

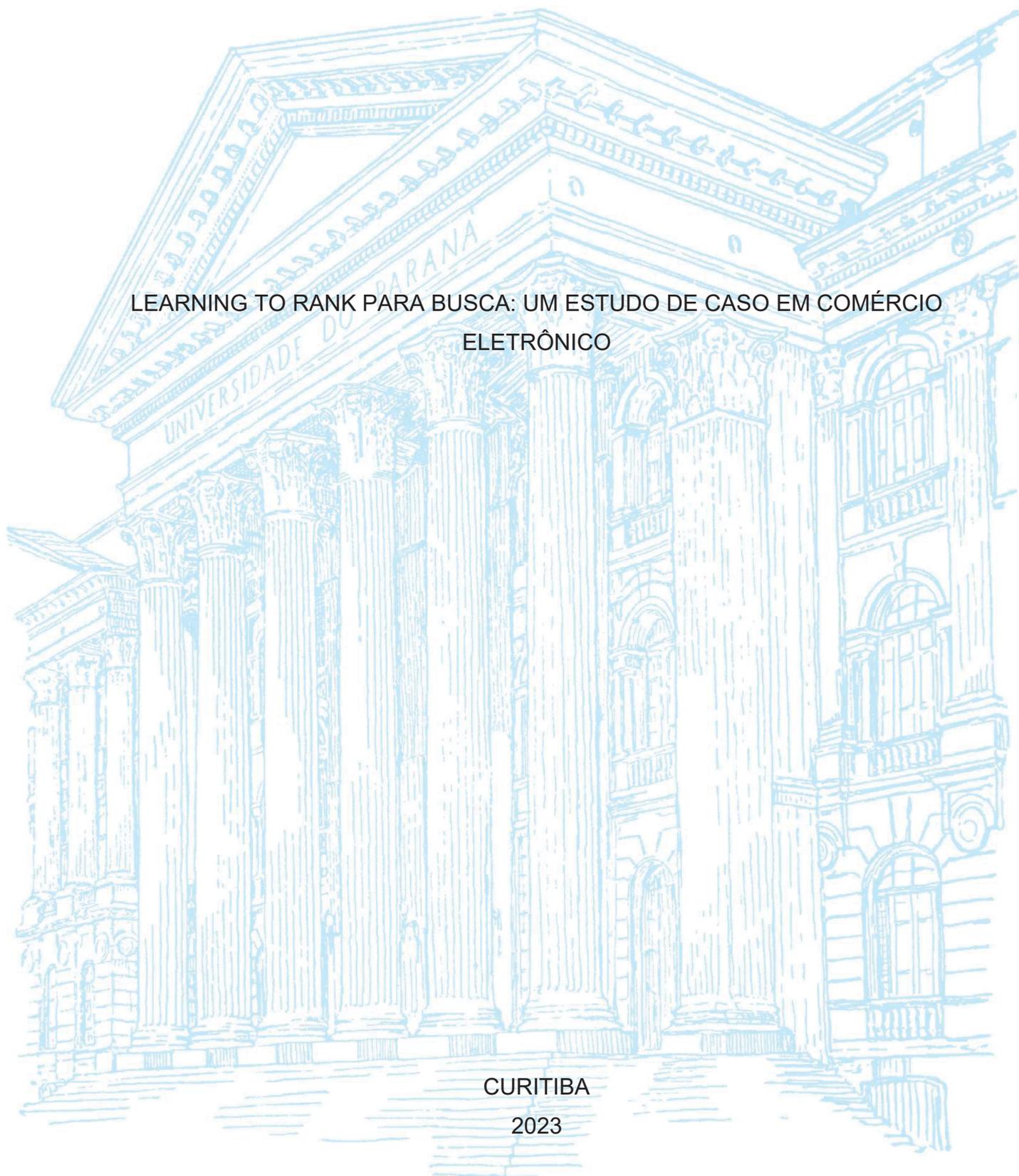
UNIVERSIDADE FEDERAL DO PARANÁ

CAIO TOMY ONO KAMINARI

LEARNING TO RANK PARA BUSCA: UM ESTUDO DE CASO EM COMÉRCIO
ELETRÔNICO

CURITIBA

2023



CAIO TOMY ONO KAMINARI

LEARNING TO RANK PARA BUSCA: UM ESTUDO DE CASO EM COMÉRCIO
ELETRÔNICO

Dissertação apresentada ao Programa de Pós-Graduação em Gestão de Organizações, Liderança e Decisão da Universidade Federal do Paraná, Setor de Ciências Sociais Aplicadas, como requisito para obtenção do título de Mestre em Gestão de Organizações, Liderança e Decisão.

Área de Concentração: Operações e Sistemas de Decisão

Orientador: Prof. Dr. Cassius Tadeu Scarpin

CURITIBA

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIAS SOCIAIS APLICADAS

Kaminari, Caio Tomy Ono

Learning to rank para busca : um estudo de caso em comércio eletrônico / Caio Tomy Ono Kaminari. – Curitiba, 2023.

1 recurso on-line : PDF.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Ciências Sociais Aplicadas, Programa de Pós-Graduação em Gestão de Organizações, Liderança e Decisão.

Orientador: Prof. Dr. Cassius Tadeu Scarpin.

1. Comércio eletrônico. 2. Recuperação da informação. 3. Aprendizado do computador. I. Scarpin, Cassius Tadeu. II. Universidade Federal do Paraná. Programa de Pós-Graduação em Gestão de Organizações, Liderança e Decisão. III. Título.

Bibliotecária: Maria Lidiane Herculano Graciosa CRB-9/2008



TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação GESTÃO DE ORGANIZAÇÕES, LIDERANÇA E DECISÃO da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **CAIO TOMY ONO KAMINARI** intitulada: **LEARNING TO RANK PARA BUSCA: UM ESTUDO DE CASO EM COMÉRCIO ELETRÔNICO**, sob orientação do Prof. Dr. CASSIUS TADEU SCARPIN, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 18 de Dezembro de 2023.

Assinatura Eletrônica
20/12/2023 13:38:10.0
CASSIUS TADEU SCARPIN
Presidente da Banca Examinadora

Assinatura Eletrônica
20/12/2023 14:27:55.0
GUSTAVO VALENTIM LOCH
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
20/12/2023 14:08:58.0
JOSÉ EDUARDO PÉCORÁ JUNIOR
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
21/12/2023 10:23:26.0
CELSO YOSHIKAZU ISHIDA
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

Dedico este trabalho à minha esposa Priscila e nossa filha Aimi.

RESUMO

Esta pesquisa apresenta técnicas de aprendizado de máquina (ML) para a ordenação de resultados de busca utilizadas em um comércio eletrônico. O intuito é melhorar a jornada do usuário após uso do sistema de busca da plataforma quando melhorada pelo algoritmo de ML proposto. O modelo LambdaMART está enquadrado na literatura das áreas de Recuperação de Informação e *Learning to Rank*. Para estimativa de relevância, utilizou-se as alternativas de taxa de clique na busca, taxa de adição no carrinho e taxa de conversão de pedido a partir de dados reais de usuários da plataforma. Os resultados obtidos mostram que o modelo LambdaMART forneceu resultados superiores a 95% nos experimentos comparado ao método atual do comércio eletrônico testado quando usado a métrica de avaliação Ganho Cumulativo Descontado Normalizado (NDCG).

Palavras-chave: Comércio eletrônico; busca; *learning to rank*.

ABSTRACT

This research presents machine learning (ML) techniques for sorting search results used in e-commerce. The objective is to improve the user journey after using the platform's search system when improved by the proposed ML algorithm. The LambdaMART's model is framed in the literature in the areas of Information Retrieval and Learning to Rank. To estimate relevance, we use search click-through rate, add to cart rate and order conversion rate as alternatives based on real data from users of the platform. The results obtained show that LambdaMART's model presented results greater than 95% in experiments compared to the current e-commerce method tested when using the Normalized Discounted Cumulative Gain (NDCG) evaluation metric.

Keywords: E-commerce; search; learning to rank.

LISTA DE FIGURAS

FIGURA 1 - FLUXOGRAMA DE UM USUÁRIO NO SISTEMA DE BUSCA.....	12
FIGURA 2 - FLUXOGRAMA DE USUÁRIO ACESSANDO O COMÉRCIO ELETRÔNICO.....	16
FIGURA 3 - FLUXOGRAMA DO CÁLCULO DO SCORE	17
FIGURA 4 - FLUXOGRAMA DO CÁLCULO DA ESTIMATIVA DE RELEVÂNCIA....	26
FIGURA 5 - FLUXOGRAMA DO TRATAMENTO NAS BASES DE DADOS.....	30
FIGURA 6 - FLUXOGRAMA DOS EXPERIMENTOS E RESULTADOS	31

LISTA DE TABELAS

TABELA 1 - EXPERIMENTOS COM ALTERAÇÃO DOS PESOS DAS VARIÁVEIS	28
TABELA 2 - EXPERIMENTOS COM ALTERAÇÃO DO NÚMERO DE VARIÁVEIS	29
TABELA 3 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE CONVERSÃO DE PEDIDOS	32
TABELA 4 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE CONVERSÃO DE PEDIDOS	32
TABELA 5 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE CONVERSÃO DE PEDIDOS	33
TABELA 6 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE CLIQUES NA BUSCA.....	34
TABELA 7 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE CLIQUES NA BUSCA.....	34
TABELA 8 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE CLIQUES NA BUSCA.....	35
TABELA 9 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE ADIÇÃO NO CARRINHO	35
TABELA 10 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE ADIÇÃO NO CARRINHO.....	36
TABELA 11 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE ADIÇÃO NO CARRINHO	37
TABELA 12 - RESULTADOS CONSOLIDADOS POR EXPERIMENTOS	37

LISTA DE ABREVIATURAS OU SIGLAS

CTR	- <i>Click-through rate</i>
DCG	- <i>Discounted cumulative gain</i>
DPSR	- <i>Deep personalized and semantic retrieval</i>
GMV	- <i>Gross merchandise volume</i>
LTR	- <i>Learning to rank</i>
NDCG	- <i>Normalized discounted cumulative gain</i>
RI	- Recuperação de Informação
TCP	- Taxa de conversão de pedidos
TAC	- Taxa de adição no carrinho
TCB	- Taxa de cliques na busca

