

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-  
47p. : 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. Palavra-chave1. 2. Palavra-chave2. 2. Palavra-chave3. I. Orientador. II. Universidade xxx. III. Faculdade de xxx. IV. 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, 24 de julho de 2020:

---

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

---

**TODO banca**  
Banca

Santa Rosa - RS  
2020

*TODO dedicatória*

# Agradecimientos

TODO agradecimientos

*“TODO quote”*  
*(TODO quote)*

# 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 a partir de diferentes conjuntos de regras baseadas no volume de vendas dos mesmos. Nesse estudo, os resultados de recomendação são analisados e comparados no que se refere a suas características estatísticas.

**Palavras-chave:** Inteligência Artificial. Machine Learning. Deep Learning. Sistemas de Recomendação. Big Data. 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 type, does not depend on numerical ratings given by customers, instead inferring the popularity of the products through different sets of rules based on their sales volume. In this study, the recommendation results are analyzed and compared with regard to their statistical features.

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

# Lista de abreviaturas e siglas

CPU	<i>Central Processing Unit</i>
ERP	<i>Enterprise Resource Planning</i>
FC	<i>Filtragem Colaborativa</i>
GPU	<i>Graphics Processing Unit</i>
IA	<i>Inteligência Artificial</i>
IDE	<i>Integrated Development Environment</i>
LTS	<i>Long-Term Support</i>
ML	<i>Machine Learning</i>
RAM	<i>Random Access Memory</i>
RNA	<i>Rede Neural Artificial</i>
UCI	<i>University of California, Irvine</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Tema</b>	<b>12</b>
<b>1.2</b>	<b>Questões</b>	<b>13</b>
<b>1.3</b>	<b>Justificativa</b>	<b>13</b>
<b>1.4</b>	<b>Objetivos</b>	<b>13</b>
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>15</b>
<b>2.1</b>	<b>Conceitos e áreas de conhecimento</b>	<b>16</b>
2.1.1	Inteligência Artificial	16
2.1.2	Redes Neurais	17
2.1.3	Machine Learning	18
2.1.4	Deep Learning	18
<b>2.2</b>	<b>Ambientes de desenvolvimento</b>	<b>19</b>
<b>2.3</b>	<b>Bibliotecas</b>	<b>20</b>
<b>2.4</b>	<b>Sistemas de recomendação</b>	<b>21</b>
2.4.1	Baseado em conteúdo	22
2.4.2	Filtragem colaborativa	23
2.4.3	Sistemas Híbridos	23
<b>2.5</b>	<b>Métricas e parâmetros</b>	<b>24</b>
2.5.1	Precisão (acc)	24
2.5.2	Perda (loss)	24
2.5.3	Dispersão	24
2.5.4	Coeficientes de Pearson	25
<b>2.6</b>	<b>Métodos de codificação</b>	<b>25</b>
2.6.1	One-hot encoding	25
2.6.2	Embedding	26
<b>2.7</b>	<b>Trabalhos Relacionados</b>	<b>26</b>
<b>2.8</b>	<b>Considerações do Capítulo</b>	<b>27</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>28</b>
<b>3.1</b>	<b>Metodologia Utilizada</b>	<b>28</b>
<b>3.2</b>	<b>Ambiente de Testes</b>	<b>29</b>
<b>3.3</b>	<b>Considerações do Capítulo</b>	<b>30</b>
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>31</b>
<b>4.1</b>	<b>Escolha do dataset</b>	<b>31</b>

4.2	Estudo de sistemas de recomendação . . . . .	31
4.3	Definição de critério de avaliação . . . . .	32
4.4	Geração de recomendações . . . . .	34
5	RESULTADOS . . . . .	36
5.1	Avaliações . . . . .	36
5.2	Recomendações - Cenário 1 . . . . .	36
6	CONCLUSÃO . . . . .	39
6.1	Conclusão . . . . .	39
	REFERÊNCIAS . . . . .	40
	ANEXOS	46
	ANEXO A – CÓDIGO - REDE NEURAL DENSE DO KERAS . . .	47

# 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 a pandemia do novo Coronavírus, observou-se um aumento das vendas online em várias partes do mundo. Enquanto no Brasil 61% dos consumidores aumentaram 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).

Esse projeto propõe a utilização de *Machine Learning* e *Deep Learning* para inferir as preferências de consumidores e fazer sugestões 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 sugestões 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 nesse 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*.

