UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Descomplicando MLOps

Uma abordagem aplicada com a arquitetura StructML

Állan Christoffer Pereira Silva

UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

O átomo, elemento fundamental da matéria, serve de inspiração para o "átomo tecnológico" apresentado na capa. Este conceito simboliza a essência da arquitetura *StructML*, desenhada para ser o pilar de sistemas que integram operações de *Machine Learning*. A arquitetura *StructML* visa fornecer uma estrutura sólida para a implementação eficiente de tais sistemas, refletindo a importância e a centralidade do átomo no universo da tecnologia aplicada ao *Machine Learning*.

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

ÁLLAN CHRISTOFFER PEREIRA SILVA

DESCOMPLICANDO MLOPS

Uma abordagem aplicada com a arquitetura StructML

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): **ALLAN CHRISTOFFER PEREIRA SILVA**

Título do trabalho:

DESCOMPLICANDO MLOPS

Uma abordagem aplicada com a arquitetura StructML

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [**X**] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Allan Christoffer Pereira Silva**, **Discente**, em 16/02/2024, às 10:02, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 10/09/2024, às 10:48, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4383335** e o código CRC **8835991F**.

Referência: Processo nº 23070.008367/2024-29 SEI nº 4383335

ÁLLAN CHRISTOFFER PEREIRA SILVA

DESCOMPLICANDO MLOPS

Uma abordagem aplicada com a arquitetura StructML

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

SILVA, ALLAN CHRISTOFFER PEREIRA
DESCOMPLICANDO MLOPS [manuscrito] : Uma abordagem aplicada com a arquitetura StructML / ALLAN CHRISTOFFER PEREIRA SILVA. - 2024.
143 f.

Orientador: Prof. Dr. FERNANDO MARQUES FEDERSON; co orientador Dr. SÁVIO SALVARINO TELES DE OLIVEIRA.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2024.

1. inteligência artificial. 2. machine learning operations. 3. StructML. I. FEDERSON, FERNANDO MARQUES, orient. II. Título.

CDU 004

ÁLLAN CHRISTOFFER PEREIRA SILVA

DESCOMPLICANDO MLOPS

Uma abordagem aplicada com a arquitetura StructML

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 08 de fevereiro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Documento assinado digitalmente
gov.br
SAVIO SALVARINO TELES DE OLIVEIRA
Data: 09/02/2024 18:45:59-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Sávio Salvarino Teles de Oliveira
Coorientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Vinícius Sebba Patto
Coordenador do BIA (INF-UFG)

Documento assinado digitalmente
gov.br
LEONARDO AFONSO AMORIM
Data: 09/02/2024 09:41:30-0300
Verifique em https://validar.iti.gov.br

Dr. Leonardo Afonso Amorim
(CEIA-UFG)

ÁLLAN CHRISTOFFER PEREIRA SILVA

DESCOMPLICANDO MLOPS

Uma abordagem aplicada com a arquitetura StructML

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **MLOps**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, machine learning operations, StructML.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **MLOps**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, machine learning operations, StructML.

Goiânia

2024

Minha Jornada

Állan Christoffer Pereira Silva

Especialista em: MLOps

Template baseado em Slidesgo e Freepik

MINHA JORNADA

Nome: Állan Christoffer Pereira Silva

Especialidade: MLOps

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

A minha jornada começou na **Semana 1** com a realização de atividades relacionadas à definição da área de conhecimento para a minha especialização. Iniciei as atividades fazendo parte de um grupo com outros dois pesquisadores e, após algumas reuniões de alinhamento, foi decidido que a área de especialização comum entre nós seria ***Machine Learning Operations (MLOps)***. Após essa definição, foi realizado um processo de busca por bases científicas que pudessem conter publicações para embasar o trabalho de cada pesquisador do grupo.

O ponto de partida foi o acervo da conferência *Computational Science and Computational Intelligence 2023 (CSCI)*, nas quais foram identificadas duas principais linhas de pesquisa: uma sobre Inteligência Artificial e outra sobre *Big Data* e Ciência de Dados. Assim, no **Apêndice 1** é referenciado um documento que contém a filtragem feita dentro dessas duas linhas temáticas para obter alguns termos-chave para a realização dos trabalhos dentro da área de *MLOps*. Além disso, também é referenciada uma tabela que lista

[1] Dez semanas, entre setembro de 2023 e janeiro de 2024.

as publicações da *CSCI* entre 2018 e 2023 relacionadas aos termos-chave definidos anteriormente para servir de fonte de pesquisa para posterior revisão bibliográfica.

As **Semanas 2 e 3** foram dedicadas ao trabalho de revisão bibliográfica e à leitura de artigos científicos tanto da base da *CSCI* quanto de outras bases renomadas na área da Computação. O **Apêndice 2** mostra algumas imagens sobre como o processo de revisão bibliográfica foi estruturado com o auxílio da ferramenta **Parsifal** destacando os objetivos da revisão, as questões de pesquisa, os critérios de inclusão e exclusão, as palavras-chave, a *string* de busca e as bases de referência (ACM, IEEE e Scopus). Além disso, nesse mesmo documento é referenciada uma tabela com os artigos que foram lidos pelos pesquisadores do grupo de *MLOps* acompanhados de algumas observações pertinentes. Por fim, é mostrado o documento referente à **Semana 3** que abrange a tabela com os artigos mas também mostra dois itens adicionais: o cronograma inicial que foi desenvolvido para guiar os trabalhos de cada pesquisador e a estrutura inicial para o vocabulário com os principais conceitos encontrados durante a leitura dos artigos.

Na **Semana 4**, foram desenvolvidas as últimas atividades em grupo: a consolidação do vocabulário de termos relacionados ao *MLOps*, incluindo também conceitos do *DevOps*, além da construção de uma lista de repositórios de códigos com tutoriais e *frameworks* para auxiliar o desenvolvimento da parte técnica dos trabalhos de cada pesquisador. Dessa forma, como primeira atividade individual, defini o objetivo central da minha especialização: propor uma nova arquitetura de dados inspirada na arquitetura Delta da empresa Databricks para servir como ponto de partida para profissionais de dados com pouca ou nenhuma experiência na tarefa de implantação de modelos de *Machine Learning*. Os resultados obtidos nessa etapa estão detalhados no **Apêndice 3.**

Posteriormente, nas **Semana 5 e 6** foram realizadas as atividades de escolha de uma aplicação para servir como estudo de caso e desenvolvimento de uma primeira versão da arquitetura de dados a ser proposta. Com isso, o estudo de caso escolhido inicialmente foi a construção de um sistema de recomendação de produtos, a partir de uma base de dados de *e-commerces* utilizando a técnica de filtragem colaborativa. O **Apêndice 4** mostra como resultados o dicionário de dados com o significado de cada coluna da base de dados escolhida, além do *Jupyter Notebook* em que foi realizada toda a análise exploratória dessa

base. Ademais, o **Apêndice 4** também traz uma explicação sobre a primeira versão da arquitetura acompanhada do arquivo com os códigos de pré-processamento dos dados.

Durante as **Semanas 7 e 8**, foram realizadas as atividades de reformulação da arquitetura de dados implementada até o momento, a primeira modelagem do sistema de recomendação utilizando filtragem colaborativa (interações dos usuários com os produtos), além de uma segunda modelagem utilizando uma abordagem híbrida, isto é, a utilização das interações juntamente com outros atributos de usuários e produtos. Os resultados relativos aos experimentos realizados, a documentação da segunda versão da arquitetura e uma breve explicação sobre as modelagens feitas estão presentes no **Apêndice 5**.

Na etapa final da Disciplina Residência em IA, acabei me deparando com resultados pouco satisfatórios com relação ao estudo de caso escolhido e percebi pouco avanço no grau de maturidade da arquitetura de dados desenvolvida até então (resultados detalhados no **Apêndice 6**). Além disso, durante a etapa de implantação dos modelos baseados em redes neurais que construí, acabei enfrentando uma série de erros de incompatibilidade de bibliotecas os quais não poderiam ser solucionados em tempo hábil e poderiam prejudicar os avanços na minha especialização. Assim, tendo em vista que a área escolhida não era a de sistemas de recomendação mas sim *MLOps*, decidi mudar o estudo de caso e escolher algum outro caso que me permitisse explorar e aprender mais sobre conceitos e ferramentas dentro da área escolhida.

Levando em conta essa decisão, após a entrega referente à **Semana 9**, pude ter um tempo maior para escolher um novo estudo de caso relacionado à construção de modelos de classificação para análise de risco de crédito, a partir de dados simulados de clientes de instituições financeiras. Com isso, na **Semana 10**, além de construir todo o fluxo de preparação e modelagem dos dados, tive êxito na etapa de implantação dos modelos preditivos, o que me permitiu explorar mais ferramentas comumente usadas em operações de *Machine Learning*. A terceira e última versão projetada da arquitetura de *MLOps*, denominada *StructML*, pode ser visualizada em sua forma conceitual e aplicada ao estudo de caso nas documentações presentes no **Apêndice 6**.

Após concluir a Disciplina Residência em IA, posso afirmar que vivenciei uma jornada extremamente gratificante, repleta de aprendizados, que me permitiu explorar uma área pela qual tenho imensa satisfação em trabalhar. Este período me deu a liberdade de aprender

ferramentas novas e absorver conceitos importantes para minha carreira futura. Almejo que este trabalho sirva como um guia útil para estudantes e pesquisadores que desejam entrar no campo de *MLOps*, oferecendo um ponto de partida claro com base nos resultados que obtive.

Quanto aos próximos passos, considero viável a implementação da arquitetura em outras plataformas de nuvem, como AWS e Microsoft Azure, e sua aplicação em diferentes casos de uso de Machine Learning, além da elaboração de um modelo para precificação de custos associados à implantação da arquitetura proposta.

Encerro expressando minha profunda gratidão aos meus orientadores, **Prof. Dr. Fernando Federson** e **Prof. Dr. Sávio Teles**, pelo suporte e pelos valiosos conselhos que foram fundamentais para meu desenvolvimento e que certamente levarei comigo ao longo de minha trajetória.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 19 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva
Gabriel da Mata Marques
Heinz Felipe Cavalcante Rahmig

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Esta entrega consistiu na classificação do grupo de trabalho responsável pela temática de MLOps.

Os requisitos básicos para a entrega eram:

- Classificar os estudos segundo a metodologia da Conference on Computational Science and Computational Intelligence (CSCI).
- Pesquisar nos anais do congresso em busca de trabalhos correlatos com o tema de MLOps.

Os produtos gerados para esta entrega encontram-se nos links abaixo:

- Classificação com os termos da CSCI: https://docs.google.com/document/d/1VPub74MceGjgfJuvjHynzP3-VOxUyI8jz2pSMHIeBnY/edit?usp=sharing
- Pesquisa pelos trabalhos correlatos publicados na CSCI entre 2018 e 2022: https://docs.google.com/spreadsheets/d/1MqcCFht8JVCCPX50I3XKQAxUTjjUlg7v4px4G1jbU3M/edit?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega do dia 26/10/2023, estão planejadas as seguintes atividades:

- Busca por artigos e publicações em outras bases científicas.
- Construção de um repositório com os trabalhos encontrados.
- Revisão e resumos dos trabalhos considerados mais relevantes.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Repositório do grupo de trabalho: https://github.com/AllanSilva156/mlops-residencia-ia

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Entrega Gate 19/10/2023

Introdução

Este documento faz parte da entrega referente ao gate do dia 19 de outubro de 2023. Nele está detalhada a classificação do grupo de pesquisa responsável pela temática de MLOps.

Responsáveis pela Entrega:

Állan Christoffer Pereira Silva
Gabriel da Mata Marques
Heinz Felipe Cavalcante Rahmig

Classificação segundo Conference on Computational Science and Computational Intelligence (CSCI):

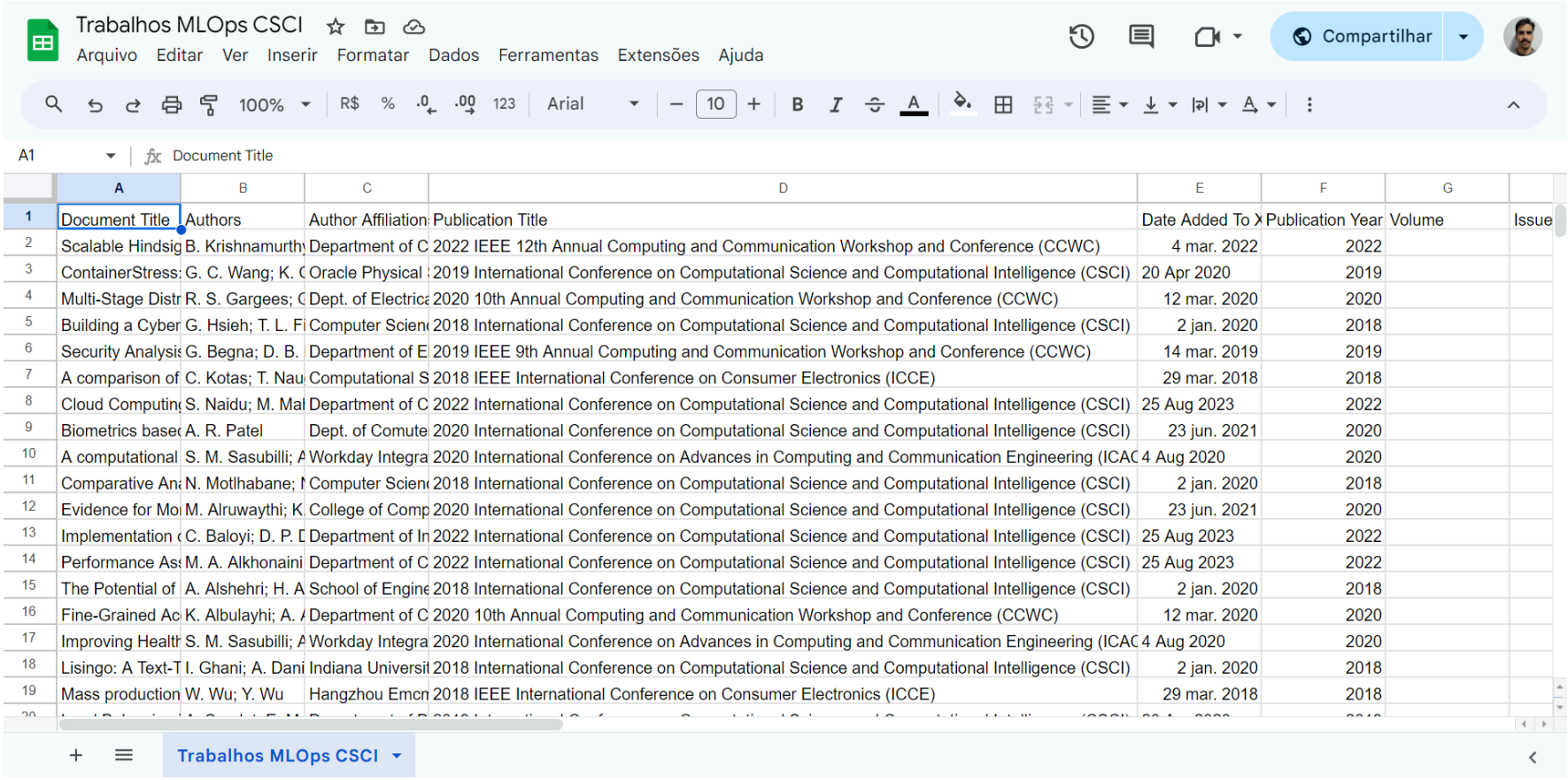
Research Track on Big Data and Data Science (CSCI-RTBD)

SECURITY & PRIVACY IN THE ERA OF DATA SCIENCE & BIG DATA:
- Privacy Preserving Big Data Collection

INFRASTRUCTURES FOR BIG DATA & DATA SCIENCE:
- Cloud Based Infrastructures (applications, storage & computing resources)
- HPC, including Parallel & Distributed Processing
- Programming Models and Environments to Support Big Data
- Software and Tools for Big Data
- Big Data Open Platforms
- Emerging Architectural Frameworks for Big Data
- Paradigms and Models for Big Data beyond Hadoop/MapReduce

BIG DATA & DATA SCIENCE MANAGEMENT AND FRAMEWORKS:
- Database and Web Applications
- Massively Parallel Processing (MPP) Databases
- Distributed Database Systems
- Distributed File Systems
- Distributed Storage Systems
- Data Preservation and Provenance
- Data Protection Methods
- Data Integrity and Privacy Standards and Policies

Revisão na *CSCI*



Trabalhos MLOps CSCI

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda

Compartilhar

A1 | Document Title

	A	B	C	D	E	F	G	
1	Document Title	Authors	Author Affiliation	Publication Title	Date Added To X	Publication Year	Volume	Issue
2	Scalable Hindsig	B. Krishnamurthy	Department of C	2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)	4 mar. 2022	2022		
3	ContainerStress:	G. C. Wang; K. C	Oracle Physical	2019 International Conference on Computational Science and Computational Intelligence (CSCI)	20 Apr 2020	2019		
4	Multi-Stage Distr	R. S. Gargees; C	Dept. of Electrica	2020 10th Annual Computing and Communication Workshop and Conference (CCWC)	12 mar. 2020	2020		
5	Building a Cyber	G. Hsieh; T. L. Fi	Computer Scien	2018 International Conference on Computational Science and Computational Intelligence (CSCI)	2 jan. 2020	2018		
6	Security Analysis	G. Begna; D. B.	Department of E	2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)	14 mar. 2019	2019		
7	A comparison of	C. Kotas; T. Nau	Computational S	2018 IEEE International Conference on Consumer Electronics (ICCE)	29 mar. 2018	2018		
8	Cloud Computing	S. Naidu; M. Mal	Department of C	2022 International Conference on Computational Science and Computational Intelligence (CSCI)	25 Aug 2023	2022		
9	Biometrics base	A. R. Patel	Dept. of Comute	2020 International Conference on Computational Science and Computational Intelligence (CSCI)	23 jun. 2021	2020		
10	A computational	S. M. Sasubilli; A	Workday Integra	2020 International Conference on Advances in Computing and Communication Engineering (ICAC	4 Aug 2020	2020		
11	Comparative Ana	N. Motlhabane; M	Computer Scien	2018 International Conference on Computational Science and Computational Intelligence (CSCI)	2 jan. 2020	2018		
12	Evidence for Mo	M. Alruwaythi; K	College of Comp	2020 International Conference on Computational Science and Computational Intelligence (CSCI)	23 jun. 2021	2020		
13	Implementation	C. Baloyi; D. P. D	Department of In	2022 International Conference on Computational Science and Computational Intelligence (CSCI)	25 Aug 2023	2022		
14	Performance Ass	M. A. Alkhonaini	Department of C	2022 International Conference on Computational Science and Computational Intelligence (CSCI)	25 Aug 2023	2022		
15	The Potential of	A. Alshehri; H. A	School of Engine	2018 International Conference on Computational Science and Computational Intelligence (CSCI)	2 jan. 2020	2018		
16	Fine-Grained Ac	K. Albulayhi; A. A	Department of C	2020 10th Annual Computing and Communication Workshop and Conference (CCWC)	12 mar. 2020	2020		
17	Improving Health	S. M. Sasubilli; A	Workday Integra	2020 International Conference on Advances in Computing and Communication Engineering (ICAC	4 Aug 2020	2020		
18	Lisingo: A Text-T	I. Ghani; A. Dani	Indiana Universit	2018 International Conference on Computational Science and Computational Intelligence (CSCI)	2 jan. 2020	2018		
19	Mass production	W. Wu; Y. Wu	Hangzhou Emcn	2018 IEEE International Conference on Consumer Electronics (ICCE)	29 mar. 2018	2018		

Trabalhos MLOps CSCI

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 26 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva
Gabriel da Mata Marques
Heinz Felipe Cavalcante Rahmig

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

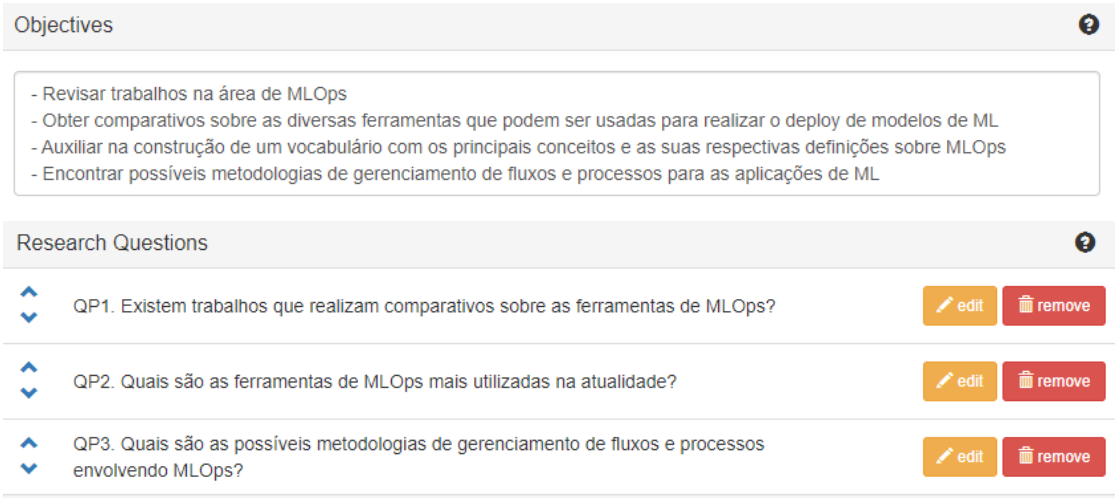
Esta entrega consistiu no início do processo de revisão bibliográfica das referências que irão fundamentar os trabalhos do grupo responsável pela temática de MLOps.

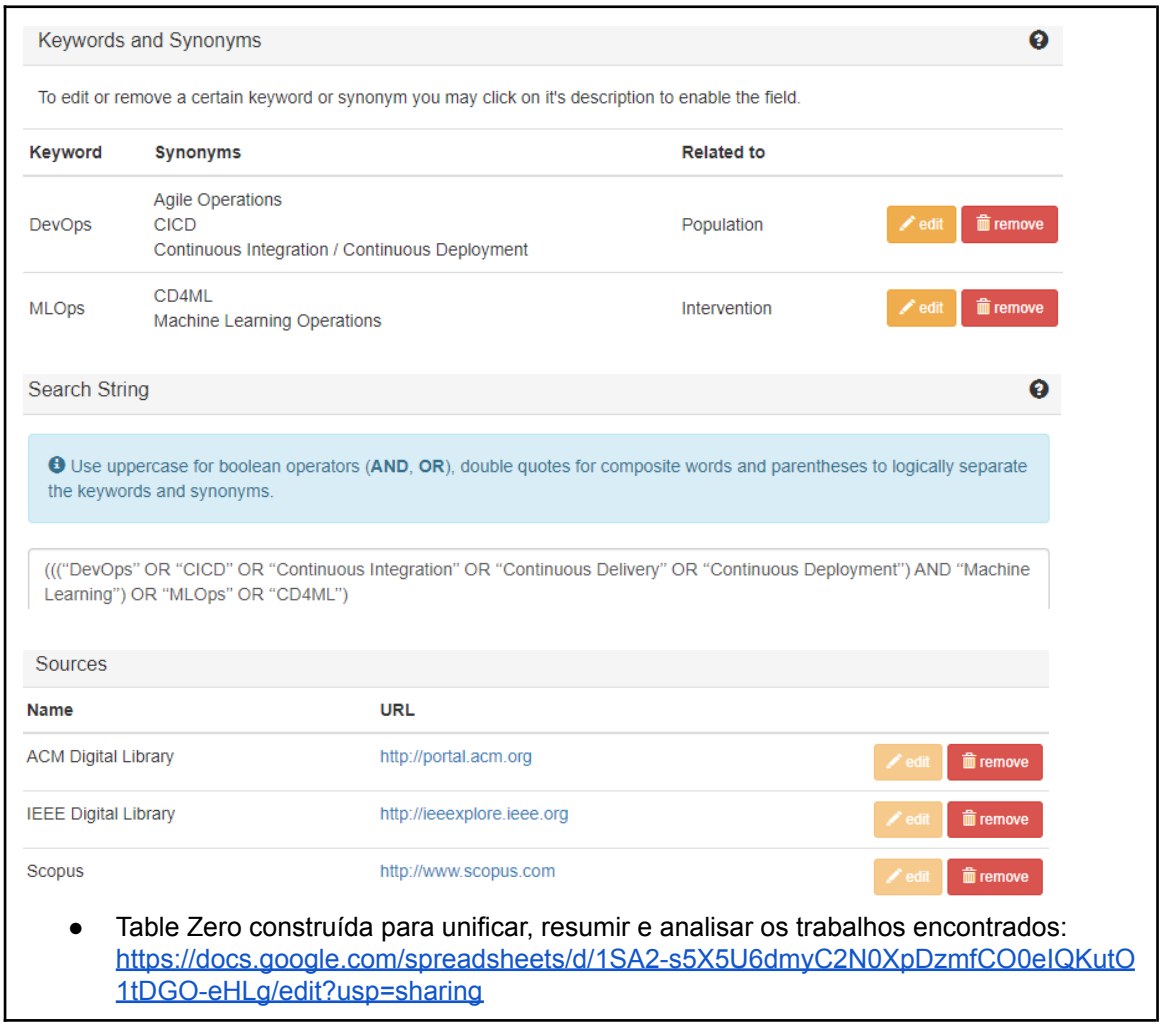
Os requisitos básicos para a entrega eram:

- Buscar artigos e publicações em outras bases científicas além da CSCI.
- Construir um repositório com os trabalhos encontrados.
- Realizar uma revisão preliminar dos trabalhos e um breve resumo sobre cada um.

Os produtos gerados para esta entrega estão descritos a seguir:

- Estruturação do processo de revisão bibliográfica utilizando a ferramenta Parsifal.





Keywords and Synonyms

To edit or remove a certain keyword or synonym you may click on it's description to enable the field.

Keyword	Synonyms	Related to	
DevOps	Agile Operations CICD Continuous Integration / Continuous Deployment	Population	edit remove
MLOps	CD4ML Machine Learning Operations	Intervention	edit remove

Search String

Use uppercase for boolean operators (**AND**, **OR**), double quotes for composite words and parentheses to logically separate the keywords and synonyms.

(((“DevOps” OR “CICD” OR “Continuous Integration” OR “Continuous Delivery” OR “Continuous Deployment”) AND “Machine Learning”) OR “MLOps” OR “CD4ML”)

Sources

Name	URL	
ACM Digital Library	http://portal.acm.org	edit remove
IEEE Digital Library	http://ieeexplore.ieee.org	edit remove
Scopus	http://www.scopus.com	edit remove

- Table Zero construída para unificar, resumir e analisar os trabalhos encontrados: https://docs.google.com/spreadsheets/d/1SA2-s5X5U6dmyC2N0XpDzmfCO0eIQKutO1tDGO-eHLg/edit?usp=sharing

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega do dia 09/11/2023, estão planejadas as seguintes atividades:

- Finalização do processo de revisão bibliográfica.
- Construção do cronograma de atividades a serem desenvolvidas até o final da Residência.
- Início da montagem de um vocabulário com termos e definições relacionadas à temática de MLOps.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Table Zero

Table Zero MLOps

Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda

Compartilhar

ID	Nome	Ano	Base	Tópico	PDF	Resumo	Responsável	Incluído
1	Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?	2021	IEEE	MLOps	Link	Pesquisa feita em 63 países com 331 profissionais da área sobre a importância do MLOps para cientistas de dados. Este trabalho pode ser usado como referência para comprovar a relevância do tema no meio acadêmico.	Állan	Sim
2	Architecting MLOps in the Cloud: From Theory to Practice	2023	IEEE	MLOps	Link	Referência a um tutorial sobre como escolher entre as opções de cloud disponíveis e um caso prático de implementação. Porém, não foi encontrado o repositório referente ao tutorial.	Állan	-
3	Practitioners guide to MLOps: A framework for continuous delivery and automation of machine learning.	2021	Google	MLOps	Link	Guia prático elaborado pela equipe do Google Cloud com definições e ferramentas relacionadas à operacionalização de modelos de ML.	Állan	-
4	MLOps: Five Steps to Guide its Effective Implementation	2022	IEEE	MLOps	Link	Artigo com 5 dicas sobre como construir um ciclo de vida de soluções de ML de forma efetiva. As dicas são um pouco genéricas mas podem ser usadas como referência.	Állan	-
5	Towards MLOps: A Framework and Maturity Model	2021	IEEE	MLOps	Link	Este trabalho conta com uma revisão da literatura sobre o estado da arte em MLOps, propõe um framework validado de desenvolvimento contínuo em ML e finaliza com um modelo de maturidade para empresas com relação ao MLOps.	Állan	Sim
6	Sustainable MLOps: Trends and Challenges	2020	IEEE	MLOps	Link	Artigo fala um pouco sobre a relevância do MLOps no contexto científico e prático, traz algumas referências relacionadas a Sustentabilidade e Interpretabilidade de soluções de ML, além de relatar alguns desafios da área.	Állan	-
7	On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps	2021	IEEE	MLOps	Link	Artigo que demonstra alguns níveis diferentes de maturidade em MLOps, traz as principais diferenças entre MLOps e DevOps e relata sobre uma metodologia de desenvolvimento baseada em push e pull chamada GitOps.	Állan	Sim
8	Towards Automation for MLOps: An Exploratory Study of Bot Usage in Deep Learning Libraries	2022	IEEE	MLOps	Link	Este trabalho relata 9 tarefas em projetos de ML que podem ser automatizadas através da criação de bots e utilização em bibliotecas de Deep Learning. Apesar de parecer promissor, acredito ser bem avançado para projetos iniciais em MLOps.	Állan	-
9	An Efficient Microservices Architecture for MLOps	2023	IEEE	MLOps	Link	Artigo que trata sobre a arquitetura de microsserviços, o padrão SAGA de arquitetura, além de propor uma nova arquitetura voltada para MLOps. Os conceitos por trás da arquitetura de microsserviços podem ser bastante úteis e merecem estudos mais aprofundados.	Állan	Sim
10	MLOps for evolvable AI intensive software systems	2022	IEEE	MLOps	Link	Este trabalho traz apenas algumas relações entre DevOps e MLOps, além de uma representação visual dos processos envolvidos no MLOps. Estudo superficial e com baixo potencial de contribuição.	Állan	-
11	K2E: Building MLOps Environments for Governing Data and Models Catalogues while Tracking Versions	2022	IEEE	MLOps	Link	O artigo trata sobre alguns desafios relacionados aos dados e modelos, baseado nesses desafios é proposta uma organização chamada Knowledge To Environment (KFE), a qual aparenta ter bom potencial para estrturar sistemas com grande volume de dados e muitas versões de modelos.	Állan	Sim

Bases	CSCI, IEEE, ACM, Scopus
Período	2018 a 2023
Strings de busca	(((“DevOps” OR “CICD” OR “Continuous Integration” OR “Continuous Delivery” OR “Continuous Deployment”) AND “Machine Learning”) OR “MLOps” OR “CD4ML”)

Página1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva
Gabriel da Mata Marques
Heinz Felipe Cavalcante Rahmig

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Esta entrega consistiu na finalização do processo de revisão bibliográfica, o qual tinha por objetivo encontrar trabalhos de referência na área de MLOps.

Os requisitos básicos para a entrega eram:

- Finalizar o processo de revisão bibliográfica.
- Construir o cronograma de atividades a serem desenvolvidas até o final da Residência.
- Iniciar a montagem de um vocabulário com termos e definições relacionadas à temática de MLOps.

Os produtos gerados para esta entrega estão descritos a seguir:

- Repositório com os artigos encontrados e suas respectivas análises (Table Zero): https://docs.google.com/spreadsheets/d/1SA2-s5X5U6dmyC2N0XpDzmfCO0eIQKutO1tDGO-eHLg/edit?usp=sharing
- Cronograma de atividades da Residência: https://docs.google.com/spreadsheets/d/16sy4Z3gcDNV2U5fZOC_mPgE7eyiGfgc5zHQFBCrxBSc/edit?usp=sharing
- Vocabulário sobre MLOps: https://docs.google.com/document/d/1AtmA9GgHnD4mlF-bbjjbt53QLZaQAjowFyjtjQHzzLA/edit?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega do dia 16/11/2023, estão planejadas as seguintes atividades:

- Finalização do vocabulário incluindo conceitos de DevOps.
- Busca por repositórios de códigos úteis.
- Decisão sobre a aplicação e levantamento de requisitos.