SUMÁRIO

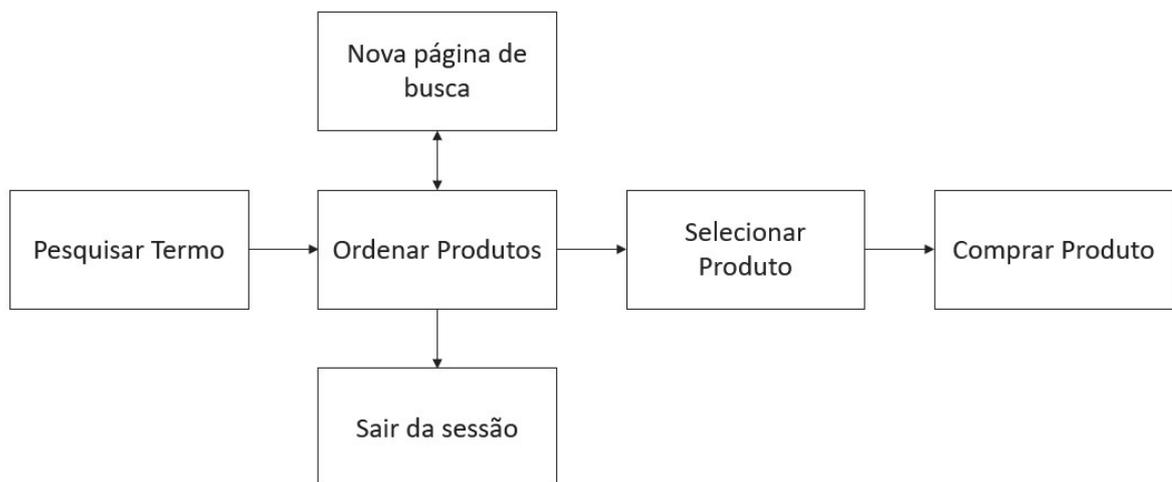
1 INTRODUÇÃO	12
1.1 OBJETIVOS	14
1.1.1 Objetivo geral	14
1.1.2 Objetivos específicos.....	14
1.2 JUSTIFICATIVA	14
1.3 LIMITAÇÕES DO TRABALHO	15
1.4 ESTRUTURA DO TRABALHO.....	15
2 DESCRIÇÃO DO PROBLEMA	16
2.1 ESTUDO DE CASO	16
2.2 SISTEMA DE BUSCA INTERNO	17
3 REVISÃO DA LITERATURA	19
3.1 RECUPERAÇÃO DE INFORMAÇÃO	19
3.2 LEARNING TO RANK	22
4 IMPLEMENTAÇÕES	26
4.1 ESTIMATIVA DE RELEVÂNCIA	26
4.2 MÉTODO DE APRENDIZAGEM DE MÁQUINA	27
4.3 EXPERIMENTOS.....	28
4.4 BASES DE DADOS.....	29
5 RESULTADOS	31
5.1 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE CONVERSÃO DE PEDIDOS 32	
5.2 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE CLIQUES NA BUSCA	33
5.3 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE ADIÇÃO NO CARRINHO ...	35
5.4 RESULTADOS CONSOLIDADOS POR EXPERIMENTOS	37
6 CONCLUSÕES	39
6.1 SUGESTÕES PARA TRABALHOS FUTUROS	39
REFERÊNCIAS	41
APÊNDICE 1 – EXEMPLO DE CÁLCULO DO NDCG	43
APÊNDICE 2 – EXEMPLO DE CÁLCULO DAS ESTIMATIVAS DE RELEVÂNCIA	45
APÊNDICE 3 – EXEMPLO DE APLICAÇÃO DO ALGORITMO LAMBDA MART ...	48

1 INTRODUÇÃO

No ano de 2020 as vendas pela internet representaram 17,8% das vendas no setor de varejo em todo o mundo, e a participação do comércio eletrônico deve saltar para 24,5% até 2025, marcando um aumento de 37,6% em apenas cinco anos (RITA; RAMOS, 2022). Grandes varejistas brasileiros como Casas Bahia, Magazine Luiza e Americanas, desenvolvem soluções em suas plataformas digitais para engajar os clientes com objetivo de aumentar a receita de vendas; por isso, um dos maiores desafios para o comércio eletrônico é de como melhorar a ordenação de produtos após algum usuário interagir com o sistema de busca. Está consolidado na literatura acadêmica e é conhecido pelos varejistas que grande parte dos clientes visualizam produtos ordenados nas primeiras posições após acionado o sistema de busca em comparação com produtos ordenados na parte inferior da página (FERREIRA et al., 2022).

Conforme a Figura 1 abaixo, nota-se o fluxograma de um usuário usando o sistema de busca de um comércio eletrônico.

FIGURA 1 - FLUXOGRAMA DE UM USUÁRIO NO SISTEMA DE BUSCA



FONTE: O autor (2023).

Uma sessão de pesquisa entre um usuário e o sistema de busca é organizado por etapas. Primeiro o usuário insere uma consulta, ou seja, digita uma palavra no campo de busca e submete ao sistema. Depois, o sistema de busca ordena os produtos relacionados à consulta e exibe os produtos principais da primeira página.

Então o usuário faz algumas operações como por exemplo, selecionar produtos para ver mais detalhes, usar filtros laterais específicos, selecionar outros critérios de ordenação ou apenas solicitar uma nova página da mesma consulta. Quando uma nova página é solicitada, o sistema de busca reclassifica o restante dos produtos e exibe os principais. Essas quatro etapas serão repetidas até que o usuário compre alguns produtos ou simplesmente saia da sessão de pesquisa. Sendo assim, uma transação bem-sucedida sempre envolve várias rodadas deste processo (HU et al., 2018).

O usuário recebe uma lista com os produtos mais relevantes de acordo com o termo buscado e a regra de ordenação; portanto, o consumidor analisa os produtos e pode selecionar àqueles com maior potencial para finalizar a compra. Nem sempre o que é mais relevante para a empresa é o que o usuário busca, ou seja, existem atributos de produto mais importantes para o usuário do que a regra imposta pela empresa, como por exemplo, preço, marca ou avaliações (ZHUANG et al., 2018). O usuário geralmente não navega por todos os produtos relevantes retornados pelo sistema de busca e pode abandonar a página se houver apenas alguns ou nenhum produto relevante encontrado nos principais resultados. O abandono da pesquisa prejudica as plataformas de comércio eletrônico porque o modelo de negócio depende de transações de compra bem-sucedidas em vez de apenas cliques em anúncios (GOSWAMI et al., 2019).

O cliente que compra no comércio eletrônico, faixa etária entre 25 a 44 anos, não tem uma boa ideia do sortimento ou do estilo dos produtos oferecidos e pode não ter em mente todas as variações de um mesmo produto que esteja procurando. O cliente forma uma primeira impressão sobre o sortimento e o estilo da página de busca após consultar os produtos nas primeiras ordenações (FERREIRA et al., 2022); portanto, quanto mais tempo o usuário interage com os produtos listados, tem-se um indicador que pode revelar que o índice principal de busca está atraente. Isto explica que empresas, como a Amazon, investem nesta área há anos; utiliza-se inteligência artificial na ordenação de produtos do sistema de busca em sua plataforma (SOROKINA; CANTU-PAZ, 2016).

Portanto, para este trabalho é necessário responder o seguinte problema de pesquisa:

O uso de aprendizado de máquina pode funcionar para ordenação de listagens de busca de um comércio eletrônico de uma empresa do segmento de bens

para o lar em comparação com o método atual de ordenação agregando maior valor financeiro?

1.1 OBJETIVOS

Esta seção é destinada para a exposição dos objetivos deste estudo.

1.1.1 Objetivo geral

Propor uma metodologia de solução, baseada em um modelo de aprendizado de máquina (ML), do sistema de busca de um comércio eletrônico, em um caso real, visando melhorar a jornada do usuário.

1.1.2 Objetivos específicos

Os objetivos específicos do trabalho são:

- Analisar a adequação de um programa computacional de um modelo de aprendizado de máquina a partir da revisão da literatura;
- Obter os dados reais para serem utilizados no estudo de caso;
- Construir o programa computacional a partir do modelo de aprendizado de máquina escolhido;
- Propor uma metodologia computacional para comparar as soluções do método de ordenação de busca atual da empresa do segmento de bens para o lar e o método de ordenação de busca do modelo de aprendizado de máquina escolhido;
- Avaliar os resultados de ordenação do modelo de aprendizado de máquina escolhido em comparação aos resultados de ordenação de busca atual da empresa do segmento de bens para o lar definindo uma métrica padronizada.

1.2 JUSTIFICATIVA

A justificativa do trabalho é apresentar um caso prático de uso de técnicas de aprendizado de máquina aplicado a um sistema de busca de uma plataforma de comércio eletrônico no segmento de bens para o lar com potencial geração de maior receita financeira frente a outras empresas de mesmo segmento no mercado.

1.3 LIMITAÇÕES DO TRABALHO

Este trabalho aplica modelos de aprendizado de máquina que se limitam a dados utilizados em uma plataforma de comércio eletrônico no segmento de bens para o lar.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em capítulos dos quais os principais assuntos abordados nestes, são:

- Capítulo 2: apresenta o ambiente onde o problema real a ser solucionado surge e descreve a principal ordenação de produtos do sistema de busca da empresa;
- Capítulo 3: apresenta a revisão da literatura com conceitos e trabalhos relacionados ao tema proposto e a principal métrica para avaliar modelos de recuperação de informação;
- Capítulo 4: apresenta o desenvolvimento completo da metodologia proposta, explicando todas as etapas necessárias abordadas neste trabalho;
- Capítulo 5: apresenta os resultados computacionais dos experimentos deste trabalho;
- Capítulo 6: apresenta as conclusões deste trabalho e sugestões para trabalhos futuros.

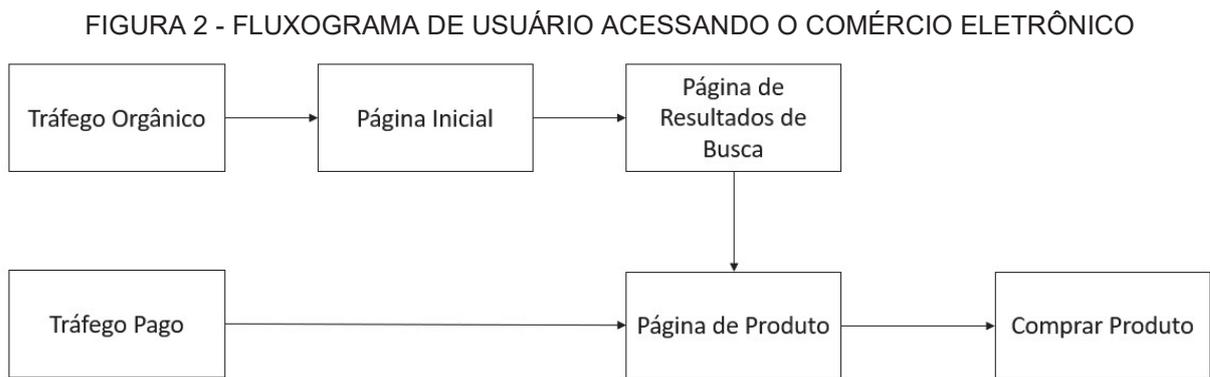
2 DESCRIÇÃO DO PROBLEMA

Neste capítulo será apresentado os principais conceitos necessários para o entendimento do problema abordado, bem como o ambiente onde este surge.

2.1 ESTUDO DE CASO

A empresa é uma das líderes do comércio eletrônico no segmento de bens para o lar no Brasil, possui mais de 2 milhões de produtos a venda e recebe mais de 20 milhões de acessos nos canais de vendas por mês.

Conforme Figura 2 abaixo, nota-se o fluxograma de um usuário acessando o comércio eletrônico.



FONTE: O autor (2023).