## 1.1 Tema

O trabalho tem como tema o desenvolvimento de uma rede neural com a finalidade de gerar sugestões de produtos para usuários de um *e-commerce* com base em seu histórico

de compra.

## 1.2 Questões

A questão central dessa pesquisa: 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. Qual é a estrutura de rede neural mínima necessária para realizar essas inferências?
2. Como os diferentes métodos de geração de avaliações interferem nas sugestões geradas?

## 1.3 Justificativa

Com a recente popularização de técnicas de Inteligência Artificial e *Machine Learning* e sua ampla implementação em aplicações empresariais, torna-se relevante 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 inferi-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.

## 1.4 Objetivos

Este trabalho tem os seguintes objetivos:

1. Obtenção de *datasets* de pedidos (*Big Data*) que não contenham avaliações de clientes, mas somente dados básicos sobre o item 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 otimizada para os problemas em questão.
6. Aplicação da IA treinada para gerar sugestões de produtos a partir dos *datasets* selecionadas.

## 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 sugestões em aplicações de *e-commerce*, foco principal desse 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 adquirí-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 a 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), 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 M., 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".

Segundo MITCHELL (1997, tradução nossa) "*Machine Learning* é uma sub-área da Inteligência Artificial 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.

Quanto 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 no 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 entremeadada 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* “descobre estruturas intrincadas em grandes conjuntos 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, equações, visualizações e textos explicativos. 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 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

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 matemáticas como TensorFlow e Theano (BROWNLEE, 2019).

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, 2020b, tradução nossa). O Keras se integra também com outros recursos e 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, 2020, 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 tem 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 Baseado 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 sugestões, 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:

- Cold start (partida fria): 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 e-commerce 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 Sistemas Híbridos

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 e parâmetros

Essa seção visa descrever métricas que são utilizadas para entender o comportamento e medir o sucesso de modelo de *Machine Learning*, bem como parâmetros utilizados. 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 Precisão (acc)

Métrica que "calcula a frequência com a qual as predições coincidem com as saídas"(KERAS, 2020a). Por exemplo, se para um dataset de 100 registros o modelo é capaz de prever as saídas corretas em 30% dos casos, a métrica tem valor 0,3 pois  $30/100 = 0,3$ .

### 2.5.2 Perda (loss)

As funções de perda "computam a quantidade que um modelo deveria minimizar durante o treinamento"(KERAS, 2020c). Enquanto para modelos de regressão a função *Mean squared Error (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).

### 2.5.3 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).

#### 2.5.4 Coeficientes de Pearson

Métricas utilizadas para definir o nível de assimetria de uma distribuição na qual média, mediana, moda e desvio padrão tenham sido medidos. Existem 2 fórmulas para obter o coeficiente, uma que utiliza a moda como referência (1º coeficiente) e outra a mediana (2º coeficiente) (WEISSTEIN, 2020). As fórmulas são detalhadas na Figura 1.

$$\frac{(\text{mean} - \text{mode})}{\sigma}, \quad \frac{3(\text{mean} - \text{median})}{\sigma},$$

Figura 1 – Fórmulas para o 1º (esq.) e 2º (dir.) coeficientes de Pearson

Quando uma distribuição é simétrica, ou seja, não possui valores em excesso tendendo acima ou abaixo da média, os coeficientes tendem a zero. Como exemplifica PEARSON (1895, tradução nossa), há diversos fatores que podem influenciar na tendência de uma determinada variável:

Em primeiro lugar, o material medido pode ser heterogêneo e constituir de uma mistura de 2 ou mais materiais homogêneos. (...) A segunda classe de curvas de frequência ocorre no caso de material homogêneo quando a tendência de desvio para um lado de média é desigual a tendência para o outro lado. Curvas desse tipo aparecem em vários estudos físicos, econômicos e biológicos, por exemplo, em curvas de frequência para a altitude de um barômetro, em preços e taxas de juros de títulos mobiliários (...)

## 2.6 Métodos de codificação

Métodos comuns para representação codificada de dados categóricos no conteúdo de *Machine Learning* serão descritos nessa seção.

### 2.6.1 One-hot encoding

Método de codificação no qual cada valor é representado em sua forma codificada por um vetor de números inteiros no qual uma das posições corresponde ao valor 1 e todas as outras ao valor 0. GOOGLE DEVELOPERS (2020, tradução nossa) exemplifica o uso desse método com uma aplicação prática:

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 esparsos 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 a sugestões 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 tendo como base somente no volume de vendas de um produto dentro de 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 informações sobre a satisfação do consumidor no contexto pós-venda.

