# iRec: Um framework para modelos interativos em Sistemas de Recomendação

**Thiago Silva**[1], **Adriano Pereira**[2], **Leonardo Rocha**[1]

[1] Departamento de Ciência da Computação – Universidade Federal de São João del-Rei (UFSJ)

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

thiagosilva@aluno.ufsj.edu.br, lcrocha@ufsj.edu.br, adrianoc@dcc.ufmg.br

***Abstract.*** *Nowadays, most e-commerce and entertainment services have adopted interactive Recommender Systems (RS) to guide the entire journey of users into the system. This task has been addressed as a Multi-Armed Bandit problem where systems must continuously learn and recommend at each iteration. However, despite the recent advances, there is still a lack of consensus on the best practices to evaluate such bandit solutions. Several variables might affect the evaluation process, but most of the works have only been concerned with the accuracy of each method. Thus, this master dissertation proposes an interactive RS framework named iRec. It covers the whole experimentation process by following the main RS guidelines. The iRec provides three modules to prepare the dataset, create new recommendation agents, and simulate the interactive scenario. Moreover, it also contains several state-of-the-art algorithms, a hyperparameter tuning module, distinct evaluation metrics, different ways of visualizing the results, and statistical validation.*

## 1. Introdução

Atualmente, os Sistemas de Recomendação (SsR) RSs não são mais algoritmos offline que fazem previsões em lote de acordo com os requisitos do negócio. Nos sistemas online atuais, eles se tornaram responsáveis por orientar toda a experiência do usuário desde suas primeiras interações como um modelo de decisão sequencial [Wu et al. 2019, Zhou et al. 2020]. Nesse caso, na interação de cada usuário, o sistema deve recomendar um ou mais itens, receber o feedback do usuário e atualizar seu conhecimento para a próxima recomendação [Wu et al. 2018]. A ideia é aprender a cada interação para aumentar o conhecimento do sistema e maximizar a satisfação do usuário no longo prazo. Os trabalhos atuais abordaram esse desafio como um problema do Multi-Armed Bandit (MAB), onde os itens são modelados como *arms* a serem selecionadas, e a experiência do usuário é representada pelo *reward* acumulado [Sanz-Cruzado et al. 2019, Zou et al. 2020, Shams et al. 2021].

Apesar dos avanços recentes, há uma completa falta de consenso sobre as melhores práticas de avaliação de um sistema de recomendação interativo. Esses novos algoritmos são baseados na teoria de Aprendizado por Reforço (*Reinforcement Learning* - RL) e geralmente exigem um ajuste de vários parâmetros que podem afetar diretamente sua eficácia. Em um cenário interativo, devemos definir pelo menos: (1) o número de tentativas a serem realizadas; (2) o número de itens a serem recomendados em cada tentativa; (3) se o usuário pode receber as mesmas recomendações

nas próximas tentativas; (4) a forma como cada usuário será escolhido para receber alguma recomendação; (5) os critérios para atualizar o conhecimento do sistema; (6) a métrica de *reward/regret* a ser otimizada; e muitos outros. Consequentemente, surgiram duas grandes preocupações: avaliação não reproduzível e comparações injustas [Sun et al. 2020]. Após uma detalhada revisão sistemática da literatura sobre MAB no campo de recomendação [Silva et al. 2022a], observamos que a maioria dos trabalhos simplesmente abstraem as noções de *reward* ou *regret* da teoria RL para avaliar seus algoritmos. Nesse sentido, eles estão preocupados apenas com a eficácia da previsão da recomendação - um conceito ultrapassado na literatura de recomendação atual.

Nesta dissertação, propomos um novo framework para avaliar sistemas de recomendação interativos, denominado *iRec*, que tem como objetivo fornecer uma comparação imparcial e justa entre distintos modelos de SsR com diversas metodologias de avaliação amplamente testadas e utilizadas na literatura. O *iRec* consiste em uma estrutura completa composta por três blocos de construção principais e uma plataforma comum que inclui bancos de dados, algoritmos e ambientes de benchmark para o cenário interativo. O primeiro componente é responsável pelo **environment setting** e tem como objetivo realizar o carregamento e o pré-processamento do conjunto de dados de recomendação. O segundo é responsável por implementar o *agent* requerido por um cenário interativo. No contexto desta dissertação, o *iRec* implementa um **recommendation agent** para treinar (se necessário) e então realizar todas as previsões necessárias. O terceiro componente é responsável por criar a **evaluation policy** dos algoritmos usuais de RL para definir como o agente irá interagir com o ambiente da tarefa. Este componente está integrado a diversos conjuntos de dados públicos de cenários distintos e contém múltiplas políticas de avaliação atualmente utilizadas na literatura. O *iRec* fornece:

- compatibilidade com vários conjuntos de dados (i.e., filmes, músicas, livros, etc.);
- uma estrutura modular e reutilizável, que permite a adição de novos conjuntos de dados, modelos, políticas e métricas de avaliação;
- execução paralelizada, *logging*, testes estatísticos e visualização dos resultados;
- abordagens distintas de filtragem e pré-processamento de dados;
- dezessete modelos de recomendação adaptados à tarefa de aprendizagem online;
- um ajuste de hiperparâmetros para permitir a adaptação de SsR complexos;
- métricas de avaliação com objetivos distintos (i.e., *accuracy*, *novelty*, *diversity*, e *coverage*).

