

UNIVERSIDADE FEDERAL DO PARANÁ

MARLON MATEUS PRUDENTE DE OLIVEIRA

MEMORIAL DE PROJETOS:
DA TEORIA DA EVOLUÇÃO À INTELIGÊNCIA ARTIFICIAL

CURITIBA

2025

MARLON MATEUS PRUDENTE DE OLIVEIRA

MEMORIAL DE PROJETOS:
DA TEORIA DA EVOLUÇÃO À INTELIGÊNCIA ARTIFICIAL

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 **MARLON MATEUS PRUDENTE DE OLIVEIRA**, intitulada: **MEMORIAL DE PROJETOS: DA TEORIA DA EVOLUÇÃO À INTELIGÊNCIA ARTIFICIAL**, 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, 26 de Agosto de 2025.



JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora



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

RESUMO

Este memorial apresenta um resumo abrangente dos projetos realizados nas disciplinas cursadas, destacando o aprendizado prático e teórico desenvolvido ao longo do período acadêmico. Cada projeto foi essencial para aprimorar conhecimentos técnicos, habilidades analíticas e abordagem criativa na resolução de problemas relacionados à Inteligência Artificial (IA), Big Data e estatísticas aplicadas. Foram 16 disciplinas ao longo do curso, todas com um trabalho em grupo ou individual ao final das aulas. Estes trabalhos evidenciam o progresso acadêmico em diversas frentes, consolidando uma base sólida de conhecimento técnico e reflexivo, aliada à capacidade de aplicação prática em diferentes contextos profissionais. Essa jornada se mostrou essencial para o preparo para desafios futuros na área de IA e tecnologia.

Palavras-chave: Inteligência Artificial; Genética; Rede Neural Artificial; Aprendizado profundo; Transformers.

ABSTRACT

This memorial presents a comprehensive summary of the projects carried out in the courses taken, highlighting the practical and theoretical learning developed throughout the academic period. Each project was essential for enhancing technical knowledge, analytical skills, and a creative approach to solving problems related to Artificial Intelligence (AI), Big Data, and applied statistics. There were 16 courses throughout the program, each concluding with a group or individual project. These works demonstrate academic progress in various areas, consolidating a solid foundation of technical and reflective knowledge, combined with the ability to apply it practically in different professional contexts. This journey has proven essential for preparing for future challenges in the field of AI and technology

Keywords: Artificial Intelligence; Genetics; Artificial Neural Network; Deep Learning; Transformers.

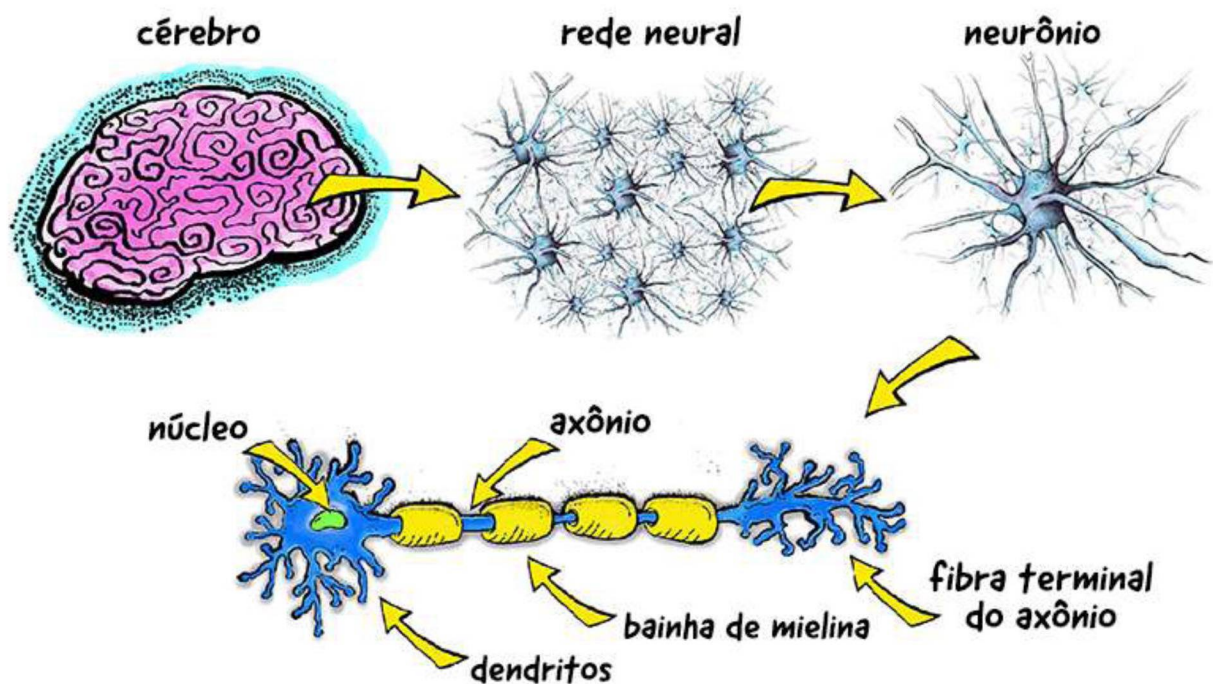
SUMÁRIO

1 PARECER TÉCNICO.....	7
REFERÊNCIAS.....	14
APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL	15
APÊNDICE 2 – LINGUAGEM DE PROGRAMAÇÃO APLICADA.....	22
APÊNDICE 3 – LINGUAGEM R	25
APÊNDICE 4 – ESTATÍSTICA APLICADA I	30
APÊNDICE 5 – ESTATÍSTICA APLICADA II	36
APÊNDICE 6 – ARQUITETURA DE DADOS	38
APÊNDICE 7 – APRENDIZADO DE MÁQUINA.....	55
APÊNDICE 8 – DEEP LEARNING	71
APÊNDICE 9 – BIG DATA.....	90
APÊNDICE 10 – VISÃO COMPUTACIONAL	95
APÊNDICE 11 – ASPECTOS FILOSÓFICOS E ÉTICOS DA IA.....	97
APÊNDICE 12 – GESTÃO DE PROJETOS DE IA.....	111
APÊNDICE 13 – FRAMEWORKS DE INTELIGÊNCIA ARTIFICIAL.....	113
APÊNDICE 14 – VISUALIZAÇÃO DE DADOS E STORYTELLING	117
APÊNDICE 15 – TÓPICOS EM INTELIGÊNCIA ARTIFICIAL	145

1 PARECER TÉCNICO

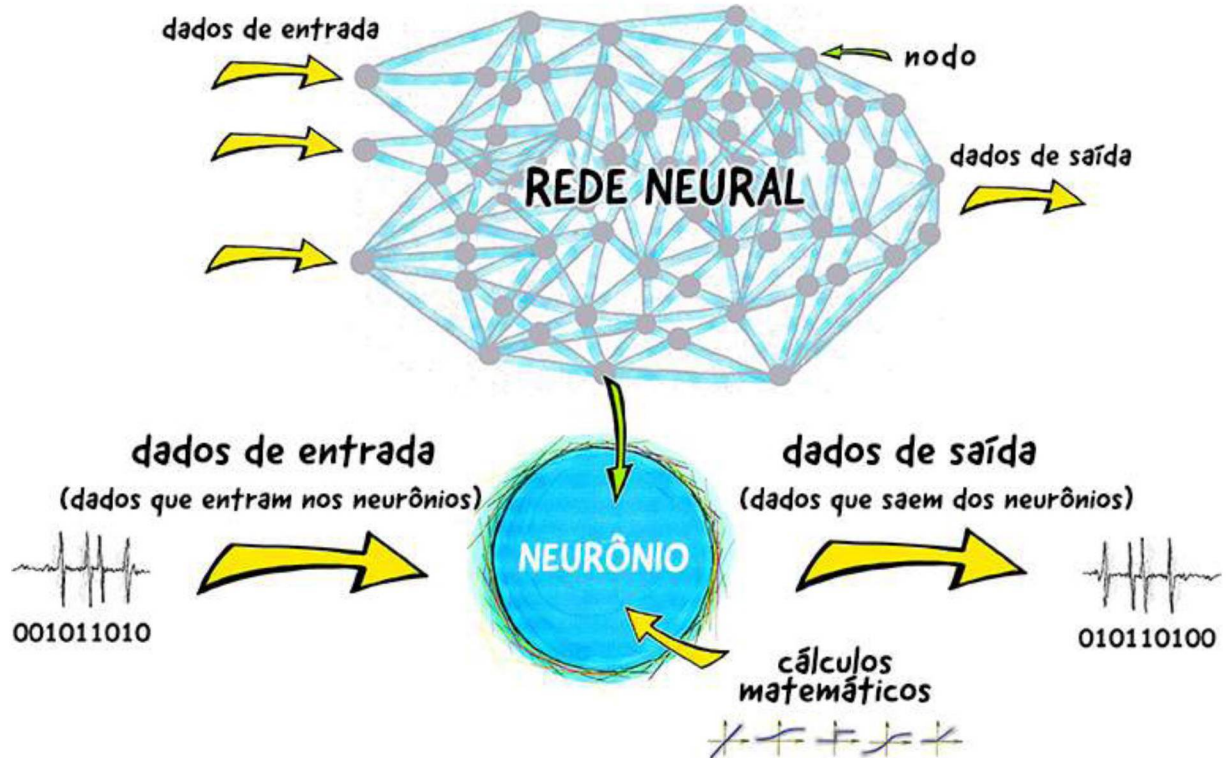
A biologia desempenhou um papel fundamental no desenvolvimento de algoritmos de inteligência artificial (IA), especialmente por meio da inspiração em processos naturais e biológicos, como por exemplo, as Redes Neurais Artificiais ilustrada na FIGURA 2, que são inspiradas no funcionamento do cérebro humano, exemplificado na FIGURA 1. Elas simulam a maneira como os neurônios se conectam e processam informações (Gerstner, 2014). Elas são amplamente utilizadas em aprendizado de máquina e reconhecimento de padrões (Gerstner; Kistler; Naud; Paninski, 2014).

FIGURA 1: Rede Neural Natural



Fonte: Pregowska e Osial (2021)

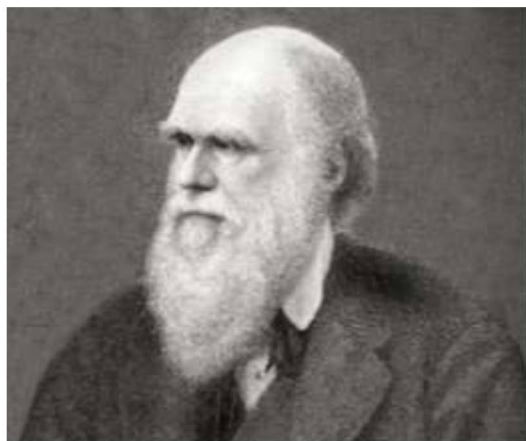
FIGURA 2: Rede Neural Artificial



Fonte: Pregowska e Osial (2021)

A teoria da evolução de Darwin influenciou significativamente o desenvolvimento de algoritmos de Inteligência Artificial (IA), especialmente por meio de conceitos como seleção natural e adaptação. Esses princípios inspiraram a criação de algoritmos evolutivos, que são usados para resolver problemas complexos de otimização e aprendizado (Pregowska; Osial, 2021).

FIGURA 3: Retrato de Charles Darwin



Fonte: Nozaki, Vasconcelos, Piteli, Longhi e Eulálio (2023)

Pode-se dizer que, inteligência artificial está completamente relacionada com a biologia, desde o desenvolvimento do Perceptron, que simula um neurônio, até os algoritmos genéticos, que simulam a teoria da evolução de Darwin (Gerstner, 2014).

Charles Darwin (FIGURA 3) desenvolveu a teoria da evolução, e John H. Holland (FIGURA 4) a aplicou em algoritmo de computadores (Holland, 1992).

FIGURA 4: Foto de John H. Holland

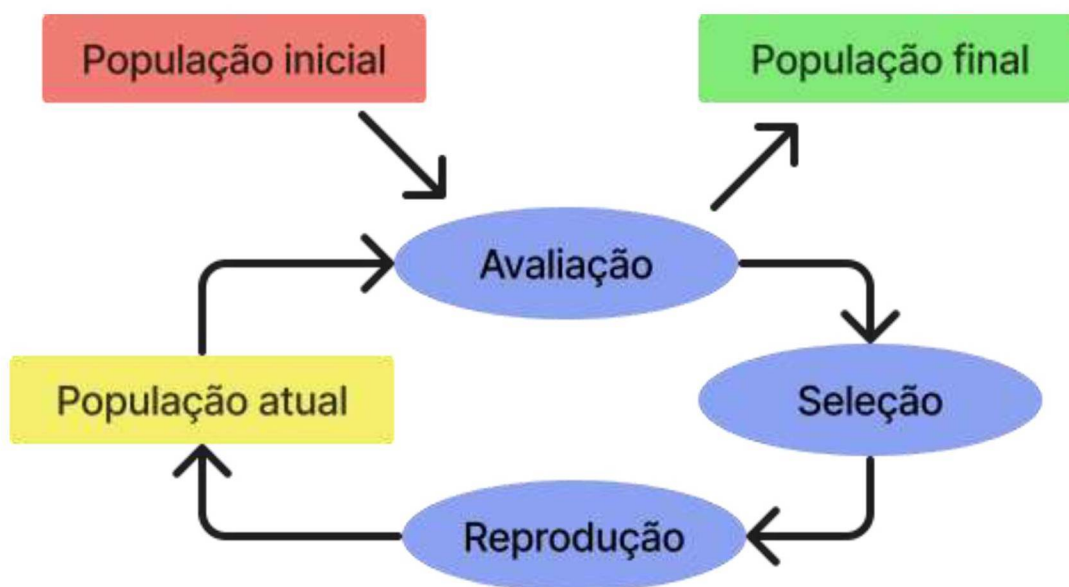


Fonte: Husbands, Holland e Wheeler (2008)

Os algoritmos evolutivos simulam processos biológicos, como reprodução, mutação e seleção, para encontrar soluções eficientes (Guizelini, 2025). Por exemplo, a seleção natural, proposta por Darwin, é aplicada para selecionar os melhores candidatos em um conjunto de soluções, permitindo que eles evoluam para resolver problemas de maneira mais eficaz. Esse é o funcionamento dos algoritmos genéticos, o fluxograma desse processo é ilustrado na FIGURA 5.

Além disso, a IA utiliza esses conceitos para criar sistemas que aprendem e se adaptam ao longo do tempo, imitando a forma como os organismos vivos evoluem para se ajustar ao ambiente. Isso é particularmente útil em áreas como aprendizado de máquina, onde os modelos precisam melhorar continuamente com base em novos dados (Gerstner, 2014).

FIGURA 5: Fluxo de um algoritmo Genético



Fonte: Nozaki, Vasconcelos, Piteli, Longhi e Eulálio (2023)

Na disciplina ministrada pelo Prof. Dr. Dieval Guizelini, foram apresentados os fundamentos dos algoritmos genéticos, o que é um cruzamento¹, uma mutação², uma seleção³ e demais termos muito utilizados em biologia e adaptados para a computação, sendo necessário fazer uma breve introdução sobre a teoria da evolução de Darwin (Guizelini, 2025).