O tráfego orgânico diz respeito à entrada de usuários na plataforma de maneira espontânea sem a necessidade de investimentos por clique. O usuário que acessa a plataforma via tráfego orgânico entra na página principal do comércio eletrônico, denominada *homepage*, e necessita navegar na plataforma para buscar produtos que deseja comprar. Já o tráfego pago diz respeito à entrada de usuários na plataforma com investimentos direto em canais de publicidade, como por exemplo, anúncios na página de busca do Google. O usuário que acessa a plataforma via tráfego pago é direcionado a uma página de produto para finalizar sua compra, ou seja, a busca do produto já foi realizada no canal onde o produto foi anunciado.

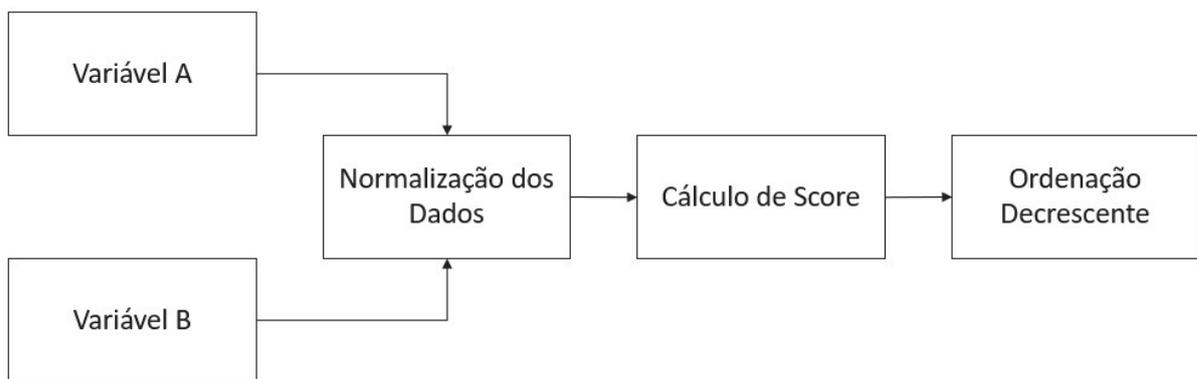
Ainda que o número de acessos no comércio eletrônico via tráfego pago é 4 vezes maior que via tráfego orgânico, o impacto transacional das sessões diárias que

interagem com o sistema de busca interno da empresa representa taxa de conversão 7 vezes maior para todos os canais de venda.

2.2 SISTEMA DE BUSCA INTERNO

Conforme Figura 3 abaixo, destaca-se o fluxograma do cálculo do *score* no método atual da empresa:

FIGURA 3 - FLUXOGRAMA DO CÁLCULO DO SCORE



FONTE: O autor (2023).

A principal ordenação de resultados do sistema de busca da empresa, denominada método atual, é construída por meio de variáveis recomendadas por especialistas da própria empresa conforme a Equação (1).

$$S = 0,4 \times A + 0,6 \times B \quad (1)$$

Por questão de sigilo da empresa, são mantidas A e B para as variáveis. Variáveis podem ser quantidade de pedidos realizados, quantidade de visualizações na página de produto, quantidade de cliques na página de busca, quantidade de adições ao carrinho, quantidade de receita gerada, quantidade de devoluções ou outras utilizadas de acordo com o padrão de mercado do comércio eletrônico.

São aplicadas técnicas de normalização de dados nas variáveis A e B para reduzir a redundância dos dados e obter o valor entre 0 e 1 por produto; após a normalização, é calculado o *product score* (S) de acordo com os pesos na Equação (1) e portanto, a ordenação final é a classificação decrescente de S (maior para o

menor valor calculado). Vale ressaltar que, embora os pesos atribuídos às variáveis sejam fixos, é possível adaptá-los a outros valores.

Isto posto, existem oportunidades para melhorar a ordenação de produtos buscados por meio de outras técnicas; neste trabalho buscou-se aplicar técnicas de aprendizado de máquina.

3 REVISÃO DA LITERATURA

3.1 RECUPERAÇÃO DE INFORMAÇÃO

A Recuperação de Informação (RI) é uma área da Ciência da Informação que se concentra em prover aos usuários o acesso às informações de seu interesse. Trata da representação, armazenamento, organização e acesso a itens de informação, como documentos, páginas da internet, catálogos online, registros estruturados e semiestruturados, objetos multimídia, entre outros. A representação e a organização dos itens de informação devem fornecer aos usuários facilidade de acesso as informações de seu interesse (BAEZA-YATES; RIBEIRO-NETO, 2013).

Em uma experiência digital diária, a RI permite que os usuários encontrem conteúdos relevantes em grandes coleções de documentos com mais facilidade. Exemplos dessa tarefa são pesquisas em *e-mails*, mensagens de mídia social, conteúdo de computador pessoal (pesquisa na área de trabalho) e páginas da internet (pesquisa *on-line*) (CAVALCANTE et al., 2020).

Existem diversos modelos de RI com objetivo de produzir uma função de ordenação atribuindo *score* a documentos em relação a uma determinada consulta. Os principais modelos clássicos são:

- O modelo booleano define que uma consulta é essencialmente uma expressão booleana convencional sobre termos de indexação com a previsão de que cada documento seja relevante ou não relevante conforme Equação (2). Suas vantagens são o formalismo claro por trás do modelo e a simplicidade com a adoção de pesos binários para os termos de indexação. Suas desvantagens são não existir ordenação para ranquear produtos e a formulação de consultas não ser apropriada para a maior parte dos usuários (BAEZA-YATES; RIBEIRO-NETO, 2013).

$$\text{sim}(d_j, q) = \begin{cases} 1, & \text{se } \exists c(q) \mid c(q) = c(d_j) \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

$c(q)$ = componente conjuntivo da consulta q

$c(d_j)$ = componente conjuntivo do documento d_j

- O modelo vetorial define que seja atribuído pesos não binários aos termos de indexação das consultas e dos documentos para computar o grau de similaridade entre eles ordenando os documentos recuperados de forma decrescente de acordo com esse grau de similaridade conforme Equação (3). Suas vantagens são o esquema de ponderação de termos para melhorar a qualidade da recuperação, a estratégia de casamento parcial que permite a recuperação de documentos aproximando as condições da consulta, a fórmula do cosseno que ordena os documentos de acordo com seu grau de similaridade em relação à consulta e a normalização pelo tamanho do documento que está naturalmente embutida no modelo. Sua principal desvantagem é que os termos de indexação são considerados independentes, o que não leva em consideração a semântica. (BAEZA-YATES; RIBEIRO-NETO, 2013).

$$\text{sim}(d_j, q) = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{i=1}^t w_{i,q}^2}} \quad (3)$$

$w_{i,j}$ = é o peso associado ao par termo – documento (k_i, d_j)

$w_{i,q}$ = é o peso associado ao par termo – consulta (k_i, q)

- O modelo probabilístico define que a partir de uma consulta do usuário, existe um conjunto (resposta ideal) de documentos que contém exatamente os documentos relevantes e nenhum outro conforme Equação (6). Sua principal vantagem é que documentos são ranqueados de forma decrescente de acordo com sua probabilidade de serem relevantes, com base na informação disponível ao sistema. Suas desvantagens são a necessidade de estimar a separação inicial dos documentos em conjuntos de relevantes e não relevantes, o método que não leva em consideração a frequência na qual um termo

de indexação ocorre em um documento e a falta de normalização pelo tamanho dos documentos (BAEZA-YATES; RIBEIRO-NETO, 2013).

$$\text{sim}(d_j, q) = \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{\rho_{iR}}{1 - \rho_{iR}}\right) + \log\left(\frac{1 - q_{iR}}{q_{iR}}\right) \quad (4)$$

ρ_{iR} = é a probabilidade que o termo de indexação k_i esteja presente em um documento aleatoriamente selecionado a partir do conjunto R de documentos relevantes (à consulta q)

q_{iR} = é a probabilidade que o termo de indexação k_i não esteja presente em um documento aleatoriamente selecionado a partir do conjunto R

A partir de então a área de Recuperação de Informação desenvolveu-se no ambiente acadêmico e diversos outros modelos com variações foram criados a partir dos modelos clássicos. Um exemplo é o modelo *BestMatch25* (BM25) que foi muito usado por se tratar de uma evolução do modelo probabilístico clássico atendendo a princípios do modelo vetorial como informações sobre a frequência dos termos e a normalização pelo tamanho dos documentos conforme Equação (5) (BAEZA-YATES; RIBEIRO-NETO, 2013).

$$\text{sim}_{\text{BM25}}(d_j, q) = \sum_{k_i \in [q, d_j]} \frac{(K_1 + 1)f_{i,j}}{K_1 \left[(1 - b) + b \frac{\text{len}(d_j)}{\text{avg_doclen}} \right] + f_{i,j}} \times \log\left(\frac{N - n_i + 0,5}{n_i + 0,5}\right) \quad (5)$$

$f_{i,j}$ = é a frequência do termo k_i na consulta q

K_1 = é uma constante que pode ser definida experimentalmente para uma coleção em particular

$\text{len}(d_j)$ = é o tamanho do documento d_j

avg_doclen = é o tamanho médio dos documentos na coleção

$k_i[q, d_j]$ = é uma notação condensada para $k_i \in q \wedge k_i \in d_j$

b = é uma nova constante introduzida com valores no intervalo $[0,1]$

As métricas de avaliação são importantes para determinar se o modelo de RI está ou não com um bom desempenho. Uma das métricas mais utilizadas na área de

RI é o Ganho Cumulativo Descontado Normalizado (*Normalized Discounted Cumulative Gain* - NDCG) conforme a Equação (6) abaixo:

$$\text{NDCG} = \sum \frac{2^{\text{rel}(p_{[i]})} - 1}{\log_2^i + 1} \text{iDCG}^{-1} \quad (6)$$

Dado um resultado ordenado de tamanho p , com i sendo a posição de algum documento neste resultado, esta métrica soma o ganho baseado na relevância atribuída $\text{rel}(p_{[i]})$ a cada documento e a divide por um fator de desconto (baseado no log da posição em que o documento foi apresentado). Este fator é o Ganho Cumulativo Descontado (DCG, do inglês - *Discounted Cumulative Gain*). O DCG é então dividido pelo DCG ideal (iDCG), ou seja, o DCG obtido dos documentos em ordem decrescente de sua relevância real conhecida. Assim é calculado o NDCG, o DCG normalizado pelo DCG ideal (FONSECA, 2018).

O NDCG mede a qualidade de uma lista de resultados retornada pelo modelo em relação aos itens que o usuário realmente indicou como relevantes. Isso facilita a comparação entre diferentes modelos de busca e recomendação usando aprendizado de máquina. Uma das maiores vantagens desta métrica é que ela leva em conta a posição dos resultados relevantes na lista, o que significa que um resultado relevante que aparece no topo da lista terá uma pontuação mais alta do que se ele aparecesse no final e sua desvantagem é a dificuldade em calcular e interpretar a fórmula.

Para mais detalhes sobre o cálculo do NDCG vide apêndice 1.

