UNIVERSIDADE FEDERAL DO PARANÁ

RENAN DE PAULA ROSA

MEMORIAL DE PROJETOS: DEEP LEARNING - INOVAÇÕES, APLICAÇÕES E DESAFIOS

CURITIBA

2025

RENAN DE PAULA ROSA

MEMORIAL DE PROJETOS: DEEP LEARNING - INOVAÇÕES, APLICAÇÕES E DESAFIOS

Memorial de Projetos apresentado ao curso de Especialização em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientadora: Profa. Dra. Rafaela Mantovani Fontana

CURITIBA
2025

MINISTÉRIO DA EDUCAÇÃO
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PÓS-GRADUAÇÃO
CURSO DE PÓS-GRADUAÇÃO INTELIGÊNCIA ARTIFICIAL APLICADA - 40001016399E1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação Inteligência Artificial Aplicada da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **RENAN DE PAULA ROSA**, intitulada: **MEMORIAL DE PROJETOS: DEEP LEARNING - INOVAÇÕES, APLICAÇÕES E DESAFIOS**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

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

Curitiba, 03 de Setembro de 2025.

RAFAELA MANTOVANI FONTANA
Presidente da Banca Examinadora

RAZER ANTHONY NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Este parecer técnico apresenta uma revisão sobre *deep learning*, subdisciplina da inteligência artificial baseada em redes neurais multicamadas. O estudo explora sua evolução desde os primeiros conceitos na década de 1940 até sua recente aceleração impulsionada por grandes conjuntos de dados, avanços computacionais e inovações algorítmicas como *backpropagation*, normalização em lote e *dropout*. Além disso, analisa as principais arquiteturas (CNNs – redes neurais convolucionais, RNNs – redes neurais recorrentes, *Transformers* e GANs – redes adversárias generativas) e suas aplicações transformadoras em setores como saúde (diagnóstico por imagem, descoberta de medicamentos), indústria (manutenção preditiva), transportes (veículos autônomos), processamento de linguagem natural e sustentabilidade ambiental. Quatro desafios fundamentais são identificados: limitada interpretabilidade dos modelos "caixa-preta", dependência de grandes volumes de dados rotulados, problemas de robustez/generalização e questões éticas relacionadas a viés. Os resultados evidenciam o profundo impacto do *deep learning* na sociedade contemporânea, enquanto apontam para a necessidade de pesquisas futuras direcionadas à criação de sistemas mais transparentes, eficientes, robustos e equitativos, que equilibrem avanço tecnológico e responsabilidade ética.

Palavras-chave: Aprendizado Profundo; Redes Neurais Artificiais; Inteligência Artificial; Aprendizado de Máquina; Transformadores.

ABSTRACT

This technical report presents a review of deep learning, a subdiscipline of artificial intelligence based on multilayer neural networks. It explores its evolution from the earliest concepts in the 1940s to its recent acceleration driven by large datasets, computational advances, and algorithmic innovations such as backpropagation, batch normalization, and dropout. The main architectures (CNNs – convolutional neural networks, RNNs – recurrent neural networks, Transformers, and GANs – generative adversarial networks) and their transformative applications are analyzed in sectors such as healthcare (image-based diagnosis, drug discovery), industry (predictive maintenance), transportation (autonomous vehicles), natural language processing, and environmental sustainability. Four key challenges are identified: limited interpretability of “black-box” models, dependence on large volumes of labeled data, robustness/generalization issues, and ethical concerns related to bias. The results highlight the profound impact of deep learning on contemporary society while pointing to the need for future research aimed at creating more transparent, efficient, robust, and fair systems that balance technological advancement with ethical responsibility.

Keywords: Deep Learning; Artificial Neural Networks; Artificial Intelligence, Machine Learning; Transformers.

SUMÁRIO

1 PARECER TÉCNICO

O *deep learning* é um campo revolucionário da inteligência artificial que permite a computadores interpretar dados complexos e executar tarefas antes restritas à inteligência humana (Lecun; Bengio; Hinton, 2015). Baseado em redes neurais artificiais com múltiplas camadas, extrai representações de alto nível diretamente dos dados brutos, dispensando a extração manual de características (Goodfellow; Bengio; Courville, 2016).

Seu desenvolvimento, iniciado nos anos 1940, alternou entre avanços e *invernos da IA*, acelerado na última década com grandes conjuntos de dados, maior poder computacional e inovações como *backpropagation*, normalização em lote e *dropout*. Inspiradas no cérebro humano, as redes profundas representam o mundo como hierarquias de conceitos, treinadas por ajuste iterativo de pesos para minimizar erros (Goodfellow; Bengio; Courville, 2016).

As principais arquiteturas incluem:

- **Redes Neurais Convolucionais (CNNs - *Convolutional Neural Networks*):** especializadas em imagens: AlexNet (Krizhevsky; Sutskever; Hinton, 2012), VGGNet - *Visual Geometry Group Network* (Simonyan; Zisserman, 2014), ResNet - *Residual Network* (He; Zhang; Ren; Sun, 2016), EfficientNet (Tan; Le, 2019);
- **Redes Neurais Recorrentes (RNNs - *Recurrent Neural Networks*):** para sequências, com variantes LSTM - *Long Short-Term Memory* (Hochreiter; Schmidhuber, 1997) e GRU - *Gated Recurrent Unit* (Cho *et al*., 2014);
- **Transformers:** revolucionaram o NLP (*Natural Language Processing* ou Processamento de Linguagem Natural) com atenção (BERT - *Bidirectional Encoder Representations from Transformers* (Devlin *et al*., 2019), GPT - *Generative Pre-trained Transformer* (Radford *et al*., 2019)) (Vaswani *et al*., 2017);
- **Redes Generativas Adversariais (GANs - *Generative Adversarial Networks*):** geram conteúdo sintético realista (Goodfellow *et al.*, 2014; StyleGAN - *Style-based Generative Adversarial Network* (Karras; Laine; Aila, 2019), CycleGAN - *Cycle-Consistent Adversarial Networks* (Zhu *et al.*, 2017)).

Essas arquiteturas não apenas diferem em estrutura e propósito, mas também se complementam em soluções híbridas. Por exemplo, combinações de CNNs e RNNs podem processar vídeo integrando análise espacial e temporal; Transformers já são aplicados além do texto, como em visão computacional; e GANs podem gerar dados sintéticos para treinar outros modelos, ampliando a aplicabilidade e superando limitações de dados.

O impacto do *deep learning* se manifesta em múltiplos setores, transformando processos, elevando a precisão de previsões e permitindo automações antes inviáveis. Em diferentes áreas, essas tecnologias impulsionam inovações capazes de redefinir cadeias produtivas e serviços.

O impacto se estende por setores como:

- **Saúde**: descoberta de medicamentos e previsão de estruturas proteicas (Goodfellow; Bengio; Courville, 2016);
- **Indústria**: inspeção visual e controle de qualidade (Krizhevsky; Sutskever; Hinton, 2012; Simonyan; Zisserman, 2014), manutenção preditiva (Hochreiter; Schmidhuber, 1997);
- **Veículos Autônomos**: fusão de sensores, navegação em tempo real, simulações (Goodfellow; Bengio; Courville, 2016);
- **Sustentabilidade**: previsão climática, otimização energética, monitoramento ambiental (Goodfellow; Bengio; Courville, 2016).

Apesar do avanço de modelos como os Transformers (Vaswani *et al*., 2017), a IA enfrenta desafios como a falta de transparência (caixa-preta), alta demanda computacional e vieses. Para solucionar isso, a pesquisa foca em **IA Explicável (*Explainable AI*)**, que busca tornar os modelos compreensíveis e auditáveis (Adadi; Berrada, 2018). O objetivo é desenvolver uma IA mais robusta, justa e confiável para aplicações críticas.

Esses desafios representam barreiras críticas para a adoção massiva e responsável do *deep learning*. Sem interpretabilidade, a confiança em aplicações de alto risco fica comprometida; a dependência de dados e recursos computacionais limita o acesso a países e organizações com menos infraestrutura; falhas de robustez podem levar a erros catastróficos em cenários reais; e vieses não tratados podem amplificar desigualdades sociais. Enfrentá-los exige abordagens multidisciplinares, combinando avanços técnicos com regulamentações, práticas éticas e colaboração global.

REFERÊNCIAS

ADADI, Amina; BERRADA, Mohammed. **Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)**. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8466590. Acesso em: 20 ago. 2025.

CHO, K. et al. **Learning phrase representations using rnn encoder-decoder for statistical machine translation**, 2014. Disponível em: https://arxiv.org/abs/1406.1078. Acesso em: 01 mai. 2025.

DEVLIN, J. et al. **BERT: Pre-training of deep bidirectional transformers for language understanding**, 2019. Disponível em: https://aclanthology.org/N19-1423.pdf. Acesso em: 01 mai. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning.** Disponível em: https://synapse.koreamed.org/pdf/10.4258/hir.2016.22.4.351. Acesso em: 01 mai. 2025.

GOODFELLOW, I. et al. **Generative adversarial nets**, 2014. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf. Acesso em: 01 mai. 2025.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition**, 2016. Disponível em: https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf. Acesso em: 01 mai. 2025.

HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory**. Disponível em: https://ieeexplore.ieee.org/abstract/document/6795963. Acesso em: 01 mai. 2025.

KARRAS, T.; LAINE, S.; AILA, T. **A style-based generator architecture for generative adversarial networks**, 2019. Disponível em: https://openaccess.thecvf.com/content_CVPR_2019/papers/Karras_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.pdf. Acesso em: 01 mai. 2025.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **Imagenet classification with deep convolutional neural networks**, 2012. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. Acesso em: 01 mai. 2025.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning.** Nature, v. 521, n. 7553, p. 436-444, 2015. Disponível em: https://www.nature.com/articles/nature14539. Acesso em: 01 mai. 2025.

RADFORD, A. et al. **Language models are unsupervised multitask learners**, 2019. Disponível em: https://storage.prod.researchhub.com/uploads/papers/2020/06/01/language-models.pdf. Acesso em: 01 mai. 2025.

SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition**. 2014. Disponível em: https://arxiv.org/pdf/1409.1556. Acesso em: 01 mai. 2025.

TAN, M.; LE, Q. V. **Efficientnet: rethinking model scaling for convolutional neural networks,** 2019. Disponível em: https://proceedings.mlr.press/v97/tan19a/tan19a.pdf. Acesso em: 01 mai. 2025.

VASWANI, A. et al. **Attention is all you need. In: Advances in Neural Information Processing Systems**, v.30, 2017. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. Acesso em: 01 mai. 2025.

ZHU, J. Y. et al. **Unpaired image-to-image translation using cycle-consistent adversarial networks**, 2017. Disponível em: https://openaccess.thecvf.com/content_ICCV_2017/papers/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.pdf. Acesso em: 01 mai. 2025.

APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

A – ENUNCIADO

1 ChatGPT

a) **(6,25 pontos)** Pergunte ao ChatGPT o que é Inteligência Artificial e cole aqui o resultado.
b) **(6,25 pontos)** Dada essa resposta do ChatGPT, classifique usando as 4 abordagens vistas em sala. Explique o porquê.
c) **(6,25 pontos)** Pesquise sobre o funcionamento do ChatGPT (sem perguntar ao próprio ChatGPT) e escreva um texto contendo no máximo 5 parágrafos. Cite as referências.
d) **(6,25 pontos)** Entendendo o que é o ChatGPT, classifique o próprio ChatGPT usando as 4 abordagens vistas em sala. Explique o porquê.

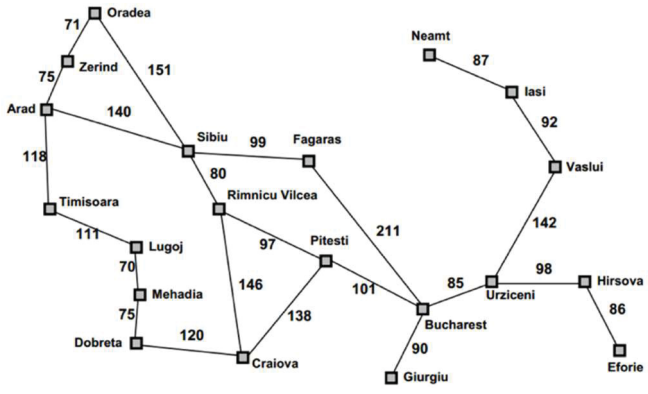
2 Busca Heurística

Realize uma busca utilizando o algoritmo A* para encontrar o melhor caminho para chegar a **Bucharest** partindo de **Lugoj**. Construa a árvore de busca criada pela execução do algoritmo apresentando os valores de *f(n)*, *g(n)* e *h(n)* para cada nó. Utilize a heurística de distância em linha reta, que pode ser observada na tabela abaixo.

Essa tarefa pode ser feita em uma **ferramenta de desenho**, ou até mesmo no **papel**, desde que seja digitalizada (foto) e convertida para PDF.

a) **(25 pontos)** Apresente a árvore final, contendo os valores, da mesma forma que foi apresentado na disciplina e nas práticas. Use o formato de árvore, não será permitido um formato em blocos, planilha, ou qualquer outra representação.

NÃO É NECESSÁRIO IMPLEMENTAR O ALGORITMO.



Arad	366	Mehadia	241
Bucareste	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Figura 3.22 Valores de *hDLR* — distâncias em linha reta para Bucareste.

3 Lógica

Verificar se o argumento lógico é válido.

Se as uvas caem, então a raposa as come
Se a raposa as come, então estão maduras
As uvas estão verdes ou caem

Logo

A raposa come as uvas se e somente se as uvas caem

Deve ser apresentada uma prova, no mesmo formato mostrado nos conteúdos de aula e nas práticas.

Dicas:

1. Transformar as afirmações para lógica:

p: as uvas caem
q: a raposa come as uvas
r: as uvas estão maduras

2. Transformar as três primeiras sentenças para formar a base de conhecimento

R1: $p \rightarrow q$
R2: $q \rightarrow r$
R3: $\neg r \vee p$

3. Aplicar equivalências e regras de inferência para se obter o resultado esperado. Isto é, com essas três primeiras sentenças devemos derivar $q \leftrightarrow p$. Cuidado com a ordem em que as fórmulas são geradas.

Equivalência Implicação: $(\alpha \rightarrow \beta)$ equivale a $(\neg\alpha \vee \beta)$

Silogismo Hipotético: $\alpha \rightarrow \beta, \beta \rightarrow \gamma \vdash \alpha \rightarrow \gamma$

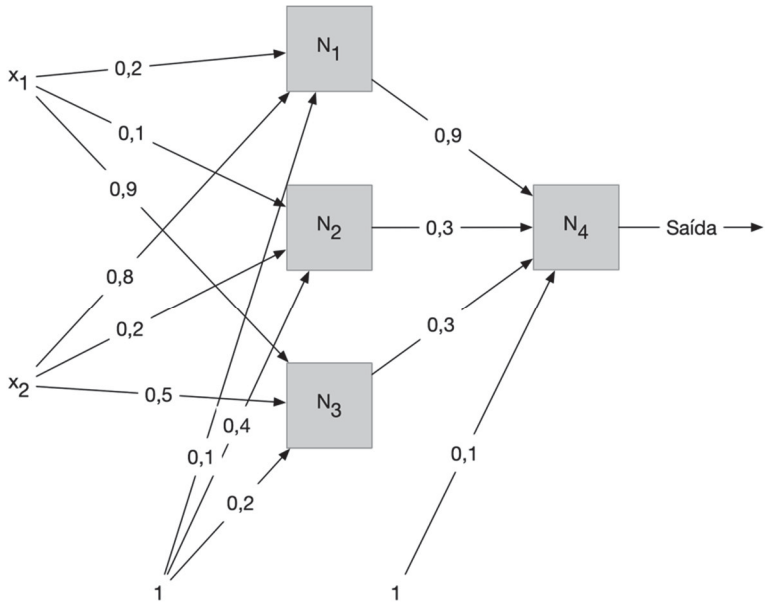
Conjunção: $\alpha, \beta \vdash \alpha \wedge \beta$

Equivalência Bicondicional: $(\alpha \leftrightarrow \beta)$ equivale a $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

a) **(25 pontos)** Deve-se mostrar todos os passos e regras aplicadas, **no mesmo formato apresentado nas aulas e nas práticas**. As equivalências e regras necessárias estão descritas acima e no material.

4 Redes Neurais Artificiais

Seja a RNA da figura abaixo.



Os neurônios N_1, N_2 e N_3 possuem função de ativação linear. Já N_4 possui função de ativação tangente hiperbólica (pesquise a fórmula e aplique).

Dada a entrada x_1=-3, x_2=1, dê:

a) **(6,25 pontos)** Valor de saída do neurônio N1
b) **(6,25 pontos)** Valor de saída do neurônio N2
c) **(6,25 pontos)** Valor de saída do neurônio N3
d) **(6,25 pontos)** Valor de saída da rede como um todo

B – RESOLUÇÃO

1)

A) Inteligência Artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas e máquinas capazes de realizar tarefas que normalmente exigiriam inteligência humana. Essas tarefas incluem reconhecimento de fala, compreensão de linguagem natural, tomada de decisão, aprendizado, visão computacional, entre outras. A IA busca criar algoritmos e modelos que possam simular certos aspectos da inteligência humana, como raciocínio, resolução de problemas, aprendizado e adaptação ao ambiente. Essa área abrange uma variedade de técnicas, incluindo aprendizado de máquina, redes neurais, processamento de linguagem natural, visão computacional e muito mais. O objetivo final da inteligência artificial é criar sistemas que possam realizar tarefas de forma autônoma e inteligente, sem intervenção humana direta.

B) Agir como humanos: A resposta dada pelo ChatGPT se enquadra bastante na abordagem “Agir como humanos”, ao citar o desenvolvimento de máquinas que executam funções que exigem inteligência humana, semelhante a definição de Kurzweil (1990). Também se enquadra nesta abordagem ao mencionar a compreensão de linguagem natural e a aprendizagem.
Pensar como os humanos: Apesar de não falar em implementar fielmente o processo de pensamento, a definição do ChatGPT se enquadra parcialmente na abordagem de “pensar como humanos” quando fala em “criar algoritmos e modelos que possam simular certos aspectos da inteligência humana, como raciocínio, resolução de problemas, aprendizado e adaptação ao ambiente”, pois está alinhada com a definição de Bellman (1978), que aborda a IA como a automatização destes tipo de atividades. Pensar racionalmente: A definição do ChatGPT não se enquadra nesta abordagem, pois não menciona modelar o processo de raciocínio ou a busca de um resultado logicamente correto. Agir racionalmente: Por não mencionar a busca do melhor resultado possível, a definição dada pelo ChatGPT não se enquadra nesta abordagem.

C) O ChatGPT é um modelo de linguagem de grande porte (LLM) desenvolvido pela OpenAI, treinado em um enorme conjunto de dados de texto e código. Essa ferramenta é capaz de gerar respostas textuais complexas e coerentes a partir de uma ampla gama de prompts e perguntas, abrindo um leque de possibilidades para interação humano-computador. Ele analisa a entrada feita pelo usuário, com base nessa informação ele ativa diferentes modelos neurais para lidar com diferentes tipos de problema. A forma como o ChatGPT gera as respostas é, de certa forma, prevendo as próximas palavras a serem escritas. Através dos dados de treinamento, ele identificou padrões nos textos produzidos por humanos, e se tornou capaz de produzir novos textos similares. Com base no prompt do usuário (e no próprio texto que vai sendo gerado), o ChatGPT busca em seu “conhecimento” a direção para a qual o texto deve continuar. Referências: ChatGPT: o que é e como usar? Veja o guia completo do chatbot da OpenAI: https://www.techtudo.com.br/guia/2023/03/chatgpt-o-que-e-e-como-usar-veja-o-guia-co mpleto-do-chatbot-da-openai-edsoftwares.ghtml What Is ChatGPT Doing … and

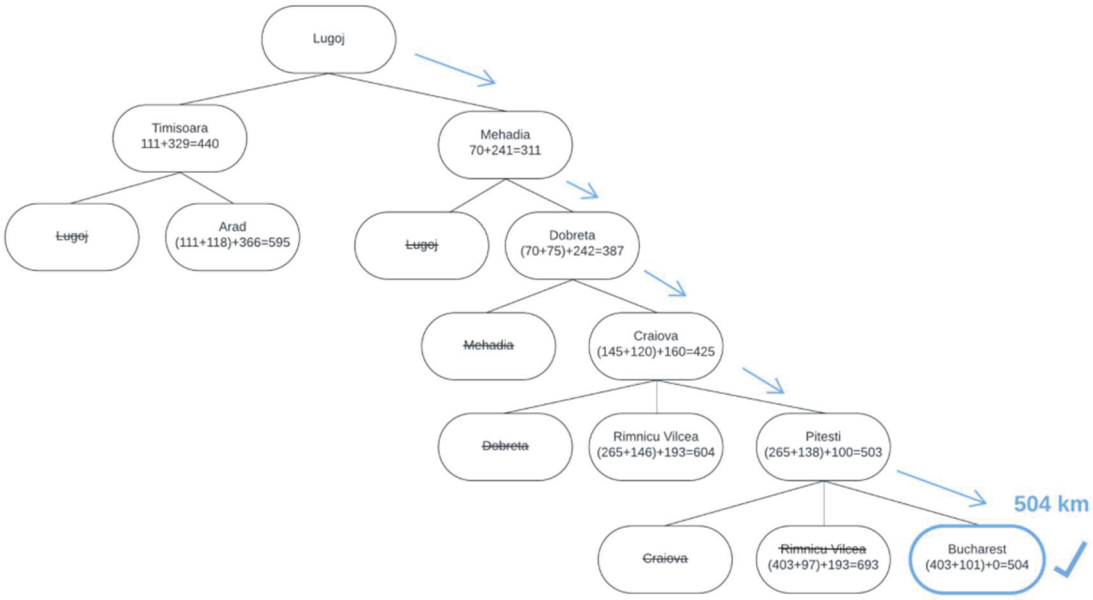
Why Does It Work?: https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

D) Pensar como os humanos: Não se enquadra nesta abordagem, por não replicar processos cognitivos reais. Pensar racionalmente: Utiliza algoritmos e modelos estatísticos para gerar respostas lógicas e coerentes com base em seu treinamento. Agir como os humanos: Destaca-se aqui, gerando interações textuais indistinguíveis das humanas em muitos contextos. Agir racionalmente: Como esta abordagem é mais ampla, e engloba técnicas das outras abordagens, o ChatGPT se enquadra nela, pois além da geração de textos bastante semelhantes aos de humanos e da utilização de modelos para gerar respostas lógicas, ele também busca maximizar relevância e utilidade das respostas, seguindo princípios de otimização.

2)

A)

FIGURA 1 – ÁRVORE DE DECISÃO



FONTE: O autor (2025)

3)

A) uvas caem: p
raposa as come: q
uvas maduras: r
Buscar: $q \leftrightarrow p$ (A raposa come as uvas se e somente se as uvas caem)
P1: $p \rightarrow q$
P2: $q \rightarrow r$

P3: $\neg r \vee p$
P4: $r \rightarrow p$ - Equivalência Implicação - P3
P5: $q \rightarrow p$ - Silogismo Hipotético - P2, P4
P6: $(q \rightarrow p) \wedge (p \rightarrow q)$ - Conjunção - P5, P1
P7: $q \leftrightarrow p$ - Equivalência Bicondicional - P6

4)

A) N1 Entrada: (1 × 0, 1) + (− 3 × 0, 2) + (1 × 0, 8) = 0, 3 N1 Saída: 0, 3
B) N2 Entrada: (1 × 0, 4) + (− 3 × 0, 1) + (1 × 0, 2) = 0, 3 N2 Saída: 0, 3
C) N3 Entrada: (1 × 0, 2) + (− 3 × 0, 9) + (1 × 0, 5) =− 2 N3 Saída: − 2
D) N4 Entrada: (1 × 0, 1) + (0, 3 × 0, 9) + (0, 3 × 0, 3) + (− 2 × 0, 3) =− 0, 14 N4 Saída: $tanh(-0,14) = -0,1391$

APÊNDICE 2 – LINGUAGEM DE PROGRAMAÇÃO APLICADA

A – ENUNCIADO

Nome da base de dados do exercício: *precos_carros_brasil.csv*

Informações sobre a base de dados:

Dados dos preços médios dos carros brasileiros, das mais diversas marcas, no ano de 2021, de acordo com dados extraídos da tabela FIPE (Fundação Instituto de Pesquisas Econômicas). A base original foi extraída do site Kaggle (Acesse aqui a base original). A mesma foi adaptada para ser utilizada no presente exercício.

Observação: As variáveis *fuel*, *gear* e *engine_size* foram extraídas dos valores da coluna *model*, pois na base de dados original não há coluna dedicada a esses valores. Como alguns valores do modelo não contêm as informações do tamanho do motor, este conjunto de dados não contém todos os dados originais da tabela FIPE.

Metadados:

Nome do campo	Descrição
year_of_reference	O preço médio corresponde a um mês de ano de referência
month_of_reference	O preço médio corresponde a um mês de referência, ou seja, a FIPE atualiza sua tabela mensalmente
fipe_code	Código único da FIPE
authentication	Código de autenticação único para consulta no site da FIPE
brand	Marca do carro
model	Modelo do carro
fuel	Tipo de combustível do carro
gear	Tipo de engrenagem do carro
engine_size	Tamanho do motor em centímetros cúbicos
year_model	Ano do modelo do carro. Pode não corresponder ao ano de fabricação
avg_price	Preço médio do carro, em reais

Atenção: ao fazer o download da base de dados, selecione o formato .csv. É o formato que será considerado correto na resolução do exercício.