Abordou-se o problema de alocação de horário de disciplinas ilustrado na FIGURA 6, utilizando algoritmos genéticos como solução computacional. Na aula síncrona, foi apresentado um software desenvolvido especificamente para alocar as o horário de aulas, desenvolvido pelo próprio professor da disciplina, permitindo ilustrar a aplicação do conteúdo apresentado de forma prática (Guizelini, 2025).

¹ É uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e de busca

² Operador randômico de manipulação; Introduz e mantém a variedade genética da população

³ Combina as informações genéticas de dois indivíduos (pais) para gerar novos indivíduos (filhos)

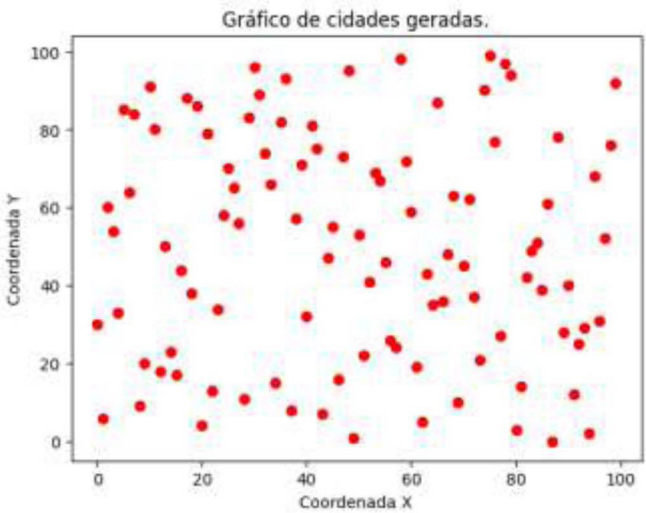
FIGURA 6 - Problema de alocação de horário



Fonte: Guizelini (2025)

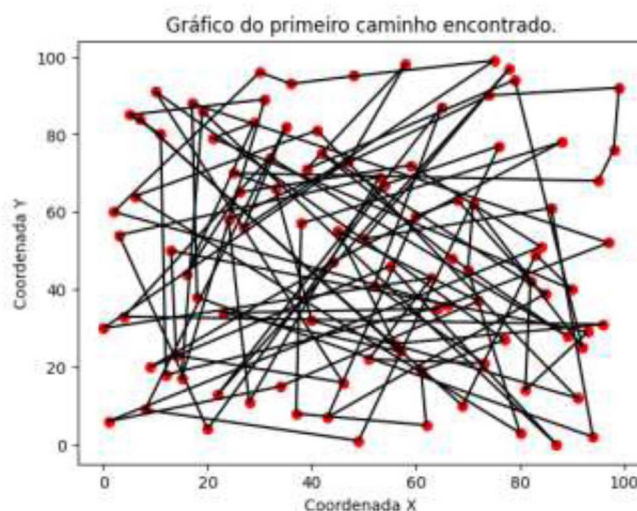
Na disciplina, foi solicitado também, o desenvolvimento de um algoritmo genético que resolva o problema do caixeiro viajante, otimizando um caminho entre 100 cidades, as cidades foram criadas de maneira aleatória, como mostra a FIGURA 7, o primeiro caminho gerado pode ser observado na FIGURA 8, e o melhor caminho encontrado, após o término do processamento do algoritmo, pode ser observado na FIGURA 9.

FIGURA 7 - Representação das cidades geradas



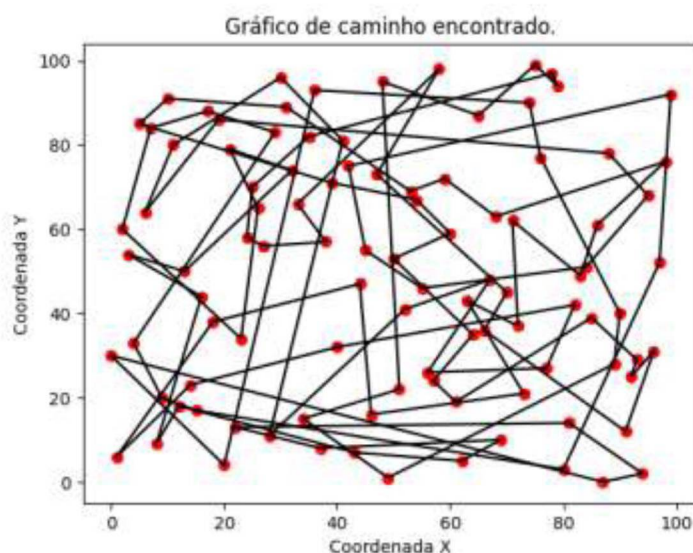
FONTE: O Autor (2025)

FIGURA 8 - Primeiro caminho gerado de maneira aleatória



FONTE: O Autor (2025)

FIGURA 9 - Melhor caminho encontrado



FONTE: O Autor (2025)

O exercício demonstrou a aplicabilidade de soluções baseadas em IA para problemas classificados como NP-Completo, encontrando, a partir de diversas iterações o melhor *indivíduo*, ou seja, o caminho com a distância mínima a percorrer.

Notavelmente, o desenvolvimento do algoritmo genético para o problema do caixeiro viajante foi realizado sem o uso de bibliotecas específicas de IA, utilizando apenas bibliotecas matemáticas.

O presente parecer técnico evidencia a relevância da interdisciplinaridade entre os campos da biologia e da inteligência artificial, destacando como fundamentos teóricos oriundos da teoria da evolução de Darwin e da neurociência têm influenciado diretamente o desenvolvimento de algoritmos computacionais avançados. A utilização de modelos bioinspirados, como redes neurais artificiais e algoritmos genéticos, demonstra não apenas a capacidade da IA em simular processos naturais, mas também sua eficácia na resolução de problemas complexos, como a alocação de horários e o problema do caixeiro viajante.

A análise dos projetos desenvolvidos ao longo da formação acadêmica revela uma sólida compreensão dos princípios teóricos e uma competência prática na aplicação de técnicas de IA em contextos diversos. Tal domínio técnico, aliado à capacidade de reflexão crítica sobre os fundamentos epistemológicos da área, contribui para a formação de profissionais aptos a enfrentar os desafios contemporâneos da ciência e da tecnologia.

Dessa forma, conclui-se que a convergência entre os saberes biológicos e computacionais constitui uma base promissora para o avanço da inteligência artificial, reafirmando seu papel como ferramenta estratégica na construção de soluções inovadoras e adaptativas, alinhadas às demandas da sociedade e aos princípios éticos e científicos que regem a pesquisa aplicada.

REFERÊNCIAS

GERSTNER, Wulfram; KISTLER, Werner M.; NAUD, Richard; PANINSKI, Liam. **Neuronal Dynamics**: from single neurons to networks and models of cognition. Uk: Cambridge University Press, 2014. 578 p.

GHOSH-DASTIDAR, Samanwoy; ADELI, Hojjat. SPIKING NEURAL NETWORKS. **International Journal Of Neural Systems**, [S.L.], v. 19, n. 04, p. 295- 308, ago. 2009. World Scientific Pub Co Pte Lt. <http://dx.doi.org/10.1142/s0129065709002002>.

GUIZELINI, Dieval. **Algoritmos Genéticos**. Curitiba, PR. UFPR. fev. 2025. Apresentação de Powerpoint. 36 slides. color. Disponível em: <https://ufprvirtual.ufpr.br/course/view.php?id=30848>. Acesso em: 21 mai. 2025.

HOLLAND, John Henry. Genetic Algorithms: computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand. **Scientific American: a division of Nature America, Inc.** [S. L.], p. 66-73. jul. 1992. Disponível em: <https://www.jstor.org/stable/24939139>. Acesso em: 06 agosto 2025.

HUSBANDS, Phil; HOLLAND, Owen; WHEELER, Michael. **The Mechanical Mind in History**. Cambridge: The MIT Press, 2008.

NOZAKI, Felipe Akira; VASCONCELOS, Vinicius Jardim; PITELI, Caio Gomes; LONGHI, Eduardo; EULÁLIO, Henrique Bexiga. **Algoritmos Genéticos**: o que abordaremos neste artigo. O que abordaremos neste artigo. 2023. Disponível em: <https://wordpress.ft.unicamp.br/revisa/introducao-aos-algoritmos-geneticos/>. Acesso em: 21 maio 2025.

PREGOWSKA, Agnieszka; OSIAL, Magdalena. What Is An Artificial Neural Network And Why Do We Need It? **Frontiers For Young Minds**: Neuroscience and Psychology, [S.L.], v. 9, n. 0, p. 1-7, 17 ago. 2021. Frontiers Media SA. <http://dx.doi.org/10.3389/frym.2021.560631>.

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.

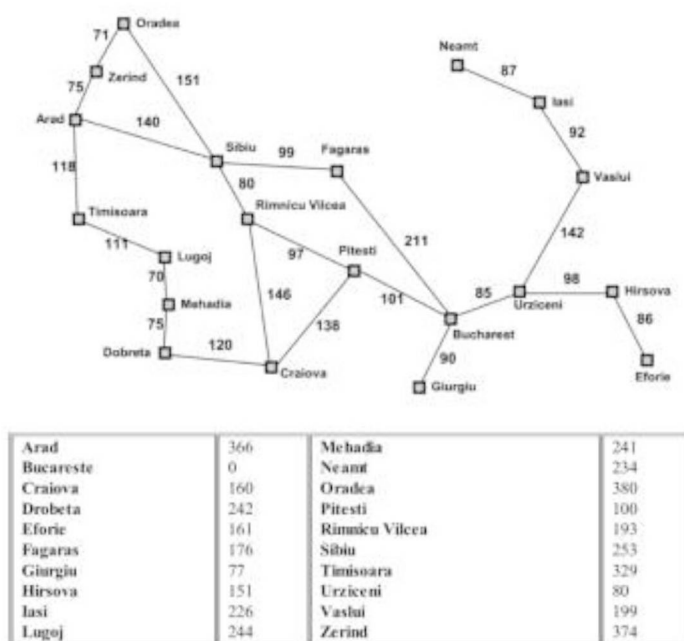


Figura 3.22 Valores de $h(n)$ — distâncias em linha reta para Bucareste.

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

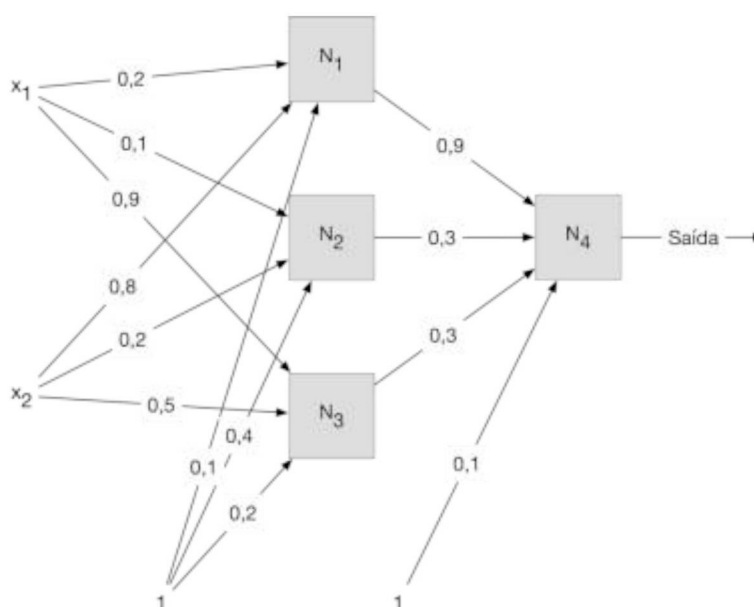
Conjunção: $\alpha, \beta \vdash \alpha \wedge \beta$

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:

- (6,25 pontos)** Valor de saída do neurônio N_1
- (6,25 pontos)** Valor de saída do neurônio N_2
- (6,25 pontos)** Valor de saída do neurônio N_3
- (6,25 pontos)** Valor de saída da rede como um todo

B – RESOLUÇÃO

1 - ChatGPT

- a) **(6,25 pontos)** Pergunte ao ChatGPT o que é Inteligência Artificial e cole aqui o resultado.

“Inteligência Artificial (IA) refere-se à capacidade de uma máquina ou software imitar a inteligência humana. Em outras palavras, é a simulação de processos de pensamento humanos por

meio de algoritmos e sistemas computacionais. A IA permite que as máquinas realizem tarefas que normalmente exigiriam inteligência humana, como aprendizado, raciocínio, reconhecimento de padrões, processamento de linguagem natural, visão computacional e tomada de decisões.

Existem dois tipos principais de IA: a estreita (ou fraca) e a geral (ou forte). A IA estreita é especializada em realizar tarefas específicas sem a capacidade de realizar atividades fora desse escopo, enquanto a IA geral busca replicar a ampla gama de habilidades cognitivas humanas.

As técnicas comuns usadas na implementação de IA incluem aprendizado de máquina (machine learning), redes neurais, algoritmos de otimização, lógica fuzzy e processamento de linguagem natural. A IA é aplicada em uma variedade de campos, como saúde, finanças, automação industrial, veículos autônomos, reconhecimento de voz, jogos, assistentes virtuais e muito mais. A pesquisa continua nesse campo visa melhorar a eficiência e a adaptabilidade dos sistemas de IA, bem como explorar novas fronteiras na inteligência artificial."

b) (6,25 pontos) Dada essa resposta do ChatGPT, classifique usando as 4 abordagens vistas em sala. Explique o porquê.

Na primeira frase do primeiro parágrafo, podemos classificar a resposta do ChatGPT como "Agir como Humanos", pois ele define que IA como imitar a inteligência humana. Já na segunda frase, podemos classificar a resposta como "Pensar como Humanos", pois ele afirma que seria a simulação do processo de pensamento humano.

No segundo parágrafo, podemos classificar a resposta como "Pensar racionalmente" quando se trata da IA estreita (ou fraca), enquanto a IA geral (ou forte) pode ser classificada como "Agir como Humano".

No terceiro parágrafo, pode-se classificar os exemplos dados de aplicações da IA como IAs que se classificam em "Agir racionalmente", pois são exemplos que foram desenvolvidos para tarefas específicas que respondem a ações e buscam o melhor resultado possível.

c) (6,25 pontos) Pesquise sobre o funcionamento do ChatGPT (sem perguntar ao próprio ChatGPT) e escreva um texto contendo no máximo 5 parágrafos. Cite as referências.

Os modelos GPT (incluindo o próprio ChatGPT), são projetados para gerar linguagem natural de maneira coerente e consistente com a linguagem humana. Esses modelos passam por um processo de pré-treinamento, onde são treinados em grandes quantidades de dados de texto não supervisionados, como páginas da web ou livros. Posteriormente, esses modelos podem ser ajustados para tarefas específicas, como classificação de texto ou resposta a perguntas.