3.2 LEARNING TO RANK

A crescente complexidade dos problemas a serem computacionalmente tratados, e da velocidade e volume de dados gerados por diferentes setores, motivou o desenvolvimento de sistemas computacionais mais sofisticadas e autônomos, mais independentes da intervenção humana, para a aquisição de conhecimento. A maioria desses sistemas é baseada em aprendizado de máquina, uma subárea da inteligência artificial que faz parte de várias das tecnologias atualmente utilizadas (FACELI et al., 2021).

Com a evolução dos modelos de RI e a crescente geração de informação, sistemas de busca modernos levam em consideração inúmeros atributos para

melhorar a ordenação para atender a necessidade de um usuário. Learning to Rank (LTR) envolve a aplicação de algoritmos de aprendizado de máquina na otimização da estratégia de ordenação sendo fundamental para ofertar melhores serviços de busca (LI et al., 2020). Na prática, os documentos listados em uma página de pesquisa, possuem diversas características que podem ser utilizados como variáveis e o objetivo do algoritmo é encontrar um conjunto de padrões dentre essas variáveis para melhorar o resultado da ordenação desses documentos através das consultas realizadas.

As três principais abordagens estudadas no campo de LTR são:

- A abordagem *pointwise* assume que o exato grau de relevância de cada documento é o que vamos prever, embora isso possa não ser necessário quando o objetivo é produzir uma lista ordenada dos documentos. Por não considerar a interdependência entre documentos, a posição de um documento na lista de classificação final é invisível para sua função de perda. Além disso, a abordagem não faz uso do fato de que alguns documentos estão realmente associados à mesma consulta (LIU et al., 2009).
- A abordagem *pairwise* não se concentra em prever com precisão o grau de relevância de cada documento; o problema de classificação é reduzido a um problema de classificação em pares de documentos. Neste caso, a função de perda considera apenas a ordem relativa entre dois documentos. Quando se olha apenas um par de documentos, no entanto, a posição dos documentos na lista de classificação final dificilmente pode ser derivada. Além disso, a abordagem ignora o fato de que alguns pares são gerados a partir dos documentos associados à mesma consulta (LIU et al., 2009).
- A abordagem *listwise* possui um valor mais direto para entender o problema de ordenação real pois considera todos os documentos associados à mesma consulta como entrada e sua lista classificada (os seus graus de relevância) como saída. A vantagem desta abordagem frente as anteriores, reside no fato de que sua função de perda pode considerar naturalmente as posições dos documentos na lista

classificada de todos os documentos associados à mesma consulta (LIU et al., 2009).

Observe que as diferentes abordagens modelam o processo de LTR de maneiras diferentes, ou seja, definem diferentes espaços de entrada e saída, usam diferentes hipóteses e empregam diferentes funções de perda. Além disso, o espaço de saída é usado para facilitar o processo de aprendizado, que pode ser diferente dos julgamentos de relevância sobre os documentos (LIU et al., 2009).

O trabalho de Keyhanipour *et al.* (2016) apresentou um novo algoritmo de LTR, chamado QRC-Rank, o qual é aplicável para conjunto de dados com ou sem informações de cliques de usuários. O QRC-Rank desenvolve um modelo de aprendizagem por reforço com base nessas representações compactas de recursos.

O trabalho de Karmaker Santu *et al.* (2017) abordou os desafios práticos na aplicação de métodos de LTR na busca do comércio eletrônico. Seu estudo inclui os desafios na aprendizagem de características, obtenção de julgamentos de relevância confiáveis e exploração otimizada de vários sinais de feedback do usuário, como taxas de cliques, taxas de adição ao carrinho, taxas de pedidos e receita. Diversos algoritmos de LTR são utilizados neste trabalho, para avaliar qual destes apresenta melhor desempenho de acordo com as etapas da jornada de compra no comércio eletrônico.

O trabalho de Fonseca (2018) utilizou técnicas de aprendizado de máquina para aprender funções de ordenação de resultados no contexto de comércio eletrônico. Diferentes formas são utilizadas para estimativas de relevância e foram aplicados os diferentes métodos LTR em dados reais de empresas de segmentos diferentes.

O trabalho de Hu *et al.* (2018) usou aprendizagem por reforço para controle da ordenação em cenários de busca de comércio eletrônico. Seu objetivo era desenvolver um método para aprender uma política de ordenação ótima, que maximize as recompensas acumulativas esperadas em uma sessão de pesquisa. Foi reformulado o conceito de processo de decisão de Markov da sessão de pesquisa.

O trabalho de Zhuang *et al.* (2018) utilizou um modelo que otimiza diretamente o Volume Bruto de Mercadorias (GMV, do inglês – *Gross Merchandise Volume*) com uso de rede neural recorrente para capturar influências relacionadas a ordenação de pedidos na estimativa de probabilidade de compra.

O trabalho de Park *et al.* (2020) apresentou uma estrutura que aproveita os registros do histórico de cliques dos usuários para compensar as interações de produtos não comprados. Um modelo de aprendizado de máquina chamado P3Stop foi desenvolvido utilizando dados de transações de compra, dados de cliques na listagem de produtos de busca e dados de cliques na listagem de produtos de busca que não resultaram em transações de compras.

O trabalho de Zhang *et al.* (2020) usou um novo modelo de aprendizado profundo chamado Recuperação Semântica e Personalizada Profunda (Deep Personalized and Semantic Retrieval - DPSR) para resolver dois problemas conhecidos na pesquisa de comércio eletrônico: 1) como recuperar itens que são semanticamente relevantes, mas não correspondem exatamente aos termos da consulta, e 2) como recuperar itens que são mais personalizados para usuários diferentes para a mesma consulta de pesquisa.

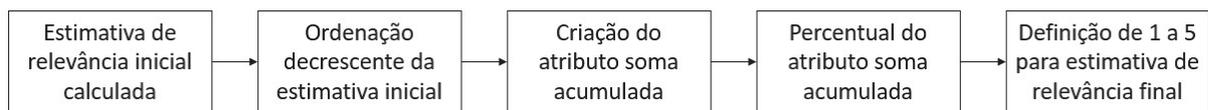
4 IMPLEMENTAÇÕES

Neste capítulo, será descrito o desenvolvimento feito para responder o problema de pesquisa abordado neste trabalho.

4.1 ESTIMATIVA DE RELEVÂNCIA

No ambiente do comércio eletrônico é possível estimar a relevância de cada produto a partir de dados reais dos usuários na plataforma. Conforme Figura 4 abaixo, destaca-se o fluxograma do cálculo da estimativa de relevância adotada neste trabalho.

FIGURA 4 - FLUXOGRAMA DO CÁLCULO DA ESTIMATIVA DE RELEVÂNCIA



FONTE: O autor (2023).

Foram realizadas três alternativas de estimativa de relevância iniciais considerando um período de 90 dias de dados acumulados: taxa de conversão de pedidos, taxa de adição no carrinho e taxa de cliques na busca. A justificativa da escolha das estimativas de relevâncias é a vantagem competitiva da empresa em coletar e armazenar os sinais de clientes durante a jornada de compra.

A seguir as Equações (7), (8) e (9) definem o cálculo de cada estimativa de relevância inicial por produto.

$$TCP = \frac{\#P}{\#V} \quad (7)$$

$$TAC = \frac{\#A}{\#I} \quad (8)$$

$$TCB = \frac{\#C}{\#I} \quad (9)$$

TCP é a taxa de conversão de pedidos;

TAC é a taxa de adição no carrinho;
TCB é a taxa de cliques na busca;
#P é a quantidade de pedidos;
#V é a quantidade de visualizações do produto;
#A é a quantidade de adição o carrinho;
#I é o número de impressões na busca;
#C é o número de cliques na busca.

Para modelos de Learning to Rank, recomenda-se que o cálculo da estimativa individual por produto esteja em uma escala de 5 pontos conforme estudo realizado de Karmaker Santu et al. (2017); portanto, para TCP, TAC e TCB foram considerados os seguintes passos:

1. Ordenação decrescente da estimativa de relevância inicial por produto (maior para o menor valor calculado);
2. Criação da coluna soma acumulada da relevância inicial;
3. Criação da coluna percentual dividindo a soma acumulada pela relevância inicial total;
4. Definição da estimativa de relevância final por produto.
 - a. Percentual até 20% - nota 5;
 - b. Percentual de 20 a 40% - nota 4;
 - c. Percentual de 40 a 60% - nota 3;
 - d. Percentual de 60 a 80% - nota 2;
 - e. Percentual de 80 a 100% - nota 1.

Para mais detalhes do cálculo das estimativas de relevância vide apêndice 2.

4.2 MÉTODO DE APRENDIZAGEM DE MÁQUINA

O método de aprendizagem de máquina escolhido para os experimentos foi o *Gradient Boosting Decision Trees* do LambdaMART. É um modelo de conjunto de árvores de decisão que são treinadas em sequência, onde em cada iteração, o algoritmo aprende as árvores de decisão observando os erros residuais.

A implementação do método escolhido foi a encontrada no pacote LightGBM. A justificativa da escolha deste método foi o resultado do trabalho de Karmaker Santu et al. (2017) que trouxe o modelo LambdaMART entre os melhores resultados dentro os modelos LTR aplicados em seu trabalho, que se assemelha com o presente trabalho proposto e por se tratar de dados de comércio eletrônico.

Para mais detalhes da aplicação do algoritmo LambdaMART vide apêndice 3.

4.3 EXPERIMENTOS

Antes da inicialização dos experimentos com o método de aprendizagem de máquina escolhido, realizou-se experimentos simulados, alterando apenas os pesos das variáveis A e B do método de ordenação atual da empresa conforme Equação (1). O objetivo foi avaliar se, necessariamente, era preciso criar um método mais robusto para atingir uma melhoria no NDCG do método atual. Nestas simulações, os parâmetros de entrada foram os pesos das variáveis A e B, ou seja, altera-se os pesos das variáveis conforme a Tabela 1. Por exemplo, para o experimento 5, considerou-se 0,5 para a variável A e 0,5 para a variável B. Portanto, em cada experimento o parâmetro de saída foi a lista ordenada de produtos conforme a ordenação decrescente de S.

TABELA 1 - EXPERIMENTOS COM ALTERAÇÃO DOS PESOS DAS VARIÁVEIS

Categoria X, Y e Z	Variável A	Variável B
Resultado Base	0,4	0,6
Experimento 1	0	1
Experimento 2	0,1	0,9
Experimento 3	0,2	0,8
Experimento 4	0,3	0,7
Experimento 5	0,5	0,5
Experimento 6	0,6	0,4
Experimento 7	0,7	0,3
Experimento 8	0,8	0,2
Experimento 9	0,9	0,1
Experimento 10	1	0

FONTE: O autor (2023).

Após estes experimentos simulados, aplicou-se o modelo LTR escolhido em mais 3 experimentos alternando a quantidade de variáveis conforme a Tabela 2. Os dois objetivos foram: 1) atingir melhoria no NDCG do método atual e, 2) avaliar se o número de variáveis traz incremento ao resultado NDCG. Portanto, os parâmetros de

entrada foram as variáveis e estimativas de relevância utilizadas no modelo e o parâmetro de saída foi a lista ordenada de produtos conforme o resultado de previsão do modelo.

