UNIVERSIDADE REGIONAL DO NOROESTE DO
ESTADO DO RIO GRANDE DO SUL
DEPARTAMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CIÊNCIA DA COMPUTAÇÃO

Gabriel Cavalheiro Ullmann

# Sistema de sugestão de produtos para e-commerce utilizando Inteligência Artificial

**Santa Rosa - RS**

**2020**

Gabriel Cavalheiro Ullmann

# Sistema de sugestão de produtos para e-commerce utilizando Inteligência Artificial

Trabalho de Conclusão de Curso do curso de graduação em Ciência da Computação apresentado ao Departamento de Ciências Exatas e Engenharias da Universidade Regional do Noroeste do Estado do Rio Grande do Sul como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Edson Luiz Padoin

Santa Rosa - RS

2020

Gabriel Cavalheiro Ullmann
Sistema de sugestão de produtos para e-commerce utilizando Inteligência Artificial/ Gabriel Cavalheiro Ullmann. – Santa Rosa - RS, 2020-
69p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Edson Luiz Padoin

Monografia – UNIVERSIDADE REGIONAL DO NOROESTE DO ESTADO DO RIO GRANDE DO SUL
DEPARTAMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CIÊNCIA DA COMPUTAÇÃO, 2020.

1. Sistemas de Recomendação. 2. Inteligência Artificial. 3. Machine Learning. I. Prof. Dr. Edson Luiz Padoin. II. Universidade Regional do Noroeste do Estado do Rio Grande do Sul. III. Título

Gabriel Cavalheiro Ullmann

# Sistema de sugestão de produtos para e-commerce utilizando Inteligência Artificial

Trabalho de Conclusão de Curso do curso de graduação em Ciência da Computação apresentado ao Departamento de Ciências Exatas e Engenharias da Universidade Regional do Noroeste do Estado do Rio Grande do Sul como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho aprovado. Santa Rosa - RS, 14 de dezembro de 2020:

---

**Prof. Dr. Edson Luiz Padoin**
Orientador

---

**Prof. Dr. Gerson Battisti**
Banca

Santa Rosa - RS
2020

*À minha mãe, Loreni e ao meu pai, Paulo*

# Agradecimentos

A meus pais, que sempre estimularam meu interesse por ciência e curiosidade sobre o mundo. Se pude seguir em frente sem medo de errar, foi graças a eles.

A meu orientador, Prof. Dr. Edson Luiz Padoin, pelo incentivo, confiança e disposição para ouvir, auxiliar e ensinar.

Aos colegas e amigos da GH Branding, empresa onde desenvolvi meu trabalho de estágio e onde nasceu a ideia que me conduziu a linha de estudo abordada neste trabalho.

Aos demais amigos e familiares que me acompanharam e auxiliaram, cada um à sua maneira, ao longo de minha jornada de aprendizado.

*"Your best and wisest refuge from all troubles is in your science."*
*(Ada Lovelace)*

# Resumo

O trabalho trata da implementação de um sistema de recomendação baseado em filtragem colaborativa, utilizando técnicas de Machine Learning e Deep Learning. Esse sistema, diferentemente de outros do tipo, não depende de avaliações numéricas dadas pelos clientes e infere a popularidade dos produtos com base em seu volume de vendas, bem como diferentes conjuntos de regras. Nesse estudo, os resultados de recomendação são analisados e comparados no que se refere a suas características estatísticas.

**Palavras-chave**: Sistemas de Recomendação. Inteligência Artificial. Machine Learning. Deep Learning. Ciência de Dados.

# Abstract

This work describes the implementation of a collaborative-filtering based recommendation system using Machine Learning and Deep Learning techniques. This system, unlike others of the same kind, does not depend on numerical ratings given by customers, instead inferring the popularity of the products based on their sales volume and different sets of rules. In this study, the recommendation results are analyzed and compared with regard to their statistical features.

**Keywords**: Recommendation Systems. Artificial Intelligence. Machine Learning. Deep Learning. Data Science.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| CPU | *Central Processing Unit* |
| ERP | *Enterprise Resource Planning* |
| FC | *Filtragem Colaborativa* |
| GPU | *Graphics Processing Unit* |
| IA | *Inteligência Artificial* |
| IBGE | *Instituto Brasileiro de Geografia e Estatística* |
| IDE | *Integrated Development Environment* |
| KDE | *Kernel Density Estimate* |
| LTS | *Long-Term Support* |
| ML | *Machine Learning* |
| MSE | *Mean Squared Error* |
| ONU | *Organização das Nações Unidas* |
| RAM | *Random Access Memory* |
| ReLu | *Rectified Linear Unit* |
| RNA | *Rede Neural Artificial* |
| UCI | *University of California, Irvine* |
| UF | *Unidade da Federação* |

# Sumário

# 1 Introdução

O Brasil é o país com maior faturamento em *e-commerce* na América Latina, faturando R$ 133 bi em 2018 (EBIT, 2019). No mundo, esse mercado somou US$ 2,8 tri em vendas no mesmo ano (ORENDORFF, 2019). Em 2020, devido à pandemia do novo Coronavírus, observou-se um aumento das vendas online em várias partes do mundo. Enquanto no Brasil 61% dos consumidores aumentaram seu volume de compras online devido ao isolamento social (SBVC, 2020), nos EUA a rede Walmart reportou um aumento de 97% nas vendas por *e-commerce* no segundo trimestre de 2020 (PEREZ, 2020).

De forma a se destacar em meio a milhares de lojas online, empresas do ramo estão buscando levar aos consumidores produtos e serviços que sejam mais relevantes em suas vidas diárias. Para cumprir esse objetivo, é fundamental entender os hábitos dos mesmos, e essa compreensão pode ser alcançada através da análise da informação que os empreendimentos possuem sobre as compras de seus clientes. Além das técnicas estatísticas convencionais, abordagens de Inteligência Artificial e *Machine Learning* têm se tornado populares recentemente para realizar esse tipo de inferência (RHEUDE, 2019).

Este projeto propõe a utilização de *Machine Learning* e *Deep Learning* para inferir as preferências de consumidores e fazer recomendações de produtos com base em um histórico de compras. Essas sugestões são geradas com base em uma métrica de avaliação, que por sua vez é inferida a partir do volume de vendas dos produtos contidos na base de dados. Clientes são correlacionados por um atributo comum - o país ou região de origem - de forma que consumidores que residem nos mesmos locais tendem a receber recomendações semelhantes.

As sugestões podem chegar ao cliente das mais diferentes formas, seja por *e-mail*, mídias sociais ou notificações dentro da loja virtual. Contudo, esse trabalho não foca na implementação prática e visual de uma aplicação de *e-commerce* mas sim na seleção e análise de dados necessária para gerar sugestões a serem consumidas por uma aplicação desse tipo.

Foram utilizados neste estudo 3 *datasets*, todos compilações de pedidos vendidos através de plataformas de *e-commerce*. Esses conjuntos de dados foram encontrados em repositórios públicos como Kaggle e UCI *Machine Learning*.

A questão central desta pesquisa é a seguinte: é possível inferir as preferências de compra de um consumidor utilizando redes neurais, mesmo sem que haja avaliação direta do consumidor quanto a sua satisfação com a compra realizada? Foram levantadas também questões secundárias, tais como:

1. É possível gerar avaliações que representem as preferências de um grupo de clientes tendo como base apenas 1 característica do mesmo, bem como seu histórico de compras?

2. Como os diferentes métodos de geração de avaliações interferem nas recomendações geradas?

3. Qual tipo de rede neural e hiperparâmetros são necessários para resolver esse tipo de problema?

A recente popularização de técnicas de Inteligência Artificial e *Machine Learning* e sua ampla implementação em aplicações empresariais justifica o foco deste estudo, visto que torna-se relevante no presente momento o desenvolvimento de um modelo de análise e recomendação de produtos que possa ser largamente utilizado em sistemas voltados para a área de vendas.

Diferentemente de trabalhos como os de CHEN et al. (2018), LI; LU; XUEFENG (2005), LI; WU; ZHANG (2019), MELVILLE; MOONEY; NAGARAJAN (2002), que criam sistemas de recomendação baseados em avaliações de produtos definidas diretamente pelos usuários, esse trabalho visa realizar recomendações através de *datasets* desprovidos de avaliações diretas e busca portanto inferí-las com base em outras variáveis. *Datasets* desse tipo podem ser encontrados em repositórios públicos na Internet e são provenientes de sistemas de controles de vendas que não permitem avaliação por parte dos usuários.

Este trabalho busca atingir os seguintes objetivos:

1. Obtenção de *datasets* de pedidos (*Big Data*) que não contenham avaliações de clientes, somente dados básicos sobre o produto comprado e o comprador.

2. Definição de métodos para inferir as avaliações de clientes a partir de métricas estatísticas.

3. Processamento dos *datasets* originais de forma a relacioná-los com as avaliações inferidas para cada caso.

4. Criação de uma rede neural em Python utilizando a biblioteca Keras.

5. Treinamento e teste da rede neural com diferentes parametrizações, a fim de definir a configuração mais adequada para os problemas em questão.

6. Aplicação da IA treinada para gerar recomendações de produtos a partir dos *datasets* selecionadas.

Seis partes compõem este estudo. A Seção 1 discorre de forma geral sobre o problema estudado e a solução proposta para o mesmo. Na Seção 2 são descritas áreas de conhecimento, ferramentas e estudos acadêmicos relacionados. A Seção 3 define a metodologia utilizada para geração das avaliações, e a Seção 4 explica o desenvolvimento da rede neural a ser alimentada por essas avaliações. Finalmente, uma análise detalhada dos resultados, bem como considerações finais e ideias para expansão do trabalho são descritas nas Seções 5 e 6, respectivamente.

# 2 Estado da arte

Esse capítulo tem por objetivo apresentar os principais conceitos abordados neste trabalho. Nas Seções 2.1 a 2.3 são apresentados as áreas de conhecimento e ferramentas de software relacionadas a aplicabilidade de Inteligência Artificial e suas sub-áreas.

Nas últimas décadas vários trabalhos foram publicados nesse campo de pesquisa, demonstrando as mais diversas aplicações práticas dessas tecnologias. Na Seção 2.7 estão dispostos esses trabalhos, de forma a demonstrar quais abordagens foram adotadas pelos autores para estudo das áreas em questão.

No âmbito da aplicação de IA para geração de recomendações em aplicações de *e-commerce*, foco principal deste trabalho, o mercado mostra que essa é uma prática que veio para ficar. Isso ocorre pois essa tecnologia traz benefícios tanto para quem a aplica quanto para o usuário-alvo. Ao utilizá-la, os comerciantes desfrutam de um volume de vendas maior, visto que conseguem levar com mais assertividade seus produtos e serviços aos consumidores que estão interessados em adquiri-los. Os consumidores, da mesma forma, podem desfrutar de uma experiência mais agradável ao utilizar o *e-commerce*, recebendo recomendações que lhe agradem e não apenas uma torrente de propagandas sem conexão com suas atividades cotidianas e hábitos de consumo.

Contudo, técnicas de IA trazem benefícios no dia a dia das pessoas em muitas outras áreas além do comércio, e frequentemente de maneira mais indireta. No campo das Ciências Sociais, por exemplo, já existem aplicações de IA na área do Direito, que permitem a advogados "trabalhar de forma mais eficiente e ampliar suas áreas de expertise" (ALARIE; NIBLETT; YOON, 2018, tradução nossa) e na área financeira para "prever relatórios financeiros fraudulentos e crises financeiras" (KUMAR; TAN, 2004, tradução nossa).

Há também algoritmos que se dedicam à análise de conteúdo textual, identificando casos de *fake news* na Internet (S; CHITTURI, 2020) ou até atuando como "jornalistas robôs"capazes de "resumir artigos científicos e transformá-los em *press releases* e matérias jornalísticas simples" (TATALOVIC, 2018, tradução nossa).

No campo das Engenharias, IA já é utilizada no design otimizado de sistemas espaciais ligados por cabos (REN; CHEN, 2019), para promover reuso de componentes de software e realizar análise de padrões de código (WANGOO, 2018) e avaliar lances em licitações de construção civil (CHUAN et al., 2011).

Em relação às Ciências da Vida, na Medicina, a IA Watson desenvolvida pela empresa IBM foi utilizada com sucesso para "acelerar a identificação de novos candidatos a medicamentos e medicamentos-alvo através da exploração do potencial do *Big*

*Data*" (CHEN; ARGENTINIS; WEBER, 2016, tradução nossa). Na agricultura de precisão, aplicações de identificação de imagem com IA foram utilizadas para "tratar aspectos relacionadas a detecção de doenças, qualidade do grão e fenotipagem" (PATRÍCIO; RIEDER, 2018, tradução nossa).

Outro foco popular da área de IA é o desenvolvimento de *chatbots*, agentes artificiais que podem atuar como interlocutores em uma conversação com um humano, auxiliar na aquisição de informações e oferecer respostas para perguntas (WANG; SIAU, 2018). Eles podem ser aplicados em áreas tais como atendimento ao consumidor (ADAM; WESSEL; BENLIAN, 2020) e Educação (KIMNA-YOUNG, 2019).

## 2.1 Conceitos e áreas de conhecimento

### 2.1.1 Inteligência Artificial

Segundo LAROUSSE (1999), IA é um "conjunto de teorias e de técnicas empregadas com a finalidade de desenvolver máquinas capazes de simular a inteligência humana".

De acordo com NIKOLOPOULOS (1997, tradução nossa):

> A Inteligência Artificial é uma área de estudos da computação que se interessa pelo estudo e criação de sistemas que possam exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência que é equivalente ou superior ao de um especialista humano.

Segundo John McCarthy, pioneiro que cunhou o termo Inteligência Artificial em 1955, o objetivo dessa área é "desenvolver máquinas que se comportam como se fossem inteligentes" (ERTEL, 2017, tradução nossa). Contudo, visto que não há consenso quanto ao que poderia ser considerado comportamento inteligente, essa definição é incompleta. A visão sobre o que é IA muda, dependendo do momento histórico considerado, o que é descrito por MCCORDUCK (2004, p.204, tradução nossa):

> É parte da história da área de Inteligência Artificial o fato de que toda vez que alguém descobre como fazer um computador fazer algo - jogar xadrez bem, resolver problemas simples mas relativamente informais - há um coro de críticos para dizer, 'isso não é raciocínio'.

Dessa forma, algumas definições de IA preferem abordar a implementação da tecnologia do ponto de vista do desenvolvimento de *software*. Conforme WATERMAN (1985, tradução nossa):

> A Inteligência Artificial é uma sub-área da Ciência da Computação que objetiva desenvolver programas computacionais inteligentes. Esses programas são: solucionadores de problemas, programas que melhoram sua

> própria performance, programas que interpretam linguagens, programas que reconhecem esquemas visuais, enfim que se comportam de maneira que seria considerada inteligente se observada num ser humano.

De forma semelhante, segundo PEREIRA (1988, p.2):

> A Inteligência Artificial é uma disciplina científica que utiliza as capacidades de processamento de símbolos da computação com o fim de encontrar métodos genéricos para automatizar actividades perceptivas, cognitivas e manipulativas, por via do computador.

Finalmente, de forma mais abrangente, RICH; KNIGHT (1993) descreve IA como "o estudo de como fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor".

### 2.1.2 Redes Neurais

Segundo ERTEL (2017, tradução nossa) redes neurais são "redes de células no cérebro de humanos e animais". Embora essa seja a definição literal do termo, quando o relacionamos com IA esse limita-se mais precisamente a descrever o processo de representação formal e reprodução artificial do funcionamento dos neurônios orgânicos. Como discorre o autor supracitado, "a partir do conhecimento do funcionamento de redes neurais naturais, tentamos modelá-las, simulá-las e até mesmo reconstruí-las em *hardware*".

A replicação das redes neurais naturais de forma artificial se dá através do entendimento que o ser humano possui atualmente sobre o funcionamento de seu cérebro. Como descreve RAUBER (2020, p.3):

> (...) a pesquisa tenta entender o funcionamento da inteligência residente nos neurônios e mapeá-la para uma estrutura artificial, por exemplo uma combinação de *hardware* e *software*, assim transformando as redes neurais biológicas em redes neurais artificiais.

Similarmente, HAYKIN (2001, tradução nossa) define que:

> "Na sua forma mais geral, uma rede neural é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse; a rede, normalmente, é implementada utilizando-se componentes eletrônicos ou é simulada por programação em um computador digital."

De forma mais pragmática, OSORIO; JOÃO; BITTENCOURT (2020) define que "as Redes Neurais Artificiais (RNAs) são ferramentas de Inteligência Artificial que possuem a capacidade de se adaptar e de aprender a realizar uma certa tarefa, ou comportamento, a partir de um conjunto de exemplos dados".

### 2.1.3 Machine Learning

Segundo BEZERRA (2019) "*Machine learning* é um nome genérico dado a um conjunto de métodos para análise de dados desenvolvidos com o intuito de fazer previsão e classificação".