O desenvolvimento do ChatGPT envolveu uma série de marcos e melhorias, incluindo a introdução da arquitetura "Transformer" que possibilitou a criação de modelos de linguagem altamente eficientes e escaláveis. Uma característica-chave dos Transformers é a "atenção multi-cabeça" ou "attention heads", onde o modelo pode ponderar diferentes partes do texto em relação a outras partes para capturar conexões significativas entre as palavras.

No caso do GPT-3.5, o modelo utiliza uma pilha de 13 blocos "Transformer", cada um com 12 cabeças de atenção e 768 unidades ocultas. A entrada para o modelo é uma sequência de tokens usando uma camada de incorporação. Os tokens incorporados são então alimentados no primeiro bloco Transformer, que produz uma sequência de representações ocultas.

As representações ocultas são então passadas através dos 12 blocos "Transformer" restantes, cada um dos quais aplicam suas camadas. A saída do bloco "Transformer" final é uma sequência de representações ocultas, que são decodificadas em uma sequência de saída.

Além da arquitetura principal do "Transformer", o GPT-3.5 também inclui vários componentes adicionais, como normalização de camadas, conexões residuais e incorporações posicionais. Esses componentes ajudam a estabilizar o treinamento e melhorar o desempenho do modelo em tarefas de modelagem de linguagem.

d) (6,25 pontos) Entendendo o que é o ChatGPT, classifique o próprio ChatGPT usando as 4 abordagens vistas em sala. Explique o porquê.

A partir do conhecimento adquirido acerca do funcionamento, treinamento e objetivos da OpenAI com o projeto do chatGPT, é possível observar duas abordagens: Agir como humanos e Agir Racionalmente.

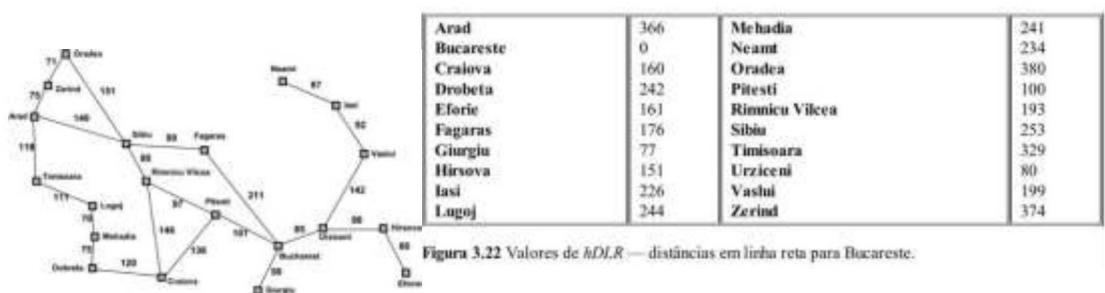
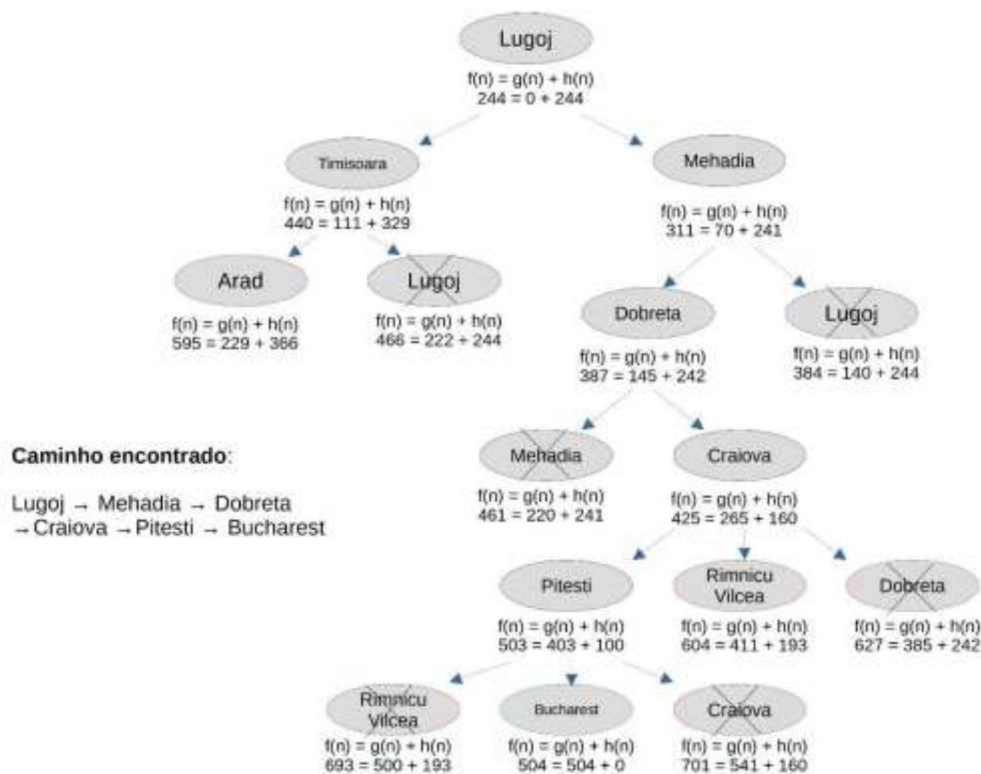
Agir como humanos ao gerar linguagem natural de forma coerente e consistente com a linguagem humana, e, Agir Racionalmente ao buscar, por meio de seus modelos, a melhor forma

possível de realizar alguma tarefa específica ou oferecer a melhor resposta à uma pergunta feita por seus usuários.

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.

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



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.

p: as uvas caem q: a raposa come as uvas

r: as uvas estão maduras

R1: $p \rightarrow q$

R2: $q \rightarrow r$

R3: $\neg r \vee p$

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.

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.

R4: $r \rightarrow p$ Equivalência condicional,
R3

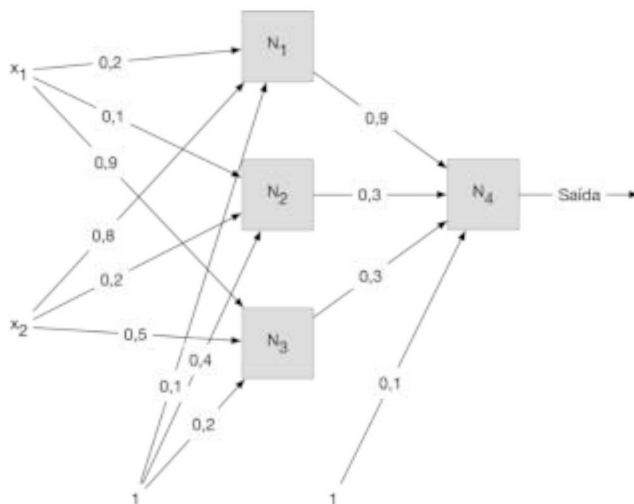
R5: $q \rightarrow p$ Silogismo Hipotético, R2,
R4

R6: $(q \rightarrow p) \wedge (p \rightarrow q)$ Conjunção, R1, R5

R7: $q \leftrightarrow p$ Equivalência
Bicondicional, R6

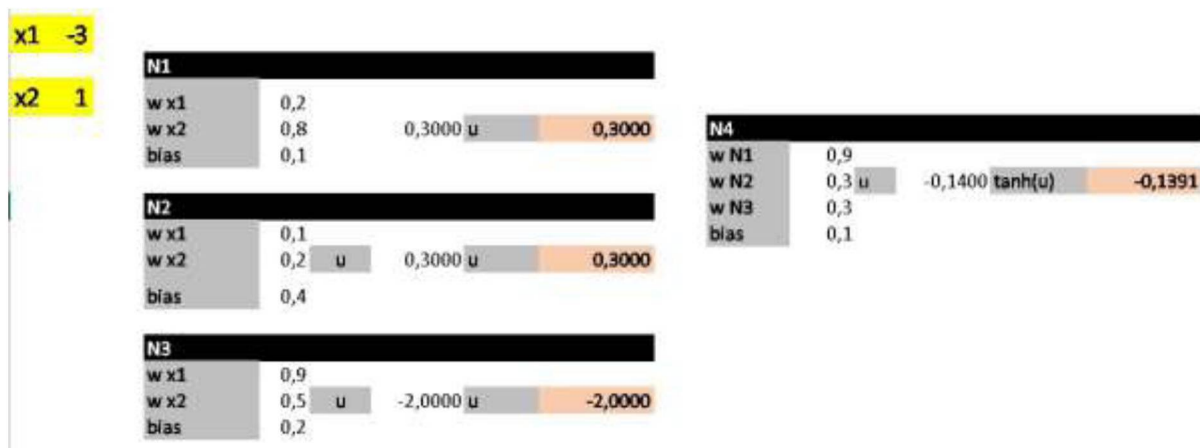
4- Redes Neurais Artificiais

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
VALOR: 0,3
- b) (6,25 pontos) Valor de saída do neurônio N2
VALOR: 0,3
- c) (6,25 pontos) Valor de saída do neurônio N3
VALOR: -2
- d) (6,25 pontos) Valor de saída da rede como um todo
VALOR: -0,1391

APÊNDICE 2 – LINGUAGEM DE PROGRAMAÇÃO APLICADA

A – ENUNCIADO

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

Informações sobre a base de dados:

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

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

Metadados:

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

year_model	Ano do modelo do carro. Pode não corresponder ao ano de fabricação
avg_price	Preço médio do carro, em reais

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

1 Análise Exploratória dos dados

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

- 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 um breve explicação (máximo de quatro linhas) sobre os principais resultados encontrados na Análise Exploratória dos dados

2 Visualização dos dados

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

- 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
- Crie partições contendo 75% dos dados para treino e 25% para teste
- 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
- Grave os valores preditos em variáveis criadas
- Realize a análise de importância das variáveis para estimar a variável target, **para cada modelo treinado**

- f. Dê uma breve explicação (máximo de quatro linhas) sobre os resultados encontrados na análise de importância de variáveis
- g. Escolha o melhor modelo com base nas métricas de avaliação MSE, MAE e R^2
- h. Dê uma breve explicação (máximo de quatro linhas) sobre qual modelo gerou o melhor resultado e a métrica de avaliação utilizada

B – RESOLUÇÃO

1 - ANÁLISE EXPLORATÓRIA DOS DADOS

f) Dê uma breve explicação (máximo de quatro linhas) sobre os principais resultados encontrados na Análise Exploratória dos dados.

Existem valores faltantes e duplicados, removemos e fizemos as análises. A contagem de registros por marca, mostrou que a Fiat possui o maior número de registros, seguida pela Volkswagen e Chevrolet. Na contagem de modelos a marca mais presentes na base aparece em outra ordem sendo Fiat, Ford e Volkswagen.

2 - VISUALIZAÇÃO DOS DADOS

e) Dê uma breve explicação (máximo de quatro linhas) sobre os resultados gerados no item d

Observou-se pouca variação na média de preços de carros automáticos por marca e manuais por marca. A exceção ocorre nos modelos da Renault, que apresentam valores significativamente inferiores tanto para os modelos automáticos quanto para os manuais. Os veículos de câmbio automático em geral apresentam maior preço médio.

g) Dê uma breve explicação (máximo de quatro linhas) sobre os resultados gerados no item f

É possível observar que veículos a diesel são mais caros. Principalmente veículos da marca VW que destaca-se significativamente em relação aos valores dos veículos movidos a gasolina e álcool. Fabricantes Renault e Nissan não oferecem veículos movidos a álcool, porém, o preço dos veículos da Renault é comparável aos da Fiat movidos a gasolina.

3 - APLICAÇÃO DE MODELOS DE MACHINE LEARNING PARA PREVER O PREÇO MÉDIO DOS CARROS

f) Dê uma breve explicação (máximo de quatro linhas) sobre os resultados encontrados na análise de importância de variáveis.

Foi observado que em ambos os algoritmos de machine learning usados as variáveis de maior importância são a potência do motor 'engine_size' e o ano do modelo. Apesar da importância das variáveis no algoritmo XGBoost ter sido menor a acurácia de ambos os modelos foi muito semelhante.

h) Dê uma breve explicação (máximo de quatro linhas) sobre qual modelo gerou o melhor resultado e a métrica de avaliação utilizada.

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:

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

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

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

Tarefas

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

- O modelo alométrico é dado por: $\text{Volume} = b_0 + b_1 * \text{dap}^2 * H_t$

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

7. Escolha o melhor modelo.

B – RESOLUÇÃO

1 Pesquisa com Dados de Satélite (Satellite)

Base de dados Satellite da biblioteca mlbench.

Treino 80% e Teste 20%

Colunas usadas no treino: x17, x18, x19, x20 e classes.

Seed = 7

Resultados obtidos

Random Forest - Acurácia: 0,8396

SVM - Acurácia: 0,8715

RNA - Acurácia: 0,6978

O melhor modelo de predição foi obtido a partir do método SVM, pois é o que apresenta maior acurácia na base de teste.

2 – Estimativa de Volume de Árvores

Base de dados utilizada disponível no link <http://www.razer.net.br/datasets/Volumes.csv>, carregamento realizado por meio da função “read.csv2” indicando os parâmetros de separador “;” e símbolo decimal “,”.

Coluna NR eliminada usando a função “subset” indicando os parâmetros do dataframe original e “-” a coluna indesejada.

Treino 80% e Teste 20%

Resultados obtidos com as funções UDF: Random

Forest

- R^2 : 0,8538019
- S_{xy} : 0,1445527
- $S_{xy}\%$: 11,07658
- RMSE: 0,1371347

SVM

- R^2 : 0,849634
- S_{xy} : 0,1470481
- $S_{xy}\%$: 11,2678
- RMSE: 0,1395021

Redes Neurais (NEURALNET)

- R^2 : 0,8825988
- S_{xy} : 0,1295415
- $S_{xy}\%$: 9,926323
- RMSE: 0,1228938 Redes Neurais (NNET)
- R^2 : 0
- S_{xy} : 0,49606
- $S_{xy}\%$: 38,01139
- RMSE: 0,4706038

Observamos que não possível executar no mesmo momento o NNET e NEURALNET. Sendo assim, para obter cada resultado foi necessário comentar as linhas de código de cada um e rodar em execuções diferentes.

Modelo alométrico de Spurr

- R^2 : 0,8271824
- S_{xy} : 0,1574296
- $S_{xy}\%$: 12,0633
- RMSE: 0,1493509

O modelo escolhido é o de Redes Neurais (NEURALNET), pois apresenta os melhores resultados em todas as métricas propostas. Caso o modelo Redes Neurais (NEURALNET) seja desconsiderado, o melhor modelo observado passa a ser o Random Forest.