Padoin: vai citando mais trabalhos e no final precisamos dizer o que o teu tem de diferente dos demais

Gabriel: alterei essa seção para citar mais trabalhos voltado a e-commerce e no último parágrafo destaco a diferença do meu para os demais

## 2.8 Considerações do Capítulo

Nesse capítulo foram apresentados áreas de conhecimento, tecnologias, técnicas e trabalho acadêmicos que possuem relação com o tema abordado nesse trabalho. No próximo capítulo serão apresentados os materiais e métodos e serem utilizados.

## 3 Materiais e Métodos

Nesse 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

Visto que um dos objetivos do estudo é aplicar a IA treinada para gerar sugestões de produtos, esse se enquadra na classe de pesquisa qualitativa, ou seja, seus resultados irão nos ajudar a entender os comportamentos dos consumidores contidos na amostra de dados a ser utilizada. Contudo, a parametrização da rede neural muitas vezes envolve valores discretos, e como um dos objetivos é também encontrar a melhor configuração de rede para esse caso de uso, os resultados serão um misto de dados qualitativos e quantitativos.

Assim como descrito no trabalho de BURKE (2002), a proposta desse trabalho será desenvolver uma aplicação que gere sugestões 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 híbrido. 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 por 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 cliente ou grupo de clientes. Essa popularidade será representado por uma avaliação numérica. Se uma combinação de cliente/produto recebe uma avaliação de valor alto, isso significará que o produto é uma boa recomendação para aquele cliente. Se a avaliação for baixa, isso significará que o produto pode não ser a recomendação mais adequada. Cada método definirá as avaliações de acordo com diferentes regras, agrupando os clientes por diferentes critérios.

Os *datasets* selecionados no início do estudo serão processados por um algoritmo que aplicará as regras definidas pelos métodos, e para cada método o *dataset* de entrada

resultará em outro de saída, contendo clientes, produtos e as avaliações inferidas. O conjunto de dados resultante desse projeto será utilizado para alimentar uma rede neural. As combinações de cliente/produto serão as entradas, e as avaliações serão as saídas.

A rede neural será criada através da biblioteca Keras em Python. 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 "células" e permite a inclusão de comentários com texto formatado, imagens e outros recursos visuais junto ao código, o Visual Studio Code funciona de forma mais parecida com um editor de texto 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 parametrizações referentes ao processo de treinamento em si (ex.: número de épocas, *batch size*) ou à rede (ex.: número de neurônios, camadas 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 sugestões.

Por fim, as avaliações e sugestões geradas por cada método serão comparadas e então analisadas qualitativamente em relação aos dados que as originaram. Ou seja, a partir de uma análise manual dos dados, será possível constatar se as previsões geradas foram relevantes ou não. Caso não sejam, novos testes serão realizados até que seja encontrado o modelo ideal. Essa pesquisa possui, portanto, caráter aplicado e exploratório.

## 3.2 Ambiente de Testes

Para desenvolvimento e prototipação foi utilizado um notebook Acer Nitro 5 com as seguintes configurações de *hardware*:

- Processador: Intel Core i5-8300H @ 2.30GHz, 4 núcleos, 8 processadores lógicos
- RAM: 8GB
- Armazenamento: 1TB

O seguinte ambiente de *software* foi configurado no notebook:

- Sistema Operacional: Windows 10 Education

- Versão do Python: 3.7.6
- Editor de texto: Jupyter Notebook v6.0.3

Os testes descritos no trabalho foram efetuados em um servidor da Unijuí com as seguintes configurações de *hardware*:

- Processador: Intel Core i7-8700 @ 3.20GHz, 6 núcleos, 12 processadores lógicos
- RAM: 16GB
- Armazenamento: 450GB

O seguinte ambiente de *software* foi configurado no servidor:

- Sistema Operacional: Ubuntu 18.04.4 LTS
- Versão do Python: 3.6.9
- Editor de texto: nano

### 3.3 Considerações do Capítulo

Nesse 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

Nessa 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 do dataset

Para o trabalho foram escolhidos 3 *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 mantêm 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 1023 votos positivos de usuários e "Usabilidade" avaliada como nota 10/10.<sup>1</sup>

Nome	Nº de 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

Tabela 1 – Dados sobre os datasets selecionados para o trabalho

### 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

<sup>1</sup> Dados referentes ao dia 19/09/2020



produto e não dados do produto em si. Contudo, há diferentes maneiras de implementar sistemas de 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 essas entradas categóricas e suas respectivas saídas, as avaliações, como exemplificado na figura 2.