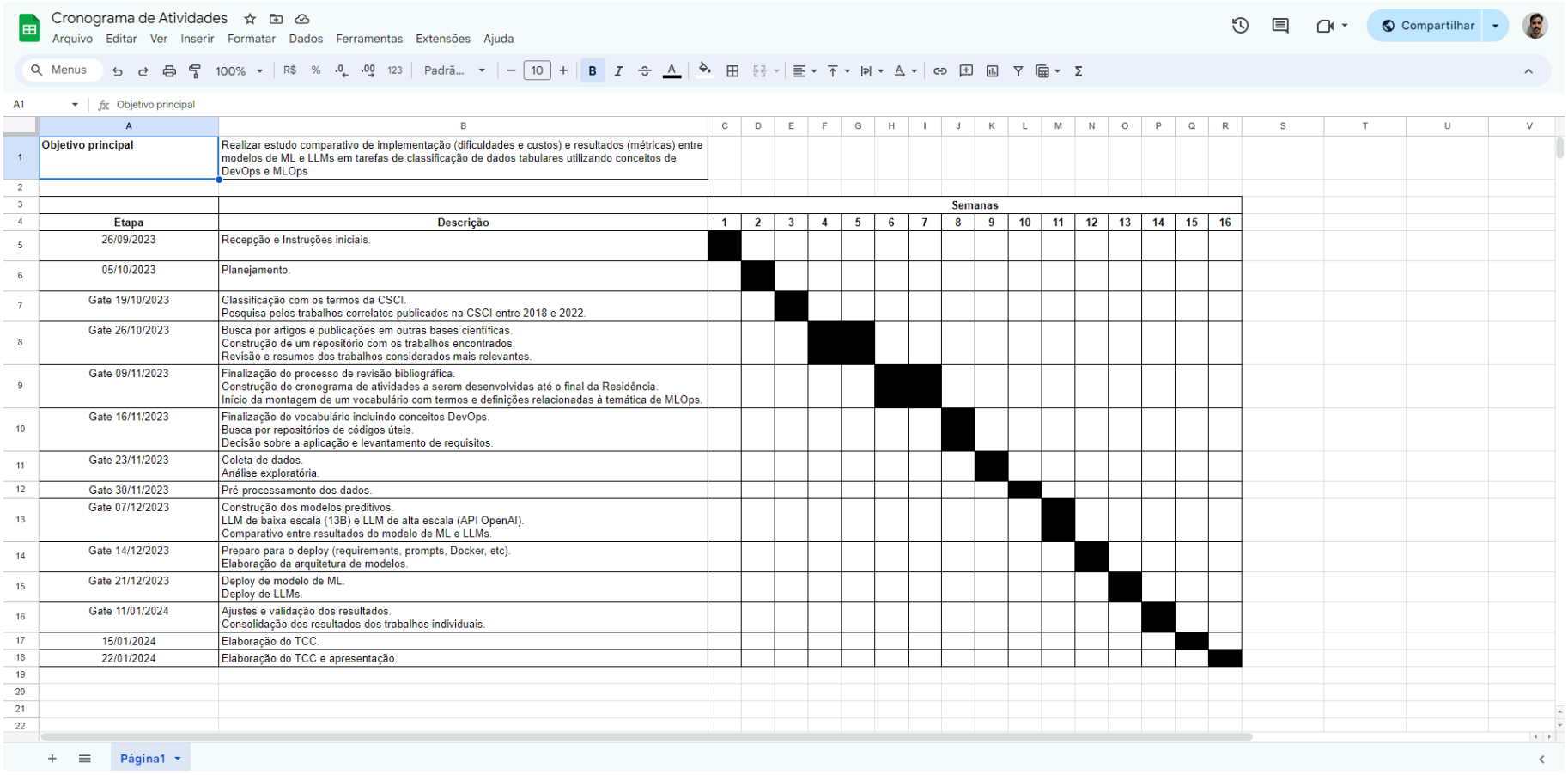
Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Cronograma de Atividades



Cronograma de Atividades

	Etapa	Descrição	Semanas 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	**Objetivo principal**	Realizar estudo comparativo de implementação (dificuldades e custos) e resultados (métricas) entre modelos de ML e LLMs em tarefas de classificação de dados tabulares utilizando conceitos de DevOps e MLOps																
5	26/09/2023	Recepção e Instruções iniciais.	■															
6	05/10/2023	Planejamento.		■														
7	Gate 19/10/2023	Classificação com os termos da CSCI. Pesquisa pelos trabalhos correlatos publicados na CSCI entre 2018 e 2022.			■													
8	Gate 26/10/2023	Busca por artigos e publicações em outras bases científicas. Construção de um repositório com os trabalhos encontrados. Revisão e resumos dos trabalhos considerados mais relevantes.				■	■											
9	Gate 09/11/2023	Finalização do processo de revisão bibliográfica. Construção do cronograma de atividades a serem desenvolvidas até o final da Residência. Início da montagem de um vocabulário com termos e definições relacionadas à temática de MLOps.						■	■									
10	Gate 16/11/2023	Finalização do vocabulário incluindo conceitos DevOps. Busca por repositórios de códigos úteis. Decisão sobre a aplicação e levantamento de requisitos.								■								
11	Gate 23/11/2023	Coleta de dados. Análise exploratória.									■							
12	Gate 30/11/2023	Pré-processamento dos dados.										■						
13	Gate 07/12/2023	Construção dos modelos preditivos. LLM de baixa escala (13B) e LLM de alta escala (API OpenAI). Comparativo entre resultados do modelo de ML e LLMs.											■					
14	Gate 14/12/2023	Preparo para o deploy (requirements, prompts, Docker, etc). Elaboração da arquitetura de modelos.												■				
15	Gate 21/12/2023	Deploy de modelo de ML. Deploy de LLMs.													■			
16	Gate 11/01/2024	Ajustes e validação dos resultados. Consolidação dos resultados dos trabalhos individuais.														■		
17	15/01/2024	Elaboração do TCC.															■	
18	22/01/2024	Elaboração do TCC e apresentação.																■

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Vocabulário sobre MLOps

Introdução

Este documento tem como objetivo principal a documentação dos principais termos e definições relacionados à área de Machine Learning Operations (MLOps). O conteúdo contido neste trabalho serve como base teórica para estudantes e profissionais que desejam se aprofundar sobre como operacionalizar modelos de Machine Learning, isto é, realizar a implantação de modelos preditivos.

Development Operations (DevOps)

DevOps é uma combinação de filosofias, práticas e ferramentas que aumenta a capacidade de uma organização de entregar aplicações e serviços em alta velocidade. Melhorando e evoluindo produtos mais rapidamente do que organizações que utilizam processos tradicionais de desenvolvimento e gerenciamento de infraestrutura. DevOps é caracterizado pela automação e monitoramento em todas as fases do desenvolvimento de software, desde a integração, testes, liberação até a implantação e gestão de infraestrutura.

Práticas Comuns de DevOps:

1. Integração Contínua (CI): Prática que incentiva desenvolvedores a integrar código em um repositório compartilhado. Cada check-in é então verificado por uma build automatizada, permitindo que equipes detectem problemas cedo.
2. Entrega Contínua (CD): Extensão da integração contínua para garantir que o código seja seguro e que possa ser liberado a qualquer momento.
3. Monitoramento e Logging: Processos que envolvem a coleta de métricas e logs para acompanhar o desempenho das aplicações e da infraestrutura.
4. Comunicação e Colaboração: Ferramentas e práticas culturais que promovem a colaboração dentro e entre as equipes.
5. Automação de Infraestrutura: Gerenciamento e provisionamento de infraestrutura através de código e ferramentas, minimizando a intervenção manual.

Ferramentas de DevOps:

1. Docker: Ferramenta de contêinerização que permite empacotar uma aplicação com todas as suas dependências em um contêiner padronizado.
2. Jenkins: Servidor de automação open source usado para CI/CD.
3. Kubernetes: Sistema de orquestração de contêineres que gerencia aplicações construídas em contêineres.
4. Ansible/Terraform: Ferramentas que permitem aos desenvolvedores provisionar e gerenciar infraestrutura através de código.
5. Git: Sistema de controle de versão distribuído para rastrear mudanças no código fonte durante o desenvolvimento de software.
6. Nagios/Grafana: Ferramentas de monitoramento que oferecem visibilidade em tempo real sobre a saúde da infraestrutura e aplicações.

Termos Chave de DevOps:

Termo	Definição
1- Pipeline de Deploy	Sequência de passos para entregar uma nova versão de software.
2- Infrastructure as Code (IaC)	Prática de gerenciar e provisionar infraestrutura de TI através de scripts de código.
3- Micro Serviços	Arquitetura que estrutura uma aplicação como uma coleção de serviços que são executados de forma independente.
4- Orquestração de Containers	Processo de gerenciar a vida útil de contêineres, especialmente em ambientes com muitos contêineres.
5- Gerenciamento de Configuração	Processo de manter computadores, servidores e software em um estado desejado e consistente.
6 - Automação de Testes	Uso de software para controlar a execução de testes, a comparação de resultados esperados com resultados reais, e a configuração de pré-condições de testes.

7- Versionamento Semântico	Convenção para nomear e gerenciar versões de software de forma a comunicar o impacto das mudanças no código.
8- Balanceamento de Carga	Distribuição automática de tráfego de rede ou pedidos entre vários servidores.
9- Código de Infraestrutura	Código que cria e configura a infraestrutura necessária para uma aplicação.
10- Dashboard de Monitoramento	Interface visual que exibe métricas importantes da aplicação e da infraestrutura.

Machine Learning Operations (MLOps)

Machine Learning Operations (MLOps) é uma área de atuação profissional responsável por dar suporte ao modelos, ao desenvolvimento e à operacionalização do ciclo de vida de Machine Learning estruturado nos princípios e práticas de DevOps.

Um fluxo de trabalho (workflow) de MLOps muito comum é composto por:

1. Extração de Dados (Data Extraction): etapa caracterizada pela integração de dados relevantes de fontes variadas.
2. Análise de Dados (Data Analysis): etapa caracterizada pela compreensão dos dados existentes nos conjuntos de dados.
3. Limpeza de Dados, Transformação e Engenharia de Atributos (Data Cleaning, Transformation and Feature Engineering): etapa caracterizada pela divisão e preparação dos conjuntos de dados de treinamento, validação e teste.
4. Treinamento do Modelo (Model Training): etapa caracterizada pelo treinamento de modelos de Machine Learning e armazenamento dos modelos com melhor desempenho, partindo de diferentes algoritmos e configurações de parâmetros.
5. Validação do Modelo (Model Validation): etapa caracterizada pela avaliação interativa da qualidade do modelo no conjunto de dados de teste e pela constatação de que se o modelo está atendendo aos critérios de qualidade baseada nas métricas de desempenho.
6. Serviço do Modelo (Model Serving): etapa caracterizada pela implantação dos modelos nos ambientes alvo integrados a outros componentes de software.

7. Monitoramento do Modelo (Model Monitoring): etapa caracterizada pela detecção da degradação do modelo através de análises de uso, dados de entrada e desempenho.

Termo	Definição
1- Open source/Código aberto	O código do software é público e disponível para uso, modificação e distribuição.
2- Escalabilidade	A capacidade de aumentar o tamanho da carga de trabalho dentro da infraestrutura existente (hardware, software, etc.) sem impactar o desempenho.
3- Elasticidade	A capacidade de expandir ou reduzir dinamicamente os recursos da infraestrutura (computacional) conforme necessário para se adaptar às mudanças na carga de trabalho de maneira autônoma.
4- Cloud agnostic	O desempenho é consistente, independentemente da plataforma em que é implantado.
5- Extensibilidade	Defina facilmente seus próprios operadores, executores e amplie a biblioteca para que ela se ajuste ao nível de abstração adequado ao seu ambiente.
6- Gestão/Coleta de metadados	A gestão de metadados é usada para coletar dados durante todo o pipeline de ML.
7- Isolamento/Fraco acoplamento	Os componentes podem ser desenvolvidos e implantados independentemente e devem depender uns dos outros na menor medida possível.
8- CI/CD	A plataforma suporta Integração Contínua (CI) e Entrega Contínua (CD) para o pipeline completo de ML.

9- UI	Interface de Usuário ou Dashboard.
10- CLI	Interface de Linha de Comando.
11- API gateway	Em vez de chamar os serviços diretamente, os clientes podem chamar o gateway de API, que encaminha a chamada para os serviços apropriados no back-end e serve como ponto de entrada para os clientes.
12- DAGs	Grafos Acíclicos Dirigidos são usados para descrever o fluxo de trabalho ou podem ser encapsulados dentro da plataforma.
13- Data streaming (real-time)	O fluxo contínuo de dados gerados por várias fontes de dados é suportado e pode ser processado, armazenado, analisado e utilizado diretamente.
14- Data storage	Um banco de dados integrado para armazenar dados brutos, projetos e metadados.
15- Data analysis	Um componente do pipeline gera estatísticas de características tanto para dados de treinamento quanto para dados de serviço, que podem ser usados por outros componentes do pipeline.
16- Data transformation	Um componente do pipeline identifica anomalias nos dados de treinamento e de serviço e prepara os dados para tarefas de ML. O resultado deste passo são as divisões de dados.
17- Data monitoring	Os dados são monitorados para manter a qualidade e inspecionar métricas gerais.
18- API endpoint	A saída da gestão de dados pode ser acessada usando um gateway de API, que encaminha os dados, metadados ou esquema de dados.
19- Automação	O processo de gestão de dados pode ser

	executado automaticamente em produção com base em uma programação ou em resposta a um gatilho.
20- Library agnostic	Todos os principais frameworks e bibliotecas de ML são suportados.
21- Model tracking	O desempenho do modelo de ML intermediário pode ser rastreado e registrado para manter a reprodutibilidade e obter insights.
22- Model registry	Um repositório centralizado usado para padronizar a definição, armazenamento e acesso de características para treinamento e serviço, que é acessível via uma API.
23- Hyper parameter tuning	Um motor de otimização é encapsulado para o ajuste de hiperparâmetros para treinar os modelos de ML de forma eficiente.
24- Teste A/B	Testes A/B podem ser usados para rastrear diferenças entre duas versões de modelos preditivos ou modelos podem ser executados em paralelo em diferentes pontos de extremidade.
25- Detecção de anomalias	Os outliers são automaticamente identificados para revelar padrões irregulares do modelo de ML.
26- Detecção de drift	Mudanças significativas nas distribuições de dados e no desempenho da previsão são automaticamente detectadas para prevenir obsolescência e diminuição da precisão.
27- Alerta de threshold	É possível configurar alertas quando a distribuição de previsões varia significativamente dos valores esperados.
28- Monitoramento de performance	O desempenho preditivo do modelo é monitorado para potencialmente invocar

	uma nova iteração no processo de ML.

Tabela 1. Definições dos principais termos em MLOps

Infrastructure as Code (IaC)

Infrastructure as Code (IaC) é uma prática da área de DevOps que consiste na criação de documentos em linguagem de codificação descritiva de alto nível para automatizar o provisionamento da infraestrutura de TI.

IaC elimina a necessidade de configuração manual de servidores, sistemas operacionais, conexões de bancos de dados, entre outras tarefas.

As ferramentas de IaC mais conhecidas no mercado são: Ansible, Terraform, Pulumi, Azure Resource Manager (ARM) e Google Cloud Deployment Manager.

Terraform é uma ferramenta popular de Infrastructure as Code (IaC) usada para provisionar infraestrutura em várias plataformas, como AWS, Azure, Google Cloud, entre outras. Ele utiliza uma linguagem de configuração chamada HashiCorp Configuration Language (HCL) para descrever a infraestrutura desejada de forma declarativa. Abaixo está um exemplo básico de um documento de configuração do Terraform para provisionar uma instância EC2 na AWS:

```
provider "aws" {
  region = "us-west-2"
}

resource "aws_instance" "example" {
  ami           = "ami-abcdefgh"
  instance_type = "t2.micro"
}

output "ip" {
  value = aws_instance.example.public_ip
}
```

Bloco Provider:

- provider "aws" { ... }: Este bloco indica que você está usando o provedor AWS. O Terraform tem provedores para várias plataformas e serviços.
- region = "us-west-2": Especifica a região da AWS onde os recursos serão provisionados.

Bloco Resource:

- resource "aws_instance" "example" { ... }: Este bloco define um recurso, que neste caso é uma instância EC2 na AWS.
- ami = "ami-abcdefgh": Especifica a Amazon Machine Image (AMI) que será usada para lançar a instância.
- instance_type = "t2.micro": Especifica o tipo de instância que será lançado.

Bloco Output:

- output "ip" { ... }: Este bloco define uma saída que será exibida após o Terraform aplicar a configuração.
- value = aws_instance.example.public_ip: Especifica que o IP público da instância EC2 será exibido como saída.

Quando este código é aplicado usando o comando terraform apply, o Terraform cria uma instância EC2 na AWS com as especificações fornecidas. A saída do comando mostrará o IP público da instância EC2 criada.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 16 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

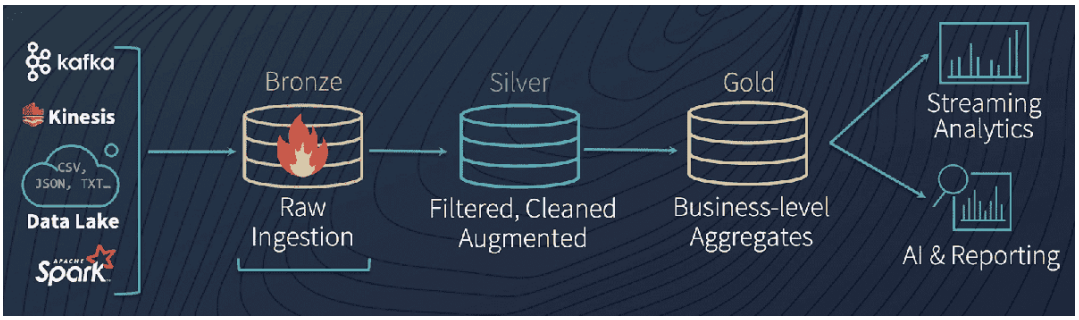
Esta entrega consistiu na finalização da construção do embasamento teórico dos trabalhos que serão desenvolvidos, além da realização do planejamento de atividades a serem executadas até o fim da Residência.

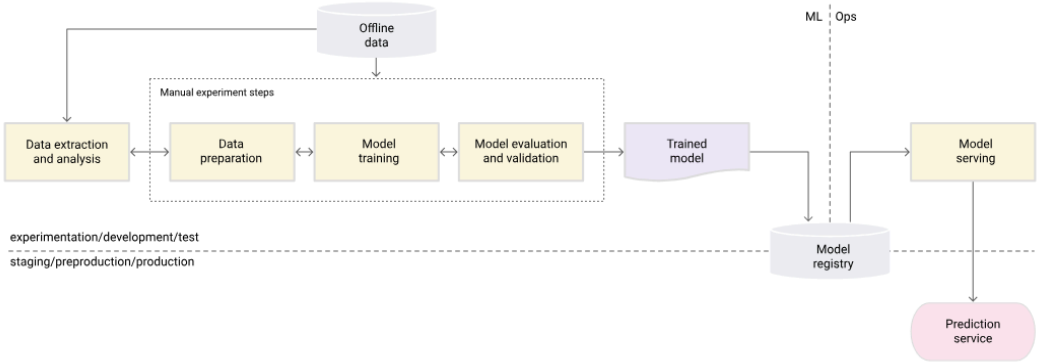
Os requisitos básicos para a entrega eram:

- Finalizar o vocabulário incluindo conceitos de DevOps.
- Buscar por repositórios de códigos úteis.
- Decidir sobre a aplicação de cada integrante e realizar o levantamento de requisitos.

Os produtos gerados para esta entrega estão descritos a seguir:

- Vocabulário completo com os principais termos e definições relacionados às áreas de DevOps e MLOps.
- Lista de repositórios de códigos úteis: https://docs.google.com/document/d/1MW3oMdNZbVLtNh4v4SLsf61smGIFkKybD_qojIRe7qY/edit?usp=sharing
- Além disso, foi decidido de forma individual, qual aplicação cada integrante do grupo irá trabalhar. No meu caso, a aplicação escolhida foi a proposição de uma nova arquitetura para implantação de modelos de Machine Learning baseada na arquitetura Delta e na arquitetura de MLOps proposta pelo time do GCP.
- Os principais requisitos levantados foram: integrar Delta Lake para gerenciamento de dados, utilizar ferramentas do GCP para ML, automatizar com pipeline de CI/CD e monitorar modelos com ferramentas GCP.





Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega do dia 23/11/2023, estão planejadas as seguintes atividades:

- Busca pela base de dados que será utilizada durante o desenvolvimento do trabalho.
- Criação de um dicionário de dados para documentar a base de dados.
- Análise exploratória de dados preliminar.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

A montagem do vocabulário e a pesquisa por repositórios de códigos úteis foram atividades realizadas de forma coletiva. Porém, foi decidido que cada integrante do grupo seguiria frentes de trabalho individuais e, por isso, a escolha da aplicação da pesquisa foi feita individualmente.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Repositórios Úteis

Abaixo estão listados alguns links que fazem referência a repositórios e frameworks importantes e que podem auxiliar no desenvolvimento dos trabalhos relacionados a MLOps e LLMOps.

- GitHub - langchain-ai/langchain: ⚡ Building applications with LLMs through composability ⚡
- GitHub - bregman-arie/devops-exercises: Linux, Jenkins, AWS, SRE, Prometheus, Docker, Python, Ansible, Git, Kubernetes, Terraform, OpenStack, SQL, NoSQL, Azure, GCP, DNS, Elastic, Network, Virtualization. DevOps Interview Questions
- GitHub - binhnguyennus/awesome-scalability: The Patterns of Scalable, Reliable, and Performant Large-Scale Systems
- DevOps Roadmap for 2023. with learning resources
- GitHub - HumanSignal/label-studio: Label Studio is a multi-type data labeling and annotation tool with standardized output format
- GitHub - EthicalML/awesome-production-machine-learning: A curated list of awesome open source libraries to deploy, monitor, version and scale your machine learning
- GitHub - microsoft/MLOps: MLOps examples
- GitHub - visenger/awesome-mlops: A curated list of references for MLOps
- GitHub - GokuMohandas/Made-With-ML: Learn how to design, develop, deploy and iterate on production-grade ML applications.
- MLOps Guide
- GitHub - jina-ai/jina: ☁️ Build multimodal AI applications with cloud-native stack
- GitHub - EthicalML/awesome-production-machine-learning: A curated list of awesome open source libraries to deploy, monitor, version and scale your machine learning

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 23 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Esta entrega consistiu na definição da aplicação a ser trabalhada durante as próximas semanas da Residência, além do início da primeira etapa de desenvolvimento da pesquisa aplicada (documentação e análise exploratória de dados).

Os requisitos básicos para a entrega eram:

- Buscar pela base de dados que será utilizada durante o desenvolvimento do trabalho.
- Criar um dicionário de dados para documentar a base de dados.
- Realizar uma análise exploratória de dados preliminar.

Os produtos gerados para esta entrega estão descritos a seguir:

- A aplicação escolhida para a pesquisa irá envolver a utilização da base de dados Brazilian E-Commerce Public Dataset para construção de um sistema de recomendação e a implantação deste sistema seguindo os princípios de MLOps. O foco da pesquisa será muito mais voltado ao detalhamento dos processos desenvolvidos para implantação e disponibilização do sistema do que à avaliação das métricas do modelo de Machine Learning em si.
- Dicionário de dados: https://docs.google.com/document/d/1rC20HQ14E0O1nC6J0Pb3Ug5pG2mDNr7xD1eG5fNNEAo/edit?usp=sharing
- Análise exploratória de dados preliminar: https://colab.research.google.com/drive/1EG_XTutJXWpMfr9W7coGLvMTaFqw8Xzv?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega do dia 30/11/2023, estão planejadas as seguintes atividades:

- Elaboração da arquitetura de dados (armazenamento bronze, silver e gold).
- Pré-processamento dos dados.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

As etapas de desenvolvimento deste trabalho estão sendo feitas sob orientação do Prof. Dr. Sávio Teles (INF-UFG) o qual é pesquisador e desenvolvedor nas áreas de Big Data e MLOps.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Dicionário de Dados

Autor
Állan Christoffer Pereira Silva

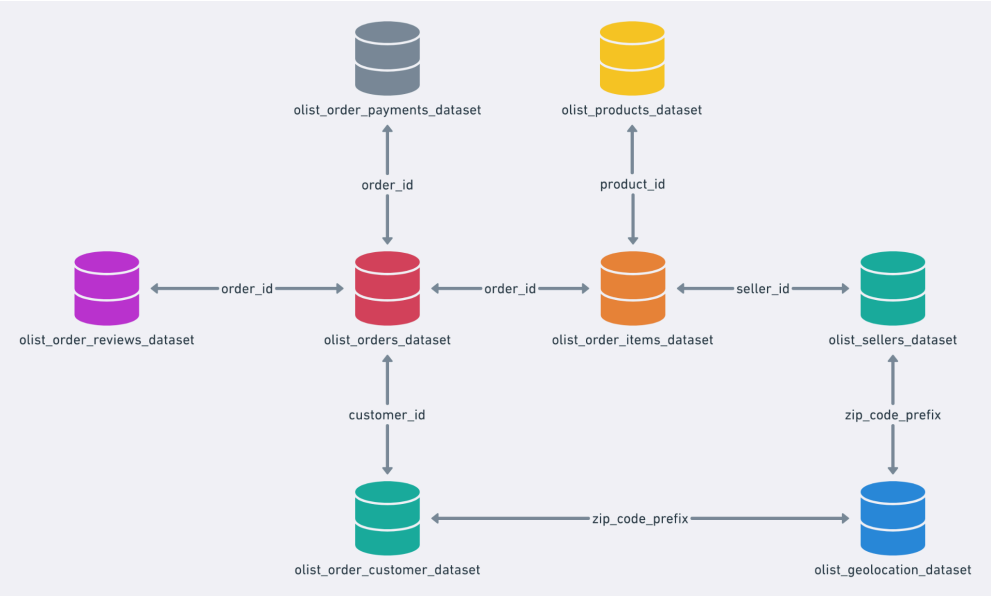
Introdução
Este documento tem como objetivo principal documentar as informações relacionadas à base de dados Brazilian E-Commerce Public Dataset para o desenvolvimento e implantação de um sistema de recomendação de acordo com os princípios do MLOps.

O Brazilian E-Commerce Public Dataset é um conjunto de dados criado pela Olist que contém cerca de 100.000 registros sobre pedidos feitos entre 2016 e 2018 em diversos marketplaces brasileiros.

Obs.: Os dados são reais e foram anonimizados para evitar qualquer tipo de problema de privacidade e segurança.

O conjunto de dados original contém 8 tabelas e as relações entre elas podem ser visualizadas no esquema abaixo.

Figura 1. Esquema dos dados



Fonte: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

Sobre os dados
Um breve resumo das informações contidas em cada tabela pode ser visto abaixo.

1. **olist_customers_dataset**: informações sobre o cliente e sua localização.
2. **olist_geolocation_dataset**: informações dos CEPs brasileiros e suas coordenadas lat/lng.
3. **olist_order_items_dataset**: dados sobre os itens adquiridos em cada pedido.
4. **olist_order_payments_dataset**: dados sobre as opções de pagamento dos pedidos.
5. **olist_order_reviews_dataset**: dados sobre as avaliações feitas pelos clientes.
6. **olist_orders_dataset**: conjunto de dados principal.
7. **olist_products_dataset**: dados sobre os produtos comercializados pela Olist.
8. **olist_sellers_dataset**: dados sobre os vendedores que atenderam aos pedidos feitos na Olist.

As informações sobre as colunas das tabelas listadas estão dispostas na tabela a seguir.

Tabela 1. Informações sobre as colunas das tabelas

Tabela	Coluna	Definição
olist_customers_dataset	customer_id	Chave para o conjunto de dados de pedidos.
	customer_unique_id	Identificador exclusivo de um cliente.
	customer_zip_cod	Primeiros cinco dígitos do CEP do cliente.
	customer_city	Nome da cidade do cliente.
	customer_state	Nome do estado do cliente.
olist_geolocation_dataset	geolocation_zip_code_prefix	Primeiros cinco dígitos do CEP.
	latitude	Coordenada de latitude.
	longitude	Coordenada de longitude.
	geolocalization_city	Nome da cidade.
	geolocalization_state	Nome do estado.
olist_order_items_dataset	order_id	Identificador exclusivo do pedido.

	order_item_id	Número sequencial que identifica a quantidade de itens incluídos no mesmo pedido.
	product_id	Identificador exclusivo do produto.
	seller_id	Identificador exclusivo do vendedor.
	shipping_limit_date	Data limite de envio do vendedor para o envio do pedido ao parceiro logístico.
	price	Preço do item.
	freight_value	Item com valor de frete (se um pedido tiver mais de um item o valor do frete é dividido entre os itens).
olist_order_payments_dataset	order_id	Identificador exclusivo de um pedido.
	payment_sequential	Um cliente pode pagar um pedido com mais de um método de pagamento. Se ele fizer isso, será criada uma sequência.
	payment_type	Forma de pagamento escolhida pelo cliente.
	payment_installments	Número de parcelas escolhido pelo cliente.
	payment_value	Valor da transação.
olist_order_reviews_dataset	review_id	Identificador exclusivo de avaliação.
	order_id	Identificador exclusivo de pedido.
	review_score	Nota que varia de 1 a 5 dada pelo cliente em pesquisa de satisfação.

	review_comment_title	Título do comentário da avaliação deixada pelo cliente.
	review_comment_message	Mensagem de comentário da avaliação deixada pelo cliente.
	review_creation_date	Data/hora em que a pesquisa de satisfação foi enviada ao cliente.
	review_answer_timestamp	Data/hora da resposta da pesquisa de satisfação.
olist_orders_dataset	order_id	Identificador exclusivo do pedido.
	customer_id	Chave para o conjunto de dados do cliente.
	order_status	Referência ao estado da encomenda (entregue, expedida, etc).
	order_purchase_timestamp	Data/hora da compra.
	order_approved_at	Data/hora da aprovação do pagamento.
	order_delivered_carrier_date	Data/hora de postagem do pedido.
	order_delivered_customer_date	Data/hora real da entrega do pedido ao cliente.
	order_estimated_delivery_date	Data/hora estimada de entrega que foi informada ao cliente no momento da compra.
olist_products_dataset	product_id	Identificador exclusivo do produto.
	product_category_name	Categoria principal do produto.
	product_name length	Número de caracteres extraídos do nome do produto.
	product_description_length	Número de caracteres extraídos da

		descrição do produto.
	product_photos_qty	Número de fotos publicadas de produtos.
	product_weight_g	Peso do produto medido em gramas.
	product_length_cm	Comprimento do produto medido em centímetros.
	product_height_cm	Altura do produto medida em centímetros.
	product_width_cm	Largura do produto medida em centímetros.
olist_sellers_dataset	seller_id	Identificador exclusivo do vendedor.
	seller_zip_code_prefix	Primeiros 5 dígitos do CEP do vendedor.
	seller_city	Nome da cidade do vendedor.
	seller_state	Nome do estado do vendedor.