Conforme MITCHELL (1997, tradução nossa) "*Machine Learning* é uma sub-área da Inteligência Artificial que diz respeito a questão de como construir programas de computador que melhoram automaticamente através da experiência". Através de um processo de treinamento com diversas iterações um programa pode encontrar relações entre dados contidos em um modelo.

Quantos aos principais tipos de algoritmos de *Machine Learning*, STIMPSON; CUMMINGS (2014, tradução nossa) descreve:

> "*Machine learning* (ou *data mining*) é uma ramificação da inteligência artificial que foca em algoritmos que identificam e aprendem as relações entre dados. Esses algoritmos, geralmente, são categorizados como não-supervisionados, que tentam identificar a estrutura intrínseca nos dados, e supervisionados, que inferem uma função para relacionar dados a uma variável 'alvo'."

Quanto ao processo de inferência realizado por esses algoritmos, ALPAYDIN (2018, tradução nossa) descreve de forma sucinta:

> "De forma geral, nossa abordagem é começar com um modelo bem generalizado com muitos parâmetros, e esse modelo geral pode fazer todo tipo de tarefa dependendo de como seus parâmetros estão definidos. Aprender corresponde a ajustar os valores desses parâmetros de forma que o modelo coincida da melhor forma com os dados vistos durante o treinamento. Baseado nos dados de treinamento, o modelo generalizado, através de uma configuração particular de seus parâmetros, torna-se especializado na tarefa distinta que encontra-se entremeada aos dados. A versão do modelo que obtemos após o treinamento, a instanciação específica do modelo padrão, torna-se o algoritmo para a tarefa."

### 2.1.4 Deep Learning

Segundo SKANSI (2018, tradução nossa), "considerando a visão mais simples possível, *deep learning* é o nome de uma classe específica da redes neurais artificiais, que por sua vez são uma classe especial de algoritmos de *Machine Learning*, aplicáveis a processamento de linguagem natural, *computer vision* e robótica."

Skansi também discorre sobre qual seria a maneira mais adequada de classificar o estudo de *Deep Learning* em relação a outras áreas de conhecimento relacionadas:

> "Um número crescente de áreas da IA como raciocínio e planejamento, outrora bastiões da IA lógica (também chamada de *Good Old-Fashioned*

> *AI*, ou GOFAI), estão sendo agora abordados de forma bem sucedida pelo *deep learning*. Nesse sentido, pode-se dizer que *deep learning* é uma abordagem de IA, e não apenas uma sub-área de uma sub-área da IA."

De maneira mais formal e relacionada com o funcionamento de redes neurais, LECUN (2015, tradução nossa) descreve que um algoritmo de *Deep Learning* é capaz de mapear a estrutura intrínseca em um conjunto de dados "utilizando-se do algoritmo de *backpropagation* para indicar como uma máquina deveria alterar seus parâmetros internos que são utilizados para computar a representação em cada camada a partir da representação na camada anterior".

Segundo AGGARWAL (2018, tradução nossa):

> "A ideia base do *deep learning* é de que a repetida composição de funções pode frequentemente reduzir os requerimentos no número de funções base (unidades computacionais) por um fator que é exponencialmente relacionado ao número de camadas da rede. Portanto, embora o número de camadas em uma rede aumente, o número de parâmetros requeridos para aproximar a mesma função reduz drasticamente. Isso aumenta o poder de generalização da rede. A ideia por trás das arquiteturas profundas é de que elas podem identificar melhor irregularidades repetidas em padrões de dados de forma a reduzir o número de unidades computacionais e portanto generalizar o aprendizado até mesmo em área do espaço amostral nos quais não existem exemplos."

## 2.2 Ambientes de desenvolvimento

Existem vários ambientes de desenvolvimento para trabalhar com RNAs em diferentes linguagens. Para fins de contextualização, será realizada aqui uma breve comparação entre ambientes para a linguagem Python.

O Spyder é uma IDE de código aberto escrita em Python. Conforme o site oficial da ferramenta, ela oferece "uma combinação única de funcionalidades de edição avançada, análise, depuração e *profiling* que podem ser encontradas em uma ferramenta de desenvolvimento completa com as capacidades de exploração de dados, execução interativa, inspeção profunda e visualização agradável de um pacote científico" (SPYDER, 2020).

O Jupyter Notebook é uma IDE de código aberto para as linguagens Python, Julia e R desenvolvida pela Project Jupyter. Diferentes de outros ambientes do mesmo tipo, o Jupyter é uma aplicação *web* que roda localmente e que possui um *front-end* acessível pelo usuário através do navegador. Através da ferramenta é possível criar e compartilhar documentos que possuem código executável, comentários com texto formatado, imagens e outros recursos visuais. Além de *Machine Learning*, a ferramenta pode ser utilizada também para simulação numérica, limpeza, transformação e visualização de dados (JUPYTER, 2020).

O PyCharm é uma IDE de código fechado desenvolvida pela empresa JetBrains. O ambiente fornece várias funcionalidades focadas em tornar o processo de desenvolvimento mais ágil e aumentar a produtividade, tais como completamento de código automático, checagem de erros em tempo real e recomendações de código baseadas no PEP8, manual de boas-práticas do Python. O PyCharm possui duas versões: a *Community*, que é gratuita mas tem funcionalidades limitadas, e a *Professional*, que não possui limitações mas é paga (PYCHARM, 2020).

O Visual Studio Code, uma IDE de código aberto desenvolvida pela Microsoft, é atualmente um dos ambientes de desenvolvimento mais amplamente utilizados no mundo (STACKOVERFLOW, 2020). Oferece suporte a uma grande variedade de linguagens, incluindo Python, bem como integração com ferramentas de depuração, versionamento e extensões produzidas pela mantenedora do software ou colaboradores (MICROSOFT, 2020).

## 2.3 Bibliotecas para Machine Learning

É possível construir e treinar redes neurais em qualquer linguagem de programação sem necessidade de instalação de bibliotecas ou *frameworks*. Contudo, nesse caso é necessário escrever a implementação de algoritmo para inicializar pesos e *biases*, funções de ativação, custo e *backpropagation* com base em suas definições matemáticas formais (PEIXEIRO, 2019), o que torna o desenvolvimento de uma aplicação completa mais lento e trabalhoso, exigindo profundo conhecimento teórico de IA por parte do programador.

Entretanto, principalmente no âmbito do desenvolvimento de *software* comercial, agilidade e facilidade de uso e manutenção são pontos fundamentais. Portanto é comum que pesquisadores e programadores busquem ferramentas que tornem o processo mais acessível tanto para iniciantes quanto profissionais experientes na área. Para fins de contextualização, será realizada aqui uma breve comparação entre bibliotecas e *frameworks* de IA e ML para a linguagem Python.

O Keras é uma biblioteca de código aberto para criação de redes neurais escrita em Python. Ela atua como uma API, uma interface consistente que permite ao usuário utilizar funções de mais baixo-nível descritas em bibliotecas como TensorFlow e Theano (BROWNLEE, 2019). Entre essas funções estão operações com vetores multi-dimensionais de tipo único, também chamados de tensores, utilizados na representação de RNAs (TENSORFLOW, 2020b).

Segundo o site oficial, a biblioteca oferece "APIs simples e consistentes, minimiza o número de ações de usuário necessárias para casos de uso comuns e provém *feedback* claro e prático em caso de erros por parte do usuário" (KERAS, 2020a, tradução nossa). O Keras se integra também com outras ferramentas, incluindo as do ecossistema TensorFlow.

O TensorFlow é uma biblioteca de código aberto que suporta linguagens como Python, JavaScript e Swift. Segundo a documentação oficial, a biblioteca possui "um completo e flexível ecossistema de ferramentas e recursos de comunidade que permitem que pesquisadores avancem o estado da arte do ML e que desenvolvedores construam e implantem facilmente aplicações com funcionalidades de ML" (TENSORFLOW, 2020a, tradução nossa). Esse ecossistema engloba ferramentas como o TensorFlow Cloud, que permite configurar o treinamento de RNAs em servidores na nuvem, e o TensorFlow Extended (TFX), *pipeline* para implantação de aplicações de *Machine Learning*.

O Theano é uma biblioteca matemática escrita em Python. Segundo a documentação oficial, a biblioteca "permite que você defina, otimize e avalie expressões matemáticas envolvendo vetores multi-dimensionais de forma eficiente" (THEANO DEVELOPMENT TEAM, 2020, tradução nossa). Essas funcionalidades são utilizadas por outras bibliotecas como o Keras, que criam abstrações para tornar mais simples a construção de redes neurais mas que para atingir esse objetivo precisam de bibliotecas de apoio (THEANO DEVELOPMENT TEAM, 2016).

O PyTorch é uma biblioteca escrita em Python baseada na biblioteca Torch, originalmente escrita em Lua. A biblioteca provém funções para definição de funções matemáticas e computação de seus respectivos gradientes, bem como funcionalidades para processamento tanto em CPU quanto em GPU (KETKAR, 2017).

O XGBoost é uma biblioteca que permite a implementação de algoritmos de *Machine Learning* através de um *framework* de *Gradient Boosting*, que consiste "em um procedimento de aprendizado que consecutivamente ajusta novos modelos para prover uma estimativa mais precisa da variável de resposta" (NATEKIN; KNOLL, 2013, tradução nossa). Segundo pesquisa conduzida pela equipe do Keras, a XGBoost é a terceira ferramenta de ML mais utilizada por equipes que ficaram no top 5 das competições do site Kaggle em 2019 (KERAS, 2020d).

O LightGBM é um *framework* de *gradient boosting* escrito em Python mas com suporte para as linguagens C e R. Suporta também processamento em CPU e GPU (LIGHTGBM, 2020). Segundo pesquisa conduzida pela equipe do Keras, a XGBoost é a segunda ferramenta de ML mais utilizada por equipes que ficaram no top 5 das competições do site Kaggle em 2019 (KERAS, 2020d).

## 2.4 Sistemas de recomendação

Visto que o objetivo principal do trabalho é criar um sistema de recomendação, essa seção irá descrever 3 abordagens populares para criação desse tipo de sistema: filtragem baseada em conteúdo, filtragem colaborativa e sistemas híbridos (FRESSATO, 2019). Outros métodos têm sido historicamente utilizados para a criação de sistemas de

recomendação, tais como redes Bayesianas, clusterização (LI; LU; XUEFENG, 2005) e a técnica denominada *Horting*, baseada na teoria dos grafos (AGGARWAL et al., 1999).

### 2.4.1 Filtragem baseada em conteúdo

Sistema que "recomenda itens com base em seus atributos" (FRESSATO, 2019). Um algoritmo que implementa essa abordagem correlaciona somente as características dos produtos comprados por um cliente, e como resultado recomendará outros produtos com características semelhantes. Como explica ACIAR et al. (2007, tradução nossa):

> (...) Métodos baseados em conteúdo fazem recomendações através da análise da descrição dos itens que foram avaliados pelo usuário. Uma variedade de algoritmos tem sido proposta para analisar o conteúdo de documentos, e encontrar padrões nesse conteúdo pode servir como base para recomendações.

As características desejadas são inferidas pelo sistema com base nas interações entre cada usuário e os produtos no catálogo da plataforma de *e-commerce*, como exemplifica BURKE (2002, tradução nossa):

> Outro paradigma de recomendação relevantes além de bases de conhecimento e recomendação colaborativa é o de recomendação baseada em conteúdo, na qual o sistema aprende um classificador para cada usuário baseado nas características de produtos curtidos e não curtidos.

Sistemas de recomendação baseados em conteúdo possuem fatores que limitam a quantidade e qualidade de suas recomendações, tais como:

- A análise dos produtos pode acabar sendo rasa, especialmente quando não há muita informação sobre os itens ou essa informação está em formatos não-textuais (ex.: filmes, música, locais geográficos) (BALABANOVIĆ; SHOHAM, 1997).
- A análise pode não ser satisfatória quando a descrição do produto é de difícil análise pelo computador, por exemplo, textos que envolvem opinião pessoal e discussão de ideias (MELVILLE; MOONEY; NAGARAJAN, 2002).
- Mesmo no caso de produtos que possuem descrição textual, essa nem sempre abrange todos os aspectos do produto (BALABANOVIĆ; SHOHAM, 1997).

### 2.4.2 Filtragem colaborativa

Um algoritmo de filtragem colaborativa "recomenda itens de acordo com o comportamento de usuários similares" (FRESSATO, 2019). O que define essa similaridade depende do contexto que estamos analisando. Se os usuários são pessoas físicas, características

como faixa etária, gênero e localização geográfica podem ser levadas em consideração. No caso de pessoas jurídicas, segmento de atuação e tamanho podem ser analisados.

De forma similar, CHEN et al. (2018, tradução nossa) define que a abordagem "faz recomendações para o usuário atualmente ativo utilizando vários históricos de avaliação de outros usuários sem analisar o conteúdo do recurso informacional". Quando cita "recurso informacional" o autor se refere ao produto, pois essa abordagem permite inferir sugestões sem que seja necessário conhecer dados do produto.

KIM; KIM (2001, tradução nossa) faz um paralelo com a abordagem baseada em conteúdo:

> Um sistema de recomendação baseado em conteúdo sugere produtos para consumidores analisando o conteúdo dos itens pelos quais eles se interessaram no passado. (...) Em contraste, a técnica de filtragem colaborativa recomenda itens pelos quais consumidores similares se interessaram.

Embora popularmente utilizados em aplicações de *e-commerce*, sistemas de FC frequentemente sofrem dos seguintes problemas:

- Partida fria (*cold start*): o sistema é incapaz de gerar recomendações precisas para novos usuários do sistema, visto que esses não possuem um histórico de avaliação ou compra de produtos (FRESSATO, 2019).
- Escalabilidade: o processamento de milhares ou milhões de produtos/clientes em um banco de dados de comércio eletrônico demanda poder computacional e tempo (LEE; YANG; PARK, 2004).
- Dispersão: visto que um *e-commerce* pode conter milhões de produtos, alguns desses itens serão pouco visualizados ou comprados pelos clientes. Os sistemas de recomendação deve gerar sugestões balanceadas, evitando que itens sejam ignorados (LEE; YANG; PARK, 2004).

### 2.4.3 Filtragem híbrida

São sistemas que combinam abordagens baseadas em conteúdo com filtragem colaborativa, com o objetivo de compensar as limitações que ocorrem ao se utilizar essas abordagens individualmente. Sua implementação pode incorporar ainda outras técnicas além das já mencionadas (FRESSATO, 2019).

Uma implementação prática de sistema híbrido é descrita por BALABANOVIĆ; SHOHAM (1997):

> Para criar um híbrido entre um sistemas colaborativos e baseados em conteúdo, nós mantivemos perfis de usuários baseados em análise de

> conteúdo, e diretamente comparamos esses perfis a fim de definir similaridade para recomendação colaborativa. Usuários recebem recomendações tanto de itens bem avaliados em relação ao próprio perfil quanto ao perfil de outros usuários similares.

LI; LU; XUEFENG (2005, tradução nossa) afirma que sistemas híbridos podem ser úteis também em casos nos quais sistemas de FC funcionam, porém, não produzem resultados satisfatórios, como em cenários onde deseja-se considerar múltiplos interesses de um usuário:

> O que pouquíssimos trabalhos mostram é que a filtragem colaborativa clássica não é adaptativa à recomendação Multi-interesse. Na verdade, a qualidade dessas recomendações é muito baixa quando usuários em sistemas de recomendação tem interesses completamente diferentes.

## 2.5 Métricas

Essa seção visa descrever métricas que são utilizadas para entender a estrutura de um conjunto de dados ou medir o sucesso de um modelo de *Machine Learning*. Nomenclaturas e siglas aqui utilizadas serão baseadas no conteúdo da documentação do Keras, visto que essa será a biblioteca a ser utilizada no trabalho.

### 2.5.1 Perda

As funções de perda "computam a quantidade que um modelo deveria minimizar durante o treinamento" (KERAS, 2020c). Enquanto para modelos de regressão a função MSE é a mais popular (NHU et al., 2020), para modelos de classificação com múltiplas categorias destaca-se a utilização de *categorical cross-entropy*, ou *binary cross-entropy* quando há somente 2 categorias (RUSIECKI, 2019). A métrica é representada no Keras por seu nome em inglês: *loss*.

### 2.5.2 Dispersão

Segundo MICHAELIS (2020), a palavra dispersão pode ser definida como "maneira com que os indivíduos de uma mesma população se acham distribuídos" ou "oscilação apresentada por uma variável aleatória".

No contexto de *Machine Learning*, um conjunto de dados é considerado disperso quando os valores nele contidos são predominantemente correspondentes a zero ou vazio. Em oposição a esse conceito, conjuntos densos são aqueles nos quais a maioria dos valores são diferentes de zero (GOOGLE DEVELOPERS, 2020).

### 2.5.3 Coeficiente de Correlação de Pearson

O coeficiente de correlação de Pearson é "a medida da associação linear entre duas variáveis" (KIRCH, 2008, tradução nossa), resultado do trabalho dos matemáticos ingleses Karl Pearson e Francis Galton (FIGUEIREDO FILHO; SILVA JUNIOR, 2010).