Nesse trabalho, optou-se por seguir a implementação de Tanner, visto que esta apresenta várias vantagens do ponto de vista de desempenho e manutenção de código que serão explicadas em mais detalhes na Seção 4.4.

Nome	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

Figura 2 – Exemplo do sistema de recomendação de Tanner

### 4.3 Definição de critério de avaliação

Para que se possa utilizar a abordagem ilustrada por TANNER (2018) com qualquer *datasets*, é necessário que avaliações numéricas (por exemplo, notas de 1 a 5) estejam relacionadas às transações analisadas. Contudo, não é todo *dataset* que possui esse tipo de dado. Os conjuntos listados na Seção 4.1, por exemplo, elencam somente pares de clientes/produtos, e portanto o único dado que pode ser inferido desses *datasets* é a frequência de compra por cliente. Visto que não há informação detalhada sobre o produto, abordagens baseadas em conteúdo (características do produto) ou objetivos (para quê o produto serve) não são viáveis.

Foram portanto definidos 3 métodos de criar avaliações para os registros nesses *datasets*, baseados na frequência de compra em diferentes agrupamentos de clientes. Quanto mais popular um produto é, mais passível de recomendação ele se torna. As regras de avaliação foram baseadas nas definições para filtragem colaborativa (FC) no contexto de sistemas de recomendação.

- FC por Categoria: se o cliente não comprou e o produto é popular dentro da categoria na qual o cliente está, define avaliação alta. Isso aumenta a probabilidade de recomendação para produtos populares na categoria (Figura 3).

- FC Intercategorias: se o cliente em questão não comprou o produto e este é popular dentro da categoria do cliente, define avaliação alta. Se o cliente não comprou, o produto não é popular, mas outro cliente da mesma categoria comprou, define avaliação média. Isso aumenta a probabilidade de recomendação para produtos populares na categoria enquanto também destaca produtos menos populares adquiridos por clientes similares. (Figura 5).
- FC Geral: segue as mesmas regras do FC por Categoria, mas considerando a popularidade dos produtos entre todos os clientes. Nesse caso, se o cliente não comprou o produto e este é popular no geral, define avaliação alta. Isso aumenta a probabilidade de recomendação para produtos populares no geral.

<b>Produto/Cliente</b>	<b>Já comprou</b>	<b>Nunca comprou</b>
<b>É popular na categoria</b>	<b>2</b>	<b>4</b>
<b>Não é popular na categoria</b>	<b>1</b>	<b>3</b>

Figura 3 – Avaliação FC geral ou por categoria.

<b>Produto/Cliente</b>	<b>Já comprou</b>	<b>Nunca comprou</b>	
<b>É popular na categoria</b>	<b>2</b>	<b>4</b>	
<b>Não é popular na categoria</b>	<b>1</b>	<b>Comprado por outro cliente na categoria</b>	<b>3</b>
		<b>Não comprado por outro cliente na categoria</b>	<b>2</b>

Figura 4 – Avaliação FC por intercategorias

Tendo os métodos de avaliação definidos, sua lógica foi descrita em *scripts* Python. Cada *script* contém instruções para percorrer todos os registros em cada um dos *datasets* selecionados, analisar os produtos mais vendidos de acordo com uma categorização de cliente e produzirem um *dataset* de saída contendo 3 colunas: código do cliente, código do produto e avaliação.

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 terem sido produzidas avaliações para os 3 *datasets* selecionados, esses valores foram analisados sob o ponto de vista da estatística descritiva, de forma a entender como cada método avaliou os dados contidos no dataset. O resultados dessa análise serão explicados em mais detalhes na Seção 5.

## 4.4 Geração de recomendações

Após a geração de avaliações conforme os métodos definidos na Seção 4.3, os *datasets* resultantes foram utilizados como entrada para o treinamento da rede neural descrita por Tanner. A rede consiste em 4 camadas iniciais, que fazem o processo de *embedding* dos produtos e clientes, e 3 camadas do tipo Dense, que permitem à rede relacionar as entradas categóricas com as avaliações durante o treinamento.