Fonte: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

Jupyter Notebook: Análise Exploratória

Exploratory Data Analysis for Brazilian E-Commerce Public Dataset

Author: Állan Christoffer Pereira Silva

Setup

```
%%capture
!pip install sweetviz

# Importing frameworks
import os
import zipfile
from google.colab import drive
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sweetviz as sv
import missingno

# Google Drive path for Kaggle API token
# NOTE: kaggle.json file must be saved in your Drive
drive.mount('/content/drive', force_remount=True)
drive_path = '/content/drive/MyDrive/'
kaggle_json_file = 'kaggle.json'

# Dataset donwload directly from Kaggle
!mkdir -p ~/.kaggle
!cp "{drive_path}/{kaggle_json_file}" ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets download -d olistbr/brazilian-ecommerce

zip_file = 'brazilian-ecommerce.zip'
with zipfile.ZipFile(zip_file, 'r') as zip_ref:
    zip_ref.extractall()
os.remove(zip_file)
```

```
Mounted at /content/drive
Downloading brazilian-ecommerce.zip to /content
 80% 34.0M/42.6M [00:00<00:00, 77.8MB/s]
100% 42.6M/42.6M [00:00<00:00, 81.9MB/s]
```

```
# Reading original tables
df_customers = pd.read_csv('./olist_customers_dataset.csv')
```

```
df_geoloc = pd.read_csv('./olist_geolocation_dataset.csv')
df_items = pd.read_csv('./olist_order_items_dataset.csv')
df_payments = pd.read_csv('./olist_order_payments_dataset.csv')
df_reviews = pd.read_csv('./olist_order_reviews_dataset.csv')
df_orders = pd.read_csv('./olist_orders_dataset.csv')
df_products = pd.read_csv('./olist_products_dataset.csv')
df_sellers = pd.read_csv('./olist_sellers_dataset.csv')

# Verifying dataframe dimensions
df_list = [df_customers, df_geoloc, df_items, df_payments,
           df_reviews, df_orders, df_products, df_sellers]

for df in df_list:
    print(df.shape)
```

```
(99441, 5)
(1000163, 5)
(112650, 7)
(103886, 5)
(99224, 7)
(99441, 8)
(32951, 9)
(3095, 4)
```

```
# Merging all dataframes
df = pd.merge(df_orders, df_reviews, on='order_id', how='left')
df = pd.merge(df, df_payments, on='order_id', how='left')
df = pd.merge(df, df_items, on='order_id', how='left')
df = pd.merge(df, df_products, on='product_id', how='left')
df = pd.merge(df, df_sellers, on='seller_id', how='left')
df = pd.merge(df, df_customers, on='customer_id', how='left')

df.head()
```

```
                           order_id                         customer_id  \
0  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
1  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
2  e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
3  53cdb2fc8bc7dce0b6741e2150273451  b0830fb4747a6c6d20dea0b8c802d7ef
4  47770eb9100c2d0c44946d9cf07ec65d  41ce2a54c0b03bf3443c3d931a367089

  order_status order_purchase_timestamp    order_approved_at  \
0    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
1    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
2    delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
3    delivered      2018-07-24 20:41:37  2018-07-26 03:24:27
```

```
4    delivered      2018-08-08 08:38:49  2018-08-08 08:55:23

  order_delivered_carrier_date order_delivered_customer_date  \
0          2017-10-04 19:55:00           2017-10-10 21:25:13
1          2017-10-04 19:55:00           2017-10-10 21:25:13
2          2017-10-04 19:55:00           2017-10-10 21:25:13
3          2018-07-26 14:31:00           2018-08-07 15:27:45
4          2018-08-08 13:50:00           2018-08-17 18:06:29

  order_estimated_delivery_date                         review_id  \
0           2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
1           2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
2           2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
3           2018-08-13 00:00:00  8d5266042046a06655c8db133d120ba5
4           2018-09-04 00:00:00  e73b67b67587f7644d5bd1a52deb1b01

   review_score  ... product_length_cm product_height_cm product_width_cm
\
0           4.0  ...             19.0              8.0            13.0
1           4.0  ...             19.0              8.0            13.0
2           4.0  ...             19.0              8.0            13.0
3           4.0  ...             19.0             13.0            19.0
4           5.0  ...             24.0             19.0            21.0

  seller_zip_code_prefix     seller_city seller_state  \
0                 9350.0           maua           SP
1                 9350.0           maua           SP
2                 9350.0           maua           SP
3                31570.0  belo horizonte           SP
4                14840.0         guariba           SP

                 customer_unique_id  customer_zip_code_prefix
customer_city  \
0  7c396fd4830fd04220f754e42b4e5bff                      3149      sao
paulo
1  7c396fd4830fd04220f754e42b4e5bff                      3149      sao
paulo
2  7c396fd4830fd04220f754e42b4e5bff                      3149      sao
paulo
3  af07308b275d755c9edb36a90c618231                     47813
barreiras
4  3a653a41f6f9fc3d2a113cf8398680e8                     75265
vianopolis

  customer_state
```

```
0             SP
1             SP
2             SP
3             BA
4             GO

[5 rows x 39 columns]
```

EDA

```
# Verifying merged dataframe dimensions
df.shape
```

```
(119143, 39)
```

```
# Verifying data types
df.dtypes
```

```
order_id                        object
customer_id                     object
order_status                    object
order_purchase_timestamp        object
order_approved_at               object
order_delivered_carrier_date    object
order_delivered_customer_date   object
order_estimated_delivery_date   object
review_id                       object
review_score                   float64
review_comment_title            object
review_comment_message          object
review_creation_date            object
review_answer_timestamp         object
payment_sequential             float64
payment_type                    object
payment_installments           float64
payment_value                  float64
order_item_id                  float64
product_id                      object
seller_id                       object
shipping_limit_date             object
price                          float64
freight_value                  float64
product_category_name           object
product_name_lenght            float64
product_description_lenght     float64
product_photos_qty             float64
product_weight_g               float64
```

```
product_length_cm              float64
product_height_cm              float64
product_width_cm               float64
seller_zip_code_prefix         float64
seller_city                     object
seller_state                    object
customer_unique_id              object
customer_zip_code_prefix         int64
customer_city                   object
customer_state                  object
dtype: object
```

```
# Column types
categ_cols = df.select_dtypes(include=['object']).columns.tolist()
print(f'Categorical columns = {categ_cols}')
num_cols = df.select_dtypes(exclude=['object']).columns.tolist()
print(f'Numerical columns = {num_cols}')
```

```
Categorical columns = ['order_id', 'customer_id', 'order_status',
'order_purchase_timestamp', 'order_approved_at',
'order_delivered_carrier_date', 'order_delivered_customer_date',
'order_estimated_delivery_date', 'review_id', 'review_comment_title',
'review_comment_message', 'review_creation_date',
'review_answer_timestamp', 'payment_type', 'product_id', 'seller_id',
'shipping_limit_date', 'product_category_name', 'seller_city',
'seller_state', 'customer_unique_id', 'customer_city', 'customer_state']
Numerical columns = ['review_score', 'payment_sequential',
'payment_installments', 'payment_value', 'order_item_id', 'price',
'freight_value', 'product_name_lenght', 'product_description_lenght',
'product_photos_qty', 'product_weight_g', 'product_length_cm',
'product_height_cm', 'product_width_cm', 'seller_zip_code_prefix',
'customer_zip_code_prefix']
```

```
# Verifying unique labels in categorical columns
for col in categ_cols:
    print(f'{col} - unique values =
{df[col].nunique()}\n{df[col].unique()}\n')
```

```
order_id - unique values = 99441
['e481f51cbdc54678b7cc49136f2d6af7' '53cdb2fc8bc7dce0b6741e2150273451'
 '47770eb9100c2d0c44946d9cf07ec65d' ... '83c1379a015df1e13d02aae0204711ab'
 '11c177c8e97725db2631073c19f07b62' '66dea50a8b16d9b4dee7af250b4be1a5']

customer_id - unique values = 99441
['9ef432eb6251297304e76186b10a928d' 'b0830fb4747a6c6d20dea0b8c802d7ef'
 '41ce2a54c0b03bf3443c3d931a367089' ... '1aa71eb042121263aafbe80c1b562c9c'
```

```
 'b331b74b18dc79bcdf6532d51e1637c1' 'edb027a75a1449115f6b43211ae02a24']

order_status - unique values = 8
['delivered' 'invoiced' 'shipped' 'processing' 'unavailable' 'canceled'
 'created' 'approved']

order_purchase_timestamp - unique values = 98875
['2017-10-02 10:56:33' '2018-07-24 20:41:37' '2018-08-08 08:38:49' ...
 '2017-08-27 14:46:43' '2018-01-08 21:28:27' '2018-03-08 20:57:30']

order_approved_at - unique values = 90733
['2017-10-02 11:07:15' '2018-07-26 03:24:27' '2018-08-08 08:55:23' ...
 '2017-08-27 15:04:16' '2018-01-08 21:36:21' '2018-03-09 11:20:28']

order_delivered_carrier_date - unique values = 81018
['2017-10-04 19:55:00' '2018-07-26 14:31:00' '2018-08-08 13:50:00' ...
 '2017-08-28 20:52:26' '2018-01-12 15:35:03' '2018-03-09 22:11:59']

order_delivered_customer_date - unique values = 95664
['2017-10-10 21:25:13' '2018-08-07 15:27:45' '2018-08-17 18:06:29' ...
 '2017-09-21 11:24:17' '2018-01-25 23:32:54' '2018-03-16 13:08:30']

order_estimated_delivery_date - unique values = 459
['2017-10-18 00:00:00' '2018-08-13 00:00:00' '2018-09-04 00:00:00'
 '2017-12-15 00:00:00' '2018-02-26 00:00:00' '2017-08-01 00:00:00'
 '2017-05-09 00:00:00' '2017-06-07 00:00:00' '2017-03-06 00:00:00'
 '2017-08-23 00:00:00' '2017-08-08 00:00:00' '2018-07-18 00:00:00'
 '2018-08-08 00:00:00' '2018-03-21 00:00:00' '2018-07-04 00:00:00'
 '2018-02-06 00:00:00' '2018-01-29 00:00:00' '2017-12-11 00:00:00'
 '2017-11-23 00:00:00' '2017-09-28 00:00:00' '2018-03-29 00:00:00'
 '2018-02-21 00:00:00' '2018-08-17 00:00:00' '2018-03-12 00:00:00'
 '2018-03-28 00:00:00' '2018-05-23 00:00:00' '2018-04-13 00:00:00'
 '2018-05-15 00:00:00' '2018-01-08 00:00:00' '2018-03-07 00:00:00'
 '2018-08-06 00:00:00' '2018-03-20 00:00:00' '2017-08-22 00:00:00'
 '2018-07-17 00:00:00' '2018-04-12 00:00:00' '2017-06-12 00:00:00'
 '2017-12-21 00:00:00' '2017-09-01 00:00:00' '2018-09-13 00:00:00'
 '2018-06-28 00:00:00' '2017-06-09 00:00:00' '2018-05-25 00:00:00'
 '2017-08-31 00:00:00' '2018-02-23 00:00:00' '2018-07-20 00:00:00'
 '2018-08-16 00:00:00' '2018-01-16 00:00:00' '2017-09-20 00:00:00'
 '2018-07-16 00:00:00' '2018-07-05 00:00:00' '2018-04-02 00:00:00'
 '2017-03-30 00:00:00' '2017-07-06 00:00:00' '2017-12-18 00:00:00'
 '2018-08-15 00:00:00' '2017-12-05 00:00:00' '2018-03-13 00:00:00'
 '2018-02-14 00:00:00' '2018-07-13 00:00:00' '2018-06-26 00:00:00'
 '2018-08-02 00:00:00' '2017-09-25 00:00:00' '2018-05-08 00:00:00'
 '2017-03-21 00:00:00' '2017-05-12 00:00:00' '2017-10-11 00:00:00'
```

```
'2018-08-30 00:00:00' '2017-08-16 00:00:00' '2018-01-19 00:00:00'
'2017-04-27 00:00:00' '2017-06-01 00:00:00' '2017-05-25 00:00:00'
'2017-11-21 00:00:00' '2018-01-03 00:00:00' '2017-09-21 00:00:00'
'2018-06-05 00:00:00' '2018-02-19 00:00:00' '2018-05-16 00:00:00'
'2017-10-13 00:00:00' '2018-05-21 00:00:00' '2018-01-22 00:00:00'
'2018-05-07 00:00:00' '2018-08-27 00:00:00' '2018-06-08 00:00:00'
'2017-04-26 00:00:00' '2018-07-23 00:00:00' '2017-06-06 00:00:00'
'2018-08-21 00:00:00' '2018-03-26 00:00:00' '2017-03-10 00:00:00'
'2017-07-25 00:00:00' '2017-10-16 00:00:00' '2017-12-22 00:00:00'
'2018-09-05 00:00:00' '2018-08-10 00:00:00' '2018-05-29 00:00:00'
'2017-12-19 00:00:00' '2017-10-17 00:00:00' '2017-07-10 00:00:00'
'2018-05-04 00:00:00' '2018-05-14 00:00:00' '2017-08-04 00:00:00'
'2017-10-03 00:00:00' '2017-12-14 00:00:00' '2017-10-31 00:00:00'
'2018-01-04 00:00:00' '2018-04-20 00:00:00' '2018-03-08 00:00:00'
'2018-07-30 00:00:00' '2017-04-17 00:00:00' '2017-07-28 00:00:00'
'2018-06-04 00:00:00' '2018-07-19 00:00:00' '2018-03-16 00:00:00'
'2018-01-31 00:00:00' '2017-05-29 00:00:00' '2017-12-27 00:00:00'
'2018-06-12 00:00:00' '2017-12-20 00:00:00' '2018-03-09 00:00:00'
'2017-06-05 00:00:00' '2018-02-07 00:00:00' '2017-06-08 00:00:00'
'2017-08-11 00:00:00' '2018-07-27 00:00:00' '2018-07-25 00:00:00'
'2017-04-11 00:00:00' '2017-11-09 00:00:00' '2018-04-11 00:00:00'
'2018-07-11 00:00:00' '2017-09-27 00:00:00' '2018-04-26 00:00:00'
'2018-02-15 00:00:00' '2018-05-02 00:00:00' '2017-10-20 00:00:00'
'2017-05-15 00:00:00' '2018-02-02 00:00:00' '2017-04-10 00:00:00'
'2018-08-23 00:00:00' '2017-07-18 00:00:00' '2017-08-07 00:00:00'
'2017-08-03 00:00:00' '2017-07-14 00:00:00' '2018-06-06 00:00:00'
'2018-08-09 00:00:00' '2017-08-21 00:00:00' '2018-07-31 00:00:00'
'2017-03-28 00:00:00' '2018-02-01 00:00:00' '2018-05-03 00:00:00'
'2017-06-16 00:00:00' '2017-12-26 00:00:00' '2017-06-28 00:00:00'
'2017-10-04 00:00:00' '2018-05-11 00:00:00' '2017-10-27 00:00:00'
'2018-03-06 00:00:00' '2017-12-06 00:00:00' '2017-06-26 00:00:00'
'2018-04-19 00:00:00' '2018-05-28 00:00:00' '2018-05-09 00:00:00'
'2017-05-11 00:00:00' '2017-12-13 00:00:00' '2018-01-24 00:00:00'
'2018-03-22 00:00:00' '2018-04-24 00:00:00' '2017-02-13 00:00:00'
'2017-05-10 00:00:00' '2018-07-12 00:00:00' '2018-04-27 00:00:00'
'2017-03-16 00:00:00' '2018-03-05 00:00:00' '2017-12-12 00:00:00'
'2018-02-08 00:00:00' '2017-03-17 00:00:00' '2018-07-24 00:00:00'
'2017-10-30 00:00:00' '2018-02-22 00:00:00' '2018-05-30 00:00:00'
'2018-03-23 00:00:00' '2018-04-16 00:00:00' '2018-05-24 00:00:00'
'2018-04-05 00:00:00' '2018-04-03 00:00:00' '2018-02-20 00:00:00'
'2017-11-27 00:00:00' '2018-03-01 00:00:00' '2018-08-14 00:00:00'
'2017-07-19 00:00:00' '2018-04-17 00:00:00' '2018-08-03 00:00:00'
'2018-04-06 00:00:00' '2018-04-09 00:00:00' '2017-03-02 00:00:00'
'2017-10-23 00:00:00' '2018-01-02 00:00:00' '2017-06-23 00:00:00'
'2018-01-30 00:00:00' '2017-09-13 00:00:00' '2018-07-03 00:00:00'
```

```
'2016-12-09 00:00:00' '2017-08-17 00:00:00' '2018-01-05 00:00:00'
'2018-08-24 00:00:00' '2018-02-05 00:00:00' '2018-05-18 00:00:00'
'2018-07-26 00:00:00' '2017-09-04 00:00:00' '2018-08-20 00:00:00'
'2018-09-21 00:00:00' '2018-03-19 00:00:00' '2018-09-12 00:00:00'
'2018-08-28 00:00:00' '2017-11-08 00:00:00' '2017-05-19 00:00:00'
'2018-04-25 00:00:00' '2018-01-17 00:00:00' '2017-11-07 00:00:00'
'2017-11-14 00:00:00' '2017-11-29 00:00:00' '2017-04-03 00:00:00'
'2017-07-11 00:00:00' '2017-06-29 00:00:00' '2018-06-14 00:00:00'
'2016-12-07 00:00:00' '2017-04-25 00:00:00' '2017-11-17 00:00:00'
'2018-08-22 00:00:00' '2017-07-05 00:00:00' '2017-05-18 00:00:00'
'2017-12-07 00:00:00' '2018-01-12 00:00:00' '2017-05-04 00:00:00'
'2017-11-06 00:00:00' '2017-09-18 00:00:00' '2017-05-31 00:00:00'
'2018-01-26 00:00:00' '2018-01-23 00:00:00' '2017-11-03 00:00:00'
'2017-10-02 00:00:00' '2017-08-14 00:00:00' '2018-09-18 00:00:00'
'2017-07-04 00:00:00' '2017-08-29 00:00:00' '2017-10-09 00:00:00'
'2018-04-04 00:00:00' '2017-12-08 00:00:00' '2017-11-01 00:00:00'
'2018-06-29 00:00:00' '2017-06-27 00:00:00' '2018-03-27 00:00:00'
'2017-08-28 00:00:00' '2017-03-09 00:00:00' '2017-05-05 00:00:00'
'2017-03-24 00:00:00' '2018-03-02 00:00:00' '2017-03-23 00:00:00'
'2017-02-20 00:00:00' '2017-06-14 00:00:00' '2017-12-01 00:00:00'
'2018-01-09 00:00:00' '2018-04-23 00:00:00' '2017-07-31 00:00:00'
'2018-06-19 00:00:00' '2017-04-12 00:00:00' '2018-08-31 00:00:00'
'2017-07-17 00:00:00' '2017-03-14 00:00:00' '2017-09-29 00:00:00'
'2018-05-22 00:00:00' '2017-10-10 00:00:00' '2017-04-20 00:00:00'
'2017-03-03 00:00:00' '2017-12-29 00:00:00' '2018-06-07 00:00:00'
'2018-01-18 00:00:00' '2018-02-09 00:00:00' '2017-05-16 00:00:00'
'2017-09-08 00:00:00' '2018-06-13 00:00:00' '2018-06-21 00:00:00'
'2018-02-16 00:00:00' '2017-08-25 00:00:00' '2017-08-15 00:00:00'
'2017-11-10 00:00:00' '2018-06-20 00:00:00' '2018-06-01 00:00:00'
'2018-03-15 00:00:00' '2017-12-28 00:00:00' '2017-03-29 00:00:00'
'2017-11-22 00:00:00' '2018-03-14 00:00:00' '2018-05-17 00:00:00'
'2017-10-19 00:00:00' '2018-08-01 00:00:00' '2017-05-08 00:00:00'
'2017-04-24 00:00:00' '2017-03-27 00:00:00' '2017-07-12 00:00:00'
'2017-07-27 00:00:00' '2018-01-11 00:00:00' '2017-06-02 00:00:00'
'2017-05-30 00:00:00' '2017-09-05 00:00:00' '2017-03-13 00:00:00'
'2017-12-04 00:00:00' '2016-11-29 00:00:00' '2017-02-16 00:00:00'
'2017-10-05 00:00:00' '2017-08-02 00:00:00' '2017-11-16 00:00:00'
'2018-05-10 00:00:00' '2018-06-11 00:00:00' '2017-06-30 00:00:00'
'2018-04-10 00:00:00' '2017-09-06 00:00:00' '2017-06-19 00:00:00'
'2017-11-13 00:00:00' '2017-04-13 00:00:00' '2017-05-02 00:00:00'
'2018-08-07 00:00:00' '2018-01-15 00:00:00' '2017-04-04 00:00:00'
'2018-08-29 00:00:00' '2017-04-07 00:00:00' '2018-06-18 00:00:00'
'2017-08-10 00:00:00' '2017-05-17 00:00:00' '2017-08-09 00:00:00'
'2017-09-26 00:00:00' '2017-08-30 00:00:00' '2017-04-06 00:00:00'
'2017-05-26 00:00:00' '2018-09-20 00:00:00' '2017-07-13 00:00:00'
```

```
 '2017-06-20 00:00:00' '2017-07-24 00:00:00' '2017-04-05 00:00:00'
 '2018-09-17 00:00:00' '2017-05-23 00:00:00' '2017-11-28 00:00:00'
 '2017-08-18 00:00:00' '2017-05-03 00:00:00' '2017-09-14 00:00:00'
 '2017-11-30 00:00:00' '2017-07-20 00:00:00' '2017-05-24 00:00:00'
 '2017-02-22 00:00:00' '2018-04-30 00:00:00' '2017-10-25 00:00:00'
 '2017-07-03 00:00:00' '2018-09-03 00:00:00' '2017-09-22 00:00:00'
 '2017-03-07 00:00:00' '2017-06-21 00:00:00' '2018-09-14 00:00:00'
 '2017-11-24 00:00:00' '2017-03-22 00:00:00' '2017-02-17 00:00:00'
 '2017-07-21 00:00:00' '2017-08-24 00:00:00' '2017-07-07 00:00:00'
 '2017-09-11 00:00:00' '2017-06-22 00:00:00' '2017-09-12 00:00:00'
 '2017-06-13 00:00:00' '2017-10-24 00:00:00' '2017-10-06 00:00:00'
 '2017-02-24 00:00:00' '2016-11-30 00:00:00' '2017-03-01 00:00:00'
 '2016-12-01 00:00:00' '2017-02-28 00:00:00' '2017-04-19 00:00:00'
 '2018-09-11 00:00:00' '2017-03-20 00:00:00' '2018-04-18 00:00:00'
 '2018-01-10 00:00:00' '2018-10-17 00:00:00' '2018-06-25 00:00:00'
 '2018-09-06 00:00:00' '2018-09-10 00:00:00' '2017-09-15 00:00:00'
 '2017-03-08 00:00:00' '2017-03-31 00:00:00' '2017-10-26 00:00:00'
 '2017-04-18 00:00:00' '2017-09-19 00:00:00' '2018-09-19 00:00:00'
 '2018-09-25 00:00:00' '2018-06-27 00:00:00' '2017-07-26 00:00:00'
 '2017-05-22 00:00:00' '2016-11-16 00:00:00' '2017-02-15 00:00:00'
 '2017-03-15 00:00:00' '2018-09-27 00:00:00' '2016-11-25 00:00:00'
 '2016-10-28 00:00:00' '2016-10-20 00:00:00' '2018-10-02 00:00:00'
 '2016-12-05 00:00:00' '2017-02-01 00:00:00' '2016-12-08 00:00:00'
 '2018-09-28 00:00:00' '2016-11-28 00:00:00' '2018-06-22 00:00:00'
 '2018-10-10 00:00:00' '2017-02-21 00:00:00' '2017-04-28 00:00:00'
 '2017-02-23 00:00:00' '2017-02-07 00:00:00' '2018-07-02 00:00:00'
 '2017-02-27 00:00:00' '2016-11-23 00:00:00' '2016-11-18 00:00:00'
 '2016-12-12 00:00:00' '2016-12-14 00:00:00' '2018-10-01 00:00:00'
 '2018-09-26 00:00:00' '2016-09-30 00:00:00' '2016-12-02 00:00:00'
 '2018-10-15 00:00:00' '2018-09-24 00:00:00' '2017-02-14 00:00:00'
 '2016-11-24 00:00:00' '2018-10-03 00:00:00' '2016-12-06 00:00:00'
 '2017-01-09 00:00:00' '2018-10-04 00:00:00' '2017-04-14 00:00:00'
 '2018-02-13 00:00:00' '2016-11-07 00:00:00' '2016-11-14 00:00:00'
 '2018-10-05 00:00:00' '2016-10-04 00:00:00' '2018-10-16 00:00:00'
 '2016-12-13 00:00:00' '2018-10-11 00:00:00' '2018-10-25 00:00:00'
 '2016-12-16 00:00:00' '2016-11-17 00:00:00' '2016-12-20 00:00:00'
 '2017-01-19 00:00:00' '2017-02-09 00:00:00' '2016-10-24 00:00:00'
 '2016-12-30 00:00:00' '2017-02-10 00:00:00' '2018-07-06 00:00:00'
 '2018-10-30 00:00:00' '2018-10-23 00:00:00' '2018-11-12 00:00:00'
 '2016-12-19 00:00:00' '2016-12-23 00:00:00' '2017-01-11 00:00:00'
 '2016-10-25 00:00:00' '2018-07-10 00:00:00' '2016-10-27 00:00:00']

review_id - unique values = 98410
['a54f0611adc9ed256b57ede6b6eb5114' '8d5266042046a06655c8db133d120ba5'
 'e73b67b67587f7644d5bd1a52deb1b01' ... '371579771219f6db2d830d50805977bb'
```

```
 '8ab6855b9fe9b812cd03a480a25058a1' 'dc9c59b4688062c25758c2be4cafc523']

review_comment_title - unique values = 4527
[nan 'Muito boa a loja' 'Nota dez' ... 'TAvaliação da entrega Car'
 'Muito frágil !!!' 'Bom.maravilha']

review_comment_message - unique values = 36159
['Não testei o produto ainda, mas ele veio correto e em boas condições.
Apenas a caixa que veio bem amassada e danificada, o que ficará chato, pois
se trata de um presente.'
 'Muito bom o produto.' nan ...
 'Ele não é um mini cajon, é um shaker, ou seja, um chocalho que imita o
cajon. Péssimo. '
 'So uma peça que veio rachado mas tudo bem rs'
 'Foi entregue somente 1. Quero saber do outro produto.']

review_creation_date - unique values = 636
['2017-10-11 00:00:00' '2018-08-08 00:00:00' '2018-08-18 00:00:00'
 '2017-12-03 00:00:00' '2018-02-17 00:00:00' '2017-07-27 00:00:00'
 '2017-05-13 00:00:00' '2017-05-27 00:00:00' '2017-02-03 00:00:00'
 '2017-08-17 00:00:00' '2017-05-30 00:00:00' '2017-07-20 00:00:00'
 '2018-06-20 00:00:00' '2018-07-31 00:00:00' '2018-03-13 00:00:00'
 '2018-07-03 00:00:00' nan '2018-01-09 00:00:00' '2017-11-28 00:00:00'
 '2017-11-09 00:00:00' '2017-10-01 00:00:00' '2018-03-20 00:00:00'
 '2018-02-09 00:00:00' '2018-01-27 00:00:00' '2018-08-14 00:00:00'
 '2018-03-16 00:00:00' '2018-06-17 00:00:00' '2018-03-22 00:00:00'
 '2018-05-17 00:00:00' '2018-03-28 00:00:00' '2017-08-19 00:00:00'
 '2018-05-06 00:00:00' '2018-01-04 00:00:00' '2018-02-23 00:00:00'
 '2018-08-01 00:00:00' '2018-03-14 00:00:00' '2017-08-09 00:00:00'
 '2018-06-19 00:00:00' '2018-04-06 00:00:00' '2017-06-01 00:00:00'
 '2017-12-22 00:00:00' '2017-08-13 00:00:00' '2018-07-01 00:00:00'
 '2017-05-26 00:00:00' '2018-05-22 00:00:00' '2018-08-03 00:00:00'
 '2018-02-15 00:00:00' '2018-07-04 00:00:00' '2017-09-05 00:00:00'
 '2018-04-04 00:00:00' '2017-04-02 00:00:00' '2017-06-22 00:00:00'
 '2017-12-15 00:00:00' '2018-08-11 00:00:00' '2018-03-10 00:00:00'
 '2018-04-25 00:00:00' '2018-01-18 00:00:00' '2018-02-21 00:00:00'
 '2018-06-23 00:00:00' '2018-07-25 00:00:00' '2017-09-17 00:00:00'
 '2018-04-20 00:00:00' '2018-02-25 00:00:00' '2017-03-11 00:00:00'
 '2017-05-11 00:00:00' '2017-09-26 00:00:00' '2018-08-07 00:00:00'
 '2018-08-30 00:00:00' '2017-08-02 00:00:00' '2018-01-10 00:00:00'
 '2017-04-12 00:00:00' '2017-05-19 00:00:00' '2017-05-12 00:00:00'
 '2017-12-20 00:00:00' '2017-09-09 00:00:00' '2018-03-17 00:00:00'
 '2018-02-16 00:00:00' '2018-05-05 00:00:00' '2018-04-28 00:00:00'
 '2018-05-01 00:00:00' '2018-08-17 00:00:00' '2018-05-19 00:00:00'
 '2017-12-13 00:00:00' '2017-04-11 00:00:00' '2018-08-29 00:00:00'
```

```
'2017-05-24 00:00:00' '2018-08-04 00:00:00' '2018-08-25 00:00:00'
'2017-02-12 00:00:00' '2018-02-10 00:00:00' '2017-07-13 00:00:00'
'2017-12-29 00:00:00' '2018-02-03 00:00:00' '2018-07-14 00:00:00'
'2018-05-23 00:00:00' '2017-06-30 00:00:00' '2018-02-08 00:00:00'
'2018-04-08 00:00:00' '2018-05-15 00:00:00' '2017-04-28 00:00:00'
'2017-12-02 00:00:00' '2017-07-22 00:00:00' '2017-10-05 00:00:00'
'2017-11-29 00:00:00' '2017-10-25 00:00:00' '2017-12-12 00:00:00'
'2018-03-30 00:00:00' '2018-01-16 00:00:00' '2018-07-27 00:00:00'
'2017-04-01 00:00:00' '2018-06-16 00:00:00' '2017-07-18 00:00:00'
'2018-05-25 00:00:00' '2018-07-12 00:00:00' '2017-09-29 00:00:00'
'2018-03-21 00:00:00' '2018-01-25 00:00:00' '2017-12-09 00:00:00'
'2018-05-16 00:00:00' '2018-05-31 00:00:00' '2017-12-06 00:00:00'
'2018-08-02 00:00:00' '2017-08-01 00:00:00' '2018-06-29 00:00:00'
'2018-07-07 00:00:00' '2017-03-21 00:00:00' '2017-10-21 00:00:00'
'2018-01-05 00:00:00' '2017-06-14 00:00:00' '2018-06-06 00:00:00'
'2018-08-15 00:00:00' '2017-12-07 00:00:00' '2018-05-03 00:00:00'
'2018-07-24 00:00:00' '2017-12-14 00:00:00' '2018-04-07 00:00:00'
'2018-06-28 00:00:00' '2018-04-21 00:00:00' '2018-01-24 00:00:00'
'2018-04-17 00:00:00' '2017-10-10 00:00:00' '2018-03-23 00:00:00'
'2017-04-29 00:00:00' '2017-09-22 00:00:00' '2018-01-21 00:00:00'
'2017-04-04 00:00:00' '2018-04-09 00:00:00' '2017-05-31 00:00:00'
'2017-06-27 00:00:00' '2017-07-05 00:00:00' '2018-06-08 00:00:00'
'2017-08-06 00:00:00' '2017-03-17 00:00:00' '2018-02-20 00:00:00'
'2017-05-16 00:00:00' '2017-06-17 00:00:00' '2017-09-23 00:00:00'
'2018-06-09 00:00:00' '2018-06-07 00:00:00' '2018-02-24 00:00:00'
'2018-06-02 00:00:00' '2018-08-21 00:00:00' '2018-04-27 00:00:00'
'2017-10-14 00:00:00' '2018-02-19 00:00:00' '2017-11-30 00:00:00'
'2018-01-23 00:00:00' '2017-06-13 00:00:00' '2018-04-14 00:00:00'
'2017-07-28 00:00:00' '2018-05-18 00:00:00' '2018-08-16 00:00:00'
'2018-05-04 00:00:00' '2017-05-03 00:00:00' '2017-12-19 00:00:00'
'2017-12-16 00:00:00' '2017-12-28 00:00:00' '2018-03-06 00:00:00'
'2017-01-18 00:00:00' '2017-05-04 00:00:00' '2018-05-10 00:00:00'
'2018-06-27 00:00:00' '2017-07-12 00:00:00' '2018-01-31 00:00:00'
'2018-03-01 00:00:00' '2017-03-02 00:00:00' '2018-03-07 00:00:00'
'2018-02-06 00:00:00' '2017-02-21 00:00:00' '2018-07-05 00:00:00'
'2017-09-13 00:00:00' '2018-02-07 00:00:00' '2017-11-17 00:00:00'
'2018-07-10 00:00:00' '2018-04-24 00:00:00' '2018-06-12 00:00:00'
'2018-03-15 00:00:00' '2017-07-11 00:00:00' '2018-04-10 00:00:00'
'2018-08-05 00:00:00' '2017-01-31 00:00:00' '2017-10-04 00:00:00'
'2018-03-25 00:00:00' '2018-03-27 00:00:00' '2017-09-20 00:00:00'
'2017-08-30 00:00:00' '2017-09-19 00:00:00' '2018-08-24 00:00:00'
'2018-06-13 00:00:00' '2016-12-11 00:00:00' '2018-03-11 00:00:00'
'2018-08-23 00:00:00' '2018-01-28 00:00:00' '2018-04-15 00:00:00'
'2018-07-08 00:00:00' '2017-06-10 00:00:00' '2018-04-05 00:00:00'
'2017-05-10 00:00:00' '2017-08-26 00:00:00' '2018-08-28 00:00:00'
```

```
'2018-07-11 00:00:00' '2018-08-31 00:00:00' '2018-03-08 00:00:00'
'2017-10-26 00:00:00' '2017-05-20 00:00:00' '2018-05-21 00:00:00'
'2018-08-22 00:00:00' '2018-01-13 00:00:00' '2017-10-29 00:00:00'
'2017-08-15 00:00:00' '2018-07-17 00:00:00' '2017-11-02 00:00:00'
'2017-03-03 00:00:00' '2018-06-05 00:00:00' '2018-05-24 00:00:00'
'2018-02-18 00:00:00' '2017-12-04 00:00:00' '2016-10-21 00:00:00'
'2018-03-19 00:00:00' '2017-04-07 00:00:00' '2017-11-15 00:00:00'
'2017-12-24 00:00:00' '2017-11-21 00:00:00' '2017-05-09 00:00:00'
'2018-02-04 00:00:00' '2017-08-22 00:00:00' '2017-09-18 00:00:00'
'2017-11-25 00:00:00' '2017-12-27 00:00:00' '2017-12-11 00:00:00'
'2017-05-06 00:00:00' '2017-10-28 00:00:00' '2017-09-14 00:00:00'
'2018-01-11 00:00:00' '2017-12-05 00:00:00' '2017-05-17 00:00:00'
'2018-01-12 00:00:00' '2017-06-23 00:00:00' '2017-09-16 00:00:00'
'2018-05-12 00:00:00' '2018-06-26 00:00:00' '2017-06-21 00:00:00'
'2017-08-18 00:00:00' '2017-10-31 00:00:00' '2017-11-14 00:00:00'
'2018-06-25 00:00:00' '2017-12-10 00:00:00' '2018-04-13 00:00:00'
'2017-10-27 00:00:00' '2018-06-22 00:00:00' '2017-05-25 00:00:00'
'2018-03-31 00:00:00' '2017-10-24 00:00:00' '2018-06-21 00:00:00'
'2018-03-04 00:00:00' '2017-02-15 00:00:00' '2017-12-23 00:00:00'
'2018-08-12 00:00:00' '2017-06-03 00:00:00' '2018-04-03 00:00:00'
'2017-10-19 00:00:00' '2017-03-28 00:00:00' '2017-12-30 00:00:00'
'2017-07-01 00:00:00' '2017-03-31 00:00:00' '2017-11-22 00:00:00'
'2017-11-26 00:00:00' '2017-03-22 00:00:00' '2018-08-19 00:00:00'
'2017-08-16 00:00:00' '2017-02-20 00:00:00' '2017-04-08 00:00:00'
'2018-03-09 00:00:00' '2017-02-25 00:00:00' '2017-02-22 00:00:00'
'2018-01-06 00:00:00' '2018-08-10 00:00:00' '2017-08-05 00:00:00'
'2018-03-03 00:00:00' '2018-06-04 00:00:00' '2018-02-22 00:00:00'
'2017-10-22 00:00:00' '2017-11-24 00:00:00' '2017-09-21 00:00:00'
'2018-03-29 00:00:00' '2018-02-01 00:00:00' '2017-09-02 00:00:00'
'2018-05-29 00:00:00' '2017-09-07 00:00:00' '2017-07-26 00:00:00'
'2018-03-24 00:00:00' '2018-06-10 00:00:00' '2017-06-02 00:00:00'
'2018-02-02 00:00:00' '2018-06-14 00:00:00' '2017-12-21 00:00:00'
'2017-07-29 00:00:00' '2017-09-28 00:00:00' '2018-02-28 00:00:00'
'2018-01-03 00:00:00' '2017-03-16 00:00:00' '2018-03-02 00:00:00'
'2017-11-01 00:00:00' '2018-07-26 00:00:00' '2017-03-29 00:00:00'
'2017-07-14 00:00:00' '2018-01-30 00:00:00' '2017-04-21 00:00:00'
'2018-04-26 00:00:00' '2017-09-25 00:00:00' '2017-03-12 00:00:00'
'2017-04-06 00:00:00' '2018-07-28 00:00:00' '2017-12-01 00:00:00'
'2018-05-27 00:00:00' '2017-07-04 00:00:00' '2018-01-07 00:00:00'
'2017-09-15 00:00:00' '2017-09-10 00:00:00' '2016-10-22 00:00:00'
'2017-05-29 00:00:00' '2018-05-07 00:00:00' '2018-08-26 00:00:00'
'2018-04-18 00:00:00' '2018-05-08 00:00:00' '2018-04-11 00:00:00'
'2018-02-27 00:00:00' '2018-05-11 00:00:00' '2017-09-27 00:00:00'
'2017-11-04 00:00:00' '2017-11-18 00:00:00' '2018-07-06 00:00:00'
'2017-08-24 00:00:00' '2018-01-14 00:00:00' '2017-04-19 00:00:00'
```

```
'2017-11-19 00:00:00' '2017-06-08 00:00:00' '2018-01-17 00:00:00'
'2017-08-12 00:00:00' '2017-03-18 00:00:00' '2018-01-19 00:00:00'
'2017-03-30 00:00:00' '2018-06-15 00:00:00' '2018-05-20 00:00:00'
'2017-03-10 00:00:00' '2018-07-29 00:00:00' '2017-04-09 00:00:00'
'2017-08-04 00:00:00' '2017-08-08 00:00:00' '2017-02-02 00:00:00'
'2017-10-06 00:00:00' '2018-06-01 00:00:00' '2017-07-10 00:00:00'
'2017-07-19 00:00:00' '2017-04-20 00:00:00' '2018-05-26 00:00:00'
'2017-04-16 00:00:00' '2017-11-08 00:00:00' '2018-04-19 00:00:00'
'2017-05-28 00:00:00' '2017-09-04 00:00:00' '2017-01-17 00:00:00'
'2017-08-11 00:00:00' '2017-02-09 00:00:00' '2018-04-01 00:00:00'
'2017-02-16 00:00:00' '2018-08-09 00:00:00' '2017-08-03 00:00:00'
'2017-09-06 00:00:00' '2017-07-15 00:00:00' '2017-03-23 00:00:00'
'2017-12-08 00:00:00' '2017-04-18 00:00:00' '2018-05-30 00:00:00'
'2017-11-10 00:00:00' '2017-06-04 00:00:00' '2018-04-12 00:00:00'
'2018-06-30 00:00:00' '2018-06-03 00:00:00' '2017-12-18 00:00:00'
'2017-10-12 00:00:00' '2017-04-30 00:00:00' '2017-05-18 00:00:00'
'2017-08-31 00:00:00' '2017-11-23 00:00:00' '2017-03-15 00:00:00'
'2017-10-18 00:00:00' '2017-10-03 00:00:00' '2018-07-21 00:00:00'
'2018-01-26 00:00:00' '2017-09-30 00:00:00' '2017-03-09 00:00:00'
'2017-11-07 00:00:00' '2018-02-26 00:00:00' '2017-04-13 00:00:00'
'2017-03-08 00:00:00' '2017-08-25 00:00:00' '2017-07-07 00:00:00'
'2017-04-05 00:00:00' '2017-01-27 00:00:00' '2018-06-11 00:00:00'
'2017-06-24 00:00:00' '2017-08-10 00:00:00' '2017-05-23 00:00:00'
'2017-10-20 00:00:00' '2017-06-20 00:00:00' '2017-11-05 00:00:00'
'2017-09-12 00:00:00' '2017-09-03 00:00:00' '2017-03-07 00:00:00'
'2018-03-18 00:00:00' '2017-12-17 00:00:00' '2017-06-26 00:00:00'
'2017-03-04 00:00:00' '2017-10-17 00:00:00' '2017-11-16 00:00:00'
'2016-10-25 00:00:00' '2017-02-10 00:00:00' '2016-10-27 00:00:00'
'2017-08-29 00:00:00' '2017-03-14 00:00:00' '2018-07-13 00:00:00'
'2017-05-05 00:00:00' '2017-07-31 00:00:00' '2017-04-14 00:00:00'
'2018-04-29 00:00:00' '2017-02-23 00:00:00' '2017-08-23 00:00:00'
'2018-05-09 00:00:00' '2017-06-07 00:00:00' '2017-01-21 00:00:00'
'2017-06-15 00:00:00' '2017-02-24 00:00:00' '2017-07-06 00:00:00'
'2017-02-04 00:00:00' '2017-07-25 00:00:00' '2017-06-28 00:00:00'
'2018-04-16 00:00:00' '2017-07-21 00:00:00' '2017-06-29 00:00:00'
'2017-06-16 00:00:00' '2017-08-20 00:00:00' '2018-07-15 00:00:00'
'2017-12-31 00:00:00' '2018-05-13 00:00:00' '2017-09-01 00:00:00'
'2017-04-23 00:00:00' '2017-11-11 00:00:00' '2017-02-26 00:00:00'
'2017-10-07 00:00:00' '2017-07-30 00:00:00' '2018-08-20 00:00:00'
'2017-06-25 00:00:00' '2017-07-08 00:00:00' '2018-02-05 00:00:00'
'2017-02-14 00:00:00' '2017-03-25 00:00:00' '2018-01-20 00:00:00'
'2018-01-29 00:00:00' '2017-11-03 00:00:00' '2017-08-27 00:00:00'
'2017-10-23 00:00:00' '2017-10-13 00:00:00' '2017-03-05 00:00:00'
'2018-04-23 00:00:00' '2018-02-11 00:00:00' '2017-07-23 00:00:00'
'2017-10-15 01:00:00' '2017-03-24 00:00:00' '2017-10-02 00:00:00'
```

```
'2016-10-28 00:00:00' '2017-06-06 00:00:00' '2017-10-09 00:00:00'
'2018-08-27 00:00:00' '2017-02-07 00:00:00' '2017-09-24 00:00:00'
'2017-04-22 00:00:00' '2018-01-01 00:00:00' '2016-11-20 00:00:00'
'2017-06-11 00:00:00' '2017-10-08 00:00:00' '2016-11-01 00:00:00'
'2017-02-01 00:00:00' '2017-01-24 00:00:00' '2018-05-02 00:00:00'
'2017-02-08 00:00:00' '2017-06-09 00:00:00' '2018-06-24 00:00:00'
'2017-02-11 00:00:00' '2018-04-22 00:00:00' '2017-08-21 00:00:00'
'2016-11-18 00:00:00' '2017-11-27 00:00:00' '2018-06-18 00:00:00'
'2017-05-01 00:00:00' '2017-03-26 00:00:00' '2017-11-12 00:00:00'
'2017-05-07 00:00:00' '2017-12-25 00:00:00' '2017-07-16 00:00:00'
'2018-07-22 00:00:00' '2018-08-06 00:00:00' '2017-05-21 00:00:00'
'2016-10-29 00:00:00' '2016-10-18 00:00:00' '2016-10-26 00:00:00'
'2017-03-19 00:00:00' '2017-02-18 00:00:00' '2016-10-19 00:00:00'
'2017-03-20 00:00:00' '2017-01-29 00:00:00' '2018-03-26 00:00:00'
'2018-07-18 00:00:00' '2017-02-17 00:00:00' '2017-07-24 00:00:00'
'2018-07-19 00:00:00' '2017-01-14 00:00:00' '2018-03-12 00:00:00'
'2018-03-05 00:00:00' '2017-05-14 00:00:00' '2017-05-08 00:00:00'
'2018-01-15 00:00:00' '2018-07-20 00:00:00' '2018-04-02 00:00:00'
'2017-07-02 00:00:00' '2016-11-23 00:00:00' '2016-11-11 00:00:00'
'2016-11-17 00:00:00' '2017-09-08 00:00:00' '2017-07-09 00:00:00'
'2017-06-18 00:00:00' '2016-11-19 00:00:00' '2017-04-15 00:00:00'
'2017-01-26 00:00:00' '2018-01-08 00:00:00' '2018-02-12 00:00:00'
'2017-11-20 00:00:00' '2016-10-20 00:00:00' '2016-10-02 00:00:00'
'2017-02-13 00:00:00' '2017-07-03 00:00:00' '2016-12-14 00:00:00'
'2016-11-02 00:00:00' '2017-05-02 00:00:00' '2017-04-10 00:00:00'
'2017-01-28 00:00:00' '2017-01-20 00:00:00' '2018-01-22 00:00:00'
'2018-07-16 00:00:00' '2017-05-22 00:00:00' '2016-12-03 00:00:00'
'2016-10-16 01:00:00' '2018-07-02 00:00:00' '2017-08-14 00:00:00'
'2016-11-10 00:00:00' '2017-04-24 00:00:00' '2017-02-19 00:00:00'
'2016-11-09 00:00:00' '2016-10-30 00:00:00' '2017-10-16 00:00:00'
'2017-01-25 00:00:00' '2017-08-28 00:00:00' '2017-09-11 00:00:00'
'2016-12-07 00:00:00' '2016-11-08 00:00:00' '2018-07-09 00:00:00'
'2016-12-16 00:00:00' '2016-11-05 00:00:00' '2017-01-19 00:00:00'
'2017-04-03 00:00:00' '2017-02-05 00:00:00' '2016-11-26 00:00:00'
'2017-06-19 00:00:00' '2016-10-15 00:00:00' '2016-12-04 00:00:00'
'2018-01-02 00:00:00' '2016-12-09 00:00:00' '2017-03-06 00:00:00'
'2018-02-13 00:00:00' '2016-10-06 00:00:00' '2017-03-13 00:00:00'
'2016-11-29 00:00:00' '2018-04-30 00:00:00' '2017-11-13 00:00:00'
'2016-12-01 00:00:00' '2017-01-22 00:00:00' '2016-12-02 00:00:00'
'2016-12-08 00:00:00' '2017-03-01 00:00:00' '2016-11-25 00:00:00'
'2017-06-12 00:00:00' '2018-05-14 00:00:00' '2016-12-10 00:00:00'
'2016-12-29 00:00:00' '2018-07-30 00:00:00' '2016-11-04 00:00:00'
'2018-08-13 00:00:00' '2016-10-23 00:00:00' '2016-11-30 00:00:00'
'2017-06-05 00:00:00' '2016-11-27 00:00:00' '2017-05-15 00:00:00'
'2016-10-09 00:00:00' '2017-01-04 00:00:00' '2016-10-24 00:00:00'
```

```
 '2017-01-13 00:00:00' '2017-07-17 00:00:00' '2016-10-31 00:00:00'
 '2016-12-25 00:00:00' '2018-05-28 00:00:00' '2017-03-27 00:00:00'
 '2017-01-12 00:00:00' '2017-02-06 00:00:00' '2016-11-06 00:00:00'
 '2017-10-30 00:00:00' '2018-07-23 00:00:00' '2016-11-15 00:00:00']

review_answer_timestamp - unique values = 98248
['2017-10-12 03:43:48' '2018-08-08 18:37:50' '2018-08-22 19:07:58' ...
 '2017-09-22 23:10:57' '2018-01-27 09:16:56' '2018-03-17 16:33:31']

payment_type - unique values = 5
['credit_card' 'voucher' 'boleto' 'debit_card' 'not_defined' nan]

product_id - unique values = 32951
['87285b34884572647811a353c7ac498a' '595fac2a385ac33a80bd5114aec74eb8'
 'aa4383b373c6aca5d8797843e5594415' ... '3d2c44374ee42b3003a470f3e937a2ea'
 'ac35486adb7b02598c182c2ff2e05254' '006619bbed68b000c8ba3f8725d5409e']

seller_id - unique values = 3095
['3504c0cb71d7fa48d967e0e4c94d59d9' '289cdb325fb7e7f891c38608bf9e0962'
 '4869f7a5dfa277a7dca6462dcf3b52b2' ... 'd263fa444c1504a75cbca5cc465f592a'
 'edf3fabebcc20f7463cc9c53da932ea8' 'f3862c2188522d89860c38a3ea8b550d']

shipping_limit_date - unique values = 93318
['2017-10-06 11:07:15' '2018-07-30 03:24:27' '2018-08-13 08:55:23' ...
 '2017-09-05 15:04:16' '2018-01-12 21:36:21' '2018-03-15 10:55:42']

product_category_name - unique values = 73
['utilidades_domesticas' 'perfumaria' 'automotivo' 'pet_shop' 'papelaria'
 nan 'moveis_decoracao' 'moveis_escritorio' 'ferramentas_jardim'
 'informatica_acessorios' 'cama_mesa_banho' 'brinquedos'
 'construcao_ferramentas_construcao' 'telefonia' 'beleza_saude'
 'eletronicos' 'bebes' 'cool_stuff' 'relogios_presentes' 'climatizacao'
 'esporte_lazer' 'livros_interesse_geral' 'eletroportateis' 'alimentos'
 'malas_acessorios' 'fashion_underwear_e_moda_praia' 'artigos_de_natal'
 'fashion_bolsas_e_acessorios' 'instrumentos_musicais'
 'construcao_ferramentas_iluminacao' 'livros_tecnicos'
 'construcao_ferramentas_jardim' 'eletrodomesticos' 'market_place'
 'agro_industria_e_comercio' 'artigos_de_festas' 'casa_conforto'
 'cds_dvds_musicais' 'industria_comercio_e_negocios' 'consoles_games'
 'moveis_quarto' 'construcao_ferramentas_seguranca' 'telefonia_fixa'
 'bebidas' 'moveis_cozinha_area_de_servico_jantar_e_jardim'
 'fashion_calcados' 'casa_construcao' 'audio' 'eletrodomesticos_2'
 'fashion_roupa_masculina' 'cine_foto' 'moveis_sala' 'artes'
 'alimentos_bebidas' 'tablets_impressao_imagem' 'fashion_esporte'
 'portateis_cozinha_e_preparadores_de_alimentos' 'la_cuisine' 'flores'
```

```
 'pcs' 'casa_conforto_2' 'portateis_casa_forno_e_cafe' 'dvds_blu_ray'
 'pc_gamer' 'construcao_ferramentas_ferramentas' 'fashion_roupa_feminina'
 'moveis_colchao_e_estofado' 'sinalizacao_e_seguranca' 'fraldas_higiene'
 'livros_importados' 'fashion_roupa_infanto_juvenil' 'musica'
 'artes_e_artesanato' 'seguros_e_servicos']

seller_city - unique values = 611
['maua' 'belo horizonte' 'guariba' 'mogi das cruzes' 'guarulhos'
 'sao paulo' 'atibaia' 'sao jose do rio pardo' 'itaquaquecetuba'
 'cariacica' 'porto alegre' 'ribeirao preto' 'ibitinga' 'itajai'
 'brasilia' 'jacarei' 'santo andre' 'itatiba' 'tabatinga' 'mococa'
 'curitiba' 'santa barbara d´oeste' 'praia grande' 'campinas'
 'campo mourao' 'ilicinea' 'rio do sul' 'joinville' 'rio claro' 'salto'
 'santo andre/sao paulo' 'capivari' 'bento goncalves'
 'sao bernardo do campo' 'pradopolis' 'sao goncalo' 'apucarana' 'pinhais'
 'goiania' 'bauru' 'rio de janeiro' 'colombo' 'santana de parnaiba'
 'maringa' 'franca' 'porto ferreira' 'osasco' 'bombinhas' 'borda da mata'
 'limeira' 'barueri' 'jundiai' 'blumenau' 'louveira' 'piracicaba'
 'londrina' 'presidente prudente' 'tres rios' 'hortolandia'
 'santa rita do sapucai' 'campo limpo paulista' 'cotia' 'betim'
 'sao jose do rio preto' 'santos' 'juiz de fora' 'caxias do sul' 'uba'
 'assis' 'campo largo' 'foz do iguacu' 'mogi guacu' 'arapongas' 'niteroi'
 'sao ludgero' 'lajeado' 'lauro de freitas' 'jaboticabal' 'brusque'
 'cruzeiro' 'formiga' 'garca' 'araguari' 'icara' 'sao caetano do sul'
 'tubarao' 'itau de minas' 'petropolis' nan 'tupa' "santa barbara d'oeste"
 'cosmopolis' 'teresopolis' 'sao luis' 'sao jose' 'sete lagoas'
 'penapolis' 'serra negra' 'guara' 'navegantes' 'jau' 'diadema'
 'porto belo' 'taruma' 'marilia' 'florianopolis' 'echapora' 'rolandia'
 'divinopolis' 'claudio' 'rio branco' 'catanduva' 'pedreira' 'ipaussu'
 'videira' 'sumare' 'sao carlos' 'taboao da serra' 'rio bonito'
 'araucaria' 'aruja' 'novo hamburgo' 'salvador' 'sao joaquim da barra'
 'laranjal paulista' 'sao jose dos campos' 'pilar do sul' 'sarandi'
 'indaiatuba' 'contagem' 'bebedouro' 'fronteira' 'itapecerica da serra'
 'jaguaruna' 'riberao preto' 'sao jose dos pinhais' 'cajamar' 'sao  paulo'
 'rolante' 'andradas' 'cornelio procopio' 'botucatu' 'birigui' 'mage'
 'tanabi' 'sao sebastiao' 'braganca paulista' 'dracena' 'paulinia'
 'uniao da vitoria' 'poa' 'cascavel' 'ribeirao pires' 'carazinho'
 'mirandopolis' 'vila velha' 'volta redonda' 'nilopolis' 'mogi mirim'
 'uberlandia' 'macae' 'itapetininga' 'sorocaba' 'porto seguro' 'mesquita'
 'sao bento do sul' 'sando andre' 'portoferreira'
 'vendas@creditparts.com.br' 'carapicuiba' 'alvorada' 'jaragua do sul'
 'sao roque' 'conchal' 'alfenas' 'garuva' 'viamao' 'cerqueira cesar'
 'mamanguape' 'cafelandia' 'cachoeirinha' 's jose do rio preto'
 'votorantim' 'tres coracoes' 'montenegro' 'pinhalzinho' 'cordeiropolis'
 'conselheiro lafaiete' 'patos de minas' 'eunapolis' 'jaguariuna' 'lages'
```