TABELA 2 - EXPERIMENTOS COM ALTERAÇÃO DO NÚMERO DE VARIÁVEIS

Categoria X, Y e Z	Número de Variáveis	Variáveis
LambdaMART 1	2	A,B
LambdaMART 2	3	A,B,C
LambdaMART 3	11	A,B,C,D,E,F,G,H,I,J,K

FONTE: O autor (2023).

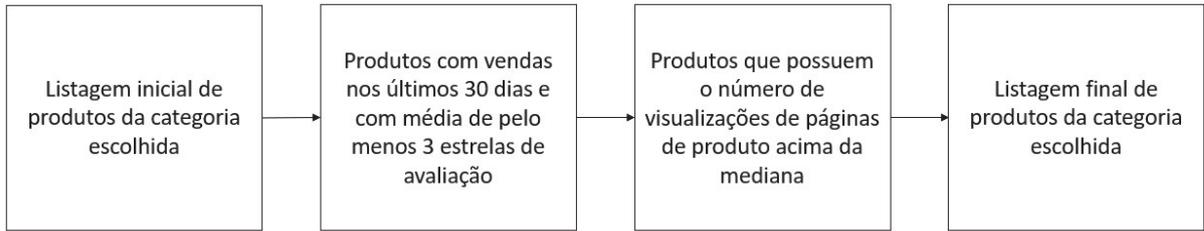
4.4 BASES DE DADOS

Todos os experimentos foram testados usando bases de dados de 3 categorias da empresa. Categorias podem ser guarda-roupas, sofás, escrivaninhas, armários de cozinha, cômodas, camas, mesas de jantar ou outras. Por sigilo, foram denominadas X, Y e Z. Até a data deste trabalho, a categoria X possuía um número total de 41.138 produtos, a categoria Y possuía 33.639 produtos e a categoria Z possuía 68.836 produtos. Para cada base de dados, foram realizados dois tratamentos de dados prévios: 1) selecionados apenas produtos que tiveram vendas nos últimos 30 dias e que possuem, individualmente, pelo menos a média 3 estrelas de avaliação no comércio eletrônico e, 2) selecionados apenas produtos que possuem o número de visualizações de páginas de produto acima da mediana da categoria. Após os tratamentos, o número de produtos da categoria X era de 656, da categoria Y era de 317 e da categoria Z era de 437.

Este procedimento foi uma recomendação de especialistas da empresa responsáveis pelas categorias testadas. O objetivo principal foi eliminar produtos sem relevância no comércio eletrônico da empresa como, por exemplo, produtos novos, produtos com poucas vendas, produtos com baixa avaliação, entre outros.

Conforme Figura 5 abaixo, destaca-se o fluxograma do tratamento nas bases de dados aplicado neste trabalho.

FIGURA 5 - FLUXOGRAMA DO TRATAMENTO NAS BASES DE DADOS



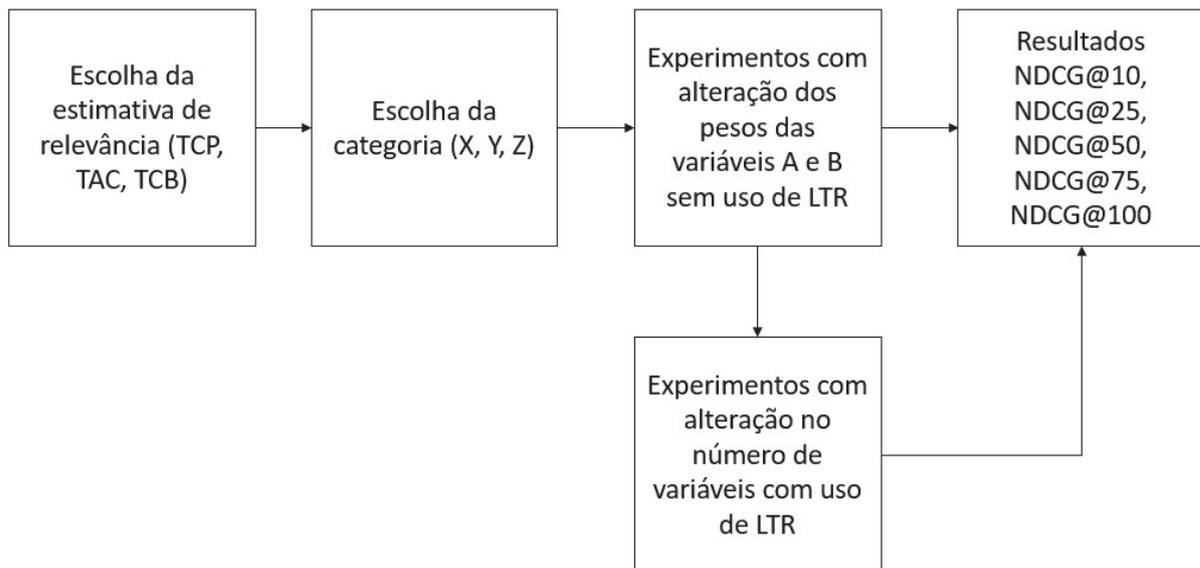
FONTE: O autor (2023).

5 RESULTADOS

Conforme recomendação de especialistas da empresa, a qualidade do sistema de busca do comércio eletrônico foi baseada na ordenação das 100 primeiras posições dos produtos listados do principal termo buscado da categoria escolhida. Sendo assim, em todos os experimentos utilizou-se a métrica $NDCG@k$, para $k \in \{10, 25, 50, 75, 100\}$ e considerando como $iDCG$ a ordenação da estimativa de relevância escolhida.

Conforme a Figura 6 abaixo, destaca-se o fluxograma dos experimentos e resultados aplicados neste trabalho.

FIGURA 6 - FLUXOGRAMA DOS EXPERIMENTOS E RESULTADOS



FONTE: O autor (2023).

O computador utilizado para os experimentos tem processador Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz-1.99 GHz e memória RAM instalada de 12,0 GB. O código usado foi o *python* e o sistema para desenvolvimento foi o *jupyter notebook* local.

Para os resultados a seguir, em destaque **negrito**, estão todos os valores superiores aos resultados do método atual do sistema de busca da empresa (resultado base).

5.1 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE CONVERSÃO DE PEDIDOS

A seguir os resultados consolidados dos experimentos das 3 categorias usando a estimativa de relevância como taxa de conversão de pedidos. Para conhecimento, a categoria X possui aproximadamente uma quantidade de pedidos 3 vezes maior que as categorias Y e Z.

TABELA 3 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE CONVERSÃO DE PEDIDOS

Categoria X	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,916642	0,793905	0,802118	0,829422	0,830639
Experimento 1	0,961521	0,883843	0,903808	0,898364	0,895763
Experimento 2	0,923042	0,883843	0,894902	0,877502	0,877540
Experimento 3	0,923042	0,869384	0,889841	0,865857	0,855534
Experimento 4	0,916642	0,837283	0,837120	0,846243	0,852330
Experimento 5	0,874963	0,779446	0,784305	0,819623	0,811387
Experimento 6	0,868563	0,777407	0,747961	0,792603	0,803047
Experimento 7	0,868563	0,772367	0,726744	0,776645	0,788058
Experimento 8	0,851877	0,727133	0,717837	0,762890	0,762955
Experimento 9	0,791605	0,713568	0,700117	0,751080	0,755832
Experimento 10	0,780062	0,685544	0,687493	0,735987	0,738196
LambdaMART 1	0,923042	0,93160	0,952499	0,944679	0,942878
LambdaMART 2	0,991657	0,975892	0,969063	0,964859	0,962366
LambdaMART 3	0,961521	0,964185	0,970408	0,965552	0,957995

FONTE: O autor (2023).

Para a categoria X, na Tabela 3, os experimentos 1, 2 e 3 atingiram resultados superiores em todas as métricas quando comparado ao resultado base. O experimento 4, atingiu resultados superiores em 4 das métricas e resultado igual ao resultado base para NDCG@10. No entanto, quando observados os resultados dos experimentos do modelo de aprendizado de máquina LambdaMART, todas as métricas estão com resultados acima do resultado base, inclusive sendo os melhores resultados atingidos.

TABELA 4 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE CONVERSÃO DE PEDIDOS

Categoria Y	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,662741	0,768131	0,798057	0,827410	0,794315
Experimento 1	0,628693	0,636021	0,656954	0,690318	0,702195
Experimento 2	0,880078	0,886333	0,892696	0,888230	0,875746
Experimento 3	0,836115	0,847726	0,862035	0,877267	0,850655
Experimento 4	0,748188	0,821468	0,831546	0,836220	0,835394
Experimento 5	0,650565	0,745301	0,775315	0,809437	0,793750
Experimento 6	0,650565	0,719043	0,744654	0,783657	0,776255
Experimento 7	0,640869	0,671357	0,709479	0,745953	0,738808
Experimento 8	0,640869	0,648527	0,691724	0,732045	0,737409
Experimento 9	0,628693	0,642975	0,661468	0,718136	0,711943

Experimento 10	0,628693	0,636021	0,656954	0,690318	0,702195
LambdaMART 1	0,994833	0,956889	0,951996	0,943865	0,941483
LambdaMART 2	0,992354	0,978940	0,969820	0,952902	0,956150
LambdaMART 3	0,992354	0,980966	0,954593	0,943823	0,948861

FONTE: O autor (2023).

Para a categoria Y, na Tabela 4, os experimentos 2, 3 e 4 atingiram resultados superiores em todas as métricas quando comparado ao resultado base. Quando observados os resultados dos experimentos do modelo de aprendizado de máquina LambdaMART, todas as métricas estão com resultados superiores, sendo os melhores resultados atingidos.

TABELA 5 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE CONVERSÃO DE PEDIDOS

Categoria Z	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,730440	0,745971	0,780625	0,762924	0,764631
Experimento 1	0,819593	0,843402	0,878753	0,861782	0,853793
Experimento 2	0,819593	0,829170	0,869004	0,839685	0,836337
Experimento 3	0,807675	0,809320	0,841100	0,822416	0,813709
Experimento 4	0,763099	0,775578	0,794572	0,789274	0,791081
Experimento 5	0,732426	0,726121	0,726538	0,724451	0,736524
Experimento 6	0,675931	0,675271	0,688036	0,693403	0,712622
Experimento 7	0,675931	0,652085	0,666355	0,667498	0,694559
Experimento 8	0,675931	0,633376	0,630233	0,645645	0,663655
Experimento 9	0,631355	0,614667	0,603007	0,626585	0,645532
Experimento 10	0,598696	0,591480	0,590411	0,599845	0,625638
LambdaMART 1	0,947858	0,971151	0,980779	0,968771	0,970485
LambdaMART 2	0,964992	0,965872	0,975085	0,969573	0,966901
LambdaMART 3	0,972193	0,974869	0,978344	0,973596	0,970912

FONTE: O autor (2023).

Para a categoria Z, na Tabela 5, os experimentos 1, 2, 3 e 4 atingiram resultados superiores em todas as métricas quando comparado ao resultado base e o experimento 5 apenas para NDCG@10. Quando observados os resultados dos experimentos do modelo de aprendizado de máquina LambdaMART, todas as métricas estão com resultados superiores para o NDCG, sendo os melhores resultados atingidos.

