, tanto baseada em usuário quanto baseada em itens (filmes) que foram consideradas como *Benchmark*, ou seja, uma base para tomar como comparação para decisões totalmente arbitrárias, então podemos considerar que qualquer métrica pior que o aleatório teve seu desempenho inferior à uma recomendação sem nenhum tipo de inteligência, porém, vale ressaltar, que devido as propriedades de aprendizado e recompensa do aprendizado por reforço, uma métrica ruim pode ser uma questão de prioridade da política em exploração ao invés de exploitação. Deve-se levar em consideração também, que, apesar de aleatória, como a quantidade de avaliações restantes diminui, consequentemente a métrica diminui, isso se deve a própria forma de como a métrica foi construída.

Considerando uma comparação entre as políticas de aprendizado por reforço, a política *ε-Greedy* se saiu melhor em todas as combinações de bases de treino e parâmetros do que a política *UCB*, em relação as técnicas de filtragem colaborativa, a política *ε-Greedy* também se saiu melhor, por outro lado, a política *UCB* não se saiu muito bem, tendo seu desempenho igual ou inferior as técnicas de filtragem colaborativa.

Dentro da política *ε-Greedy*, as combinações com a base Contextual acabaram performando melhor do que as combinações que utilizaram a base de Matriz de Avaliações para seu treinamento.

Na questão do parâmetro, aqueles que consideravam uma exploração e exploitação equilibrada acabava se saindo melhor quando olhávamos as mesmas políticas e bases de treinos, já nas extremidades os parâmetros que utilizavam mais da exploitação em relação à exploração acabaram tendo uma métrica melhor.

A melhor combinação nesse experimento, para essa métrica, levando em consideração a base de contexto disponível e a base de avaliações com seu

respectivo filtro, foi a política ε-Greedy, utilizando a base de treino de Contexto, com os parâmetros de 0,1 e 0,5, que ficaram ligeiramente diferentes, mas muito próximos.

## 5.2. Resultados das combinações na $Metrica_{Item}$

Os resultados obtidos através da $Metrica_{Item}$ foram considerados em médias móveis de 1 mil rodadas, isso devido a sua alta variância entre rodadas, o que dificultava a visualização e análise, portanto, essa média móvel foi feita de forma a minimizar essas variações e gerar números mais estáveis. As duas figuras abaixo apresentam os resultados, tanto para as bases de Contexto quanto para as bases de Matriz de avaliações, ambas as figuras contam com as abordagens de comparação, tanto a recomendação aleatória quanto a baseada na filtragem colaborativa, além disso, as cores se repetem para ambos os gráficos abaixo, sendo que cada cor é uma técnica diferente, sendo que a diferença entre os dois gráficos está no fato de que a base de treino do primeiro gráfico (Figura 12) é a base contextual e do segundo gráfico (Figura 13) é a base de avaliações, além disso, a legenda com as cores e sua respectiva técnica está abaixo dos dois gráficos (Figura 14).