'araraquara' 'votuporanga' 'muqui' 'canoas' 'fazenda rio grande'
'ponta grossa' 'nova friburgo' 'monte siao' 'armacao dos buzios' 'itauna'
'centro' 'criciuma' 'vicente de carvalho' 'recife' 'concordia'
'queimados' 'sp' 'araras' 'balneario camboriu' 'nova iguacu' 'bertioga'
'brejao' 'pocos de caldas' 'novo horizonte' 'loanda' 'campo do meio'
'sao paulo - sp' 'parana' 'congonhas' 'canoinhas' 'mirassol' 'tatui'
'fortaleza' 'sao joao de meriti' 'sinop' 'lencois paulista' 'barra velha'
'anapolis' 'itu' 'itajobi' 'gama' 'aperibe' 'guaruja' 'ponte nova'
'congonhal' 'ferraz de vasconcelos' 'batatais' 'pompeia' 'ipatinga'
'presidente epitacio' 'sao joao del rei' 'dois corregos' 'suzano'
'socorro' 'embu guacu' 'aracatuba' 'viana' 'monte alto' 'toledo'
'rio verde' 'amparo' 'vassouras' 'santa maria' 'descalvado' 'americana'
'itaborai' 'ribeirao das neves' 'boituva' 'ampere'
'bonfinopolis de minas' 'vicosa' 'umuarama' 'cuiaba' 'baependi' 'vitoria'
'jussara' 'valinhos' 'floresta' 'muriae' 'jales' 'itaipulandia'
'pirituba' 'teresina' 'luziania' 'varginha' 'mandirituba' 'itapevi'
'ferraz de vasconcelos' 'campina grande' 'scao jose do rio pardo'
'bofete' 'camanducaia' 'imbituba' 'jacutinga' 'timbo' 'itaporanga'
'palhoca' 'parai' 'sao pedro' 'tiete' 'lagoa santa' 'varzea paulista'
'monteiro lobato' 'registro' 'curitibanos' 'francisco beltrao'
'carapicuiba / sao paulo' 'taubate' 'barra mansa' 'mogi das cruzes / sp'
'engenheiro coelho' 'teixeira soares' 'braco do norte' 'caucaia'
'mucambo' 'alambari' 'mogi das cruses' 'ribeirao preto / sao paulo'
'parnamirim' 'mateus leme' 'fernandopolis' 'uberaba' 'pinhalao' 'serrana'
'flores da cunha' 'saquarema' 'santo antonio de padua'
'vargem grande do sul' 'aparecida de goiania' 'angra dos reis'
'farroupilha' 'mairipora' 'pederneiras' 'tiradentes' 'ouro preto'
'paraiba do sul' 'ilheus' 'coronel fabriciano'
"arraial d'ajuda (porto seguro)" 'mineiros do tiete' 'timoteo' 'guanambi'
'torres' 'joao pessoa' 'campo magro' 'aparecida' 'campina das missoes'
'mombuca' 'tres de maio' 'arinos' 'bocaiuva do sul' 'barretos'
'andira-pr' 'ronda alta' 'nova lima' 'massaranduba' 'pitanga'
'sao pedro da aldeia' 'paranavai' 'santo angelo' 'coxim'
'franco da rocha' 'buritama' 'resende' 'campo grande' 'extrema'
'santa catarina' 'barrinha' 'marechal candido rondon' 'bady bassitt'
'fernando prestes' 'ubatuba' 'caruaru' 'sombrio' 'cambe' 'tabao da serra'
'guaratuba' 'bariri' 'santa cruz do sul' 'sao leopoldo' 'arvorezinha'
'rodeio' 'santa terezinha de itaipu' 'uruacu' 'california' 'sertanopolis'
'mandaguacu' 'mairinque' 'marica' 'paulo lopes' 'carmo do cajuru'
'brasilia df' 'sp / sp' 'lagoa da prata' 'laurentino' 'bom jardim'
'auriflama/sp' 'formosa do oeste' 'paincandu' 'campos dos goytacazes'
'nova odessa' 'rio grande' 'portao' 'venancio aires' 'campo bom' 'natal'
'jambeiro' 'duque de caxias' 'vargem grande paulista' 'cachoeira do sul'
'santa maria da serra' 'auriflama' 'manaus' 'vespasiano' 'castro'
'guanhaes' 'aracaju' 'presidente bernardes' 'joao pinheiro' 'morrinhos'

```
'floranopolis' 'dores de campos' 'entre rios do oeste' 'feira de santana'
'itapeva' 'artur nogueira' 'pelotas' 'governador valadares'
'igaracu do tiete' 'sabara' 'holambra' 'francisco morato' 'pinhais/pr'
'ipe' 'joao monlevade' 'santa barbara d oeste' 'santo antonio de posse'
'porto velho' 'xanxere' 'pedregulho' 'afonso claudio' 'indaial' 'serra'
'ouro fino' 'jaci' 'luiz alves' 'garopaba' 'miguelopolis' 'cananeia'
'04482255' 'irece' 'japira' 'nova petropolis' 'olimpia' 'sao bento'
'cataguases' 'paraiso do sul' 'montes claros' 'taio' 'clementina'
'formosa' 'rio negrinho' 'treze tilias' 'tocantins'
'sao bernardo do capo' 'cariacica / es' 'janauba' 'pato bragado' 'ibia'
'jacarei / sao paulo' 'estancia velha' 'angra dos reis rj' 'ji parana'
'leme' 'colorado' 'chapeco' 'santa rosa de viterbo' 'divisa nova' 'bahia'
'sbc/sp' 'guaratingueta' 'garulhos' 'sao paulop' 'lorena' 'cianorte'
'sao jose do rio pret' 'jaragua' 'embu das artes' 'presidente getulio'
'lambari' 'goioere' 'pato branco' 'araxa' 'domingos martins'
'rio de janeiro / rio de janeiro' 'irati' 'piracanjuba' 'xaxim'
'bandeirantes' 'laguna' 'ipira' 'santa rosa' 'imbituva' 'horizontina'
'osvaldo cruz' 'avare' 'guarapuava' 'serra redonda' 'neopolis'
'belo horizont' 'cachoeiro de itapemirim' 'guaimbe' 'alvares machado'
'gaspar' 'imbe' 'tres coroas' 'palotina' 'erechim' 'caieiras' 'imigrante'
'sao joao da boa vista' 'soledade' 'campos novos' 'messias targino'
'sao paluo' 'sbc' 'cascavael' 'pedro leopoldo' 'barro alto'
'sao paulo / sao paulo' 'bom jesus dos perdoes' 'condor'
'pedrinhas paulista' 'caratinga' 'sao paulo sp' 'igrejinha' 'mandaguari'
'cravinhos' 'rio do oeste' 'triunfo' 'aguas claras df' 'colatina'
'sao sebastiao da grama/sp' 'passos' 'picarras' 'sao  jose dos pinhais'
'castro pires' 'santa terezinha de goias' 'jarinu' 'cacador' 'guiricema'
'rio de janeiro \\rio de janeiro' 'robeirao preto' 'itabira' 'nhandeara'
'terra boa' 'monte alegre do sul' 'santa cecilia' 'sao vicente'
'barbacena/ minas gerais' 'ararangua' 'ao bernardo do campo' 'macatuba'
'rio de janeiro, rio de janeiro, brasil' 'sao jose dos pinhas'
'itirapina' 'belford roxo' 'brotas' "sao miguel d'oeste" 'maua/sao paulo'
'balenario camboriu' 'pouso alegre' 'itapema' 'nova trento'
'laranjeiras do sul' 'sao miguel do oeste' 'lages - sc' 'marapoama'
'medianeira' 'ivoti' 'varzea alegre' 'cordilheira alta'
'novo hamburgo, rio grande do sul, brasil' 'vera cruz' 'eusebio'
'vitoria de santo antao' 'sapiranga' 'paracambi' 'ribeirao pretp'
'camboriu' 'ourinhos' 'oliveira' 'ibirite' 'uruguaiana' 'tambau'
'pacatuba' 'orleans' 'jaciara' 'carmo da mata' 'marialva' 'minas gerais'
'sao francisco do sul' 'barbacena' 'bage' 'campanha'
'almirante tamandare' 'rio das pedras' 'itapui' 'abadia de goias'
'prados' 'pitangueiras' 'sao pauo' 'santo antonio da patrulha'
'juzeiro do norte' 'ipua' 'araquari' 'guaira' 'orlandia' 'são paulo'
'pirassununga']
```

```
seller_state - unique values = 23
['SP' 'MG' 'ES' 'RS' 'DF' 'PR' 'SC' 'RJ' 'GO' 'BA' nan 'MA' 'AC' 'PB' 'PE'
 'CE' 'MT' 'PI' 'RN' 'MS' 'PA' 'AM' 'SE' 'RO']

customer_unique_id - unique values = 96096
['7c396fd4830fd04220f754e42b4e5bff' 'af07308b275d755c9edb36a90c618231'
 '3a653a41f6f9fc3d2a113cf8398680e8' ... '737520a9aad80b3fbbdad19b66b37b30'
 '5097a5312c8b157bb7be58ae360ef43c' '60350aa974b26ff12caad89e55993bd6']

customer_city - unique values = 4119
['sao paulo' 'barreiras' 'vianopolis' ... 'messias targino'
 'campo do tenente' 'nova vicosa']

customer_state - unique values = 27
['SP' 'BA' 'GO' 'RN' 'PR' 'RS' 'RJ' 'MG' 'SC' 'RR' 'PE' 'TO' 'CE' 'DF'
 'SE' 'MT' 'PB' 'PA' 'RO' 'ES' 'AP' 'MS' 'MA' 'PI' 'AL' 'AC' 'AM']


df

                                order_id                       customer_id
\
0       e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
1       e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
2       e481f51cbdc54678b7cc49136f2d6af7  9ef432eb6251297304e76186b10a928d
3       53cdb2fc8bc7dce0b6741e2150273451  b0830fb4747a6c6d20dea0b8c802d7ef
4       47770eb9100c2d0c44946d9cf07ec65d  41ce2a54c0b03bf3443c3d931a367089
...                                  ...                               ...
119138  63943bddc261676b46f01ca7ac2f7bd8  1fca14ff2861355f6e5f14306ff977a7
119139  83c1379a015df1e13d02aae0204711ab  1aa71eb042121263aafbe80c1b562c9c
119140  11c177c8e97725db2631073c19f07b62  b331b74b18dc79bcdf6532d51e1637c1
119141  11c177c8e97725db2631073c19f07b62  b331b74b18dc79bcdf6532d51e1637c1
119142  66dea50a8b16d9b4dee7af250b4be1a5  edb027a75a1449115f6b43211ae02a24

       order_status order_purchase_timestamp    order_approved_at  \
0         delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
1         delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
2         delivered      2017-10-02 10:56:33  2017-10-02 11:07:15
3         delivered      2018-07-24 20:41:37  2018-07-26 03:24:27
4         delivered      2018-08-08 08:38:49  2018-08-08 08:55:23
...             ...                      ...                  ...
119138    delivered      2018-02-06 12:58:58  2018-02-06 13:10:37
119139    delivered      2017-08-27 14:46:43  2017-08-27 15:04:16
119140    delivered      2018-01-08 21:28:27  2018-01-08 21:36:21
119141    delivered      2018-01-08 21:28:27  2018-01-08 21:36:21
119142    delivered      2018-03-08 20:57:30  2018-03-09 11:20:28
```

```
       order_delivered_carrier_date order_delivered_customer_date  \
0               2017-10-04 19:55:00           2017-10-10 21:25:13
1               2017-10-04 19:55:00           2017-10-10 21:25:13
2               2017-10-04 19:55:00           2017-10-10 21:25:13
3               2018-07-26 14:31:00           2018-08-07 15:27:45
4               2018-08-08 13:50:00           2018-08-17 18:06:29
...                             ...                           ...
119138          2018-02-07 23:22:42           2018-02-28 17:37:56
119139          2017-08-28 20:52:26           2017-09-21 11:24:17
119140          2018-01-12 15:35:03           2018-01-25 23:32:54
119141          2018-01-12 15:35:03           2018-01-25 23:32:54
119142          2018-03-09 22:11:59           2018-03-16 13:08:30

       order_estimated_delivery_date                         review_id  \
0                2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
1                2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
2                2017-10-18 00:00:00  a54f0611adc9ed256b57ede6b6eb5114
3                2018-08-13 00:00:00  8d5266042046a06655c8db133d120ba5
4                2018-09-04 00:00:00  e73b67b67587f7644d5bd1a52deb1b01
...                              ...                               ...
119138           2018-03-02 00:00:00  29bb71b2760d0f876dfa178a76bc4734
119139           2017-09-27 00:00:00  371579771219f6db2d830d50805977bb
119140           2018-02-15 00:00:00  8ab6855b9fe9b812cd03a480a25058a1
119141           2018-02-15 00:00:00  8ab6855b9fe9b812cd03a480a25058a1
119142           2018-04-03 00:00:00  dc9c59b4688062c25758c2be4cafc523

        review_score  ... product_length_cm product_height_cm  \
0                4.0  ...              19.0               8.0
1                4.0  ...              19.0               8.0
2                4.0  ...              19.0               8.0
3                4.0  ...              19.0              13.0
4                5.0  ...              24.0              19.0
...              ...  ...               ...               ...
119138           4.0  ...              40.0              10.0
119139           5.0  ...              32.0              90.0
119140           2.0  ...              20.0              20.0
119141           2.0  ...              20.0              20.0
119142           5.0  ...              16.0               7.0

       product_width_cm seller_zip_code_prefix     seller_city seller_state
\
0                  13.0                 9350.0           maua           SP
1                  13.0                 9350.0           maua           SP
2                  13.0                 9350.0           maua           SP
```

```
3                 19.0                31570.0  belo horizonte           SP
4                 21.0                14840.0         guariba           SP
...                ...                    ...             ...          ...
119138            40.0                17602.0           tupa            SP
119139            22.0                 8290.0       sao paulo           SP
119140            20.0                37175.0        ilicinea           MG
119141            20.0                37175.0        ilicinea           MG
119142            15.0                14407.0          franca           SP

                  customer_unique_id  customer_zip_code_prefix  \
0       7c396fd4830fd04220f754e42b4e5bff                      3149
1       7c396fd4830fd04220f754e42b4e5bff                      3149
2       7c396fd4830fd04220f754e42b4e5bff                      3149
3       af07308b275d755c9edb36a90c618231                     47813
4       3a653a41f6f9fc3d2a113cf8398680e8                     75265
...                                  ...                       ...
119138  da62f9e57a76d978d02ab5362c509660                     11722
119139  737520a9aad80b3fbbdad19b66b37b30                     45920
119140  5097a5312c8b157bb7be58ae360ef43c                     28685
119141  5097a5312c8b157bb7be58ae360ef43c                     28685
119142  60350aa974b26ff12caad89e55993bd6                     83750

        customer_city customer_state
0           sao paulo             SP
1           sao paulo             SP
2           sao paulo             SP
3           barreiras             BA
4          vianopolis             GO
...               ...            ...
119138   praia grande             SP
119139    nova vicosa             BA
119140        japuiba             RJ
119141        japuiba             RJ
119142           lapa             PR

[119143 rows x 39 columns]
```

```
# Top 10 customers with the most orders
df_orders_per_customer = df.groupby(['customer_id', 'customer_city',
'customer_state']) \
                           .agg({'price': 'mean', 'review_score': 'mean',
'product_category_name': 'unique', 'order_item_id': 'count'}) \
                           .reset_index().rename(columns={'price':
'average_price_per_order', 'review_score': 'average_score_per_order',
'product_category_name': 'ordered_product_categories', 'order_item_id':
```

```
'orders_qty'}) \
                          .round(2) \
                          .sort_values('orders_qty',
ascending=False).head(10)
df_orders_per_customer
```

```
                           customer_id   customer_city customer_state  \
15183  270c23a11d024a44c896d1894b261a83       sao paulo             SP
7586   13aa59158da63ba0e93ec6ac2c07aacb  rio de janeiro             RJ
60184  9af2372a1e49340278e7c1ef8d749f34          cuiaba             MT
56942  92cd3ec6e2d643d4ebd0e3d6238f69e2       sao paulo             SP
52255  86cc80fef09f7f39df4b0dbce48e81cb          itaqui             RS
42897  6ee2f17e3b6c33d6a9557f280edd2925       guarulhos             SP
38590  63b964e79dee32a3587651701a2b8dbf         atibaia             SP
81894  d22f25a9fadfb1abbc2e29395b1239f4           sinop             MT
16961  2ba91e12e5e4c9f56b82b86d9031d329          suzano             SP
74119  be1c4e52bb71e0c54b11a26b8e8d59f2       sao paulo             SP

       average_price_per_order  average_score_per_order  \
15183                    36.59                      5.0
7586                     79.99                      5.0
60184                   392.55                      1.0
56942                    49.99                      5.0
52255                   121.44                      1.0
42897                   189.90                      4.0
38590                   412.00                      5.0
81894                    14.99                      3.0
16961                    99.90                      1.0
74119                    65.01                      1.0

                     ordered_product_categories  orders_qty
15183    [cama_mesa_banho, utilidades_domesticas]          63
7586                          [moveis_escritorio]          38
60184                        [ferramentas_jardim]          29
56942                           [cama_mesa_banho]          26
52255  [informatica_acessorios, malas_acessorios]          24
42897                        [ferramentas_jardim]          24
38590               [agro_industria_e_comercio]          24
81894                  [informatica_acessorios]          24
16961                               [perfumaria]          24
74119                           [cama_mesa_banho]          22
```

```
# Top 10 most ordered products
df_orders_per_product = df.groupby(['product_id', 'product_category_name'])
\
                          .agg({'price': 'mean', 'review_score': 'mean',
```

```
'order_item_id': 'count'}) \
                          .reset_index().rename(columns={'price':
'average_price_per_order', 'review_score': 'average_score_per_order',
'order_item_id': 'orders_qty'}) \
                          .round(2) \
                          .sort_values('orders_qty',
ascending=False).head(10)
df_orders_per_product
```

