



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
CAMPUS FORTALEZA
DIRETORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
TELECOMUNICAÇÕES**

FELIPE DE BRITO VIANA

**DEGRADAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA EM
VISÃO COMPUTACIONAL**

FORTALEZA

2024

FELIPE DE BRITO VIANA

DEGRADAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA EM
VISÃO COMPUTACIONAL

Projeto de dissertação apresentado ao Programa de Pós-graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará como requisito parcial para a obtenção do título de mestre em Engenharia de Telecomunicações. Área de concentração: Processamento Digital de Sinais.

Prof. Dr. Auzuir Ripardo de Alexandria

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- V614d Viana, Felipe de Brito.
Degradação de modelos de aprendizado de máquina em visão computacional / Felipe de Brito Viana. - 2024.
133 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações,
Campus Fortaleza, 2024.
Orientação: Prof. Dr. Auzuir Ripardo de Alexandria.
Coorientação: Prof. Dr. João Alexandre Lobo Marques.
1. Degradação de Modelos. 2. Machine Learning. 3. Visão Computacional. 4. Data Drift e Concept Drift.
5. Dataset MURA. I. Título.

CDD - 621.382



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
Av. 13 de maio, 2081 - Bairro Benfica - CEP 60040-531 - Fortaleza - CE - www.ifce.edu.br

AVALIAÇÃO

Processo: 23256.014876/2023-31

Interessado: Felipe de Brito Viana

FELIPE DE BRITO VIANA

DEGRADAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA EM VISÃO COMPUTACIONAL

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus Fortaleza*, como requisito parcial para obtenção do Título de Mestre em Engenharia de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Aprovado em: **13 / 11 / 2023**

BANCA EXAMINADORA

Prof. Dr. **Auzuir Ripardo de Alexandria** (orientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. **João Alexandre Lobo Marques** (coorientador)

University of Saint Joseph

Profa. Dra. **Jéssyca Almeida Bessa**

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Prof. Dr. **Sergio Silva Ribeiro**

University of Crandall, Moncton, Canadá



Documento assinado eletronicamente por **Auzuir Ripardo de Alexandria, Professor do Ensino Básico, Técnico e Tecnológico**, em 22/11/2023, às 21:42, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **JOAO ALEXANDRE LOBO MARQUES, Usuário Externo**, em 28/02/2024, às 09:52, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Sergio Silva Ribeiro, Usuário Externo**, em 07/03/2024, às 10:33, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Jessyca Almeida Bessa, Professora do Ensino Básico, Técnico e Tecnológico**, em 22/03/2024, às 09:36, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade do documento pode ser conferida no site https://sei.ifce.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 informando o código verificador **5529373** e o código CRC **1B62E719**.

Agradecimentos

Primeiramente, quero expressar minha profunda gratidão a Deus, a força maior que guiou cada passo da minha jornada acadêmica, proporcionando-me saúde, determinação e serenidade nos momentos mais desafiadores.

À minha família, minha base sólida, que me ofereceu suporte, amor e compreensão em todos os momentos, sem os quais eu não teria a perseverança necessária para alcançar este marco em minha vida.

Um agradecimento especial ao meu orientador, Dr. Auzuir Ripardo de Alexandria, cuja dedicação, paciência e didática não apenas enriqueceram este trabalho, mas também contribuíram imensamente para meu crescimento pessoal e acadêmico. Sua orientação sábia e constante apoio foram fundamentais para a concretização deste projeto.

A todos vocês, minha eterna gratidão.

Resumo

A degradação de modelos de aprendizado de máquina (ML, *Machine Learning*) em visão computacional constitui o foco principal desta dissertação, voltando-se intensamente para a questão do *Data Drift* e *Concept Drift*. Em um cenário onde o crescimento e a aplicação de ML prosperam rapidamente, o objetivo é avaliar, através de experimentos em visão computacional, as técnicas de detecção de degradação, fundamentando-se firmemente em objetivos específicos que circulam desde a coleta de dados via revisão bibliográfica até a aplicação prática de detectores de *data drift* e *concept drift*. A motivação está firmemente na atualidade e aplicabilidade generalizada de ML, destacando uma preocupação essencial com a eficácia e adequação contínuas de modelos em um mercado que demanda de forma crescente por aplicações no campo da inteligência artificial. A pesquisa é justificada pela importância dos modelos de ML na atualidade e pela necessidade crítica de monitoramento e ajustes destes para garantir a sua eficiência e aplicabilidade a longo prazo. O objetivo desta dissertação é avaliar dentro do contexto de ML e MLOps a degradação de modelos de aprendizado de máquina em visão computacional exemplificando por meio de experimentos as técnicas de detecção de degradação. A fundamentação teórica tem foco em *machine learning*, técnicas de aprendizado, redes neurais convolucionais e MLOps, onde é explorada técnicas de monitoramento de *drift*. No contexto dos materiais e métodos, o *dataset MURA* é utilizado como base para o experimento, plataformas como *Pytorch* e *Deepchecks* são utilizadas para execução do modelo de classificação em visão computacional e o monitor de *drift* respectivamente. Os resultados alcançados são destacados pelo entendimento e análise dos impactos de diferentes níveis de ruído na ordem de 0,005, 0,03, 0,05 e 0,07 do tipo *salt and pepper* produziram *drift's* com valores 6,8802%, 19,8075%, 43,7801%, 52,3139% e 63,7912% respectivamente. Concluímos que a aplicabilidade de plataformas na criação e monitoramento de modelos é de grande relevância para os desafios enfrentados na criação e manutenção dos produtos desenvolvidos na área de aprendizado de máquina.

Palavras-chaves: Degradação de Modelos, *Machine Learning*, Visão Computacional, *data drift*, *concept drift*, *Dataset MURA*.

Abstract

The degradation of machine learning (ML) models in computer vision is the primary focus of this dissertation, heavily turning towards the issue of Data Drift and Concept Drift. In a scenario where the growth and application of ML prosper rapidly, the goal is to evaluate, through experiments in computer vision, degradation detection techniques, firmly grounding on specific objectives that circulate from data collection via literature review to the practical application of data drift and concept drift detectors. The motivation is firmly rooted in the contemporaneity and widespread applicability of ML, highlighting an essential concern with the continuous efficacy and suitability of models in a market that increasingly demands applications in the field of artificial intelligence. The research is justified by the importance of ML models at present and the critical need for monitoring and adjusting them to ensure their efficiency and applicability in the long term. The aim of this dissertation is to evaluate, within the context of ML and MLOps, the degradation of machine learning models in computer vision, exemplifying through experiments the degradation detection techniques. The theoretical foundation focuses on machine learning, learning techniques, convolutional neural networks, and MLOps, where drift monitoring techniques are explored. In the context of materials and methods, the MURA dataset is used as the basis for the experiment and platforms like Pytorch and Deepchecks are used for the execution of the computer vision classification model and drift monitoring, respectively. The results obtained are highlighted by understanding and analyzing the impacts of different levels of noise in the order of 0.005, 0.03, 0.05, and 0.07 of the salt and pepper type produced drifts with values 6.8802%, 19.8075%, 43.7801%, 52.3139%, and 63.7912% respectively. We conclude that the applicability of platforms in creating and monitoring models is of great relevance for the challenges faced in creating and maintaining the products developed in the machine learning area.

Key-words: Model Degradation, Machine Learning, Computer Vision, Data Drift, Concept Drift, MURA Dataset.

Lista de ilustrações

Figura 1 – Representação Gráfica da Degradação de modelos de Aprendizado de Máquina	20
Figura 2 – Mapa mental ilustrando de forma macro a estrutura do trabalho	31
Figura 3 – Ilustração de um Classificador de <i>Spam</i> para E-mail.....	36
Figura 4 – Exemplo de entrada, processamento e saída de sinais em sistema biológico	38
Figura 5 – Ilustração de uma rede neural com uma camada intermediária.....	39
Figura 6 – Ilustração Neurônio Biológico e fluxo de Sinal	40
Figura 7 – Neurônio de McCulloch-Pitts	42
Figura 8 – Gráfico representando a função Sigmoide	44
Figura 9 – Rede Neural Convolucional etapa de extração de atributos.....	45
Figura 10 – Rede Neural Convolucional etapa de classificação	46
Figura 11 – Implementação dos Princípios e componentes técnicos.....	49
Figura 12 – Relacionamento entre DevOps e Qualidade de <i>software</i>	51
Figura 13 – Fluxo dos dados em Treinamento Contínuo de modelo de ML.....	53
Figura 14 – Exemplo de <i>Data Drift</i> , fotografias em estações diferente mesmo lugar	56
Figura 15 – Exemplo de <i>Data Drift</i> , fotografias em estações diferente mesmo lugar caatinga.....	57
Figura 16 – Ilustração <i>Concept Drift</i> em pontos de dados com limite de decisão.	58
Figura 17 – Tipos de <i>Concept Drift</i>	59
Figura 18 – <i>Framework</i> para detecção de <i>Concept Drift</i>	60
Figura 19 – Exemplo Gráfico de uma curva ROC.....	81
Figura 20 – <i>Drift Score</i> para todos os níveis de ruído	98
Figura 21 – <i>Train test label drift score</i>	100
Figura 22 – <i>Train test label drift comparativo distribuição</i>	101
Figura 23 – <i>Image Property Drift - RMS Contrast</i> comparativo distribuição	103
Figura 24 – <i>Image Property Drift Score - RMS Contrast</i>	104
Figura 27 – <i>Train Test Prediction Drift Score</i>	108
Figura 28 – <i>Train Test Prediction Drift</i> comparativo distribuição	109
Figura 29 – Comparativo <i>Image dataset drift</i> para todos os níveis de ruído	111
Figura 30 – Comparativo <i>Image Dataset Drift</i> raio-X.....	111
Figura 31 – Comparativo distribuições e impacto <i>Image Dataset Drift</i>	112
Figura 25 – <i>Image Property Drift - Brightness</i> comparativo distribuição.....	114
Figura 26 – <i>Image Property Drift Score - Brightness</i>	115

Lista de tabelas

Tabela 1 – Sumário dos artigos breve estado da arte.	25
Tabela 2 – Sumário dos dados dos experimentos.....	95
Tabela 3 – Valores de KS e <i>p-value</i> para percentual de ruído.	96

Lista de abreviaturas e siglas

ML	Machine Learning
MLOps	Machine Learning Operations
DEVOps	Development and Operations
CD	Concept Drift
DD	Data Drift
IDC	International Data Corporation
CNN	Convolutcional Neural Network
IA	Inteligência Artificial
MSE	Mean Square Error
CE	Cross Entropy
EMD	Earth Mover's Distance
KL	Kullback-Leibler
MLP	Multilayer Perceptron
RNC	Rede Neural Convolutcional
AR	Aspect Ratio
AL	Altura
LG	Largura
A	Área
I	Intensidade
BR	Brilho
IM	Intensidade Média
RI	Intensidade Vermelho
GI	Intensidade Verde

BI	Intensidade Azul
RMSC	Root Mean Square Contrast
MRRI	Intensidade Relativa Média do Vermelho
MGRI	Intensidade Relativa Média do Verde
MBRI	Intensidade Relativa Média do Azul
GPU	Graphics Processing Unit
PNG	Portable Network Graphics
MURA	Musculoskeletal Radiographs
EMD	Earth Mover Distance
PwC	Price water house Coopers
PIB	Producto Interno Bruto
CAGR	Compound Annual Growth
ICML	International Conference on Machine Learning
AAAI	Association for the Advancement of Artificial Intelligence
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
SGD	Stochastic Gradient Descent
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
IoT	Internet of Things
5G	Quinta geração de tecnologia de redes móveis
3D	Três dimensões
MNIST	Modified National Institute of Standards and Technology)
ETL	Extract, Transform, Load

Lista de símbolos

Σ	Soma
$ a $	Módulo de a
$\log$	Logaritmo com base 10
$\ln$	Logaritmo natural
ϑ	Letra grega Theta
φ	Letra grega Phi
σ	Letra grega Sigma
e	Número de Euler
$\int$	Integral Definida
$\bar{x}$	Média
M_d	Mediana
M_o	Moda
AT	Amplitude
cov	Covariância
var	Variância
V	Resultado teste de Cramer's V
$W_p(P, Q)$	Classificador pelo Domínio

Sumário

1	INTRODUÇÃO	16
1.1	Objetivos	19
1.1.1	Objetivo geral.....	19
1.1.2	Objetivos específicos.....	20
1.2	Breve Estado da Arte	20
1.2.1	Do CD à degradação do modelo: uma visão geral sobre detectores de desvio com reconhecimento de desempenho.	21
1.2.2	Uma pesquisa sobre aprendizado de máquina para recorrentes CD em <i>Data Stream</i>	22
1.2.3	Corrigindo DD e restabelecendo modelos de.....	22
1.2.4	Deteção de DD em um fluxo bidimensional usando o teste de Kolmogorov-Smirnov	23
1.2.5	Máquinas de vetores de suporte incremental-decremental ponderadas para CD com janela de deslocamento.....	23
1.2.6	Conclusão.....	26
1.3	Motivação	26
1.4	Justificativa	27
1.5	Organização da Dissertação	29
2	FUNDAMENTAÇÃO TEÓRICA.....	32
2.1	<i>Machine Learning</i>	32
2.1.1	Conceitos básicos.....	32
2.1.2	Técnicas de aprendizado de máquina.....	32
2.1.2.1	Redes Neurais Convolucionais (CNNs).....	33
2.1.2.2	Máquinas de Vetores de Suporte (SVMs).....	34
2.1.2.3	Árvores de Decisão e Florestas Aleatórias	34
2.1.2.4	Técnicas de Aprendizado Profundo	34
2.1.3	Pré-processamento de dados	34
2.1.4	Avaliação de modelos.....	34
2.1.5	Modelos de Classificação.....	35
2.1.5.1	Função de perda ou <i>Loss Function</i>	35
2.1.6	Redes Neurais Artificiais.....	38
2.1.6.1	Neurônio de McCulloch-Pitts	40
2.1.6.2	Redes Perceptron	41
2.1.6.3	Função de ativação Sigmoide.....	43

2.1.6.4	Neurônio com função de ativação Sigmoid	43
2.1.7	Redes Neurais Convolucionais	44
2.1.7.1	Conceitos básicos de redes neurais convolucionais	44
2.1.7.2	Arquitetura e funcionamento das camadas convolucionais	45
2.1.7.3	Treinamento e ajuste dos parâmetros da rede	47
2.1.7.4	Transferência de aprendizado e pré-treinamento	48
2.2	<i>Machine Learning Operations - MLOps</i>	48
2.3	<i>Data Drift</i>	54
2.4	<i>Concept Drift</i>	55
2.5	Modelos de Visão computacional	59
2.5.1	Propriedades da Imagem	60
2.5.2	Arquiteturas CNN's	62
2.5.2.1	LeNet-5	62
2.5.2.2	AlexNet	62
2.5.2.3	VGGNet	63
2.5.2.4	ResNet	63
2.5.2.5	DenseNet	63
2.6	Considerações Finais	64
3	MATERIAL E MÉTODOS	65
3.1	Base de dados	65
3.1.1	Descrição do <i>Dataset</i> MURA	65
3.2	Tecnologias	66
3.2.1	Python	66
3.2.2	Pytorch	67
3.2.3	<i>Deepchecks</i>	68
3.2.4	Colab	68
3.3	Métricas	69
3.3.1	Métricas Univariadas e Multivariadas	69
3.3.1.1	Métricas Univariadas	69
3.3.1.2	Métricas multivariadas	71
3.3.2	Teste de Cramer's V	73
3.3.3	Wasserstein ou <i>Earth Movers Distance</i>	75
3.3.4	O teste de Kolmogorov-Smirnov	76
3.3.4.1	Funcionamento do Teste K-S	77
3.3.5	Deteção pelo Domínio do Classificador	78
3.3.6	Métricas usadas em Classificação	79
3.3.7	Exemplos de aplicação das Métricas Univariadas e Multivariadas	82
3.3.7.1	Métricas Univariadas e seus Casos de Uso	82

3.3.7.2	Métricas Multivariadas e seus Casos de Uso	82
3.3.8	Exemplos de Métricas de Classificação e seus Casos de Uso.....	83
3.4	O modelo de Classificação	84
3.5	Testes selecionados.....	84
3.6	<i>Dataset</i> selecionados.....	85
3.6.1	Separando o <i>Dataset Elbow</i> MURA em treino e teste.	86
3.7	Densenet selecionada.	87
3.7.1	Características Técnicas DenseNet121.	88
3.8	Escolha do <i>Deepchecks</i>	89
3.9	Simulando degradação dos <i>dataset</i>.....	89
3.9.1	Falhas em máquinas de raio-X.	90
3.9.2	<i>Salt and Pepper</i>	91
3.9.2.1	Escolha do nível de ruído <i>Salt and Pepper</i>	92
3.10	Experimento.....	93
3.11	Considerações Finais.....	93
4	RESULTADOS E DISCUSSÕES	94
4.1	Definição dos limites de <i>drift</i> no <i>deepchecks</i>.....	94
4.2	Resultados Gerais	95
4.3	Resultados <i>Train Test Label Drift</i>.....	99
4.4	Resultados <i>Image Property Drift</i>	102
4.5	Resultados <i>Train Test Prediction Drift</i>.....	107
4.6	Resultados <i>Image Dataset Drift</i>.....	110
5	CONCLUSÕES	116
5.1	Trabalhos Futuros	117
	REFERÊNCIAS.....	119

1 Introdução

Modelos de Aprendizado de máquina estão presentes em grande parte das aplicações e produtos de tecnologia atualmente (CHOE et al., 2023; DU et al., 2023a; RAO et al., 2023). Em alguns casos são tecnologias invisíveis ao usuário médio. As aplicações em inteligência artificial (IA) tem prosperado com muitas soluções práticas e diversos tópicos de pesquisa (GOODFELLOW; COURVILLE, 2016a).

A inteligência desenvolvida e aplicada por meio de software para automatizar as atividades de rotina, compreensão de fala (LIU et al., 2023), reconhecimento de imagens (ZHANG et al., 2022a), diagnósticos em medicina (HUTTUNEN et al., 2022), apoio a pesquisa científica (YANG, 2022) plataformas como telecomunicações (HIJJI et al., 2023), sistema financeiro (LIU et al., 2022b), agricultura (WANG et al., 2022), medicina (ZHANG et al., 2022b), direito (BARRIENTOS et al., 2022), segurança pública (BOLKIAH et al., 2022), mobilidade urbana (GUPTA et al., 2022), indústria (LIU et al., 2022a) e outros mais que tornam essa lista quase incontável (BISHOP, 2006b).

Assim a necessidade de operacionalizar e manter com eficiência modelos de aprendizado de máquina é essencial. A escalada do uso aplicações que utilizam de aprendizado de máquina só é possível graças a evolução dos computadores, do aumento significativo na produção e armazenamento de dados e da evolução dos próprios modelos matemáticos e algoritmos (OSIPOV, 2022).

A área de aprendizado de máquina se beneficiou de forma significativa na massificação do uso de modelos de *Machine Learning* (ML) nas mais diversas áreas. Cada área possui demandas específicas com características e desafios próprios. Portanto operacionalizar modelo de ML é fundamental para o sucesso desses projetos (TREVEIL; TEAM, 2020).

O aprendizado de máquina, com sua ampla adoção, criou a necessidade de uma abordagem sistemática e eficiente para a construção de sistemas de ML. Soluções na área de desenvolvimento de software estão disponíveis para desenvolvimento e operação de aplicações, *Development Operations* (DevOps) (GIFT, 2021). Nas disciplinas de engenharia, os padrões de projeto capturam as melhores práticas e soluções para problemas comuns (WATSON; CUTTING, 2022). Codificar conhecimento e experiência em boas práticas, técnicas e ferramentas que profissionais da área de aprendizado de máquina podem seguir é fundamental para garantir uma escalabilidade saudável (LAKSHMANAN; MUNN, 2021).

A evolução das tecnologias é dinâmica bem como novos dados são gerados a cada instante. Esses novos dados podem representar novos objetos no mundo real ou novas

características de objetos já existentes. Como modelos de aprendizado de máquina são programas de computadores que aprendem com os dados (GERON, 2019) esse aprendizado pode ficar defasado ou equivocado para o caso em que o aprendizado diverge do cenário representado pelos dados atuais. Nesse contexto é possível falar em degradação do modelo ou *model decay*. *Machine learning Operations* (MLOps), possui princípios dedicados a monitorar e avaliar a performance dos modelos com objetivo de identificar e mitigar os impactos da degradação ao longo do tempo. Ao longo do tempo novos dados são produzidos e os atributos utilizados para classificar um objeto podem mudar, por exemplo.

MLOps é uma prática emergente que combina ML, DevOps e engenharia de dados para automatizar e aprimorar o ciclo de vida de desenvolvimento e operacionalização de modelos de ML. Esse conjunto de práticas, metodologias e tecnologias visa facilitar a colaboração entre cientistas de dados e profissionais de operações para implementar e manter modelos de ML de forma eficaz e eficiente em ambientes de produção. A importância do MLOps reside na sua capacidade de acelerar o tempo de entrega, melhorar a qualidade e a confiabilidade dos modelos de ML, além de promover uma gestão eficiente do ciclo de vida dos modelos (AMEISEN, 2020).

Um dos desafios críticos no ciclo de vida de um modelo de ML é o fenômeno conhecido como *drift* do modelo, que pode ocorrer imediatamente após o *deploy* devido a falhas em seu desenvolvimento, bem como ao longo do tempo, devido a mudanças no ambiente de dados ou no comportamento subjacente que o modelo tenta prever. O *drift* do modelo pode significativamente deteriorar a performance dos modelos de ML, levando a decisões baseadas em previsões imprecisas ou irrelevantes (BISHOP, 2006b).

Dentro do contexto de MLOps, a aplicação de detectores de *drift* e a implementação de estratégias de monitoramento e atualização automáticas são fundamentais para um desenvolvimento saudável de modelos de ML. Estas práticas permitem a identificação e correção oportuna do *drift* do modelo, garantindo que os modelos permaneçam precisos e relevantes ao longo do tempo. Além disso, o MLOps promove a integração e entrega contínuas (CI/CD) de modelos de ML, facilitando a rápida adaptação e reimplantação de modelos atualizados em resposta a detecções de *drift* (ZHENG,).

A adoção de princípios e métodos de MLOps não apenas melhora a eficiência operacional, mas também minimiza riscos associados à degradação do desempenho dos modelos de ML, assegurando que as soluções baseadas em ML sejam sustentáveis, confiáveis e alinhadas às necessidades em constante mudança do negócio. Portanto, a integração efetiva do MLOps no ciclo de vida de desenvolvimento de ML é essencial para enfrentar o desafio do *drift* do modelo, promovendo um ecossistema de ML robusto, escalável e adaptável (AMEISEN, 2020).

Novos dados podem agora sugerir uma nova classe ou referenciar a outro objeto.

Portanto é necessário e fundamental o monitoramento baseado em critérios bem definidos para identificar essas mudanças (YOUNG; STEELE, 2022). Na Figura 1 há a representação gráfica da evolução positiva da performance do modelo de ML na fase de treino e aprendizagem. Já no ambiente de produção podemos observar o ponto de degradação indicando que a performance está reduzindo com o passar do tempo.

O *Concept Drift* (CD) tem origem em diversos contextos. Este pode surgir na degradação da qualidade dos materiais, dos equipamentos, do sistema, sazonalidade, mudança de preferências. No mundo real modelos de aprendizado de máquina precisam lidar com sistemas não estacionários. CD refere-se a uma situação em que a relação entre os dados de entrada e a saída desejada muda ao longo do tempo em um problema de aprendizado de máquina. Em outras palavras, é uma alteração nas distribuições dos dados subjacentes após o modelo ter sido treinado, tornando o modelo possivelmente desatualizado ou menos preciso. CD é especialmente relevante em sistemas que operam em ambientes dinâmicos, onde os padrões subjacentes podem mudar com o tempo. Por exemplo, um modelo de detecção de fraude treinado com dados de transações de um ano atrás pode não funcionar bem hoje, se os padrões de fraude tiverem mudado. Desta forma pesquisadores desenvolveram várias técnicas, desde re-treinamento regular dos modelos com novos dados até abordagens mais complexas que detectam e se adaptam automaticamente às mudanças. Detectar o CD pode ser tão crucial quanto lidar com ele, uma vez que uma detecção prévia pode ajudar a manter a performance e a relevância de um modelo em operação (COLLEDANI; ANSELMINI; ROBUSTO, 2023).

A ausência de monitoramento automático para detecção e correção de drift em modelos de Machine Learning (ML) pode ter impactos significativos, tanto em termos de custos quanto de riscos à vida, especialmente em aplicações críticas (AHMED et al., 2023). Aqui estão alguns dos maiores impactos relacionados:

- Custos Financeiros Elevados.

Reparação de Erros: Sem monitoramento e correção proativos, os erros gerados por modelos desatualizados podem acumular custos substanciais antes de serem detectados e corrigidos. Isso inclui o custo de identificar a origem do problema, re-treinar o modelo e implementar a solução.

Decisões Baseadas em Dados Incorretos: Decisões automatizadas baseadas em previsões imprecisas podem levar a ações ineficazes ou erradas, resultando em perdas financeiras diretas, desperdício de recursos e oportunidades perdidas.

- Riscos à Vida e à Segurança.

Aplicações na Saúde: Em contextos médicos e de saúde, um modelo de ML que não se adapta a mudanças nos padrões de doenças ou na eficácia do tratamento

pode levar a diagnósticos errados ou tratamentos ineficazes, colocando vidas em risco.

Sistemas de Transporte Autônomo: Nos sistemas de veículos autônomos, a incapacidade de se adaptar a mudanças nas condições de tráfego, ambiente ou regulamentações pode aumentar o risco de acidentes.

- **Riscos Legais e de Reputação.**

Não Conformidade com Regulamentações: Em muitos setores, especialmente em finanças e saúde, existem regulamentações rigorosas sobre a precisão dos sistemas automatizados. A falha em monitorar e corrigir *drift* pode levar a violações regulatórias, multas e outras sanções legais.

Dano à Reputação: A falha em manter a precisão dos modelos de ML pode prejudicar a reputação de uma organização, afetando a confiança dos clientes e parceiros e podendo levar a uma perda de negócios.

- **Impacto em Sistemas Críticos**

Infraestrutura Crítica e Segurança Pública: Em sistemas críticos, como monitoramento de redes de energia, sistemas de alerta de desastres naturais e segurança pública, a falha em detectar e corrigir *drift* em tempo hábil pode ter consequências catastróficas, incluindo falhas de sistema, desastres não mitigados e ameaças à segurança pública.

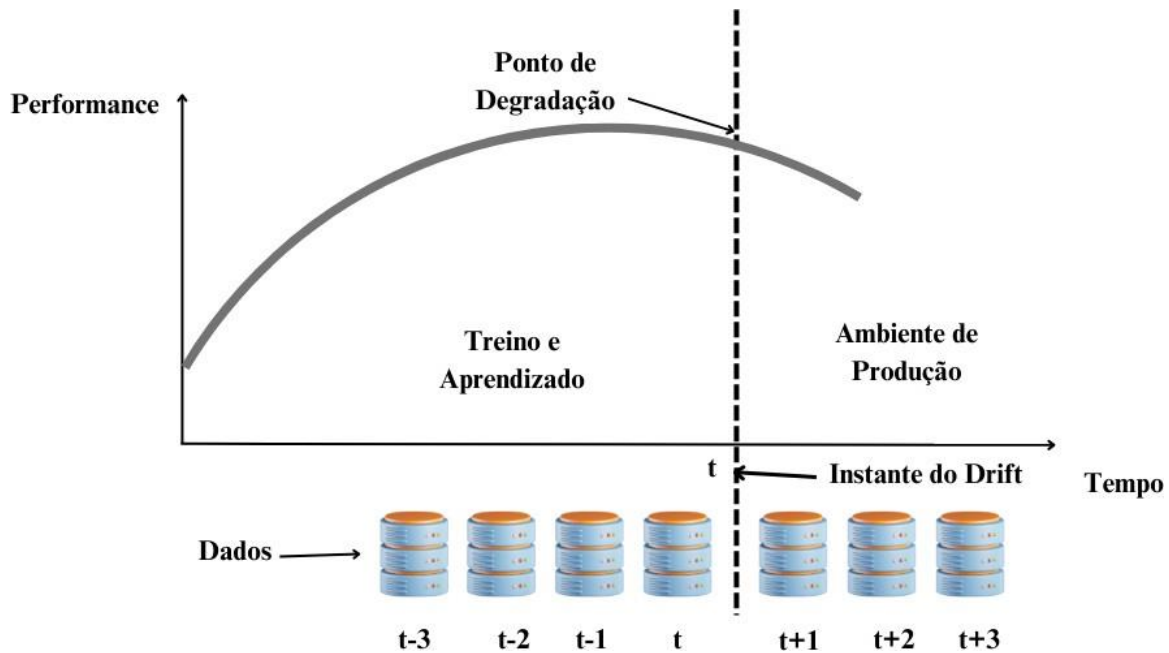
A ausência de monitoramento automático e correção de *drift* em modelos de ML pode ter implicações profundas e de longo alcance, destacando a importância de integrar sistemas de detecção de *drift* e mecanismos de atualização automática, especialmente em aplicações onde a precisão é crítica para a segurança, saúde e bem-estar das pessoas, além da sustentabilidade financeira e operacional das organizações (SHAKER; HÜLLERMEIER, 2015; ATBAN; EKINCI; GARIP, 2023) . Implementar estratégias de monitoramento contínuo e adaptação não só minimiza riscos e custos mas também garante a relevância e eficácia contínua dos sistemas de ML.

1.1 Objetivos

1.1.1 Objetivo geral

Avaliar dentro do contexto de ML e MLOps a degradação de modelos de aprendizado de máquina em visão computacional exemplificando por meio de experimentos as técnicas de detecção de degradação.

Figura 1 – Representação Gráfica da Degradação de modelos de Aprendizado de Máquina



Fonte: Adaptado de [Bayram, Ahmed e Kassler \(2022\)](#).

1.1.2 Objetivos específicos

- Coletar por meio de revisão de obras e trabalhos acadêmicos informações e conhecimento sobre ML e MLOps.
- Identificar as estratégias de monitoramento de modelos por meio de detecção de degradação.
- Identificar os testes e métricas mais adequados para monitorar a degradação de um modelo de classificação de imagens.
- Aplicação de detectores de *data drift* (DD) e CD na validação da degradação de modelos de classificação de imagens.
- Analisar como os detectores de *drift* foram capazes de evidenciar a degradação do modelo.
- Discutir os resultados alcançados avaliando impactos e avanços.

1.2 Breve Estado da Arte

Nesta seção são descritos trabalhos relacionados à degradação de modelos de aprendizado de máquina.

1.2.1 Do CD à degradação do modelo: uma visão geral sobre detectores de desvio com reconhecimento de desempenho.

Bayram, Ahmed e Kassler (2022) apresentam em seu trabalho que, a realidade das aplicações de modelos de aprendizado de máquina é cercada de complexidades que de forma quase inevitável produz degradação e perda na capacidade dos modelos de ML realizarem aquilo ao qual foram desenvolvidos. Os sistemas são sujeitos a diversas interferências e mudanças. Em Bayram, Ahmed e Kassler (2022) a formulação de três perguntas foi realizada:

1. O por que da existência de diferentes termos para descrever CD?
2. Métodos de detecção de degradação com base em desempenho do modelo propostos nos últimos dez anos?
3. Validar o desempenho preditivo do modelo e descrever a utilização dessa técnica para detecção de desvios?

As perguntas formuladas por Bayram, Ahmed e Kassler (2022) foram avaliadas quanto a possibilidade de replicação neste trabalho. No entanto ao elaborar a presente dissertação foi necessário estabelecer prioridades e focar nos aspectos mais relevantes para o objetivo principal deste trabalho. Neste contexto, optou-se por não abordar conforme explicado abaixo:

1. A existência de diferentes termos para descrever CD é uma questão mais linguística e histórica do que técnica. Embora seja um ponto interessante, sua discussão detalhada não contribuiria diretamente para a compreensão da degradação dos modelos no âmbito da visão computacional.
2. Os métodos de detecção de degradação baseados no desempenho do modelo propostos nos últimos dez anos são vastos e variados. Em vez de abordar cada um deles, optou-se por focar nas técnicas mais relevantes e amplamente adotadas na área de visão computacional, garantindo assim uma discussão mais aprofundada e pertinente.
3. A validação do desempenho preditivo do modelo e a descrição da utilização dessa técnica para detecção de desvios são, sem dúvida, fundamentais. No entanto, este trabalho se concentra mais na identificação da degradação em si do que nas nuances específicas da validação preditiva. Acredita-se que a abordagem adotada oferece um equilíbrio mais adequado entre teoria e aplicação prática na temática da dissertação.

Assim, enquanto reconhecemos a importância desses tópicos, a decisão de não os incluir foi tomada para manter o trabalho conciso, focado e alinhado com os principais objetivos propostos.

Os autores [Bayram, Ahmed e Kassler \(2022\)](#) focam em degradação com base na análise da performance. Há métodos para detecção de CD. Métodos com base na distribuição, performance, múltiplas hipóteses e contexto, controle no processo estatístico, técnica de janelamento e aprendizagem conjunta.

CD e degradação da performance dos modelos de aprendizagem de máquina são dois fenômenos interligados em sistemas preditivos segundo [Bayram, Ahmed e Kassler \(2022\)](#). A existência de um implica na ocorrência do outro. Após a introdução a uma visão geral sobre CD, as principais causas foram discutidas, seguidas por definições pertinentes. Foram compilados diversos termos encontrados na literatura relacionados ao CD. Em seguimento, conforme apresentado por [Bayram, Ahmed e Kassler \(2022\)](#), é destacada a detecção de CD e as abordagens que monitoram a degradação da performance com o objetivo de identificar mudanças. Nota-se que esses métodos baseados em performance operam de uma forma peculiar: um mecanismo de alerta é acionado quando o CD atinge um determinado patamar, geralmente quando a performance decresce para um limite estabelecido.

1.2.2 Uma pesquisa sobre aprendizado de máquina para recorrentes CD em *Data Stream*.

[Suárez-Cetrulo, Quintana e Cervantes \(2023\)](#) apresentam uma pesquisa em literatura relevante para mudanças de comportamento em *Data Stream*. O estudo começa descrevendo mecanismos passivos ou ativos para adaptar ou detectar CD, desafios frequentes nesta área, e métricas de desempenho relacionadas. Diferentes abordagens supervisionadas e não supervisionadas, como online, ensembles, meta-learning e agrupamento baseado em modelo que podem ser usados para lidar com sazonalidades em um *Data Stream*. O autor tem por objetivo apontar novas tendências de pesquisa direcionando pesquisas futuras sobre o tema.

1.2.3 Corrigindo DD e restabelecendo modelos de .

[Ashok, Ezhumalai e Patwa \(2023\)](#) destacam a degradação da performance ao longo do tempo devido a vários motivos, causando comportamento imprevisto e problemas de desempenho. O *corona virus disease* - 2019, COVID-19 tem um impacto significativo nos negócios, resultando em grandes dispersões de dados com características instáveis. Dados com desvios significativos devido às mudanças em suas propriedades estatísticas, geram impacto em modelos existentes baseados em dados históricos. Assim as saídas dos

modelos podem gerar resultados ruins para os negócios. Antecipar todos os eventos futuros significativos e criar um modelo de ML que seja resiliente as mudanças é um desafio complexo. O autor reforça a necessidade de processo automatizado para monitoramento contínuo no ciclo de vida do ML para identificar os possíveis problemas usando metodologias de detecção de desvio com ações adequadas para evitar falsas previsões. Este estudo é focado em séries temporais onde o fluxo de dados é representado como observações em intervalos regulares. Muitos processos dinâmicos do mundo real podem ser modelados usando séries temporais em que os dados podem fluir rapidamente e mudar ao longo do tempo, o que é prejudicial para a análise e previsão de séries temporais. O trabalho gerou um estudo comparativo de diferentes abordagens para desvios de dados em modelos de séries temporais.

1.2.4 Detecção de DD em um fluxo bidimensional usando o teste de Kolmogorov-Smirnov

Porwik e Dadzie (2022) pontuam que nos últimos anos ocorre uma crescente na quantidade de informações via *streaming* oriundo de séries temporais. Aprender com os dados que aparecem em tempo real é um grande desafio, em parte devido à velocidade com que novos dados aparecem. Mudanças de dados ocultas que não são previamente conhecidas pelos algoritmos de aprendizado são referidas na literatura como DD ou CD. A precisão do classificador se deteriora devido ao DD, que ocorre em dados não estacionários. Em tais situações, o classificador deve detectar uma mudança significativa nos dados para adaptar sua previsão ao longo do tempo.

1.2.5 Máquinas de vetores de suporte incremental-decremental ponderadas para CD com janela de deslocamento

Galmeanu e Andonie (2022) analisam o problema de aprender sobre a distribuição dos dados à medida que esses mudam com o passar do tempo. Essa mudança, conhecida como CD, traz problema para o desempenho do modelo. As previsões perdem sua eficácia. As *Support Vector Machines* (SVMs) podem aprender instâncias de entrada ponderadas e também podem ser treinadas *online* (aprendizagem incremental-decremental). Combinando essas duas propriedades das SVMs, o problema proposto é definir um modelo de CD para SVM online com janela ponderada de deslocamento. O modelo SVM clássico deve ser retreinado do zero após cada mudança de janela. É apresentado o *Weighted Incremental-Decremental SVM* (WIDSVM), uma generalização do SVM incremental-decremental para janelas de deslocamento. O WIDSVM é capaz de aprender a partir de fluxos de dados com CD, usando a técnica de janela de deslocamento ponderada. Os experimentos em conjuntos de dados de CD artificiais e do mundo real mostram que a precisão da

classificação do WIDSVM melhora significativamente em comparação com um SVM sem janela de deslocamento.

Autores	Título	Ano	Palavras-chaves	Core do Artigo
Firas Bayram, Bestoun S. Ahmed, Andreas Kassler	<i>From concept drift to model degradation: An overview on performance-aware drift detectors</i>	2022	<i>Concept drift, Model degradation, Data stream, Machine learning</i>	Pesquisa ordenada e abrangente sobre terminologia e matemática dos tipos de <i>concept drift</i>
Andrés L. Suárez-Cetrulo, David Quintana, Alejandro Cervantes	<i>A survey on machine learning for recurring concept drifting data streams</i>	2023	<i>Regime change, Online machine learning, Data streams, Concept drift, Meta learning</i>	Apontar tendência de pesquisas sobre mudanças e recorrências (<i>drift</i>) em cenários de ML
Sreeja Ashok, Sangeetha Ezhumalai, Tanvi Patwa	<i>Remediating data drifts and re-establishing ML models</i>	2023	<i>Data drifts, Time series, Statistical methods, Sequential methods, Window-based drift detection</i>	Comparar e avaliar diferentes abordagens para <i>drifts</i> de dados
Piotr Porwik, Benjamin Mensah Dadzie	<i>Detection of data drift in a two-dimensional stream using the Kolmogorov-Smirnov test</i>	2022	<i>Kolmogorov-Smirnov test, Classifiers, Data drift</i>	Avaliação sem <i>label</i> nos dados para contexto de fluxo contínuo de dados e mudanças abruptas
Honorius Gâlmeanu, Răzvan Andonie	<i>Weighted Incremental-Decremental Support Vector Machines for concept drift with shifting window</i>	2022	<i>Support Vector Machines, Concept drift, Incremental learning, Shifting window</i>	Aprender a partir de fluxos de dados que experimentam <i>concept drift</i> utilizando a técnica de janela ponderada móvel

Tabela 1 – Sumário dos artigos breve estado da arte.