1 Análise Exploratória dos dados

A partir da base de dados **precos_carros_brasil.csv**, execute as seguintes tarefas:

a. Carregue a base de dados media_precos_carros_brasil.csv
b. Verifique se há valores faltantes nos dados. Caso haja, escolha uma tratativa para resolver o problema de valores faltantes
c. Verifique se há dados duplicados nos dados
d. Crie duas categorias, para separar colunas numéricas e categóricas. Imprima o resumo de informações das variáveis numéricas e categóricas (estatística descritiva dos dados)
e. Imprima a contagem de valores por modelo (model) e marca do carro (brand)
f. Dê um breve explicação (máximo de quatro linhas) sobre os principais resultados encontrados na Análise Exploratória dos dados

2 Visualização dos dados

A partir da base de dados **precos_carros_brasil.csv,** execute as seguintes tarefas:

a. Gere um gráfico da distribuição da quantidade de carros por marca
b. Gere um gráfico da distribuição da quantidade de carros por tipo de engrenagem do carro
c. Gere um gráfico da evolução da média de preço dos carros ao longo dos meses de 2022 (variável de tempo no eixo X)
d. Gere um gráfico da distribuição da média de preço dos carros por marca e tipo de engrenagem
e. Dê uma breve explicação (máximo de quatro linhas) sobre os resultados gerados no item d
f. Gere um gráfico da distribuição da média de preço dos carros por marca e tipo de combustível
g. Dê uma breve explicação (máximo de quatro linhas) sobre os resultados gerados no item f

3 Aplicação de modelos de machine learning para prever o preço médio dos carros

A partir da base de dados **precos_carros_brasil.csv**, execute as seguintes tarefas:

a. Escolha as variáveis **numéricas** (modelos de Regressão) para serem as variáveis independentes do modelo.A variável target é **avg_price. Observação:** caso julgue necessário, faça a transformação de variáveis categóricas em variáveis numéricas para inputar no modelo. Indique **quais variáveis** foram transformadas e **como** foram transformadas
b. Crie partições contendo 75% dos dados para treino e 25% para teste
c. Treine modelos RandomForest (biblioteca RandomForestRegressor) e XGBoost (biblioteca XGBRegressor) para predição dos preços dos carros. **Observação**: caso julgue necessário, mude os parâmetros dos modelos e rode novos modelos. Indique quais parâmetros foram inputados e indique o treinamento de cada modelo
d. Grave os valores preditos em variáveis criadas
e. Realize a análise de importância das variáveis para estimar a variável target, **para cada modelo treinado**
f. Dê uma breve explicação (máximo de quatro linhas) sobre os resultados encontrados na análise de importância de variáveis
g. Escolha o melhor modelo com base nas métricas de avaliação MSE, MAE e R^2
h. Dê uma breve explicação (máximo de quatro linhas) sobre qual modelo gerou o melhor resultado e a métrica de avaliação utilizada

B - RESOLUÇÃO

1)

A)

```
dados = pd.read_csv("precos_carros_brasil.csv", low_memory=False)
dados.head()
```

B)

```
dados.isna().any()
year_of_reference True
month_of_reference True
fipe_code True
authentication True
brand True
model True
fuel True
gear True
engine_size True
year_model True
avg_price_brl True
dtype: bool

dados.isna().sum()
year_of_reference 65245
month_of_reference 65245
fipe_code 65245
authentication 65245
brand 65245
model 65245
fuel 65245
gear 65245
engine_size 65245
year_model 65245
avg_price_brl 65245
dtype: int64

dados.dropna(inplace=True)
dados.isna().sum()


year_of_reference 0
month_of_reference 0
fipe_code 0
authentication 0
brand 0
model 0
fuel 0
gear 0
engine_size 0
year_model 0
avg_price_brl 0
dtype: int64
```

C)

```
dados.duplicated().sum()
```

D)

```
numericas_cols = [col for col in dados.columns if dados[col].dtype !=
"object”]   categoricas_cols = [col for col in dados.columns if
dados[col].dtype ==           "object”]

dados[numericas_cols].describe()
```

	year_of_reference	year_model	avg_price_brl
count	202295.000000	202295.000000	202295.000000
mean	2021.564695	2011.271514	52756.765713
std	0.571904	6.376241	51628.912116
min	2021.000000	2000.000000	6647.000000
25%	2021.000000	2006.000000	22855.000000
50%	2022.000000	2012.000000	38027.000000
75%	2022.000000	2016.000000	64064.000000
max	2023.000000	2023.000000	979358.000000

E)

```
dados["model"].value_counts()

model
Palio Week. Adv/Adv TRYON 1.8 mpi Flex 425
Focus 1.6 S/SE/SE Plus Flex 8V/16V 5p 425
Focus 2.0 16V/SE/SE Plus Flex 5p Aut. 400
Saveiro 1.6 Mi/ 1.6 Mi Total Flex 8V 400
Corvette 5.7/ 6.0, 6.2 Targa/Stingray 375
...
STEPWAY Zen Flex 1.0 12V Mec. 2
Saveiro Robust 1.6 Total Flex 16V CD 2
Saveiro Robust 1.6 Total Flex 16V 2
Gol Last Edition 1.0 Flex 12V 5p 2
Polo Track 1.0 Flex 12V 5p 2
Name: count, Length: 2112, dtype: int64
dados["brand"].value_counts()
brand
Fiat 44962
VW - VolksWagen 44312
GM - Chevrolet 38590
Ford 33150
Renault 29191
Nissan 12090
Name: count, dtype: int64
```

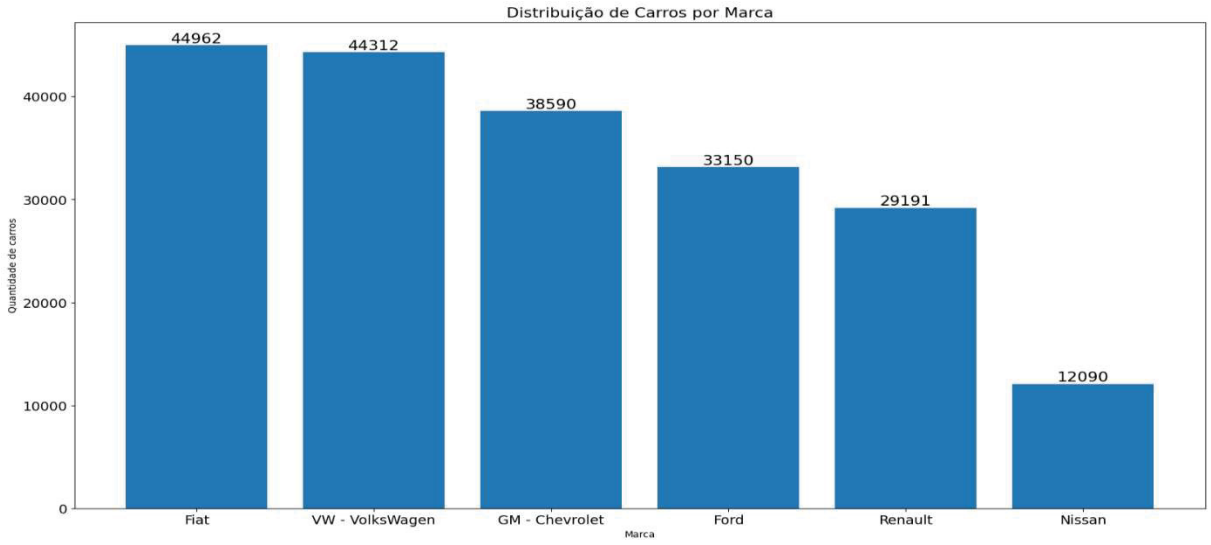
F)

Foram identificados 6 marcas de carros e 2112 modelos diferentes. A base de dados inicial contem 65.245 valores NaN 2 valores duplicados, apos o tratamento desses dados a base de dados contem 202.295 linhas. É notável que a fabricante que mais tem veículos na base é a Fiat, seguida da VW, e GM em terceiro lugar. A fabricante com a menor quantidade é a Nissan.

2)

A)

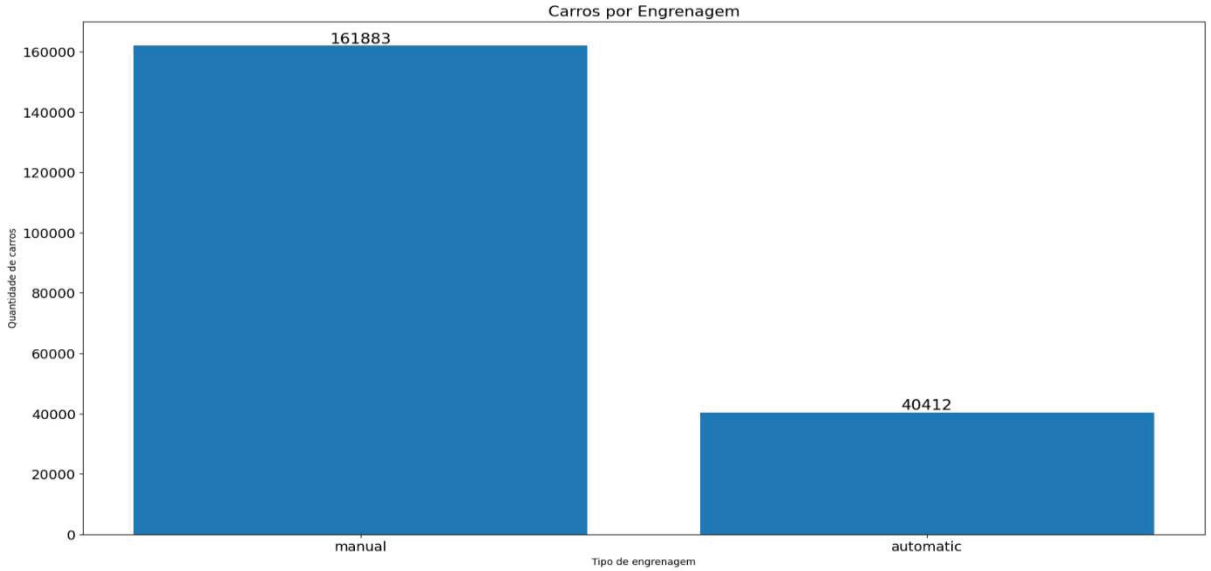
GRÁFICO 2 – DISTRIBUIÇÃO DE CARROS POR MARCA



FONTE: O autor (2025).

B)

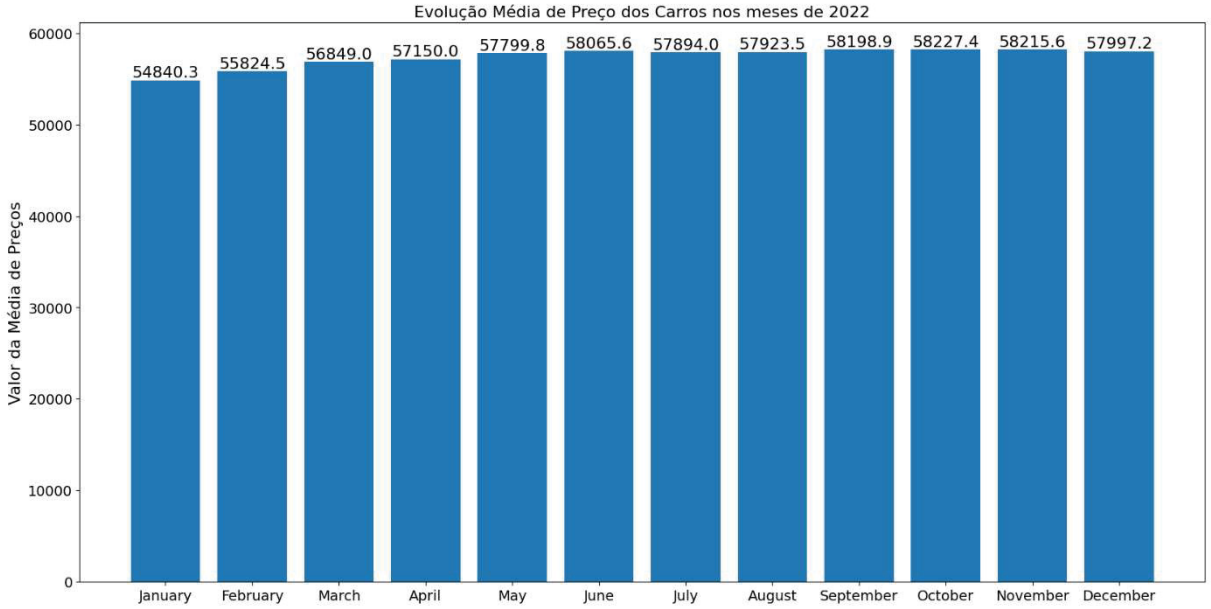
GRÁFICO 2 – DISTRIBUIÇÃO DE CARROS POR ENGRENAGEM



FONTE: O autor (2025).

C)

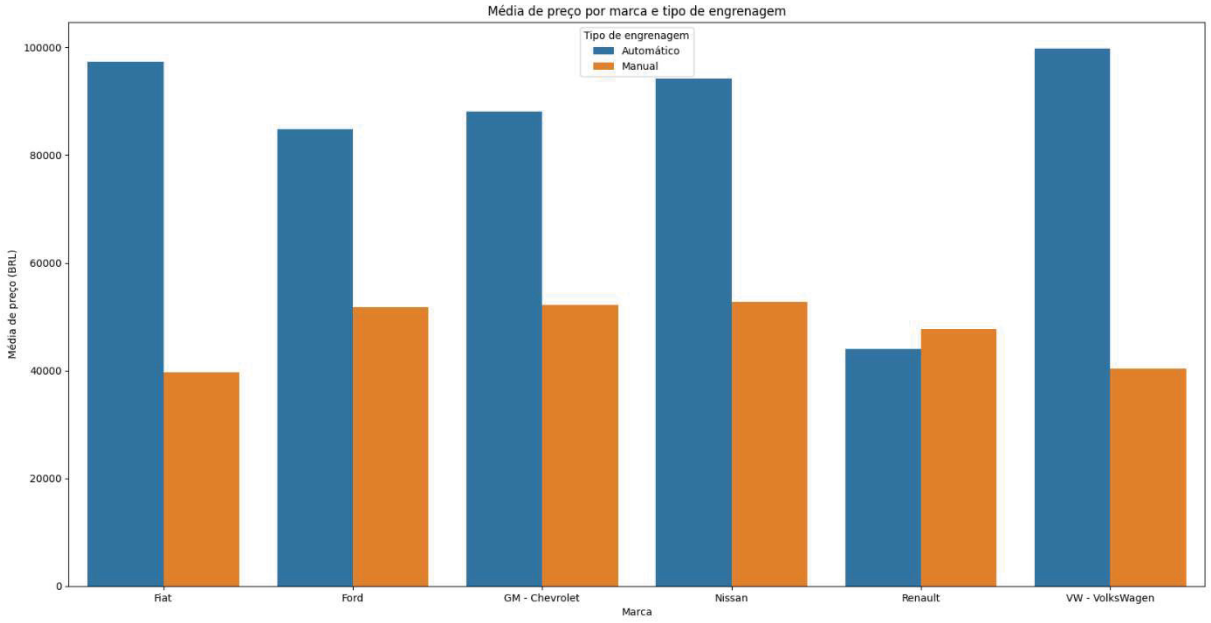
GRÁFICO 3 – DISTRIBUIÇÃO DE EVOLUÇÃO MÉDIA DE PREÇO DOS CARROS



FONTE: O autor (2025).

D)

GRÁFICO 4 – DISTRIBUIÇÃO DA MÉDIA DE PREÇO POR MARCA E TIPO DE ENGRENAGEM



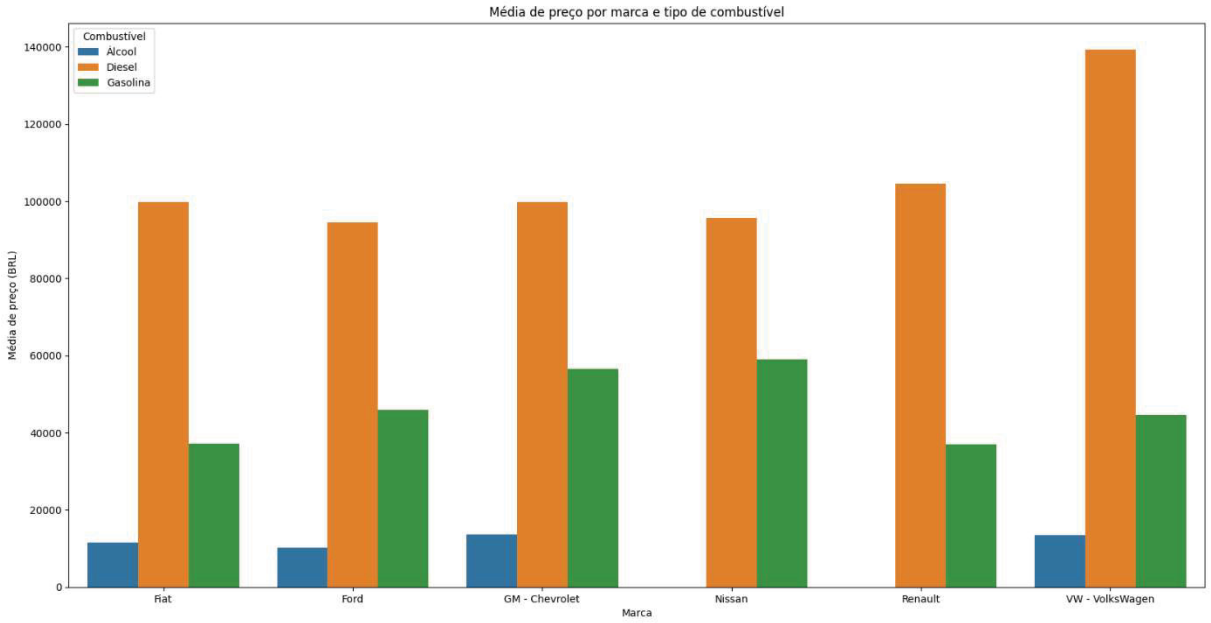
FONTE: O autor (2025).

E)

Os preços de carros automáticos são, em geral, maiores que os preços de carros manuais. A única marca onde isso não ocorreu foi a Renault, onde a média dos preços é semelhante, inclusive sendo um pouco mais elevada nos carros manuais.

F)

GRÁFICO 5 – DISTRIBUIÇÃO DA MÉDIA DE PREÇO POR MARCA E TIPO DE COMBUSTÍVEL



FONTE: O autor (2025).

G)

Os veículos movidos a diesel apresentam um preço médio mais elevado do que os demais tipos de combustíveis. Os veículos movidos a álcool apresentam um preço médio menor do que os demais tipos de combustíveis. Os veículos movidos a gasolina apresentam um preço médio entre os veículos movidos a diesel e a álcool.

3)

A)

TABELA 1 – VARIÁVEIS ESCOLHIDAS

brand	model	gear	fuel	engine	Model_year	year	month	avg_price
2	297	1	2	1.0	2002	2021	4	9162
2	297	1	2	1.0	2001	2021	4	8832
2	297	1	2	1.0	2000	2021	4	8388
2	297	1	0	1.0	2000	2021	4	8453
2	260	1	2	1.6	2001	2021	4	12525

FONTE: O autor (ano).

B)

```
x_train, x_test, y_train, y_test = train_test_split( x, y,
test_size=0.25, random_state=42 )
```

C)

```
model_rf = RandomForest
Regressor() model_rf.fit(x_train, y_train)
model_xgb = XGBRegressor()
model_xgb.fit(x_train, y_train)
```

D)

```
valores_preditos_rf = model_rf.predict(x_test)
valores_preditos_xgb = model_xgb.predict(x_test)
```

E)

TABELA 1 – RANDOM FOREST

variable	importance
engine_size	0.449773
year_model	0.390370
Model	0.059092
gear	0.033169
fuel	0.032056
brand	0.017997
year	0.012330
month	0.005213

FONTE: O autor (2025).

TABELA 3 – XGBOOST

variable	importance
engine_size	0.438108
year_model	0.200562
Model	0.022404
gear	0.113473
fuel	0.146407
brand	0.055805
year	0.018362
month	0.004878

FONTE: O autor (2025).

F)

O ano do carro (year_model) e o tamanho do motor (engine_size) se mostraram as variáveis mais importantes na predição do valor dos automóveis. O tipo de combustível acabou não se mostrando tão importante, o que é curioso pois a média de preço dos veículos a diesel era superior às outras. O ano e mês de referência dos dados foram as variáveis de menos importância.

G)

```
RandomForest
mse_rf = mean_squared_error(y_test, valores_preditos_rf)
mse_rf
11681843.441155115


mae_rf = mean_absolute_error(y_test, valores_preditos_rf)
mae_rf
1732.1978164799061

r2_rf = r2_score(y_test, valores_preditos_rf)
r2_rf
0.995659336124847


XGBoost

mse_xgb = mean_squared_error(y_test, valores_preditos_xgb)
mse_xgb
29555221.170828193

mae_xgb = mean_absolute_error(y_test, valores_preditos_xgb)
mae_xgb
3173.461879807312

r2_xgb = r2_score(y_test, valores_preditos_xgb)
r2_xgb
0.98901806196046
```

H)

O modelo Random Forest apresentou melhor desempenho em todas as métricas avaliadas: menor MSE (11.681.843,44), menor MAE (1.732,12) e R^2 mais próximo de 1 (0,9957). Portanto, o Random Forest teve um desempenho superior ao XGBoost na previsão com base nas métricas utilizadas.

APÊNDICE 3 – LINGUAGEM R

A – ENUNCIADO

1 Pesquisa com Dados de Satélite (Satellite)

O banco de dados consiste nos valores multiespectrais de pixels em vizinhanças 3x3 em uma imagem de satélite, e na classificação associada ao pixel central em cada vizinhança. O objetivo é prever esta classificação, dados os valores multiespectrais.

Um quadro de imagens do Satélite Landsat com MSS (*Multispectral Scanner System*) consiste em quatro imagens digitais da mesma cena em diferentes bandas espectrais. Duas delas estão na região visível (correspondendo aproximadamente às regiões verde e vermelha do espectro visível) e duas no infravermelho (próximo). Cada pixel é uma palavra binária de 8 bits, com 0 correspondendo a preto e 255 a branco. A resolução espacial de um pixel é de cerca de 80m x 80m. Cada imagem contém 2340 x 3380 desses pixels. O banco de dados é uma subárea (minúscula) de uma cena, consistindo de 82 x 100 pixels. Cada linha de dados corresponde a uma vizinhança quadrada de pixels 3x3 completamente contida dentro da subárea 82x100. Cada linha contém os valores de pixel nas quatro bandas espectrais (convertidas em ASCII) de cada um dos 9 pixels na vizinhança de 3x3 e um número indicando o rótulo de classificação do pixel central.

As classes são: solo vermelho, colheita de algodão, solo cinza, solo cinza úmido, restolho de vegetação, solo cinza muito úmido.

Os dados estão em ordem aleatória e certas linhas de dados foram removidas, portanto você não pode reconstruir a imagem original desse conjunto de dados. Em cada linha de dados, os quatro valores espectrais para o pixel superior esquerdo são dados primeiro, seguidos pelos quatro valores espectrais para o pixel superior central e, em seguida, para o pixel superior direito, e assim por diante, com os pixels lidos em sequência, da esquerda para a direita e de cima para baixo. Assim, os quatro valores espectrais para o pixel central são dados pelos atributos 17, 18, 19 e 20. Se você quiser, pode usar apenas esses quatro atributos, ignorando os outros. Isso evita o problema que surge quando uma vizinhança 3x3 atravessa um limite.

O banco de dados se encontra no pacote **mlbench** e é completo (não possui dados faltantes).

Tarefas:

1. Carregue a base de dados Satellite
2. Crie partições contendo 80% para treino e 20% para teste
3. Treine modelos RandomForest, SVM e RNA para predição destes dados.
4. Escolha o melhor modelo com base em suas matrizes de confusão.
5. Indique qual modelo dá o melhor o resultado e a métrica utilizada

2 Estimativa de Volumes de Árvores

Modelos de aprendizado de máquina são bastante usados na área da engenharia florestal (mensuração florestal) para, por exemplo, estimar o volume de madeira de árvores sem ser necessário abatê-las.

O processo é feito pela coleta de dados (dados observados) através do abate de algumas árvores, onde sua altura, diâmetro na altura do peito (dap), etc, são medidos de forma exata. Com estes dados, treina-se um modelo de AM que pode estimar o volume de outras árvores da população.

Os modelos, chamados alométricos, são usados na área há muitos anos e são baseados em regressão (linear ou não) para encontrar uma equação que descreve os dados. Por exemplo, o modelo de Spurr é dado por:

$$\textbf{Volume} = b0 + b1 * dap^2 * Ht$$

Onde dap é o diâmetro na altura do peito (1,3metros), Ht é a altura total. Tem-se vários modelos alométricos, cada um com uma determinada característica, parâmetros, etc. Um modelo de regressão envolve aplicar os dados observados e encontrar b0 e b1 no modelo apresentado, gerando assim uma equação que pode ser usada para prever o volume de outras árvores.

Dado o arquivo **Volumes.csv**, que contém os dados de observação, escolha um modelo de aprendizado de máquina com a melhor estimativa, a partir da estatística de correlação.

Tarefas

1. Carregar o arquivo Volumes.csv (http://www.razer.net.br/datasets/Volumes.csv)
2. Eliminar a coluna NR, que só apresenta um número sequencial
3. Criar partição de dados: treinamento 80%, teste 20%
4. Usando o pacote "caret", treinar os modelos: Random Forest (rf), SVM (svmRadial), Redes Neurais (neuralnet) e o modelo alométrico de SPURR

    - O modelo alométrico é dado por: Volume = b0 + b1 * dap^2 * Ht

      alom <- nls(VOL ~ b0 + b1*DAP*DAP*HT, dados, start=list(b0=0.5, b1=0.5))

5. Efetue as predições nos dados de teste
6. Crie suas próprias funções (UDF) e calcule as seguintes métricas entre a predição e os dados observados

    - Coeficiente de determinação: R^2

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \widehat{y_i})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

onde y_i é o valor observado, $\widehat{y_i}$ é o valor predito e $\bar{y}$ é a média dos valores y_i observados. Quanto mais perto de 1 melhor é o modelo;

- Erro padrão da estimativa: S_{yx}

$$S_{yx} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n-2}}$$

esta métrica indica erro, portanto quanto mais perto de 0 melhor é o modelo;

- Syx%

$$S_{yx}\% = \frac{S_{yx}}{\bar{y}} * 100$$

esta métrica indica porcentagem de erro, portanto quanto mais perto de 0 melhor é o modelo;

7. Escolha o melhor modelo.

B – RESOLUÇÃO

1)

```
# carregar pacotes
> library("mlbench")
> library("caret")
> data("Satellite")
> Satellite <- Satellite[c("x.17", "x.18", "x.19", "x.20", "classes")]
```

2)

```
> indices <- createDataPartition(Satellite$classes, p=0.8,
list=FALSE)
> treino <- Satellite[indices,] > teste <- Satellite[-indices, ]
```

3)

```
> rf <- train(classes~., data=treino, method="rf")
> svm <- train(classes~., data=treino, method="svmRadial")
> rna <- train(classes~., data=treino, method="nnet")
> predicoes.rf <- predict(rf, teste)
> predicoes.svm <- predict(svm, teste)
> predicoes.rna <- predict(rna, teste)
```

4)

```
> matrizesConfusao.rf <- confusionMatrix(predicoes.rf, teste$classes)
> matrizesConfusao.svm <- confusionMatrix(predicoes.svm,
teste$classes)
> matrizesConfusao.rna <- confusionMatrix(predicoes.rna,
teste$classes)
> matrizesConfusao.rf
```

TABELA 4 – MATRIZ DE CONFUSÃO

Reference \ Predicted	Red soil	Cotton crop	Grey soil	Damp grey soil	Vegetation stubble	Very damp grey soil
Red soil	293	0	2	1	4	0
Cotton crop	0	129	0	0	5	1
Grey soil	5	0	249	27	3	12
Damp grey soil	0	1	16	61	1	34
Vegetation stubble	7	9	0	1	117	11
Very damp grey soil	1	1	4	35	11	243

FONTE: O autor (2025).

```
Overall Statistics
Accuracy:0.8505
95%CI:(0.8298,0.8695)
No Information Rate:0.2383
P-Value[Acc>NIR]:<2.2e-16
Kappa:0.8152
Mcnemar'sTestP-Value:NA
>matrizesConfusao.svm
```

TABELA 5 – MATRIZ DE CONFUSÃO

Reference \ Predicted	Red soil	Cotton crop	Grey soil	Damp grey soil	Vegetation stubble	Very damp grey soil
Red soil	294	0	1	1	5	0
Cotton crop	0	126	0	0	0	0
Grey soil	5	0	263	26	3	11
Damp grey soil	0	2	7	75	1	38
Vegetation stubble	7	11	0	0	122	6
Very damp grey soil	0	1	0	23	10	246

FONTE: O autor (2025).

```
Overall Statistics
Accuracy:0.8769
95%CI:(0.8577,0.8944)
NoInformationRate:0.2383
P-Value[Acc>NIR]:<2.2e-16
Kappa:0.8481
Mcnemar'sTestP-Value:NA
>matrizesConfusao.rna
```

TABELA 6 – MATRIZ DE CONFUSÃO

Reference \ Predicted	Red soil	Cotton crop	Grey soil	Damp grey soil	Vegetation stubble	Very damp grey soil
Red soil	291	0	3	1	3	2
Cotton crop	1	128	0	0	0	0
Grey soil	8	0	261	31	3	16
Damp grey soil	1	1	6	33	1	17
Vegetation stubble	5	10	0	0	119	12
Very damp grey soil	0	1	1	60	15	254

FONTE: O autor (2025).

```
Overall Statistics

Accuracy:0.8458
95%CI:(0.8249,0.8651)
No Information Rate:0.2383
P-Value[Acc>NIR]:<2.2e-16
Kappa:0.8081
Mcnemar'sTestP-Value:NA
```

5)

O melhor modelo é o svm, obtendo uma acurácia de 87,69%. Os modelos RandomForest e RNA atingiram 85,05% e 84,58%, respectivamente.

2)
1)

```
>df<-read.csv("http://www.razer.net.br/datasets/Volumes.csv",
sep=";",dec=",")
```

2)

```
>df$NR<-NULL
```

3)

```
>indices<-createDataPartition(df$VOL,p=0.8,list=FALSE)
>treino<-df[indices,]
>teste<-df[-indices,]
```

4)

```
> rf <- train(VOL~., data=treino, method="rf")
> svm <- train(VOL~., data=treino, method="svmRadial")
> rna <- train(VOL~., data=treino, method="nnet")
> alom <- nls(VOL ~ b0 + b1*DAP*DAP*HT, treino, start=list(b0=0.5,
b1=0.5))
```

5)

```
> predicoes.rf <- predict(rf, teste)
> predicoes.svm <- predict(svm, teste)
> predicoes.rna <- predict(rna, teste)
> predicoes.alom <- predict(alom, teste)
```

6)

```
> func_r2 <- function(predicoes, observacoes) {
+ res <- sum( (observacoes- predicoes) ^ 2 )
+ tot <- sum( (observacoes- mean(observacoes)) ^2 )
+ return (1- ( res / tot ))
+ }
>  func_Syx <- function(predicoes, observacoes) {
+ res <- sum( (observacoes- predicoes) ^ 2 )
+ gl <- length(predicoes)- 2
+ return (sqrt (res / gl) )
+ }
> func_Syx_p <- function(predicoes, observacoes) {
+ syx <- func_Syx(predicoes, observacoes)
+ return ( (syx / mean(observacoes)) * 100 )
+ }
```

7)

O melhor modelo é o Random Forest.