Assim, o *iRec* apresenta um novo ambiente de reprodutibilidade para esse cenário de aprendizado on-line difícil na área de recomendação. Uma vez que ele armazena todas as informações sobre a execução de cada pipeline (ou seja, os melhores parâmetros encontrados e quais campos de busca foram usados para encontrar esses valores), podemos reproduzir a execução de vários algoritmos usando políticas de avaliação distintas. Dessa forma, *iRec* pode ajudar a comunidade RS orientando novos pesquisadores no campo da aprendizagem on-line e também oferece a possibilidade de pesquisadores especialistas validarem e contrastarem seus algoritmos com várias linhas de base. Demonstramos todas essas vantagens através de dois estudos de caso de cenários distintos. Em cada um dos estudos, elaboramos um experimento completo a fim de exemplificar todo o processo de configuração, desde a entrada dos dados até avaliação e visualização dos resultados.

## 2. Contribuições

A **primeira grande contribuição dessa dissertação** é uma revisão sistemática completa da literatura sobre MAB no campo de recomendação, visando responder duas perguntas:

**QP1**: Como os modelos MAB vêm sendo avaliados no cenário de sistemas de recomendação interativa?

**QP2**: Existe na literatura algum *framework* capaz de englobar todas etapas (desde a coleta dos dados até a avaliação dos resultados) para experimentação off-line para o cenário de sistemas de recomendação interativos?

Com relação à primeira pergunta, em revisão sistemática da literatura (RSL) [Silva et al. 2022a], foram levantadas diversas informações que nos permitem fazer inúmeras análises sobre como os trabalhos atuais avaliam seus modelos de MAB. Podemos observar que a grande maioria dos trabalhos optam por estratégias que dificultam o processo de reprodutibilidade. Percebe-se que não há um consenso quanto às etapas de pré-processamento que devem ser aplicadas. Além disso, as avaliações utilizadas para medir o quão bom é um recomendador ou estão focadas apenas no processo de aprendizagem, o que está muito relacionado à teoria de aprendizagem por reforço, ou estão focadas apenas no resultado final das recomendaçãoes (acurácia das recomendações aos usuários). Sendo que conceitos como novidade, diversidade e cobertura também são de extrema importância para medir o desempenho de um recomendador.

Para responder a segunda pergunta deste trabalho, exploramos e analisamos, profundamente, diversos *frameworks* e bibliotecas de avaliação já existentes na literatura. Dessa forma, encontramos alguns trabalhos que tentam lidar com a avaliação dos modelos MAB no cenário de recomendação interativa: *BEARS* [Barraza-Urbina et al. 2018] *MABWise* [Strong et al. 2021, Strong et al. 2019] e *OBP* [Saito et al. 2020]. Para cada uma dessas propostas, apresentamos detalhadamente a arquitetura de cada uma delas, bem como seus principais componentes. Em seguida, realizamos uma comparação dessas três propostas encontradas na literatura com o nosso *Framework iRec*. Através desse estudo, verificamos que nenhum dos trabalhos existentes até o momento, engloba integralmente o processo experimental de avaliação de um sistema de recomendação desde a entrada dos dados até a metodologia de avaliação. Apenas o *iRec* é capaz de cobrir todo esse processo.

A **segunda grande contribuição desta dissertação** é ser uma referência no processo de avaliações reprodutíveis de SsR interativos. Nesse contexto, a direção seguida para se chegar à essa contribuição foi responder as seguintes perguntas:

**QP3**: É possível prover um ambiente de execução reprodutível para SsR interativos capaz de amenizar a lacuna presente no processo de avaliação das soluções de MAB?

**QP4**: O quão abrangente um ambiente reprodutível pode ser a fim de suprir as necessidades dos diversos cenários em SsR interativos?

Com objetivo de responder a terceira pergunta, propomos o *iRec* apresentando toda a sua estrutura, bem como bases de dados, algoritmos, métricas e políticas de avaliação de última geração integradas a ele, além de demonstrar como o *iRec* pode ser facilmente configurado para criação de diversos experimentos no cenário de recomendação interativa. Por fim, para responder a quarta pergunta, apresentamos a utilização de nosso *framework*, que foi instanciado em diversos cenários de recomendação

interativa, tais como: músicas, filmes, livros e pontos de interesse (POI), demonstrando toda a amplitude que um ambiente como *iRec* pode trazer [Silva et al. 2022c].

