

UNIVERSIDADE FEDERAL DO PARANÁ

IVO ASSMANN JUNIOR

MEMORIAL DE PROJETOS: ANÁLISE EXPLORATÓRIA DE DADOS

CURITIBA

2025

IVO ASSMANN JUNIOR

MEMORIAL DE PROJETOS: ANÁLISE EXPLORATÓRIA DE DADOS

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.

Orientador: Prof. Dr. Jaime Wojciechowski

CURITIBA

2025

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 **IVO ASSMANN JUNIOR**, intitulada: **MEMORIAL DE PROJETOS: ANÁLISE EXPLORATÓRIA DE DADOS**, 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, 11 de Setembro de 2025.



JAIME WOJCIECHOWSKI

Presidente da Banca Examinadora



RAFAELA MANTOVANI FONTANA

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

A análise exploratória de dados representa uma das etapas mais fundamentais no processo de descoberta de conhecimento e construção de soluções baseadas em ciência de dados. Seu principal objetivo é permitir uma compreensão inicial dos dados por meio da investigação de suas características, distribuição, correlações e possíveis inconsistências. Ao longo do curso, percebi a importância desse processo não apenas como uma fase preliminar, mas como um recurso estratégico para orientar decisões e escolhas metodológicas mais assertivas. Através de técnicas como estatísticas descritivas, visualização gráfica e identificação de padrões, a análise exploratória oferece insights valiosos que muitas vezes passam despercebidos em análises superficiais. Além disso, ela desempenha papel crucial na detecção de valores atípicos, dados ausentes e possíveis vieses que poderiam comprometer resultados posteriores. Refletindo de forma pessoal, compreendo que essa prática amplia a visão crítica e analítica do profissional, já que exige não apenas domínio técnico, mas também sensibilidade para interpretar os contextos em que os dados estão inseridos. Para mim, trabalhar com análise exploratória significa desenvolver a habilidade de dialogar com os dados, de modo a construir narrativas coerentes e embasadas, capazes de apoiar tanto a pesquisa acadêmica quanto a tomada de decisão em cenários organizacionais. Por isso, considero que esse tema é central dentro da ciência de dados e que seu estudo contribuiu de forma significativa para a consolidação do meu aprendizado e para o desenvolvimento das competências necessárias à minha formação.

Palavras-chave: análise exploratória de dados; ciência de dados; estatística descritiva; visualização de dados; tomada de decisão.

ABSTRACT

Exploratory data analysis is one of the most fundamental stages in the process of knowledge discovery and the development of data science-based solutions. Its main objective is to provide an initial understanding of the dataset by investigating its characteristics, distribution, correlations, and potential inconsistencies. Throughout the course, I realized the importance of this process not only as a preliminary step, but also as a strategic resource that guides decisions and methodological choices with greater accuracy. By applying techniques such as descriptive statistics, graphical visualization, and pattern identification, exploratory analysis provides valuable insights that often go unnoticed in more superficial approaches. Furthermore, it plays a crucial role in detecting outliers, missing values, and potential biases that could compromise later results. From a personal perspective, I understand that this practice enhances critical and analytical thinking, as it requires not only technical skills but also the ability to interpret the contexts in which data is embedded. For me, working with exploratory data analysis means developing the capacity to "dialogue with the data," building coherent and evidence-based narratives that support both academic research and organizational decision-making. Therefore, I consider this topic central within the field of data science, and its study has significantly contributed to consolidating my learning process and to the development of the competencies necessary for my professional formation.

Keywords: exploratory data analysis; data science; descriptive statistics; data visualization; decision-making.

SUMÁRIO

1 PARECER TÉCNICO.....	7
REFERÊNCIAS.....	10
APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL	11
APÊNDICE 2 – LINGUAGEM DE PROGRAMAÇÃO APLICADA	18
APÊNDICE 3 – LINGUAGEM R	32
APÊNDICE 4 – ESTATÍSTICA APLICADA I.....	39
APÊNDICE 5 – ESTATÍSTICA APLICADA II.....	46
APÊNDICE 6 – ARQUITETURA DE DADOS	49
APÊNDICE 7 – APRENDIZADO DE MÁQUINA.....	55
APÊNDICE 8 – DEEP LEARNING	66
APÊNDICE 9 – BIG DATA.....	82
APÊNDICE 10 – VISÃO COMPUTACIONAL	86
APÊNDICE 11 – ASPECTOS FILOSÓFICOS E ÉTICOS DA IA.....	90
APÊNDICE 12 – GESTÃO DE PROJETOS DE IA.....	97
APÊNDICE 13 – FRAMEWORKS DE INTELIGÊNCIA ARTIFICIAL	100
APÊNDICE 14 – VISUALIZAÇÃO DE DADOS E STORYTELLING	114
APÊNDICE 15 – TÓPICOS EM INTELIGÊNCIA ARTIFICIAL	116

1 PARECER TÉCNICO

A Análise Exploratória de Dados (AED) ocupa papel central dentro do ciclo de análise estatística e da ciência de dados, sendo reconhecida como um processo fundamental para a compreensão inicial dos conjuntos de informações. Não se trata apenas de um conjunto de técnicas preparatórias que antecedem a modelagem estatística ou o uso de algoritmos de inteligência artificial. Ao contrário, representa um momento de descoberta, de formulação de hipóteses preliminares e de interpretação crítica, no qual os dados revelam padrões, irregularidades e relações não evidentes à primeira vista. Segundo Tukey (1977), a AED deve ser entendida como um processo investigativo contínuo, em vez de um simples requisito técnico a ser cumprido.

A análise exploratória assume caráter investigativo, no qual cada conjunto de dados é tratado como um objeto a ser decifrado. Nesse contexto, o analista busca valores ausentes, detecta inconsistências, identifica padrões recorrentes e examina correlações que podem passar despercebidas em uma primeira análise superficial. Cada gráfico elaborado, cada estatística descritiva calculada e cada medida de dispersão observada funciona como um indício que aproxima da essência do conjunto de informações. Essa etapa é indispensável para reduzir vieses interpretativos e garantir maior robustez às conclusões (Fávero; Belfiore, 2017).

Outro aspecto de grande relevância consiste no uso de técnicas de visualização de dados. Recursos como histogramas, diagramas de caixa (*boxplots*) e gráficos de dispersão (*scatterplots*) permitem observar de forma intuitiva e direta a distribuição, os agrupamentos e as discrepâncias presentes nos dados. McKinney (2017) salienta que a visualização exploratória é uma das ferramentas mais eficazes para identificar estruturas subjacentes e direcionar análises subsequentes. Wickham (2016), ao discutir a importância do *ggplot2* como ferramenta de visualização, reforça que gráficos não devem ser considerados apenas como complementos ilustrativos, mas como instrumentos de raciocínio estatístico.

No âmbito prático, a AED apresenta-se como etapa crítica para a tomada de decisão baseada em dados. Ao identificar valores extremos (*outliers*), erros de

digitação, duplicidades ou padrões ocultos, evita-se que informações equivocadas distorçam os resultados de análises posteriores. Gelman e Hill (2007) afirmam que a investigação exploratória é essencial para a formulação de hipóteses adequadas e para a seleção de modelos estatísticos consistentes. Assim, a ausência dessa fase pode comprometer a validade de todo o processo analítico.

Além disso, a AED possibilita compreender a qualidade e a confiabilidade das bases de dados utilizadas. Em muitos projetos de ciência de dados, especialmente em ambientes corporativos, os registros coletados podem apresentar falhas decorrentes de processos manuais de entrada, sistemas integrados ou até mesmo erros de coleta automatizada. Nesse sentido, a análise exploratória atua como um filtro inicial, capaz de revelar a extensão desses problemas e fornecer subsídios para as etapas de limpeza e pré-processamento dos dados. James et al. (2013) destacam que uma modelagem eficiente depende de bases bem compreendidas e estruturadas, sendo a AED indispensável para esse diagnóstico preliminar.

O valor da AED não se restringe ao campo acadêmico, mas se estende a diferentes áreas aplicadas. Na saúde, por exemplo, análises exploratórias podem revelar padrões epidemiológicos em dados clínicos, permitindo identificar surtos ou tendências de doenças. Na economia, a exploração inicial de séries temporais auxilia na detecção de ciclos e anomalias que impactam diretamente a formulação de políticas públicas. No marketing, a AED contribui para compreender o comportamento do consumidor, agrupando perfis de clientes e revelando preferências implícitas. Hair et al. (2019) destacam que a análise exploratória deve ser vista como etapa estratégica que orienta decisões baseadas em evidências concretas.

Outro ponto a ser destacado é que a AED possui caráter iterativo. Diferentemente de etapas rigidamente sequenciais, a exploração de dados não ocorre de forma linear e definitiva. Muitas vezes, o processo exige revisões constantes, retornos às etapas anteriores e refinamentos sucessivos. Esse movimento cíclico é necessário porque novas informações emergem a cada análise, o que leva à reformulação de hipóteses ou até mesmo à redefinição dos objetivos da

pesquisa. Essa visão está alinhada ao que defendem Cleveland (1993) e Chambers (1998), ao apontarem que a AED deve ser compreendida como um ciclo dinâmico, em constante interação com o objeto de estudo.

É importante ressaltar que a Análise Exploratória de Dados transcende o caráter meramente técnico, assumindo uma dimensão metodológica e reflexiva. Ela estimula uma postura investigativa, baseada no questionamento da confiabilidade das informações, na atenção aos detalhes e na disposição para identificar o inesperado. Ao transformar dados brutos em conhecimento aplicável, a AED se consolida como ponto de partida essencial para análises estatísticas inferenciais e preditivas.

Em síntese, pode-se afirmar que a Análise Exploratória de Dados não deve ser vista apenas como um procedimento inicial, mas como um eixo estruturante de qualquer estudo analítico. Sua prática assegura maior clareza, solidez e confiabilidade às interpretações, permitindo que decisões, sejam elas científicas ou organizacionais, sejam fundamentadas em evidências consistentes. Dessa forma, a AED configura-se não apenas como etapa metodológica, mas como exercício de reflexão crítica, sem o qual a ciência de dados e a estatística aplicada perderiam grande parte de sua robustez e utilidade prática.

REFERÊNCIAS

CHAMBERS, J. M. **Programming with Data**: A Guide to the S Language. New York: Springer, 1998.

CLEVELAND, W. S. **Visualizing Data**. Summit: Hobart Press, 1993.

FÁVERO, L. P.; BELFIORE, P. **Manual de Análise de Dados**: Estatística e Modelagem Multivariada com Excel®, SPSS® e Stata. Rio de Janeiro: Elsevier, 2017.

GELMAN, A.; HILL, J. **Data Analysis Using Regression and Multilevel/Hierarchical Models**. Cambridge: Cambridge University Press, 2007.

HAIR, J. F. et al. **Análise Multivariada de Dados**. 7. ed. Porto Alegre: Bookman, 2019.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning**: with Applications in R. New York: Springer, 2013.

MCKINNEY, W. **Python for Data Analysis**: Data Wrangling with Pandas, NumPy, and IPython. 2. ed. Sebastopol: O'Reilly Media, 2017.

TUKEY, J. W. **Exploratory Data Analysis**. Reading: Addison-Wesley, 1977.

WICKHAM, H. ggplot2: **Elegant Graphics for Data Analysis**. New York: Springer, 2016.

APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

A – ENUNCIADO

1 ChatGPT

- (6,25 pontos)** Pergunte ao ChatGPT o que é Inteligência Artificial e cole aqui o resultado.
- (6,25 pontos)** Dada essa resposta do ChatGPT, classifique usando as 4 abordagens vistas em sala. Explique o porquê.
- (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.
- (6,25 pontos)** Entendendo o que é o ChatGPT, classifique o próprio ChatGPT usando as 4 abordagens vistas em sala. Explique o porquê.

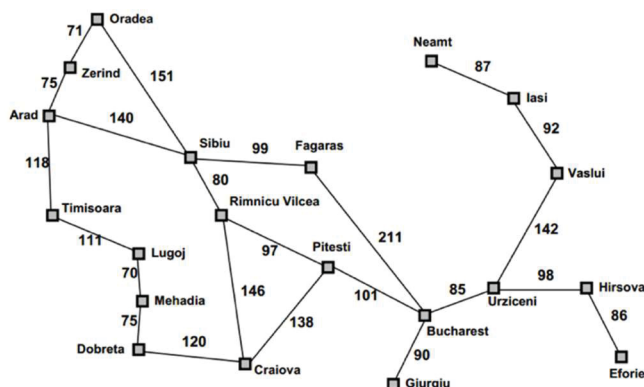
2 Busca Heurística

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.

- (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: $\alpha \rightarrow \beta, \beta \rightarrow \gamma \vdash \alpha \rightarrow \gamma$

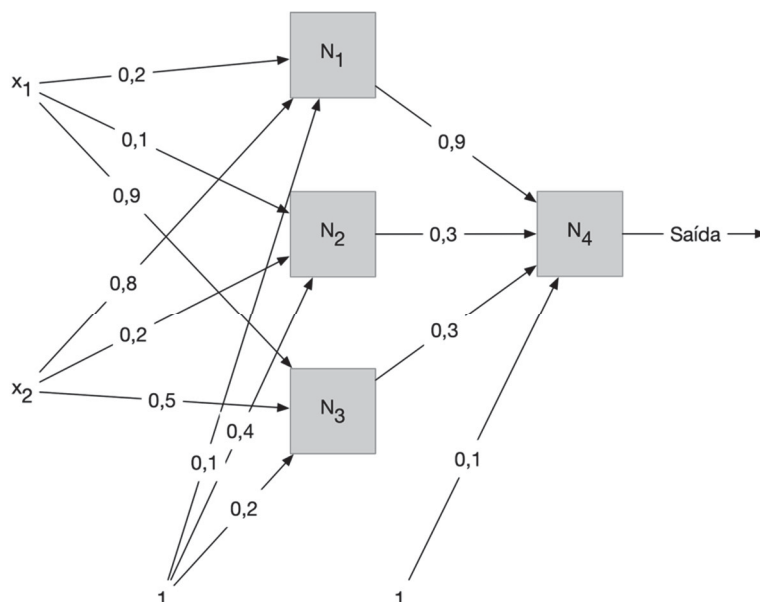
Conjunção: $\alpha, \beta \vdash \alpha \wedge \beta$

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

Questão 1:

a) Model GPT 3.5 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 requerem inteligência humana. Esses sistemas são projetados para simular processos cognitivos humanos, como aprendizado, raciocínio, resolução de problemas, percepção e compreensão da linguagem natural. A IA abrange uma ampla gama de técnicas e abordagens, incluindo aprendizado de máquina, redes neurais, processamento de linguagem natural, visão computacional e muito mais. Seu objetivo é criar sistemas que possam aprender e se adaptar a novas situações, tornando-se cada vez mais eficientes e autônomos em suas operações. A IA tem aplicações em diversos setores, como saúde, finanças, automotivo, jogos, assistentes virtuais, entre outros.

Model GPT-4 Inteligência Artificial (IA) refere-se à simulação da inteligência humana em máquinas que são programadas para pensar como humanos e imitar suas ações. O termo também pode ser aplicado a qualquer máquina que exiba traços associados à mente humana, como aprendizado e resolução de problemas. A capacidade ideal de uma máquina de IA é realizar tarefas que normalmente requerem inteligência humana, incluindo reconhecimento de padrões, interpretação de linguagem natural, raciocínio, aprendizado a partir da experiência, adaptação a novas situações e a capacidade de manipular objetos no mundo físico. A IA é um campo amplo com muitas subáreas, incluindo:

1. Aprendizado de Máquina (Machine Learning): Uma abordagem de IA que envolve o treinamento de algoritmos para aprender padrões em dados e fazer previsões ou decisões baseadas em dados novos, sem ser explicitamente programado para realizar a tarefa.

2. Processamento de Linguagem Natural (PLN): Uma subárea da IA focada em dar às máquinas a capacidade de entender, interpretar e responder à linguagem humana de forma útil.

3. Visão Computacional: A capacidade de máquinas e sistemas para derivar informações significativas de imagens digitais, vídeos e outros inputs visuais e tomar ações ou fazer recomendações baseadas nessa informação.

4. Robótica Inteligente: A integração da IA em robôs, permitindo que eles realizem tarefas complexas, aprendam com suas experiências e interajam com o ambiente de maneira mais sofisticada.

A IA pode variar desde sistemas simples, como uma calculadora ou um aplicativo de reconhecimento de voz, até sistemas extremamente complexos, como carros autônomos ou robôs com capacidade de aprendizado profundo. O desenvolvimento da IA promete transformar muitos aspectos da vida cotidiana e tem aplicação em diversos campos, incluindo medicina, educação, finanças, logística, arte, entretenimento e muito mais.

b) Conforme a resultado acima classifco a resposta do ChatGPT em AGIR RACIONALMENTE, pois o objetivo GPT é sempre fornecer uma resposta com o melhor resultado e rapidamente conforme as informações fornecidas, ou seja, ele adapta da situação (no caso o usuário), sempre se preocupando em tomar uma ação racional. Ao contrário da abordagem AGIR COMO HUMANO, onde o objetivo é imitar o comportamento humano, algo que não é esperado de uma IA, no qual o objetivo é ter resultados precisos, racionais e rápidos diante das mais diversas situações. O mesmo podemos desconsiderar a abordagem de PENSAR COMO OS HUMANOS, cujo objetivo é implementar fielmente o processo de pensamento, e PENSAR RACIONALMENTE, que visa sempre fornecer um resultado logicamente correto.

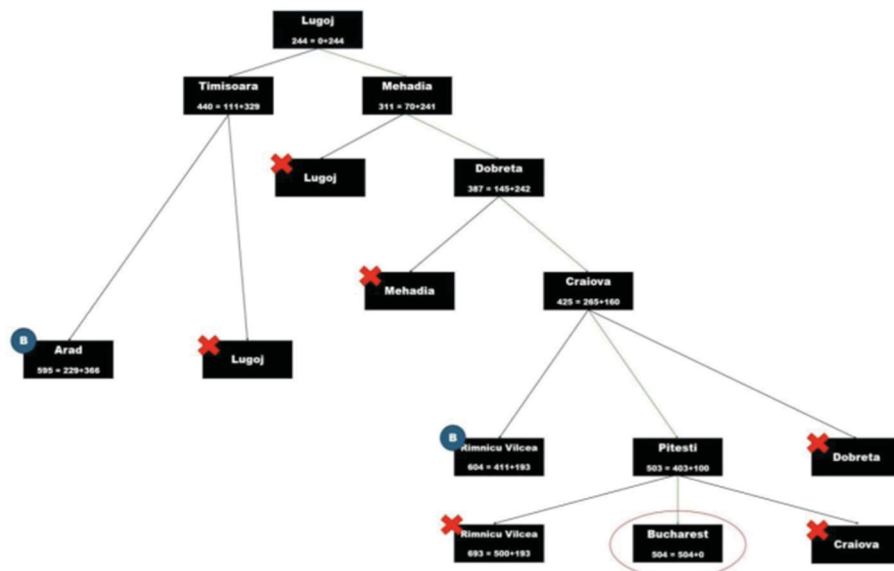
c) ChatGPT é uma inteligência artificial baseada em Generative Pre-Trained Transformer, traduzido do inglês significa Transformador Generativo PréTreinado, disponibilizado via chatbot por meio de uma interface web e via API REST, permitindo a integração com diversas aplicações. O ChatGPT foi treinado com uma vasta gama de informações (dados), permitindo que ele compreenda e responda a diversas perguntas dos mais variados temas, por exemplo, matemática, saúde, pontos turísticos e entre outros. Por isso, ele também é considerado um modelo de IA conversacional e linguagem natural. Os modelos de redes neurais utilizados na criação do ChatGPT são baseados na técnica transformer mencionada no artigo Vaswani et al. (2017). Com isso, modelos do ChatGPT conseguem realizar geração de textos, interpretação de áudios, criação de imagens, análise de imagens se baseando nas informações fornecidas pelo usuário via plataforma web ou API rest.

d) Agir racionalmente (classificação definida)

O ChatGPT se alinha mais estreitamente com a abordagem de agir racionalmente. Ele é projetado para realizar tarefas de maneira lógica e eficiente, produzindo respostas que atendem aos requisitos de uma dada solicitação de acordo com seu treinamento e as diretrizes predefinidas. Essa capacidade de gerar respostas apropriadas e úteis num vasto domínio de conhecimento, otimizando para objetivos específicos (como responder a perguntas, fornecer informações, criar imagens, interpretar áudio, etc.), exemplifica a ação racional conforme definido na inteligência artificial.

Questão 2:

a)



Questão 3:

Provar que: $p \rightarrow q, q \rightarrow r, \sim r \vee p \vdash q \leftrightarrow p$

R1 : $p \rightarrow q$

R2 : $q \rightarrow r$

R3 : $\sim r \vee p$

R4 : $r \rightarrow p$ (EI R3)

R5 : $q \rightarrow p$ (SH R2 e R4)

R6 : $(p \rightarrow q) \wedge (q \rightarrow p)$ (CONJ R1 e R5)

R7 : $p \leftrightarrow q$ (BICOND R6)

Conclusão: Q: $q \leftrightarrow p$

O argumento é VÁLIDO A raposa come as uvas se e somente se as uvas caem.

Questão 4:

a) (6,25 pontos) Valor de saída do neurônio N1

$$N1 = (0,2 \times -3) + (0,8 \times 1) + (0,1 \times 1)$$

$$N1 = (-0,6) + 0,8 + 0,1$$

$$N1 = 0,3$$

b) (6,25 pontos) Valor de saída do neurônio N2

$$N2 = (0,1 \times -3) + (0,2 \times 1) + (0,4 \times 1)$$

$$N2 = (-0,3) + 0,2 + 0,4$$

$$N2 = 0,3$$

c) (6,25 pontos) Valor de saída do neurônio N3

$$N3 = (0,9 \times -3) + (0,5 \times 1) + (0,2 \times 1)$$

$$N3 = (-2,7) + 0,5 + 0,2$$

$$N3 = -2,00$$

d) (6,25 pontos) Valor de saída da rede como um todo

$$N4 = (0,9 \times 0,3) + (0,3 \times 0,3) + (0,3 \times -2,0) + (0,1 \times 1)$$

$$N4 = 0,27 + 0,09 + (-0,6) + 0,1$$

$$N4 = -0,14$$

$$N4 = \tanh(-0,14) = -0,13909$$

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:

- Carregue a base de dados **media_precos_carros_brasil.csv**
- Verifique se há valores faltantes nos dados. Caso haja, escolha uma tratativa para resolver o problema de valores faltantes
- Verifique se há dados duplicados nos dados
- Crie duas categorias, para separar colunas numéricas e categóricas. Imprima o resumo de informações das variáveis numéricas e categóricas (estatística descritiva dos dados)
- Imprima a contagem de valores por modelo (model) e marca do carro (brand)
- Dê uma 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:

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

- 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

Questão 1:

a)

```
dados = pd.read_csv('precos_carros_brasil.csv')
dados.shape
(267542, 11)
```

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_br1      65245
dtype: int64
```

```
dados = dados.dropna(axis=0)
dados.shape
(202297, 11)
```

c)

```
dados.duplicated().sum()
2
```

```
dados = dados.drop_duplicates()
dados.shape
(202295, 11)
```

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_br1
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

	year_of_reference	year_model	avg_price_br1
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

```
dados[categoricas_cols].describe()
```

	month_of_reference	fipe_code	authentication	brand	model	fuel	gear	engine_size
count	202295	202295	202295	202295	202295	202295	202295	202295
unique	12	2091	202295	6	2112	3	2	29
top	January	003281-6	cfzlcztzfwrcp	Fiat	Palio Week: Adv/Adv TRYON 1.8 mpi Flex	Gasoline	manual	1,6
freq	24260	425	1	44962	425	168684	161883	47420

e)

```
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
```

```
contagem_modelos = dados['model'].value_counts()
total_modelos_unicos = len(contagem_modelos)
```