Figura 1 – Fórmula para o coeficiente de correlação de Pearson

$$r = \frac{1}{n-1}\sum\left(\frac{xi-\overline{X}}{Sx}\right)\left(\frac{yi-\overline{Y}}{Sy}\right)$$

Fonte: adaptado de FIGUEIREDO FILHO; SILVA JUNIOR (2010)

Os valores obtidos através da fórmula da Figura 1 indicam a força da correlação: valores mais próximos de zero indicam associação fraca entre as variáveis, enquanto resultados próximos de -1 ou 1 indicam forte relação direta ou inversa, dependendo do sinal (AKOGLU, 2018). Há várias interpretações dos coeficientes, mas nesse trabalho será utilizada a abordagem de Dancey & Reidy, descrita na Tabela 1.

Tabela 1 – Interpretação de Dancey & Reidy para a correlação de Pearson

| **Coeficiente** | **Correlação** |
|---|---|
| 1 | Perfeita |
| 0,7 a 0,9 | Forte |
| 0,4 a 0,6 | Moderada |
| 0,1 a 0,3 | Fraca |
| 0 | Zero |

Fonte: adaptado de AKOGLU (2018, tradução nossa)

## 2.6 Métodos de codificação

Métodos comuns para representação codificada de dados no contexto de *Machine Learning* serão descritos nesta seção. Técnicas desse tipo são utilizadas com o objetivo de simplificar e padronizar a representação de grandes vetores de valores.

### 2.6.1 One-hot encoding

Ao aplicar-se o método de one-hot encoding aos elementos de um vetor, cada item passa a ser representado por um novo vetor de números inteiros no qual uma das posições corresponde ao valor 1, enquanto o valor 0 ocupa as demais. O tamanho total e a posição definida como 1 na lista gerada pela codificação é relativa posição do item na lista original.

O uso desse método é exemplificado com uma aplicação prática por GOOGLE DEVELOPERS (2020, tradução nossa):

> Suponhamos que um dado *dataset* de botânica descreva 15.000 diferentes espécies, cada uma indicada por uma *string* identificadora única. Como parte da engenharia de parâmetros, você provavelmente codificaria esses identificadores como vetores *one-hot*, os quais teriam tamanho 15.000.

### 2.6.2 Embedding

Método de codificação utilizado como alternativa ao *one-hot encoding*. GOOGLE DEVELOPERS (2020, tradução nossa) define *embedding* como "uma característica categórica representada como um valor contínuo"e "a tradução de um vetor de alta dimensionalidade para um espaço de baixa dimensionalidade". O mesmo autor exemplifica a aplicação prática do método:

> Você poderia representar as palavras em uma frase da Língua Inglesa de duas maneiras: - Como um vetor esparso de milhões de elementos (alta dimensionalidade) no qual todos são valores inteiros. (...) - Como vários vetores densos de centenas de posições nos quais cada posição corresponde a um valor de ponto flutuante entre 0 e 1. Isso é um *embedding*.

## 2.7 Trabalhos relacionados

No contexto de *e-commerce*, há trabalhos acadêmicos descrevendo diferentes abordagens, tais como sistemas de filtragem colaborativa que predizem avaliações por meio de fatorização de matrizes (HE et al., 2017), representam relações cliente/item utilizando grafos bipartidos (WANG et al., 2019) e consideram fatores como críticas do usuário às recomendações geradas (BURKE, 2002), sazonalidade e intenção de compra (HWANGBO; KIM; CHA, 2018).

Em sistemas baseados em conteúdo ou com abordagens híbridas, é comum a utilização de resenhas escritas por usuários sobre os produtos como base para geração de sugestões, seja focando apenas na informação textual em si (SHOJA; TABRIZI, 2019) ou relacionando com outros dados sobre os produtos em questão, tais como imagens (WU; ZHAO; CUI, 2020).

Ainda no contexto de sistemas híbridos, há uma variedade de abordagens utilizadas para correlacionar produtos e clientes, tais como análise conjunta de múltiplos *datasets* e integração com dados provenientes de redes sociais (LI; LIU; HUANG, 2016), cadeias de Markov (YANG; JANG; KIM, 2020), uso de agentes inteligentes baseados em lógica *fuzzy* (YAGER, 2000), sistemas multi-agente (ACIAR et al., 2007), entre outras.

No presente trabalho, em vez de trabalhar com avaliações numéricas dadas pelos usuários, as recomendações serão produzidas com base somente na frequência da ocorrência

de pedidos para um produto em um determinado segmento de clientes. Isso permite que recomendações sejam geradas mesmo em aplicações que não coletam avaliações de clientes, ou em ERPs que possuem cadastros de pedidos simples, contando apenas com informações básicas do cliente, produto e transação, mas sem dados sobre a satisfação do consumidor no contexto pós-venda.

Conforme definido na Seção 1, o trabalho tem como objetivo também a proposta de métodos de avaliação e a posterior comparação e análise estatística das recomendações geradas através desses métodos. Ou seja, diferente dos trabalhos apresentados nesta seção, o estudo vai além da construção de uma estrutura através de uma técnica, visto que busca também avaliar se a estrutura resultante está em conformidade com a proposta inicial, bem como extrair *insights* dos *datasets* no tocante à vendas e comportamento de grupos de consumidores.

## 2.8 Considerações do capítulo

Neste capítulo foram apresentados áreas de conhecimento, tecnologias, técnicas e trabalhos acadêmicos que possuem relação com o tema abordado no trabalho. No próximo capítulo serão apresentados os métodos e ferramentas a serem utilizadas.

# 3 Materiais e Métodos

Neste capítulo serão delimitados detalhes relativos à metodologia empregada para o desenvolvimento do trabalho, bem como especificidades do ambiente e ferramentas de *software* a serem utilizadas.

## 3.1 Metodologia utilizada

Similarmente ao descrito no trabalho de BURKE (2002), a proposta deste trabalho será desenvolver uma aplicação que gere recomendações com base em transações comerciais passadas, mas sem que haja a necessidade de que o comprador informe o tipo de produto que está buscando ou por qual motivo a compra ou pesquisa está sendo efetuada. A abordagem escolhida para atingir esse objetivo consiste no desenvolvimento de um sistema de recomendação por filtragem híbrida. Como descrito na Seção 2.4, esse tipo de sistema integra características de diferentes abordagens, visando mitigar os problemas específicos de cada uma.

Inicialmente, será realizada a seleção da base de pedidos a ser utilizada para o treinamento da IA. Serão utilizados repositórios online como Kaggle, GitHub e UCI ML para buscar conjuntos de dados. As bases escolhidas devem conter dados que permitam identificar clientes ou produtos de forma simples (ex.: nome, marca, categoria) e um código único para identificar essas entidades.

Após a seleção, serão definidos métodos que permitam inferir a popularidade de um produto em relação a um determinado grupo de clientes, a ser expressa por uma avaliação numérica. Estes procedimentos são baseados na frequência de pedidos em diferentes agrupamentos de clientes e nas definições para filtragem colaborativa no contexto de sistemas de recomendação:

- **FC por Categoria**: se um produto possui alta frequência de pedido dentro de uma categoria, mas um cliente daquela categoria nunca comprou o item, uma avaliação alta será definida para esse par de cliente/produto. Avaliações mais baixas serão definidas caso o cliente em questão já tenha comprado um produto, independentemente deste ser popular ou não. Essa regra, além de evitar que o cliente receba sugestões de produtos já adquiridos, favorece a recomendação de itens que possa gostar considerando que estes são populares em clientes que são da mesma categoria e com os quais, portanto, compartilha alguma similaridade (Figura 2).
- **FC Geral**: segue as mesmas regras do FC por Categoria, porém considerando a

frequência total de pedidos, sem distinção de categoria. Se um produto tem alta frequência de pedido no contexto geral, mas um cliente daquele conjunto nunca comprou o item, uma avaliação alta será definida para esse par de cliente/produto. Essa regra também evita a repetição de sugestões de itens já adquiridos, além de buscar um equilíbrio na recomendação de produtos com alta e baixa frequência de pedido (Figura 3).

Figura 2 – Geração de avaliações para FC por Categoria

| Produto/Cliente | Já comprou | Nunca comprou |
|---|---|---|
| É popular na categoria | 2 | 4 |
| Não é popular na categoria | 1 | 3 |

Fonte: Autor

Figura 3 – Geração de avaliações para FC Geral

| Produto/Cliente | Já comprou | Nunca comprou |
|---|---|---|
| É popular no geral | 2 | 4 |
| Não é popular no geral | 1 | 3 |

Fonte: Autor

Após a elaboração de uma definição textual, a lógica dos métodos de avaliação foi descrita em *scripts* Python. Cada *script* contém instruções para percorrer todos os registros em cada um dos *datasets* selecionados e analisar os 10 produtos mais vendidos de acordo com uma categorização de cliente. No caso dos *datasets* provenientes do repositório UCI ML, decidiu-se categorizar os clientes com base na coluna *country* (país de origem). No caso do conjunto do Kaggle, que aborda transações exclusivamente realizadas no Brasil, a coluna *customer_state* (UF de origem) serviu de base para categorização de clientes.

Após executados, os *scripts* produzirão um *dataset* de saída contendo 3 colunas: código do cliente, código do produto e avaliação. Esse conjunto de resultados será utilizado para alimentar uma rede neural. As combinações de cliente/produto serão as entradas, e as avaliações serão as saídas que se deseja prever. A rede neural será criada utilizando a biblioteca Keras. Utilizando o editor Jupyter Notebook será possível prototipar uma estrutura inicial e testá-la até chegar em um código minimamente funcional, uma rede que

possa ser treinada e que retorne predições corretas, ainda que considerando apenas uma fração do conjunto de dados total.

Desse ponto em diante, o desenvolvimento será realizado utilizando o Visual Studio Code. Diferente do Jupyter, que organiza o código em blocos e permite a inclusão de comentários e mídia junto ao código, o Visual Studio Code funciona de forma mais parecida com uma IDE convencional. Esse ambiente permite que o código seja organizado com mais concisão e clareza, o que se torna importante à medida que o projeto avança e o número de linhas cresce.

Se necessário, a rede neural passará então por diversas iterações de treinamento, na qual serão testados diferentes hiperparâmetros (ex.: número de épocas, *batch size*, número de camadas ocultas e funções de ativação, erro e otimização). Cada teste será registrado de modo que possam ser identificadas as configurações que resultem nos modelos mais precisos e rápidos. Ao final dos testes, os melhores modelos serão utilizados para a geração de recomendações.

Por fim, as avaliações e recomendações geradas por cada método serão comparadas e então analisadas qualitativamente em relação aos dados que as originaram. Através dessa observação, será possível constatar se as previsões geradas pelo modelo treinado foram relevantes ou não, e os métodos e estrutura de rede neural poderão ser ajustados ou refinados conforme a necessidade. Os resultados dessa análise serão detalhados na Seção 5.

## 3.2 Ambientes de teste

Para desenvolvimento e prototipação do projeto foi utilizado um notebook Acer Nitro 5, de propriedade do autor. A execução dos métodos de avaliação e posterior treinamento das redes neurais foram realizados em um servidor da Unijuí. As especificações de *hardware* e *software* desses ambientes estão descritos nas Tabelas 2 e 3.

Tabela 2 – Especificações do ambiente de prototipação

| **Recurso** | **Especificação** |
|---|---|
| CPU | Intel Core i5-8300H @ 2,30GHz, 4 núcleos |
| GPU | NVIDIA GeForce GTX 1050 @ 1,35MHz, 640 Cuda® Cores |
| RAM | 8GB |
| Armazenamento | 1TB |
| Sistema Operacional | Windows 10 Education |
| Versão do Python | 3.7.6 |
| Editores de texto | Jupyter Notebook 6.0.3<br>Visual Studio Code 1.51.0 |

Fonte: Autor

Tabela 3 – Especificações do ambiente de teste

| **Recurso** | **Especificação** |
|---|---|
| CPU | Intel Core i7-8700 @ 3.20GHz, 6 núcleos |
| GPU | NVIDIA GeForce GTX 1050 Ti @ 1,39MHz, 768 Cuda® Cores |
| RAM | 16GB |
| Armazenamento | 450GB |
| Sistema Operacional | Ubuntu 18.04.4 LTS |
| Versão do Python | 3.6.9 |
| Editores de texto | nano 2.9.3 |

Fonte: Autor

## 3.3 Considerações do capítulo

Neste capítulo foram apresentadas as metodologias empregadas no desenvolvimento do trabalho, bem como especificidades do ambiente de desenvolvimento e ferramentas, tanto no âmbito do *software* quanto do *hardware*.

# 4 Desenvolvimento

Nesta seção serão descritas, de forma mais detalhada, as etapas de análise dos *datasets* e os processos realizados em cada etapa, visando descrever como as recomendações foram obtidas através do uso das ferramentas e técnicas descritas nas seções anteriores.

## 4.1 Escolha dos datasets

Para o trabalho foram escolhidos três *datasets* de *e-commerce* entre dezenas disponíveis nos repositórios Kaggle e UCI ML. Estes conjuntos foram escolhidos pois possuem características que favorecem sua utilização para testes de sistemas de recomendação, tais como:

- Estão disponíveis de forma integral, pública e gratuita.
- Possuem mais de 100 mil registros.
- Embora os clientes estejam anonimizados, os conjuntos mantém dados que permitem categorizá-los (ex.: cidade/estado/país de origem).
- Os *datasets* provenientes do UCI ML já foram utilizados com propósitos acadêmicos por CHEN (2010).
- O *dataset* proveniente do Kaggle recebeu boas avaliações: possui classificação "Ouro" pela plataforma, ganhou 1172 votos positivos de usuários e "Usabilidade" avaliada com nota 10/10. [1]

Tabela 4 – Dados sobre os *datasets* selecionados para o trabalho

| **Nome** | **Registros** | **Tamanho (MB)** | **Período** | **Origem** |
|---|---|---|---|---|
| Online Retail | 541.909 | 43,4 | 2010-2011 | UCI ML |
| Online Retail II | 1.067.371 | 85,7 | 2009-2011 | UCI ML |
| Brazilian E-Commerce | 112.650 | 7,5 | 2016-2018 | Kaggle |

Fonte: Autor

[1] Dados referentes ao dia 19/12/2020.

## 4.2 Estudo de sistemas de recomendação

Decidiu-se por seguir um modelo de filtragem colaborativa, que tal como descrito na Seção 2.4 leva em conta apenas as avaliações dadas pelos clientes a um determinado produto e não dados do produto em si. Contudo, há diferentes maneiras de implementar sistemas desse tipo e, portanto, antes de iniciar-se o desenvolvimento, foi realizada uma busca por tutoriais e projetos de exemplo na Internet, que explicassem mais detalhadamente essas implementações.

Em um dos projetos encontrados, TANNER (2018) explica em vídeo uma abordagem semelhante, na qual ele treina e testa um modelo de sugestão de livros baseado em avaliações (notas de 1 a 5) das pessoas que os compraram. A rede criada por ele é treinada de forma a aprender a relação entre a forma codificada dessas entradas e suas respectivas saídas, as avaliações, como exemplificado na Tabela 5.

Neste trabalho, optou-se por seguir a implementação de Tanner, visto que esta apresenta diversas vantagens do ponto de vista de desempenho e manutenção de código, que serão explicadas em mais detalhes na Seção 4.3.

Tabela 5 – Avaliações de produto no sistema de recomendação de Tanner

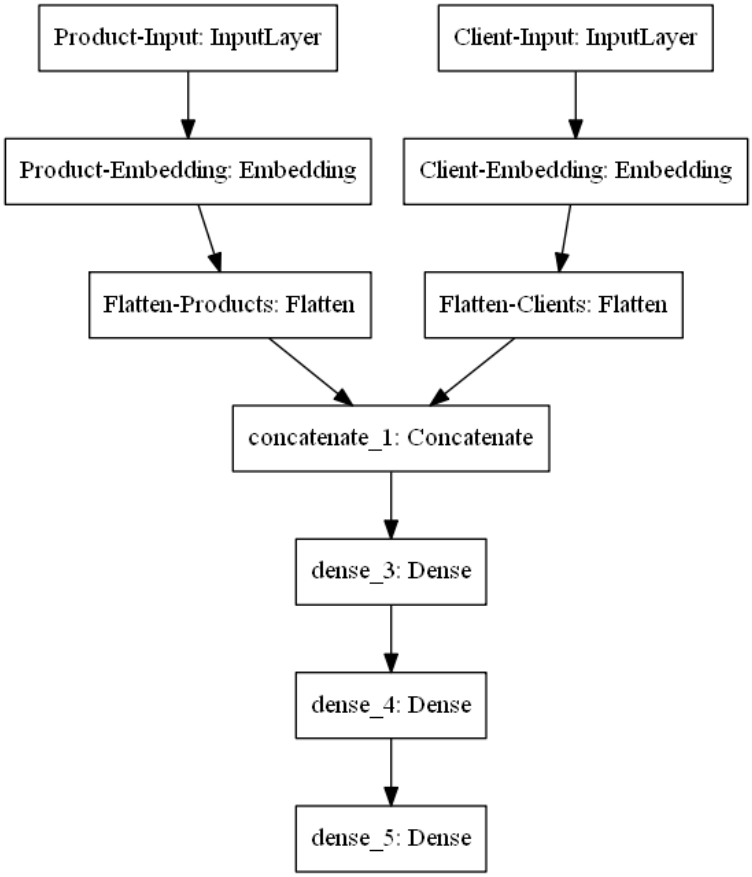
| **Cliente** | **Livro 1** | **Livro 2** | **Livro 3** | **Livro 4** |
|---|---|---|---|---|
| João | 5 | 3 | 4 | 3 |
| Maria | 3 | 2 | 4 | 1 |
| Pedro | 4 | 1 | 5 | 4 |
| Joana | 5 | 2 | 4 | 5 |

Fonte: adaptado de TANNER (2018)

## 4.3 Geração de recomendações

Após a geração de avaliações conforme os métodos definidos na Seção 3.1, os *datasets* resultantes foram utilizados como entradas para o treinamento da rede neural descrita por Tanner. A Figura 4 consiste no diagrama do modelo Keras, gerado através da função *keras.plot_model* no Jupyter Notebook para visualização e posteriormente importado como imagem. No diagrama estão expostas quatro camadas de pré-processamento, que criam os *embeddings* de produtos/clientes e posteriormente os unem, e três camadas do tipo Dense, que permitem à rede relacionar as entradas categóricas com as avaliações durante o treinamento.