APÊNDICE 4 – ESTATÍSTICA APLICADA I

A – ENUNCIADO

1) Gráficos e tabelas

(15 pontos) Elaborar os gráficos box-plot e histograma das variáveis “age” (idade da esposa) e “husage” (idade do marido) e comparar os resultados

(15 pontos) Elaborar a tabela de frequencias das variáveis “age” (idade da esposa) e “husage” (idade do marido) e comparar os resultados

2) Medidas de posição e dispersão

(15 pontos) Calcular a média, mediana e moda das variáveis “age” (idade da esposa) e “husage” (idade do marido) e comparar os resultados

(15 pontos) Calcular a variância, desvio padrão e coeficiente de variação das variáveis “age” (idade da esposa) e “husage” (idade do marido) e comparar os resultados

3) Testes paramétricos ou não paramétricos

(40 pontos) Testar se as médias (se você escolher o teste paramétrico) ou as medianas (se você escolher o teste não paramétrico) das variáveis “age” (idade da esposa) e “husage” (idade do marido) são iguais, construir os intervalos de confiança e comparar os resultados.

Obs:

Você deve fazer os testes necessários (e mostra-los no documento pdf) para saber se você deve usar o unpaired test (paramétrico) ou o teste U de Mann-Whitney (não paramétrico), justifique sua resposta sobre a escolha.

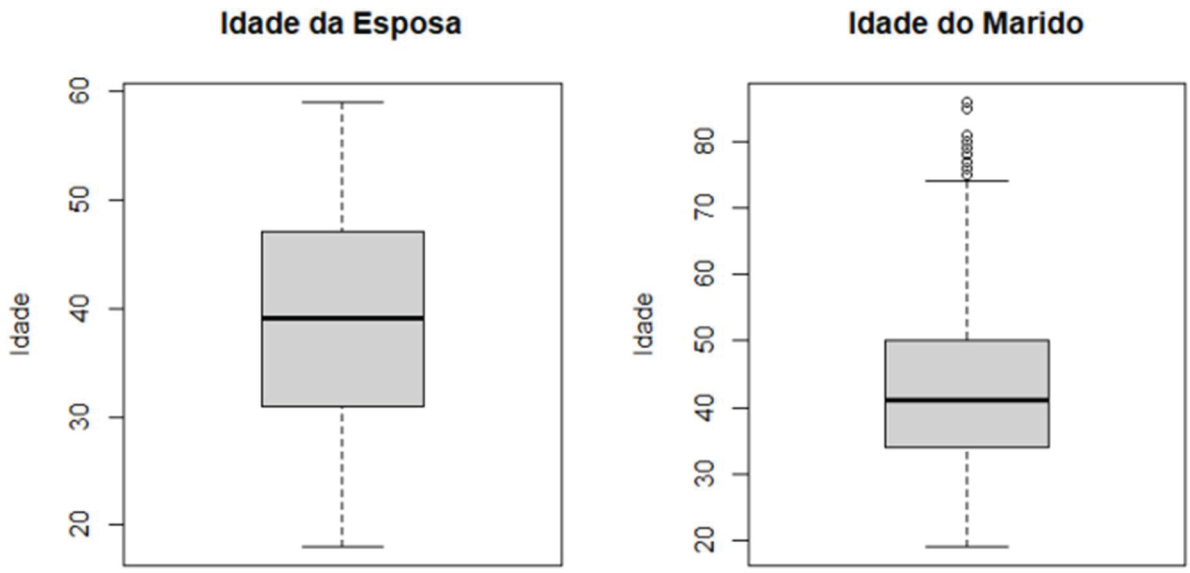
Lembre-se de que os intervalos de confiança já são mostrados nos resultados dos testes citados no item 1 acima.

B – RESOLUÇÃO

1)

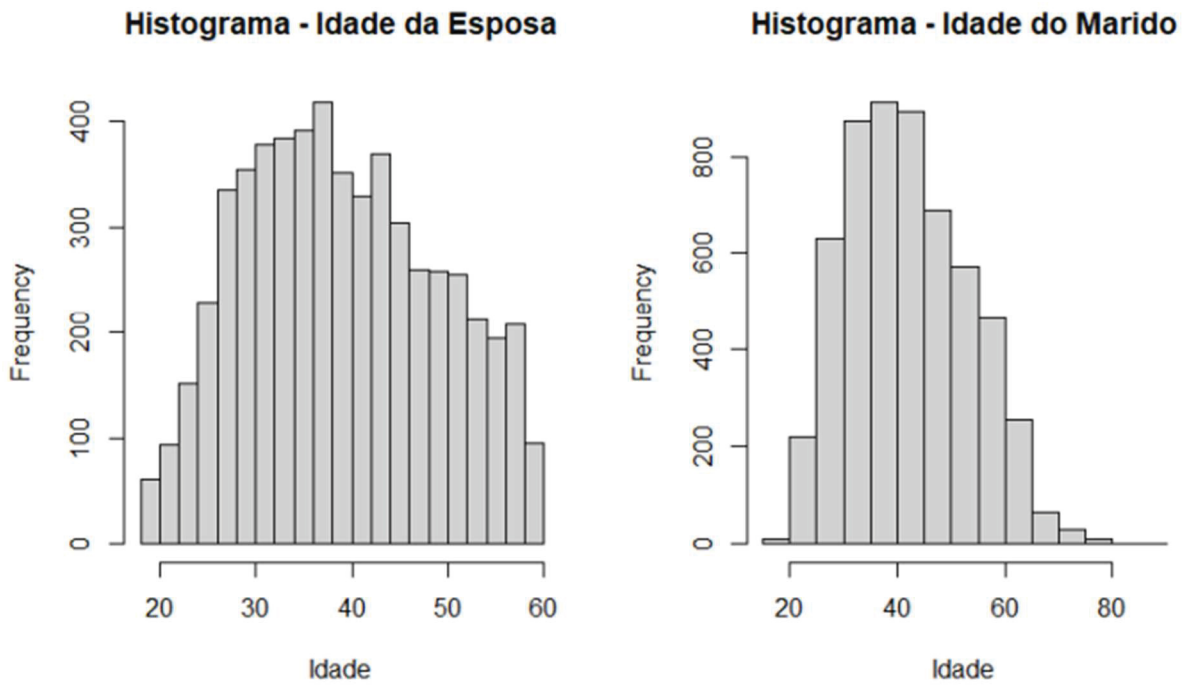
A)

GRÁFICO 6 – BOXPLOT DAS IDADES



FONTE: O autor (2025).

GRÁFICO 7 – HISTOGRAMA DAS IDADES



FONTE: O autor (2025).

Analisando os gráficos gerados, podemos verificar, que as idades dos maridos (husage) se distribuem em uma faixa maior de valores, com alguns registros chegando na casa dos 80 anos. Também podemos verificar, no box-plot, que a mediana das duas variáveis está próxima de 40.

B)

TABELA 7 – FREQUENCIA DE “AGE”

Class Limits	rf	rf (%)	cf	cf (%)
[17.82, 20.804)	61	0.01	61	1.08
[20.804, 23.787)	161	0.03	222	3.94
[23.787, 26.771)	312	0.06	534	9.48
[26.771, 29.754)	505	0.09	1039	18.44
[29.754, 32.738)	562	0.10	1601	28.42
[32.738, 35.721)	571	0.10	2172	38.55
[35.721, 38.705)	624	0.11	2796	49.63
[38.705, 41.689)	510	0.09	3306	58.68
[41.689, 44.672)	542	0.10	3848	68.30
[44.672, 47.656)	432	0.08	4280	75.97
[47.656, 50.639)	389	0.07	4669	82.87
[50.639, 53.623)	358	0.06	5027	89.23
[53.623, 56.606)	304	0.05	5331	94.62
[56.606, 59.590)	303	0.05	5634	100.00

FONTE: O autor (2025).

TABELA 8 – FREQUENCIA DE “HUSAGE”

Class Limits	f	rf (%)	cf	cf (%)
[18.81, 23.671)	102	0.02	102	1.81
[23.671, 28.531)	466	0.08	568	10.08
[28.531, 33.392)	809	0.14	1377	24.44
[33.392, 38.253)	895	0.16	2272	40.33
[38.253, 43.114)	917	0.16	3189	56.60
[43.114, 47.974)	629	0.11	3818	67.77
[47.974, 52.835)	649	0.12	4467	79.29
[52.835, 57.696)	541	0.10	5008	88.89
[57.696, 62.556)	394	0.07	5402	95.88
[62.556, 67.417)	152	0.03	5554	98.58
[67.417, 72.278)	51	0.01	5605	99.49
[72.278, 77.139)	21	0.00	5626	99.86
[77.139, 81.999)	6	0.00	5632	99.96
[81.999, 86.86)	2	0.00	5634	100.00

FONTE: O autor (2025).

2)

A)

Idade da Esposa:
Média: 39.42758
Mediana: 39
Moda: 37

Idade do Marido:
Média: 42.45296
Mediana: 41
Moda: 44

A idade média dos maridos é 7,67% superior à das esposas.
A mediana da idade dos maridos é 5,12% superior à das esposas.
A moda da idade dos maridos é 18,92% superior à das esposas.

B)

Idade da Esposa:
Variância: 99.75234
Desvio Padrão: 9.98761
Coeficiente de Variação: 25.33153%

Idade do Marido:
Variância: 126.0717
Desvio Padrão: 11.22817
Coeficiente de Variação: 26.44849%

Portanto, a variância da idade dos maridos é 26,38% superior à da idade das esposas. O desvio padrão da idade dos maridos é 12,42% superior ao desvio padrão da idade das esposas. Quanto aos coeficientes de variação, que ficaram em 25,33% e 26,45%, podemos dizer que existe média dispersão entre os valores destas variáveis nessa base de dados.

3)
A)

Vamos fazer checagens preliminares para verificar as exigencias do teste: Amostras independentes, normalidade e homogeneidade das variancias entre grupos.
Premissa 1: As duas amostras sao independentes? Sim, pois os são dados dos maridos e esposas, e não dados do mesmo indivíduo em dois momentos. Nao se trata de uma amostra ou grupos emparelhados.
Premissa 2: Os dados de cada amostra/grupo possuem distribuicao normal? Vamos usar o teste de normalidade com o seguinte teste de hipoteses:
- H0: os dados sao normalmente distribuidos
- Ha: os dados nao sao normalmente distribuidos
Primeiro vamos fazer o teste de normalidade para a Idade das esposas

Anderson-Darling normality test
data: salarios$age
A = 31.828, p-value < 0.00000000000000022

P-value < 0.00000000000000022 (ou seja, <0.05), então rejeitamos H0 e identificamos que os dados não são normalmente distribuídos.

Agora vamos fazer o teste de normalidade para a Idade dos maridos
> ad.test(salarios$husage)
Anderson-Darling normality test data:
salarios$husage A = 28.176, p-value < 0.00000000000000022

P-value < 0.00000000000000022 (ou seja, <0.05), então rejeitamos H0 e identificamos que os dados não são normalmente distribuídos.
Conclusão: não é possível usar o unpaired test (paramétrico), pois não foi atendida a premissa de que os dados do grupo possuem distribuição normal.

Portanto, vamos utilizar o teste não paramétrico:
Vamos fazer o teste se a idade mediana dos maridos é estatisticamente igual a idade mediana das mulheres.
Hipoteses do teste:
H0: A idade mediana dos maridos é igual estatisticamente a idade mediana das mulheres;
Ha: A idade mediana dos maridos não é igual estatisticamente a idade mediana das mulheres

```
> res <- wilcox.test(salarios$age, salarios$husage, + exact = FALSE,
conf.int=TRUE)
> res
```

```
Wilcoxon rank sum test with continuity correction
data: salarios$age and salarios$husage
W = 13619912, p-value < 0.00000000000000022
alternative hypothesis: true location shift is not equal to 0 95 percent
confidence interval:

-3.000024 -2.000033

sample estimates:
difference in location -2.999966
```

Com base no resultado do teste de Wilcoxon, rejeitamos H0, pois p-value é menor que 0,05. Portanto, concluímos que a idade mediana dos maridos é estatisticamente diferente da idade mediana das esposas. O intervalo de confiança da diferença entre as medianas está entre -3 e -2, com uma diferença da mediana de -2,999966, ou seja, a mediana da idade das mulheres aproximadamente 3 anos inferior à dos maridos.

APÊNDICE 5 – ESTATÍSTICA APLICADA II

A – ENUNCIADO

Regressões Ridge, Lasso e ElasticNet

(100 pontos) Fazer as regressões Ridge, Lasso e ElasticNet com a variável dependente "lwage" (salário-hora da esposa em logaritmo neperiano) e todas as demais variáveis da base de dados são variáveis explicativas (todas essas variáveis tentam explicar o salário-hora da esposa). No pdf você deve colocar a rotina utilizada, mostrar em uma tabela as estatísticas dos modelos (RMSE e R^2) e concluir qual o melhor modelo entre os três, e mostrar o resultado da predição com intervalos de confiança para os seguintes valores:

husage = 40 (anos – idade do marido)
husunion = 0 (marido não possui união estável)
husearns = 600 (US$ renda do marido por semana)
huseduc = 13 (anos de estudo do marido)
husblck = 1 (o marido é preto)
hushisp = 0 (o marido não é hispânico)
hushrs = 40 (horas semanais de trabalho do marido)
kidge6 = 1 (possui filhos maiores de 6 anos)
age = 38 (anos – idade da esposa)
black = 0 (a esposa não é preta)
educ = 13 (anos de estudo da esposa)
hispanic = 1 (a esposa é hispânica)
union = 0 (esposa não possui união estável)
exper = 18 (anos de experiência de trabalho da esposa)
kidlt6 = 1 (possui filhos menores de 6 anos)

obs: lembre-se de que a variável dependente "lwage" já está em logarítmo, portanto voçê não precisa aplicar o logaritmo nela para fazer as regressões, mas é necessário aplicar o antilog para obter o resultado da predição.

B – RESOLUÇÃO

TABELA 9 – DESEMPENHO DOS MODELOS

Regressão	RMSE (treino)	R^2 (treino)	RMSE (teste)	R^2 (teste)
Ridge	0.2930866	0.6867948	0.281537	0.7009349
Lasso	0.2931455	0.686669	0.2797465	0.7047266
ElasticNet	0.292994	0.6869926	0.2800212	0.7041465

FONTE: O autor (2025).

Ao analisar as métricas dos diferentes modelos, percebemos que nos três casos os valores ficaram muito próximos, com diferenças mínimas entre eles. Sendo necessária a seleção de um deles, poderia ser selecionado o ElasticNet, que obteve os melhores números na base de treino e o segundo melhor na base de teste. Porém, como a diferença observada é muito pequena, qualquer um dos modelos, neste caso, deverá apresentar resultados semelhantemente corretos.

Predição e intervalos de confiança:

Na tabela a seguir podemos ver os valores preditos de salário-hora, bem como o intervalo de confiança, exibindo o valor mínimo e máximo:

TABELA 10 – PREVISÕES E INTERVALOS DE CONFIANÇA

Regressão	Valor predito	Intervalo de confiança
Ridge	US$ 8.79	US$ 8.59 a US$ 8.99
Lasso	US$ 8.83	US$ 8.63 a US$ 9.03
ElasticNet	US$ 8.89	US$ 8.69 a US$ 9.09

FONTE: O autor (2025).

APÊNDICE 6 – ARQUITETURA DE DADOS

A – ENUNCIADO

1 Construção de Características: Identificador automático de idioma

O problema consiste em criar um modelo de reconhecimento de padrões que dado um texto de entrada, o programa consegue classificar o texto e indicar a língua em que o texto foi escrito.

Parta do exemplo (notebook produzido no Colab) que foi disponibilidade e crie as funções para calcular as diferentes características para o problema da identificação da língua do texto de entrada.

Nessa atividade é para "construir características".

Meta: a acurácia deverá ser maior ou igual a 70%.

Essa tarefa pode ser feita no Colab (Google) ou no Jupiter, em que deverá exportar o notebook e imprimir o notebook para o formato PDF. Envie no UFPR Virtual os dois arquivos.

2 Melhore uma base de dados ruim

Escolha uma base de dados pública para problemas de classificação, disponível ou com origem na UCI Machine Learning.

Use o mínimo de intervenção para rodar a SVM e obtenha a matriz de confusão dessa base.

O trabalho começa aqui, escolha as diferentes tarefas discutidas ao longo da disciplina, para melhorar essa base de dados, até que consiga efetivamente melhorar o resultado.

Considerando a acurácia para bases de dados balanceadas ou quase balanceadas, se o percentual da acurácia original estiver em até 85%, a meta será obter 5%. Para bases com mais de 90% de acurácia, a meta será obter a melhora em pelo menos 2 pontos percentuais (92% ou mais).

Nessa atividade deverá ser entregue o script aplicado (o notebook e o PDF correspondente).

B – RESOLUÇÃO

Apresentar a resolução (somente o resultado) das questões do trabalho.

```
        from sklearn.model_selection import train_test_split
        import numpy as np

        vet = np.array(padroes) classes = vet[:, -1] # classes = [p[-1] for
p in padroes]
        padroes_sem_classe = vet[:, 0:-1]
        X_train, X_test, y_train, y_test = train_test_split(
padroes_sem_classe, classes, test_size=0.25, stratify=classes )
        treinador = svm.SVC()
        modelo = treinador.fit(X_train, y_train)
        acuracia = modelo.score(X_train, y_train)
        print("Acurácia nos dados de treinamento: {:.2f}%".format(acuracia *
100)
        y_pred = modelo.predict(X_train)
        cm = confusion_matrix(y_train, y_pred)
        print(cm)
        print(classification_report(y_train, y_pred))
        print("métricas mais confiáveis")
        y_pred2 = modelo.predict(X_test)
        cm = confusion_matrix(y_test, y_pred2)
        print(cm)
        print(classification_report(y_test, y_pred2))
```

Acurácia nos dados de treinamento: 75.36%

TABELA 11 – MATRIZ DE CONFUSÃO

9	5	8
2	20	1
1	0	23

FONTE: O autor (2025).

TABELA 12 – DESEMPENHO DAS PREVISÕES

	Precision	recall	F1-score	support
Espanhol	0,75	0,41	0,53	22
Inglês	0,80	0,87	0,83	23
Português	0,72	0,96	0,82	24

FONTE: O autor (2025).

TABELA 13 – DESEMPENHO DAS PREVISÕES

accuracy			0,75	69
macro avg	0,76	0,75	0,73	69
weighted avg	0,76	0,75	0,73	69

FONTE: O autor (2025).

Métricas mais confiáveis

TABELA 14 – MATRIZ DE CONFUSÃO

1	3	4
0	6	1
0	1	7

FONTE: O autor (2025).

TABELA 15 – DESEMPENHO DAS PREVISÕES

	Precision	recall	F1-score	support
espanhol	1,00	0,12	0,22	8
inglês	0,60	0,86	0,71	7
português	0,58	0,88	0,70	8

FONTE: O autor (2025).

TABELA 16 – DESEMPENHO DAS PREVISÕES

accuracy			0,61	23
macro avg	0,73	0,62	0,54	23
weighted avg	0,73	0,61	0,54	23

FONTE: O autor (2025).

APÊNDICE 7 – APRENDIZADO DE MÁQUINA

A – ENUNCIADO

Para cada uma das tarefas abaixo (Classificação, Regressão etc.) e cada base de dados (Veículo, Diabetes etc.), fazer os experimentos com todas as técnicas solicitadas (KNN, RNA etc.) e preencher os quadros com as estatísticas solicitadas, bem como os resultados pedidos em cada experimento.

B – RESOLUÇÃO

TABELA 17 – VEÍCULO

Técnica	Parâmetro	Acurácia	Matriz de Confusão				
SVM-CV	C=100 Sigma=0,015	0,8503	P/R	bus	opel	saab	van
			bus	42	0	0	0
			opel	0	28	7	1
			saab	0	14	35	1
			van	1	0	1	37
RNA-CV	Size=21 Decay=0,7	0,8323	P/R	bus	opel	saab	van
			bus	40	2	0	0
			opel	0	28	9	0
			saab	2	11	33	1
			van	1	1	1	38
SVM-Hold-out	C=1 Sigma=0,06667012	0,7844	P/R	bus	opel	saab	van
			bus	41	0	0	0
			opel	0	24	13	0
			saab	0	17	28	1
			van	2	1	2	38
RF-CV	Mtry=5	0,7665	P/R	bus	opel	saab	van
			bus	42	0	0	0
			opel	0	24	15	0
			saab	0	17	25	2
			van	1	1	3	37
RF-Hold-out	Mtry=10	0,7545	P/R	bus	opel	saab	van
			bus	42	0	0	0
			opel	0	25	17	1
			saab	0	16	23	2
			van	1	1	3	36
KNN	K=1	0,6168	P/R	bus	opel	saab	Van
			bus	36	2	3	1
			opel	1	18	21	3

			saab	4	20	17	3
			van	2	2	2	32
RNA-Hold-out	Size=5	0,485	P/R	bus	opel	saab	van
	Decay=0,1		bus	42	40	42	1
			opel	1	1	0	0
			saab	0	0	0	0
			van	0	1	1	38

FONTE: O autor (2025).

TABELA 18 – DIABETES

Técnica	Parâmetro	Acurácia	Matriz de Confusão		
RNA-CV	Size=11	0,7908	P/R	neg	pos
	Decay=0,4		neg	87	19
			pos	13	34
SVM-CV	C=2	0,7843	P/R	neg	pos
	Sigma=0,01		neg	87	20
			pos	13	33
RNA – Hold - out	Size=3	0,7778	P/R	neg	pos
	decay=0,1		neg	85	19
			pos	15	34
SVM - Hold - out	C=0,5	0,7778	P/R	neg	pos
	Sigma=0,1449667		neg	86	20
			pos	14	33
RF - Hold - out	Mtry=2	0,7778	P/R	neg	pos
			neg	87	21
			pos	13	32
RF - CV	Mtry=2	0,7712	P/R	neg	pos
			neg	86	21
			pos	14	32
KNN	K=9	0,7647	P/R	neg	pos
			neg	85	21
			pos	15	32

FONTE: O autor (2025).

TABELA 19 – ADMISSÃO

Técnica	Parâmetro	R2	Syx	Pearson	RMSE	MAE

SVM – CV	C=2 Sigma= 0,01	0,8576092	0,5451641	0,05673658	0,05506989	0,03948337
RF – Hold-out	mtry=2	0,8304361	0,5949125	0,05589008	0,06009524	0,04293873
RF - CV	mtry=2	0,8292982	0,5969053	0,05583056	0,06029654	0,04278709
RNA - CV	size=9 decay=0,1	0,822511	0,6086562	0,05306953	0,06148356	0,04558092
SVM – Hold-out	C=1 Sigma= 0,201492	0,8079223	0,6331766	0,05438461	0,06396049	0,04449268
RNA – Hold-out	Size=5 Decay=0,1	0,8032007	0,6409116	0,05182541	0,06474185	0,04748661
KNN	K=9	0,7832191	0,6726619	0,05741561	0,06794912	0,05113379

FONTE: O autor (2025).

TABELA 20 – BIOMASSA

Técnica	Parâmetro	R2	Syx	Pearson	RMSE	MAE
RF - CV	mtry=2	0,9155052	2986,813	0,837266	385,596	112,6905
RNA - CV	size=10 decay=0,1	0,908381	3110,182	0,8334981	401,5228	125,549
RF – Hold-out	mtry=2	0,8994709	3257,909	0,8316976	420,5943	117,8495
SVM – CV	C=100 Sigma= 0,01	0,8725528	3668,245	0,8137989	473,5684	159,3758
KNN	K=1	0,8113007	4463,528	0,8015866	576,239	152,3858
SVM – Hold-out	C=1 Sigma= 1,144223	0,4505385	7616,611	0,6136272	983,3002	236,1239
RNA – Hold-out	Size=5 Decay=0,1	-1,259574	15445,66	0,3510669	1994,026	518,2475

FONTE: O autor (2025).

TABELA 21 – AGRUPAMENTO DE VEÍCULO

Cluster	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis
0	89.9600	39.1733	75.7067	168.4267	65.4933	9.4000	149.6933	44.1733	19.0000	135.4533	173.8800
1	107.2500	55.4167	103.4167	190.2500	65.7917	5.7917	248.7083	26.8750	27.0833	136.0833	271.2917
2	91.0495	45.5455	80.1485	157.9505	63.0792	10.0198	156.2574	43.0198	19.5646	152.1881	176.2475
3	93.4714	41.3286	80.1429	166.9571	61.1429	9.1429	165.7857	39.6143	21.3642	156.4286	192.6857
4	104.1241	53.6569	102.6715	201.8759	62.4234	10.4745	217.1825	30.7007	24.4236	169.2847	227.1022
5	85.6667	36.7407	87.3148	122.3446	56.1852	6.4444	120.8148	55.7222	22.4943	140.3148	203.7407
6	95.9851	45.9552	90.6418	196.2537	64.3433	8.3134	182.4179	35.8955	21.9254	148.6716	203.2687
7	102.1413	49.7051	100.5128	204.0513	63.6667	9.3718	200.6667	32.8077	22.4974	186.5897	218.5897
8	85.6107	43.1985	70.4122	138.4962	58.9313	7.1985	148.4885	45.2971	18.9389	134.6794	170.3905
9	88.8807	38.8899	66.8269	138.2477	57.8073	7.2661	134.4128	49.9174	18.0257	135.4128	156.9083

FONTE: O autor (2025).

TABELA 22 – REGRAS DE ASSOCIAÇÃO

Nº	lhs	rhs	support	confidence	coverage	lift	count
1	{Afundo}	=> {Gemeos}	0.3461538	1.0000000	0.3461538	1.529412	9
2	{AgachamentoSmith}	=> {Extensor}	0.3461538	0.9000000	0.3846154	1.800000	9
3	{Esteira}	=> {Extensor}	0.4230769	0.9166667	0.4615385	1.833333	11
4	{Extensor}	=> {Bicicleta}	0.4615385	0.9230769	0.5000000	1.714286	12
5	{Esteira, Extensor}	=> {Bicicleta}	0.3846154	0.9090909	0.4230769	1.688312	10
6	{Bicicleta, Esteira}	=> {Extensor}	0.3846154	1.0000000	0.3846154	2.000000	10

FONTE: O autor (2025).

APÊNDICE 8 – DEEP LEARNING

A – ENUNCIADO

1 Classificação de Imagens (CNN)

Implementar o exemplo de classificação de objetos usando a base de dados CIFAR10 e a arquitetura CNN vista no curso.

2 Detector de SPAM (RNN)

Implementar o detector de spam visto em sala, usando a base de dados SMS Spam e arquitetura de RNN vista no curso.

3 Gerador de Dígitos Fake (GAN)

Implementar o gerador de dígitos *fake* usando a base de dados MNIST e arquitetura GAN vista no curso.

4 Tradutor de Textos (Transformer)

Implementar o tradutor de texto do português para o inglês, usando a base de dados e a arquitetura Transformer vista no curso.