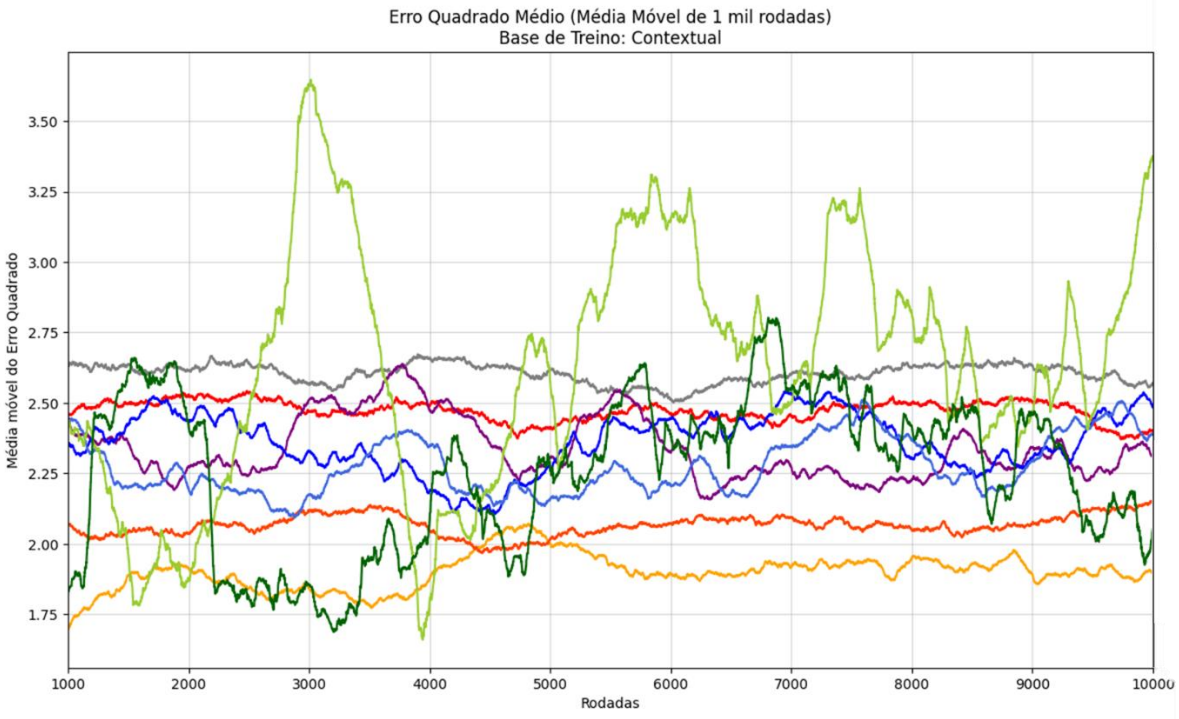


*Figura 12: Gráfico do Erro Quadrado Médio para as bases de Contexto em uma média móvel de mil rodadas.*

