UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Sistema de recomendação para produtos financeiros

Luiz Carlos Barbosa e Silva

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Luiz Carlos Barbosa e Silva

Sistema de recomendação para produtos financeiros

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Marcelo Garcia Manzato

Versão original

São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

S586s

Silva, Luiz Carlos Barbosa e
Sistema de recomendação para produtos financeiros / Luiz Carlos Barbosa e Silva; orientador Marcelo Garcia Manzato. -- São Carlos, 2024.
57 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Sistema de recomendação. 2. Filtragem colaborativa. 3. Feedback implícito. 4. Produtos financeiros. 5. Produtos bancários. I. Manzato, Marcelo Garcia, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Luiz Carlos Barbosa e Silva

Recommendation system for financial products

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Marcelo Garcia Manzato

Original version

São Carlos

2024

Aos meus filhos, Carmen, Rodrigo e Miguel, e a meu enteado, Gustavo, para que seja uma inspiração na busca constante por conhecimento, em qualquer área que queiram.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão ao meu orientador, Prof. Dr. Marcelo Garcia Manzato, por suas valiosas orientações, paciência e apoio ao longo deste trabalho. Sua expertise, disponibilidade e dedicação foram cruciais para o desenvolvimento deste projeto e para o meu crescimento acadêmico.

Agradeço também ao Itaú Unibanco, onde tive a oportunidade de construir minha carreira e adquirir um conhecimento profundo sobre o negócio bancário. A experiência e os aprendizados que acumulei ao longo desses anos foram essenciais para a realização deste trabalho.

Por fim, meu mais sincero e profundo agradecimento à minha esposa, Flavia. Seu constante incentivo, compreensão e apoio incondicional em diversas áreas foram fundamentais para que eu pudesse manter o foco necessário e concluir este trabalho com sucesso. Sou eternamente grato, amor meu!

RESUMO

SILVA, L.C.B. **Sistema de recomendação para produtos financeiros**. 2024. 57 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Considerando um cenário de mercado cada vez mais competitivo, com clientes cada vez mais exigentes, os sistemas de recomendação têm se tornado uma ferramenta essencial para o sucesso dos negócios, incluindo o setor financeiro. Recomendações personalizadas de produtos aumentam o engajamento e a satisfação dos clientes, reduzem custos de vendas e incrementam receitas, especialmente em um ambiente altamente regulado, como o das instituições financeiras brasileiras. Neste contexto desafiador, este trabalho apresenta o desenvolvimento de um sistema de recomendação de produtos financeiros voltado para clientes pessoa física de um banco, utilizando dados históricos de comportamento de consumo desses clientes, frequentemente representados por *feedback* implícito, cuja disponibilidade é bastante comum em ambientes de negócios reais. Embora o dataset utilizado contenha dados fictícios de uma competição promovida no site *Kaggle* por um banco espanhol, ele foi escolhido por sua adequação para testar a viabilidade do sistema proposto. Os resultados obtidos com o algoritmo escolhido, *Bayesian Personalized Ranking* (BPR), superaram as alternativas, como o algoritmo *ItemKNN* e as recomendações aleatórias, especialmente nas métricas de precisão das recomendações. Este trabalho pretende ainda contribui para intensificar o uso de dados em instituições financeiras, demonstrando como é possível aumentar a eficiência e personalização das ofertas de produtos financeiros aos clientes. Futuras pesquisas poderiam focar na incorporação de abordagens híbridas e na utilização de dados reais de clientes bancários brasileiros, para validar e potencializar a eficácia do modelo proposto.

Palavras-chave: Sistema de recomendação. Filtragem colaborativa. Feedback implícito. Produtos financeiros. Produtos bancários.

ABSTRACT

SILVA, L.C.B. **Recommendation system for financial products**. 2024. 57 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

In an increasingly competitive market environment, with ever more demanding customers, recommendation systems have become an essential tool for business success, including in the financial sector. Personalized product recommendations enhance customer engagement and satisfaction, reduce sales costs, and increase revenues, especially in a highly regulated environment like that of Brazilian financial institutions. In this challenging context, this work presents the development of a recommendation system for financial products aimed at individual clients of a bank, using historical consumption behavior data, often represented by implicit feedback, which is commonly available in real business environments. Although the dataset used contains fictitious data from a competition hosted on the Kaggle website by a Spanish bank, it was chosen for its suitability in testing the feasibility of the proposed system. The results obtained with the selected algorithm, Bayesian Personalized Ranking (BPR), outperformed alternatives such as the ItemKNN algorithm and random recommendations, particularly in recommendation accuracy metrics. This work also aims to contribute to the increased use of data in financial institutions, demonstrating how it can enhance the efficiency and personalization of financial product offerings to clients. Future research could focus on incorporating hybrid approaches and using real data from Brazilian banking clients to validate and further improve the effectiveness of the proposed model.

Keywords: Recommendation system. Collaborative filtering. Implicit feedback. Financial products. Banking products.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BPR	*Bayesian personalized ranking*
FBC	Filtragem baseada em conteúdo
FC	Filtragem colaborativa
SR	Sistema de recomendação
SRH	Sistema de recomendação híbrido

SUMÁRIO

1 INTRODUÇÃO

1.1 Contexto

Em um mundo com cada vez mais abundância de dados, existe grande expectativa, tanto por parte dos clientes quanto dos fornecedores de produtos e serviços, de se extrair o máximo de valor desses dados. Nesse contexto, os sistemas de recomendação têm desempenhado um papel fundamental em muitas indústrias, incluindo os setores bancário e financeiro.

Com a crescente demanda por experiências de consumo personalizadas, os sistemas de recomendação têm sido decisivos nesse mercado, oferecendo benefícios tanto para clientes quanto para as instituições financeiras. São eles (Bouvier, 2021; Drozdov, 2023):

- **Aumento de engajamento e satisfação de clientes**: ao apresentar recomendações personalizadas, que vão de encontro com seus interesses e necessidades específicos, os clientes têm uma melhor experiência de consumo, evitando a sobrecarga de informações e alternativas de escolha e resultando em um incremento de relacionamento e fidelidade com o banco;
- **Redução de custo**: ofertar produtos e serviços personalizados minimiza o esforço de venda, pois reduz a necessidade de campanhas de marketing massivas e, muitas vezes, ineficazes;
- **Incremento de vendas e receita**: ao utilizar recomendações personalizadas, aumenta-se a chance do cliente adquirir produtos e serviços adicionais, trazendo maior resultado para a instituição.

Vale ressaltar ainda que, no Brasil, o setor bancário é altamente regulado, o que limita muito as possibilidades de inovação e diferenciação de produtos e serviços entre os bancos concorrentes. Isso faz com que a utilização de sistemas de recomendação tenha um papel estratégico no sucesso do negócio bancário.

1.2 Objetivo

Considerando a alta disponibilidade de dados de clientes que os bancos têm a disposição, podemos dizer que hoje não é apenas condição competitiva, mas sim de existência, que o uso dos dados seja intensificado e maximizado, de forma que todo o potencial de conhecimento sobre o cliente e eficiência operacional possam ser explorados. Caso não o faça, provavelmente a concorrência o fará, levando a perda de clientes e resultado.

No mercado atual, não há mais espaço para listas frias de prospecção de clientes, baseadas em dados estáticos e categorizações básicas, que elevam o esforço e o custo de oferta de produtos, tendo um baixo índice de aproveitamento.

Além de tornar o processo de vendas mais eficiente, personalizar ofertas para os clientes tem um impacto direto no nível de satisfação, sobretudo no caso dos bancos, para que a instituição financeira possa ser percebida como uma conselheira confiável, capaz de antecipar necessidades os seus clientes, ao invés de uma empresa que insiste em "empurrar" seus produtos de acordo com seus próprios interesses.

Com base nisso, este trabalho tem como objetivo desenvolver um sistema de recomendação de produtos financeiros para clientes pessoa física de um banco, utilizando dados históricos de consumo. O sistema visa prever quais produtos terão maior interesse para cada cliente, ou seja, aqueles com maior probabilidade de serem contratados.

Atualmente, existem diversas técnicas e abordagens para a construção de ferramentas de recomendação, porém devemos aprofundar o entendimento de suas aplicações e avaliar qual é a mais aderente para o atingimento do objetivo proposto.

1.3 Desenvolvimento do trabalho

O desenvolvimento deste trabalho começa com a seleção de um conjunto de dados (*dataset*) e a escolha de um algoritmo apropriado para gerar as recomendações. Em seguida, procede-se à preparação dos dados, quando eles são transformados para se adequarem ao algoritmo escolhido. Durante a fase de treinamento, são realizados ciclos de execução utilizando os dados de treinamento e validação, avaliando os resultados e ajustando os parâmetros até atingir a configuração ideal.

Na etapa final, aplicamos o algoritmo com a configuração de parâmetros definida anteriormente, agora utilizando os dados de teste. Essa fase permite comparar as recomendações geradas pelo modelo com os resultados observados na prática, possibilitando a avaliação da qualidade do sistema de recomendação.

1.4 Organização do trabalho

Este trabalho está organizado em capítulos, da seguinte forma:

No capítulo 2, apresentamos a definição de sistemas de recomendação e suas principais classificações. Em seguida, discutimos questões específicas relacionadas ao uso de *feedback* implícito e detalhamos o funcionamento do algoritmo *Bayesian Personalized* Ranking (*BPR*), que é especialmente adequado para esse tipo de cenário. Também introduzimos as principais métricas utilizadas para avaliar a performance de sistemas de recomendação.

No capítulo 3, descrevemos o desenvolvimento do sistema de recomendação com base no dataset selecionado, além da metodologia empregada para alcançar os objetivos do trabalho.

No capítulo 4, exploramos o *dataset* em maior profundidade e apresentamos os resultados obtidos durante a fase de treinamento e ajuste de parâmetros, assim como a aplicação do algoritmo nos dados de teste.

Finalmente, no capítulo 5, discutimos as conclusões derivadas das análises dos resultados obtidos, as limitações da solução proposta, e sugerimos possíveis melhorias para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Definição de sistema de recomendação (SR)

Algumas das definições encontradas na literatura para SR dizem respeito a um sistema para calcular e prover conteúdo relevante para o usuário com base no conhecimento do próprio usuário, do conteúdo e das interações entre o usuário e o item, além de ser o conjunto de ferramentas e técnicas de programação que fornece sugestões para tomadas de decisão relativas de itens que podem ser úteis para um usuário (Falk, 2019; Ricci *et al.*, 2011).