**Publicações:** Esta dissertação de mestrado resultou em 6 artigos: (1) SIGIR 2022[1] (A1) [Silva et al. 2022c]; (2) periódico Expert Systems With Applications 2021 (A1) [Silva et al. 2022a]; (3) periódico ACM Transactions on Recommender Systems [Silva et al. 2023]. (4) WebMedia 2020 (A3) [Silva et al. 2020]; (5) WebMedia 2021 (A3) [Silva et al. 2021]; (6) WebMedia 2022 (A3) [Silva et al. 2022b];

## 3. iRec - Um framework para Recomendações Interativas

O *iRec*[2] é um *framework* proposto para viabilizar o uso de modelos interativos, em especial aqueles baseados em modelos Multi-Armed Bandit, no domínio de recomendação. O *iRec* é composto de três componentes principais que abrangem todo o processo de experimentação: (1) a construção de um **Environment**; (2) a definição de **Recommendation Agent**; e (3) a definição de uma **Experimental Evaluation**. Esses componentes são ilustrados na Figura 1 por cores diferentes e são discutidos a seguir.

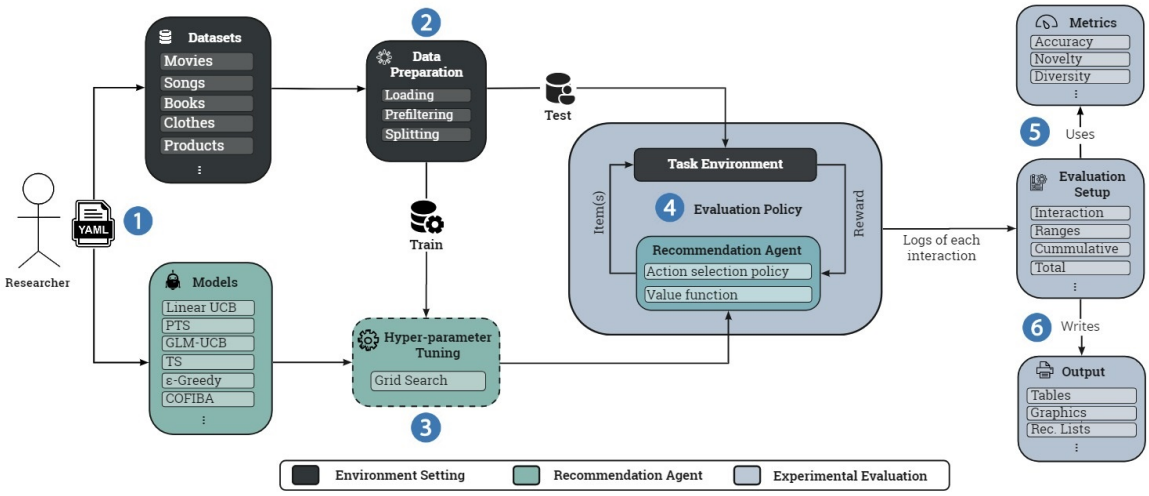


**Figure 1. Uma visão geral da estrutura iRec.**

No componente *Environment* configuramos toda a estrutura dos dados a serem processados pelo framework. Nele, carregamos as bases de dados desejadas e definimos todos os módulos de preparação de dados a serem aplicados a elas, tais como as estratégias de pré-filtragem e divisão dos conjuntos de treino e teste. Atualmente, o *iRec* possui 17 datasets públicos relacionados a diversos cenarios de aplicação, tais como filmes, músicas, pontos de interesse, produtos e roupas. Por sua vez, no componente *Recommendation Agent* seleciona o modelo que será utilizado para definir o(s) melhor(es) item(ns) para cada usuário em cada iteração. Em outras palavras, é nesse componente que implementam-se os SsR que serão utilizados na recomendação. O *iRec* possui diversos algoritmos estado-da-arte já implementados. Por fim, o componente *Experimental Evaluation* é responsável por realizar a integração dos modelos propostos no *Recommendation Agent* sobre os dados especificados no primeiro componente - *Environment*. Primeiramente, precisamos definir como deve ser a interação entre esses dois componentes por meio de um módulo denominado *Evaluation Policy*. Nele,

[1]Conferência mais importante do mundo na área de recuperação de informação - h5-index: 75
[2]Disponível em: `https://github.com/irec-org`

configuramos um ambiente de Aprendizado por Reforço, no qual um item (ou um conjunto de itens) é recomendado em cada iteração do algoritmo (ou seja, tentativa). Por um tempo pré-determinado (número de tentativas), o *Recommendation Agent* realiza as recomendações dentro do *Environment*, recebendo uma recompensa positiva ou negativa e atualizando seu conhecimento (se necessário). Todos os *logs* (i.e., registro das ações realizadas pelo *Recommendation Agent* e recompensas fornecidas pelo usuários, de acordo com os dados no conjunto de teste) são armazenados. Por fim, o *Experimental Evaluation* analisa esses logs, aplica as métricas de avaliação de recomendações e realiza os testes estatísticos necessários para realizar uma avaliação adequada do desempenho dos SsR implementados no *Recommendation Agent*. Nesta dissertação, consideramos os principais SsR baseados em modelos MAB (*Recommendation Agent*) para serem avaliados (*Experimental Evaluation*) em dois cenários distintos: entretenimento e Pontos de Interesse (POIs). Por um questão de limitação de espaço, na próxima seção, apresentaremos apenas a aplicação do *iRec* no cenário de entretenimento (i.e., música, filmes e livros). Os resultados relacionados ao outro cenário, bem como o detalhamento dos componentes do *iRec* podem ser encontrados no arquivo original da dissertação.