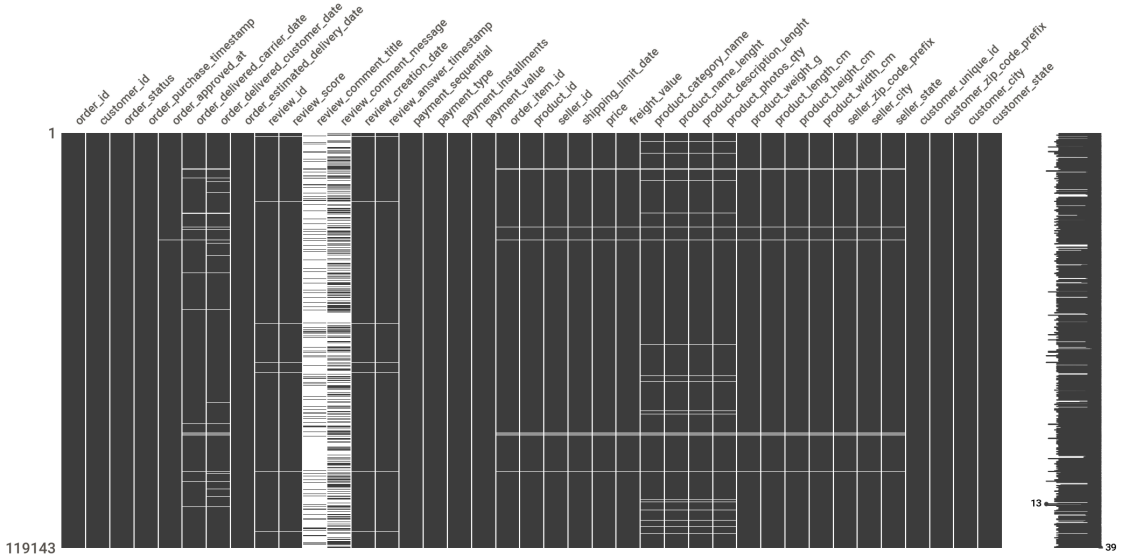
```
                            product_id   product_category_name  \
21724  aca2eb7d00ea1a7b8ebd4e68314663af        moveis_decoracao
19394  99a4788cb24856965c36a24e339b6058         cama_mesa_banho
8456   422879e10f46682990de24d770e7f83d      ferramentas_jardim
7231   389d119b48cf3043d311335e499d9c6b      ferramentas_jardim
6950   368c6c730842d78016ad823897a372db      ferramentas_jardim
10638  53759a2ecddad2bb87a079a1f1519f73      ferramentas_jardim
26545  d1c427060a0f73f6b889a5c7c61f2ac4  informatica_acessorios
10665  53b36df67ebb7c41585e8d54d6772e08      relogios_presentes
2743   154e7e31ebfa092203795c972e5804a6            beleza_saude
7905   3dd2a17168ec895c781a9191c1e95ad7  informatica_acessorios

       average_price_per_order  average_score_per_order  orders_qty
21724                    71.36                     4.02         536
19394                    88.21                     3.91         528
8456                     54.83                     3.93         508
7231                     54.63                     4.11         406
6950                     54.27                     3.91         398
10638                    54.71                     3.88         391
26545                   137.65                     4.10         357
10665                   116.69                     4.20         327
2743                     22.53                     4.32         295
7905                    149.94                     4.21         278
```

```
# EDA dashboard
analysis = sv.analyze(df)
analysis.show_notebook()
```

```
{"model_id":"f51e1b40a5b2414fb068a2db4654373a","version_major":2,"version_m
inor":0}
```

```
<IPython.core.display.HTML object>
```

```
# Data missing patterns
missingno.matrix(df)
plt.show()
```

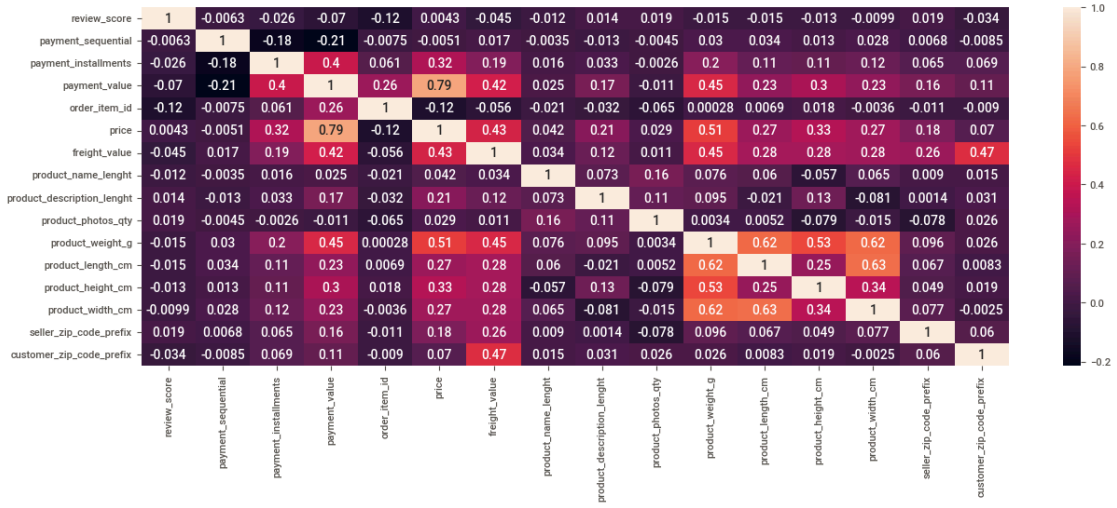



```
# Checking for missing values
df.isnull().sum()
```

```
order_id                            0
customer_id                         0
order_status                        0
order_purchase_timestamp            0
order_approved_at                 177
order_delivered_carrier_date     2086
order_delivered_customer_date    3421
order_estimated_delivery_date       0
review_id                         997
review_score                      997
review_comment_title           105154
review_comment_message          68898
review_creation_date              997
review_answer_timestamp           997
payment_sequential                  3
payment_type                        3
payment_installments                3
payment_value                       3
order_item_id                     833
product_id                        833
seller_id                         833
shipping_limit_date               833
price                             833
freight_value                     833
product_category_name            2542
product_name_lenght              2542
```

```
product_description_lenght          2542
product_photos_qty                  2542
product_weight_g                     853
product_length_cm                    853
product_height_cm                    853
product_width_cm                     853
seller_zip_code_prefix               833
seller_city                          833
seller_state                         833
customer_unique_id                     0
customer_zip_code_prefix               0
customer_city                          0
customer_state                         0
dtype: int64
```

```
# Correlation matrix
plt.figure(figsize=(15, 5))
sns.heatmap(df.corr(method='spearman', numeric_only=True), annot=True)
plt.show()
```




```
# Exporting merged dataframe
df.to_parquet(f'raw.parquet', index=False)
```

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: **30 de nov. de 2023**

Participantes da Entrega **[matriculados em Residência em IA]**:

Állan Christoffer Pereira Silva

Entrega: **[descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]**

Esta entrega consistiu na segunda etapa de desenvolvimento relacionada à organização dos dados de acordo com o nível de valor agregado, além da definição do fluxo de pré-processamento desses dados para a aplicação de algoritmos de ML. Vale relembrar que o objetivo da pesquisa aplicada é a construção de um sistema de recomendação a partir de dados de e-commerces brasileiros e a implantação deste sistema seguindo os princípios de MLOps.

Os requisitos básicos para a entrega eram:

- Elaboração da arquitetura de dados (armazenamento bronze, silver e gold).
- Pré-processamento dos dados.

Os produtos gerados para esta entrega estão descritos a seguir:

- Documentação acerca da arquitetura projetada para a aplicação escolhida: https://docs.google.com/document/d/1cPn4_dbWwUiqKpcKB7jHICjFXWreU2QhZcEzhlwAQ7M/edit?usp=sharing. A arquitetura Delta Lake foi escolhida inicialmente como base para a construção da arquitetura idealizada para o desenvolvimento e a implantação do sistema de recomendação de forma a seguir os princípios da cultura MLOps. O documento traz a arquitetura projetada e os respectivos serviços da plataforma Google Cloud Platform que vão ser utilizados.
- Scripts realizados para preparação dos dados e formação da camada silver: https://colab.research.google.com/drive/1Hlyb3QfSA1iVisBpZQ6m5ZcekKCruVIA?usp=sharing. Os pré-processamentos realizados foram, de forma geral, a conversão dos tipos de dados de algumas colunas (string para data, por exemplo), remoção de quebras de linhas na coluna de comentários das avaliações, tratamento de dados faltantes (via exclusão de registros ou inserção de valores), além da seleção das colunas mais relevantes para o contexto de recomendações (características dos usuários e dos pedidos).

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega do dia 07/12/2023, estão planejadas as seguintes atividades:

- Configuração dos scripts de carregamento dos dados e pré-processamento no serviço Dataproc.
- Exploração do serviço Cloud Composer para definição e gerenciamento do fluxo de trabalho.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

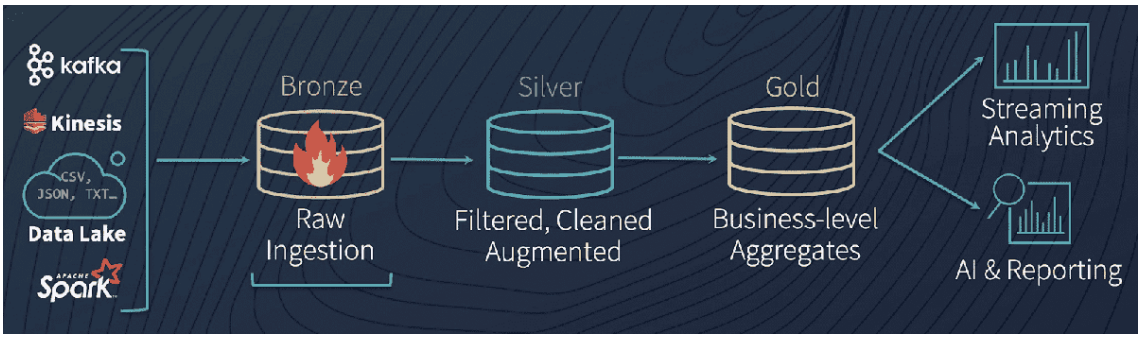
Arquitetura de Dados v1

Autor
Állan Christoffer Pereira Silva

Introdução
Este documento tem como objetivo principal documentar a estruturação da arquitetura de dados referente à pesquisa aplicada que está sendo desenvolvida envolvendo sistema de recomendação e princípios de MLOps.

A arquitetura que será utilizada como base para o desenvolvimento deste trabalho pode ser vista a seguir.

Figura 1. Arquitetura Delta Lake



Fonte:
https://www.databricks.com/blog/2020/11/20/delta-vs-lambda-why-simplicity-trumps-complexity-for-data-pipelines.html

A arquitetura Delta Lake é uma abordagem para construir um data lakehouse, que combina as características de um data warehouse e um data lake. Ela é construída em cima de um sistema de arquivos distribuídos (como HDFS ou S3) e fornece um formato de armazenamento que oferece ACID transactions, escalabilidade, e integração com ferramentas de big data. Ela é composta por algumas camadas de organização de dados de acordo com o nível de valor agregado. São elas:

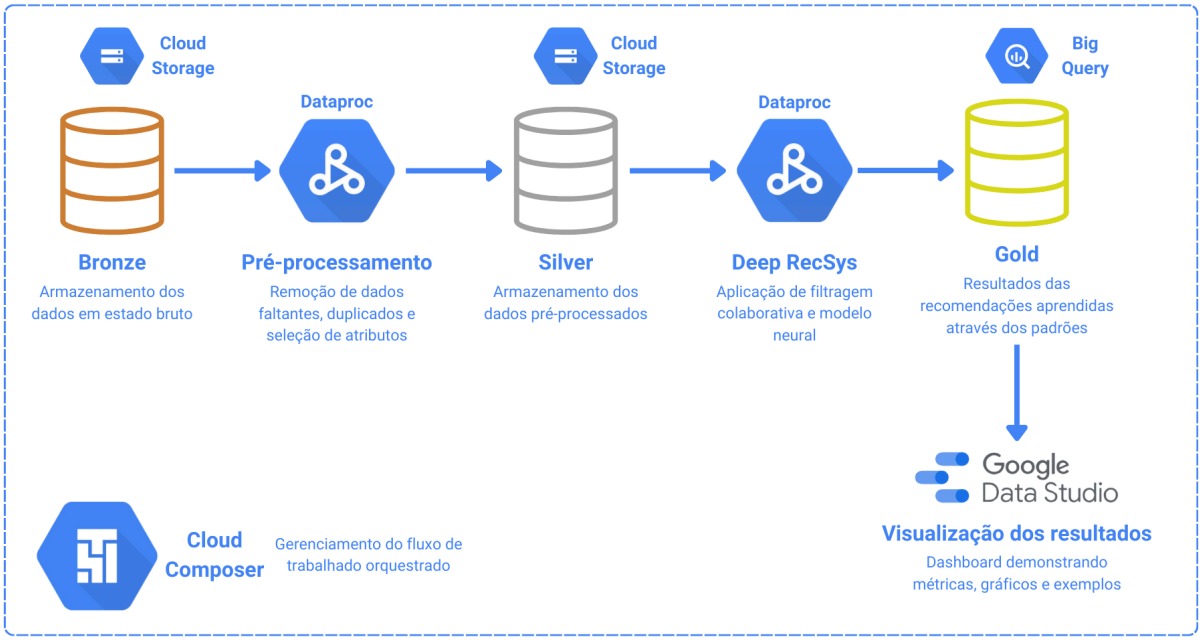
- **Bronze:** armazena dados brutos em seu formato original; usada para ingestão de dados de diversas fontes.
- **Silver:** contém dados que foram limpos, transformados e enriquecidos; usada para unificar, padronizar e melhorar a qualidade dos dados.
- **Gold:** armazena dados agregados e otimizados para análises e relatórios de negócios; usada para insights e tomada de decisão baseada em dados.

A arquitetura Delta Lake oferece gerenciamento robusto de transações ACID, escalabilidade, integração fácil com ferramentas de big data, governança de dados aprimorada e otimização de consultas, tudo dentro de uma estrutura de camadas de dados organizada para eficiência e clareza.
No escopo de sistemas de recomendação, essa arquitetura é essencial dentro dos princípios de MLOps, pois proporciona transações ACID para integridade de dados, suporta versionamento para reprodutibilidade, oferece esquemas evolutivos para adaptabilidade, e assegura otimização de consultas para eficiência, tudo isso em um ambiente escalável que se integra bem com ferramentas de machine learning, essencial para o rápido desenvolvimento e manutenção de modelos de recomendação robustos.

Arquitetura de dados projetada

Para este trabalho, a arquitetura projetada vai ter como enfoque principal a organização eficiente dos dados e dos fluxos de processamento. A plataforma em nuvem escolhida foi o Google Cloud Platform (GCP) tendo em vista a sua grande disponibilidade de serviços e o seu acervo de documentações sobre a utilização de cada serviço.

Figura 2. Arquitetura projetada



Fonte: próprio autor.

A arquitetura ilustrada acima pode ser detalhada da seguinte forma:

- **Cloud Storage - Bronze:** Armazena os dados brutos coletados de diferentes fontes. Esta é a camada de ingestão onde os dados ainda não foram processados ou filtrados.

- **Dataproc - Pré-processamento:** Utiliza o serviço de processamento de dados Dataproc para realizar o pré-processamento dos dados brutos. Isso inclui a remoção de dados faltantes, duplicados e a seleção de atributos relevantes para análise.
- **Cloud Storage - Silver:** Armazena os dados que foram pré-processados pelo Dataproc. Estes dados são mais limpos e estruturados do que os dados brutos e estão prontos para análises mais aprofundadas.
- **Dataproc - Deep RecSys:** Realiza processamento avançado usando algoritmos de sistema de recomendação, como filtragem colaborativa e modelos neurais, para gerar recomendações personalizadas.
- **BigQuery - Gold:** Os resultados das recomendações são armazenados em BigQuery, o data warehouse do Google, onde podem ser analisados e consultados rapidamente. Esta é a camada de valor agregado onde os dados são transformados em insights.
- **Google Data Studio:** Os resultados e insights obtidos são visualizados no Google Data Studio através de dashboards que apresentam métricas, gráficos e exemplos, facilitando a interpretação e a tomada de decisão baseada em dados.
- **Cloud Composer:** Orquestra todo o fluxo de trabalho, garantindo que cada etapa seja executada na sequência correta e gerenciando as dependências entre as tarefas.

Jupyter Notebook: Pré-processamento

Data Preprocessing Scripts

Author: Állan Christoffer Pereira Silva

Setup

```
# Importing frameworks
import pandas as pd

# Reading raw dataset
df = pd.read_csv('raw.csv')
```

Data Preprocessing

```
# Converting numeric columns to object columns
categ_cols = ['order_item_id', 'payment_sequential',
'seller_zip_code_prefix',
              'customer_zip_code_prefix']

for col in categ_cols:
    df[col] = df[col].astype(str)

df[categ_cols].dtypes
```

```
order_item_id               object
payment_sequential          object
seller_zip_code_prefix      object
customer_zip_code_prefix    object
dtype: object
```

```
# Converting textual columns to datetime columns
datetime_cols = ['order_purchase_timestamp', 'order_approved_at',
'order_delivered_carrier_date',
                 'order_delivered_customer_date',
'order_estimated_delivery_date', 'review_creation_date',
                 'review_answer_timestamp', 'shipping_limit_date']

for col in datetime_cols:
    df[col] = pd.to_datetime(df[col])

df[datetime_cols].dtypes
```

```
order_purchase_timestamp         datetime64[ns]
order_approved_at                datetime64[ns]
order_delivered_carrier_date     datetime64[ns]
order_delivered_customer_date    datetime64[ns]
```

```
order_estimated_delivery_date    datetime64[ns]
review_creation_date             datetime64[ns]
review_answer_timestamp          datetime64[ns]
shipping_limit_date              datetime64[ns]
dtype: object
```

```
# Removing line break in textual columns
print(f'BEFORE = {df.iloc[76:77, :]["review_comment_message"].values[0]}')
for col in df.select_dtypes(include=['object']).columns:
    df[col] = df[col].str.replace('\r', '', regex=False)
    df[col] = df[col].str.replace('\n', ' ', regex=False)
print(f'AFTER = {df.iloc[76:77, :]["review_comment_message"].values[0]}')
```

```
BEFORE = Pensei que a cinta seria mais larga.
É muito estreita para cargas pesadas
AFTER = Pensei que a cinta seria mais larga. É muito estreita para cargas
pesadas
```

```
# Handling null data through deletion
deletion_nan_columns = ['order_approved_at',
'order_delivered_carrier_date',
       'order_delivered_customer_date', 'review_creation_date',
       'review_answer_timestamp', 'shipping_limit_date',
'payment_sequential',
       'payment_type', 'payment_installments', 'payment_value',
'order_item_id',
       'product_id', 'seller_id', 'price', 'freight_value',
'product_category_name',
       'product_name_lenght', 'product_description_lenght',
'product_photos_qty',
       'product_weight_g', 'product_length_cm', 'product_height_cm',
'product_width_cm',
       'seller_zip_code_prefix', 'seller_city', 'seller_state',
'review_id', 'review_score',
       'review_creation_date', 'review_answer_timestamp']
df = df.dropna(subset=deletion_nan_columns, axis=0)
df[deletion_nan_columns].isnull().sum()
```

```
order_approved_at                0
order_delivered_carrier_date     0
order_delivered_customer_date    0
review_creation_date             0
review_answer_timestamp          0
shipping_limit_date              0
payment_sequential               0
payment_type                     0
```

```
payment_installments            0
payment_value                   0
order_item_id                   0
product_id                      0
seller_id                       0
price                           0
freight_value                   0
product_category_name           0
product_name_lenght             0
product_description_lenght      0
product_photos_qty              0
product_weight_g                0
product_length_cm               0
product_height_cm               0
product_width_cm                0
seller_zip_code_prefix          0
seller_city                     0
seller_state                    0
review_id                       0
review_score                    0
review_creation_date            0
review_answer_timestamp         0
dtype: int64
```

```
# Handling null data through insertion
df.loc[:, 'review_comment_title'] = df.loc[:,
'review_comment_title'].fillna('Sem título')
df.loc[:, 'review_comment_message'] = df.loc[:,
'review_comment_message'].fillna('Sem comentário')
df[['review_comment_title', 'review_comment_message']].isnull().sum()
```

```
review_comment_title      0
review_comment_message    0
dtype: int64
```

```
# Selecting best-applicable columns for recommendation
df = df[['customer_id', 'product_id', 'product_category_name',
        'product_name_lenght', 'product_description_lenght',
'product_photos_qty',
        'product_weight_g', 'product_weight_g', 'product_height_cm',
'product_height_cm',
        'seller_state', 'customer_state', 'payment_value']]

# Verifying merged dataframe dimensions after preprocessing
df.shape
```

```
(113216, 13)
```

```
# Verifying data types after preprocessing
df.dtypes
```

```
customer_id                    object
product_id                     object
product_category_name          object
product_name_lenght           float64
product_description_lenght    float64
product_photos_qty            float64
product_weight_g              float64
product_weight_g              float64
product_height_cm             float64
product_height_cm             float64
seller_state                   object
customer_state                 object
payment_value                 float64
dtype: object
```

```
# Checking for missing values after preprocessing
df.isnull().sum()
```

```
customer_id                   0
product_id                    0
product_category_name         0
product_name_lenght           0
product_description_lenght    0
product_photos_qty            0
product_weight_g              0
product_weight_g              0
product_height_cm             0
product_height_cm             0
seller_state                  0
customer_state                0
payment_value                 0
dtype: int64
```

```
# Exporting preprocessed dataframe
df.to_csv('preprocessed.csv', index=False)
```

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Esta entrega consistiu na reformulação da arquitetura de dados previamente elaborada, além da implementação dos scripts de carregamento dos dados e de pré-processamento nos serviços correspondentes do Google Cloud Platform.

Os requisitos básicos para a entrega eram:

- Configurar os scripts de carregamento dos dados e pré-processamento no serviço Dataproc.
- Explorar o serviço Cloud Composer para definição e gerenciamento do fluxo de trabalho.

Os produtos gerados para esta entrega estão descritos a seguir:

- Documentação acerca da reformulação da arquitetura projetada: https://docs.google.com/document/d/1zWKTvRJUOfavGSF2LlBeVwIbqzAJY_TbjcRuzUiU1Pc/edit?usp=sharing. De forma geral, a principal mudança foi com relação ao serviço de processamento de dados, que anteriormente foi planejado para ser o Dataproc, mas agora passa a ser o Vertex AI. A justificativa para essa mudança foi a grande variedade de ferramentas oferecidas pelo Vertex AI especificamente para a construção de aplicações de Machine Learning. Além disso, o serviço do Vertex AI Workbench possibilita carregar os scripts de carregamento dos dados e de pré-processamento em formato de Jupyter Notebook e oferece uma melhor integração com outros serviços (Cloud Storage, BigQuery, entre outros) do que quando comparado ao Dataproc.
- Sobre a exploração do serviço Cloud Composer, foi decidido que o melhor caminho era implementar, primeiramente, todo o pipeline de preparação dos dados e de treinamento do modelo de recomendação e somente após a consolidação dessas etapas, partir para a configuração do gerenciamento dos fluxos de trabalho.
- Por fim, foi testada uma abordagem inicial de filtragem colaborativa com o uso de Autoencoder. Este foi somente um experimento preliminar para avaliar a possibilidade de uso com mais atributos envolvidos. O teste construído pode ser visualizado neste link:

https://colab.research.google.com/drive/12M00l9bp37kSAg0xOn5LNV8BEglpz0-4?usp=sharing.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega do dia 14/12/2023, estão planejadas as seguintes atividades:

- Realização da filtragem colaborativa com Autoencoder envolvendo os outros atributos do dataset.
- Implementação da camada gold com os resultados das recomendações e visualizações desses resultados.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Em análise!

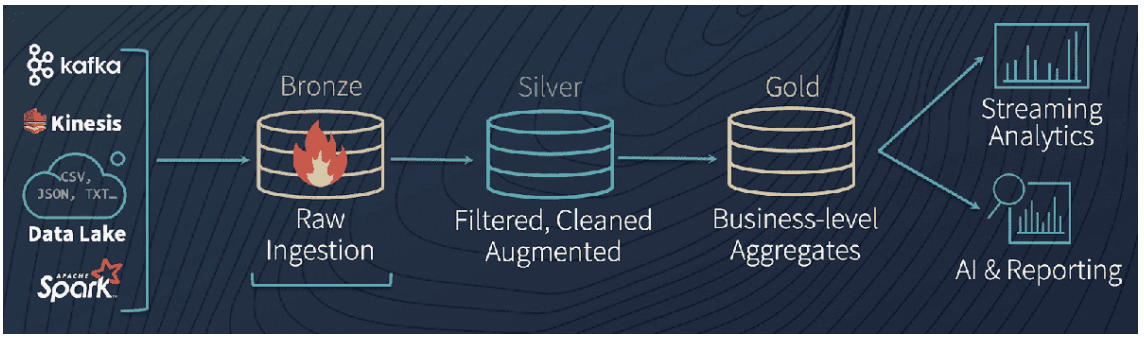
Arquitetura de Dados v2

Autor
Állan Christoffer Pereira Silva

Introdução
Este documento tem como objetivo principal documentar a estruturação da arquitetura de dados referente à pesquisa aplicada que está sendo desenvolvida envolvendo sistema de recomendação e princípios de MLOps.