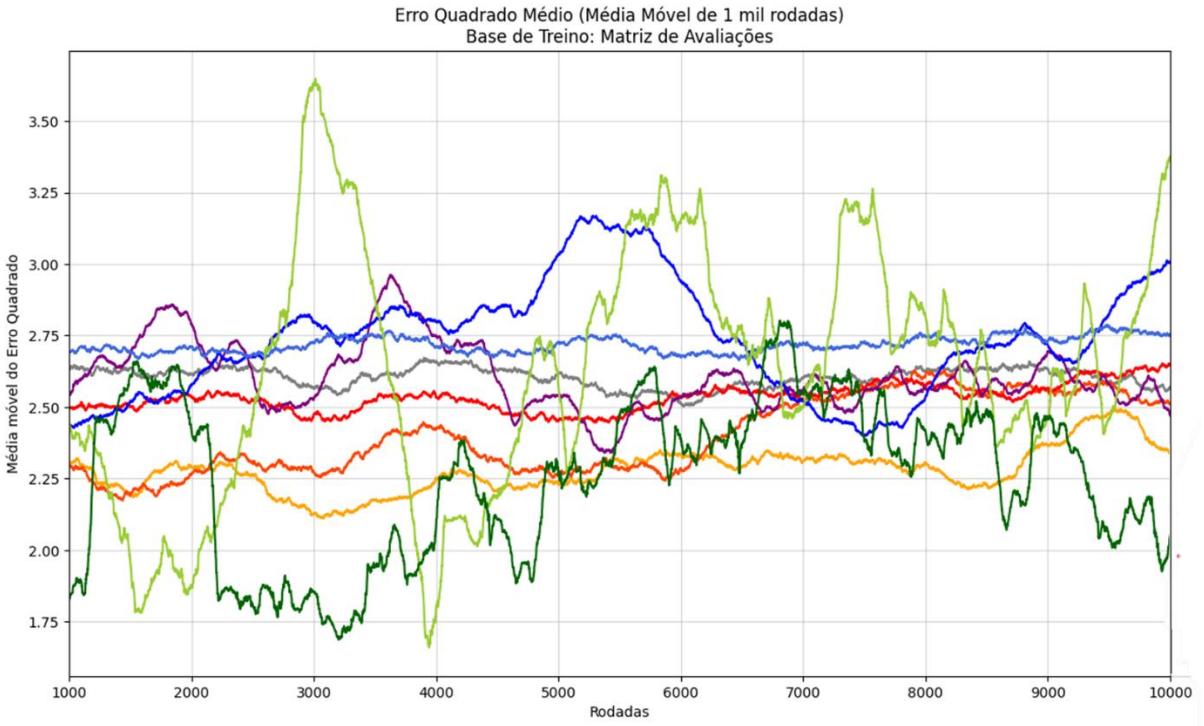


*Figura 13: Gráfico do Erro Quadrado Médio para as bases de Matriz de avaliações em uma média móvel de mil rodadas.*

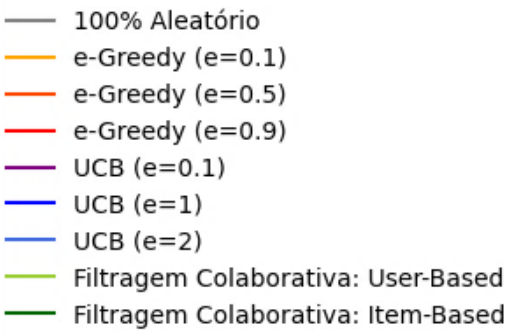


*Figura 14: Políticas utilizadas nos gráficos de média móvel de MetricaItem*

Como é possível visualizar nas figuras acima, o desempenho da maioria das técnicas foi razoavelmente estável, com exceção das abordagens que utilizaram filtragem colaborativa, que variaram bastante durante o treinamento, tem seus picos e vales com diferenças consideráveis. As abordagens que utilizaram UCB também foram mais instáveis e com algumas variações mais consideráveis que as técnicas que utilizaram ε-Greedy. Já a abordagem de recomendação aleatória, apesar de não ter ido bem, manteve uma estabilidade durante todo o processo.

A Tabela 5, está resumindo as informações dos gráficos acima de forma a facilitar a interpretação e comparação dos resultados, essa tabela possui um tratamento nos dados e foi dividida em três grupos de média móvel da $Metrica_{Item}$, que são, média

móvel da rodada inicial até a rodada 1 mil, da rodada 4 mil até 5 mil e por fim, da rodada 9 mil até 10 mil, sendo essa a última média móvel de mil rodadas disponível, a ordenação da tabela está baseada no desempenho na última média móvel, ou seja, de 9 mil a 10 mil rodadas. Nessas métricas, quanto mais verde, melhor foi a métrica, quanto mais vermelho, pior foi o desempenho.

| **Combinação** | | | ***Metricaltem* (Média móvel - 1 mil rodadas)** | | |
|---|---|---|---|---|---|
| **Base de Treino** | **Política** | **Parâmetro** | **até 1 mil** | **4 mil a 5 mil** | **9 mil a 10 mil** |
| Contextual | e-Greedy | 0,1 | 1,7 | 2,0 | 1,9 |
| | Filtragem Colaborativa Item-Based | | 1,8 | 2,3 | 2,1 |
| Contextual | e-Greedy | 0,5 | 2,1 | 2,0 | 2,1 |
| Matriz de avaliações | e-Greedy | 0,1 | 2,3 | 2,2 | 2,3 |
| Contextual | UCB | 0,1 | 2,4 | 2,3 | 2,3 |
| Contextual | e-Greedy | 0,9 | 2,5 | 2,4 | 2,4 |
| Contextual | UCB | 2 | 2,4 | 2,2 | 2,4 |
| Matriz de avaliações | e-Greedy | 0,5 | 2,3 | 2,3 | 2,5 |
| Contextual | UCB | 1 | 2,4 | 2,3 | 2,5 |
| Matriz de avaliações | UCB | 0,1 | 2,5 | 2,5 | 2,5 |
| Matriz de avaliações | e-Greedy | 0,9 | 2,5 | 2,5 | 2,6 |
| | Aleatório | | 2,6 | 2,6 | 2,6 |
| Matriz de avaliações | UCB | 2 | 2,7 | 2,7 | 2,8 |
| Matriz de avaliações | UCB | 1 | 2,4 | 3,0 | 3,2 |
| | Filtragem Colaborativa User-Based | | 2,4 | 2,6 | 3,4 |