Figura 4 – Diagrama do modelo Keras




Fonte: Autor

A Tabela 6 detalha os hiperparâmetros de cada camada e, através dela, pode-se constatar que as quatro primeiras camadas não possuem neurônios. Isso ocorre pois essas são apenas abstrações da biblioteca Keras para indicar que, antes das entradas serem efetivamente alimentadas para a rede neural, elas passarão pelo pré-processamento já descrito. Ou seja, essas camadas representam processos externos à RNA.

A camada Input é um objeto que representa um tensor do TensorFlow e que é utilizado como base para a criação da estrutura do modelo Keras. Nesta camada são definidos hiperparâmetros como dimensões e *batch size*. Esses dados são então processados pela camada Embedding, que realiza o processo descrito no Item 2.6.2. Em seguida, a camada Flatten planifica a matriz de *embeddings*, transformando-a em um vetor. Esse processo é repetido para cada conjunto que deseja-se utilizar como entrada, nesse caso, para códigos de clientes e produtos.

Tabela 6 – Tipo de camadas da rede neural

| # | **Tipo** | **Dimensões** | **Neurônios** | **Ativação** |
|---|---|---|---|---|
| 1 | Input | 1 | | |
| 2 | Embedding | Nº de registros + 1 | | |
| 3 | Flatten | | | |
| 4 | Concatenate | | | |
| 5 | Dense | | 128 | ReLu |
| 6 | Dense | | 32 | ReLu |
| 7 | Dense | | 1 | Linear |

Fonte: adaptado de TANNER (2018)

Antes dos vetores de *embeddings* serem alimentados às camadas Dense, estes são combinados. A camada Concatenate realiza essa operação recebendo como entrada um número qualquer de tensores de igual dimensão, e produz como saída um tensor que representa a concatenação de todas as entradas KERAS (2020b).

As camadas do tipo Dense utilizam função de ativação ReLu, com exceção da camada de saída que utiliza a função linear padrão do Keras. Esse tipo de configuração é frequentemente utilizada e apresenta bons resultados em RNAs com múltiplas camadas, principalmente em problemas envolvendo conjuntos de dados dispersos (GLOROT; BORDES; BENGIO, 2010).

De forma geral, as escolhas de tipos e parametrizações de camadas, hiperparâmetros de treinamento (listado na tabela 7) e métodos de pré-processamento são justificadas pelas seguintes vantagens:

- Conforme detalhado na Seção 2.4, o uso de *embeddings* em *Machine Learning* permite a redução de dimensionalidade e por consequência uma melhor utilização de recursos computacionais tais como RAM e processamento

Tabela 7 – Outros hiperparâmetros da rede neural

| **Hiperparâmetro** | **Valor** |
|---|---|
| Épocas | 4 |
| Batch size | 32 |
| Função de perda | MSE |
| Otimizador | Adam |
| Divisão treinamento/teste | 80/20 |

Fonte: adaptado de TANNER (2018)

- Consolida o processo de criação e aprendizado do *embedding* na mesma estrutura, facilitando o entendimento e manutenção do código.
- O funcionamento da estrutura foi documentado pelo autor tanto em texto quanto em vídeo, o que permite entendê-lo e validá-lo de forma mais assertiva.

A rede neural foi treinada utilizando um *dataset* de entrada de cada vez. Após o treinamento, a rede foi testada com pares de produtos/clientes do *dataset*, o que comprovou que o treinamento ocorreu de forma bem sucedida. A média de perda do modelo para 10 treinamentos com 4 épocas foi de 0,0116, o que corrobora esse resultado.

## 4.4 Considerações do capítulo

Neste capítulo foi descrito o processo de seleção dos *datasets* e a construção das estruturas de rede neural, detalhando os tipos de camadas e hiperparâmetros utilizados para o treinamento. O capítulo discorreu também sobre os motivos pelos quais as metodologias foram escolhidas, e como elas se aplicam no contexto da implementação do sistema de recomendação proposto.

# 5 Resultados

Neste capítulo serão apresentadas análises dos resultados da aplicação dos dois métodos de avaliação aos *datasets* selecionados. Essa análise é baseada na interpretação de gráficos estatísticos, produzidos utilizando a biblioteca Seaborn em Python.

Todos os gráficos tiveram seus eixos X/Y limitados de forma a facilitar a visualização, visto que em muitos casos as variáveis analisadas abrangem um grande intervalo de valores. No caso da Figura 21, as frequências são tão altas que optou-se por representá-las na escala logarítmica e dividir o histograma em 12 intervalos[1]. Para a amostra em questão, a cada marca de 1000 valores no eixo X, são exibidos dois intervalos. Isso ocorre em todos os gráficos representados nesta escala, embora o número de intervalos varie dependendo do número de registros no conjunto de dados.

Os gráficos de linhas foram gerados através do método KDE, padrão do Seaborn (WASKOM, 2020). Nesta análise, todos os gráficos do tipo consideram não a quantidade absoluta de ocorrências, mas sim a densidade de probabilidade, ou seja, a probabilidade relativa de que um determinado valor na distribuição ocorra no intervalo dado. Nas situações onde isso se aplica, o eixo Y estará identificado com o título "Densidade". Caso contrário, estará identificado com o título "Contagem".

Nas análises relacionadas ao método de FC por Categoria, todas as categorizações utilizadas envolvem regiões geográficas. A justificativa para a escolha e representação dessas regiões será detalhada nos respectivos capítulos. Quando necessário, a representação abreviada do nome de países será feita utilizando códigos de três caracteres, conforme a especificação ISO 3166-1 (ISO, 2013). As siglas dos estados brasileiros seguem o padrão usual de dois caracteres, tal como apresentado no site do IBGE (IBGE, 2020).

Quando citados em tabelas, os nomes de produtos foram encurtados, de forma a facilitar a leitura. Porém, o código do produto que consta nas bases de dados originais foi mantido ao lado do nome como referência. No caso do *dataset* Brazilian E-Commerce, visto que os códigos de produtos são representados por *hashes* MD5 com 32 caracteres cada, apenas os seis primeiros são exibidos, o que é suficiente para identificar o item em conjunto com a respectiva descrição da categoria.

Embora para a contabilização dos produtos mais populares tenham sido considerados os 10 itens mais vendidos no conjunto (FC Geral) ou nas categorias (FC por Categoria), apenas os 5 primeiros serão considerados nesta análise, de forma a facilitar a visualização principalmente na análise de popularidade de produtos entre regiões.

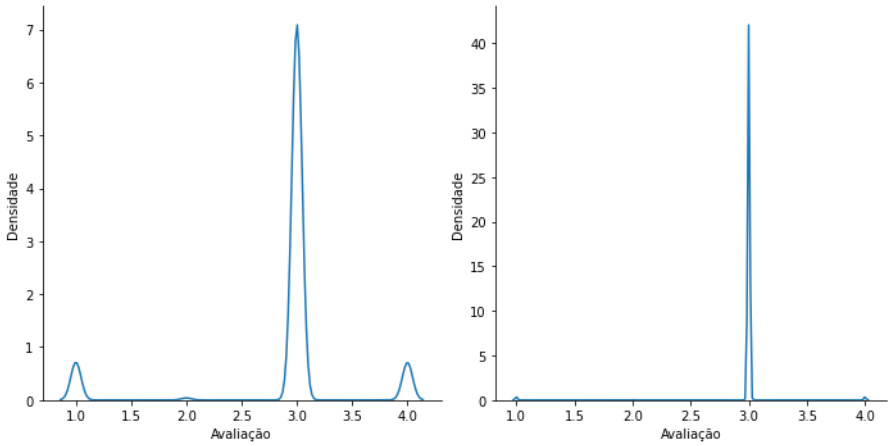
[1] Em algumas ocasiões, mesmo em artigos de língua portuguesa, os intervalos do histograma podem ser também referidos pelo termo inglês "*bin*".

## 5.1 Online Retail - FC Geral

Neste cenário de testes, o método de FC Geral para geração de avaliações foi aplicado ao *dataset* Online Retail. O processamento de 541.909 pedidos resultou em 5.079.419 registros de cliente/produto/avaliação.

O gráfico da Figura 5 mostra a densidade de avaliações para os primeiros 500 mil pares de cliente/produto, bem como para o total. Em ambas as visualizações, é possível constatar que a avaliação mais frequente é 3, seguida pelas notas 4 e 1.

Figura 5 – Densidade de avaliações para o *dataset* Online Retail - FC Geral



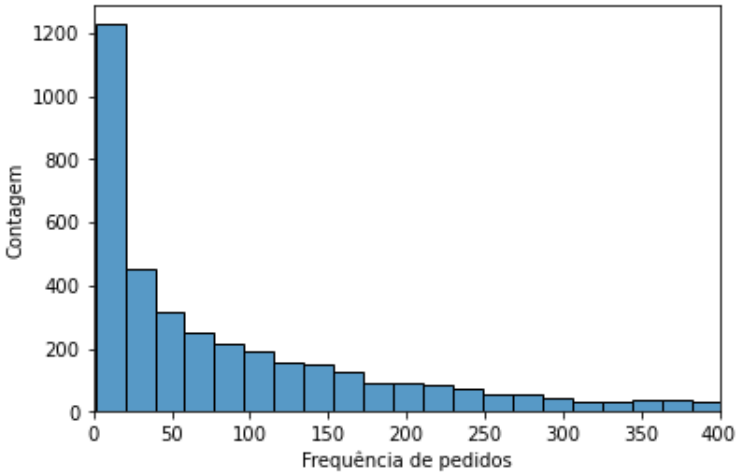

Fonte: Autor

A partir do gráfico da Figura 6, pode-se constatar que a frequência de pedidos se concentra nos valores mais baixos. Esse valor foi inferior a 400 pedidos para 92,3% dos produtos, e por volta de 25% dos produtos foi comprado em apenas 15 ocasiões ou menos. Isso indica que compras recorrentes não são comuns nesse conjunto, embora não seja possível determinar se esse comportamento se deve ao tipo de produto vendido pela loja, o período ou perfil de clientes considerado.

No que se refere as avaliações geradas e sua relação com as vendas, a Figura 7 mostra que avaliações 4 e 2 são as mais comuns para produtos com alta frequência de pedidos, portanto, é possível concluir que as recomendações foram geradas corretamente para o conjunto em questão seguindo as regras do método FC Geral.

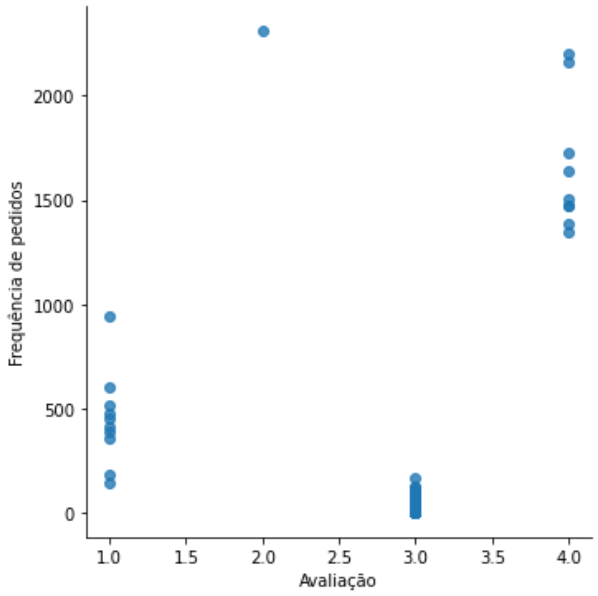
O fato de haver mais avaliações com nota 4 do que 2 aponta que a maior parte dos clientes nunca adquiriu produtos que estão entre os mais vendidos, embora esse grupo também não seja grande. Isso indica que o alto volume de venda dos produtos populares é resultado dos pedidos de uma pequena parcela dos clientes. Por outro lado, a alta incidência de avaliações 3 revela que muitos dos produtos não-populares também venderam pouco, e a partir disso pode-se concluir que, em geral, os clientes nessa base fizeram poucos pedidos.

Figura 6 – Frequência de pedidos para o *dataset* Online Retail - FC Geral




Fonte: Autor

Figura 7 – Distribuição de avaliações e pedidos para o *dataset* Online Retail - FC Geral




Fonte: Autor

A Tabela 8 ilustra as preferências de compra dos clientes desse conjunto. Os valores das colunas "Pedidos" e "Avaliação média" corroboram o que ilustra a Figura 7: a relação entre avaliações e pedidos não é linear, visto que o quinto produto mais vendido recebeu, em média, avaliações melhores que o mais vendido.

Através das descrições fornecidas pelo *dataset*, pode-se concluir que os itens mais populares dessa loja são utensílios domésticos, tais como suportes para velas, bandejas para bolo, bandeirinhas de festa e sacolas térmicas para transporte de alimentos.

Tabela 8 – Produtos mais vendidos para o *dataset* Online Retail

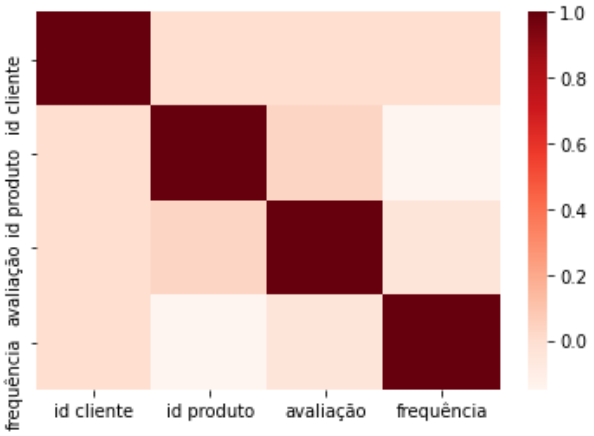
| ID | Nome do Produto | Pedidos | Avaliação média |
|---|---|---|---|
| 85123A | T-LIGHT HOLDER | 2312 | 3,81 |
| 22423 | CAKESTAND 3 TIER | 2203 | 3,83 |
| 85099B | JUMBO BAG | 2159 | 3,88 |
| 47566 | PARTY BUNTING | 1727 | 3,88 |
| 20725 | LUNCH BAG | 1639 | 3,92 |

Fonte: Autor

Esse padrão de não-linearidade é apresentado também pelo mapa de calor (Figura 8). No que se refere às avaliações e frequência de pedidos, o coeficiente fica abaixo de 0,2, o que indica correlação muito baixa ou inexistente. A partir desse resultado e demais métricas, pode-se concluir que para esse conjunto:

- Utensílios domésticos são os produtos mais populares.
- De forma geral, a maior parte dos clientes realizou poucos pedidos, ou seja, há baixa incidência de compras recorrentes.
- A alta densidade de notas 3 e 4 aponta abaixa incidência de compras de produtos em todos os níveis de popularidade. O alto volume de venda dos produtos populares é resultado dos pedidos de uma parcela representativa dos clientes, embora haja uma proporção muito maior de produtos que nunca tenham sido comprados.
- Há correlação fraca entre avaliações e frequência de pedidos, o que é o comportamento esperado dada a natureza do método da avaliação.