```
print("Total de modelos únicos:", total_modelos_unicos)
```

```
Total de modelos únicos: 2112
```

```
dados['brand'].value_counts()
```

```
brand
```

```
Fiat          44962
```

```
VW - Volkswagen  44312
```

```
GM - Chevrolet  38590
```

```
Ford          33150
```

```
Renault       29191
```

```
Nissan         12090
```

```
Name: count, dtype: int64
```

f)

Ao carregar o arquivo, identificou-se um total de 267.542 linhas e 11 colunas.

Posteriormente, observou-se que 65.245 linhas continham valores f)

faltantes, as quais foram removidas, resultando em 202.297 linhas. Além disso, foram

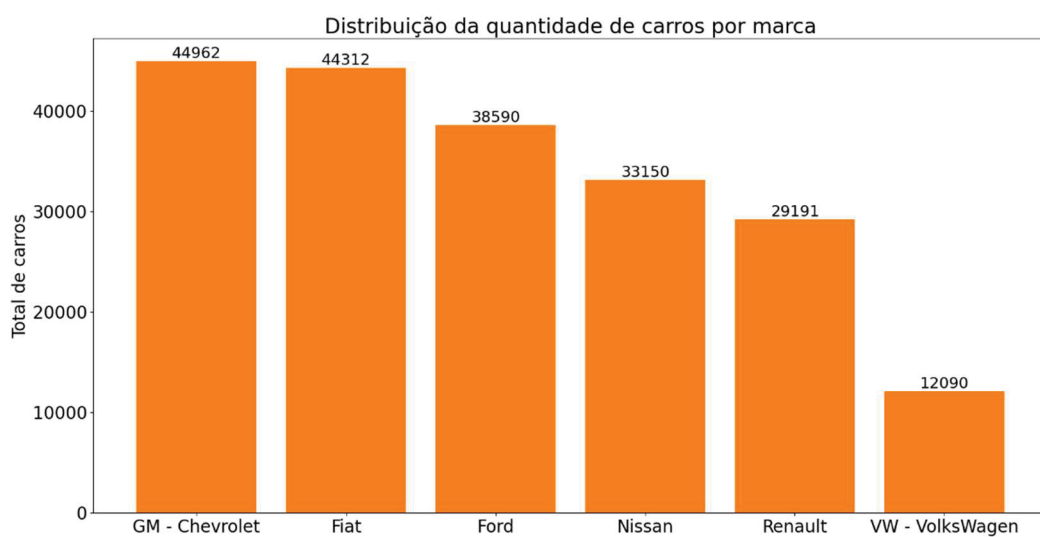
detectadas e excluídas 2 linhas duplicadas, resultando em 202.295 linhas únicas. Foi

realizada uma contagem de valores por modelo, totalizando 2.112, e por marca, totalizando

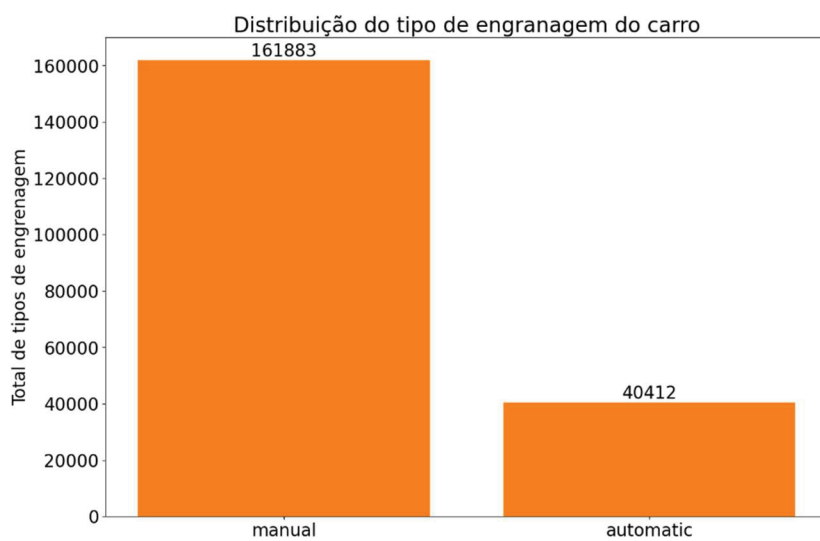
6.

Questão 2:

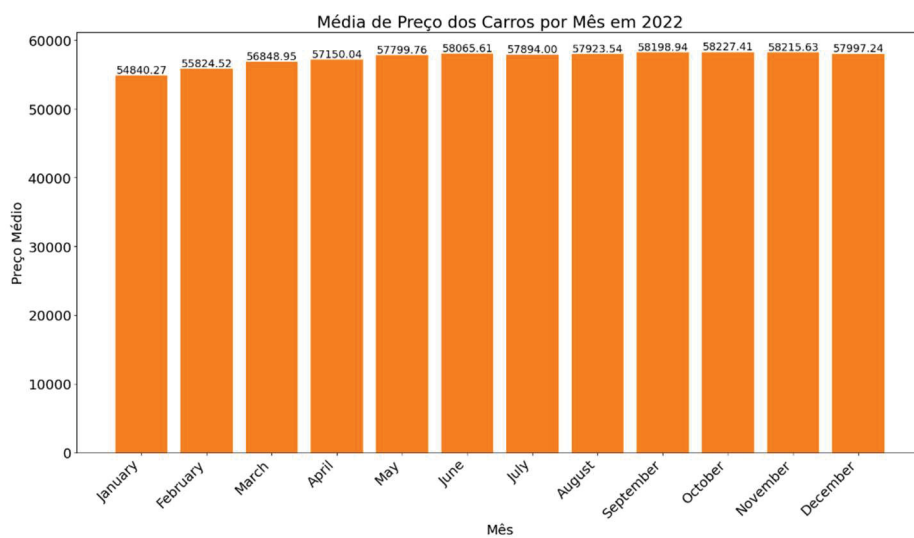
a)



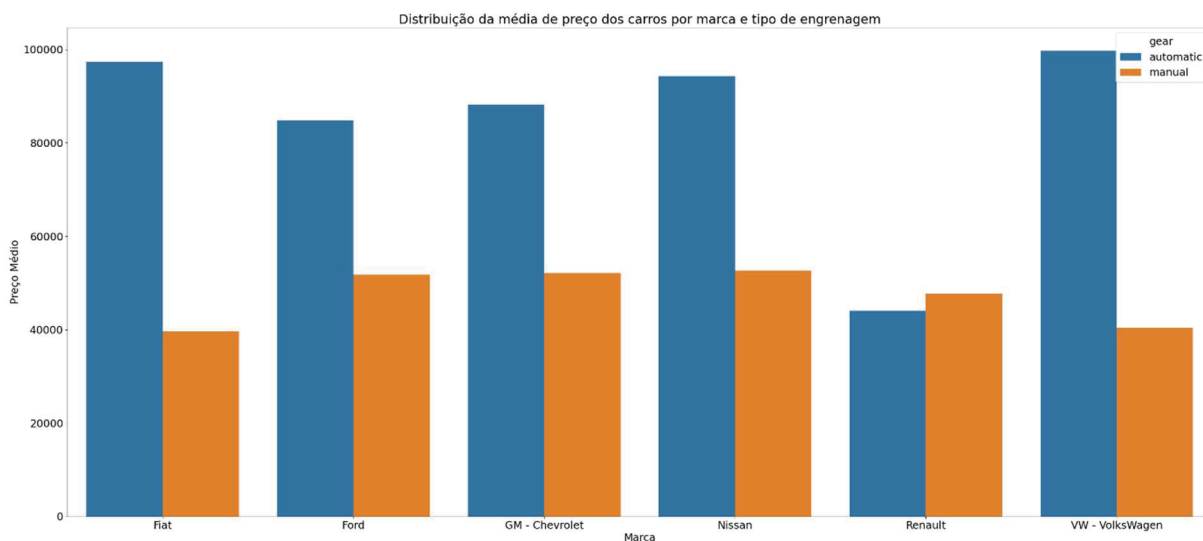
b)



c)



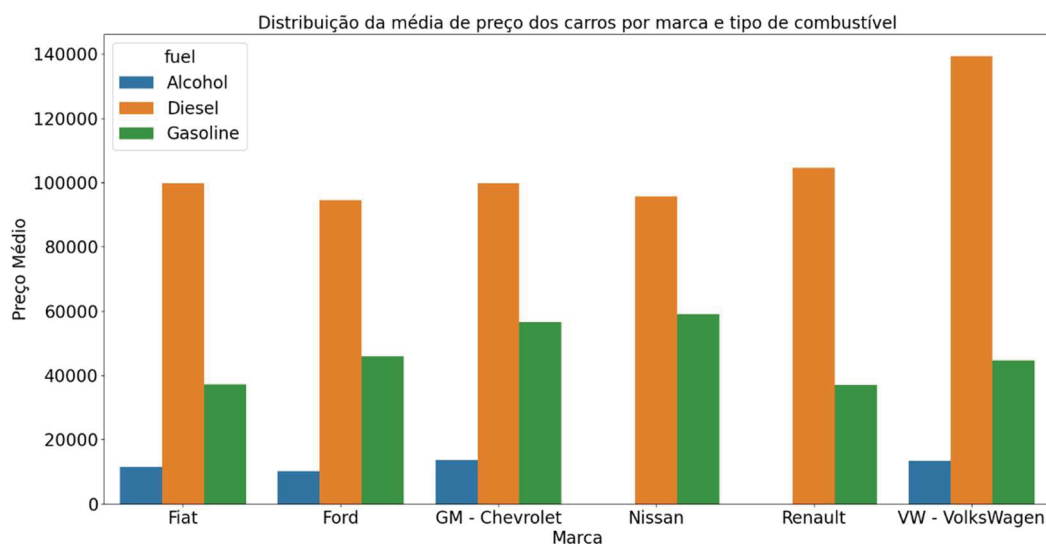
d)



e)

No gráfico da média de preço dos carros por marca e tipo de engrenagem, destaca-se que a Volkswagen (VW) lidera com o valor médio mais alto em carros automáticos, seguida pela Nissan em carros manuais. Enquanto isso, a Renault e a Fiat registram os valores médios mais baixos, respectivamente, nessas categorias. Essa análise evidencia as variações nos preços médios de acordo com a marca e o tipo de transmissão.

f)



g)

No gráfico de distribuição da média de preço dos carros por marca e tipo de combustível, destaca-se que o Diesel é o tipo mais caro, seguido por Gasolina e Álcool, respectivamente. As marcas com os valores mais elevados nas respectivas categorias são Volkswagen (VW),

Nissan e GM - Chevrolet. Por outro lado, as marcas com os valores mais baixos são Ford, Renault e Ford, respectivamente. Essa análise evidencia as variações nos preços médios conforme o tipo de combustível e a marca do carro.

Questão 3:

a)

```
#brand          object
#model          object
#fuel          object
#gear          object
#engine_size    object
```

```
dadosPredicao['brand']=
abelEncoder().fit_transform(dadosPredicao['brand'])
dadosPredicao['model']=
abelEncoder().fit_transform(dadosPredicao['model'])
dadosPredicao['fuel']                                     =
LabelEncoder().fit_transform(dadosPredicao['fuel'])
dadosPredicao['gear']                                     =
LabelEncoder().fit_transform(dadosPredicao['gear'])
dadosPredicao['engine_size']= LabelEncoder().fit_transform(dadosPredicao['engine_size'])
```

```
dadosPredicao.value_counts()
```

year_of_reference	brand	model	fuel	gear	engine_size	year_model	avg_price_brl	
2021	5	667	2	1	5	2014	27590.0	10
	1	1415	1	0	16	2022	236250.0	8
	3	1728	2	0	8	2022	98633.0	7
		1334	1	0	12	2009	70000.0	6
		1003	2	0	5	2022	71489.0	6
	4	967	2	1	5	2017	44451.0	1
		968	2	1	0	2008	15830.0	1
							15976.0	1
							16089.0	1
2023	5	2111	2	1	0	2017	49807.0	1

Name: count, Length: 201105, dtype: int64

b)

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25,
random_state = 42)
```

```
print(X_train.shape)
```

```
X_train.head(10)
```

	year_of_reference	brand	model	fuel	gear	engine_size	year_model
156364	2022	0	364	1	1	10	2020
93758	2021	4	1099	2	1	5	2011
43313	2021	1	549	2	1	8	2008
114972	2022	0	385	1	1	10	2013
69993	2021	4	970	2	1	0	2018
125124	2022	1	898	2	1	0	2016
171836	2022	0	1994	2	1	0	2013
87814	2021	5	1695	2	1	5	2017
59771	2021	1	1469	1	1	15	2007
85504	2021	4	204	2	1	0	2004

```
print(X_test.shape)
```

```
X_test.head(10)
```

	year_of_reference	brand	model	fuel	gear	engine_size	year_model
180633	2022	0	1235	2	1	5	2015
13130	2021	4	224	2	1	0	2005
163315	2022	0	1984	2	1	2	2003
121464	2022	2	1114	2	1	6	2008
14044	2021	4	310	2	0	5	2019
79483	2021	5	1373	2	1	8	2012
104655	2022	2	1879	1	1	8	2004
84287	2021	1	451	1	1	21	2007
103710	2022	5	717	2	1	0	2012
100088	2022	1	460	1	1	24	2006

```
Y_test.head()
```

```
180633    42595.0
13130     10989.0
163315     9087.0
121464    26965.0
14044     57102.0
Name: avg_price_brl, dtype: float64
```

c)



d)

```
valores_preditos_rf = model_rf.predict(X_test)
valores_preditos_xgboost = model_xgboost.predict(X_test)
valores_preditos_rf_parametros = model_rf_parametros.predict(X_test)
```

e)

Random Forest

	importance
engine_size	0.452190
year_model	0.393490
model	0.057937
gear	0.032744
fuel	0.032458
brand	0.018702
year_of_reference	0.012478

XGboost

	importance
engine_size	0.443018
year_model	0.197655
fuel	0.154716
gear	0.120949
brand	0.048413
model	0.020340
year_of_reference	0.014909

Random Forest com alteração de parâmetros

	importance
engine_size	0.466002
year_model	0.411574
model	0.038422
fuel	0.034104
gear	0.023301
brand	0.016202
year_of_reference	0.010395

f)

Na análise de importância de variáveis, observou-se uma consistência em relação à influência das características do veículo, como o tamanho do motor e o ano do modelo, na predição do preço médio. Além disso, fatores como o tipo de combustível e a transmissão também desempenharam um papel significativo. A marca do veículo, o modelo específico e o ano de referência da observação, tiveram uma importância menor, indicando que outros atributos podem ser mais relevantes na determinação do preço médio.

g)

Random Forest

```
mse = mean_squared_error(Y_test, valores_preditos_rf)
```

```
mse
```

```
20013214.759729225
```

```
mae = mean_absolute_error(Y_test, valores_preditos_rf)
```

```

mae
2336.850652570212

r2_score(Y_test, valores_preditos_rf)
0.9925636190237587

# XGBoost
mse = mean_squared_error(Y_test, valores_preditos_xgboost)
mse
39450171.01302586

mae = mean_absolute_error(Y_test, valores_preditos_xgboost)
mae
3669.041110367435

r2_score(Y_test, valores_preditos_xgboost)
0.9853413604584382

# Random Forest com alteração de parâmetros
mse = mean_squared_error(Y_test, valores_preditos_rf_parametros)
mse
146317605.61954153

mae = mean_absolute_error(Y_test, valores_preditos_rf_parametros)
mae
4557.4308940321735

r2_score(Y_test, valores_preditos_rf_parametros)
0.9456322498918177

```

h)

O modelo Random Forest teve o menor MAE (2336.85) e o maior R^2 (0.992563), indicando melhor precisão e explicação da variabilidade dos dados. O XGBoost apresentou um desempenho ligeiramente inferior ao Random Forest, mas ainda muito bom. A alteração dos parâmetros do Random Forest não resultou em um melhor desempenho

Modelo	Métrica	Resultado
Random Forest	MAE	2336.850652570212
Random Forest	MSE	20013214.759729225
Random Forest	R ²	0.9925636190237587
Random Forest com alteração de parâmetros	MAE	4557.4308940321735
Random Forest com alteração de parâmetros	MSE	146317605.61954153
Random Forest com alteração de parâmetros	R ²	0.9456322498918177
XGBoost	MAE	3669.041110367435
XGBoost	MSE	39450171.01302586
XGBoost	R ²	0.9853413604584382

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:

$$\text{Volume} = b_0 + b_1 * \text{dap}^2 * \text{Ht}$$

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 b_0 e b_1 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: $\text{Volume} = b_0 + b_1 * \text{dap}^2 * \text{Ht}$

```
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

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

onde y_i é o valor observado, $\hat{y}_i$ 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;

- $S_{yx}\%$

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

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

7. Escolha o melhor modelo.

B – RESOLUÇÃO

Questão 1:

1)

```
data(Satellite)
```

.

2)

```
dados_selecionados <- Satellite[, c("x.17", "x.18", "x.19", "x.20",  
"classes")]
```

```
cat("\nEstrutura inicial dos dados:\n")
```

```
str(dados_selecionados)
```

```
set.seed(123)
```

```

indice <- createDataPartition(dados_selecionados$classes, p = 0.8, list =
FALSE)
dados_treino <- dados_selecionados[indice, ]
dados_teste <- dados_selecionados[-indice, ]

```

3)

```

cat("Treinando Modelo Random Forest:\n")
modelo_rf <- randomForest(classes ~ ., data = dados_treino)
cat("\nTreinando Modelo SVM:\n")
modelo_svm <- svm(classes ~ ., data = dados_treino, kernel = "radial")
cat("\nTreinando Modelo RNA:\n")
modelo_rna <- caret::train(classes ~ ., data = dados_treino, method =
"nnet")

```

4)

O modelo SVM foi identificado como o melhor dos 3 modelos, pois leva um tempo de execução semelhante ao RandomForest, porém tem uma acurácia levemente superior de acordo com a comparação entre as matrizes de confusão. O RNA(nnet) teve um acurácia inferior ao SVM e ao RandomForest e ainda levou muito mais tempo de execução.

Complementando a escolha pelo SVM, a equipe fez testes usando o pacote caret para o treinar os modelos RandomForest e SVM, porém o treinamento levava vários minutos, o que dificulta a execução de testes e por isso optamos por utilizar os pacotes “randomForest” para o RandomForest e os pacotes “e1071” e “kernlab” para usar no SVM. Estes pacotes deixaram a execução muitos mais rápida, pois levava segundos, enquanto usando o pacote caret, levava alguns minutos para os mesmos algoritmos. Adicionalmente, além do ganho de tempo, os resultados foram semelhantes em relação a acurácia dos modelos.

5)

As métricas utilizadas para escolher o SVM como o melhor modelo para usar neste conjunto de dados foram a acurácia apresentada nas matrizes de confusão em conjunto com o tempo de execução do treinamento. Abaixo constam os resultados das acurácias da matriz de confusão de cada modelo que foram usadas para a escolha do melhor modelo.

```

Resumo dos Modelos> print(métricas)
      Modelo Acuracia
1 Random Forest 0.8512461
2          SVM 0.8551402
3          RNA 0.8045171

```

Questão 2:**1)**

```
volumes <- read.csv2("http://www.razer.net.br/datasets/Volumes.csv", sep =
";", header = TRUE)
```

2)

```
volumes <- volumes[, -1] # Exclui a primeira coluna
```

3)

```
set.seed(123)
indice <- createDataPartition(volumes$VOL, p = 0.8, list = FALSE)
```

4)

```
# Criar conjuntos de treinamento e teste
```

```
dados_treino <- volumes[indice, ]
```

```
dados_teste <- volumes[-indice, ]
```

```
cat(paste("\nQuantidade de dados no conjunto de treino:",
nrow(dados_treino)))
```

```
cat(paste("\nQuantidade de dados no conjunto de teste:",
nrow(dados_teste)))
```

```
# Treinar os modelos
```

```
# Random Forest
```

```
cat("\n\nTreinando Modelo Random Forest:\n\n")
```

```
modelo_rf <- train(VOL ~ ., data = dados_treino, method = "rf")
```

```
# SVM
```

```
cat("Treinando Modelo SVM\n\n")
```

```
modelo_svm <- train(VOL ~ ., data = dados_treino, method = "svmRadial")
```

```
# Redes Neurais
```

```
cat("Treinando Modelo Redes Neurais\n\n")
```

```
modelo_neuralnet <- neuralnet(VOL ~ ., data = dados_treino)
```

```
cat("Treinando Modelo alométrico de SPURR \n\n")
```

```
# Modelo alométrico de SPURR
```

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

5)

```
# Efetuar as previsões nos dados de teste
pred_rf <- predict(modelo_rf, newdata = dados_teste)
pred_svm <- predict(modelo_svm, newdata = dados_teste)
pred_neuralnet <- predict(modelo_neuralnet, newdata = dados_teste)
pred_alom <- predict(alom, newdata = dados_teste)
```

6)

```
# Criar funções para calcular as métricas
```

```
# Coeficiente de determinação: R2
```

```
calcular_R2 <- function(y_real, y_predito) {
  media_y_real <- mean(y_real)
  ss_tot <- sum((y_real - media_y_real)^2)
  ss_res <- sum((y_real - y_predito)^2)
  r2 <- 1 - (ss_res / ss_tot)
  return(r2)
}
```

```
# Erro padrão da estimativa: Syx
```

```
calcular_Syx <- function(y_real, y_predito) {
  n <- length(y_real)
  syx <- sqrt(sum((y_real - y_predito)^2) / (n - 2))
  return(syx)
}
```

```
# Syx%
```

```
calcular_Syx_percent <- function(y_real, y_predito) {
  syx <- calcular_Syx(y_real, y_predito)
  media_y_real <- mean(y_real)
  syx_percent <- (syx / media_y_real) * 100
  return(syx_percent)
}
```

```

# Calculando as métricas para cada modelo
R2_rf <- calcular_R2(dados_teste$VOL, pred_rf)
Syx_rf <- calcular_Syx(dados_teste$VOL, pred_rf)
Syx_percent_rf <- calcular_Syx_percent(dados_teste$VOL, pred_rf)

R2_svm <- calcular_R2(dados_teste$VOL, pred_svm)
Syx_svm <- calcular_Syx(dados_teste$VOL, pred_svm)
Syx_percent_svm <- calcular_Syx_percent(dados_teste$VOL, pred_svm)

R2_neuralnet <- calcular_R2(dados_teste$VOL, pred_neuralnet)
Syx_neuralnet <- calcular_Syx(dados_teste$VOL, pred_neuralnet)
Syx_percent_neuralnet <- calcular_Syx_percent(dados_teste$VOL,
pred_neuralnet)

R2_alom <- calcular_R2(dados_teste$VOL, pred_alom)
Syx_alom <- calcular_Syx(dados_teste$VOL, pred_alom)
Syx_percent_alom <- calcular_Syx_percent(dados_teste$VOL, pred_alom)

```

7)

Após o treinamento dos modelos Random Forest (rf), SVM (svmRadial), Redes Neurais (neuralnet) e modelo alométrico de SPURR, o modelo usando Redes Neurais foi considerado o melhor, pois foi superior aos demais em todas as métricas de comparação, pois foi o que ficou mais próximo a 1 na métrica R2 e nas métricas Syx e Syx% foi o que ficou mais próximo de zero. A escolha se baseou nos resultados da tabela abaixo:

	Modelo	R2	Syx	Syx_percent
3	Redes Neurais	0.9099191	0.1208231	8.976552
4	Alométrico de SPURR	0.8694429	0.1454567	10.806705
1	Random Forest	0.8486654	0.1566040	11.634890
2	SVM	0.7899082	0.1845178	13.708744

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 frequências 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:

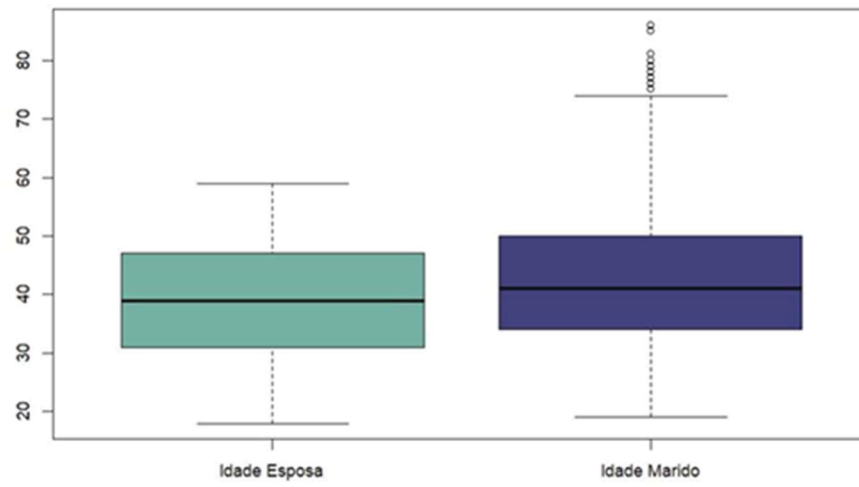
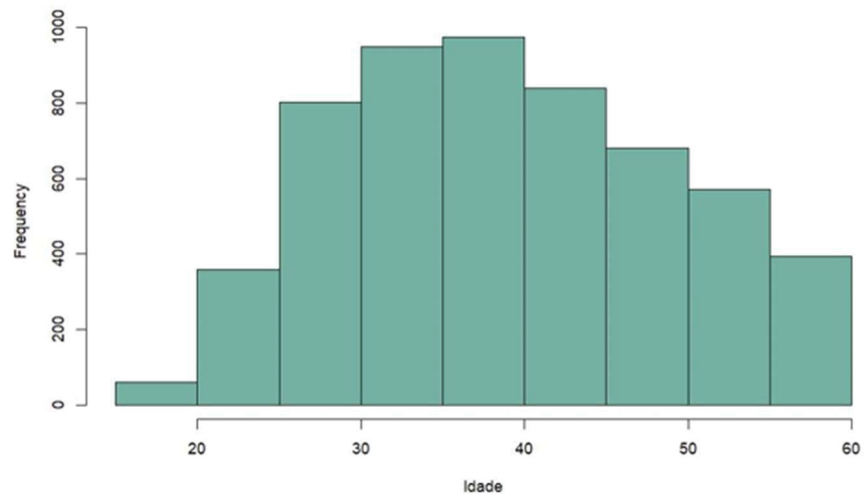
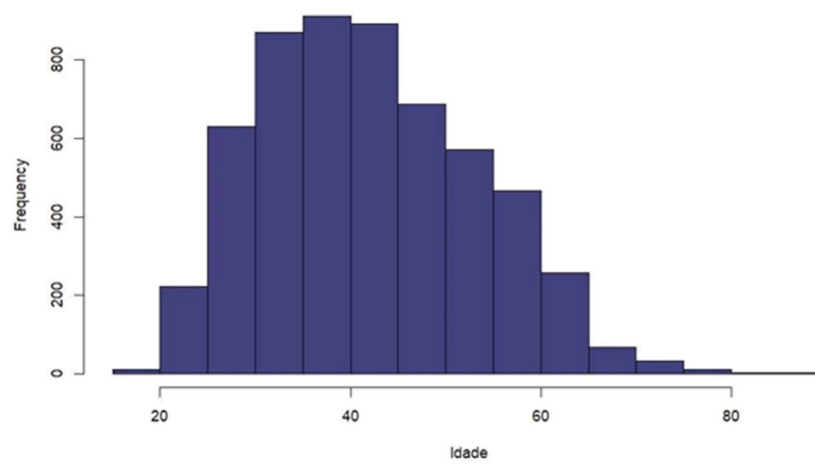
Você deve fazer os testes necessários (e mostra-los no documento pdf) para saber se você deve usar o unpaired test (paramétrico) ou o teste U de Mann-Whitney (não paramétrico), justifique sua resposta sobre a escolha.

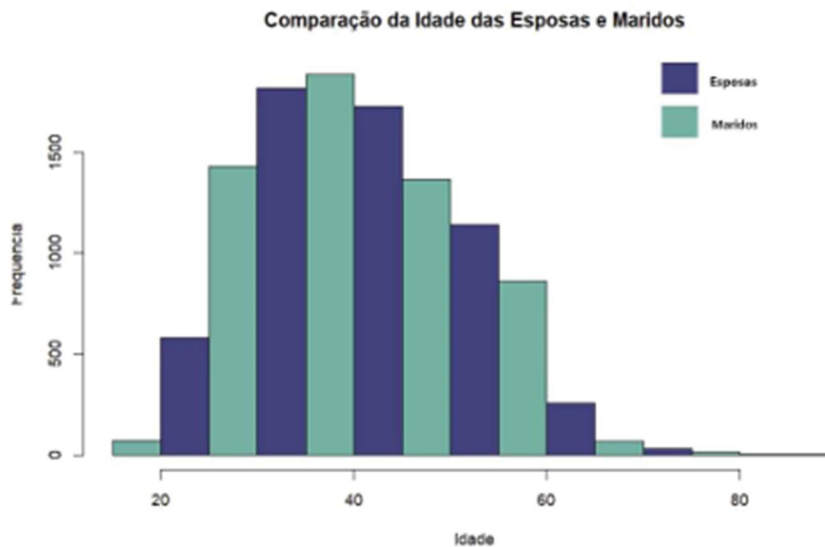
Lembre-se de que os intervalos de confiança já são mostrados nos resultados dos testes citados no item 1 acima.

B – RESOLUÇÃO

Questão 1:

a)

Boxplot das Idades da Esposa e do Marido**Idade da Esposa****Idade do Marido**



Comparação dos Histogramas

No histograma com as idades dos maridos existe uma distribuição maior dos dados, visto que existe uma maior variedade de valores observados. Os valores mais elevados podem ser considerados outliers devido a distância que possuem da média porém a grande parte dos dados tem maior proximidade com a média.

No histograma com as idades das esposas existe uma concentração ainda maior em relação a média dos dados, além disso existem uma menor variedade de valores observados.

Comparação dos gráficos box plot

O limite inferior do gráfico com as idades dos maridos é por volta de 20 anos e o limite superior é por volta de 80 anos. Nos valores próximos a 80 anos existem alguns outliers, que já foram identificados anteriormente na análise do histograma. # O limite superior do gráfico das esposas é menor em relação ao gráfico dos maridos, chegando apenas a 60 anos. O limite inferior também é por volta de 20 anos, ficando semelhante com o limite inferior dos maridos. Nessa amostra das idades das esposas não constam outliers.

b)

Mostrando a distribuicao de frequencia