Em se tratando de SR, devemos ainda considerar as seguintes definições para alguns termos utilizados (Falk, 2019):

- **Item**: termo normalmente usado para se referir a um determinado objeto que pode ser adquirido e/ou um serviço que pode ser contratado;
- **Predição**: é uma estimativa de como um usuário irá avaliar (positiva ou negativamente) ou tomar a decisão pela aquisição de um item;
- **Relevância**: é a ordem de prioridade dos itens de acordo com o que é mais importante para o usuário naquele momento;
- **Recomendação**: os primeiros N itens mais relevantes;
- **Personalização**: o quanto uma recomendação é individualizada para cada usuário.

Ainda que existam diferentes níveis de personalização para as recomendações, que vão desde as não-personalizadas, em que todos os usuários recebem as mesmas recomendações (uma lista dos itens mais populares, por exemplo), passando por recomendações semi personalizadas (ou segmentadas), em que usuários membros de um determinado segmento (por idade, por localização etc.) recebem as mesmas recomendações, nosso foco aqui é desenvolver um SR personalizado, que gere sugestões de itens para um determinado usuário (Falk, 2019).

De forma geral, podemos dizer que um SR serve para dois propósitos diferentes. De um lado, podem estimular os usuários a agirem de determinada forma, como consumir um item. Do outro, são ferramentas úteis para lidar com a sobre carga de informações (ou opções), já que indicam os itens mais relevantes dentro de um grande conjunto (Jannach *et al.*, 2010).

Apesar de muitas pessoas considerarem o uso de SR como uma forma de manipulação, já que apresentam para os usuários itens com maior propensão de serem adquiridos por

eles, com o objetivo de aumentar as vendas, é importante fazer essa distinção. Manipulação, na verdade, diz respeito mais à motivação para recomendar determinado item do que ao fato de indicá-lo, como quando, por exemplo, um vendedor lhe oferece um produto não porque é o mais indicado para a sua necessidade, mas sim porque este lhe rende uma comissão maior (Falk, 2019).

Dentro do contexto de produtos financeiros para clientes pessoa física de um banco, consideraremos que um SR desenvolvido para esta aplicação é o conjunto de ferramentas e técnicas de programação capaz de prever a necessidade mais relevante ou benéfica para o cliente naquele momento, ou seja, qual será o próximo produto (item) a ser contratado (adquirido) pelos clientes (usuários) com base nos dados disponíveis, com o objetivo de personalizar a experiência bancária, aumentando a satisfação do cliente e otimizando o esforço de vendas.

2.2 Classificação de sistemas de recomendação

Os algoritmos para SR são basicamente classificados de acordo com o tipo de dado utilizado para se fazer recomendações. Os algoritmos que empregam dados da interação entre usuário e item, como avaliações ou histórico de consumo, são chamados de Filtragem Colaborativa. Já aqueles que se baseiam no perfil dos usuários, bem como nos metadados dos itens, são conhecidos como Filtragem Baseada em Conteúdo. Há ainda a possibilidade de combinação entre os dois, resultando em um Sistema Híbrido (Aggarwal, 2016).

2.2.1 Filtragem colaborativa (FC)

A ideia básica da FC é aproveitar os dados sobre o comportamento histórico de consumo ou as opiniões de um conjunto de usuários existentes para prever quais itens os usuários atuais provavelmente gostarão ou terão mais interesse no momento. Trata-se da abordagem mais proeminente na geração de recomendações, tendo sido testada e utilizada com sucesso de forma ampla nos negócios, principalmente em sites de varejo (Jannach *et al.*, 2010).

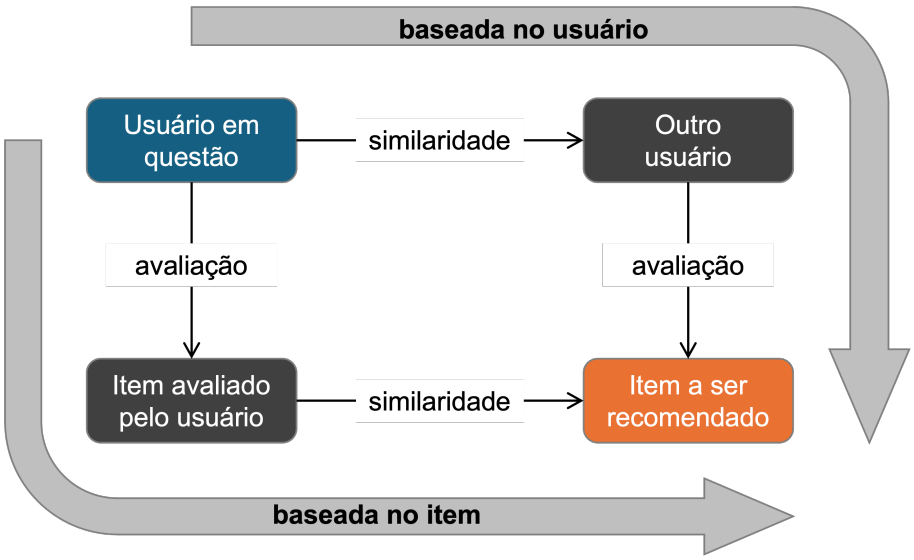
Os dados sobre o comportamento dos usuários podem ser coletados de forma explícita, através de algum sistema em que o usuário possa fornecer seu *feedback* sobre um determinado item (por exemplo, *Likert* de cinco pontos), ou de forma implícita, quando uma ação do cliente, como a aquisição ou a busca por mais detalhes em um site, é interpretada pelo SR como interesse do usuário por aquele item. Apesar do *feedback* explícito fornecer informações mais precisar sobre a opinião do cliente, esta demanda um esforço do usuário para que seja feita, o que pode reduzir a quantidade de dados disponíveis (Jannach *et al.*, 2010).

Além da necessidade de que esses dados de comportamento estejam disponíveis, para ao utilizar a FC assume-se também que as pessoas mantêm suas preferências constantes

ao longo do tempo e que as similaridades passadas entre pessoas se mantêm no futuro (Falk, 2019).

A FC baseada em similaridade pode ser realizada de duas maneiras. Uma delas, chamada de filtragem baseada no usuário, é encontrar os usuários com preferências semelhantes às do usuário em questão e recomendar itens que esses usuários gostaram ou escolheram. A outra, conhecida como filtragem baseada no item, consiste em recomendar itens semelhantes aos que o usuário em questão já gostou ou escolheu. Essas duas formas são mostradas na Figura 1 (Jannach *et al.*, 2010).

Figura 1 – As duas formas de se realizar a FC baseada em similaridade



Fonte: Falk (2019)

A FC pura utiliza como entrada uma matriz de avaliações (ou consumo) usuário x item e tem como resultado: 1) a predição (numérica) do interesse do usuário em relação a um item e 2) lista de N itens recomendados para o usuário, em ordem de prioridade (Jannach *et al.*, 2010).

A principal vantagem da FC é o fato de ser agnóstica ao conteúdo, ou seja, não é necessário esforço para adicionar metadados dos itens ou coletar características dos usuários. Isso faz com que a FC possa ser aplicada a qualquer tipo de item, desde que haja dados suficientes quando as interações usuário-item. Além disso, pela possibilidade de recomendar itens de interesse que o usuário não encontraria por conta própria, aumenta a diversidade de recomendações, adicionando valor para o usuário por causa do uso de um SR (Falk, 2019).

Entretanto, existem algumas limitações para sua aplicação. A esparsidade, um dos maiores problemas, ocorre quando a baixa densidade de dados dificulta a recomendação de itens além dos mais populares. Existe ainda a questão do "início frio", que afeta tanto usuários novos quanto itens novos, quando ainda não há dados para que recebam

recomendações ou sejam recomendados, respectivamente. De forma parecida, temos também as chamadas “ovelhas cinza”, que são usuários com preferências tão incomuns que é difícil encontrar similares. Por fim, como o algoritmo segue as tendências comportamentais dos usuários, os itens mais populares acabam sendo recomendados com mais frequência, o que limita a descoberta de itens menos conhecidos (Falk, 2019).

2.2.2 Filtragem baseada em conteúdo (FBC)

A FBC consiste em um sistema de recomendação que considera o conjunto de características de itens já avaliados positivamente ou consumidos por um usuário para construir um perfil de interesse e, com base nele, são recomendados novos itens para este usuário (Ricci, 2011, p.75). Neste tipo de filtragem, somente as qualificações de itens realizadas pelo próprio usuário já são suficientes para gerar recomendações para ele, ou seja, as avaliações de itens por outros usuários não são necessárias (Aggarwal, 2016).

Dessa forma, para que um SR baseado em FBC possa existir é necessário ter disponível: 1) informações que representem cada item (por exemplo, no caso típico de livros, seria o gênero, o autor, a editora, a resenha, o preço etc.) e 2) perfil de cada usuário que descrevem seus interesses (passados), talvez na forma de atributos de itens preferidos. A partir disso, a tarefa do sistema é determinar os itens que melhor correspondem com às preferências do usuário (Jannach *et al.*, 2010).

Em comparação com a FC, a complexidade FBC é maior porque é necessário extrair conhecimento a partir de conteúdo (por exemplo, atributos, palavras-chave e textos). De forma resumida, o processo de recomendação ocorre em 3 etapas (Falk, 2019; Ricci *et al.*, 2011):

1. **Analisador de conteúdo**: tem a tarefa de transformar os dados sobre cada item em uma representação (por exemplo, um vetor) que possa ser processada por uma máquina. É nesta etapa que o treinamento do modelo acontece. Se a informação disponível é não estruturada (por exemplo, um texto), será preciso fazer algum processamento prévio, o que aumenta ainda mais a complexidade;

2. **Aprendizagem de perfil**: cria o perfil de usuário a partir dos dados representativos de suas preferências.

3. **Buscador de item**: encontra itens relevantes comparando a representação do perfil de usuário com a dos itens que podem ser recomendados.

As principais vantagens em se utilizar a FBC envolvem a facilidade em se adicionar novos itens, pois basta criar o vetor de características para que eles sejam passíveis de recomendação (não há o problema de “início frio” para itens) e a característica de não

ser afetada pela popularidade de itens, já que pode recomendar itens que sequer tiveram avaliações (Falk, 2019).

Porém, há o problema de mal interpretação das preferências do usuário, pois nem toda característica de um item faz com que este seja escolhido por um determinado usuário, bem como a característica que o usuário avalia como positiva pode não estar presente no algoritmo. Ou seja, há uma dependência direta da completude e da qualidade dos metadados dos itens. Além disso, acaba especializando as recomendações para o usuário, impedindo a introdução de itens mais variados que poderiam expandir seus interesses (Falk, 2019).