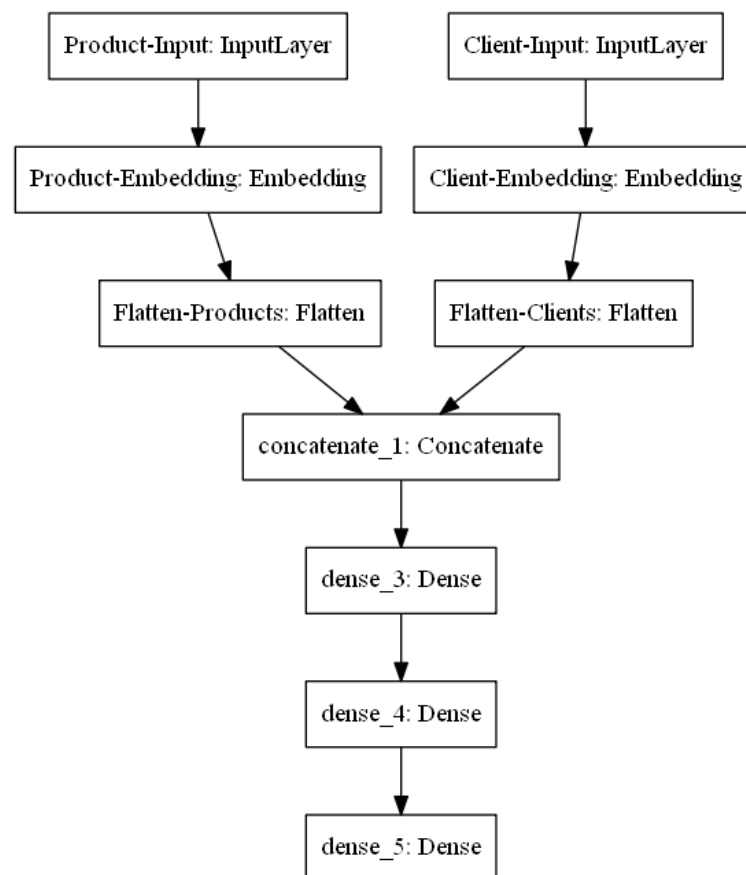


Figura 5 – Estrutura de rede neural utilizada

Essa estrutura de rede foi escolhida para o trabalho pois apresenta as 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 memória RAM e processamento
- 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 comprová-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 manualmente com pares de produtos/clientes do *dataset* para atestar que o treinamento ocorreu de forma bem sucedida.

## 5 Resultados

### 5.1 Avaliações

Após terem sido geradas para todos os datasets, as medidas estatísticas de posição e dispersão dessas avaliações foram compiladas (Tabela 2). A partir da análise do 1º coeficiente de Pearson, pode-se concluir que as avaliações para todos os conjuntos apresentaram assimetria negativa, ou seja, valores de avaliação altos aparecem com mais frequência na amostra do que os baixos. Isso se comprova na observação da mediana e moda, que foi maior ou igual a 3 para todos os datasets.

A partir desses dados, pode-se concluir que os produtos presentes nos datasets analisados apresentaram alto volume de vendas na maioria dos casos. Caso houvesse um dataset repleto de produtos com baixo volume de vendas, as avaliações teriam comportamento inverso, tendendo a valores mais baixos (assimetria positiva).

Método	Dataset	Média	Mediana	Moda	Coef. Pearson
FC-Inter-Categoria	Brazilian retail	3,41377	4	4	-184,38%
FC-Geral	Brazilian retail	3,52950	4	4	-174,78%
FC-Categoria	Brazilian retail	3,55913	4	4	-167,57%
FC-Inter-Categoria	Online Retail 1	2,89565	3	3	-87,41%
FC-Inter-Categoria	Online Retail 2	2,90716	3	3	-84,47%
FC-Geral	Online Retail 1	2,99107	3	3	-13,03%
FC-Categoria	Online Retail 1	2,99187	3	3	-11,77%
FC-Geral	Online Retail 2	2,99361	3	3	-11,06%
FC-Categoria	Online Retail 2	2,99415	3	3	-10,02%

Tabela 2 – Atributos estatísticos das previsões geradas

Também para determinar se as avaliações pertencem a uma distribuição normal, a amostra foi analisada utilizando o teste de Anderson-Darling. O valor obtido foi de 1811063,02. Para um intervalo de confiança de 95%, pode-se rejeitar a hipótese de normalidade dos dados.

Gabriel: o teste que fiz com o coeficiente de pearson não é suficiente para determinar que a distribuição não é normal? é preciso definir esses testes na fundamentação?

### 5.2 Recomendações - Cenário 1

No primeiro cenário de testes, recomendações foram geradas para o *dataset* Online Retail utilizando a abordagem de FC por categoria. Nesse caso, o critério considerado

para a categorização foi o país de origem do cliente.