1.2.6 Conclusão

Em conclusão, o CD é um fator crítico que afeta o desempenho de modelos de em ambientes em constante mudança. Na Tabela 1 são tabulados os artigos utilizados no estado da arte de forma a consolidar de forma visual como esse tema é atual e relevante em diversos cenários. Ao lidar com dados que evoluem ao longo do tempo, é essencial monitorar e adaptar continuamente os modelos para que possam continuar a produzir resultados precisos e confiáveis. Ignorar o CD pode levar a decisões incorretas e prejudicar a eficácia do modelo. Portanto, compreender a importância do CD e ter estratégias eficazes para lidar com ele são fundamentais para garantir que os modelos de permaneçam relevantes e úteis em um mundo em constante mudança.

1.3 Motivação

ML é um tema atual e sua aplicação nos mais diversos modelos de negócio é uma realidade. A compreensão de que modelos de ML são produtos de desenvolvimento que precisam de avaliação de qualidade, monitoramento de performance e eventuais ajustes e correções é fundamental para que sua utilização obtenha os benefícios desejados de forma eficiente e adequada.

Portanto tratar de uma tema que faz parte do processo de manutenção e adequação dos modelos de ML é fundamental para a longevidade e boa conduta no campo do aprendizado de máquina. A degradação de modelos de ML em aplicações e produtos de tecnologia, especialmente em contextos de visão computacional, pode ter consequências significativas. A motivação para estudar o CD nesses cenários é ainda mais relevante dadas as peculiaridades e a importância das aplicações de visão computacional. Aqui estão alguns exemplos sobre o impacto da degradação desses modelos:

- **Segurança e Vigilância:** Modelos de reconhecimento facial ou detecção de objetos em sistemas de segurança podem começar a ter uma taxa alta de falsos positivos ou falsos negativos se estiverem degradados. Um falso negativo em um sistema de segurança pode permitir uma ameaça, enquanto um falso positivo pode desperdiçar recursos em respostas desnecessárias (BABANNE et al., 2019).
- **Veículos Autônomos:** A visão computacional é fundamental para veículos autônomos. Um modelo degradado pode falhar em detectar obstáculos ou interpretar sinais de trânsito corretamente, levando a potenciais acidentes (BACHUTE; SUBHEDAR, 2021).
- **Saúde:** Em aplicações médicas, como diagnóstico assistido por computador, a degradação do modelo pode resultar em diagnósticos incorretos, afetando tratamentos

e, em alguns casos, a vida dos pacientes (NAYYAR; GADHAVI; ZAMAN, 2021).

- **Varejo e Comércio:** Sistemas que dependem da visão computacional para gerenciamento de estoque, reconhecimento de produtos ou análise de comportamento do cliente podem começar a tomar decisões comerciais ineficientes se os modelos estiverem degradados (J et al., 2023).
- **Interação Homem-Máquina:** Sistemas que usam visão computacional para rastreamento ocular, gestos ou outras formas de interação podem se tornar frustrantes ou inutilizáveis se a precisão do modelo diminuir (XIONG et al., 2021).
- **Agricultura de Precisão:** Modelos que identificam pragas ou doenças nas plantações podem perder sua eficácia, levando a perdas agrícolas (CONDRAN et al., 2022).
- **Redes Sociais e Mídia:** Plataformas que dependem de reconhecimento de imagens para categorização de conteúdo, filtragem ou recomendação podem começar a fornecer experiências de usuário abaixo do ideal ou até mesmo permitir conteúdos inapropriados (T.K.; ANNAVARAPU; BABLANI, 2021).
- **Fabricação e Inspeção Industrial:** Na detecção de defeitos ou na garantia da qualidade, a degradação do modelo pode resultar em produtos defeituosos ou custos adicionais para retrabalho (SCHMITT et al., 2020).

Quando modelos de visão computacional são degradados devido ao CD, não apenas a eficácia do sistema individual é comprometida, mas também há implicações de custo, eficiência e, em muitos casos, segurança. Estudar e abordar o CD nesses contextos é, portanto, crucial para garantir que tais sistemas se mantenham confiáveis e eficazes ao longo do tempo.

1.4 Justificativa

A alta demanda por produtos que ofereçam recursos de IA é evidente. Modelos de aprendizado de máquina são construídos e colocados em produção em uma escala maior e tempo menor. Devido a busca pelo diferencial competitivo empresas buscam soluções tecnológicas que aumentem sua atuação e participação no mercado.

O foco maior é concentrado na produção e disponibilização de modelos de ML. Em alguns casos é possível que essa seja a única preocupação. Dessa forma com a passagem do tempo a relação entre os dados de treino e dados de entrada podem mudar bem como a relação entre os dados e variável *target*.

Trazer luz sobre a degradação de modelos de aprendizado de máquina e as técnicas para detecção é uma forma de atentar para a eficiência dos modelos de ML e melhor resultado para as organizações que a utilização e consequentemente seus clientes.

- **Adoção de IA pelas Empresas:** Segundo uma pesquisa da *McKinsey & Company* de 2020, a adoção de tecnologias de IA estava crescendo rapidamente, com cerca de 50% das empresas adotando IA em pelo menos uma função de negócios ([GLOBAL... , 2020](#)).
- **Investimento em IA e ML:** De acordo com o IDC (International Data Corporation), os gastos globais com sistemas de IA estavam previstos para atingir mais de \$110 bilhões em 2024, um grande aumento em relação aos \$50 bilhões gastos em 2020 ([WORLDWIDE... , 2020](#)).
- **Aplicação em Setores Diversos:** A PwC fez uma análise que sugeriu que a IA poderia adicionar até \$15,7 trilhões ao PIB global até 2030, com setores como saúde, finanças e automotivo sendo os maiores beneficiários ([PWC'S... , 2030](#)).
- **Crescimento do Mercado de ML:** O mercado de plataformas de Machine Learning estava projetado para crescer a uma CAGR (Taxa Composta de Crescimento Anual) de 40% durante o período de 2020 a 2027, de acordo com um relatório da *Grand View Research* ([MACHINE... , 2030](#)).
- **Pesquisas Acadêmicas:** A quantidade de pesquisas publicadas relacionadas à IA e ML aumentou exponencialmente na última década. Conferências de prestígio como NeurIPS, ICML e AAAI têm visto um número crescente de inscrições e publicações ano após ano ([FORECASTING... , 2023](#)).

Benefícios da Detecção e Tratamento da Degradação de Modelos de ML

A identificação e o tratamento da degradação de modelos de Machine Learning são críticos para manter e melhorar a operação eficaz de sistemas de IA. Os principais benefícios incluem:

- **Melhoria no Desempenho dos Modelos:** Modelos regularmente monitorados e atualizados se adaptam às mudanças nos dados e nas condições do mundo real, garantindo que permaneçam relevantes e eficazes ao longo do tempo.
- **Maior Precisão nas Previsões:** Um modelo que é corrigido em resposta à detecção de degradação é mais provável que faça previsões precisas, levando a melhores decisões baseadas em dados.

- **Redução de Custos:** Evitando o uso de modelos imprecisos, as organizações podem evitar decisões caras e ineficazes. Além disso, a detecção precoce de problemas pode prevenir custos futuros associados à falha do sistema ou à correção de erros.
- **Aumento da Confiabilidade dos Sistemas de IA:** Sistemas de IA que incluem mecanismos de detecção e tratamento da degradação de modelos são percebidos como mais confiáveis, aumentando a confiança dos stakeholders e dos usuários finais.

A medida que a IA se torna cada vez mais integrada em processos de negócios, operações e produtos, a importância de manter a qualidade e a precisão dos modelos se torna ainda mais crítica. Ao reconhecer e abordar a degradação dos modelos, as organizações estão bem posicionadas para aproveitar ao máximo os benefícios da IA, ao mesmo tempo em que minimizam riscos.

1.5 Organização da Dissertação

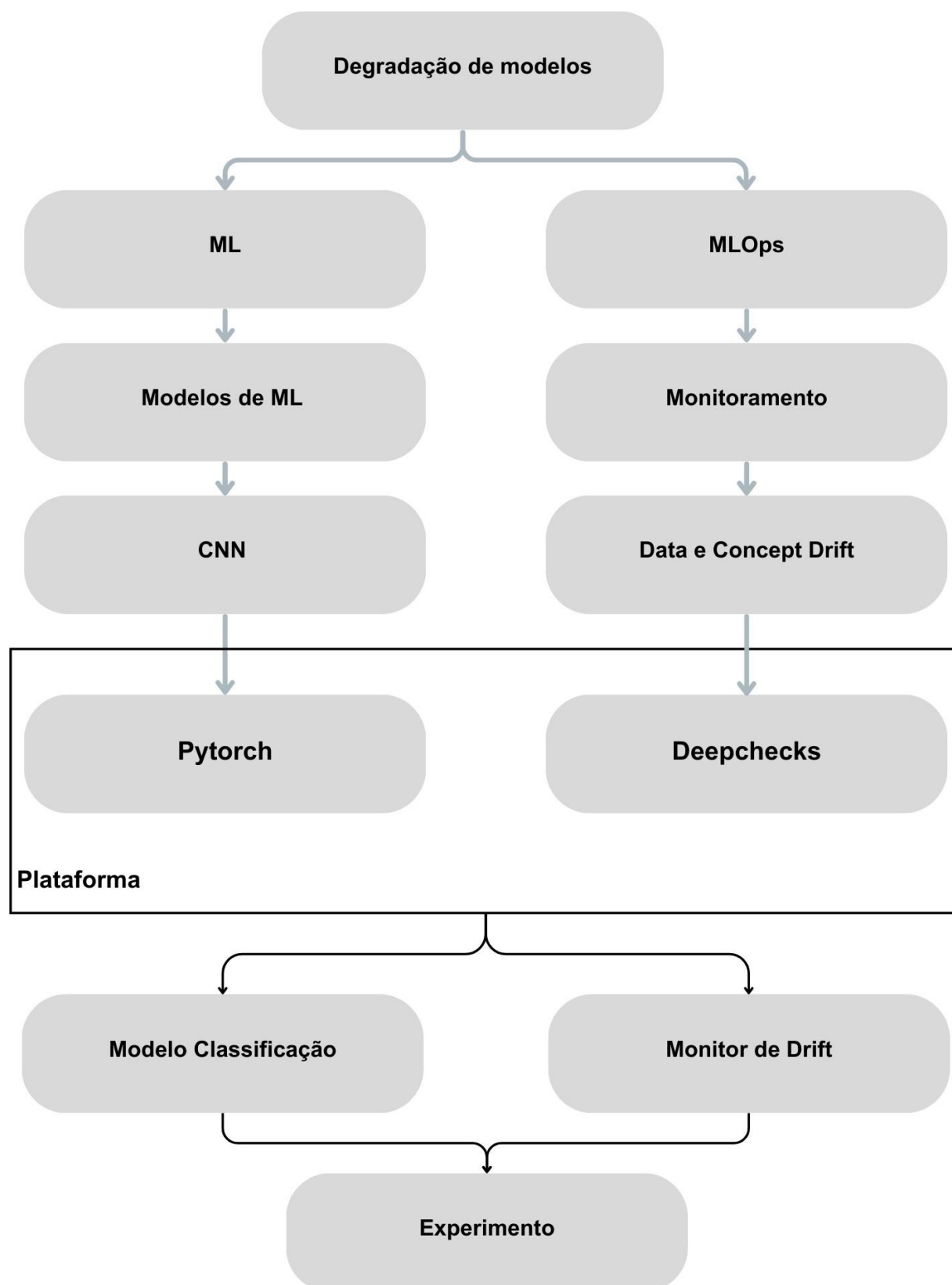
- **Capítulo 1:** Introduz a pesquisa, definindo objetivos gerais e específicos focados em Concept Drift e Data Drift em modelos de Machine Learning. Uma revisão do estado da arte aborda técnicas de detecção e correção de drifts, destacando a relevância do estudo em ambientes dinâmicos. Motivação e justificativa sublinham a importância da pesquisa para avançar no conhecimento e prática na área. A organização da dissertação é brevemente esboçada, preparando o terreno para os capítulos subsequentes.
- **Capítulo 2:** Nesse capítulo, é realizada uma revisão bibliográfica abrangente sobre a temática em foco. A revisão tem como base diversas fontes, incluindo artigos científicos reconhecidos, aplicações práticas relevantes no campo, e livros autorais que são referências na área. Essa exploração aprofundada visa construir um sólido fundamento teórico para o estudo e contextualizar a pesquisa dentro do corpus existente de conhecimento.
- **Capítulo 3:** O terceiro capítulo é dedicado à exposição detalhada dos algoritmos escolhidos para esta investigação. Além disso, são discutidos testes específicos realizados e as métricas utilizadas para avaliar o desempenho dos algoritmos. Um componente crucial deste capítulo será a apresentação da base de dados empregada, elucidando sua origem, características e a relevância para o estudo em questão.
- **Capítulo 4:** Aqui, o foco é para a apresentação e avaliação dos resultados obtidos a partir dos experimentos conduzidos. Esse capítulo é crítico para entender a eficácia e as limitações dos algoritmos testados, bem como para interpretar esses resultados

à luz da revisão bibliográfica previamente realizada. Gráficos, tabelas e análises estatísticas são utilizados para oferecer uma compreensão clara e objetiva dos achados.

- **Capítulo 5:** No último capítulo, há uma reflexão sobre todo o trabalho realizado, apresentando as conclusões derivadas da pesquisa. São abordados as principais descobertas, suas implicações e o impacto potencial na área de estudo. Na sequência são apresentadas as considerações finais, incluindo limitações do estudo e sugestões para pesquisas futuras, visando a continuidade e expansão do conhecimento na temática abordada.

O mapa mental para elaboração do presente trabalho é ilustrado na Figura 2. A concepção ou origem ocorre na degradação de modelos de esse é subdividido em ML e MLOps. ML refere-se parte conceitual ou a modelagem. MLOps refere-se a parte operacional. As definições seguintes são qual o modelo e monitoramento definidos para ML e MLOPS respectivamente temos as *Convolutional neural network* (CNN) e o *Data and Concept Drift*. A etapa adiante é a plataforma. Plataforma é definida neste cenário como o conjunto de ferramentas e bibliotecas usadas para desenvolver aplicações. Nela são definidos pytorch para construção do modelo de classificação de imagens e o *framework Deepcheks* para o monitoramento e calculo do *Drift*. Com o modelo pronto e monitor implementador é iniciado o experimento. O Experimento tem por objetivo avaliar o impacto de diferentes níveis de ruído aplicados a amostras do conjunto de dados e confrontados com o modelo de desenvolvido.

Figura 2 – Mapa mental ilustrando de forma macro a estrutura do trabalho



Fonte: autor.

2 Fundamentação Teórica

A fundamentação teórica serve como alicerce para toda pesquisa acadêmica, garantindo que o trabalho seja construído sobre um conhecimento prévio sólido e reconhecido. Este capítulo é dedicado a apresentar e discutir as principais teorias, conceitos, e estudos anteriores que são pertinentes ao tema desta dissertação. Ao revisar literaturas relevantes, este capítulo visa não apenas contextualizar o estudo dentro de um campo mais amplo de conhecimento, mas também identificar lacunas que a presente pesquisa pretende abordar.

2.1 *Machine Learning*

O ML é uma área da inteligência artificial que se dedica a desenvolver algoritmos capazes de aprender a partir de dados, sem que sejam explicitamente programados para isso (GOODFELLOW; COURVILLE, 2016b; DISABATO; ROVERI, 2022). Esses algoritmos são usados para criar modelos preditivos (SAYED et al., 2023), reconhecimento de padrões (BU; SHI; MA, 2023), classificação (WANG; JIA; XIAN, 2023), segmentação (KNOTT; PEREZ-CRUZ; DEFRAEYE, 2023), entre outras tarefas. Nesta seção, é apresentada uma visão geral do aprendizado de máquina, descrevendo seus principais conceitos e técnicas.

2.1.1 Conceitos básicos

O aprendizado de máquina pode ser dividido em três categorias principais: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (GOODFELLOW; COURVILLE, 2016b; BISHOP, 2006a). No aprendizado supervisionado, o algoritmo é alimentado com um conjunto de dados rotulados e deve aprender a relacionar as entradas com as saídas correspondentes (JAMALEDDYN; El ayachi; BINIZ, 2023). No aprendizado não supervisionado, o algoritmo deve identificar padrões ou estruturas nos dados sem que haja informação prévia sobre as classes ou categorias a que pertencem (POPOV et al., 2023). No aprendizado por reforço, o algoritmo deve aprender a tomar decisões em um ambiente dinâmico, recebendo recompensas ou punições de acordo com suas ações (JAENSCH et al., 2022; LI et al., 2023b).

2.1.2 Técnicas de aprendizado de máquina

Dentro de cada categoria de aprendizado, existem diversas técnicas e algoritmos que podem ser utilizados. No aprendizado supervisionado, os principais algoritmos são as redes neurais (WANG; QIAO; LIN, 2022), árvores de decisão (COLLEDANI; ANSELMINI; ROBUSTO, 2023), regressão linear (NGUYEN et al., 2021) e regressão logística (JEP-

PESEN et al., 2023). No aprendizado não supervisionado, destaca-se o agrupamento ou *clustering* (MOHAMMADI et al., 2023; VIVERIT et al., 2023), que pode ser realizado por meio de técnicas como *k-means* (HALDAR et al., 2023) e DBSCAN (PERAFÁN-LÓPEZ; SIERRA-PÉREZ, 2021). No aprendizado por reforço, estão os algoritmos *Q-learning* (AGHASI et al., 2023), SARSA (MOHAMMED et al., 2023) e Monte Carlo (BAYAR; HAMBARCI, 2022; HUANG; BAI, 2023).

2.1.2.1 Redes Neurais Convolucionais (CNNs)

As Redes Neurais Convolucionais (CNNs) têm se destacado no campo da visão computacional devido à sua capacidade de lidar eficazmente com imagens e aprender características hierárquicas dos dados (TEOH, 2023).

Detalhes Técnicos: Uma CNN é composta de uma sequência de camadas. Essas camadas incluem camadas convolucionais, camadas de pooling (agrupamento) e camadas totalmente conectadas. A principal inovação é a camada convolucional, que aplica filtros à imagem para detectar características de baixo nível, como bordas e texturas. As camadas subsequentes da rede aprendem representações de nível mais alto (OZTURK, 2023).

Algoritmo: O treinamento de uma CNN é geralmente realizado usando uma variação do algoritmo de retropropagação, otimizado por técnicas como o Gradiente Descendente Estocástico (SGD). Durante o treinamento, os pesos dos filtros são atualizados para minimizar uma função de perda, geralmente uma forma de erro de classificação (OZTURK, 2023).

Exemplo de Aplicação: Uma aplicação comum de CNNs é a classificação de imagens. Por exemplo, a arquitetura AlexNet, introduzida em 2012, demonstrou uma performance significativamente superior em um *dataset* popular chamado *ImageNet*, revolucionando o campo da visão computacional (GOODFELLOW; COURVILLE, 2016b).

Aplicações Específicas em Visão Computacional:

- **Classificação de Imagens:** Identificar a classe a que uma imagem pertence, como distinguir entre diferentes animais em fotos.
- **Deteção de Objetos:** Localizar e identificar múltiplos objetos em uma imagem.
- **Segmentação Semântica:** Classificar cada pixel de uma imagem, útil para tarefas como a identificação de objetos em primeiro plano e fundo.
- **Reconhecimento Facial:** Identificar ou verificar uma pessoa a partir de uma imagem facial. (TEOH, 2023)

O sucesso das CNNs na visão computacional deve-se, em grande parte, à sua capacidade de aprender características diretamente dos dados, eliminando a necessidade

de extração manual de características, um processo que era comum em abordagens anteriores à visão computacional ([GONZALEZ; WOODS, 2018](#)).

2.1.2.2 Máquinas de Vetores de Suporte (SVMs)

Embora mais tradicionais, as SVMs ainda são amplamente utilizadas em visão computacional, especialmente em tarefas de classificação. A técnica tenta encontrar um hiperplano que melhor separe as diferentes classes de dados. Em visão computacional, pode ser usada para reconhecimento facial, detecção de objetos, entre outros ([BISHOP, 2006a](#)).

2.1.2.3 Árvores de Decisão e Florestas Aleatórias

Estas são técnicas que dividem o espaço de entrada em regiões, tomando decisões com base em características específicas dos dados. Em visão computacional, elas são frequentemente usadas para tarefas como classificação de imagens e detecção de características ([BISHOP, 2006a](#)).

2.1.2.4 Técnicas de Aprendizado Profundo

Além das CNNs, existem outras arquiteturas de aprendizado profundo, como RNNs e LSTMs, que têm aplicações em análise de sequência temporal. Em visão computacional, elas podem ser usadas, por exemplo, para análise de vídeos ou situações onde o contexto temporal é importante.

Cada uma dessas técnicas tem suas próprias vantagens, limitações e casos de uso específicos em visão computacional. A escolha da técnica depende do problema específico a ser tratado, da natureza dos dados e dos requisitos de desempenho ([GOODFELLOW; COURVILLE, 2016b](#)).

2.1.3 Pré-processamento de dados

Antes de aplicar as técnicas de aprendizado de máquina, é necessário realizar o pré-processamento dos dados ([ZIEGENBEIN; METTERNICH, 2021](#)). Isso envolve tarefas como limpeza dos dados (remoção de ruídos e dados faltantes), normalização (para garantir que todas as variáveis tenham a mesma escala) e seleção de características (para reduzir a dimensionalidade dos dados) ([ZELAYA, 2019](#)).

2.1.4 Avaliação de modelos

Após aplicar as técnicas de aprendizado de máquina, é necessário avaliar a qualidade dos modelos gerados ([GOODFELLOW; COURVILLE, 2016b](#)). Isso pode ser feito por

meio de métricas como acurácia, precisão, revocação, *F1-score* e área sob a curva ROC (ZHANG et al., 2023; FU et al., 2023).

2.1.5 Modelos de Classificação

Os modelos de classificação em aprendizado de máquina são treinados usando um conjunto de dados de treinamento que contém exemplos rotulados com suas respectivas classes (SARAN; KESSWANI, 2023). O objetivo do modelo é aprender a mapear as características de entrada para as classes corretas (BIRCHHA; NIGAM, 2023).

O processo de treinamento envolve ajustar os parâmetros do modelo para minimizar uma função de perda que mede a diferença entre as previsões do modelo e as classes verdadeiras (SHARMA; SHARMA; Kr. Dwivedi, 2023; BISHOP, 2006a). Os parâmetros são ajustados por meio de um algoritmo de otimização que tenta encontrar a combinação ideal de parâmetros que minimizam a função de perda (GHIASI; NG; SHEIKH, 2022; GOODFELLOW; COURVILLE, 2016a).

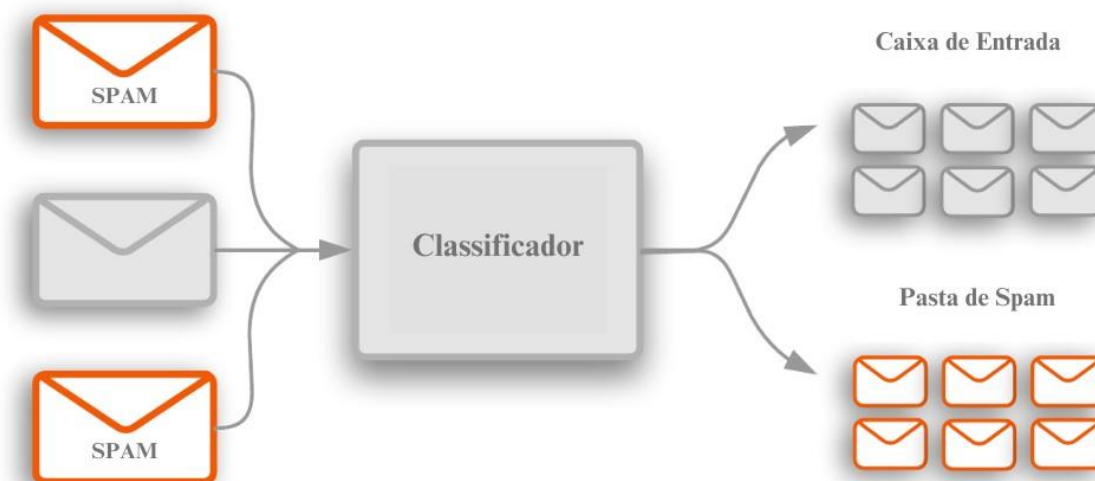
Uma vez que o modelo tenha sido treinado, ele pode ser usado para prever a classe de novos exemplos com base em suas características de entrada. O modelo aplica a função aprendida durante o treinamento aos novos dados e retorna a classe mais provável (HASTIE ROBERT TIBSHIRANI, 2009; TALUKDAR; GOGOI; SINGH, 2023).

Por exemplo, dado um modelo de classificação, supervisionado, para detecção de *Spam* de *e-mails*, ele é treinado em um conjunto de dados de *e-mails* rotulados como *Spam* ou não *Spam* (KONTSEWAYA; ANTONOV; ARTAMONOV, 2021). Durante o treinamento, o modelo aprende a mapear as características do *e-mails*, como palavras-chave ou comprimento do texto, para as classes corretas (SALAMA; ALY; ABOUELSEUD, 2023). Uma vez que o modelo tenha sido treinado, ele pode ser usado para prever se um novo *e-mails* é *Spam* ou não, com base em suas características de entrada (JANEZ-MARTINO et al., 2023). A Figura 3 ilustra um classificador de spam para e-mail onde e-mail são recebidos passando pelo classificador o qual define a caixa de entrada ou pasta de *Spam*.

2.1.5.1 Função de perda ou *Loss Function*

A função de perda, também chamada de função de custo ou função objetivo, é uma medida da discrepância entre as previsões do modelo e as classes verdadeiras nos dados de treinamento (OLSEN et al., 2022). O objetivo do modelo é minimizar a função de perda durante o processo de treinamento.

A escolha da função de perda depende do tipo de problema de classificação que estamos tentando resolver. Por exemplo, a função de perda usada para um problema de classificação binária pode ser diferente da usada para um problema de classificação

Figura 3 – Ilustração de um Classificador de *Spam* para E-mail

Fonte: Extraído de [ladjs cabinjs e lassjs \(2022\)](#).

multiclasse ([CHEN et al., 2022](#)). Algumas das funções de perda comuns usadas em modelos de classificação incluem:

- Erro Médio Quadrático ou *Mean Square Error (MSE)*: é uma função de perda comum usada em problemas de regressão e classificação binária ([BOMMIDI; TEEPARTHI; KOSANA, 2023](#)). Calcula a média dos quadrados das diferenças entre as previsões do modelo e as classes verdadeiras ([OLSEN et al., 2022](#)).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.1)$$

Em que:

- n é o número de exemplos no conjunto de dados de treinamento.
- y_i é o valor verdadeiro do alvo para o i -ésimo exemplo.
- $\hat{y}_i$ é a previsão do modelo para o i -ésimo exemplo.

A equação calcula a média dos quadrados das diferenças entre as previsões do modelo e as classes verdadeiras (BOMMIDI; TEEPARTHI; KOSANA, 2023).

- Entropia Cruzada ou *Cross-Entropy (CE)*: é uma função de perda comum usada em problemas de classificação binária e multiclasse (BOSMAN; ENGELBRECHT; HELBIG, 2020). Mede a discrepância entre a distribuição de probabilidade prevista pelo modelo e a distribuição verdadeira das classes (YEUNG et al., 2022). Para classificação Binária é apresentado a equação 2.2 e para classificação Multiclasse segue a Equação 2.3 (XUE et al., 2023).

$$CE = -\frac{1}{n} \sum_{i=1}^n y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (2.2)$$

$$CE = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k y_{ij} \log(\hat{y}_{ij}) \quad (2.3)$$

Em que na Equação 2.2:

- n é o número de exemplos no conjunto de dados de treinamento.
- y_i é a classe verdadeira para o i -ésimo exemplo (sendo 0 ou 1).
- $\hat{y}_i$ é a probabilidade prevista pelo modelo de que o exemplo i pertence à classe positiva (também conhecida como classe 1).

Em que na Equação 2.3:

- n é o número de exemplos no conjunto de dados de treinamento.
- k é o número de classes.
- y_{ij} é 1 se o exemplo i pertence à classe j e 0 caso contrário.
- $\hat{y}_{ij}$ é a probabilidade prevista pelo modelo de que o exemplo i pertence à classe j .

Em resumo ambas equações medem a discrepância entre a distribuição de probabilidade prevista pelo modelo e a distribuição verdadeira das classes (MEHNI; MEHNI, 2023).

- Erro Absoluto Médio ou *Mean absolute error (MAE)*: é uma função de perda comum usada em problemas de regressão e classificação binária (FRIAS-PAREDES et al., 2018). Calcula a média das diferenças absolutas entre as previsões do modelo e as classes verdadeiras.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.4)$$

Em que:

- n é o número de exemplos no conjunto de dados de treinamento.
- y_i é o valor verdadeiro do alvo para o i -ésimo exemplo.
- $\hat{y}_i$ é a previsão do modelo para o i -ésimo exemplo.

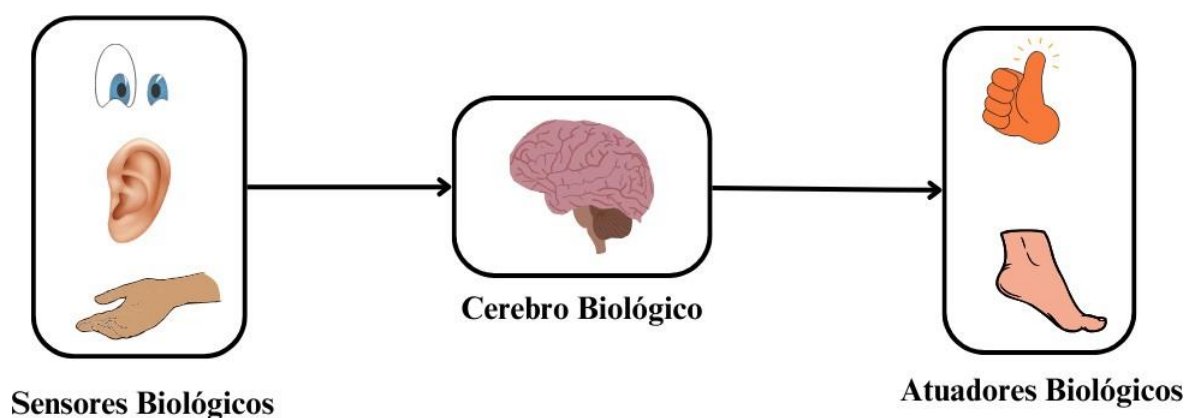
A equação calcula a média das diferenças absolutas entre as previsões do modelo e as classes verdadeiras (IQBAL; SHAD; YASSEN, 2023).

A escolha da função de perda é importante porque afeta diretamente a forma como o modelo é treinado e como as previsões são realizadas. Cada função de perda tem seus prós e contras e deve ser escolhida com base nas características específicas do problema de classificação que estamos tentando resolver (OLSEN et al., 2022; BRAGA; CARVALHO; LUDERMIR, 2007).

2.1.6 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são modelos matemáticos inspirados no funcionamento do cérebro humano, que são utilizados em diversas aplicações de inteligência artificial e aprendizado de máquina (HAYKIN, 2009; BRAGA; CARVALHO; LUDERMIR, 2007; ZHANG; MA, 2023). Na Figura 4 há um exemplo ilustrado do processamento de sinais em sistema biológico onde, nossos sensores como olhos, ouvido e tato percebem o meio ao seu redor enviam essa informação para o cérebro que processa e devolve em ações que são realizadas por atuadores como as mãos e pés (HAYKIN, 2009).

Figura 4 – Exemplo de entrada, processamento e saída de sinais em sistema biológico

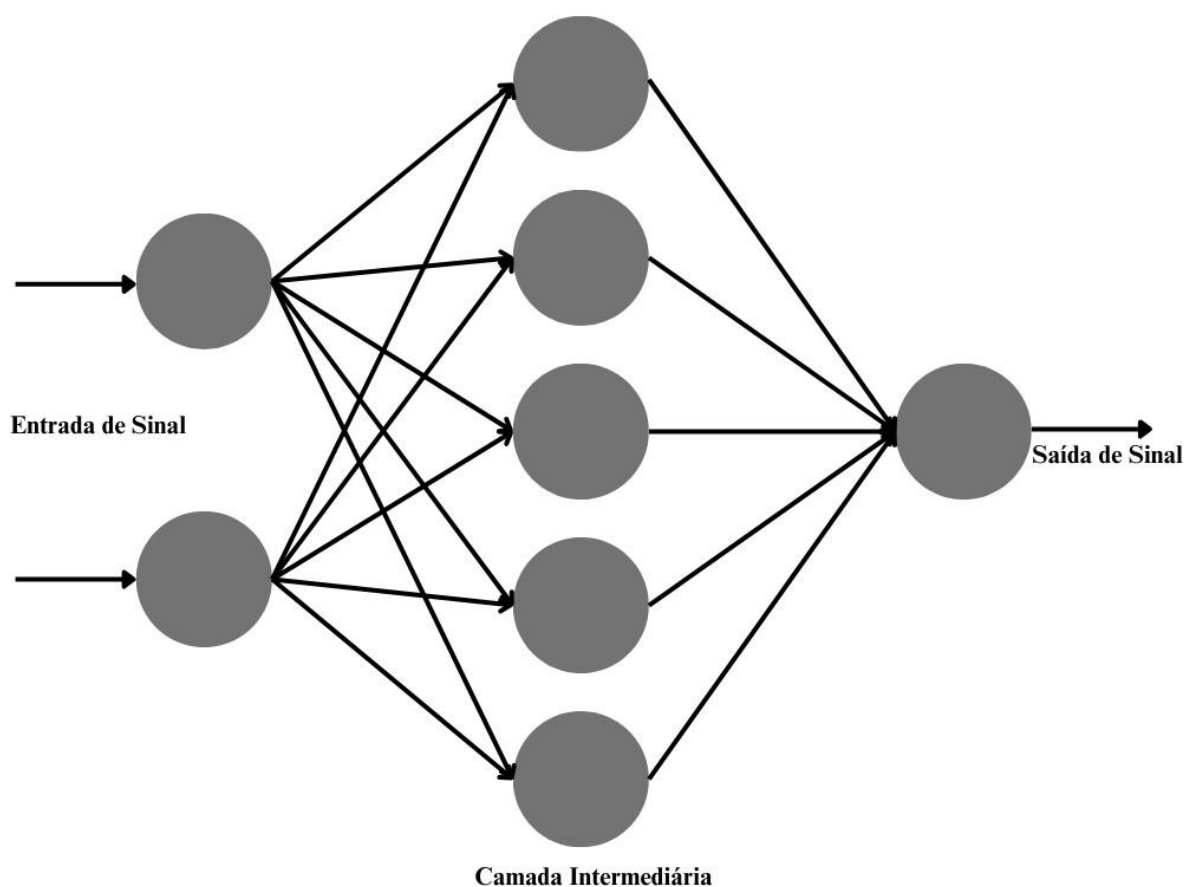


Fonte: Adaptado de Haykin (2009).

As RNAs são compostas por neurônios artificiais interconectados em camadas, formando uma rede de processamento de informação (REZAALIPOUR; FURIA, 2023). Na

Figura 5 cada neurônio recebe uma entrada, realiza um cálculo e produz uma saída, que é enviada para outros neurônios na rede até ser enviado para o neurônio de saída (BRAGA; CARVALHO; LUDERMIR, 2007; ALGHAMDI, 2023). Esses neurônios artificiais tem comportamento espelhado nos neurônios biológicos representados na Figura 6 onde, os dendritos captam os sinais de entrada concentram esses no núcleo o qual processa e envia resposta ao longo do Axônio até as terminações do mesmo (BRAGA; CARVALHO; LUDERMIR, 2007; SADEQ; QADIR; ABBAS, 2023).

Figura 5 – Ilustração de uma rede neural com uma camada intermediária



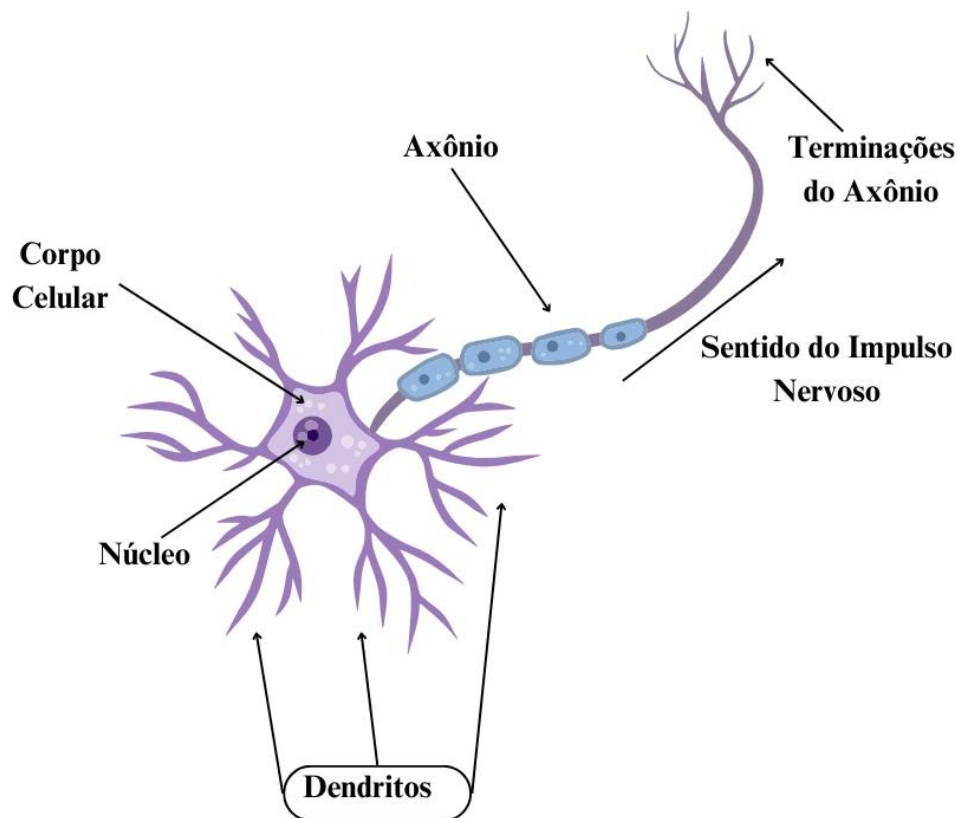
Fonte: Adaptado de Braga, Carvalho e Ludermir (2007).

As redes neurais podem ser treinadas utilizando algoritmos de aprendizado de máquina, para que possam aprender a reconhecer padrões em dados de entrada e fazer previsões ou tomar decisões com base nessas informações (KHAN et al., 2023).

Entre as aplicações mais comuns das redes neurais estão reconhecimento de fala (ADOLFI; BOWERS; POEPEL, 2023), reconhecimento de imagens (DAS et al., 2023) e processamento de linguagem natural (GE et al., 2023; MAGALHÃES; LIMA; POZO, 2023).

Uma rede neural é um processador maciçamente paralelamente distribuído consti-

Figura 6 – Ilustração Neurônio Biológico e fluxo de Sinal



Fonte: Adaptado de [Braga, Carvalho e Ludermir \(2007\)](#).

tuído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso [Haykin \(2009\)](#).

2.1.6.1 Neurônio de McCulloch-Pitts

O neurônio de McCulloch-Pitts (MCP) é um modelo matemático simples de um neurônio artificial, proposto por Warren McCulloch e Walter Pitts em 1943. O modelo é baseado em um circuito lógico binário e foi desenvolvido para explicar como o cérebro humano pode realizar operações lógicas ([MCCULLOCH; PITTS, 1943](#)).

O neurônio de MCP, Figura 7 é composto por um conjunto de entradas binárias (0 ou 1) que são somadas com pesos associados a cada entrada. O resultado da soma é então comparado a um valor de limiar (também binário) e o neurônio emite um sinal de saída (0 ou 1) com base nessa comparação ([HAYKIN, 2009](#); [GOODFELLOW; COURVILLE, 2016a](#)).