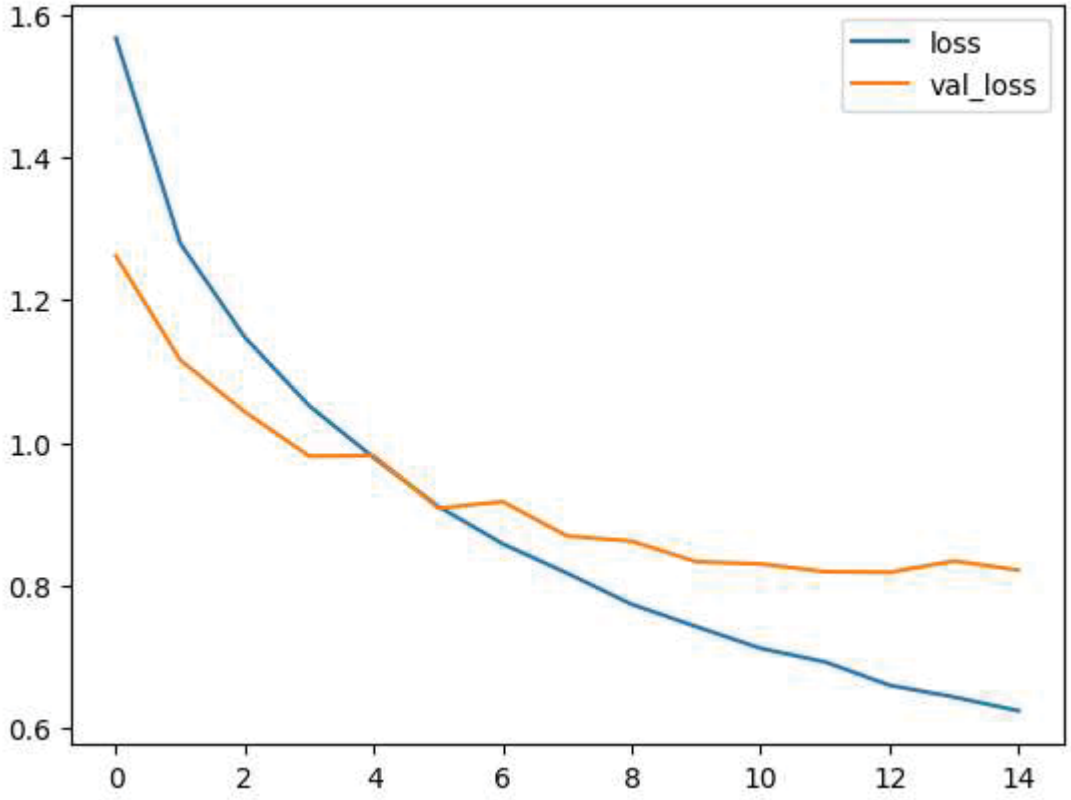
B – RESOLUÇÃO

1)

```
K = len(set(y_train))
# Estágio 1
i = Input(shape=x_train[0].shape)
x = Conv2D(32, (3, 3), strides=2, activation="relu")(i)
x = Conv2D(64, (3, 3), strides=2, activation="relu")(x)
x = Conv2D(128, (3, 3), strides=2, activation="relu")(x)

x = Flatten()(x)
# Estágio 2
x = Dropout(0.5)(x)
x = Dense(1024, activation="relu")(x)
x = Dropout(0.2)(x)
x = Dense(K, activation="softmax")(x)

model = Model(i, x)
```

```
model.compile(optimizer="adam",
loss="sparse_categorical_crossentropy", metrics=["accuracy"])
r = model.fit(x_train, y_train, validation_data=(x_test, y_test),
epochs=15)

# Plotar a função de perda, treino e validação
plt.plot(r.history["loss"], label="loss")
plt.plot(r.history["val_loss"], label="val_loss")
plt.legend()
plt.show()
```

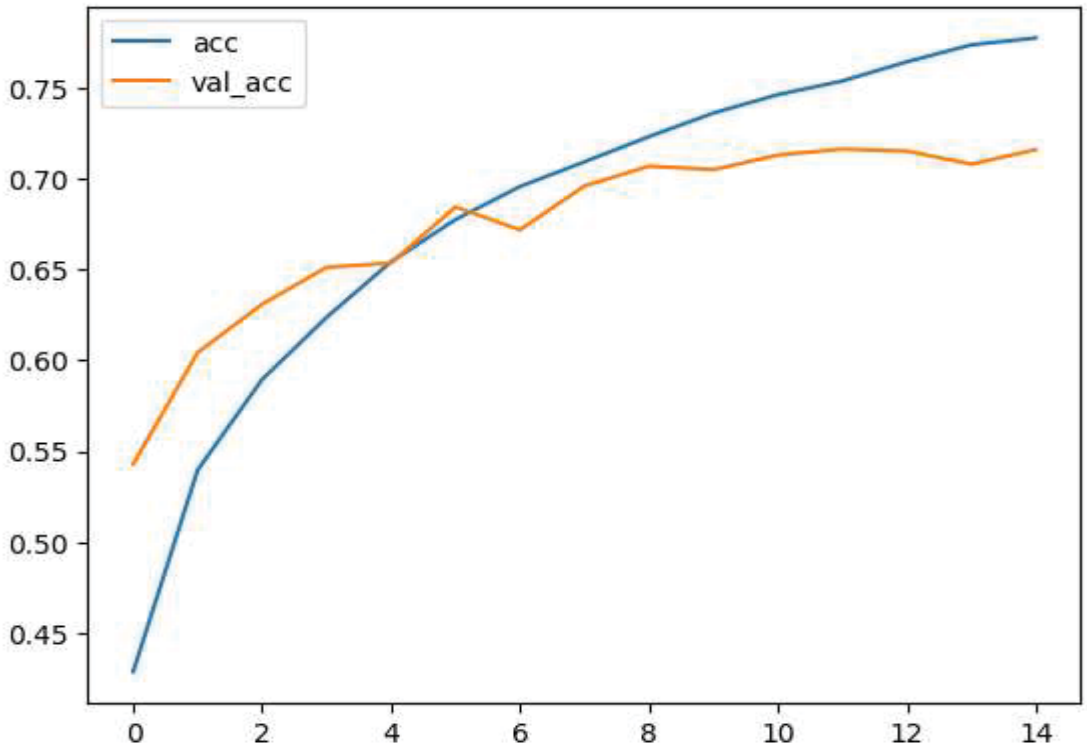
GRÁFICO 8 – EVOLUÇÃO DA FUNÇÃO DE PERDA E VALIDAÇÃO



FONTE: O autor (2025).

```
# Plotar acurácia, treino e validação
plt.plot(r.history["accuracy"], label="acc")
plt.plot(r.history["val_accuracy"], label="val_acc")
plt.legend()
plt.show()
```

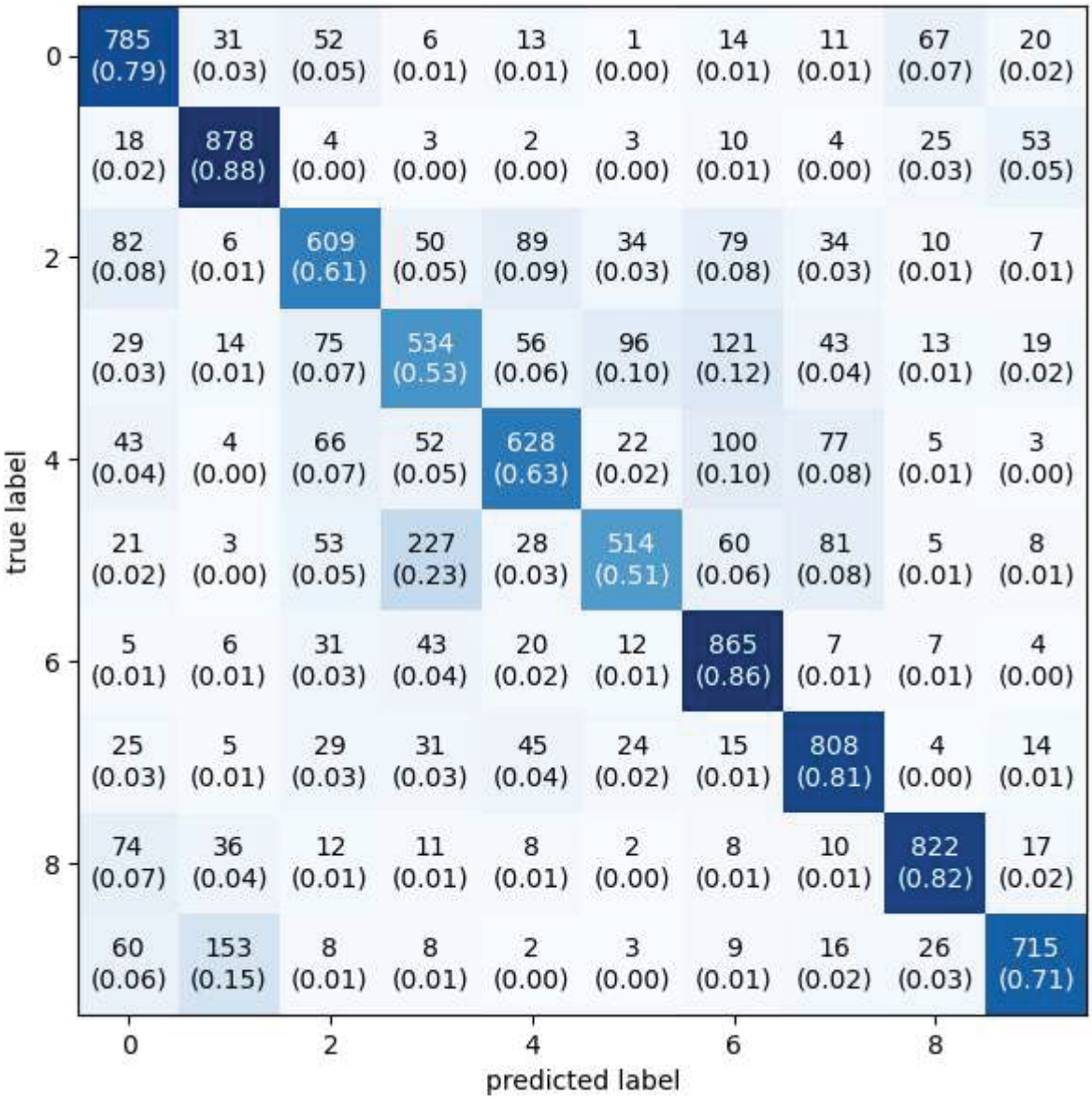
GRÁFICO 9 – EVOLUÇÃO DA ACURACIDADE E VALIDAÇÃO



Fonte: O autor (2025).

```
#Matriz de confusão
cm = confusion_matrix(y_test, y_pred)
plot_confusion_matrix(conf_mat=cm, figsize=(7, 7), show_normed=True)
```

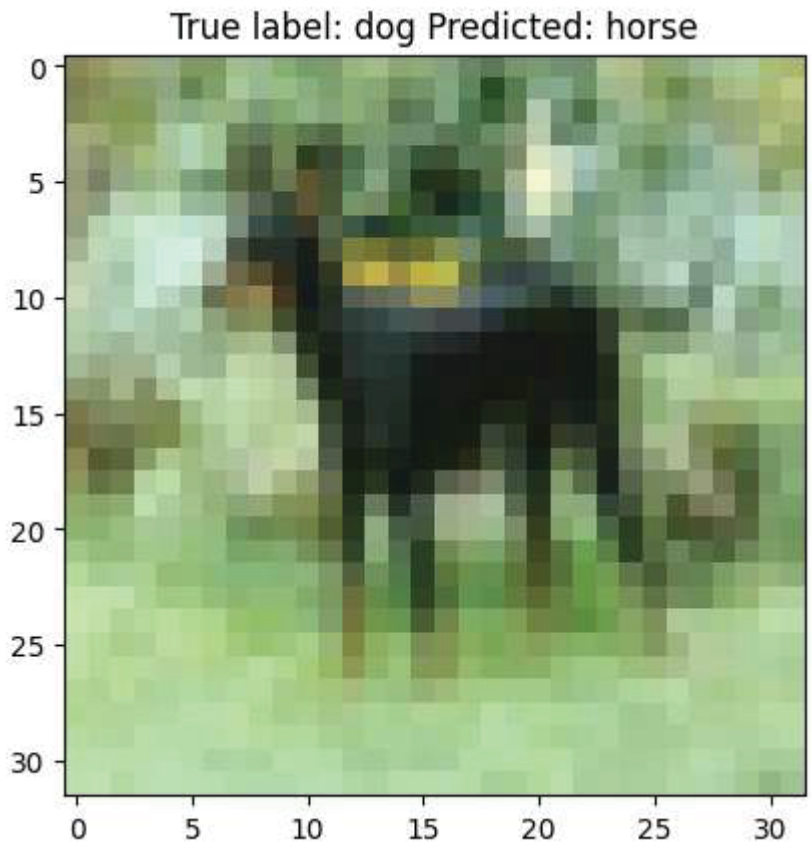
FIGURA 2 – MATRIZ DE CONFUSÃO



Fonte: O autor (2025).

```
# Mostrar uma classificação errada
labels= ["airplane", "automobile", "bird", "cat", "deer", "dog",
"frog", "horse", "ship", "truck"]
misclassified = np.where(y_pred != y_test)[0]
i = np.random.choice(misclassified)
plt.imshow(x_test[i], cmap="gray")plt.title("True label: %s
Predicted: %s" % (labels[y_test[i]], labels[y_pred[i]]))
```

FIGURA 3 – CLASSIFICAÇÃO INCORRETA



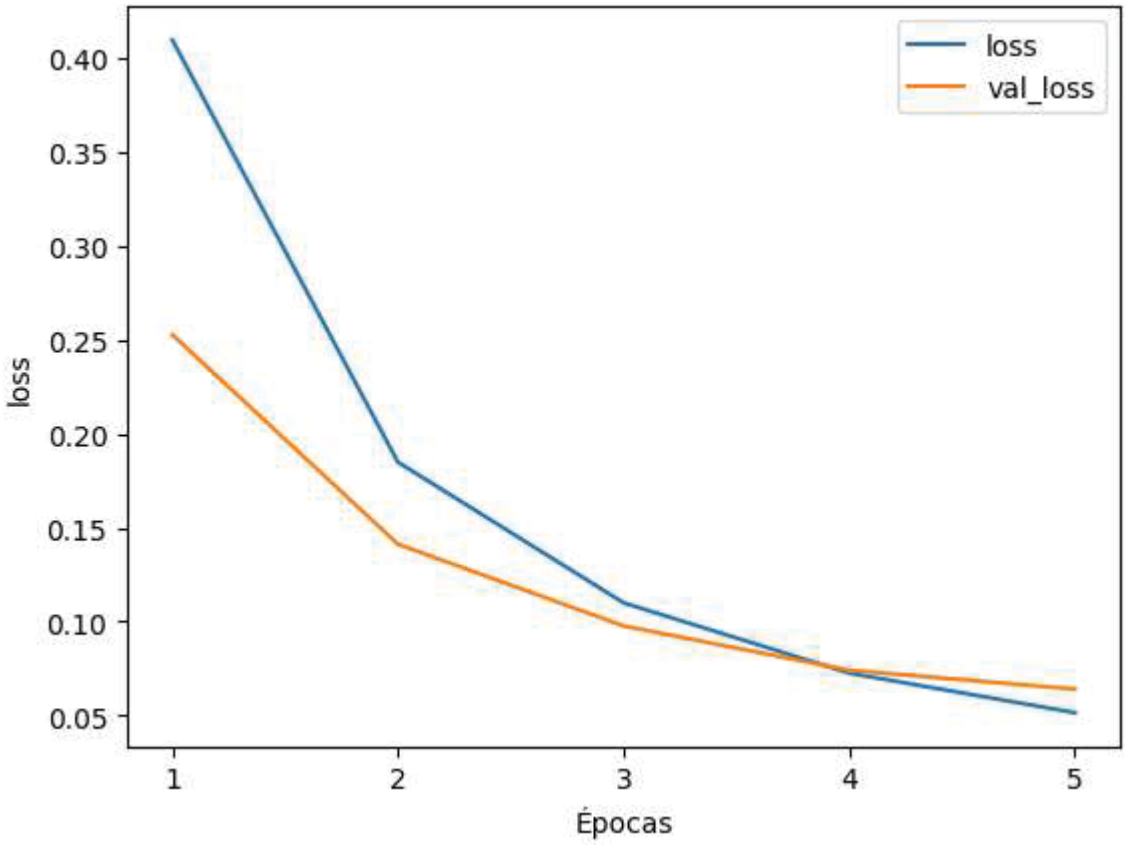
FONTE: O autor (2025).

2)

```
# Número máximo de palavras para considerar
# São consideradas as mais frequentes, as demais são ignoradas
num_words = 20000
tokenizer = Tokenizer(num_words=num_words)
tokenizer.fit_on_texts(x_train)
sequences_train = tokenizer.texts_to_sequences(x_train)
sequences_test = tokenizer.texts_to_sequences(x_test)
word2index = tokenizer.word_index
V = len(word2index)
print("%s tokens" % V)


# Define o modelo
D = 20 # tamanho do embedding, hiperparâmetro que pode ser escolhido
M = 5 # tamanho do hidden state, quantidade de unidades LSTM

i = Input(shape=(T,))
x = Embedding(V+1, D)(i)
x = LSTM(M)(x)
x = Dense(1, activation="sigmoid")(x)

model = Model(i, x)
```

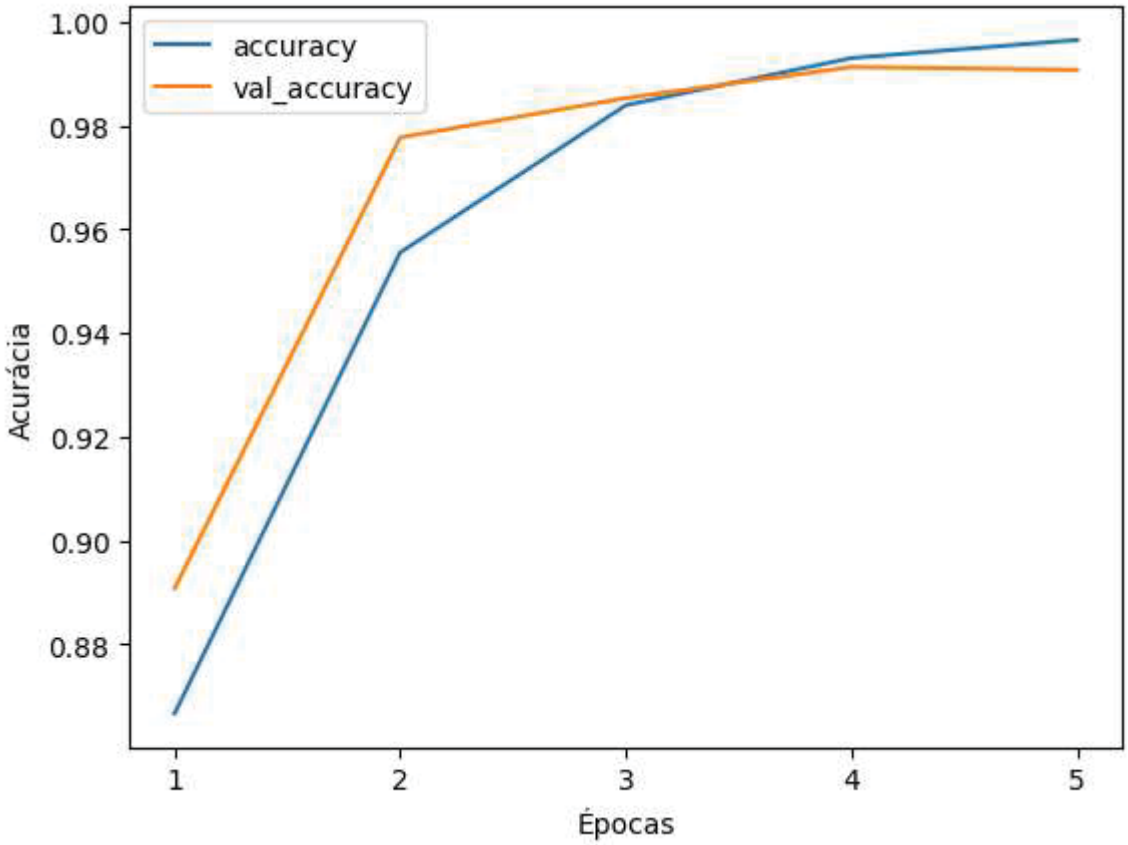
```
# Compilar e treinar o modelo
model.compile(loss="binary_crossentropy", optimizer="adam",
metrics=["accuracy"])

epochs = 5

r = model.fit(data_train, y_train, epochs=epochs,
validation_data=(data_test, y_test))

# Plota função de perda e acurácia
plt.plot(r.history["loss"], label="loss")
plt.plot(r.history["val_loss"], label="val_loss")
plt.xlabel("Épocas")
plt.ylabel("loss")
plt.xticks(np.arange(0, epochs, step=1), labels=range(1, epochs+1))
plt.legend()
plt.show()

plt.plot(r.history["accuracy"], label="accuracy")
plt.plot(r.history["val_accuracy"], label="val_accuracy")
plt.xlabel("Épocas")
plt.ylabel("Acurácia")
plt.xticks(np.arange(0, epochs, step=1), labels=range(1, epochs+1))
plt.legend()
plt.show()
```

GRÁFICO 10 – EVOLUÇÃO DA FUNÇÃO DE PERDA E VALIDAÇÃO



FONTE: O autor (2025).

GRÁFICO 11 – EVOLUÇÃO DA ACURACIDADE E VALIDAÇÃO



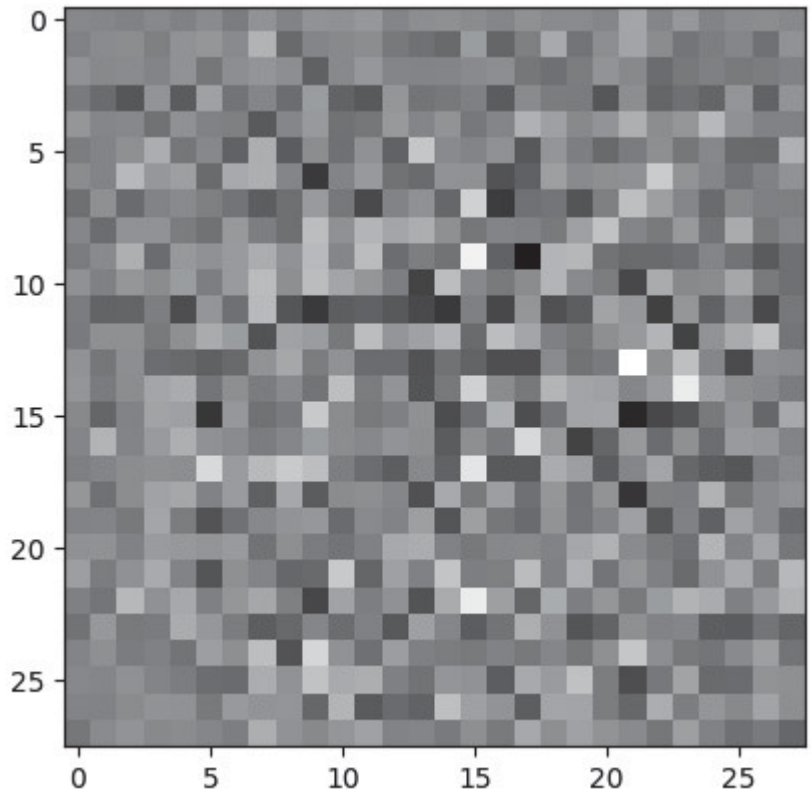

```
# Efetua a predição de um texto novo
#texto = "Hi, my name is Razer and want to tell you something."
texto = "Is your car dirty? Discover our new product. Free for all.
Click the link."
seq_texto = tokenizer.texts_to_sequences([texto]) # Tokeniza
data_texto = pad_sequences(seq_texto, maxlen=T) # Padding

pred = model.predict(data_texto) # Predição
print(pred)
print ("SPAM" if pred >= 0.5 else "OK")

1/1 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
 [[0.6188728]] SPAM
```

3)

```
# Cria o GERADOR
def make_generator_model():
  model = tf.keras.Sequential()
  model.add(layers.Dense(7*7*256, use_bias=False,
input_shape=(100,)))
  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())


  model.add(layers.Reshape((7, 7, 256)))
  assert model.output_shape == (None, 7, 7, 256)
```

```
  model.add(layers.Conv2DTranspose(128, (5, 5), strides=(1, 1),
padding='same', use_bias=False))
  assert model.output_shape == (None, 7, 7, 128)

  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())

  model.add(layers.Conv2DTranspose(64, (5, 5), strides=(2, 2),
padding='same', use_bias=False))
  assert model.output_shape == (None, 14, 14, 64)
  model.add(layers.BatchNormalization())
  model.add(layers.LeakyReLU())

  model.add(layers.Conv2DTranspose(1, (5, 5), strides=(2, 2),
padding='same', use_bias=False, activation='tanh'))
  assert model.output_shape == (None, 28, 28, 1)

  return model

# Teste do GERADOR, ainda não treinado
generator = make_generator_model()
noise = tf.random.normal([1, 100])
generated_image = generator(noise, training=False)
plt.imshow(generated_image[0, :, :, 0], cmap='gray')
```

FIGURA 4 – IMAGEM DE EXEMPLO DO GERADOR



Fonte: O autor (2025).

```
# Cria o DISCRIMINADOR
def make_discriminator_model():
  model = tf.keras.Sequential()
  model.add(layers.Conv2D(64, (5, 5), strides=(2, 2), padding='same',
input_shape=[28, 28, 1]))
  model.add(layers.LeakyReLU())
  model.add(layers.Dropout(0.3))

  model.add(layers.Conv2D(128, (5, 5), strides=(2, 2),
padding='same'))
  model.add(layers.LeakyReLU())
  model.add(layers.Dropout(0.3))

  model.add(layers.Flatten())
  model.add(layers.Dense(1))

  return model

# Gerar e salvar imagens
def generate_and_save_images(model, epoch, test_input):
  predictions = model(test_input, training=False)
  fig = plt.figure(figsize=(4, 4))

  for i in range(predictions.shape[0]):
    plt.subplot(4, 4, i+1)
    plt.imshow(predictions[i, :, :, 0] * 127.5 + 127.5, cmap='gray')
    plt.axis('off')

  plt.savefig('image_at_epoch_{:04d}.png'.format(epoch))
  plt.show()
```

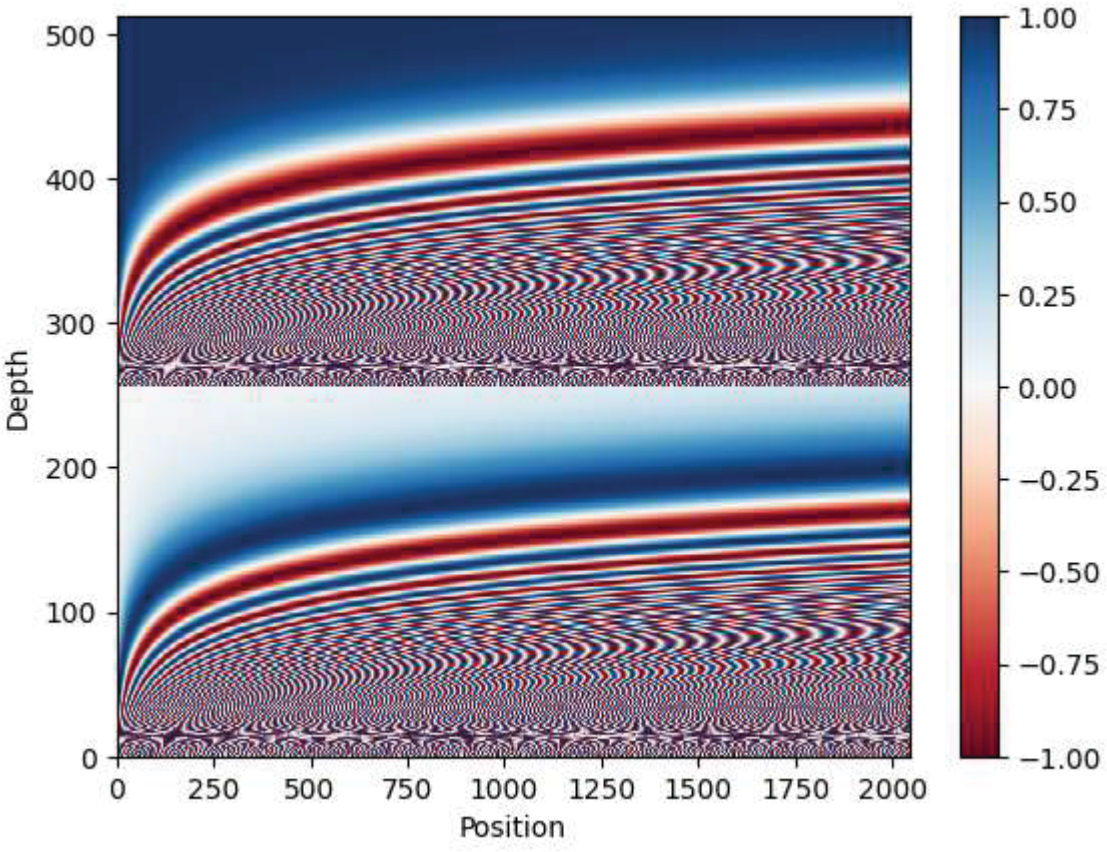
FIGURA 5 – CARACTERES GERADOS PELA REDE NEURAL ARTIFICIAL

Fonte: O autor (2025).