*Tabela 5: Tabela de Resultados baseado na Metrica_Item*

Como é possível observar, o desempenho das combinações na $Metrica_{Item}$ refletem muito o desempenho dessas mesmas combinações a longo prazo, dado que elas são uma forma de medição de um certo ‘erro instantâneo’ do modelo.

Dentro das melhores combinações estão as que utilizaram a base de Contexto para treinar, além de fatores de exploração baixos, ou seja, são modelos que priorizam o conhecimento já adquirido em detrimento de testar novas possibilidades em possíveis recomendações que poderiam dar um retorno maior, além disso a abordagem via filtragem colaborativa baseada em item obteve um excelente desempenho nas últimas etapas de treinamento.

O erro na abordagem de recomendação aleatória obteve a mesma faixa de erro (2,6) do começo ao fim do experimento, o que mostra coerência nessa abordagem, dado que não houve qualquer tipo de aprendizado, o que torna o desempenho das 3 abordagens que obtivem desempenho inferior bem curiosas, que são as combinações que utilizaram a matriz de avaliações para treinar e filtragem colaborativa baseada no usuário, umas das possíveis explicações para o desempenho das combinações de aprendizado por reforço é pelo fato de serem as combinações com o maior fator de exploração, ou seja, estavam mais interessados em descobrir novas possibilidades de recomendações boas do que se ‘garantirem’ naquelas que tinham uma maior confiança de um bom resultado.

## 5.3. Discussão

### 5.3.1. Sobre aprendizado por reforço

Na comparação das duas políticas utilizadas, a ε-Greedy teve um desempenho bem superior à UCB, para quaisquer outras combinações feitas de base de treino e controle de exploração e exploitação.

Analisando apenas a utilização das bases de treinamento contextual e da matriz de avaliações, para quaisquer combinações feitas, aquela que utilizava a base contextual obteve um desempenho superior àquela que utilizou a base com a matriz de avaliações, mostrando assim, um maior ganho de desempenho ao utilizar o contexto para um sistema de recomendações em relação a utilização de apenas a base com as avaliações dos usuários.

Em relação ao parâmetro de controle de exploração e exploitação, aqueles que obtiveram melhor desempenho foram os que priorizaram a exploitação em relação à exploração, isso é, priorizaram o ganho de recompensas a curto prazo, recompensas que havia mais garantia de serem ganhas, porém, caso as simulações fossem mais adiante, ou houvesse um conjunto de dados maiores, a longo prazo, provavelmente as técnicas que favoreceram uma exploitação poderiam adquirir recompensas mais valiosas.

### 5.3.2. Comparativo com técnicas já conhecidas

Nas técnicas de sistemas de recomendação mais conhecidos, como de filtragem colaborativa, a técnica baseada em item teve um desempenho bem superior à baseada em usuário, sendo que, a baseada em item obteve métricas boas o suficiente para estar entre as melhores técnicas, já a baseada em usuário, acabou ficando entre as piores abordagens.

Essa diferença considerável, provavelmente se deve ao fato de que, ao reduzir consideravelmente a base de dados, foram mantidos os filmes com mais avaliações, provavelmente filmes parecidos que agradam pessoas que gostam do mesmo estilo, ou seja, na técnica baseada em item, ao comparar filmes semelhantes e recomendar para aquele usuário que gostou de um, mas ainda não assistiu outro parecido, teve um desempenho muito melhor do que comparar usuários que gostaram dos mesmos filmes e sugerir filmes de um usuário para outro, como a técnica baseada em usuário propõe.