APÊNDICE 4 – ESTATÍSTICA APLICADA I

A – ENUNCIADO

1) Gráficos e tabelas

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

(15 pontos) Elaborar a tabela de 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

1 – Gráficos e Tabelas

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

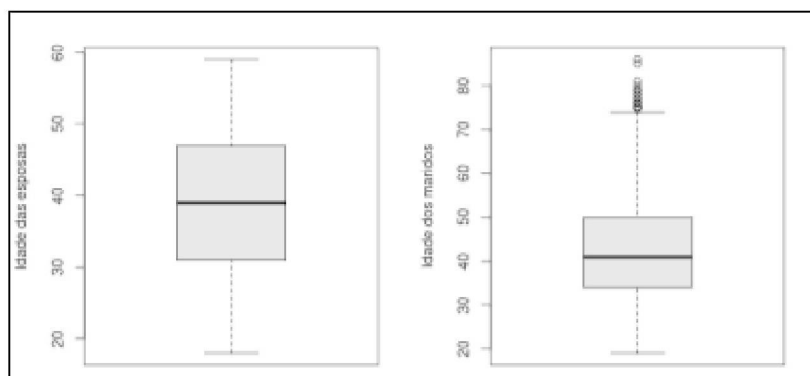


Figura 1 - Boxplot das variáveis "age" e "husage"

Ao observar os gráficos do tipo box plot, podemos observar que as idades dos maridos possuem *outliers* enquanto as idades das esposas não.

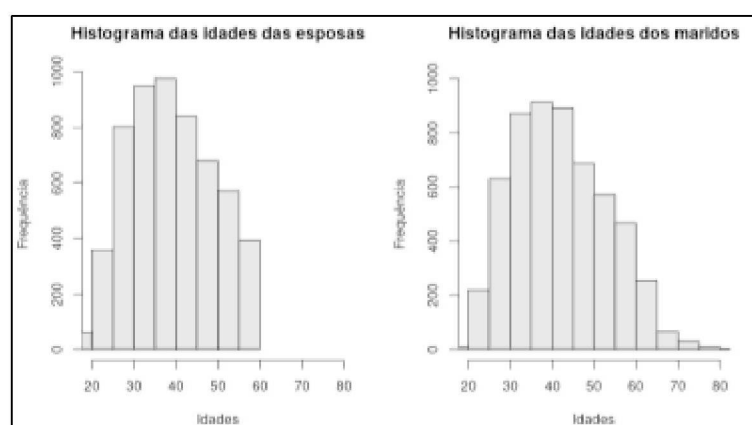


Figura 2 – Histogramas das variáveis "age" e "husage"

Observando os histogramas da frequência de idades das esposas e maridos podemos concluir que possuem diferentes limites em x.

b) Elaborar a tabela de frequências das variáveis "age" (idade da esposa) e "husage" (idade do marido) e comparar os resultados

Tabela de frequências da variável "age"					
Classes	Frequência	Frequência relativa	Frequência relativa (%)	Frequência acumulada	Frequência relativa acumulada (%)
17.82- 20.804	61	0.01	1.08	61	1.08
20.804 - 3.787	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

Tabela 1 - Tabela de frequências da variável "age"

Tabela de frequências da variável "husage"					
Classes	Frequência	Frequência relativa	Frequência relativa (%)	Frequência acumulada	Frequência rel. acumulada (%)
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

Tabela 2 - Tabela de frequência da variável "husage"

2 – Medidas de posição e dispersão

- a) Calcular a média, mediana e moda das variáveis "age" (idade da esposa) e "husage" (idade do marido) e comparar os resultados

Medida	Idade da esposa	Idade do marido
Média	39.42758	42.45296
Mediana	39	41
Moda	37 (217 ocorrências)	44 (201 ocorrências)

b) Calcular a variância, desvio padrão e coeficiente de variação das variáveis “age”

(idade da esposa) e “husage” (idade do marido) e comparar os resultados

Medida	Idade da esposa	Idade do marido
Variância	99.75234	126.0717
Desvio Padrão	9.98761	11.22817
Coeficiente de variação	25.33153	26.44849

3 – Testes paramétricos ou não paramétricos

- a) 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.

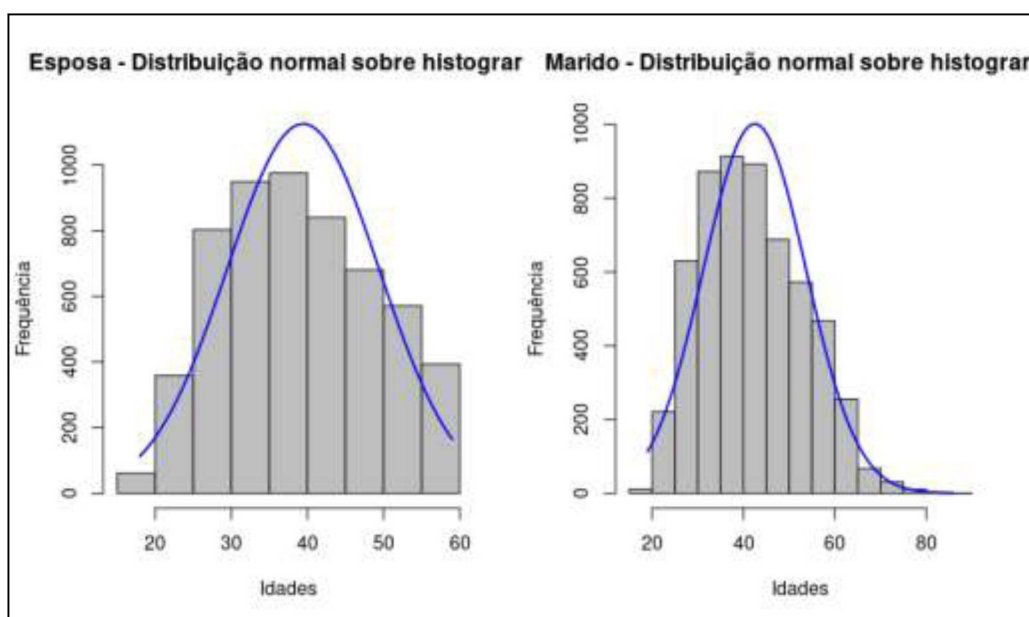


Figura 3 - Distribuição normal sobre o histograma

Para decidir se as médias serão testadas com testes paramétricos ou não paramétricos verificamos pelos métodos Kolmogorov-Smirnov e Shapiro-Wilk se a distribuição dos dados das variáveis analisadas são normais:

```
Lilliefors (Kolmogorov-Smirnov) normality test
data:  salarios$age
D = 0.058909, p-value < 0.00000000000000022
```

```
Lilliefors (Kolmogorov-Smirnov) normality test
data:  salarios$husage
D = 0.059662, p-value < 0.00000000000000022
```

```

Shapiro-Wilk normality test

data:  age[name == "WIFE"][0:5000]
W = 0.97695, p-value < 0.00000000000000022

Shapiro-Wilk normality test

data:  age[name == "HUSBAND"][0:5000]
W = 0.98151, p-value < 0.00000000000000022

```

Por ambos os testes, é possível observar que *p-value* é inferior ao nível de significância 0,05, portanto os dados a serem comparados não são normais, demandando assim a abordagem por meio de testes não-paramétricos.

H0: A idade mediana dos maridos é estatisticamente igual à mediana da idade das esposas.

H1: A idade mediana dos maridos é estatisticamente diferente à mediana da idade das esposas.

O teste usado é o U de Mann-Whitney, que gerou o resultado abaixo:

```

Wilcoxon rank sum test with continuity correction

data:  age by name
W = 18122044, p-value < 0.00000000000000022
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

```

Figura 4 - Teste U de Mann-Whitney - Estatisticamente Igual

Ao realizar o teste U de Mann-Whitney é possível verificar que as medianas das duas variáveis não são estatisticamente iguais, pois *p-value* inferior ao nível de significância 0,05, logo, rejeitamos H0. O intervalo de confiança da diferença entre as medianas está entre 2,00 e 3,00, com uma mediana de 2,99.

H0: A idade mediana dos maridos não é estatisticamente menor à mediana da idade das esposas.

H1: A idade mediana dos maridos é estatisticamente menor à mediana da idade das esposas.

```

> #testando se a mediana das idades das esposas e estatisticamente menor
> wilcox.test(age ~ name, data = allAges,
+             exact = FALSE, alternative = "less",
+             conf.int=TRUE)

Wilcoxon rank sum test with continuity correction

data:  age by name
W = 18122044, p-value = 1
alternative hypothesis: true location shift is less than 0
95 percent confidence interval:
 -Inf 3.000034
sample estimates:
difference in location
      2.999966

```

Figura 5 - Teste U de Mann-Whitney - Estatisticamente Menor

Ao testar se a mediana do marido não é estatisticamente menor que a mediana das esposas, não rejeitamos H_0 , e verificamos que a mediana da idade dos maridos é estatisticamente maior que a mediana da idade das esposas, pois p-value foi superior a 0,05, com um intervalo de confiança entre -Inf e 3, e mediana 2,99.

Para confirmar a hipótese encontrada, realiza o teste para verificar se a mediana da idade dos maridos não é estatisticamente maior que a mediana da idade das esposas, caso o p-value fique abaixo de 0,05, confirmamos que, a mediana da idade dos maridos é estatisticamente maior à mediana da idade das esposas.

H_0 : A idade mediana da idade dos maridos não é estatisticamente maior à mediana da idade das esposas.

H_1 : A idade mediana da idade dos maridos é estatisticamente maior à mediana da idade das esposas.

```

wilcoxon rank sum test with continuity correction

data: age by name
w = 18122044, p-value < 0.00000000000000022
alternative hypothesis: true location shift is greater than 0
95 percent confidence interval:
 2.000046      Inf
sample estimates:
difference in location
      2.999966

```

Figura 6 - Teste U de Mann-Whitney - Estatisticamente Maior

Rejeitamos H_0 , e confirmamos que a mediana da idade dos maridos é estatisticamente maior que a mediana da idade das esposas.

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

1. Regressões Ridge, Lasso e ElasticNet

a) Feitas as regressões, tivemos os seguintes resultados:

	RMSE	<i>R</i> ²
--	-------------	-----------------------

Treino Ridge	0.8487758	0.2792296
Teste Ridge	0.8489331	0.3002941
Treino Lasso	0.8498015	0.2774865
Teste Lasso	0.8491485	0.2999389
Treino Elasticnet	0.8499146	0.2772942
Teste Elasticnet	0.8497128	0.2990082

Observamos que todas as regressões tiveram valores parecidos, mas mesmo assim escolhemos o padrão Ridge por apresentar a menor diferença dos erros entre o treino e teste e melhores valores nos erros se comparados com as outras regressões.

O resultado da predição usando o Ridge para os seguintes valores passados no exercício e depois de aplicar o antilog são:

Resultado da predição: 8.346

Intervalos de confiança: Entre 8.324 e 8.369

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

1- Identificador automático de idioma

Problema: Dados um texto de entrada, é possível identificar em qual língua o texto está escrito?

Entrada: "texto qualquer"

Saída: português ou inglês ou francês ou italiano ou...

O processo de Reconhecimento de Padrões

O objetivo desse trabalho é demonstrar o processo de "construção de atributos" e como ele é fundamental para o **Reconhecimento de Padrões (RP)**.

Primeiro um conjunto de "amostras" previamente conhecido (classificado)

```
#
# amostras de texto em diferentes línguas
#
ingles = [
    "Hello, how are you?",
    "I love to read books.",
    "The weather is nice today.",
    "Where is the nearest restaurant?",
    "What time is it?",
    "I enjoy playing soccer.",
    "Can you help me with this?",
    "I'm going to the movies tonight.",
    "This is a beautiful place.",
    "I like listening to music.",
    "Do you speak English?",
    "What is your favorite color?",
    "I'm learning to play the guitar.",
    "Have a great day!",
    "I need to buy some groceries.",
    "Let's go for a walk.",
    "How was your weekend?",
    "I'm excited for the concert.",
    "Could you pass me the salt, please?",
    "I have a meeting at 2 PM.",
    "I'm planning a vacation.",
    "She sings beautifully.",
    "The cat is sleeping.",
    "I want to learn French.",
    "I enjoy going to the beach.",
    "Where can I find a taxi?",
    "I'm sorry for the inconvenience.",
    "I'm studying for my exams.",
    "I like to cook dinner at home.",
    "Do you have any recommendations for restaurants?",
]

espanhol = [
    "Hola, ¿cómo estás?",
    "Me encanta leer libros.",
    "El clima está agradable hoy.",
    "¿Dónde está el restaurante más cercano?",
    "¿Qué hora es?",
    "Voy al parque todos los días.",
    "¿Puedes ayudarme con esto?",
    "Me gustaría ir de vacaciones.",
    "Este es mi libro favorito.",
    "Me gusta bailar salsa.",
    "¿Hablas español?",
```



```

"¿Cuál es tu comida favorita?",
"Estoy aprendiendo a tocar el piano.",
"¡Que tengas un buen día!",
"Necesito comprar algunas frutas.",
"Vamos a dar un paseo.",
"¿Cómo estuvo tu fin de semana?",
"Estoy emocionado por el concierto.",
"¿Me pasas la sal, por favor?",
"Tengo una reunión a las 2 PM.",
"Estoy planeando unas vacaciones.",
"Ella canta hermosamente.",
"El perro está jugando.",
"Quiero aprender italiano.",
"Disfruto ir a la playa.",
"¿Dónde puedo encontrar un taxi?",
"Lamento las molestias.",
"Estoy estudiando para mis exámenes.",
"Me gusta cocinar la cena en casa.",
"¿Tienes alguna recomendación de restaurantes?",
]

```

```

portugues = [
"Estou indo para o trabalho agora.",
"Adoro passar tempo com minha família.",
"Preciso comprar leite e pão.",
"Vamos ao cinema no sábado.",
"Gosto de praticar esportes ao ar livre.",
"O trânsito está terrível hoje.",
"A comida estava deliciosa!",
"Você já visitou o Rio de Janeiro?",
"Tenho uma reunião importante amanhã.",
"A festa começa às 20h.",
"Estou cansado depois de um longo dia de trabalho.",
"Vamos fazer um churrasco no final de semana.",
"O livro que estou lendo é muito interessante.",
"Estou aprendendo a cozinhar pratos novos.",
"Preciso fazer exercícios físicos regularmente.",
"Vou viajar para o exterior nas férias.",
"Você gosta de dançar?",
"Hoje é meu aniversário!",
"Gosto de ouvir música clássica.",
"Estou estudando para o vestibular.",
"Meu time de futebol favorito ganhou o jogo.",
"Quero aprender a tocar violão.",
"Vamos fazer uma viagem de carro.",
"O parque fica cheio aos finais de semana.",
"O filme que assisti ontem foi ótimo.",
"Preciso resolver esse problema o mais rápido possível.",
"Adoro explorar novos lugares.",
"Vou visitar meus avós no domingo.",
"Estou ansioso para as férias de verão.",
"Gosto de fazer caminhadas na natureza.",
"O restaurante tem uma vista incrível.",
"Vamos sair para jantar no sábado.",
]