4)

```
# CODIFICAÇÃO POSICIONAL
n, d = 2048, 512
pos_encoding = positional_encoding(n, d)
print(pos_encoding.shape)
pos_encoding = pos_encoding[0]

# Arrumar as dimensões
pos_encoding = tf.reshape(pos_encoding, (n, d//2, 2))
pos_encoding = tf.transpose(pos_encoding, (2, 1, 0))
pos_encoding = tf.reshape(pos_encoding, (d, n))
plt.pcolormesh(pos_encoding, cmap='RdBu')
plt.ylabel('Depth')
plt.xlabel('Position')
plt.colorbar()
plt.show()
```

GRÁFICO 12 – RELAÇÃO ENTRE DEPTH E POSITION



Fonte: O autor (2025).

```
# Cria uma máscara de 0 e 1, 0 para quando há valor e 1 quando não há
def create_padding_mask(seq):
  seq = tf.cast(tf.math.equal(seq, 0), tf.float32)

  # add extra dimensions to add the padding
  # to the attention logits.
  return seq[:, tf.newaxis, tf.newaxis, :] # (batch_size, 1, 1,
seq_l)
```

```
# Máscara futura, usada no decoder
def create_look_ahead_mask(size):
  # zera o triângulo inferior
  mask = 1 - tf.linalg.band_part(tf.ones((size, size)), -1, 0)
  return mask # (seq_len, seq_len)

# Função de Atenção
def scaled_dot_product_attention(q, k, v, mask):
  # Q K^T
  matmul_qk = tf.matmul(q, k, transpose_b=True) # (..., seq_len_q,
seq_len_k)

  # converte matmul_qk para float32
  dk = tf.cast(tf.shape(k)[-1], tf.float32)

  # divide por sqrt(d_k)
  scaled_attention_logits = matmul_qk / tf.math.sqrt(dk)

  # Soma a máscara, e os valores faltantes serão um número próximo a
-inf if mask is not None:
  scaled_attention_logits += (mask * -1e9)

  # softmax normaliza os dados, soman 1. // (..., seq_len_q,
seq_len_k)
  attention_weights = tf.nn.softmax(scaled_attention_logits, axis=-1)

  output = tf.matmul(attention_weights, v) # (..., seq_len_q,
depth_v)

  return output, attention_weights

# Atenção Multi-cabeças
class MultiHeadAttention(tf.keras.layers.Layer):
  def __init__(self, d_model, num_heads):
    super(MultiHeadAttention, self).__init__()
    self.num_heads = num_heads
    self.d_model = d_model

    assert d_model % self.num_heads == 0

    self.depth = d_model // self.num_heads

    self.wq = tf.keras.layers.Dense(d_model)
    self.wk = tf.keras.layers.Dense(d_model)
    self.wv = tf.keras.layers.Dense(d_model)

    self.dense = tf.keras.layers.Dense(d_model)

  def split_heads(self, x, batch_size):
    """Separa a última dimensão em (num_heads, depth).
    Transpõe o resultado para o shape (batch_size, num_heads,
seq_len, depth)
    """
    x = tf.reshape(x, (batch_size, -1, self.num_heads, self.depth))
    return tf.transpose(x, perm=[0, 2, 1, 3])

  def call(self, v, k, q, mask):
```

```
    batch_size = tf.shape(q)[0]

    q = self.wq(q) # (batch_size, seq_len, d_model)
    k = self.wk(k) # (batch_size, seq_len, d_model)
    v = self.wv(v) # (batch_size, seq_len, d_model)

    q = self.split_heads(q, batch_size) # (batch_size, num_heads,
seq_len_q, depth)
    k = self.split_heads(k, batch_size) # (batch_size, num_heads,
seq_len_k, depth)
    v = self.split_heads(v, batch_size) # (batch_size, num_heads,
seq_len_v, depth)

    # Calcula a atenção para cada cabeça (de forma matricial)
    # scaled_attention.shape == (batch_size, num_heads, seq_len_q,
depth)
    # attention_weights.shape == (batch_size, num_heads, seq_len_q,
seq_len_k)
    scaled_attention, attention_weights =
scaled_dot_product_attention(q, k, v, mask)

    # Troca a dimensão 2 com 1, para acertar o num_heads
    # (batch_size, seq_len_q, num_heads, depth)
    scaled_attention = tf.transpose(scaled_attention, perm=[0, 2, 1,
3])

    # Concatena os valores em: (batch_size, seq_len_q, d_model)
    concat_attention = tf.reshape(scaled_attention, (batch_size, -1,
self.d_model))

    output = self.dense(concat_attention) # (batch_size, seq_len_q,
d_model)

    return output, attention_weights

def point_wise_feed_forward_network(d_model, dff):
  return tf.keras.Sequential([
      tf.keras.layers.Dense(dff, activation='relu'), # (batch_size,
seq_len, dff)
      tf.keras.layers.Dense(d_model) # (batch_size, seq_len, d_model)
  ])
class EncoderLayer(tf.keras.layers.Layer):
  def __init__(self, d_model, num_heads, dff, rate=0.1):
    super(EncoderLayer, self).__init__()

    self.mha = MultiHeadAttention(d_model, num_heads)
    self.ffn = point_wise_feed_forward_network(d_model, dff)

    self.layernorm1 = tf.keras.layers.LayerNormalization(epsilon=1e-
6)
    self.layernorm2 = tf.keras.layers.LayerNormalization(epsilon=1e-
6)

    self.dropout1 = tf.keras.layers.Dropout(rate)
    self.dropout2 = tf.keras.layers.Dropout(rate)

  def call(self, x, training, mask):
```

```
    attn_output, _ = self.mha(x, x, x, mask) # (batch_size,
input_seq_len, d_model)
    attn_output = self.dropout1(attn_output, training=training)
    out1 = self.layernorm1(x + attn_output) # (batch_size,
input_seq_len, d_model)

    ffn_output = self.ffn(out1) # (batch_size, input_seq_len,
d_model)
    ffn_output = self.dropout2(ffn_output, training=training)
    out2 = self.layernorm2(out1 + ffn_output) # (batch_size,
input_seq_len, d_model)

    return out2
class DecoderLayer(tf.keras.layers.Layer):
  def __init__(self, d_model, num_heads, dff, rate=0.1):
    super(DecoderLayer, self).__init__()

    self.mha1 = MultiHeadAttention(d_model, num_heads)
    self.mha2 = MultiHeadAttention(d_model, num_heads)

    self.ffn = point_wise_feed_forward_network(d_model, dff)

    self.layernorm1 = tf.keras.layers.LayerNormalization(epsilon=1e-
6)
    self.layernorm2 = tf.keras.layers.LayerNormalization(epsilon=1e-
6)
    self.layernorm3 = tf.keras.layers.LayerNormalization(epsilon=1e-
6)

    self.dropout1 = tf.keras.layers.Dropout(rate)
    self.dropout2 = tf.keras.layers.Dropout(rate)
    self.dropout3 = tf.keras.layers.Dropout(rate)

  def call(self, x, enc_output, training, look_ahead_mask,
padding_mask):
    # enc_output.shape == (batch_size, input_seq_len, d_model)

    # (batch_size, target_seq_len, d_model)
    attn1, attn_weights_block1 = self.mha1(x, x, x, look_ahead_mask)
    attn1 = self.dropout1(attn1, training=training)
    out1 = self.layernorm1(attn1 + x)

    # (batch_size, target_seq_len, d_model)
    attn2, attn_weights_block2 = self.mha2(enc_output, enc_output,
out1, padding_mask)
    attn2 = self.dropout2(attn2, training=training)
    out2 = self.layernorm2(attn2 + out1) # (batch_size,
target_seq_len, d_model)

    ffn_output = self.ffn(out2) # (batch_size, target_seq_len,
d_model)
    ffn_output = self.dropout3(ffn_output, training=training)
    out3 = self.layernorm3(ffn_output + out2) # (batch_size,
target_seq_len, d_model)

    return out3, attn_weights_block1, attn_weights_block2
class Encoder(tf.keras.layers.Layer):
```

```
  def __init__(self, num_layers, d_model, num_heads, dff,
               input_vocab_size, maximum_position_encoding,
rate=0.1):
    super(Encoder, self).__init__()

    self.d_model = d_model
    self.num_layers = num_layers
    self.embedding = tf.keras.layers.Embedding(input_vocab_size,
d_model)
    self.pos_encoding =
positional_encoding(maximum_position_encoding, self.d_model)
    self.enc_layers = [EncoderLayer(d_model, num_heads, dff, rate)
for _ in range(num_layers)]
    self.dropout = tf.keras.layers.Dropout(rate)

  def call(self, x, training, mask):
    seq_len = tf.shape(x)[1]
    # adding embedding and position encoding.
    x = self.embedding(x) # (batch_size, input_seq_len, d_model)
    x *= tf.math.sqrt(tf.cast(self.d_model, tf.float32))
    x += self.pos_encoding[:, :seq_len, :]
    x = self.dropout(x, training=training)

    for i in range(self.num_layers):
      x = self.enc_layers[i](x, training, mask)

    return x # (batch_size, input_seq_len, d_model)
class Decoder(tf.keras.layers.Layer):
  def __init__(self, num_layers, d_model, num_heads, dff,
target_vocab_size,
               maximum_position_encoding, rate=0.1):
    super(Decoder, self).__init__()

    self.d_model = d_model
    self.num_layers = num_layers

    self.embedding = tf.keras.layers.Embedding(target_vocab_size,
d_model)
    self.pos_encoding =
positional_encoding(maximum_position_encoding, d_model)

    self.dec_layers = [DecoderLayer(d_model, num_heads, dff, rate)
for _ in range(num_layers)]
    self.dropout = tf.keras.layers.Dropout(rate)

  def call(self, x, enc_output, training, look_ahead_mask,
padding_mask):
    seq_len = tf.shape(x)[1]
    attention_weights = {}

    x = self.embedding(x) # (batch_size, target_seq_len, d_model)
    x *= tf.math.sqrt(tf.cast(self.d_model, tf.float32))
    x += self.pos_encoding[:, :seq_len, :]

    x = self.dropout(x, training=training)

    for i in range(self.num_layers):
```

```
      x, block1, block2 = self.dec_layers[i](x, enc_output, training,
                                             look_ahead_mask,
padding_mask)
      attention_weights[f'decoder_layer{i+1}_block1'] = block1
      attention_weights[f'decoder_layer{i+1}_block2'] = block2

    # x.shape == (batch_size, target_seq_len, d_model)
    return x, attention_weights
class Transformer(tf.keras.Model):
  def __init__(self, num_layers, d_model, num_heads, dff,
input_vocab_size,
               target_vocab_size, pe_input, pe_target, rate=0.1):
    super().__init__()
    self.encoder = Encoder(num_layers, d_model, num_heads, dff,
input_vocab_size, pe_input, rate)
    self.decoder = Decoder(num_layers, d_model, num_heads, dff,
target_vocab_size, pe_target, rate)
    self.final_layer = tf.keras.layers.Dense(target_vocab_size)

  def call(self, inputs, training):
    # Keras models prefer if you pass all your inputs in the first
argument
    inp, tar = inputs

    enc_padding_mask, look_ahead_mask, dec_padding_mask =
self.create_masks(inp, tar)

    # (batch_size, inp_seq_len, d_model)
    enc_output = self.encoder(inp, training, enc_padding_mask)

    # dec_output.shape == (batch_size, tar_seq_len, d_model)
    dec_output, attention_weights = self.decoder(tar, enc_output,
training, look_ahead_mask, dec_padding_mask)

    # (batch_size, tar_seq_len, target_vocab_size)
    final_output = self.final_layer(dec_output)

    return final_output, attention_weights

  def create_masks(self, inp, tar):
    # Encoder padding mask
    enc_padding_mask = create_padding_mask(inp)

    # Used in the 2nd attention block in the decoder.
    # This padding mask is used to mask the encoder outputs.
    dec_padding_mask = create_padding_mask(inp)

    # Used in the 1st attention block in the decoder.
    # It is used to pad and mask future tokens in the input received
by
    # the decoder.
    look_ahead_mask = create_look_ahead_mask(tf.shape(tar)[1])
    dec_target_padding_mask = create_padding_mask(tar)
    look_ahead_mask = tf.maximum(dec_target_padding_mask,
look_ahead_mask)

    return enc_padding_mask, look_ahead_mask, dec_padding_mask
```

```
# Otimizador

class
CustomSchedule(tf.keras.optimizers.schedules.LearningRateSchedule):
  def __init__(self, d_model, warmup_steps=4000):
    super(CustomSchedule, self).__init__()
    self.d_model = d_model
    self.d_model = tf.cast(self.d_model, tf.float32)
    self.warmup_steps = warmup_steps

  def __call__(self, step):
    step = tf.cast(step, tf.float32) # Adicionado para evitar ERRO
    arg1 = tf.math.rsqrt(step)
    arg2 = step * (self.warmup_steps ** -1.5)
    return tf.math.rsqrt(self.d_model) * tf.math.minimum(arg1, arg2)

learning_rate = CustomSchedule(d_model)
optimizer = tf.keras.optimizers.Adam(learning_rate, beta_1=0.9,
beta_2=0.98, epsilon=1e-9)

# Processo de Treinamento
EPOCHS = 20
train_step_signature = [
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
]
@tf.function(input_signature=train_step_signature)
def train_step(inp, tar):
  tar_inp = tar[:, :-1]
  tar_real = tar[:, 1:]
  with tf.GradientTape() as tape:
    predictions, _ = transformer([inp, tar_inp], training = True)
    loss = loss_function(tar_real, predictions)
  gradients = tape.gradient(loss, transformer.trainable_variables)
  optimizer.apply_gradients(zip(gradients,
transformer.trainable_variables))
  train_loss(loss)
  train_accuracy(accuracy_function(tar_real, predictions))

class Translator(tf.Module):
  def __init__(self, tokenizers, transformer):
    self.tokenizers = tokenizers
    self.transformer = transformer
  def __call__(self, sentence, max_length=20):
    # input sentence is portuguese, hence adding the start and end
token
    assert isinstance(sentence, tf.Tensor)
    if len(sentence.shape) == 0:
      sentence = sentence[tf.newaxis]
    sentence = self.tokenizers.pt.tokenize(sentence).to_tensor()
    encoder_input = sentence
    # as the target is english, the first token to the transformer
should be the
    # english start token.
    start_end = self.tokenizers.en.tokenize([''])[0]
    start = start_end[0][tf.newaxis]
    end = start_end[1][tf.newaxis]
```

```
    output_array = tf.TensorArray(dtype=tf.int64, size=0,
dynamic_size=True)
    output_array = output_array.write(0, start)

    for i in tf.range(max_length):
      output = tf.transpose(output_array.stack())
      predictions, _ = self.transformer([encoder_input, output],
training=False)
      predictions = predictions[:, -1:, :] # (batch_size, 1,
vocab_size)
      predicted_id = tf.argmax(predictions, axis=-1)
      output_array = output_array.write(i+1, predicted_id[0])
      if predicted_id == end:
        break
    output = tf.transpose(output_array.stack())
    # output.shape (1, tokens)
    text = tokenizers.en.detokenize(output)[0]
    tokens = tokenizers.en.lookup(output)[0]
    _, attention_weights = self.transformer([encoder_input,
output[:,:-1]], training=False)
    return text, tokens, attention_weights
# Efetuar uma tradução
translator = Translator(tokenizers, transformer)
sentence = "Eu li sobre triceratops na enciclopédia."
translated_text, translated_tokens, attention_weights =
translator(tf.constant(sentence))
print(f'{"Prediction":15s}: {translated_text}')
Prediction : b'i read about trifuges in the encyclopedia .'
```

APÊNDICE 9 – BIG DATA

A – ENUNCIADO

Enviar um arquivo PDF contendo uma descrição breve (2 páginas) sobre a implementação de uma aplicação ou estudo de caso envolvendo Big Data e suas ferramentas (NoSQL e NewSQL). Caracterize os dados e Vs envolvidos, além da modelagem necessária dependendo dos modelos de dados empregados.

B – RESOLUÇÃO

1. Estudo de Caso: Sabe-se que a enorme produção de dados, hoje conhecido popularmente como o novo petróleo, pode gerar inúmeros insights e informações valiosas que a mente humana não consegue processar e assim obter conhecimentos sobre os mais variados assuntos. Pensando desta forma, um hospital pode extrair, aprender e gerar conteúdo muito mais positivo do que aquele gerado em anos de pesquisas e convenções das quais os médicos são convidados a participar. Mas como o hospital pode conhecer melhor os tratamentos feitos aos pacientes? Como é mensurado? Existe um feedback do paciente dizendo esse remédio funcionou? Ou o seu 'não retorno' consta como resultado válido? Essas respostas todas podem ser respondidas utilizando Big Data. A coleta de dados pelos médicos em grande quantidade, incluindo registros de sangue, qualidade de vida, exames, histórico de tratamentos, medicamentos receitados, acompanhamentos, pósconsulta, comentários em redes sociais… Todas essas informações analisadas geram uma base de dados rica e sólida (5Vs), sendo uma potencial ferramenta preditiva para auxiliar a melhora dos tratamentos convencionais. A solução está em utilizar Machine Learning. É possível prever crises de saúde, ataques cardíacos ou infecções; Com base em sinais vitais e histórico médico e familiar é possível prever, com mais eficiência, futuras doenças; e até personalizar os tratamentos. Com isso, a precisão dos diagnósticos pode reduzir as readmissões (retornos) hospitalares e ser mais assertivo quanto as prescrições, exames e acompanhamentos. Aplicado a rede pública de saúde (SUS), essa aplicação pode trazer vários benefícios desde um melhor controle da saúde pública, até um menor custo e consumo de insumos hospitalares.

2. Dados Processados: Para implementar uma solução de Big Data em um hospital, especialmente utilizando Machine Learning para melhorar tratamentos e prever crises de saúde, são necessárias diversas ferramentas e tecnologias. Desta forma, os dados a serem processados segundo as fontes, caracterização e objetivos das ferramentas aplicadas são:

- Nome, idade, sexo, peso
  Tipo de Dados: Estruturados
  Vs: Volume e Velocidade no acesso

- Histórico de doenças e Tratamentos
  Tipo de Dados: Estruturados, Temporais e Interação
  Vs: Volume e Veracidade

- Registros de exames de sangue
  Tipo de Dados: Estruturados e Temporais
  Vs: Volume e Velocidade no acesso

- Exames de Ressonância e Raio-X
  Tipo de Dados: Estruturados e Temporais
  Vs: Volume e Velocidade no acesso

- Relatório de Consultas
  Tipo de Dados: Estruturados e Interação
  Vs: Volume e Velocidade no acesso

- Medicamentos Receitados
  Tipo de Dados: Estruturados
  Vs: Volume

- Notas Médicas
  Tipo de Dados: Estruturados e Semi-Estruturados
  Vs: Volume e Veracidade

- Prescrições (Receitas)
  Tipo de Dados: Estruturados e Semi-Estruturados
  Vs: Volume e Velocidade

- Relatório de Consultas
  Tipo de Dados: Semi-Estruturados
  Vs: Volume, Velocidade e Veracidade

- Pós-consulta - Comentários em Mídias Sociais
  Tipo de Dados: Não-Estruturados e Interação
  Vs: Volume, Velocidade, Variedade, Veracidade e Valor

- Imagens de Ressonância e Raio-X
  Tipo de Dados: Não-Estruturados
  Vs: Volume, Variedade e Valor

- Sensores diversos - Dados de equipamentos médicos (monitores, elétrons)
  Tipo de Dados: Não-Estruturados e Temporais
  Vs: Volume, Variedade e Valor

- Transcrições de conversas – Médico → Paciente
  Tipo de Dados: Estruturados, Semi-Estruturados, Temporais e Interação
  Vs: Variedade e Valor

3. Ferramentas

Objetivos para Alcance

- Sistemas de Registro Eletrônico de Saúde (EHR): Ferramentas como Epic, Cerner, ou sistemas personalizados de EHR, que coletam e armazenam dados de pacientes de maneira estruturada.
- Plataformas de Big Data: Soluções como Hadoop e Spark para processar e analisar grandes volumes de dados. O Hadoop é útil para armazenar grandes quantidades de dados distribuídos, enquanto o Spark oferece processamento em tempo real e análise de dados com suporte para Machine Learning.
- Bancos de Dados NoSQL: Ferramentas como MongoDB ou Cassandra são essenciais para armazenar dados não estruturados e semiestruturados, como notas médicas, relatórios de exames e históricos de tratamentos.
- Plataformas de Machine Learning: Bibliotecas e frameworks como TensorFlow, ScikitLearn, ou PyTorch são fundamentais para desenvolver e treinar modelos preditivos que auxiliem na personalização de tratamentos e na previsão de crises de saúde.
- Ferramentas de Visualização: Soluções como Tableau, Power BI ou mesmo dashboards customizados em Python ou R para apresentar os dados analisados de forma compreensível e acessível aos médicos e administradores.
- Ferramentas de Integração e ETL: Apache Nifi ou Talend podem ser usadas para extrair, transformar e carregar dados de diversas fontes, integrando dados de EHRs, sistemas de laboratório, e outros repositórios.

4. Da modelagem de dados

A modelagem de dados em um hospital utilizando Big Data pode variar dependendo do tipo de dados e do objetivo da análise.

- Modelo Relacional: Pode ser usado para armazenar informações estruturadas como registros de pacientes, tratamentos prescritos, e resultados de exames. Bancos de dados relacionais como MySQL ou PostgreSQL são úteis para consultas complexas e integrações com outros sistemas hospitalares.

- Modelo Chave-Valor: Adequado para armazenar informações de acesso rápido, como consultas recentes dos pacientes ou dados de sensores médicos em tempo real. Ferramentas como Redis são frequentemente utilizadas para esse tipo de modelagem.
- Modelo Documento: Ideal para armazenar registros médicos eletrônicos, relatórios de consultas, e anotações clínicas, onde cada documento pode representar um paciente com todas as suas informações de saúde. MongoDB é uma escolha comum para esse tipo de modelagem.
- Modelo Colunar: Utilizado para grandes volumes de dados analíticos, como dados históricos de pacientes, para identificar padrões e tendências. Apache Cassandra é uma opção popular para essa modelagem, especialmente em análises de larga escala.
- Modelo de Grafo: Pode ser usado para mapear relações complexas entre sintomas, doenças, tratamentos e pacientes, facilitando a análise de interações e concorrências. Neo4j é um exemplo de banco de dados de grafos que pode ser utilizado nesse contexto.

Essas ferramentas, caracterizações, Vs e modelagens formam a base para a implementação eficaz de Big Data em um ambiente hospitalar, com o objetivo de melhorar a qualidade dos cuidados de saúde e a eficiência operacional.

APÊNDICE 10 – VISÃO COMPUTACIONAL

A – ENUNCIADO

1) Extração de Características

Os bancos de imagens fornecidos são conjuntos de imagens de 250x250 pixels de imuno-histoquímica (biópsia) de câncer de mama. No total são 4 classes (0, 1+, 2+ e 3+) que estão divididas em diretórios. O objetivo é classificar as imagens nas categorias correspondentes. Uma base de imagens será utilizada para o treinamento e outra para o teste do treino.

As imagens fornecidas são recortes de uma imagem maior do tipo WSI *(Whole Slide Imaging*) disponibilizada pela Universidade de Warwick (link). A nomenclatura das imagens segue o padrão XX_HER_YYYY.png, onde XX é o número do paciente e YYYY é o número da imagem recortada. Separe a base de treino em 80% para treino e 20% para validação. **Separe por pacientes (XX), não utilize a separação randômica! Pois, imagens do mesmo paciente não podem estar na base de treino e de validação, pois isso pode gerar um viés.** No caso da CNN VGG16 remova a última camada de classificação e armazene os valores da penúltima camada como um vetor de características. Após o treinamento, os modelos treinados devem ser validados na base de teste.

Tarefas:

a) Carregue a base de dados de **Treino.**
b) Crie partições contendo 80% para treino e 20% para validação (atenção aos pacientes).
c) Extraia características utilizando LBP e a CNN VGG16 (gerando um csv para cada extrator).
d) Treine modelos Random Forest, SVM e RNA para predição dos dados extraídos.
e) Carregue a base de **Teste** e execute a tarefa 3 nesta base.
f) Aplique os modelos treinados nos dados de treino
g) Calcule as métricas de Sensibilidade, Especificidade e F1-Score com base em suas matrizes de confusão.
h) Indique qual modelo dá o melhor o resultado e a métrica utilizada

2) Redes Neurais

Utilize as duas bases do exercício anterior para treinar as Redes Neurais Convolucionais VGG16 e a Resnet50. Utilize os pesos pré-treinados (*Transfer Learning*), refaça as camadas *Fully Connected* para o problema de 4 classes. Compare os treinos de 15 épocas com e sem *Data Augmentation*. Tanto a VGG16 quanto a Resnet50 têm como camada de entrada uma imagem 224x224x3, ou seja, uma imagem de 224x224 pixels coloridos (3 canais de cores). Portanto, será necessário fazer uma transformação de 250x250x3 para 224x224x3. Ao fazer o *Data Augmentation* **cuidado** para não alterar demais as cores das imagens e atrapalhar na classificação.

Tarefas:

a) Utilize a base de dados de **Treino** já separadas em treino e validação do exercício anterior
b) Treine modelos VGG16 e Resnet50 adaptadas com e sem *Data Augmentation*
c) Aplique os modelos treinados nas imagens da base de **Teste**
d) Calcule as métricas de Sensibilidade, Especificidade e F1-Score com base em suas matrizes de confusão.
e) Indique qual modelo dá o melhor o resultado e a métrica utilizada

B – RESOLUÇÃO

1)

```
--- Métricas para Random Forest com LBP no CONJUNTO DE TESTE ---

              precision    recall  f1-score   support

           0     0.8721    0.7426    0.8021       101
           1     0.5259    0.6778    0.5922        90
           2     0.5641    0.4889    0.5238        90
           3     0.9890    1.0000    0.9945        90

    accuracy                         0.7278       371
   macro avg     0.7378    0.7273    0.7282       371
weighted avg     0.7417    0.7278    0.7304       371


--- Métricas para SVM com LBP no CONJUNTO DE TESTE ---

              precision    recall  f1-score   support

           0     0.9565    0.6535    0.7765       101
           1     0.5197    0.7333    0.6083        90
           2     0.5647    0.5333    0.5486        90
           3     0.9889    0.9889    0.9889        90

    accuracy                         0.7251       371
   macro avg     0.7575    0.7273    0.7306       371
weighted avg     0.7634    0.7251    0.7319       371

--- Métricas para RNA com LBP no CONJUNTO DE TESTE ---
              precision    recall  f1-score   support

           0     0.9718    0.6832    0.8023       101
           1     0.5464    0.5889    0.5668        90
           2     0.5766    0.7111    0.6368        90
           3     0.9783    1.0000    0.9890        90

    accuracy                         0.7439       371
   macro avg     0.7683    0.7458    0.7487       371
weighted avg     0.7743    0.7439    0.7503       371
```

```
--- Métricas para Random Forest com VGG16 no CONJUNTO DE TESTE ---
              precision    recall  f1-score   support

           0     0.8000    0.9901    0.8850       101
           1     0.6957    0.3556    0.4706        90
           2     0.6881    0.8333    0.7538        90
           3     0.9780    0.9889    0.9834        90

    accuracy                         0.7978       371
   macro avg     0.7904    0.7920    0.7732       371
weighted avg     0.7907    0.7978    0.7765       371


--- Métricas para SVM com VGG16 no CONJUNTO DE TESTE ---
              precision    recall  f1-score   support

           0     0.8807    0.9505    0.9143       101
           1     0.7073    0.3222    0.4427        90
           2     0.6167    0.8222    0.7048        90
           3     0.8911    1.0000    0.9424        90

    accuracy                         0.7790       371
   macro avg     0.7740    0.7737    0.7511       371
weighted avg     0.7771    0.7790    0.7559       371

--- Métricas para RNA com VGG16 no CONJUNTO DE TESTE ---
              precision    recall  f1-score   support

           0     0.8142    0.9109    0.8598       101
           1     0.6061    0.2222    0.3252        90
           2     0.5923    0.8556    0.7000        90
           3     0.9368    0.9889    0.9622        90

    accuracy                         0.7493       371
   macro avg     0.7373    0.7444    0.7118       371
weighted avg     0.7396    0.7493    0.7162       371
```

O melhor modelo foi o SVM utilizando as características extraídas com VGG16. A métrica usada para essa conclusão foi a Sensibilidade, que é importante em contextos onde não identificar um positivo pode ter consequências graves (como em diagnósticos médicos, por exemplo).