2.2.3 Sistema de recomendação híbrido (SRH)

Como explicado anteriormente, a FC e a FBC são abordagens distintas, que tem suas vantagens e limitações próprias, além de fazerem uso de fontes de dados de natureza diferentes. Considerando que em muitas situações há disponibilidade tanto de dados relativos à interação usuário-item quanto dos metadados dos itens, faz bastante sentido que se queira utilizar todas as informações disponíveis, bem como o poder algorítmico de mais de um tipo de SR, para entregar recomendações mais robustas. Para explorar essas possibilidades, desenvolvem-se SRHs, que mitigam as limitações dos métodos puros de filtragem e otimizam a relevância e a precisão das recomendações para os usuários (Aggarwal, 2016).

Existem três formas principais de se criar SRHs (Falk, 2019; Jannach *et al.*, 2010; Manzato, 2020):

- **Hibridização Monolítica**: trata-se da um único componente que combina diferentes tipos de filtragem e fontes de informação para melhorar o desempenho (por exemplo, um FBC que utiliza dados de *feedback* de usuários para determinar similaridade entre itens). Esse tipo de hibridização também inclui o enriquecimento das características dos itens com dados externos;
- **Hibridização Paralela**: utiliza mais de um recomendador lado a lado e emprega um mecanismo para agregar os resultados. Pode ser projetado com diversas modelagens, entre elas a mesclada (que combina os resultados dos diferentes recomendadores e os apresenta da forma como estiver definida previamente), a ponderada (que combina os resultados considerando pesos definidos/apreendidos para cada recomendador) e a comutada (que define qual recomendador utilizar em cada situação, como por exemplo: se for usuário novo, usar a FBC, se não, usar a FC), entre outras possibilidades;
- **Hibridização Canalizada**: consiste em estabelecer um processo em que diferentes recomendadores são colocados em sequência, gerando uma única lista final de recomendações. A modelagem pode ser feita de duas formas: “em cascata” (em que os

recomendadores são dispostos em um fluxo em que a saída do anterior é a entrada para o seguinte) e a "meta-nível" (em que um recomendador secundário gera um modelo que é explorado por um principal).

2.3 SR para *feedback* implícito

Ainda que exista uma série de algoritmos desenvolvidos para os casos em que dados de *feedback* explícito estão disponíveis, vemos que na realidade de negócios a maior parte dos dados disponíveis são avaliações de natureza implícita. Além de não demandarem nenhum esforço do usuário para serem gerados, como já mencionado anteriormente, esses dados estão disponíveis em praticamente qualquer sistema de informação, podendo ser obtidos através do rastreamento das interações do usuário, como por exemplo o monitoramento de cliques e tempo de visualização, bem como a efetivação de compra ou contratação de produto e serviços (Rendle *et al.*, 2009).

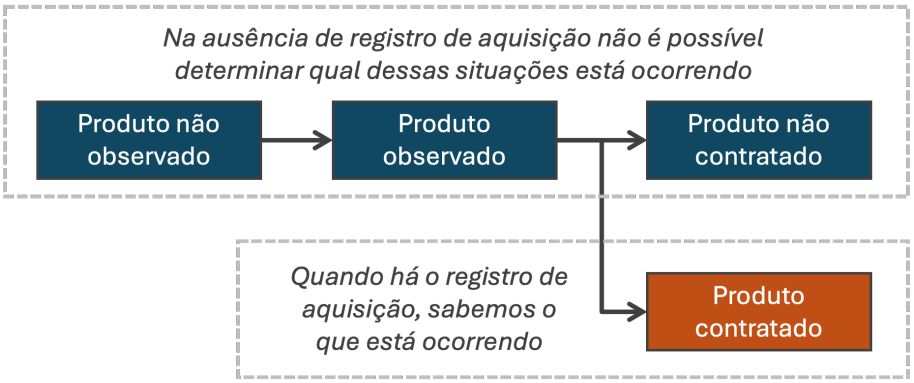
A questão principal a ser tratada quando há disponibilidade apenas de dados decorrentes de avaliações implícitas é que só existem as avaliações positivas, ou seja, o registro das interações dos usuários, que demonstram seu interesse por um determinado item. O fato de não haver avaliações negativas faz com que o algoritmo tenha dificuldades para entender que está fazendo algo "errado" no processo de aprendizagem, assim como saber quando está fazendo algo "certo" (Falk, 2019).

É importante destacar que, nesse cenário, os termos "certo" e "errado" referem-se ao desempenho do algoritmo em recomendar itens de maneira eficaz. "Certo" significa que o item recomendado foi efetivamente adquirido pelo usuário, confirmando a recomendação. Já "errado" se refere à recomendação de um item que o usuário não adquiriu, indicando uma falha na previsão do algoritmo. No entanto, com dados implícitos, essa distinção entre acertos e erros é mais desafiadora, pois o sistema não possui um feedback negativo explícito para ajustar o processo de aprendizado.

Como ilustrado na Figura 2, na ausência de um evento positivo, não é possível determinar se o usuário não sabe que o item existe (não lhe foi apresentado); se ele viu, mas não se interessou; ou se houve interesse, mas ele ainda não comprou. De qualquer forma, podemos assumir que todas essas condições são "piores" do que o registro da aquisição do item pelo usuário (Falk, 2019).

Se tivermos dois itens, um adquirido e o outro não, podemos definir que o item comprado é sempre mais interessante do que o que não foi. Dessa forma, trabalhando por pares, podemos fazer um ranqueamento de itens, em que o primeiro é o mais interessante de todos, seguido pelo segundo mais interessante e assim por diante. Ao estabelecer um ranqueamento personalizado para cada usuário, conseguimos solucionar a questão de recomendação (Falk, 2019).

Figura 2 – Diferentes situações na relação do usuário com o item



Fonte: Falk (2019)

2.3.1 *Bayesian Personalized Ranking* (BPR)

O algoritmo chamado *Bayesian Personalized Ranking* (BPR), apresentado por Steffen Rendle et al. no trabalho intitulado "*BPR: Bayesian Personalized Ranking from Implicit Feedback*", é capaz de gerar uma lista de itens ranqueados para cada usuário, obtida a partir de dados de *feedback* implícito apenas, atendendo assim a necessidade de fornecer recomendações personalizadas para os clientes (Rendle *et al.*, 2009).

Para que o BPR possa funcionar, é preciso observar o atendimento das seguintes propriedades (considerando i, j, k como itens passíveis de recomendação para o usuário u) na ordenação dos itens no ranqueamento (Rendle *et al.*, 2009):

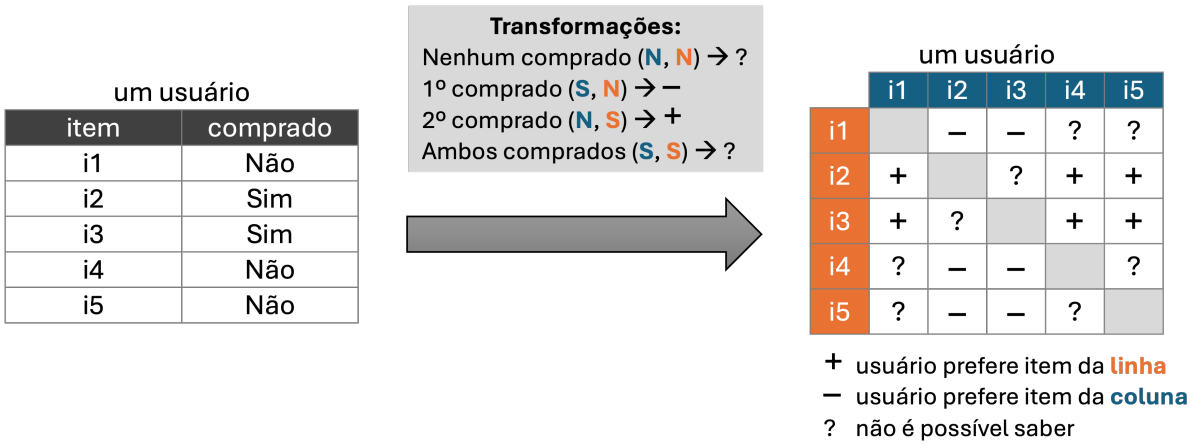
- **Totalidade**: se $i \neq j$, então teremos $i >_u j$ ou $j >_u i$;
- **Antissimetria**: se $i >_u j$ e $j >_u i$, então $i = j$;
- **Transitividade**: se $i >_u j$ ou $j >_u k$, então $i >_u k$

Considerando um conjunto de dados para treinamento que contenham os registros das compras de itens, é possível determinar uma matriz de ordenação para cada usuário, conforme exemplo mostrado na Figura 3.

A abordagem usual para recomendadores é prever a pontuação para o par usuário-item que reflita a preferência do usuário, para então classificar os itens, ordenando os mesmos de acordo com essa pontuação. Se aplicada em um contexto em que só há *feedback* implícito (quando só existem dado positivos), um treinamento de modelo não conseguiria classificar nada, pois iria atribuir a mesma classificação (negativa) para tudo que não fosse positivo (Rendle *et al.*, 2009).

O BPR faz uso de uma abordagem distinta, utilizando pares de dados para treinamento, procurando otimizar a classificação correta desses pares de itens, ao invés de

Figura 3 – Exemplo de transformação dos dados de transação de um usuário em uma matriz de ordenação



Fonte: Falk (2019)

atribuir uma pontuação individual para cada item. Isso representa melhor a situação do que considerar tudo que não for positivo como se fosse negativo (Rendle *et al.*, 2009).

Para encontrar uma classificação personalizada para todos os itens e usuários, o BPR utiliza a estatística bayesiana (como o próprio nome diz). Esse ramo da estatística se baseia na equação abaixo, que diz que a probabilidade do evento A ocorrer, dado que o evento B aconteceu, é igual à probabilidade de B ocorrer quando A acontece, multiplicada pela probabilidade de A ocorrer e divida pela probabilidade de B acontecer.

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)}$$

O problema da classificação pode ser estabelecido como a preferência de ordenação desconhecida para cada usuário, que podemos chamar de $>_u$ para o usuário u. Esta ordenação é total, pois dados quaisquer dois itens, o usuário irá preferir um ou outro.

Considerando θ como a lista de parâmetros de um SR, em um BPR busca-se encontrar o modelo θ com a maior probabilidade de produzir a ordenação perfeita para todos os usuários, ou seja, maximizar $p(\theta| >_u)$. Através da equação de Bayes, sabemos que isso é equivalente a maximizar $p(>_u |\theta) \cdot p(\theta)$, pois são diretamente proporcionais (Falk, 2019).

2.4 Avaliação em Sistemas de Recomendação

Avaliar a qualidade de um SR consiste em mensurar o quão aderente são as recomendações geradas em relação aos interesses e às preferências do usuário. Sem uma avaliação sistemática, é impossível verificar o sucesso de um SR, bem como compará-lo objetivamente com alternativas.