```


As “amostras” de texto precisa ser “transformada” em **padrões**

Um padrão é um conjunto de características, geralmente representado por um vetor e um conjunto de padrões no formato de tabela. Onde cada linha é um padrão e as colunas as características e, geralmente, na última coluna a **classe**

```
import random

pre_padroes = []
for frase in ingles:
    pre_padroes.append( [frase, 'inglês'])

for frase in espanhol:
    pre_padroes.append( [frase, 'espanhol'])

for frase in portugueses:
    pre_padroes.append( [frase, 'português'])

random.shuffle(pre_padroes)
print(pre_padroes)
```

```
[['Estou aprendendo a cozinhar pratos novos.', 'português'], ['Vamos fazer
um churrasco no final de semana.', 'português'], ['Tenho uma reunião import
ante amanhã.', 'português'], ['Me encanta leer libros.', 'espanhol'], ['¿Qu
é hora es?', 'espanhol'], ["I'm studying for my exams.", 'inglês'], ['Gosto
de ouvir música clássica.', 'português'], ['Hola, ¿cómo estás?', 'espanhol'
], ['Preciso comprar leite e pão.', 'português'], ['I enjoy going to the be
ach.', 'inglês'], ['Hello, how are you?', 'inglês'], ['Do you speak English
?', 'inglês'], ['Adoro explorar novos lugares.', 'português'], ['A comida e
stava deliciosa!', 'português'], ['Vamos fazer uma viagem de carro.', 'port
uguês'], ['Me gusta bailar salsa.', 'espanhol'], ['O filme que assisti onte
m foi ótimo.', 'português'], ['Adoro passar tempo com minha família.', 'por
tuguês'], ['Preciso resolver esse problema o mais rápido possível.', 'portu
guês'], ['Necesito comprar algunas frutas.', 'espanhol'], ['¿Tienes alguna
recomendación de restaurantes?', 'espanhol'], ['Você já visitou o Rio de Ja
neiro?', 'português'], ['I need to buy some groceries.', 'inglês'], ['How w
as your weekend?', 'inglês'], ["I'm sorry for the inconvenience.", 'inglês'
], ['Lamento las molestias.', 'espanhol'], ['I enjoy playing soccer.', 'ing
lês'], ["Let's go for a walk.", 'inglês'], ['¿Dónde puedo encontrar un taxi
?', 'espanhol'], ['A festa começa às 20h.', 'português'], ['Estou indo para
o trabalho agora.', 'português'], ['Estou ansioso para as férias de verão.'
, 'português'], ['Preciso fazer exercícios físicos regularmente.', 'portugu
ês'], ['O parque fica cheio aos finais de semana.', 'português'], ["I'm exc
ited for the concert.", 'inglês'], ['¿Que tengas un buen día!', 'espanhol'
], ['Where is the nearest restaurant?', 'inglês'], ['The cat is sleeping.',
'inglês'], ['Vamos a dar un paseo.', 'espanhol'], ["I'm going to the movies
tonight.", 'inglês'], ['Vamos ao cinema no sábado.', 'português'], ['Quiero
aprender italiano.', 'espanhol'], ['Quero aprender a tocar violão.', 'portu
guês'], ['I like listening to music.', 'inglês'], ['Me gusta cocinar la cen
a en casa.', 'espanhol'], ['Have a great day!', 'inglês'], ['Do you have an
y recommendations for restaurants?', 'inglês'], ['Estou cansado depois de u
m longo dia de trabalho.', 'português'], ['El perro está jugando.', 'espanh
ol'], ['Gosto de praticar esportes ao ar livre.', 'português'], ['Estoy emo
cionado por el concierto.', 'espanhol'], ['I want to learn French.', 'inglê
s'], ['O trânsito está terrível hoje.', 'português'], ['Gosto de fazer cami
nhadas na natureza.', 'português'], ['Can you help me with this?', 'inglês'
], ['I like to cook dinner at home.', 'inglês'], ['Voy al parque todos los
días.', 'espanhol'], ["I'm learning to play the guitar.", 'inglês'], ['Teng
```

o una reunión a las 2 PM.', 'espanhol'], ['The weather is nice today.', 'inglês'], ['¿Cuál es tu comida favorita?', 'espanhol'], ['Hoje é meu aniversário!', 'português'], ['I love to read books.', 'inglês'], ['What is your favorite color?', 'inglês'], ['She sings beautifully.', 'inglês'], ['Vamos sair para jantar no sábado.', 'português'], ['O livro que estou lendo é muito interessante.', 'português'], ['Este es mi libro favorito.', 'espanhol'], ['¿Puedes ayudarme con esto?', 'espanhol'], ["I'm planning a vacation.", 'inglês'], ['Vou visitar meus avós no domingo.', 'português'], ['Disfruto ir a la playa.', 'espanhol'], ['Estou estudando para o vestibular.', 'português'], ['Vou viajar para o exterior nas férias.', 'português'], ['Me gustaría ir de vacaciones.', 'espanhol'], ['I have a meeting at 2 PM.', 'inglês'], ['Could you pass me the salt, please?', 'inglês'], ['Estoy estudiando para mis exámenes.', 'espanhol'], ['Estoy aprendiendo a tocar el piano.', 'espanhol'], ['This is a beautiful place.', 'inglês'], ['¿Cómo estuvo tu fin de semana?', 'espanhol'], ['Meu time de futebol favorito ganhou o jogo.', 'português'], ['Estoy planeando unas vacaciones.', 'espanhol'], ['¿Hablas español?', 'espanhol'], ['El clima está agradable hoy.', 'espanhol'], ['¿Dónde está el restaurante más cercano?', 'espanhol'], ['Você gosta de dançar?', 'português'], ['Ella canta hermosamente.', 'espanhol'], ['¿Me pasas la sal, por favor?', 'espanhol'], ['O restaurante tem uma vista incrível.', 'português'], ['What time is it?', 'inglês'], ['Where can I find a taxi?', 'inglês']]

O DataFrame do pandas facilita a visualização.

```
import pandas as pd
dados = pd.DataFrame(pre_padroes)
dados
```

	0	1
0	Estou aprendendo a cozinhar pratos novos.	português
1	Vamos fazer um churrasco no final de semana.	português
2	Tenho uma reunião importante amanhã.	português
3	Me encanta leer libros.	espanhol
4	¿Qué hora es?	espanhol
...	...	...
87	Ella canta hermosamente.	espanhol
88	¿Me pasas la sal, por favor?	espanhol

0 1

89 O restaurante tem uma vista incrível. português

90 What time is it? inglês

91 Where can I find a taxi? inglês

92 rows × 2 columns

Construção dos atributos

Esse é o coração desse trabalho e que deverá ser desenvolvido por vocês. Pensem em como podemos "medir" cada frase/sentença e extrair características que melhorem o resultado do processo de identificação.

Após a criação de cada novo atributo, execute as etapas seguintes e registre as métricas da matriz de confusão. Principalmente acurácia e a precisão.

```
# a entrada é o vetor pre_padroes e a saída desse passo deverá ser
"padrões"
import re

def tamanhoMedioFrases(texto):
    palavras = re.split("\s", texto)
    #print(palavras)
    tamanhos = [len(s) for s in palavras if len(s)>0]
    #print(tamanhos)
    soma = 0
    for t in tamanhos:
        soma=soma+t
    return soma / len(tamanhos)

def contar_consoantes(texto):
    # retorna a quantidade de consoantes em uma frase
    padrao_consoantes = r'[bcdfghjklmnpqrstvwxyzBCDFGHJKLMNOPQRSTUVWXYZ]'
    #padrão regex dos elementos que serão contados

    consoantes = re.findall(padrao_consoantes, texto)

    return len(consoantes)

def contar_acentuados(texto):

    #retorna o numero de caracteres com acentos gráficos
    padrao_acentos = r'[âéíóúâêîôûâêîôûãõäëïöüñçÁÉÍÓÚÀÈÌÒÙÂÊÎÔÛÃÕÄËÏÖÜÇÑ]'
    #padrão regex dos elementos que serão contados

    acentos = re.findall(padrao_acentos, texto)

    return len(acentos)

def contar_dupla_vogal(texto):
    padrao_duplas_vogais = r'(?i)[aeiouy]{2}'
    duplas_vogais = re.findall(padrao_duplas_vogais, texto)
```

```

    return len(duplas_vogais)

def contar_termino_palavras(texto):
    padrao_terminos = r'\b\w*(?:m|an|ón|ing)\b'
    terminos = re.findall(padrao_terminos, texto, re.IGNORECASE)
    return len(terminos)

def contar_pontuacao(texto):
    padrao_pontuacao = r'(|;|!|)'
    pontuacao = re.findall(padrao_pontuacao, texto)

    return len(pontuacao)

def caracteristicas_ingles(texto):
    padrao_ingles = r'(\'|wh|th|ing|oo|ee|nn)'
    pd_ingles = re.findall(padrao_ingles, texto)

    return len(pd_ingles)

def caracteristicas_espanhol(texto):
    padrao_espanhol = r'\w(?:ç|el|la|ue|ie|ón|an|oy|ay|ñ)'
    pd_espanhol = re.findall(padrao_espanhol, texto)

    return len(pd_espanhol)

def caracteristicas_portugues(texto):
    padrao_portugues = r'\w(?:nh|lh|rr|ss|am)'
    pd_portugues = re.findall(padrao_portugues, texto)

    return len(pd_portugues)

def extraiCaracteristicas(frase):
    # frase é um vetor [ 'texto', 'lingua' ]
    texto = frase[0]
    pattern_regex = re.compile('[^\w+]', re.UNICODE)
    texto = re.sub(pattern_regex, ' ', texto)
    #print(texto)
    caracteristica1=tamanhoMedioFrases(texto)
    caracteristica2=contar_consoantes(texto)
    caracteristica3=contar_acentuados(texto)
    caracteristica4=caracteristicas_ingles(texto)
    caracteristica5=caracteristicas_espanhol(texto)
    caracteristica6=caracteristicas_portugues(texto)
    # acrescente as suas funcoes no vetor padrao
    padrao = [caracteristica1, caracteristica2, caracteristica3,
    caracteristica4,
                caracteristica5,caracteristica6, frase[1] ]
    return padrao

def geraPadroes(frases):
    padroes = []
    for frase in frases:
        padrao = extraiCaracteristicas(frase)
        padroes.append(padrao)
    return padroes

# converte o formato [frase classe] em
# [caracteristica_1, caracteristica_2,... caracteristica n, classe]
padroes = geraPadroes(pre_padroes)

#

```



```
# apenas para visualizacao
print(padroes)
```

```
dados = pd.DataFrame(padroes)
dados
```

```
[[5.833333333333333, 20, 0, 0, 0, 1, 'português'], [4.5, 21, 0, 0, 1, 2, 'p
ortuguês'], [6.2, 15, 2, 0, 2, 2, 'português'], [4.75, 11, 0, 1, 1, 0, 'esp
anhol'], [3.0, 4, 1, 0, 0, 0, 'espanhol'], [3.3333333333333335, 14, 0, 1, 0
, 1, 'inglês'], [5.2, 14, 2, 0, 0, 1, 'português'], [4.333333333333333, 7,
2, 0, 1, 0, 'espanhol'], [4.6, 12, 1, 0, 0, 0, 'português'], [3.5, 12, 0, 2
, 1, 0, 'inglês'], [3.5, 7, 0, 0, 1, 0, 'inglês'], [4.25, 10, 0, 0, 0, 0, '
inglês'], [6.25, 14, 0, 0, 0, 0, 'português'], [5.5, 10, 0, 0, 1, 0, 'portu
guês'], [4.333333333333333, 14, 0, 0, 0, 2, 'português'], [4.5, 10, 0, 0, 1
, 0, 'espanhol'], [4.142857142857143, 14, 1, 0, 1, 1, 'português'], [5.1666
66666666667, 17, 1, 0, 0, 3, 'português'], [5.75, 26, 2, 0, 1, 2, 'portuguê
s'], [7.0, 17, 0, 0, 0, 0, 'espanhol'], [7.8, 21, 1, 0, 3, 0, 'espanhol'],
[3.7142857142857144, 11, 2, 0, 1, 0, 'português'], [3.8333333333333335, 12,
0, 1, 1, 0, 'inglês'], [4.25, 10, 0, 1, 0, 0, 'inglês'], [4.333333333333333
, 16, 0, 1, 1, 1, 'inglês'], [6.333333333333333, 11, 0, 0, 0, 1, 'espanhol'
], [4.75, 12, 0, 1, 2, 0, 'inglês'], [2.3333333333333335, 9, 0, 0, 0, 0, 'i
nglês'], [5.0, 14, 1, 0, 2, 0, 'espanhol'], [3.4, 7, 2, 0, 0, 0, 'português
'], [4.5, 13, 0, 0, 0, 1, 'português'], [4.428571428571429, 14, 2, 0, 0, 0,
'português'], [8.2, 23, 2, 0, 1, 0, 'português'], [4.125, 15, 0, 0, 2, 0, '
português'], [3.6666666666666665, 14, 0, 1, 0, 0, 'inglês'], [3.6, 9, 1, 0,
2, 0, 'espanhol'], [5.4, 16, 0, 1, 1, 0, 'inglês'], [4.0, 10, 0, 2, 0, 0, '
inglês'], [3.2, 8, 0, 0, 0, 1, 'espanhol'], [3.5714285714285716, 15, 0, 2,
1, 0, 'inglês'], [4.2, 10, 1, 0, 0, 1, 'português'], [7.333333333333333, 10
, 0, 0, 2, 0, 'espanhol'], [5.0, 12, 1, 0, 1, 0, 'português'], [4.2, 12, 0,
1, 0, 0, 'inglês'], [3.7142857142857144, 14, 0, 0, 0, 0, 'espanhol'], [3.25
, 7, 0, 0, 1, 0, 'inglês'], [5.857142857142857, 24, 0, 0, 1, 0, 'inglês'],
[4.4444444444444445, 21, 0, 0, 1, 1, 'português'], [4.5, 10, 1, 0, 1, 1, 'es
panhol'], [4.571428571428571, 18, 0, 0, 0, 0, 'português'], [5.8, 15, 0, 0,
2, 0, 'espanhol'], [3.6, 12, 0, 0, 1, 0, 'inglês'], [5.0, 14, 3, 0, 1, 1, '
português'], [5.333333333333333, 18, 0, 0, 0, 2, 'português'], [3.333333333
3333335, 13, 0, 2, 2, 0, 'inglês'], [3.2857142857142856, 12, 0, 2, 0, 0, 'i
nglês'], [3.8333333333333335, 13, 1, 0, 2, 0, 'espanhol'], [3.5714285714285
716, 15, 0, 2, 1, 0, 'inglês'], [3.142857142857143, 11, 1, 0, 1, 0, 'espanh
ol'], [4.2, 12, 0, 1, 1, 0, 'inglês'], [4.4, 11, 1, 0, 0, 0, 'espanhol'], [
4.75, 8, 2, 0, 0, 0, 'português'], [3.2, 8, 0, 1, 0, 0, 'inglês'], [4.6, 13
, 0, 0, 0, 0, 'inglês'], [6.333333333333333, 12, 0, 1, 0, 0, 'inglês'], [4.
5, 15, 1, 0, 1, 1, 'português'], [4.625, 18, 1, 0, 2, 1, 'português'], [4.2
, 11, 0, 0, 0, 0, 'espanhol'], [5.25, 11, 0, 0, 1, 0, 'espanhol'], [3.8, 11
, 0, 2, 1, 0, 'inglês'], [4.5, 14, 1, 1, 0, 0, 'português'], [3.6, 10, 0, 0
, 1, 0, 'espanhol'], [5.8, 15, 0, 0, 2, 0, 'português'], [4.428571428571429
, 15, 1, 0, 0, 0, 'português'], [4.8, 12, 1, 0, 0, 0, 'espanhol'], [2.57142
85714285716, 9, 0, 2, 0, 0, 'inglês'], [3.857142857142857, 16, 0, 1, 0, 1,
'inglês'], [6.0, 16, 1, 0, 2, 0, 'espanhol'], [4.833333333333333, 15, 0, 0,
3, 0, 'espanhol'], [4.2, 11, 0, 0, 1, 0, 'inglês'], [3.8333333333333335, 12
, 1, 0, 1, 0, 'espanhol'], [4.375, 17, 0, 0, 1, 1, 'português'], [7.0, 15,
0, 0, 3, 0, 'espanhol'], [6.5, 7, 1, 0, 2, 0, 'espanhol'], [4.6, 13, 1, 0,
1, 0, 'espanhol'], [5.333333333333333, 18, 3, 0, 3, 0, 'espanhol'], [4.25,
9, 2, 0, 1, 0, 'português'], [7.0, 12, 0, 0, 2, 1, 'espanhol'], [3.33333333
33333335, 12, 0, 0, 0, 0, 'espanhol'], [5.166666666666667, 17, 1, 0, 2, 0,
'português'], [3.0, 7, 0, 0, 0, 0, 'inglês'], [3.0, 10, 0, 0, 1, 0, 'inglês
']]
```