```
> print (tabela_husage)
  class limits    f   rf rf(%)   cf  cf(%)
[18.81,23.671) 102 0.02  1.81  102   1.81
[23.671,28.531) 466 0.08  8.27  568  10.08
[28.531,33.392) 809 0.14 14.36 1377  24.44
[33.392,38.253) 895 0.16 15.89 2272  40.33
[38.253,43.114) 917 0.16 16.28 3189  56.60
[43.114,47.974) 629 0.11 11.16 3818  67.77
[47.974,52.835) 649 0.12 11.52 4467  79.29
[52.835,57.696) 541 0.10  9.60 5008  88.89
[57.696,62.556) 394 0.07  6.99 5402  95.88
[62.556,67.417) 152 0.03  2.70 5554  98.58
[67.417,72.278)  51 0.01  0.91 5605  99.49
[72.278,77.139)  21 0.00  0.37 5626  99.86
[77.139,81.999)   6 0.00  0.11 5632  99.96
[81.999,86.86)   2 0.00  0.04 5634 100.00
```

```
> print(tabela_age)
  class limits    f   rf rf(%)   cf  cf(%)
[17.82,20.804)  61 0.01  1.08   61   1.08
[20.804,23.787) 161 0.03  2.86  222   3.94
[23.787,26.771) 312 0.06  5.54  534   9.48
[26.771,29.754) 505 0.09  8.96 1039  18.44
[29.754,32.738) 562 0.10  9.98 1601  28.42
[32.738,35.721) 571 0.10 10.13 2172  38.55
[35.721,38.705) 624 0.11 11.08 2796  49.63
[38.705,41.689) 510 0.09  9.05 3306  58.68
[41.689,44.672) 542 0.10  9.62 3848  68.30
[44.672,47.656) 432 0.08  7.67 4280  75.97
[47.656,50.639) 389 0.07  6.90 4669  82.87
[50.639,53.623) 358 0.06  6.35 5027  89.23
[53.623,56.606) 304 0.05  5.40 5331  94.62
[56.606,59.59)  303 0.05  5.38 5634 100.00
```

Comparação das Tabelas de Frequência

A maior frequência das idades das esposas está entre 35,721 e 38,705 (624 ocorrências) e em seguida o intervalo entre 32,738 e 35,721 (571 ocorrências). A menor ocorrência está no intervalo entre 17,82 e 20,804, contado com apenas 61 ocorrências.

Na frequência das idades dos maridos constam 917 ocorrências nos intervalos de 38,253 e 43,114, portanto esse intervalo tem a maior frequência. A menor frequência está no intervalo entre 81,999 e 86,86, contando com apenas 2 ocorrências. Outra observação são os intervalos entre 72,278 e 77,139 e entre 77,139 e 81,999 que constam com apenas 6 e 2 ocorrências respectivamente. #Esses dados citados por último se caracterizam como os outliers da amostra.

Questão 2:

a)

```

      Variavel      Media Mediana Moda
1 Idade da Esposa 39.42758      39   37
2 Idade do Marido 42.45296      41   44
> |

```

Ao analisar as idades de esposas e maridos, percebemos que os maridos geralmente são mais velhos com idade máxima de 86 e para as esposas a idade máxima é 59.

```

> summary(salarios$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00  31.00   39.00  39.43  47.00   59.00
> summary(salarios$husage)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  19.00  34.00   41.00  42.45  50.00   86.00

```

Com uma média de 42,45 anos para os maridos e 39,43 anos para as esposas, a diferença é clara. A mediana também suporta isso, mostrando maridos com 41 anos e esposas com 39.

Isso indica que, mesmo removendo outliers, a tendência de os maridos serem mais velhos se mantém. Além disso, a moda nos dá uma diferença maior ainda, com 44 anos para maridos e 37 para esposas, o que pode sugerir uma maior dispersão nas idades dos maridos.

Esses números mostram uma tendência consistente de os maridos serem mais velhos nas relações.

b)

Variavel	variância	Desvio_Padrao	Coeficiente_de_Variacao
Idade da Esposa	99.75234	9.98761	25.33153
Idade do Marido	126.07173	11.22817	26.44849

É notável que os maridos são mais velhos, mas também exibem maior variabilidade em suas idades. A análise das variâncias e desvios padrões das idades dá uma ideia de como esses valores se espalham ao redor da média, e os resultados são bem interessantes.

A variância das idades dos maridos é de 126.07, enquanto a das esposas é de 99.75. Isso mostra que as idades dos maridos estão mais espalhadas, indicando uma diversidade maior. O desvio padrão, dá uma ideia da dispersão em relação à média, segue o mesmo padrão: 11.23 para maridos contra 9.99 para esposas.

Além disso, o coeficiente de variação, nos ajuda a comparar a dispersão entre conjuntos de dados com médias diferentes. Os maridos têm um coeficiente de 26.44%, enquanto as esposas têm 25.33%. Isso significa que, relativamente, a idade dos maridos varia mais em torno da média do que a das esposas.

Esses números revelam a heterogeneidade dentro dos grupos. Uma maior variância e coeficiente de variação nos maridos sugerem que existe uma gama mais ampla de idades entre eles, talvez refletindo normas sociais ou culturais que influenciam a idade em que os homens se casam em comparação com as mulheres.

Questão 3:

```
>lillie.test(salarios$age)

> lillie.test(salarios$age)

Lilliefors (Kolmogorov-Smirnov) normality test

data:  salarios$age
D = 0.058909, p-value < 0.00000000000000022

>lillie.test(salarios$husage)
> lillie.test(salarios$husage)

Lilliefors (Kolmogorov-Smirnov) normality test

data:  salarios$husage
D = 0.059662, p-value < 0.00000000000000022

>
> |
```

Após análise de normalidade das variáveis “husage” e “age” utilizando o teste Lilliefors e Jaquer-Bera, no qual ambos tiveram resultado de $p\text{-value} < 0.00000000000000022$, que é menor que o nível de significância 0,05.

Podemos concluir que a idade mediana dos homens é estatisticamente diferente da idade mediana das mulheres (rejeitamos a hipótese nula de que as localizações das distribuições são iguais). Então, optou-se pela escolha do teste não paramétrico Mann-Whitney "U" test.

O intervalo de confiança da diferença entre as medianas está entre 2.000033 e 3.000024, isso significa que há uma confiança de 95% de que as localizações das distribuições estão entre esse intervalo.

A estimativa da diferença nas localizações das distribuições é de 2,999966. Isso indica que a média da variável “age” (idade das esposas) é aproximadamente 3 unidades menor que a média da variável “husage” (idade dos maridos).

Por isso, infere-se que, há uma diferença estatisticamente significativa na distribuição da variável Idade entre os grupos de Maridos ("husage") e Esposas ("age").

#Segue abaixo o resultado do teste.

```
> print(test_result)

wilcoxon rank sum test with continuity correction

data:  Idade by Grupo
W = 18122044, p-value < 0.000000000000000022
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 2.000033 3.000024
sample estimates:
difference in location
      2.999966
```

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 logaritmo, portanto você não precisa aplicar o logaritmo nela para fazer as regressões, mas é necessário aplicar o antilog para obter o resultado da predição.

B – RESOLUÇÃO

Rotina utilizada – Modelos Ridge, Lasso e ElasticNet

1. **Carregamento e preparação dos dados:**
 - O conjunto de dados trabalhosalarios foi carregado e armazenado como `dat`.
 - A estrutura do dataset foi explorada com funções como `glimpse()` e `summary()`.
2. **Particionamento da amostra:**
 - A base foi dividida aleatoriamente em:
 - **80% para treinamento**, e
 - **20% para teste**, usando uma semente aleatória fixa (`set.seed(302)`), garantindo reprodutibilidade dos resultados.
3. **Padronização das variáveis:**
 - As variáveis numéricas contínuas foram padronizadas (média zero e desvio padrão um) usando a função `preProcess()` do pacote `caret`.
 - Variáveis dummies/binárias não foram padronizadas.
4. **Tratamento de variáveis categóricas:**
 - Variáveis categóricas foram convertidas em variáveis dummies por meio da função `dummyVars()` para permitir o uso em modelos penalizados (Ridge, Lasso e ElasticNet).
5. **Separação entre variáveis dependente e explicativas:**
 - A variável dependente foi `lwage` (logaritmo do salário-hora da esposa).
 - As demais variáveis disponíveis foram utilizadas como explicativas, estruturadas em matrizes (`x`, `x_test`) e vetores (`y_train`, `y_test`).
6. **Ajuste dos modelos:**
 - **Ridge**: Regressão com penalização L2, ajustada com `glmnet` utilizando `alpha = 0`. O valor ótimo de λ foi selecionado via validação cruzada.
 - **Lasso**: Regressão com penalização L1, ajustada com `glmnet` utilizando `alpha = 1`. Também selecionado via validação cruzada.
 - **ElasticNet**: Modelo misto com penalização L1 e L2. Foi ajustado com a função `train()` do pacote `caret`, com `method = "glmnet"`, realizando busca automática dos melhores valores de `alpha` e `lambda` por validação cruzada com 10 folds e 5 repetições.
7. **Avaliação de desempenho dos modelos:**
 - As métricas de desempenho aplicadas foram:
 - **RMSE (Root Mean Squared Error)**: medida do erro quadrático médio.
 - **R² (Coeficiente de Determinação)**: mede o grau de explicação da variância da variável resposta.
 - Os modelos foram avaliados tanto nos dados de **treinamento** quanto de **teste**, por meio de uma função definida manualmente.
8. **Predição com perfil específico:**

- Realizou-se uma predição do salário-hora da esposa com base em valores fornecidos no enunciado (idade, escolaridade, cor, experiência, união, filhos, etc.).
- As variáveis contínuas foram padronizadas com os parâmetros do conjunto de treinamento.
- O resultado predito (em escala padronizada e logarítmica) foi **despadronizado e exponenciado** para retornar à escala original (US\$).

9. Intervalo de confiança:

- Foi construído um **intervalo de confiança de 95%** para o salário predito, com base na variabilidade da variável resposta e no tamanho da amostra.
- Os limites inferior e superior do intervalo também foram transformados para a escala original

Modelo	RMSE	R ²	Clwr (Intervalo Inferior)	Clupr (Intervalo Superior)	Predição var. "lwage" (salário-hora esposa)
Ridge Time: 1.37s Lambda: 0.0316	Treino: 0.8411446 Teste: 0.9893328	Treino: 0.292132 Teste: 0.259084	8.493945	8.87351	8.681654
Lasso Time: 1.55s Lambda: 0.01	Treino: 0.8420298 Teste: 0.9894877	Treino: 0.2906413 Teste: 0.258852	7.84636	8.196938	8.01971
ElasticNet Time: 7.91s Lambda: 0.0125	Treino: 0.8410462 Teste: 0.9894186	Treino: 0.2922976 Teste: 0.2589555	8.587557	8.971305	8.777334

Considerando a aplicação dos modelos Ridge, Lasso e ElasticNet, após criação de nova matriz com os valores da tabela acima, o melhor desempenho para a estimativa do valor hora de salário para a esposa foi o do modelo Ridge, pois este modelo apresentou um RMSE (Erro Quadrático Médio) levemente inferior aos demais e também apresentou um R² (R Quadrado) mais próximo de 1, indicando que se ajusta melhor aos dados observados.

Outra métrica considerada para a escolha do modelo Ridge foi o tempo de execução, pois o modelo Ridge levou menos tempo e ainda apresentou resultados melhores. O modelo Lasso foi levemente superior em relação aos intervalos de confiança, porém em um contexto geral o Ridge se mostrou melhor na maioria das métricas avaliadas.

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 disponibilizado 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

Questão 1:

		0	1
0	How was your weekend?	inglês	
1	Do you speak English?	inglês	
2	I need to buy some groceries.	inglês	
3	Meu time de futebol favorito ganhou o jogo.	português	
4	Vamos sair para jantar no sábado.	português	
...		...	...
87	Quiero aprender italiano.	espanhol	
88	O restaurante tem uma vista incrível.	português	
89	Me gustaría ir de vacaciones.	espanhol	
90	The cat is sleeping.	inglês	
91	I want to learn French.	inglês	

92 rows × 2 columns

0	0.333333	0.500000	0.000000	0.000000	0.200000	inglês
1	0.333333	0.500000	0.200000	0.000000	0.200000	inglês
2	0.200000	0.250000	0.000000	0.000000	0.142857	inglês
3	0.142857	0.166667	0.222222	0.222222	0.000000	português
4	0.200000	0.250000	0.285714	0.285714	0.000000	português
...	...	...	...	...	...	...
87	0.500000	1.000000	0.000000	0.000000	0.000000	espanhol
88	0.200000	0.250000	0.285714	0.142857	0.000000	português
89	0.250000	0.333333	0.166667	0.166667	0.000000	espanhol
90	0.333333	0.500000	0.000000	0.000000	0.400000	inglês
91	0.250000	0.333333	0.000000	0.000000	0.166667	inglês

92 rows × 6 columns

↩ Acurácia nos dados de treinamento: 91.30%

```
[[19 0 3]
 [ 1 21 1]
 [ 1 0 23]]
```

	precision	recall	f1-score	support
espanhol	0.90	0.86	0.88	22
inglês	1.00	0.91	0.95	23
português	0.85	0.96	0.90	24
accuracy			0.91	69
macro avg	0.92	0.91	0.91	69
weighted avg	0.92	0.91	0.91	69

métricas mais confiáveis

```
[[7 0 1]
 [1 5 1]
 [1 0 7]]
```

	precision	recall	f1-score	support
espanhol	0.78	0.88	0.82	8
inglês	1.00	0.71	0.83	7
português	0.78	0.88	0.82	8
accuracy			0.83	23
macro avg	0.85	0.82	0.83	23
weighted avg	0.85	0.83	0.83	23

Por favor, insira um texto (máximo de 500 caracteres): trabalho de arquitetura de dados
 O seu texto está escrito no idioma português.

Acurácia geral: 80.00%

Matriz de Confusão:

```
[[4 1 0]
 [0 5 0]
 [0 2 3]]
```

Relatório de Classificação:

	precision	recall	f1-score	support
inglês	1.00	0.80	0.89	5
espanhol	0.62	1.00	0.77	5
português	1.00	0.60	0.75	5
accuracy			0.80	15
macro avg	0.88	0.80	0.80	15
weighted avg	0.88	0.80	0.80	15

Questão 2:

```
# Imprimindo as 5 primeiras linhas
titanic.head()
```

```
survived  pclass   sex  age  sibsp  parch   fare  embarked  class  who  adult_male  deck  embark_town  alive  alone
0         0       3  male  22.0    1     0   7.2500         S   Third   man         True   NaN   Southampton    no   False
1         1       1  female  38.0    1     0  71.2833         C   First  woman        False    C    Cherbourg   yes   False
2         1       3  female  26.0    0     0   7.9250         S   Third  woman        False   NaN   Southampton   yes    True
3         1       1  female  35.0    1     0  53.1000         S   First  woman        False    C    Southampton   yes   False
4         0       3  male   35.0    0     0   8.0500         S   Third   man         True   NaN   Southampton    no    True
```

```
[ ] # Observando número de linhas e colunas
titanic.shape
```

```
(891, 15)
```

```
[ ] # Selecionar as features (X) e o target (y)
x = titanic.drop('survived', axis=1)
y = titanic['survived']
```

```
[ ] x.head()
```

```
pclass  sex  age  sibsp  parch   fare  embarked  class  who  adult_male  deck  embark_town  alive  alone
0       3   1  22.0    1     0   7.2500         2     2   1         1     7         2     0     0
1       1   0  38.0    1     0  71.2833         0     0   2         0     2         0     1     0
2       3   0  26.0    0     0   7.9250         2     2   2         0     7         2     1     1
3       1   0  35.0    1     0  53.1000         2     0   2         0     2         2     1     0
4       3   1  35.0    0     0   8.0500         2     2   1         1     7         2     0     1
```

```
[ ] y.head()
```

```
0     0
1     1
2     1
3     1
4     0
Name: survived, dtype: int64
```

```
[ ] # Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Inicializar e treinar o classificador SVM
svm_classifier = SVC()
svm_classifier.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = svm_classifier.predict(X_test)
```

Verificando a Acurácia do Modelo

```
[ ] # Calcular a acurácia do modelo em porcentagem
accuracy_percentage = accuracy_score(y_test, y_pred) * 100
print("Acurácia do modelo:", accuracy_percentage, "%")
```

↗ Acurácia do modelo: 62.93706293706294 %

```
[ ] # Calcular a matriz de confusão
conf_matrix = confusion_matrix(y_test, y_pred)
print("Matriz de Confusão:")
print(conf_matrix)
```

↗ Matriz de Confusão:

```
[[71 16]
 [37 19]]
```

✓ Melhorando a base de dados

```
[ ] # Verificando a prevalência de sobreviventes
# A porcentagem resultante representa a prevalência de sobreviventes na base de dados do Titanic.
# Com 40.62% entendemos que os dados estão razoavelmente balanceados.
prevalence = titanic['survived'].mean() * 100 # Calculando a média da coluna 'survived' e multiplicando por 100 para obter a porcentagem
print("Prevalência de sobreviventes: {:.2f}%".format(prevalence))
```

↗ Prevalência de sobreviventes: 40.62%

```
[ ] # Criar uma nova coluna 'FamilySize' representando o tamanho da família
titanic['FamilySize'] = titanic['sibsp'] + titanic['parch'] + 1
```

```
• # Agrupamento de idade
idade = [0, 12, 18, 30, 50, 200]
labels = ['Child', 'Teenager', 'Young Adult', 'Adult', 'Senior']
titanic['AgeGroup'] = pd.cut(titanic['age'], bins=idade, labels=labels)

titanic['AgeGroup'] = LabelEncoder().fit_transform(titanic['AgeGroup'])
```

```
[ ] # Imprimindo as 5 primeiras linhas
titanic.head()
```

↗

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	FamilySize	AgeGroup
0	0	3	1	22.0	1	0	7.2500	2	2	1	1	7	2	0	0	2	4
1	1	1	0	38.0	1	0	71.2833	0	0	2	0	2	0	1	0	2	0
2	1	3	0	26.0	0	0	7.9250	2	2	2	0	7	2	1	1	1	4
3	1	1	0	35.0	1	0	53.1000	2	0	2	0	2	2	1	0	2	0

```
[ ] # Manter apenas as colunas relevantes
#titanic = titanic[['survived', 'pclass', 'sex', 'embarked', 'FamilySize', 'AgeGroup' ]]
# Manter apenas as colunas relevantes
titanic = titanic[['survived', 'pclass', 'sex', 'FamilySize', 'AgeGroup' ]]
```

```
[ ] # Imprimindo as 5 primeiras linhas
titanic.head()
```

```
survived  pclass  sex  FamilySize  AgeGroup
0         0      3    1           2         4
1         1      1    0           2         0
2         1      3    0           1         4
3         1      1    0           2         0
4         0      3    1           1         0
```

```
[ ] # Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Inicializar e treinar o classificador SVM
svm_classifier = SVC()
svm_classifier.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = svm_classifier.predict(X_test)
```

```
[ ] # Calcular a acurácia do modelo em porcentagem
accuracy_percentage = accuracy_score(y_test, y_pred) * 100
print("Acurácia do modelo:", accuracy_percentage, "%")
```

```
Acurácia do modelo: 79.72027972027972 %
```

Matriz de Confusão Base Tratada

```
[75 12]
```

```
[17 39]
```

```
[ ] # Calcular a matriz de confusão
conf_matrix = confusion_matrix(y_test, y_pred)
print("Matriz de Confusão:")
print(conf_matrix)
```