Em termos matemáticos, o neurônio de MCP pode ser representado por uma função que recebe um vetor de entrada $x = [x_1, x_2, \dots, x_m]$, um vetor de pesos $w =$

$[w_{k1}, w_{k2}, \dots, w_{kp}]$ e um valor de limiar ϑ_k como entrada e retorna a saída y_k . Onde $\varphi(\cdot)$ é uma função de ativação binária que retorna 1 se o valor de entrada é maior ou igual ao valor de limiar e 0 caso contrário (BRAGA; CARVALHO; LUDERMIR, 2007; HAYKIN, 2009). Na Equação 2.5 há a representação matemática do Neurônio de MCP (BRAGA; CARVALHO; LUDERMIR, 2007; HAYKIN, 2009; ÜNAL; BASCIFTCI, 2023).

$$y_k = f\left(\sum_{i=1}^n w_i x_i - \vartheta_k\right) \quad (2.5)$$

Onde:

- y_k é a saída do neurônio.
- f é a função de ativação, tipicamente uma função degrau.
- w_i são os pesos sinápticos.
- x_i são as entradas do neurônio.
- ϑ_k é o limiar de ativação.
- n é o número de entradas.

A função de limiar também conhecida com função de *Heaviside* é expressa conforme as Equações 2.6 e 2.7 e tomando o valor de $\vartheta = 0$ y_k representa a saída do neurônio k, essa é a propriedade conhecida como tudo ou nada do modelo de MCCulloch-Pitts Haykin (2009).

$$y_k = \begin{cases} 1, & \text{se } v_k \geq \vartheta \end{cases} \quad (2.6)$$

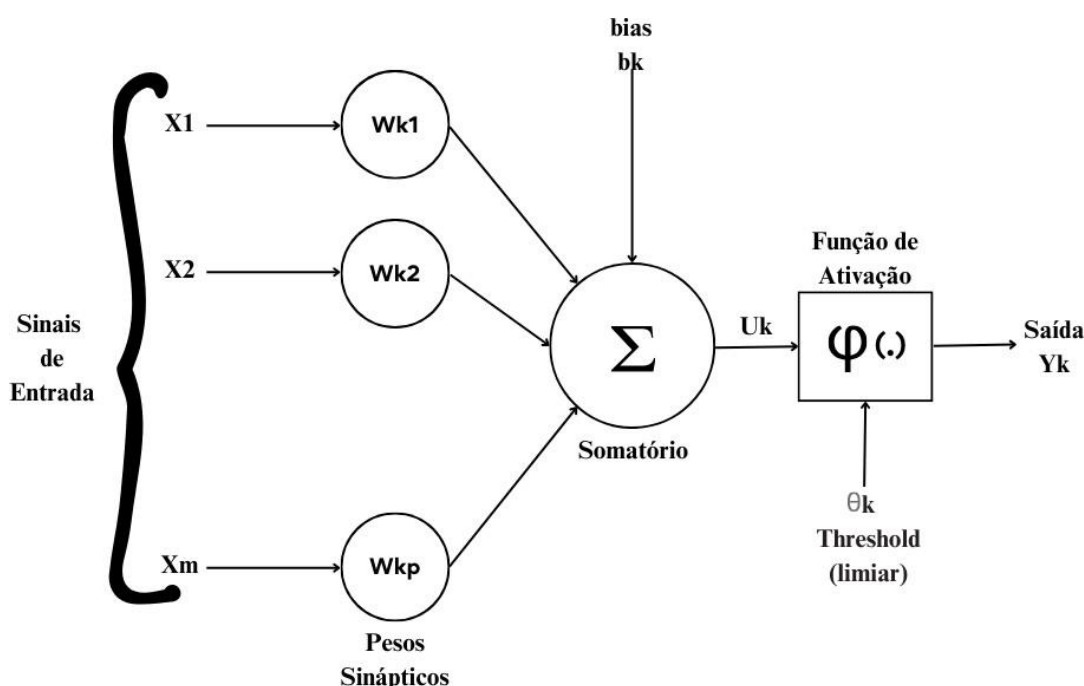
$$y_k = \begin{cases} 0, & \text{se } v_k < \vartheta \end{cases} \quad (2.7)$$

Apesar de sua simplicidade, o modelo de McCulloch-Pitts foi fundamental para o desenvolvimento de redes neurais artificiais e inspirou modelos mais complexos de neurônios artificiais (ÜNAL; BASCIFTCI, 2023).

2.1.6.2 Redes Perceptron

O perceptron é um modelo matemático de um neurônio artificial que foi desenvolvido na década de 1950 pelo cientista da computação Frank Rosenblatt (ROSENBLATT, 1958). O conceito de aprendizado em Redes Neurais Artificiais foi então introduzido pelo trabalho de Rosenblatt (BRAGA; CARVALHO; LUDERMIR, 2007; ROSENBLATT,

Figura 7 – Neurônio de McCulloch-Pitts



Fonte: Adaptado de Haykin (2009).

1958). Construído com base no neurônio não-linear (HAYKIN, 2009), esse neurônio é o modelo proposto por McCulloch e Pitts (1943). Portanto o perceptron é um vetor que recebe entradas binárias $x = [x_1, x_2, \dots, x_n]$ e um vetor de pesos $w = [w_1, w_2, \dots, w_n]$, que são multiplicados e somados para gerar uma entrada líquida v . A entrada que é comparada a um valor de limiar ϑ e o perceptron emite uma saída binária y :

$$v_k = \sum_{i=1}^n x_i w_{ki} \quad (2.8)$$

Em que x_i são as entradas binárias, w_{ki} são os pesos associados a cada entrada, ϑ é o valor de limiar e n é o número de entradas. A entrada líquida z é calculada como a soma ponderada das entradas e pesos, e a saída y é emitida com base na comparação entre z e ϑ .

O perceptron é um modelo linear, o que significa que ele é capaz de aprender apenas funções lineares dos dados de entrada Haykin (2009). No entanto, ele pode ser usado como um bloco de construção em redes neurais mais complexas, como as redes neurais *multilayer* perceptron (MLP) (ABU-DOUSH et al., 2023).

O perceptron é usado em várias aplicações de aprendizado de máquina, incluindo classificação binária, análise de sentimentos e reconhecimento de padrões (BISHOP, 2006a; NANDA et al., 2023; LI et al., 2023a).

2.1.6.3 Função de ativação Sigmoid

A função sigmoide é uma função matemática que transforma qualquer valor de entrada em um valor entre 0 e 1 (GOODFELLOW; COURVILLE, 2016a; HAYKIN, 2009; BRAGA; CARVALHO; LUDERMIR, 2007). A função mais comum usada como função de ativação é a logística representada na Equação 2.10. A representação matemática pode ser escrita conforme Equação 2.9.

$$y = f\left(\sum_{i=1}^n x_i w_i\right) \quad (2.9)$$

em que,

- x_i são as entradas,
- w_i são os pesos associados a cada entrada,
- n é o número de entradas e

2.1.6.4 Neurônio com função de ativação Sigmoid

O neurônio com função de ativação Sigmoid é amplamente usado em redes neurais artificiais devido à sua capacidade de modelar funções não lineares e à sua suavidade em relação à função de ativação binária (DEMIR; SAGLAM, 2023). A função sigmoide também é utilizada em outros modelos de redes neurais, como a rede neural recorrente e a rede neural convolucional (ASRAV; AYDIN, 2023).

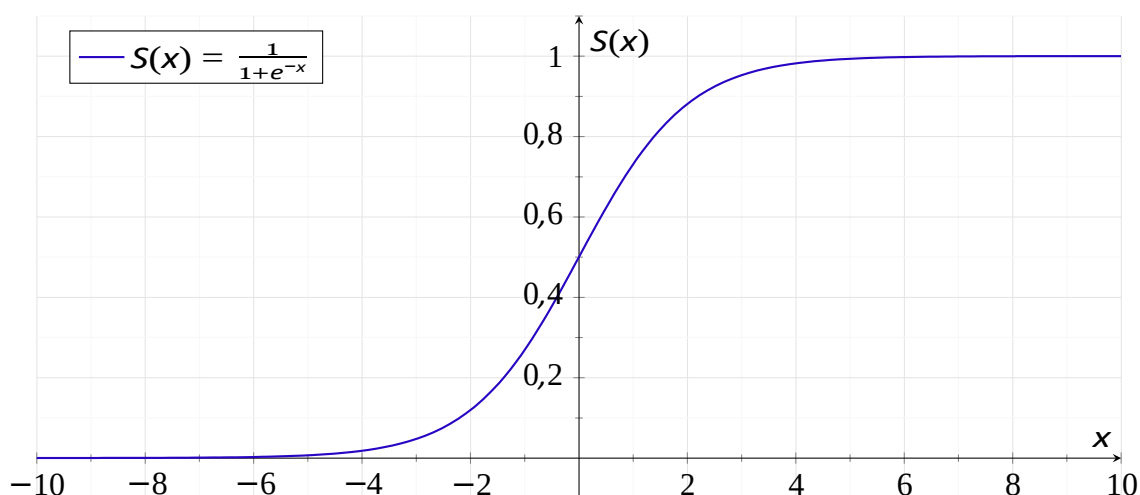
$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

Em que,

- σ é a função de ativação sigmoide Equação 2.10.

A Figura 8 representa a função sigmoide $\sigma(x)$, comumente usado na camada oculta de uma rede neural. (HASTIE ROBERT TIBSHIRANI, 2009).

Figura 8 – Gráfico representando a função Sigmoide



2.1.7 Redes Neurais Convolucionais

Redes neurais convolucionais (RNCs) são um tipo de modelo de aprendizado amplamente utilizados em tarefas de visão computacional (GOODFELLOW; COURVILLE, 2016a; ISLAM et al., 2023). Elas se destacam por sua capacidade de reconhecer padrões em dados de alta dimensionalidade, como imagens e aprender representações hierárquicas desses dados (RAHMAN; ISLAM, 2023).

2.1.7.1 Conceitos básicos de redes neurais convolucionais

As RNCs são compostas por várias camadas, cada uma com uma função específica. As camadas convolucionais são responsáveis por identificar os padrões nos dados de entrada, como bordas, formas e objetos (HAYKIN, 2009; GOODFELLOW; COURVILLE, 2016a). Essas camadas aplicam uma série de filtros ou *kernels* sobre a entrada, que são deslocados através de toda a imagem, produzindo um mapa de características que representa a presença de diferentes padrões em diferentes partes da imagem (GONZALEZ; WOODS, 2018; ISLAM et al., 2023).

As camadas de *pooling* são frequentemente adicionadas após as camadas convolucionais para reduzir a dimensionalidade dos mapas de características, agregando informações de regiões adjacentes em uma única unidade (RAJPUT et al., 2023; FAN et al., 2023). Isso reduz o número de parâmetros e a complexidade da rede, além de ajudar a lidar com a variabilidade nos dados de entrada (DU et al., 2023b).

As camadas totalmente conectadas são responsáveis por combinar as informações de todas as unidades nas camadas anteriores para gerar uma saída final, que pode ser usada para classificação, detecção de objetos ou outras tarefas (YANG et al., 2023; PANG; WANG, 2023).

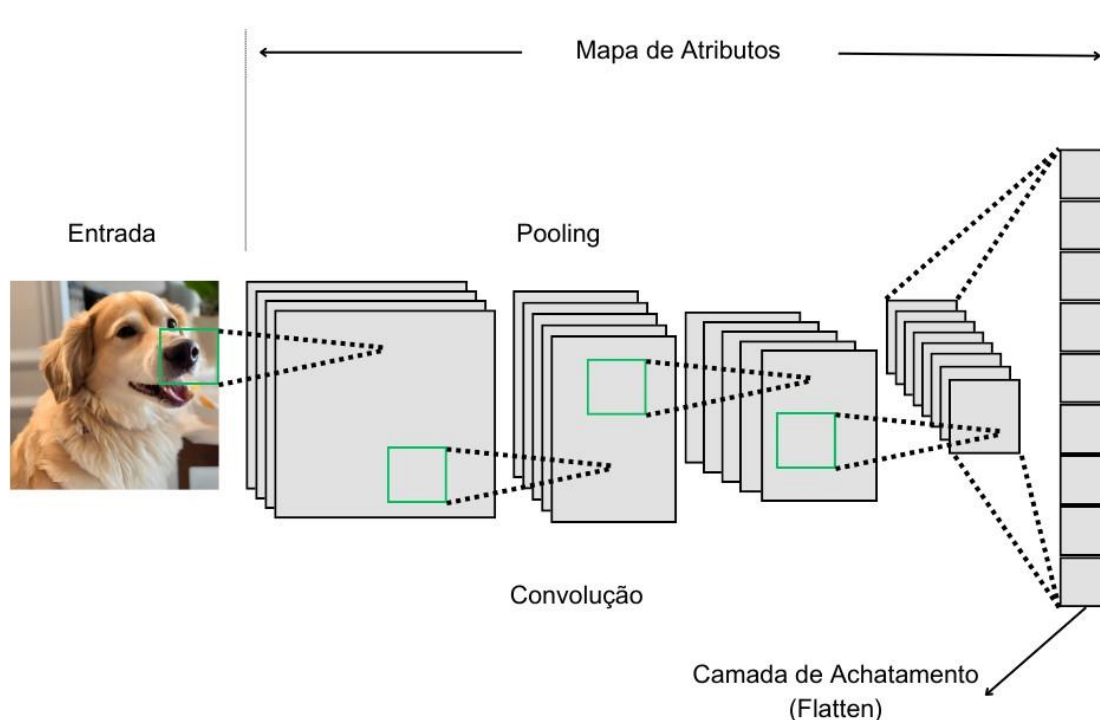
As RNCs são treinadas usando um algoritmo de otimização, como descida de gradiente estocástica, para ajustar os pesos e os vieses das diferentes camadas da rede (GOODFELLOW; COURVILLE, 2016a; DU et al., 2023b). O treinamento ocorre usando um conjunto de dados de treinamento rotulados para ajustar a rede a tarefas específicas, como reconhecimento de objetos ou classificação de imagens (GONZALEZ; WOODS, 2018; OSBORNE; DORVILLE; ROMANO, 2023; HUANG et al., 2023).

2.1.7.2 Arquitetura e funcionamento das camadas convolucionais

Redes convolucionais são redes neurais que usam convolução no lugar de uma matriz multiplicação em pelo menos uma de suas camadas Goodfellow e Courville (2016b).

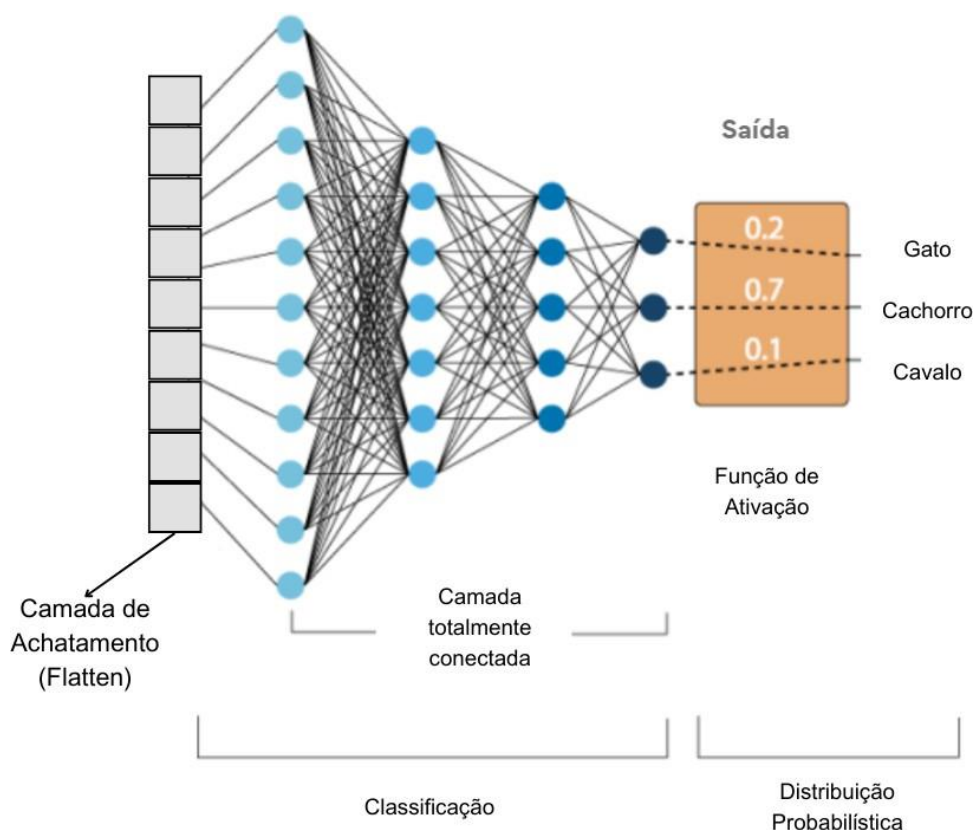
As várias camadas das redes neurais convolucionais (RNCs) tem funções específicas. As camadas convolucionais, de *pooling* e totalmente conectadas são as principais camadas em uma RNC e são responsáveis por extrair recursos das entradas, reduzir a dimensionalidade dos recursos extraídos e gerar uma saída final, respectivamente.

Figura 9 – Rede Neural Convolutacional etapa de extração de atributos



Fonte: Adaptado de SwapnaKE (2023).

Figura 10 – Rede Neural Convolutacional etapa de classificação



Fonte: Adaptado de [SwapnaKE \(2023\)](#).

As camadas convolucionais são a base das RNCs e têm a função de identificar características importantes nas entradas, como bordas, texturas e objetos ([OSBORNE; DORVILLE; ROMANO, 2023](#)). Na Figura 9 é observado a aplicação de um conjunto de filtros ou *kernels* a cada parte da entrada e geram um mapa de características que indica a presença de características específicas em diferentes partes da entrada. Os filtros são ajustados durante o treinamento para identificar as características mais importantes para a tarefa em questão ([YANG et al., 2023](#)).

Ainda na Figura 9 as camadas de *pooling* são frequentemente adicionadas após as camadas convolucionais para reduzir a dimensionalidade do mapa de características, agregando informações de regiões adjacentes em uma única unidade ([RAJPUT et al., 2023](#)). Isso ajuda a reduzir a variabilidade nos dados de entrada e a tornar a rede mais robusta. As operações de *pooling* mais comuns são o *max pooling* e o *average pooling* ([RAJPUT et al., 2023](#)).

As camadas totalmente conectadas são responsáveis por combinar as informações de todas as unidades nas camadas anteriores e gerar uma saída final. Essas camadas

recebem como entrada um vetor de recursos extraídos pelas camadas convolucionais e de *pooling* e geram uma saída que pode ser usada para classificação, detecção de objetos ou outras tarefas (HUANG et al., 2023). A saída da última camada é geralmente passada por uma função de ativação, como a função *softmax*, para gerar uma distribuição de probabilidade sobre as classes de saída (GOODFELLOW; COURVILLE, 2016a).

A arquitetura geral de uma RNC é formada por várias camadas convolucionais, de *pooling* e totalmente conectadas, empilhadas uma em cima da outra. As camadas convolucionais e de *pooling* geralmente se alternam, reduzindo a dimensionalidade do mapa de características e permitindo que as camadas posteriores tenham um número manejável de parâmetros (GOODFELLOW; COURVILLE, 2016a; OSBORNE; DORVILLE; ROMANO, 2023; ISLAM et al., 2023).

2.1.7.3 Treinamento e ajuste dos parâmetros da rede

O treinamento de uma rede neural convolucional (RNC) envolve a otimização dos parâmetros da rede para minimizar a função de perda, que mede a diferença entre as saídas previstas e as saídas reais. Os parâmetros da rede incluem os pesos das conexões entre as unidades, os parâmetros dos filtros das camadas convolucionais e os pesos das camadas totalmente conectadas (GOODFELLOW; COURVILLE, 2016a; HAYKIN, 2009).

O processo de treinamento geralmente envolve a alimentação dos dados de treinamento na rede e a atualização iterativa dos parâmetros usando um algoritmo de otimização, como o gradiente descendente estocástico (SGD) (HASTIE ROBERT TIBSHIRANI, 2009; BRAGA; CARVALHO; LUDERMIR, 2007; DU et al., 2023b). Durante o treinamento, a rede é avaliada em um conjunto de validação para garantir que não esteja superajustando os dados de treinamento.

Para evitar o *overfitting*, muitas técnicas de regularização são usadas em RNCs. Uma técnica comum é o *dropout*, que desativa aleatoriamente algumas das unidades da rede durante o treinamento, impedindo que elas sejam muito dependentes de outras unidades (GOODFELLOW; COURVILLE, 2016a; PANG; WANG, 2023). Outra técnica é a regularização L2, que adiciona uma penalidade na norma dos pesos da rede durante o treinamento, incentivando pesos menores.

Além disso, a escolha dos hiperparâmetros da rede, como o tamanho do filtro, o número de filtros, o tamanho do *batch* e a taxa de aprendizado, é crucial para o desempenho da RNC. Esses hiperparâmetros afetam a complexidade da rede e a velocidade e estabilidade do treinamento (BRAGA; CARVALHO; LUDERMIR, 2007; PANG; WANG, 2023).

A busca pelos melhores hiperparâmetros é um processo iterativo que envolve a avaliação do desempenho da rede em um conjunto de validação para diferentes combinações de hiperparâmetros. Técnicas de busca de hiperparâmetros, como a busca em grade e a

busca aleatória, podem ser usadas para automatizar esse processo (HUANG et al., 2023).

Em resumo, o treinamento de uma RNC envolve a otimização iterativa dos parâmetros da rede usando um algoritmo de otimização, como o SGD. Técnicas de regularização, como o *dropout* e a regularização L2, são usadas para evitar o *overfitting*. A escolha dos hiperparâmetros é crucial para o desempenho da rede e pode ser feita por meio de técnicas de busca de hiperparâmetros (GOODFELLOW; COURVILLE, 2016a; HAYKIN, 2009; HUANG et al., 2023; PANG; WANG, 2023).

2.1.7.4 Transferência de aprendizado e pré-treinamento

A transferência de aprendizado e o pré-treinamento são técnicas que podem ser usadas em redes neurais convolucionais (RNCs) para melhorar o desempenho em tarefas de classificação de imagem, mesmo com conjuntos de dados pequenos (YU et al., 2021).

A transferência de aprendizado é uma técnica em que os pesos de uma rede neural pré-treinada em uma tarefa semelhante são usados como ponto de partida para treinar uma nova rede neural para uma tarefa relacionada (ALATRANY et al., 2023). Por exemplo, uma RNC treinada para reconhecer objetos em um conjunto de dados grande, como o ImageNet (PINTOR et al., 2023), pode ser usada como ponto de partida para treinar uma nova rede para reconhecer objetos em um conjunto de dados menor e mais específico, como um conjunto de dados de carros (BEHERA; KHAN; BAKSHI, 2022).

O pré-treinamento é outra técnica que pode ser usada em RNCs para melhorar o desempenho em tarefas de classificação de imagem. Em vez de treinar a RNC do zero, a rede é pré-treinada em um conjunto de dados grande, como o *ImageNet*, em uma tarefa de classificação geral de imagens (PINTOR et al., 2023). Em seguida, a RNC é ajustada em um conjunto de dados menor e mais específico para a tarefa desejada, como a classificação de imagens médicas (PRAKASH et al., 2023).

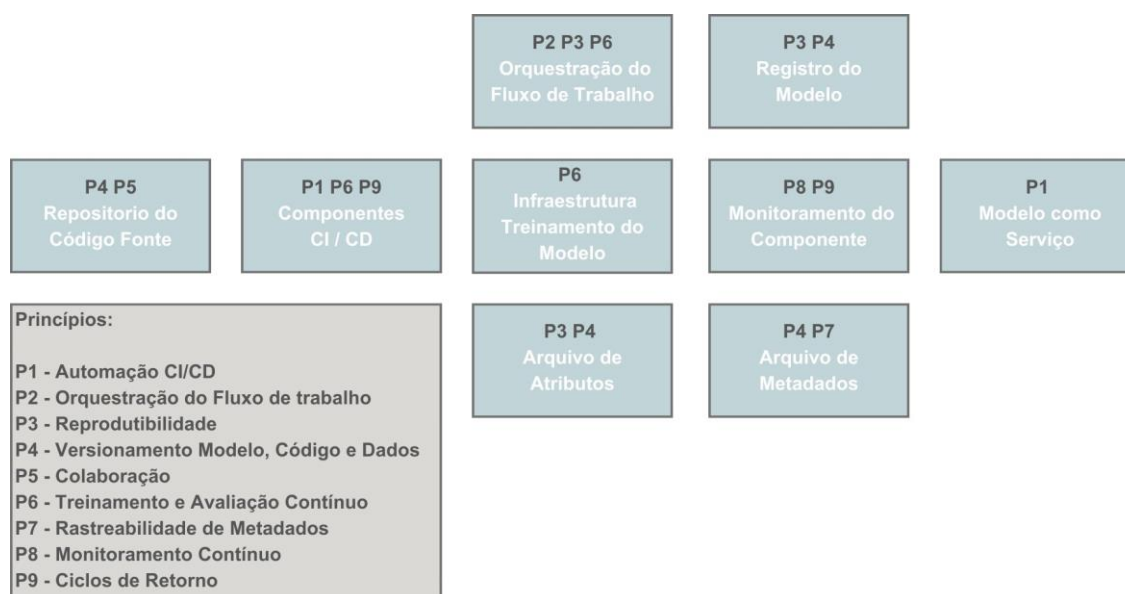
Essas técnicas de transferência de aprendizado e pré-treinamento são eficazes porque as camadas iniciais de uma RNC são geralmente responsáveis por extrair características gerais de uma imagem, como linhas e formas (SADIK et al., 2023). Essas características são úteis para tarefas de classificação de imagem em geral e podem ser transferidas ou pré-treinadas para novas tarefas relacionadas (STIGT et al., 2023; AHMED et al., 2023).

2.2 Machine Learning Operations - MLOps

Machine Learning Operations (MLOps) pode ser definido como o processo de automatizar o aprendizado de máquina usando metodologias e melhores práticas de integração contínua e entrega contínua unido assim operação e desenvolvimento (GIFT,

2021). Em MLOps princípio é um guia de como as atividades devem ser realizadas. (KREUZBERGER; KÜHL; HIRSCHL, 2023).

Figura 11 – Implementação dos Princípios e componentes técnicos



Fonte: Adaptado de Kreuzberger, Kühl e Hirschl (2023).

Na Figura 11 há um quadro que ilustra a implementação de princípios e componentes técnicos para MLOps. No quadro há os princípios que são boas praticas sugeridas para a melhor utilização dos componentes descritos em cada quadro (KREUZBERGER; KÜHL; HIRSCHL, 2023). Os componentes Técnicos por sua vez são as tecnologia ou o conceito tecnológico que deve utilizar o principio descrito em cada quadro. Abaixo segue definição de cada princípio.

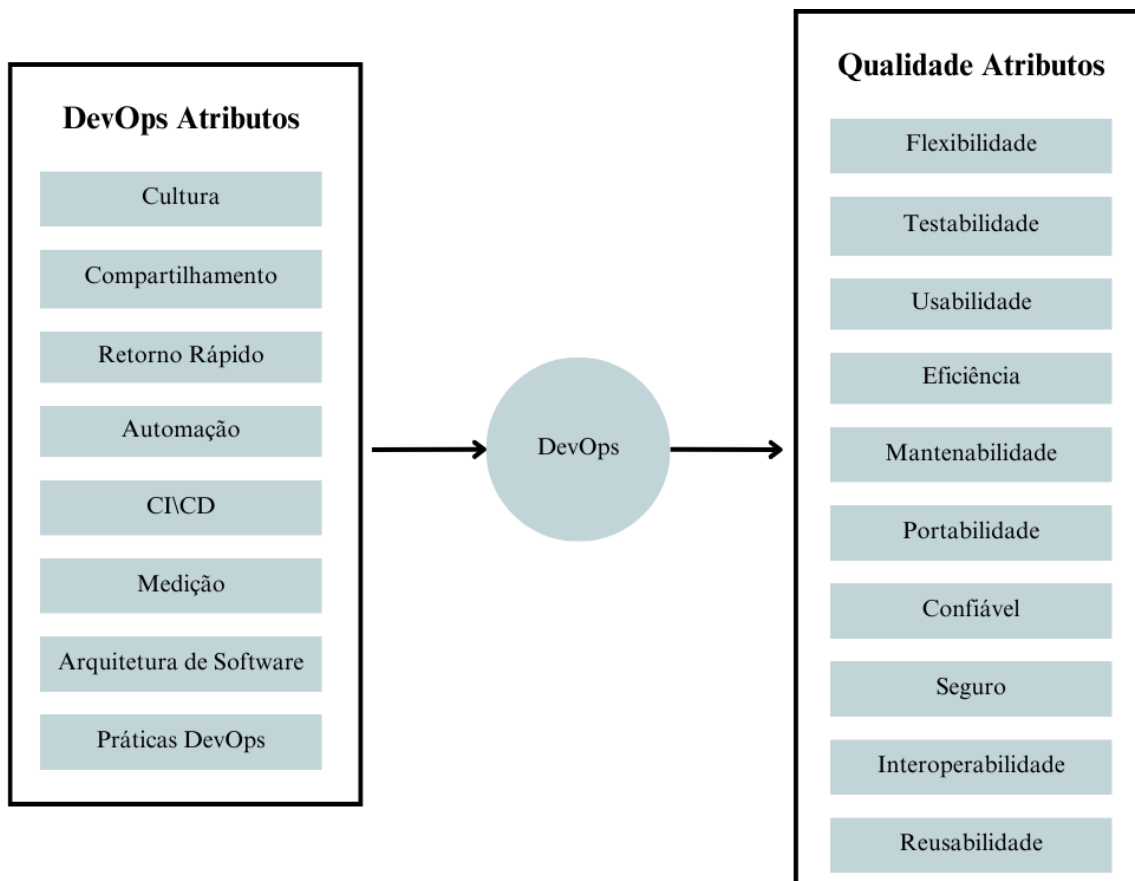
- P1 Automação CI/CD:** *Continuous Integration (CI) e Continuous Deployment (CD)*. O CI é implementado com intuito de que os membros do time de desenvolvimento que trabalham em colaboração possam realizar suas entregas de partes do código o qual estão trabalhando com frequência de forma continua com entregas menores permitindo melhorias na qualidade do *software* (SHAHIN; BABAR; ZHU, 2017). CD é possível compreender como *continuous delivery* e *continuous deployment*. O CD *delivery* garante que a aplicação está pronta para ser colocada no ambiente de produção. Todos os testes foram realizados e a integração funcionou conforme esperado. No CD *deploy* a aplicação é disponibilizada no ambiente de produção de forma totalmente automática (SHAHIN; BABAR; ZHU, 2017; KREUZBERGER; KÜHL; HIRSCHL, 2023).

- **P2 Orquestração do fluxo de trabalho:** Atividades em geral tem sequências e regras para que essas aconteçam. Da mesma forma as atividades em MLOps precisam ocorrer com base em regras e sequencias ou *steps* bem definidos. Quando uma atividade depende da conclusão de uma ou mais atividades anteriores chamamos de dependências. Ferramentas de orquestração gerenciam essas dependências com objetivo de que cada *step* seja executado apenas quando todas as dependências estiverem prontas (KREUZBERGER; KÜHL; HIRSCHL, 2023; DANDEKAR, 2021).
- **P3 Reprodutibilidade:** Capacidade de um experimento de ML reproduzir sob as mesmas condições exatamente o mesmo resultado (KREUZBERGER; KÜHL; HIRSCHL, 2023).
- **P4 Versionamento:** Versionamento do modelo de dados ou seja deve ser conforme P3 mas também deve ser rastreável (KREUZBERGER; KÜHL; HIRSCHL, 2023). E extremamente importante saber qual versão do modelo está em produção. Cada versão deve ser identificada de forma adequada, rastreável e recuperável.
- **P5 Colaboração:** Possibilidade de desenvolvimento de forma colaborativa reduzido a diferença entre as diversas áreas de domínio. Uma equipe de desenvolvimento pode ser composta por diversos desenvolvedores com habilidades e conhecimentos diversificados. A colaboração acelera e melhora a qualidade do desenvolvimento de código (FARUK et al., 2022; KREUZBERGER; KÜHL; HIRSCHL, 2023).
- **P6 Treinamento e Avaliação Contínuo:** Retreinamento periódico dos modelos de ML com base em monitoramento das métricas é uma prática adequada e um principio valioso quando olhamos para a degradação dos modelos de ML (THONGLEK et al., 2022; KIDANE et al., 2022).
- **P7 Rastreabilidade de Metadados:** Metadados são dados gerados ao longo da esteira de modelos que carregam informações por exemplo do autor, tipo do modelo, versão, ambiente, testes, dependências e outros. Manter o registro desses metadados é necessário para cada iteração de treinamento, por exemplo, treinamento data e hora, duração, incluindo o modelo específico metadados parâmetros usados e o desempenho resultante, linhagem de modelo, dados e código usados para garantir o pleno rastreabilidade de execuções de experimentos (KREUZBERGER; KÜHL; HIRSCHL, 2023).
- **P8 Monitoramento Contínuo:** O monitoramento contínuo implica em avaliações automáticas e periódicas de dados, código, infraestrutura recursos e desempenho do modelo para detectar possíveis erros ou mudanças que influenciam o qualidade do produto (KREUZBERGER; KÜHL; HIRSCHL, 2023).

- **P9 Ciclos de retorno:** Múltiplos ciclos de retorno ou *feedback* são necessários para avaliar se realmente o que foi idealizado é o que o modelo está entregando e com os critérios de qualidade definidos (KREUZBERGER; KÜHL; HIRSCHL, 2023).

Esses princípios são encontrados nos conceitos de desenvolvimento e operação (DevOps). O termo DevOps representa um conjunto de métodos em que desenvolvedores e operações se comunicam e colaboram para entregar *software* e serviços de forma rápida, confiável e com maior qualidade (MISHRA; OTAIWI, 2020). Na Figura 12 é possível ver a relação entre os atributos de DevOps e os de qualidade de *software*. DevOps foi aplicado como metodologia e técnicas de desenvolvimento de *software* ágil e enxuto assim sua utilização em aprendizado de máquina foi relatado em vários domínios (LIU et al., 2020).

Figura 12 – Relacionamento entre DevOps e Qualidade de *software*



Fonte: Adaptado de Mishira e Otaiwi (2020).

MLOps tem por objetivo permitir que a automação assuma o controle das atividades de rotina para que os desenvolvedores possam concentrar esforços na solução do problema de negócio (TAMBURRI, 2020). Dessa forma ocorre uma evolução na qualidade do modelos de ML. O Ciclo de vida de um modelo é complexo e repleto de vários componentes. A

necessidade de padronizar processos é extremamente importante para o sucesso de longo prazo de aplicações.

Práticas de MLOps são extremamente necessárias visto a alta complexidade e possibilidade de inúmeras etapas com diversas iterações. É um desafio comercializar soluções de aprendizado de máquina. O ciclo de vida do aprendizado de máquina compreende muitos componentes complexos, incluindo coleta de dados, preparação de dados, treinamento de modelo, ajuste de modelo, implantação de modelo, monitoramento de modelo. A comunicação entre equipes de Engenharia de Dados, Ciência de Dados e Engenharia de *Machine Learning* é fundamental para manter todos esses processos sincronizados e funcionando em uníssono. O MLOps chega com o propósito de tornar tudo isso menos complexo. A disponibilidade constante de *endpoints* para realizar previsão e treinamento também deve fazer parte de uma solução de aprendizado de máquina (SILVA et al., 2020). O treinamento e monitoramento contínuo é parte essencial do processo de MLOps. Com a evolução do tempo o problema de degradação da qualidade do modelo é uma realidade e para mitigar esse impacto na qualidade retreinar o modelo periodicamente é uma alternativa, isso pode ser realizado usando o histórico dos dados mais recentes (DERAKHSHAN; MARKL, 2019). No caso de MLOps, o monitoramento de recursos é necessário como qualquer outro recurso. No entanto, o monitoramento precisa ser dentro do contexto do ML levando em consideração as características como vieses e desvios que podem surgir ao longo do tempo (GRANLUND et al., 2021).

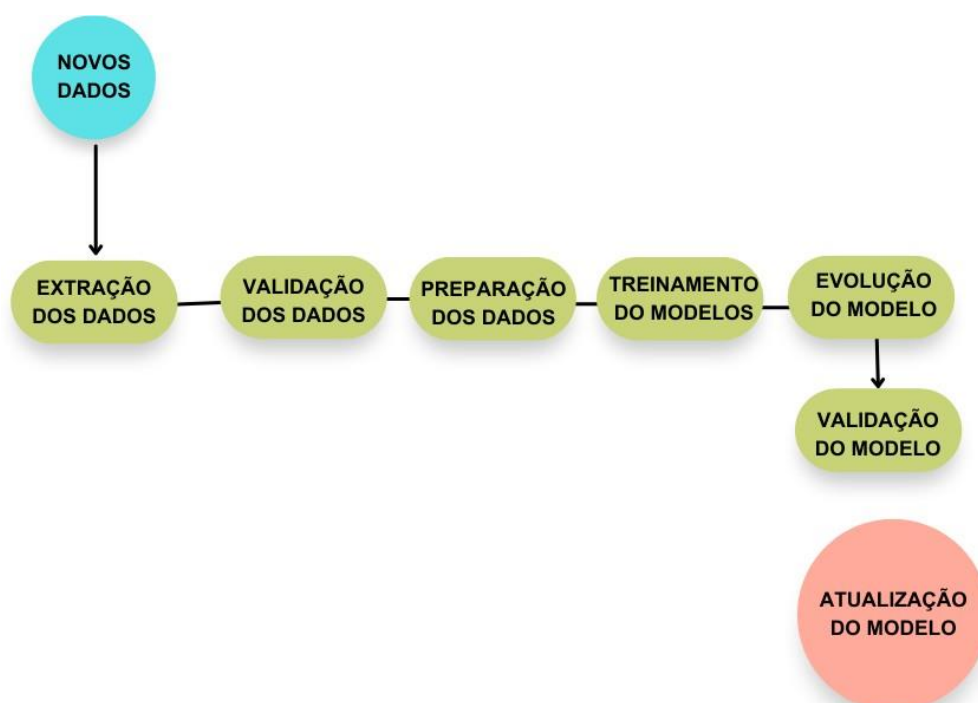
Com o advento do *Big Data*, Internet das Coisas (*Internet of things, IoT*) e 5G a quantidade e frequência de novos dados é bastante intensa. Manter os modelos generalizando de forma adequada é papel da etapa de monitoramento contínuo e consequentemente treinamento contínuo (PRAPAS et al., 2021). Os dados do mundo real são carregados de entropia. Isso impacta na degradação dos modelos e a engenharia de *machine learning* deve atuar para reduzir ou eliminar esses impactos. Os modelos tem seu desempenho degradado com o tempo (ZLIOBAITE, 2011). Isso ocorre porque os dados mudam em tempo real devido a mudanças nas tendências.

Há 3 principais formas de treinar continuamente um modelo de aprendizado de máquina:

- Incremental - treinar o modelo com novos dados à medida que os dados chegam, mantendo o modelo existente.
- Lote ou *Batch* - treinar o modelo assim que uma quantidade significativa de novos dados estiver disponível, mantendo o modelo existente.
- Retreinamento - retreinar todo o modelo do zero quando uma quantidade significativa de dados estiver disponível.

Cada método pode ser implementado a depender do cenário e da escolha da equipe de engenharia de ML. Ponto em comum é que independente da escolha qualquer um desses métodos precisará ser automatizado. Realizar esse processo de forma manual quando há poucos modelos pode ser tentador. Porém em organizações que tem alta demanda e escalabilidade de modelos com diversidade de aplicações optar pela automação é a solução adequada para sustentabilidade da operação. Dessa forma aplicar os conceitos, técnicas e ferramentas de MLOps são fundamentais (BAYRAM; AHMED; KASSLER, 2022).

Figura 13 – Fluxo dos dados em Treinamento Contínuo de modelo de ML



Fonte: O Autor.