Para realizar essa mensuração, é necessário associar uma métrica quantitativa aos resultados produzidos pelo SR em questão, fornecendo uma avaliação da performance do SR em relação a referências previamente estabelecidas. Essa avaliação pode ser feita de três diferentes formas (Aggarwal, 2016; Manzato, 2020):

- **Estudos com usuários**: envolve recrutar usuários reais para, em um ambiente controlado, observar a interação deles com o SR, bem como coletar feedbacks. A vantagem é capturar diretamente as percepções do usuário sobre diferentes aspectos do SR. Entretanto, como o usuário sabe que está sendo observado, pode enviesar sua avaliação, além do que geralmente a quantidade de usuários é bem pouco representativa em comparação com o conjunto total;
- **Métodos on-line**: consiste em capturar a opinião dos usuários de forma remota, permitindo ampliar muito a quantidade de participantes. Este método está menos suscetível aos efeitos de vieses, pois o usuário opera o sistema normalmente, sem saber que está fazendo uma avaliação. Um exemplo típico são os testes A/B, em que se compara a taxa de conversão de dois algoritmos diferentes;
- **Métodos off-line com dados históricos**: considera os dados históricos das interações já efetuadas. É a forma mais comum de se avaliar SRs, pois são experimentos mais fáceis de serem reproduzidos e permitem avaliar diversas métricas.

Vale observar que, na avaliação *off-line*, é de suma importância dividir os conjuntos de treinamento, validação e teste de forma a não enviesar os resultados. A divisão desses conjuntos pode ser feita de várias maneiras e deve ser adequada a cada caso específico. Por exemplo, em se tratando de dados temporais, a forma mais indicada é a chamada *hold-out*, que consiste em fazer uma única divisão, separando as interações mais recentes como conjunto de testes (que não pode ser usado para ajuste do modelo) e as demais para treinamento e validação (Manzato, 2020).

2.4.1 Métricas de Avaliação

Existem diversas métricas para avaliação um SR. A mais conhecida e fundamental é a acurácia, entretanto outras métricas, como cobertura, confiança, novidade, serendipidade, diversidade, robustez /estabilidade e escalabilidade, que podem ser utilizadas em complemento para um entendimento mais profundo da performance, de acordo com o interesse específico do caso (Aggarwal, 2016).

2.4.2 Acurácia para predição de notas de avaliação

Mensura a precisão da predição de notas de avaliação para cada par usuário-item e é aferida utilizando-se o conjunto de teste.

Considerando que r_{uj} é a nota explícita real do usuário u para o item j e que $\hat{r}_{uj}$ é a predição desta nota dada pelo algoritmo de recomendação, o erro de predição (e_{uj}) é expresso por (Aggarwal, 2016):

$$e_{uj} = \hat{r}_{uj} - r_{uj}$$

Dado que E é o conjunto teste de avaliações, temos as seguintes métricas utilizadas para se calcular o erro total:

Erro médio quadrado (*mean squared error – MSE*):

$$MSE = \frac{\sum_{(u,j)\in E} e_{uj}^2}{|E|}$$

Erro quadrático médio quadrado (*root mean squared error – RMSE*):

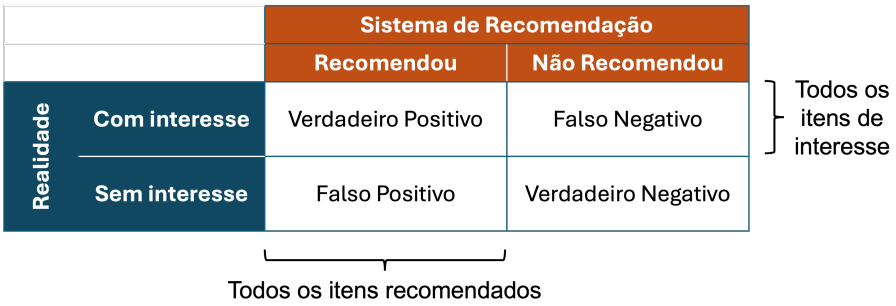
$$RMSE = \sqrt{\frac{\sum_{(u,j)\in E} e_{uj}^2}{|E|}}$$

Uma observação relevante é que nenhuma dessas métricas consegue mensurar o impacto da melhoria da recomendação para o usuário, por isso deve-se buscar a combinação com avaliações realizadas diretamente com o usuário.

2.4.3 Acurácia para predição de ranqueamento

Mede a precisão da predição do ranqueamento dos k itens recomendados para cada usuário, utilizando o conjunto de teste, e é utilizada principalmente nos casos em que há disponibilidade somente de avaliações implícitas. Considerando as recomendações geradas pelo SR e o interesse real do usuário, temos as seguintes combinações possíveis, mostrada na Figura 4 (Falk, 2019):

Figura 4 – Matriz de confusão entre a recomendação gerada e a realidade



Fonte: Falk (2019)

- **Verdadeiro positivo** (*True positive* - TP): item recomendado e de interesse do usuário.

- **Falso positivo** (*False positive* - FP): item foi recomendado, mas não é de interesse do usuário.
- **Falso negativo** (*False negative* - FN): item não foi recomendado, mas era de interesse do usuário.
- **Verdadeiro Negativo** (*True negative* - TN): item não foi nem recomendado e nem era de interesse do usuário.

Dessa forma, é possível definir duas diferentes métricas:

- **Precisão**: fração de itens recomendados que são de interesse do usuário, ou seja, quanto o SR acertou de tudo que ele recomendou.

$$Precisão = \frac{TP}{TP + FP}$$

- **Revocação**: fração de itens de interesse do usuário que foram recomendados, ou seja, quanto o SR acertou de tudo que é de interesse do usuário.

$$Revocação = \frac{TP}{TP + FN}$$

No contexto de SR que utiliza ranqueamento, geralmente o que importa é a quantidade de itens relevantes apresentados no topo das recomendações. Com isso, muitas vezes busca-se otimizar a precisão sem dar tanta atenção se o usuário está recebendo todos os itens de interesse possíveis (Revocação). A métrica que faz essa mensuração é conhecida como $P@k$ (*Precision at k*) e é calculada pela quantidade de itens de interesse nas k primeiras posições em relação aos primeiros k itens no ranking de recomendações (Falk, 2019):

$$P@k(u) = \frac{\#\ \{itens\ de\ interesse\ nas\ k\ primeiras\ posições\}}{k}$$

A métrica mais comum para SR com avaliações implícitas é a Média das Precisões Médias (*mean average precision* - MAP). A Precisão Média (*average precision* - AP) avalia a qualidade do ranqueamento ao considerar a posição dos itens de interesse no topo da lista de recomendações. Considerando a P@k para valores de 1 a m (em que m é o número de itens recomendados), a AP do usuário u é dada por (Falk, 2019):

$$AP\,(u) = \frac{\sum_{k=1}^{m} P@k(u)}{m}$$

Para avaliação do SR, obtém-se a MAP através da média das APs de todos os usuários:

$$MAP = \frac{\sum_{u \in U} AP(u)}{|U|}$$

2.4.4 Demais métricas

Para mensuração de outros aspectos de um SR utilizam-se as seguintes métricas (Manzato, 2020):

- **Cobertura**: é fração de usuários para os quais pelo menos N predições podem ser calculadas (em que N é o número de listados no ranking). Deve ser reportada em conjunto com a acurácia, para garantir que o sistema não esteja gerando recomendações aleatórias;
- **Confiança**: indica o quanto o usuário confia nas recomendações geradas. A melhor maneira de mensurar é através de métodos on-line;
- **Novidade**: capacidade do sistema recomendar itens que o usuário não conhece, mas que são de interesse. Isso aumenta a percepção de valor que o usuário atribui a um SR, melhorando a taxa de conversão;
- **Serendipidade**: é semelhante a métrica de novidade, mas considerando itens que surpreendem o usuário em recomendações bem sucedidas, ou seja, são recomendações não óbvias. Da mesma forma, adiciona valor para o usuário no uso do SR;
- **Diversidade**: mede a variedade entre si dos itens de uma mesma lista de recomendações. Maior diversidade pode trazer mais novidade ou serendipidade;
- **Robustez/estabilidade**: diz o quanto um sistema é resistente a ataques de hackers ou avaliações falsas de usuários maliciosos;
- **Escalabilidade**: avalia a performance do sistema ao se trabalhar com grandes bases de dados, mensurando itens como tempo de processamento em treinamentos e tempo de resposta na geração de recomendações.

3 DESENVOLVIMENTO

Para criação de um SR de produtos financeiros precisamos dispor de dados de clientes e produtos contratados em uma instituição financeira. Entretanto, em se tratando de dados pessoais e sensíveis, amparados pela LGPD e pelo sigilo bancário, não será possível ter acesso à dados reais dos clientes de um banco. Ainda que esses dados fossem anonimizados, de forma a não permitir a identificação das pessoas, a exposição de uma base de dados que permitisse caracterizar o perfil de uma carteira de clientes e seus produtos contratados poderia ser considerada uma violação de segredo comercial, caracterizada na legislação vigente.

A alternativa que encontramos foi utilizar o conjunto de dados (*dataset*) disponibilizado por ocasião de uma competição promovida em 2016 por um banco espanhol no site Kaggle. Na página de publicação há um disclamer que deixa claro que este *dataset* não contém dados de nenhum cliente real, bem como não é representativo da base de clientes do banco em questão (Risdal; Piedra; Kan, 2016). A descrição deste *dataset* será aprofundada a seguir.

Considerando o *dataset* selecionado, devemos encontrar um algoritmo que possa gerar recomendações relevantes aos usuários com base nesses dados, que contém seus comportamentos anteriores relativos à contratação de produtos. Como veremos a seguir, esses dados se caracterizam como *feedbacks* implícitos. Sendo assim, conforme já exposto anteriormente, o algoritmo que se adequa bem a essa característica é o *Bayesian Personalized Ranking* (*BPR*), por isso é o escolhido para o nosso SR.

3.1 Conjunto de dados (*Dataset*)

Na página do Kaggle referente à competição promovida em 2016 encontramos o arquivo contendo o *dataset* a ser utilizado (Risdal; Piedra; Kan, 2016).

Neste *dataset*, temos dados referentes ao comportamento mensal de consumo de produtos bancários de aproximadamente 960 mil clientes ao longo de um período de 17 meses (entre janeiro/2015 e maio/2016). Em resumo, são 23 atributos com dados sobre as características de cada cliente e mais 24 atributos de produtos, referentes a situação no último dia do mês ("1" quando tiver o produto contratado e "0" quando não possuir).