Com os conjuntos separados, podemos "treinar" o modelo usando a SVM.

```
from sklearn import svm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

treinador = svm.SVC() #algoritmo escolhido
modelo = treinador.fit(X_train, y_train)

#
# score com os dados de treinamento
acuracia = modelo.score(X_train, y_train)
print("Acurácia nos dados de treinamento: {:.2f}%".format(acuracia * 100))

#
# melhor avaliar com a matriz de confusão
y_pred = modelo.predict(X_train)
cm = confusion_matrix(y_train, y_pred)
print(cm)
print(classification_report(y_train, y_pred))
#
# com dados de teste que não foram usados no treinamento
print('métricas mais confiáveis')
y_pred2 = modelo.predict(X_test)
cm = confusion_matrix(y_test, y_pred2)
print(cm)
print(classification_report(y_test, y_pred2))
```

Acurácia nos dados de treinamento: 71.01%

```
[[19  2  2]
 [ 5 14  3]
 [ 8  0 16]]
```

	precision	recall	f1-score	support
espanhol	0.59	0.83	0.69	23
inglês	0.88	0.64	0.74	22
português	0.76	0.67	0.71	24
accuracy			0.71	69
macro avg	0.74	0.71	0.71	69
weighted avg	0.74	0.71	0.71	69

métricas mais confiáveis

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

	precision	recall	f1-score	support
espanhol	0.40	0.57	0.47	7
inglês	0.80	0.50	0.62	8
português	0.62	0.62	0.62	8
accuracy			0.57	23
macro avg	0.61	0.57	0.57	23
weighted avg	0.62	0.57	0.57	23

2 - Melhorar o desempenho de RP em conjunto de dados existentes

A atividade 02 visa trabalhar com um conjunto de dados pré-construído, onde as opções que o desenvolvedor tem, são de aplicar as técnicas de pré-processamento abaixo relacionadas:

- Seleção
- Limpeza
- Codificação
- Enriquecimento
- Normalização
- Construção de Atributos
- Correção de Prevalência
- Partição do Conjunto de Dados

Busque uma base de dados na UCI Machine Learning que seja indicada para problemas de classificação. (<https://archive.ics.uci.edu/datasets>)

Para esse exemplo, vamos usar a base Secondary Mushroom (<https://archive.ics.uci.edu/static/public/848/data.csv>)

Opção 01 - carregando o arquivo de dados da pasta local para o colab.

```
from sklearn.preprocessing import scale
from sklearn.preprocessing import minmax_scale
import pandas as pd

from sklearn import svm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

from sklearn.model_selection import train_test_split
import numpy as np
url = "https://archive.ics.uci.edu/static/public/848/data.csv"
colunas = ["class", "cap-diameter", "cap-shape", "cap-surface", "cap-color", "does-bruise-or-bleed",
           "gill-attachment", "gill-spacing", "gill-color", "stem-height",
           "stem-width", "stem-root",
           "stem-surface", "stem-color", "veil-type", "veil-color",
           "has-ring", "ring-type",
           "spore-print-color", "habitat", "season"]

colunas_categoricas = [
    "cap-shape", "cap-surface", "cap-color",
    "does-bruise-or-bleed",
    "gill-attachment", "gill-spacing", "gill-color", "stem-root",
    "stem-surface", "stem-color", "veil-type", "veil-color",
    "has-ring", "ring-type",
    "spore-print-color", "habitat", "season"]

colunas_numericas = ["cap-diameter", "stem-height", "stem-width"]

sm = pd.read_csv(url, header=None, low_memory=False, names=colunas)
sm = sm.drop(index=0)

print(type(sm))
print(sm.head())
<class 'pandas.core.frame.DataFrame'>
```

```

class cap-diameter cap-shape cap-surface cap-color does-bruise-or-bleed
\
1      p          15.26          x          g          o          f
2      p          16.6           x          g          o          f
3      p          14.07          x          g          o          f
4      p          14.17          f          h          e          f
5      p          14.64          x          h          o          f

gill-attachment gill-spacing gill-color stem-height ... stem-root \
1              e          NaN          w          16.95 ...      s
2              e          NaN          w          17.99 ...      s
3              e          NaN          w          17.8   ...      s
4              e          NaN          w          15.77 ...      s
5              e          NaN          w          16.53 ...      s

stem-surface stem-color veil-type veil-color has-ring ring-type \
1              y          w          u          w          t          g
2              y          w          u          w          t          g
3              y          w          u          w          t          g
4              y          w          u          w          t          p
5              y          w          u          w          t          p

spore-print-color habitat season
1              NaN          d          w
2              NaN          d          u
3              NaN          d          w
4              NaN          d          w
5              NaN          d          w
[5 rows x 21 columns]

```

Hora de realizar os tratamentos

no exemplo, iremos normalizar as colunas, remover a coluna de identificação e separar a classe dos atributos.

```

sm_dummies = pd.get_dummies(sm[colunas_categoricas])
X = pd.concat([sm[colunas_numericas], sm_dummies], axis=1)

```

```

#X = sm.iloc[:,1:]
cols = sm[0:]
print(X.head())
Y = sm['class']
Y_orig = sm['class']
print(Y.unique())

```

	cap-diameter	stem-height	stem-width	cap-shape_b	cap-shape_c	cap-shape_
f \						
1	15.26	16.95	17.09	False	False	Fals
e						
2	16.6	17.99	18.19	False	False	Fals
e						
3	14.07	17.8	17.74	False	False	Fals
e						
4	14.17	15.77	15.98	False	False	Tru
e						
5	14.64	16.53	17.2	False	False	Fals
e						

	cap-shape_o	cap-shape_p	cap-shape_s	cap-shape_x	...	habitat_h	\
1	False	False	False	True	...	False	
2	False	False	False	True	...	False	
3	False	False	False	True	...	False	
4	False	False	False	False	...	False	
5	False	False	False	True	...	False	

	habitat_l	habitat_m	habitat_p	habitat_u	habitat_w	season_a	season_
1	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	Fals
5	False	False	False	False	False	False	Fals

	season_u	season_w
1	False	True
2	True	False
3	False	True
4	False	True
5	False	True

[5 rows x 119 columns]

['p' 'e']

Na próxima seção que deverão ser realizada as tentativas de tratamento de dados, visando a melhoria no desempenho do classificador (SVM).

```
X_orig = X.copy()
print(X_orig.head())
```

```
print(Y_orig.unique() )
```

```
# normalização min-max
```

```
X = pd.DataFrame( minmax_scale(X) )
```

```
#retirada de dados faltantes
```

```
X.dropna(axis = 1, how ='any')
```

```
print(X_orig.head())
```

```
print(X.head())
```

	cap-diameter	stem-height	stem-width	cap-shape_b	cap-shape_c	cap-shape_
1	15.26	16.95	17.09	False	False	Fals
2	16.6	17.99	18.19	False	False	Fals
3	14.07	17.8	17.74	False	False	Fals
4	14.17	15.77	15.98	False	False	Tru

```

5      14.64      16.53      17.2      False      False      Fals
e

      cap-shape_o  cap-shape_p  cap-shape_s  cap-shape_x  ...  habitat_h  \
1      False      False      False      True      ...      False
2      False      False      False      True      ...      False
3      False      False      False      True      ...      False
4      False      False      False      False     ...      False
5      False      False      False      True      ...      False

      habitat_l  habitat_m  habitat_p  habitat_u  habitat_w  season_a  season_
s  \
1      False      False      False      False      False      False      Fals
e
2      False      False      False      False      False      False      Fals
e
3      False      False      False      False      False      False      Fals
e
4      False      False      False      False      False      False      Fals
e
5      False      False      False      False      False      False      Fals
e

      season_u  season_w
1      False      True
2      True      False
3      False      True
4      False      True
5      False      True

[5 rows x 119 columns]
['p' 'e']
      cap-diameter  stem-height  stem-width  cap-shape_b  cap-shape_c  cap-shape_
f  \
1      15.26      16.95      17.09      False      False      Fals
e
2      16.6      17.99      18.19      False      False      Fals
e
3      14.07      17.8      17.74      False      False      Fals
e
4      14.17      15.77      15.98      False      False      Tru
e
5      14.64      16.53      17.2      False      False      Fals
e

      cap-shape_o  cap-shape_p  cap-shape_s  cap-shape_x  ...  habitat_h  \
1      False      False      False      True      ...      False
2      False      False      False      True      ...      False
3      False      False      False      True      ...      False
4      False      False      False      False     ...      False
5      False      False      False      True      ...      False

      habitat_l  habitat_m  habitat_p  habitat_u  habitat_w  season_a  season_
s  \
1      False      False      False      False      False      False      Fals
e

```

```

2      False      False      False      False      False      False      False      Fals
e
3      False      False      False      False      False      False      False      Fals
e
4      False      False      False      False      False      False      False      Fals
e
5      False      False      False      False      False      False      False      Fals
e

```

```

      season_u  season_w
1      False      True
2      True      False
3      False      True
4      False      True
5      False      True

```

```
[5 rows x 119 columns]
```

```

      0      1      2      3      4      5      6      7      8      9      ...    10
9  \
0  0.240155  0.499705  0.164469  0.0  0.0  0.0  0.0  0.0  0.0  1.0  ...  0.
0
1  0.261782  0.530366  0.175055  0.0  0.0  0.0  0.0  0.0  0.0  1.0  ...  0.
0
2  0.220949  0.524764  0.170725  0.0  0.0  0.0  0.0  0.0  0.0  1.0  ...  0.
0
3  0.222563  0.464917  0.153787  0.0  0.0  1.0  0.0  0.0  0.0  0.0  ...  0.
0
4  0.230148  0.487323  0.165528  0.0  0.0  0.0  0.0  0.0  0.0  1.0  ...  0.
0

```

```

      110  111  112  113  114  115  116  117  118
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
1  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0  0.0
2  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
3  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
4  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0

```

```
[5 rows x 119 columns]
```

A próxima seção trata da construção do modelo, dos testes e das métricas da matriz de confusão.

```

# com os dados originais
X_oring_train, X_orig_test, y_orig_train, y_orig_test =
train_test_split(X_orig,
                  Y_orig, test_size=0.25,
                  stratify=Y_orig, random_state=10)

# com os dados tratados
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25,
                                                    stratify=Y, random_state=10)

```

Treina o modelo com base nos dados originais (SVM).

```

treinador = svm.SVC() #algoritmo escolhido

modelo_orig = treinador.fit(X_oring_train, y_orig_train)

```

```

# score com os dados de treinamento
acuracia_orig = modelo_orig.score(X_oring_train, y_orig_train)
print("Acurácia nos dados de treinamento ORIGINAIS:
{:.2f}%".format(acuracia_orig * 100))

# predição com os mesmos dados usados para treinar
y_orig_pred = modelo_orig.predict(X_oring_train)
cm_orig_train = confusion_matrix(y_orig_train, y_orig_pred)
print('Matriz de confusão - com os dados ORIGINAIS usados no TREINAMENTO')
print(cm_orig_train)
print(classification_report(y_orig_train, y_orig_pred))

# predição com os mesmos dados usados para testar
print('Matriz de confusão - com os dados ORIGINAIS usados para TESTES')
y2_orig_pred = modelo_orig.predict(X_orig_test)
cm_orig_test = confusion_matrix(y_orig_test, y2_orig_pred)
print(cm_orig_test)
print(classification_report(y_orig_test, y2_orig_pred))
Acurácia nos dados de treinamento ORIGINAIS: 95.03%
Matriz de confusão - com os dados ORIGINAIS usados no TREINAMENTO
[[19080  1305]
 [   970 24446]]

```

	precision	recall	f1-score	support
e	0.95	0.94	0.94	20385
p	0.95	0.96	0.96	25416
accuracy			0.95	45801
macro avg	0.95	0.95	0.95	45801
weighted avg	0.95	0.95	0.95	45801

```

Matriz de confusão - com os dados ORIGINAIS usados para TESTES
[[6274  522]
 [ 327 8145]]

```

	precision	recall	f1-score	support
e	0.95	0.92	0.94	6796
p	0.94	0.96	0.95	8472
accuracy			0.94	15268
macro avg	0.95	0.94	0.94	15268
weighted avg	0.94	0.94	0.94	15268

Como os dados ficam após os processos de tratamento dos dados?

```

from sklearn import svm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

treinador = svm.SVC() #algoritmo escolhido

modelo = treinador.fit(X_train, y_train)

# score com os dados de treinamento
acuracia = modelo.score(X_train, y_train)
print("Acurácia nos dados de treinamento TRATADOS: {:.2f}%".format(acuracia
* 100))

```



```
# predição com os mesmos dados usados para treinar
y_pred = modelo.predict(X_train)
cm_train = confusion_matrix(y_train, y_pred)
print('Matriz de confusão - com os dados TRATADOS usados no TREINAMENTO')
print(cm_train)
print(classification_report(y_train, y_pred))
```

```
# predição com os mesmos dados usados para testar
print('Matriz de confusão - com os dados TRATADOS usados para TESTES')
y2_pred = modelo.predict(X_test)
cm_test = confusion_matrix(y_test, y2_pred)
print(cm_test)
print(classification_report(y_test, y2_pred))
```

Acurácia nos dados de treinamento TRATADOS: 99.96%

Matriz de confusão - com os dados TRATADOS usados no TREINAMENTO

```
[[20368    17]
 [     3 25413]]
```

	precision	recall	f1-score	support
e	1.00	1.00	1.00	20385
p	1.00	1.00	1.00	25416
accuracy			1.00	45801
macro avg	1.00	1.00	1.00	45801
weighted avg	1.00	1.00	1.00	45801