5.2 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE CLIQUES NA BUSCA

A seguir os resultados consolidados dos experimentos das 3 categorias usando a estimativa de relevância como taxa de clique na busca. Para conhecimento, a categoria X possui aproximadamente uma quantidade de cliques na busca 2 vezes maior que a categoria Y e 4,5 vezes maior que a categoria Z.

TABELA 6 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE CLIQUES NA BUSCA

Categoria X	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,938777	0,904079	0,843456	0,829396	0,828368
Experimento 1	0,859362	0,848844	0,822841	0,820247	0,819256
Experimento 2	0,886963	0,850079	0,823592	0,820243	0,825907
Experimento 3	0,909860	0,869755	0,839278	0,818211	0,831189
Experimento 4	0,938777	0,881092	0,841942	0,835412	0,823461
Experimento 5	0,938777	0,907390	0,846919	0,823830	0,836255
Experimento 6	0,944798	0,926584	0,849583	0,835015	0,845746
Experimento 7	0,921901	0,945777	0,856062	0,823432	0,840195
Experimento 8	0,944798	0,945295	0,880119	0,827697	0,834779
Experimento 9	0,949502	0,945295	0,880119	0,828878	0,837016
Experimento 10	0,948185	0,927336	0,880119	0,833890	0,833803
LambdaMART 1	0,936144	0,915776	0,879373	0,860601	0,862942
LambdaMART 2	0,995296	0,982571	0,909146	0,885721	0,890315
LambdaMART 3	0,977103	0,972075	0,956470	0,936399	0,934199

FONTE: O autor (2023).

Para a categoria X, na Tabela 6, os experimentos 6 e 10 atingiram resultados superiores em todas as métricas quando comparado ao resultado base. Os experimentos 3 e 4, atingiram resultados superiores em apenas uma das métricas avaliadas, enquanto os experimentos 5 e 7 em três das métricas avaliadas. Os experimentos 8 e 9 só não atingiram resultados maiores ao resultado base para NDCG@75. Quando observados os resultados dos experimentos do modelo de aprendizado de máquina LambdaMART 2 e 3, todas as métricas estão com resultados superiores e apenas o LambdaMART 1 não superou o resultado base para NDCG@10. Ainda assim, os melhores resultados atingidos são com o modelo de LTR.

TABELA 7 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE CLIQUES NA BUSCA

Categoria Y	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,906712	0,864185	0,802310	0,825021	0,818966
Experimento 1	0,827680	0,758418	0,770673	0,810522	0,791084
Experimento 2	0,827680	0,796568	0,772175	0,818050	0,798400
Experimento 3	0,814901	0,796568	0,784358	0,812724	0,810218
Experimento 4	0,857983	0,837666	0,801550	0,823155	0,816249
Experimento 5	0,920448	0,873917	0,835492	0,831951	0,815517
Experimento 6	0,920448	0,904624	0,832065	0,834255	0,829330
Experimento 7	0,920448	0,907571	0,834158	0,843154	0,823066
Experimento 8	0,920448	0,907571	0,834538	0,849149	0,829905
Experimento 9	0,920448	0,900127	0,853824	0,853423	0,823642
Experimento 10	0,920448	0,894312	0,856660	0,855392	0,831478
LambdaMART 1	0,946006	0,900127	0,863817	0,867019	0,869072
LambdaMART 2	0,933227	0,909199	0,914023	0,918473	0,916275
LambdaMART 3	0,998132	0,975381	0,959297	0,949222	0,944428

FONTE: O autor (2023).

Para a categoria Y, na Tabela 7, a partir do experimento 6 até o experimento 10, os resultados são superiores em todas as métricas. O experimento 5 não superou apenas para NDCG@100. Os experimentos do modelo de aprendizado de máquina LambdaMART, superaram todas as métricas e os melhores resultados atingidos fazem parte do LambdaMART 3.

TABELA 8 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE CLIQUES NA BUSCA

Categoria Z	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,947916	0,859072	0,853583	0,852916	0,869296
Experimento 1	0,947916	0,870475	0,857941	0,855679	0,851703
Experimento 2	0,947916	0,870475	0,857267	0,860706	0,852742
Experimento 3	0,947916	0,858844	0,857941	0,864605	0,862807
Experimento 4	0,947916	0,846412	0,850987	0,858533	0,869339
Experimento 5	0,947916	0,870400	0,855887	0,862887	0,863720
Experimento 6	0,941989	0,841749	0,839155	0,843277	0,865401
Experimento 7	0,941989	0,840418	0,845363	0,844401	0,854655
Experimento 8	0,941989	0,839086	0,847375	0,840244	0,847129
Experimento 9	0,914332	0,835203	0,839536	0,831788	0,827729
Experimento 10	0,917031	0,848566	0,836559	0,829603	0,834349
LambdaMART 1	0,978271	0,929263	0,905473	0,901133	0,900261
LambdaMART 2	0,975573	0,945217	0,926983	0,899738	0,902848
LambdaMART 3	0,969645	0,985381	0,955286	0,952825	0,950675

FONTE: O autor (2023).

Para a categoria Z, na Tabela 8, nenhum experimento de 1 a 10 superou todos os valores das métricas. Os melhores foram os experimentos 1, 2 e 5 que tiveram 3 métricas maiores que o resultado base, enquanto os experimentos 3 e 4 tiveram apenas 2 métricas maiores. No entanto, novamente os experimentos do modelo de aprendizado de máquina LambdaMART, superaram todas as métricas com os melhores resultados atingidos.

5.3 ESTIMATIVA DE RELEVÂNCIA COMO TAXA DE ADIÇÃO NO CARRINHO

A seguir os resultados consolidados dos experimentos das 3 categorias usando a estimativa de relevância como taxa de adição no carrinho. Para conhecimento, a categoria X possui aproximadamente uma quantidade de adições no carrinho 2 vezes maior que a categorias Y e 5 vezes maior que a categoria Z.

TABELA 9 - EXPERIMENTOS DA CATEGORIA X USANDO TAXA DE ADIÇÃO NO CARRINHO

Categoria X	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,963985	0,903975	0,766015	0,789049	0,758641

Experimento 1	0,937541	0,878009	0,760146	0,759913	0,757517
Experimento 2	0,937541	0,878009	0,766304	0,753713	0,761441
Experimento 3	0,937541	0,877166	0,773051	0,762506	0,765895
Experimento 4	0,963985	0,890570	0,780675	0,754609	0,760571
Experimento 5	0,963985	0,918223	0,772173	0,760810	0,757580
Experimento 6	0,968771	0,918223	0,766607	0,787990	0,758980
Experimento 7	0,995215	0,937613	0,763096	0,788411	0,756457
Experimento 8	1,000000	0,933355	0,754595	0,782211	0,757857
Experimento 9	0,995215	0,933355	0,753717	0,781573	0,755587
Experimento 10	0,995215	0,919107	0,752839	0,779240	0,751133
LambdaMART 1	0,990430	0,966953	0,824655	0,830175	0,843912
LambdaMART 2	0,990430	0,930745	0,881697	0,861438	0,861658
LambdaMART 3	0,906312	0,859503	0,781428	0,792701	0,842355

FONTE: O autor (2023).

Para a categoria X, na Tabela 9, o experimento 6 foi o que teve melhor desempenho comparado ao resultado base, superando 4 métricas, enquanto os experimentos 2, 3, 4, 5, 7, 8, 9, 10 só superaram 2 métricas. Os experimentos do modelo de aprendizado de máquina LambdaMART 1 e 2, superaram todas as métricas, enquanto o LambdaMART 3 superou apenas 3 métricas. Com exceção do NDCG@10, os melhores resultados das métricas fazem parte do modelo proposto.

TABELA 10 - EXPERIMENTOS DA CATEGORIA Y USANDO TAXA DE ADIÇÃO NO CARRINHO

Categoria Y	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,951521	0,903708	0,777906	0,781836	0,769378
Experimento 1	0,908831	0,819795	0,790448	0,742192	0,798518
Experimento 2	0,908831	0,819795	0,802869	0,764949	0,755237
Experimento 3	0,908831	0,819795	0,815291	0,766665	0,756725
Experimento 4	0,945732	0,881182	0,815291	0,775967	0,761564
Experimento 5	0,951521	0,906803	0,788288	0,791138	0,769378
Experimento 6	0,951521	0,909897	0,792366	0,792854	0,775704
Experimento 7	0,951521	0,912992	0,800709	0,800439	0,780166
Experimento 8	0,951521	0,912992	0,800709	0,800439	0,780166
Experimento 9	0,951521	0,912992	0,802748	0,800439	0,772352
Experimento 10	0,951521	0,912992	0,792366	0,800439	0,773839
LambdaMART 1	0,994211	0,971285	0,861085	0,830199	0,852481
LambdaMART 2	0,994211	0,971285	0,900509	0,906572	0,881935
LambdaMART 3	0,945732	0,920045	0,879744	0,843346	0,846155

FONTE: O autor (2023).

Para a categoria Y, na Tabela 10, os experimentos de 6 a 10 tiveram melhores resultados quando comparados ao resultado base, superando 4 métricas. O experimento 5 superou o em apenas 3 métricas, seguido dos experimentos 1 com apenas 2 métricas e os experimentos 2, 3 e 4 com apenas 1 métrica superior. Os experimentos do modelo de aprendizado de máquina LambdaMART 1 e 2, tiveram

todos resultados superiores e o LambdaMART 3 só não superou o NDCG@10. Ainda assim, os maiores resultados foram atingidos pelo modelo proposto.

TABELA 11 - EXPERIMENTOS DA CATEGORIA Z USANDO TAXA DE ADIÇÃO NO CARRINHO

Categoria Z	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100
Resultado Base	0,942502	0,874800	0,808283	0,788636	0,792665
Experimento 1	0,987487	0,931695	0,866094	0,802014	0,810976
Experimento 2	0,987487	0,931695	0,833828	0,807527	0,810976
Experimento 3	0,942502	0,895356	0,838013	0,784740	0,809335
Experimento 4	0,942502	0,895356	0,808283	0,793056	0,788365
Experimento 5	0,938723	0,874800	0,792379	0,770411	0,796408
Experimento 6	0,932150	0,868163	0,779234	0,763855	0,764737
Experimento 7	0,932150	0,861526	0,777965	0,758484	0,763829
Experimento 8	0,932150	0,856409	0,776996	0,754014	0,759353
Experimento 9	0,932150	0,853091	0,770953	0,730277	0,735658
Experimento 10	0,932150	0,853091	0,770953	0,742846	0,746852
LambdaMART 1	0,927194	0,886457	0,879909	0,827746	0,839462
LambdaMART 2	0,913697	0,886737	0,925321	0,919726	0,914397
LambdaMART 3	0,974974	0,942113	0,915392	0,874294	0,868269

FONTE: O autor (2023).

Para a categoria Z, na Tabela 11, os experimentos 1 e 2 tiveram resultados melhores em todas as métricas quando comparados ao resultado base. O experimento 3 superou em apenas 3 métricas, seguido do experimento 4 com 2 métricas e o experimento 5 com apenas 1 métrica melhor. Os resultados dos experimentos do modelo LambdaMART 3 foram superiores em todas as métricas e os demais só não tiveram resultados superiores em apenas 1 métrica. Ainda assim, o LambdaMART 2 teve maiores resultados para as métricas NDCG@50, NDCG@75 e NDCG@100, seguido do LambdaMART 3 que teve o maior resultado para NDCG@25. Os experimentos 1 e 2 tiveram o maior resultado para NDCG@10.