Os 5 produtos que mais receberam avaliação 4 conforme a análise constam na Tabela 3. Nessa amostra, todos os produtos foram considerados boas recomendações para 8 em cada 10 clientes da base. A partir disso, pode-se concluir que esse produtos possuem alta demanda em todos os países analisados. Os produtos nessa amostra, assim como os demais no *dataset*, são itens de utilidade doméstica (bandeja e potes para bolo e sacolas térmicas para alimentos).

Quanto aos produtos que mais frequentemente receberam avaliação 3 (Tabela 4), pode-se concluir que foram os mesmos para todos os clientes. Isso indica que há produtos no conjunto que não foram adquiridos por nenhum cliente, e que por consequência também não figuraram entre os mais vendidos nos países de origem dos clientes. Esses produtos correspondem a 27,22% do total, ou seja, essa amostra engloba 1108 dos 4070 produtos presentes na base.

Analisando os produtos que mais frequentemente receberam avaliações 2 e 1 (Tabelas 5 e 6), pode-se concluir que o produto que mais frequentemente recebeu notas baixas foi assim avaliado para somente 4,6% dos clientes. Esse cenário indica que a maior parte dos produtos na base é vendido com frequência igual ou próxima, enquanto somente uma parcela menor tem baixo de volume venda em cada país analisado. Essa análise, portanto, condiz também com a encontrada para os itens com avaliação 3.

Portanto, pode-se concluir que a abordagem de FC por Categoria aplicada ao *dataset* citado trabalhou de forma a avaliar com pontuação alta todos os produtos populares nos países analisados, e com valor ligeiramente mais baixo os produtos que nenhum dos clientes comprou. Dessa forma, o sistema de recomendação é capaz de incentivar clientes antigos a recomprarem produtos populares, clientes novos a comprarem esses mesmos produtos, bem tornar produtos com baixo volume de venda mais populares entre a base de consumidores como um todo. Todos esses cenários são positivos para os vendedores que ofertam seus itens em *e-commerce*.

ID produto	Nome produto	Nº avaliações 4	% clientes
180	LUNCH BAG RED RETROSPOT	3916	89,55
1348	REGENCY CAKESTAND 3 TIER	3866	88,41
1314	LUNCH BAG SUKI DESIGN	3827	87,51
182	LUNCH BAG BLACK SKULL	3818	87,31
1631	SET OF 3 CAKE TINS PANTRY DESIGN	3806	87,03

Tabela 3 – Análise dos 5 produtos que mais receberam avaliação 4

ID produto	Nome produto	Nº avaliações 3	% clientes
4068	GIFT VOUCHER £50.00	4373	100,00
2610	BIG POLKADOT MUG	4373	100,00
2601	CHRISTMAS TREE T-LIGHT HOLD	4373	100,00
2600	FOLKART HEART NAPKIN RINGS	4373	100,00
2603	ART METAL HEART T-LIGHT HOLDER	4373	100,00

Tabela 4 – Análise dos 5 produtos que mais receberam avaliação 3

ID produto	Nome produto	Nº avaliações 2	% clientes
3536	HANGING HEART T-LIGHT HOLDER	432	9,88
1348	REGENCY CAKESTAND 3 TIER	393	8,99
3305	ASSORTED COLOUR BIRD ORNAMENT	339	7,75
4062	POSTAGE	253	5,79
3515	JUMBO BAG RED RETROSPOT	229	5,24

Tabela 5 – Análise dos 5 produtos que mais receberam avaliação 2

ID produto	Nome produto	Nº avaliações 1	% clientes
339	REX CASH+CARRY JUMBO SHOPPER	208	4,76
1862	JAM MAKING SET WITH JARS	179	4,09
1043	PAPER CHAIN KIT 50'S CHRISTMAS	172	3,93
1128	VICTORIAN GLASS HANGING T-LIGHT	170	3,89
3346	ANTIQUE SILVER TEA GLASS ETCHED	160	3,66

Tabela 6 – Análise dos 5 produtos que mais receberam avaliação 1

## 6 Conclusão

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 o volume de vendas de um item e os clientes e segmentos relacionados a essa venda. Em trabalhos futuros, pretende-se validar esse modelo através de sua implementação em uma aplicação de *e-commerce* real, bem como testar diferentes abordagens para geração de avaliações e recomendações.