## 4. Avaliação Experimental

Com o objetivo de demonstrar a amplitude que toda essa estrutura do *iRec* tem sobre as pesquisas atuais focadas em SsR interativos, na dissertação construímos dois estudos de caso. Em cada um dos estudos, definimos as configurações de cada conjunto de dados utilizados, as configurações de cada agente, como realizar a tunagem de parâmetros e, por fim, como avaliar adequadamente os resultados dos modelos MAB. Dessa forma, a única coisa que difere os dois estudos são os parâmetros utilizados em cada um deles. Conforme mencionado anteriormente, por restrições de espaço, apresentamos nesse artigo a avaliação apenas para o cenário de entretenimento.

### 4.1. Configurações

Para a recomendação no cenário de entretenimento, selecionamos três conjuntos de dados: *Netflix* (Filmes), *Good Books* (Livros) e *Yahoo Music* (Música), detalhados na Tabela 1.

| Coleções de Dados | Domínio | # Usuários | # Itens | Esparsidade |
|---|---|---|---|---|
| Netflix | Filmes | 10.000 | 17.372 | 98,67% |
| GoodBooks | Livros | 53.423 | 10.000 | 98,88% |
| Yahoo Music R1 | Música | 10.000 | 13.214 | 99,22% |

**Table 1. Visão geral dos três conjuntos de dados do cenário de entretenimento.**

De maneira geral, o *iRec* pode ser instanciado como uma biblioteca, estendendo os módulos e executando cada ação, ou mesmo executando linhas de comando. Para ambos os estudos, optamos por linhas de comando para demonstrar como configurar cada arquivo *yaml* necessário no processo. Dessa forma, demonstramos o processo de configuração de um experimento para os três componentes principais de *iRec*. Em resumo, nossa estrutura se inicia com o componente Environment Setting. Por este componente, um pesquisador pode escolher uma base de dados a ser carregada para o experimento e definir e executar todos os módulos de preparação de dados a serem aplicados a ela, como a pré-filtragem e a estratégia de divisão de dados para criar os

conjuntos de treinamento e teste. No *iRec*, enquanto o conjunto de treinamento é usado para ajustar os algoritmos de recomendação, o conjunto de teste representa o público-alvo das recomendações no cenário interativo (task environment). Neste trabalho, o *iRec* é focado em recomendações interativas personalizadas. Consequentemente, todos os modelos visam identificar o(s) melhor(es) item(ns) para cada usuário em cada iteração. O Agente de Recomendação escolhe o(s) item(ns) de acordo com o modelo implementado. A interação do agente com o ambiente ocorre por meio de uma *Evaluation Policy* definida pelo pesquisador. Essa política é semelhante às abordagens clássicas de Aprendizado por Reforço, onde um item (ou um conjunto de itens) é recomendado em cada iteração do algoritmo (ou seja, tentativa). Por um tempo pré-determinado (número de tentativas), o agente realiza as recomendações dentro do *task environment*, recebendo uma recompensa positiva ou negativa e atualizando seu conhecimento (se necessário). Todos os logs (ou seja, ações realizadas pelo agente e recompensas fornecidas pelo público-alvo) são então registrados pela estrutura para permitir uma avaliação adequada. O componente *Experimental Evaluation* analisa esses logs, aplica as métricas de avaliação de recomendações e realiza todos os testes estatísticos necessários para realizar uma avaliação justa. Todos os arquivos *yaml* podem ser observados no arquivo original da dissertação.

### 4.2. Resultados

A Tabela 2 mostra os resultados experimentais gerados pela configuração de *iRec* com as configurações descritos anteriormente. Essa tabela foi gerada automaticamente pelo nosso *framework*, no qual podemos analisar o comportamento de diferentes modelos de recomendação nos diferentes cenários selecionados. Nesta avaliação experimental, aplicamos o teste estatístico *Wilcoxon* (computado em pares de modelos com sua melhor configuração) e utilizamos métricas de accuracy, coverage, diversity, e novelty, apresentando diferentes formas de avaliar o desempenho de um modelo. Além dos modelos MAB clássicos ($\epsilon$-Greedy, UCB e Thompson Sampling), executamos outros modelos MAB mais recentes para enriquecer a análise dos dados obtidos.