Os metadados dos atributos de partição, características do cliente e produtos contratados são listados nas Tabela 1, Tabela 2 e Tabela 3, respectivamente (descrição em tradução livre do autor):

Tabela 1 – Metadados do atributo de partição

#	Atributo	Descrição
0	fecha_dato	Partição pela data (mês e ano)

Tabela 2 – Metadados dos atributos das características do cliente

#	Atributo	Descrição
1	ncodpers	Código identificador do cliente
2	ind_empleado	Indicador de vínculo empregatício com o banco
3	pais_residencia	País de residência do cliente
4	sexo	Sexo informado pelo cliente
5	age	Idade do cliente
6	fecha_alta	Data de abertura da conta/relacionamento com o banco
7	ind_nuevo	Indicador de cliente novo (até 6 meses de relacionamento)
8	antiguedad	Tempo como cliente (em meses)
9	indrel	Indicador do tipo de cliente
10	ult_fec_cli_1t	Data mais recente como cliente principal
11	indrel_1mes	Indicador do tipo de cliente no início do mês
12	tiprel_1mes	Indicador do relacionamento no início do mês
13	indresi	Indicador de residência no mesmo país do banco
14	indext	Indicador de nascimento em um país diferente do banco
15	conyuemp	Indicador de cônjuge com vínculo empregatício com o banco
16	canal_entrada	Canal de aquisição do cliente
17	indfall	Indicador de falecido
18	tipodom	Tipo de endereço
19	cod_prov	Código da província
20	nomprov	Nome da província
21	ind_actividad_cliente	Indicador de cliente ativo
22	renta	Renda do cliente
23	segmento	Segmento do cliente

Tabela 3 – Metadados dos atributos indicadores de produto contratado

#	Atributo	Descrição
24	ind_ahor_fin_ult1	Conta investimento
25	ind_aval_fin_ult1	Conta garantia
26	ind_cco_fin_ult1	Conta corrente
27	ind_cder_fin_ult1	Conta derivada
28	ind_cno_fin_ult1	Conta salário
29	ind_ctju_fin_ult1	Conta júnior
30	ind_ctma_fin_ult1	Conta mais especial
31	ind_ctop_fin_ult1	Conta especial
32	ind_ctpp_fin_ult1	Conta especial plus
33	ind_deco_fin_ult1	Investimentos de curto prazo
34	ind_deme_fin_ult1	Investimentos de médio prazo
35	ind_dela_fin_ult1	Investimentos de longo prazo
36	ind_ecue_fin_ult1	Conta digital
37	ind_fond_fin_ult1	Investimentos em fundos
38	ind_hip_fin_ult1	Crédito imobiliário
39	ind_plan_fin_ult1	Plano de previdência
40	ind_pres_fin_ult1	Crédito pessoal
41	ind_reca_fin_ult1	Pagamentos de impostos
42	ind_tjcr_fin_ult1	Cartão de crédito
43	ind_valo_fin_ult1	Investimentos em ações
44	ind_viv_fin_ult1	Conta moradia
45	ind_nomina_ult1	Recebimento de salário
46	ind_nom_pens_ult1	Pensionista
47	ind_recibo_ult1	Pagamentos por débito direto

3.2 Algoritmo de Recomendação

Existem diversas bibliotecas de algoritmos para sistema de recomendação com o BPR incluído, entre elas a Implicit (https://benfred.github.io/implicit/index.html), a LightFM (https://making.lyst.com/lightfm/docs/home.html) e a Cornac (https://cornac.readthedocs.io/en/stable/index.html).

Entretanto, escolhemos usar a CaseRecommender (https://github.com/caserec/CaseRecommender/wiki) por ser de fácil instalação e utilização. Trata-se de uma biblioteca em linguagem Python que fornece uma ampla variedade de algoritmos, entre eles o BPR, além de inclui as métricas de avaliação para predição de ranqueamento (Costa; Manzato, 2016).

Os requisitos para uso são simples: a instalação é feita por meio de uma única linha de comando e pode ser executada dentro do Google Colab. A entrada de dados é feita através de um arquivo de texto com apenas 3 colunas:

- u_id: identificação do usuário, em formato integer;

- $i_i d$: identificação do item, em formato integer;
- *rating*: *feedback* do usuário em relação ao item, em formato floating.

Para uma análise comparativa dos resultados do BPR, selecionamos do CaseRecommender um outro algoritmo de FC, o *ItemKNN*, que utiliza a similaridade entre k itens vizinhos para fazer recomendações, utilizando também *feedback* implícito para isto. Além desse, também usaremos as recomendações geradas aleatoriamente, disponível na biblioteca indicando *Random* como algoritmo.

Para avaliação dos algoritmos selecionados (*BPR*, *ItemKNN* e *Random*), o CaseRecommender dispõe das seguintes métricas precisão em N (*Prec@N*), revocação em N (*Recall@N*) e média das precisões médias (*Map@N*).

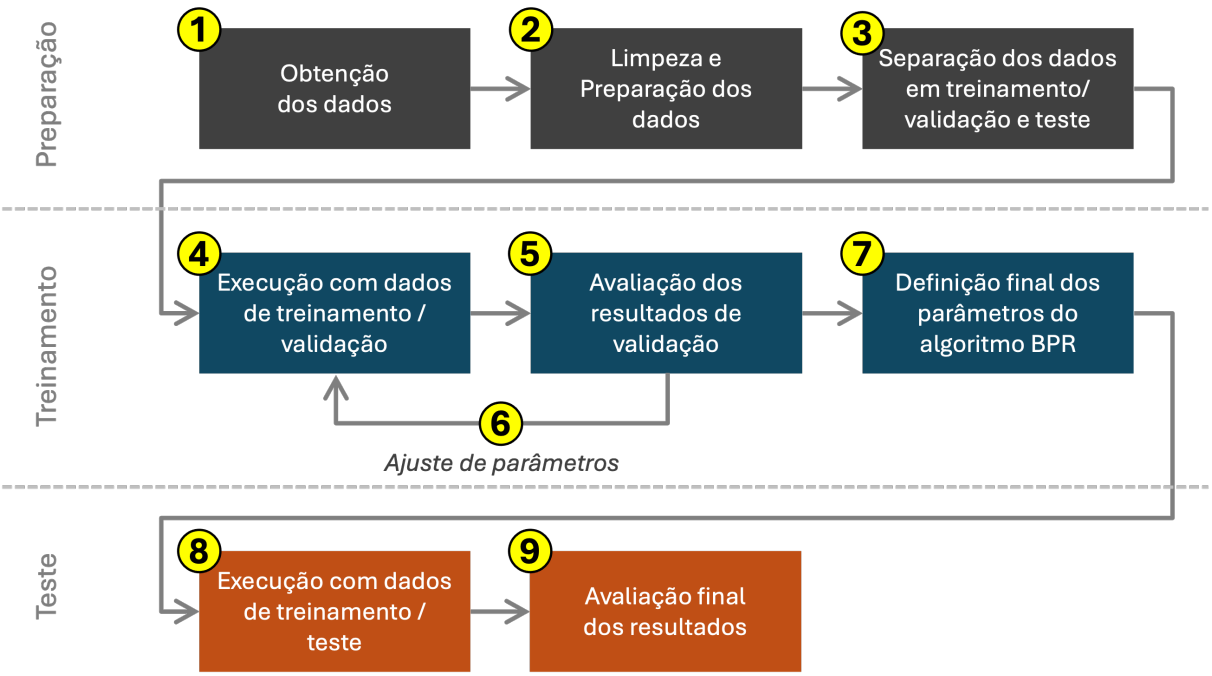
Os parâmetros de ajuste para algoritmo BPR no CaseRecommender são os seguintes:

- *factors*: Número de fatores latentes por usuário/item;
- *learn_rate*: Taxa de aprendizagem (alfa);
- *epochs*: Número de épocas de treinamento;
- *batch_size*: Reduz o número de interações em cada época; se for 0, usa o número de interações positivas no conjunto de treinamento;
- *rank_length*: Tamanho do ranking que deve ser gerado pelas previsões do algoritmo de recomendação;
- *init_mean*: Média da distribuição normal usada para inicializar os fatores latentes;
- *init_stdev*: Desvio padrão da distribuição normal usada para inicializar os fatores latentes;
- *reg_u*: Parâmetro de regularização para os fatores dos usuários;
- *reg_i*: Parâmetro de regularização para os fatores dos itens positivos;
- *reg_j*: Parâmetro de regularização para os fatores dos itens negativos;
- *reg_bias*: Parâmetro de regularização para o termo de viés (bias);
- *random_seed*: Semente aleatória;
- *items_test*: Se for verdadeiro, atualiza o conjunto não observado de cada usuário com amostras no conjunto de teste;

3.3 Metodologia

As macro etapas para a construção do SR são mostradas na Figura 5 e descritas a seguir.

Figura 5 – Macro etapas de desenvolvimento



1. **Obtenção dos dados**: consiste em acessar o site, baixar o arquivo com os dados e armazená-lo em local que possa ser acessado por ferramenta para execução de código em linguagem Python. Utilizamos o Google Drive para armazenamento e o Google Colab como ambiente de execução;

2. **Limpeza e Preparação dos dados**: nesta etapa carregamos os dados no ambiente de execução, substituímos os valores nulos por zeros, harmonizamos os tipos dos dados e eliminamos os atributos que não são necessários para execução dos algoritmos *BPR*, *ItemKNN* e *Random*. Além disso, alteramos a disposição dos dados para o formato requerido (usuário, item e *feedback*) Por fim, eliminamos os registros de *feedbacks* iguais a zero, pois devemos manter somente os *feedbacks* positivos (iguais a um) para adequar o arquivo de entrada ao formato exigido pela biblioteca;

3. **Separação dos dados em treinamento/validação e teste**: em se tratando de dados temporais, a divisão é feita na forma *hold-out*, em que as interações do mês mais recentes constituem o conjunto de teste. No restante, separamos o mês mais recente para validação e os demais para treinamento. A Figura 6 ilustra como a divisão é realizada.

Figura 6 – Separação dos dados em treinamento/validação e teste

4. **Execução com dados de treinamento/validação**: subimos os dados de treinamento e de validação para execução dos algoritmos e registramos os resultados, os parâmetros utilizados e eventuais mensagens de erro/alerta;

5. **Avaliação dos resultados de validação**: analisamos os resultados obtidos para as métricas *Prec@N*, *Recall@N* e *Map@N* do algoritmo *BPR*, comparando-os tanto com os resultados dos ciclos anteriores quanto com as saídas dos algoritmos *ItemKNN* e *Random*;

6. **Ajuste de parâmetros**: criamos uma hipótese de configuração de parâmetros do algoritmo *BPR* para uma melhora gradativa do resultado e/ou correção de erros, retornando a etapa 4;

7. **Definição final dos parâmetros do algoritmo BPR**: após diversos ciclos de experimentação nas etapas 4, 5 e 6, chegamos a configuração final de parâmetros, que será utilizada com o conjunto de teste.