2)

O melhor resultado encontrado foi VGG16 com Data Augmentation. Para essa conclusão foi considerada a métrica de Sensibilidade, que é importante em contextos onde não identi car um positivo pode ter consequências graves. Além disso, nas outras duas métricas veri cadas, este mesmo modelo também apresentou o melhor resultado.

APÊNDICE 11 – ASPECTOS FILOSÓFICOS E ÉTICOS DA IA

A – ENUNCIADO

Título do Trabalho: "Estudo de Caso: Implicações Éticas do Uso do ChatGPT"

Trabalho em Grupo: O trabalho deverá ser realizado em grupo de alunos de no máximo seis (06) integrantes.

Objetivo do Trabalho: Investigar as implicações éticas do uso do ChatGPT em diferentes contextos e propor soluções responsáveis para lidar com esses dilemas.
Parâmetros para elaboração do Trabalho:

1. Relevância Ética: O trabalho deve abordar questões éticas significativas relacionadas ao uso da inteligência artificial, especialmente no contexto do ChatGPT. Os alunos devem identificar dilemas éticos relevantes e explorar como esses dilemas afetam diferentes partes interessadas, como usuários, desenvolvedores e a sociedade em geral.
2. Análise Crítica: Os alunos devem realizar uma análise crítica das implicações éticas do uso do ChatGPT em estudos de caso específicos. Eles devem examinar como o algoritmo pode influenciar a disseminação de informações, a privacidade dos usuários e a tomada de decisões éticas. Além disso, devem considerar possíveis vieses algorítmicos, discriminação e questões de responsabilidade.
3. Soluções Responsáveis: Além de identificar os desafios éticos, os alunos devem propor soluções responsáveis e éticas para lidar com esses dilemas. Isso pode incluir sugestões para políticas, regulamentações ou práticas de design que promovam o uso responsável da inteligência artificial. Eles devem considerar como essas soluções podem equilibrar os interesses de diferentes partes interessadas e promover valores éticos fundamentais, como transparência, justiça e privacidade.
4. Colaboração e Discussão: O trabalho deve envolver discussões em grupo e colaboração entre os alunos. Eles devem compartilhar ideias, debater diferentes pontos de vista e chegar a conclusões informadas através do diálogo e da reflexão mútua. O estudo de caso do ChatGPT pode servir como um ponto de partida para essas discussões, incentivando os alunos a aplicar conceitos éticos e legais aprendidos ao analisar um caso concreto.
5. Limite de Palavras: O trabalho terá um limite de 6 a 10 páginas teria aproximadamente entre 1500 e 3000 palavras.
6. Estruturação Adequada: O trabalho siga uma estrutura adequada, incluindo introdução, desenvolvimento e conclusão. Cada seção deve ocupar uma parte proporcional do total de páginas, com a introdução e a conclusão ocupando menos espaço do que o desenvolvimento.
7. Controle de Informações: Evitar incluir informações desnecessárias que possam aumentar o comprimento do trabalho sem contribuir significativamente para o conteúdo. Concentre-se em informações relevantes, argumentos sólidos e evidências importantes para apoiar sua análise.

8. Síntese e Clareza: O trabalho deverá ser conciso e claro em sua escrita. Evite repetições desnecessárias e redundâncias. Sintetize suas ideias e argumentos de forma eficaz para transmitir suas mensagens de maneira sucinta.
9. Formatação Adequada: O trabalho deverá ser apresentado nas normas da ABNT de acordo com as diretrizes fornecidas, incluindo margens, espaçamento, tamanho da fonte e estilo de citação. Deve-se seguir o seguinte template de arquivo: hfps://bibliotecas.ufpr.br/wp-content/uploads/2022/03/template-artigo-de-periodico.docx

B – RESOLUÇÃO

Casos de uso de ChatGPT e suas dimensões éticas: uma análise crítica

RESUMO

Inteligência Artificial (IA) é definida como o campo que emprega recursos científicos e de engenharia para atribuir inteligência a máquinas e programas de computador, capacitando-os a realizar tarefas anteriormente exclusivas aos humanos (BOLANDER, 2019). Uma aplicação recente da IA é o Processamento de Linguagem Natural (NLP), exemplificado pelos Large Language Models (LLMs), que são treinados com extensos volumes de texto para prever palavras em sentenças parciais (DEVLIN et al., 2018). LLMs, como ChatGPT da OpenAI e Gemini do Google, são usados em tradução, classificação, escrita criativa e criação de código (ELOUNDOU et al., 2023). Desde seu lançamento, o ChatGPT tem sido utilizado por milhões para tarefas como agendamento, lembretes e gerenciamento de listas, além de melhorar campanhas de marketing, reduzir custos e gerar insights em empresas e organizações (AYINDE et al., 2023). Há usos possíveis também na educação, como na escrita de artigos e ensaios acadêmicos, por exemplo (MOLLICK; MOLLICK, 2022). No entanto, o crescimento do uso do ChatGPT destaca seus impactos sociais, econômicos, éticos e culturais, especialmente em relação aos vieses incorporados nos modelos (VIDHYA; DEVI; A.; MANJU, 2023). Este trabalho apresenta estudos de caso sobre o uso do ChatGPT, analisando criticamente seus dilemas éticos e propondo soluções com base na literatura especializada.
Palavras-chave: Estudo de Caso. Ética. Inteligência Artificial. ChatGPT. Interação Humano-Máquina.

ABSTRACT

Artificial Intelligence (AI) is defined as the field that employs scientific and engineering resources to attribute intelligence to machines and computer programs, enabling them to perform tasks previously exclusive to humans (BOLANDER, 2019). A recent application of AI is Natural Language Processing (NLP), exemplified by Large Language Models (LLMs), which are trained with extensive volumes of text to predict words in partial sentences (DEVLIN et al., 2018). LLMs, such as OpenAI's ChatGPT and Google's Gemini, are used in translation, classification, creative writing, and code generation (ELOUNDOU et al., 2023). Since its launch, ChatGPT has been utilized by millions for tasks like

scheduling, reminders, and list management, as well as improving marketing campaigns, reducing costs, and generating insights in businesses and organizations (AYINDE et al., 2023). There are 3 also possible uses in education, such as writing articles and academic essays, for example (MOLLICK; MOLLICK, 2022). However, the growing use of ChatGPT highlights its social, economic, ethical, and cultural impacts, especially concerning the biases incorporated into the models (VIDHYA; DEVI; A.; MANJU, 2023). This work presents case studies on the use of ChatGPT, critically analyzing its ethical dilemmas and proposing solutions based on specialized literature. Keywords: Case Study. Ethics. Artificial Intelligence. ChatGPT. Human-Machine Interaction.

1 INTRODUÇÃO

Há mais de 60 anos, John McCarthy definiu Inteligência Artificial (AI) como sendo o campo capaz de empregar recursos científicos e de engenharia para atribuir inteligência (ou comportamentos inteligentes) a máquinas e programas de computador. Embora essa definição tenha problemas crônicos e inerentes, o objetivo deste campo quase sempre se traduz no esforço de construir computadores ou robôs capazes de realizar tarefas que antes apenas humanos teriam capacidade de executar (BOLANDER, 2019). Preenchendo algumas das lacunas nessa definição, Nilsson (2009) afirma que AI é "[...] é a atividade dedicada a tornar as máquinas inteligentes, e inteligência é a qualidade que permite a uma entidade funcionar apropriadamente e com perspicácia em seu ambiente".

Uma das aplicações recentes derivadas dos estudos de Inteligência Artificial e de Aprendizado de Máquina, pertencente a subcategoria de aplicações de Processamento de Linguagem Natural (NLP) são os chamados Large Language Models (LLM), ou no português Grandes Modelos de Linguagem. Esses modelos são treinados através de extensos volumes de dados de texto e são capazes de prever, de forma auto supervisionada, uma palavra em uma sentença parcial (DEVLIN et al., 2018). Implementados em diferentes soluções (como o ChatGPT, da OpenAI, ou o Gemini, do Google), estes modelos passaram a ser utilizados para diferentes tarefas, como tradução, classificação, escrita criativa e criação de código. LLMs podem processar e produzir várias outras formas de dados sequenciais, não se limitando ao processamento de linguagem natural (ELOUNDOU et al., 2023). Lançado ao público há pouco tempo, o ChatGPT já é utilizado por milhões de pessoas em todo o mundo nas mais diversas tarefas como agendamento de 4 compromissos, definindo lembretes e gerenciando listas de afazeres (BISWAS, 2023b). Além de tarefas de uso geral, o ChatGPT também pode ser utilizado de uma ampla gama de atividades e empreendimentos humanos tais como: em empresas e organizações governamentais ou do terceiro setor, melhorando campanhas de marketing, reduzindo custos, gerando insights de comportamento de consumidores (AYINDE et al., 2023); na educação, por meio da escrita de artigos e ensaios acadêmicos, elaboração de perguntas e questionários, além de ampliar o escopo de possibilidades de interação entre alunos e professores em sala de aula (MOLLICK; MOLLICK, 2022).

No entanto, conforme o uso do ChatGPT cresce e se populariza, tornam-se cada vez mais evidentes seus impactos sociais, econômicos, éticos e culturais. Na medida em que inputs e dados dos usuários são incorporados aos modelos e algoritmos da ferramenta, vieses dos mais variados tipos também podem ser reforçados, a depender dos objetivos corporativos e da conduta dos grupos

empresariais que administram essas ferramentas, o que lança um grande desafio na fronteira dos estudos da interação humano-máquina (VIDHYA; DEVI; A.; MANJU, 2023).

No presente trabalho, são apresentados estudos de caso que envolvam o uso e aplicações de ChatGPT em diferentes contextos e atividades, com ênfase especial na análise crítica das dimensões e dilemas éticos contidos em cada um dos casos analisados. Em seguida, serão listadas, com base na literatura especializada sobre o tema, possíveis soluções para os problemas apontados nos estudos de caso. Por fim, são apresentadas as considerações finais.

2 REVISÃO DE LITERATURA

O ChatGPT é uma ferramenta de Inteligência Artificial Generativa, que são tecnologias capazes de gerar novos conteúdos (textos, códigos, imagens e vídeos) a partir de entradas de dados do usuário, também conhecidas como “prompts”. Essas ferramentas são treinadas com base em uma grande variedade de dados obtidos pela Internet, por meio de uma arquitetura de Redes Neurais Artificiais de tipo Transformer. A ideia de sistemas de IA generativos, por meio do qual seja possível interagir com máquinas por meio de texto e linguagem natural não é nova, e remonta ao início das pesquisas no campo da IA e computação (STAHL; EKE, 2024).

O notório avanço nas pesquisas e, consequentemente nas aplicações que envolvam algum grau de Processamento de Linguagem Natural (NLP), presenciados nos últimos anos tiveram por consequência importantes marcos para acadêmicos e profissionais da indústria da área Interação Humano-Máquina, como por exemplo o comportamento inadequado do bot Tay da Microsoft no Twitter; as infrações de privacidade da Alexa, da Amazon, além das limitações e vieses presentes no próprio ChatGPT (HIRSCHBERG; MANNING, 2015).

No que se refere especificamente ao ChatGPT, diversas pesquisas acadêmicas já foram conduzidas demonstrando algumas das falhas apresentadas pelo modelo, entre as quais destacam-se: criação de conteúdo falso (alucinações), reiteração dos pontos de vista apresentados por um usuário; reforço de vieses de caráter discriminatório presentes nos dados de treinamento do modelo; e perpetuação de comportamentos que humanos possam tentar evitar em razão de potenciais riscos (EIGNER; HÄNDLER, 2024).

Em razão do extenso uso e atenção que ferramentas como o ChatGPT vem recebendo atualmente, vários estudos têm sido conduzidos para explorar os potenciais benefícios no uso dessa tecnologia, e dilemas éticos que surgem por este motivo. As pesquisas que se dedicam sobre a temática da ética em aplicações de LLM abrangem desde a detecção de condutas imorais até a presença de vieses de cunho discriminatório (VIDHYA, 2023). No entanto, de acordo com Stahl e Eke (2024), muitas dessas pesquisas têm focado em questões específicas, e deixam a desejar no que diz respeito à capacidade de considerar, simultaneamente, benefícios advindos da ferramenta e preocupações éticas resultantes. O autor também propõe que é de interesse coletivo que esse reequilíbrio aconteça, de modo a beneficiar toda a comunidade interessada em tópicos ligados a IA e LLMs. Considerando o contexto apresentado, e as considerações levantadas por autores e

especialistas da área, são apresentados à seguir alguns estudos de caso que demonstram os dilemas éticos que surgem no uso da ferramenta.

2.1 CHATGPT E O USO NA EDUCAÇÃO

Diversos estudos e publicações destacam os potenciais benefícios do ChatGPT na educação, sugerindo até mesmo diretrizes para sua aplicação em sala de aula. No entanto, as preocupações relacionadas aos chatbots, como a falta de recursos, inadequação tecnológica para tarefas específicas, regulações insuficientes e questões de segurança de dados, ainda não foram suficientemente exploradas. Especificamente, não está claro se o ChatGPT, um chatbot avançado, conseguirá superar essas questões ou se agravará os problemas já presentes em outras soluções de chatbot, o que poderia resultar em reações protetivas rápidas e severas, como a proibição do ChatGPT em redes educacionais de grandes cidades devido ao risco de facilitar trapaças em tarefas escolares (Shen-Berro, 2023).

O estudo conduzido por Tlili et. al (2023) realizou entrevistas com diferentes stakeholders de uma escola (estudantes, professores, diretores) a fim de entender melhor sua relação com a ferramenta. Em geral, todos os entrevistados tinham alguma familiaridade e experiência prévia com o uso de chatbots no contexto educacional. O foco da pesquisa foi compreender a experiência de usuário destes stakeholders com a ferramenta. A pesquisa então aponta 10 cenários que levantam preocupações de cunho ético e educacional.

Um dos cenários apresentados é a possibilidade real de alunos trapacearem em atividades como escrita de artigos ou mesmo ao responder questionários de múltipla escolha. É possível submeter todas essas atividades a uma ferramenta como o ChatGPT e obter respostas relativamente acuradas. Isso se torna mais crítico ainda porque é possível fugir de ferramentas de detecção de plágio de conteúdo gerado por IA simplesmente adicionando algumas palavras ao conteúdo gerado, por exemplo.

Outros cenários apontados na pesquisa demonstram a opacidade nos dados de treinamento desses modelos. Em muitos casos, a resposta gerada à partir de uma pergunta feita pelo usuário era inadequada ou mesmo errada. Em outras situações, a resposta gerada pelo modelo era completamente diferente, ainda que diferentes usuários fizessem exatamente a mesma pergunta. Nesse sentido, outra preocupação que aparece é quanto a clareza sobre como os dados das conversas com os usuários são coletados, armazenados e utilizados pela OpenAI para retreinar seus modelos. Em sua política de privacidade, há menção a essa coleta de dados, e no entanto, a empresa nega que faça qualquer coleta. Outra questão importante é a falta de feedback emocional das ferramentas de IA, algo que é relativamente comum em uma relação entre professores e alunos, quando há a oportunidade de falar sobre os métodos de aprendizado utilizados e como a experiência de aprendizado em si tem se dado no curso do processo pedagógico.

De maneira geral, as preocupações apontadas por professores e alunos estão relacionadas a questões éticas fundamentais como o encorajamento ao plágio e práticas de trapaça; a falta de acurácia das respostas geradas pelo modelo, e a notável presença de vieses e informações falsas. Além disso,

os respondentes da pesquisa se mostraram bastante preocupados com o fato do ChatGPT criar referências falsas quando solicitado a fazer tarefas como essa.

2.2 CHATGPT E O USO MILITAR

O uso da IA (e do ChatGPT) em contextos militares levanta questões éticas, como a responsabilidade por decisões autônomas, a minimização de danos colaterais e a conformidade com leis internacionais. Essas aplicações requerem uma abordagem cuidadosa para equilibrar benefícios operacionais com preocupações éticas e legais (MACEY-DARE, 2023).

Há uma série de riscos ao usar a inteligência artificial e muitos países e organizações internacionais estão buscando estabelecer normas regulamentações para garantir o uso responsável e ético da inteligência artificial no contexto militar. Os modelos de dados que alimentam os algoritmos podem ter desvios, como erros de identificação facial biométrica ou inexistência de uma avaliação profunda do seu ambiente. No nível civil, especialistas em Inteligência Artificial pediram uma suspensão de seis meses do desenvolvimento desta tecnologia para estudar as suas possíveis consequências. (EURONEWS, 2023).

Quanto à violação dos Direitos Humanos (DH), há um risco significativo de que a IA seja usada para vigiar e controlar populações, infringindo DH fundamentais. Sistemas de IA podem ser usados para coletar e analisar grandes quantidades de dados sobre indivíduos, potencialmente levando a abusos de privacidade e liberdade. Em regimes autoritários, essa tecnologia pode ser empregada para suprimir dissidências e manter o controle social (BISWAS, 2023a).

A legislação internacional atual pode não estar preparada para lidar com os desafios trazidos pelo uso militar de IA. A ausência de normas claras sobre a utilização de tecnologias autônomas em combate cria um vácuo legal, onde ações questionáveis podem ser tomadas sem consequências claras. Há uma necessidade 8 urgente de desenvolver um quadro legal robusto que regule o uso de IA em contextos militares (ESMAILZADEH, 2023).

Em suma, enquanto o uso de IA, como o Chat GPT, em operações militares pode oferecer vantagens estratégicas significativas, os dilemas éticos são profundos e complexos. É imperativo que essas tecnologias sejam desenvolvidas e implementadas com um forte enfoque ético, garantindo que seu uso esteja alinhado com os princípios fundamentais dos direitos humanos e da dignidade humana. O desenvolvimento de políticas e regulamentações claras será crucial para mitigar os riscos e assegurar que a IA seja usada de maneira responsável e ética no campo militar (MONTEIRO, 2022), (SILVA, 2022).

3 SOLUÇÕES RESPONSÁVEIS

Uma das primeiras e mais importantes considerações a fazer quando se trata de soluções responsáveis é compreender o princípio de que incorporar a tecnologia é preferível a rejeitá-la, ainda que ela apresente desafios éticos significativos. De acordo com o autor, é necessário mais análise e discussão sobre como adotar o ChatGPT de forma responsável em ambientes educacionais, em vez

de simplesmente proibi-lo. Um importante fator nessa equação é sempre considerar que os conteúdos gerados pelas IAs generativas podem ser conceitualmente errados e também radicalmente diferentes de usuário para usuário, ainda que estes usem o mesmo prompt como input de dados (KUNG et. al, 2023). Em suma, o potencial revolucionário das ferramentas de IA generativa não podem ser descartados e precisam de atenção especial (TLILI et. al, 2023).

Tlili et. al (2023) ainda ressalta que o componente emocional de IAs generativas ainda aparece como uma limitação da tecnologia. A grande maioria dos chatbots são desenhados para atender a tarefas específicas, e não possuem qualidades socioemocionais, o que pode se apresentar como uma barreira para ampliar a efetividade da interação entre humanos e agentes virtuais, especialmente em um contexto de aprendizado. Por outro lado, há considerações importantes quanto a atribuir características humanas a inteligências artificiais como atribuir a chatbots a autoria ou coautoria sobre um conteúdo, por exemplo. De acordo com Tlili et. al (2023), o design de soluções de ChatBots como o ChatGPT devem considerar aspectos como a inclusão e usabilidade da ferramenta, além dos aspectos técnicos, como a arquitetura da rede neural de treinamento, o tamanho da base e os dados utilizados no processo.

De acordo com o autor, é necessário ir além da privacidade e segurança dos dados pessoais. Deve-se desenvolver diretrizes e estratégias que estejam alinhadas com valores humanos fundamentais e com o sistema legal.

4 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo demonstrar, por meio de estudos de caso disponíveis na literatura, de que forma dilemas éticos e suas nuances aparecem a partir do uso de ferramentas de IA Generativa, em especial o ChatGPT. Uma análise crítica foi realizada com base nos artigos selecionados, e ao final, buscou-se levantar também com base na literatura especializada, soluções responsáveis, possíveis e viáveis para mitigar alguns dos problemas que surgem em cenários de caso de uso de IAs generativas.

No contexto da educação, o estudo de caso apresentado demonstrou que ainda que os benefícios dessas ferramentas seja relevante, alguns desafios éticos gerados pelo uso da ferramenta, como o plágio ou a criação de conteúdo sem autenticidade, além de não term sido devidamente explorados pela literatura no momento, também podem ser facilmente contornadas pelos alunos, gerando ainda mais problemas eticos. Enquanto no estudo de caso do setor militar, questões como a coleta massiva de dados por sistemas de inteligência artificial, e a dificuldade em responsabilizar e coibir agentes autônomos por decisões de algoritmos de IA levanta profundos questionamentos sobre a segurança e confiabilidade do uso dessas ferramentas em situações reais, como em um campo de batalha. Todos esses desafios evidenciam os limites da legislação internacional para lidar com os problemas advindos desse tipo de tecnologia.

Apesar de todos os desafios apresentados, Tlili et. al (2023) demonstra que o potencial revolucionário das ferramentas de Inteligência Artificial e do ChatGPT não pode ser descartado, e que, por isso, o princípio a ser adotado é de que é melhor adotar a tecnologia de modo responsável, e não

ignorá-la completamente, ou desprezar seu uso. Entre as recomendações para utilizar essas tecnologias de modo responsável, estão a consideração do componente socioemocional na interação 10 entre humano-máquina, análise crítica das respostas e conteúdos gerados por IA, além de considerar questões legais, regulatórias e estratégias que se estendem para além do escopo da privacidade e segurança dos usuários.

REFERÊNCIAS

AYINDE, Lateef et al. ChatGPT as an important tool in organizational management: A review of the literature. Business Information Review, v. 40, n. 3, p. 137-149, 2023.
BISWAS, Som. Prospective role of chat gpt in the military: According to chatgpt. Qeios, 2023a.

BISWAS, Som S. Role of chat gpt in public health. Annals of biomedical engineering, v. 51, n. 5, p. 868-869, 2023b.

BOLANDER, Thomas. What do we lose when machines take the decisions?. Journal of Management and Governance, v. 23, p. 849-867, 2019.

DEVLIN, Jacob et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

EIGNER, Eva; HÄNDLER, Thorsten. Determinants of llm-assisted decision-making. arXiv preprint arXiv:2402.17385, 2024.
ELOUNDOU, Tyna et al. Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130, 2023.

EURONEWS.Que questões levanta o uso da inteligência artificial na guerra. Disponível em: https://pt.euronews.com/2023/05/08/que-questoes-levanta-o-uso-da-inteligencia-artifi cial-na-guerra.

ESMAILZADEH, Yaser. Potential Risks of ChatGPT: Implications for Counterterrorism and International Security. International Journal of Multicultural and Multireligious Understanding (IJMMU) Vol, v. 10, 2023.

KUNG, Tiffany H. et al. Performance of ChatGPT on USMLE: potential for AI-assisted medical education using large language models. PLoS digital health, v. 2, n. 2, p. e0000198, 2023.

HIRSCHBERG, Julia; MANNING, Christopher D. Advances in natural language processing. Science, v. 349, n. 6245, p. 261-266, 2015.

MACEY-DARE, Rupert. ChatGPT and Generative AI Systems as Military Ethics Advisors. Available at SSRN 4413206, 2023. 11 MOLLICK, Ethan R.;

MOLLICK, Lilach. New modes of learning enabled by ai chatbots: Three methods and assignments. Available at SSRN 4300783, 2022.

MONTEIRO, António Pedro Lopes. A inteligência artificial nos processos de modernização e edificação das capacidades militares do exército: vetor de desenvolvimento, liderança e formação. 2022. 56 f. Monografia (Especialização) - Curso de Departamento de Estudos Pós-Graduados, Instituto Universitário Militar, Pedrouços, 2022

NILSSON, Nils J. The quest for artificial intelligence. Cambridge University Press, 2009.

SHEN-BERRO, J. New York City Schools blocked ChatGPT. Here's what other large districts are doing. Chalkbeat, 2023. Disponível em: https://www.chalkbeat.org/2023/1/6/23543039/chatgpt-school-districts-ban-block-artifi cial-intelligence-open-ai.

SILVA, Luciano Santos da. A inteligência artificial e sua aplicação no mundo militar. A Lucerna, [s. l], v. 11, n. 1, p. 85-89, 25 mar. 2022.

STAHL, Bernd Carsten; EKE, Damian. The ethics of ChatGPT–Exploring the ethical issues of an emerging technology. International Journal of Information Management, v. 74, p. 102700, 2024.

TLILI, Ahmed et al. What if the devil is my guardian angel: ChatGPT as a case study of using chatbots in education. Smart learning environments, v. 10, n. 1, p. 15, 2023.
VIDHYA, N. Gowri et al. Prognosis of exploration on Chat GPT with artificial intelligence ethics. Brazilian Journal of Science, v. 2, n. 9, p. 60-69, 2023.

APÊNDICE 12 – GESTÃO DE PROJETOS DE IA

A – ENUNCIADO

1 Objetivo

Individualmente, ler e resumir – seguindo o *template* fornecido – **um** dos artigos abaixo:

AHMAD, L.; ABDELRAZEK, M.; ARORA, C.; BANO, M; GRUNDY, J. Requirements practices and gaps when engineering human-centered Artificial Intelligence systems. Applied Soft Computing. 143. 2023. DOI https://doi.org/10.1016/j.asoc.2023.110421

NAZIR, R.; BUCAIONI, A.; PELLICCIONE, P.; Architecting ML-enabled systems: Challenges, best practices, and design decisions. The Journal of Systems & Software. 207. 2024. DOI https://doi.org/10.1016/j.jss.2023.111860

SERBAN, A.; BLOM, K.; HOOS, H.; VISSER, J. Software engineering practices for machine learning – Adoption, effects, and team assessment. The Journal of Systems & Software. 209. 2024. DOI https://doi.org/10.1016/j.jss.2023.111907

STEIDL, M.; FELDERER, M.; RAMLER, R. The pipeline for continuous development of artificial intelligence models – Current state of research and practice. The Journal of Systems & Software. 199. 2023. DOI https://doi.org/10.1016/j.jss.2023.111615

XIN, D.; WU, E. Y.; LEE, D. J.; SALEHI, N.; PARAMESWARAN, A. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. In CHI Conference on Human Factors in Computing Systems (CHI'21), Maio 8-13, 2021, Yokohama, Japão. DOI https://doi.org/10.1145/3411764.3445306

2 Orientações adicionais

Escolha o artigo que for mais interessante para você. Utilize tradutores e o Chat GPT para entender o conteúdo dos artigos – caso precise, mas escreva o resumo em língua portuguesa e nas suas palavras.

Não esqueça de preencher, no trabalho, os campos relativos ao seu nome e ao artigo escolhido.

No *template*, você deverá responder às seguintes questões:

- Qual o objetivo do estudo descrito pelo artigo?
- Qual o problema/oportunidade/situação que levou a necessidade de realização deste estudo?
- Qual a metodologia que os autores usaram para obter e analisar as informações do estudo?
- Quais os principais resultados obtidos pelo estudo?

Responda cada questão utilizando o espaço fornecido no *template*, sem alteração do tamanho da fonte (Times New Roman, 10), nem alteração do espaçamento entre linhas (1.0).

Não altere as questões do template.

Utilize o editor de textos de sua preferência para preencher as respostas, mas entregue o trabalho em PDF.