### 5.3.3. Desempenho Geral

Considerando apenas as duas métricas e ignorando os nuances e desafios da avaliação de técnicas de aprendizado por reforço, tanto no fator de longo prazo (*MétricaFull*), quanto de curto prazo (*MétricaItem*), as abordagens que mais se destacaram foram as abordagens que utilizaram técnicas de aprendizado por reforço, em especial, a política ε-Greedy, utilizando a base de contexto e com fatores de exploitação igual ou maior que a exploração, além disso, uma abordagem já conhecida e utilizada no contexto de sistemas de recomendação também teve um certo destaque, que foi a filtragem colaborativa baseada em item, que obteve métricas muito boas também.

As abordagens que tiveram pior desempenho foram aquelas que utilizaram a política UCB e a matriz de avaliações como base de treino, independente do fator de exploração e exploitação, além da filtragem colaborativa baseada em usuário, já que todas essas técnicas obtiveram um desempenho igual ou inferior a uma recomendação aleatória, ou seja, para esse conjunto de dados, partindo de um conhecimento nulo até 10 mil recomendações, fazer recomendações aleatórias garantiu uma maior chance de acertar um filme bem avaliado do que essas técnicas mencionadas.

Dentre as três abordagens que obtiveram um melhor desempenho nas duas métricas, a combinação de política ε-Greedy, com a base de contexto e o fator de exploração igual a 0,1 foi a que mais se destacou e merece um destaque pelo ótimo equilíbrio entre recompensas de curto e longo prazo.

### 5.3.4. Desafios

Há um grande desafio na avaliação de técnicas que utilizam aprendizado por reforço, exatamente pelo fato dessas técnicas, dependendo da configuração do ambiente, da forma de recompensas e da própria arquitetura da técnica priorizarem o aprendizado à longo prazo em detrimento de um melhor desempenho no curto prazo, um grande exemplo disso são as abordagens que utilizaram fatores de exploração alto, ou seja, priorizavam adquirir um conhecimento de longo prazo sobre o ambiente e sobre as possíveis ações em detrimento de recomendar filmes que possuíam uma maior certeza de que acertariam na recomendação.

# 6. Conclusão

O estudo realizado neste trabalho abordou a aplicação de técnicas de aprendizado por reforço, especificamente *Contextual Bandits*, em sistemas de recomendação, utilizando conjuntos de dados provenientes do MovieLens e do IMDb. A análise foi conduzida com o objetivo de comparar o desempenho dessas técnicas com abordagens tradicionais de filtragem colaborativa e recomendação aleatória. Além disso, foram realizados tratamentos nas bases de treinamento

A escolha da Regressão Linear como modelo base para os experimentos permitiu uma análise focada nas políticas e parâmetros de controle de exploração e exploitação, sem a complexidade adicional de modelos mais sofisticados. As políticas ε-Greedy e UCB foram implementadas e testadas em diferentes cenários, utilizando tanto a matriz de avaliações quanto informações contextuais dos filmes.

Os resultados, considerando os 0,2% dos avaliadores com mais avaliações e os 0,2% dos filmes mais avaliados do conjunto de dados de 25 milhões do MovieLens, que resultaram em uma base de aproximadamente 38 mil avaliações (das quais, 90% estavam preenchidas com alguma nota), indicaram que, em geral, as abordagens baseadas em *Contextual Bandits*, especialmente aquelas usando a política ε-Greedy, superaram as técnicas de filtragem colaborativa e recomendação aleatória. Notavelmente, a utilização do contexto dos filmes como base de treinamento mostrou-se mais eficaz do que a simples matriz de avaliações. Além disso, abordagens que favoreciam a exploração moderada em relação à exploração excessiva tendiam a ter um desempenho superior.