8. **Execução com dados de treinamento/teste**: subimos os dados de treinamento e teste para execução dos algoritmos e registramos os resultados;

9. **Avaliação final dos resultados**: analisamos os resultados obtidos para as métricas *Prec@N*, *Recall@N* e *Map@N* dos algoritmos *BPR*, *ItemKNN* e *Random*.

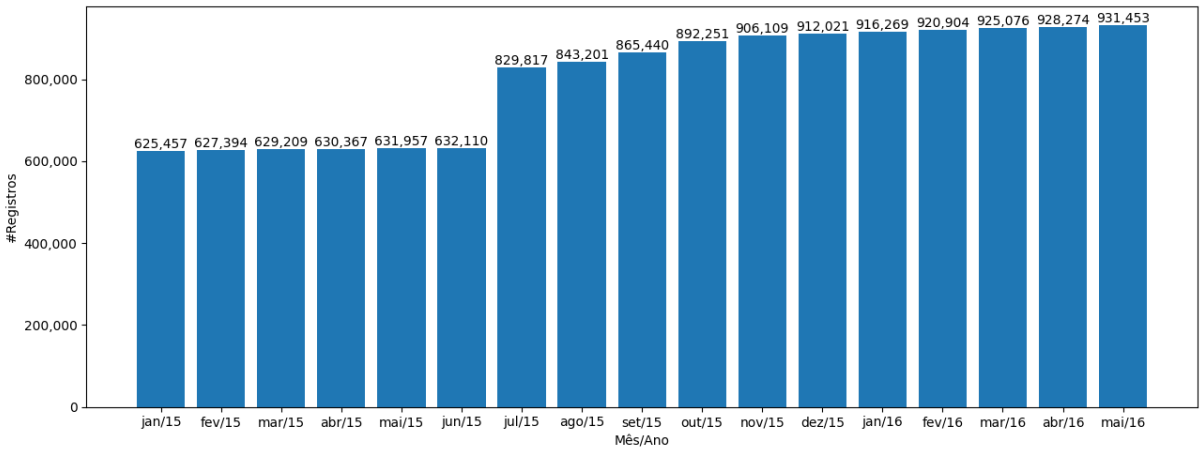
4 RESULTADOS

4.1 Exploração do Dataset

Após a etapa de limpeza e preparação, mantivemos apenas os dados descritos nas Tabela 1 e Tabela 3, bem como apenas o Código identificador do cliente (*ncodpers*) da Tabela 2, pois os algoritmos de recomendação utilizados são de FC e não fazem uso dos metadados dos clientes.

Na Figura 7 podemos ver a evolução da quantidade de registros, ou seja, clientes, por mês. No período de jan-jun/15, a quantidade se mantém em torno de 630 mil registros. Em jul/15, essa quantidade salta para aproximadamente 830 mil e segue aumentando gradativamente, até atingir pouco mais de 930 mil em mai/16.

Figura 7 – Quantidade de clientes por mês/ano



Em termos de clientes distintos, são 956.645 em todo dataset. Ao eliminarmos os clientes que não contém nenhum registro de produto contratado ao longo de todos os meses, esse número cai para 731.162 clientes. Após separar a base em treino / validação / teste, temos em número de clientes distintos:

- Treino: 724.392
- Validação: 695.044
- Teste: 696.539
- em comum nas 3 bases: 688.947

Ao analisarmos os dados dos produtos contratados, em que temos apenas 24 distintos, podemos verificar nas Figura 8, Figura 9 e Figura 10 que a distribuição % dos

produtos nas 3 bases é praticamente a mesma. Além disso, fica evidente que há uma concentração muito alta no produto conta corrente (cta_corrente), acima de 42-45% (em função dessa concentração, para melhor visualização do valor dos demais produtos, optamos por utilizar os gráficos em escala logarítmica).

Figura 8 – Distribuição % de produto contratado - dados de treino

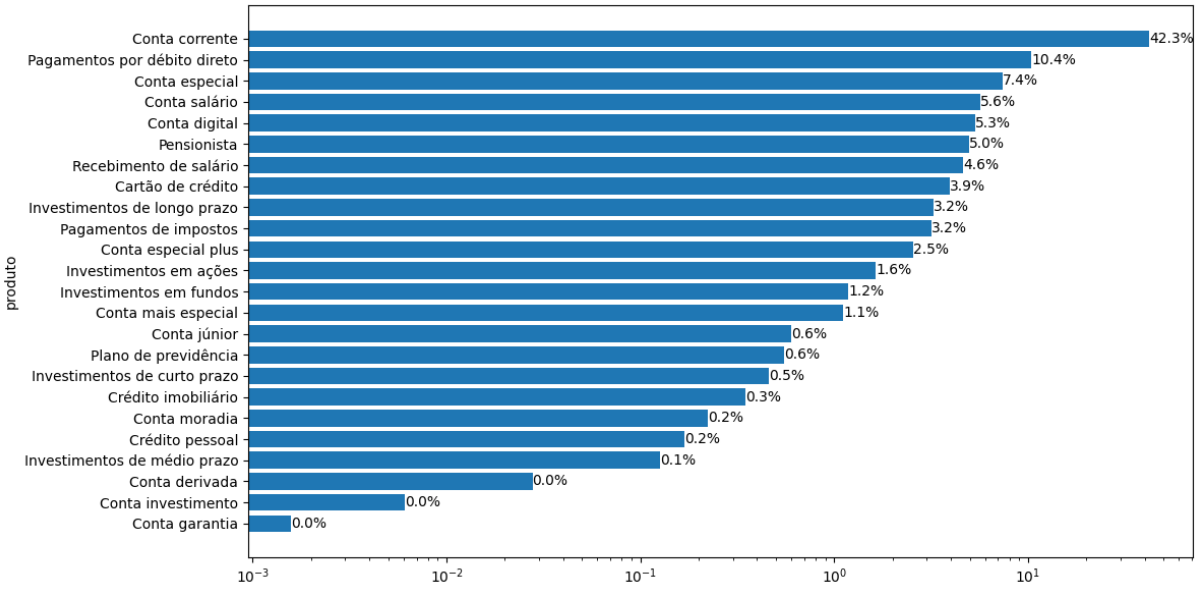


Figura 9 – Distribuição % de produto contratado - dados de validação

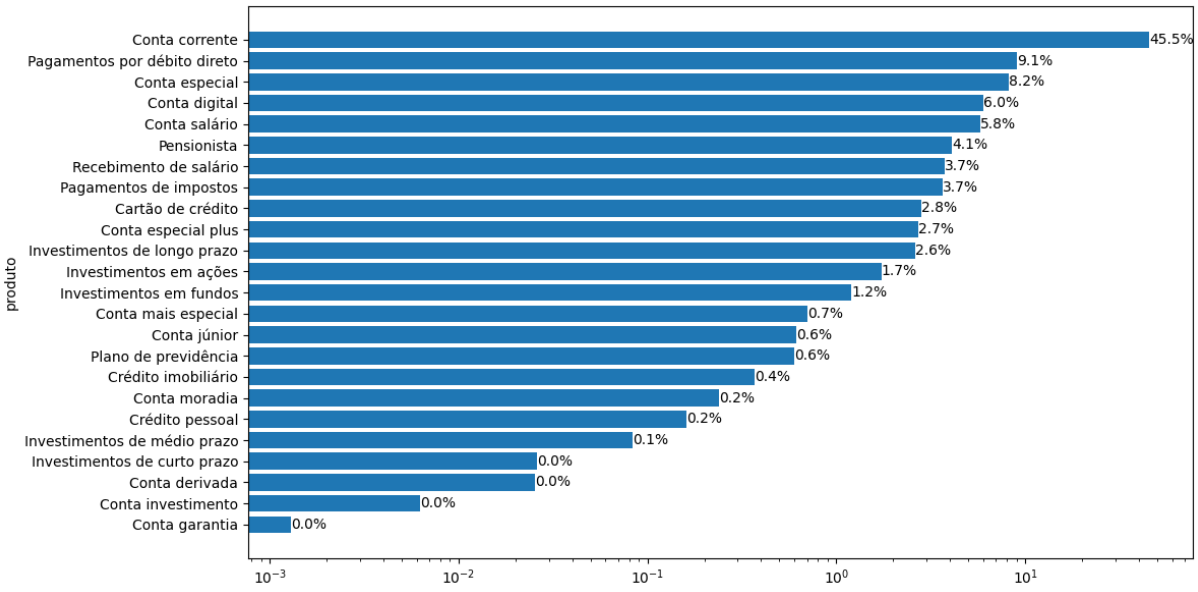
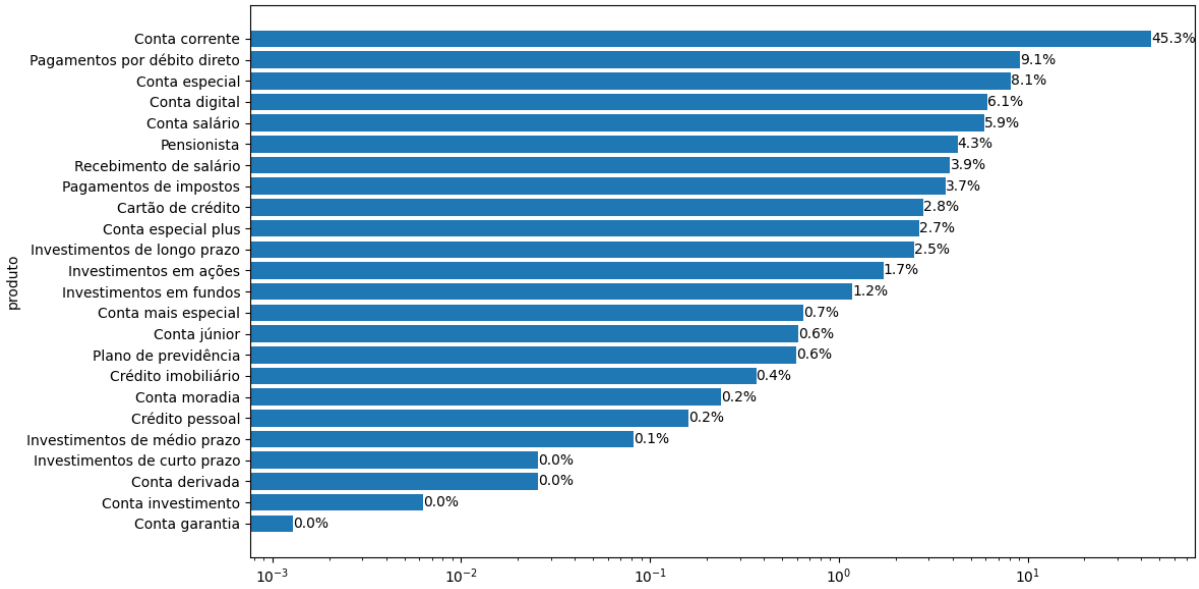


Figura 10 – Distribuição % de produto contratado - dados de teste



4.2 Ciclos de execução, análise de resultados e ajuste de parâmetros

Iniciando os ciclos de execução, rodamos pela primeira vez os algoritmos de recomendação, considerando os parâmetros *default* para o *BPR*, e obtivemos os resultados mostrados na Figura 11:

Figura 11 – Resultados - parâmetros default do algoritmo *BPR*

	Precisão				Revocação				Média das precisões médias			
	Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF (default)	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
ItemKNN	0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random	0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

Entretanto, na saída do algoritmo *BPR* recebemos a seguinte mensagem de alerta: "*RuntimeWarning: overflow encountered in scalar subtract*", indicando que houve um "*overflow*"(estouro) gerado pelo resultado de uma operação exceder o limite que pode ser representado pelo tipo de dado utilizado, provavelmente por ser extremamente grande.