# 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](https://www.researchgate.net/publication/247834709_Increasing_effectiveness_in_e-commerce_Recommendations_applying_intelligent_agents)>. Citado 2 vezes nas páginas 22 e 26.
- ADAM M., W. M. . B. 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 16.
- AGGARWAL, C. C. *Neural Networks and Deep Learning - A Textbook*. 1. ed. Cham: Springer, 2018. Citado na página 19.
- 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 22.
- 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 15.
- ALPAYDIN, E. *Introduction to Machine Learning*. 4. ed. Cambridge: MIT, 2018. Citado na página 18.
- 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 22 e 23.
- 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 18.
- 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 20.
- 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 22, 26 e 28.
- 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 31.
- 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 13 e 23.

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 16.

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 15.

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](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 12.

ERTEL, W. *An Introduction to Artificial Intelligence*. 2. ed. Cham: Springer, 2017. Citado 2 vezes nas páginas 16 e 17.

FRESSATO, E. P. *Incorporação de metadados semânticos para recomendação no cenário de partida fria*. Tese (Doutorado), 2019. Disponível em: <<https://doi.org/10.11606/d.55.2019.tde-09082019-134753>>. Citado 2 vezes nas páginas 22 e 23.

GOOGLE DEVELOPERS. *Machine Learning Glossary*. 2020. Disponível em: <<https://developers.google.com/machine-learning/glossary>>. Acesso em: 18 out 2020. Citado 2 vezes nas páginas 25 e 26.

HAYKIN, S. *Redes neurais: princípios e prática*. 1. ed. Porto Alegre: Bookman, 2001. Citado na página 17.

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 26.

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 26.

JUPYTER. *Project Jupyter*. 2020. Disponível em: <<https://jupyter.org/>>. Acesso em: 18 out 2020. Citado na página 19.

KERAS. *Accuracy metrics - Accuracy class*. 2020. Disponível em: <[https://keras.io/api/metrics/accuracy\\_metrics/#accuracy-class](https://keras.io/api/metrics/accuracy_metrics/#accuracy-class)>. Acesso em: 18 out 2020. Citado na página 24.

KERAS. *The Keras ecosystem*. 2020. Disponível em: <[https://keras.io/getting\\_started/ecosystem/](https://keras.io/getting_started/ecosystem/)>. Acesso em: 18 out 2020. Citado na página 20.

KERAS. *Losses*. 2020. Disponível em: <<https://keras.io/api/losses/>>. Acesso em: 18 out 2020. Citado na página 24.

KERAS. *Why Keras?* 2020. Disponível em: <[https://keras.io/why\\_keras](https://keras.io/why_keras)>. Acesso em: 18 out 2020. Citado na página 21.

KETKAR, N. *Introduction to PyTorch*. In: *Deep Learning with Python*. 1. ed. New York: Apress, 2017. Citado na página 21.

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 23.

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 16.

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](https://www.researchgate.net/publication/237301698_Artificial_Intelligence_in_Financial_Distress_Prediction)>. Acesso em: 18 out 2020. Citado na página 15.

LAROUSSE. *Grande Enciclopédia Larousse Cultural*. São Paulo: Nova Cultural, 1999. Citado na página 16.

LECUN, Y. *Deep learning*. 2015. Disponível em: <<https://www.nature.com/articles/nature14539>>. Acesso em: 18 out 2020. Citado na página 19.

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](https://doi.org/10.1007/978-3-540-30214-8_36)>. Citado na página 23.

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 13.

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 26.

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 13, 22 e 24.

LIGHTGBM. *Welcome to LightGBM's documentation!* 2020. Disponível em: <<https://lightgbm.readthedocs.io/en/latest/>>. Acesso em: 18 out 2020. Citado na página 21.

MCCORDUCK, P. *Machines Who Think*. 2. ed. Natick: A. K. Peters, Ltd., 2004. Citado na página 16.

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 13 e 22.

MICHAELIS. *Dispersão*. 2020. Disponível em: <<https://michaelis.uol.com.br/palavra/90A8/dispers~ao>>. Acesso em: 18 out 2020. Citado na página 24.

MICROSOFT. *Visual Studio Code - Code Editing. Redefined*. 2020. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 18 out 2020. Citado na página 20.

MITCHELL, T. *Machine Learning*. 1. ed. New York: McGraw Hill, 1997. Citado na página 18.

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 21.

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 24.

NIKOLOPOULOS, C. *Expert Systems – Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York: Marcel Dekker, 1997. Citado na página 16.

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 12.

OSORIO, F.; JOÃO, R.; BITTENCOURT. *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](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 17.