5.4 RESULTADOS CONSOLIDADOS POR EXPERIMENTOS

A seguir o consolidado de todos os experimentos deste trabalho, o qual cada métrica teve a chance de superar o resultado base até 9 vezes.

TABELA 12 - RESULTADOS CONSOLIDADOS POR EXPERIMENTOS

Consolidado	NDCG@10	NDCG@25	NDCG@50	NDCG@75	NDCG@100	> Resultado Base
Experimento 1	3	4	5	4	4	44%
Experimento 2	4	5	7	5	5	58%
Experimento 3	3	4	7	4	6	53%

Experimento 4	2	4	5	6	5	49%
Experimento 5	1	5	5	3	2	36%
Experimento 6	3	4	4	3	4	40%
Experimento 7	2	4	3	2	3	31%
Experimento 8	3	4	3	2	3	33%
Experimento 9	3	4	3	2	3	33%
Experimento 10	3	4	3	3	3	36%
LambdaMART 1	7	9	9	9	9	96%
LambdaMART 2	8	9	9	9	9	98%
LambdaMART 3	7	8	9	9	9	93%

FONTE: O autor (2023).

Com esses resultados consolidados, pode-se concluir que os experimentos do modelo proposto atingiram desempenho melhores que o resultado base de 96% para LambdaMART 1, 98% para LambdaMART 2 e 93% para LambdaMART 3 mostrando muito mais eficiência quando comparado aos experimentos com a regra atual. O melhor desempenho dos experimentos alternando os pesos da metodologia atual foi obtido com o experimento 2 que obteve 58% dos resultados melhores que o resultado base.

6 CONCLUSÕES

O presente trabalho abordou um estudo de caso com uso de técnicas de aprendizado de máquina para ordenação de resultados de busca com dados reais de um comércio eletrônico do segmento de bens para o lar. A partir da literatura sobre Learning to Rank (LTR) foi construído um programa computacional do método escolhido. A justificativa do trabalho é o crescente avanço da inteligência artificial e desafios no ambiente corporativo.

Ao analisar os resultados dos experimentos, com base na métrica $NDCG@k$, concluiu-se a necessidade de construção do modelo LambdaMART. Em geral, os resultados apresentados revelam que somente ajustes de parâmetros nos pesos das variáveis do método da empresa atual, não trouxe resultados superiores ao resultado base em 80% dos experimentos. Dos 10 experimentos testados, apenas os experimentos 2 e 3 ficaram acima de 50% com 58% e 53%, respectivamente.

Como apresentado anteriormente, a proposta foi a elaboração de um modelo com maior robustez para o problema real de um sistema de busca no comércio eletrônico de uma indústria de bens para o lar, por meio de técnicas de aprendizado de máquina. Sendo assim, após aplicada uma técnica de LTR, pode-se concluir que os resultados são superiores. Conforme a Tabela 12 de resultados consolidados por experimento, a aplicação do modelo LambdaMART nos 3 experimentos testados obteve um desempenho médio maior que 95%.

Não obstante, vale ressaltar que o aumento do número de variáveis, não necessariamente aumenta o desempenho do modelo LTR. Em alguns casos até pioraram as métricas apresentadas. Por isso, vale sempre ter cautela na implementação de novas variáveis no modelo, devendo fazer os testes estatísticos adequados sobre as variáveis que explicam o fenômeno para serem utilizadas de entrada nos modelos de aprendizado de máquina. Portanto, foi possível concluir que a aplicação de uma técnica LTR fornece resultados melhores (maior $NDCG@k$) em relação ao método atual da empresa.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Ao que tange o campo de estudo LTR, existem diversos modelos sendo desenvolvidos e abordados na literatura acadêmica como o apresentado neste

trabalho. Este trabalho se limitou ao uso do modelo LambdaMART e comparar com outros modelos de LTR pode ser uma alternativa para avaliar se existe um modelo que performa melhor no estudo de caso. Além disso, uma outra possibilidade é aplicar na prática testes online, ou seja, testes A/B no comércio eletrônico para dar mais embasamento nos resultados apresentados e se de fato houve melhoria na receita da empresa.

O modelo apresentado também se limitou com o número de experimentos e a aplicação de estimativas de relevância baseada em avaliação de usuários como taxa de conversão de pedidos, taxa de cliques na busca e taxa de adição no carrinho. Recomenda-se gerar mais amostras aleatórias com cálculos de intervalos de confiança de 95%, além de usar outros métodos de relevância para estudos futuros como, por exemplo, estimativas de relevância que considera o viés da posição do produto na página de listagem de busca.

Por fim, este trabalho também se restringiu a um segmento específico de um comércio eletrônico, uma indústria de bens para o lar com apenas 3 categorias sendo testadas, e pode ser interessante aplicar o modelo LambdaMART para diferentes segmentos do comércio eletrônico e avaliar se os resultados também serão positivos.

REFERÊNCIAS

Al, Qingyao et al. Unbiased learning to rank: online or offline?. **ACM Transactions on Information Systems (TOIS)**, v. 39, n. 2, p. 1-29, 2021.

BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca**. Bookman Editora, 2013.

CAVALCANTE, Larícia et al. Improving Search Quality with Automatic Ranking Evaluation and Tuning. In: **Anais do XXXV Simpósio Brasileiro de Bancos de Dados**. SBC, 2020. p. 157-162.

FACELI, Katti et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Livros Técnicos e Científicos Editora Ltda, 2021.

FERREIRA, Kris J.; PARTHASARATHY, Sunanda; SEKAR, Shreyas. Learning to rank an assortment of products. **Management Science**, v. 68, n. 3, p. 1828-1848, 2022.

FONSECA, Roberto Cidade. **Learning to rank para busca em Comércio Eletrônico**. Dissertação – Universidade Federal do Amazonas. 2018.

GOSWAMI, Anjan; ZHAI, Chengxiang; MOHAPATRA, Prasant. Learning to Diversify for E-commerce Search with Multi-Armed Bandit. In: **eCOM@ SIGIR**. 2019.

HU, Yujing et al. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In: **Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining**. 2018. p. 368-377.

KARMAKER SANTU, Shubhra Kanti; SONDHI, Parikshit; ZHAI, ChengXiang. On application of learning to rank for e-commerce search. In: **Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval**. 2017. p. 475-484.

KEYHANIPOUR, Amir Hosein et al. Learning to rank with click-through features in a reinforcement learning framework. **International Journal of Web Information Systems**, 2016.

LI, Pengcheng et al. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space. In: **Proceedings of the 29th ACM International Conference on Information & Knowledge Management**. 2020. p. 2605-2612.

LIU, Tie-Yan et al. Learning to rank for information retrieval. **Foundations and Trends® in Information Retrieval**, v. 3, n. 3, p. 225-331, 2009.

PARK, Chanyoung et al. Click-aware purchase prediction with push at the top. **Information Sciences**, v. 521, p. 350-364, 2020.

RITA, Paulo; RAMOS, Ricardo F. Global research trends in consumer behavior and sustainability in E-Commerce: A bibliometric analysis of the knowledge structure. **Sustainability**, v. 14, n. 15, p. 9455, 2022.

SOROKINA, Daria; CANTU-PAZ, Erick. Amazon search: The joy of ranking products. In: **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**. 2016. p. 459-460.

ZHANG, Han et al. Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In: **Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval**. 2020. p. 2407-2416.

ZHUANG, Tao; OU, Wenwu; WANG, Zhirong. Globally optimized mutual influence aware ranking in e-commerce search. **arXiv preprint arXiv:1805.08524**, 2018.

APÊNDICE 1 – EXEMPLO DE CÁLCULO DO NDCG

Para o cálculo do NDCG é necessário entender o *Discounted Cumulative Gain* (DCG), o qual é calculado atribuindo pontos aos resultados relevantes de acordo com a relevância e a posição deles na lista.

A fórmula do DCG para uma lista de resultados é:

$$DCG = \sum_{i=1}^n \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

- n é o número de resultados na lista
- i é a posição do item na lista
- rel_i é a relevância do item i

Para exemplificar o cálculo do DCG temos a tabela abaixo:

EXEMPLO DE CÁLCULO DO DCG

Posição	ID do produto	Relevância	$2^{rel_i} - 1$	$\log_2(i + 1)$	DCG posição
1	A	3	7	1,000000	7,000000
2	B	5	31	1,584963	19,55882
3	C	1	1	2,000000	0,500000
4	D	2	3	2,321928	1,292030
5	E	4	15	2,584963	5,802792

FONTE: O autor (2023).

- A coluna “Posição” é a posição do produto na lista retornada, sendo 1 para o produto A e 5 para o produto E.
- A coluna “ID do Produto” é o produto recomendado de acordo com a categoria escolhida.
- A coluna “Relevância” é a relevância do produto para o usuário, sendo 5 para o produto B (mais relevante) e 1 para o produto C (menos relevante).
- A coluna “ $2^{rel_i} - 1$ ” é o numerador da fórmula do DCG, onde cada ponto de relevância traz ganhos exponenciais.

- A coluna “ $\log_2(i + 1)$ ” é o denominador da fórmula do DCG, onde quanto mais baixa a posição do produto na lista, maior será a penalidade sobre a relevância do cálculo do DCG, mas numa escala logarítmica.
- A coluna “DCG” mostra o cálculo da fração para cada produto, onde o somatório dessa coluna mostra o DCG da lista retornada.

Para obter o DCG desta lista, basta somar os valores de cada produto, que dá um total aproximado de 34,15.

O NDCG é o DCG dos resultados normalizado pelo DCG ideal, ou seja, é a divisão do DCG calculado anteriormente pelo DCG dos resultados ordenados por relevância (a lista perfeita) conforme tabela abaixo.

EXEMPLO DE CÁLCULO DO DCG IDEAL

Posição	ID do produto	Relevância	$2^{rel_i} - 1$	$\log_2(i + 1)$	DCG posição
1	B	5	31	1,000000	31,00000
2	E	4	15	1,584963	9,463946
3	A	3	7	2,000000	3,500000
4	D	2	3	2,321928	1,292030
5	C	1	1	2,584963	0,386853

FONTE: O autor (2023).

Nesta lista, temos um total aproximado de 45,64 que é o IDCG ou DCG ideal.

Portanto, para calcular o NDCG basta dividir o DCG pelo IDCG. Obtém-se um valor entre 0 e 1, sendo que quanto mais próximo de 1, melhor.

No exemplo apresentado, NDCG@5 seria de aproximadamente 0,75.