```
Matriz de Confusão:
[[75 12]
 [17 39]]
```

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

Classificação

Veículo

Técnica	Parâmetro	Acurácia	Matriz de Confusão
SVM – Hold-out	C=1 Sigma=0.06	0.784	Confusion Matrix and Statistics Reference Prediction bus opel saab van bus 41 2 1 0 opel 0 20 9 0 saab 0 19 32 1 van 2 1 1 38

RNA – CV	size=5 decay=0.1	0.7545	Reference Prediction bus opel saab van bus 38 0 1 0 opel 1 26 17 1 saab 4 16 25 1 van 0 0 0 37
RF – CV	mtry=2	0.7485	Confusion Matrix and Statistics Reference Prediction bus opel saab van bus 42 3 0 0 opel 0 21 15 0 saab 0 18 25 2 van 1 0 3 37
RF – Hold-out	mtry=2	0.7365	Confusion Matrix and Statistics Reference Prediction bus opel saab van bus 42 3 1 0 opel 0 20 15 0 saab 0 19 24 2 van 1 0 3 37
SVM – CV	C=0.5 Sigma=0.06	0.713	Confusion Matrix and Statistics Reference Prediction bus opel saab van bus 40 2 1 0 opel 0 12 10 0 saab 0 27 30 2 van 3 1 2 37
KNN	k=1	0.647	Confusion Matrix and Statistics Reference Prediction bus opel saab van bus 39 2 4 4 opel 2 18 15 0 saab 3 18 19 0 van 2 5 5 34
RNA – Hold-out	size=3 decay=0.1	0.479	Reference Prediction bus opel saab van bus 42 41 42 1 opel 0 0 0 0 saab 0 0 0 0 van 1 1 1 38

Após cálculos

Melhor técnica: SVM – Hold-out

3 Casos Desconhecidos:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis	ScVarmaxis
1	95	45	83	178	72	10	162	42	20	100	176	379
2	100	41	84	141	57	9	149	45	19	143	170	330
3	104	50	100	209	66	10	300	32	23	158	223	635

RaGyr	SkewMaxis	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa	tipo
184	70	6	16	187	200	?
158	72	9	14	189	199	?
220	73	14	9	188	196	?

Resultados:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis	ScVarmaxis
1	95	45	83	178	72	10	162	42	20	100	176	379
2	100	41	84	141	57	9	149	45	19	143	170	330
3	104	50	100	209	66	10	300	32	23	158	223	635

RaGyr	SkewMaxis	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa	predict.svm
184	70	6	16	187	200	opel
158	72	9	14	189	199	van
220	73	14	9	188	196	opel

Diabetes

Técnica	Parâmetro	Acurácia	Matriz de Confusão
RF – CV	mtry=2	0.7451	Reference Prediction neg pos neg 82 21 pos 18 32
SVM – Hold-out	C=0.25 Sigma=0.13	0.739	Confusion Matrix and Statistics Reference Prediction neg pos neg 87 27 pos 13 26
RF – Hold-out	mtry=2	0.7386	Reference Prediction neg pos neg 83 23 pos 17 30
SVM – Hold-out	C=0.25 Sigma=0.13	0.732	Confusion Matrix and Statistics Reference Prediction neg pos neg 87 27 pos 13 26
SVM – CV	C=0.25 Sigma=0.13	0.732	Confusion Matrix and Statistics Reference Prediction neg pos neg 84 25 pos 16 28
RNA – Hold-out	size=1 decay=0.1	0.7255	Reference Prediction neg pos neg 81 23 pos 19 30
KNN	k=9	0.721	Confusion Matrix and Statistics Reference Prediction neg pos neg 89 27 pos 16 22
RNA – CV	size=3 decay=0.4	0.6863	Reference Prediction neg pos neg 78 26 pos 22 27

Melhor técnica: RF – CV

3 Casos Desconhecidos:

	num	preg0nt	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes
1	1	10	200	74	0	0	38.0	0.937	34	?
2	2	10	200	80	0	0	27.1	1.441	57	?
3	3	1	60	60	23	300	30.1	0.398	59	?

Resultados:

	num	preg0nt	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes	predict.rf_cv
1	1	10	200	74	0	0	38.0	0.937	34	?	pos
2	2	10	200	80	0	0	27.1	1.441	57	?	neg
3	3	1	60	60	23	300	30.1	0.398	59	?	neg

Regressão

Admissão

Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
SVM – Hold-out	C=1 Sigma=0.11	0.8492141	0.05683194	0.9285196	0.05654124	0.03801399
SVM – CV	C=1 Sigma=0.11	0.8492141	0.05683194	0.9285196	0.05654124	0.03801399
RF – Hold-out	mtry=2	0.8206227	0.06230834	0.9065653	0.06166926	0.04253831

RF – CV	mtry=2	0.8153338	0.06322024	0.9035653	0.06257181	0.04345472
RNA – Hold-out	size=3 decay=0.1	0.8139129	0.06346299	0.9068822	0.06281207	0.04790741
RNA – CV	size=5 decay=0.1	0.8107109	0.06400666	0.9025809	0.06335017	0.04759592
KNN	K=7	0.089	0.09	0.8	0.62	0.068

Após os cálculos

Melhor técnica: SVM – Hold-out

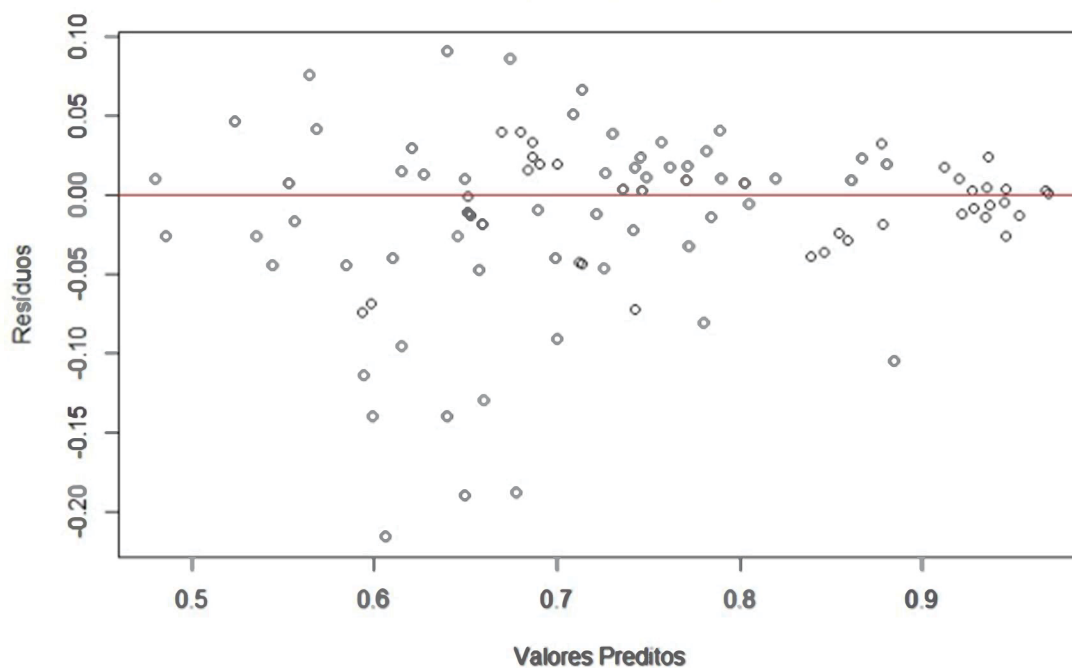
3 Casos Desconhecidos:

	num	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	ChanceOfAdmit
1	1	412	99	3	4	2	7.9	1	?
2	2	324	114	4	4	2	10.4	0	?
3	3	358	113	5	5	9	9.5	1	?

Resultados:

	num	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	predict.svm
1	1	412	99	3	4	2	7.9	1	0.6910434
2	2	324	114	4	4	2	10.4	0	0.7658169
3	3	358	113	5	5	9	9.5	1	0.7104633

Gráfico de Resíduos



Biomassa

Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
RF – Hold-out	mtry=2	0.8709691	532.4229	0.9797789	523.474	157.3707
RF – CV	mtry=2	0.8679476	538.6207	0.9805176	529.5676	165.2688
KNN	K=1	0.67	850	0.91	836	232
SVM – Hold-out	C=1 Sigma=0.91	0.34	1205	0.73	1184	362
SVM – CV	C=1 Sigma=0.91	0.34	1205	0.73	1184	362
RNA – Hold-out	size=3 decay=0.1	0.1884613	1335.256	0.8232358	1312.813	570.528
RNA – CV	size=3 decay=0.4	-0.09940084	1554.132	0.8481104	1528.01	502.1445

Após os cálculos

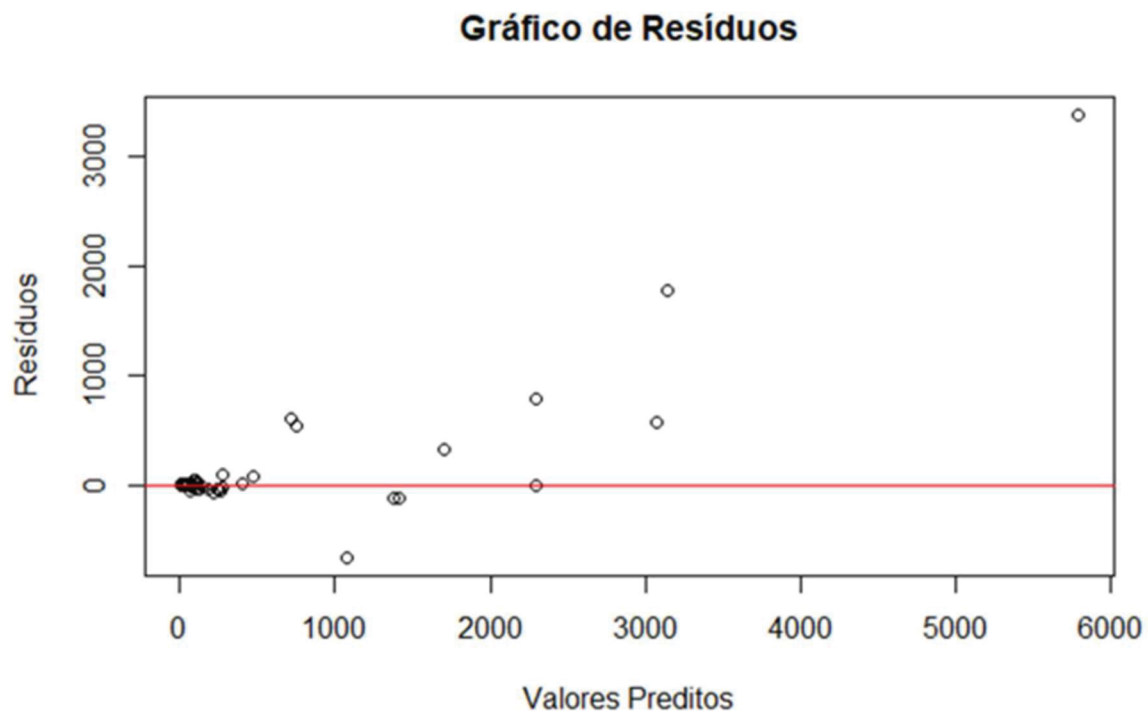
Melhor técnica: RF – Hold-out

3 Casos Desconhecidos:

	dap	h	Me	biomassa
1	11.3	10.7	0.9	?
2	10.3	13.0	0.9	?
3	17.5	11.0	0.5	?

Resultados:

	dap	h	Me	biomassa	predict.rf
1	11.3	10.7	0.9	?	77.56292
2	10.3	13.0	0.9	?	52.75713
3	17.5	11.0	0.5	?	89.92782



Agrupamento

Lista completa dos comandos emitidos no Rstudio:

```

1  ### Pacotes necessários
2  #install.packages("mlbench")
3  library(mlbench)
4
5  #install.packages("mice")
6  library(mice)
7
8  ## para o knodes
9  #install.packages("klaR")
10 library(klaR)
11
12 ### Leitura dos dados
13 setwd("C:\\faa\\Trabalho")
14 dados <- read.csv("4 - Veiculos - Dados.csv")
15 view(dados)
16
17 # Remoção da coluna desnecessária (id)
18 dados$a <- NULL
19
20 ## Executa o cluster - usar 10 clusters
21 set.seed(202470)
22 cluster.results <- knodes(dados, 10, weighted = FALSE )
23 cluster.results
24
25 ### Resultado do agrupamento
26 resultado <- cbind(dados, cluster.results$cluster)
27
28 ### Exibição do resultado
29 head(resultado, 10)

```

Após a leitura dos dados e remoção da coluna de ID, é lançada a seed e depois o comando com 10 clusters é executado:

```
## Executa o cluster - Usar 10 clusters
set.seed(202470)
cluster.results <- kmodes(dados, 10, weighted = FALSE)
cluster.results
```

O resultado é este abaixo, onde consta a Lista de Clusters gerados:

K-modes clustering with 10 clusters of sizes 63, 109, 84, 79, 75, 57, 93, 97, 76, 113

```
Cluster modes:
  a Comp CIRC DCIRC RadRa PRaXisRa MaxLRa ScatRa Elong PRaXisRect MaxLRect SCVarMaxis SCVarMaxis RaGyr SkewMaxis SkewMaxis KurtMaxis KurtMaxis MolRea Class
1 8 89 36 69 127 64 8 140 48 18 143 161 281 123 66 7 6 193 201 opel
2 1 101 53 108 203 61 10 217 31 24 162 226 712 214 70 0 24 189 197 saab
3 7 86 42 71 141 54 7 153 44 19 144 175 341 172 69 3 7 179 182 bus
4 9 85 37 86 121 55 6 122 57 17 128 237 209 127 64 4 14 180 181 van
5 2 90 46 85 169 56 11 157 43 20 160 170 373 185 74 1 11 183 192 van
6 3 89 45 82 136 61 10 155 43 19 148 173 351 174 71 1 9 188 196 van
7 15 104 55 105 197 61 11 218 31 24 173 221 706 216 71 0 4 187 196 opel
8 16 100 52 100 197 62 10 201 33 23 161 214 602 178 67 0 11 192 198 opel
9 27 91 38 66 130 59 7 128 52 18 127 148 246 139 72 1 20 183 187 saab
10 3 85 43 68 120 56 7 150 45 19 145 169 330 171 85 4 1 181 183 bus

Clustering vector:
[1] 2 5 6 6 10 10 3 1 4 4 1 4 5 6 7 8 4 10 7 2 3 4 10 8 7 6 9 7 3 9 6 1 4 2 7 5 3 3 2 3 8 4 5 2 2 5 4 3 5 6 9 2 7 3 4 1 2 5 7
[60] 9 8 9 9 10 10 7 10 7 5 10 2 8 8 4 1 2 9 10 7 10 3 2 1 10 13 10 10 4 1 4 7 8 8 3 1 2 1 10 8 3 10 1 9 9 6 2 7 10 10 10 6 9 4 6 5 8 2
[119] 10 2 9 5 6 4 6 5 10 10 6 8 2 10 9 4 7 5 5 1 9 3 5 3 8 10 8 7 5 10 3 1 2 10 10 2 6 10 4 5 6 1 9 8 2 9 2 7 3 2 10 5 7 10 6 7 5 5 7
[178] 10 3 4 2 1 3 9 8 2 3 6 2 1 8 2 2 9 5 10 2 3 3 6 5 1 7 7 4 6 8 3 5 7 8 2 3 7 1 5 9 7 10 4 7 5 8 3 9 6 7 1 2 10 4 4 10 8 1 5
[237] 4 10 8 3 1 1 3 2 5 1 10 8 7 4 9 7 5 10 6 7 1 3 2 7 9 2 4 4 8 10 3 5 4 4 10 9 10 6 10 10 10 4 2 10 8 3 10 2 5 4 3 6 8 4 9 5 3 1 4
[296] 9 8 9 9 3 2 10 4 1 10 10 2 2 8 8 2 6 10 8 10 9 9 6 4 3 2 10 10 2 9 1 8 6 10 9 6 8 7 3 5 7 5 8 6 1 10 1 4 8 7 2 2 3 6 1 1 10 5 3
[355] 2 3 2 6 8 2 8 5 3 3 8 6 2 10 7 8 3 6 4 4 8 2 9 9 7 1 8 3 2 1 5 2 3 6 8 10 8 3 6 2 5 10 3 3 1 2 3 2 1 8 3 9 9 3 9 1 4 10
[414] 1 5 2 1 1 10 9 9 7 9 10 1 1 2 9 7 1 2 8 9 10 2 4 10 8 9 8 5 4 1 2 5 8 2 2 6 10 5 8 10 8 10 4 6 7 1 9 8 8 4 10 10 10 7 2 9 9 7
[473] 3 7 3 7 7 5 10 3 7 7 8 9 5 3 7 2 3 2 2 5 10 7 3 3 7 6 4 2 8 7 9 9 2 2 2 10 5 7 10 8 7 5 9 9 3 2 9 1 2 4 1 10 7 9 9 8 5 3 8
[532] 3 7 4 5 6 7 4 8 4 1 9 9 2 3 5 4 1 3 2 4 1 2 3 8 8 10 9 10 2 7 5 7 3 6 2 2 9 3 9 2 8 6 5 7 2 2 8 5 9 10 4 2 7 8 4 7 9 7 7
[591] 10 1 3 6 9 7 7 4 10 9 5 4 8 1 8 2 4 4 1 10 7 5 4 4 4 1 1 4 9 7 2 6 8 2 1 2 1 3 8 9 2 7 10 8 9 1 9 10 10 7 8 10 2 10 6 7 4 9 7
[650] 6 7 6 8 8 2 4 1 3 1 2 9 8 10 7 4 5 8 2 5 10 8 7 8 10 6 8 10 8 10 5 5 10 5 10 7 3 1 4 4 7 2 10 8 5 8 1 6 4 3 5 7 8 1 1 8 7 1 4
[709] 5 8 4 8 8 2 3 8 9 5 8 7 7 2 5 7 6 4 8 10 1 6 7 5 6 4 4 7 2 2 2 6 2 10 1 2 6 3 9 10 5 2 2 4 9 10 1 4 2 10 10 6 8 2 7 7 2 10 9
[768] 3 8 1 10 8 8 10 4 5 2 10 7 9 8 8 8 7 4 10 7 4 2 2 10 1 8 2 10 4 4 7 7 10 6 9 1 3 8 10 6 7 3 4 7 1 10 5 10 10 5 2 2 2 9 7 4 8 7 5
[827] 8 8 3 5 10 3 7 5 9 10 6 6 5 6 1 1 9 5 7 1 4

Within-cluster simple-matching distance by cluster:
[1] 1082 1785 1400 1360 1204 934 1486 1603 1295 1873

Available components:
[1] "cluster" "size" "modes" "withindiff" "iterations" "weighted"
```

Exibição gerada pelo comando acima, com o cluster de cada linha na última coluna a direita:

```
> ## Resultado do agrupamento
> resultado <- cbind(dados, cluster.results$cluster)
>
> ## Exibição do resultado
> head(resultado, 10)
  a Comp CIRC DCIRC RadRa PRaXisRa MaxLRa ScatRa Elong PRaXisRect MaxLRect SCVarMaxis SCVarMaxis RaGyr SkewMaxis SkewMaxis KurtMaxis KurtMaxis MolRea Class cluster.results$cluster
1 1 99 48 83 178 72 10 182 47 20 159 176 379 184 70 8 14 187 187 van 2
2 2 92 43 84 141 57 9 149 45 18 143 170 330 158 72 9 14 189 199 van 5
3 3 104 50 106 209 66 10 207 32 23 158 223 635 220 73 14 9 188 196 saab 6
4 4 93 41 82 139 65 9 144 48 19 143 160 389 127 65 6 10 199 207 van 6
5 5 85 44 70 205 101 12 149 45 19 144 141 325 188 127 9 11 180 181 bus 10
6 6 107 57 106 172 50 6 255 28 18 168 280 957 264 85 5 8 181 182 bus 10
7 7 97 43 73 173 65 6 153 42 19 143 176 381 172 66 13 1 200 204 bus 3
8 8 90 45 88 137 65 9 157 48 18 148 162 281 164 67 3 3 195 202 van 1
9 9 88 34 82 149 61 7 122 54 17 127 141 274 132 66 2 14 200 208 van 4
10 10 92 44 98 187 62 11 183 36 22 146 202 505 152 68 4 14 195 204 saab 4
```

Regras de associação

Musculação

Lista completa dos comandos emitidos no RStudio

```
1 ## Instalação dos pacotes necessários
2 #install.packages('arules', dep=T)
3 library(arules)
4 library(datasets)
5
6 ## Leitura dos dados
7 setwd("C:\\faa\\trabalho")
8 dados <- read.transactions(file="2 - Musculacao - dados.csv", format="basket", sep=";")
9
10 # Exibição dos 5 primeiros itens
11 inspect(dados[1:5])
12
13 # Gráfico dos 10 primeiros itens mais frequentes
14 itemFrequencyPlot(dados, topN=10, type='absolute')
15
16 # Visão geral dos dados
17 summary(dados)
18
19 ## Extração das regras
20 # TESTES EFETUADOS COM SUPORTE ENTRE 0,1 A 0,5 E CONFIANÇA ENTRE 0,5 E 0,9
21 # A ESCOLHA FINAL FOI SUPORTE = 0,3 E CONFIANÇA = 0,7
22 # FORAM GERADAS 18 REGRAS
23
24 set.seed(202470)
25 rules <- apriori(dados, parameter = list(supp=0.3, conf = 0.7, target = "rules")) # > 18 regras
26
27 # Visão geral das regras
28 summary(rules)
29
30 # EXIBIÇÃO DAS REGRAS POR ORDEN DE CONFIANÇA
31 inspect(sort(rules, by="confidence"))
```

As saídas dos principais comandos serão mostradas abaixo.

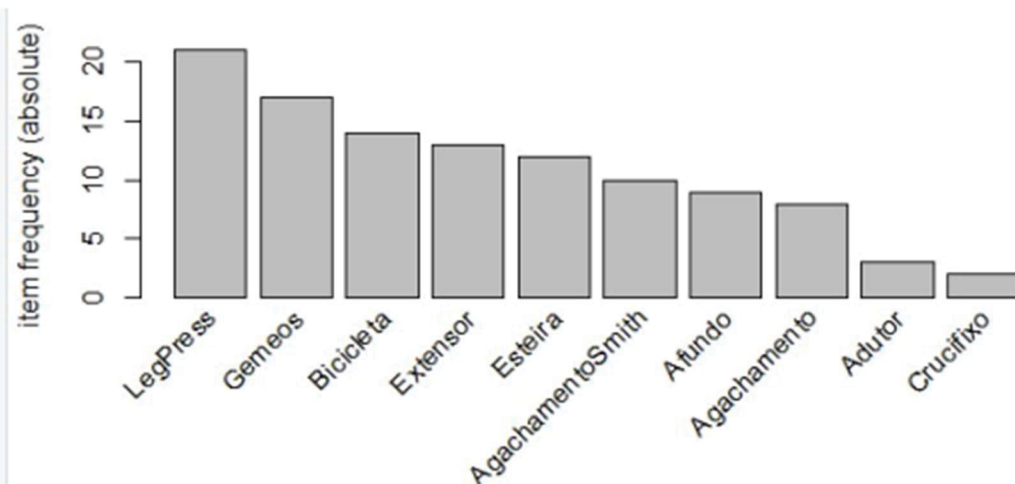
Exibição dos primeiros itens:

```
> # Exibição dos 5 primeiros itens
> inspect(dados[1:5])
  items
[1] {Afundo, crucifixo, Geneos, LegPress}
[2] {Agachamento, Geneos, LegPress}
[3] {Afundo, Agachamento, Geneos, LegPress}
[4] {Adutor, Agachamento, LegPress}
[5] {Afundo, Bicicleta, Geneos, LegPress}
```

O comando abaixo gera o gráfico apresentando na sequência do documento:

```
# Gráficos dos 10 primeiros itens mais frequentes
itemFrequencyPlot(dados, topN=10, type='absolute')
```

Exibição do gráfico gerado com o comando anterior:



A execução do código com o apriori exhibe o resultado abaixo:

```
> set.seed(202470)
> rules <- apriori(dados, parameter = list(supp=0.3, conf = 0.7, target = "rules")) # > 18 regras
Apriori

Parameter specification:
confidence minval smax arem aval originalsupport maxtime support minlen
0.7 0.1 1 none FALSE TRUE 5 0.3 1
maxlen target ext
10 rules TRUE

Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 7

set item appearances ... [0 item(s)] done [0.00s].
set transactions ... [11 item(s), 26 transaction(s)] done [0.00s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [18 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
```

Comando para exibir uma visão geral das regras encontradas:

```
> summary(rules)
set of 18 rules

rule length distribution (lhs + rhs): sizes
1 2 3
1 12 5

    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
1.000  2.000  2.000  2.222  2.750  3.000

summary of quality measures:
      support      confidence      coverage      lift      count
Min.   :0.3077  Min.   :0.7059  Min.   :0.3077  Min.   :0.8739  Min.   : 8.00
1st Qu.:0.3173  1st Qu.:0.8141  1st Qu.:0.3846  1st Qu.:1.5340  1st Qu.: 8.25
Median :0.3846  Median :0.8730  Median :0.4423  Median :1.7013  Median :10.00
Mean   :0.3996  Mean   :0.8742  Mean   :0.4658  Mean   :1.6109  Mean   :10.39
3rd Qu.:0.4231  3rd Qu.:0.9215  3rd Qu.:0.5000  3rd Qu.:1.8042  3rd Qu.:11.00
Max.   :0.8077  Max.   :1.0000  Max.   :1.0000  Max.   :2.0000  Max.   :21.00

mining info:
data ntransactions support confidence
dados      26      0.3      0.7 apriori(data = dados, parameter = list(supp = 0.3, conf = 0.7, target = "rules"))
```

Exibição das regras encontradas por ordem de confiança:

```
> # EXIBIÇÃO DAS REGRAS POR ORDEM DE CONFIANÇA
> inspect(sort(rules, by="confidence"))
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{Agachamento}	=> {LegPress}	0.3076923	1.0000000	0.3076923	1.2380952	8
[2]	{Afundo}	=> {Gemeos}	0.3461538	1.0000000	0.3461538	1.5294118	9
[3]	{AgachamentoSmith, Bicicleta}	=> {Extensor}	0.3076923	1.0000000	0.3076923	2.0000000	8
[4]	{Bicicleta, Esteira}	=> {Extensor}	0.3846154	1.0000000	0.3846154	2.0000000	10
[5]	{Extensor}	=> {Bicicleta}	0.4615385	0.9230769	0.5000000	1.7142857	12
[6]	{Esteira}	=> {Extensor}	0.4230769	0.9166667	0.4615385	1.8333333	11
[7]	{Esteira, Extensor}	=> {Bicicleta}	0.3846154	0.9090909	0.4230769	1.6883117	10
[8]	{AgachamentoSmith}	=> {Extensor}	0.3461538	0.9000000	0.3846154	1.8000000	9
[9]	{AgachamentoSmith, Extensor}	=> {Bicicleta}	0.3076923	0.8888889	0.3461538	1.6507937	8
[10]	{Bicicleta}	=> {Extensor}	0.4615385	0.8571429	0.5384615	1.7142857	12
[11]	{Extensor}	=> {Esteira}	0.4230769	0.8461538	0.5000000	1.8333333	11
[12]	{Esteira}	=> {Bicicleta}	0.3846154	0.8333333	0.4615385	1.5476190	10
[13]	{Bicicleta, Extensor}	=> {Esteira}	0.3846154	0.8333333	0.4615385	1.8055556	10
[14]	{}	=> {LegPress}	0.8076923	0.8076923	1.0000000	1.0000000	21
[15]	{AgachamentoSmith}	=> {Esteira}	0.3076923	0.8000000	0.3846154	1.7333333	8
[16]	{AgachamentoSmith}	=> {Bicicleta}	0.3076923	0.8000000	0.3846154	1.4857143	8
[17]	{Bicicleta}	=> {Esteira}	0.3846154	0.7142857	0.5384615	1.5476190	10
[18]	{Gemeos}	=> {LegPress}	0.4615385	0.7058824	0.6538462	0.8739496	12

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

Questão 1:

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Input, Conv2D, Dense, Flatten, Dropout
from tensorflow.keras.models import Model
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix
```

```

# Carga da base
cifar10 = tf.keras.datasets.cifar10

# Já está separado em dados de treino e teste
# Não precisa separar
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# Imagens em pixels de 0 - 255
# / 255.0 transforma em 0 - 1
x_train, x_test = x_train / 255.0, x_test / 255.0
# O dado y é a classe a qual faz parte
# O flatten torna os dados vetorizados
y_train, y_test = y_train.flatten(), y_test.flatten()
# Dimensão dos dados
print("x_train.shape: ", x_train.shape)
print("y_train.shape: ", y_train.shape)
print("x_test.shape: ", x_test.shape)
print("y_test.shape: ", y_test.shape)

K = len(set(y_train))
# Aqui começa o 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)
# Todas as imagens são do mesmo tamanho, não precisa de Global Pooling
x = Flatten()(x)
# Aqui começa o 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 ( lista entrada, lista saída)
model = Model(i, x)

# Relatório sobre a arquitetura da rede
model.summary()

```

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_3 (Conv2D)	(None, 15, 15, 32)	896
conv2d_4 (Conv2D)	(None, 7, 7, 64)	18,496
conv2d_5 (Conv2D)	(None, 3, 3, 128)	73,856
flatten_1 (Flatten)	(None, 1152)	0
dropout_2 (Dropout)	(None, 1152)	0
dense_2 (Dense)	(None, 1024)	1,180,672
dropout_3 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 10)	10,250

Total params: 1,284,170 (4.90 MB)

Trainable params: 1,284,170 (4.90 MB)

Non-trainable params: 0 (0.00 B)

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