Analisando os resultados, é importante observar que nenhum deles obteve os melhores resultados para todas as métricas avaliadas, demonstrando a importância de utilizar diferentes métricas de avaliação adequadas ao cenário de recomendação. Os SsR devem ser capazes de fornecer itens relevantes, diversos e novos, suprindo as necessidades de consumo da maioria dos usuários de um sistema de recomendação. Outro ponto a destacar é a importância dos testes estatísticos para validar a comparação dos resultados. Por exemplo, analisando a métrica *Hits* na 10ª interação, no conjunto de dados da *Yahoo Music*, vemos que, apesar da superioridade do NICF, o modelo do PTS empata estatisticamente com ele. Dessa forma, esses resultados mostram o quão importante é definir bem o protocolo de avaliação e analisar o comportamento dos modelos em diferentes etapas. Podemos ver que nenhum modelo foi superior em todas as interações. O CLinUCB, apesar de perder nas primeiras 10 interações, em *Hits* e *Recall* em todas as bases, observa-se que a médio prazo (50 interações) e longo prazo (100 interações), este mesmo modelo se torna extremamente superior aos demais. Conforme apresentado na Seção 1.1, o número de tentativas a serem realizadas, o número de itens a serem recomendados em cada tentativa e se o usuário pode receber as mesmas recomendações nas próximas tentativas, entre muitos outros parâmetros de avaliação, impactam diretamente nos resultados finais.

Além disso, Métricas de diversidade, novidade e cobertura também devem ser

| Dataset | Yahoo Music | | | Netflix | | | Good Books | | |
|---|---|---|---|---|---|---|---|---|---|
| Métrica | Hits | | | Hits | | | Hits | | |
| T | 10 | 50 | 100 | 10 | 50 | 100 | 10 | 50 | 100 |
| CLinUCB | 2.336 | **17.922▲** | **27.260▲** | 1.461 | **15.184▲** | **25.947▲** | 1.776 | **11.759▲** | **18.623▲** |
| UCB | 1.358 | 7.330 | 13.277 | 1.284 | 5.440 | 9.915 | 0.764 | 2.984 | 5.493 |
| e-Greedy | 1.460 | 7.424 | 13.360 | 1.320 | 5.390 | 9.936 | 0.800 | 3.072 | 5.633 |
| TS | 1.907 | 8.356 | 14.720 | 1.882 | 7.498 | 12.959 | 1.361 | 4.528 | 7.216 |
| Linear UCB | 3.157 | 15.514 | 25.361 | 1.980 | 12.076 | 22.361 | 1.586 | 6.848 | 12.593 |
| Linear e-Greedy | 0.011 | 0.316 | 1.059 | 0.158 | 2.303 | 6.037 | 0.060 | 0.815 | 2.543 |
| Cluster Bandit | 2.428 | 10.095 | 16.777 | 2.082 | 8.649 | 16.265 | 2.187 | 7.863 | 12.173 |
| NICF | **3.345●** | 11.457 | 14.951 | **2.837▲** | 10.268 | 15.690 | **2.960▲** | 7.785 | 10.452 |
| PTS | **3.329●** | 14.363 | 19.634 | 0.442 | 3.635 | 8.901 | 1.687 | 7.386 | 12.951 |
| ICTRTS | 0.138 | 6.149 | 13.446 | 0.046 | 2.865 | 6.205 | 1.159 | 7.013 | 11.293 |
| Métrica | Recall | | | Recall | | | Recall | | |
| T | 10 | 50 | 100 | 10 | 50 | 100 | 10 | 50 | 100 |
| CLinUCB | 0.059 | **0.455▲** | **0.648▲** | 0.022 | **0.283▲** | **0.442▲** | 0.024 | **0.154▲** | **0.241▲** |
| UCB | 0.034 | 0.179 | 0.321 | 0.017 | 0.075 | 0.141 | 0.010 | 0.039 | 0.072 |
| e-Greedy | 0.036 | 0.183 | 0.323 | 0.017 | 0.074 | 0.141 | 0.011 | 0.041 | 0.074 |
| TS | 0.048 | 0.204 | 0.354 | 0.025 | 0.101 | 0.177 | 0.018 | 0.060 | 0.095 |
| Linear UCB | 0.086 | 0.388 | 0.597 | 0.031 | 0.208 | 0.360 | 0.021 | 0.089 | 0.163 |
| Linear e-Greedy | 0.000 | 0.006 | 0.018 | 0.001 | 0.014 | 0.042 | 0.001 | 0.011 | 0.034 |
| Cluster Bandit | 0.064 | 0.254 | 0.413 | 0.043 | 0.147 | 0.273 | 0.030 | 0.105 | 0.161 |
| NICF | **0.086●** | 0.280 | 0.352 | **0.051▲** | 0.168 | 0.247 | **0.040▲** | 0.103 | 0.137 |
| PTS | **0.088●** | 0.359 | 0.473 | 0.007 | 0.061 | 0.155 | 0.023 | 0.098 | 0.170 |
| ICTRTS | 0.003 | 0.153 | 0.327 | 0.001 | 0.045 | 0.092 | 0.015 | 0.093 | 0.149 |
| Métrica | ILD | | | ILD | | | ILD | | |
| T | 10 | 50 | 100 | 10 | 50 | 100 | 10 | 50 | 100 |
| CLinUCB | 0.462 | 0.423 | 0.435 | 0.399 | 0.372 | 0.378 | 0.472 | 0.453 | 0.464 |
| UCB | 0.465 | 0.463 | 0.465 | 0.404 | 0.416 | 0.421 | **0.490▲** | **0.495▲** | **0.495▲** |
| e-Greedy | 0.461 | 0.462 | 0.465 | 0.401 | 0.416 | 0.421 | 0.489 | 0.494 | 0.495 |
| TS | 0.431 | 0.452 | 0.459 | 0.336 | 0.373 | 0.387 | 0.467 | 0.487 | 0.492 |
| Linear UCB | 0.387 | 0.418 | 0.436 | 0.375 | 0.387 | 0.394 | 0.428 | 0.469 | 0.476 |
| Linear e-Greedy | 0.466 | **0.478▲** | **0.488▲** | 0.481 | **0.482▲** | **0.481▲** | 0.487 | 0.488 | 0.488 |
| Cluster Bandit | 0.437 | 0.448 | 0.455 | 0.429 | 0.392 | 0.391 | 0.384 | 0.451 | 0.468 |
| NICF | 0.405 | 0.433 | 0.464 | 0.339 | 0.368 | 0.394 | 0.392 | 0.461 | 0.479 |
| PTS | 0.406 | 0.434 | 0.461 | 0.475 | 0.464 | 0.462 | 0.465 | 0.471 | 0.477 |
| ICTRTS | **0.490▲** | 0.475 | 0.467 | **0.493▲** | 0.460 | 0.446 | 0.480 | 0.465 | 0.474 |
| Métrica | UsersCoverage | | | UsersCoverage | | | UsersCoverage | | |
| T | 10 | 50 | 100 | 10 | 50 | 100 | 10 | 50 | 100 |
| CLinUCB | 0.850 | **0.983▲** | **0.992▲** | 0.690 | **0.967▲** | **0.986●** | **0.764▲** | **0.934▲** | **0.968▲** |
| UCB | 0.664 | 0.960 | 0.986 | 0.544 | 0.872 | 0.949 | 0.483 | 0.842 | 0.930 |
| e-Greedy | 0.684 | 0.960 | 0.985 | 0.545 | 0.857 | 0.954 | 0.499 | 0.853 | 0.932 |
| TS | 0.748 | 0.960 | 0.986 | 0.631 | 0.895 | 0.957 | 0.636 | 0.893 | 0.944 |
| Linear UCB | 0.806 | 0.967 | 0.990 | 0.588 | 0.920 | 0.963 | 0.489 | 0.819 | 0.915 |
| Linear e-Greedy | 0.005 | 0.038 | 0.075 | 0.078 | 0.129 | 0.244 | 0.037 | 0.181 | 0.306 |
| Cluster Bandit | 0.814 | 0.965 | 0.986 | 0.844 | 0.963 | **0.986●** | 0.673 | 0.926 | 0.964 |
| NICF | 0.839 | 0.956 | 0.967 | **0.845▲** | 0.959 | 0.977 | 0.747 | 0.920 | 0.957 |
| PTS | **0.858▲** | 0.977 | 0.989 | 0.320 | 0.866 | 0.952 | 0.673 | 0.908 | 0.961 |
| ICTRTS | 0.117 | 0.954 | 0.982 | 0.045 | 0.806 | 0.928 | 0.608 | 0.922 | 0.962 |