Para evitar a ocorrência de "*overflow*", procedemos a alteração dos parâmetros de regularização para os fatores, tanto dos itens positivos (*reg_i*) quanto dos negativos (*reg_j*), avaliando o impacto nos resultados.

Para o parâmetro de regularização para os fatores dos itens positivos (*reg_i*), vemos na Figura 12 que o *overflow* permanece para valores abaixo de 0.0155. A partir desse valor, o *overflow* deixa de ocorrer e os resultados apresentados melhoram significativamente.

Da mesma forma, para o parâmetro de regularização para os fatores dos itens negativos (*reg_j*), vemos na Figura 13 que o valor mínimo deve ser 0.00795 para que não

Figura 12 – Resultados - teste do parâmetro de regularização para os fatores dos itens positivos (*reg_i*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	reg_i=0.0025 (default)	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
	reg_i=0.0154	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
	reg_i=0.0155	0,005036	0,003021	0,002330	0,001523	0,002304	0,003746	0,004539	0,005737	0,005036	0,006625	0,006996	0,007216
	reg_i=0.0171	0,004972	0,003016	0,002334	0,001524	0,002273	0,003740	0,004544	0,005742	0,004972	0,006587	0,006964	0,007186
	reg_i=0.0186	0,004903	0,003000	0,002339	0,001524	0,002242	0,003719	0,004551	0,005743	0,004903	0,006535	0,006924	0,007146
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

ocorra *overflow*.

Figura 13 – Resultados - teste do parâmetro de regularização para os fatores dos itens negativos (*reg_j*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	reg_j=0.00025 (default)	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
	reg_j=0.00794	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
	reg_j=0.00795	0,005653	0,003133	0,002269	0,001466	0,002446	0,003768	0,004402	0,005554	0,005653	0,006967	0,007210	0,007420
	reg_j=0.00875	0,005615	0,003131	0,002271	0,001465	0,002429	0,003765	0,004405	0,005552	0,005615	0,006945	0,007188	0,007400
	reg_j=0.00954	0,005538	0,003119	0,002274	0,001470	0,002397	0,003769	0,004422	0,005573	0,005538	0,006912	0,007157	0,007371
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

Para prosseguir com os testes de parâmetro, decidimos aplicar um incremento de 10% sobre os valores mínimos para *reg_i* e *reg_j*, de forma a garantirmos uma margem para que não ocorra *overflow* na fase de teste e no uso do algoritmo para geração de recomendações futuras. Assim, fixamos *reg_i*=0.0171 e *reg_j*=0.00875 e verificamos o resultado combinado na Figura 14.

Figura 14 – Resultados - fixação dos parâmetros de regularização para os fatores dos itens positivos (*reg_i*) e negativos (*reg_j*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	reg_i=0.0025, reg_j=0.00025 (default)	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
	reg_i=0.0171, reg_j=0.00875	0,004355	0,002939	0,002265	0,001516	0,001989	0,003663	0,004440	0,005708	0,004355	0,006185	0,006575	0,006812
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

Superada a questão de *overflow*, passamos a testar alterações nos demais parâmetros.

Para o número de fatores latentes por usuário/item (*factors*), variando os valores em torno do default, obtivemos os resultados que constam na Figura 15. Dessa forma, fixamos o parâmetro no valor que apresentou o melhor resultado: *factors*=9.

Em seguida, passamos a testar alterações na taxa de aprendizagem (*learn_rate*). A Figura 16 mostra que os resultados obtidos melhoram conforme este parâmetro é aumentado, até atingir o *overflow* com *learn_rate*=0.12.

A exceção fica por conta das métricas de precisão e revocação para recomendações de *rankings* de 10 itens (@10), que se comportam de forma inversa, ou seja, em que

Figura 15 – Resultados - teste do número de fatores latentes por usuário/item (*factors*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	factors=6	0,003536	0,002908	0,002226	0,001500	0,001519	0,003674	0,004403	0,005667	0,003536	0,005727	0,006117	0,006350
	factors=8	0,004440	0,002946	0,002269	0,001525	0,002031	0,003697	0,004454	0,005740	0,004440	0,006224	0,006612	0,006842
	factors=9	0,004479	0,002952	0,002314	0,001522	0,002050	0,003701	0,004519	0,005725	0,004479	0,006246	0,006653	0,006874
	factors=10 (default)	0,004355	0,002939	0,002265	0,001516	0,001989	0,003663	0,004440	0,005708	0,004355	0,006185	0,006575	0,006812
	factors=11	0,004427	0,002944	0,002276	0,001521	0,002014	0,003679	0,004462	0,005726	0,004427	0,006231	0,006626	0,006863
	factors=12	0,004426	0,002944	0,002276	0,001522	0,002014	0,003661	0,004462	0,005735	0,004426	0,006226	0,006623	0,006857
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

Figura 16 – Resultados - teste da taxa de aprendizagem (*learn_rate*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	learn_rate=0.05 (default)	0,004479	0,002952	0,002314	0,001522	0,002050	0,003701	0,004519	0,005725	0,004479	0,006246	0,006653	0,006874
	learn_rate=0.06	0,004725	0,002998	0,002293	0,001519	0,002143	0,003740	0,004483	0,005721	0,004725	0,006440	0,006814	0,007049
	learn_rate=0.07	0,004942	0,003031	0,002308	0,001517	0,002231	0,003761	0,004503	0,005720	0,004942	0,006588	0,006952	0,007184
	learn_rate=0.08	0,005162	0,003063	0,002328	0,001516	0,002340	0,003786	0,004536	0,005717	0,005162	0,006735	0,007081	0,007307
	learn_rate=0.09	0,005338	0,003093	0,002344	0,001512	0,002419	0,003799	0,004555	0,005709	0,005338	0,006845	0,007180	0,007399
	learn_rate=0.10	0,005510	0,003113	0,002344	0,001508	0,002502	0,003806	0,004553	0,005699	0,005510	0,006941	0,007263	0,007476
	learn_rate=0.11	0,005636	0,003132	0,002343	0,001507	0,002541	0,003823	0,004548	0,005699	0,005636	0,007016	0,007320	0,007533
	learn_rate=0.12	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

os valores vão piorando a medida que se aumenta o parâmetro. Também é interessante observar que isso não ocorre para a média das precisões médias (*MAP@10*). Uma explicação possível é que como o *MAP@10* avalia a qualidade geral da ordenação dos itens relevantes ao longo do *ranking*, acaba sendo mais estável e menos suscetível a flutuações específicas na posição dos itens, especialmente nas posições finais, considerando um *ranking* de 10 itens.

Para tentarmos evitar a ocorrência de *overflow* na fase de teste e no uso posterior do algoritmo, fixamos o parâmetro em um valor intermediário: *learn_rate*=0.08.

Por fim, buscamos o ajuste do número de épocas de treinamento (*epochs*). Na Figura 17 podemos verificar que não há uma linearidade no comportamento das métricas frente ao aumento gradativo do parâmetro, convergindo para os melhores resultados em diversos valores. Dessa forma, ficaremos com o máximo obtido para o *MAP*, ou seja, *epochs*=35.

Concluindo a fase de treinamento, definimos os valores finais para os parâmetros do algoritmo *BPR*, mostrados na Tabela 4.

Tabela 4 – Parâmetros finais otimizados para o algoritmo BPR

Parâmetro	Valor *default*	Valor otimizado
reg_i	0.0025	0.0171
reg_j	0.00025	0.00875
factors	10	9
learn_rate	0.05	0.08
epochs	30	35

Figura 17 – Resultados - teste do número de épocas de treinamento(*epochs*)

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	epochs=25	0,004844	0,003007	0,002258	0,001488	0,002176	0,003733	0,004467	0,005638	0,004844	0,006538	0,006875	0,007048
	epochs=28	0,005026	0,002969	0,002202	0,001507	0,002286	0,003696	0,004390	0,005706	0,005026	0,006572	0,006913	0,007151
	epochs=29	0,004889	0,003001	0,002226	0,001489	0,002215	0,003761	0,004435	0,005643	0,004889	0,006546	0,006889	0,007120
	epochs=30	0,005162	0,003063	0,002328	0,001516	0,002340	0,003786	0,004536	0,005717	0,005162	0,006735	0,007081	0,007307
	epochs=31	0,004805	0,003013	0,002229	0,001493	0,002156	0,003743	0,004439	0,005671	0,004805	0,006534	0,006846	0,007017
	epochs=32	0,002988	0,002972	0,002235	0,001507	0,001291	0,003716	0,004401	0,005699	0,002988	0,005567	0,005936	0,006222
	epochs=33	0,004469	0,002977	0,002191	0,001486	0,002010	0,003720	0,004346	0,005632	0,004469	0,006315	0,006638	0,006888
	epochs=34	0,005148	0,003027	0,002228	0,001510	0,002299	0,003751	0,004437	0,005710	0,005148	0,006716	0,007020	0,007194
	epochs=35	0,005451	0,003061	0,002244	0,001483	0,002437	0,003783	0,004475	0,005636	0,005451	0,006906	0,007194	0,007355
	epochs=36	0,004847	0,002984	0,002234	0,001509	0,002194	0,003695	0,004385	0,005699	0,004847	0,006475	0,006806	0,007071
	epochs=37	0,005069	0,003076	0,002322	0,001514	0,002273	0,003792	0,004532	0,005722	0,005069	0,006686	0,007016	0,007237
	epochs=38	0,003583	0,002954	0,002199	0,001490	0,001585	0,003673	0,004346	0,005639	0,003583	0,005829	0,006162	0,006424
	epochs=39	0,004339	0,003027	0,002257	0,001490	0,001925	0,003753	0,004482	0,005658	0,004339	0,006292	0,006618	0,006820
	epochs=40	0,005165	0,003066	0,002297	0,001500	0,002324	0,003771	0,004518	0,005674	0,005165	0,006718	0,007028	0,007207
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

Na Figura 18 podemos comparar os resultados obtidos inicialmente, com os parâmetros *default* com os atingidos após a otimização dos parâmetros para o algoritmo *BPR*, verificando valores superiores em todas as métricas, também em comparação com o algoritmo *ItemKNN* e o ranqueamento aleatório (*Random*).