A arquitetura que será utilizada como base para o desenvolvimento deste trabalho pode ser vista a seguir.

Figura 1. Arquitetura Delta Lake



Fonte:
https://www.databricks.com/blog/2020/11/20/delta-vs-lambda-why-simplicity-trumps-complexity-for-data-pipelines.html

A arquitetura Delta Lake é uma abordagem para construir um data lakehouse, que combina as características de um data warehouse e um data lake. Ela é construída em cima de um sistema de arquivos distribuídos (como HDFS ou S3) e fornece um formato de armazenamento que oferece ACID transactions, escalabilidade, e integração com ferramentas de big data. Ela é composta por algumas camadas de organização de dados de acordo com o nível de valor agregado. São elas:

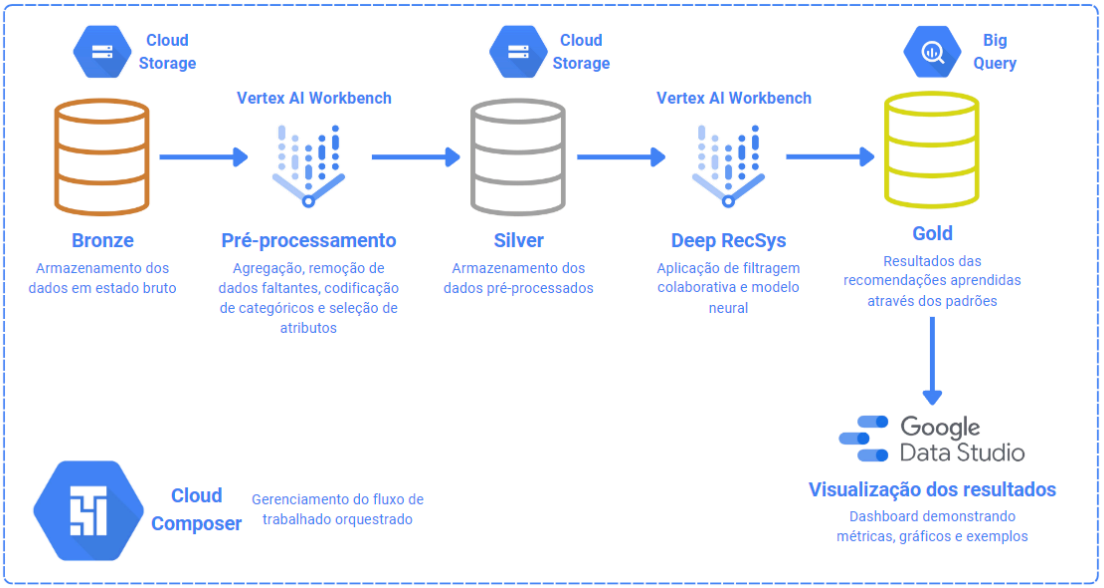
- **Bronze:** armazena dados brutos em seu formato original; usada para ingestão de dados de diversas fontes.
- **Silver:** contém dados que foram limpos, transformados e enriquecidos; usada para unificar, padronizar e melhorar a qualidade dos dados.
- **Gold:** armazena dados agregados e otimizados para análises e relatórios de negócios; usada para insights e tomada de decisão baseada em dados.

A arquitetura Delta Lake oferece gerenciamento robusto de transações ACID, escalabilidade, integração fácil com ferramentas de big data, governança de dados aprimorada e otimização de consultas, tudo dentro de uma estrutura de camadas de dados organizada para eficiência e clareza.
No escopo de sistemas de recomendação, essa arquitetura é essencial dentro dos princípios de MLOps, pois proporciona transações ACID para integridade de dados, suporta versionamento para reprodutibilidade, oferece esquemas evolutivos para adaptabilidade, e assegura otimização de consultas para eficiência, tudo isso em um ambiente escalável que se integra bem com ferramentas de machine learning, essencial para o rápido desenvolvimento e manutenção de modelos de recomendação robustos.

Arquitetura de dados projetada

Para este trabalho, a arquitetura projetada vai ter como enfoque principal a organização eficiente dos dados e dos fluxos de processamento. A plataforma em nuvem escolhida foi o Google Cloud Platform (GCP) tendo em vista a sua grande disponibilidade de serviços e o seu acervo de documentações sobre a utilização de cada serviço.

Figura 2. Arquitetura projetada



Fonte: próprio autor. Link: https://www.canva.com/design/DAF2Ipv-WV4/4gqsVxJzJGEvc9AEsfvBJA/edit?utm_content=DAF2Ipv-WV4&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

A arquitetura ilustrada acima pode ser detalhada da seguinte forma:

- **Cloud Storage - Bronze:** Armazena os dados brutos coletados de diferentes fontes. Esta é a camada de ingestão onde os dados ainda não foram processados ou

filtrados. As fontes de dados que serão utilizadas assim como a suas respectivas descrições estão todas detalhadas no Dicionário de Dados.

- **Dataproc - Pré-processamento:** Utiliza o serviço de processamento de dados Dataproc para realizar o pré-processamento dos dados brutos. Isso inclui a remoção de dados faltantes, duplicados e a seleção de atributos relevantes para análise. Além disso, nas colunas categóricas será realizado um processamento do tipo binary encoding (boa eficiência com muitas categorias). Já nos atributos textuais, será aplicada a transformação TF-IDF para representação das informações. Dentre os atributos selecionados para a realização do treinamento do modelo de recomendação estão:
    1. customer_id e product_id: para realizar a identificação de cada par de interação cliente-produto.
    2. product_category_name: esse atributo pode conter padrões de consumo do tipo cross sell, isto é, categorias que são frequentemente compradas juntas.
    3. product_name_length, product_description_length e product_photos_qty: essas informações quantitativas sobre os textos descritivos e as imagens podem revelar padrões de criação de anúncios de produtos. Esses padrões podem indicar grupos de produtos com anúncios semelhantes.
    4. product_weight_g, product_length_cm, product_height_cm e product_width_cm: as características físicas dos produtos podem revelar padrões que permitem separar produtos de pequeno, médio e grande porte.
    5. seller_state, seller_city, customer_city e customer_state: essas características geográficas podem revelar padrões de consumo parecidos entre clientes de regiões próximas ou de vendedores próximos.
    6. review_comment_title e review_comment_message: esses atributos são fundamentais pois contém informações sobre os comentários realizados por cada cliente sobre suas compras. Padrões de consumo podem ser identificados por meio do histórico de avaliações.
    7. review_score: esse é o atributo alvo o qual será previsto para determinar uma possível avaliação de um cliente a um produto que ele ainda não comprou. A partir desse score previsto, a recomendação será feita.
- **Cloud Storage - Silver:** Armazena os dados que foram pré-processados pelo Dataproc. Estes dados são mais limpos e estruturados do que os dados brutos e estão prontos para análises mais aprofundadas. Como serviço de armazenamento das versões dos dados, o Cloud Storage foi escolhido por conta do custo mais baixo, da possibilidade de armazenamento de formatos diferentes (csv, json, parquet, entre outros), além da facilidade em conectar com outros serviços como Dataproc e VertexAI.
- **Dataproc - Deep RecSys:** Realiza processamento avançado usando algoritmos de sistema de recomendação, como filtragem colaborativa e modelos neurais, para gerar recomendações personalizadas.

- **BigQuery - Gold:** Os resultados das recomendações são armazenados em BigQuery, o data warehouse do Google, onde podem ser analisados e consultados rapidamente. Esta é a camada de valor agregado onde os dados são transformados em insights.
- **Google Data Studio:** Os resultados e insights obtidos são visualizados no Google Data Studio através de dashboards que apresentam métricas, gráficos e exemplos, facilitando a interpretação e a tomada de decisão baseada em dados.
- **Cloud Composer:** Orquestra todo o fluxo de trabalho, garantindo que cada etapa seja executada na sequência correta e gerenciando as dependências entre as tarefas.

Jupyter Notebook: Deep RecSys

Collaborative Filtering with Autoencoder

Author: Állan Christoffer Pereira Silva

Setup

```
# Importing frameworks
import pandas as pd
from keras.optimizers import Adam
from keras.layers import Input, Dense, Dropout
from keras.models import Model

# Reading preprocessed dataset
df = pd.read_csv('preprocessed.csv')
```

Predictive Modeling

```
# Filtering by customers with a history of at least 5 orders
df_orders_per_customer = df.groupby(['customer_id', 'customer_city',
'customer_state']) \
                           .agg({'price': 'mean', 'review_score': 'mean',
'product_category_name': 'unique', 'order_item_id': 'count'}) \
                           .reset_index().rename(columns={'price':
'average_price_per_order', 'review_score': 'average_score_per_order',
'product_category_name': 'ordered_product_categories', 'order_item_id':
'orders_qty'}) \
                           .round(2)

df_orders_per_customer =
df_orders_per_customer.loc[df_orders_per_customer['orders_qty'] >= 5]

print(f'Dimensions before filtering = {df.shape}')

customer_ids = df_orders_per_customer['customer_id'].values
df = df.loc[df['customer_id'].isin(customer_ids)]

print(f'Dimensions after filtering = {df.shape}')
```

```
Dimensions before filtering = (119143, 39)
Dimensions after filtering = (4930, 39)
```

```
# Creating interaction matrix
df = df[['customer_id', 'product_id', 'product_category_name',
'review_score']]
customers_products_matrix = df.pivot_table(index='customer_id',
```

```
                                   columns='product_id',
                                   values='review_score',
                                   aggfunc='mean').fillna(0)
customers_products_matrix.head()
```

```
product_id                    00250175f79f584c14ab5cecd80553cd  \
customer_id
00426311a53f3c052943c88b692a3be2                               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                               0.0
0097c5abeb126a90646370f4a1cf3d93                               0.0
00f394e6fc446865ac4097b6db69ef4a                               0.0
00f6217307f712298d8e47215f0bf2ad                               0.0

product_id                    002af88741ba70c7b5cf4e4a0ad7ef85  \
customer_id
00426311a53f3c052943c88b692a3be2                               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                               0.0
0097c5abeb126a90646370f4a1cf3d93                               0.0
00f394e6fc446865ac4097b6db69ef4a                               0.0
00f6217307f712298d8e47215f0bf2ad                               0.0

product_id                    00ba6d766f0b1d7b78a5ce3e1e033263  \
customer_id
00426311a53f3c052943c88b692a3be2                               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                               0.0
0097c5abeb126a90646370f4a1cf3d93                               0.0
00f394e6fc446865ac4097b6db69ef4a                               0.0
00f6217307f712298d8e47215f0bf2ad                               0.0

product_id                    01422266d7a3131403364787ef9dab11  \
customer_id
00426311a53f3c052943c88b692a3be2                               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                               0.0
0097c5abeb126a90646370f4a1cf3d93                               0.0
00f394e6fc446865ac4097b6db69ef4a                               0.0
00f6217307f712298d8e47215f0bf2ad                               0.0

product_id                    0152f69b6cf919bcdaf117aa8c43e5a2  \
customer_id
00426311a53f3c052943c88b692a3be2                               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                               0.0
0097c5abeb126a90646370f4a1cf3d93                               0.0
00f394e6fc446865ac4097b6db69ef4a                               0.0
00f6217307f712298d8e47215f0bf2ad                               0.0
```

```
product_id                      01ff1ff8aa5dec93e9938b989393a4ca  \
customer_id
00426311a53f3c052943c88b692a3be2                              0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0
0097c5abeb126a90646370f4a1cf3d93                              0.0
00f394e6fc446865ac4097b6db69ef4a                              0.0
00f6217307f712298d8e47215f0bf2ad                              0.0

product_id                      0302c3fcf5e2d9526e243db50d30d5e3  \
customer_id
00426311a53f3c052943c88b692a3be2                              0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0
0097c5abeb126a90646370f4a1cf3d93                              0.0
00f394e6fc446865ac4097b6db69ef4a                              0.0
00f6217307f712298d8e47215f0bf2ad                              0.0

product_id                      034abfb9b758233fd393bd361d4ec599  \
customer_id
00426311a53f3c052943c88b692a3be2                              0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0
0097c5abeb126a90646370f4a1cf3d93                              0.0
00f394e6fc446865ac4097b6db69ef4a                              0.0
00f6217307f712298d8e47215f0bf2ad                              0.0

product_id                      0364c36f8e845e4d309c0a3accc04b1c  \
customer_id
00426311a53f3c052943c88b692a3be2                              0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0
0097c5abeb126a90646370f4a1cf3d93                              0.0
00f394e6fc446865ac4097b6db69ef4a                              0.0
00f6217307f712298d8e47215f0bf2ad                              0.0

product_id                      03d817e5e392e78674ed0bd8195f9159  ...  \
customer_id                                                       ...
00426311a53f3c052943c88b692a3be2                              0.0 ...
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0 ...
0097c5abeb126a90646370f4a1cf3d93                              0.0 ...
00f394e6fc446865ac4097b6db69ef4a                              0.0 ...
00f6217307f712298d8e47215f0bf2ad                              0.0 ...

product_id                      fd471a043ee8b8dd27f4086495e0724c  \
customer_id
00426311a53f3c052943c88b692a3be2                              0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                              0.0
0097c5abeb126a90646370f4a1cf3d93                              0.0
```

```
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       fd5c5a67cd369732c2ac20dbd574d1d4  \
customer_id
00426311a53f3c052943c88b692a3be2                                  0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                  0.0
0097c5abeb126a90646370f4a1cf3d93                                  0.0
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       fdba6f37e0258dbca54eb0ed8b293ae0  \
customer_id
00426311a53f3c052943c88b692a3be2                                  0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                  0.0
0097c5abeb126a90646370f4a1cf3d93                                  0.0
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       fdbfd4cec6db6e89f22d9aa2c4839eed  \
customer_id
00426311a53f3c052943c88b692a3be2                                  0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                  0.0
0097c5abeb126a90646370f4a1cf3d93                                  0.0
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       fdff6fbfaeefd11afe77be6f416e9c4e  \
customer_id
00426311a53f3c052943c88b692a3be2                                  0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                  0.0
0097c5abeb126a90646370f4a1cf3d93                                  0.0
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       fe83af233315b04d9093c7edbcf789dd  \
customer_id
00426311a53f3c052943c88b692a3be2                                  0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                  0.0
0097c5abeb126a90646370f4a1cf3d93                                  0.0
00f394e6fc446865ac4097b6db69ef4a                                  0.0
00f6217307f712298d8e47215f0bf2ad                                  0.0

product_id                       feb4ade62e32b8d74c6f69f635057964  \
customer_id
```

```
00426311a53f3c052943c88b692a3be2                                    0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                    0.0
0097c5abeb126a90646370f4a1cf3d93                                    0.0
00f394e6fc446865ac4097b6db69ef4a                                    0.0
00f6217307f712298d8e47215f0bf2ad                                    0.0

product_id                      ff2c1ec09b1bb340e84f0d6b21cc7dbb  \
customer_id
00426311a53f3c052943c88b692a3be2                                    0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                    0.0
0097c5abeb126a90646370f4a1cf3d93                                    0.0
00f394e6fc446865ac4097b6db69ef4a                                    0.0
00f6217307f712298d8e47215f0bf2ad                                    0.0

product_id                      ff6caf9340512b8bf6d2a2a6df032cfa  \
customer_id
00426311a53f3c052943c88b692a3be2                                    0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                    0.0
0097c5abeb126a90646370f4a1cf3d93                                    0.0
00f394e6fc446865ac4097b6db69ef4a                                    0.0
00f6217307f712298d8e47215f0bf2ad                                    0.0

product_id                      ff95ac47246ef13e48712ea1ff8df0d9
customer_id
00426311a53f3c052943c88b692a3be2                                    0.0
007e99fec9d53dfa4e5d8be9c2b36ca7                                    0.0
0097c5abeb126a90646370f4a1cf3d93                                    0.0
00f394e6fc446865ac4097b6db69ef4a                                    0.0
00f6217307f712298d8e47215f0bf2ad                                    0.0

[5 rows x 907 columns]
```

```
# Defining autoencoder architecture
def autoEncoder(X):
  '''
  Autoencoder for Collaborative Filter Model
  '''

  # Input
  input_layer = Input(shape=(X.shape[1],), name='UserScore')

  # Encoder
  # ------------------------------
  enc = Dense(512, activation='selu', name='EncLayer1')(input_layer)
```

```
  # Latent Space
  # ----------------------------
  lat_space = Dense(256, activation='selu', name='LatentSpace')(enc)
  lat_space = Dropout(0.8, name='Dropout')(lat_space) # Dropout

  # Decoder
  # ----------------------------
  dec = Dense(512, activation='selu', name='DecLayer1')(lat_space)

  # Output
  output_layer = Dense(X.shape[1], activation='linear',
name='UserScorePred')(dec)

  # This model maps an input to its reconstruction
  model = Model(input_layer, output_layer)

  return model

# Input
X = customers_products_matrix.values
y = customers_products_matrix.values

# Build model
model = autoEncoder(X)
model.compile(optimizer = Adam(learning_rate=0.0001), loss='mse')
```

Figure 1 - Autoencoder Architecture

Fonte:
https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-2-filtragem-colaborativa-com-autoencoders-347ba7d53bae

```
# Train
hist = model.fit(x=X, y=y,
                 epochs=50,
                 batch_size=64,
                 shuffle=True,
                 validation_split=0.1)

Epoch 1/50
10/10 [==============================] - 1s 16ms/step - loss: 0.3004 -
val_loss: 0.0741
Epoch 2/50
10/10 [==============================] - 0s 8ms/step - loss: 0.2804 -
val_loss: 0.0702
```

```
Epoch 3/50
10/10 [==============================] - 0s 8ms/step - loss: 0.2611 -
val_loss: 0.0667
Epoch 4/50
10/10 [==============================] - 0s 8ms/step - loss: 0.2429 -
val_loss: 0.0636
Epoch 5/50
10/10 [==============================] - 0s 8ms/step - loss: 0.2224 -
val_loss: 0.0608
Epoch 6/50
10/10 [==============================] - 0s 9ms/step - loss: 0.2145 -
val_loss: 0.0583
Epoch 7/50
10/10 [==============================] - 0s 9ms/step - loss: 0.2040 -
val_loss: 0.0560
Epoch 8/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1919 -
val_loss: 0.0538
Epoch 9/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1805 -
val_loss: 0.0518
Epoch 10/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1702 -
val_loss: 0.0500
Epoch 11/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1620 -
val_loss: 0.0483
Epoch 12/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1544 -
val_loss: 0.0467
Epoch 13/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1450 -
val_loss: 0.0453
Epoch 14/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1389 -
val_loss: 0.0439
Epoch 15/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1323 -
val_loss: 0.0426
Epoch 16/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1259 -
val_loss: 0.0414
Epoch 17/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1190 -
val_loss: 0.0403
```

```
Epoch 18/50
10/10 [==============================] - 0s 8ms/step - loss: 0.1160 -
val_loss: 0.0393
Epoch 19/50
10/10 [==============================] - 0s 10ms/step - loss: 0.1087 -
val_loss: 0.0383
Epoch 20/50
10/10 [==============================] - 0s 9ms/step - loss: 0.1043 -
val_loss: 0.0374
Epoch 21/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0998 -
val_loss: 0.0366
Epoch 22/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0964 -
val_loss: 0.0357
Epoch 23/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0931 -
val_loss: 0.0350
Epoch 24/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0901 -
val_loss: 0.0343
Epoch 25/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0871 -
val_loss: 0.0336
Epoch 26/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0825 -
val_loss: 0.0329
Epoch 27/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0792 -
val_loss: 0.0323
Epoch 28/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0770 -
val_loss: 0.0317
Epoch 29/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0754 -
val_loss: 0.0312
Epoch 30/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0729 -
val_loss: 0.0307
Epoch 31/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0698 -
val_loss: 0.0302
Epoch 32/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0673 -
val_loss: 0.0297
```

```
Epoch 33/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0659 -
val_loss: 0.0293
Epoch 34/50
10/10 [==============================] - 0s 10ms/step - loss: 0.0638 -
val_loss: 0.0289
Epoch 35/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0622 -
val_loss: 0.0285
Epoch 36/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0602 -
val_loss: 0.0281
Epoch 37/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0587 -
val_loss: 0.0277
Epoch 38/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0570 -
val_loss: 0.0274
Epoch 39/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0557 -
val_loss: 0.0270
Epoch 40/50
10/10 [==============================] - 0s 9ms/step - loss: 0.0541 -
val_loss: 0.0267
Epoch 41/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0531 -
val_loss: 0.0264
Epoch 42/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0510 -
val_loss: 0.0261
Epoch 43/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0502 -
val_loss: 0.0259
Epoch 44/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0490 -
val_loss: 0.0256
Epoch 45/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0485 -
val_loss: 0.0253
Epoch 46/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0470 -
val_loss: 0.0251
Epoch 47/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0463 -
val_loss: 0.0249
```

```
Epoch 48/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0452 -
val_loss: 0.0247
Epoch 49/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0444 -
val_loss: 0.0244
Epoch 50/50
10/10 [==============================] - 0s 8ms/step - loss: 0.0430 -
val_loss: 0.0242
```

```
# New matrix with predicted recommendations
new_matrix = model.predict(X) * (X == 0)

new_customers_products_matrix = pd.DataFrame(new_matrix,

columns=customers_products_matrix.columns,

index=customers_products_matrix.index)
new_customers_products_matrix.head()
```

```
product_id                          00250175f79f584c14ab5cecd80553cd  \
customer_id
00426311a53f3c052943c88b692a3be2                          -0.027811
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.117858
0097c5abeb126a90646370f4a1cf3d93                          -0.037638
00f394e6fc446865ac4097b6db69ef4a                          -0.027006
00f6217307f712298d8e47215f0bf2ad                          -0.045436

product_id                          002af88741ba70c7b5cf4e4a0ad7ef85  \
customer_id
00426311a53f3c052943c88b692a3be2                          -0.098069
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.052274
0097c5abeb126a90646370f4a1cf3d93                           0.000797
00f394e6fc446865ac4097b6db69ef4a                           0.005310
00f6217307f712298d8e47215f0bf2ad                           0.046506

product_id                          00ba6d766f0b1d7b78a5ce3e1e033263  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.113313
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.008069
0097c5abeb126a90646370f4a1cf3d93                          -0.039476
00f394e6fc446865ac4097b6db69ef4a                          -0.000916
00f6217307f712298d8e47215f0bf2ad                           0.034681

product_id                          01422266d7a3131403364787ef9dab11  \
customer_id
```

```
00426311a53f3c052943c88b692a3be2                          -0.084298
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.092103
0097c5abeb126a90646370f4a1cf3d93                          -0.056825
00f394e6fc446865ac4097b6db69ef4a                           0.012633
00f6217307f712298d8e47215f0bf2ad                           0.011346

product_id                      0152f69b6cf919bcdaf117aa8c43e5a2  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.015485
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.055790
0097c5abeb126a90646370f4a1cf3d93                           0.008036
00f394e6fc446865ac4097b6db69ef4a                          -0.009118
00f6217307f712298d8e47215f0bf2ad                           0.048540

product_id                      01ff1ff8aa5dec93e9938b989393a4ca  \
customer_id
00426311a53f3c052943c88b692a3be2                          -0.060569
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.067538
0097c5abeb126a90646370f4a1cf3d93                          -0.057377
00f394e6fc446865ac4097b6db69ef4a                           0.021934
00f6217307f712298d8e47215f0bf2ad                          -0.033586

product_id                      0302c3fcf5e2d9526e243db50d30d5e3  \
customer_id
00426311a53f3c052943c88b692a3be2                          -0.050228
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.058481
0097c5abeb126a90646370f4a1cf3d93                          -0.037165
00f394e6fc446865ac4097b6db69ef4a                           0.026552
00f6217307f712298d8e47215f0bf2ad                          -0.047179

product_id                      034abfb9b758233fd393bd361d4ec599  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.122928
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.048592
0097c5abeb126a90646370f4a1cf3d93                           0.040647
00f394e6fc446865ac4097b6db69ef4a                           0.021764
00f6217307f712298d8e47215f0bf2ad                          -0.144405

product_id                      0364c36f8e845e4d309c0a3accc04b1c  \
customer_id
00426311a53f3c052943c88b692a3be2                          -0.004387
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.183007
0097c5abeb126a90646370f4a1cf3d93                           0.027601
00f394e6fc446865ac4097b6db69ef4a                          -0.014361
00f6217307f712298d8e47215f0bf2ad                           0.095142
```

```
product_id                          03d817e5e392e78674ed0bd8195f9159  ...  \
customer_id                                                           ...
00426311a53f3c052943c88b692a3be2                          -0.040119  ...
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.064682  ...
0097c5abeb126a90646370f4a1cf3d93                          -0.045859  ...
00f394e6fc446865ac4097b6db69ef4a                           0.029808  ...
00f6217307f712298d8e47215f0bf2ad                           0.017937  ...

product_id                          fd471a043ee8b8dd27f4086495e0724c  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.021898
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.007638
0097c5abeb126a90646370f4a1cf3d93                          -0.076501
00f394e6fc446865ac4097b6db69ef4a                          -0.007021
00f6217307f712298d8e47215f0bf2ad                          -0.059173

product_id                          fd5c5a67cd369732c2ac20dbd574d1d4  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.096527
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.048283
0097c5abeb126a90646370f4a1cf3d93                           0.097357
00f394e6fc446865ac4097b6db69ef4a                           0.055532
00f6217307f712298d8e47215f0bf2ad                          -0.053911

product_id                          fdba6f37e0258dbca54eb0ed8b293ae0  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.010593
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.034171
0097c5abeb126a90646370f4a1cf3d93                          -0.021234
00f394e6fc446865ac4097b6db69ef4a                           0.004298
00f6217307f712298d8e47215f0bf2ad                           0.046047

product_id                          fdbfd4cec6db6e89f22d9aa2c4839eed  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.039226
007e99fec9d53dfa4e5d8be9c2b36ca7                          -0.019479
0097c5abeb126a90646370f4a1cf3d93                          -0.003099
00f394e6fc446865ac4097b6db69ef4a                           0.006090
00f6217307f712298d8e47215f0bf2ad                           0.005825

product_id                          fdff6fbfaeefd11afe77be6f416e9c4e  \
customer_id
00426311a53f3c052943c88b692a3be2                           0.107248
007e99fec9d53dfa4e5d8be9c2b36ca7                           0.094361
```

```
0097c5abeb126a90646370f4a1cf3d93                          0.066441
00f394e6fc446865ac4097b6db69ef4a                          0.013408
00f6217307f712298d8e47215f0bf2ad                         -0.054118

product_id                      fe83af233315b04d9093c7edbcf789dd  \
customer_id
00426311a53f3c052943c88b692a3be2                         -0.093388
007e99fec9d53dfa4e5d8be9c2b36ca7                         -0.010750
0097c5abeb126a90646370f4a1cf3d93                          0.045257
00f394e6fc446865ac4097b6db69ef4a                          0.084576
00f6217307f712298d8e47215f0bf2ad                         -0.114798

product_id                      feb4ade62e32b8d74c6f69f635057964  \
customer_id
00426311a53f3c052943c88b692a3be2                         -0.006843
007e99fec9d53dfa4e5d8be9c2b36ca7                         -0.046327
0097c5abeb126a90646370f4a1cf3d93                         -0.047629
00f394e6fc446865ac4097b6db69ef4a                          0.015064
00f6217307f712298d8e47215f0bf2ad                          0.054364

product_id                      ff2c1ec09b1bb340e84f0d6b21cc7dbb  \
customer_id
00426311a53f3c052943c88b692a3be2                         -0.002289
007e99fec9d53dfa4e5d8be9c2b36ca7                         -0.070099
0097c5abeb126a90646370f4a1cf3d93                          0.009877
00f394e6fc446865ac4097b6db69ef4a                          0.005560
00f6217307f712298d8e47215f0bf2ad                          0.136693

product_id                      ff6caf9340512b8bf6d2a2a6df032cfa  \
customer_id
00426311a53f3c052943c88b692a3be2                          0.008615
007e99fec9d53dfa4e5d8be9c2b36ca7                         -0.064075
0097c5abeb126a90646370f4a1cf3d93                         -0.002227
00f394e6fc446865ac4097b6db69ef4a                          0.033758
00f6217307f712298d8e47215f0bf2ad                         -0.087225

product_id                      ff95ac47246ef13e48712ea1ff8df0d9
customer_id
00426311a53f3c052943c88b692a3be2                          0.049512
007e99fec9d53dfa4e5d8be9c2b36ca7                         -0.023826
0097c5abeb126a90646370f4a1cf3d93                         -0.054357
00f394e6fc446865ac4097b6db69ef4a                          0.053520
00f6217307f712298d8e47215f0bf2ad                         -0.138406
```

```
[5 rows x 907 columns]
```

```
# Recommender function
def recommender_for_customer(customer_id, interact_matrix, df_content, topn
= 10):
  '''
  Recommender product for customer
  '''
  pred_scores = interact_matrix.loc[customer_id].values

  df_scores = pd.DataFrame({'product_id':
list(customers_products_matrix.columns),
                            'score': pred_scores})

  df_rec = df_scores.set_index('product_id')\
                    .join(df_content.set_index('product_id'))\
                    .sort_values('score', ascending=False)\
                    .head(topn)[['score', 'product_category_name']]

  return df_rec[df_rec.score > 0]

# Customer with the longest order history
id = df_orders_per_customer.loc[df_orders_per_customer['orders_qty'] ==
df_orders_per_customer['orders_qty'].max(), 'customer_id'].values[0]

recommender_for_customer(customer_id=id,
                         interact_matrix=customers_products_matrix,
                         df_content=df)
```

product_id	score	product_category_name
5ddab10d5e0a23acb99acf56b62b3276	5.0	utilidades_domesticas
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
5ddab10d5e0a23acb99acf56b62b3276	5.0	utilidades_domesticas
5ddab10d5e0a23acb99acf56b62b3276	5.0	utilidades_domesticas
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho
ebf9bc6cd600eadd681384e3116fda85	5.0	cama_mesa_banho

```
# Test with the customer with the longest order history
recommender_for_customer(customer_id=id,
```

```
                        interact_matrix=new_customers_products_matrix,
                        df_content=df)

                                                    score
product_category_name
product_id
ad00a218e16f65efb3dfebe514994ca1  0.326867
brinquedos
4a5c3967bfd3629fe07ef4d0cc8c3818  0.299479
construcao_ferramentas_construcao
4a5c3967bfd3629fe07ef4d0cc8c3818  0.299479
construcao_ferramentas_construcao
99a4788cb24856965c36a24e339b6058  0.273854
cama_mesa_banho
99a4788cb24856965c36a24e339b6058  0.273854
cama_mesa_banho
99a4788cb24856965c36a24e339b6058  0.273854
cama_mesa_banho
99a4788cb24856965c36a24e339b6058  0.273854
cama_mesa_banho
99a4788cb24856965c36a24e339b6058  0.273854
cama_mesa_banho
b114bf337c0626166abe574eee9e3f32  0.255011
moveis_escritorio
b114bf337c0626166abe574eee9e3f32  0.255011
moveis_escritorio
```

References

SANTANA, Marlesson. **Deep Learning para Sistemas de Recomendação - Parte 2: Filtragem Colaborativa com Autoencoders**. Medium, 2019. Disponível em: https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-2-filtragem-colaborativa-com-autoencoders-347ba7d53bae. Acesso em: 06 de dezembro de 2023.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: **14 de dez. de 2023**

Participantes da Entrega **[matriculados em Residência em IA]**:

Állan Christoffer Pereira Silva

Entrega: **[descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]**

Esta entrega consistiu na consolidação da base de dados pré-processada para o treinamento de um sistema de recomendação usando a abordagem de filtragem colaborativa com rede Autoencoder. Além disso, foi implementada a camada gold da arquitetura, responsável por armazenar os dados de saída do treinamento do modelo.

Os requisitos básicos para a entrega eram:

- Realizar a filtragem colaborativa com Autoencoder envolvendo os outros atributos do dataset.
- Implementar a camada gold com os resultados das recomendações e visualizações desses resultados.