**Table 2. Performance dos modelos *bandit* no cenário de entretenimento. Os resultados foram comparados com o teste de Wilcoxon com p-value $= 0.05$. O símbolo ▲ denota ganhos estatísticos e o símbolo ● representa empates.**

consideradas ao avaliar um sistema de recomendação. Em se tratando de MAB, recomendar itens mais diversos, está relacionado ao conceito de *exploration*, ou seja, o sistema está optando por recomendar um item mais distinto capaz de agregar mais conhecimento ao sistema. Por outro lado, ao recomendar itens mais populares, por exemplo, o sistema está em busca do item potencialmente mais relevante até o momento para satisfazer ao usuário imediatamente. Com base no próprio consenso existente na literatura, sabemos que explorar apenas *exploration* ou *explotation* pode trazer prejuízos ao sistema. Um sistema que exige um esforço inicial ou que apresenta itens pouco relevantes para os usuário pode se comprometer facilmente. Em contrapartida, apresentar itens potencialmente relevantes, explorando o conhecimento existente no sistema, como por exemplo, a popularidade dos

itens, adiciona pouco conhecimento sobre o usuário, uma vez que todo mundo tende a se interessar por tais itens (ou eles não seriam os populares). Logo, estamos atrás de um modelo que consiga equilibrar esses dois conceitos. Sendo assim, se analisarmos a tabela 2, apesar de não haver nenhum algoritmo capaz de demonstrar superioridade em todas métricas e interações apresentadas, vemos que o CLinUCB [Silva et al. 2021] é o que possui melhor desempenho em todas as bases de dados. Além dos melhores resultados em *Hits* e *Recall*, este modelo consegue cobrir uma grande porcentagem de usuários distintos que estão interessados nos itens a eles recomendados (*Users Coverage*).