APÊNDICE 2 – EXEMPLO DE CÁLCULO DAS ESTIMATIVAS DE RELEVÂNCIA

Para exemplificar o cálculo das estimativas de relevância temos a tabela abaixo:

EXEMPLO DE DADOS DE PRODUTOS COM ESTIMATIVAS DE RELEVÂNCIAS INICIAIS

ID do produto	Quantidade de pedidos	Quantidade de visualizações de produtos	Quantidade de adição no carrinho	Número de impressões na busca	Número de cliques na busca	Taxa de conversão de pedidos	Taxa de adição no carrinho	Taxa de cliques na busca
A	39	5217	231	1790	514	0,007476	0,12905	0,287151
B	30	6641	150	1076	487	0,004517	0,139405	0,452602
C	38	7348	225	1155	604	0,005171	0,194805	0,522944
D	22	9570	110	1631	473	0,002299	0,067443	0,290006
E	38	6478	70	1018	335	0,005866	0,068762	0,329077
F	15	5052	194	1306	559	0,002969	0,148545	0,428025
G	30	5026	248	1613	336	0,005969	0,153751	0,208308
H	29	7032	130	1863	313	0,004124	0,06978	0,168009
I	17	6033	63	1199	667	0,002818	0,052544	0,556297
J	15	8848	235	1521	390	0,001695	0,154504	0,256410

FONTE: O autor (2023).

- A coluna “ID do Produto” é o produto recomendado de acordo com a categoria escolhida.
- A coluna “Quantidade de pedidos” é a quantidade de pedidos de compras realizada no comércio eletrônico durante o período de 90 dias.
- A coluna “Quantidade de visualizações de produto” é a quantidade de visualizações na página de produto antes do produto ser adicionado ao carrinho, podendo ser acessada via tráfego orgânico ou pago durante o período de 90 dias.
- A coluna “Quantidade de adição no carrinho” é a quantidade de adições ao carrinho após a jornada de busca durante o período de 90 dias.
- A coluna “Número de impressões na busca” é a quantidade de impressões que um produto teve na listagem de busca após um termo pesquisado durante o período de 90 dias.

- A coluna “Número de cliques na busca” é a quantidade de cliques que um produto teve na listagem de busca após um termo pesquisado durante o período de 90 dias.
- A coluna “Taxa de conversão de pedidos” é a divisão da coluna “Quantidade de pedidos” pela coluna “Quantidade de visualizações de produto”.
- A coluna “Taxa de adição no carrinho” é a divisão da coluna “Quantidade de adição no carrinho” pela coluna “Número de impressões na busca”.
- A coluna “Taxa de cliques na busca” é a divisão da coluna “Número de cliques na busca” pela coluna “Número de impressões na busca”.

Portanto, para cálculo da estimativa de relevância temos com os seguintes passos aplicados na tabela anterior:

1. Ordenação decrescente da estimativa de relevância inicial por produto (maior para o menor valor calculado);
2. Criação da coluna soma acumulada da relevância inicial;
3. Criação da coluna percentual dividindo a soma acumulada pela relevância inicial total;
4. Definição da estimativa de relevância final por produto.
 - a. Percentual até 20% - nota 5;
 - b. Percentual de 20 a 40% - nota 4;
 - c. Percentual de 40 a 60% - nota 3;
 - d. Percentual de 60 a 80% - nota 2;
 - e. Percentual de 80 a 100% - nota 1.

EXEMPLO DE ESTIMATIVA DE RELEVÂNCIA PARA TAXA DE CONVERSÃO DE PEDIDOS

ID do produto	Taxa de conversão de pedidos	Soma acumulada	Percentual da soma acumulada	Relevância
A	0,0075	0,0075	0,1742	5
G	0,0060	0,0134	0,3134	4
E	0,0059	0,0193	0,4501	3
C	0,0052	0,0245	0,5706	3
B	0,0045	0,0290	0,6759	2

H	0,0041	0,0331	0,7720	2
F	0,0030	0,0361	0,8412	1
I	0,0028	0,0389	0,9069	1
D	0,0023	0,0412	0,9605	1
J	0,0017	0,0429	1,0000	1

FONTE: O autor (2023).

EXEMPLO DE ESTIMATIVA DE RELEVÂNCIA PARA TAXA DE ADIÇÃO NO CARRINHO

ID do produto	Taxa de adição no carrinho	Soma acumulada	Percentual da soma acumulada	Relevância
C	0,1948	0,1948	0,1653	5
J	0,1545	0,3493	0,2964	4
G	0,1538	0,5031	0,4268	3
F	0,1485	0,6516	0,5529	3
B	0,1394	0,7910	0,6711	2
A	0,1291	0,9201	0,7806	2
H	0,0698	0,9898	0,8399	1
E	0,0688	1,0586	0,8982	1
D	0,0674	1,1260	0,9554	1
I	0,0525	1,1786	1,0000	1

FONTE: O autor (2023).

EXEMPLO DE ESTIMATIVA DE RELEVÂNCIA PARA TAXA DE CLIQUES NA BUSCA

ID do produto	Taxa de cliques na busca	Soma acumulada	Percentual da soma acumulada	Relevância
I	0,5563	0,5563	0,1590	5
C	0,5229	1,0792	0,3085	4
B	0,4526	1,5318	0,4378	3
F	0,4280	1,9599	0,5601	3
E	0,3291	2,2889	0,6542	2
D	0,2900	2,5790	0,7371	2
A	0,2872	2,8661	0,8192	1
J	0,2564	3,1225	0,8924	1
G	0,2083	3,3308	0,9520	1
H	0,1680	3,4988	1,0000	1

FONTE: O autor (2023).

APÊNDICE 3 – EXEMPLO DE APLICAÇÃO DO ALGORITMO LAMBDMART

Importar as bibliotecas que serão usadas no trabalho

```
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import lightgbm as lgb
from sklearn.metrics import ndcg_score
```

Importar os arquivos que serão usados no trabalho

```
train = pd.read_csv("treino.csv")
test = pd.read_csv("teste.csv")
predict = pd.read_csv("predicao.csv")
```

Preparar os dados que serão usados no modelo

```
dados = dados[dados['gmv_d30'] > 0]
teste = teste[teste['gmv_d30'] > 0]
predicao = predicao[predicao['gmv_d30'] > 0]

dados = dados[dados["rate"] > 3]
teste = teste[teste["rate"] > 3]
predicao = predicao[predicao["rate"] > 3]

dados = dados[dados["pageviews_d30"] > dados["pageviews_d30"].median()]
teste = teste[teste["pageviews_d30"] > teste["pageviews_d30"].median()]
predicao = predicao[predicao["pageviews_d30"] >
predicao["pageviews_d30"].median()]
```

Calcular o score atual da empresa

```
scaler = MinMaxScaler()
scaled = dados[['pk_produto', "A", "B"]]
scaled.iloc[:, 1:3] = scaler.fit_transform(scaled.iloc[:, 1:3])
scaled['score'] = 0.4*scaled['A'] + 0.6*scaled['B']
scaled_dados = scaled[['pk_produto', 'score']]
dados = dados.merge(scaled_dados, how = 'inner', on='pk_produto')
scaled2 = teste[['pk_produto', "A", "B"]]
scaled2.iloc[:, 1:3] = scaler.fit_transform(scaled2.iloc[:, 1:3])
scaled2['score'] = 0.4*scaled2['A'] + 0.6*scaled2['B']
scaled_teste = scaled2[['pk_produto', 'score']]
teste = teste.merge(scaled_teste, how = 'inner', on='pk_produto')
scaled3 = predicao[['pk_produto', "A", "B"]]
scaled3.iloc[:, 1:3] = scaler.fit_transform(scaled3.iloc[:, 1:3])
scaled3['score2'] = 0.4*scaled3['A'] + 0.6*scaled3['B']
scaled_predicao = scaled3[['pk_produto', 'score']]
```

```
predicao = predicao.merge(scaled_predicao, how = 'inner', on='pk_produto')
```

Calcular a estimativa de relevância que será usada no modelo

```
dados = dados.sort_values('conversao', ascending=False)
dados.reset_index(drop = True, inplace = True)
teste = teste.sort_values('conversao', ascending=False)
teste.reset_index(drop = True, inplace = True)
predicao = predicao.sort_values('conversao', ascending=False)
predicao.reset_index(drop = True, inplace = True)
```

```
dados['percentual'] = dados[target].cumsum()/dados[target].sum()
teste['percentual'] = teste[target].cumsum()/teste[target].sum()
predicao['percentual'] = predicao[target].cumsum()/predicao[target].sum()
```

```
def checar_relevancia(percentual):
    if percentual <= 1 and percentual > 0.8:
        return 1
    elif percentual <= 0.8 and percentual > 0.6:
        return 2
    elif percentual <= 0.6 and percentual > 0.4:
        return 3
    elif percentual <= 0.4 and percentual > 0.2:
        return 4
    elif percentual <= 0.2:
        return 5
```

```
dados['relevancia'] = dados['percentual'].apply(checar_relevancia)
teste['relevancia'] = teste['percentual'].apply(checar_relevancia)
predicao['relevancia'] = predicao['percentual'].apply(checar_relevancia)
```

Separar os dados em treino e teste para serem usados no modelo

```
qids_train =
dados.groupby("data_processamento")["data_processamento"].count().to_numpy(
)
X_train = dados[['A', 'B']]
y_train = dados['relevancia'].astype(int)

qids_test =
teste.groupby("data_processamento")["data_processamento"].count().to_numpy(
)
X_test = teste[['A', 'B']]
y_test = teste['relevancia'].astype(int)
```

Configuração de parâmetros do modelo LambdaMART

```
ranker = lgb.LGBMRanker(
    objective="lambdarank",
    boosting_type = "gbdt",
```

```

n_estimators = 100,
importance_type = "gain",
metric= "ndcg",
num_leaves = 31,
learning_rate = 0.1,
max_depth=-1)

```

Treino e teste do modelo LambdaMART

```

ranker.fit(
    X=X_train,
    y=y_train,
    group=qids_train,
    eval_set=[(X_train, y_train), (X_test, y_test)],
    eval_group=[qids_train, qids_test],
    eval_at=[10, 25, 50, 75, 100])

```

Predição do modelo LambdaMART

```

X_predict = predicacao[['A', 'B']]
test_pred = ranker.predict(X_predict)
X_predict["predicted_ranking"] = test_pred
X_predict = X_predict.sort_values("predicted_ranking", ascending=False)

```

NDCG@k do modelo LambdaMART com a estimativa de relevância

```

avaliacao = X_predict.sort_values('relevancia', ascending=False)
avaliacao = avaliacao[avaliacao.index < 'k']
true_relevance = np.asarray([avaliacao['relevancia']]) #Ideal DCG
avaliacao_2 = X_predict.sort_values('predicted_ranking', ascending=False)
avaliacao_2 = avaliacao_2[avaliacao_2.index < 'k']
score_teste = np.asarray([avaliacao_2['relevancia']]) #DCG
ndcg_k_teste = ndcg_score(true_relevance, score_teste)

```

NDCG@k do método atual da empresa com a estimativa de relevância

```

avaliacao = X_predict.sort_values('relevancia', ascending=False)
avaliacao = avaliacao[avaliacao.index < 'k']
true_relevance = np.asarray([avaliacao['relevancia']]) #Ideal DCG
avaliacao_2 = X_predict.sort_values('score', ascending=False)
avaliacao_2 = avaliacao_2[avaliacao_2.index < 'k']
score_teste = np.asarray([avaliacao_2['relevancia']]) #DCG
ndcg_k_atual = ndcg_score(true_relevance, score_teste)

```