Para que o processo de Aprendizado Contínuo ocorra são necessários gatilhos ou *triggers*. Esses gatilhos disparam eventos em um *pipeline* qual o processo pode ser visto na Figura 13. A Figura 13 ilustra a entrada de novos dados no processo de retreinamento, esses novos dados passam pelas fases de extração, validação e preparação dos dados onde seguem para o treinamento do modelo que tem sua evolução e validação avaliadas para servir ou não para atualizar o modelo. Abaixo algumas formas de acionamento do retreinamento.

- Acionamento manual.
- De tempos em tempos ou seja a cada janela de tempo semanal, mensal, trimestral a depender das regras de negócio.
- Acionado quando novos dados chegam.
- Alterações na distribuição de dados.
- Deterioração do desempenho do modelo.

2.3 Data Drift

Quando um modelo entra em produção ele deve permanecer em monitoramento e avaliação. Essa etapa tem por objetivo que variações nas características estatísticas dos dados que ocorrem com o tempo degradem o modelo a tal ponto de sua inviabilidade (MANIAS; CHOUMAN; SHAMI, 2022). Sensores usados para adquirir dados podem perder a sensibilidade com o tempo, as populações podem envelhecer, e as tendências da moda podem mudar (ACKERMAN et al., 2021). A degradação do modelo pode levar a níveis de precisão abaixo de limites aceitáveis em áreas por exemplo medicina, indústria petroquímica, aeroespacial. Quando a degradação do modelo tem origem na mudança das variáveis de entrada ou atributos chamamos de *Data Drift*. Na equação 2.11 temos a descrição matemática do *Data Drift*.

$$P_{t1}(X) \neq P_{t2}(X) \quad (2.11)$$

Outro caso em que a precisão pode diminuir é quando os sensores usado para capturar dados, como uma lâmpada de raios X que, conforme o tempo avança, suas propriedades mudam e consequentemente os dados capturados. Neste caso, a degradação pode ser fatal (ACKERMAN et al., 2021).

Muitas técnicas de classificação calculam uma medida de confiança em seus resultados. Esta medida pode não refletir o real desempenho de um modelo.

O problema da mudança geralmente envolve detectar quando a distribuição dos dados, observados ao longo do tempo, mudaram (ACKERMAN et al., 2020).

Data Drift, também é conhecido como *Feature Drift* ou *Covariate Shift*, ocorre quando a distribuição dos dados de produção é diferente dos dados de treinamento. O modelo não está equipado para lidar com esse desvio portanto, suas previsões podem não ser confiáveis. Na Figura 14 é um exemplo de mudança na distribuição dos dados devido a mudança no clima. O modelo pode não reconhecer como um farol e sim como uma montanha de gelo. Essa mudança impacta na capacidade de um modelo de visão

computacional de reconhecer objetos ou locais. A causa real do desvio pode ser atribuída a mudanças naturais no mundo real, mas também a problemas sistêmicos, como dados ausentes, erros de pipeline e alterações de esquema. É importante inspecionar os dados e rastreá-los ao longo do *pipeline* para identificar quando e onde o *drift* ocorre (GAMA et al., 2014):

- Janela de referência: Conjunto de pontos para comparar as distribuições de dados de produção para identificar o *drift*.
- Janela de teste: Conjunto de pontos a serem comparados com a janela de referência para determinar se ocorreu um *drift*.

Ao utilizar testes estatísticos, deve-se notar que testes paramétricos podem ser mais sensíveis a desvios em comparação com testes não paramétricos que são menos sensíveis. O bom senso e critério técnico na escolha dos recursos para monitorar e detectar variações são importantes. Escolher vários recursos pode dificultar o ajuste fino. Outro ponto extremamente importante é conhecer os dados e sua origem. Avaliar se há sazonalidade, mudança geográficas, mudanças culturais, janela de tempo ampla para coleta. Todos esses pontos podem influenciar no desempenho do modelo (MALLICK et al., 2022).

2.4 Concept Drift

O *Concept Drift* $P(Y|X)$ é uma mudança na relação real entre as entradas do modelo e a saída. Um exemplo de desvio de conceito é quando fatores macroeconômicos tornam os empréstimos mais arriscados e há um padrão mais alto para se qualificar para um empréstimo. Nesse caso, um nível de renda que antes era considerado digno de crédito não é mais digno de crédito. A mudança da características estatísticas da variável de destino implica em redução da acurácia do modelo (GRABUS; LOGAN; GREENBERG, 2022). Na Figura 16 é possível visualizar o limite de decisão alterado devido ao *Concept Drift* ao longo do tempo e o impacto nas predições.

$$P_{t1}(Y|X) \neq P_{t2}(Y|X) \quad (2.12)$$

O *Concept Drift* faz parte do ciclo de vida de modelos de *Machine Learning*. Ele é um forte indicador de necessidade de atualização e retreino de modelos de ML. À medida que os dados recebidos se afastam dos dados históricos que foram usados para treinamento, os relacionamentos e as correlações entre os recursos também mudam (SUÁREZ-CETRULO; QUINTANA; CERVANTES, 2023). Sob a ótica da matemática, *Concept Drift* é a alteração na distribuição $P(y|X)$, onde y é o rótulo real e X são os atributos disponíveis. O problema alcançado com *Concept Drift* é que a a predefinição de

Figura 14 – Exemplo de *Data Drift*, fotografias em estações diferente mesmo lugar



Fonte: Extraído de [Povenz \(2012\)](#).

dos modelos de *Machine Learning* é que a distribuição de treinamento reflete a distribuição de produção, caso contrário, não há segurança que o modelo treinado seja adequado para identificar corretamente o rótulo de saída ([AGRAHARI; SINGH, 2021](#)).

Na Figura 17 encontra-se a representação gráfica dos tipos de *Concept Drift* com base na variação da distribuição dos dados ao longo do tempo ([GAMA et al., 2014](#)).

Dessa forma é observado:

- **Abrupto:** A mudança Abrupta do conceito é caracterizada pela mudança rápida da relação distribuição X tempo, Figura 17 (a) ([GAMA et al., 2014](#); [GUO et al., 2022](#)).
- **Gradual:** Na gradual ocorrem mudanças que, ao longo do tempo, resultam em uma transição que resulta em mudança das distribuição, Figura 17 (b) ([GAMA et al., 2014](#); [GUO et al., 2022](#)).
- **Incremental:** O incremental tem uma razão de mudanças muito mais lenta, geralmente não é estatisticamente significativo quando comparado com a distribuição anterior porém ao longo do tempo ocorre a mudança na distribuição, Figura 17 (c) ([GAMA et al., 2014](#); [GUO et al., 2022](#)).

Figura 15 – Exemplo de *Data Drift*, fotografias em estações diferente mesmo lugar caatinga

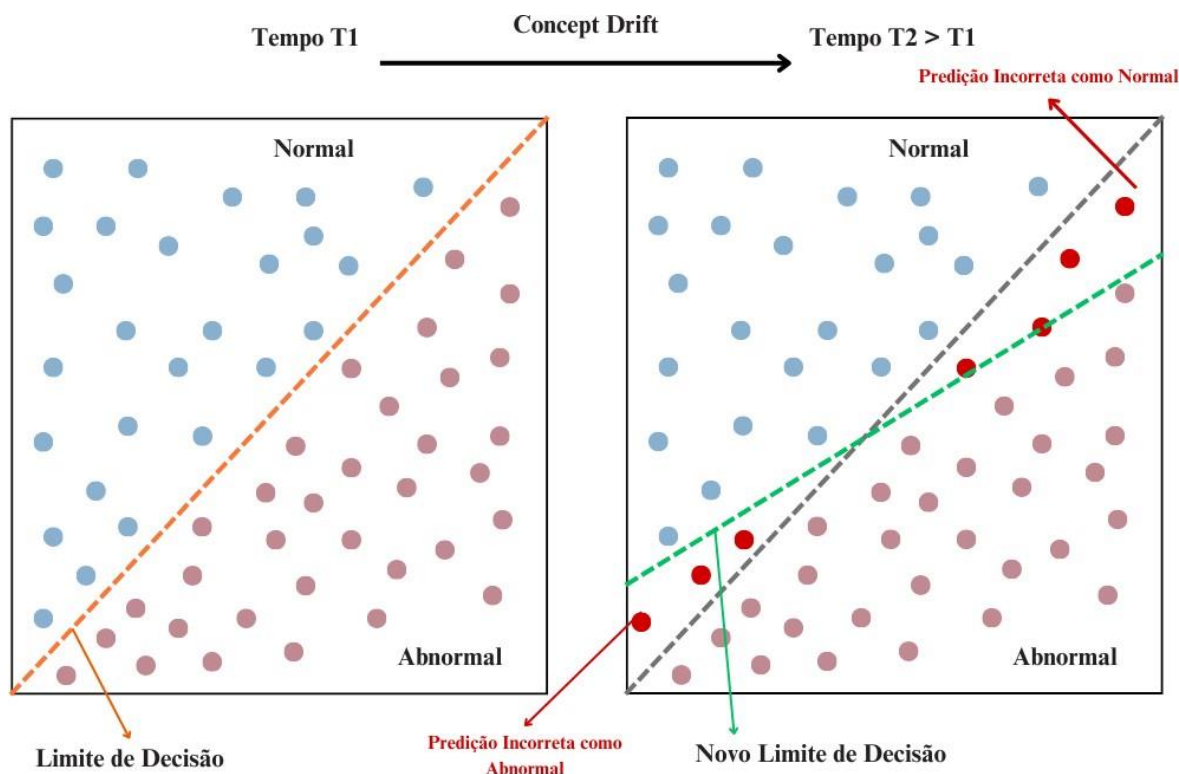
Fonte: Extraído de [Jurumi \(2022\)](#).

- **Recorrente:** No recorrente estados anteriores pode ressurgir de forma periódica, Figura 17 (d) ([GAMA et al., 2014](#); [GUO et al., 2022](#)).

Esses são os principais tipos de *Concept Drift* ([KRAWCZYK; CANO, 2018](#))

Pode ocorrer a mistura dos tipos de mudança isso pode apresentar características híbridas de tipos de *Concept Drift*. Ruídos não devem ser considerados como *drift* esses são sempre um problema real e parte de qualquer *pipeline* de dados. O Ruído pode ter uma natureza transitória e sendo assim não pode influenciar de forma definitiva o conceito. Filtros para garantir a qualidade dos dados devem ser implementados para reduzir a incidências de ruído nos dados ([BRZEZINSKI; STEFANOWSKI, 2014](#)).

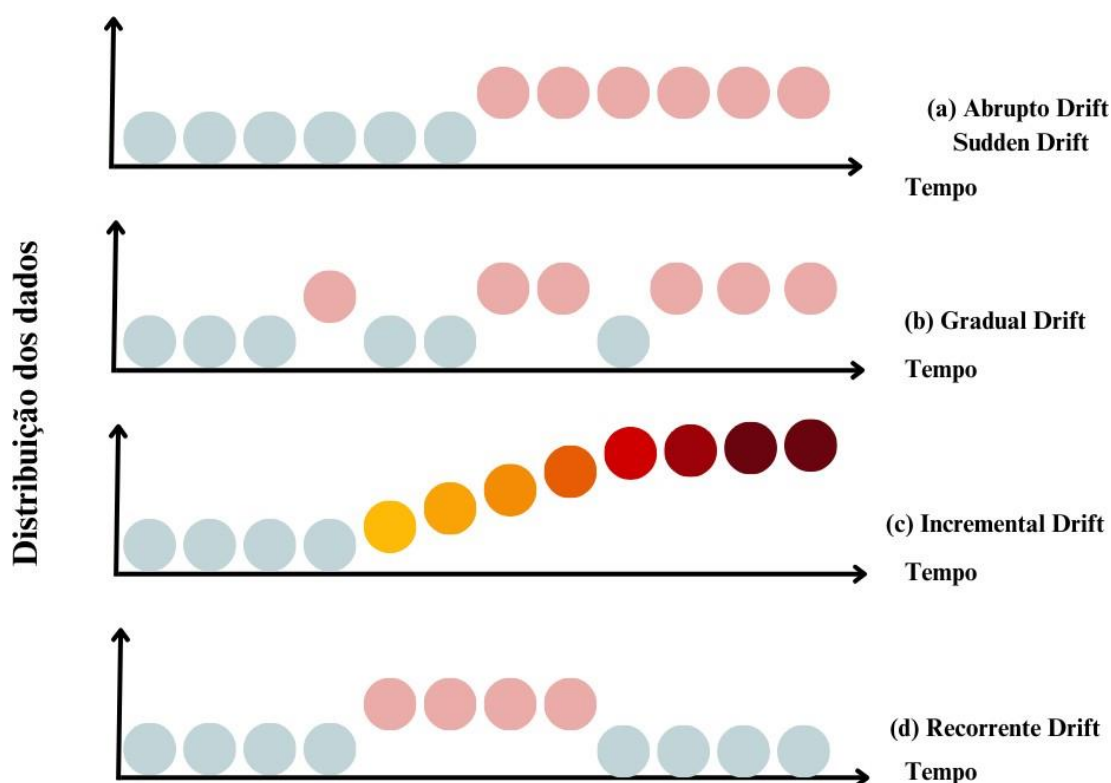
Pode ser visto como flutuações temporais no conceito de entrada que não se traduzem em mudanças reais, são prejudiciais ao sistema de classificação e, portanto, devem ser filtrados. Existem várias soluções para tratamento de fluxos de dados ruidosos propostas na literatura. No entanto, sua aplicação é limitada, pois é de se esperar que o ruído apareça. No entanto, em cenários da vida real ruído pode aparecer inesperadamente ou periodicamente ([KRAWCZYK; CANO, 2018](#)). Uma abordagem prática para detectar o *Concept Drift* é verificar a relação entre as estimativas do modelo e valores oriundos

Figura 16 – Ilustração *Concept Drift* em pontos de dados com limite de decisão.

Fonte: Adaptado de [Suárez-Cetrulo, Quintana e Cervantes \(2023\)](#).

dos dados de entrada. Se o erro de estimativa do modelo aumentar ao longo do tempo, os modelos não se ajustam mais aos dados atuais, o que pode indicar mudança no comportamento do sistema ([ZENISEK; HOLZINGER; AFFENZELLER, 2019](#)). Para lidar com a presença de *Concept Drift*, pode-se usar três soluções gerais:

- **Monitorar o desempenho do modelo:** As métricas de desempenho do modelo mais comuns incluem matriz de confusão, precisão, *recall*, *F1-Score* e Curva ROC ([TRIOLA, 2017](#); [HASTIE ROBERT TIBSHIRANI, 2009](#); [ALKAISSY et al., 2023](#)).
- **Monitorar as estatísticas descritivas:** As métricas da estatística descritiva oferecem medidas de distribuição e localização. O comportamento dessas métricas ao longo do tempo fornecem informações ricas sobre o possível *drift* dos modelos ([TRIOLA, 2017](#); [HASTIE ROBERT TIBSHIRANI, 2009](#); [ALKAISSY et al., 2023](#)).
- **Testes estatísticos:** Diversos testes estatístico e de hipótese podem auxiliar a entender o comportamento das distribuições dos dados ([TRIOLA, 2017](#); [HASTIE ROBERT TIBSHIRANI, 2009](#); [ALKAISSY et al., 2023](#)).

Figura 17 – Tipos de *Concept Drift*.

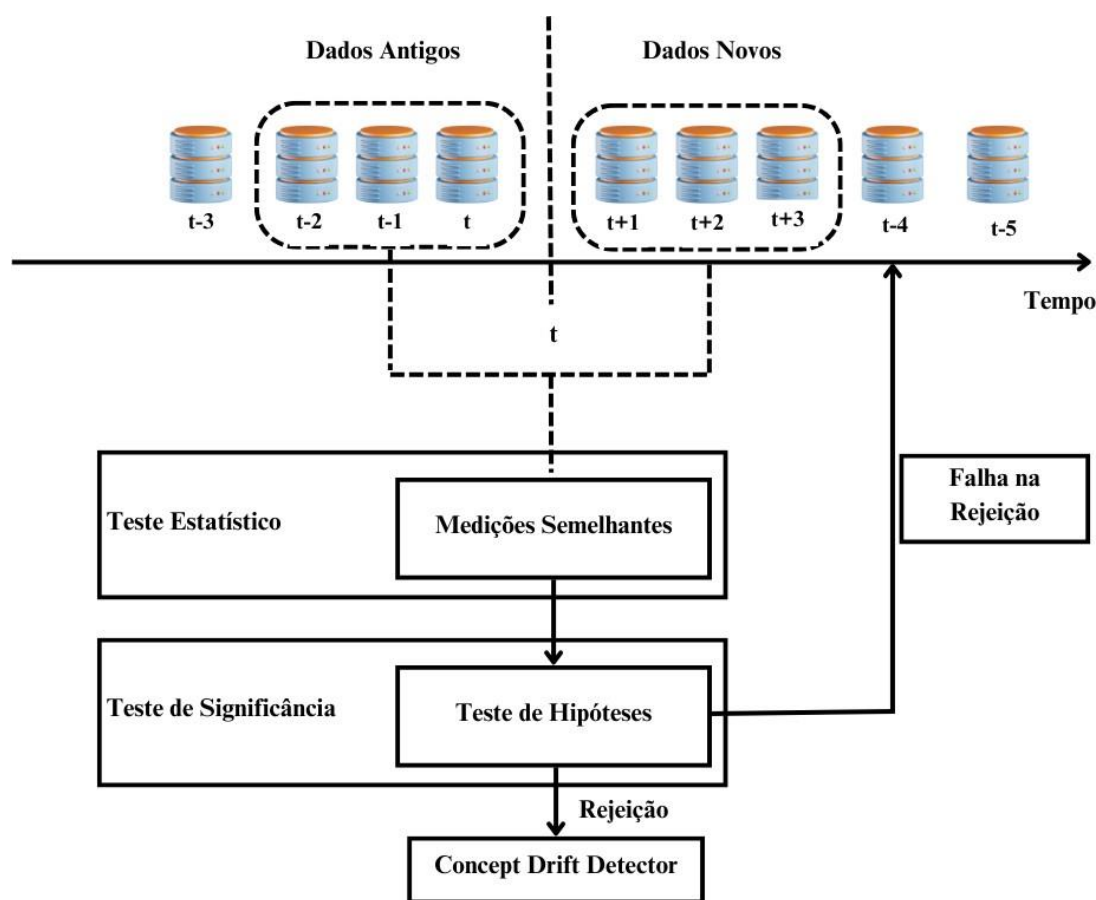
Fonte: Adaptado de Gama et al (2014).

Na Figura 18 é possível ver um exemplo de *framework* para detecção de *Concept Drift*.

2.5 Modelos de Visão computacional

A visão computacional busca replicar ou simular habilidades da visão humana, fornecendo métodos para a formação de imagens (imitando o sistema sensorial humano) e a percepção da máquina (imitando o sistema cognitivo humano) (GONZALEZ; WOODS, 2018). A abordagem moderna para imitar o sistema cognitivo humano consiste em métodos de aprendizado de máquina (ML) que são usados para extrair informações de imagens (LAKSHMANAN; GILLARD, 2021).

Visão computacional é utilizada na atualidade em diversas aplicações e sistemas. Inspeções de Máquina, locação, logística, imagens médicas, veículos autônomos, modelagem 3D, vigilância, reconhecimento biométrico são algumas das diversas aplicações (SZELISKI,

Figura 18 – *Framework* para detecção de *Concept Drift*.

Fonte: Adaptado de Bayram, Ahmed, Kassler (2022).

2022). Em todas as aplicações citadas há o fator tecnológico, ambiental e humano os quais fazem parte do processo de captura ou registro do dado. Aprendizado de máquina e a visão computacional possui aplicações inovadoras em engenharia, medicina, agricultura, astronomia, esportes, educação dentre outras (KHAN; AL-HABSI, 2020).

A visão computacional tem muitas aplicações que impactaram diversas indústrias, no entanto há algumas limitações. Assim como qualquer outro dispositivo conectado, há uma preocupação de que os dispositivos falhem devido a vírus computacional, defeitos de *hardware* ou *software* são sempre uma possibilidade (PATEL; ALNASER, 2020).

2.5.1 Propriedades da Imagem

- Proporção da tela ou *Aspect Ratio*, AR: Relação entre Altura (AL) e Largura da imagem (LG) (GONZALEZ; WOODS, 2018; SUN et al., 2021).

$$AR = \left(\frac{AL}{LG} \right) \quad (2.13)$$

- Área A: Área da imagem em *pixels* (GONZALEZ; WOODS, 2018; SUN et al., 2021).

$$A = \int_a^b f(x) dx \quad (2.14)$$

- Brilho ou *Brightness* (BR): Intensidade média dos *pixels* da imagem. Os canais de cores têm pesos diferentes de acordo com a fórmula para os canais *Red* (R), *Green* (G) and *Blue* (B) - RGB para escala de cinza. Na Equação 2.15, n refere-se ao número total de *pixels* na imagem e *Intensity* (I) refere-se ao valor de brilho do i -ésimo *pixels* na imagem. A Equação 2.15 calcula o brilho médio de todos os *pixels* da imagem (GONZALEZ; WOODS, 2018; LAVRUKHIN et al., 2021).

$$BR = \frac{1}{n} \sum_{i=1}^n I_i \quad (2.15)$$

- Contraste *Root Mean Square* (RMSC): Contraste da imagem, calculado pelo desvio padrão de *pixels*. Na Equação 2.16, n refere-se ao número total de *pixels* na imagem, *Intensity* (I) refere-se ao valor de brilho do i -ésimo *pixels* na imagem e *Mean Intensity* (MI) refere-se ao brilho médio de todos os *pixels* na imagem. A Equação 2.16 calcula a raiz quadrada média das diferenças entre o brilho de cada *pixels* e o brilho médio da imagem (GONZALEZ; WOODS, 2018; LAVRUKHIN et al., 2021).

$$RMSC = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (I_i - MI)^2} \quad (2.16)$$

- Intensidade Relativa Média do Vermelho (MRRI): Média de todos os *pixels* do canal vermelho, dimensionados para sua intensidade relativa em comparação com os outros canais. Na Equação 2.17, n refere-se ao número total de *pixels* na imagem, *Red Intensity* (RI) refere-se à intensidade da cor vermelha do i -ésimo *pixels* na imagem e *Intensity* (I) refere-se ao brilho geral ou intensidade de o i -ésimo *pixels* na imagem. A fórmula calcula a intensidade média da cor vermelha em relação ao brilho geral de todos os *pixels* da imagem (GONZALEZ; WOODS, 2018; LAVRUKHIN et al., 2021).

$$MRRI = \frac{1}{n} \sum_{i=1}^n \frac{RI_i}{I_i} \quad (2.17)$$

- Intensidade Relativa Média do Verde (MGRI): Média de todos os *pixels* do canal verde, dimensionados para sua intensidade relativa em comparação com os outros canais. Na Equação 2.18, n refere-se ao número total de *pixels* na imagem, *Green Intensity* (GI) refere-se à intensidade da cor verde do i -ésimo *pixel* na imagem e *Intensity* (I) refere-se ao brilho geral ou intensidade de o i -ésimo *pixel* na imagem. A Equação 2.18 calcula a intensidade média da cor verde em relação ao brilho geral

de todos os *pixels* da imagem (GONZALEZ; WOODS, 2018; LAVRUKHIN et al., 2021).

$$MGRI = \frac{1}{n} \sum_{i=1}^n \frac{GI_i}{I_i} \quad (2.18)$$

- Intensidade Relativa Média do Azul (MBRI): Média de todos os *pixels* do canal azul, dimensionados para sua intensidade relativa em comparação com os outros canais. Na Equação 2.19, n refere-se ao número total de *pixels* na imagem, *Blue Intensity* (BI) refere-se à intensidade da cor azul do i -ésimo *pixel* na imagem e *Intensity* (I) refere-se ao brilho geral ou intensidade de o i -ésimo *pixel* na imagem. A Equação 2.19 calcula a intensidade média da cor azul em relação ao brilho geral de todos os *pixels* da imagem (GONZALEZ; WOODS, 2018; LAVRUKHIN et al., 2021).

$$MBRI = \frac{1}{n} \sum_{i=1}^n \frac{BI_i}{I_i} \quad (2.19)$$

2.5.2 Arquiteturas CNN's

O campo da visão computacional viu uma revolução nos últimos anos, particularmente com a introdução de redes neurais convolucionais. Estas arquiteturas, otimizadas para lidar com imagens, têm estabelecido padrões de referência em várias tarefas. Nesta seção, discutiremos algumas das arquiteturas mais influentes e suas aplicações.

2.5.2.1 LeNet-5

Introduzida por Yann LeCun nos anos 90, a LeNet-5 é considerada uma das primeiras redes neurais convolucionais. Ela foi projetada principalmente para reconhecimento de dígitos manuscritos.

Estrutura: A LeNet-5 consiste em duas camadas convolucionais, seguidas por camadas de *pooling* e, finalmente, camadas totalmente conectadas.

Aplicações: Originalmente usada para o *dataset* de dígitos manuscritos MNIST, tornou-se o padrão de referência para tarefas similares.

2.5.2.2 AlexNet

A AlexNet, apresentada por Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton em 2012, é frequentemente creditada por reviver o interesse nas redes neurais convolucionais devido ao seu desempenho impressionante no desafio *ImageNet*.

Estrutura: Esta rede é composta por cinco camadas convolucionais, seguidas por três camadas totalmente conectadas.

Aplicações: AlexNet foi especificamente projetada para classificação de imagens e venceu o desafio *ImageNet ILSVRC* em 2012.

2.5.2.3 VGGNet

Desenvolvida pelo *Visual Geometry Group* da Universidade de Oxford em 2014, a VGGNet é conhecida por sua simplicidade e eficácia, utilizando apenas convoluções 3x3.

Estrutura: Oferece variantes com diferentes números de camadas (como VGG-16 e VGG-19), todas usando pequenas convoluções e camadas de *pooling*.

Aplicações: Além da classificação de imagens, a VGGNet é frequentemente usada como base para extração de características em outras tarefas, como detecção de objetos e segmentação semântica.

2.5.2.4 ResNet

Introduzida pela Microsoft Research em 2015, a ResNet (ou Redes Residuais) permite treinar redes muito mais profundas, abordando o problema do desvanecimento do gradiente.

Estrutura: Usa conexões de atalho ou saltos para evitar o problema do desvanecimento do gradiente em redes profundas. Variáveis como ResNet-50, ResNet-101 e ResNet-152 indicam o número de camadas na rede.

Aplicações: Embora seja usada para classificação de imagens, a ResNet também tem sido aplicada em outras tarefas de visão computacional devido à sua capacidade de treinar redes profundas.

2.5.2.5 DenseNet

Proposta em 2016 por Gao Huang, Zhuang Liu, Laurens van der Maaten e Kilian Q. Weinberger, a DenseNet (ou Rede Densa Conectada) é notável por sua estrutura de conexão densa, onde cada camada recebe entradas de todas as camadas anteriores e passa suas próprias características para todas as camadas subsequentes.

Estrutura: Em vez de usar apenas conexões de atalho (como a ResNet), a DenseNet conecta cada camada a todas as outras camadas em um padrão *feed-forward*. Isso resulta em conexões $l(l - 1)/2$ em uma rede com l camadas. Isso ajuda a melhorar a eficiência do fluxo de informações e gradientes ao longo da rede.

Aplicações: A DenseNet tem mostrado desempenho notável em várias tarefas de visão computacional, especialmente na classificação de imagens. Devido à sua eficiente reutilização de características, também é eficaz em conjuntos de dados com imagens de resolução comparativamente menor.

Estas arquiteturas desempenharam um papel fundamental na formação do campo moderno da visão computacional e servem como base para muitas outras pesquisas e aplicações.

2.6 Considerações Finais

Com base na fundamentação teórica apresentada neste capítulo, é possível concluir que a pesquisa desenvolvida possui uma base sólida e fundamentada em conhecimentos consolidados no campo de estudo. Através da revisão da literatura, foram discutidos os principais conceitos e teorias relacionados ao tema. Isso permite que as próximas seções tenham o arcabouço necessário para sua construção.

3 Material e Métodos

O capítulo de Material e Métodos é fundamental para a compreensão do processo de pesquisa empregado nesta dissertação. Ele fornece uma descrição detalhada dos instrumentos, técnicas, procedimentos e métodos utilizados durante o estudo. Além disso, este capítulo garante que o estudo possa ser replicado por outros pesquisadores, o que é uma parte essencial da validação e verificação científica.

3.1 Base de dados

3.1.1 Descrição do *Dataset* MURA

O *dataset* MURA (*Musculoskeletal Radiographs*) é um conjunto de dados médicos de larga escala que contém radiografias de partes do corpo humano. Ele foi desenvolvido para permitir a pesquisa em algoritmos de aprendizado de máquina para detectar anormalidades musculoesqueléticas.

1. **Origem:** MURA é uma das várias iniciativas do grupo de pesquisa em *Machine Learning* para Saúde da Stanford University.
2. **Imagens e Estudos:**
 - Contém **40.895** radiografias de **14.863** estudos.
 - As radiografias são tiradas em diferentes posições e ângulos, por exemplo, vistas anteroposteriores (AP) e laterais.
3. **Partes do corpo cobertas:** O conjunto de dados cobre diversas regiões do corpo, incluindo:
 - Ombro
 - Cotovelo
 - Punho
 - Mão
 - Dedo
 - Joelho
 - Tornozelo
4. **Anotações:**

- Cada estudo no MURA é anotado por radiologistas como normal ou anormal.
- Essas anotações ajudam os pesquisadores a treinar e validar algoritmos de diagnóstico automático.

5. **Divisão do conjunto de dados:**

- O *dataset* MURA é dividido em conjuntos de treinamento e validação.

6. **Fonte das Imagens:** As imagens do MURA provêm de pacientes que buscaram atendimento no Stanford Hospital.

7. **Formato das Imagens:** As imagens estão no formato PNG.

8. **Utilização na pesquisa:** MURA tem sido usado como um *benchmark* na comunidade de pesquisa.

9. **Disponibilidade e Licença:** O *dataset* MURA é de acesso público, mas é importante verificar os termos de uso específicos.

10. **Desafios:** Além de apenas fornecer o conjunto de dados, a equipe por trás do MURA também propôs desafios para encorajar a comunidade de pesquisa a desenvolver algoritmos que superassem os métodos existentes.

A intenção do MURA é avançar na capacidade dos algoritmos de aprendizado de máquina de diagnosticar de forma eficaz e precisa anormalidades musculoesqueléticas.

As imagens neste conjunto de testes, foram rotuladas com a seguinte metodologia conforme (RAJPURKAR et al., 2017). O voto majoritário de um grupo de três radiologistas serve como padrão ouro.

3.2 Tecnologias

Esta seção destina-se a fornecer uma visão abrangente das tecnologias utilizadas ao longo desta pesquisa. O objetivo é dar contexto sobre as ferramentas e plataformas que facilitaram a realização do estudo, permitindo uma compreensão mais profunda dos métodos e procedimentos descritos em seções anteriores e garantindo a replicabilidade da pesquisa por outros estudiosos interessados.

3.2.1 Python

A linguagem de programação adotada é a linguagem Python na versão 3.9.5. Os Códigos de *Extract, Transform, Load* - (ETL), geração de ruído e parte do modelo é desenvolvida em Python (FOUNDATION, 2023).

O Python é uma linguagem de programação de alto nível, amplamente reconhecida por sua sintaxe clara e legível, tornando-a especialmente adequada para aplicações educacionais, pesquisa e desenvolvimento. Neste estudo, Python foi empregado por várias razões:

Python é uma linguagem versátil que pode ser executada em quase todos os sistemas operacionais modernos. Seu extenso ecossistema de bibliotecas torna a linguagem apropriada para uma variedade de tarefas, desde análise de dados até desenvolvimento *web* e aprendizado de máquina.

3.2.2 Pytorch

PyTorch é uma biblioteca de alto nível para *Machine Learning*, manipulação de tensores e otimizada para aprendizado profundo usando unidade de processamento gráfico, *Graphics Processing Unit* (GPUs). Dessa forma é possível obter uma melhor qualidade e velocidade no desenvolvimento. O Modelo é desenvolvido em PyTorch ([PYTORCH, 2023](#)).

O PyTorch, desenvolvido pelo Facebook's AI Research lab, é uma biblioteca de computação tensorial com aceleração GPU, amplamente reconhecida por sua flexibilidade e eficiência, tornando-a uma escolha predominante para pesquisadores e desenvolvedores que trabalham com aprendizado profundo. Neste estudo, o PyTorch foi escolhido por várias razões:

Diferentemente de outras bibliotecas de aprendizado profundo, o PyTorch opera com grafos computacionais dinâmicos, também conhecidos como *define-by-run*. Isso significa que o grafo é construído *on-the-fly* à medida que as operações são realizadas, permitindo uma modelagem mais intuitiva e uma depuração mais fácil.

O PyTorch oferece eficiência em suas operações, especialmente com o uso de GPUs, o que é crucial para treinar modelos de aprendizado profundo, que geralmente requerem grandes quantidades de dados e poder computacional.

Com uma variedade de ferramentas e extensões, como o *torchvision* para tarefas de visão computacional e o *torchaudio* para áudio, o PyTorch fornece recursos abrangentes para muitas aplicações de aprendizado profundo.

A comunidade PyTorch é ativa e em crescimento, com muitos recursos, tutoriais e uma vasta literatura, o que facilitou a resolução de problemas e a implementação de técnicas avançadas durante a pesquisa.

Neste estudo, o PyTorch foi empregado para construir e treinar um modelo de rede neural convolucional para classificação de imagens.

A decisão de usar PyTorch nesta pesquisa foi baseada em sua flexibilidade, eficiência e rica base de recursos. Ele provou ser uma ferramenta valiosa, permitindo a implementação

eficaz e eficiente dos modelos de aprendizado profundo necessários para este estudo.

3.2.3 *Deepchecks*

O *Deepchecks* é um pacote *Open Source* para validação de modelos de *Machine Learning*. Há duas aplicações do *Deepchecks* uma para dados tabulares e outra para visão computacional. Este último é o utilizado no desenvolvimento deste trabalho.

Nos últimos anos, à medida que os modelos de aprendizado de máquina tornaram-se mais complexos e sua adoção aumentou em diversas aplicações críticas, a necessidade de validações rigorosas também cresceu. Neste estudo, a biblioteca *deepchecks* foi utilizada para garantir a integridade e confiabilidade de nosso modelo.

Deepchecks inclui uma ampla gama de verificações específicas para validação de modelos. De inconsistências de dados a problemas de distribuição e erros no modelo, a biblioteca proporciona uma visão aprofundada dos possíveis problemas que um modelo pode ter.

O *deepchecks* permite:

- Identificar rapidamente problemas nos dados que poderiam afetar adversamente o desempenho do modelo.
- Garantir que as distribuições dos conjuntos de treinamento e teste fossem consistentes.
- Detectar possíveis vieses no modelo, assegurando assim que as inferências do modelo fossem justas e imparciais.

Com foco na integridade dos dados ingeridos, validação dos teste e treinamento do modelo e na avaliação da evolução do modelo o *Deepchecks* concentra recursos para auxiliar as esteiras automatizadas de ML ([DEEPCHECKS, 2023](#)).

3.2.4 Colab

Google Colab é uma ferramenta *web* disponibilizada pelo Google para programação em Python sem necessidade de configurações, fácil compartilhamento e com acesso a recursos de GPU de forma gratuita. Com interface bastante amigável e com a infraestrutura necessária para a utilização do Python e Pytorch. Por ser do Google oferece integração ao *Drive* do Google ([GOOGLE, 2023](#)).

À medida que os modelos de aprendizado de máquina tornam-se mais sofisticados, o poder computacional necessário para treinar esses modelos também cresce. Para superar

as limitações de *hardware* e facilitar o desenvolvimento, este estudo utilizou o Google Colab, uma plataforma de pesquisa gratuita fornecida pela Google.

Google Colab oferece um ambiente de *notebook Jupyter* que não requer configuração e fornece acesso gratuito a GPUs e TPUs. As principais vantagens de usar o Google Colab em nossa pesquisa incluem:

- **Acesso Gratuito a GPUs:** Permite treinar modelos de aprendizado profundo de forma eficiente.
- **Facilidade de Uso:** Oferece um ambiente familiar para quem já está acostumado com *Jupyter Notebooks*.
- **Integração com Google Drive:** Facilitou o armazenamento e o acesso a *datasets* e outros arquivos relacionados.

Neste estudo, o Google Colab foi primordial para treinar a rede neural devido ao seu alto requisito computacional que não poderia ser atendido por máquinas locais. Apesar de suas vantagens, também enfrentamos algumas limitações com o Google Colab, incluindo sessões temporárias e limitações de tempo de uso da GPU. No entanto, com backups regulares e gestão eficiente do tempo, essas questões foram contornadas.

3.3 Métricas

A seção de métricas tem como objetivo apresentar e explicar diferentes medidas utilizadas na análise de dados. Serão abordados conceitos como média, mediana, moda, desvio padrão, variância, correlação, covariância, entre outros.

3.3.1 Métricas Univariadas e Multivariadas

Métricas univariadas são usadas para descrever e analisar uma única variável em um conjunto de dados. Já as métricas multivariadas são usadas para analisar as relações entre duas ou mais variáveis em um conjunto de dados. Abaixo, estão algumas das métricas mais comuns para ambas abordagens.

3.3.1.1 Métricas Univariadas

- **Média:** a média é a soma de todos os valores de uma variável dividida pelo número de observações.

$$\bar{x} = \left(\frac{1}{n} \sum_{i=1}^n x_i \right) \quad (3.1)$$

- Mediana: a mediana é o valor que separa a metade superior e inferior de uma distribuição de dados.

$$M_d(x) = x_{[\frac{n+1}{2}]}, \text{ quando } n \text{ for Impar.} \quad (3.2)$$

$$M_d(x) = \frac{x_{[\frac{n}{2}]} + x_{[\frac{n}{2} + 1]}}{2}, \text{ quando } n \text{ for Par.} \quad (3.3)$$

- Moda: a moda é o valor que ocorre com a maior frequência em um conjunto de dados. Na Equação 3.4, $f(x)$ representa a frequência do valor x . A expressão $\arg \max_x$ identifica o valor de x que maximiza a frequência $f(x)$.

$$M_o(x) = \arg \max_x f(x) \quad (3.4)$$

Por exemplo, considerando um conjunto de dados simples como $\{1, 2, 2, 3, 3, 3, 4\}$, a função $f(x)$ retornaria $f(1) = 1$, $f(2) = 2$, $f(3) = 3$ e $f(4) = 1$. Assim, aplicando a equação 3.4, encontraríamos a moda $M_o(x)$ sendo 3, pois é o valor que maximiza a frequência $f(x)$.

- Desvio padrão: o desvio padrão mede o quão dispersos os valores de uma variável estão em relação à média. A equação para o desvio padrão amostral, tal que $x_1, x_2, \dots, x_n$ são os n elementos do conjunto de dados, e n é o tamanho da amostra. O símbolo sigma representa o desvio padrão e o símbolo $\bar{x}$ representa a média dos elementos do conjunto de dados.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (3.5)$$

A Equação 3.5 calcula o desvio padrão como a raiz quadrada da soma dos quadrados das diferenças entre cada elemento e a média, dividido pelo tamanho da amostra menos 1. Isso mede a variabilidade dos dados em relação à média.

- Variância: a variância (var) é a medida de quão longe cada valor em um conjunto de dados está do valor médio.

$$var(X) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (3.6)$$

Na Equação 3.6, X é uma variável aleatória, n é o tamanho da amostra, X_i é a i -ésima observação de X e $\bar{X}$ é a média da amostra.