Figura 8 – Correlação entre variáveis do *dataset* Online Retail



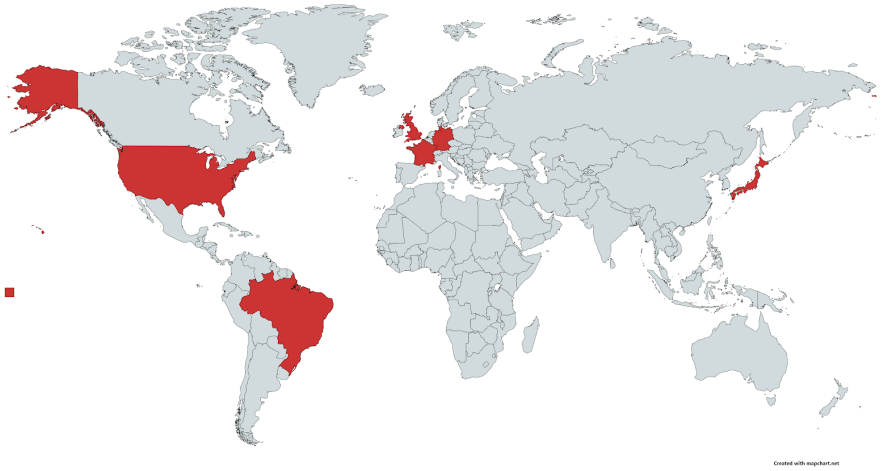

Fonte: Autor

## 5.2 Online Retail - FC por Categoria

No contexto deste *dataset*, o país de origem foi a variável utilizada para categorizar os clientes. Embora o conjunto contenha dados de 37 países, para essa análise foram considerados apenas os seis maiores países em população conforme estatísticas da ONU para 2020: Estados Unidos, Brasil, Japão, Alemanha, Reino Unido e França (ONU, 2020). As nações, distribuídas em três dos cinco continentes do mundo, estão destacadas na Figura 9.

Foram desconsiderados nesta análise os registros categorizados como *Unspecified* ("Não-especificado", ou seja, sem país definido) e também os classificados como *European Community* (Comunidade Européia), visto que os autores do *dataset* não especificam quais países estão incluídos nessa classificação.

Figura 9 – Países considerados na análise do *dataset* Online Retail - FC por Categoria

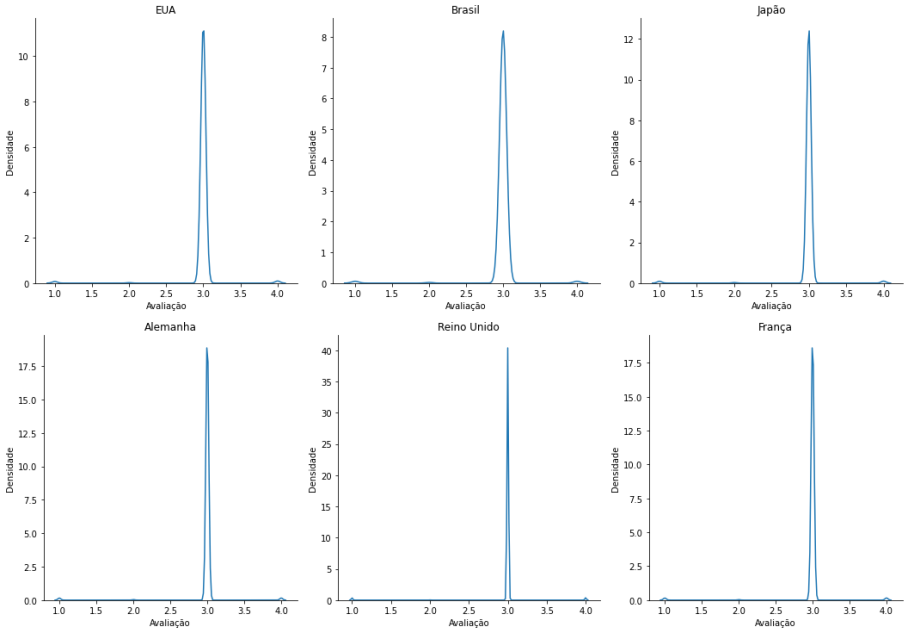


Fonte: Autor

No que se refere às avaliações (Figura 10), a tendência geral se reflete também em cada um dos países analisados, visto que a nota 3 é a mais frequente. Quanto à frequência de pedidos, os valores mais baixos exibidos no histograma (Figura 11) para EUA, Brasil e Japão mostram que essas nações representam apenas uma pequena parcela do total de pedidos nessa base.

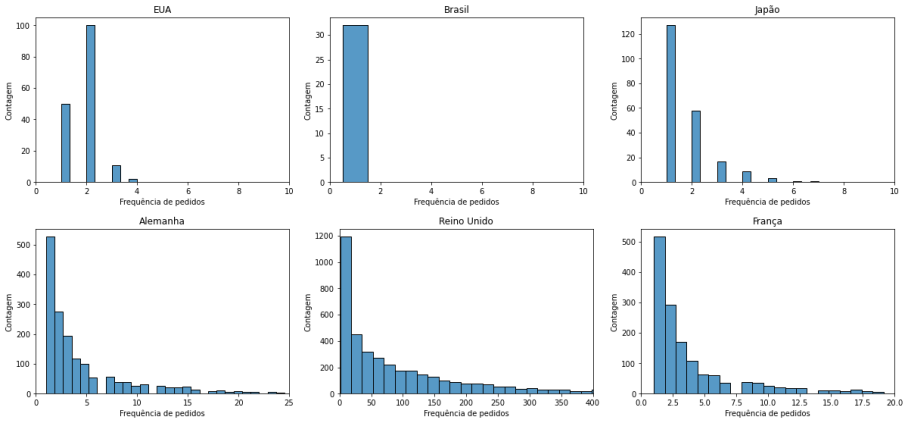
Alemanha, Reino Unido e França, por outro lado, apresentam maior número de pedidos e, em muitas ocasiões, itens são comprados de forma recorrente por dezenas ou centenas de vezes nessas regiões. Como já explicado no item 5.1, embora o hábito de compra recorrente possa ser identificado neste *dataset*, os dados disponíveis sobre clientes e produtos não permitem investigar essa questão mais a fundo a fim de buscar as razões para esse comportamento.

Figura 10 – Densidade de avaliações para o *dataset* Online Retail - FC por Categoria




Fonte: Autor

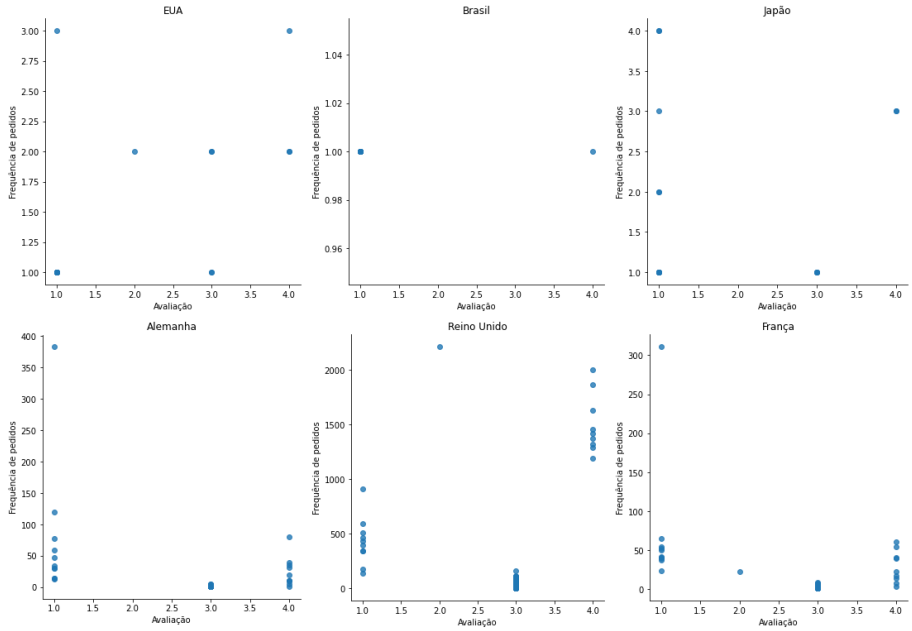
Figura 11 – Frequência de pedidos para o *dataset* Online Retail - FC por Categoria




Fonte: Autor

Como pode-se verificar na Figura 12, as avaliações 4 e 2 frequentemente estão relacionadas às frequências mais altas, embora não seja possível visualizar uma divisão tão clara entre pedidos populares e não-populares, tal como na análise geral (Figura 12). O alto nível de dispersão do gráfico dificulta a análise para os países não-europeus, embora seja possível notar que há várias ocorrências de produtos com avaliações diferentes para frequências iguais ou muito próximas. Isso ocorre pois nessas categorias a diferença de frequência de pedidos entre os produtos mais populares e os demais itens é pequena.

Figura 12 – Distribuição de avaliações e pedidos para o *dataset* Online Retail - FC por Categoria




Fonte: Autor

A Tabela 9 ilustra a comparação de popularidade dos produtos na análise geral e na análise por categoria. Assim como em análises anteriores, pode-se constatar que os pedidos do Reino Unido representam uma parcela significativa do total, visto que todos os produtos que estão entre os mais populares no mundo são populares também nesta região. Somente dois dos cinco produtos mais populares mundialmente figuram também entre os preferidos fora do Reino Unido.

Tabela 9 – Produtos mais vendidos no geral x categorias para o *dataset* Online Retail

| **Nome do Produto** | **USA** | **BRA** | **JPN** | **DEU** | **GBR** | **FRA** |
|---|---|---|---|---|---|---|
| T-LIGHT HOLDER | | | | | x | |
| CAKESTAND 3 TIER | | | | x | x | |
| JUMBO BAG | | | | | x | |
| PARTY BUNTING | | | | | x | |
| LUNCH BAG | | | | | x | x |

Fonte: Autor

Portanto, essa análise nos permite concluir que para a amostra considerada desse conjunto:

- Embora estejam entre os países mais populosos do mundo, EUA, Brasil e Japão representam uma parcela minoritária dos pedidos realizados.
- Por representarem uma grande parcela dos pedidos, produtos que são populares no

Reino Unido e em países europeus aparecem frequentemente entre os mais populares no mundo.

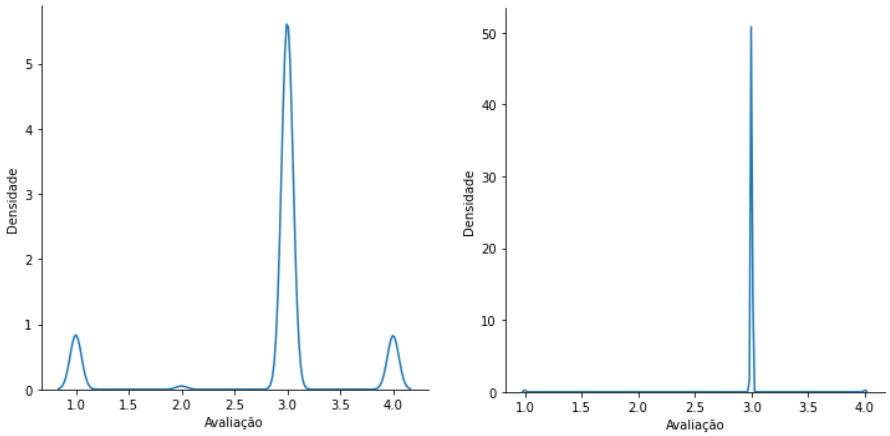
- Embora baixa no geral, a incidência de compras recorrentes é mais alta na Europa do que em outras regiões.
- Em países com menor volume de vendas, a diferença entre a frequência de pedidos dos produtos mais e menos populares é pequena.

## 5.3 Online Retail II - FC Geral

Neste cenário de testes, o método de FC Geral para geração de avaliações foi aplicado ao *dataset* Online Retail II. O processamento dos 1.067.371 pedidos resultou em 9.541.831 registros de cliente/produto/avaliação.

O gráfico da Figura 13 mostra a densidade de avaliações para os primeiros 500 mil pares de cliente/produto, bem como para o total. Em ambas as visualizações, é possível constatar que a avaliação mais frequente é 3 e a menos frequente é 2.

Figura 13 – Densidade de avaliações para o *dataset* Online Retail II - FC Geral

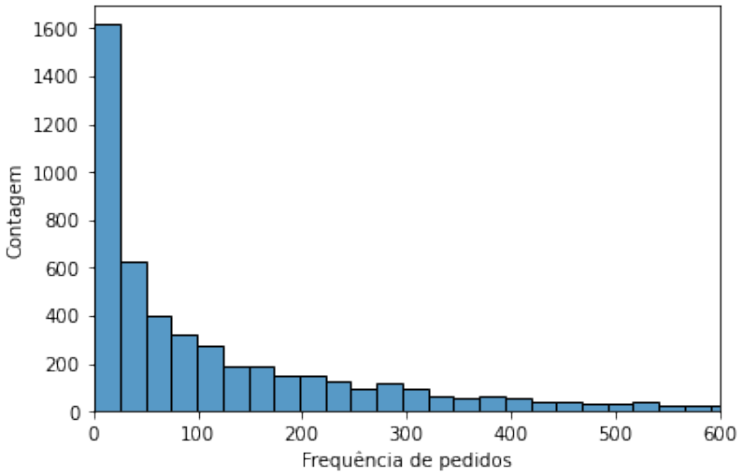



Fonte: Autor

Visto que esse *dataset* se refere a mesma loja e possui uma composição de clientes/produtos muito similar ao Online Retail, é possível concluir através da análise do histograma de frequência de pedidos (Figura 14) que um padrão semelhante ocorre: 91,5% dos produtos foram comprados com frequência inferior a 600 pedidos, o que indica uma baixa incidência de compras recorrentes.

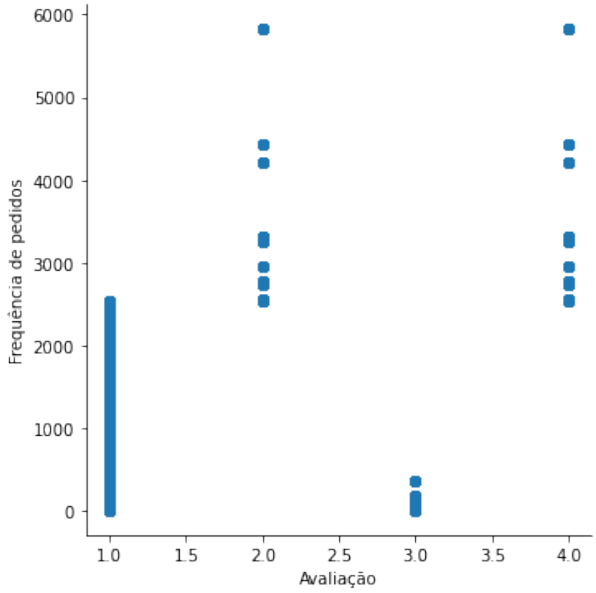
A Figura 15 relaciona a densidade de avaliações em relação a frequência de pedidos. As avaliações 2 e 4 são as mais comuns para produtos com alta frequência, portanto, é possível concluir que as recomendações foram geradas corretamente para o conjunto em questão seguindo as regras do método FC Geral. O pico de notas 3 em contraste com as notas 4 indica um baixo número de pedidos por cliente.

Figura 14 – Frequência de pedidos para o *dataset* Online Retail II - FC Geral




Fonte: Autor

Figura 15 – Distribuição de avaliações e pedidos para o *dataset* Online Retail II - FC Geral




Fonte: Autor

Uma análise semelhante pode ser observada na Tabela 10, relacionando as variáveis já citadas com os cinco produtos mais vendidos. Assim como no *dataset* Online Retail, os itens mais populares dessa loja são utensílios domésticos, e quatro dos cinco produtos figuram entre os mais vendidos em ambos os conjuntos. Visto que há pelo menos um ano de diferença entre as amostras, pode-se concluir que os interesses de compra dos clientes não se alteraram significativamente durante esse período.

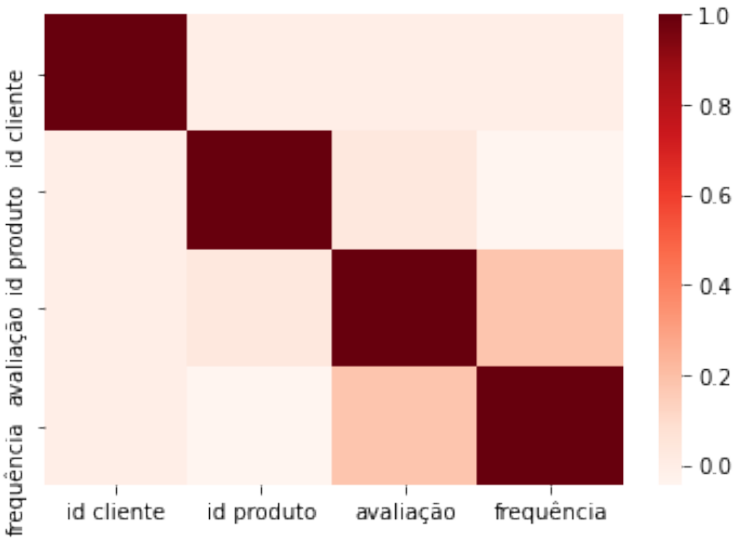
Tabela 10 – Produtos mais vendidos para o *dataset* Online Retail II

| ID | Nome do Produto | Pedidos | Avaliação média |
|---|---|---|---|
| 85123A | T-LIGHT HOLDER | 5829 | 3,74 |
| 22423 | CAKESTAND 3 TIER | 4424 | 3,83 |
| 85099B | JUMBO BAG | 4216 | 3,87 |
| 21212 | PACK OF 72 CAKE CASES | 3318 | 3,92 |
| 20725 | LUNCH BAG | 3259 | 3,91 |

Fonte: Autor

Os dados da Tabela 10 apontam também para o mesmo padrão de não-linearidade entre avaliações e pedidos já identificado no *dataset* Online Retail. Contudo, no mapa de calor (Figura 16) é possível observar que no caso do Online Retail II a correlação entre essas variáveis é um pouco mais proeminente, chegando a 0,2 (fraca).