Portanto, para dizer que um modelo é melhor que outro, não basta avaliar apenas uma métrica e os resultados a curto, médio ou longo prazo. Um modelo de recomendação deve permanecer constante, apresentando bons resultados em diferentes etapas e demonstrar superioridade ou ao menos competitividade em relação a diferentes métricas de avaliação. Ademais, o *framework iRec* também permite aos pesquisadores melhorar a comparação dos modelos e garantir a reprodutibilidade de todos os experimentos. O processo de ajuste dos hiperparâmetros para todos os modelos de recomendação permite aos pesquisadores obter os melhores resultados para cada modelo e fornecer uma comparação justa entre as linhas de base [Dacrema et al. 2021]. Desconsiderando o processo de ajuste e definindo arbitrariamente os parâmetros o TS marcou 1,572, 6,943 e 11,291, na base da *Netflix*, em termos de Hits nas interações 10º, 50º, e 100º, respectivamente. Como esperado, após o ajuste, o algoritmo TS obteve os melhores resultados para essas interações, conforme mostrado na Tabela 2. Como o *iRec* armazena todas as sementes aleatórias desde a preparação dos dados até o processo de avaliação, além dos melhores parâmetros obtidos após o ajuste dos hiperparâmetros, podemos garantir que os resultados sejam os mesmos independentemente do computador. Esses experimentos, por exemplo, foram executados em cinco computadores diferentes com diferentes configurações de hardware e obtivemos os mesmos resultados para os métodos determinísticos.

## 5. Conclusão

Nesta dissertação, apresentamos o *iRec*[3], um framework completo para avaliação de sistemas interativos de recomendação (RS) que visa lidar com a falta de consenso sobre as melhores práticas de avaliação nesta área [Sun et al. 2020], fornecendo um ambiente completo para uma avaliação reprodutível e comparações justas de sistemas de recomendação. Sua estrutura oferece compatibilidade com conjuntos de dados abrangentes adotados na literatura, diferentes estratégias de pré-processamento de dados, dezessete modelos de recomendação, otimização completa de hiperparâmetros e nove métricas de avaliação com objetivos distintos (i.e., accuracy, novelty, diversity, coverage, etc.). Além disso, o *iRec* fornece várias políticas de avaliação atualmente utilizadas na literatura, testes estatísticos para comparar o desempenho dos algoritmos e diferentes visualizações de resultados.

Para demonstrar os benefícios proporcionados pelo nosso framework, avaliamos o *iRec* com base em dois estudos de caso para demonstrar toda a abrangência que sua estrutura pode trazer nas pesquisas atuais focadas em SsR interativos. Em cada um dos estudos, elaboramos um experimento completo a fim de exemplificar todo o processo de configuração, desde a entrada dos dados até avaliação e visualização dos resultados. Em

[3]Disponível em: `https://github.com/irec-org`

suma, no primeiro estudo focamos no cenário de entretenimento, no qual selecionamos três conjuntos de dados amplamente conhecidas em seus domínios: *Netflix* (Filmes), *Good Books* (Livros) e *Yahoo Music* (Música). Por sua vez, no segundo estudo, focamos em um cenário novo de aplicação de modelos de MAB para recomendação interativa de POIs. Para este segundo cenário, selecionamos três coleções de dados reais provenientes da Yelp, relacionadas a três cidades estadunidenses: *Philadelphia*, *Nashville* e *New Orleans*. Dessa forma, para ambos, demonstramos o processo de configuração de um experimento, em detalhes, para os três componentes principais de *iRec*: *Environment*, *Recommendation Agent* e *Experimental Evaluation*. Enfim, para cada estudo, apresentamos os resultados nos quais pudemos avaliar o comportamento desses modelos nos diferentes cenários utilizados. Analisando os resultados obtidos no Estudo 1, vimos a capacidade que o *iRec* possui em promover um ambiente totalmente completo para avaliação em um dos cenários mais utilizados nos trabalhos atuais: o de entretenimento. No Estudo 2 (apenas na dissertação), apesar dos modelos MAB ainda não serem explorados em POI, demonstramos como é fácil a criação de um experimento nessa área. Portanto, deixamos claro que o *iRec* pode ser estendido, incluindo diferentes coleções de dados, políticas de avaliação, estratégias de recomendação e fontes de dados, de maneira extremamente intuitiva e prática. trata-se de um marco na literatura de sistemas de recomendação interativas, uma vez que, além da oportunidade de aplicação dos modelos de recomendação de forma sistematizada e simplificada, os pesquisadores da área podem usar o *iRec* para avançar em seus estudos e experimentos, garantindo o uso de melhores práticas em SsR. Com o *iRec* é possível estender os trabalhos na área e também estabelecer um *benchmark* para os trabalhos da literatura de sistemas de recomendação.