- Quartis: os quartis são valores que dividem uma distribuição de dados em quatro partes iguais, cada parte representando 25% do conjunto de dados.

$$Q1 = x_{(n+1)/4} \quad (3.7)$$

$$Q2 = x_{(n+1)/2} \quad (3.8)$$

$$Q3 = x_{3(n+1)/4} \quad (3.9)$$

- Amplitude (AT): a amplitude é a diferença entre o valor máximo e mínimo em um conjunto de dados.

$$AT = \max(X) - \min(X) \quad (3.10)$$

- Coeficiente de variação (cov): o coeficiente de variação é o desvio padrão dividido pela média, expresso em porcentagem, e é usado para comparar a variação de diferentes conjuntos de dados.

$$cov = \frac{\sigma}{\bar{x}} \times 100\% \quad (3.11)$$

([TRIOLA, 2017](#); [ANDAU NAVARRO et al., 2023](#); [LAMB et al., 2022](#))

3.3.1.2 Métricas multivariadas

- Covariância: a covariância mede a relação linear entre duas variáveis. Uma covariância positiva significa que as duas variáveis aumentam juntas, enquanto uma covariância negativa significa que elas se movem em direções opostas. Na Equação 3.12 X e Y são variáveis aleatórias, n é o tamanho da amostra, X_i é a i -ésima observação de X , Y_i é a i -ésima observação de Y , $\bar{X}$ é a média amostral de X e $\bar{Y}$ é a média amostral de Y . A expressão $cov(X, Y)$ representa a covariância entre X e Y , que mede o quanto duas variáveis variam juntas.

$$cov(X, Y) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) \quad (3.12)$$

- Coeficiente de correlação: o coeficiente de correlação é uma medida padronizada da relação linear entre duas variáveis, variando de -1 a +1. Um coeficiente de correlação de 0 indica nenhuma correlação, enquanto um coeficiente de 1 ou -1 indica uma correlação perfeita. Na Equação 3.13 X e Y são variáveis aleatórias, n é o tamanho da amostra, X_i é a i -ésima observação de X , Y_i é a i -ésima observação de Y , $\bar{X}$ é a média amostral de X e $\bar{Y}$ é a média amostral de Y . A expressão r_{XY} representa o coeficiente de correlação entre X e Y , que mede a força e a direção da relação linear entre as duas variáveis. O valor de r_{XY} varia de -1 a 1, onde -1 indica uma relação

linear negativa perfeita, 0 indica nenhuma relação linear e 1 indica uma relação linear positiva perfeita.

$$r^{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (3.13)$$

- **Análise de regressão:** a análise de regressão é uma técnica usada para modelar a relação entre duas ou mais variáveis. A regressão linear é uma das técnicas mais comuns, onde uma Equação é criada para prever uma variável de resposta com base em uma ou mais variáveis preditoras. Na Equação 3.14 Y é a variável dependente, X é a variável independente, θ_0 é o coeficiente de interceptação, θ_1 é o coeficiente de inclinação e ϵ é o termo de erro. A Equação representa uma relação linear entre Y e X, onde Y é uma função de X com uma interceptação de θ_0 e uma inclinação de θ_1 . O termo de erro ϵ representa o desvio do valor observado de Y do valor previsto com base na Equação de regressão.

Os coeficientes θ_0 e θ_1 podem ser estimados pelo método dos mínimos quadrados. O objetivo da análise de regressão linear é encontrar os valores de θ_0 e θ_1 que minimizem a soma dos erros quadrados entre os valores observados de Y e os valores previstos com base na Equação de regressão. Uma vez estimados os coeficientes, a Equação de regressão pode ser usada para prever o valor de Y para um determinado valor de X.

$$Y = \theta_0 + \theta_1 X + \epsilon \quad (3.14)$$

- **Análise de componentes principais:** a análise de componentes principais é uma técnica usada para reduzir a dimensionalidade de um conjunto de dados, identificando as principais fontes de variação nos dados. Ela procura transformar as variáveis originais em um conjunto menor de variáveis não correlacionadas, conhecidas como componentes principais. Na Equação 3.15 T são os dados transformados, X são os dados originais e W é a matriz de transformação. O objetivo do PCA é encontrar um novo conjunto de variáveis, chamadas de componentes principais, que capturem a quantidade máxima de variância nos dados originais. Para calcular os componentes principais, primeiro é calculado a matriz de covariância dos dados originais X. Em seguida, é realizado uma decomposição de autovalor da matriz de covariância para obter os autovetores e autovalores. Os autovetores correspondem aos componentes principais e os autovalores representam a quantidade de variância explicada por cada componente principal. A matriz de transformação W é formada selecionando os k autovetores superiores com os maiores autovalores, onde k é o número desejado de componentes principais. Os dados T transformados são obtidos multiplicando os dados originais X pela matriz de transformação W. O T resultante contém o novo conjunto de variáveis que capturam a quantidade máxima de variância nos dados

originais.

$$\mathbf{T} = \mathbf{XW} \quad (3.15)$$

- **Análise discriminante:** a análise discriminante é uma técnica usada para determinar as diferenças entre grupos de observações com base em várias variáveis medidas. Ela procura encontrar as variáveis mais importantes que discriminam entre os grupos. Na Equação 3.16 w é o vetor discriminante, S_w é a matriz de covariância agrupada dentro da classe, m_1 e m_2 são as médias das duas classes e -1 indica o inverso de uma matriz. O objetivo é encontrar uma combinação linear das variáveis originais que maximize a separação entre as duas classes. Para calcular o vetor discriminante, primeiro é calculado a matriz de covariância dentro da classe S_w . Essa matriz representa a variância agrupada de cada variável dentro de cada classe. Em seguida, é calculada as médias m_1 e m_2 de cada classe. Finalmente, é calculado o vetor discriminante w tomando o inverso de S_w e multiplicando-o pela diferença entre as médias das duas classes. Uma vez que definido o vetor discriminante w , é possível classificar novas observações calculando sua combinação linear com w e comparando-a com um valor limite. Se a combinação linear for maior que o limiar, a observação é classificada como pertencente a uma classe, caso contrário é classificada como pertencente à outra classe.

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \quad (3.16)$$

(TRIOLA, 2017; STEFANIA et al., 2023; CAO et al., 2023).

3.3.2 Teste de Cramer's V

O teste de Cramer's V (V), também conhecido como teste de associação de Cramer, é um teste estatístico usado para avaliar a associação entre duas variáveis categóricas. Ele mede a força da associação entre as variáveis usando um coeficiente de associação chamado de coeficiente de Cramer (CRAMER, 2016).

O coeficiente de Cramer é calculado a partir de uma tabela de contingência, que mostra a distribuição conjunta das duas variáveis categóricas. O coeficiente varia entre 0 e 1, onde 0 indica nenhuma associação e 1 indica uma associação perfeita. Valores intermediários indicam uma associação parcial (CRAMER, 2016; BABU; GAJANAN, 2022).

O teste de Cramer é aplicado para verificar se a relação entre as variáveis categóricas é significativa ou não. A hipótese nula é que não há associação entre as variáveis, enquanto a hipótese alternativa é que há associação entre elas. O teste produz um valor de estatística e um valor-p, que é interpretado como a probabilidade de obter uma associação tão forte

ou mais forte que a observada, sob a hipótese nula (CRAMER, 2016; WU; ZHANG; ZHAO, 2014).

Se o valor-p for menor do que o nível de significância pré-definido, geralmente 0,05, então a hipótese nula é rejeitada e conclui-se que há evidência suficiente para afirmar que há associação entre as variáveis categóricas. Caso contrário, não há evidência suficiente para afirmar que há associação entre as variáveis categóricas.

$$V = \sqrt{\frac{\chi^2}{n(k-1)}} \quad (3.17)$$

Em que χ^2 é a estatística qui-quadrado, n é o número total de observações e k é o número de categorias para cada variável (CRAMER, 2016; TAO et al., 2023).

Limites Aceitáveis para *drift* usando Cramer's V

O valor de Cramer's V varia entre 0 e 1. Os limites aceitáveis podem ser definidos como:

O *Drift Score* usando o Cramer's V é uma métrica comum para avaliar a magnitude da diferença entre duas distribuições categóricas. O valor do Cramer's V varia entre 0 e 1, onde:

- **0** indica que as duas distribuições são idênticas.
- **1** indica que as duas distribuições são completamente diferentes.

Para definir limites aceitáveis para o *drift* usando o Cramer's V, é necessário levar em consideração o domínio específico do problema, o tamanho da amostra e a importância de detectar pequenas mudanças. No entanto, uma regra geral frequentemente usada é:

- $0 \leq V < 0.1$: Muito pequena associação (*drift* aceitável).
- $0.1 \leq V < 0.3$: Pequena associação (leve preocupação, monitorar de perto).
- $0.3 \leq V < 0.5$: Associação moderada (preocupante, talvez requerendo ação).
- $V \geq 0.5$: Forte associação (forte evidência de *drift*, ação provavelmente necessária).

Porém, é essencial destacar que a decisão sobre o que é aceitável é fortemente dependente do contexto. Em algumas aplicações críticas, mesmo um Cramer's V de 0.1 pode ser motivo de preocupação, enquanto em outros cenários, um valor de 0.3 pode ser aceitável (CRAMER, 2016; TAO et al., 2023).

3.3.3 Wasserstein ou *Earth Movers Distance*

A distância de Wasserstein, $W_p(P, Q)$, também conhecida como Distância do Transporte Terrestre (*Earth Mover's Distance* - EMD), é uma métrica utilizada para medir a distância entre duas distribuições de probabilidade. O nome Distância do Transporte Terrestre vem da analogia de imaginar as duas distribuições de probabilidade como pilhas de terra e a distância de Wasserstein como a quantidade de trabalho (ou distância) necessário para moldar uma pilha de terra em outra (CHEN et al., 2023).

A distância de Wasserstein leva em consideração as relações espaciais entre os pontos das duas distribuições de probabilidade, em vez de apenas sua forma ou densidade geral. Isso a torna uma métrica mais poderosa do que outras medidas de distância, como a distância Euclidiana, que considera apenas a distância entre pontos individuais (CHEN et al., 2023).

A distância de Wasserstein tem aplicações em muitos campos, incluindo processamento de imagens, visão computacional e processamento de linguagem natural. É particularmente útil em situações em que as distribuições sendo comparadas têm uma estrutura complexa ou de alta dimensionalidade (FENG; LIU; ZHAO, 2023).

Em resumo, a distância de Wasserstein (ou Distância do Transporte Terrestre) é uma ferramenta útil para medir a distância entre duas distribuições de probabilidade, levando em consideração suas relações espaciais em vez de apenas sua forma ou densidade geral. Dessa modo observamos a distância entre duas distribuições de probabilidade P e Q em um espaço métrico (X, d) dada pela Equação 3.18, onde $\Gamma(P, Q)$ é o conjunto de todas as distribuições conjuntas $\gamma(x, y)$ cujos marginais são P e Q , respectivamente, e $p \geq 1$ é um parâmetro que determina a ordem da distância. Às vezes, a distância também é indicada como $W_p(P, Q)$, onde $p = 1$ corresponde à distância do motor terrestre e $p = 2$ corresponde à distância de Wasserstein (GAUNT; LI, 2023; WEI et al., 2023).

$$W_p(P, Q) = \inf_{\gamma \in \Gamma(P, Q)} \left(\int_{X \times X} d(x, y)^p d\gamma(x, y) \right)^{1/p} \quad (3.18)$$

A Wasserstein *distance* por ser uma métrica de distância entre duas distribuições de probabilidade é frequentemente utilizada em domínios onde é importante considerar a diferença de localização entre as amostras nas distribuições.

Diferente de métricas como a distância de Kullback-Leibler (KL) ou o Cramer's V, a Wasserstein *distance* tem uma interpretação mais geométrica. É uma medida do custo mínimo de transformar uma distribuição na outra, considerando o transporte das massas de probabilidade.

Quanto aos limites aceitáveis para a Wasserstein *distance*:

- **0:** As duas distribuições são idênticas.
- Valores maiores indicam uma diferença crescente entre as distribuições.

Não há um valor estrito ou universal para determinar o que é alto ou baixo. A interpretação do valor da Wasserstein *distance* depende do contexto e da aplicação específica. Por exemplo:

- Em um domínio onde as distribuições tendem a ser muito semelhantes, uma Wasserstein *distance* de 0.1 pode ser considerada alta.
- Em contraste, em um domínio onde as distribuições são naturalmente variadas, um valor de 0.1 pode ser considerado baixo.

O mais importante ao usar a Wasserstein *distance* é:

- **Estabelecer uma base de referência:** Isso pode ser feito calculando a Wasserstein *distance* entre distribuições conhecidas ou utilizando um conjunto de treinamento e comparação históricos para determinar o que é típico para o seu domínio.
- **Monitorar ao longo do tempo:** Detectar mudanças na Wasserstein *distance* ao longo do tempo pode ser mais informativo do que qualquer valor absoluto.
- **Considerar o contexto:** Em algumas aplicações, mesmo pequenas mudanças na distribuição podem ser críticas. Em outros casos, podem ser aceitáveis variações maiores.

Se for necessário definir um limite específico, recomenda-se trabalhar em estreita colaboração com especialistas no domínio e considerar testes de hipóteses ou técnicas de *bootstrapping* para determinar os limites aceitáveis com base em dados históricos ou de referência.

3.3.4 O teste de Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov (K-S) é um teste não paramétrico utilizado em estatística para determinar se duas distribuições de probabilidade unidimensionais subjacentes diferem, ou para comparar uma amostra com uma distribuição de probabilidade de referência. Sua utilidade advém do fato de não fazer suposições sobre a distribuição dos dados, sendo, portanto, não paramétrico, e de sua capacidade de comparar as formas das distribuições, não se restringindo apenas a seus parâmetros (PORWIK; DADZIE, 2022).

Existem duas formas do teste K-S:

1. **Teste K-S de uma amostra:** Compara a função de distribuição acumulada (CDF) de uma amostra com a CDF de uma distribuição teórica especificada. Esta modalidade é útil para testar se uma amostra origina-se de uma população com uma distribuição específica (PORWIK; DADZIE, 2022).
2. **Teste K-S de duas amostras:** Compara as $CDFs$ de duas amostras independentes para determinar se elas provêm da mesma distribuição. Este formato não pressupõe que as distribuições comparadas sejam normais, nem que apresentem a mesma forma, mas sim que os parâmetros das distribuições sejam os mesmos (PORWIK; DADZIE, 2022).

3.3.4.1 Funcionamento do Teste K-S

A estatística K-S quantifica a distância entre a função de distribuição empírica da amostra e a função de distribuição acumulada da distribuição de referência, ou entre as funções de distribuição empíricas de duas amostras. Esta distância é definida como a máxima diferença absoluta entre as duas funções (OHUNAKIN et al., 2024; PORWIK; DADZIE, 2022).

Para o teste K-S de uma amostra, a estatística D é calculada como:

$$D = \sup_x |F_n(x) - F(x)| \quad (3.19)$$

onde $F_n(x)$ é a função de distribuição empírica da amostra e $F(x)$ é a função de distribuição acumulada da distribuição teórica especificada.

Para o teste K-S de duas amostras, a estatística D é calculada como:

$$D = \sup_x |F_{n1}(x) - F_{n2}(x)| \quad (3.20)$$

onde $F_{n1}(x)$ e $F_{n2}(x)$ são as funções de distribuição empíricas das duas amostras.

O símbolo $\sup$ na equação do teste de K-S representa o supremo, que é um conceito matemático usado para descrever o limite superior dentro de um conjunto de valores. No contexto das equações do teste K-S, o $\sup$ é usado para encontrar a máxima diferença absoluta entre as funções de distribuição acumulada ($CDFs$) comparadas (PORWIK; DADZIE, 2022; ALABBASI; KHALIL; MCGRAIL, 2023; OHUNAKIN et al., 2024).

Regra de Decisão

A hipótese nula (H_0) para o teste K-S afirma que a amostra origina-se da distribuição especificada, ou, no caso do teste K-S de duas amostras, que as duas amostras provêm da mesma distribuição.

Se o D calculado for maior que o valor crítico da tabela do teste K-S em um nível de significância escolhido (α), a hipótese nula é rejeitada, sugerindo que as distribuições são significativamente diferentes (ZHOU et al., 2024).

Vantagens e Limitações

Vantagens:

- O teste é não paramétrico e não requer a suposição de normalidade.
- É mais sensível a diferenças nas formas das distribuições do que testes que comparam apenas parâmetros, como médias ou variâncias.

Limitações:

- O teste K-S é sensível ao tamanho da amostra, com amostras muito grandes podendo levar à rejeição da hipótese nula mesmo para pequenos desvios da distribuição teórica.
- Para o teste de uma amostra, pressupõe-se que a distribuição teórica é completamente especificada sem parâmetros estimados a partir dos dados. Caso os parâmetros sejam estimados, ajustes são necessários, ou devem ser considerados testes alternativos.

O teste K-S é amplamente utilizado em diversos campos, incluindo controle de qualidade, ciência ambiental e economia, para testar hipóteses sobre formas distribucionais ou para comparar dados empíricos com modelos teóricos (COHEN; GRINSHPOUN; WEISS, 2024).

3.3.5 Detecção pelo Domínio do Classificador

A detecção pelo domínio do classificador é uma técnica usada em aprendizado de máquina para lidar com dados desbalanceados. Nesse contexto, dados desbalanceados são aqueles em que uma classe de interesse é muito menor em número do que outras classes.

Na detecção pelo domínio do classificador, o objetivo é identificar padrões nas características dos dados que são exclusivos da classe de interesse. Isso pode ser feito treinando um classificador para distinguir entre dados da classe de interesse e dados das outras classes, usando características que são específicas para a classe de interesse.

Considere o cenário em que se deseja treinar um classificador para detectar fraudes em transações financeiras. Se a taxa de fraude é muito baixa, o conjunto de dados será desbalanceado, pois a maioria das transações será legítima. Nesse caso, a detecção pelo domínio do classificador poderia ser usada para identificar características exclusivas das

transações fraudulentas, como valores anormais, locais incomuns de transação ou padrões de comportamento suspeitos.

A detecção pelo domínio do classificador é uma técnica útil para lidar com dados desbalanceados, mas requer um conhecimento profundo do domínio do problema e uma seleção cuidadosa de características relevantes. Além disso, é importante avaliar cuidadosamente a eficácia do classificador usando métricas adequadas, como precisão, *recall* e *F1-score*.

O treinamento de um classificador de domínio é um método para detectar desvios multivariados, o que significa que ele pode ser executado em várias variáveis e até mesmo em todo o conjunto de dados. Isso é feito treinando um modelo para classificar se uma amostra veio do conjunto de dados do trem ou do conjunto de dados mais recente (teste ou produção). Se o classificador puder prever facilmente qual amostra é de qual conjunto de dados, isso significaria que existem diferenças significativas entre esses conjuntos de dados (DEEPCHECKS, 2023).

A principal vantagem desse método é que ele também pode descobrir desvios de covariáveis, ou seja, desvios nos dados que não afetam a distribuição de cada variável individual, mas afetam a relação entre eles (DEEPCHECKS, 2023).

Nas verificações *Multivariate Drift* e *Image Dataset Drift* são mesclados os dados de Treino e de Teste e atribuímos rótulo 0 às amostras que vêm do conjunto de treinamento e 1 às que são do conjunto de teste. Em seguida, treinamos um classificador binário do tipo Árvore de classificação de aumento de gradiente baseada em histograma. Em seguida, normalizamos a pontuação AUC desse classificador e a usamos como pontuação de *Drift*, pois quanto maior a AUC, melhor o modelo, o que significa que os conjuntos de dados são significativamente diferentes (DEEPCHECKS, 2023).

3.3.6 Métricas usadas em Classificação

- Acurácia ou *Accuracy* é a porcentagem de previsões corretas em relação ao total de previsões. Essa métrica é fácil de entender, mas pode ser enganosa se houver um desequilíbrio entre as classes, ou seja, se uma classe tiver muito mais observações do que as outras.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.21)$$

Em que:

- TP: Verdadeiros positivos (*True Positives*)
- TN: Verdadeiros negativos (*True Negatives*)
- FP: Falsos positivos (*False Positives*)

- FN: Falsos negativos (*False Negatives*)
- Precisão ou *Precision* é a proporção de observações positivas corretamente classificadas em relação a todas as observações classificadas como positivas. A precisão é uma métrica importante quando o foco é minimizar os falsos positivos, ou seja, quando é desejável ter certeza de que uma observação classificada como positiva realmente é positiva.

$$Precision = \frac{TP}{TP + FP} \quad (3.22)$$

Em que:

- TP: Verdadeiros positivos (*True Positives*)
- FP: Falsos positivos (*False Positives*)
- *Recall* ou Sensibilidade (*Sensitivity*) é a proporção de observações positivas corretamente classificadas em relação a todas as observações que são realmente positivas. O *Recall* é uma métrica importante quando o foco é minimizar os falsos negativos, ou seja, quando é desejável ter certeza de que todas as observações positivas são classificadas como tal.

$$Recall = \frac{TP}{TP + FN} \quad (3.23)$$

Em que:

- TP: Verdadeiros positivos (*True Positives*)
- FN: Falsos negativos (*False Positives*)
- F1 Score é a média harmônica da precisão e *Recall*. Essa métrica é útil quando o equilíbrio entre precisão e *Recall* é importante.

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.24)$$

Em que:

- *Precision*: Precisão
- *Recall*: Sensibilidade
- Área sob a curva ROC *Area under the ROC curve - AUC* é uma medida da habilidade do modelo de distinguir entre classes. A curva ROC é um gráfico que representa a taxa de verdadeiros positivos em relação à taxa de falsos positivos para diferentes pontos de corte. A AUC é a área sob essa curva e varia de 0 a 1. Um modelo com AUC igual a 1 é considerado perfeito. O valor de AUC é obtido pela integral da curva ROC em relação ao eixo dos FPR, ou seja, a área abaixo da curva ROC 3.25. Quanto mais próxima de 1 for a AUC, melhor é o desempenho do modelo de

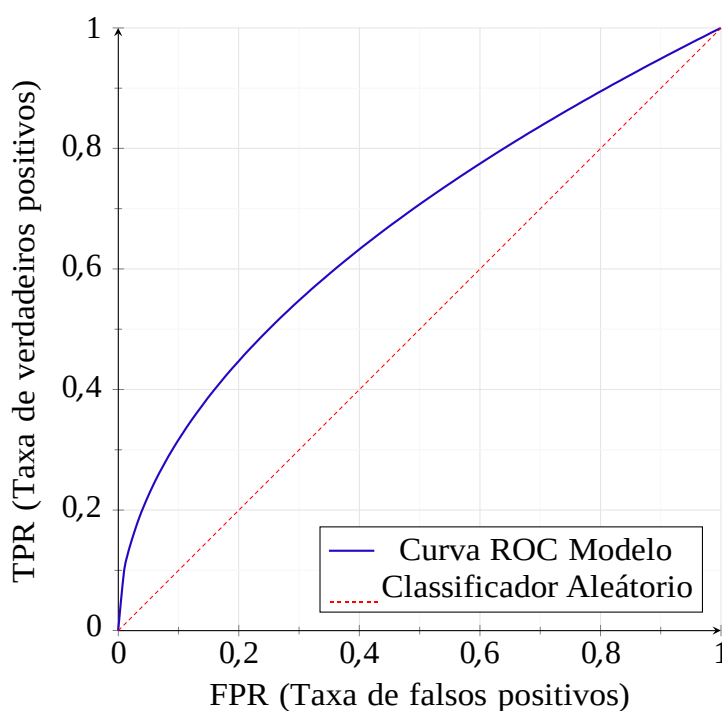
classificação, pois indica que ele é capaz de distinguir com precisão os exemplos positivos dos negativos. Quando a AUC é igual a 0,5, o modelo é equivalente a um classificador aleatório, que não é capaz de discriminar entre exemplos positivos e negativos. E quando a AUC é menor que 0,5, o modelo está piorando a classificação, sendo necessário considerar a inversão da classe positiva para aumentar a acurácia.

$$AUC = \int_0^1 TPR(FPR)d(FPR) \quad (3.25)$$

Em que:

- TPR: Taxa de verdadeiros positivos (*True Positive Rate*)
- FPR: Taxa de falsos positivos (*False Positive Rate*)

Figura 19 – Exemplo Gráfico de uma curva ROC.



Fonte: O autor.

Na Figura 19 a linha azul representa a curva ROC e a linha vermelha tracejada representa os valores de um classificador Aleatório. No Eixo horizontal estão os valores para FPR e no eixo vertical os valores para TPR. Utilizando a Equação 3.25 é encontrado o valor para $AUC = 0,87$.

(HASTIE ROBERT TIBSHIRANI, 2009; POOLA; PL; Y, 2023; ALKAISSY et al., 2023)

3.3.7 Exemplos de aplicação das Métricas Univariadas e Multivariadas

3.3.7.1 Métricas Univariadas e seus Casos de Uso

1. **Média:** Uma escola quer calcular a média geral das notas dos alunos em matemática.
2. **Mediana:** Durante a venda de casas em um bairro, a mediana do preço de venda determina o preço típico.
3. **Moda:** Uma loja de roupas analisa a cor de camiseta mais popular vendida.
4. **Variância:** Um fundo de investimento deseja entender a variabilidade dos retornos mensais.
5. **Desvio padrão:** Uma empresa de transporte mede a consistência do tempo de entrega ao longo de um mês.
6. **Assimetria (Skewness):** Um economista analisa a distribuição de renda de um país.
7. **Curtose (Kurtosis):** Um analista financeiro estuda os retornos diários de uma ação.

(TRIOLA, 2017; STEFANIA et al., 2023; CAO et al., 2023)

3.3.7.2 Métricas Multivariadas e seus Casos de Uso

1. **Covariância:** Um analista verifica a relação entre o número de horas estudadas e as notas obtidas pelos alunos.
2. **Correlação:** Uma empresa farmacêutica investiga a relação entre a dose de um medicamento e a eficácia do tratamento.
3. **Regressão Linear:** Um varejista online prevê as vendas futuras com base no número de visitantes do site.
4. **Análise de Componentes Principais (PCA):** Utilizado em reconhecimento facial para reduzir a dimensionalidade das imagens.
5. **Análise Discriminante:** Um banco classifica os candidatos a empréstimos como alto risco ou baixo risco.

(TRIOLA, 2017; STEFANIA et al., 2023; CAO et al., 2023)

3.3.8 Exemplos de Métricas de Classificação e seus Casos de Uso.

1. **Acurácia (*Accuracy*):**

- Um software de reconhecimento facial identifica funcionários em uma empresa, medindo a porcentagem total de identificações corretas.

2. **Precisão (*Precision*):**

- Sistema de detecção de *spam* para *e-mails* onde é vital garantir que *e-mails* legítimos não sejam erroneamente marcados como *spam*.

3. **Recall ou Sensibilidade (*Sensitivity*):**

- Teste médico para detectar uma doença rara focado em minimizar os falsos negativos.

4. **Especificidade (*Specificity*):**

- Testes de drogas que visam corretamente identificar indivíduos não usuários como negativos.

5. **F1-Score:**

- Sistema de moderação de conteúdo para mídia social que equilibra a remoção de conteúdo ofensivo e a preservação de conteúdo legítimo.

6. **Curva ROC e Área sob a Curva (AUC):**

- Modelos de risco de crédito usados por bancos para distinguir entre bons e maus tomadores de crédito.

7. **Log Loss:**

- Competições de modelagem preditiva que avaliam não apenas a classificação correta, mas também a confiança associada.

8. **Taxa de Erro (*Error Rate*):**

- Sistema de reconhecimento de voz que mede a porcentagem de palavras ou comandos erroneamente reconhecidos.

9. **Matriz de Confusão *Confusion Matrix*:**

- Campanhas de *marketing* que visualizam a eficácia de um modelo em prever respostas a uma oferta.

(HASTIE ROBERT TIBSHIRANI, 2009; POOLA; PL; Y, 2023; ALKAISSY et al., 2023)

3.4 O modelo de Classificação

Neste estudo, são empregadas técnicas avançadas de aprendizado de máquina para classificar imagens, com foco em identificar anomalias. Um modelo de *Machine Learning* é implementado utilizando a biblioteca *PyTorch*, conhecida por sua eficiência e flexibilidade em tarefas de *Deep Learning*.

O conjunto de imagens é categorizado em duas classes: 1 para as consideradas normais e 0 para aquelas que apresentam alguma anormalidade. Estabelecer uma divisão clara e rotulada dessa forma é crucial para assegurar um treinamento eficaz do modelo e previsões confiáveis em novos dados.

Uma característica notável do projeto é a forma como o conjunto de dados é gerenciado e acessado. As imagens rotuladas estão armazenadas no *Google Drive*, uma plataforma de armazenamento em nuvem reconhecida. Através da integração do *Google Drive* com o *Google Colab*, é possível realizar uma ingestão direta das imagens no *notebook* do *Colab*, simplificando o processo e evitando a necessidade de *uploads* manuais.

No código que implementa o modelo, a classe *ImageFolder* do *PyTorch* é utilizada. Esta classe fornece os métodos necessários para direcionar o diretório das imagens e para segmentar o conjunto de dados entre as categorias 'normal' e 'anormal', garantindo o treinamento e teste adequados do modelo.

Visando otimizar o desempenho, optou-se pela utilização de um modelo pré-treinado chamado *densenet*. Ao adotar pesos pré-treinados, reduz-se o tempo de treinamento e incorporam-se *insights* de tarefas similares, enriquecendo a capacidade do modelo de identificar características pertinentes nas imagens.

Após a execução e classificação pelo modelo, um arquivo no formato *HyperText Markup Language* (HTML) é gerado, apresentando os resultados de maneira estruturada. Este arquivo pode ser visualizado localmente ou armazenado no *Google Drive*, facilitando o compartilhamento e acesso dos resultados em diferentes localidades.

3.5 Testes selecionados.

No contexto de aprendizado de máquina, a ocorrência de *drift* refere-se a mudanças no padrão dos dados ao longo do tempo, o que pode afetar adversamente o desempenho do modelo. É crucial detectar e lidar com *drift* para manter a confiabilidade do modelo. Assim, foram selecionados os seguintes testes para identificar e evidenciar essa ocorrência:

1. **Train Test Label Drift:** A medição univariada é aplicada, centrando-se na propriedade do rótulo (*Label*). No contexto da classificação de imagens, a propriedade mais relevante é o número de imagens por classe. Esta seleção permite uma análise

baseada na distribuição discreta das imagens. Dada essa natureza discreta, a métrica de Cramer, conforme delineada na Equação 3.17, foi escolhida por sua capacidade de detectar associações entre variáveis categóricas.

2. **Image Property Drift:** Esta análise também faz uso da abordagem univariada, mas se concentra nas propriedades intrínsecas das imagens. As características examinadas incluem Contraste RMS, Brilho, Intensidade relativa do azul médio, Intensidade relativa do verde médio, Intensidade relativa do vermelho médio, Área e *Aspect Ratio*. Dado que estas propriedades são contínuas, a métrica de Wasserstein ou *Earth Movers Distance*, conforme detalhada na Equação 3.18, foi adotada. Esta métrica é particularmente adequada para comparar distribuições de dados contínuos.
3. **Train Test Prediction Drift:** Esta análise se concentra nas amostras por classe e no volume de dados analisados. A métrica de Cramer, devido à sua eficácia na detecção de diferenças entre categorias, foi escolhida para calcular o *Drift Score*.
4. **Image Dataset Drift:** Ao contrário dos testes anteriores, esta análise se vale de uma abordagem multivariada, considerando o domínio global em que o classificador opera. Essa métrica é fundamental para avaliar a estabilidade e confiabilidade do modelo, especialmente quando confrontado com variações nos dados de teste ou em cenários de produção.

Em conjunto, esses testes fornecem uma visão abrangente das potenciais fontes de *drift*, permitindo a adoção de medidas corretivas adequadas e assegurando a eficácia contínua do modelo.

3.6 Dataset selecionados.

O MURA é um dos *datasets* de imagem médica de grande escala mais reconhecidos no campo da radiologia. Ele é composto por radiografias anotadas de várias regiões do corpo humano, fornecendo um terreno fértil para pesquisas e aplicações de aprendizado de máquina voltadas para a área médica.

Dentro deste conjunto, a subseção *elbow* (cotovelo) merece destaque. Esta subseção é exclusivamente focada em radiografias da região do cotovelo, proporcionando uma análise detalhada e concentrada dessa área específica. Radiografias do cotovelo são fundamentais para diagnosticar diversas condições médicas, desde fraturas traumáticas até degenerações crônicas, como osteoartrite.

A escolha do *dataset elbow* do MURA para o estudo presente foi motivada por várias razões:

1. **Especificidade:** Ao concentrar-se apenas na região do cotovelo, é possível obter um entendimento mais profundo e focado das variações e patologias comuns a esta região.
2. **Variedade de Casos:** O *dataset* apresenta uma ampla gama de patologias, tornando-o ideal para treinar modelos robustos capazes de reconhecer diversas anomalias.
3. **Relevância Clínica:** Dada a prevalência de lesões no cotovelo, especialmente em determinadas populações (como atletas), existe uma demanda significativa por ferramentas automatizadas de diagnóstico nessa área.
4. **Desafio Técnico:** A região do cotovelo é intrincada, com muitos ossos e articulações próximos uns dos outros. Isso representa um desafio adicional para algoritmos de aprendizado de máquina, tornando a pesquisa nessa área tanto desafiadora quanto gratificante.

Em resumo, o *dataset elbow* do MURA foi selecionado para este estudo devido à sua relevância clínica, variedade de casos, especificidade, e o desafio técnico que apresenta. Acredita-se que as descobertas obtidas a partir deste *dataset* podem contribuir significativamente para avanços na área de diagnóstico por imagem, com implicações diretas no cuidado ao paciente.

3.6.1 Separando o *Dataset Elbow* MURA em treino e teste.

A divisão de conjuntos de dados em porções específicas para treino e teste é uma prática consagrada em aprendizado de máquina e visão computacional. A escolha da proporção de 90% para treino e 10% para teste é baseada nos seguintes critérios:

- **Maximização do Conjunto de Treino:** Garantir um volume substancial de dados para treino é essencial para aprimorar a capacidade do modelo de aprender características intrínsecas dos dados. Uma alocação de 90% dos dados para treino assegura que o modelo tenha acesso a uma ampla variedade de exemplos, promovendo uma aprendizagem mais robusta e generalizada.
- **Validação Confidencial:** Embora seja crucial ter um conjunto de treino vasto, a reserva de uma porção para teste é imperativa para avaliar o desempenho do modelo em dados não vistos anteriormente. Os 10% reservados para teste fornecem uma amostra representativa, permitindo uma avaliação justa e imparcial do modelo.
- **Equilíbrio entre *Overfitting* e *Underfitting*:** A divisão na proporção de 90%-10% busca atingir um equilíbrio onde o modelo é exposto a uma quantidade significativa de dados (reduzindo a chance de *underfitting*), mas ainda reserva uma porção suficiente para teste (prevenindo o risco de *overfitting*).

- **Consistência com Práticas Padrão:** Em muitos cenários, especialmente quando o conjunto de dados é grande, uma divisão de 90%-10% é comum e aceita pela comunidade, tornando os resultados mais comparáveis com outras pesquisas e aplicações.
- **Otimização de Recursos Computacionais:** Ao se utilizar uma proporção maior para treinamento, maximiza-se o uso de recursos computacionais durante essa fase, que frequentemente é mais exigente em termos de cálculos e processamento.

Consequentemente, a proporção escolhida de 90% para treino e 10% para teste busca garantir que o modelo seja treinado de forma eficaz, ao mesmo tempo em que sua performance é rigorosamente avaliada em um conjunto de dados separado, garantindo uma estimativa fidedigna de seu desempenho em cenários reais.

3.7 Densenet selecionada.

O *DenseNet121*, que pertence à família das redes *DenseNet* (Redes Densamente Conectadas), é uma arquitetura de rede neural convolucional projetada para otimizar a eficiência na formação de características. Ao contrário de outras arquiteturas convencionais, onde as camadas estão ligadas sequencialmente, as *DenseNets* possuem conexões entre cada par de camadas. Esta característica fundamental melhora o gradiente no processo de treinamento, reforça a propagação de características e reduz a necessidade de neurônios, tornando a rede mais compacta e eficiente.

A escolha do *DenseNet121* para o estudo atual foi baseada nas seguintes razões:

1. **Eficiência na Extração de Características:** A conectividade densa entre camadas assegura que o modelo não apenas aprenda características novas, mas também reutilize as já existentes, otimizando o processo de aprendizado.
2. **Redução de Desvanecimento do Gradiente:** A arquitetura densamente conectada auxilia na propagação do gradiente, tornando o treinamento mais estável e acelerando a convergência.
3. **Menor Necessidade de Parâmetros:** A *DenseNet121*, apesar de sua conectividade, requer menos parâmetros em comparação com outras arquiteturas de profundidade semelhante, tornando-a mais eficiente em termos de memória e computacionalmente.
4. **Resultados de Referência:** Em várias tarefas de visão computacional, especialmente na classificação de imagens, as *DenseNets* demonstraram desempenho superior ou comparável às arquiteturas estado-da-arte.

Em resumo, a escolha do *DenseNet121* foi motivada pela sua capacidade demonstrada de fornecer representações hierárquicas ricas em conjuntos de dados complexos, sua eficiência em termos de parâmetros, e seus resultados sólidos em tarefas de classificação de imagens. Tais características tornam o *DenseNet121* um modelo apropriado e promissor para muitos desafios no domínio do aprendizado profundo.

3.7.1 Características Técnicas DenseNet121.

A DenseNet121 é uma variante da arquitetura DenseNet, notória por sua capacidade de conectar cada camada diretamente a todas as camadas subsequentes em um padrão denso. Este padrão de conexões densas é fundamental para garantir a reutilização de características e a eficiência na propagação de gradientes durante o treinamento.

- **Conexões Densas:** Em contraste com arquiteturas tradicionais onde camadas subsequentes recebem entradas apenas da camada anterior, na DenseNet, camadas são diretamente conectadas a todas as camadas seguintes. Isto resulta em uma conservação de características e garante um fluxo de informação mais consistente.
- **Crescimento de Canais:** A DenseNet121 possui um fator de crescimento k , que define o número de canais produzidos por cada camada. Este parâmetro é essencial para controlar a expansão das características em camadas subsequentes.
- **Blocos Densos e Camadas de Transição:** A arquitetura é organizada em blocos densos intercalados por camadas de transição. Enquanto os blocos densos ajudam na propagação de características e gradientes, as camadas de transição reduzem a dimensionalidade dos dados, usando geralmente uma combinação de operações como convolução e *pooling*.
- **Número de Parâmetros:** A DenseNet121 é projetada para ser eficiente em termos de parâmetros. Graças à sua natureza densa e à reutilização de características, ela pode obter uma performance competitiva com um número menor de parâmetros em comparação a outras arquiteturas profundas.
- **Aplicabilidade em Diversas Tarefas:** Embora a DenseNet121 seja frequentemente associada à classificação de imagens, sua arquitetura versátil permite que seja adaptada para uma variedade de tarefas, incluindo segmentação e detecção de objetos.

Em resumo, a DenseNet121 combina eficiência e performance, sendo uma escolha relevante para uma série de aplicações no campo de visão computacional.

3.8 Escolha do *Deepchecks*

Em análises de visão computacional, é essencial ter mecanismos rigorosos para validar e interpretar os resultados. Especialmente ao avaliar *drifts* em conjuntos de dados, a precisão, clareza e interpretabilidade dos resultados tornam-se fundamentais para compreender o comportamento dos modelos e garantir sua robustez e confiabilidade.

O *Deepchecks* surge como um framework contemporâneo focado na validação e interpretação de modelos de aprendizado profundo. Sua capacidade de apresentar uma visão detalhada e abrangente dos resultados faz dele uma ferramenta valiosa, especialmente em cenários de visão computacional.

A escolha do *Deepchecks* foi baseada nas seguintes considerações:

1. **Visão Holística dos Modelos:** *Deepchecks* oferece uma visão abrangente do modelo, considerando não apenas métricas de desempenho, mas também características intrínsecas do modelo e do conjunto de dados.
2. **Especialização em *Drifts*:** Este framework possui mecanismos robustos para detectar e interpretar *drifts*, ajudando os pesquisadores a compreender as nuances das mudanças nos dados ao longo do tempo.
3. **Visualizações Claras:** As visualizações fornecidas por *Deepchecks* são projetadas para serem intuitivas, permitindo uma interpretação mais direta e transparente dos resultados.
4. **Flexibilidade e Integração:** Com uma interface amigável e capacidade de se integrar a diversos outros frameworks e ferramentas, *Deepchecks* oferece uma experiência fluida e adaptável.