Figura 16 – Correlação entre variáveis do *dataset* Online Retail II




Fonte: Autor

Em resumo, essa análise nos permite concluir que para esse conjunto:

- Utensílios domésticos são os produtos mais populares.
- De forma geral, a maior parte dos clientes realizou poucos pedidos, ou seja, há baixa incidência de compras recorrentes.
- A alta densidade de notas 3 e 4 aponta abaixa incidência de compras de produtos em todos os níveis de popularidade.
- Há maior densidade de notas 3 do que 4, o que indica o alto volume de venda dos produtos populares é resultado dos pedidos de uma parcela representativa dos

clientes, embora haja uma proporção muito maior de produtos que nunca tenham sido comprados.

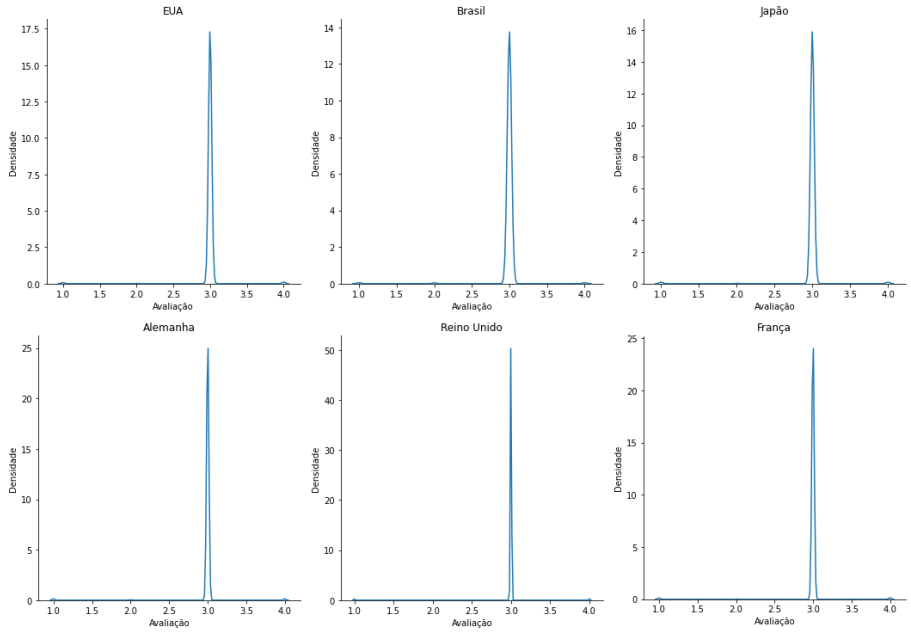
- Há correlação fraca entre avaliações e frequência de pedidos, o que é o comportamento esperado dada a natureza do método da avaliação.

## 5.4 Online Retail II - FC por Categoria

Recomendações foram geradas para o *dataset* Online Retail II utilizando o método de FC por Categoria neste cenário. Visto que esse conjunto de dados compartilha parte dos clientes e produtos do Online Retail, os mesmos países relacionados na Figura 9 serão utilizados como base para categorização dos clientes.

Quanto à densidade de avaliações, na Figura 17 pode-se observar que a análise de cada um dos seis países reflete a tendência geral: as avaliações 3 são as mais frequentes, o que evidencia baixo volume de pedidos por cliente.

Figura 17 – Densidade de avaliações para o *dataset* Online Retail II - FC por Categoria

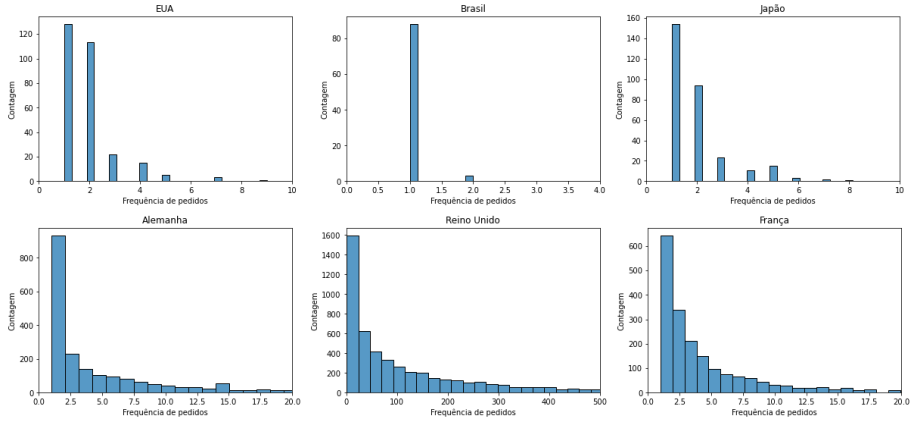



Fonte: Autor

Da mesma forma, a frequência de pedidos se concentra nos valores mais baixos, conforme ilustra o painel da Figura 18. EUA, Brasil e Japão apresentam um número pequeno de barras no histograma e escassas ocorrências de frequência acima de 1, o que indica que os pedidos realizados para esses países representam apenas uma pequena fração do total. Por outro lado, a frequência de pedidos na Alemanha, Reino Unido e França exibe um padrão muito próximo a uma função logarítmica inversa, visto que quanto maior a frequência, menor o número de ocorrências. A partir disso, pode-se concluir que nos três

países europeus analisados há maior incidência de compras recorrentes que nas demais regiões.

Figura 18 – Frequência de pedidos para o *dataset* Online Retail II - FC por Categoria

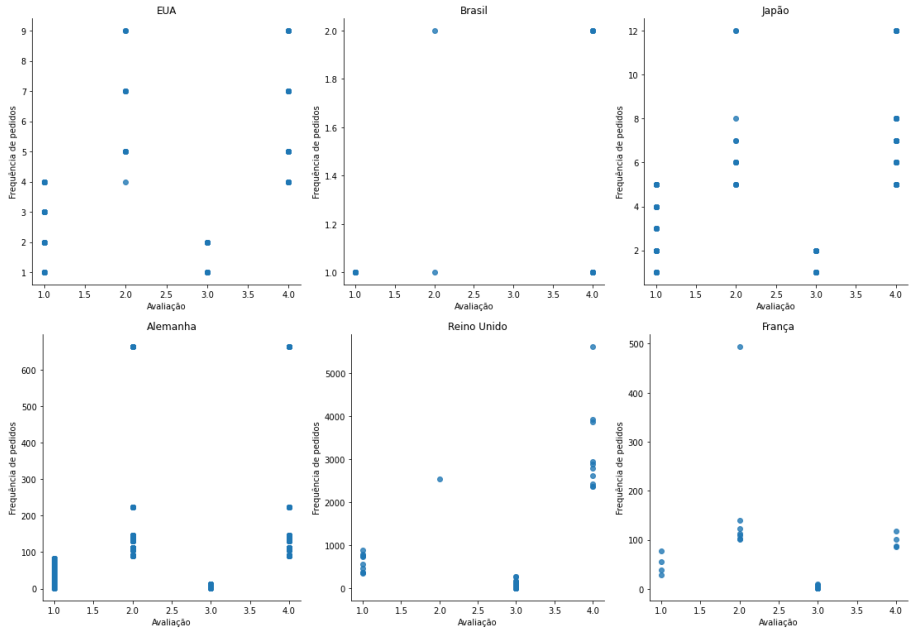



Fonte: Autor

Também em concordância com a análise geral, através da Figura 19 pode-se constatar que para os produtos com maior frequência de venda as avaliações 4 e 2 são mais frequentes, enquanto as notas 1 e 3 ocorrem nos demais casos. Contudo, é possível notar peculiaridades locais nessa amostra. Nos EUA, Brasil e Japão, as ocorrências estão esparsamente distribuídas no gráfico e a diferença entre a frequência máxima e mínima da série é pequena em comparação aos países europeus. Isso indica que nestes países mesmo os produtos mais populares não foram vendidos de forma recorrente, ou que o número de pedidos que compõe essa categoria não nos permite identificar uma relação de recorrência clara.

A Tabela 11 ilustra os produtos mais vendidos no mundo e sua popularidade nos países analisados. Pode-se verificar que todos os itens mundialmente populares são também populares no Reino Unido, o que se deve principalmente a alta representatividade desse país em relação ao número total de pedidos. Apenas três dos cinco produtos mundialmente populares repetiram esse padrão em 50% dos países analisados, e apenas um deles foi popular em mais de 30% das regiões, o que indica que existem diferenças locais significativas em relação aos interesses de compra dos consumidores.

Figura 19 – Distribuição de avaliações e pedidos para o *dataset* Online Retail II - FC por Categoria




Fonte: Autor

Tabela 11 – Produtos mais vendidos no geral x categorias para o *dataset* Online Retail II

| **Nome do Produto** | **USA** | **BRA** | **JPN** | **DEU** | **GBR** | **FRA** |
|---|---|---|---|---|---|---|
| T-LIGHT HOLDER | | | | | x | |
| CAKESTAND 3 TIER | x | | | x | x | |
| JUMBO BAG | | | | | x | |
| PACK OF 72 CAKE CASES | x | | | | x | |
| LUNCH BAG | | | x | | x | x |

Fonte: Autor

Portanto, essa análise nos permite concluir que para a amostra considerada desse conjunto:

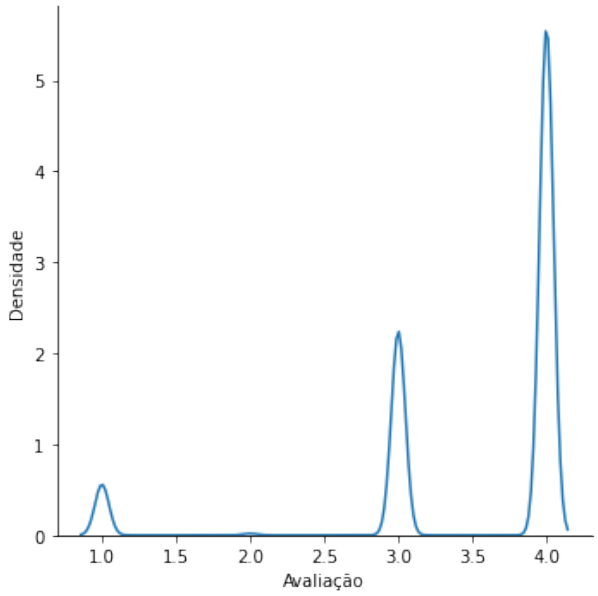
- A maior parte dos compradores está em países europeus.
- Dois dos produtos mais vendidos no mundo são também populares em pelo menos metade dos países analisados.
- Há maior incidência de compras recorrentes em países europeus do que nas demais partes do mundo.

## 5.5 Brazilian E-Commerce - FC Geral

Neste cenário de testes, o método de FC Geral para geração de avaliações foi aplicado ao *dataset* Brazilian E-commerce. O processamento de 112.650 pedidos resultou em 1.430.959 registros de cliente/produto/avaliação.

Em relação às avaliações, a Figura 20 permite visualizar que as avaliações 4 e 3 são as mais frequentes. Isso indica que o alto volume de venda dos produtos populares é resultado dos pedidos de uma pequena parcela dos clientes. Além disso, a alta incidência de notas 3 revela que muitos dos produtos não-populares também venderam pouco. Contudo, diferente do que ocorre para os *datasets* Online Retail, neste conjunto os produtos não-populares são vendidos em maior proporção, visto a menor incidência de notas 3.

Figura 20 – Densidade de avaliações para o *dataset* Brazilian E-Commerce - FC Geral

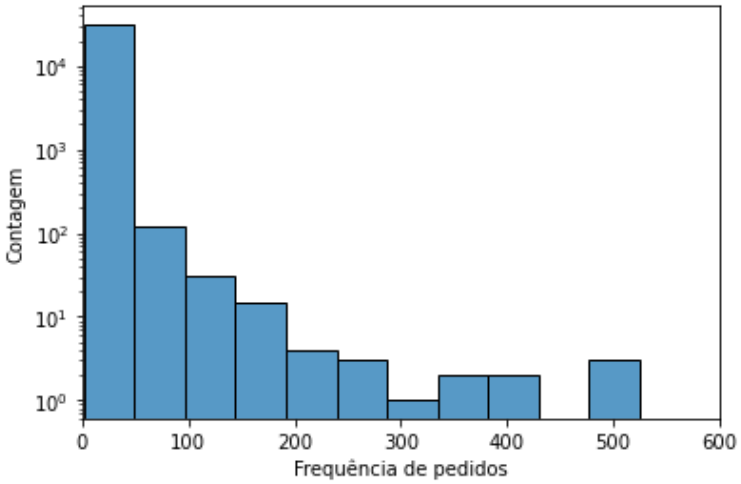



Fonte: Autor

A Figura 21 ilustra o baixo índice de pedidos por cliente: 99% dos produtos foram comprados com frequência igual ou inferior a 50 pedidos. Ou seja, assim como ocorre para os *datasets* Online Retail, a maior parte dos produtos não é comprada de maneira recorrente pelos clientes.

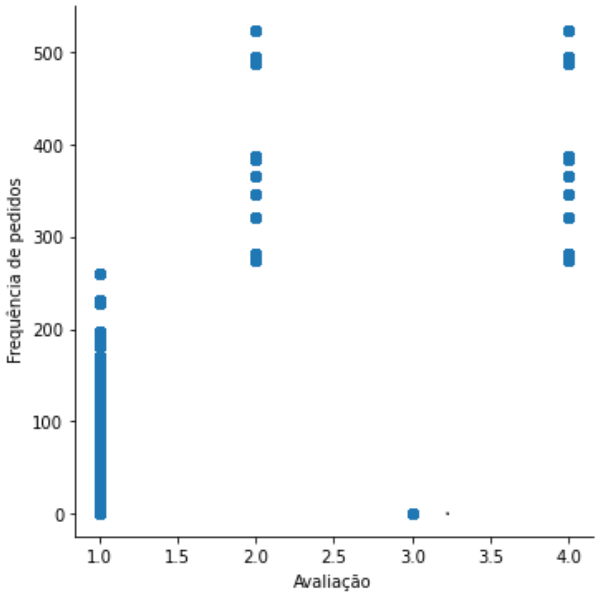
Conforme Figura 22, pode-se concluir que a geração das avaliações ocorreu corretamente para esse conjunto, visto que a divisão entre avaliações para produtos com alta (notas 4 e 2) e baixa (notas 3 e 1) frequência de pedidos é representada pelos quatro agrupamentos de pontos que podem ser visualizados próximos às extremidades no gráfico de dispersão.

Figura 21 – Frequência de pedidos para o *dataset* Brazilian E-Commerce - FC Geral




Fonte: Autor

Figura 22 – Distribuição de avaliações e pedidos para o *dataset* Brazilian E-Commerce - FC Geral




Fonte: Autor

Ferramentas de jardinagem são os artigos mais populares entre os clientes, correspondendo a três dos cinco itens mais vendidos, conforme a Tabela 12. Todos esses produtos obtiveram avaliação média de 3,99, o que indica que a maioria dos clientes nunca os comprou. Visto que estes são os produtos mais populares, essa afirmação pode parecer um contrassenso, mas corrobora a mesma tendência ilustrada pela Figura 20.