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 16.

PEARSON, K. X. contributions to the mathematical theory of evolution.—II. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. (A.)*, The Royal Society, v. 186, p. 343–414, dez. 1895. Disponível em: <<https://doi.org/10.1098/rsta.1895.0010>>. Citado na página 25.

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 20.

PEREIRA, L. M. *Inteligência Artificial Mito e Ciência*. 1988. Disponível em: <[https://www.researchgate.net/publication/237130636\\_Inteligencia\\_Artificial\\_Mito\\_e\\_Ciencia](https://www.researchgate.net/publication/237130636_Inteligencia_Artificial_Mito_e_Ciencia)>. Acesso em: 18 out 2020. Citado na página 17.

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 12.

PYCHARM. *PyCharm - The Python IDE for Professional Developers*. 2020. Disponível em: <<https://www.jetbrains.com/pycharm/>>. Acesso em: 18 out 2020. Citado na página 20.

RAUBER, T. *Redes Neurais Artificiais*. 2020. Disponível em: <[https://www.researchgate.net/publication/228686464\\_Redes\\_neurais\\_artificiais](https://www.researchgate.net/publication/228686464_Redes_neurais_artificiais)>. Acesso em: 18 out 2020. Citado na página 17.

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 15.

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 12.

RICH, E.; KNIGHT, K. *Inteligência Artificial*. 2. ed. São Paulo: Makron Books do Brasil, 1993. Citado na página 17.

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 24.

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 15.

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 12.

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 26.

SKANSI, S. *An Introduction to Deep Learning - From Logical Calculus to Artificial Intelligence*. 1. ed. [S.l.]: Springer, 2018. Citado na página 18.

SPYDER. *Spyder - The Scientific Python Development Environment*. 2020. Disponível em: <<https://www.spyder-ide.org/>>. Acesso em: 18 out 2020. Citado na página 19.

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 20.

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 18.

TANNER, G. *Keras Tutorial 10 - Book Recommendation System*. 2018. Disponível em: <<https://www.youtube.com/watch?v=4vwNkHFuZBk>>. Acesso em: 18 out 2020. Citado na página 32.

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 15.

TENSORFLOW. *Introduction to TensorFlow*. 2020. Disponível em: <<https://www.tensorflow.org/learn>>. Acesso em: 18 out 2020. Citado na página 21.

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 21.

THEANO DEVELOPMENT TEAM. *Theano Docs - Welcome*. 2020. Disponível em: <<http://deeplearning.net/software/theano/>>. Acesso em: 18 out 2020. Citado na página 21.

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 16.

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 26.

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 15.

WATERMAN, D. *A guide to Expert Systems*. Boston: Addison-Wesley, 1985. Citado na página 16.

WEISSTEIN, E. W. *Pearson's Skewness Coefficients*. 2020. Disponível em: <<https://mathworld.wolfram.com/PearsonsSkewnessCoefficients.html>>. Acesso em: 18 out 2020. Citado na página 25.

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 26.

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 26.

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 26.

## Anexos

# ANEXO A – Código - Rede neural Densa do Keras

---

```
1 import pandas as pd
2 from keras.models import Model
3 from keras.layers import Input, Embedding, Flatten, Dense, Concatenate
4 from sklearn.model_selection import train_test_split
5
6 # carregar arquivo, dividir dataset em treinamento/teste
7 dataset = pd.read_csv('arquivo.csv')
8 train, test = train_test_split(dataset, test_size=0.2, random_state=42)
9
10 # criar embedding para produtos
11 prod_input = Input(shape=[1], name="Prod-Input")
12 prod_embedding = Embedding(n_prods+1, 5, name="Prod-Embedding")(prod_input)
13 prod_vec = Flatten(name="Flatten-Prods")(prod_embedding)
14
15 # criar embedding para segmentos de cliente
16 client_input = Input(shape=[1], name="Client-Input")
17 client_embedding = Embedding(n_users+1, 5,
18                             name="Client-Embedding")(client_input)
19 client_vec = Flatten(name="Flatten-Clients")(client_embedding)
20
21 # concatenar conjuntos
22 conc = Concatenate()([prod_vec, client_vec])
23
24 # adicionar camadas
25 fc1 = Dense(128, activation='relu')(conc)
26 fc2 = Dense(32, activation='relu')(fc1)
27 out = Dense(1)(fc2)
28
29 # treinamento
30 model2 = Model([client_input, prod_input], out)
31 model2.compile('adam', 'mean_squared_error')
```

---