Os produtos gerados para esta entrega estão descritos a seguir:

- Documentação acerca da modelagem utilizada para a construção do sistema de recomendação: https://docs.google.com/document/d/1S5H_ep963bf7s8G8D-u9UlHDQpgPgfvhqJnZrVGZ_n4/edit?usp=sharing. De forma geral, foi explicado sobre a abordagem híbrida, isto é, a realização de filtragem colaborativa considerando mais atributos na matriz de interação, além das próprias interações.
- A implementação da filtragem colaborativa utilizando os demais atributos pode ser encontrada no notebook a seguir: https://colab.research.google.com/drive/1yv5GxLpMRTvTysPl9aIhES6-pCzf0EGC?usp=sharing. Um ponto importante a ser destacado é que para reduzir a dimensionalidade da matriz de interação resultante foram considerados os usuários com um histórico de no mínimo 5 pedidos. Dessa forma, o treinamento dos modelos preditivos ganha maior eficiência computacional.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Para a próxima entrega do dia 21/12/2023, estão planejadas as seguintes atividades:

- Utilização do Model Registry para armazenamento das diferentes arquiteturas Autoencoder testadas.
- Construção de um painel de visualização dos resultados com Looker Studio.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Modelagem de RecSys

Autor
Állan Christoffer Pereira Silva

Introdução
Este documento tem como objetivo principal apresentar a abordagem utilizada para a construção do sistema de recomendação. Nele é apresentado como o método da filtragem colaborativa foi utilizada inicialmente com a base de dados de e-commerces já pré-processada. Além disso, é explicado como a utilização de redes Autoencoder podem gerar previsões de interações de clientes com produtos que nunca compraram através do processo de compressão e reconstrução dos dados de entrada. Por fim, é detalhado como foi construído um método híbrido para o sistema de recomendação, isto é, como outros atributos além das próprias interações entre clientes e produtos foram agregados.

Filtragem colaborativa
A filtragem colaborativa em sistemas de recomendação é uma técnica amplamente utilizada para prever as preferências de um usuário com base nas preferências de outros usuários. Essa abordagem baseia-se no princípio de que usuários com preferências similares em alguns itens tendem a ter preferências similares em outros itens.

Figura 1. Matriz de interação cliente-produto

	product_id			
customer_id	002501	002af8	00ba6d	014222
004263	-	-	4,5	-
007e99	-	4,9	-	5,0
0097c5	3,7	-	-	-
00f394	-	2,2	4,8	-

Fonte: próprio autor

Os elementos centrais desse método são:

- **Princípio Básico:** Baseia-se em criar um sistema de recomendação que aproveita as avaliações ou interações dos usuários com produtos ou serviços. No contexto da aplicação do projeto, em um site de e-commerce, as recomendações são geradas considerando as avaliações que diferentes clientes dão aos produtos.

- **Matriz de Interação:** a matriz de interação usuário-item é o elemento central. Cada linha representa um cliente e cada coluna um item (produto). Os valores na matriz são as avaliações ou a intensidade da interação (compra). A Figura 1 mostra um exemplo de como essa matriz pode ser representada.
- **Preenchimento de Lacunas:** Um grande desafio é lidar com a “esparsidade'” da matriz, já que muitos clientes interagem com apenas uma pequena fração dos produtos disponíveis. O objetivo é preencher as lacunas predizendo como um cliente pode classificar ou interagir com itens não avaliados.

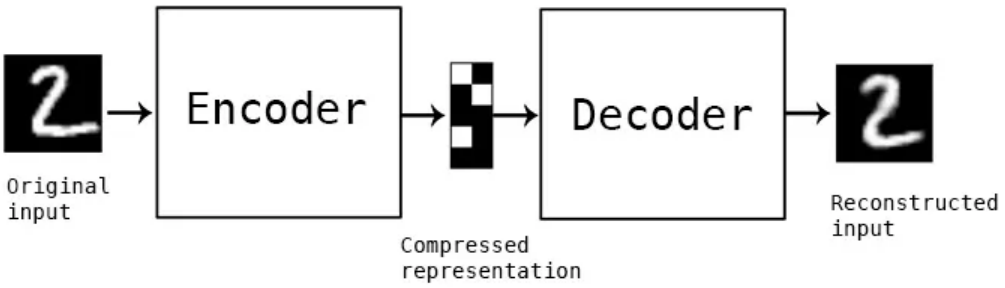
A montagem da matriz de interação é um passo fundamental em qualquer sistema de filtragem colaborativa. A coleta e normalização desses dados são passos essenciais, garantindo que a matriz reflita com precisão as interações dos usuários. No entanto, um desafio significativo é a esparsidade da matriz, pois muitos usuários interagem com apenas um subconjunto limitado de itens.

Para abordar isso, técnicas como imputação de valores podem ser utilizadas, ou modelos que naturalmente toleram altos níveis de esparsidade, como os autoencoders, são empregados. Além disso, a matriz é geralmente dividida em conjuntos de treinamento e teste para permitir a avaliação precisa do modelo de recomendação. A precisão das recomendações geradas depende em grande parte da qualidade da montagem e do processamento desta matriz de interação.

Redes Autoencoder em Filtragem Colaborativa

Redes autoencoder desempenham um papel crucial na filtragem colaborativa, especialmente no contexto de sistemas de recomendação. Essas redes são uma forma de rede neural projetada para aprender representações codificadas de dados.

Figura 2. Representação conceitual de um Autoencoder



Fonte: https://blog.keras.io/building-autoencoders-in-keras.html

Um autoencoder consiste em duas partes principais (Figura 2): um codificador, que comprime os dados em uma representação de menor dimensão, e um decodificador, que reconstrói os dados a partir desta representação compacta. Quando aplicados à filtragem colaborativa, os autoencoders

são usados para aprender representações densas da matriz de interação, que é tipicamente esparsa. Isso é feito alimentando a matriz de interação no autoencoder, que então aprende a compactar e a reconstruir esses dados.

A vantagem dos autoencoders em comparação com métodos mais tradicionais de filtragem colaborativa, como a fatoração de matriz, é notável, especialmente em sua capacidade de lidar com desafios específicos na análise de dados de recomendação.

- **Captura de Relações Não-Lineares:** Um dos aspectos mais significativos dos autoencoders é sua habilidade de capturar relações não-lineares complexas entre usuários e itens. Enquanto métodos como a fatoração de matriz assumem relações lineares ou simplificadas, os autoencoders, através de suas múltiplas camadas e estruturas de rede neural, podem identificar e modelar padrões e dependências muito mais complexos e sutis nos dados. Isso permite que eles descubram e utilizem nuances nas interações dos usuários que podem ser perdidas em abordagens mais simples.
- **Eficiência no Tratamento da Esparsidade de Dados:** Outra grande vantagem dos autoencoders é sua eficiência em lidar com a esparsidade dos dados, um problema comum em sistemas de recomendação. Muitas vezes, os usuários interagem com apenas um pequeno subconjunto de itens disponíveis, resultando em uma matriz de interação usuário-item amplamente não preenchida. Enquanto a fatoração de matriz pode lutar para lidar com essa esparsidade, os autoencoders são projetados para reconstruir com eficácia os dados a partir de informações limitadas, preenchendo as lacunas e fazendo previsões precisas mesmo com dados esparsos.
- **Flexibilidade e Personalização:** Os autoencoders oferecem uma flexibilidade significativa na modelagem de sistemas de recomendação. Eles podem ser ajustados e personalizados para se adequar a diferentes tipos de dados e requisitos específicos, tornando-os uma ferramenta poderosa para uma ampla gama de aplicações em sistemas de recomendação.

Método Híbrido

A implementação de um esquema híbrido de filtragem colaborativa usando Autoencoders representa um grande avanço nos sistemas de recomendação, oferecendo recomendações altamente personalizadas e contextualmente relevantes. Este método não se limita a aprender a partir das interações dos usuários, mas também incorpora uma gama de características adicionais, como as dimensões físicas dos produtos, categorias de produtos, dados geográficos dos clientes e análises de sentimentos das avaliações.

O processo começa com a normalização e integração destes dados adicionais na matriz de interação usuário-item, expandindo a capacidade do sistema de entender as preferências dos usuários. O Autoencoder é treinado com esta matriz enriquecida, aprendendo não só as interações diretas, mas também as nuances proporcionadas por essas características adicionais. O resultado

é um sistema que não só compreende as preferências passadas dos usuários, mas também antecipa suas necessidades e interesses futuros, considerando um espectro mais amplo de fatores.

Referências

SANTANA, MARLESSON. **Deep Learning para Sistemas de Recomendação Parte 2: Filtragem Colaborativa com Autoencoders**. Medium, 4 de março de 2019. Disponível em: <https://medium.com/data-hackers/deep-learning-para-sistemas-de-recomenda%C3%A7%C3%A3o-parte-2-filtragem-colaborativa-com-autoencoders-347ba7d53bae>. Acesso em: 6 dez. 2023.

POLIGNANO, M. et al. **Together is Better: Hybrid Recommendations Combining Graph Embeddings and Contextualized Word Representations**. Fifteenth ACM Conference on Recommender Systems, 13 set. 2021.

ACM RECSYS. R**ecSys 2015 Session 2b: Cold Start and Hybrid Recommender Systems.** Disponível em: <https://youtu.be/wEbatX4J-1g?si=ISDtglTcX1JAHZZg>. Acesso em: 12 dez. 2023.

Jupyter Notebook: RecSys Hybrid

Recsys Hybrid Method with Autoencoder

Author: Állan Christoffer Pereira Silva

Setup

```
%%capture
!pip install torchviz
```

```
# Importing frameworks
import pandas as pd
from scipy.sparse import csr_matrix

import torch
from torch import nn
from torch.optim import Adam

from torchviz import make_dot
from IPython.display import Image
```

```
# Reading preprocessed dataset
df = pd.read_csv('preprocessed.csv')
```

```
# Verifying dataframe dimensions
df.shape
```

```
(4843, 45)
```

Predictive Modeling

```
# Selecting columns of interest
user_id_col = 'customer_id'
item_id_col = 'product_id'
rating_col = 'review_score'
feature_cols = [col for col in df.columns if col not in [user_id_col,
item_id_col, rating_col]]

# Creating the interaction matrix with user ratings
interaction_matrix = df.pivot_table(index=user_id_col, columns=item_id_col,
values=rating_col)

# Including additional features for each user
for col in feature_cols:
    user_features = df.groupby(user_id_col)[col].mean()  # Average of
features per user
    interaction_matrix[col] = interaction_matrix.index.map(user_features)
```

```
interaction_matrix.head()
```

```
product_id                          00250175f79f584c14ab5cecd80553cd  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                          002af88741ba70c7b5cf4e4a0ad7ef85  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                          00ba6d766f0b1d7b78a5ce3e1e033263  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                          01422266d7a3131403364787ef9dab11  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                          0152f69b6cf919bcdaf117aa8c43e5a2  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                          01ff1ff8aa5dec93e9938b989393a4ca  \
customer_id
```

```
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                    0302c3fcf5e2d9526e243db50d30d5e3  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                    034abfb9b758233fd393bd361d4ec599  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                    0364c36f8e845e4d309c0a3accc04b1c  \
customer_id
00426311a53f3c052943c88b692a3be2                                NaN
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN
0097c5abeb126a90646370f4a1cf3d93                                NaN
00f394e6fc446865ac4097b6db69ef4a                                NaN
00f6217307f712298d8e47215f0bf2ad                                NaN

product_id                    03d817e5e392e78674ed0bd8195f9159  ...  \
customer_id                                                     ...
00426311a53f3c052943c88b692a3be2                                NaN  ...
007e99fec9d53dfa4e5d8be9c2b36ca7                                NaN  ...
0097c5abeb126a90646370f4a1cf3d93                                NaN  ...
00f394e6fc446865ac4097b6db69ef4a                                NaN  ...
00f6217307f712298d8e47215f0bf2ad                                NaN  ...

product_id                    customer_state_1  customer_state_2  \
customer_id
00426311a53f3c052943c88b692a3be2               0.0               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7               0.0               0.0
0097c5abeb126a90646370f4a1cf3d93               0.0               1.0
00f394e6fc446865ac4097b6db69ef4a               0.0               1.0
00f6217307f712298d8e47215f0bf2ad               0.0               0.0
```

```
product_id                       customer_state_3  customer_state_4  \
customer_id
00426311a53f3c052943c88b692a3be2               1.0               0.0
007e99fec9d53dfa4e5d8be9c2b36ca7               1.0               1.0
0097c5abeb126a90646370f4a1cf3d93               1.0               0.0
00f394e6fc446865ac4097b6db69ef4a               1.0               1.0
00f6217307f712298d8e47215f0bf2ad               1.0               0.0

product_id                       product_name_lenght  \
customer_id
00426311a53f3c052943c88b692a3be2            -2.569230
007e99fec9d53dfa4e5d8be9c2b36ca7             0.715151
0097c5abeb126a90646370f4a1cf3d93             0.232154
00f394e6fc446865ac4097b6db69ef4a            -1.603236
00f6217307f712298d8e47215f0bf2ad            -0.540642

product_id                       product_description_lenght  \
customer_id
00426311a53f3c052943c88b692a3be2                   -0.822483
007e99fec9d53dfa4e5d8be9c2b36ca7                   -0.285343
0097c5abeb126a90646370f4a1cf3d93                   -0.121804
00f394e6fc446865ac4097b6db69ef4a                    0.867888
00f6217307f712298d8e47215f0bf2ad                   -0.639207

product_id                       product_weight_g  product_length_cm  \
customer_id
00426311a53f3c052943c88b692a3be2         -0.505128           1.971207
007e99fec9d53dfa4e5d8be9c2b36ca7         -0.501022          -1.036639
0097c5abeb126a90646370f4a1cf3d93          0.047661           1.971207
00f394e6fc446865ac4097b6db69ef4a         -0.348454          -0.689580
00f6217307f712298d8e47215f0bf2ad         -0.316039          -0.342520

product_id                       product_height_cm  product_width_cm
customer_id
00426311a53f3c052943c88b692a3be2          -0.296959         -0.703035
007e99fec9d53dfa4e5d8be9c2b36ca7          -0.427471         -0.863276
0097c5abeb126a90646370f4a1cf3d93          -0.296959          1.940940
00f394e6fc446865ac4097b6db69ef4a          -0.035936         -0.382553
00f6217307f712298d8e47215f0bf2ad           0.420855          0.178290

[5 rows x 940 columns]
```

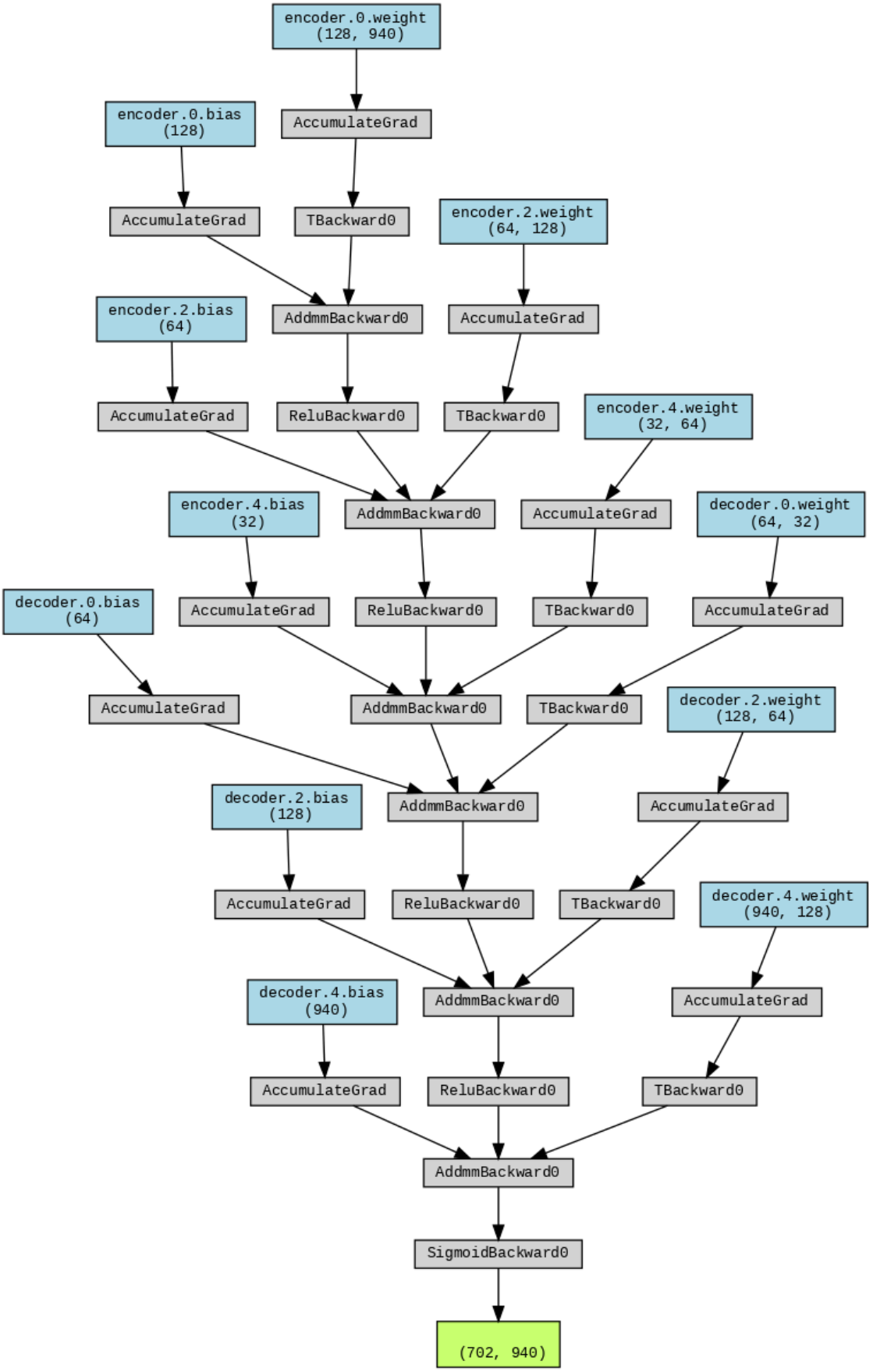
```
# Defining the Autoencoder architecture
class Autoencoder(nn.Module):
```

```
    def __init__(self, num_features):
        super(Autoencoder, self).__init__()
        # Encoder
        self.encoder = nn.Sequential(
            nn.Linear(num_features, 128),  # Linear layer from num_features
to 128
            nn.ReLU(),                     # ReLU activation
            nn.Linear(128, 64),            # Linear layer from 128 to 64
            nn.ReLU(),                     # ReLU activation
            nn.Linear(64, 32)              # Latent layer, linear from 64
to 32
        )
        # Decoder
        self.decoder = nn.Sequential(
            nn.Linear(32, 64),             # Linear layer from 32 to 64
            nn.ReLU(),                     # ReLU activation
            nn.Linear(64, 128),            # Linear layer from 64 to 128
            nn.ReLU(),                     # ReLU activation
            nn.Linear(128, num_features),  # Linear layer back to original
size
            nn.Sigmoid()                   # Sigmoid activation (if data is
normalized between 0 and 1)
        )

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x

# Number of features (columns) in the interaction matrix
num_features = interaction_matrix.shape[1]

# Instantiating the model
model = Autoencoder(num_features)

# Displaying the model architecture
model
```

```
Autoencoder(
  (encoder): Sequential(
    (0): Linear(in_features=940, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): ReLU()
    (4): Linear(in_features=64, out_features=32, bias=True)
```

```
  )
  (decoder): Sequential(
    (0): Linear(in_features=32, out_features=64, bias=True)
    (1): ReLU()
    (2): Linear(in_features=64, out_features=128, bias=True)
    (3): ReLU()
    (4): Linear(in_features=128, out_features=940, bias=True)
    (5): Sigmoid()
  )
)
```

```
# Convert the interaction matrix to a sparse matrix
sparse_interaction_matrix = csr_matrix(interaction_matrix.fillna(0))

# Convert the interaction matrix to a PyTorch tensor
interaction_tensor = torch.FloatTensor(sparse_interaction_matrix.toarray())

output = model(interaction_tensor)
graph = make_dot(output, params=dict(model.named_parameters()))
graph_file = 'autoencoder_graph'
graph.render(graph_file, format='png')

# Displaying architecture image
display(Image(filename='autoencoder_graph.png'))
```


encoder.0.weight
(128, 940)
encoder.0.bias
(128)
AccumulateGrad
AccumulateGrad
TBackward0
encoder.2.weight
(64, 128)
encoder.2.bias
(64)
AddmmBackward0
AccumulateGrad
AccumulateGrad
ReluBackward0
TBackward0
encoder.4.weight
(32, 64)
encoder.4.bias
(32)
AddmmBackward0
AccumulateGrad
decoder.0.weight
(64, 32)
decoder.0.bias
(64)
AccumulateGrad
ReluBackward0
TBackward0
AccumulateGrad
AccumulateGrad
AddmmBackward0
TBackward0
decoder.2.weight
(128, 64)
decoder.2.bias
(128)
AddmmBackward0
AccumulateGrad
AccumulateGrad
ReluBackward0
TBackward0
decoder.4.weight
(940, 128)
decoder.4.bias
(940)
AddmmBackward0
AccumulateGrad
AccumulateGrad
ReluBackward0
TBackward0
AddmmBackward0
SigmoidBackward0
(702, 940)

```
# Define loss function and optimizer
criterion = nn.MSELoss()
optimizer = Adam(model.parameters(), lr=0.001)

# Training parameters
epochs = 20
batch_size = 64

# Train the model
model.train()
for epoch in range(epochs):
    train_loss = 0.0
    for i in range(0, len(interaction_tensor), batch_size):
        # Data batch
        inputs = interaction_tensor[i:i+batch_size]

        # Zero the optimizer's gradients
        optimizer.zero_grad()

        # Forward pass
        outputs = model(inputs)

        # Calculate loss
        loss = criterion(outputs, inputs)

        # Backward pass and optimization
        loss.backward()
        optimizer.step()

        train_loss += loss.item()

    # Calculate average loss
    train_loss /= len(interaction_tensor) / batch_size

    print(f"Epoch {epoch+1}/{epochs}, Loss: {train_loss:.4f}")

print("Training completed.")
```

```
Epoch 1/20, Loss: 0.2682
Epoch 2/20, Loss: 0.1949
Epoch 3/20, Loss: 0.0487
Epoch 4/20, Loss: 0.0373
Epoch 5/20, Loss: 0.0370
Epoch 6/20, Loss: 0.0369
Epoch 7/20, Loss: 0.0369
```

```
Epoch 8/20, Loss: 0.0368
Epoch 9/20, Loss: 0.0367
Epoch 10/20, Loss: 0.0367
Epoch 11/20, Loss: 0.0367
Epoch 12/20, Loss: 0.0367
Epoch 13/20, Loss: 0.0367
Epoch 14/20, Loss: 0.0367
Epoch 15/20, Loss: 0.0367
Epoch 16/20, Loss: 0.0367
Epoch 17/20, Loss: 0.0367
Epoch 18/20, Loss: 0.0367
Epoch 19/20, Loss: 0.0367
Epoch 20/20, Loss: 0.0367
Training completed.
```

```
def make_recommendations(user_id, interaction_matrix, model, top_k=5):
    # Check if the user is in the interaction matrix
    if user_id in interaction_matrix.index:
        user_interactions =
torch.FloatTensor(interaction_matrix.loc[user_id].values).unsqueeze(0)

        # Put the model in evaluation mode
        model.eval()
        with torch.no_grad():
            # Generate predictions for the user
            predictions = model(user_interactions).squeeze()

        # Get indices of items with the highest predictions
        recommended_items_idx =
predictions.argsort(descending=True)[:top_k]

        # Map indices to product IDs
        recommended_products =
interaction_matrix.columns[recommended_items_idx]
        return recommended_products
    else:
        return "User not found in the database."

# Example of making a recommendation for a specific user
user_id = interaction_matrix.index[0]  # Taking the first user as an
example
recommended_products = make_recommendations(user_id, interaction_matrix,
model)

recommended_products
```

```
Index(['b3e40ff639c185b9d726b4b19c17e6cd',
'ae2f987a774d66463c1bfcc0a82c3456',
       'aeb767ca82c5a6cca8bbac33c4e21579',
'af35be35db4ad0dc288b571453337376',
       'afc6bc70dc56fcf15c7f9f1e4bc67dda'],
      dtype='object', name='product_id')
```

```
# Reading original datasets
df_customers = pd.read_csv('olist_customers_dataset.csv')
df_products = pd.read_csv('olist_products_dataset.csv')
```

```
# Retrieving customer information
df_customers.loc[df_customers['customer_id']==user_id]
```

```
                          customer_id                customer_unique_id
\
10225  00426311a53f3c052943c88b692a3be2  aed838b04abeb2fb94566d0a073bd718

       customer_zip_code_prefix customer_city customer_state
10225                    4551.0     sao paulo             SP
```

```
# Retrieving recommended product information
df_products.loc[df_products['product_id'].isin(recommended_products.tolist(
))]
```

```
                           product_id             product_category_name
\
3239   b3e40ff639c185b9d726b4b19c17e6cd                        cool_stuff
6171   afc6bc70dc56fcf15c7f9f1e4bc67dda               utilidades_domesticas
8624   aeb767ca82c5a6cca8bbac33c4e21579  construcao_ferramentas_construcao
24599  af35be35db4ad0dc288b571453337376                  moveis_decoracao
29600  ae2f987a774d66463c1bfcc0a82c3456                           pet_shop

       product_name_lenght  product_description_lenght  product_photos_qty
\
3239                  55.0                       934.0                 1.0
6171                  59.0                       303.0                 1.0
8624                  31.0                      1505.0                 1.0
24599                 24.0                       457.0                 4.0
29600                 49.0                      1180.0                 1.0

       product_weight_g  product_length_cm  product_height_cm  \
3239             2150.0               42.0               25.0
6171             9900.0               61.0               31.0
8624             1425.0               35.0               13.0
24599             100.0               80.0               10.0
29600             200.0               16.0               14.0
```

	product_width_cm
3239	15.0
6171	31.0
8624	20.0
24599	60.0
29600	11.0

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 21 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Esta entrega consistiu na construção de diferentes arquiteturas Autoencoder para posterior registro no serviço Model Registry do Google Cloud Platform. Além disso, também foi requisitada a criação de um painel para permitir ao usuário a visualização dos dados resultantes do treinamento.

Os requisitos básicos para a entrega eram:

- Utilizar o Model Registry para armazenamento das diferentes arquiteturas Autoencoder testadas.
- Construir um painel de visualização dos resultados com Looker Studio.

Os produtos gerados para esta entrega estão descritos a seguir:

- O documento a seguir descreve brevemente as arquiteturas testadas e relata sobre o processo de registro no Model Registry: https://docs.google.com/document/d/1OocHrBRLBk97xRlxMdbbLd48kl3o-MOQ99Ww2s0oaWo/edit?usp=sharing. De forma geral, foram testadas variações de arquiteturas autoencoder, quatro no total, das mais básicas até as mais robustas. Vale ressaltar que esses testes tiveram como principal objetivo a geração de modelos para registro e futuro deploy. No entanto, foi encontrado um empecilho no registro dos modelos que pode ter sido causado por incompatibilidade de pacotes. Assim, fica como próximos passos o estudo e a correção do erro gerado.
- Link para a interface do relatório de resultados obtidos a partir da matriz de interação de clientes e produtos construída no Looker Studio: https://lookerstudio.google.com/reporting/7230687b-442d-43c0-9797-b6ee9f1c5812

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega do dia 11/01/2024, estão planejadas as seguintes atividades:

- Investigação e correção do erro encontrado no processo de registro dos modelos no Model Registry.

- Estudo da plataforma Apache Airflow para construção de um fluxo de trabalho automatizado do sistema de recomendação.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Arquiteturas Testadas

Autor
Állan Christoffer Pereira Silva

Introdução
Este documento tem como objetivo principal apresentar as arquiteturas de redes neurais testadas para abordar a aplicação de sistema de recomendação híbrido. Nele é possível compreender as diferenças entre cada uma delas, destacando vantagens e desvantagens. Por fim, foi feito um relato sobre o processo de registro dos modelos no serviço Model Registry do Google Cloud Platform.

Modelo 1: Autoencoder Básico
O Autoencoder Básico é uma arquitetura neural simples, adequada para sistemas de recomendação híbridos, que visa aprender representações compactas dos dados de entrada (por exemplo, interações de usuários com produtos). Neste modelo, a codificação (encoder) reduz progressivamente a dimensionalidade dos dados, capturando as características essenciais em um espaço latente. O decodificador (decoder) então reconstrói os dados de entrada a partir deste espaço latente, permitindo que o modelo aprenda a estrutura subjacente dos dados. Essa arquitetura é particularmente útil para identificar padrões ocultos nas preferências dos clientes e recomendar produtos com base nessas preferências aprendidas.

Figura 1. Esquema do Modelo 1

```
BasicAutoencoder(
  (encoder): Sequential(
    (0): Linear(in_features=940, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=64, bias=True)
    (3): ReLU()
    (4): Linear(in_features=64, out_features=32, bias=True)
  )
  (decoder): Sequential(
    (0): Linear(in_features=32, out_features=64, bias=True)
    (1): ReLU()
    (2): Linear(in_features=64, out_features=128, bias=True)
    (3): ReLU()
    (4): Linear(in_features=128, out_features=940, bias=True)
    (5): Sigmoid()
  )
)
```

Fonte: próprio autor

Sendo o primeiro modelo da série, o Autoencoder Básico serve como ponto de partida. Ele não possui as camadas adicionais ou técnicas de regularização encontradas nos modelos subsequentes. A simplicidade desta arquitetura a torna fácil de entender e implementar, mas menos capaz de capturar complexidades em dados de recomendação do que os modelos mais avançados.

A principal vantagem deste modelo é sua simplicidade e eficiência computacional. Ele pode ser um bom ponto de partida para sistemas de recomendação com conjuntos de dados menos complexos. No entanto, a principal desvantagem é sua limitação em captar relações complexas e nuances nos dados, o que pode resultar em recomendações menos precisas em comparação com modelos mais sofisticados.

Modelo 2: Autoencoder com Dropout e Batch Normalization

Esta versão do autoencoder incorpora dropout e normalização em lote para melhorar a generalização e acelerar o treinamento. No contexto de sistemas de recomendação híbridos, estas adições ajudam o modelo a evitar o overfitting aos dados de treinamento e a responder melhor a variações nos padrões de consumo dos clientes. O dropout desativa aleatoriamente alguns neurônios durante o treinamento, forçando o modelo a não depender excessivamente de qualquer parte da rede. A normalização em lote ajuda a estabilizar o aprendizado, acelerando a convergência.

Figura 2. Esquema do Modelo 2

```
EnhancedAutoencoder(
  (encoder): Sequential(
    (0): Linear(in_features=940, out_features=256, bias=True)
    (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Dropout(p=0.3, inplace=False)
    (4): Linear(in_features=256, out_features=128, bias=True)
    (5): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): Dropout(p=0.3, inplace=False)
    (8): Linear(in_features=128, out_features=64, bias=True)
  )
  (decoder): Sequential(
    (0): Linear(in_features=64, out_features=128, bias=True)
    (1): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Dropout(p=0.3, inplace=False)
    (4): Linear(in_features=128, out_features=256, bias=True)
    (5): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): Dropout(p=0.3, inplace=False)
    (8): Linear(in_features=256, out_features=940, bias=True)
    (9): Sigmoid()
  )
)
```

Fonte: próprio autor

Em comparação com o Autoencoder Básico, este modelo adiciona complexidade e robustez. As técnicas de dropout e batch normalization são projetadas para melhorar a capacidade do modelo de generalizar para dados não vistos, um aspecto crucial para sistemas de recomendação que lidam com uma variedade de preferências de clientes e padrões de consumo.

A principal vantagem desta arquitetura é sua maior robustez e capacidade de generalização, o que é crítico para sistemas de recomendação que operam em ambientes dinâmicos. No entanto, estas melhorias vêm à custa de uma complexidade aumentada e um tempo de treinamento potencialmente maior. Além disso, o ajuste fino dos hiperparâmetros, como taxas de dropout, torna-se mais crítico para o desempenho do modelo.

Modelo 3: Autoencoder com Camadas Densas e Skip Connections

O autoencoder com camadas densas e conexões residuais (skip connections) representa um avanço significativo em termos de complexidade. As camadas densas permitem ao modelo aprender representações mais ricas dos dados, o que é benéfico em sistemas de recomendação para capturar uma gama mais ampla de padrões de interação entre usuários e produtos. As conexões residuais facilitam o fluxo de gradientes durante o treinamento, permitindo que o modelo aprenda eficientemente mesmo quando a arquitetura se torna mais profunda.