```

Epoch 1/15
1563/1563 ————— 17s 7ms/step - accuracy: 0.3379 - loss: 1.7806 - val_accuracy: 0.5092 - val_loss: 1.3535
Epoch 2/15
1563/1563 ————— 10s 3ms/step - accuracy: 0.5252 - loss: 1.3135 - val_accuracy: 0.5903 - val_loss: 1.1472
Epoch 3/15
1563/1563 ————— 10s 3ms/step - accuracy: 0.5749 - loss: 1.1797 - val_accuracy: 0.6316 - val_loss: 1.0380
Epoch 4/15
1563/1563 ————— 5s 3ms/step - accuracy: 0.6168 - loss: 1.0740 - val_accuracy: 0.6461 - val_loss: 1.0043
Epoch 5/15
1563/1563 ————— 4s 3ms/step - accuracy: 0.6427 - loss: 1.0002 - val_accuracy: 0.6687 - val_loss: 0.9473
Epoch 6/15
1563/1563 ————— 11s 6ms/step - accuracy: 0.6684 - loss: 0.9332 - val_accuracy: 0.6863 - val_loss: 0.9019
Epoch 7/15
1563/1563 ————— 5s 3ms/step - accuracy: 0.6848 - loss: 0.8805 - val_accuracy: 0.6948 - val_loss: 0.8800
Epoch 8/15
1563/1563 ————— 5s 3ms/step - accuracy: 0.7011 - loss: 0.8388 - val_accuracy: 0.6977 - val_loss: 0.8795
Epoch 9/15
1563/1563 ————— 10s 3ms/step - accuracy: 0.7127 - loss: 0.7942 - val_accuracy: 0.7012 - val_loss: 0.8672
Epoch 10/15
1563/1563 ————— 5s 3ms/step - accuracy: 0.7288 - loss: 0.7610 - val_accuracy: 0.7146 - val_loss: 0.8245
Epoch 11/15
1563/1563 ————— 4s 3ms/step - accuracy: 0.7445 - loss: 0.7199 - val_accuracy: 0.7109 - val_loss: 0.8251
Epoch 12/15
1563/1563 ————— 6s 3ms/step - accuracy: 0.7507 - loss: 0.7000 - val_accuracy: 0.7179 - val_loss: 0.8205
Epoch 13/15
1563/1563 ————— 10s 3ms/step - accuracy: 0.7522 - loss: 0.7009 - val_accuracy: 0.7152 - val_loss: 0.8279
Epoch 14/15
1563/1563 ————— 9s 3ms/step - accuracy: 0.7658 - loss: 0.6593 - val_accuracy: 0.7218 - val_loss: 0.8140
Epoch 15/15
1563/1563 ————— 5s 3ms/step - accuracy: 0.7742 - loss: 0.6372 - val_accuracy: 0.7205 - val_loss: 0.8157

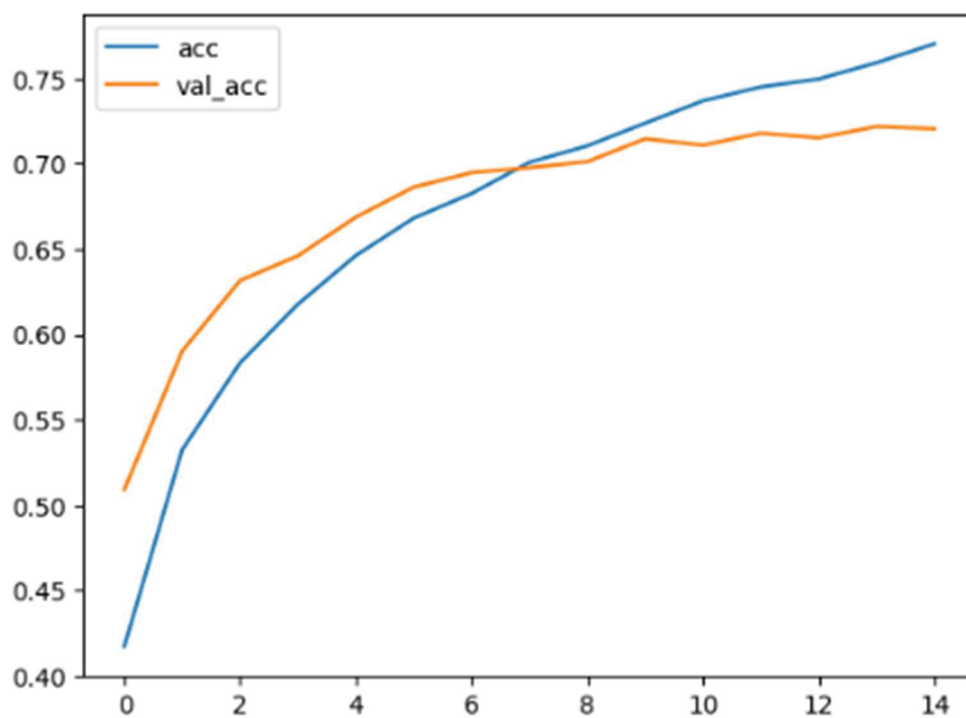
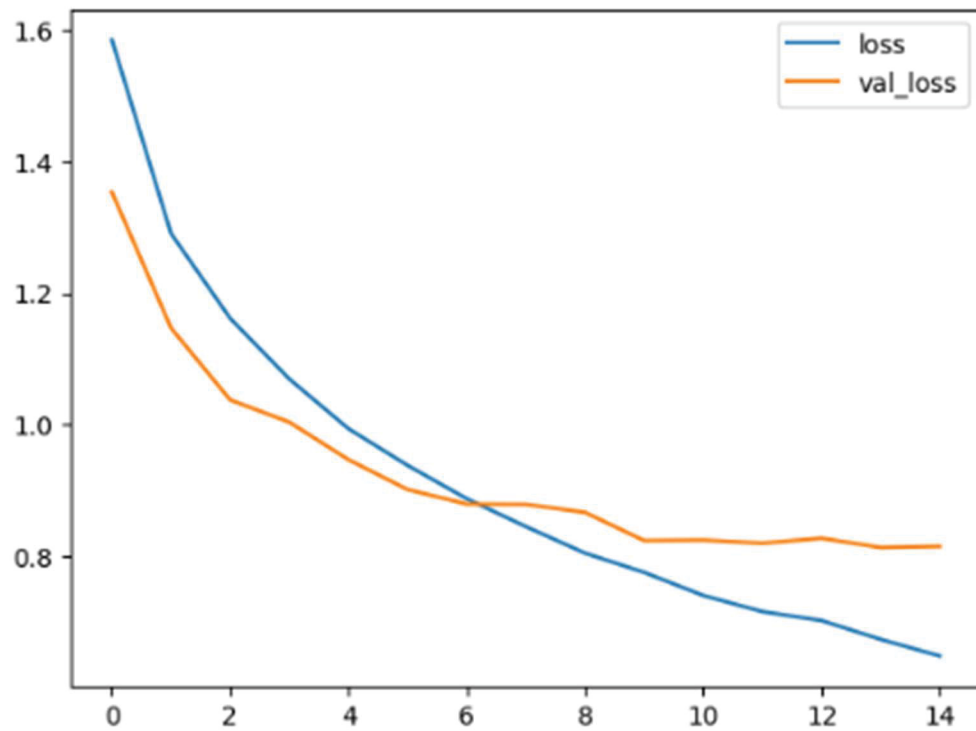
```

```

# 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()

# 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()

```

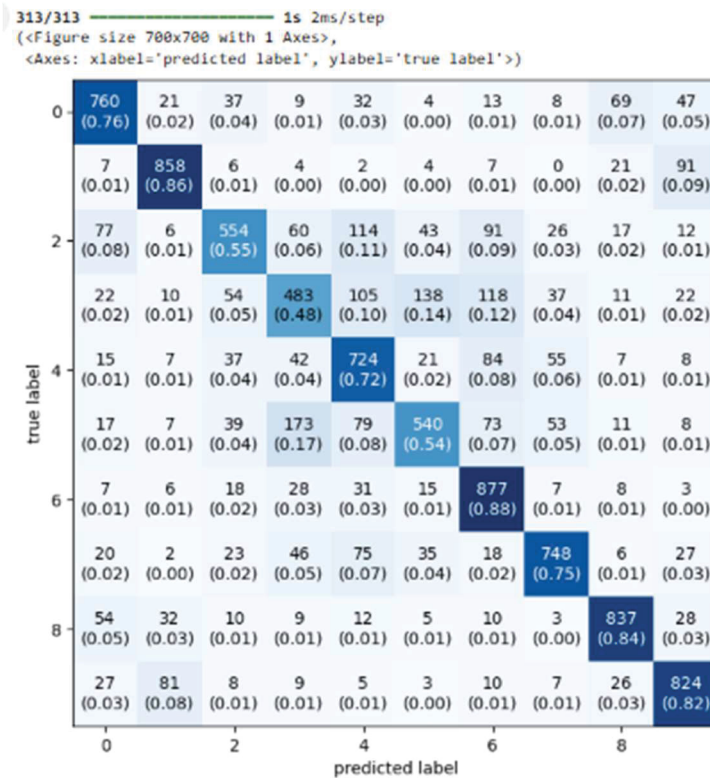


```
# Efetuar predições na base de teste
# argmax é usado pois a função de ativação da saída é softmax
# argmax pega o neurônio que deu o maior resultado, isto é,
# a maior probabilidade de saída
```

```

y_pred = model.predict(x_test).argmax(axis=1)
# Mostrar a matriz de confusão
cm = confusion_matrix(y_test, y_pred)
plot_confusion_matrix(conf_mat=cm, figsize=(7, 7),
show_normed=True)

```



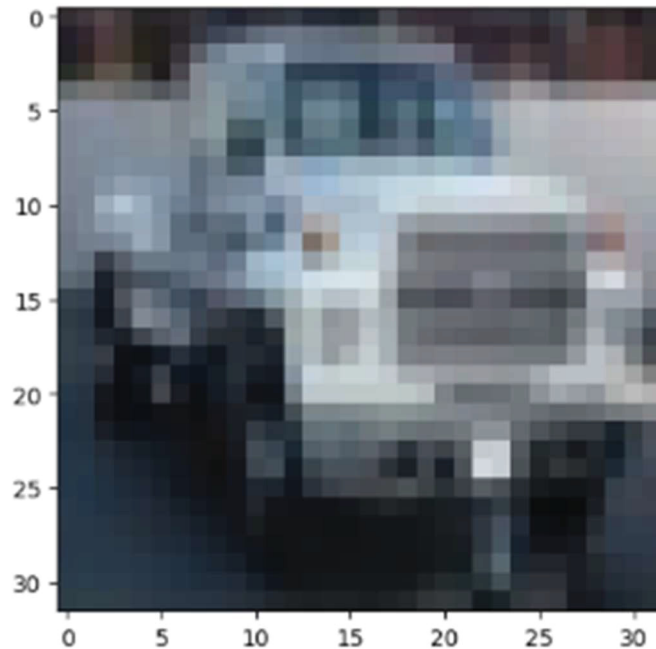
```

#Mostrar algumas classificações erradas
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]]))

```

```
Text(0.5, 1.0, 'True label: truck Predicted: automobile')
```

True label: truck Predicted: automobile



Questão 2:

```
# Importação das Bibliotecas
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Input, Embedding, LSTM, Dense
from tensorflow.keras.layers import GlobalMaxPooling1D
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer

# carrega e arruma a base
!wget http://www.razer.net.br/datasets/spam.csv
df = pd.read_csv("spam.csv", encoding="ISO-8859-1")
df.head()

df = df.drop(["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axis=1)
df.columns = ["labels", "data"]
```

```
df["b_labels"] = df["labels"].map({ "ham": 0, "spam": 1})
y = df["b_labels"].values
```

```
--2024-08-02 17:55:43-- http://www.razer.net.br/datasets/spam.csv
Resolving www.razer.net.br (www.razer.net.br)... 178.128.150.229
Connecting to www.razer.net.br (www.razer.net.br)|178.128.150.229|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 503663 (492K) [text/csv]
Saving to: 'spam.csv.4'
```

```
spam.csv.4          100%[=====>] 491.86K  --.-KB/s   in 0.04s
```

```
2024-08-02 17:55:43 (12.0 MB/s) - 'spam.csv.4' saved [503663/503663]
```

```
# Separa a base em treino e teste
x_train, x_test, y_train, y_test = train_test_split(df["data"],
y,test_size=0.33)
```

```
# 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)

# Acerta o tamanho das sequências (padding)
data_train = pad_sequences(sequences_train) # usa o tamanho da maior seq.
T = data_train.shape[1] # tamanho da sequência
data_test = pad_sequences(sequences_test, maxlen=T)
print("data_train.shape: ", data_train.shape)
print("data_test.shape: ", data_test.shape)

# 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,)) # Entra uma frase inteira
x = Embedding(V+1, D)(i)
x = LSTM(M)(x)
x = Dense(1, activation="sigmoid")(x) # Sigmoide pois só tem 2 valores
model = Model(i, x)

```

model.summary()

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 189)	0
embedding_1 (Embedding)	(None, 189, 20)	144,060
lstm_1 (LSTM)	(None, 5)	520
dense_1 (Dense)	(None, 1)	6

Total params: 144,586 (564.79 KB)
Trainable params: 144,586 (564.79 KB)
Non-trainable params: 0 (0.00 B)

```

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))

```

```

Epoch 1/5
117/117 — 11s 67ms/step - accuracy: 0.8475 - loss: 0.5297 - val_accuracy: 0.9233 - val_loss: 0.2555
Epoch 2/5
117/117 — 10s 69ms/step - accuracy: 0.9589 - loss: 0.2163 - val_accuracy: 0.9723 - val_loss: 0.1583
Epoch 3/5
117/117 — 7s 56ms/step - accuracy: 0.9854 - loss: 0.1252 - val_accuracy: 0.9804 - val_loss: 0.1132
Epoch 4/5
117/117 — 8s 71ms/step - accuracy: 0.9892 - loss: 0.0907 - val_accuracy: 0.9848 - val_loss: 0.0890
Epoch 5/5
117/117 — 10s 71ms/step - accuracy: 0.9917 - loss: 0.0622 - val_accuracy: 0.9842 - val_loss: 0.0714

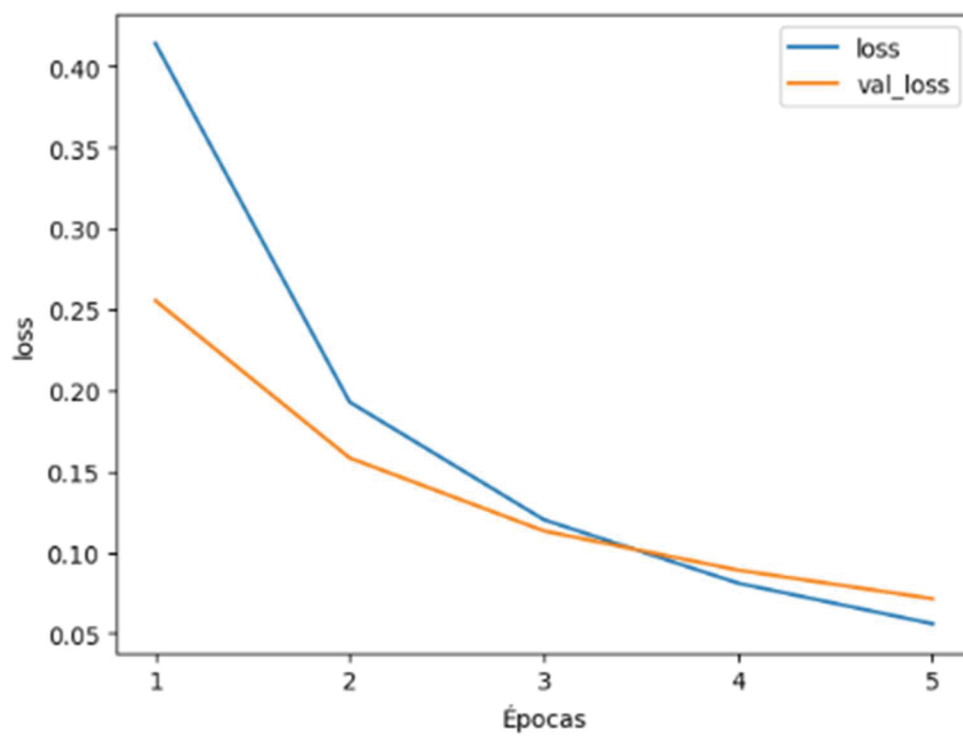
```

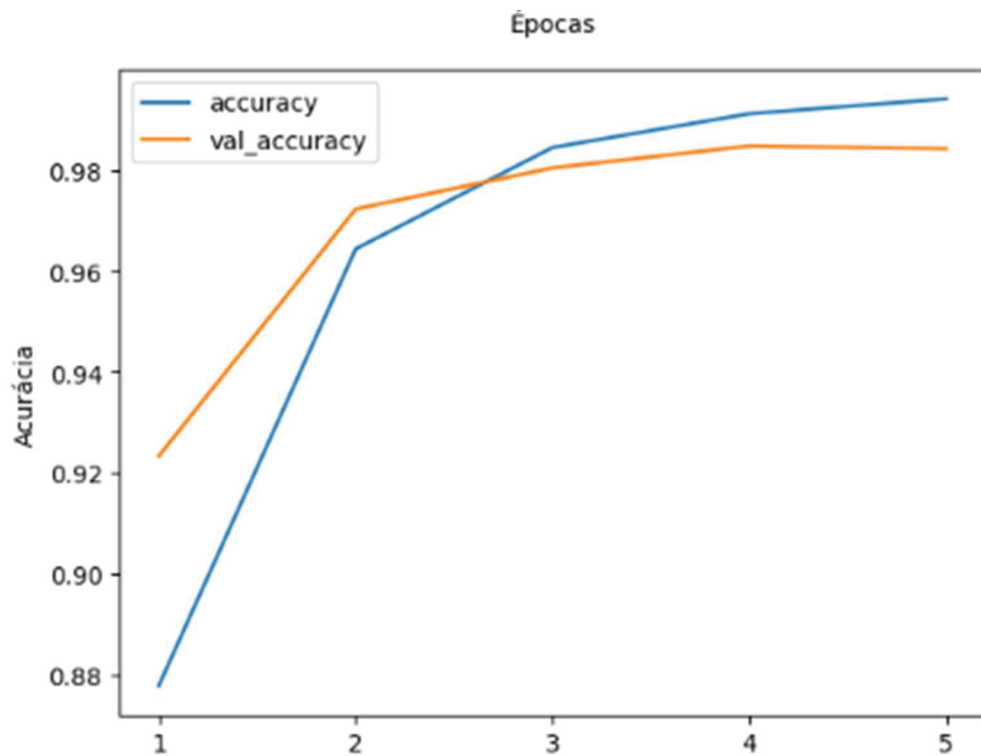
```

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()
```





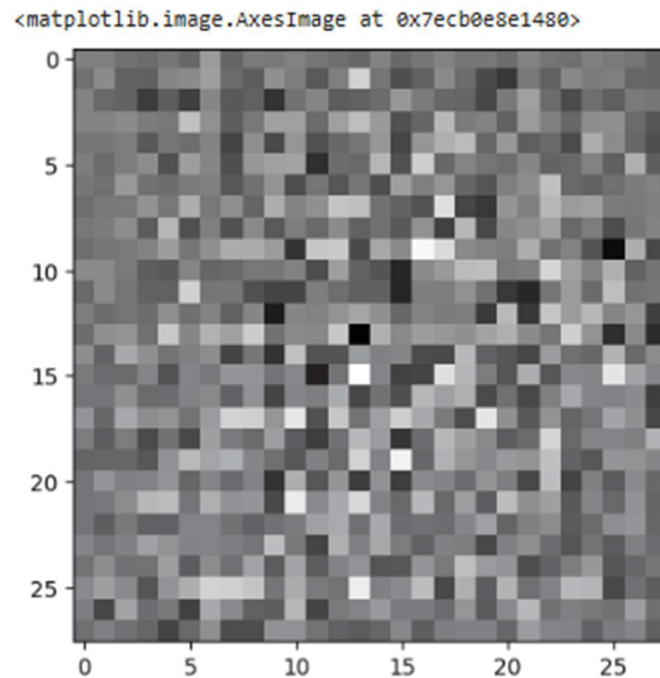
```
# Efetua a predição de um texto novo
texto = "Hi, my name is Razer and want to tell you something."
#texto = "Is your car dirty? Discover our new product. Free for all. Click
the link."
seq_texto = tokenizer.texts_to_sequences([texto]) # Tokeniza
data_texto = pad_sequences(seq_texto, maxlen=T) # Padding
pred = model.predict(data_texto) # Predição
print(pred)
print ("SPAM" if pred >= 0.5 else "OK")
```

```
1/1 ————— 0s 28ms/step
[[0.01949643]]
OK
```

Questão 3:

```
# 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')
```



```
# Treinar o modelo e restaurar o último ponto de verificação
train(train_dataset, EPOCHS)
checkpoint.restore(tf.train.latest_checkpoint(checkpoint_dir))
```



```
:tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x7eca5f6f3400>
```

```
# Criar um GIF
```

```
# Display a single image using the epoch number
def display_image(epoch_no):
    return PIL.Image.open('image_at_epoch_{:04d}.png'.format(epoch_no))

display_image(EPOCHS)

anim_file = 'dcgan.gif'
with imageio.get_writer(anim_file, mode='I') as writer:
    filenames = glob.glob('image*.png')
    filenames = sorted(filenames)
    for filename in filenames:
        image = imageio.imread(filename)
        writer.append_data(image)

import tensorflow_docs.vis.embed as embed
embed.embed_file(anim_file)
```

```
<ipython-input-85-a4048264b71f>:13: DeprecationWarning: Starting with ImageIO
image = imageio.imread(filename)
```



Questão 4:

```
# Carregar a base de dados
examples, metadata = tfds.load('ted_hrlr_translate/pt_to_en',
with_info=True, as_supervised=True)
```

```
train_examples, val_examples = examples['train'], examples['validation']
```

```
Downloading and preparing dataset 124.94 MiB (download: 124.94 MiB, generated: Unknown size, total: 124.94 MiB) to /root/tensorflow_datasets/ted_hrlr_translate/pt_to_en/1.0.0...
Dl Completed...: 100%  1/1 [00:10<00:00, 3.68s/ un]
Dl Size...: 100%  124/124 [00:10<00:00, 35.48 MiB/s]
Extraction completed...: 100%  112/112 [00:10<00:00, 10.28s/ file]
```

```
# 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))
```

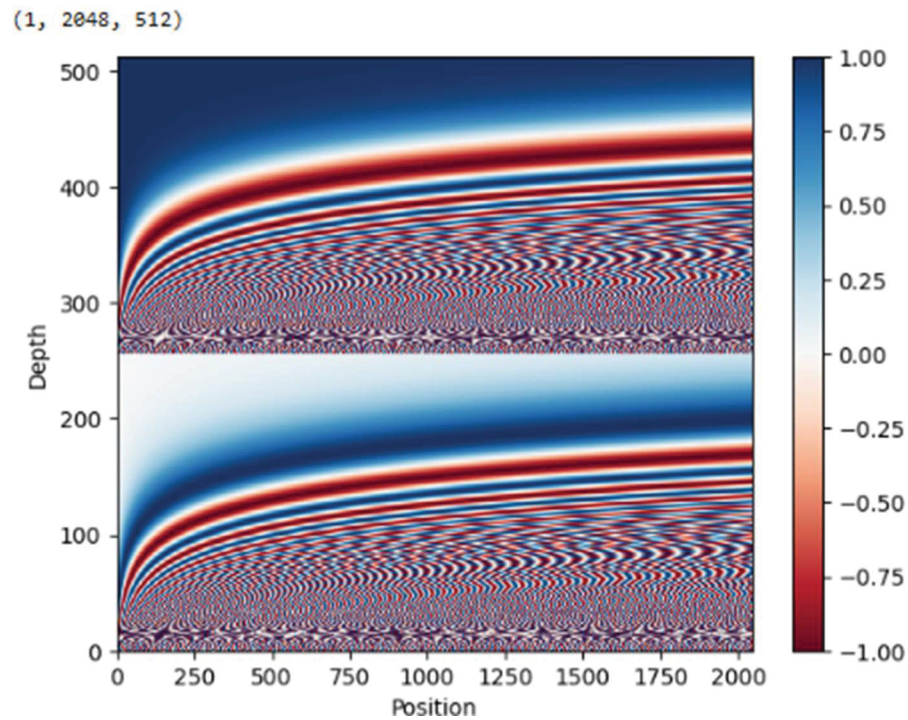
```
plt.pcolormesh(pos_encoding, cmap='RdBu')
```

```
plt.ylabel('Depth')
```

```
plt.xlabel('Position')
```

```
plt.colorbar()
```

```
plt.show()
```



```

for epoch in range(EPOCHS):
    start = time.time()
    train_loss.reset_state()
    train_accuracy.reset_state()

    # inp -> português, tar -> inglês
    for batch, (inp, tar) in enumerate(train_batches):
        train_step(inp, tar)

        if batch % 50 == 0:
            print(f'Epoch {epoch + 1} Batch {batch} Loss
{train_loss.result():.4f} Accuracy {train_accuracy.result():.4f}')

        if (epoch + 1) % 5 == 0:
            ckpt_save_path = ckpt_manager.save()
            print(f'Saving checkpoint for epoch {epoch + 1} at
{ckpt_save_path}')

    print(f'Epoch {epoch + 1} Loss {train_loss.result():.4f} Accuracy
{train_accuracy.result():.4f}')

```

```
print(f'Time taken for 1 epoch: {time.time() - start:.2f} secs\n')
```

```
Epoch 20 Batch 0 Loss 1.3898 Accuracy 0.6773
Epoch 20 Batch 50 Loss 1.4245 Accuracy 0.6815
Epoch 20 Batch 100 Loss 1.4248 Accuracy 0.6825
Epoch 20 Batch 150 Loss 1.4271 Accuracy 0.6831
Epoch 20 Batch 200 Loss 1.4320 Accuracy 0.6823
Epoch 20 Batch 250 Loss 1.4372 Accuracy 0.6813
Epoch 20 Batch 300 Loss 1.4426 Accuracy 0.6805
Epoch 20 Batch 350 Loss 1.4468 Accuracy 0.6799
Epoch 20 Batch 400 Loss 1.4487 Accuracy 0.6795
Epoch 20 Batch 450 Loss 1.4513 Accuracy 0.6789
Epoch 20 Batch 500 Loss 1.4506 Accuracy 0.6790
Epoch 20 Batch 550 Loss 1.4534 Accuracy 0.6786
Epoch 20 Batch 600 Loss 1.4563 Accuracy 0.6780
Epoch 20 Batch 650 Loss 1.4609 Accuracy 0.6773
Epoch 20 Batch 700 Loss 1.4653 Accuracy 0.6764
Epoch 20 Batch 750 Loss 1.4675 Accuracy 0.6762
Epoch 20 Batch 800 Loss 1.4698 Accuracy 0.6759
Saving checkpoint for epoch 20 at ./checkpoints/train/ckpt-4
Epoch 20 Loss 1.4702 Accuracy 0.6759
Time taken for 1 epoch: 97.68 secs
```

```
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 trivias in enclos 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

Contexto

Este estudo de caso baseia-se na rotina de um integrante da equipe que atua no departamento de TI de uma rede de supermercados, composta por 29 lojas no estado do Paraná. Diante da necessidade de entender melhor o comportamento dos clientes, foi proposto o desenvolvimento de um escopo de Big Data para realizar uma análise detalhada do perfil dos clientes cadastrados via aplicativo da rede. O objetivo principal é segmentar os clientes de acordo com seus hábitos de compra, utilizando dados extraídos de sistemas CRM (Customer Relationship Management) e ERP (Enterprise Resource Planning), como histórico de compras, cadastro de clientes e produtos. Com esses dados, será realizada uma análise comportamental para identificar padrões de consumo e auxiliar na personalização de campanhas de marketing.