Figura 18 – Treinamento - Resultados antes e após o ajuste dos parâmetros

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	(default)	0,000906	0,000678	0,001474	0,001499	0,000242	0,000654	0,002828	0,005536	0,000906	0,001376	0,002521	0,003286
BPRMF	reg_i=0.0171, reg_j=0.00875, factors=9, learn_rate=0.08, epochs=35	0,005451	0,003061	0,002244	0,001483	0,002437	0,003783	0,004475	0,005636	0,005451	0,006906	0,007194	0,007355
ItemKNN		0,002608	0,002090	0,001410	0,000941	0,000770	0,002028	0,002306	0,003456	0,002608	0,003462	0,003610	0,003889
Random		0,000780	0,000753	0,000750	0,000748	0,000297	0,000859	0,001427	0,002855	0,000780	0,001369	0,001662	0,002021

4.3 Execução com dados de teste e análise de resultados

Passando para a fase de teste, configuramos os parâmetros para o algoritmo *BPR* definidos na fase de treinamento para otimização dos resultados. Também carregamos os dados de treinamento e de teste, sendo que este último corresponde ao mês mais recente do dataset original (*mai/16*).

Os resultados atingidos estão expostos na Figura 19, em que podemos verificar que a performance do algoritmo *BPR* foi bastante superior ao algoritmo *ItemKNN* em todas as métricas, principalmente na média das precisões médias (*MAP*), que alcançou valores da ordem de 100% melhores que o outro algoritmo.

Figura 19 – Teste - Resultados antes e após o ajuste dos parâmetros

		Precisão				Revocação				Média das precisões médias			
		Prec@1	Prec@3	Prec@5	Prec@10	Recall@1	Recall@3	Recall@5	Recall@10	MAP@1	MAP@3	MAP@5	MAP@10
BPRMF	reg_i=0.0171, reg_j=0.00875, factors=9, learn_rate=0.08, epochs=35	0,008940	0,005211	0,003893	0,002557	0,003936	0,006410	0,007749	0,009792	0,008940	0,011477	0,011952	0,012118
ItemKNN		0,004330	0,003427	0,002337	0,001645	0,001234	0,003146	0,003643	0,006091	0,004330	0,005592	0,005841	0,006373
Random		0,001259	0,001277	0,001274	0,001296	0,000472	0,001440	0,002404	0,004906	0,001259	0,002257	0,002760	0,003366
BPRMF x ItemKNN		106%	52%	67%	55%	219%	104%	113%	61%	106%	105%	105%	90%
BPRMF x Random		610%	308%	206%	97%	734%	345%	222%	100%	610%	409%	333%	260%

5 CONCLUSÃO

5.1 Considerações Finais

Este trabalho de conclusão de curso teve como objetivo desenvolver um sistema de recomendação de produtos financeiros voltado para clientes pessoa física de um banco, utilizando dados históricos de consumo para prever quais produtos seriam de maior interesse para cada cliente, ou seja, aqueles com maior probabilidade de contratação. Ao longo do desenvolvimento, foram exploradas as etapas essenciais e os desafios envolvidos na criação de um sistema capaz de fornecer recomendações personalizadas de forma eficiente.

O algoritmo selecionado para o sistema foi o *Bayesian Personalized Ranking* (BPR), reconhecido por sua eficácia na geração de recomendações ranqueadas com base em *feedback* implícito. Essa característica é amplamente encontrada nos dados disponíveis em ambientes de negócios reais, o que torna esse sistema de recomendação altamente aplicável em situações do cotidiano empresarial.

Os resultados obtidos após ajuste dos parâmetros do algoritmo demonstraram que o BPR pode fornecer recomendações satisfatórias, superando o algoritmo *ItemKNN* e, em muito, a geração aleatória.

A análise dos resultados com o conjunto de dados de teste validou a capacidade do sistema identificar os produtos de maior interesse para cada cliente. A média das precisões médias (MAP) mostrou-se uma métrica robusta para avaliar a qualidade das recomendações geradas, confirmando a eficácia do sistema desenvolvido.

Em resumo, o sistema desenvolvido neste trabalho apresenta um potencial significativo para melhorar a eficiência das ofertas de produtos financeiros, contribuindo para a satisfação do cliente, a redução de custos e o aumento da receita das instituições bancárias. A contínua evolução dos sistemas de recomendação, especialmente com a adoção de novas técnicas de inteligência artificial e aprendizado de máquina, sem dúvida impulsionará ainda mais a capacidade de personalização e a experiência dos usuários no setor financeiro.

5.2 Limitações do trabalho

Tendo em vista tudo o que foi desenvolvido, é crucial destacar algumas limitações encontradas neste trabalho.

Em primeiro lugar, o modelo foi treinado utilizando um *dataset* fictício proveniente de uma competição realizada no conhecido site Kaggle. Embora esse conjunto de dados tenha sido adequado para experimentação e ajuste do algoritmo, ele pode não capturar totalmente as nuances e complexidades dos dados reais de comportamento de consumo

dos clientes de um banco.

Além disso, os produtos financeiros considerados pertencem ao contexto de mercado da Espanha, que difere significativamente da realidade encontrada no Brasil, tanto pela existência de produtos específicos em cada país quanto pelas diferentes importâncias relativas atribuídas a esses produtos.

No que diz respeito ao conjunto de dados de treinamento, que abrange um período de 15 meses, não foram consideradas as particularidades do momento exato (mês) em que o cliente contratou um produto, nem se ocorreram movimentos de contratação e cancelamento do produto ao longo desse período. Essas variáveis podem ter uma influência nas recomendações geradas pelo modelo, mas não foram exploradas neste trabalho.

Por fim, este sistema de recomendação se restringiu à abordagem de filtragem colaborativa (FC), utilizando exclusivamente dados de *feedback* implícito. Dessa forma, os dados das características dos clientes presentes no *dataset* não foram explorados, embora sua inclusão pudesse enriquecer o modelo e provavelmente melhorar a qualidade das recomendações. No entanto, incorporar essas informações teria aumentado significativamente a complexidade e o tempo de desenvolvimento do sistema, exigindo a implementação de filtragem baseada em conteúdo (FBC) e transformando o modelo em um sistema de recomendação híbrido (SRH). Dado o tempo disponível para a conclusão deste curso, não haveria tempo hábil para realizar esse desenvolvimento adicional de forma adequada.

5.3 Trabalhos Futuros

Com base nos aprendizados obtidos durante o desenvolvimento deste trabalho, destacamos algumas perspectivas de evolução que podem ser implementadas para aprimorar o modelo, alinhadas ao objetivo proposto:

- **Utilização de dados reais**: aplicar o modelo a dados reais de comportamento de consumo de clientes de um banco brasileiro, retreinando o algoritmo e comparando seu desempenho. Além disso, se possível, estender o período dos dados de treinamento para pelo menos três anos, a fim de identificar e analisar possíveis efeitos de sazonalidade;
- **Clientes pessoa jurídica**: ampliar a aplicação do modelo considerando também empresas que são clientes de um banco, verificando-se as particularidades dos clientes e do portfólio de produtos destinado a esse público, de forma que as recomendações geradas sejam aderentes às suas jornadas de consumo;
- **Desenvolvimento de um sistema de recomendação híbrido**: evoluir o sistema de recomendação para incorporar dados das características dos clientes e dos produtos, adotando uma abordagem híbrida que combine as vantagens da filtragem colaborativa

(FC) e da filtragem baseada em conteúdo (FBC). Essa evolução pode, entre outros benefícios, ajudar a superar as dificuldades na geração de recomendações para novos clientes ou novos produtos introduzidos no portfólio do banco;

- **Incorporação da temporalidade na contratação de produtos**: integrar informações sobre o momento de contratação e cancelamento de produtos no treinamento do modelo, enriquecendo a compreensão do comportamento do cliente e melhorando a precisão das recomendações.

REFERÊNCIAS

AGGARWAL, C. C. **Recommender Systems**. Springer Cham, 2016. ISBN 978-3-319-29659-3. Disponível em: https://link.springer.com/book/10.1007/978-3-319-29659-3.

BOUVIER, A. **Recommender Systems and Applications in Banking**. 2021. Disponível em: https://medium.com/genifyai/recommender-systems-and-applications-in-banking-f0cef8f87fa6. Acesso em: 06 jan. 2024.

COSTA, A. F. da; MANZATO, M. G. Case recommender: A recommender framework. *In*: SBC. **Anais Estendidos do XXII Simpósio Brasileiro de Sistemas Multimídia e Web**. 2016. p. 99–102. Disponível em: https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/4891/4797.

DROZDOV, A. **Recommender Systems for Banking and Financial Services: How AI is Transforming the Way We Manage Our Finances**. 2023. Disponível em: https://yellow.systems/blog/recommender-systems-for-banking-and-financial-services. Acesso em: 06 jan. 2024.

FALK, K. **Practical Recommender Systems**. Manning Publications Co., 2019. ISBN 9781617292705. Disponível em: https://manning.com/books/practical-recommender-systems.

JANNACH, D. *et al.* **Recommender Systems An Introduction**. Cambridge University Press, 2010. ISBN 978-0-521-49336-9. Disponível em: https://www.cambridge.org/9780521493369https://www.cambridge.org/core/books/recommender-systems/C6471B59388D8A9F684C49C198691B53.

MANZATO, M. **Sistemas de Recomendação**. 2020. Disponível em: https://youtube.com/playlist?list=PLih3eXOECaZZxbYKN5OEPQAmgefS6YzF0. Acesso em: 09 mar. 2024.

RENDLE, S. *et al.* Bpr: Bayesian personalized ranking from implicit feedback. *In*: **UAI '09: 25 Conference on Uncertainty in Artificial Intelligence**. [*S.l.: s.n.*], 2009. p. 452–461. Disponível em: https://arxiv.org/pdf/1205.2618.

RICCI, F. *et al.* **Recommender Systems Handbook**. Springer US, 2011. ISBN 978-0-387-85819-7. Disponível em: https://link.springer.com/book/10.1007/978-0-387-85820-3.

RISDAL, M.; PIEDRA, M.; KAN, W. **Santander Product Recommendation**. 2016. Disponível em: https://www.kaggle.com/competitions/santander-product-recommendation/. Acesso em: 19 jan. 2024.