Concluindo, em um domínio tão desafiador e dinâmico como a visão computacional, ferramentas que proporcionam insights claros e detalhados sobre o comportamento dos modelos são essenciais. *Deepchecks*, com suas características únicas e foco em interpretabilidade, emerge como uma escolha acertada para essa análise, garantindo que os resultados dos *drifts* sejam compreendidos de maneira abrangente e rigorosa.

3.9 Simulando degradação dos *dataset*

Em processos de análise e modelagem de dados, surgem desafios frequentes relacionados à avaliação da robustez e eficácia dos métodos em cenários subótimos. *Datasets* do mundo real, frequentemente, apresentam imperfeições. Estas imperfeições podem ser originadas por erros de coleta, falhas em equipamentos ou corrupções intencionais. A

simulação dessas degradações torna-se essencial para entender a resposta dos métodos em contextos adversos e desenvolver soluções mais robustas.

Esta seção destina-se a apresentar as principais falhas técnicas de máquinas de raio-X bem como a escolha de um modelo de degradação que possa representar em sua maioria as falhas, as respectivas características e potenciais impactos na qualidade e confiabilidade dos dados.

3.9.1 Falhas em máquinas de raio-X.

1. **Problemas de Calibração:** Se a máquina não estiver corretamente calibrada, pode não produzir a quantidade adequada de radiação.
 - **Consequência:** Imagens muito claras ou escuras, dificultando a visualização de detalhes.
2. **Defeitos no Tubo de Raios-X:** O tubo de raios-X pode sofrer desgaste ao longo do tempo ou ter defeitos de fabricação.
 - **Consequência:** Pode resultar em pontos ou áreas de sombra na imagem, interferindo na interpretação.
3. **Falhas no Sistema de Detecção:** Se os detectores que capturam a radiação não estiverem funcionando corretamente, podem não capturar a imagem com precisão.
 - **Consequência:** Imagens com ruído, manchas ou distorções.
4. **Problemas de Movimentação do Paciente:** Se o paciente se mover durante o exame, pode causar uma imagem borrada.
 - **Consequência:** Redução da clareza da imagem, tornando difícil ou impossível a interpretação.
5. **Problemas com o Software de Processamento:** O software que processa e exibe as imagens pode ter falhas ou configurações incorretas.
 - **Consequência:** Imagens distorcidas, artefatos ou alteração nas escalas de cinza.
6. **Mau Alinhamento ou Posicionamento Incorreto:** Se a máquina ou o paciente não estiverem alinhados corretamente, a imagem pode não capturar a área de interesse.
 - **Consequência:** Áreas de interesse podem ser cortadas ou vistas sob um ângulo não ideal.

7. **Falhas de Manutenção:** A falta de manutenção regular pode resultar em várias falhas, desde defeitos elétricos a problemas mecânicos.

- **Consequência:** Risco de exposição excessiva à radiação, imagens de baixa qualidade ou falha total do equipamento.

8. **Contaminação ou Sujeira:** Sujeira ou contaminação nos componentes que entram em contato direto com os raios-X (como janelas ou placas) podem afetar a imagem.

- **Consequência:** Manchas, sombras ou artefatos na imagem.

3.9.2 *Salt and Pepper.*

A escolha do ruído do tipo salt and pepper para simular falhas em imagens de raios-X é justificável por várias razões:

1. **Natureza Esparsa das Falhas:** Muitas das falhas descritas no texto, como defeitos no tubo de raios-X e contaminação ou sujeira, resultam em perturbações pontuais e esparsas na imagem. O ruído *salt and pepper* se caracteriza por alterações brancas e pretas esparsas, que se assemelham a estas perturbações pontuais.
2. **Representação de Extremos:** O ruído *salt and pepper* introduz valores extremos (geralmente preto e branco) na imagem. Estes valores extremos podem simular situações como áreas superexpostas ou subexpostas na imagem de raios-X, refletindo problemas como calibração inadequada.
3. **Visibilidade Clara:** Ao contrário de outras formas de ruído, como o ruído gaussiano, o ruído *salt and pepper* é claramente visível e identificável. Isso facilita a compreensão de onde o erro está localizado na imagem, tornando mais intuitiva a identificação e eventual correção.
4. **Simplicidade e Reprodutibilidade:** Para fins experimentais, é vital que o ruído introduzido seja fácil de simular e reproduzir em diferentes conjuntos de dados. O ruído *salt and pepper* pode ser facilmente implementado e repetido em diferentes imagens, garantindo consistência nos experimentos.
5. **Desafio em Correção:** Em aplicações práticas, especialmente no contexto médico, é crucial que as técnicas de processamento de imagens sejam capazes de lidar com ruídos desafiadores. O ruído *salt and pepper*, devido à sua natureza abrupta e extrema, apresenta um desafio significativo para a correção, tornando-o ideal para testar a robustez dos algoritmos.

Em resumo, a escolha do ruído do tipo *salt and pepper* para representar falhas nas imagens de raios-X é coerente tanto com a natureza das falhas mencionadas no texto quanto com os requisitos experimentais. Ele fornece um modelo representativo e desafiador de erros reais, enquanto mantém a simplicidade e a reproducibilidade essenciais para o ambiente experimental.

3.9.2.1 Escolha do nível de ruído *Salt and Pepper*.

Ao investigar degradações em conjuntos de dados, particularmente através da introdução de ruído do tipo *salt and pepper*, a escolha de taxas de ruído apropriadas é de suma importância. As taxas de ruído selecionadas, isto é, 0.005, 0.03, 0.05 e 0.07, foram determinadas com base em vários critérios técnicos e de relevância prática.

- **Representatividade de Diferentes Intensidades:** A inclusão de taxas de ruído tanto mais baixas (como 0.005) quanto mais altas (como 0.07) permite uma avaliação ampla das degradações. Esta variedade serve para simular situações de pouca interferência até contextos onde o ruído é mais pronunciado.
- **Relevância Prática:** As taxas escolhidas são muitas vezes encontradas em literaturas e estudos anteriores, tornando-as pertinentes para fins de comparação e *benchmarking*. A seleção destes valores específicos pode ajudar na correlação de resultados com pesquisas existentes e no entendimento de como o ruído *salt and pepper* em diferentes taxas impacta os modelos.
- **Observações Experimentais:** Em experimentações preliminares, taxas de ruído extremamente baixas ou altas demais podem não apresentar diferenças significativas ou podem ser excessivamente degradantes, respectivamente. Os valores selecionados oferecem um equilíbrio, proporcionando uma gama de cenários que são tanto desafiadores quanto representativos de situações reais.
- **Consistência na Variação:** A seleção de valores em um intervalo progressivo permite uma análise sistemática da resposta dos modelos à introdução gradativa do ruído *salt and pepper*.

Em resumo, as taxas de ruído escolhidas para *salt and pepper* proporcionam um espectro representativo de degradações, facilitando a avaliação robusta e abrangente dos impactos deste tipo de ruído nos conjuntos de dados.

3.10 Experimento

A fim de avaliar e validar a degradação potencial dos modelos em circunstâncias controladas, conduziu-se um experimento criteriosamente estruturado. As etapas desse experimento são delineadas a seguir:

1. O conjunto de dados foi segmentado em grupos de treino e teste. O grupo de treino consistia em 1.993 imagens da classe *Abnormal* e 1.742 imagens da classe *Normal*. Para o conjunto de teste, foram alocadas 228 imagens da classe *Abnormal* e 164 da classe *Normal*.
2. A escolha recaiu sobre a arquitetura *Densenet 121*, um modelo consolidado que dispõe de pesos pré-treinados, assegurando assim um ponto de partida robusto para o treinamento.
3. Utilizando a plataforma *Pytorch* no ambiente *Google Colab*, o modelo foi implementado e treinado. Inicialmente, procedeu-se ao treinamento e validação sem introdução de qualquer forma de ruído nos dados.
4. Como etapa subsequente, e visando simular potenciais degradações nos dados, foram inseridos ruídos do tipo *salt and pepper* nos dados de teste. Estes ruídos foram aplicados em quatro níveis distintos: 0.005, 0.03, 0.05 e 0.07. Em cada uma dessas intensidades, o modelo foi submetido a nova execução, com os resultados devidamente catalogados para análises futuras.

Com esta abordagem meticulosa, busca-se compreender em profundidade o comportamento do modelo diante de variações e perturbações nos dados, um passo crucial para assegurar sua confiabilidade em cenários reais e variáveis.

3.11 Considerações Finais

Com base nas etapas descritas neste capítulo, é possível obter um panorama claro dos materiais utilizados, das ferramentas adotadas e dos métodos empregados no desenvolvimento deste projeto de pesquisa. Essas informações são fundamentais para a compreensão e interpretação dos resultados apresentados no capítulo subsequente, que aborda os resultados e a análise dos experimentos realizados.

4 Resultados e Discussões

Neste capítulo, serão apresentados os resultados obtidos a partir da análise dos dados coletados durante o experimento, bem como serão discutidas as implicações desses resultados para o campo de estudo em questão.

Inicialmente, serão apresentados os resultados de forma objetiva, utilizando tabelas, gráficos e outras formas de visualização de dados para facilitar a compreensão. Em seguida, serão realizadas análises e comentários sobre os resultados encontrados.

Além disso, será feita uma discussão mais ampla dos resultados, relacionando-os com a literatura existente sobre o assunto e buscando compreender suas implicações teóricas e práticas. Serão levantadas questões importantes a serem consideradas, como limitações da pesquisa, possíveis explicações para os resultados encontrados e sugestões para pesquisas futuras.

4.1 Definição dos limites de *drift* no *deepchecks*.

Os limites padrão de *drift* no *deepchecks* estão definidos conforme descrito a seguir:

- *Train Test Label Drift*:
 - A pontuação de *drift* categórico é inferior a 0,15;
 - A pontuação de *drift* numérico é inferior a 0,075.
- *Image Property Drift*:
 - O *Earth Mover's Distance* deve ser inferior a 0,1 para *drift* de propriedades de imagem.
- *Train Test Prediction Drift*:
 - A pontuação de *drift* categórico é inferior a 0,15 para *drift* de previsão;
 - A pontuação de *drift* numérico é inferior a 0,075 para *drift* de previsão.
- *Image Dataset Drift*:
 - O *Earth Mover's Distance* deve ser inferior a 0,1 para *drift* de propriedades de imagem.

Estes são os valores predefinidos e fornecem uma linha de base para avaliar possíveis desvios nos conjuntos de dados ou nas propriedades de imagem.

4.2 Resultados Gerais

Esta seção tem o propósito de apresentar os resultados gerais obtidos a partir do experimento realizado para avaliar o impacto do *drift* no desempenho do classificador de imagens. Serão discutidas as métricas de desempenho, as variações observadas e as potenciais implicações desses resultados.

Experimentos					
Ruído	0,00%	0,05%	0,30%	0,50%	0,70%
Label <i>drift</i> Amostra por Classe Treino e Teste					
<i>Drift Score</i> Cramer's V	2,2573%	8,0675%	8,0675%	8,0675%	8,0675%
Image Property <i>drift</i> RMS Contraste					
<i>Drift Score</i> Wasserstein Metric	4,3483%	4,0404%	8,2770%	11,3515%	14,2890%
Image Property <i>drift</i> Mean Blue Relative Intensity					
<i>Drift Score</i> Wasserstein Metric	2,0311%	1,6945%	1,8216%	1,9074%	1,9922%
Image Property <i>drift</i> Mean Green Relative Intensity					
<i>Drift Score</i> Wasserstein Metric	2,0272%	1,6896%	1,8161%	1,9013%	1,9844%
Image Property <i>drift</i> Brightness					
<i>Drift Score</i> Wasserstein Metric	1,9441%	3,2012%	4,5018%	5,5865%	6,6803%
Image Property <i>drift</i> Area					
<i>Drift Score</i> Wasserstein Metric	0,0000%	0,0000%	0,0000%	0,0000%	0,0000%
Image Property <i>drift</i> Aspect Ratio					
<i>Drift Score</i> Wasserstein Metric	0,0000%	0,0000%	0,0000%	0,0000%	0,0000%
Predição de <i>drift</i> Treino e Teste					
<i>Drift Score</i> Cramer's V	0,0000%	20,3998%	51,9072%	68,9568%	71,9248%
Image Dataset <i>drift</i>					
<i>Drift Score</i>	6,8802%	19,8075%	43,7801%	52,3139%	63,7912%
Multivariable Total					

Tabela 2 – Sumário dos dados dos experimentos.

A Tabela 2 fornece um sumário detalhado dos resultados de uma série de experimentos relacionados ao *drift*. O foco principal desses experimentos é quantificar o *drift* em várias propriedades de imagens e nas previsões de treino e teste.

Os resultados apresentados na Tabela 2 indicam que o nível de ruído introduzido tem um impacto significativo no *drift* observado nas propriedades das imagens e nas previsões. Especificamente, à medida que o nível de ruído aumenta, a maioria dos *drift*

scores também aumenta. É importante destacar que algumas propriedades das imagens, como a área e o aspecto, não apresentaram *drift* independentemente do nível de ruído.

Percentual de Ruído	KS Statistic	P-value
0.07	0.6204	2.04×10^{-44}
0.05	0.5714	2.82×10^{-37}
0.03	0.5714	2.82×10^{-37}
0.01	0.5918	3.72×10^{-40}
0.005	0.5755	7.68×10^{-38}

Tabela 3 – Valores de KS e *p-value* para percentual de ruído.

A Tabela 3 apresenta os resultados do KS e *p-value*. A estatística do teste de KS é fundamental para avaliar a aderência de uma amostra a uma distribuição teórica ou para comparar as distribuições de duas amostras independentes. O valor de KS varia de 0 a 1 e é interpretado da seguinte forma:

- **Valor 0:** Indica que as funções de distribuição acumulada (CDFs) comparadas são idênticas em todos os pontos. No contexto do teste de uma amostra, isso significa que a amostra segue perfeitamente a distribuição teórica de referência. No teste de duas amostras, implica que as duas amostras provêm da mesma distribuição.
- **Valor 1:** Representa a máxima discrepância possível entre as CDFs, indicando que as distribuições são completamente diferentes em pelo menos um ponto.

A estatística KS ou D é utilizada para testar hipóteses sobre a semelhança entre as distribuições. O procedimento é o seguinte:

1. Se D for pequeno e não ultrapassar um valor crítico específico (dependente do tamanho da amostra e do nível de significância α escolhido), não rejeitamos a hipótese nula (H_0), concluindo que não há evidência suficiente para afirmar que as distribuições são diferentes.
2. Se D for maior que o valor crítico, rejeitamos a hipótese nula, indicando que as distribuições comparadas são significativamente diferentes.

No âmbito dos testes estatísticos, incluindo o teste de Kolmogorov-Smirnov (K-S), o valor-p é definido como a probabilidade de observar um resultado tão extremo quanto, ou mais extremo que, o resultado atual, sob a premissa de que a hipótese nula seja verdadeira. Este documento visa elucidar a interpretação do valor-p no teste K-S, bem como detalhar seu cálculo e utilização.

- Interpretação do Valor-p

- **Baixo Valor-p:** Um valor-p inferior ao nível de significância α estabelecido, como 0.05 ou 0.01, sugere uma diferença estatisticamente significativa entre as distribuições comparadas. Neste contexto, a hipótese nula é rejeitada, indicando que as amostras não provêm da mesma distribuição, ou que a amostra não segue a distribuição teórica especificada.
- **Alto Valor-p:** Um valor-p elevado indica a ausência de evidência estatística suficiente para rejeitar a hipótese nula, sugerindo que as diferenças observadas entre as distribuições podem ser atribuídas ao acaso.

- Cálculo do Valor-p

O cálculo do valor-p no teste K-S baseia-se na estatística do teste D , que reflete a máxima diferença absoluta entre as funções de distribuição acumulada (CDFs) comparadas. A determinação do valor-p a partir da estatística D observada envolve o uso de tabelas ou algoritmos computacionais, considerando o tamanho da amostra ou os tamanhos das amostras no caso de dois grupos. A distribuição exata de D sob a hipótese nula é afetada pelo tamanho da amostra.

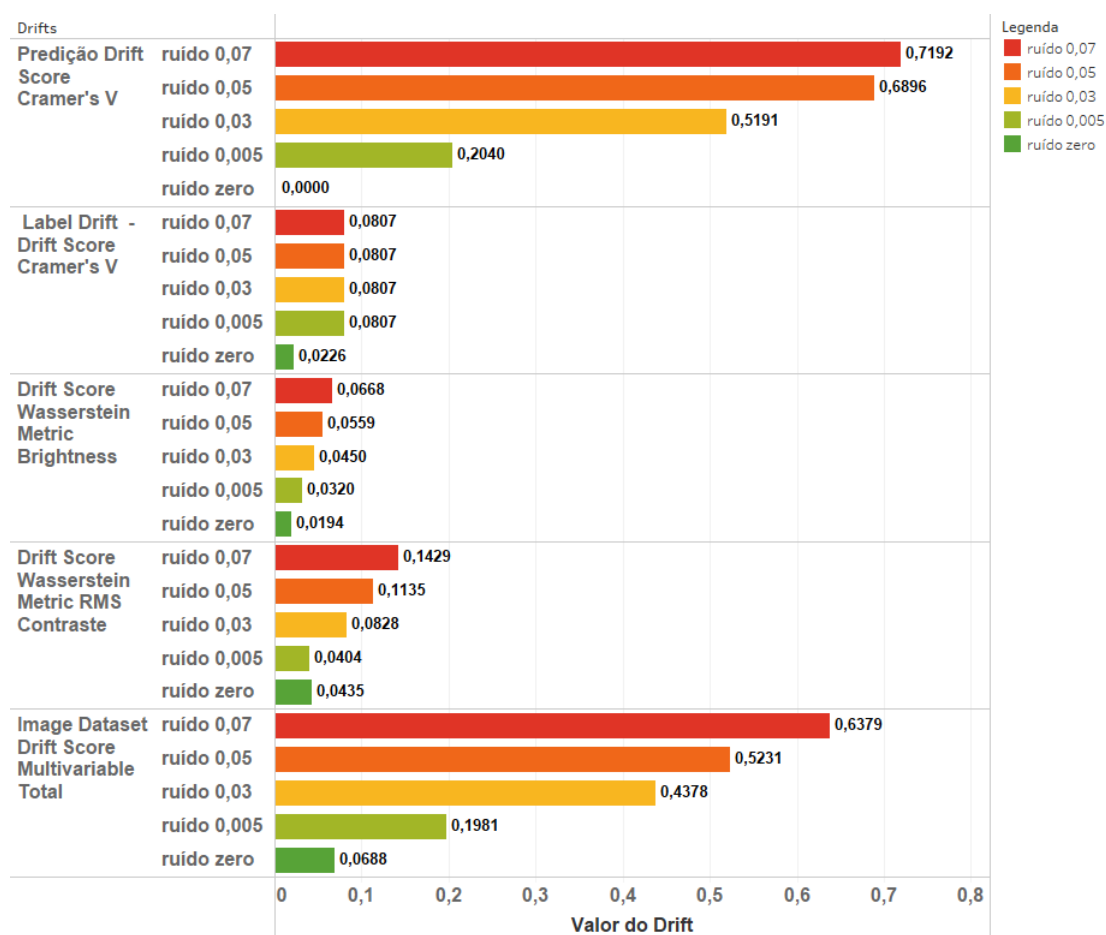
- Utilização do Valor-p

A aplicação do valor-p no teste K-S implica a definição a priori de um nível de significância α , que serve como critério para a rejeição da hipótese nula. Se o valor-p é inferior a α , conclui-se pela rejeição da hipótese nula. Importa salientar que um valor-p reduzido não implica na magnitude da discrepância entre as distribuições, mas apenas na sua significância estatística. Analogamente, um valor-p elevado não comprova a identidade entre as distribuições, mas indica a insuficiência de evidências para afirmar a existência de uma diferença significativa ao nível de significância selecionado.

Analizando os dados encontrados para um percentual de ruído 0,07 por exemplo temos o seguinte: Estatística KS A estatística KS obtida, 0.6204081632653061, indica uma diferença substancial entre as distribuições comparadas. Considerando que a estatística KS varia de 0 a 1, um valor próximo a 1, como o observado, sinaliza uma discrepância significativa entre as funções de distribuição acumulada (CDFs) das distribuições em análise. Valor-p O valor-p calculado, $2.0370909332351388 \times 10^{-44}$, é extremamente baixo, sugerindo que a probabilidade de observar uma diferença tão expressiva ou maior, sob a hipótese nula de que as amostras provêm da mesma distribuição ou que a amostra segue a distribuição teórica especificada, é praticamente nula. Tal resultado fornece uma forte evidência para a rejeição da hipótese nula. Interpretação dos Resultados A combinação de uma estatística KS elevada com um valor-p extremamente baixo constitui uma evidência robusta contra a hipótese nula.

No contexto de um teste de uma amostra, tal resultado implicaria que a amostra analisada não segue a distribuição teórica predefinida. Estes resultados apontam para uma diferença estatisticamente significativa entre as distribuições analisadas, o que pode justificar a realização de ações de retreinamento ou reavaliação completa do modelo.

Figura 20 – *Drift Score* para todos os níveis de ruído



Fonte: Adaptado de [deepchecks \(2023\)](#).

Os gráficos de barra na Figura 20 representam os resultados dos experimentos da Tabela 2, destacando a relação entre os níveis de ruído e os respectivos *Drift Scores*.

No gráfico relacionado ao *Label Drift* Amostra por Classe Treino e Teste, é observável que o *Drift Score Cramer's V* mantém-se relativamente estável para ruídos acima de 0,05%, tendo um ligeiro aumento inicial do ruído de 0,00% para 0,05%.

Os gráficos referentes às propriedades das imagens, como *RMS Contraste*, *Mean Blue Relative Intensity*, *Mean Green Relative Intensity*, *Brightness*, *Area* e *Aspect Ratio*,

apresentam diferentes comportamentos. Notavelmente, *Area* e *Aspect Ratio* permanecem constantes em 0% através de todos os níveis de ruído, indicando que essas propriedades não são afetadas pelo ruído. Por outro lado, propriedades como *Brightness* mostram uma tendência crescente com o aumento do ruído.

O gráfico *Predição de drift Treino e Teste* demonstra uma relação quase exponencial entre o nível de ruído e o *Drift Score Cramer's V*. Há um aumento marcante na pontuação à medida que o ruído aumenta, indicando uma deterioração significativa na previsão à medida que o ruído aumenta.

O gráfico *Image Dataset Drift*, é perceptível uma tendência crescente no *Drift Score Multivariable Total* com o aumento do ruído, reiterando o impacto do ruído no conjunto geral de dados da imagem.

Os gráficos de barra contidos na Figura 20 ilustram claramente as variações nos *Drift Scores* em relação aos diferentes níveis de ruído, fornecendo uma visualização intuitiva dos resultados dos experimentos.

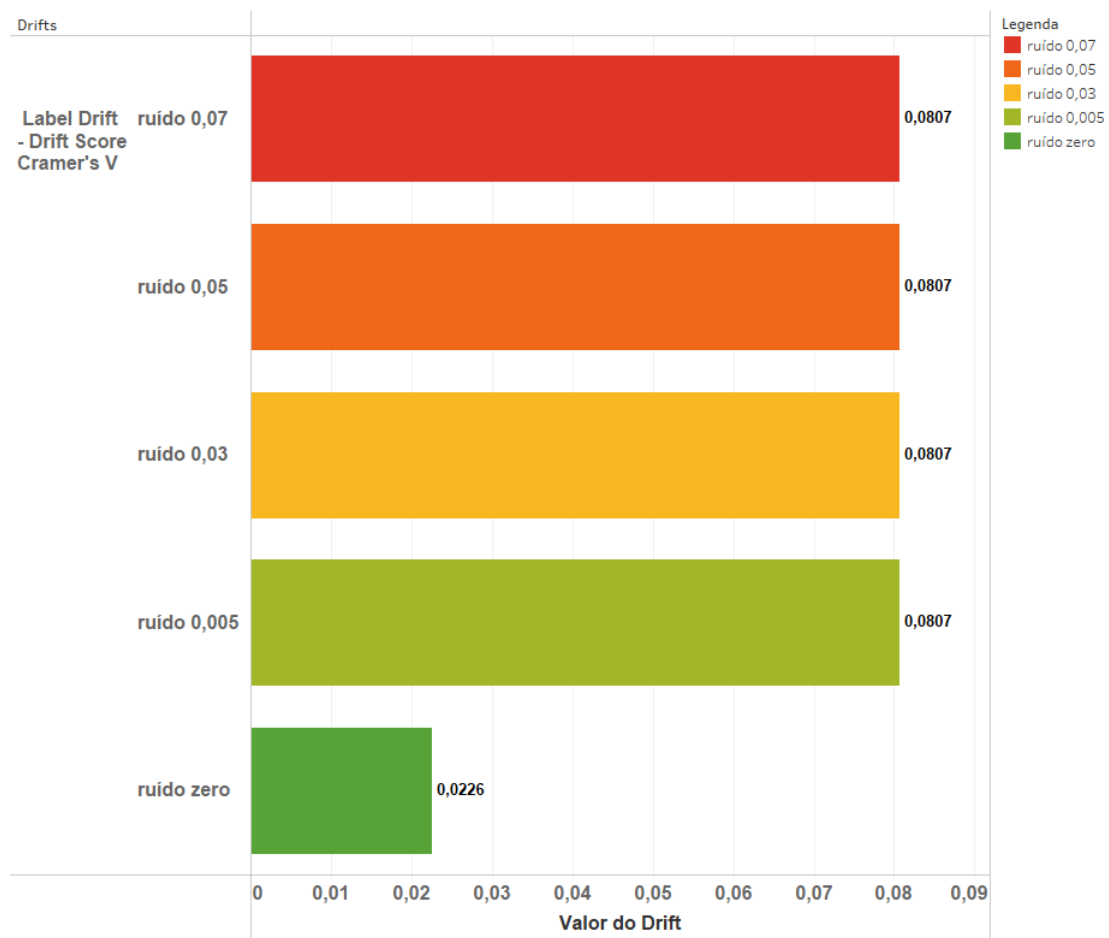
4.3 Resultados *Train Test Label Drift*.

A avaliação do *Label Drift* é fundamental para entender a consistência entre os conjuntos de treino e teste. Essencialmente, o *Label Drift* refere-se ao desvio observado nas distribuições de etiquetas entre os dois conjuntos. Isso é essencialmente a discrepância entre as distribuições de dados de treino e teste com base em amostras por classe.

Dada a natureza categórica dessa distribuição, a métrica de Cramer's V é empregada para quantificar o desvio. Esta métrica é útil para avaliar a diferença entre duas distribuições de variáveis categóricas. Assim, a magnitude do *Drift* é determinada calculando a diferença entre a amostra mais recente e a amostra anterior.

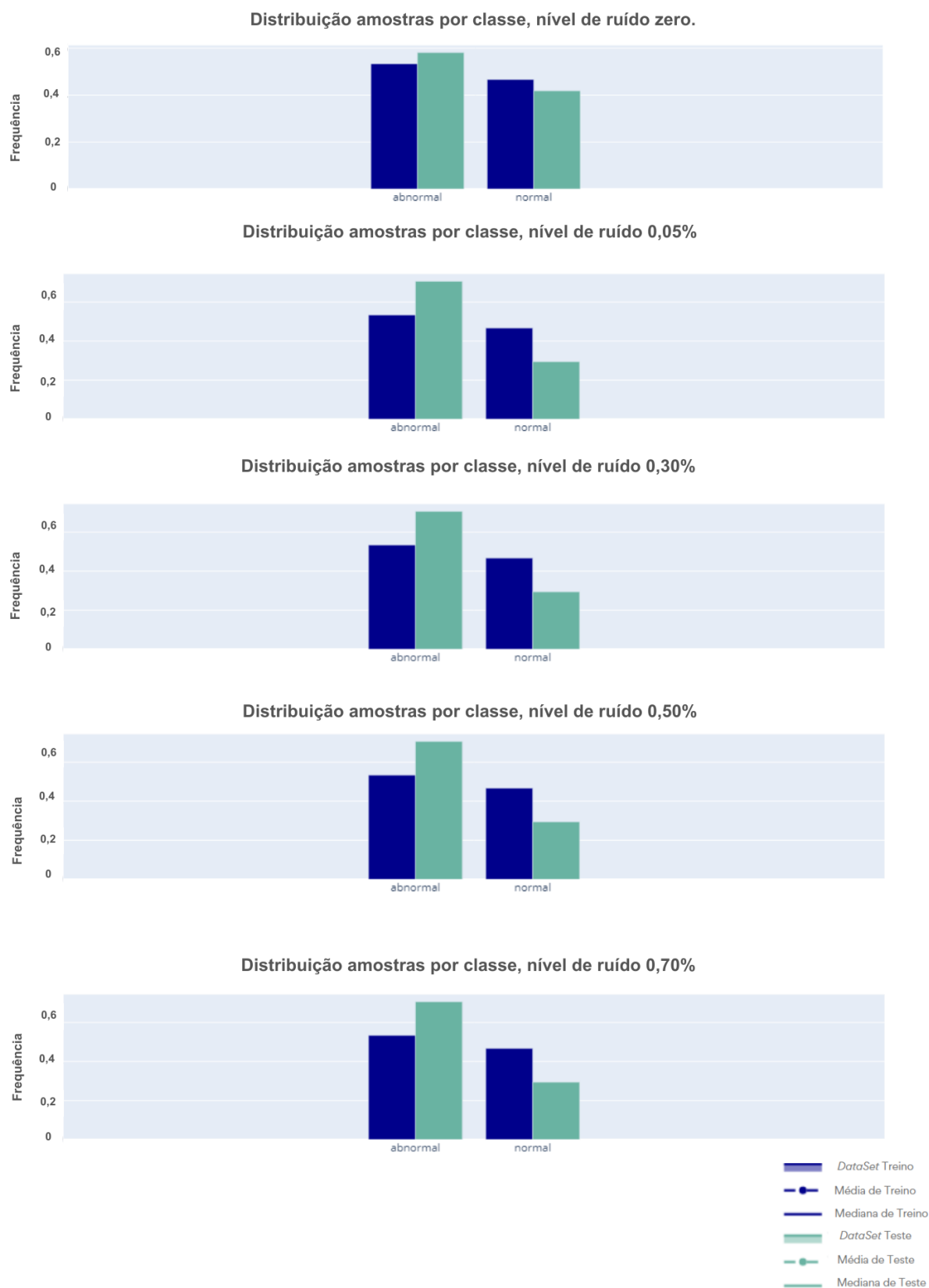
Os dados da Tabela 2 indicam que o *Drift* para um nível de ruído de 0% é de 2,2573%. Isso sugere que, mesmo na ausência de ruído externo, existe uma discrepância inerente entre os conjuntos de treino e teste.

Figura 21 – Train test label drift score



Fonte: Adaptado de [deepchecks \(2023\)](#).

Figura 22 – Train test label drift comparativo distribuição



Fonte: Adaptado de [deepchecks \(2023\)](#).

Além disso, ao examinar a Figura 22, obtemos uma representação gráfica da comparação entre as distribuições com base na frequência. Isso nos permite visualizar o *range* do *Drift Score* para um nível de ruído de 0%. Segundo os padrões estabelecidos, um *Drift Score* categórico menor que 0,15 é considerado satisfatório. Isso é avaliado comparando-se a densidade da probabilidade das distribuições.

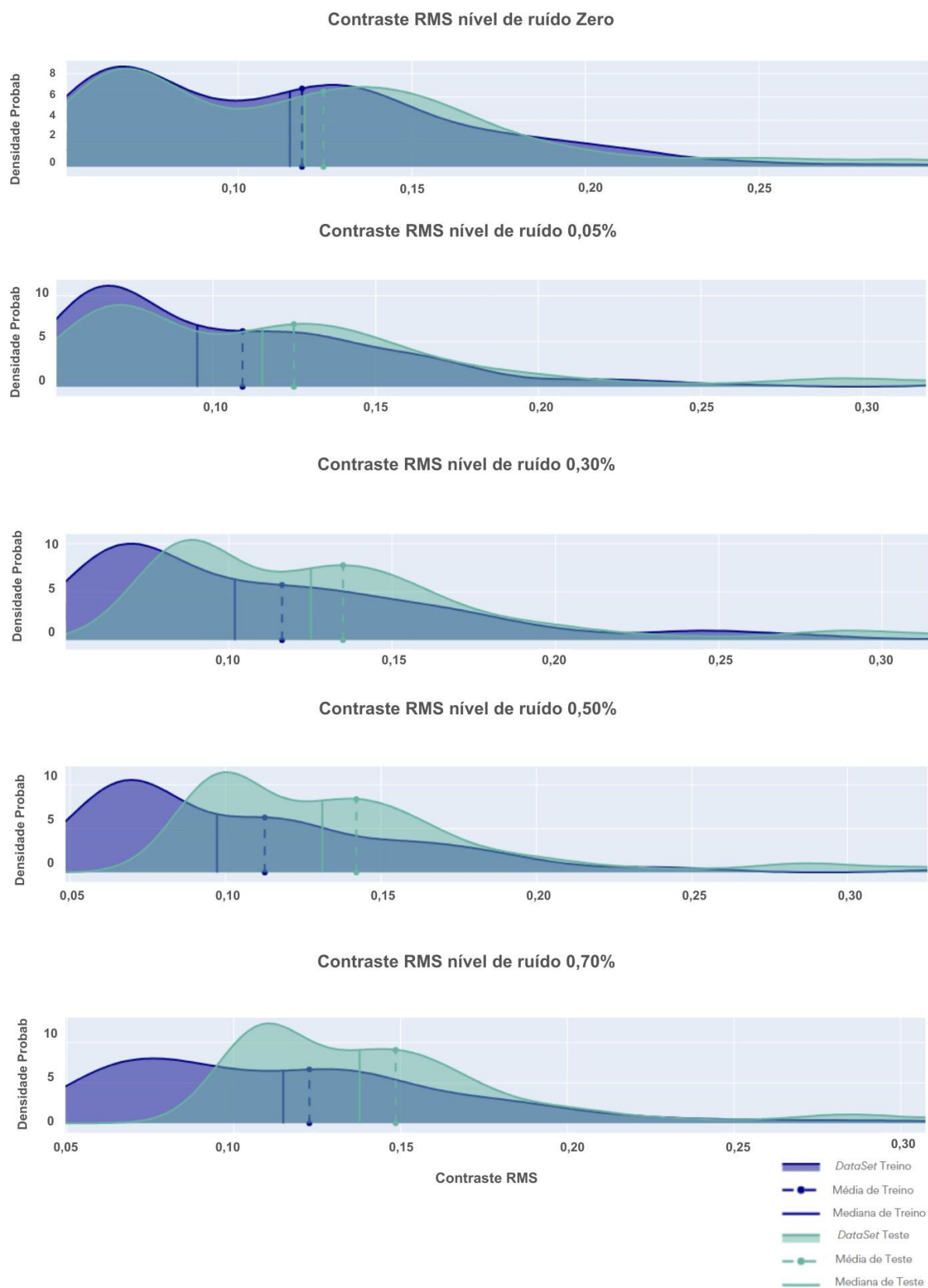
Para distribuições numéricas contínuas, a métrica de Wasserstein, também conhecida como *Earth Mover's Distance*, é adotada. Neste contexto, para que o resultado seja considerado satisfatório, o valor de *Earth Mover's Distance* deve ser inferior a 0,1.

É fundamental notar que a detecção de *Label Drift* é crucial para garantir a confiabilidade dos modelos de aprendizado de máquina. Discrepâncias não detectadas ou não corrigidas entre os conjuntos de treino e teste podem resultar em um desempenho subótimo do modelo, afetando negativamente sua capacidade de generalização em ambientes não vistos.

4.4 Resultados *Image Property Drift*

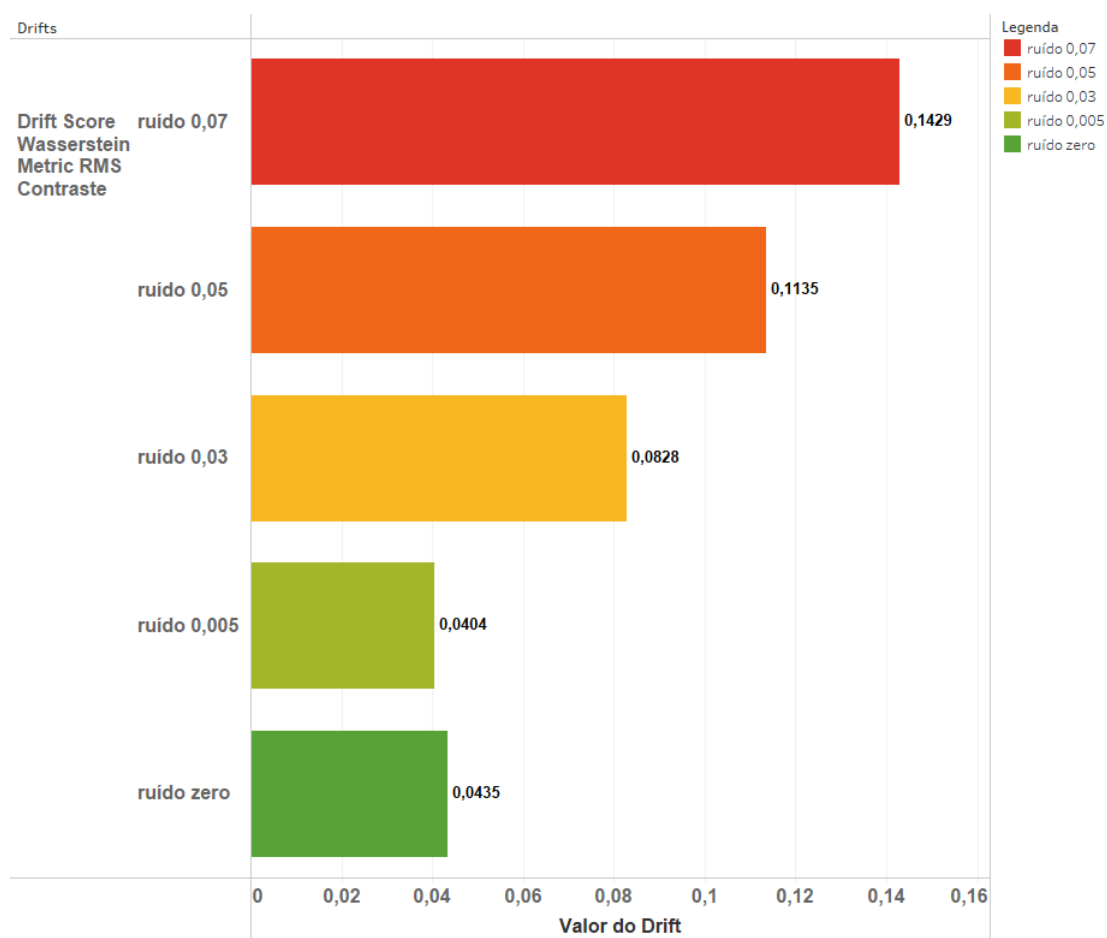
A verificação de *Image Property Drift* é uma etapa crítica na análise de conjuntos de dados de imagem para treinamento e teste em aprendizado de máquina. Esse desvio destaca as variações nas propriedades intrínsecas das imagens entre os dois conjuntos. Em contraste com o *Label Drift*, que se concentra na consistência das etiquetas, o *Image Property Drift* concentra-se nas características subjacentes das imagens, como contraste, intensidade relativa de cores e brilho, entre outras.

A Tabela 2 apresenta uma análise detalhada dos *Drift Scores* para várias propriedades da imagem em diferentes níveis de ruído. Por exemplo, observa-se que o RMS Contraste mostra variações crescentes com o aumento do nível de ruído. Este é um indicativo de que o contraste das imagens pode ser significativamente afetado mesmo por pequenas quantidades de ruído. Da mesma forma, as propriedades *Mean Blue Relative Intensity* e *Mean Green Relative Intensity* apresentam variações menores nos *Drift Scores* à medida que o ruído aumenta, indicando uma resistência relativa dessas propriedades ao ruído. Em contraste, as propriedades *Area* e *Aspect Ratio* permanecem inalteradas através dos níveis de ruído, conforme refletido por seus *Drift Scores* constantes de 0%. Isso sugere que essas características da imagem são robustas e não são facilmente perturbadas pelo ruído. Para quantificar esses desvios, a métrica de Wasserstein, ou *Earth Mover's Distance*, é utilizada, particularmente para distribuições contínuas.