Ferramentas e Tecnologias utilizadas

- **Sistemas de Log Management**

IBM Qradar para correlação de eventos de segurança em tempo real.

- **Plataforma de Big Data**

Apache Hadoop núcleo do processamento de dados devido à sua capacidade de lidar com grandes volumes de dados distribuídos. Utilizando o MapReduce, os dados serão processados de forma distribuída, permitindo análises rápidas e eficientes, como a classificação de perfis de clientes.

- **NoSql**

Cassandra e HBase, serão empregadas para armazenar os dados não estruturados e semiestruturados, como o histórico de compras e as preferências dos clientes. Estes sistemas são ideais para lidar com grandes volumes de dados com velocidade e flexibilidade

- **Machine Learning**

Para a análise comportamental dos clientes, serão aplicadas técnicas de Machine Learning, utilizando bibliotecas como Mahout e Spark MLlib. Permitindo a criação de algoritmos de clustering, que serão aplicados para segmentar os clientes em três grupos distintos: clientes que gastam muito, clientes com gastos moderados e clientes que gastam pouco. Modelos preditivos também serão desenvolvidos para recomendar produtos com base no perfil de cada cliente, auxiliando nas campanhas de marketing personalizadas.

Organização dos dados

- **Modelagem dos Dados**

Conforme os tipos de dados existentes no CRM e ERP, diferentes abordagens de modelagem serão utilizadas, tais como:

- **Modelo Colunar:** O HBase, como banco de dados NoSQL colunar, será utilizado para armazenar dados do histórico de compras e cadastros de clientes, proporcionando um modelo eficiente para leituras rápidas.
- **Modelo Chave-Valor:** Sistemas como o Redis podem ser usados para armazenar preferências instantâneas dos clientes, como produtos visualizados recentemente.
- **Spark;** será empregado para processar grandes volumes de dados, rodar algoritmos de clustering (K-means) para segmentar os clientes, e aplicar modelos de aprendizado de máquina para prever comportamentos de consumo e recomendar produtos. Essa arquitetura permite um fluxo contínuo de dados entre diferentes componentes.

Captura de dados em tempo real do PDV (ponto de venda)

1. Será utilizado Kafka e Storm. Pois o Kafka terá o papel de gestão em tempo real dos dados de CRM/ERP, consolidando o fluxo de transações e comportamento dos clientes em tempo real. A tecnologia Storm pode ser usada para processar esses dados conforme chegam, com uma arquitetura distribuída que garante alta disponibilidade, semelhante ao que é feito no pipeline de análise mostrado no exemplo da figura 1.

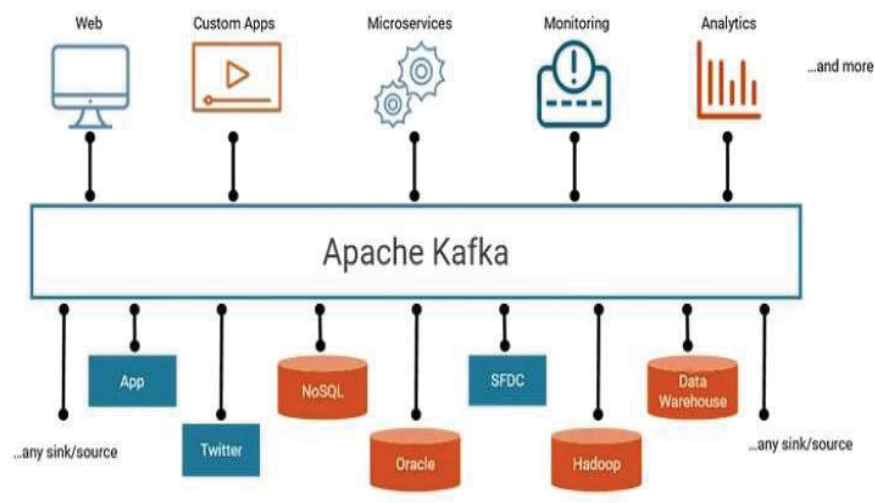


Figura 1: Exemplo de um fluxo de armazenamento e processamento de dados usando Kafka. Fonte : <https://medium.com/trainingcenter/apache-kafka-838882261e83>.

Visualização dos dados

Grafana para visualização em tempo real das requisições realizadas e informações relacionadas a performances do produto no geral. Além do Power BI para criação de dashboard personalizados de acordo com a necessidade da equipe de marketing, e-commerce e demais que compoem o setor administrativo da rede de supermercado.

Arquitetura Resumida

1. **Ingestão de Dados:** Kafka para coleta de dados em tempo real dos sistemas CRM/ERP.
2. **Processamento de Dados:** Storm processando streams de dados, seguido de persistência no **Hadoop HDFS** e bancos **NoSQL** (HBase ou Cassandra).
3. **Análise:** Spark para análise distribuída e segmentação de clientes com aprendizado de máquina.
4. **Visualização:** Dashboards interativos com Grafana e Power BI para análise do comportamento dos clientes.
5. **Recomendação de Produtos:** Sistemas baseados em aprendizado de máquina para recomendar produtos em tempo real de acordo com o comportamento.

Especificação dos Vs

Volume: É esperado um volume considerável de dados que serão processados, provenientes do histórico de compras dos clientes das 29 lojas, cadastros no app.

Variedade: O conjunto de dados é altamente heterogêneo, incluindo dados estruturados (como cadastros de clientes e produtos) e não estruturados (como logs de navegação(app e e-commerce) e preferências de produtos). A utilização de sistemas NoSQL é crucial para lidar com essa variedade.

Velocidade: A análise em tempo real é necessária para recomendações instantâneas de produtos e personalização de campanhas de marketing. Para isso, será utilizado processamento de dados em movimento com ferramentas como Apache Storm e Kafka (possui ótima performance em envio de mensageria que serão consumidos via tópicos), que fornecem capacidades de streaming.

Veracidade: Garantir a integridade e a qualidade dos dados é fundamental para todo o processo executado. Sistemas distribuídos, como o Hadoop, têm mecanismos para garantir a consistência dos dados, mesmo em grandes volumes e alta diversidade,.Desde modo enfatiza ainda mais a escolha do Hadoop

Valor: A análise comportamental trará valor ao negócio ao identificar os padrões de compra, permitindo a segmentação precisa dos clientes e gerando insights valiosos para campanhas de marketing personalizadas e auxílio na compras de produtos no geral

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:

- Carregue a base de dados de **Treino**.
- Crie partições contendo 80% para treino e 20% para validação (atenção aos pacientes).
- Extraia características utilizando LBP e a CNN VGG16 (gerando um csv para cada extrator).
- Treine modelos Random Forest, SVM e RNA para predição dos dados extraídos.
- Carregue a base de **Teste** e execute a tarefa 3 nesta base.
- Aplice os modelos treinados nos dados de treino
- Calcule as métricas de Sensibilidade, Especificidade e F1-Score com base em suas matrizes de confusão.
- 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:

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

B – RESOLUÇÃO

Questão 1:

Resultados obtidos com a Base de dados: Test_4cl_amostra

Desempenho dos Modelos com Características LBP					
Classe	Modelo	Acurácia	Sensibilidade	Especificidade	F1-Score
0	Random Forest	0.66	0.50	0.80	0.56
1	Random Forest	0.66	0.63	0.81	0.56
2	Random Forest	0.66	0.56	0.87	0.55
3	Random Forest	0.66	0.97	0.98	0.97
0	SVM	0.51	0.02	0.75	0.04
1	SVM	0.51	0.72	0.98	0.51
2	SVM	0.51	0.38	0.84	0.37
3	SVM	0.51	0.98	0.98	0.90
0	RNA	0.50	0.00	0.65	0.00
1	RNA	0.50	0.77	0.68	0.51
2	RNA	0.50	0.31	0.83	0.34
3	RNA	0.50	1.00	0.97	0.87

Desempenho dos Modelos com Características VGG16					
Classe	Modelo	Acurácia	Sensibilidade	Especificidade	F1-Score
0	SVM	0.91	0.97	0.92	0.93
1	SVM	0.91	0.91	0.88	0.90
2	SVM	0.91	0.84	0.83	0.90
3	SVM	0.91	0.92	0.97	0.93

0	Random Forest	0.87	0.95	0.85	0.91
1	Random Forest	0.87	0.83	0.88	0.88
2	Random Forest	0.87	0.81	0.85	0.82
3	Random Forest	0.87	0.86	0.93	0.84
0	RNA	0.89	0.94	0.93	0.89
1	RNA	0.89	0.79	0.88	0.82
2	RNA	0.89	0.90	0.96	0.91
3	RNA	0.89	0.93	0.96	0.95

Melhor Modelo por Característica

- LBP: O Random Forest apresentou uma Acurácia de 66% com um F1-Score elevado na classe 3 (0.97).
- VGG16: O SVM teve a melhor Acurácia de 91% com um F1-Score alto em todas as classes.

Conclusão

O modelo SVM com características VGG16 teve o melhor desempenho geral, superando o Random Forest e o RNA em termos de Acurácia e F1-Score, evidenciando sua eficácia na classificação.

Questão 2:

Resultados obtidos com a Base de dados: Test_4cl_amostra

Desempenho dos Modelos RESNET50 e VGG16					
Classe	Modelo	Acurácia	Sensibilidade	Especificidade	F1-Score
0	Resnet50 COM Data Augmentation	24.26%	0.059406	0.985185	0.112055
1	Resnet50 COM Data Augmentation	24.26%	0.022222	0.971530	0.043451
2	Resnet50 COM Data Augmentation	24.26%	0.155556	0.697509	0.254380
3	Resnet50 COM Data Augmentation	24.26%	0.944444	0.405694	0.567579
0	Resnet50 SEM Data Augmentation	81.13%	0.930693	1.000000	0.964103

1	Resnet50 SEM Data Augmentation	81.13%	0.700000	0.889680	0.783524
2	Resnet50 SEM Data Augmentation	81.13%	0.688889	0.882562	0.773791
3	Resnet50 SEM Data Augmentation	81.13%	0.933333	0.985765	0.958833
0	VGG16 COM Data Augmentation	44.47%	0.029703	1.000000	0.057692
1	VGG16 COM Data Augmentation	44.47%	0.000000	1.000000	0.000000
2	VGG16 COM Data Augmentation	44.47%	0.011111	1.000000	0.021978
3	VGG16 COM Data Augmentation	44.47%	1.000000	0.014235	0.028070
0	VGG16 SEM Data Augmentation	63.07%	0.811881	0.951852	0.876312
1	VGG16 SEM Data Augmentation	63.07%	0.466667	0.882562	0.610515
2	VGG16 SEM Data Augmentation	63.07%	0.466667	0.797153	0.588699
3	VGG16 SEM Data Augmentation	63.07%	0.833333	0.903915	0.867190

Conclusão

Com base nas métricas encontradas em todos os modelos treinados, o modelo ResNet50 SEM Data Augmentation é a melhor escolha. Ele combina uma alta acurácia com boas métricas de sensibilidade e F1-Score, sugerindo que ele generaliza bem e é eficaz em classificar corretamente as diferentes classes.

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: <https://bibliotecas.ufpr.br/wp-content/uploads/2022/03/template-artigo-de-periodico.docx>

B – RESOLUÇÃO

INTRODUÇÃO

A inteligência artificial (IA) tem mudado muitos aspectos da vida moderna, desde a automação de processos industriais até a personalização de serviços ao consumidor e questões básicas do dia a dia. Uma das inovações mais notáveis é o desenvolvimento de modelos de linguagem avançados, como o ChatGPT da OpenAI. Esses modelos são capazes de gerar textos coerentes e relevantes, interagindo com os usuários de maneira cada vez mais sofisticada. No entanto, o uso crescente dessas tecnologias levanta importantes questões éticas que precisam ser cuidadosamente examinadas e tratadas.

Este estudo de caso tem como objetivo investigar as implicações éticas do uso do ChatGPT em diferentes contextos, explorando como esses desafios afetam desenvolvedores, usuários e a sociedade em geral. Abordaremos questões cruciais, como a privacidade dos dados dos usuários, o potencial de disseminação de desinformação, os vieses algorítmicos e a responsabilidade dos desenvolvedores. Além disso, discutiremos possíveis soluções para mitigar esses dilemas éticos e promover um uso mais responsável e transparente da IA.

A importância deste estudo está na necessidade de equilibrar o avanço tecnológico com a manutenção de princípios éticos fundamentais. À medida que a IA continua a evoluir, é essencial que seu desenvolvimento e uso sejam guiados por normas éticas rigorosas que protejam os direitos e a privacidade das pessoas, promovam a equidade e garantam a transparência. Com este trabalho, esperamos contribuir para um debate mais informado e construtivo sobre as melhores práticas no uso da IA, com foco específico no ChatGPT.

DESENVOLVIMENTO

Relevância Ética

Os desafios éticos relacionados ao uso do ChatGPT são amplos e complexos. Alguns dos principais dilemas, percebemos que, incluem a privacidade dos usuários, a disseminação de desinformação e a discriminação algorítmica e questões relacionadas a dificuldade de avaliar as responsabilidades efeitos do seu uso e a necessidade de análise humana antes de usar as informações em outros contextos.

A privacidade dos dados é uma preocupação central, especialmente considerando que o ChatGPT processa grandes volumes de informações pessoais.

De acordo com a Lei Geral de Proteção de Dados (LGPD) no Brasil, a privacidade e a proteção dos dados pessoais são direitos fundamentais (Brasil, 2018).

A disseminação de desinformação é outro problema crítico, que vem tomando grandes proporções com os avanços da tecnologia, uma vez que o ChatGPT pode gerar informações imprecisas ou enganosas e que são expostas de uma forma persuasiva, porém muitas vezes sem fontes consideradas confiáveis ou que possam ser checadas de forma independente.

Além disso, há preocupações sobre a discriminação algorítmica, onde vieses presentes nos dados de treinamento podem resultar em respostas prejudiciais ou injustas. Outra questão relevante, é necessidade de um debate maior na sociedade para elencar responsabilidades quando existe alguma consequência no uso das informações recebidas via ChatGPT.

Análise Crítica

Analisando alguns casos específicos, o uso do ChatGPT revelou várias implicações éticas significativas. Em um estudo recente, Floridi (2023) discute os desafios éticos emergentes relacionados ao uso de IA em geral, destacando a necessidade de frameworks robustos para lidar com a transparência e a responsabilidade.

No contexto brasileiro, a pesquisadora Virgínia Dignum (2020) ressalta que a IA deve ser desenvolvida e utilizada com responsabilidade social, promovendo a justiça e a inclusão.

Além disso, um relatório recente da OpenAI (2023) identificou riscos associados à geração de desinformação, onde o ChatGPT foi capaz de criar conteúdo falso de maneira convincente, potencialmente influenciando negativamente a opinião pública e decisões políticas.

No Brasil, a questão da desinformação é particularmente relevante, dada a recente onda de fake news que tem impactado processos eleitorais e a confiança pública nas instituições (Ruediger, 2023).

Os vieses algorítmicos representam um dos maiores desafios éticos no desenvolvimento e uso de inteligência artificial. Estes vieses podem surgir de diversas maneiras, incluindo dados de treinamento enviesados, design inadequado do algoritmo e falta de diversidade nas equipes de desenvolvimento. Esses vieses podem resultar em discriminação e injustiça, afetando negativamente grupos minoritários e perpetuando desigualdades existentes na sociedade.

Por exemplo, estudos demonstram que modelos de IA treinados em dados históricos podem reproduzir e até amplificar preconceitos presentes nos dados, resultando em decisões enviesadas (Bender et al., 2021).

No contexto do ChatGPT, esses vieses podem se manifestar na geração de respostas que reforcem estereótipos ou discriminam certos grupos, como aconteceu em um workshop promovido pela Gradio, uma empresa especializada em testes de machine learning, onde preconceitos foram reproduzidos na maioria dos cenários de testes apresentados durante uma demonstração do uso do chatGPT (JOHNSON, 2021).

Para mitigar os vieses algorítmicos, é fundamental adotar práticas de design e desenvolvimento inclusivas, como a auditoria constante dos dados de treinamento e dos modelos, e a incorporação de diversas perspectivas na criação e avaliação dos algoritmos (Nascimento, 2021).

Ferramentas de auditoria de IA, como as desenvolvidas por Raji et al. (2020), são essenciais para identificar e corrigir esses vieses antes que causem impactos significativos.

A responsabilidade dos desenvolvedores de IA é um aspecto crucial da ética em tecnologia. Desenvolvedores têm o dever de garantir que seus sistemas sejam seguros, justos e transparentes. Isso inclui não apenas a criação de algoritmos éticos, mas também a implementação de práticas de monitoramento contínuo e a manutenção de um diálogo aberto com a sociedade sobre os usos e limitações dessas tecnologias.

Conforme destacado por Floridi (2023), a responsabilidade ética na IA envolve a necessidade de accountability, ou seja, a capacidade de os desenvolvedores e as empresas de tecnologia serem responsabilizados pelas consequências de seus sistemas.

No caso do ChatGPT, isso pode incluir garantir que as respostas geradas sejam verificáveis e que os dados dos usuários sejam tratados com o máximo respeito à privacidade e segurança (Silva & Souza, 2021).

Além disso, os desenvolvedores devem promover a transparência em seus processos, permitindo que usuários e reguladores compreendam como as decisões são tomadas pelos algoritmos.

Iniciativas como a criação de comitês de ética e a publicação de relatórios de impacto de IA são passos importantes para assegurar essa responsabilidade (Dignum, 2020). Os autores Rossetti e Angeluci (2021) também destacam a necessidade e importância da transparência algorítmica para possibilitar a correção dos processos que envolvem a compressão a avaliação das informações pelos algoritmos.

Soluções Responsáveis

Para abordar esses dilemas éticos, é importante se pensar em implementar soluções responsáveis que garantam a transparência, justiça e privacidade. Algumas das propostas incluem

a adoção de políticas rigorosas de privacidade de dados, conforme estabelecido pela LGPD, o desenvolvimento de algoritmos de mitigação de vieses e a implementação de práticas de design responsável que priorizem a ética. Sugere-se também a criação de regulamentos específicos que guiem o desenvolvimento e uso da IA, como o regulamento da União Europeia para IA, que estabelece diretrizes claras para a transparência e a responsabilização no uso de tecnologias de IA.

No Brasil, a Estratégia Brasileira de Inteligência Artificial (EBIA) destaca a importância de desenvolver IA com responsabilidade, ética e respeito aos direitos humanos (Ministério da Ciência, Tecnologia e Inovações, 2023).

Colaboração e Discussão

A colaboração entre desenvolvedores, pesquisadores, formuladores de políticas e a sociedade em geral é crucial para abordar as questões éticas associadas ao ChatGPT. Discussões abertas e inclusivas permitem a troca de diferentes perspectivas, enriquecendo o debate e promovendo soluções mais equilibradas e justas. A reflexão contínua e a adaptação das práticas de desenvolvimento de IA são necessárias para garantir que essas tecnologias beneficiem a sociedade como um todo.

Propostas de Ação Concreta para Garantir a Privacidade e Ética no Uso do ChatGPT.

1. Criação de Ferramentas de Auditoria e Transparência:

Uma proposta concreta é o desenvolvimento de ferramentas de auditoria que permitam monitorar e verificar o comportamento dos sistemas de IA, como o ChatGPT. Essas ferramentas poderiam incluir:

Auditoria Algorítmica: Ferramentas que permitam a análise dos algoritmos utilizados pelo ChatGPT para identificar e corrigir vieses e discriminações. Essas ferramentas poderiam ser disponibilizadas como software open source para promover a transparência e a colaboração entre diferentes partes interessadas.

Transparência de Dados: Plataformas que permitam aos usuários visualizar e controlar os dados que são coletados e utilizados pelos modelos de IA. Isso poderia incluir dashboards que mostram quais dados estão sendo usados e como, além de opções para que os usuários possam editar ou deletar suas informações.

2. Implementação de Políticas de Privacidade Rigorosas:

Para garantir a privacidade dos usuários, é essencial implementar políticas de privacidade rigorosas e assegurar que estas sejam comunicadas de maneira clara e acessível aos usuários. Medidas específicas podem incluir:

Consentimento Informado: Garantir que os usuários forneçam consentimento explícito e informado antes que seus dados sejam coletados ou utilizados. Isso pode ser feito através de interfaces amigáveis que expliquem de maneira clara como os dados serão usados.

Anonimização de Dados: Utilização de técnicas avançadas de anonimização e pseudonimização para proteger a identidade dos usuários. Isso ajuda a minimizar os riscos associados ao vazamento de dados pessoais.

Dados sensíveis: Desenvolver/aperfeiçoar ferramentas que consigam identificar quando vários dados sensíveis são utilizados e assim evitar que esses dados sejam armazenados ou mesmo processados.

3. Desenvolvimento de Práticas de Design Responsável:

A integração de princípios éticos no processo de design e desenvolvimento de IA pode ajudar a mitigar riscos e promover o uso responsável da tecnologia. Isso pode incluir:

Design Centrado no Usuário: Adotar práticas de design que priorizem as necessidades e direitos dos usuários, garantindo que os sistemas de IA sejam seguros e respeitosos da privacidade.

Inclusão e Diversidade: Envolver diversos grupos de stakeholders no processo de design e desenvolvimento para garantir que diferentes perspectivas sejam consideradas e que os sistemas de IA sejam inclusivos e justos.

4. Fomento à Educação e Conscientização:

Promover a educação e a conscientização sobre as implicações éticas da IA é fundamental para criar uma cultura de responsabilidade e ética. Isso pode ser alcançado através de:

Programas de Capacitação: Oferecer programas de capacitação e treinamento para desenvolvedores, formuladores de políticas e o público em geral sobre os princípios éticos da IA e as melhores práticas para seu desenvolvimento e uso. **Campanhas de Conscientização:** Lançar campanhas de conscientização pública sobre os direitos dos usuários e as implicações éticas da IA, incentivando uma participação ativa e informada da sociedade no debate sobre essas tecnologias.

CONCLUSÃO

O estudo das implicações éticas do uso do ChatGPT mostra a necessidade urgente de abordagens responsáveis na implementação de tecnologias de inteligência artificial. É essencial identificar e resolver dilemas éticos, como privacidade, disseminação de desinformação e discriminação algorítmica, para garantir que o desenvolvimento e uso da IA estejam alinhados com valores de justiça, transparência e respeito aos direitos humanos.

Propostas concretas, como a criação de ferramentas de auditoria e transparência, a implementação de políticas de privacidade rigorosas, o desenvolvimento de práticas de design

responsável e a promoção da educação e conscientização, são fundamentais para garantir um uso ético e responsável da IA. A adoção dessas medidas pode ajudar a mitigar riscos, promover a confiança dos usuários e assegurar que a IA seja utilizada de maneira justa e benéfica.

Além disso, é crucial promover a colaboração entre desenvolvedores, pesquisadores, formuladores de políticas e a sociedade em geral. Discussões abertas permitem a troca de diferentes perspectivas, enriquecendo o debate e promovendo soluções mais equilibradas e justas. A reflexão contínua e a adaptação das práticas de desenvolvimento de IA são necessárias para garantir que essas tecnologias beneficiem a sociedade como um todo.

Portanto, integrar princípios éticos no desenvolvimento e uso do ChatGPT e outras tecnologias de IA não é apenas necessário, mas uma responsabilidade coletiva. Somente através de uma abordagem ética e colaborativa podemos garantir que a inteligência artificial seja uma força positiva na sociedade, promovendo o bem-estar e respeitando os direitos de todos.

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

Nome do artigo escolhido:

Requirements practices and gaps when engineering human-centered Artificial Intelligence systems.