B – RESOLUÇÃO

Qual o objetivo do estudo descrito pelo artigo?	Qual o problema/oportunidade/situação que levou à necessidade da realização desse estudo?	Qual a metodologia que os autores usaram para obter e analisar as informações do estudo?	Quais os principais resultados obtidos pelo estudo?
O objetivo do estudo é analisar práticas e lacunas nas práticas de Engenharia de Requisitos para sistemas de Inteligência Artificial centrados no ser humano. O estudo busca entender como essas práticas são adotadas na indústria, para fornecer recomendações que ajudem a mitigar problemas como falta de explicabilidade, controle do usuário, transparência e redução de vieses em sistemas de IA.	Há uma falta de diretrizes e práticas consolidadas para o desenvolvimento de sistemas de Inteligência Artificial centrados no ser humano. Grandes empresas, como Google, Microsoft e Apple, publicaram diretrizes para auxiliar no desenvolvimento de IA centrada no ser humano, mas ainda existe pouca pesquisa sobre como essas práticas são realmente adotadas na indústria, especialmente na fase de Engenharia de Requisitos.	Utilizaram questionário e revisão da literatura para obtenção e análise das informações do estudo. Primeiramente, 29 profissionais da área de IA foram entrevistados para entender as práticas atuais de Engenharia de Requisitos adotadas na indústria. Além disso, eles realizaram um mapeamento das diretrizes industriais existentes (como as da Google, Microsoft e Apple) e as compararam com a literatura acadêmica. Essa metodologia	Embora algumas práticas centradas no ser humano estejam sendo adotadas na Engenharia de Requisitos para IA, ainda há lacunas importantes. Ferramentas tradicionais, como UML e Microsoft Office, não capturam bem requisitos específicos de IA, como explicabilidade e controle do usuário. Requisitos não funcionais, como ética e transparência, são pouco considerados, enquanto requisitos funcionais e de dados são priorizados. O estudo recomenda o desenvolvimento de

		permitiu aos autores identificar quais práticas centradas no ser humano estão sendo implementadas e quais lacunas ainda existem entre a teoria e a prática industrial.	novas ferramentas e métodos que integrem aspectos humanos e éticos, além de mais atenção a falhas e feedback dos usuários para melhorar a qualidade dos sistemas de IA centrados no ser humano.

APÊNDICE 13 – FRAMEWORKS DE INTELIGÊNCIA ARTIFICIAL

A – ENUNCIADO

1 Classificação (RNA)

Implementar o exemplo de Classificação usando a base de dados Fashion MNIST e a arquitetura RNA vista na aula **FRA - Aula 10 - 2.4 Resolução de exercício de RNA - Classificação**. Além disso, fazer uma breve explicação dos seguintes resultados:

- Gráficos de perda e de acurácia;
- Imagem gerada na seção "**Mostrar algumas classificações erradas"**, apresentada na aula prática.
  Informações:

- **Base de dados:** Fashion MNIST Dataset
- **Descrição**: Um dataset de imagens de roupas, onde o objetivo é classificar o tipo de vestuário. É semelhante ao famoso dataset MNIST, mas com peças de vestuário em vez de dígitos.
- **Tamanho**: 70.000 amostras, 784 features (28x28 pixels).
- **Importação do dataset**: Copiar código abaixo.

```
data = tf.keras.datasets.fashion_mnist
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

2 Regressão (RNA)

Implementar o exemplo de Classificação usando a base de dados Wine Dataset e a arquitetura RNA vista na aula **FRA - Aula 12 - 2.5 Resolução de exercício de RNA - Regressão**. Além disso, fazer uma breve explicação dos seguintes resultados:

- Gráficos de avaliação do modelo (loss);
- Métricas de avaliação do modelo (pelo menos uma entre MAE, MSE, R^2).

Informações:

- **Base de dados:** Wine Quality
- **Descrição**: O objetivo deste dataset prever a qualidade dos vinhos com base em suas características químicas. A variável target (y) neste exemplo será o score de qualidade do vinho, que varia de 0 (pior qualidade) a 10 (melhor qualidade)
- **Tamanho**: 1599 amostras, 12 features.
- **Importação**: Copiar código abaixo.

```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
data = pd.read_csv(url, delimiter=';')
```

Dica 1. Para facilitar o trabalho, renomeie o nome das colunas para português, dessa forma:

```
data.columns = [
    'acidez_fixa',            # fixed acidity
    'acidez_volatil',         # volatile acidity
    'acido_citrico',          # citric acid
    'acucar_residual',        # residual sugar
    'cloretos',               # chlorides
    'dioxido_de_enxofre_livre', # free sulfur dioxide
    'dioxido_de_enxofre_total', # total sulfur dioxide
    'densidade',              # density
    'pH',                     # pH
    'sulfatos',               # sulphates
    'alcool',                 # alcohol
    'score_qualidade_vinho'              # quality
]
```

Dica 2. Separe os dados (x e y) de tal forma que a última coluna (índice -1), chamada score_qualidade_vinho, seja a variável target (y)

3 Sistemas de Recomendação

Implementar o exemplo de Sistemas de Recomendação usando a base de dados Base_livos.csv e a arquitetura vista na aula **FRA - Aula 22 - 4.3 Resolução do Exercício de Sistemas de Recomendação**. Além disso, fazer uma breve explicação dos seguintes resultados:

- Gráficos de avaliação do modelo (loss);
- Exemplo de recomendação de livro para determinado Usuário.

Informações:

- **Base de dados:** Base_livros.csv
- **Descrição**: Esse conjunto de dados contém informações sobre avaliações de livros (Notas), nomes de livros (Titulo), ISBN e identificação do usuário (ID_usuario)
- **Importação:** Base de dados disponível no Moodle (UFPR Virtual), chamada Base_livros (formato .csv).

4 Deepdream

Implementar o exemplo de implementação mínima de Deepdream usando uma imagem de um felino - retirada do site Wikipedia - e a arquitetura Deepdream vista na aula **FRA - Aula 23 - Prática Deepdream**. Além disso, fazer uma breve explicação dos seguintes resultados:

- Imagem onírica obtida por *Main Loop*;
- Imagem onírica obtida ao levar o modelo até uma oitava;
- Diferenças entre imagens oníricas obtidas com *Main Loop* e levando o modelo até a oitava.

Informações:

- **Base de dados:** https://commons.wikimedia.org/wiki/File:Felis_catus-cat_on_snow.jpg
- **Importação da imagem**: Copiar código abaixo.

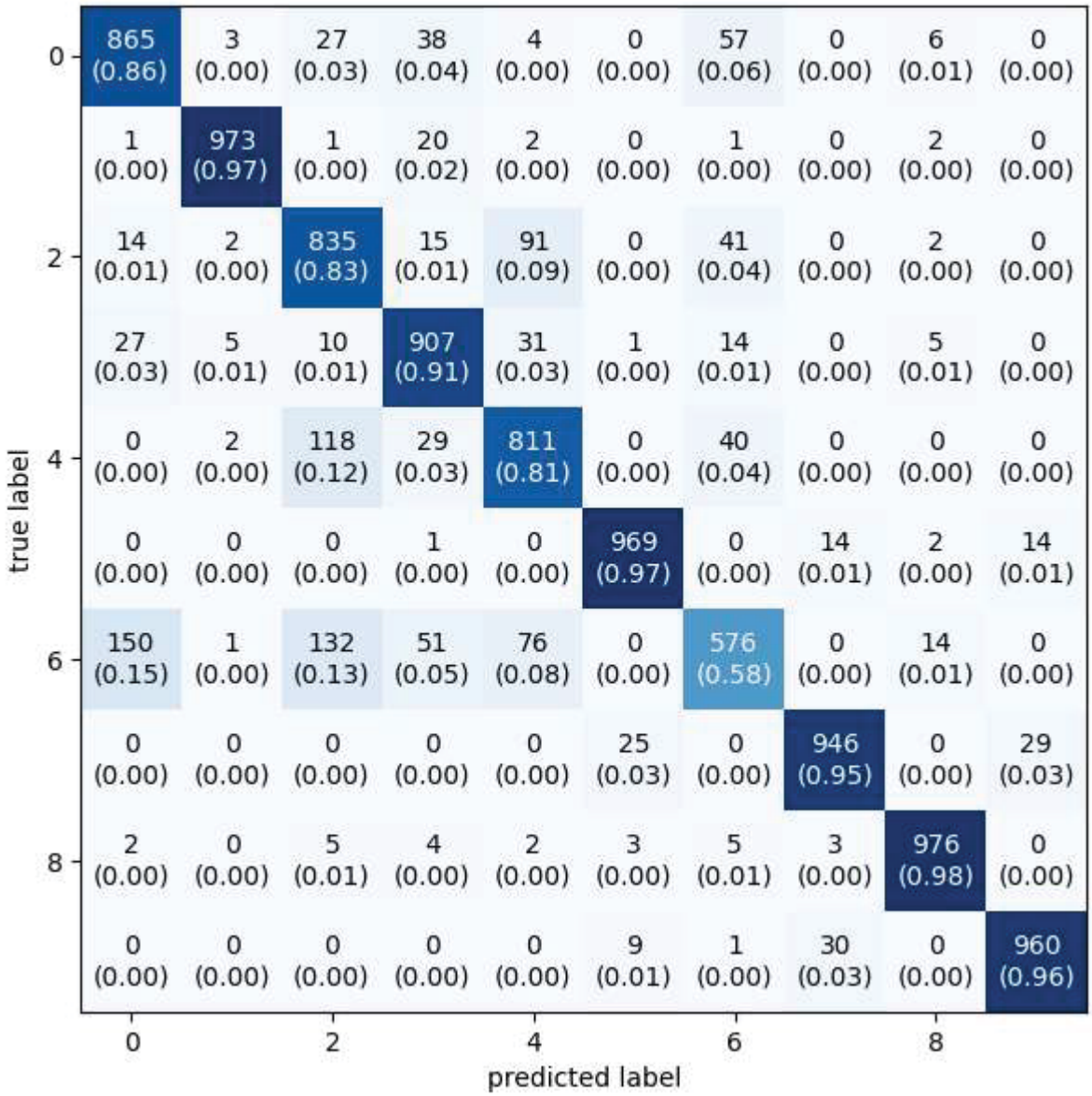
```
url = "https://commons.wikimedia.org/wiki/Special:FilePath/Felis_catus-cat_on_snow.jpg"
```