Figura 3. Esquema do Modelo 3

```
DenseAutoencoder(
  (encoder): Sequential(
    (0): Linear(in_features=940, out_features=512, bias=True)
    (1): ReLU()
    (2): Linear(in_features=512, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=128, bias=True)
  )
  (skip): Sequential(
    (0): Linear(in_features=940, out_features=128, bias=True)
    (1): ReLU()
  )
  (decoder): Sequential(
    (0): Linear(in_features=128, out_features=256, bias=True)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=512, bias=True)
    (3): ReLU()
    (4): Linear(in_features=512, out_features=940, bias=True)
    (5): Sigmoid()
  )
)
```

Fonte: próprio autor

Este modelo difere significativamente do anterior pela introdução de camadas mais densas e conexões residuais. Essas adições tornam o modelo mais apto para capturar relações complexas e não lineares nos dados, superando as limitações dos modelos anteriores que poderiam ter dificuldades com a profundidade e a complexidade da rede.

A capacidade de capturar relações mais complexas nos dados é uma vantagem significativa, tornando este modelo particularmente adequado para sistemas de recomendação com grande diversidade de produtos e comportamentos de usuários. A desvantagem é o aumento na complexidade computacional e na necessidade de mais dados para treinar efetivamente o modelo, além de uma maior atenção ao risco de overfitting.

Modelo 4: Autoencoder Variacional

O Autoencoder Variacional (VAE) é uma abordagem avançada que utiliza um espaço latente probabilístico. Esta arquitetura é especialmente interessante para sistemas de recomendação, pois permite não apenas aprender representações compactas dos dados, mas também explorar essas representações para gerar novas amostras. Em um contexto de recomendação, isso significa que o modelo pode gerar novos padrões de interação, potencialmente descobrindo novas recomendações que não são explicitamente presentes nos dados de treinamento.

Figura 4. Esquema do Modelo 4

```
VariationalAutoencoder(
  (fc1): Linear(in_features=940, out_features=256, bias=True)
  (fc21): Linear(in_features=256, out_features=128, bias=True)
  (fc22): Linear(in_features=256, out_features=128, bias=True)
  (fc3): Linear(in_features=128, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=940, bias=True)
)
```

Fonte: próprio autor

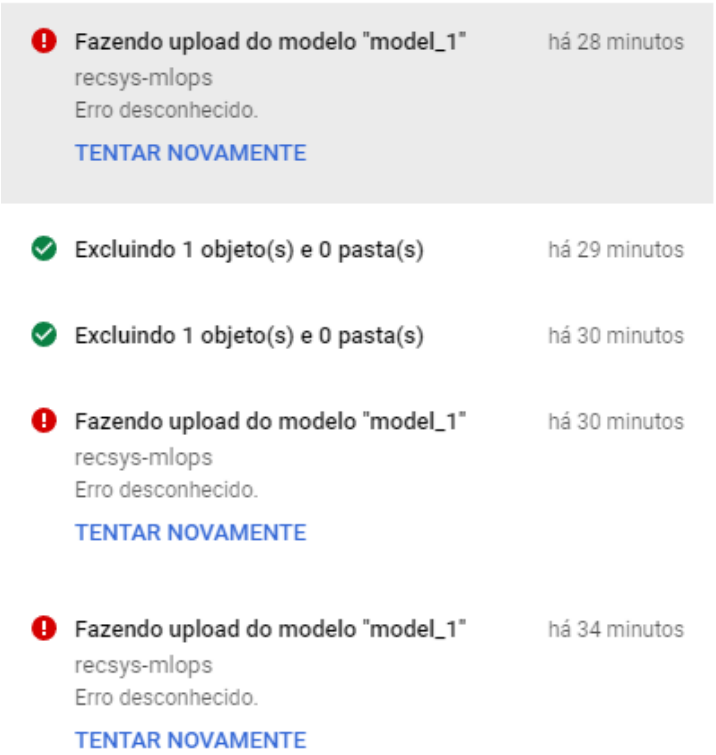
O grande diferencial do VAE em relação aos modelos anteriores é sua capacidade de modelar o espaço latente como uma distribuição probabilística. Isso permite que o modelo capture não apenas a estrutura dos dados, mas também a incerteza e a variação inerentes às preferências dos usuários, oferecendo um nível de flexibilidade e capacidade de generalização significativamente maior.

A principal vantagem do VAE é sua capacidade de gerar novos dados e descobrir padrões latentes complexos, tornando-o extremamente valioso para sistemas de recomendação que buscam inovação e personalização. No entanto, essa arquitetura é também a mais complexa e desafiadora para treinar e afinar, exigindo um entendimento profundo de técnicas de aprendizado de máquina e um conjunto de dados substancial para treinamento eficaz. Além disso, a interpretação dos resultados pode ser menos intuitiva em comparação com modelos mais simples.

Registro no Model Registry

O Model Registry no Vertex AI da Google Cloud Platform (GCP) é um recurso que permite aos usuários gerenciar, versionar e manter um repositório centralizado de modelos de Machine Learning (ML). Este registro de modelos oferece uma maneira organizada e eficiente para armazenar, compartilhar e acessar modelos de ML dentro de uma organização ou entre projetos. A sequência de erros pode ser visualizada na figura abaixo.

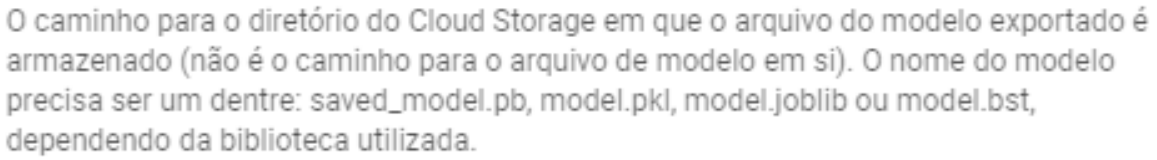
Figura 5. Erros no registro dos modelos



Fonte: próprio autor

Ao realizar o processo de registro dos modelos treinados que foram descritos anteriormente, houve um erro de incompatibilidade, o qual, até o momento não foi contornado. Uma das hipóteses para a ocorrência desse erro é o fato dos modelos serem salvos em uma extensão padrão do framework PyTorch (pth) e o serviço do Model Registry esperar alguma das extensões abaixo.

Figura 6. Extensões aceitas pelo Model Registry



Fonte: próprio autor

Dessa forma, fica como próximos passos a investigação e a correção dos causas relacionadas ao erro de incompatibilidade encontrado para que os modelos testados possam ser registrados e, posteriormente, disponibilizados para um serviço de recomendação.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 11 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Állan Christoffer Pereira Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Até a entrega passada, a aplicação escolhida para a implementação de uma arquitetura própria de MLOps era um sistema de recomendação de produtos de e-commerce utilizando a abordagem de filtragem colaborativa. No entanto, devido a problemas relacionados ao tipo de modelagem escolhida e a erros de incompatibilidade de frameworks encontrados e que não puderam ser solucionados em tempo hábil, foi decidido a alteração da aplicação. Dessa forma, a nova aplicação escolhida foi o desenvolvimento de um modelo preditivo de análise de risco de crédito para a classificar clientes que poderiam ou não receber empréstimos. Vale ressaltar que a mudança da aplicação foi benéfica do ponto de vista da especialização na área de MLOps, pois permitiu uma maior exploração de ferramentas e serviços relacionados ao desenvolvimento e implantação de modelos de ML.

As entregas relacionadas a este último gate estão listadas a seguir:

- Documentação completa sobre o projeto: Arquitetura de Dados v3
- Aplicação web que consome o serviço preditivo: link
- Acompanhamento de experimentos e registro de modelos: link (usuário = mlflow, senha = mlflow!@#123)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Como etapas finais do processo, as atividades planejadas são:

- Consolidação das versões finais dos códigos implementados e dos experimentos.
- Escrita do artigo referente ao Trabalho de Conclusão de Curso.
- Construção dos slides de apresentação do TCC.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Repositório no GitHub: link

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

LUANA GUEDES BARROS MARTINS: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Arquitetura de Dados v3

Autor
Állan Christoffer Pereira Silva

Introdução

A ascensão das soluções de Inteligência Artificial (IA) e Aprendizado de Máquina (Machine Learning ou ML) tem revolucionado o modo como pessoas e organizações enfrentam desafios e otimizam processos, redefinindo as fronteiras do possível em diversos setores. Essa transformação digital, no entanto, é impulsionada por um trabalho árduo e muitas vezes invisível de analistas, cientistas e engenheiros de dados. Estes profissionais desempenham um papel crucial na construção e implantação dessas soluções inovadoras, um processo complexo que envolve não apenas o desenvolvimento de algoritmos, mas também sua integração e manutenção em ambientes operacionais reais.

No centro desta dinâmica de desenvolvimento está o Machine Learning Operations (MLOps), um derivado do DevOps, que vai além somente da implementação de ferramentas e tecnologias, estabelecendo-se como uma cultura indispensável. MLOps abrange práticas e princípios que facilitam o ciclo de vida de modelos de ML, desde a concepção até a produção, assegurando eficiência e qualidade. É fundamental para os profissionais de dados não apenas dominar as técnicas de modelagem, mas também compreender como "produtizar" seus códigos, pipelines e modelos, alinhando-os com os princípios de MLOps para garantir aplicações robustas e escaláveis.

Nesse contexto, torna-se crucial estabelecer uma organização inicial clara sobre a arquitetura desses sistemas, especialmente para aqueles profissionais que ainda não possuem a experiência relacionada com o desafio de colocar modelos em produção. Uma arquitetura bem definida não apenas mapeia os fluxos e ferramentas necessários, mas também guia o desenvolvimento de aplicações de forma estruturada e eficiente.

Esta pesquisa aborda exatamente este desafio, propondo a arquitetura StructML, baseada na arquitetura Delta da Databricks, oferecendo um guia prático para sua implementação em plataformas de nuvem como Google Cloud Platform (GCP), Amazon Web Service (AWS) e Microsoft Azure. A aplicabilidade e eficácia desta arquitetura são demonstradas através de um estudo de caso prático: a construção de um sistema de análise de risco de crédito no GCP, utilizando um modelo preditivo para avaliar a elegibilidade de clientes a empréstimos. Este trabalho, portanto, não apenas propõe uma arquitetura de sistemas inovadora, mas também ilustra sua aplicação prática em um cenário real, destacando sua relevância e potencial para auxiliar o trabalho de profissionais da área de dados e otimizar a entrega de projetos de IA e ML.

Arquiteturas de Dados

Uma das principais partes quando se projeta algum software é o planejamento da organização dos fluxos de informação e dos componentes de serviço necessários para manter tudo funcionando de forma adequada. Para isso, é essencial a elaboração da arquitetura, isto é, uma representação visual capaz de descrever as partes que compõem o sistema e as relações que devem existir entre essas partes para garantir o funcionamento do todo. Como as aplicações que envolvem Inteligência Artificial e Machine Learning são, acima de tudo, recursos de software, o desenho da arquitetura é uma etapa fundamental do desenvolvimento.

No entanto, o que torna mais complexa a tarefa de projetar as arquiteturas desses sistemas é a existência de elementos além do código-fonte propriamente dito, no caso, os dados e os modelos. Os dados representam as diferentes informações que podem ser utilizadas como entrada para que os algoritmos de Aprendizado de Máquina possam passar por um processo de treinamento para aprender padrões considerados relevantes. Os modelos são os resultados do treinamento e juntamente com os dados são os componentes fundamentais de qualquer solução de Inteligência Artificial.

O desafio encontra-se justamente na elaboração de uma organização capaz de realizar a integração correta entre código, dados e modelos de forma escalável, eficiente e que proporcione uma boa experiência do usuário. Essa dificuldade torna-se ainda maior quando os profissionais da área de dados não possuem uma sólida base de conhecimento sobre arquiteturas e padrões de software. Diante desse cenário, profissionais e empresas se empenharam em idealizar arquiteturas que facilitassem as tarefas desses desenvolvedores. Dentre elas, as mais conhecidas são:

- Arquitetura Lambda: Proposta por Nathan Marz, surgiu para enfrentar os desafios no processamento de Big Data, combinando a eficiência no processamento de dados em tempo real e em lotes. Ela se estrutura em três camadas: a Batch Layer para o processamento de grandes volumes de dados armazenados, a Speed Layer para dados em tempo real e a Serving Layer para a visualização dos resultados. Esta abordagem permite análises abrangentes e em tempo real, tornando-se fundamental em aplicações de Big Data que exigem tanto a precisão quanto a rapidez.
- Arquitetura Kappa: Idealizada por Jay Kreps, co-criador do Apache Kafka, é uma simplificação da Arquitetura Lambda. Centrada exclusivamente no processamento de fluxo de dados, a Kappa elimina a necessidade de sistemas separados para processamento de dados em tempo real e em lotes. Em vez disso, utiliza um único sistema de processamento de fluxo para ambos os propósitos, simplificando a arquitetura e facilitando a manutenção e a escalabilidade em aplicações de streaming de dados.
- Arquitetura Delta: Desenvolvida pela Databricks, introduz o conceito de Delta Lake, visando melhorar os Data Lakes tradicionais. Combinando as características de Data Lakes e Data Warehouses, ela oferece um armazenamento confiável que suporta transações ACID e otimiza as consultas de dados. Esta arquitetura, conhecida como "Lakehouse", se destaca pela sua flexibilidade e escalabilidade de um Data Lake, junto com a gestão e eficiência de consulta de um Data Warehouse, sendo

particularmente útil para organizações que lidam com enormes volumes de dados variados.

A partir do desenvolvimento da arquitetura Delta, surgiu uma prática para gerenciar grandes volumes de dados em ambientes de armazenamento como Data Lakes. Essa abordagem foi desenvolvida para superar os desafios enfrentados por arquiteturas de dados tradicionais, proporcionando um gerenciamento de dados mais eficiente, confiável e escalável. Assim, a prática consiste na implementação de três camadas de armazenamento:

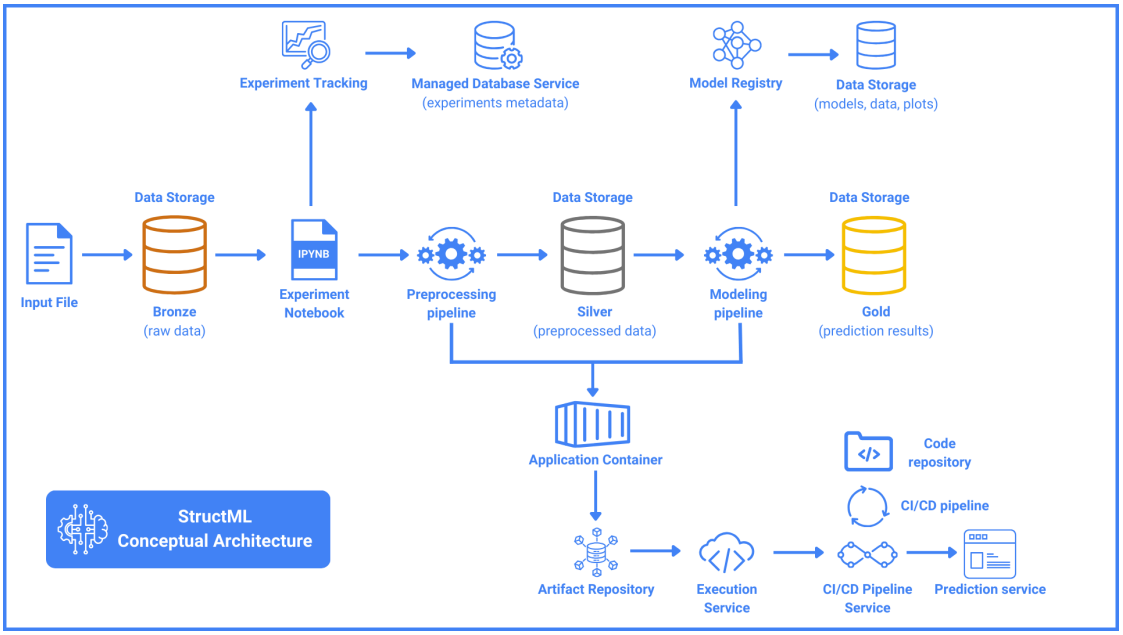
- Camada Bronze: Esta é a primeira camada de armazenamento na arquitetura Delta. Ela contém dados brutos, geralmente em seu formato original. Estes dados podem incluir registros de logs, arquivos de texto, dados de streaming brutos, etc. A ideia é que nesta camada, os dados sejam armazenados de maneira barata e eficiente, estando disponíveis para processamento e análise futuros.
- Camada Silver: Neste nível, os dados da camada Bronze são processados e refinados. Este processamento pode incluir a limpeza, a normalização, a deduplicação e a agregação de dados. A camada Silver armazena esses dados transformados, que estão em um estado mais útil para análises e insights de negócios, mas ainda podem não estar completamente otimizados para consultas de alto desempenho.
- Camada Gold: A camada final é onde os dados estão no seu estado mais refinado. Aqui, os dados são frequentemente estruturados e otimizados para suportar tipos específicos de consultas e relatórios, como dados de desempenho de negócios, métricas de KPIs, e relatórios agregados. Esta camada é ideal para usuários finais que precisam de acesso rápido e confiável a dados de alta qualidade para tomada de decisões.

Levando em consideração a dificuldade envolvida na tarefa de colocar modelos de ML em produção, este trabalho propõe uma arquitetura conceitual para que profissionais de dados sem experiência nesta tarefa possam ter um ponto de partida no desenvolvimento dos seus projetos. Vale ressaltar que a arquitetura proposta tem como principal inspiração a arquitetura Delta anteriormente descrita juntamente com as camadas de armazenamento Bronze, Silver e Gold.

Arquitetura StructML

A arquitetura StructML é a representação proposta neste trabalho para a construção e implantação de sistemas que envolvam processamento de dados e modelos de Aprendizado de Máquina. Ela foi criada pensando em profissionais que estão começando na área de MLOps e precisam de um caminho mais claro para desenvolver e aplicar modelos de ML. Com uma estrutura que facilita desde a entrada dos dados até a análise final, a StructML torna o processo menos intimidador e mais acessível para aqueles que não têm experiência prévia. É uma solução prática que ajuda a colocar modelos em funcionamento de maneira eficiente, guiando o usuário por cada etapa importante do MLOps.

Figura 1. Arquitetura Conceitual StructML



Fonte: próprio autor

Conforme a Figura 1 mostra, na arquitetura StructML, os dados iniciais são armazenados na camada Bronze. Após o pré-processamento, eles são movidos para a camada Silver, que contém dados prontos para análise. A modelagem é executada a seguir, e os resultados são salvos na camada Gold. A arquitetura também engloba funcionalidades para o acompanhamento de experimentos e o gerenciamento de modelos. Adicionalmente, dispõe de recursos para a containerização de aplicações, armazenamento de artefatos digitais, execução de serviços e um pipeline de integração e entrega contínua (CI/CD), que leva a um serviço de previsão. Todos esses elementos são integrados para fornecer uma plataforma eficiente para a operação de modelos de machine learning.

A StructML é uma arquitetura composta por elementos essenciais para o gerenciamento eficaz de projetos de Machine Learning. O armazenamento de dados é categorizado em três níveis: a camada Bronze guarda os dados brutos, a Silver contém dados que passaram pelo pré-processamento, e a Gold armazena os resultados finais das análises. Os Experiment Notebooks, como Jupyter Notebooks, são usados para experimentação e desenvolvimento de modelos, enquanto as Processing Pipelines cuidam do pré-processamento dos dados. O Model Registry e o Experiment Tracking são utilizados para gerenciar modelos e registrar detalhes dos experimentos, respectivamente.

Para garantir a consistência entre os ambientes de desenvolvimento e produção, a StructML utiliza Application Containerization, empacotando aplicações e seus ambientes. Os artefatos gerados, como modelos e gráficos, são armazenados no Artifact Repository. A execução e

operação dos modelos são realizadas pelo Execution Service. A integração e entrega contínuas de código e modelos atualizados são automatizadas pelo CI/CD Pipeline Service, enquanto o Code Repository permite a armazenagem e versionamento colaborativo do código-fonte. Por fim, o Prediction Service disponibiliza as funcionalidades do modelo treinado para fazer previsões e entregar resultados.

Tabela 1. Detalhamento dos componentes da StructML

Nome do Componente	Função	Ferramenta	Serviço (GCP)	Serviço (AWS)	Serviço (Azure)
Data Storage	Armazenamento de diferentes tipos de dados	-	Cloud Storage	S3	Data Lake Storage
Experiment Notebook	Exploração e experimentação de dados	Jupyter Notebooks, RStudio	Colab Entreprise Vertex AI	SageMaker Notebooks	Azure Notebooks
Processing Pipelines	Pipelines para pré-processamento e modelagem de dados	Scikit-learn, TensorFlow, PyTorch	Dataflow, Vertex AI	Data Pipeline, Glue, SageMaker	Data Factory, Machine Learning
Model Registry	Registro e versionamento de modelos de ML	MLflow, DVC	Vertex AI Model Registry	SageMaker Model Registry	Machine Learning Model Registry
Experiment Tracking	Rastreamento de experimentos e metadados	MLflow, TensorBoard	Vertex AI Experiments	SageMaker Experiments	Azure Machine Learning
Application Containerization	Criação de containers para ambientes de execução	Docker, Kubernetes	Container Registry	Elastic Container Service (ECS), Elastic Kubernetes Service (EKS)	Container Instances, Kubernetes Service (AKS)
Artifact Repository	Armazenamento de artefatos de modelos	Docker Hub, JFrog Artifactory	Artifact Registry	Elastic Container Registry (ECR)	Container Registry

Execution Service	Execução e hospedagem de modelos em ambientes de produção	-	Cloud Run, Kubernetes Engine	Lambda, ECS, EKS	Functions, Container Instances, AKS
CI/CD Pipeline Service	Automatização de integração, testes e entrega	Jenkins, GitLab CI, CircleCI	Cloud Build	CodePipelin e, CodeBuild	Azure Pipelines
Code Repository	Gerenciamento de código fonte	Git, Subversion	Cloud Source Repositories	CodeCommit	Azure Repos
Prediction Service	Interface para interação com modelos de ML	Flask, Dash, Streamlit	Vertex AI Predictions	SageMaker Endpoints	Machine Learning Endpoints

Fonte: próprio autor

A Tabela 1 traz um guia resumido sobre a arquitetura de MLOps proposta, definindo a função de cada componente, mostrando ferramentas e/ou frameworks possíveis de serem utilizados e os serviços correspondentes de cada uma das plataformas de nuvem mais utilizadas na atualidade. O objetivo deste guia é orientar desenvolvedores e gestores de projetos de Machine Learning sobre como a arquitetura conceitual detalhada anteriormente pode ser implementada na prática utilizando diferentes soluções disponíveis no mercado. Dessa forma, a nível de utilização em projetos reais, a arquitetura StructML cumpre com os princípios do MLOps ao proporcionar:

- Integração e Modularidade: Facilita o gerenciamento de diferentes estágios do pipeline de ML, desde o armazenamento de dados até a produção, devido à sua estrutura modular.
- Rastreabilidade e Versionamento: Oferece rastreabilidade completa dos modelos e experimentos com o uso de Experiment Tracking e Model Registry, o que é vital para auditoria e conformidade.
- Agilidade no Ciclo de Desenvolvimento: A utilização de pipelines de CI/CD e containerização permite atualizações rápidas e consistentes de modelos e aplicações, reduzindo o tempo de colocação no mercado.
- Flexibilidade de Implantação: Com a containerização de aplicações e serviços de execução, a StructML suporta a implantação flexível em várias plataformas e ambientes, facilitando a escalabilidade e a gestão de infraestrutura.

Porém, como toda proposição de organização de sistemas, ela está sujeita a pontos que necessitam de atenção por parte do desenvolvedor que a está implementando como:

- Sobrecarga de Gerenciamento de Dados: A separação em múltiplas camadas de armazenamento pode levar a uma sobrecarga de gerenciamento de dados, exigindo etapas adicionais de coordenação e transferência de dados entre as camadas.
- Integração de Componentes: A necessidade de integrar diversas ferramentas e plataformas distintas pode resultar em complexidade adicional, especialmente quando se tratam de atualizações e compatibilidade entre diferentes versões dessas ferramentas.
- Desafios de Performance em Escala: Embora a arquitetura seja projetada para escalar, pode haver desafios em manter a performance e eficiência quando o volume de dados ou a complexidade dos modelos aumenta significativamente, exigindo otimizações constantes.

Portanto, para comprovar a aplicabilidade da organização proposta, foi escolhido um estudo de caso de Machine Learning relacionado à análise de risco de crédito a partir de dados de clientes que realizaram requisições de empréstimos a uma determinada instituição. Na aplicação em questão a arquitetura StructML é implementada utilizando a linguagem de programação, alguns frameworks de Ciência de Dados e Machine Learning e os serviços disponibilizados pela Google Cloud Platform.

Estudo de Caso

O estudo de caso escolhido para que a arquitetura StructML fosse implementada aborda uma aplicação de Machine Learning na análise de risco de crédito. Este processo é fundamental em diversos setores como instituições financeiras, empresas de cartão de crédito, agências de crédito, empresas de locação e fintechs de empréstimos online. A aplicação desenvolvida visa permitir que usuários insiram seus dados através de uma interface e recebam uma decisão sobre a aprovação ou rejeição de crédito.

A metodologia de desenvolvimento adotada baseia-se no modelo CRISP-DM, um processo estruturado que orienta a abordagem sistemática para solucionar problemas complexos por meio de dados. O estudo começa com a compreensão do negócio, focando no equilíbrio entre atender às necessidades dos solicitantes de empréstimos e mitigar os riscos para as instituições financeiras. Em seguida, passa pela compreensão dos dados, preparação dos dados, modelagem, avaliação e, finalmente, desenvolvimento do sistema de apoio à decisão.

Este estudo de caso não só aborda o problema de negócio da análise de crédito, envolvendo múltiplos stakeholders como solicitantes, instituições financeiras, analistas de crédito, reguladores e investidores, mas também detalha a compreensão dos dados utilizados, incluindo informações como idade, renda anual, tipo de residência, tempo de emprego, intenção do empréstimo, valor do empréstimo, taxa de juros, status do empréstimo, histórico de não pagamentos e tempo de histórico de crédito (Tabela 2).

Tabela 2. Dicionário de dados do estudo de caso

Nome do atributo	Descrição

Idade	Idade do solicitante
Renda anual	Renda anual do solicitante (reais)
Tipo residência	Tipo da residência do solicitante (ALUGUEL, PRÓPRIA, HIPOTECA, OUTRO)
Tempo no emprego	Tempo que o solicitante está no emprego atual (anos)
Intenção de empréstimo	A intenção do solicitante com relação ao empréstimo (PESSOAL, EDUCAÇÃO, MÉDICO, INVESTIMENTO, REFORMA, PAGAMENTO DE DIVIDAS)
Valor do empréstimo	Valor do empréstimo solicitado (reais)
Taxa de juros	Taxa de juros para o empréstimo (percentual)
Status do empréstimo	Negado (0) ou Aprovado (1)
Histórico não pagamentos	Se existe histórico de não pagamentos do solicitantes (S ou N)
Tempo histórico de crédito (anos)	Quanto tempo o solicitante possui de histórico de crédito

Fonte: próprio autor

No estudo de caso sobre análise de risco de crédito, a arquitetura StructML foi aplicada utilizando uma combinação de frameworks da linguagem Python e serviços do Google Cloud Platform (GCP) para demonstrar um fluxo de trabalho de Machine Learning completo e integrado. A imagem ilustra como diferentes componentes da arquitetura interagem para processar dados, treinar modelos e servir previsões, destacando a viabilidade prática da StructML.

Conforme a Figura 2, o processo começa com a ingestão de dados brutos, representados pelo arquivo CSV, que são armazenados na camada Bronze do Cloud Storage, servindo como o ponto de entrada de dados. Esses dados são então processados através de uma pipeline de pré-processamento utilizando o PyCaret, uma ferramenta automatizada de ML, resultando em dados limpos e transformados que são armazenados na camada Silver do Cloud Storage. O próximo passo é a modelagem, onde o PyCaret é novamente utilizado para treinar e avaliar modelos de ML. Os modelos gerados e os insights são registrados e mantidos no Model Registry do MLflow, com metadados de experimentos armazenados no Cloud SQL e artefatos adicionais armazenados no Cloud Storage.

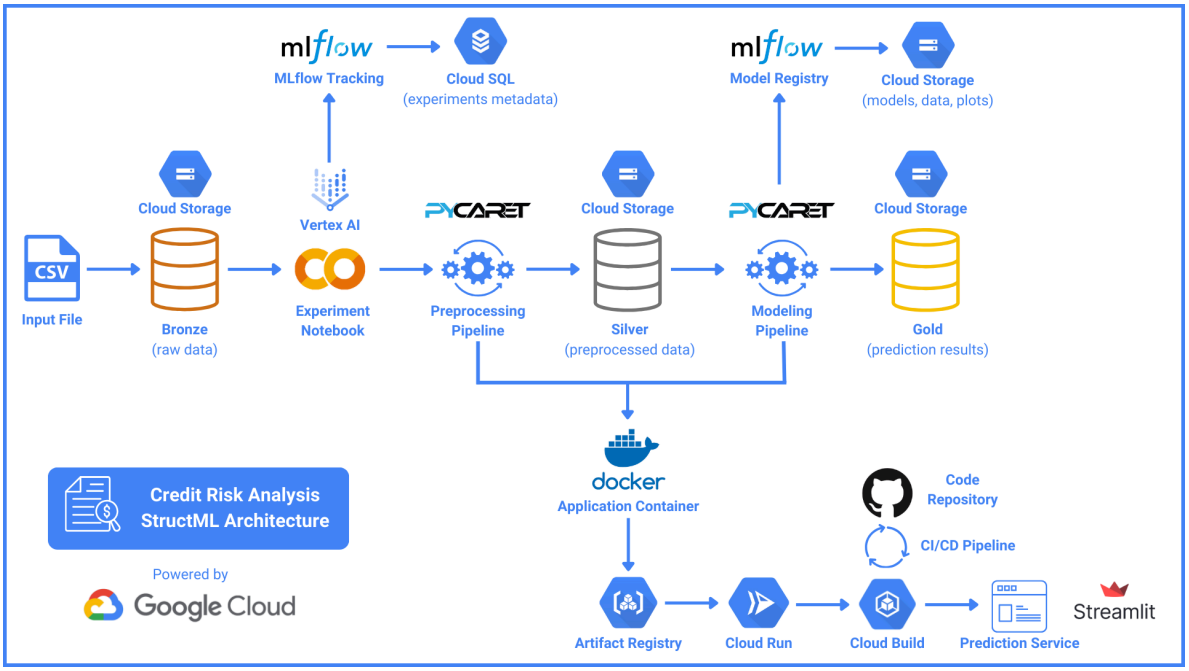
A implementação faz uso de Docker para containerização, empacotando a aplicação de ML para garantir a consistência em diferentes ambientes e facilitar a implantação. Os containers são gerenciados pelo Artifact Registry, e o serviço de execução é realizado pelo Cloud Run, que oferece uma plataforma gerenciada para executar containers escaláveis. O ciclo de desenvolvimento é

suportado por um pipeline de CI/CD fornecido pelo Cloud Build, que automatiza a integração e a entrega de aplicações atualizadas.

Finalmente, a aplicação de ML é disponibilizada aos usuários finais através de um serviço de previsão construído com Streamlit, fornecendo uma interface amigável para interação do usuário. Isso permite que os usuários insiram seus dados e recebam avaliações de risco de crédito em tempo real, com a aplicação do modelo treinado.

A integração desses componentes na arquitetura StructML, todos facilitados pelos serviços do GCP, demonstra não só a capacidade da arquitetura de suportar um processo de MLOps completo, mas também seu potencial para ser replicada e adaptada em diferentes contextos de negócios. O uso dessa arquitetura em um estudo de caso real valida sua aplicabilidade e robustez, reforçando a ideia de que StructML pode ser uma fundação sólida para projetos de ML em diferentes domínios.

Figura 2. Arquitetura StructML implementada no estudo de caso



Fonte: próprio autor

Referências

- https://pycaret.gitbook.io/docs/
- https://towardsdatascience.com/easy-mlops-with-pycaret-mlflow-7fbcbf1e38c6
- https://towardsdatascience.com/deploy-machine-learning-app-built-using-streamlit-and-pycaret-on-google-kubernetes-engine-fd7e393d99cb
- https://www.restack.io/docs/mlflow-knowledge-mlflow-pycaret-integration
- https://github.com/pycaret/pycaret-streamlit-google
- https://www.restack.io/docs/mlflow-knowledge-mlflow-on-gcp-integration

- https://moez-62905.medium.com/simplify-mlops-with-pycaret-mlflow-and-dagshub-366c768f0dac
- https://dlabs.ai/blog/a-step-by-step-guide-to-setting-up-mlflow-on-the-google-cloud-platform/
- https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning?hl=pt-br

Interface Web do Estudo de Caso

Interface MLflow