Matriz de confusão - com os dados TRATADOS usados para TESTES

```
[[6795     1]
 [     3 8469]]
```

	precision	recall	f1-score	support
e	1.00	1.00	1.00	6796
p	1.00	1.00	1.00	8472
accuracy			1.00	15268
macro avg	1.00	1.00	1.00	15268
weighted avg	1.00	1.00	1.00	15268

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

Para o experimento de Classificação:

- Ordenar pela Acurácia (descendente), ou seja, a técnica de melhor acurácia ficará em primeiro na tabela.
- Após o quadro colocar:
 - Um resultado com 3 linhas com a predição de novos casos para a técnica/parâmetro de maior Acurácia (criar um arquivo com novos casos à sua escolha)
 - A lista de comandos emitidos no RStudio para conseguir os resultados obtidos

• Veículo

Técnica	Parâmetro	Acurácia	Matriz de Confusão
SVM – CV	C=50 Sigma=0,021	0,8802	<pre> Reference Prediction bus opel saab van bus 43 0 0 0 opel 0 35 12 0 saab 0 6 30 0 van 0 1 1 39 </pre>
SVM – Hold-out	C = 100 Sigma = 0,021	0,8743	<pre> Reference Prediction bus opel saab van bus 43 0 0 0 opel 0 34 11 1 saab 0 8 31 0 van 0 0 1 38 </pre>
RF – CV	mtry=18	0,7784	<pre> Reference prediction bus opel saab van bus 42 0 3 0 opel 0 27 17 0 saab 1 13 23 1 van 0 2 2 36 </pre>
RNA – CV	size=05 decay=0,1	0,7725	<pre> Reference prediction bus opel saab van bus 39 1 3 0 opel 1 32 20 1 saab 2 9 20 0 van 3 0 0 38 </pre>

RF – Hold-out	mtry=02	0,7605	<pre> Reference Prediction bus opel saab van bus 41 0 3 0 opel 0 27 14 0 saab 0 17 24 1 van 0 3 2 38 </pre>
---------------	---------	--------	---

RNA – Hold-out	size=05 decay=0,1	0,7006	<pre> Reference Prediction bus opel saab van bus 42 2 1 0 opel 1 35 40 0 saab 0 2 1 1 van 0 2 1 35 </pre>
KNN	k=01	0,6467	<pre> Reference Prediction bus opel saab van bus 37 4 8 3 opel 2 18 13 0 saab 3 18 18 1 van 1 2 2 35 </pre>

Novos casos:

	Co mp	Circ	DCir c	Rad Ra	PrA xisR a	Max LRa	Scat Ra	Elon g	PrA xisR ect	Max LRe ct	ScV arM axis	ScV arm axis	RaG yr	Ske wM axis	Ske wm axis	Kurt max is	Kurt Max is	Holl Ra	pred ict_s vm
1	76	45	69	306	112	42	174	59	24	182	222	301	200	96	6	1	204	198	van
2	100	54	106	288	118	17	181	48	28	167	209	971	210	65	2	39	178	198	opel
3	103	51	75	283	95	10	182	50	24	140	132	519	146	83	11	7	179	205	bus

Lista de comandos:

```

#carregamento da base
dados <- read.csv("databases/6 - Veiculos - Dados.csv")
head(dados) dados$a <- NULL

#separando em bases de treino e teste set.seed(2034)
indice <- createDataPartition(dados$tipo, p = 0.8, list =
FALSE)
treino <- dados[indice, ] teste <-
dados[-indice, ]

#treinando o modelo usando svm com crossvalidarion e mostrando as metricas
set.seed(2034)
ctrl <- trainControl(method = "cv", number = 10)
rf <- train(tipo ~ ., data = treino, method = "rf", trControl = ctrl)
rf

#fazendo a predicaõ na base de teste predito_rf <-
predict(rf, teste)

#apresentando a matriz de confusãõ
confusionMatrix(predito_rf, as.factor(teste$tipo))

```

DIABETES

Técnica	Parâmetro	Acurácia	Matriz de Confusão												
RNA – Hold-out	size=03 decay=0,1	0,817	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>88</td><td>16</td></tr><tr><td>pos</td><td>12</td><td>37</td></tr></table>			Reference	Prediction	neg	pos	neg	88	16	pos	12	37
		Reference													
Prediction	neg	pos													
neg	88	16													
pos	12	37													
SVM – Hold-out	C=0,25 Sigma=0,1090209	0,7908	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>93</td><td>25</td></tr><tr><td>pos</td><td>7</td><td>28</td></tr></table>			Reference	Prediction	neg	pos	neg	93	25	pos	7	28
		Reference													
Prediction	neg	pos													
neg	93	25													
pos	7	28													
SVM – CV	C=0,25 Sigma=0,1090209	0,7908	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>93</td><td>25</td></tr><tr><td>pos</td><td>7</td><td>28</td></tr></table>			Reference	Prediction	neg	pos	neg	93	25	pos	7	28
		Reference													
Prediction	neg	pos													
neg	93	25													
pos	7	28													
RNA – CV	size=05 decay=0,1	0,7843	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>91</td><td>24</td></tr><tr><td>pos</td><td>9</td><td>29</td></tr></table>			Reference	Prediction	neg	pos	neg	91	24	pos	9	29
		Reference													
Prediction	neg	pos													
neg	91	24													
pos	9	29													
RF – CV	mtry=02	0,7843	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>85</td><td>18</td></tr><tr><td>pos</td><td>15</td><td>35</td></tr></table>			Reference	Prediction	neg	pos	neg	85	18	pos	15	35
		Reference													
Prediction	neg	pos													
neg	85	18													
pos	15	35													
RF – Hold-out	mtry=02	0,7712	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>85</td><td>20</td></tr><tr><td>pos</td><td>15</td><td>33</td></tr></table>			Reference	Prediction	neg	pos	neg	85	20	pos	15	33
		Reference													
Prediction	neg	pos													
neg	85	20													
pos	15	33													
KNN	k=10	0,7451	<table><tr><td></td><td></td><td>Reference</td></tr><tr><td>Prediction</td><td>neg</td><td>pos</td></tr><tr><td>neg</td><td>88</td><td>27</td></tr><tr><td>pos</td><td>12</td><td>26</td></tr></table>			Reference	Prediction	neg	pos	neg	88	27	pos	12	26
		Reference													
Prediction	neg	pos													
neg	88	27													
pos	12	26													

Novos casos:

num	preg0nt	glucose	pressure	triceps	insulin	mass	pedigree	age	diabetes	predict.rna
1	16	148	16	86	756	202	2025	24	?	pos
2	15	197	81	25	80	665	1351	30	?	pos
3	16	128	4	50	134	321	1738	36	?	pos

Lista de comandos:

```
##Leitura da database
setwd("D:/GIT/IAA008-AprendizadoMaquina/")

dados <- read.csv("databases/10 - Diabetes - Dados.csv", header
= T)
dados$num <- NULL

#View(dados)

#(Ano atual com 4 dígitos + 2 algarismos do dígito verificador do CPF de
um dos integrantes) set.seed(2034)

ind <- createDataPartition(dados$diabetes, p=0.80, list=FALSE) treino <-
dados[ind,] teste <- dados[-ind,]

##executa o RNA com esse grid

rna <- train(diabetes~., data = treino, method =
"nnet", trace=FALSE) rna

##Aplica o modelo no arquivo de teste predict.rna <-
predict(rna, teste)

##mostra as métricas
confusionMatrix(predict.rna, as.factor(teste$diabetes))
```

• REGRESSÃO

Para o experimento de Regressão:

- Ordenar por R2 descendente, ou seja, a técnica de melhor R2 ficará em primeiro na tabela.
- Após o quadro, colocar:
 - Um resultado com 3 linhas com a predição de novos casos para a técnica/parâmetro de maior R2 (criar um arquivo com novos casos à sua escolha)
 - O Gráfico de Resíduos para a técnica/parâmetro de maior R2 o A lista de comandos emitidos no RStudio para conseguir os resultados obtidos

• Admissão

Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
RF – CV	mtry=02	0,8139	0,0646	0,9064	0,0623	0,0429
RF – Hold-out	mtry=02	0,8134	0,0647	0,9057	0,0624	0,0424
SVM – CV	C=01 Sigma=0,1599276	0,8108	0,0652	0,9046	0,0628	0,0451

SVM – Hold-out	C=0,5 Sigma=0,1599276	0,8034	0,0664	0,9032	0,064	0,0452
RNA – Hold-out	size=05 decay=0,1	0,76	0,073	0,89	0,07	0,054
RNA – CV	size=05 decay=0,1	0,76	0,073	0,89	0,07	0,054
KNN	K=10	0,7337	0,0773	0,8578	0,0745	0,0531

Novos Casos:

num	GRE.Score	TOEFL.Score	University.Rating	SOP	LOR	CGPA	Research	ChanceOfAdmit	predict.rf
1	338	96	1	43	3	865	0	?	0,7
2	299	108	4	36	37	704	0	?	0,58
3	324	109	2	4	42	6	0	?	0,66

Lista de comandos:

```
dados <- read.csv("databases/9 - Admissao - Dados.csv", header
= T)
dados$num <- NULL

#View(dados)
#(Ano atual com 4 dígitos + 2 algarismos do dígito verificador do CPF de
um dos integrantes) set.seed(2034)

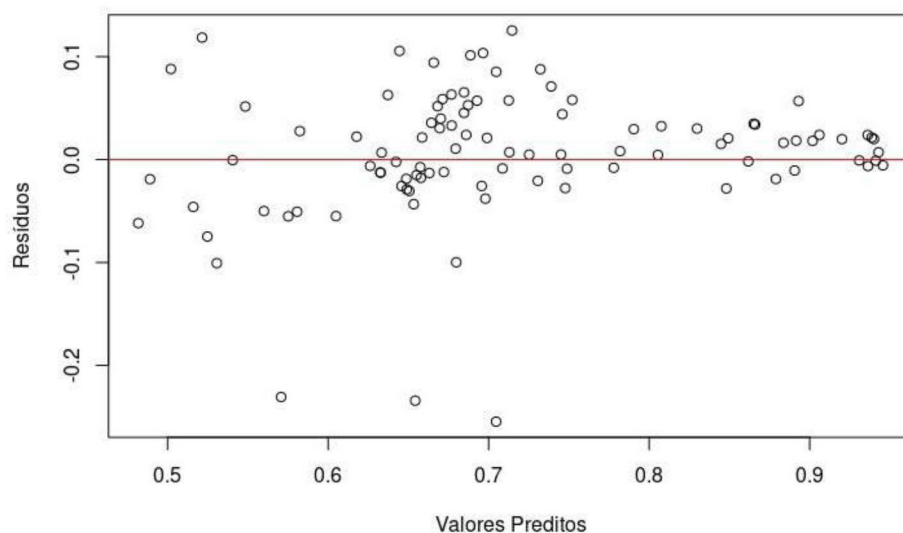
ind <- createDataPartition(dados$ChanceOfAdmit, p=0.80,
list=FALSE) treino <- dados[ind,] teste <- dados[-ind,]

##Cross-Validation
control <- trainControl(method='cv', number = 10)
##executa o RF com esse grid
rf <- train(ChanceOfAdmit~., data = treino, method = "rf", trControl =
control)
rf

##Aplica o modelo no arquivo de teste predict.rf <-
predict(rf, teste)

##mostra as métricas
rmse(teste$ChanceOfAdmit, predict.rf) mae(teste$ChanceOfAdmit,
predict.rf)
cor(teste$ChanceOfAdmit, predict.rf, method = "pearson")
Syx <- function(predito, observado, p) {
return(sqrt(sum((observado - predito)^2) /
(length(observado) - p)))
}
Syx(predict.rf, teste$ChanceOfAdmit, ncol(teste) - 1) r2 <- function(predito,
observado){ return (1 - (sum((predito-observado)^2)/sum((observado-
mean(observado))^2)))}
r2(predict.rf, teste$ChanceOfAdmit)
```

Gráfico de Resíduos - modelo RF com Crossvalidation



- Biomassa**

Técnica	Parâmetro	R2	Syx	Pearson	Rmse	MAE
SVM – Holdout	C=100 Sigma=0,021	0,9622214	481,2785	0,9981336	469,0922	127,2661
SVM – CV	C=100 Sigma=0,021	0,9622214	481,2785	0,9981336	469,0922	127,2661
RNA – Holdout	size=05 decay=0,1	0,71	1324	0,97	1290	227
RNA – CV	size=05 decay=0,1	0,71	1324	0,97	1290	227
RF – Hold-out	mtry=02	0,6397223	1486,252	0,9529809	1448,619	241,0877
RF – CV	mtry=03	0,6501568	1464,571	0,9574702	1427,487	238,9709
KNN	K=03	0,6579974	1448,066	0,959401	1411,4	233,0013

Novos casos				
dap	h	Me	biomassa	Predict_svm
12.1	20.5	0.467	?	75.95294
7.2	12.3	0.514	?	22.16095
19.1	14.9	0.617	?	170.43377

Lista de comandos:

```
#carregamento da base
dados <- read.csv("databases/5 - Biomassa - Dados.csv") head(dados)

#separando em bases de treino e teste set.seed(2034)
indice <- createDataPartition(dados$biomassa, p = 0.8, list =
FALSE)
treino <- dados[indice, ] teste <-
dados[-indice, ]

#funcoes das metricas

syx <- function(observado, predito, p) {
  return(sqrt(sum((observado - predito)^2) / (length(observado)
- p)))
}
r2 <- function(observado, predito){
  return(1 - (sum((predito - observado)^2) / sum((observado -
mean(observado))^2)))
}
metricas <- function(observado, predito, p) {
  #calculando as metricas
  r2_valor <- r2(observado, predito) syx_valor <-
syx(observado, predito, p)
  pearson_valor <- cor(observado, predito, method = "pearson")
  rmse_valor <- rmse(observado, predito) mae_valor <- mae(observado,
predito)

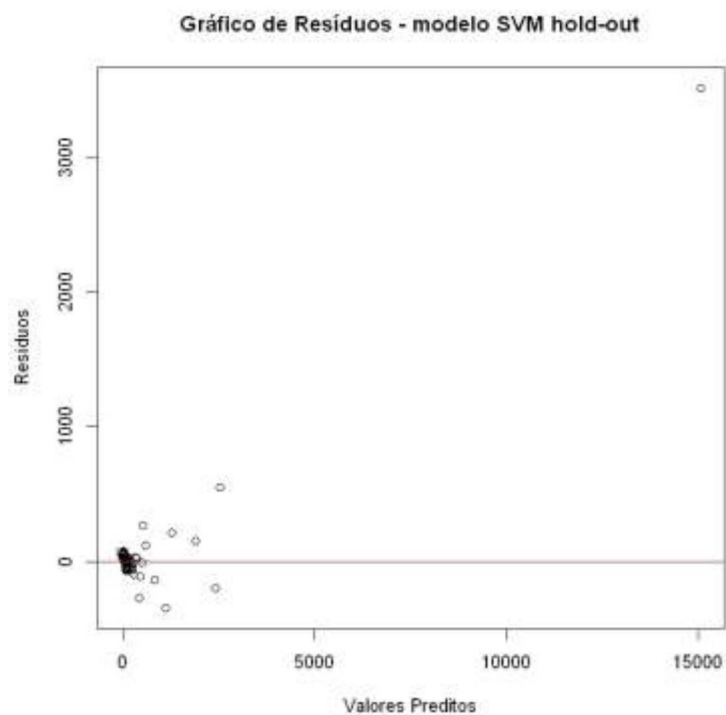
  #apresentando as métricas
  cat("R²:", r2_valor, "\n")
  cat("Syx:", syx_valor, "\n") cat("Pearson:",
pearson_valor, "\n") cat("RMSE:", rmse_valor,
"\n") cat("MAE:", mae_valor, "\n")
}

#treinando o modelo usando svm com hold-out e mostrando as metricas
set.seed(2034)
tune_grid <- expand.grid(C = c(1, 2, 10, 50, 100), sigma = c(.021, .015,
0.2))

svm1 <- train(biomassa ~ ., data = treino, method =
"svmRadial", tuneGrid = tune_grid) svm1

#fazendo a predicaõ na base de teste predito_svm1 <-
predict(svm1, teste)

#apresentando as metricas
metricas(teste$biomassa, predito_svm1, ncol(teste) - 1)
```