Qual o objetivo do estudo descrito pelo artigo?	Qual o problema/oportunidade/situação que levou à necessidade de 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?
--	--	---	--

Este estudo propôs-se a mapear diretrizes centradas no ser humano relevantes para a Engenharia de Requisitos aplicada à Inteligência Artificial (RE4AI) e a examinar a literatura atual sobre o tema. A investigação teve como foco identificar discrepâncias entre as práticas industriais e as orientações acadêmicas, buscando entender de que forma abordagens centradas no ser humano estão sendo incorporadas ao desenvolvimento de sistemas de IA, com ênfase na experiência do usuário.	O problema abordado é a dificuldade de integrar efetivamente as diretrizes centradas no ser humano nas práticas de Engenharia de Requisitos para IA. Apesar da literatura fornecer recomendações para o desenvolvimento de IA com foco no ser humano, os profissionais enfrentam desafios na aplicação dessas diretrizes, em parte devido à escassez de ferramentas e metodologias adequadas. O estudo foi motivado pela necessidade de explorar como essas diretrizes podem ser mais bem aplicadas na prática, visando reduzir o descompasso entre a teoria e a prática neste campo.	O estudo foi dividido em duas etapas: Primeiro, eles mapearam as diretrizes da indústria sobre IA centrada no ser humano e revisaram a literatura existente sobre Engenharia de Requisitos para IA (RE4AI). Na segunda etapa, eles realizaram uma pesquisa com profissionais da área para entender as práticas atuais em RE4AI e identificar como as abordagens centradas no ser humano estão sendo aplicadas na criação de sistemas de IA. Para analisar as informações, usaram uma combinação de revisão teórica e pesquisa baseada na experiência dos participantes.	Os principais resultados do estudo foram: • Existe uma diferença entre o que a literatura recomenda para IA centrada no ser humano e as práticas realmente usadas na indústria, especialmente em relação às ferramentas e métodos para modelagem de requisitos. • As ferramentas como JIRA e Excel são muito utilizadas, mas têm limitações para gerenciar requisitos específicos de IA, como questões éticas e dados. • A UML é amplamente usada na indústria, mas apresenta limitações quando se trata de modelar requisitos não funcionais. • Apesar de a literatura focar em áreas como veículos autônomos e saúde, os profissionais estão explorando mais áreas como educação, governo e defesa na prática. • A maioria dos profissionais entrevistados não tem experiência específica em Engenharia de Requisitos para IA, o que aponta para uma oportunidade de aprimorar a formação na área.
---	--	---	---

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"
```

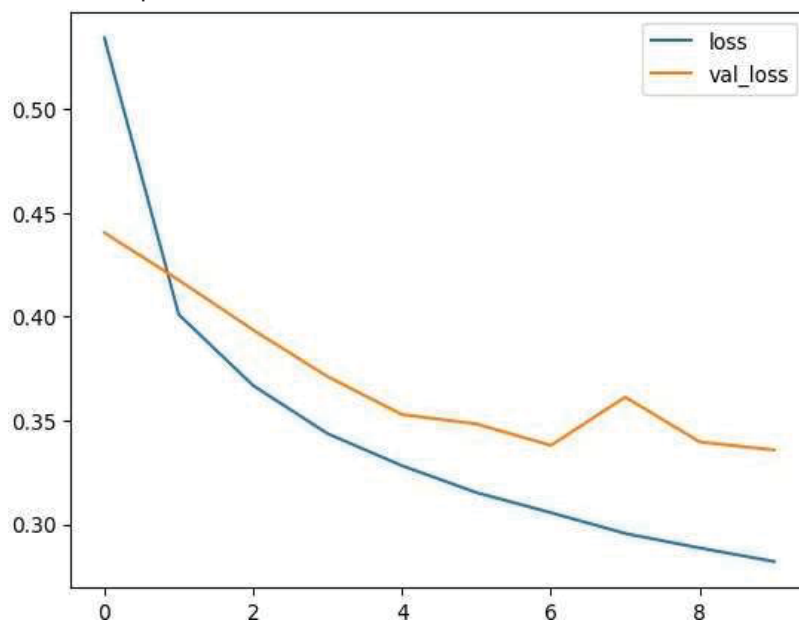
Dica: Para exibir a imagem utilizando display (display.html) use o link https://commons.wikimedia.org/wiki/File:Felis_catus-cat_on_snow.jpg

B – RESOLUÇÃO

- 1 Classificação (RNA)
https://colab.research.google.com/drive/1uHmWhj0Vu2Bt0uSEhvUnmiRgJi_bVNzE?usp=sharing

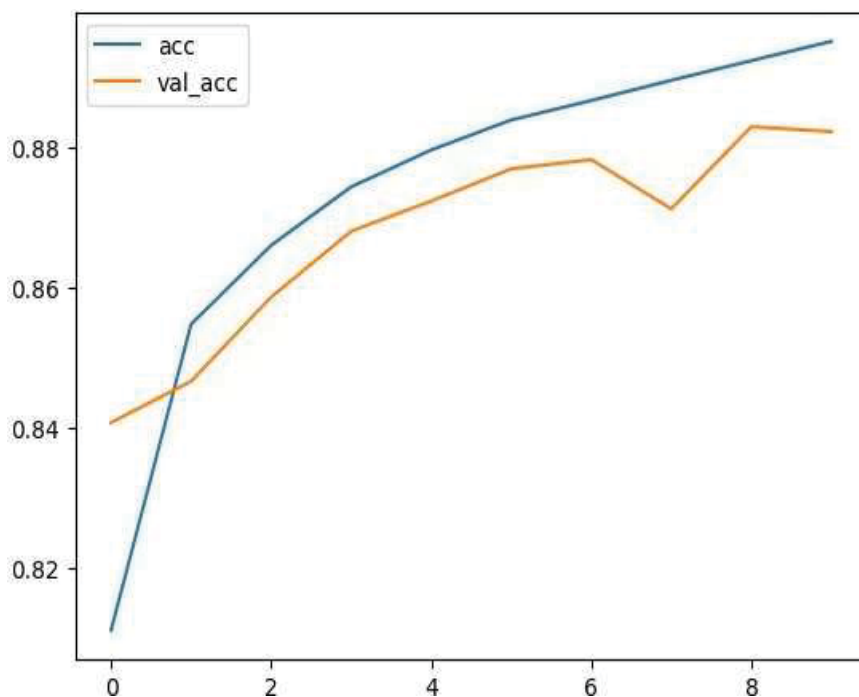
Função de Perda: 0.3329

O gráfico mostra que as perdas de treinamento (**loss**) e validação (**val_loss**) diminuem com as épocas, indicando aprendizado do modelo.



- **Primeiras épocas:** Queda acentuada em ambas as perdas, mostrando rápido aprendizado inicial.
- **Últimas épocas:** **loss** continua caindo, mas **val_loss** estabiliza com pequenas flutuações.

Função de Acurácia: 0.8826

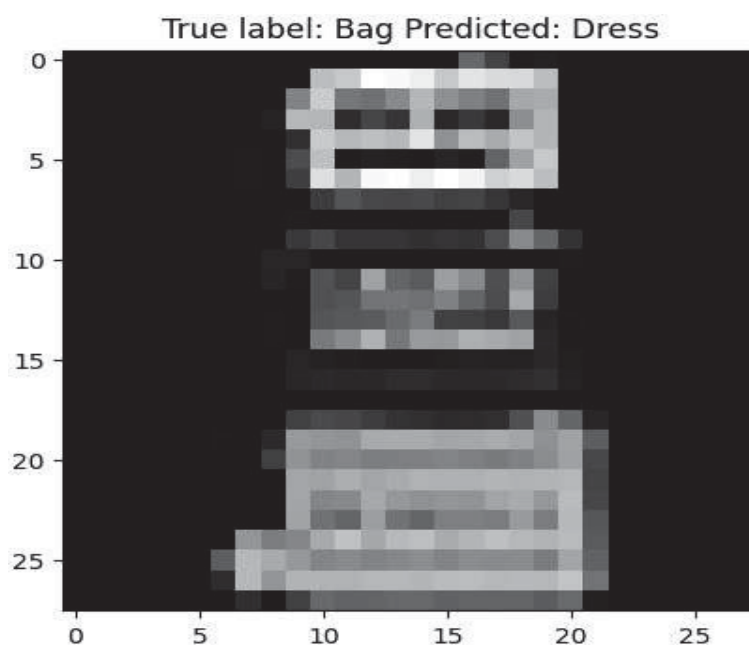


O gráfico mostra que, ao longo das épocas, a acurácia no conjunto de treinamento (**acc**)

cresce de forma consistente, enquanto a acurácia no conjunto de validação (**val_acc**) também aumenta, mas apresenta pequenas flutuações.

- **Primeiras épocas:** Melhorias rápidas em ambas as acurácias, indicando aprendizado inicial eficiente.
- **Últimas épocas:** **acc** continua crescendo, mas **val_acc** se estabiliza e começa a divergir.

Mostrar algumas classificações erradas: A predição aponta que esta imagem é um Dress mas na verdade é um Bag



- 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
```

```

'dioido_de_enxofre_livre', # free sulfur dioxide 'dioido_de_enxofre_total', # total sulfur
dioxide 'densidade',      # density
'pH', # pH
'sulfatos', # sulphates
'alcool',    # alcohol 'score_qualidade_vinho' # quality
]

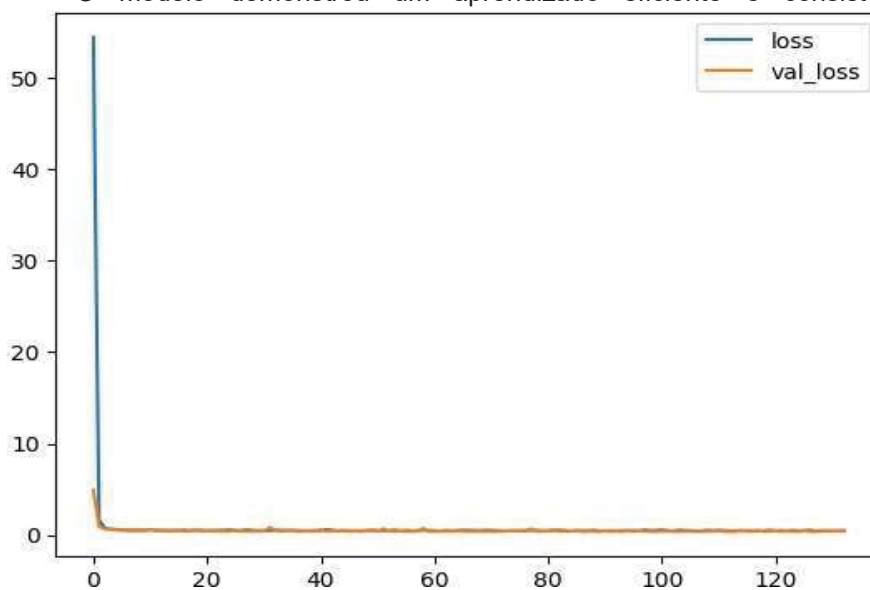
```

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)

<https://colab.research.google.com/drive/1zzi9c5piUZ-3VORkjMoUvIpra3CmiSHD?usp=sharing>

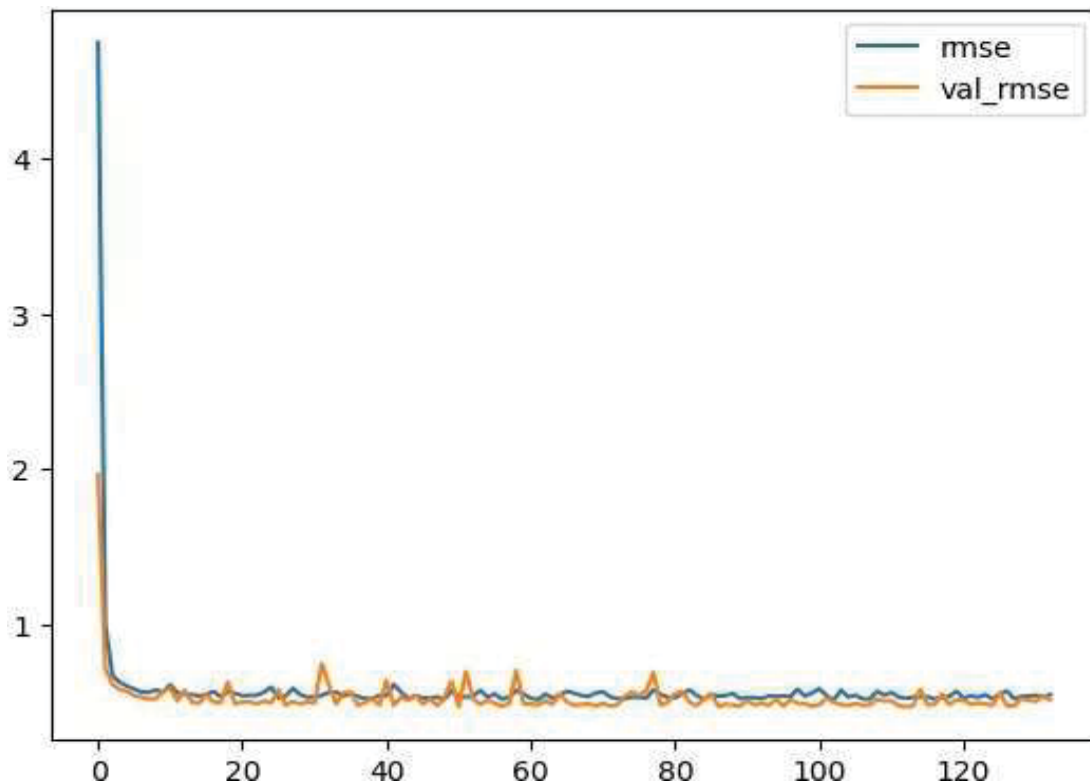
- **Função de Perda:**

O modelo demonstrou um aprendizado eficiente e consistente, alcançando boa



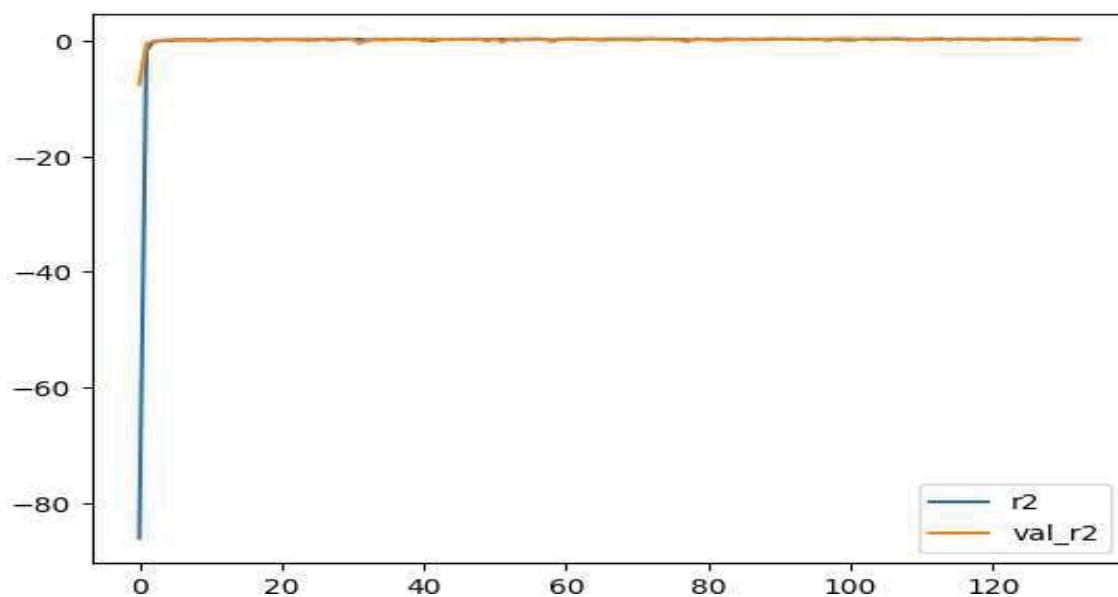
generalização e evitando problemas de sobreajuste ou subajuste. Esses resultados indicam que a arquitetura e os parâmetros utilizados foram adequados para a tarefa de regressão.

- **Função RMSE:**



O gráfico de RMSE indica um bom aprendizado do modelo, com rápida redução do erro inicial e estabilização em valores baixos após cerca de 10-15 épocas. As curvas de treino e validação são próximas, mostrando boa generalização. Não há sinais de overfitting, e o desempenho foi consistente nos dados de validação. O modelo é eficaz para a tarefa proposta.

- **Função R2:**



O gráfico do coeficiente R^2 mostra uma rápida melhoria inicial, alcançando valores estáveis após poucas épocas. As curvas de treino e validação são muito próximas, indicando boa generalização do modelo. O R^2 próximo de 0 ou levemente negativo sugere que o modelo ainda não explica completamente a variação dos dados. Ajustes adicionais podem melhorar a qualidade do ajuste.

- **Conclusão:**

Os resultados indicam um MSE de 0.35 e um RMSE de 0.59, representando um erro moderado na previsão. O R^2 de 0.43 mostra que o modelo explica cerca de 43% da variância dos dados, sugerindo espaço para melhorias no desempenho preditivo.

- 3 Sistemas de Recomendação

Implementar o exemplo de Sistemas de Recomendação usando a base de dados `Base_livros.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.

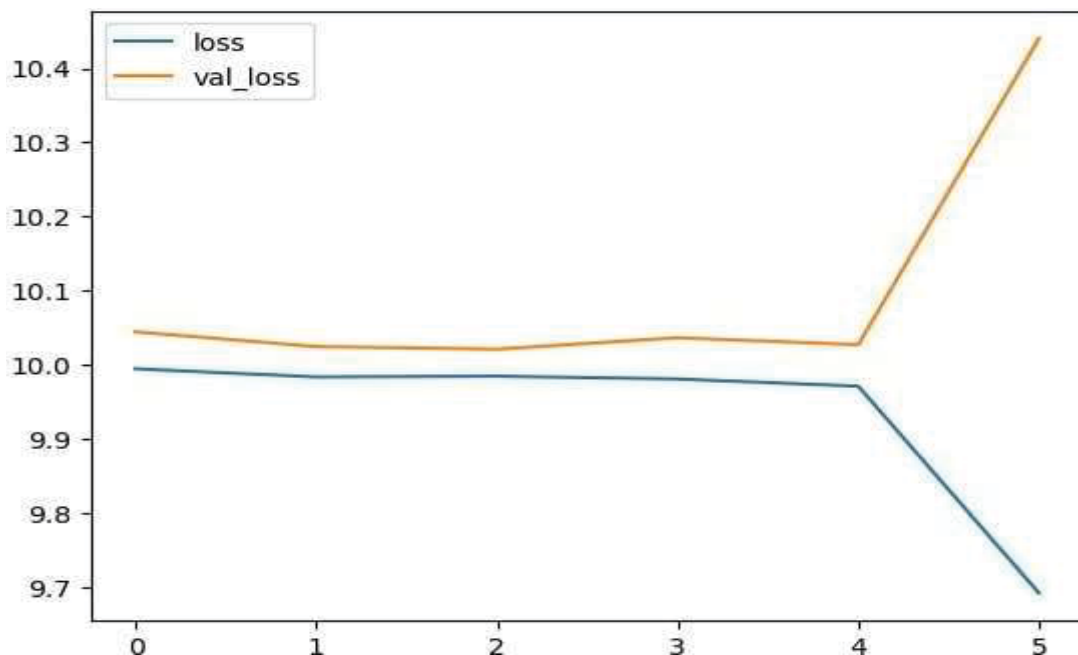
Informações:

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

[https://colab.research.google.com/drive/1HmgSvY8L_E7yWjWUjA1dPbokYt57LFfd#scrollTo=W863](https://colab.research.google.com/drive/1HmgSvY8L_E7yWjWUjA1dPbokYt57LFfd#scrollTo=W863APQBsYWT)

APQBsYWT

- **Função de perda:**



Análise e Observações: O gráfico mostra que durante as primeiras épocas do treinamento do modelo, o aprendizado ainda é baixo, por isso os valores de loss e val_loss praticamente não sofrem alteração, porém a partir da quinta época, o valor de loss diminuiu de forma agressiva, mas o val_loss chega a aumentar.

Devido a função early stopping configurada para monitorar o val_loss (função de perda dos dados de validação), o treinamento foi interrompido, pois o val_loss não estava melhorando.

Essa parada antecipada mostra que a continuidade do treinamento resultaria em overfitting, visto que os dados de validação não estavam tendo evolução no aprendizado e por isso o modelo não está generalizando bem os novos dados.

Justificativa dos resultados e sugestão de Intervenção: Alguns dos parâmetros utilizados podem ser ajustados para buscar um modelo mais eficiente, além disso, outras questões podem ser verificadas. Algumas ações que podem tornar o modelo melhor seriam a redução da taxa de aprendizado (foi usado 0,05) visando evitar oscilações e a redução do número de neurônios (foi usado 1024) na camada densa, visto que a redução do número de pesos pode auxiliar na generalização. O número de épocas pode ser aumentado para um número entre 50 e 100, pois desta maneira o modelo poderá ver os dados mais vezes e consequentemente pode ajustar os pesos de uma forma mais gradual para chegar de uma convergência com melhor precisão. A alteração do batch_size também poderá melhorar o modelo, pois o número usado (512) acabou não capturando bem a variação dos dados usados. Outros testes que podem ser feitos são a alteração do algoritmo otimizador para verificação se o resultado final tem um ganho na generalização e testar outros valores mais altos no

dropout na camada densa (neste caso, também pode ser necessário ajustar o número de épocas para uma avaliação mais eficiente).

- **Exemplo de recomendação de livro:**

- ✓ 9. Recomendações para o usuário 20584

```
# Gerar o array com o usuário único
# repete a quantidade de livros
input_usuario = np.repeat(a=20584, repeats=M)
livro = np.array(list(set(ISBN_ids)))

preds = model.predict( [input_usuario, livro] )

# descentraliza as predições
rat = preds.flatten() + avg_notas

# índice da maior nota
idx = np.argmax(rat)

print("Recomendação: Livro - ", livro[idx], " / ", rat[idx] , "**")
```

4028/4028 ————— 6s 1ms/step
Recomendação: Livro - 77837 / 5.0071154 *

No exemplo acima, é feita uma sugestão de livro para o usuário 20584. Além de apontar o id do livro sugerido, é feita a predição de uma nota aproximada que o usuário deve atribuir ao livro, considerando todas as notas dadas ao livro por outros usuários e também as notas que o próprio usuário tende a atribuir aos livros que ele já fez a leitura.

- 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

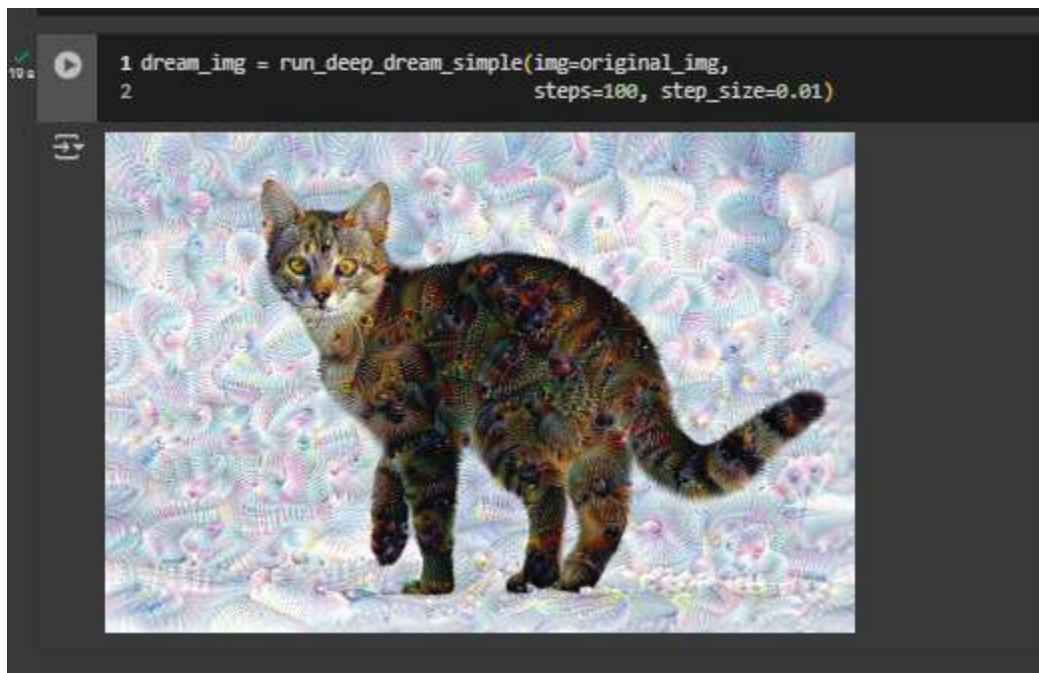
- **Importação da imagem:** Copiar código abaixo.

url = "https://commons.wikimedia.org/wiki/Special:FilePath/Felis_catus-cat_o_n_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

Link do colab : https://colab.research.google.com/drive/1s3oRhAKXCTitpEWZvyDQ8boh_57P7LoC#scrollTo=0eGDSdatLT-8

- Imagem onírica obtida por *Main Loop*;



A imagem onírica obtida no Main Loop resulta do processo de amplificação de padrões que o modelo pré-treinado reconhece nos dados de entrada. Esse efeito é gerado utilizando a técnica conhecida como Deep Dream.

Explicação Breve dos Resultados O que é o Deep Dream?

O Deep Dream é uma técnica baseada em redes neurais convolucionais (como InceptionV3) que maximiza as ativações de camadas específicas. Isso amplifica os padrões que a rede reconhece na imagem, criando formas "surreais" e texturas complexas. Como o Main Loop funciona?

A imagem de entrada é processada iterativamente. O modelo tenta aumentar as ativações dos filtros das camadas especificadas no dream_model. Em cada passo, o modelo ajusta os valores dos pixels da imagem de entrada para amplificar os padrões. Características da Imagem Resultante:

Texturas repetitivas: Elementos visuais, como curvas, olhos ou formas geométricas, são amplificados, criando um aspecto "onírico". Padrões vibrantes: As cores e os detalhes tornam-se mais pronunciados à medida que os filtros são maximizados. Integração dos padrões: As texturas não são adicionadas aleatoriamente; elas emergem de áreas da imagem onde os filtros da rede detectaram elementos significativos. Por que a imagem do gato ficou surreal?

O modelo identificou características específicas do gato (como olhos, pelos e contornos) e amplificou esses padrões. Camadas diferentes da rede contribuem para texturas e formas mais complexas à medida que os filtros interpretam a imagem.

- **Imagem onírica obtida ao levar o modelo até uma oitava**



A imagem onírica obtida com o uso de oitavas apresenta padrões mais refinados e complexos, pois combina informações em múltiplas escalas de resolução. Detalhes maiores, como o corpo do gato, são destacados em resoluções menores, enquanto texturas finas, como olhos e contornos, são refinadas em resoluções maiores. Esse processo multiescala reduz o ruído e cria uma integração mais harmoniosa dos padrões amplificados. No código, a imagem é redimensionada iterativamente com o fator OCTAVE_SCALE, passando pelo Deep Dream em cada nível antes de ser retornada ao tamanho original. O resultado é uma obra visual rica, detalhada e surreal.

- **Diferenças entre imagens oníricas obtidas com *Main Loop* e levando o modelo até a oitava.**

As diferenças entre as imagens oníricas obtidas com o Main Loop e levando o modelo até uma oitava são as seguintes:

Detalhamento Multiescala: No Main Loop, os padrões são amplificados em uma única resolução, resultando em detalhes uniformes e, às vezes, caóticos. Com oitavas, o modelo processa a imagem em diferentes escalas, refinando padrões em níveis globais e locais.