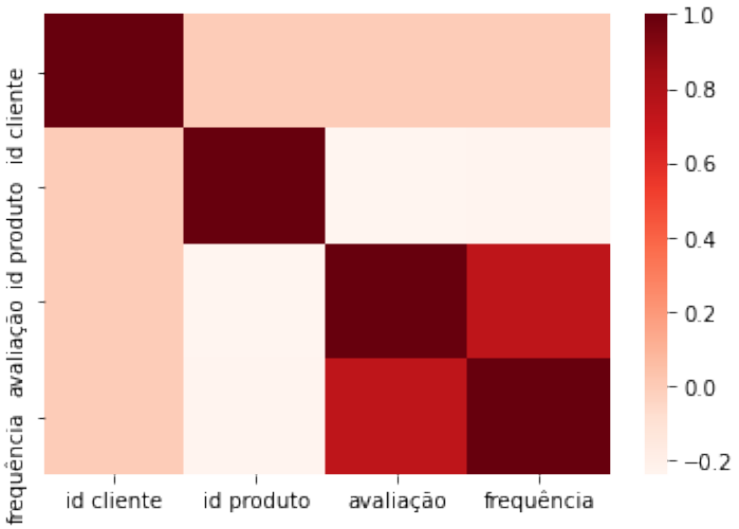
Tabela 12 – Produtos mais vendidos para o *dataset* Brazilian E-Commerce

| **Início ID** | **Categoria** | **Pedidos** | **Avaliação média** |
|---|---|---|---|
| aca2eb | moveis_decoracao | 525 | 3,99 |
| 422879 | ferramentas_jardim | 496 | 3,99 |
| 99a478 | cama_mesa_banho | 489 | 3,99 |
| 389d11 | ferramentas_jardim | 387 | 3,99 |
| 368c6c | ferramentas_jardim | 385 | 3,99 |

Fonte: Autor

Nesse conjunto é possível identificar alta correlação entre avaliações e frequência de pedidos (Figura 23). Embora a avaliação 3 seja a segunda mais frequente, essa ocorreu apenas em frequências de pedido muito próximas ou iguais a 1, enquanto tanto notas 2 quanto 4 corresponderam a frequências na casa das dezenas de milhares. Esse cenário demonstra uma tendência que, embora não coincida perfeitamente, é próxima de um padrão linear.

Figura 23 – Correlação entre variáveis do *dataset* Brazilian E-Commerce




Fonte: Autor

Em resumo, essa análise nos permite concluir que para esse conjunto:

- Ferramentas de jardinagem são os itens mais populares.
- De forma geral, a maior parte dos clientes realizou poucos pedidos, ou seja, há baixa incidência de compras recorrentes.
- A alta densidade de notas 3 e 4 aponta abaixa incidência de compras de produtos em todos os níveis de popularidade.

- Há maior densidade de notas 4 do que 3, o que indica o alto volume de venda dos produtos populares é resultado dos pedidos de uma pequena parcela dos clientes.
- Há forte correlação entre a volume de vendas de um produto e a avaliação que ele receberá conforme o método FC Geral.

## 5.6 Brazilian E-Commerce - FC por Categoria

Para esse conjunto, a UF de origem foi a variável utilizada para categorizar os clientes. Embora o *dataset* contenha pedidos provenientes dos 26 estados brasileiros, bem como do Distrito Federal, para essa análise foram considerados somente os seis estados mais populosos segundo o IBGE (IBGE, 2020): São Paulo, Minas Gerais, Rio de Janeiro, Bahia, Paraná e Rio Grande do Sul. As UFs, distribuídas ao longo de três das cinco regiões do Brasil, estão destacadas na Figura 24.

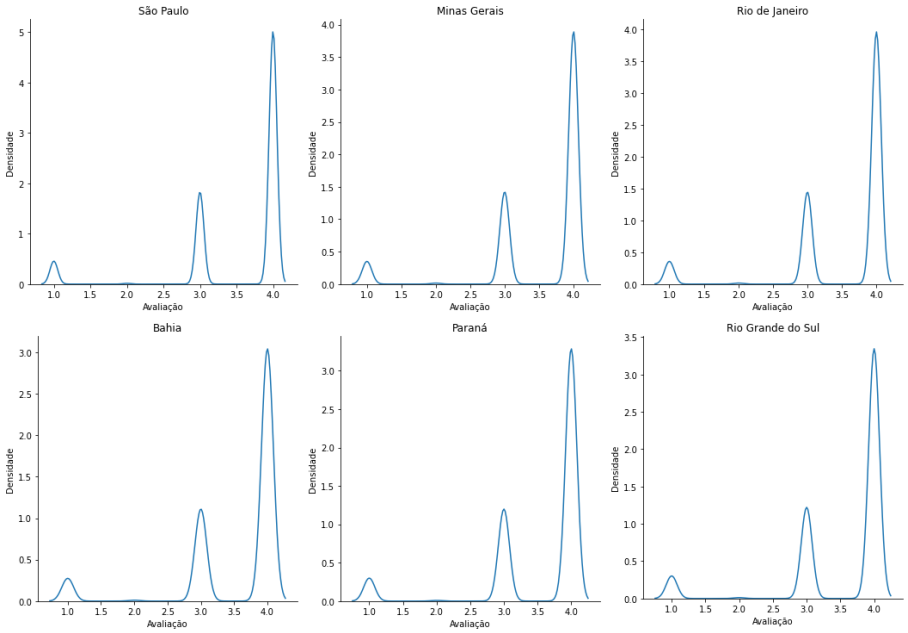
A Figura 25 ilustra o mesmo padrão de avaliações para todos os estados analisados: as avaliações 4 e 3 são as mais frequentes. Quanto à frequência de pedidos (Figura 26), pode-se visualizar uma maior ocorrência de valores baixos, seguindo a tendência identificada na análise geral.

Figura 24 – Estados brasileiros considerados na análise do *dataset* Brazilian E-Commerce - FC por Categoria

Fonte: Autor

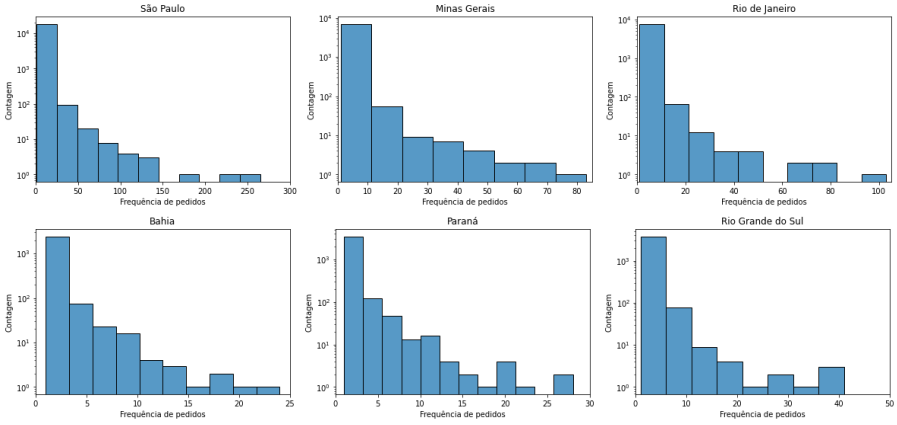
São Paulo, Minas Gerais e Rio de Janeiro são os estados onde produtos foram comprados apenas uma vez com maior frequência (SP chega a ter mais de 10 mil ocorrências desse tipo). Por outro lado, essas regiões foram também as únicas a registrar recorrência de compra igual ou superior a 50 pedidos. Conclui-se a partir disso que, apesar da baixa incidência de compras recorrentes, no geral os estados do Sudeste são os que apresentam maior recorrência entre as UFs analisadas.

Figura 25 – Densidade de avaliações para o *dataset* Brazilian E-Commerce - FC por Categoria




Fonte: Autor

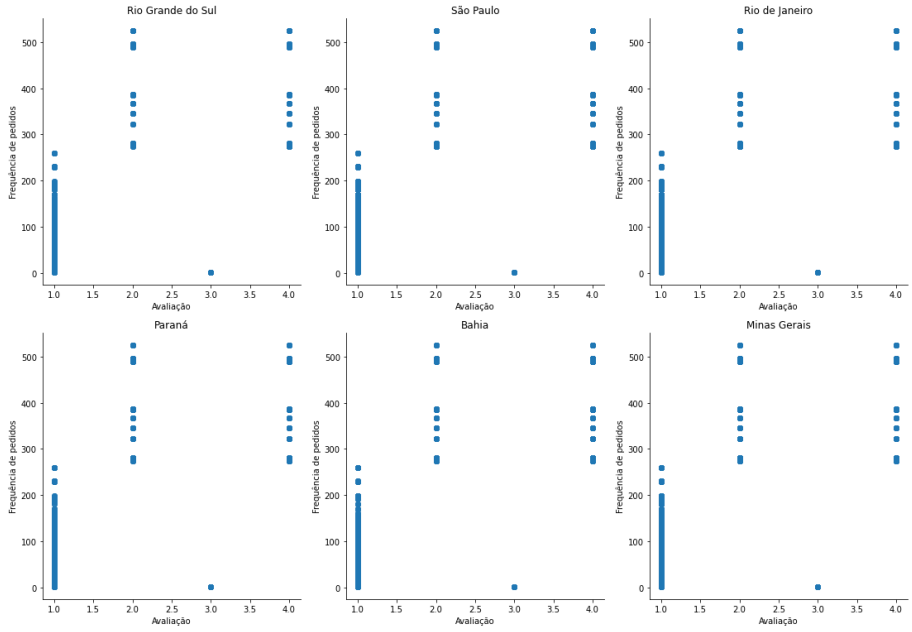
Figura 26 – Frequência de pedidos para o *dataset* Brazilian E-Commerce - FC por Categoria




Fonte: Autor

Conforme a Figura 27, a relação entre pedidos e avaliações é quase idêntica em todos os estados analisados, o que indica que esses clientes, embora estejam em categorias distintas, possuem hábitos de compra semelhantes. Nos demais aspectos, a análise por UF segue a mesma tendência da análise geral: avaliações 4 são as mais frequentes, seguidas pelas notas 3 e 1.

Figura 27 – Distribuição de avaliações e pedidos para o *dataset* Brazilian E-Commerce - FC por Categoria




Fonte: Autor

Os itens mais populares a nível nacional frequentemente alcançam popularidade equivalente nos estados analisados, conforme mostra a Tabela 13. Somente clientes da Bahia e Paraná apresentaram interesses de compra que diferem um pouco da preferência geral. Pode-se concluir a partir disso que, para esse conjunto, a frequência de pedidos e interesses dos consumidores são distribuídos de forma mais uniforme do que nos *datasets* Online Retail, nos quais apenas uma região seguia consistentemente a tendência geral de popularidade de produtos.

Tabela 13 – Produtos mais vendidos no geral x categorias para o *dataset* Brazilian E-Commerce

| **Início ID** | **Categoria** | **SP** | **MG** | **RJ** | **BA** | **PR** | **RS** |
|---|---|---|---|---|---|---|---|
| aca2eb | moveis_decoracao | x | x | x | | x | x |
| 422879 | ferramentas_jardim | x | x | x | x | x | x |
| 99a478 | cama_mesa_banho | x | x | x | x | | x |
| 389d11 | ferramentas_jardim | x | x | x | | x | x |
| 368c6c | ferramentas_jardim | x | x | x | x | x | x |

Fonte: Autor

A análise nos permite concluir que para a amostra considerada desse conjunto:

- A frequência de pedidos e interesses dos consumidores são distribuídos de forma mais uniforme entre as regiões do que nos *datasets* Online Retail.
- Os produtos que estão entre os mais vendidos em todo o Brasil apresentam popularidade equivalente nos estados analisados.
- Os estados do Sudeste são os que apresentam maior incidência de compras recorrentes.
- Os produtos mais populares foram comprados com menor frequência por clientes da Bahia, Paraná e Rio de Janeiro.
- Os clientes da Bahia e Paraná apresentaram interesses de compra que diferem um pouco da preferência geral.

## 5.7 Considerações do capítulo

Neste capítulo foram descritos os resultados obtidos a partir da análise das avaliações geradas através dos 2 métodos apresentados na Seção 3.1 para os 3 *datasets* expostos na Seção 4.1. Além da exposição de dados estatísticos em formato gráfico e tabular, o capítulo buscou estabelecer paralelos entre a proposta inicial e os resultados obtidos, bem como extrair *insights* que permitem entender os hábitos de compra e comportamento dos grupos de consumidores representados nos conjuntos de dados em questão.

# 6 Conclusão e Trabalhos Futuros

Neste capítulo serão apresentadas as conclusões obtidas no desenvolvimento do presente trabalho, bem como possibilidades de expansão do estudo em trabalhos futuros.

## 6.1 Conclusão

Este trabalho apresentou o desenvolvimento de um sistema de recomendação de produtos para *e-commerce* com utilização de técnicas de IA e ML aliadas com análise estatística. O sistema descrito sugere produtos com base em diferentes análises da relação entre a frequência de pedido de um item e os clientes e segmentos relacionados a essa venda.

O trabalho alcançou o objetivo de gerar recomendações de produtos, aplicando primeiramente métodos estatísticos de avaliação a um *dataset* de pedidos, e utilizando as avaliações geradas para alimentar um modelo conforme descrito por TANNER (2018), que, por sua vez, resulta nas sugestões de produtos.

Além de ser utilizado para alimentar uma aplicação de *e-commerce*, as sugestões geradas podem ser analisadas também através de ferramentas estatísticas, de forma a produzir gráficos e tabelas que forneçam *insights* sobre o *dataset* analisado, tal como foi feito neste trabalho no item 5. Através dessa análise, foi possível constatar que os métodos para avaliação definidos na Seção 3.1 puderam ser utilizados de forma consistente para gerar recomendações condizentes com as características dos dados nas quais se baseiam e que podem ter sua consistência verificada.

A análise dos resultados, portanto, nos permite também responder as questões de pesquisa expostas na Seção 1, que levantam hipóteses quanto à viabilidade da geração de sugestões dadas as limitações propostas, bem como questionam qual seria a configuração de rede neural ideal para a resolução do problema. Considerando que sugestões consistentes com os dados puderam ser geradas com base na abordagem de TANNER (2018), pode-se concluir que a estrutura por ele proposta é efetiva e utilizável na prática.

## 6.2 Trabalhos futuros

Em trabalhos futuros, pretende-se validar esse modelo através de sua implementação em uma aplicação de comércio eletrônico em produção, avaliando as interações do usuário com o sistema de recomendação e posteriores avaliações do mesmo quanto ao pedido realizado. É possível testar diferentes abordagens para geração de avaliações e

recomendações através de teste A/B, e sistemas de filtragem baseados em conteúdo podem ser utilizados a fim de complementar ou refinar os resultados produzidos pelos métodos descritos neste estudo.

Outros métodos de avaliação e categorização podem também ser desenvolvidos e testados. Enquanto neste trabalho a localização geográfica foi utilizada como forma de agrupar os clientes, outras características podem ser utilizadas separadamente ou em conjunto de forma a analisar interesses e produzir recomendações mais precisas, tais como faixa etária, gênero, interesses (no caso de indivíduos), segmento de mercado e porte (no caso de empresas).

Outra possível linha de estudo é a análise da estrutura de rede neural utilizada buscando otimizá-la. Através de testes com diferentes hiperparâmetros é possível determinar se há possibilidade de melhoria na precisão ou velocidade de treinamento do modelo.

# Referências

ACIAR, S. V. et al. Increasing effectiveness in e-commerce: recommendations applying intelligent agents. *International Journal of Business and Systems Research*, v. 1, n. 1, p. 81–97, 2007. Disponível em: <https://www.researchgate.net/publication/247834709_Increasing_effectiveness_in_e-commerce_Recommendations_applying_intelligent_agents>. Citado 2 vezes nas páginas 24 e 28.

ADAM, M.; WESSEL, M.; BENLIAN, A. *AI-based chatbots in customer service and their effects on user compliance*. 2020. Disponível em: <https://link.springer.com/article/10.1007/s12525-020-00414-7>. Acesso em: 18 out 2020. Citado na página 18.

AGGARWAL, C. C. *Neural Networks and Deep Learning - A Textbook*. 1. ed. Cham: Springer, 2018. Citado na página 21.

AGGARWAL, C. C. et al. Horting hatches an egg. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '99*. ACM Press, 1999. Disponível em: <https://doi.org/10.1145/312129.312230>. Citado na página 24.

AKOGLU, H. *User's guide to correlation coefficients*. 2018. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6107969>. Acesso em: 18 out 2020. Citado na página 27.

ALARIE, B.; NIBLETT, A.; YOON, A. H. How artificial intelligence will affect the practice of law. *University of Toronto Law Journal*, 2018. Disponível em: <https://doi.org/10.3138/utlj.2017-0052>. Acesso em: 18 out 2020. Citado na página 17.

ALPAYDIN, E. *Introduction to Machine Learning*. 4. ed. Cambridge: MIT, 2018. Citado na página 20.

BALABANOVIĆ, M.; SHOHAM, Y. Fab. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 40, n. 3, p. 66–72, mar. 1997. Disponível em: <https://doi.org/10.1145/245108.245124>. Citado 2 vezes nas páginas 24 e 25.

BEZERRA, E. *Métodos de Machine Learning para seleção de variáveis com aplicações ao Rugby Sevens Feminino*. 2019. Disponível em: <https://www.lume.ufrgs.br/handle/10183/203926>. Acesso em: 18 out 2020. Citado na página 20.

BROWNLEE, J. *Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. 1. ed. [S.l.]: Jason Brownlee, 2019. Citado na página 22.

BURKE, R. Interactive critiquing for catalog navigation in e-commerce. *Artificial Intelligence Review*, v. 18, n. 1, p. 245–267, 2002. Disponível em: <https://link.springer.com/article/10.1023/A:1020701617138>. Citado 3 vezes nas páginas 24, 28 e 30.

CHEN, D. *Online Retail Data Set*. 2010. Disponível em: <https://archive.ics.uci.edu/ml/datasets/Online+Retail>. Acesso em: 18 out 2020. Citado na página 34.