Figura 23 – *Image Property Drift - RMS Contrast* comparativo distribuição

Fonte: Adaptado de [deepchecks \(2023\)](#).

Figura 24 – Image Property Drift Score - RMS Contrast



Fonte: Adaptado de [deepchecks \(2023\)](#).

a Figura 23 permite avaliação do desvio nas propriedades das imagens para o contraste RMS. Essa análise é fundamental para compreender as nuances subjacentes que podem afetar a qualidade do treinamento e a validação de modelos de aprendizado de máquina. Dentro das variadas propriedades de uma imagem, o *RMS Contrast* ou *Contraste RMS (Root Mean Square Contrast)* é um parâmetro crucial que descreve a magnitude das variações de intensidade em uma imagem.

O *RMS Contrast* é uma métrica que quantifica a diferença de contraste entre regiões claras e escuras de uma imagem. Um desvio nesta propriedade pode indicar variações nos detalhes da imagem, o que pode ser fundamental, especialmente em aplicações como reconhecimento de imagens e diagnóstico médico por imagem.

De acordo com a Figura 23 é perceptível a variação das distribuições e aumento da distância.

Esta observação sugere que a presença de ruído pode afetar de maneira significativa

o contraste das imagens, possivelmente obscurecendo detalhes importantes ou introduzindo artefatos que não estavam presentes na imagem original.

Ao considerar as métricas utilizadas para avaliar esse desvio, a métrica de Wasserstein, também conhecida como *Earth Mover's Distance*, é aplicada. Este é um indicador robusto que quantifica a diferença entre duas distribuições. Em relação ao *RMS Contrast*, um *Drift Score* satisfatório, quando medido pela distância de Wasserstein, é considerado abaixo de 0,1 *deepchecks*. Isso sugere uma sobreposição substancial nas distribuições de contraste entre os conjuntos de treino e teste.

A avaliação do *Image Property Drift* no que tange ao *RMS Contrast* fornece insights valiosos sobre a qualidade e a consistência das imagens usadas no treinamento e na validação. Garantir a homogeneidade nessa propriedade é vital para o desempenho ótimo de algoritmos de processamento e análise de imagens.

A Figura 24 apresenta graficamente o *RMS Contrast* que representa uma métrica essencial e reflete a variação de intensidades em uma imagem. Como um indicador da nitidez e clareza da imagem, variações no *RMS Contrast* podem impactar o discernimento de detalhes finos e características sutis, essenciais em aplicações de visão computacional e análise de imagens.

Ao avaliar o *Drift Score* associado ao *RMS Contrast*, estamos essencialmente quantificando o quanto esta propriedade específica varia entre amostras de treino e teste ao longo do tempo ou sob diferentes condições. Este tipo de análise é fundamental para garantir a consistência e a confiabilidade das imagens que estão sendo utilizadas.

De acordo com a Tabela 2 e a A Figura 24, os *Drift Scores* relacionados ao *RMS Contrast* mostram uma elevação à medida que o nível de ruído aumenta. Para um nível de ruído de 0,00%, o *Drift Score* calculado com a métrica de Wasserstein é de 4,3483%. Este valor indica uma discrepância entre as distribuições de contraste das imagens nos conjuntos de treino e teste. À medida que o ruído cresce, essa discrepância torna-se mais pronunciada, sugerindo que o ruído tem um impacto direto na propriedade de contraste das imagens.

O brilho ou *Brightness*, em análise de imagens, reflete a luminância geral percebida em uma imagem e é uma das propriedades fundamentais que afetam diretamente a percepção e a interpretação visual de uma imagem. Variações no brilho podem alterar o destaque de objetos, nuances e texturas, influenciando decisivamente na performance de algoritmos e na clareza com que informações visuais são transmitidas.

Analisando a Figura 25 percebemos que os *Drift Scores* associados ao brilho apresentam um padrão ascendente com o aumento dos níveis de ruído. No nível de ruído de 0,00%, o *Drift Score* usando a métrica de Wasserstein é de 1,9441%. Tal valor sugere uma leve divergência entre as distribuições de brilho nos conjuntos de treino e teste. Com

a escalada do ruído, essa divergência se amplifica, ainda que modestamente, denotando o efeito adverso do ruído na propriedade de brilho das imagens.

Na Figura 26 o *Drift* associado ao brilho, fica evidente no gráfico de barras quanto a sua escalada juntamente com o incremento do ruído.

O ruído do tipo *salt-and-pepper* é uma perturbação visual comum que introduz *pixels* brancos e pretos aleatórios em uma imagem. Sua presença afeta diferentes propriedades da imagem de maneiras distintas, e a compreensão do motivo pelo qual suas interferências variam para diferentes propriedades é crucial para análises precisas e para a criação de modelos robustos de visão computacional.

Impacto no RMS Contraste (90%): O RMS (*Root Mean Square*) Contraste é uma medida que avalia a variação de intensidade luminosa em uma imagem. Um alto contraste significa que há uma grande diferença nas intensidades dos *pixels* na imagem. O ruído *salt-and-pepper*, ao inserir *pixels* extremamente claros (sal) e escuros (pimenta) de forma aleatória, provoca uma variação drástica nas intensidades luminosas. Dessa forma, a métrica de contraste, que é sensível a mudanças repentinas e variações de intensidade, é fortemente influenciada. Por isso, o impacto deste tipo de ruído no RMS Contraste é significativo, resultando em um aumento médio de 90

Impacto no Brightness (5%): O brilho é uma métrica que reflete a média das intensidades de luminância em toda a imagem. Ele fornece uma visão geral da iluminação ou luminosidade da imagem. O ruído *salt-and-pepper*, embora introduza variações pontuais nas intensidades luminosas, não altera de maneira substancial a média global de intensidades. A introdução de *pixels* extremamente claros e escuros tende a se equilibrar, resultando em uma mudança mínima na métrica de brilho. Assim, o impacto deste ruído no brilho é consideravelmente menor, levando a uma composição total de *Drift* de apenas 5

Desta forma enquanto o RMS Contraste é altamente sensível a variações locais de intensidade, o brilho foca na média global de luminosidade. Portanto, a natureza pontual e extrema do ruído *salt-and-pepper* tem um impacto desproporcional no contraste em comparação com o brilho. Isso justifica a diferença substancial observada nos *Drift Scores* para essas duas propriedades da imagem.

Para compreender o impacto mínimo desse ruído em propriedades como a Intensidade Relativa Média Azul (*Mean Blue Relative Intensity*), Vermelha (*Mean Red Relative Intensity*) e Verde (*Mean Green Relative Intensity*), é preciso analisar a natureza dessas métricas e a interação delas com o tipo de ruído em questão.

Natureza das Métricas de Intensidade Relativa: Estas métricas avaliam a média das intensidades das cores específicas em toda a imagem. Elas não são sensíveis a variações pontuais ou a alterações extremas em locais específicos, mas sim à tendência global da imagem.

Efeito do Ruído *Salt-and-Pepper* : O *salt-and-pepper* introduz *pixels* brancos e pretos. *pixels* brancos são representados como (255, 255, 255) em termos de RGB, enquanto *pixels* pretos são (0, 0, 0). Isso significa que ambos os tipos de ruído têm o mesmo impacto em todas as componentes de cor seja vermelho, verde ou azul.

Dessa forma, quando o ruído *salt-and-pepper* é introduzido, ele afeta igualmente todas as componentes de cor da imagem. Como tal, a proporção ou a relação relativa entre essas cores não muda significativamente, mesmo que a intensidade absoluta possa variar.

Equilíbrio entre *Salt-and-Pepper* : A natureza aleatória do ruído *salt-and-pepper* significa que, em média, a introdução de *pixels* brancos e pretos é equilibrada. Assim, os efeitos desses *pixels* em termos de intensidades relativas tendem a se anular, resultando em um impacto mínimo nas métricas de intensidade relativa média.

Em resumo, enquanto o ruído *salt-and-pepper* pode alterar as intensidades absolutas dos *pixels*, ele não discrimina entre diferentes cores. Como resultado, a intensidade relativa média das cores específicas seja azul, vermelho ou verde permanece quase inalterada, levando a um impacto quase zero nas métricas em questão.

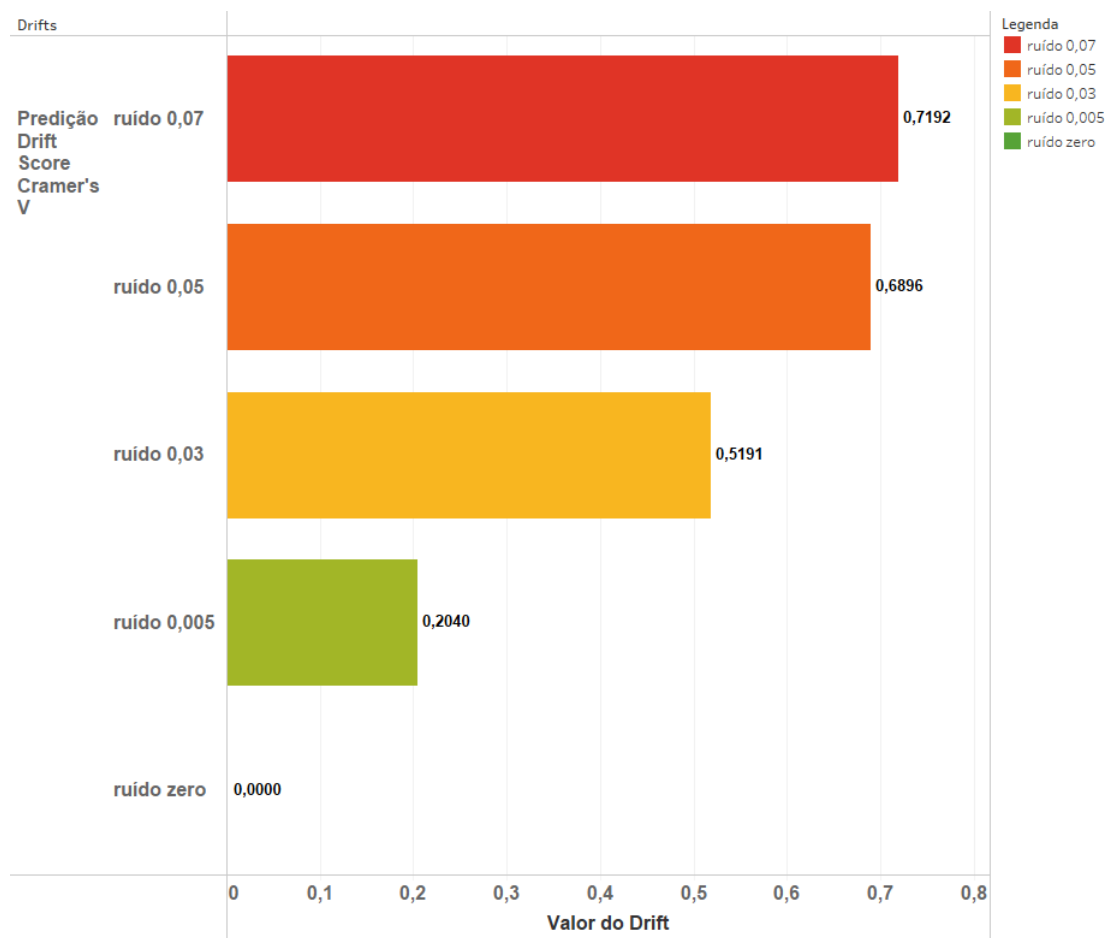
4.5 Resultados *Train Test Prediction Drift*

O *Train Test Prediction Drift* refere-se à variação entre as previsões dos dados de treino e de teste. Este tipo de *Drift* é crucial, pois identifica diferenças na qualidade das previsões entre os conjuntos de treino e teste, ajudando a entender como o modelo poderia se comportar ao ser exposto a dados novos ou desconhecidos.

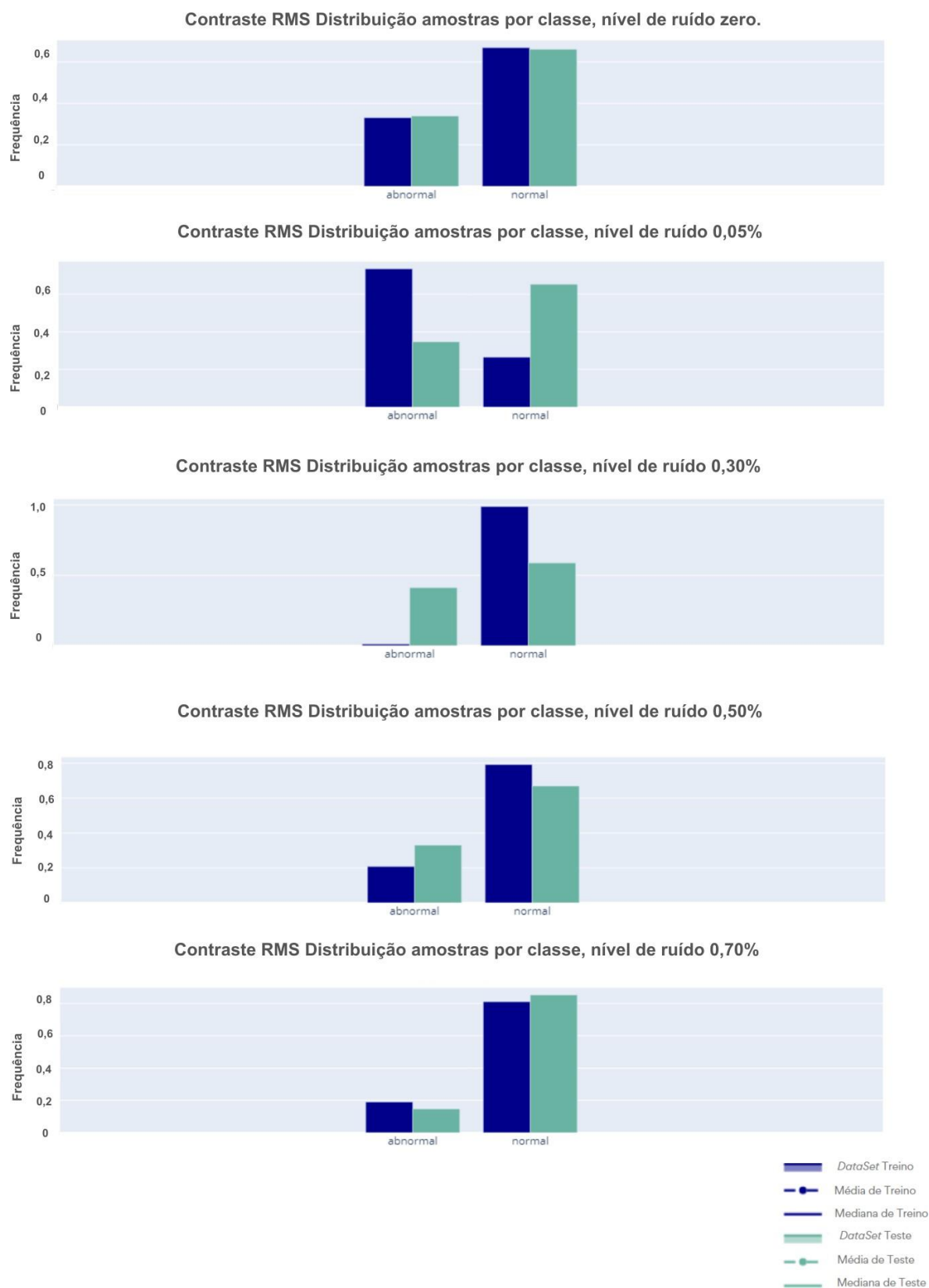
A métrica padrão para avaliar o *Drift Score* categórico é definida como um valor abaixo de 0,15, sendo considerado satisfatório. Isso significa que, em condições ideais, a diferença entre as distribuições de previsão do conjunto de treino e teste deve ser pequena, resultando em um *Drift Score* menor que 0,15.

Ao analisar a Tabela 2, observamos que para um nível de ruído de 0,00 %, o *Drift Score Cramer's V* para *Train Test Prediction Drift* é de 0,0000%, o que é excelente. No entanto, com o aumento incremental do ruído, este score aumenta significativamente, atingindo 20,3998% para um ruído de 0,05% e 71,9248% para um ruído de 0,70%. Estes números indicam que, à medida que o ruído aumenta, a capacidade do modelo de fazer previsões consistentes entre os conjuntos de treino e teste deteriora-se acentuadamente.

Figura 27 – Train Test Prediction Drift Score



Fonte: Adaptado de [deepchecks \(2023\)](#).

Figura 28 – *Train Test Prediction Drift* comparativo distribuição

Fonte: Adaptado de [deepchecks \(2023\)](#).

A Figura 28 ilustra a variação das distribuições para o incremento de ruído, é observado uma separação crescente entre as distribuições de treino e teste à medida que o nível de ruído aumenta. Este comportamento indica que o ruído *salt and pepper* tem um impacto substancial nas previsões do modelo, tornando-as cada vez mais inconsistentes.

A Figura 27, que representa o *Drift Score* em barras, ilustra o aumento no *Drift Score* à medida que o ruído é incrementado. Este gráfico reitera a observação da Tabela 2, enfatizando o impacto negativo do ruído nas previsões do modelo.

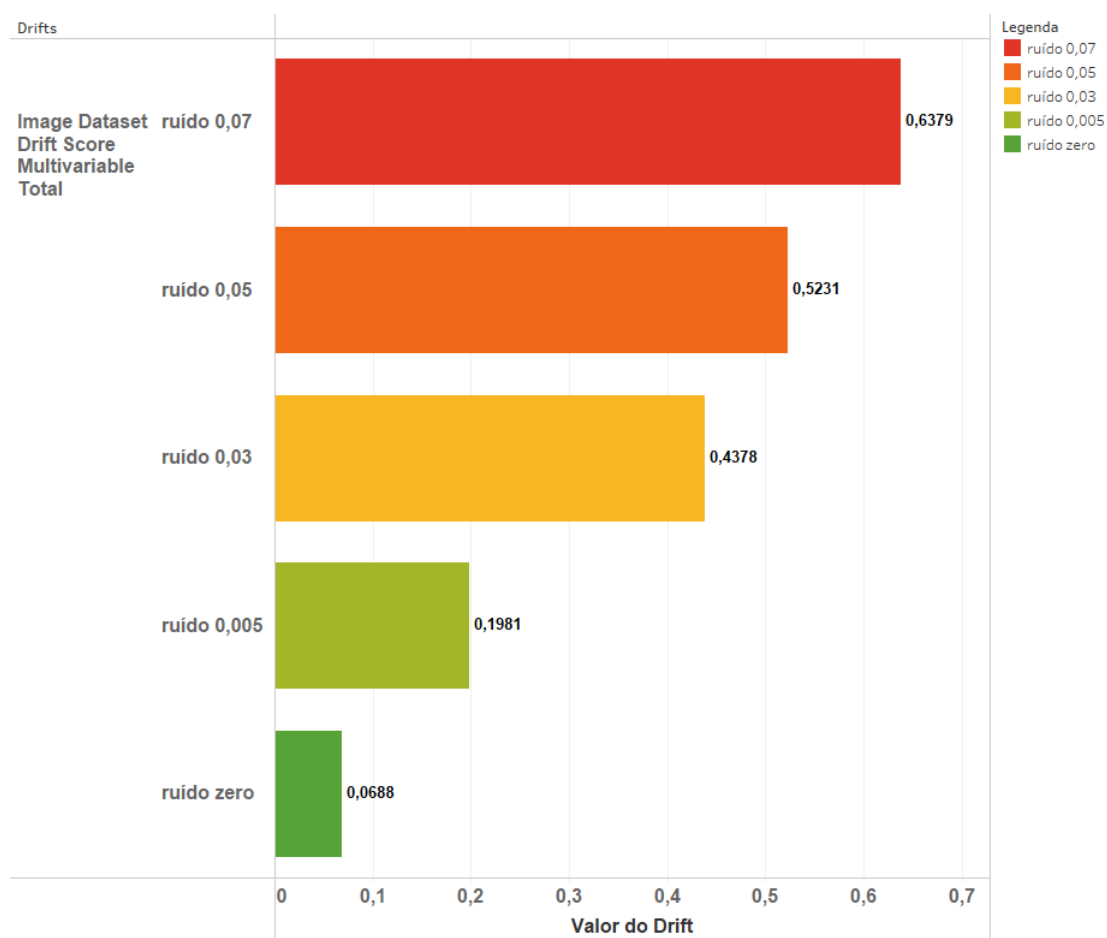
E evidente que a introdução de ruído, mesmo em pequenas quantidades, pode ter um impacto profundo na qualidade e consistência das previsões do modelo. Portanto, é imperativo monitorar e avaliar regularmente o *Train Test Prediction Drift* ao trabalhar em ambientes onde a qualidade dos dados pode variar ou ser comprometida.

4.6 Resultados *Image Dataset Drift*

O *Image Dataset Drift* avalia a variação nos conjuntos de imagens de treino e teste considerando o conjunto total de propriedades das imagens. A análise deste *Drift* é fundamental para entender como alterações em características intrínsecas das imagens podem impactar a performance de modelos de aprendizado de máquina, especialmente quando se trabalha com imagens médicas, como raios-X. A Tabela 2 revela que, para um nível de ruído de 0,00%, o *Drift Score Multivariable Total* para o *Image Dataset Drift* é 0,0000%, indicando uma perfeita consistência entre os conjuntos de treino e teste. Entretanto, ao introduzir ruído no dataset, observamos um aumento progressivo no *Drift Score*, demonstrando que mesmo pequenas perturbações podem afetar a homogeneidade do conjunto.

A Figura 29, por sua vez, ilustra a evolução do *Drift Score* em relação ao aumento do ruído. Esta representação em barras destaca os pontos de inflexão e permite uma compreensão clara de quão rapidamente a consistência do *dataset* é comprometida ao introduzir ruídos.

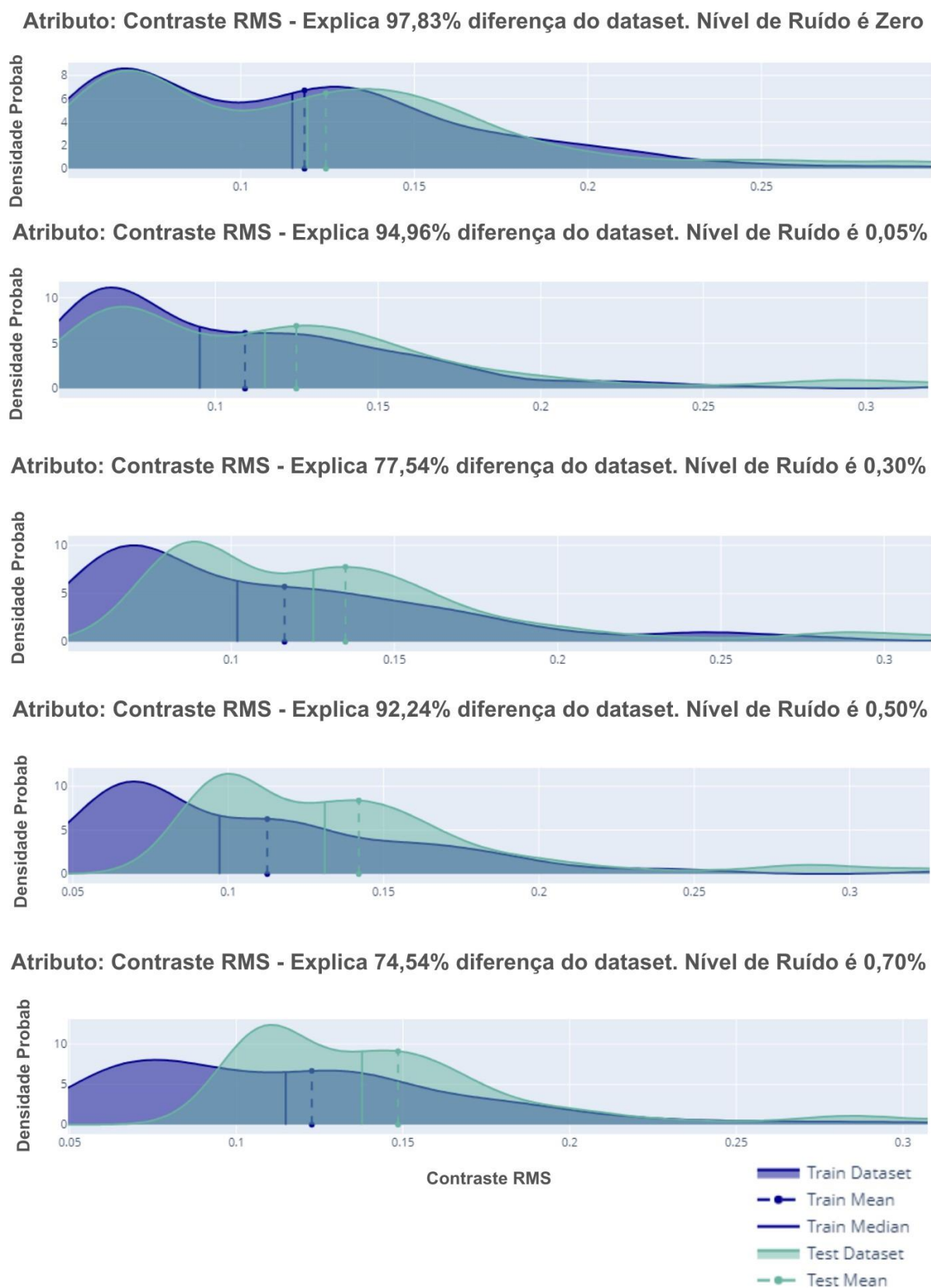
A Figura 30 é apresentado imagens de raios-X do *dataset* MURA para Cotovelo, as imagens dispostas em sequência lado a lado permitem comparar os níveis de ruído com a degradação visual da imagem oferece uma perspectiva visual direta das perturbações introduzidas. Ao comparar estas imagens, é possível discernir as mudanças estéticas causadas pelo ruído *salt and pepper*, reiterando a necessidade de assegurar a integridade e qualidade das imagens, especialmente quando usadas para diagnósticos médicos ou treinamento de modelos.

Figura 29 – Comparativo *Image dataset drift* para todos os níveis de ruído

Fonte: Adaptado de [deepchecks \(2023\)](#).

Figura 30 – Comparativo *Image Dataset Drift* raio-X

Fonte: Adaptado de [deepchecks \(2023\)](#).

Figura 31 – Comparativo distribuições e impacto *Image Dataset Drift*

Fonte: Adaptado de [deepchecks \(2023\)](#).

A Figura 31 traça uma comparação visual das distribuições para cada nível de ruído. Foi destacado o atributo contraste RMS visto que ele é o maior responsável pela ocorrência de *drift*. É possível observar uma crescente divergência entre as distribuições de treino e teste à medida que o nível de ruído aumenta. Esse afastamento entre as distribuições sublinha a sensibilidade do conjunto de dados a perturbações e a importância de monitorar a qualidade das imagens.

Em resumo, os resultados do *Image Dataset Drift* enfatizam a sensibilidade dos conjuntos de dados de imagem a variações e a importância de manter uma qualidade consistente nas imagens. O monitoramento regular deste *Drift* é crucial, particularmente em aplicações críticas como diagnósticos médicos.

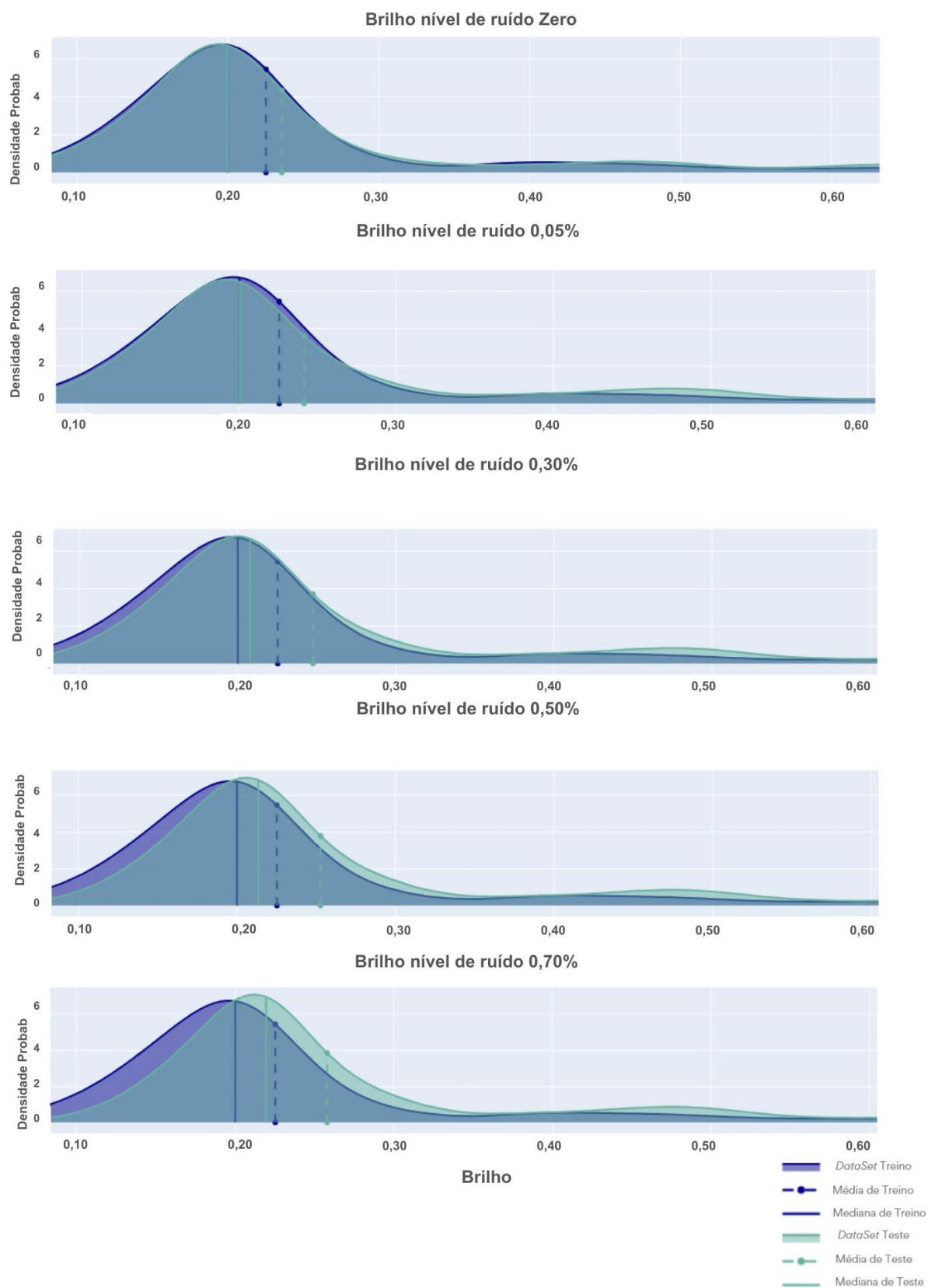
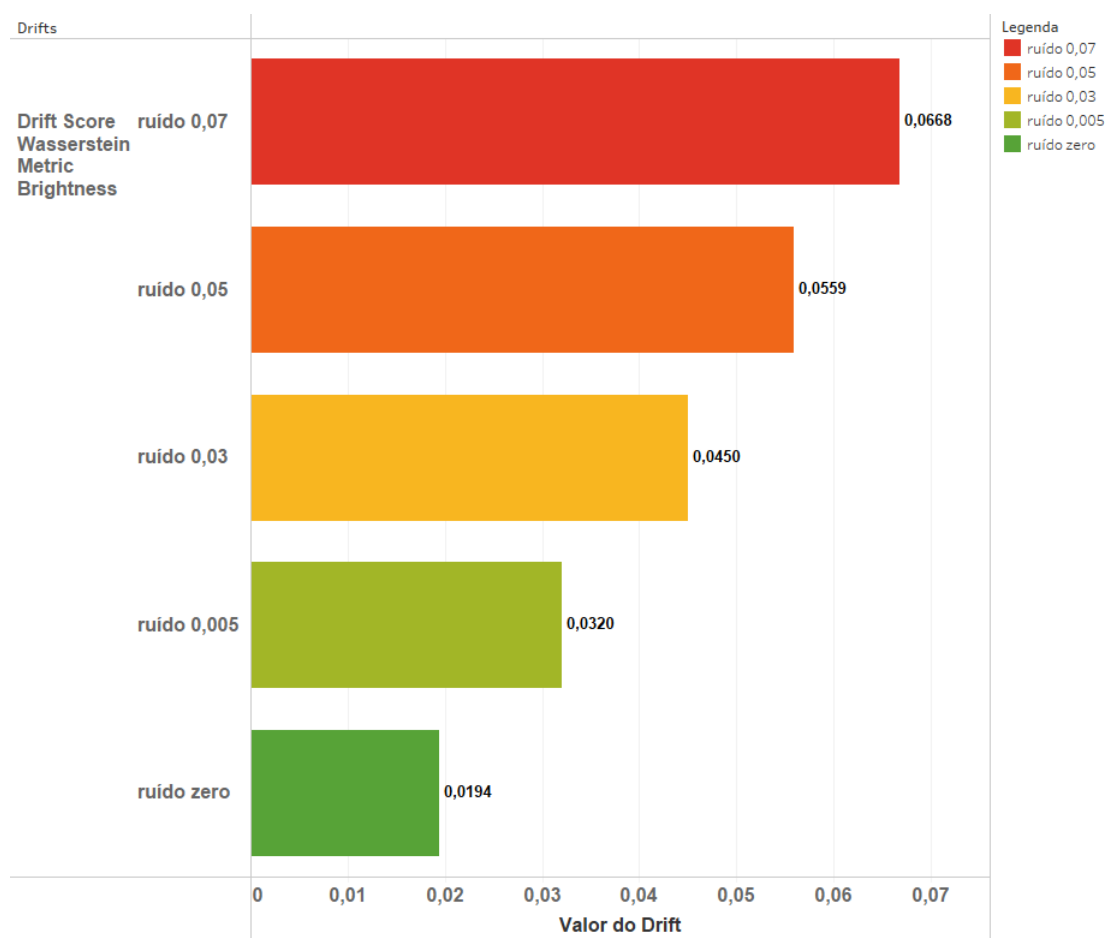
Figura 25 – *Image Property Drift - Brightness* comparativo distribuiçãoFonte: Adaptado de [deepchecks \(2023\)](#).

Figura 26 – Image Property Drift Score - Brightness



Fonte: Adaptado de [deepchecks \(2023\)](#).

5 Conclusões

Ao decorrer deste estudo, exploramos profundamente os conceitos de *Data Drift* e *Concept Drift* em visão computacional. Na aplicação prática o experimento tornou possível a visualização do impacto do ruído, propositalmente, adicionado as imagens. A medida que esse ruído aumenta o detector de *drift* não apenas identifica como quantifica e exhibe essa evolução de forma gráfica. Com o *deepchecks* é possível também identificar quais atributos da imagem estão impactando na produção de *drifts*, que para as características do experimento desenvolvidos tivemos o brilho e principalmente o contraste como atributos mais impactados ou que degradaram com maior impacto. A evolução dos dados e conceitos subjacentes têm o potencial de debilitar significativamente a habilidade dos modelos de aprendizado de máquina de generalizar e, portanto, de produzir resultados precisos e confiáveis. Esta seção enfatiza as principais descobertas e implicações práticas relacionando-se com as discussões e resultados evidenciados nos capítulos anteriores.

A necessidade de detecção e monitoramento contínuo dos *Drifts* foi afirmada como crucial para assegurar a qualidade e eficácia dos modelos de visão computacional. Os métodos e técnicas apresentados e discutidos, como re-treinamento de modelos e uso de algoritmos de detecção de *Drift*, não apenas abordam as diversas facetas do problema, mas também servem como um guia para os praticantes na escolha de estratégias eficazes na gestão do *Drift*.

Nos capítulos anteriores, debatemos que o *Data Drift* pode ser particularmente mitigado mediante a escolha adequada do conjunto de dados de treinamento e sua manutenção qualitativa ao longo do tempo. Foi enfatizado que a utilização de dados diversificados e representativos, juntamente com técnicas de pré-processamento adequadas, podem atenuar o impacto do *Drift*.

Os desafios inerentes ao gerenciamento do *Drift*, e, particularmente, no contexto da visão computacional, foram alvo de discussão detalhada. Foi postulado que o *Drift*, que é caracterizado pela mudança não planejada nas características dos dados de entrada, pode ser gerido adequadamente através de estratégias e técnicas adequadas.

Além disso, é crucial investir em pesquisa e desenvolvimento contínuos para aprimorar as soluções existentes e explorar novas abordagens para lidar com o *Drift*. A comunidade científica e a indústria devem trabalhar em conjunto para compartilhar conhecimentos, dados e experiências, a fim de avançar no campo do gerenciamento do *Drift* e garantir a confiabilidade e a efetividade dos sistemas de visão computacional em diferentes cenários.

O gerenciamento do *Drift* é um desafio em constante evolução que requer atenção contínua e a adoção de técnicas avançadas. Somente através do monitoramento adequado, da detecção oportuna e da adaptação dos modelos, será possível garantir que os sistemas de visão computacional continuem a fornecer resultados confiáveis e relevantes em um ambiente em constante mudança.

5.1 Trabalhos Futuros

Embora tenham sido apresentadas diversas técnicas e abordagens para lidar com o problema do *Data Drift* e *Concept Drift* em visão computacional, ainda há espaço para a investigação de novas soluções e abordagens. Algumas possibilidades para trabalhos futuros incluem:

- Desenvolvimento de algoritmos mais robustos e eficientes de detecção de *Drift*: embora existam algumas soluções disponíveis, ainda há muitas situações em que a detecção de *Drift* pode ser desafiadora. Novos algoritmos que explorem diferentes aspectos dos dados e dos modelos podem ser desenvolvidos para melhorar a precisão e eficácia da detecção.
- Estudo de novas técnicas de adaptação de modelos: embora as técnicas de adaptação de modelos sejam amplamente utilizadas, ainda há espaço para o desenvolvimento de novas abordagens que possam ser mais efetivas em situações específicas. O uso de técnicas de transferência de aprendizado e aprendizado incremental pode ser investigado em detalhes para entender sua aplicabilidade em diferentes cenários.
- Investigação de métodos para lidar com *Drift* em aplicações de visão computacional em tempo real: muitas aplicações de visão computacional requerem resultados em tempo real, o que torna o gerenciamento do *Drift* ainda mais desafiador. Estudos podem ser realizados para investigar abordagens específicas para lidar com *Drifts* em tempo real, incluindo o uso de técnicas de aprendizado por reforço.
- Avaliação de técnicas de geração de dados sintéticos para treinamento de modelos: a qualidade e a diversidade do conjunto de dados de treinamento são fundamentais para evitar o surgimento de *Drifts*. O uso de técnicas de geração de dados sintéticos pode ser avaliado para entender sua eficácia em melhorar a qualidade e diversidade do conjunto de dados de treinamento.
- Estudo do impacto do *Drift* em diferentes tarefas de visão computacional: o impacto do *Drift* pode variar dependendo da tarefa específica de visão computacional. Estudos podem ser realizados para avaliar o impacto do *Drift* em diferentes tarefas, incluindo reconhecimento de objetos, detecção de anomalias e análise de comportamento.

Em resumo, há diversas possibilidades de trabalhos futuros relacionados ao tema do *Data Drift* e *Concept Drift* em visão computacional. Investigar novas abordagens e soluções pode contribuir para melhorar a qualidade e eficácia dos modelos de visão computacional em diferentes aplicações e cenários.

Referências

ABU-DOUSH, I. et al. Enhancing multilayer perceptron neural network using archive-based harris hawks optimizer to predict gold prices. **Journal of King Saud University - Computer and Information Sciences**, p. 101557, 2023. ISSN 1319-1578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157823001039>.