Redução de Ruído: A imagem do Main Loop tende a ser mais ruidosa devido à amplificação direta. Usando oitavas, o redimensionamento multiescala suaviza transições e integra melhor os padrões.

Complexidade Visual: No Main Loop, os detalhes podem parecer menos harmoniosos. Já com oitavas, os padrões são mais ricos e detalhados, devido ao refinamento em várias resoluções.

Resolução e Granularidade: A imagem do Main Loop é processada diretamente na resolução original, enquanto com oitavas, a granularidade dos padrões melhora em várias escalas de tamanho.

Harmonia Visual: A técnica com oitavas cria uma composição mais equilibrada, combinando detalhes finos e amplos, enquanto o Main Loop é limitado à resolução única do processamento.

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)

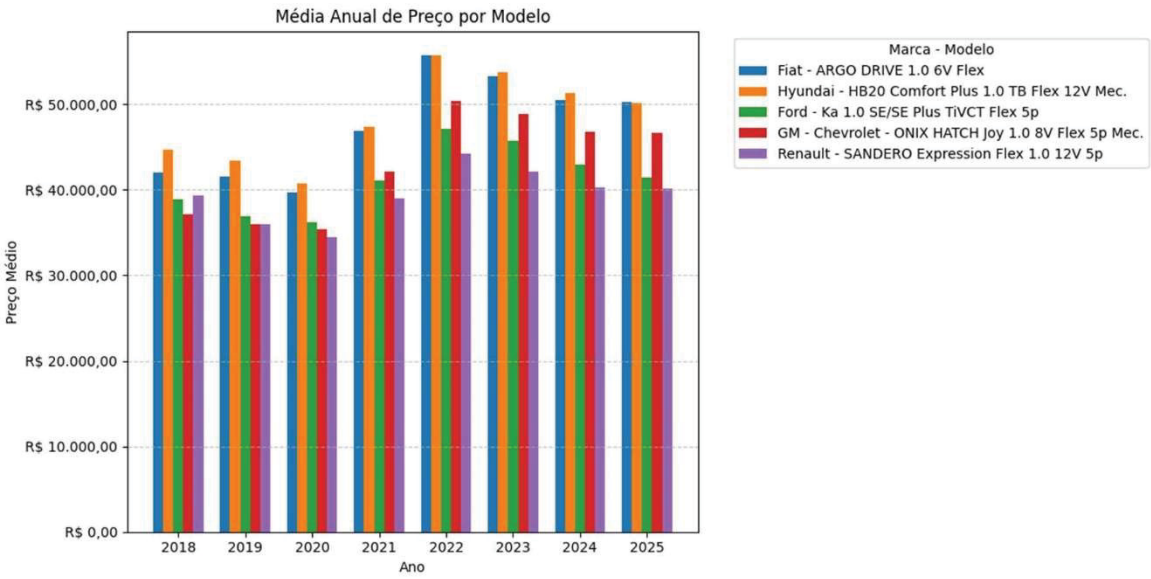
Desenvolva uma narrativa/storytelling para essa visualização de dados considerando os conceitos e informações que foram discutidas nesta disciplina. Não esqueça de deixar claro para seu possível público alvo qual **o objetivo dessa visualização de dados, o que esses dados significam, quais possíveis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

Quais modelos de carros hatch compactos fabricados em 2018 tiveram a menor desvalorização nos últimos 7 anos?



Amostra dos dados analisados:

index	MesReferencia	Valor	Marca	Modelo	AnoModelo	Combustivel
252	2024-07-01 00:00:00	5067800.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
327	2024-08-01 00:00:00	5055500.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
320	2024-09-01 00:00:00	5005800.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
335	2024-10-01 00:00:00	4973400.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
317	2024-11-01 00:00:00	5028000.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
281	2024-12-01 00:00:00	5037700.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
282	2025-01-01 00:00:00	5037000.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
256	2025-02-01 00:00:00	5065000.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
263	2025-03-01 00:00:00	4975400.0	Fiat	ARGO DRIVE 1.0 6V Flex	2018.0	Gasolina
70	2018-04-01 00:00:00	3799800.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
64	2018-05-01 00:00:00	3852400.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
79	2018-06-01 00:00:00	3810000.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
29	2018-07-01 00:00:00	3915400.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
81	2018-08-01 00:00:00	3954100.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
32	2018-09-01 00:00:00	3920000.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
22	2018-10-01 00:00:00	3973100.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
38	2018-11-01 00:00:00	3899200.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina
7	2018-12-01 00:00:00	3905600.0	Ford	Ka 1.0 SE/SE Plus TIVCT Flex 5p	2018.0	Gasolina

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

Contexto

Os carros hatches compactos são o segmento mais comercializado do Brasil. Com base nos dados históricos da tabela FIPE, esta análise revela quais são os 5 modelos mais comercializado e fabricados em 2018 que perderam menos valor nos últimos sete anos, destacando tendências do mercado e fatores que influenciam a valorização/desvalorização.

Público-Alvo

Compradores de seminovos – Buscam modelos com menor depreciação. **Revendedores e investidores** – Avaliam veículos mais rentáveis na revenda. **Entusiastas do setor** – Interessados em tendências automotivas.

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

Algoritmo Genético

```
import random
import numpy as np
import matplotlib.pyplot as plt

NUM_CIDADES = 100 # 100 cidades
ESPACO = 100 # Espaço de 100x100 pixels
POPULACAO_SIZE = 100 # (3) População mínima de 100 indivíduos
GERACOES = 1000 # Número de gerações
TAXA_MUTACAO = 0.01 # (4) 1% de mutação
TAXA_CRUZAMENTO = 0.9 # (5) 90% de crossover

## Gerar cidades com coordenadas aleatórias dentro do espaço definido
cidades = np.random.randint(0, ESPACO, (NUM_CIDADES, 2))

# Imprimir as coordenadas das cidades geradas
print("Coordenadas das cidades:\n", cidades)

# Criar um indivíduo (permutação das cidades, sempre começando na cidade 0)
def criar_individuo():
    percurso = list(range(1, NUM_CIDADES)) # Lista de cidades sem a cidade 0
    random.shuffle(percurso) # Embaralha as cidades
    return [0] + percurso + [0] # Inclui a cidade 0 no início e no final

# Criar população inicial com indivíduos aleatórios
def populacao_inicial():
    return [criar_individuo() for _ in range(POPULACAO_SIZE)]

# Execução do Algoritmo Genético
```

```

# Contar o número de indivíduos na população numero_individuos = len(populacao)

# Imprimir o número de indivíduos
print(f"Número de indivíduos na população: {numero_individuos}") populacao =
populacao_inicial()
print(populacao)

# Função para calcular a distância euclidiana entre duas cidades def distancia(c1,
c2):
return np.linalg.norm(c1 - c2)

# Função de avaliação (fitness): distância total do percurso def
avaliacao(percurso):
dist = 0 # Inicializa a variável dist para armazenar a soma das distâncias

# Loop para percorrer todas as cidades no percurso (exceto a última) for i in
range(len(percurso) - 1):
# Calcula a distância entre a cidade atual (percurso[i]) e a próxima cidade
(percurso[i + 1])
dist += distancia(cidades[percurso[i]], cidades[percurso[i + 1]])
# Retorna o inverso da distância total, pois queremos um valor maior para percursos
mais curtos
return 1 / dist # Quanto menor a distância, melhor o fitness

# Percorrendo toda a população para encontrar o melhor percurso for percurso in
populacao:
if avaliacao(percurso) > avaliacao(melhor_inicial): melhor_inicial = percurso

#melhor_inicial = max(populacao, key=avaliacao) print(melhor_inicial)

# Função para plotar o percurso e numerar as cidades def plotar(percurso, titulo):
plt.figure(figsize=(8, 8))
x, y = zip(*[cidades[i] for i in percurso]) plt.plot(x, y, 'bo-', alpha=0.6) #
Traça o caminho
plt.plot([x[0]], [y[0]], 'ro', markersize=10) # Destaca a cidade inicial

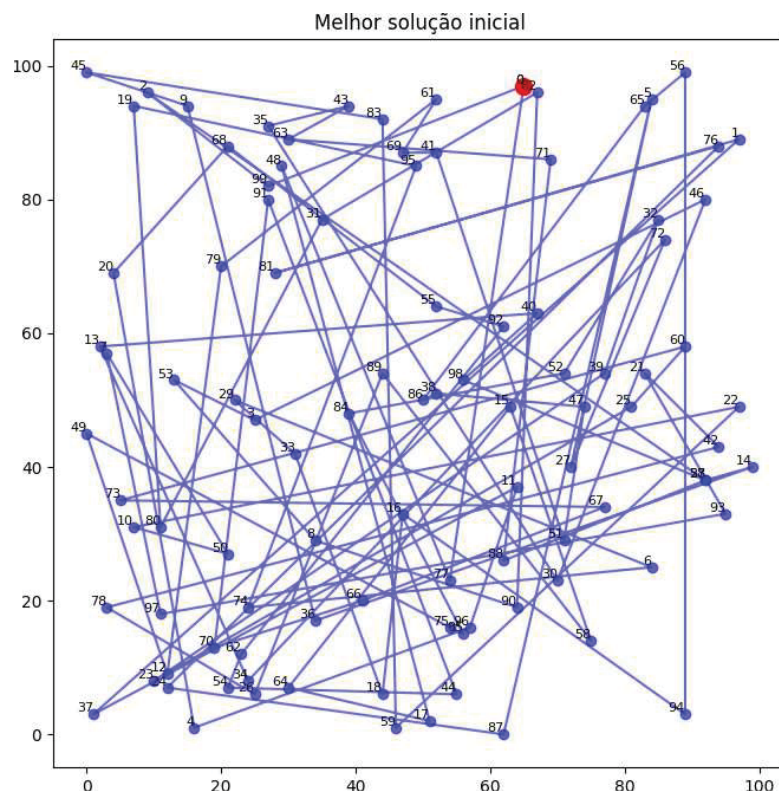
# Adiciona os números das cidades

```

```
for i, (x_coord, y_coord) in enumerate(zip(x, y)): plt.text(x_coord, y_coord,
str(percurso[i]), fontsize=8,
ha='right', va='bottom', color='black')
```

```
plt.title(titulo) plt.show()
```

```
plotar(melhor_inicial, "Melhor solução inicial")
```



```
# Seleção dos pais por torneio (escolhe o melhor entre 5 indivíduos aleatórios)
```

```
def selecao(populacao):
```

```
    return sorted(random.sample(populacao, 5), key=avaliacao, reverse=True)[0]
```

```
# Cruzamento OX (Order Crossover) entre dois pais def cruzamento(pai1, pai2):
```

```
    tamanho = len(pai1)
```

```
    inicio, fim = sorted(random.sample(range(1, tamanho - 1), 2)) # Ponto de corte pai
```

```
    filho = [-1] * tamanho # Inicializa filho com valores inválidos filho[inicio:fim]
```

```
    = pai1[inicio:fim] # Copia segmento do primeiro
```

```
    p2_idx = fim # Índice do segundo pai for i in range(fim, tamanho - 1):
```

```

while pai2[p2_idx] in filho:
    p2_idx = (p2_idx + 1) % (tamanho - 1) filho[i] = pai2[p2_idx]

for i in range(1, inicio):
    while pai2[p2_idx] in filho:
        p2_idx = (p2_idx + 1) % (tamanho - 1) filho[i] = pai2[p2_idx]

    filho[0] = filho[-1] = 0 # Mantém a cidade inicial no começo e no return filho
f
i
m

# Mutação (troca de duas cidades aleatórias no percurso) def mutacao(individuo):
if random.random() < TAXA_MUTACAO:
    i, j = random.sample(range(1, NUM_CIDADES - 1), 2) individuo[i], individuo[j] =
    individuo[j], individuo[i]

# Evolução da população através de seleção, cruzamento e mutação

def evoluir(populacao):
    nova_populacao = [max(populacao, key=avaliacao)] # Preserva o melhor indivíduo
    num_filhos = int(TAXA_CRUZAMENTO * POPULACAO_SIZE)

    while len(nova_populacao) < num_filhos:
        pai1, pai2 = selecao(populacao), selecao(populacao) filho = cruzamento(pai1, pai2)
        mutacao(filho) nova_populacao.append(filho)

    while len(nova_populacao) < POPULACAO_SIZE:
        nova_populacao.append(criar_individuo()) # Adiciona novos
        indivíduos aleatórios

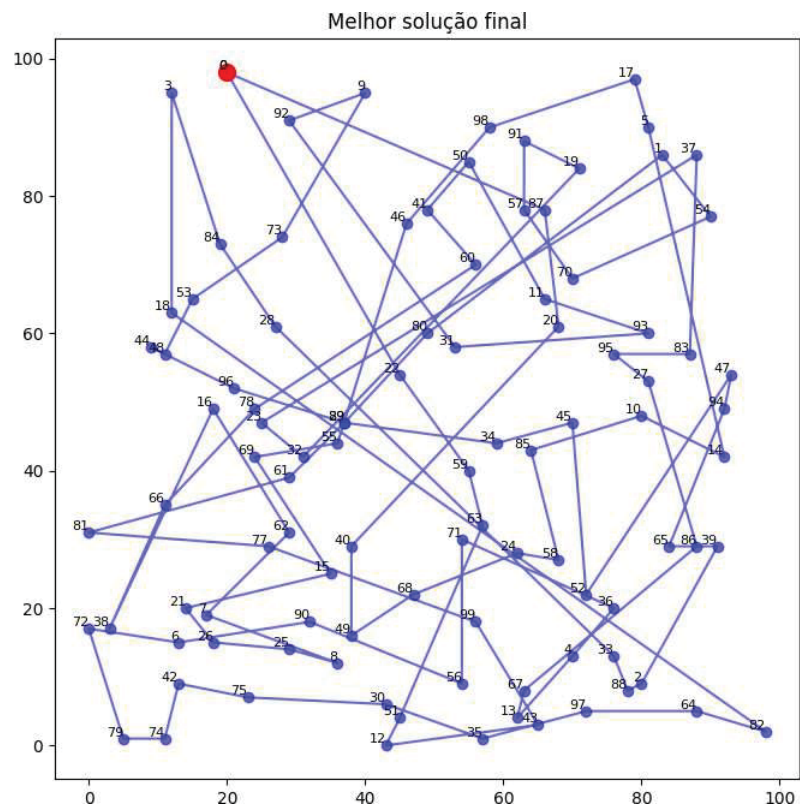
    return nova_populacao

# Loop de gerações para evoluir a população for i in range(GERACOES):
populacao = evoluir(populacao)
if i % 100 == 0: # Exibe progresso a cada 100 gerações melhor = max(populacao,
    key=avaliacao) print(f'Geração {i}: Melhor distância = {1 /
    avaliacao(melhor):.2f}')

```

```
# Exibir a melhor solução final
top_individuo = max(populacao, key=avaliacao) plotar(top_individuo, "Melhor
solução final")
```

Geração 0: Melhor distância = 4338.38
 Geração 100: Melhor distância = 2723.20
 Geração 200: Melhor distância = 2526.27
 Geração 300: Melhor distância = 2416.34
 Geração 400: Melhor distância = 2368.94
 Geração 500: Melhor distância = 2363.90
 Geração 600: Melhor distância = 2311.31
 Geração 700: Melhor distância = 2202.36
 Geração 800: Melhor distância = 2145.22
 Geração 900: Melhor distância = 2091.68



Compare a representação de dois modelos vetoriais

```
# =====
```

```

# Instalação de bibliotecas (Colab) # =====
!pip install -q sentence-transformers scikit-learn

# =====
# Imports
# =====
from sklearn.feature_extraction.text import TfidfVectorizer from
sentence_transformers import SentenceTransformer
from sklearn.decomposition import PCA import matplotlib.pyplot as plt import numpy
as np from sklearn.metrics.pairwise import cosine_similarity

# =====
# Entrada do usuário
# =====
def verificar_qtde_sentencas_validas(): while True:
print("Digite um texto com até 1000 caracteres (mínimo 6 frases com algumas
parecidas):\n")
texto = input()[:1000]

# Fragmentar em sentenças
import re

sentencas = re.split(r'(?<=[.!?])\s+', texto.strip())

# Filtro: entre 6 e 10 sentenças
sentencas = [s for s in sentencas if s] # Remover sentenças

if len(sentencas) < 6:

```

```

print(".1 continuar.\n")
continue
sentencas = sentencas[:10] # limite para clareza do gráfico

print("\n•, ' )  Sentenças selecionadas:") for i, s in enumerate(sentencas):
print(f"{i+1}. {s}")

return sentencas

sentencas = verificar_qtde_sentencas_validas()

# =====
# Modelo 1: TF-IDF
# =====
vectorizer = TfidfVectorizer()
tfidf_vectors = vectorizer.fit_transform(sentencas).toarray()

# =====
# Modelo 2: Sentence Embedding (BERT) # =====

model      =      SentenceTransformer('all-MiniLM-L6-v2')      bert_vectors      =
model.encode(sentencas)

# =====
# Calcular Similaridades
# =====

def calc_similaridade_tfidf(vetores, limiar=0.6): matriz_similaridade_tfidf =
cosine_similarity(vetores)
np.fill_diagonal(matriz_similaridade_tfidf, 0) # Zerar diagonal para não contar
similaridade
return (matriz_similaridade_tfidf >= limiar).sum() >= 2 # Pelo menos
2 sentenças com similaridade

def calc_similaridade_bert(vetores, limiar=0.7): matriz_similaridade_bert =
cosine_similarity(vetores) np.fill_diagonal(matriz_similaridade_bert, 0) # Zerar
diagonal para não contar similaridade

```

```

return (matriz_similaridade_bert >= limiar).sum() >= 2 # Pelo menos
2 sentenças com similaridade

### Função para exibir mapa de calor

import seaborn as sns
def plot_heatmap(sim_matrix, titulo): plt.figure(figsize=(8, 6))
sns.heatmap(sim_matrix, annot=True, cmap="coolwarm", fmt=".2f",
xticklabels=range(1, len(sim_matrix) + 1), yticklabels=range(1, len(sim_matrix) +
1))
plt.title(f"Matriz de Similaridade - {titulo}") plt.xlabel("Sentenças")

plt.ylabel("Sentenças") plt.show()

# Verificar se pelo menos 2 sentenças têm similaridade while True:

tfidf_sim_matrix = cosine_similarity(tfidf_vectors) bert_sim_matrix =
cosine_similarity(bert_vectors)

plot_heatmap(tfidf_sim_matrix, "TF-IDF") plot_heatmap(bert_sim_matrix, "BERT")
if calc_similaridade_tfidf(tfidf_vectors) or
calc_similaridade_bert(bert_vectors):
break # Se pelo menos um critério for atendido, continue

print("\n+ 0 texto não tem pelo menos 2 sentenças similares (≥ 0.6 para TF-IDF ou
≥ 0.7 para BERT). Tente novamente.\n")

# Solicitar novas sentenças e recalcular vetores sentencas =
verificar_qtde_sentencas_validas()
tfidf_vectors = vectorizer.fit_transform(sentencas).toarray() bert_vectors =
model.encode(sentencas)
# =====
# Redução PCA
# =====
def reduzir_pca(vetores): pca = PCA(n_components=2)
return pca.fit_transform(vetores)

```

```
tfidf_2d = reduzir_pca(tfidf_vectors) bert_2d = reduzir_pca(bert_vectors)

# =====
# Plot dos Vetores (TF-IDF vs BERT) # =====
def plot_vetores(vetores_2d, sentencas, titulo): plt.figure(figsize=(8, 6))
for i, coord in enumerate(vetores_2d): plt.scatter(coord[0], coord[1],
label=f"{i+1}")
plt.text(coord[0]+0.02, coord[1]+0.02, f"{i+1}", fontsize=9) plt.title(f'Projeção
PCA - {titulo}')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2') plt.grid(True)
plt.legend() plt.show()

plot_vetores(tfidf_2d, sentencas, "TF-IDF") plot_vetores(bert_2d, sentencas, "BERT
Embedding")
```

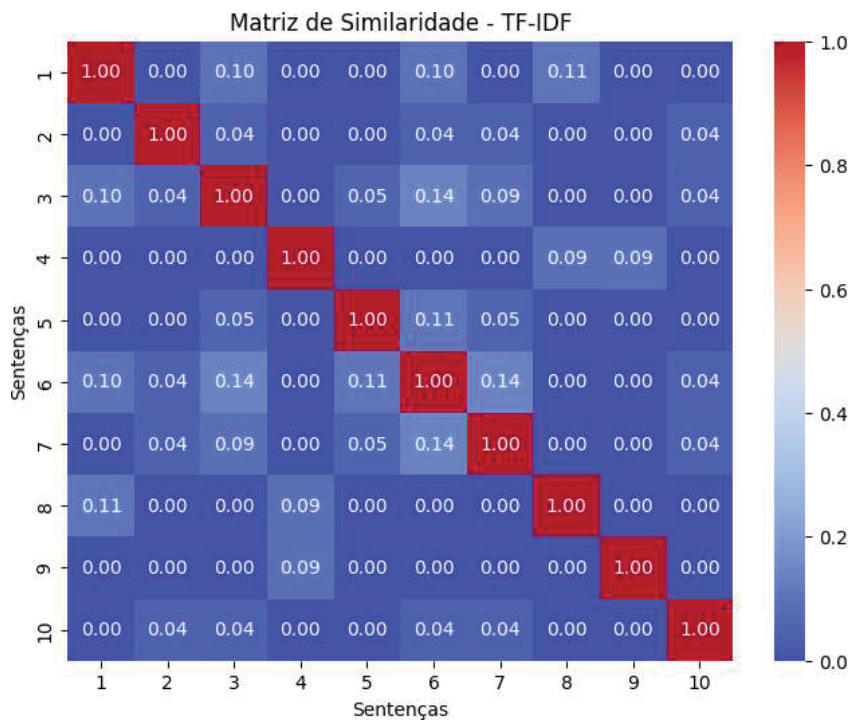
Digite um texto com até 1000 caracteres (mínimo 6 frases com algumas parecidas):

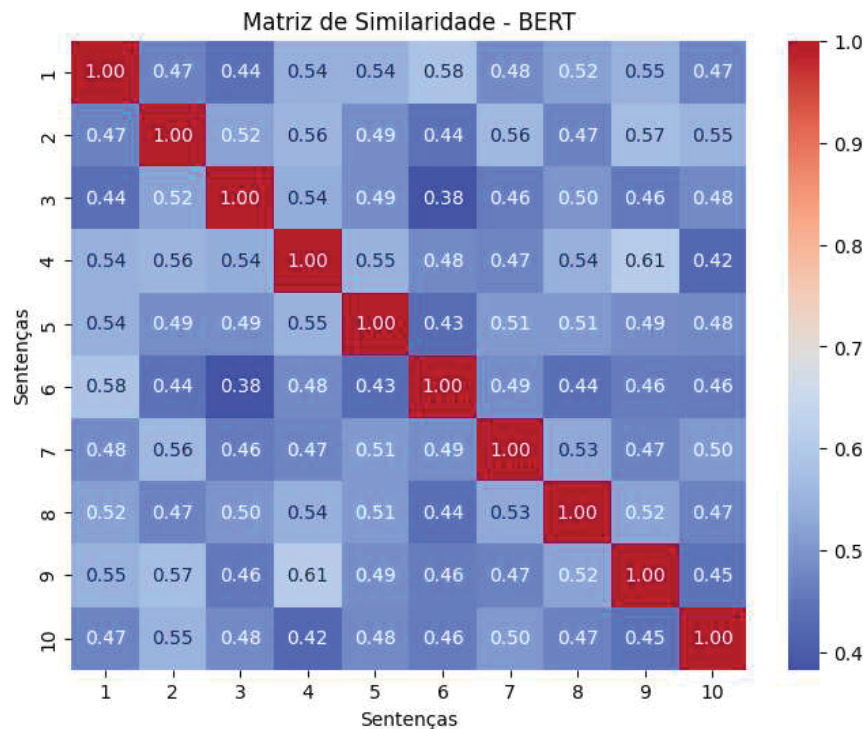
O pôr do sol na praia era deslumbrante. As ondas quebravam suavemente, criando um som relaxante. Um grupo de amigos ria enquanto jogava bola na areia. Crianças corriam perto da água, desenhando formas com os pés. O céu exibia tons vibrantes de laranja, rosa e roxo. Era o tipo de cenário que parecia saído de um filme. Mais adiante, um casal caminhava de mãos dadas, trocando sorrisos. O vento trazia o cheiro do mar e da brisa salgada. Os pássaros voavam em formação, cruzando o horizonte. Um senhor tocava violão, cantando baixinho uma melodia nostálgica. • ‘)

Sentenças selecionadas:

O pôr do sol na praia era deslumbrante.
 As ondas quebravam suavemente, criando um som relaxante.
 Um grupo de amigos ria enquanto jogava bola na areia.
 Crianças corriam perto da água, desenhando formas com os pés.
 O céu exibia tons vibrantes de laranja, rosa e roxo.
 Era o tipo de cenário que parecia saído de um filme.
 Mais adiante, um casal caminhava de mãos dadas, trocando sorrisos.
 O vento trazia o cheiro do mar e da brisa salgada.
 Os pássaros voavam em formação, cruzando o horizonte

Um senhor tocava violão, cantando baixinho uma melodia nostálgica.





+ O texto não tem pelo menos 2 sentenças similares (≥ 0.6 para TF-IDF ou ≥ 0.7 para BERT). Tente novamente.

Digite um texto com até 1000 caracteres (mínimo 6 frases com algumas parecidas):
 O gato preto pulou o muro. O gato preto saltou sobre a cerca. O cachorro marrom correu pelo jardim. O cachorro preto pulou o portão. O pássaro azul voou alto no céu. O gato e o cachorro brincaram no quintal.

) ' • Sentenças selecionadas:

O gato preto pulou o muro.

O gato preto saltou sobre a cerca.

O cachorro marrom correu pelo jardim.

O cachorro preto pulou o portão.

O pássaro azul voou alto no céu.

O gato e o cachorro brincaram no quintal.