Para trabalhos futuros, de forma mais imediata, planejamos estender *iRec* para incluir novas estratégias de filtragem e divisão de dados ideais para cada um dos cenários suportados por *iRec*. Sabemos que, dependendo do cenário aplicado, algumas técnicas de divisão de dados, por exemplo, podem ser melhores para desenvolvimento de um processo experimental. Além disso, temos como objetivo adicionar novas técnicas de ajuste de parâmetros, como a estratégia Bayesiana, randômica, dentre outras que são amplamente utilizadas no processo de otimização (tunagem) de parâmetros de algoritmos de aprendizado de máquina. Novas políticas e métricas de avaliações também podem ser adicionadas para tornar ainda mais completa a avaliação de modelos *Multi-Armed Bandits*. A longo prazo, nossa meta é utilizar o *iRec* nos demais trabalhos de nosso grupo de pesquisa relacionado a modelos MAB em recomendação interativas. Nosso laboratório conta hoje com outros alunos de iniciação científica, mestrado e doutorado trabalhando com soluções para diversos problemas de recomendação interativas, tais como esparsidade, privacidade, *cold-start* e explicabilidade. Além de utilizarem o *iRec* como ferramenta de apoio, em alguns casos também serão feitas propostas de melhorias e extensões para que o mesmo também seja capaz de realizar experimentações detalhadas de recomendação interativas para os problemas mencionados.

## Agradecimentos

Esse trabalho foi parcialmente financiado por AWS, CNPq, CAPES, FINEP e Fapemig.

## References

Barraza-Urbina, A., Koutrika, G., d'Aquin, M., and Hayes, C. (2018). Bears: Towards an evaluation framework for bandit-based interactive recommender systems. *REVEAL 18, October 6-7, 2018, Canada*.

Dacrema, M. F., Boglio, S., Cremonesi, P., and Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM TOIS*, 39(2).

Saito, Y., Shunsuke, A., Megumi, M., and Yusuke, N. (2020). Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*.

Sanz-Cruzado, J., Castells, P., and López, E. (2019). A simple multi-armed nearest-neighbor bandit for interactive recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 358–362.

Shams, S., Anderson, D., and Leith, D. (2021). Cluster-based bandits: Fast cold-start for recommender system new users.

Silva, N., Silva, T., Werneck, H., Rocha, L., and Pereira, A. (2023). User cold-start problem in multi-armed bandits: When the first recommendations guide the user's experience. *ACM Trans. Recomm. Syst.*, 1(1).

Silva, N., Werneck, H., Silva, T., Pereira, A. C., and Rocha, L. (2021). A contextual approach to improve the user's experience in interactive recommendation systems. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 89–96.

Silva, N., Werneck, H., Silva, T., Pereira, A. C., and Rocha, L. (2022a). Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. *Expert Systems with Applications*, 197:116669.

Silva, T., Silva, N., Mito, C., Pereira, A. C. M., and Rocha, L. (2022b). Interactive poi recommendation: Applying a multi-armed bandit framework to characterise and create new models for this scenario. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, WebMedia '22, page 211–221, New York, NY, USA. Association for Computing Machinery.

Silva, T., Silva, N., Werneck, H., Mito, C., Pereira, A. C., and Rocha, L. (2022c). Irec: An interactive recommendation framework. In *Proceedings of the 45th International ACM SIGIR*, page 3165–3175, New York, NY, USA. Association for Computing Machinery.

Silva, T., Silva, N., Werneck, H., Pereira, A. C., and Rocha, L. (2020). The impact of first recommendations based on exploration or exploitation approaches in recommender systems' learning. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 173–180.

Strong, E., Kleynhans, B., and Kadioglu, S. (2019). Mabwiser: A parallelizable contextual multi-armed bandit library for python. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019*, pages 909–914. IEEE.

Strong, E., Kleynhans, B., and Kadioglu, S. (2021). MABWiser: parallelizable contextual multi-armed bandits. *Int. J. Artif. Intell. Tools*, 30(4).

Sun, Z., Yu, D., Fang, H., Yang, J., Qu, X., Zhang, J., and Geng, C. (2020). Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In *Fourteenth ACM conference on recommender systems*, pages 23–32.

Wu, Q., Iyer, N., and Wang, H. (2018). Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR*, pages 495–504.

Wu, Q., Zhang, H., Gao, X., He, P., Weng, P., Gao, H., and Chen, G. (2019). Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*, pages 2091–2102.

Zhou, S., Dai, X., Chen, H., Zhang, W., Ren, K., Tang, R., He, X., and Yu, Y. (2020). Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR*.

Zou, L., Xia, L., Gu, Y., Zhao, X., Liu, W., Huang, J. X., and Yin, D. (2020). Neural interactive collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR*, pages 749–758.