ACKERMAN, S. et al. **Detection of data drift and outliers affecting machine learning model performance over time**. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2012.09258>.

ACKERMAN, S. et al. Automatically detecting data drift in machine learning classifiers. **arXiv e-prints**, p. arXiv:2111.05672, nov. 2021.

ADOLFI, F.; BOWERS, J. S.; POEPEL, D. Successes and critical failures of neural networks in capturing human-like speech recognition. **Neural Networks**, v. 162, p. 199–211, 2023. ISSN 0893-6080. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608023001016>.

AGHASI, A. et al. A decentralized adaptation of model-free q-learning for thermal-aware energy-efficient virtual machine placement in cloud data centers. **Computer Networks**, v. 224, p. 109624, 2023. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128623000695>.

AGRAHARI, S.; SINGH, A. K. Concept drift detection in data stream mining : A literature review. **Journal of King Saud University - Computer and Information Sciences**, 2021. ISSN 1319-1578. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157821003062>.

AHMED, M. et al. An inception v3 approach for malware classification using machine learning and transfer learning. **International Journal of Intelligent Networks**, v. 4, p. 11–18, 2023. ISSN 2666-6030. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666603022000252>.

ALABBASI, A.; KHALIL, M.; MCGRAIL, T. Integrating narx neural network with k-s test for accurate partial discharge detection in transformers. In: **2023 18th Conference on Electrical Machines, Drives and Power Systems (ELMA)**. [S.l.: s.n.], 2023. p. 1–7.

ALATRANY, A. S. et al. Transfer learning for classification of alzheimer's disease based on genome wide data. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, p. 1–12, 2023.

ALGHAMDI, W. Y. A novel deep learning method for predicting athletes' health using wearable sensors and recurrent neural networks. **Decision Analytics Journal**, v. 7, p. 100213, 2023. ISSN 2772-6622. Disponível em: <https://www.sciencedirect.com/science/article/pii/S277266222300053X>.

ALKAISSY, M. et al. Enhancing construction safety: Machine learning-based classification of injury types. **Safety Science**, v. 162, p. 106102, 2023. ISSN 0925-7535. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925753523000449>.

AMEISEN, E. **Building Machine Learning Powered Applications: Going from Idea to Product**. [S.l.]: O'Reilly Media, 2020. ISBN 978-1492045113.

ANDAUER NAVARRO, C. L. et al. Systematic review identifies the design and methodological conduct of studies on machine learning-based prediction models. **Journal of Clinical Epidemiology**, v. 154, p. 8–22, 2023. ISSN 0895-4356. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0895435622003006>.

ASHOK, S.; EZHUMALAI, S.; PATWA, T. Remediating data drifts and re-establishing ml models. **Procedia Computer Science**, v. 218, p. 799–809, 2023. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050923000601>.

ASRAV, T.; AYDIN, E. Physics-informed recurrent neural networks and hyper-parameter optimization for dynamic process systems. **Computers Chemical Engineering**, v. 173, p. 108195, 2023. ISSN 0098-1354. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0098135423000649>.

ATBAN, F.; EKINCI, E.; GARIP, Z. Traditional machine learning algorithms for breast cancer image classification with optimized deep features. **Biomedical Signal Processing and Control**, v. 81, p. 104534, 2023. ISSN 1746-8094. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1746809422009880>.

BABANNE, V. et al. Machine learning based smart surveillance system. In: **2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)**. [S.l.: s.n.], 2019. p. 84–86.

BABU, S. C.; GAJANAN, S. N. Chapter 4 - effects of technology adoption and gender of household head: the issue, its importance in food security—application of cramer's v and phi coefficient. In: BABU, S. C.; GAJANAN, S. N. (Ed.). **Food Security, Poverty and Nutrition Policy Analysis (Third Edition)**. Third edition. San Diego: Academic Press, 2022. p. 105–133. ISBN 978-0-12-820477-1. Disponível em: <https://www.sciencedirect.com/science/article/pii/B978012820477100019X>.

BACHUTE, M. R.; SUBHEDAR, J. M. Autonomous driving architectures: Insights of machine learning and deep learning algorithms. **Machine Learning with Applications**, v. 6, p. 100164, 2021. ISSN 2666-8270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827021000827>.

BARRIENTOS, D. et al. A legal information system for intelligent sentence mining applied to civil law. In: **2022 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. Montevideo, Uruguay: IEEE, 2022. p. 1–6.

BAYAR, G.; HAMBARCI, G. Improving measurement accuracy of indoor positioning system of a mecanum wheeled mobile robot using monte carlo - latin hypercube sampling based machine learning algorithm. **Journal of the Franklin Institute**, 2022. ISSN 0016-0032. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0016003222005051>.

- BAYRAM, F.; AHMED, B. S.; KASSLER, A. From concept drift to model degradation: An overview on performance-aware drift detectors. **Knowledge-Based Systems**, v. 245, p. 108632, 2022. ISSN 0950-7051. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0950705122002854>>.
- BEHERA, T. K.; KHAN, M. A.; BAKSHI, S. Brain mr image classification using superpixel-based deep transfer learning. **IEEE Journal of Biomedical and Health Informatics**, p. 1–11, 2022.
- BIRCHHA, V.; NIGAM, B. Performance analysis of averaged perceptron machine learning classifier for breast cancer detection. **Procedia Computer Science**, v. 218, p. 2181–2190, 2023. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050923001941>>.
- BISHOP, C. **Pattern Recognition and Machine Learning**. Springer, 2006. Disponível em: <<https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/>>.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. Printed in Singapore: Springer Science+Business Media, LLC, 2006.
- BOLKIAH, A. H. A. A. et al. Crime scene prediction using the integration of k-means clustering and support vector machine. In: **2022 IEEE 10th Conference on Systems, Process Control (ICSPC)**. Malacca, Malaysia: IEEE, 2022. p. 242–246.
- BOMMIDI, B. S.; TEEPARTHI, K.; KOSANA, V. Hybrid wind speed forecasting using iceemdan and transformer model with novel loss function. **Energy**, v. 265, p. 126383, 2023. ISSN 0360-5442. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360544222032698>>.
- BOSMAN, A. S.; ENGELBRECHT, A.; HELBIG, M. Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions. **Neurocomputing**, v. 400, p. 113–136, 2020. ISSN 0925-2312. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231220303593>>.
- BRAGA, A. de P.; CARVALHO, A. de L. F.; LUDERMIR, T. **Redes neurais artificiais: teoria e aplicações**. LTC Editora, 2007. ISBN 9788521615644. Disponível em: <<https://books.google.com.br/books?id=R-p1GwAACAAJ>>.
- BRZEZINSKI, D.; STEFANOWSKI, J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. **Neural Networks and Learning Systems, IEEE Transactions on**, v. 25, p. 81–94, 05 2014.
- BU, Y.; SHI, L.; MA, B. Method for establishing predictive models for total organic halogen based on piecewise interpolation and machine learning. **Journal of Environmental Chemical Engineering**, p. 109928, 2023. ISSN 2213-3437. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S221334372300667X>>.
- CAO, Z. et al. Spectral classification by generative adversarial linear discriminant analysis. **Analytica Chimica Acta**, p. 341129, 2023. ISSN 0003-2670. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0003267023003501>>.

CHEN, A. et al. Multi-source monitoring information fusion method for dam health diagnosis based on wasserstein distance. **Information Sciences**, v. 632, p. 378–389, 2023. ISSN 0020-0255. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0020025523003523>>.

CHEN, X. et al. A novel loss function of deep learning in wind speed forecasting. **Energy**, v. 238, p. 121808, 2022. ISSN 0360-5442. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360544221020569>>.

CHOE, G. et al. Machine learning-assisted statistical variation analysis of ferroelectric transistor: From experimental metrology to adaptive modeling. **IEEE Transactions on Electron Devices**, p. 1–6, 2023.

COHEN, L.; GRINSHPOUN, T.; WEISS, G. Efficient optimal kolmogorov approximation of random variables. **Artificial Intelligence**, v. 329, p. 104086, 2024. ISSN 0004-3702. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0004370224000225>>.

COLLEDANI, D.; ANSELMINI, P.; ROBUSTO, E. Machine learning-decision tree classifiers in psychiatric assessment: An application to the diagnosis of major depressive disorder. **Psychiatry Research**, v. 322, p. 115127, 2023. ISSN 0165-1781. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016517812300080X>>.

CONDRAN, S. et al. Machine learning in precision agriculture: A survey on trends, applications and evaluations over two decades. **IEEE Access**, v. 10, p. 73786–73803, 2022.

CRAMER, H. **Mathematical Methods of Statistics (Pms-9), Volume 9**. ebook. Princeton University Press, 2016. (Princeton landmarks in mathematics and physics). ISBN 1400883865; 9781400883868. Disponível em: <libgen.li/file.php?md5=4e273160396c8e82c11095462f1a0b7d>.

DANDEKAR, A. **Towards autonomic orchestration of machine learning pipelines in future networks**. 2021.

DAS, S. et al. Interpretation of ekg with image recognition and convolutional neural networks. **Current Problems in Cardiology**, p. 101744, 2023. ISSN 0146-2806. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0146280623001615>>.

DEEPCHECKS. **deepchecks-opensource**. 2023. Disponível em: <<https://deepchecks.com/open-source/>>, Acesso em: 2023-01-09.

DEMIR, A. S.; SAGLAM, E. B. Electricity consumption estimation with differential polynomial and artificial neural networks: Case of türkiye. **Journal of Engineering Research**, p. 100073, 2023. ISSN 2307-1877. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2307187723000743>>.

DERAKHSHAN, B.; MARKL, V. Continuous deployment of machine learning pipelines. In: TECHNISCHE UNIVERSITÄT BERLIN. **DFKI GmbH**. Berlin: Open Proceedings, 2019. Disponível em: <https://openproceedings.org/2019/conf/edbt/EDBT19_paper_23.pdf>.

DISABATO, S.; ROVERI, M. Tiny machine learning for concept drift. **IEEE Transactions on Neural Networks and Learning Systems**, p. 1–12, 2022.

DU, H. et al. An ensemble learning approach for land use/land cover classification of arid regions for climate simulation: A case study of xinjiang, northwest china. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, p. 1–15, 2023.

DU, Y. et al. Diagnoses in multiple types of cancer based on serum raman spectroscopy combined with a convolutional neural network: gastric cancer, colon cancer, rectal cancer, lung cancer. **Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy**, p. 122743, 2023. ISSN 1386-1425. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1386142523004286>>.

FAN, T. et al. A new deep convolutional neural network incorporating attentional mechanisms for ecg emotion recognition. **Computers in Biology and Medicine**, p. 106938, 2023. ISSN 0010-4825. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0010482523004031>>.

FARUK, M. J. H. et al. Software engineering process and methodology in blockchain-oriented software development: A systematic study. In: **2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA)**. Las Vegas, NV, USA: IEEE, 2022. p. 120–127.

FENG, C.; LIU, Y.; ZHAO, H. Periodic measures and wasserstein distance for analysing periodicity of time series datasets. **Communications in Nonlinear Science and Numerical Simulation**, v. 120, p. 107166, 2023. ISSN 1007-5704. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1007570423000849>>.

FORECASTING the future of artificial intelligence with machine learning-based link prediction in an exponentially growing knowledge network. **Nature Machine Intelligence**, 2023. Disponível em: <<https://www.nature.com/articles/s42256-023-00498-4>>.

FOUNDATION, P. S. **Python Programming**. 2023. Disponível em: <<https://www.python.org/>>, Acesso em: 2023-10-09.

FRIAS-PAREDES, L. et al. Dynamic mean absolute error as new measure for assessing forecasting errors. **Energy Conversion and Management**, v. 162, p. 176–188, 2018. ISSN 0196-8904. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0196890418301341>>.

FU, Y. et al. Exploring the effects of training samples on the accuracy of crop mapping with machine learning algorithm. **Science of Remote Sensing**, v. 7, p. 100081, 2023. ISSN 2666-0172. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2666017223000068>>.

GALMEANU, H.; ANDONIE, R. Weighted incremental–decremental support vector machines for concept drift with shifting window. **Neural Networks**, v. 152, p. 528–541, 2022. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0893608022001927>>.

GAMA, J. a. et al. A survey on concept drift adaptation. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 46, n. 4, mar 2014. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/2523813>>.

GAUNT, R. E.; LI, S. Bounding kolmogorov distances through wasserstein and related integral probability metrics. **Journal of Mathematical Analysis and Applications**, v. 522, n. 1, p. 126985, 2023. ISSN 0022-247X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022247X22009994>>.

GE, Z. et al. Reading is not believing: A multimodal adversarial attacker for chinese-nlp model. **Computers Security**, v. 125, p. 103052, 2023. ISSN 0167-4048. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167404822004448>>.

GERON, A. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2019.

GHIASI, A.; NG, C.-T.; SHEIKH, A. H. Damage detection of in-service steel railway bridges using a fine k-nearest neighbor machine learning classifier. **Structures**, v. 45, p. 1920–1935, 2022. ISSN 2352-0124. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S235201242200916X>>.

GIFT, A. D. N. **Practical MLOps: Operationalizing Machine Learning Models**. Operationalizing Machine Learning Models: O'Reilly Media, Inc., 2021.

GLOBAL survey: The state of AI in 2020. 2020. Disponível em: <<https://www.mckinsey.com/capabilities/quantumblack/our-insights/global-survey-the-state-of-ai-in-2020>>.

GONZALEZ, R.; WOODS, R. **Processamento Digital De Imagens**. 4. ed. India: Pearson India, 2018. ISBN 9353062985.

GOODFELLOW, Y. B. I.; COURVILLE, A. **Deep learning**. Adaptive computation and machine learning series: Cambridge, MA : MIT Press, 2016.

GOODFELLOW, Y. B. I.; COURVILLE, A. **Deep learning**. Adaptive computation and machine learning series: Cambridge, MA : MIT Press, 2016.

GOOGLE. **Google Colab**. 2023. Disponível em: <<https://colab.research.google.com/>>, Acesso em: 2023-10-09.

GRABUS, S.; LOGAN, P. M.; GREENBERG, J. Temporal Concept Drift and Alignment: An empirical approach to comparing Knowledge Organization Systems over time. **arXiv e-prints**, p. arXiv:2208.07835, ago. 2022.

GRANLUND, T. et al. **MLOps Challenges in Multi-Organization Setup: Experiences from Two Real-World Cases**. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2103.08937>>.

GUO, H. et al. Concept drift type identification based on multi-sliding windows. **Information Sciences**, v. 585, p. 1–23, 2022. ISSN 0020-0255. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0020025521011439>>.

GUPTA, B. B. et al. Novel graph-based machine learning technique to secure smart vehicles in intelligent transportation systems. **IEEE Transactions on Intelligent Transportation Systems**, p. 1–9, 2022.

HALDAR, D. et al. Unsupervised machine learning using k-means identifies radiomic subgroups of pediatric low-grade gliomas that correlate with key molecular markers. **Neoplasia**, v. 36, p. 100869, 2023. ISSN 1476-5586. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S147655862200094X>>.

HASTIE ROBERT TIBSHIRANI, J. F. T. **The elements of statistical learning: Data mining, inference, and prediction**. 2nd ed. 2009. corr. 3rd printing 5th printing.. ed. Springer, 2009. (Springer Series in Statistics). ISBN 0387848576; 9780387848570; 9780387848587; 0387848584. Disponível em: <libgen.li/file.php?md5=0161e6689920acb72e562a5b8d726f4d>.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.

HIJJI, M. et al. 6g connected vehicle framework to support intelligent road maintenance using deep learning data fusion. **IEEE Transactions on Intelligent Transportation Systems**, p. 1–10, 2023.

HUANG, S.-Y. et al. Image classification and adversarial robustness analysis based on hybrid quantum–classical convolutional neural network. **Optics Communications**, v. 533, p. 129287, 2023. ISSN 0030-4018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030401823000329>>.

HUANG, W.; BAI, X.-M. Machine learning based on-the-fly kinetic monte carlo simulations of sluggish diffusion in ni-fe concentrated alloys. **Journal of Alloys and Compounds**, v. 937, p. 168457, 2023. ISSN 0925-8388. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925838822048484>>.

HUTTUNEN, R. et al. A comparison of signal combinations for deep learning-based simultaneous sleep staging and respiratory event detection. **IEEE Transactions on Biomedical Engineering**, p. 1–12, 2022.

IQBAL, A.; SHAD, M. Y.; YASSEN, M. F. Empirical e-bayesian estimation of hierarchical poisson and gamma model using scaled squared error loss function. **Alexandria Engineering Journal**, v. 69, p. 289–301, 2023. ISSN 1110-0168. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S111001682300090X>>.

ISLAM, M. A. et al. An integrated convolutional neural network and sorting algorithm for image classification for efficient flood disaster management. **Decision Analytics Journal**, v. 7, p. 100225, 2023. ISSN 2772-6622. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2772662223000656>>.

J, S. et al. E-commerce customer churn prevention using machine learning-based business intelligence strategy. **Measurement: Sensors**, v. 27, p. 100728, 2023. ISSN 2665-9174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2665917423000648>>.

JAENSCH, F. et al. Test-driven reward function for reinforcement learning: A contribution towards applicable machine learning algorithms for production systems. **Procedia CIRP**, v. 112, p. 103–108, 2022. ISSN 2212-8271. 15th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 14-16 July 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827122011970>.

JAMALEDDYN, I.; El ayachi, R.; BINIZ, M. An improved approach to arabic news classification based on hyperparameter tuning of machine learning algorithms. **Journal of Engineering Research**, v. 11, n. 2, p. 100061, 2023. ISSN 2307-1877. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2307187723000627>.

JANEZ-MARTINO, F. et al. Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach. **Applied Soft Computing**, v. 139, p. 110226, 2023. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494623002442>.

JEPPESEN, J. et al. Personalized seizure detection using logistic regression machine learning based on wearable ecg-monitoring device. **Seizure**, v. 107, p. 155–161, 2023. ISSN 1059-1311. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1059131123001061>.

JURUMI, I. **Estação Caatinga**. 2022. Disponível em: <https://www.institutojurumi.org.br/2022/04/estacao-caatinga.html>, Acesso em: 2023-10-09.

KHAN, A. et al. Computational and topological properties of neural networks by means of graph-theoretic parameters. **Alexandria Engineering Journal**, v. 66, p. 957–977, 2023. ISSN 1110-0168. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1110016822007347>.

KHAN, A. I.; AL-HABSI, S. Machine learning in computer vision. **Procedia Computer Science**, v. 167, p. 1444–1451, 2020. ISSN 1877-0509. International Conference on Computational Intelligence and Data Science. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050920308218>.

KIDANE, L. et al. When and how to retrain machine learning-based cloud management systems. In: **2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)**. Lyon, France: IEEE, 2022. p. 688–698.

KNOTT, M.; PEREZ-CRUZ, F.; DEFRAEYE, T. Facilitated machine learning for image-based fruit quality assessment. **Journal of Food Engineering**, v. 345, p. 111401, 2023. ISSN 0260-8774. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0260877422004551>.

KONTSEWAYA, Y.; ANTONOV, E.; ARTAMONOV, A. Evaluating the effectiveness of machine learning methods for spam detection. **Procedia Computer Science**, v. 190, p. 479–486, 2021. ISSN 1877-0509. 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921013016>.

KRAWCZYK, B.; CANO, A. Online ensemble learning with abstaining classifiers for drifting and noisy data streams. **Applied Soft Computing**, v. 68, p. 677–692, 07 2018.

KREUZBERGER, D.; KÜHL, N.; HIRSCHL, S. Machine learning operations (mlops): Overview, definition, and architecture. **IEEE Access**, p. 1–1, 2023.

LADJS CABINJS, b.; LASSJS. **Spam detector github**. 2022. Disponível em: <<https://github.com/topics/spam-classifier>>, Acesso em: 2023-10-09.

LAKSHMANAN, M. G. alliappa; GILLARD, R. **Practical Machine Learning for Computer Vision**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2021.

LAKSHMANAN, S. R. V.; MUNN, M. **Machine Learning Design Patterns**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2021.

LAMB, L. R. et al. Predicting ipsilateral recurrence in women treated for ductal carcinoma in situ using machine learning and multivariable logistic regression models. **Clinical Imaging**, v. 92, p. 94–100, 2022. ISSN 0899-7071. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0899707122002315>>.

LAVRUKHIN, E. V. et al. Assessing the fidelity of neural network-based segmentation of soil xct images based on pore-scale modelling of saturated flow properties. **Soil and Tillage Research**, v. 209, p. 104942, 2021. ISSN 0167-1987. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016719872100012X>>.

LI, M. et al. A deep learning convolutional neural network and multi-layer perceptron hybrid fusion model for predicting the mechanical properties of carbon fiber. **Materials Design**, v. 227, p. 111760, 2023. ISSN 0264-1275. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0264127523001752>>.

LI, W. et al. Multi-agent evolution reinforcement learning method for machining parameters optimization based on bootstrap aggregating graph attention network simulated environment. **Journal of Manufacturing Systems**, v. 67, p. 424–438, 2023. ISSN 0278-6125. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0278612523000390>>.

LIU, B. et al. An adaptive parallel feature learning and hybrid feature fusion-based deep learning approach for machining condition monitoring. **IEEE Transactions on Cybernetics**, p. 1–12, 2022.

LIU, Q. et al. Rmt-net: Reject-aware multi-task network for modeling missing-not-at-random data in financial credit scoring. **IEEE Transactions on Knowledge and Data Engineering**, p. 1–1, 2022.

LIU, Y. et al. Building a platform for machine learning operations from open source frameworks. **IFAC-PapersOnLine**, v. 53, n. 5, p. 704–709, 2020. ISSN 2405-8963. 3rd IFAC Workshop on Cyber-Physical Human Systems CPHS 2020. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896321003013>>.

LIU, Y. et al. A discriminative feature representation method based on cascaded attention network with adversarial strategy for speech emotion recognition. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, p. 1–11, 2023.

MACHINE Learning Market Size, Share & Growth Report, 2030. 2030. Disponível em: <<https://www.grandviewresearch.com>>.

- MAGALHÃES, D.; LIMA, R. H.; POZO, A. Creating deep neural networks for text classification tasks using grammar genetic programming. **Applied Soft Computing**, v. 135, p. 110009, 2023. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494623000273>>.
- MALLICK, A. et al. Matchmaker: Data drift mitigation in machine learning for large-scale systems. In: MARCULESCU, D.; CHI, Y.; WU, C. (Ed.). **Proceedings of Machine Learning and Systems**. MLSys Proceedings, 2022. v. 4, p. 77–94. Disponível em: <https://proceedings.mlsys.org/paper_files/paper/2022/file/069a002768bcb31509d4901961f23b3c-Paper.pdf>.
- MANIAS, D. M.; CHOUMAN, A.; SHAMI, A. A Model Drift Detection and Adaptation Framework for 5G Core Networks. **arXiv e-prints**, p. arXiv:2209.06852, ago. 2022.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <<https://doi.org/10.1007/BF02478259>>.
- MEHNI, M. B.; MEHNI, M. B. Reliability analysis with cross-entropy based adaptive markov chain importance sampling and control variates. **Reliability Engineering System Safety**, v. 231, p. 109014, 2023. ISSN 0951-8320. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0951832022006299>>.
- MISHRA, A.; OTAIWI, Z. Devops and software quality: A systematic mapping. **Computer Science Review**, v. 38, p. 100308, 2020. ISSN 1574-0137. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1574013720304081>>.
- MOHAMMADI, T. et al. Unsupervised machine learning with cluster analysis in patients discharged after an acute coronary syndrome: Insights from a 23,270-patient study. **The American Journal of Cardiology**, v. 193, p. 44–51, 2023. ISSN 0002-9149. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S000291492300067X>>.
- MOHAMMED, M. A. et al. A hybrid cancer prediction based on multi-omics data and reinforcement learning state action reward state action (sarsa). **Computers in Biology and Medicine**, v. 154, p. 106617, 2023. ISSN 0010-4825. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0010482523000823>>.
- NANDA, A. K. et al. Multi-layer perceptron's neural network with optimization algorithm for greenhouse gas forecasting systems. **Environmental Challenges**, v. 11, p. 100708, 2023. ISSN 2667-0100. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2667010023000306>>.
- NAYYAR, A.; GADHAVI, L.; ZAMAN, N. Chapter 2 - machine learning in healthcare: review, opportunities and challenges. In: SINGH, K. K. et al. (Ed.). **Machine Learning and the Internet of Medical Things in Healthcare**. Academic Press, 2021. p. 23–45. ISBN 978-0-12-821229-5. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128212295000112>>.
- NGUYEN, V. L. et al. Comparison of multivariate linear regression and a machine learning algorithm developed for prediction of precision warfarin dosing in a korean population. **Journal of Thrombosis and Haemostasis**,

v. 19, n. 7, p. 1676–1686, 2021. ISSN 1538-7836. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1538783622018049>>.

OHUNAKIN, O. S. et al. Conditional monitoring and fault detection of wind turbines based on kolmogorov–smirnov non-parametric test. **Energy Reports**, v. 11, p. 2577–2591, 2024. ISSN 2352-4847. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352484724000817>>.

OLSEN, F. et al. Borough-level covid-19 forecasting in london using deep learning techniques and a novel mse-moran's i loss function. **Results in Physics**, v. 35, p. 105374, 2022. ISSN 2211-3797. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2211379722001450>>.

OSBORNE, A.; DORVILLE, J.; ROMANO, P. Upsampling monte carlo neutron transport simulation tallies using a convolutional neural network. **Energy and AI**, v. 13, p. 100247, 2023. ISSN 2666-5468. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2666546823000198>>.

OSIPOV, C. **MLOps Engineering at Scale**. Shelter Island, NY 11964: Manning Publications Co, 2022.

OZTURK, S. (Ed.). **Convolutional Neural Networks for Medical Image Processing Applications**. 1st edition. ed. [S.l.]: CRC Press, 2023. 274 p. ISBN 9781032104003.

PANG, Y.; WANG, Y. Water spatial distribution in polymer electrolyte membrane fuel cell: Convolutional neural network analysis of neutron radiography. **Energy and AI**, v. 14, p. 100265, 2023. ISSN 2666-5468. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S266654682300037X>>.

PATEL, S.; ALNASER, A. J. A mathematical overview of machine vision. In: **2020 SoutheastCon**. Raleigh, NC, USA: IEEE, 2020. p. 1–6.

PERAFÁN-LÓPEZ, J. C.; SIERRA-PÉREZ, J. An unsupervised pattern recognition methodology based on factor analysis and a genetic-dbscan algorithm to infer operational conditions from strain measurements in structural applications. **Chinese Journal of Aeronautics**, v. 34, n. 2, p. 165–181, 2021. ISSN 1000-9361. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1000936120304775>>.

PINTOR, M. et al. Imagenet-patch: A dataset for benchmarking machine learning robustness against adversarial patches. **Pattern Recognition**, v. 134, p. 109064, 2023. ISSN 0031-3203. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0031320322005441>>.

POOLA, R. G.; PL, L.; Y, S. S. Covid-19 diagnosis: A comprehensive review of pre-trained deep learning models based on feature extraction algorithm. **Results in Engineering**, v. 18, p. 101020, 2023. ISSN 2590-1230. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2590123023001470>>.

POPOV, K. et al. Data-driven track geometry fault localisation using unsupervised machine learning. **Construction and Building Materials**, v. 377, p. 131141, 2023. ISSN 0950-0618. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095006182300853X>>.

- PORWIK, P.; DADZIE, B. M. Detection of data drift in a two-dimensional stream using the kolmogorov-smirnov test. **Procedia Computer Science**, v. 207, p. 168–175, 2022. ISSN 1877-0509. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S187705092200922X>>.
- POVENZ, K. **St Joseph Lighthouse**. 2012. Disponível em: <<https://www.flickr.com/photos/kevinpovenz/8140373638>>, Acesso em: 2023-10-09.
- PRAKASH, N. N. et al. A densenet cnn-based liver lesion prediction and classification for future medical diagnosis. **Scientific African**, v. 20, p. e01629, 2023. ISSN 2468-2276. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2468227623000868>>.
- PRAPAS, I. et al. Continuous training and deployment of deep learning models. **Datenbank-Spektrum**, v. 21, n. 3, p. 203–212, Nov 2021. ISSN 1610-1995. Disponível em: <<https://doi.org/10.1007/s13222-021-00386-8>>.
- PWC'S Global Artificial Intelligence Study. 2030. Disponível em: <<https://www.pwc.com>>.
- PYTORCH, F. **PyTorch Foundation**. 2023. Disponível em: <<https://pytorch.org/>>, Acesso em: 2023-10-09.
- RAHMAN, T.; ISLAM, M. S. Mri brain tumor detection and classification using parallel deep convolutional neural networks. **Measurement: Sensors**, v. 26, p. 100694, 2023. ISSN 2665-9174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2665917423000302>>.
- RAJPURKAR, P. et al. **MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs**. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1712.06957>>.
- RAJPUT, D. S. et al. Fault prediction using fuzzy convolution neural network on iot environment with heterogeneous sensing data fusion. **Measurement: Sensors**, v. 26, p. 100701, 2023. ISSN 2665-9174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2665917423000375>>.
- RAO, M. V. V. et al. Monitor the strength status of buildings using hybrid machine learning technique. **IEEE Access**, p. 1–1, 2023.
- REZAALIPOUR, M.; FURIA, C. A. An annotation-based approach for finding bugs in neural network programs. **Journal of Systems and Software**, v. 201, p. 111669, 2023. ISSN 0164-1212. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016412122300064X>>.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65 6, p. 386–408, 1958.
- SADEQ, J. M.; QADIR, B. A.; ABBAS, H. H. Cars logo recognition by using of backpropagation neural networks. **Measurement: Sensors**, v. 26, p. 100702, 2023. ISSN 2665-9174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2665917423000387>>.

SADIK, R. et al. An in-depth analysis of convolutional neural network architectures with transfer learning for skin disease diagnosis. **Healthcare Analytics**, v. 3, p. 100143, 2023. ISSN 2772-4425. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2772442523000102>>.

SALAMA, W. M.; ALY, M. H.; ABOUELSEUD, Y. Deep learning-based spam image filtering. **Alexandria Engineering Journal**, v. 68, p. 461–468, 2023. ISSN 1110-0168. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1110016823000741>>.

SARAN, N.; KESSWANI, N. A comparative study of supervised machine learning classifiers for intrusion detection in internet of things. **Procedia Computer Science**, v. 218, p. 2049–2057, 2023. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050923001813>>.

SAYED, Y. A. et al. Machine-learning-based models versus design-oriented models for predicting the axial compressive load of frp-confined rectangular rc columns. **Engineering Structures**, v. 285, p. 116030, 2023. ISSN 0141-0296. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0141029623004443>>.

SCHMITT, J. et al. Predictive model-based quality inspection using machine learning and edge cloud computing. **Advanced Engineering Informatics**, v. 45, p. 101101, 2020. ISSN 1474-0346. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1474034620300707>>.

SHAHIN, M.; BABAR, M. A.; ZHU, L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. **IEEE Access**, PP, 03 2017.

SHAKER, A.; HÜLLERMEIER, E. Handling concept drift: Importance, challenges and solutions. **PLoS ONE**, v. 10, n. 7, p. e0131955, 2015.

SHARMA, A.; SHARMA, M.; Kr. Dwivedi, R. Exploratory data analysis and deception detection in news articles on social media using machine learning classifiers. **Ain Shams Engineering Journal**, p. 102166, 2023. ISSN 2090-4479. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2090447923000552>>.

SILVA, L. C. et al. Benchmarking machine learning solutions in production. In: **2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)**. Miami, FL, USA: IEEE, 2020. p. 626–633.

STEFANIA, G.-C. et al. Credit risk scoring model based on the discriminant analysis technique. **Procedia Computer Science**, v. 220, p. 928–933, 2023. ISSN 1877-0509. The 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT 2022) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050923006622>>.

STIGT, M. V. et al. High performance clean versus artifact dry electrode eeg data classification using convolutional neural network transfer learning.

Clinical Neurophysiology Practice, 2023. ISSN 2467-981X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2467981X23000094>.

SUN, Z. et al. An artificial neural network model for accurate and efficient optical property mapping from spatial-frequency domain images. **Computers and Electronics in Agriculture**, v. 188, p. 106340, 2021. ISSN 0168-1699. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169921003574>.

SUÁREZ-CETRULO, A. L.; QUINTANA, D.; CERVANTES, A. A survey on machine learning for recurring concept drifting data streams. **Expert Systems with Applications**, v. 213, p. 118934, 2023. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417422019522>.

SWAPNAKE. **Convolution Neural Networks**. 2023. Disponível em: <https://developersbreach.com/convolution-neural-network-deep-learning/>, Acesso em: 2023-10-09.

SZELISKI, R. **Computer Vision: Algorithms and Applications (Texts in Computer Science)**. 2nd ed. 2022. ed. Springer Nature Switzerland: Springer, 2022. ISBN 3030343715; 9783030343712.

TALUKDAR, J.; GOGOI, D. K.; SINGH, T. P. A comparative assessment of most widely used machine learning classifiers for analyzing and classifying autism spectrum disorder in toddlers and adolescents. **Healthcare Analytics**, p. 100178, 2023. ISSN 2772-4425. Disponível em: <https://www.sciencedirect.com/science/article/pii/S277244252300045X>.

TAMBURRI, D. A. Sustainable mlops: Trends and challenges. In: **2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)**. Timisoara, Romania: IEEE, 2020. p. 17–23.

TAO, H. et al. Machine learning algorithms for high-resolution prediction of spatiotemporal distribution of air pollution from meteorological and soil parameters. **Environment International**, p. 107931, 2023. ISSN 0160-4120. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0160412023002040>.

TEOH, T. T. **Convolutional Neural Networks for Medical Applications**. 1. ed. Springer Singapore, 2023. XI, 96 p. (SpringerBriefs in Computer Science). ISBN 978-981-19-8814-1. Disponível em: <https://doi.org/10.1007/978-981-19-8814-1>.

THONGLEK, K. et al. Automated quantization and retraining for neural network models without labeled data. **IEEE Access**, v. 10, p. 73818–73834, 2022.

T.K., B.; ANNAVARAPU, C. S. R.; BABLANI, A. Machine learning algorithms for social media analysis: A survey. **Computer Science Review**, v. 40, p. 100395, 2021. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013721000356>.

TREVEIL, M.; TEAM the D. **Introducing MLOps**. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2020.

TRIOLA, M. **Introdução à estatística**. 12. ed. Rio de Janeiro: LTC Editora, 2017.

- VIVERIT, L. et al. Application of machine learning to cluster hotel booking curves for hotel demand forecasting. **International Journal of Hospitality Management**, v. 111, p. 103455, 2023. ISSN 0278-4319. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0278431923000294>>.
- WANG, T. et al. Hybrid machine learning approach for evapotranspiration estimation of fruit tree in agricultural cyber-physical systems. **IEEE Transactions on Cybernetics**, p. 1–15, 2022.
- WANG, W.; QIAO, L.; LIN, B. Tabular machine learning using conjunctive threshold neural networks. **Machine Learning with Applications**, v. 10, p. 100429, 2022. ISSN 2666-8270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2666827022001049>>.
- WANG, Y.; JIA, B.; XIAN, C. Machine learning and unet++ based microfracture evaluation from ct images. **Geoenergy Science and Engineering**, v. 226, p. 211726, 2023. ISSN 2949-8910. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2949891023003135>>.
- WATSON, E. M.; CUTTING, D. Engagement contexts of software engineering education projects. In: **2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEIE)**. Coimbra, Portugal: IEEE, 2022. p. 1–6.
- WEI, X. et al. Wasserstein distance-based expansion planning for integrated energy system considering hydrogen fuel cell vehicles. **Energy**, v. 272, p. 127011, 2023. ISSN 0360-5442. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S036054422300405X>>.
- WORLDWIDE AI spending expected to reach 110billionin2024.2020.Disponívelem : <>.
- WU, B.; ZHANG, L.; ZHAO, Y. Feature selection via cramer’s v-test discretization for remote-sensing image classification. **IEEE Transactions on Geoscience and Remote Sensing**, v. 52, n. 5, p. 2593–2606, 2014.
- XIONG, D. et al. Deep learning for emg-based human-machine interaction: A review. **IEEE/CAA Journal of Automatica Sinica**, v. 8, n. 3, p. 512–533, 2021.
- XUE, Y. et al. Template-guided frequency attention and adaptive cross-entropy loss for uav visual tracking. **Chinese Journal of Aeronautics**, 2023. ISSN 1000-9361. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1000936123001097>>.
- YANG, L. et al. Rema-net: An efficient multi-attention convolutional neural network for rapid skin lesion segmentation. **Computers in Biology and Medicine**, v. 159, p. 106952, 2023. ISSN 0010-4825. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0010482523004171>>.

YANG, S. Research on ink speed recognition method of hyperspectral imaging ink pad based on machine learning. In: **2022 International Conference on 3D Immersion, Interaction and Multi-sensory Experiences (ICDIIME)**. Madrid, Spain: IEEE, 2022. p. 22–26.

YEUNG, M. et al. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. **Computerized Medical Imaging and Graphics**, v. 95, p. 102026, 2022. ISSN 0895-6111. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0895611121001750>>.

YOUNG, Z.; STEELE, R. Empirical evaluation of performance degradation of machine learning-based predictive models – a case study in healthcare information systems. **International Journal of Information Management Data Insights**, v. 2, n. 1, p. 100070, 2022. ISSN 2667-0968. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2667096822000143>>.

YU, X. et al. Faulty feeder identification in resonant grounding distribution networks based on deep learning and transfer learning. **CSEE Journal of Power and Energy Systems**, p. 1–10, 2021.

ZELAYA, C. V. G. Towards explaining the effects of data preprocessing on machine learning. In: **2019 IEEE 35th International Conference on Data Engineering (ICDE)**. Macao, China: IEEE, 2019. p. 2086–2090.

ZENISEK, J.; HOLZINGER, F.; AFFENZELLER, M. Machine learning based concept drift detection for predictive maintenance. **Computers Industrial Engineering**, v. 137, p. 106031, 2019. ISSN 0360-8352. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360835219304905>>.

ZHANG, L.; MA, J. A spatiotemporal graph wavelet neural network for traffic flow prediction. **Journal of Information and Intelligence**, 2023. ISSN 2949-7159. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2949715923000021>>.

ZHANG, M. et al. Sequential learning for ingredient recognition from images. **IEEE Transactions on Circuits and Systems for Video Technology**, p. 1–1, 2022.

ZHANG, N. et al. Discriminative few shot learning of facial dynamics in interview videos for autism trait classification. **IEEE Transactions on Affective Computing**, p. 1–1, 2022.

ZHANG, W. et al. Explaining of prediction accuracy on phase selection of amorphous alloys and high entropy alloys using support vector machines in machine learning. **Materials**

Today Communications, v. 35, p. 105694, 2023. ISSN 2352-4928. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352492823003859>.

ZHENG, A. **Evaluating Machine Learning Models**. [S.l.]: O'Reilly Media.

ZHOU, B. et al. Stochastic generalized kolmogorov systems with small diffusion: I. explicit approximations for invariant probability density function. **Journal of Differential Equations**, v. 382, p. 141–210, 2024. ISSN 0022-0396. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022039623007131>.

ZIEGENBEIN, A.; METTERNICH, J. Data-based quality analysis in machining production: Influence of data pre-processing on the results of machine learning models. **Procedia CIRP**, v. 104, p. 869–874, 2021. ISSN 2212-8271. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827121010441>.

ZLIOBAITE. Combining similarity in time and space for training set formation under concept drift. **Intelligent Data Analysis**, IOS Press, v. 15, p. 589–611, 2011. ISSN 1571-4128. 4. Disponível em: <https://doi.org/10.3233/IDA-2011-0484>.

ÜNAL, H. T.; BASCIFTCI, F. Neural logic circuits: An evolutionary neural architecture that can learn and generalize. **Knowledge-Based Systems**, v. 265, p. 110379, 2023. ISSN 0950-7051. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705123001296>.