CHEN, R. et al. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, v. 6, p. 64301–64320, 2018. Citado 2 vezes nas páginas 15 e 25.

CHEN, Y.; ARGENTINIS, J. E.; WEBER, G. *IBM Watson: How Cognitive Computing Can Be Applied to Big Data Challenges in Life Sciences Research*. 2016. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0149291815013168>. Acesso em: 18 out 2020. Citado na página 18.

CHUAN, M. Z. et al. Applying artificial neural network to build engineering project bid evaluation system. In: *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*. [S.l.: s.n.], 2011. p. 211–214. Citado na página 17.

EBIT. *Webshoppers 40 - Versão Free*. 2019. Disponível em: <https://www.ebit.com.br/webshoppers/download?pathFile=D%3A%5CEbit%5CSites%5Cwww.ebit.com.br%5CPDF_WS%5C40.webshoppers_2019.pdf&fileName=Webshoppers_40.pdf>. Acesso em: 18 out 2020. Citado na página 14.

ERTEL, W. *An Introduction to Artificial Intelligence*. 2. ed. Cham: Springer, 2017. Citado 2 vezes nas páginas 18 e 19.

FIGUEIREDO FILHO, D.; SILVA JUNIOR, J. A. Desvendando os mistérios do coeficiente de correlação de pearson. *Revista Política Hoje*, v. 18, n. 1, 2010. ISSN 0104-7094. Disponível em: <https://periodicos.ufpe.br/revistas/politicahoje/article/view/3852>. Citado na página 27.

FRESSATO, E. P. *Incorporação de metadados semânticos para recomendação no cenário de partida fria*. Tese (Doutorado) — Universidade de Sao Paulo, 2019. Disponível em: <https://doi.org/10.11606/d.55.2019.tde-09082019-134753>. Citado 3 vezes nas páginas 23, 24 e 25.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: . [S.l.: s.n.], 2010. v. 15, p. 1. Citado na página 37.

GOOGLE DEVELOPERS. *Machile Learning Glossary*. 2020. Disponível em: <https://developers.google.com/machine-learning/glossary>. Acesso em: 18 out 2020. Citado 2 vezes nas páginas 26 e 28.

HAYKIN, S. *Redes neurais: princípios e prática*. 1. ed. Porto Alegre: Bookman, 2001. Citado na página 19.

HE, X. et al. Neural collaborative filtering. In: *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017. Disponível em: <https://doi.org/10.1145/3038912.3052569>. Citado na página 28.

HWANGBO, H.; KIM, Y. S.; CHA, K. J. Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, Elsevier BV, v. 28, p. 94–101, mar. 2018. Disponível em: <https://doi.org/10.1016/j.elerap.2018.01.012>. Citado na página 28.

IBGE. *Cidades e Estados*. 2020. Disponível em: <https://www.ibge.gov.br/cidades-e-estados>. Acesso em: 18 out 2020. Citado 2 vezes nas páginas 39 e 55.

ISO. *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*. [S.l.], 2013. Citado na página 39.

JUPYTER. *Project Jupyter*. 2020. Disponível em: <https://jupyter.org/>. Acesso em: 18 out 2020. Citado na página 21.

KERAS. *The Keras ecosystem*. 2020. Disponível em: <https://keras.io/getting_started/ecosystem/>. Acesso em: 18 out 2020. Citado na página 22.

KERAS. *Keras layers API*. 2020. Disponível em: <https://keras.io/api/layers/>. Acesso em: 18 out 2020. Citado na página 37.

KERAS. *Losses*. 2020. Disponível em: <https://keras.io/api/losses/>. Acesso em: 18 out 2020. Citado na página 26.

KERAS. *Why Keras?* 2020. Disponível em: <https://keras.io/why_keras>. Acesso em: 18 out 2020. Citado na página 23.

KETKAR, N. *Introduction to PyTorch. In: Deep Learning with Python*. 1. ed. New York: Apress, 2017. Citado na página 23.

KIM, B.-D.; KIM, S.-O. A new recommender system to combine content-based and collaborative filtering systems. *Journal of Database Marketing & Customer Strategy Management*, Springer Science and Business Media LLC, v. 8, n. 3, p. 244–252, abr. 2001. Disponível em: <https://doi.org/10.1057/palgrave.jdm.3240040>. Citado na página 25.

KIMNA-YOUNG. A study on the use of artificial intelligence chatbots for improving english grammar skills. *Journal of Digital Convergence*, v. 17, n. 8, p. 37–46, 2019. Acesso em: 18 out 2020. Citado na página 18.

Pearson's correlation coefficient. In: KIRCH, W. (Ed.). *Encyclopedia of Public Health*. Dordrecht: Springer Netherlands, 2008. p. 1090–1091. ISBN 978-1-4020-5614-7. Disponível em: <https://doi.org/10.1007/978-1-4020-5614-7_2569>. Citado na página 27.

KUMAR, K.; TAN, C. *Artificial Intelligence in Financial Distress Prediction*. 2004. Disponível em: <https://www.researchgate.net/publication/237301698_Artificial_Intelligence_in_Financial_Distress_Prediction>. Acesso em: 18 out 2020. Citado na página 17.

LAROUSSE. *Grande Enciclopédia Larousse Cultural*. São Paulo: Nova Cultural, 1999. Citado na página 18.

LECUN, Y. *Deep learning*. 2015. Disponível em: <https://www.nature.com/articles/nature14539>. Acesso em: 18 out 2020. Citado na página 21.

LEE, S.; YANG, J.; PARK, S.-Y. Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. In: *Discovery Science*. Springer Berlin Heidelberg, 2004. p. 396–402. Disponível em: <https://doi.org/10.1007/978-3-540-30214-8_36>. Citado na página 25.

LI, M.; WU, H.; ZHANG, H. Matrix factorization for personalized recommendation with implicit feedback and temporal information in social ecommerce networks. *IEEE Access*, v. 7, n. 1, p. 141268–141276, 2019. Citado na página 15.

LI, T.; LIU, A.; HUANG, C. A similarity scenario-based recommendation model with small disturbances for unknown items in social networks. *IEEE Access*, v. 4, p. 9251–9272, 2016. Citado na página 28.

LI, Y.; LU, L.; XUEFENG, L. A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in e-commerce. *Expert Systems with Applications*, Elsevier BV, v. 28, n. 1, p. 67–77, jan. 2005. Disponível em: <https://doi.org/10.1016/j.eswa.2004.08.013>. Citado 3 vezes nas páginas 15, 24 e 26.

LIGHTGBM. *Welcome to LightGBM's documentation!* 2020. Disponível em: <https://lightgbm.readthedocs.io/en/latest/>. Acesso em: 18 out 2020. Citado na página 23.

MCCORDUCK, P. *Machines Who Think*. 2. ed. Natick: A. K. Peters, Ltd., 2004. Citado na página 18.

MELVILLE, P.; MOONEY, R. J.; NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, v. 23, p. 187–192, 2002. Citado 2 vezes nas páginas 15 e 24.

MICHAELIS. *Dispersão*. 2020. Disponível em: <https://michaelis.uol.com.br/palavra/9OA8/dispers~ao>. Acesso em: 18 out 2020. Citado na página 26.

MICROSOFT. *Visual Studio Code - Code Editing. Redefined*. 2020. Disponível em: <https://code.visualstudio.com/>. Acesso em: 18 out 2020. Citado na página 22.

MITCHELL, T. *Machine Learning*. 1. ed. New York: McGraw Hill, 1997. Citado na página 20.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, v. 7, p. 21, 2013. Disponível em: <https://www.frontiersin.org/article/10.3389/fnbot.2013.00021>. Acesso em: 18 out 2020. Citado na página 23.

NHU, V.-H. et al. Effectiveness assessment of keras based deep learning with different robust optimization algorithms for shallow landslide susceptibility mapping at tropical area. *CATENA*, Elsevier BV, v. 188, p. 104458, maio 2020. Disponível em: <https://doi.org/10.1016/j.catena.2020.104458>. Citado na página 26.

NIKOLOPOULOS, C. *Expert Systems – Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York: Marcel Dekker, 1997. Citado na página 18.

ONU. *World Population Prospects - Population Division - United Nations*. 2020. Disponível em: <https://population.un.org/wpp/DataQuery/>. Acesso em: 18 out 2020. Citado na página 43.

ORENDORFF, A. *Global Ecommerce Statistics and Trends to Launch Your Business Beyond Borders*. 2019. Disponível em: <https://www.shopify.com/enterprise/global-ecommerce-statistics>. Acesso em: 18 out 2020. Citado na página 14.

OSORIO, F.; JOÃO, R.; BITTENCOURT, J. *Sistemas Inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens*. 2020. Disponível em: <https://www.researchgate.net/publication/228588719_Sistemas_Inteligentes_baseados_em_redes_neurais_artificiais_aplicados_ao_processamento_de_imagens>. Acesso em: 18 out 2020. Citado na página 19.

PATRÍCIO, D. I.; RIEDER, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*, v. 153, p. 69–81, 2018. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168169918305829>. Acesso em: 18 out 2020. Citado na página 18.

PEIXEIRO, M. *How to Build a Deep Neural Network Without a Framework*. 2019. Disponível em: <https://towardsdatascience.com/how-to-build-a-deep-neural-network-without-a-framework-5d46067754d5>. Acesso em: 18 out 2020. Citado na página 22.

PEREIRA, L. M. *Inteligência Artificial Mito e Ciência*. 1988. Disponível em: <https://www.researchgate.net/publication/237130636_Inteligencia_Artificial_Mito_e_Ciencia>. Acesso em: 18 out 2020. Citado na página 19.

PEREZ, S. *Pandemic helped drive Walmart e-commerce sales up 97% in second quarter*. 2020. Disponível em: <https://techcrunch.com/2020/08/18/pandemic-helped-drive-walmart-e-commerce-sales-up-97-in-second-quarter>. Acesso em: 18 out 2020. Citado na página 14.

PYCHARM. *PyCharm - The Python IDE for Professional Developers*. 2020. Disponível em: <https://www.jetbrains.com/pycharm/>. Acesso em: 18 out 2020. Citado na página 22.

RAUBER, T. *Redes Neurais Artificiais*. 2020. Disponível em: <https://www.researchgate.net/publication/228686464_Redes_neurais_artificiais>. Acesso em: 18 out 2020. Citado na página 19.

REN, X.; CHEN, Y. How can artificial intelligence help with space missions - a case study: Computational intelligence-assisted design of space tether for payload orbital transfer under uncertainties. *IEEE Access*, v. 7, p. 161449–161458, 2019. Citado na página 17.

RHEUDE, J. *Will E-Commerce Benefit from Machine Learning or Face a New Threat?* 2019. Disponível em: <https://neilpatel.com/blog/will-e-commerce-benefit-from-machine-learning>. Acesso em: 18 out 2020. Citado na página 14.

RICH, E.; KNIGHT, K. *Inteligência Artificial*. 2. ed. São Paulo: Makron Books do Brasil, 1993. Citado na página 19.

RUSIECKI, A. Trimmed categorical cross-entropy for deep learning with label noise. *Electronics Letters*, Institution of Engineering and Technology (IET), v. 55, n. 6, p. 319–320, mar. 2019. Disponível em: <https://doi.org/10.1049/el.2018.7980>. Citado na página 26.

S, D.; CHITTURI, B. Deep neural approach to fake-news identification. *Procedia Computer Science*, 2020. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050920307420>. Citado na página 17.

SBVC. *Novos Hábitos Digitais em Tempos de COVID-19*. 2020. Disponível em: <http://sbvc.com.br/novos-habitos-digitais-em-tempos-de-covid-19/>. Acesso em: 18 out 2020. Citado na página 14.

SHOJA, B.; TABRIZI, N. Customer reviews analysis with deep neural networks for e-commerce recommender systems. *IEEE Access*, v. 7, p. 119121–119130, 2019. Citado na página 28.

SKANSI, S. *An Introduction to Deep Learning - From Logical Calculus to Artifical Intelligence*. 1. ed. [S.l.]: Springer, 2018. Citado na página 20.

SPYDER. *Spyder - The Scientific Python Development Environment*. 2020. Disponível em: <https://www.spyder-ide.org/>. Acesso em: 18 out 2020. Citado na página 21.

STACKOVERFLOW. *StackOverflow Developer Survey Results 2019*. 2020. Disponível em: <https://insights.stackoverflow.com/survey/2019#development-environments-and-tools>. Acesso em: 18 out 2020. Citado na página 22.

STIMPSON, A. J.; CUMMINGS, M. L. Assessing intervention timing in computer-based education using machine learning algorithms. *IEEE Access*, 2014. Disponível em: <https://ieeexplore.ieee.org/document/6730683>. Acesso em: 18 out 2020. Citado na página 20.

TANNER, G. *Keras Tutorial #10 - Book Recommendation System*. 2018. Disponível em: <https://www.youtube.com/watch?v=4vwNkHFuZBk>. Acesso em: 18 out 2020. Citado 4 vezes nas páginas 35, 37, 38 e 59.

TATALOVIC, M. Ai writing bots are about to revolutionise science journalism: we must shape how this is done. *JCOM 17 (01)*, 2018. Disponível em: <https://doi.org/10.22323/2.17010501>. Citado na página 17.

TENSORFLOW. *Introduction to TensorFlow*. 2020. Disponível em: <https://www.tensorflow.org/learn>. Acesso em: 18 out 2020. Citado na página 23.

TENSORFLOW. *Introduction to Tensors*. 2020. Disponível em: <https://www.tensorflow.org/guide/tensor>. Acesso em: 18 out 2020. Citado na página 22.

THEANO DEVELOPMENT TEAM. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, 2016. Disponível em: <http://arxiv.org/abs/1605.02688>. Acesso em: 18 out 2020. Citado na página 23.

THEANO DEVELOPMENT TEAM. *Theano Docs - Welcome*. 2020. Disponível em: <http://deeplearning.net/software/theano/>. Acesso em: 18 out 2020. Citado na página 23.

WANG, W.; SIAU, K. L. *Living with Artificial Intelligence: Developing a Theory on Trust in Health Chatbots - Research in Progress*. 2018. Disponível em: <https://aisel.aisnet.org/sighci2018/4/>. Acesso em: 18 out 2020. Citado na página 18.

WANG, X. et al. Neural graph collaborative filtering. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2019. Disponível em: <https://doi.org/10.1145/3331184.3331267>. Citado na página 28.

WANGOO, D. P. Artificial intelligence techniques in software engineering for automated software reuse and design. In: *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. [S.l.: s.n.], 2018. p. 1–4. Citado na página 17.

WASKOM, M. *seaborn.kdeplot — seaborn 0.11.0 documentation*. 2020. Disponível em: <https://seaborn.pydata.org/generated/seaborn.kdeplot.html#seaborn.kdeplot>. Acesso em: 18 out 2020. Citado na página 39.

WATERMAN, D. *A guide to Expert Systems*. Boston: Addison-Wesley, 1985. Citado na página 18.

WU, Q.; ZHAO, P.; CUI, Z. Visual and textual jointly enhanced interpretable fashion recommendation. *IEEE Access*, v. 8, p. 68736–68746, 2020. Citado na página 28.

YAGER, R. R. Targeted e-commerce marketing using fuzzy intelligent agents. *IEEE Intelligent Systems and their Applications*, v. 15, n. 6, p. 42–45, 2000. Disponível em: <https://ieeexplore.ieee.org/abstract/document/895859>. Citado na página 28.

YANG, Y.; JANG, H. J.; KIM, B. A hybrid recommender system for sequential recommendation: Combining similarity models with markov chains. *IEEE Access*, p. 1–1, 2020. Citado na página 28.

# Anexos

# ANEXO A – Script do modelo de recomendação

```
import pandas as pd
from keras.models import Model
from keras.layers import Input, Embedding, Flatten, Dense, Concatenate
from sklearn.model_selection import train_test_split

# carregar arquivo, dividir dataset em treinamento/teste
dataset = pd.read_csv('arquivo.csv')
train, test = train_test_split(dataset, test_size=0.2, random_state=42)

# criar embedding para produtos
n_prods = len(dataset['product_id'])
prod_input = Input(shape=[1], name="Prod-Input")
prod_embedding = Embedding(n_prods+1, 5, name="Prod-Embedding")(prod_input)
prod_vec = Flatten(name="Flatten-Prods")(prod_embedding)

# criar embedding para segmentos de cliente
client_input = Input(shape=[1], name="Client-Input")
n_clients = len(dataset['client_id'])
client_embedding = Embedding(n_clients+1, 5,
    name="Client-Embedding")(client_input)
client_vec = Flatten(name="Flatten-Clients")(client_embedding)

# concatenar conjuntos
conc = Concatenate()([prod_vec, client_vec])

# adicionar camadas
fc1 = Dense(128, activation='relu')(conc)
fc2 = Dense(32, activation='relu')(fc1)
out = Dense(1)(fc2)

# treinamento
model2 = Model([client_input, prod_input], out)
model2.compile('adam', 'mean_squared_error')
```