No entanto, alguns desafios foram identificados, especialmente relacionados à avaliação de técnicas de aprendizado por reforço. A natureza dessas técnicas pode priorizar o aprendizado a longo prazo em detrimento de ganhos imediatos, levando a resultados menos favoráveis em cenários de curto prazo. Isso foi evidenciado pela baixa performance de certas combinações que enfatizavam a exploração em detrimento da exploração.

Em resumo, os resultados deste estudo destacam a eficácia das técnicas de aprendizado por reforço, especialmente *Contextual Bandits* com a política ε-Greedy, na construção de sistemas de recomendação. No entanto, é fundamental considerar cuidadosamente os desafios específicos dessas técnicas, especialmente em relação

ao equilíbrio entre exploração e exploração e à avaliação de desempenho a curto e longo prazo.

## 6.1. Trabalhos futuros

Esse trabalho apresenta algumas aplicações de técnicas de aprendizado por reforço em sistemas de recomendação, porém, ainda há muito que possa evoluir para melhores resultados, dentre algumas dessas possibilidades estão:

- Política Thompson Sampling [18], uma política muito popular que poderia substituir o ε-Greedy ou UCB, porém essa política assume uma distribuição a priori, o que aumentaria o grau de complexidade e combinações possíveis para resolver problemas de recomendação. Há diversas outras políticas que poderiam ser avaliadas também, como Epoch-Greedy [19] ou Lin-UCB [20].
- Outros modelos de aprendizado de máquina, como modelos com maior complexidade, por exemplo *Redes Neurais Artificiais* ou modelos de Boosting, como *Xgboost* ou *LightGBM*. A grande dificuldade desses modelos é sua baixa explicabilidade, maior custo de processamento, além da necessidade de otimizar os hiperparâmetros do modelo.
- Combinação entre a base contextual (que possui informações sobre os filmes) e a base de recomendações, de forma a maximizar o ganho entre as duas bases disponíveis.
- Construção de modelos offline, com menor impacto aos usuários, dado que recomendar itens muito irrelevantes à usuários pode fazer com que troquem de plataforma, ocasionando na perda completado do usuário, uma técnica de teste A/B para testes offline é encontrada nesse artigo [21], além da necessidade de adaptação dos algoritmos de aprendizado por reforço também para ambientes offline, como sugerem esse artigos [22] e esse [23]

Os problemas na implementação dessas possíveis alterações são na evolução da complexidade, tanto para mudança de política, de modelos, e de base, tornando assim uma solução menos interpretável e, possivelmente, mais necessitada de recursos computacionais, tornando sua execução mais lenta e menos interpretável.

# 7. Referências


[1] A. Dereventsov e A. Bibin, “Simulated Contextual Bandits for Personalization Tasks from Recommendation Datasets,” *IEEE ICDMW - International Conference on Data Mining Workshops,* 2022.

[2] L. Tang, Y. Jiang, L. Li e T. Li, “Ensemble Contextual Bandits for Personalized Recommendation,” *Proceedings of the 8th ACM Conference on Recommender systems,* pp. 73-80, 2014.

[3] X. Xu, F. Dong, Y. Li, S. He e X. Li, “Contextual-Bandit Based Personalized Recommendation with Time-Varying User Interests,” *Proceedings of the AAAI Conference on Artificial Intelligence,* 2020.

[4] A. Pilani, K. Mathura, H. Agrawalda, D. Chandolab, V. A. Tikkiwalb e a. A. Kumar, “Contextual Bandit Approach-based Recommendation System for Personalized Web-based Services,” *Applied Artificial Intelligence,* vol. 35, nº 7, pp. 489-504, 2021.

[5] F. Ricci, L. Rokach e B. Shapira, “Recommender Systems Handbook,” 2015.

[6] K. Shah, A. Salunke, S. Dongare e K. Antala, “Recommender Systems: An overview of different approaches to recommendations,” *International Conference on Innovations in information Embedded and Communication Systems,* 2017.

[7] J. Ni, Y. Cai e G. Tang, “Collaborative Filtering Recommendation Algorithm Based on TF-IDF and User Characteristics,” *Applied Sciences,* 14 Outubro 2021.