```
Dica: Para exibir a imagem utilizando display (display.html) use o link https://commons.wikimedia.org/wiki/File:Felis_catus-cat_on_snow.jpg
```

B – RESOLUÇÃO

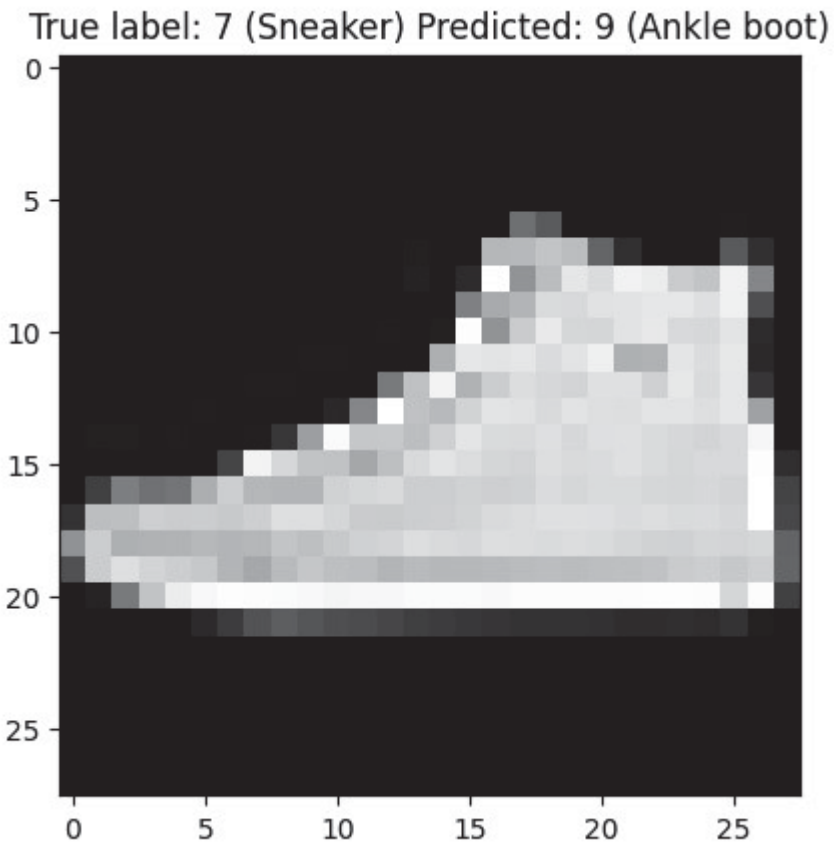
1)

FIGURA 6 – MATRIZ DE CONFUSÃO



FONTE: O autor (2025).

FIGURA 7 – EXEMPLO DE IMAGEM



FONTE: O autor (2025).

Imagem com classificação errada

Breve explicação dos seguintes resultados:

Gráficos de perda e de acurácia

No gráfico de perda, vemos que a perda com os dados de treino diminuiu rapidamente, enquanto que com os dados de validação chegou até a aumentar em uma das épocas. Como os valores de peda ainda estavam diminuindo, talvez o treinamento com mais épocas gere um melhor resultado.

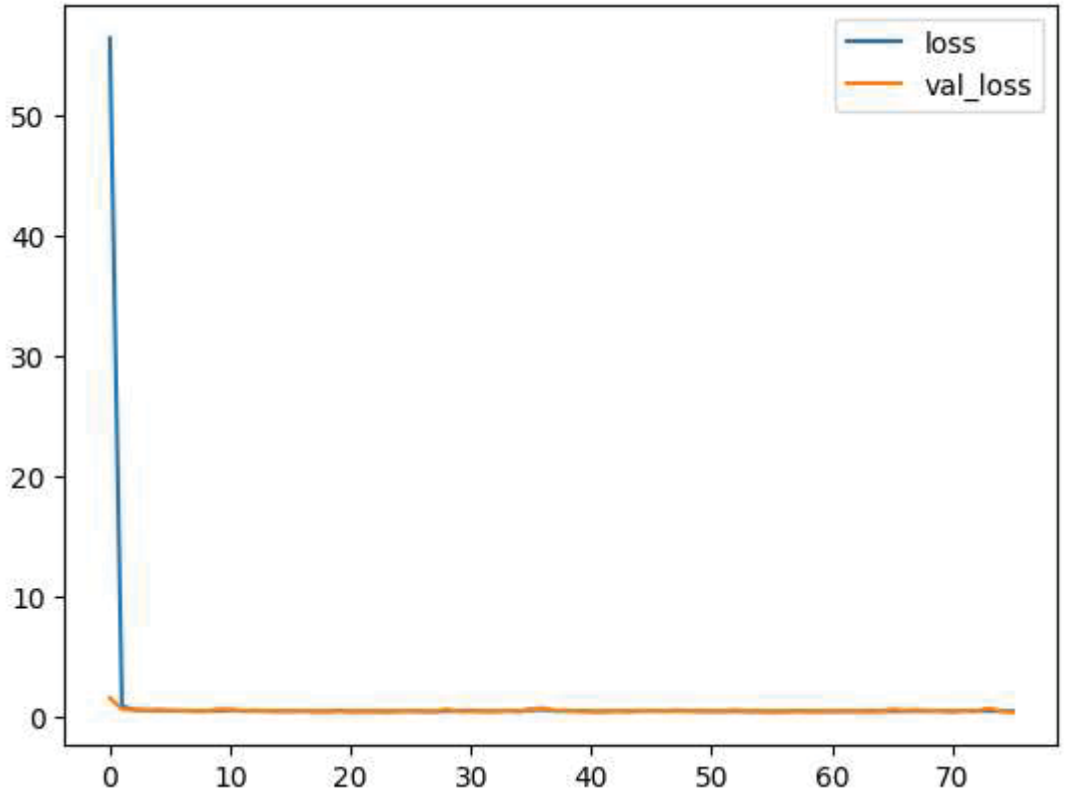
No gráfico de acurácia também verificamos que com os dados de treino o resultado melhora rapidamente, e com os dados de validação ocorre uma oscilação em determinada época. Do mesmo modo, como o gráfico mostra a acurácia melhorando a cada época, sem uma estabilização, é provável que o aumento das épocas auxilie a melhorar o resultado.

Imagem gerada na seção “Mostrar algumas classificações erradas”

Na exibição de uma classificação errada, vemos um tênis (sneaker) que foi incorretamente classificado como um tipo de bota (ankle boot). O fato das imagens terem baixa definição, e o modelo do tênis (com a canela mais alta) devem ter contribuído para este caso de classificação incorreta.

2)

GRÁFICO 13 – AVALIAÇÃO DO MODELO



FONTE: O autor (2025).

Explicação dos seguintes resultados:

Gráficos de avaliação do modelo (loss)

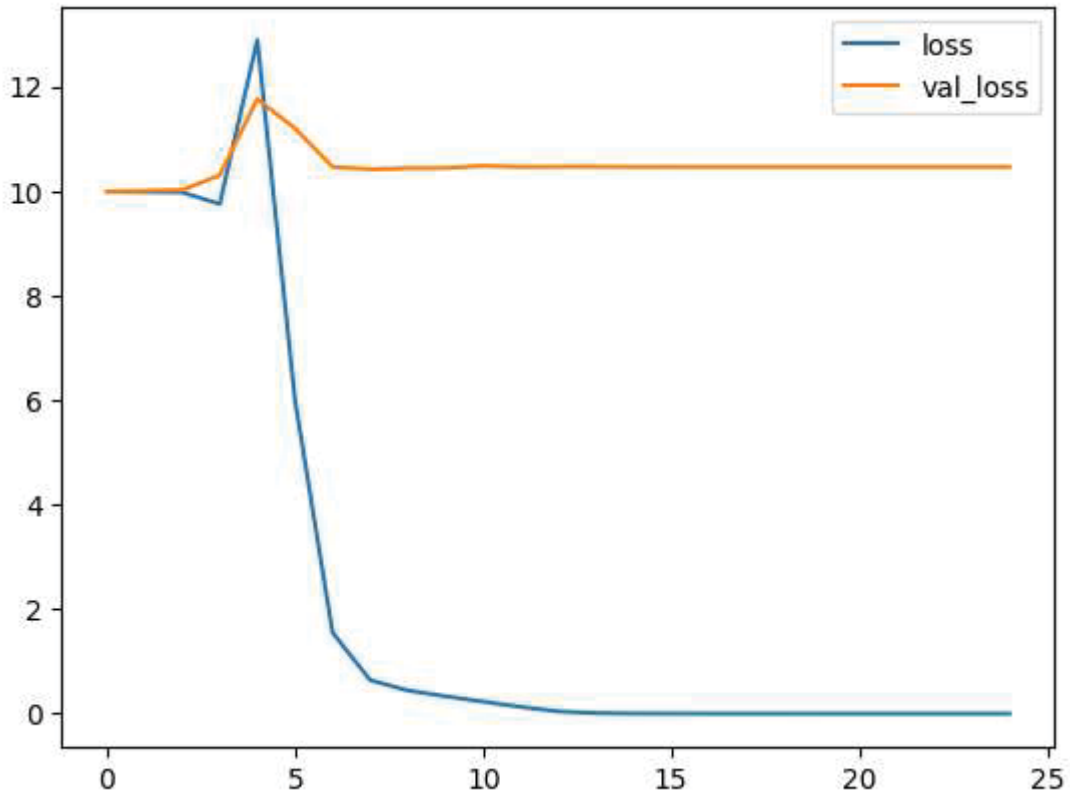
Verificando o gráfico de perda, percebemos que apenas na primeira época a perda com os dados de treino foi alta, diminuindo bastante logo em seguida. Com os dados de validação, as diferenças foram sutis de uma época para outra, não sendo obtida melhora muito significativa ao longo do treinamento. A configuração de "Early Stop" também fez com que o treinamento fosse interrompido bem antes das 1500 épocas definidas como limite.

Métricas de avaliação do modelo:

Pelos valores verificados na avaliação, o modelo precisa de melhorias. O R2 baixo (0.33) mostra que o modelo tem dificuldade em explicar os dados. Alterar as camadas da rede, normalizar os dados e ajustar hiperparâmetros são técnicas que podem auxiliar a obter melhores resultados.

3)

GRÁFICO 14 – AVALIAÇÃO DO MODELO



FONTE: O autor (2025).

Breve explicação dos resultados:

Gráficos de avaliação do modelo (loss);
No gráfico de perda, verificamos que, com os dados de treinamento, a perda ficou bem baixa, e com os dados de validação ficou muito maior. Isso é um indicativo de overfitting, ou seja, o modelo não está muito ajustado aos dados de treinamento, conseguindo generalizar para outro conjunto de dados.

Exemplo de recomendação de livro para determinado usuário:
No exemplo de recomendação, foi selecionado um usuário, e o modelo recomendou o livro "Legal Tender" para ele, prevendo uma possível nota "10".

4)

FIGURA 8 – IMAGEM ORIGINAL

FONTE: O autor (2025).

FIGURA 9 – IMAGEM DEPOIS QUE O MODELO FOI LEVADO ATÉ UMA OITAVA

FONTE: O autor (2025)

Breve explicação dos seguintes resultados:

Imagem onírica obtida por Main Loop:
O main loop é o ciclo principal do DeepDream. Nele, são feitas várias iterações modificando a imagem original, gerando outra formada por repetições de padrões, formas e cores detectados pela rede neural. O modelo "sonha" com as características que ele reconhece nas imagens e tenta realçá-las de maneira exagerada, gerando uma imagem com visual mais surreal.

Imagem onírica obtida ao levar o modelo até uma oitava:
Levar até uma oitava faz com que a nova imagem tenha mais variações, sendo mais complexa que a imagem gerada anteriormente no main loop. As texturas e padrões gerados são mais diversos ao levar o modelo até uma oitava.

APÊNDICE 14 – VISUALIZAÇÃO DE DADOS E STORYTELLING

A – ENUNCIADO

Escolha um conjunto de dados brutos (ou uma visualização de dados que você acredite que possa ser melhorada) e faça uma visualização desses dados (de acordo com os dados escolhidos e com a ferramenta de sua escolha)

Desenvolva uma narrativa/storytelling para essa visualização de dados considerando os conceitos e informações que foram discutidas nesta disciplina. Não esqueça de deixar claro para seu possível público alvo qual **o objetivo dessa visualização de dados, o que esses dados significam, quais possíveis ações podem ser feitas com base neles.**

Entregue em um PDF:

- O **conjunto de dados brutos** (**ou uma visualização de dados** que você acredite que possa ser **melhorada**);

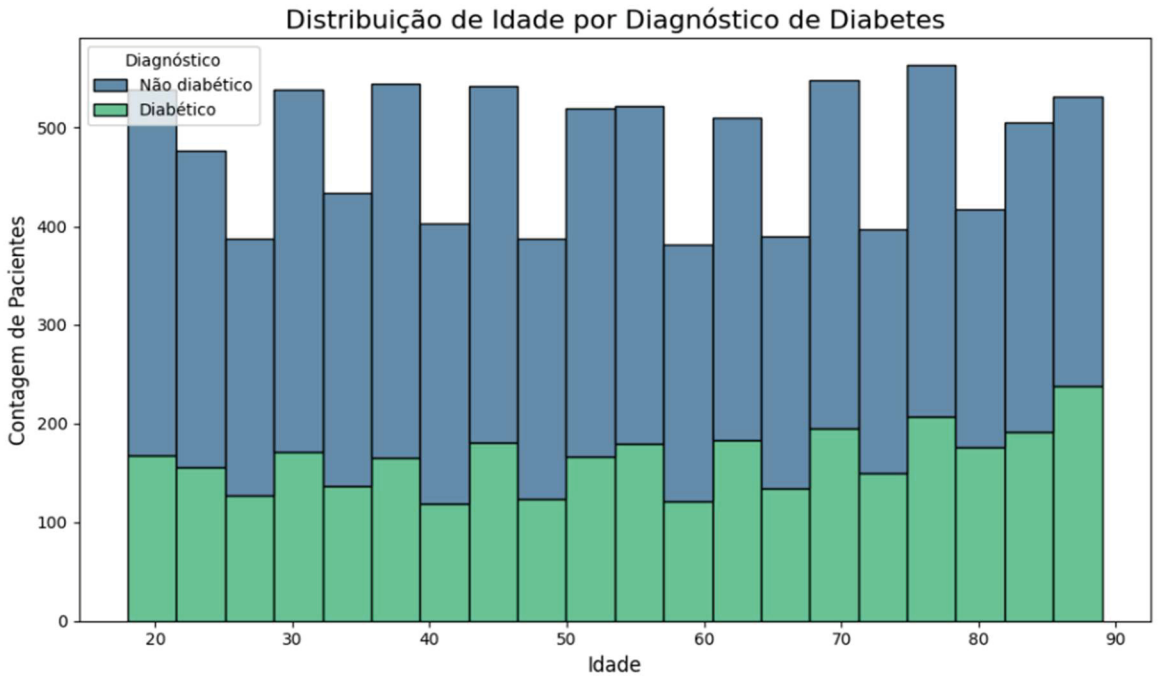
- Explicação do **contexto e o publico-alvo** da visualização de dados e do storytelling que será desenvolvido;

- A **visualização desses dados** (de acordo com os dados escolhidos e com a ferramenta de sua escolha) **explicando a escolha do tipo de visualização e da ferramenta usada**; (**50 pontos**)

B – RESOLUÇÃO

1)

GRÁFICO 15 – DISTRIBUIÇÃO DE IDADE POR DIAGNÓSTICO DE DIABETES

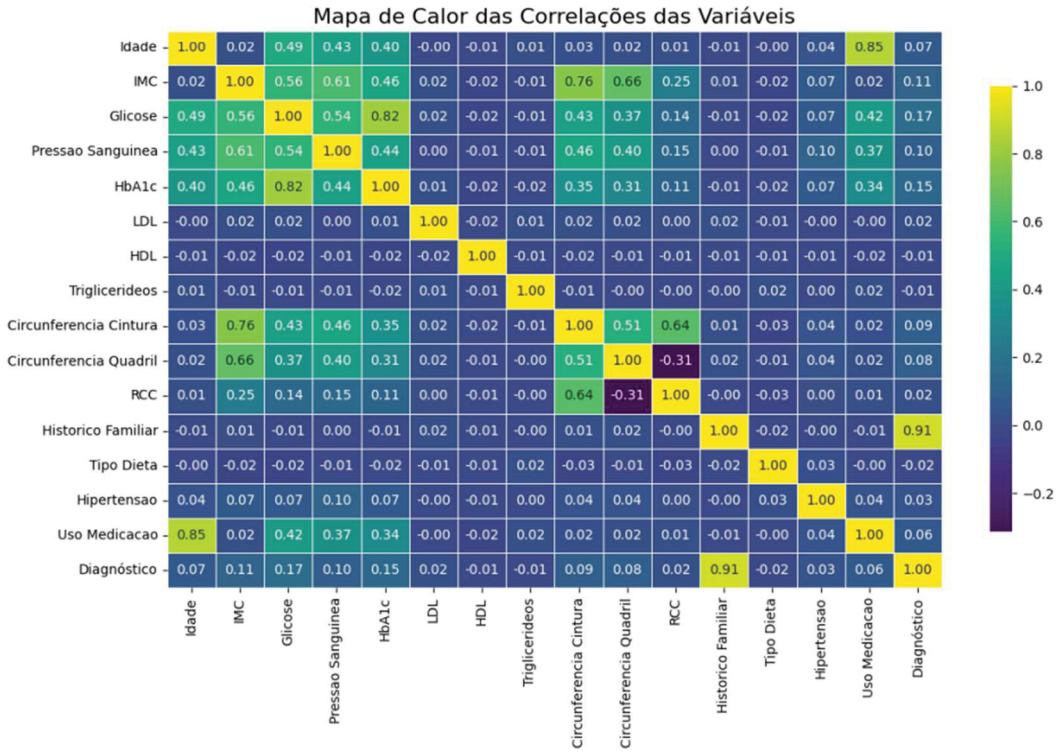


FONTE: O autor (2025).

Este histograma mostra a distribuição da idade dos pacientes, separando aqueles com e sem diabetes. Isso pode ajudar a identificar se há uma faixa etária mais propensa ao diagnóstico de diabetes. Analisando o gráfico, podemos observar as seguintes características principais:

- A quantidade total de pacientes (soma das barras verdes e azuis) se mantém relativamente estável entre as diferentes faixas etárias, com valores aproximados entre 400-550 pacientes por faixa.
- Existe um aumento visível na proporção de pacientes diabéticos (barras verdes) com o avanço da idade. Aos 20 anos, aproximadamente 150-170 pacientes são diabéticos. Aos 90 anos, este número sobe para cerca de 230- 240 pacientes.
- A proporção de pacientes diabéticos aumenta gradualmente a partir dos 60 anos.
- O grupo mais afetado pela diabetes é o de 90 anos, onde a barra verde representa quase metade do total de pacientes nessa idade.
- Entre 70-90 anos ocorre o maior aumento proporcional de casos de diabetes.

GRÁFICO 16 – MAPA DE CALOR DAS CORRELAÇÕES DAS VARIÁVEIS



FONTE: O autor (2025).

Ao analisar o gráfico de correlações, podemos inferir que:

Correlações fortes (próximas a 1.0):

- Histórico Familiar e Resultado (0.91): Existe uma correlação muito forte entre histórico familiar e o resultado, sugerindo que o histórico familiar é um forte preditor do resultado avaliado.
- Idade e Uso de Medicação (0.85): Há uma correlação forte positiva, indicando que pessoas mais velhas tendem a usar mais medicações. Glicose e HbA1c (0.82): Como esperado, níveis de glicose e hemoglobina glicada são fortemente correlacionados, o que é consistente com o conhecimento médico sobre diabetes.

Correlações moderadas a fortes:

- IMC e Circunferência da Cintura (0.76): Indica que pessoas com maior IMC tendem a ter maior circunferência da cintura.
- IMC e Circunferência do Quadril (0.66): Similar à relação anterior, mostrando que o IMC está relacionado às medidas de circunferência corporal.
- RCC e Circunferência da Cintura (0.64): Relação entre a razão cintura-quadril e a circunferência da cintura.

- IMC e Glicose (0.56): Sugere que pessoas com maior IMC tendem a ter níveis mais elevados de glicose.

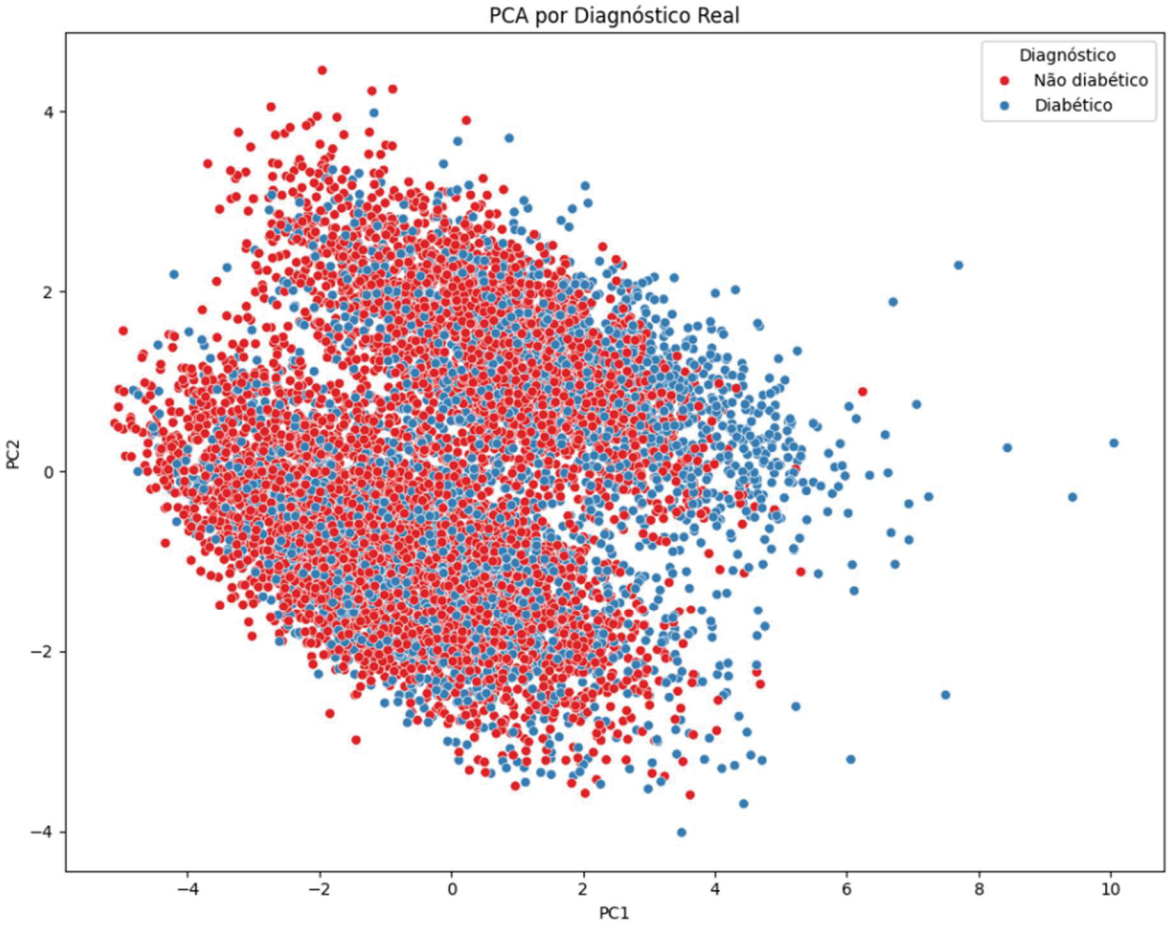
Correlações que sugerem fatores de risco metabólico:

- Glicose e Pressão Sanguínea (0.54): Mostra relação entre níveis de glicose e pressão arterial.
- Pressão Sanguínea e IMC (0.61): Sugere que pessoas com maior IMC tendem a ter pressão arterial mais elevada

Observações interessantes:

HDL apresenta correlações negativas com várias variáveis, o que é consistente com seu papel protetor na saúde cardiovascular. LDL mostra correlações relativamente baixas com outras variáveis. Tipo de Dieta tem correlações fracas com a maioria das variáveis. Este mapa de calor sugere um quadro de síndrome metabólica, onde fatores como idade, IMC, circunferência da cintura, níveis de glicose e pressão sanguínea estão interconectados. A forte correlação entre histórico familiar e resultado sugere uma componente genética importante no desfecho analisado, possivelmente relacionado a diabetes ou doença cardiovascular.

GRÁFICO 18 - COMPONENTES PRINCIPAIS

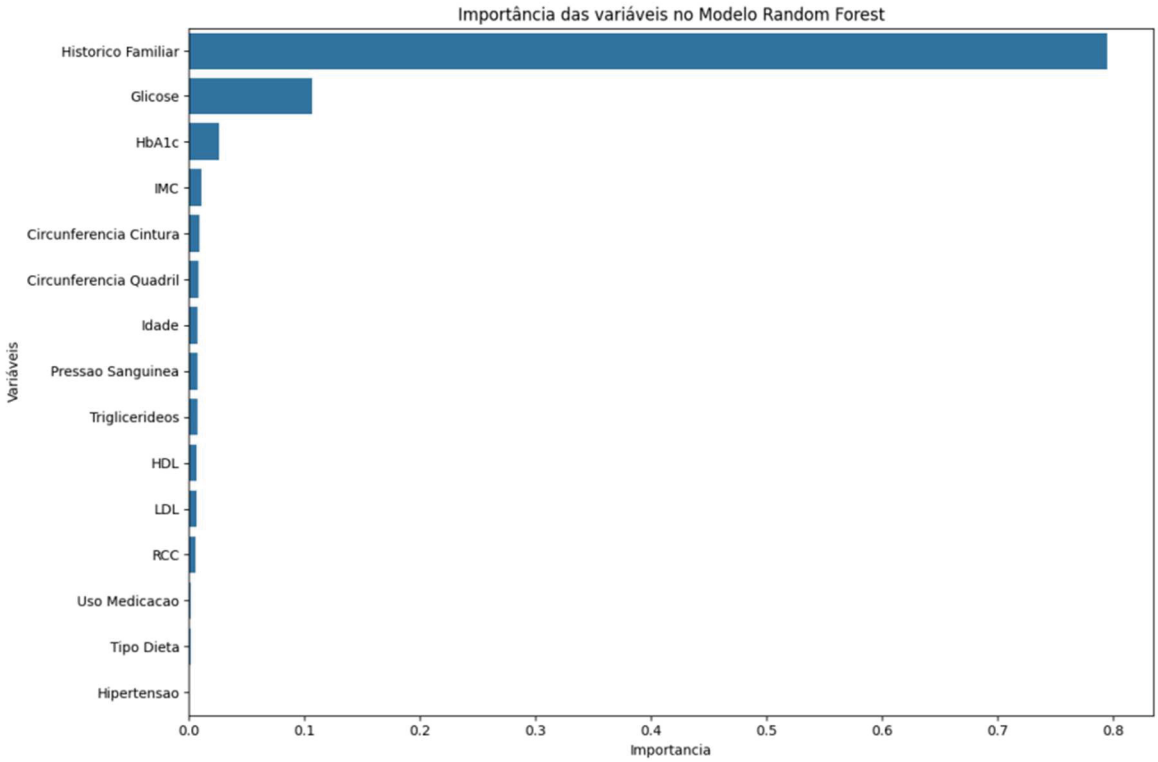


FONTE: O autor (2025).

O gráfico de análise de componentes principais nos mostra que:

- Os pacientes diabéticos (azul) tendem a se concentrar mais no lado direito do gráfico (valores mais altos de PC1).
- Os valores extremamente altos de PC1 (>5) são quase exclusivamente de pacientes diabéticos.
- A região esquerda contém uma mistura mais equitativa de ambos os diagnósticos.

GRÁFICO 19 – IMPORTÂNCIA DAS VARIÁVEIS NO MODELO RANDOM FOREST



FONTE: O autor (2025).

Este gráfico apresenta a importância relativa de cada variável no modelo de classificação Random Forest para prever diabetes. A importância indica quanto cada característica contribui para a capacidade preditiva do modelo. Podemos observar que:

- Histórico Familiar é, de longe, a característica mais importante, com uma pontuação de aproximadamente 0,8 (80%). Isso indica que a presença de histórico familiar de diabetes é o fator mais determinante para o diagnóstico neste modelo, sugerindo uma forte componente genética ou familiar na condição.
- Glicose aparece como o segundo fator mais importante, com pontuação em torno de 0,1 (10%). Isso é consistente com o conhecimento médico, já que níveis elevados de glicose no sangue são um indicador direto de diabetes.
- HbA1c (hemoglobina glicada) é o terceiro fator mais relevante, com importância de aproximadamente 0,03 (3%). Este teste reflete os níveis médios de glicose no sangue nos últimos 2-3 meses e é um indicador padrão para diagnóstico de diabetes.
- Características de importância secundária incluem IMC, medidas de circunferência e perfil lipídico (HDL, LDL, Triglicerídeos), todas com importância abaixo de 0,02 (2%).
- Características com importância mínima: Uso de Medicação, Tipo de Dieta e Hipertensão apresentam contribuição quase nula para o modelo, o que pode ser surpreendente considerando que são fatores frequentemente associados ao diabetes na literatura médica.

APÊNDICE 15 – TÓPICOS EM INTELIGÊNCIA ARTIFICIAL

A – ENUNCIADO

1) Algoritmo Genético

Problema do Caixeiro Viajante

A Solução poderá ser apresentada em: Python (preferencialmente), ou em R, ou em Matlab, ou em C ou em Java.

Considere o seguinte problema de otimização (a escolha do número de 100 cidades foi feita simplesmente para tornar o problema intratável. A solução ótima para este problema não é conhecida).

Suponha que um caixeiro deva partir de sua cidade, visitar clientes em outras 99 cidades diferentes, e então retornar à sua cidade. Dadas as coordenadas das 100 cidades, descubra o percurso de menor distância que passe uma única vez por todas as cidades e retorne à cidade de origem.

Para tornar a coisa mais interessante, as coordenadas das cidades deverão ser sorteadas (aleatórias), considere que cada cidade possui um par de coordenadas (x e y) em um espaço limitado de 100 por 100 pixels.

O relatório deverá conter no mínimo a primeira melhor solução (obtida aleatoriamente na geração da população inicial) e a melhor solução obtida após um número mínimo de 1000 gerações. Gere as imagens em 2d dos pontos (cidades) e do caminho.

Sugestão:

(1) considere o cromossomo formado pelas cidades, onde a cidade de início (escolhida aleatoriamente) deverá estar na posição 0 e 100 e a ordem das cidades visitadas nas posições de 1 a 99 deverão ser definidas pelo algoritmo genético.
(2) A função de avaliação deverá minimizar a distância euclidiana entre as cidades (os pontos).
(3) Utilize no mínimo uma população com 100 indivíduos;
(4) Utilize no mínimo 1% de novos indivíduos obtidos pelo operador de mutação;
(5) Utilize no mínimo de 90% de novos indivíduos obtidos pelo método de cruzamento (crossover-ox);
(6) Preserve sempre a melhor solução de uma geração para outra.

Importante: A solução deverá implementar os operadores de “cruzamento” e “mutação”.

2) Compare a representação de dois modelos vetoriais

Pegue um texto relativamente pequeno, o objetivo será visualizar a representação vetorial, que poderá ser um vetor por palavra ou por sentença. Seja qual for a situação, considere a quantidade de

palavras ou sentenças onde tenha no mínimo duas similares e no mínimo 6 textos, que deverão produzir no mínimo 6 vetores. Também limite o número máximo, para que a visualização fique clara e objetiva.

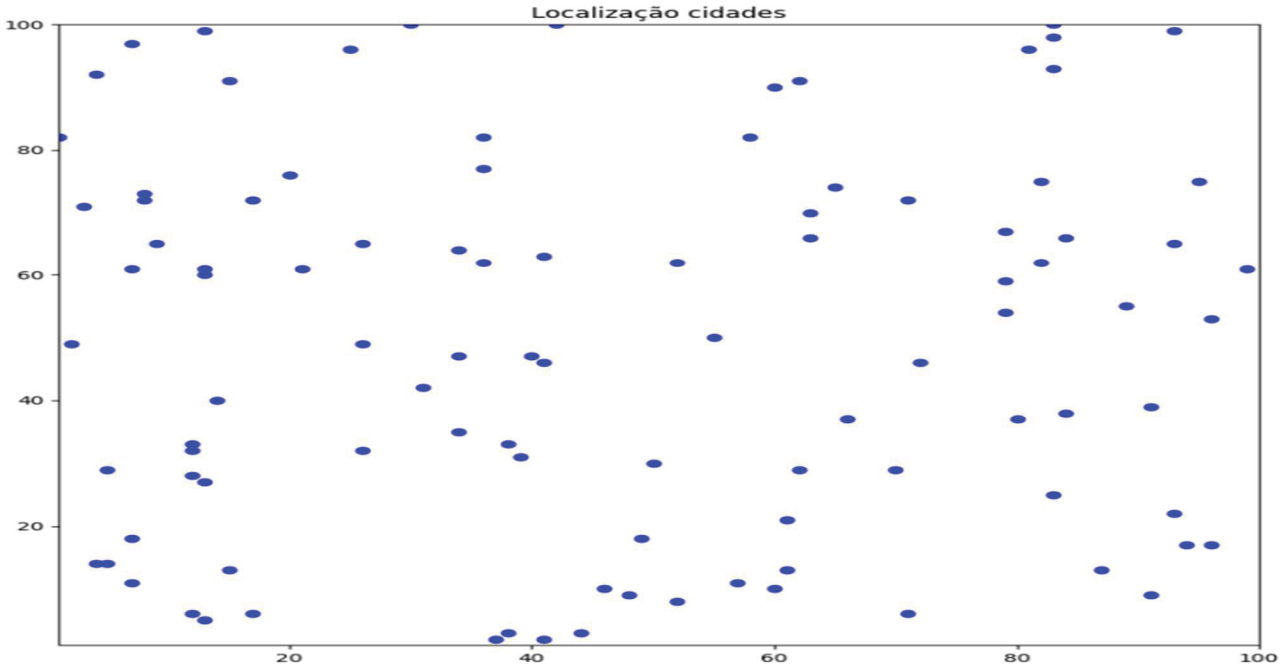
O trabalho consiste em pegar os fragmentos de texto e codificá-las na forma vetorial. Após obter os vetores, imprima-os em figuras (plot) que demonstrem a projeção desses vetores usando a PCA.

O PDF deverá conter o código-fonte e as imagens obtidas.

B – RESOLUÇÃO

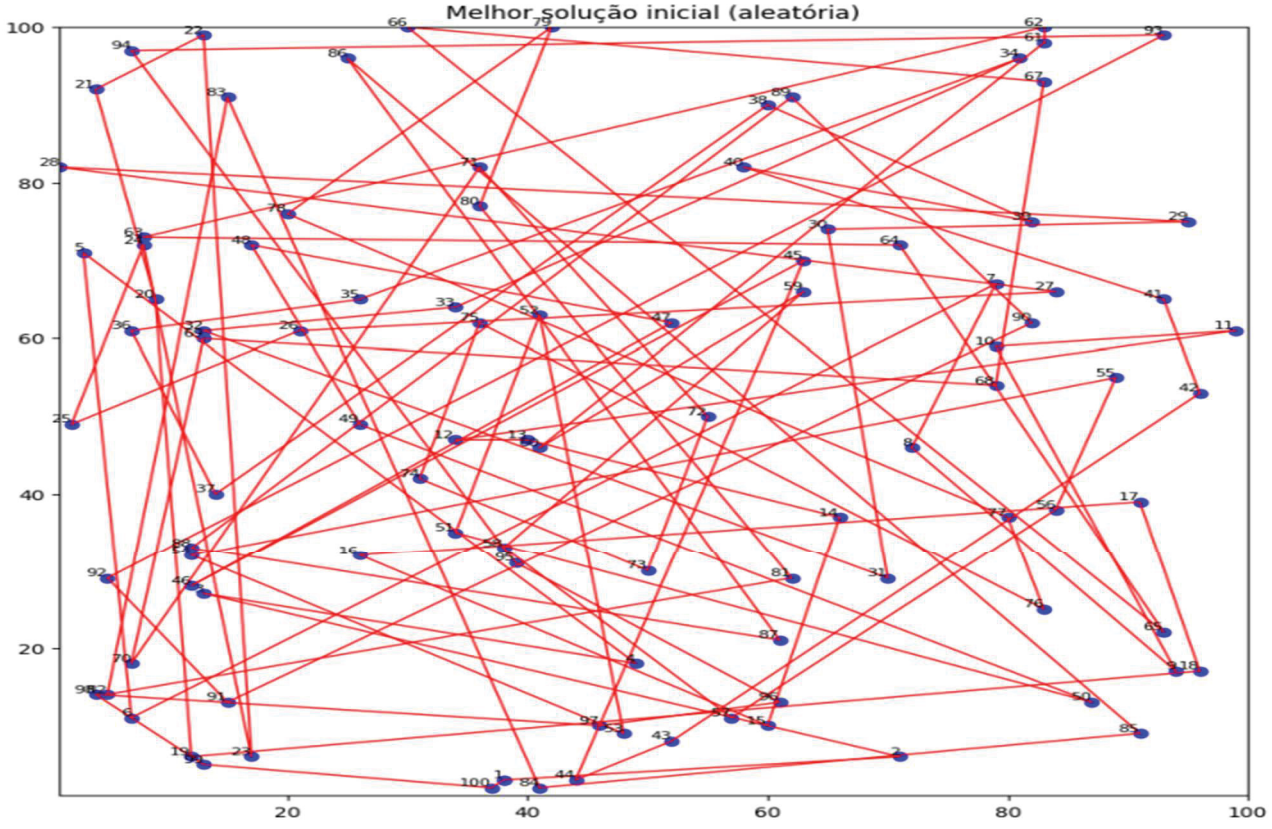
1)

FIGURA 10 – LOCALIZAÇÃO DAS CIDADES



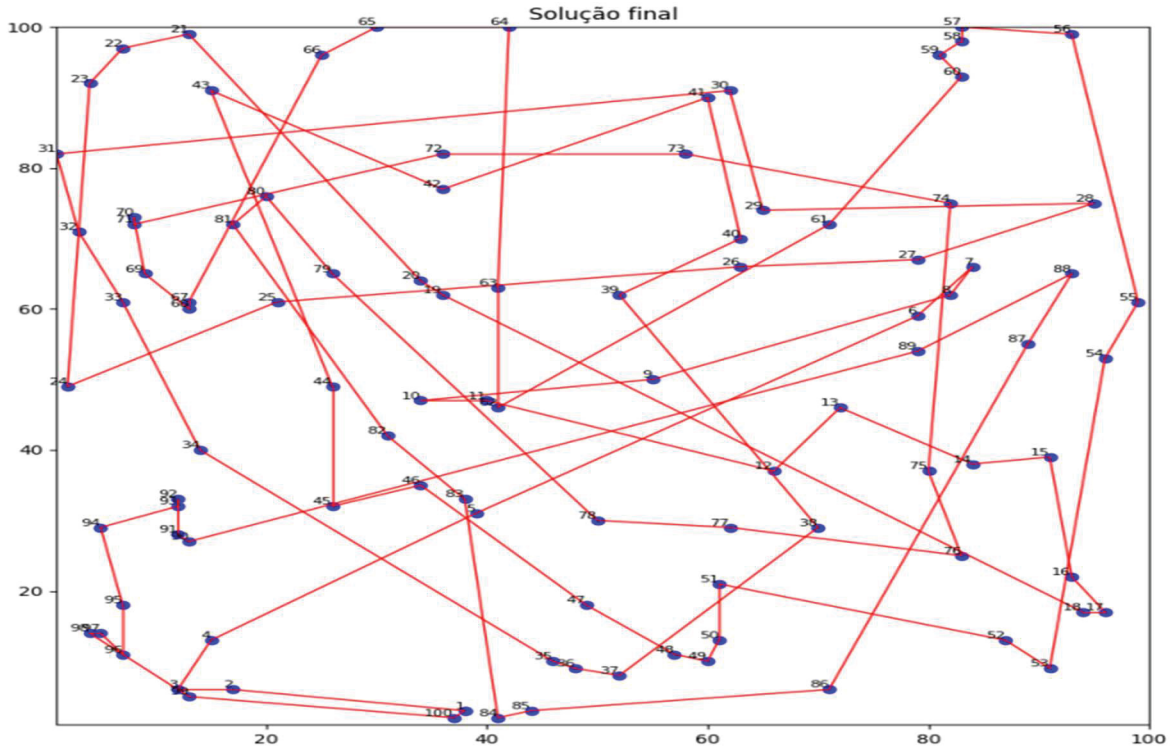
FONTE: O autor (2025).

FIGURA 11 – MELHOR SOLUÇÃO INICIAL (ALEATÓRIA)



FONTE: O autor (2025).

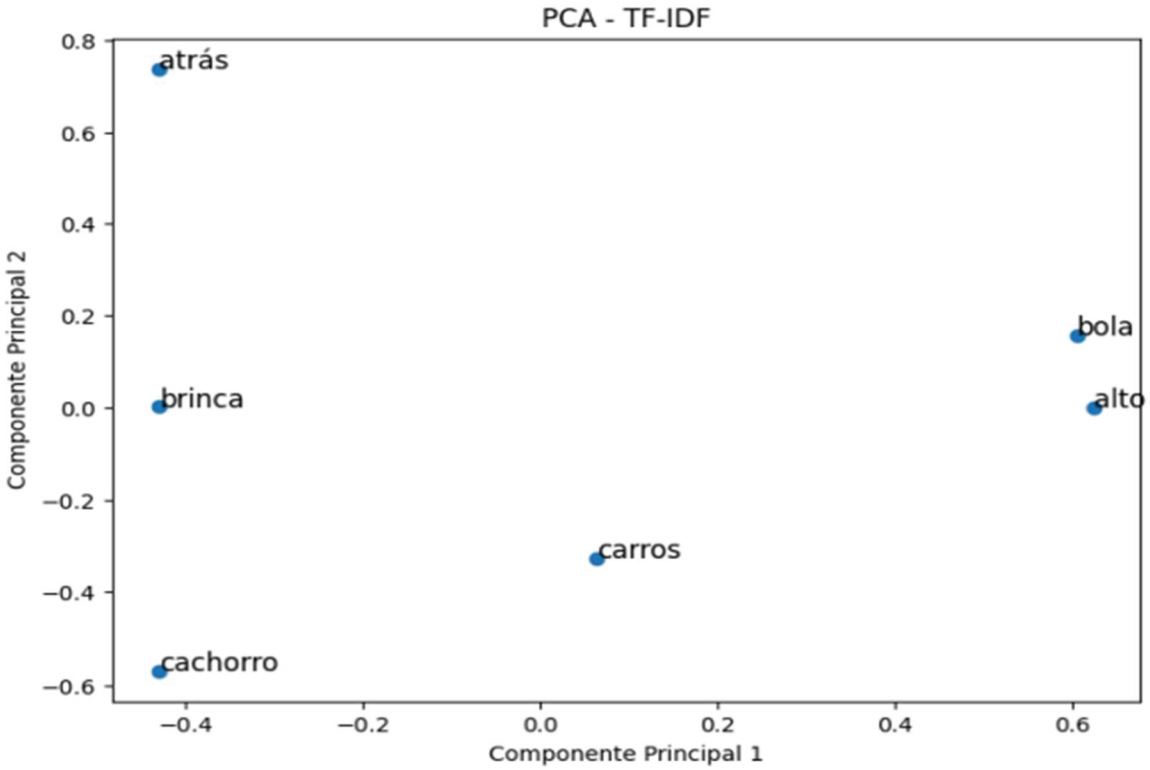
FIGURA 12 – SOLUÇÃO FINAL



FONTE: O autor (2025).

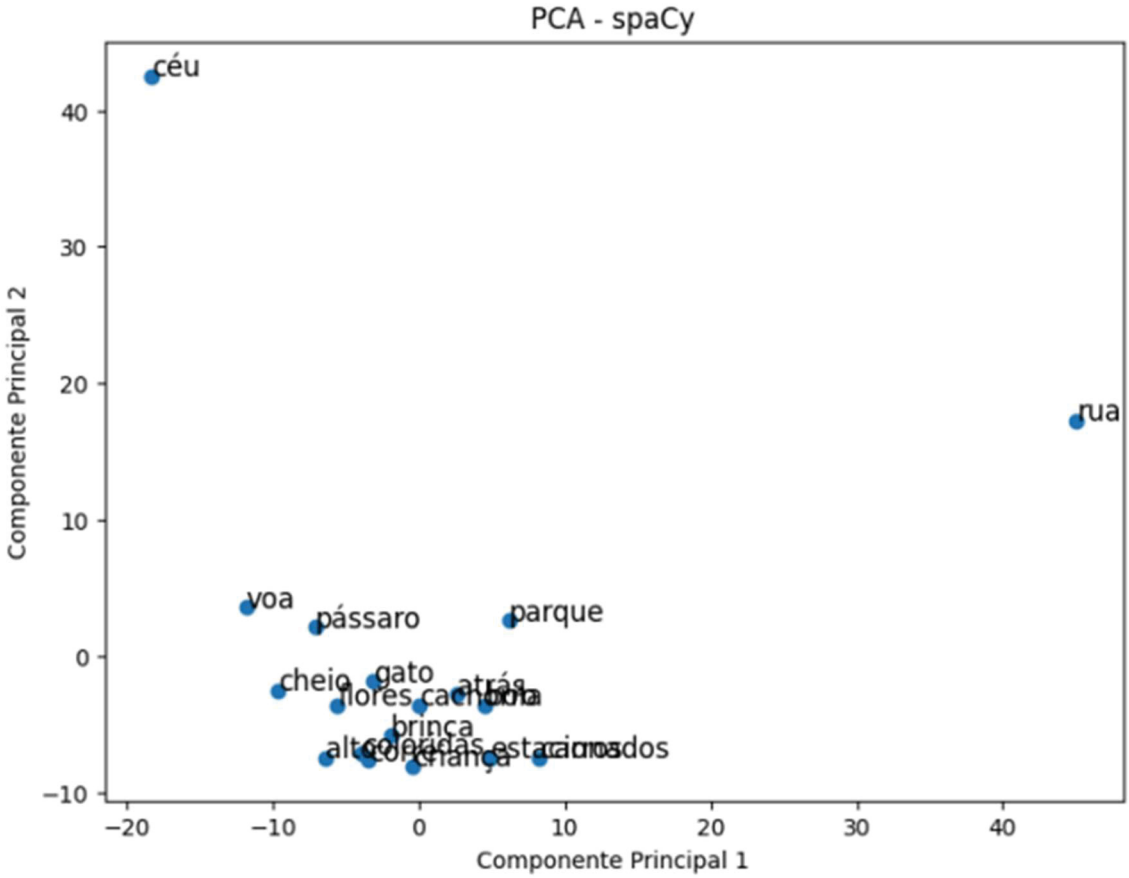
2)

FIGURA 13 – COMPONENTES PRINCIPAIS



FONTE: O autor (2025).

FIGURA 14 – COMPOENTES PRINCIPAIS



FONTE: O autor (2025).