[8] C. C. Aggarwal, Recommender Systems: The Textbook, Springer, 2016.

[9] G. James, D. Witten, T. Hastie, R. Tibshirani e J. Taylor, “Linear Regression,” em *An Introduction to Statistical Learning*, Springer, 2023, pp. 69-134.

[10] R. S. Sutton e A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 2018.

[11] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen e C. Miao, “A Survey on Reinforcement Learning for Recommender Systems,” *IEEE Transactions on Neural Networks and Learning System,* 11 Junho 2023.

[12] Y. Chi, “Multi-arm bandits: stochastic bandits,” em *Foundations of Reinforcement Learning*, Carnegie Mellon University, 2023.

[13] A. White e M. White, *Reinforcement Learning Specialization.*

[14] “Machine learning series - part 5: Exploration vs. Exploitation dilemma in Reinforcement learning,” Steemit, 2017. [Online]. Available: https://steemit.com/technology/@mor/machine-learning-series-part-5-exploration-vs-exploitation-dilemma-in-reinforcement-learning.

[15] G. Fei, “Contextual Bandit for Marketing Treatment Optimization,” Wayfair | Careers, 7 Outubro 2021. [Online]. Available: https://www.aboutwayfair.com/careers/tech-blog/contextual-bandit-for-marketing-treatment-optimization.

[16] “MovieLens Dataset,” 2019. [Online]. Available: https://grouplens.org/datasets/movielens/25m/.

[17] “IMDb Dataset,” 2023. [Online]. Available: https://developer.imdb.com/non-commercial-datasets/.

[18] D. J. Russo, B. V. Roy, A. Kazerouni, I. Osband e Z. Wen, “A Tutorial on Thompson Sampling,” *Foundations and Trends® in Machine Learning,* pp. 1-96, 2018.

[19] J. Langford e T. Zhang, “The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information,” *NeurIPS Proceedings,* 2007.

[20] L. Li, W. Chu, J. Langford e R. E. Schapire, “A Contextual-Bandit Approach to Personalized News Article Recommendation,” *Presented at the Nineteenth International Conference on World Wide Web,* 2010.

[21] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham e S. Dollé, “Offline A/B testing for Recommender Systems,” *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining,* 2018.

[22] R. Agarwal, D. Schuurmans e M. Norouzi, “An Optimistic Perspective on Offline Reinforcement Learning,” *Proceedings of the 37th International Conference on Machine Learning,* pp. 104-114, 2020.

[23] S. Levine, A. Kumar, G. Tucker e J. Fu, “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems,” *DBLP Journals,* 2005.

[24] D. Bouneffouf e I. Rish, “A Survey on Practical Applications of Multi-Armed and Contextual Bandits,” *CoRR - Computing Research Repository,* 2 Abril 2019.

[25] R. Johari, L. Pekelis, P. Koomen e D. Walsh, “Peeking at A/B Tests,” *KDD 2017 Applied Data Science Paper,* pp. 1517-1525, 2017.

[26] G. Krishnan, “Selecting the best artwork for videos through A/B testing,” 3 Maio 2016. [Online]. Available: https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6.

[27] D. Cortes, “Adapting multi-armed bandits policies to contextual bandits scenarios,” *CoRR - Computing Research Repository,* 2018.

[28] Y. Zhu, D. J. Foster, J. Langford e P. Mineiro, “Contextual Bandits with Large Action Spaces: Made Practical,” *ICML - International Conference on Machine Learning,* 2022.

[29] J. R. Chumley, “Bandits for Recommender System Optimization,” Medium, 2018. [Online]. Available: https://towardsdatascience.com/bandits-for-recommender-system-optimization-1d702662346e.

[30] T. Sauerwald, “Bandit Algorithms,” em *Randomised Algorithms*, 2021-2022.

[31] D. Roy e M. Dutta, “A systematic review and research perspective on recommender systems,” 2022.