- **AGRUPAMENTO**
- **Veículo**

Lista de Clusters gerados:

K-means clustering with 10 clusters of sizes 56, 24, 78, 112, 71, 135, 47, 77, 184, 62

Cluster means:

	Comp	Circ	DCirc	RadRa	PrAxisRa	MaxLRa	ScatRa	Elong	PrAxisRect	MaxLRect	ScVarMaxis	ScVarmaxis		
	RaGyr	SkewMaxis	Skewmaxis	Kurtmaxis	KurtMaxis	HollRa								
1	102	49	99	206	64	9.2	199	33		23	156	217	596	
198		69		6.1	13.6		192	199						
2	107	56	103	190	56	5.8	249	27		27	168	271	911	
247		84		6.5	14.2		183	184						
3	104	53	103	197	61	10.2	212	31		24	165	224	664	
212		71		7.2	16.4		188	197						
4	90	46	79	154	62	9.1	156	43		20	152	175	359	
180		73		6.6	9.1		187	195						
5	94	43	83	187	64	7.5	171	38		21	142	193	443	
164		68		5.9	14.7		195	201						
6	88	38	62	130	57	6.4	128	53		18	132	149	243	
139		73		6.6	11.5		188	193						
7	98	47	93	199	64	8.6	186	35		22	151	207	523	
182		68		7.0	12.7		194	200						
8	104	54	103	207	63	10.6	221	30		25	171	229	717	212
	72		7.9	16.0		188	198							
9	87	42	71	144	60	7.4	146	46		19	141	168	318	
163		75		5.9	10.4		187	192						
10	92	39	77	187	71	12.8	155	42		19	135	187	363	143
73		4.3		14.9	194	201								

10 primeiras linhas do arquivo com o cluster correspondente.

Cluster	bus	opel	saab	van
1	10	24	22	0
2	24	0	0	0
3	5	33	40	0
4	28	11	9	64
5	22	22	27	0
6	0	27	26	82
7	10	18	19	0
8	6	41	30	0
9	91	25	25	43
10	22	11	19	10

Usa 10 clusters no experimento.

Colocar a lista de comandos emitidos no RStudio para conseguir os resultados obtidos

```
### Pacotes necessários library("caret")
library(Metrics)

##Leitura da database
setwd("D:/GIT/IAA008-AprendizadoMaquina/")

dados <- read.csv("databases/6 - Veiculos - Dados.csv", header
= T)
dados$a <- NULL

View(dados)

#(Ano atual com 4 dígitos + 2 algarismos do dígito verificador do CPF de
um dos integrantes) set.seed(2034)

km.res = kmeans(dados[, 0:(ncol(dados) - 1)],10) print(km.res)

table(km.res$cluster, dados$tipo)

resultado <- cbind(dados,km.res$cluster) resultado
```

- **REGRAS DE ASSOCIAÇÃO**
- **Musculação**

Regras geradas com uma configuração de Suporte e Confiança.

lhs	1	rhs	support	confidence
{V2= Gemeos}	=>	{V3=Bicicleta}	0.02564	1
{V2= Gemeos}	=>	{V4=Esteira}	0.02564	1
{V2= Gemeos}	=>	{V1=Extensor}	0.02564	1
{V3=Gemeos}	=>	{V2=AgachamentoSmith}	0.05128	1
{V3=Gemeos}	=>	{V4=Esteira}	0.05128	1
{V4=Crucifixo}	=>	{V3=Afundo}	0.05128	1
{V4=Crucifixo}	=>	{V2=Gemeos}	0.05128	1
{V4=Crucifixo}	=>	{V1=LegPress}	0.05128	1
{V2=Flexor}	=>	{V3=Bicicleta}	0.05128	1
{V2=Flexor}	=>	{V4=Esteira}	0.05128	1
{V2=Flexor}	=>	{V1=Extensor}	0.05128	1
{V4=Gemeos}	=>	{V2=AgachamentoSmith}	0.05128	1
{V4=Gemeos}	=>	{V3=Bicicleta}	0.05128	1
{V4=Gemeos}	=>	{V1=Extensor}	0.05128	1
{V4=Bicicleta}	=>	{V3=Afundo}	0.05128	1
{V4=Agachamento}	=>	{V2=LegPress}	0.05128	1
{V4=Agachamento}	=>	{V1=Gemeos}	0.05128	1
{V4=Agachamento}	=>	{V3=Afundo}	0.05128	1
{V4=Adutor}	=>	{V3=LegPress}	0.07692	1
{V3=LegPress}	=>	{V4=Adutor}	0.07692	1
{V4=Adutor}	=>	{V2=Agachamento}	0.07692	1
{V2=Agachamento}	=>	{V4=Adutor}	0.07692	1
{V4=Adutor}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor}	=>	{V4=Adutor}	0.07692	1
{V3=LegPress}	=>	{V2=Agachamento}	0.07692	1
{V2=Agachamento}	=>	{V3=LegPress}	0.07692	1
{V3=LegPress}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor}	=>	{V3=LegPress}	0.07692	1
{V2=Agachamento}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor}	=>	{V2=Agachamento}	0.07692	1
{V2=LegPress}	=>	{V1=Gemeos}	0.07692	1
{V2=LegPress}	=>	{V3=Afundo}	0.07692	1
{V3=Agachamento}	=>	{V2=Gemeos}	0.07692	1
{V3=Agachamento}	=>	{V1=LegPress}	0.07692	1
{V3=Agachamento}	=>	{V4=}	0.07692	1
{V3=Bicicleta}	=>	{V1=Extensor}	0.30769	1

{V2=}	=>	{V3=}	0.33333	1
{V3=}	=>	{V2=}	0.33333	1
{V2=}	=>	{V4=}	0.33333	1
{V3=}	=>	{V4=}	0.33333	1
{V2= Gemeos, V3=Bicicleta}	=>	{V4=Esteira}	0.02564	1
{V2= Gemeos, V4=Esteira}	=>	{V3=Bicicleta}	0.02564	1
{V2= Gemeos, V3=Bicicleta}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2= Gemeos}	=>	{V3=Bicicleta}	0.02564	1
{V2= Gemeos, V4=Esteira}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2= Gemeos}	=>	{V4=Esteira}	0.02564	1
{V1=Afundo, V3=Gemeos}	=>	{V2=AgachamentoSmith}	0.02564	1
{V1=Afundo, V2=AgachamentoSmith}	=>	{V3=Gemeos}	0.02564	1
{V1=Afundo, V3=Gemeos}	=>	{V4=Esteira}	0.02564	1
{V1=Afundo, V4=Esteira}	=>	{V3=Gemeos}	0.02564	1
{V2=AgachamentoSmith, V3=Gemeos}	=>	{V4=Esteira}	0.05128	1
{V3=Gemeos, V4=Esteira}	=>	{V2=AgachamentoSmith}	0.05128	1
{V1=Extensor, V3=Gemeos}	=>	{V2=AgachamentoSmith}	0.02564	1
{V1=Extensor, V3=Gemeos}	=>	{V4=Esteira}	0.02564	1
{V3=Afundo, V4=Crucifixo}	=>	{V2=Gemeos}	0.05128	1
{V2=Gemeos, V4=Crucifixo}	=>	{V3=Afundo}	0.05128	1
{V3=Afundo, V4=Crucifixo}	=>	{V1=LegPress}	0.05128	1
{V1=LegPress, V4=Crucifixo}	=>	{V3=Afundo}	0.05128	1
{V2=Gemeos, V4=Crucifixo}	=>	{V1=LegPress}	0.05128	1
{V1=LegPress, V4=Crucifixo}	=>	{V2=Gemeos}	0.05128	1
{V2=Flexor, V3=Bicicleta}	=>	{V4=Esteira}	0.05128	1
{V2=Flexor, V4=Esteira}	=>	{V3=Bicicleta}	0.05128	1
{V2=Flexor, V3=Bicicleta}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V2=Flexor}	=>	{V3=Bicicleta}	0.05128	1
{V2=Flexor, V4=Esteira}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V2=Flexor}	=>	{V4=Esteira}	0.05128	1
{V2=AgachamentoSmith, V4=Gemeos}	=>	{V3=Bicicleta}	0.05128	1
{V3=Bicicleta, V4=Gemeos}	=>	{V2=AgachamentoSmith}	0.05128	1
{V2=AgachamentoSmith, V4=Gemeos}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V4=Gemeos}	=>	{V2=AgachamentoSmith}	0.05128	1
{V3=Bicicleta, V4=Gemeos}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V4=Gemeos}	=>	{V3=Bicicleta}	0.05128	1
{V2=LegPress, V4=Bicicleta}	=>	{V1=Gemeos}	0.02564	1
{V1=Gemeos, V4=Bicicleta}	=>	{V2=LegPress}	0.02564	1
{V2=LegPress, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1

{V1=Gemeos, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1
{V2=Gemeos, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1
{V1=LegPress, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1
{V2=Gemeos, V4=Bicicleta}	=>	{V1=LegPress}	0.02564	1
{V1=LegPress, V4=Bicicleta}	=>	{V2=Gemeos}	0.02564	1
{V2=LegPress, V4=Agachamento}	=>	{V1=Gemeos}	0.05128	1
{V1=Gemeos, V4=Agachamento}	=>	{V2=LegPress}	0.05128	1
{V2=LegPress, V4=Agachamento}	=>	{V3=Afundo}	0.05128	1
{V3=Afundo, V4=Agachamento}	=>	{V2=LegPress}	0.05128	1
{V1=Gemeos, V4=Agachamento}	=>	{V3=Afundo}	0.05128	1
{V3=Afundo, V4=Agachamento}	=>	{V1=Gemeos}	0.05128	1
{V1=Afundo, V2=AgachamentoSmith}	=>	{V4=Esteira}	0.02564	1
{V1=Afundo, V4=Esteira}	=>	{V2=AgachamentoSmith}	0.02564	1
{V1=Afundo, V2=}	=>	{V3=}	0.05128	1
{V1=Afundo, V3=}	=>	{V2=}	0.05128	1
{V1=Afundo, V2=}	=>	{V4=}	0.05128	1
{V1=Afundo, V4=}	=>	{V2=}	0.05128	1
{V1=Afundo, V3=}	=>	{V4=}	0.05128	1
{V1=Afundo, V4=}	=>	{V3=}	0.05128	1
{V3=LegPress, V4=Adutor}	=>	{V2=Agachamento}	0.07692	1
{V2=Agachamento, V4=Adutor}	=>	{V3=LegPress}	0.07692	1
{V2=Agachamento, V3=LegPress}	=>	{V4=Adutor}	0.07692	1
{V3=LegPress, V4=Adutor}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor, V4=Adutor}	=>	{V3=LegPress}	0.07692	1
{V1=Adutor, V3=LegPress}	=>	{V4=Adutor}	0.07692	1
{V2=Agachamento, V4=Adutor}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor, V4=Adutor}	=>	{V2=Agachamento}	0.07692	1
{V1=Adutor, V2=Agachamento}	=>	{V4=Adutor}	0.07692	1
{V2=Agachamento, V3=LegPress}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor, V3=LegPress}	=>	{V2=Agachamento}	0.07692	1
{V1=Adutor, V2=Agachamento}	=>	{V3=LegPress}	0.07692	1
{V1=Gemeos, V2=LegPress}	=>	{V3=Afundo}	0.07692	1
{V2=LegPress, V3=Afundo}	=>	{V1=Gemeos}	0.07692	1
{V1=Gemeos, V3=Afundo}	=>	{V2=LegPress}	0.07692	1
{V2=Gemeos, V3=Agachamento}	=>	{V1=LegPress}	0.07692	1
{V1=LegPress, V3=Agachamento}	=>	{V2=Gemeos}	0.07692	1
{V2=Gemeos, V3=Agachamento}	=>	{V4=}	0.07692	1
{V3=Agachamento, V4=}	=>	{V2=Gemeos}	0.07692	1
{V2=Gemeos, V4=}	=>	{V3=Agachamento}	0.07692	1

{V1=LegPress, V3=Agachamento}	=>	{V4=}	0.07692	1
{V3=Agachamento, V4=}	=>	{V1=LegPress}	0.07692	1
{V1=Gemeos, V2=}	=>	{V3=}	0.05128	1
{V1=Gemeos, V3=}	=>	{V2=}	0.05128	1
{V1=Gemeos, V2=}	=>	{V4=}	0.05128	1
{V1=Gemeos, V4=}	=>	{V2=}	0.05128	1
{V1=Gemeos, V3=}	=>	{V4=}	0.05128	1
{V1=Gemeos, V4=}	=>	{V3=}	0.05128	1
{V2=Gemeos, V3=Afundo}	=>	{V1=LegPress}	0.07692	1
{V1=LegPress, V3=Afundo}	=>	{V2=Gemeos}	0.07692	1
{V2=Gemeos, V3=Bicicleta}	=>	{V4=Esteira}	0.02564	1
{V2=Gemeos, V4=Esteira}	=>	{V3=Bicicleta}	0.02564	1
{V2=Gemeos, V3=Bicicleta}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2=Gemeos}	=>	{V3=Bicicleta}	0.02564	1
{V2=Gemeos, V4=Esteira}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2=Gemeos}	=>	{V4=Esteira}	0.02564	1
{V2=Gemeos, V4=}	=>	{V1=LegPress}	0.07692	1
{V2=AgachamentoSmith, V3=Bicicleta}	=>	{V1=Extensor}	0.20512	1
{V3=Bicicleta, V4=Esteira}	=>	{V1=Extensor}	0.25641	1
{V1=LegPress, V2=}	=>	{V3=}	0.23076	1
{V1=LegPress, V3=}	=>	{V2=}	0.23076	1
{V2=, V3=}	=>	{V4=}	0.33333	1
{V2=, V4=}	=>	{V3=}	0.33333	1
{V3=, V4=}	=>	{V2=}	0.33333	1
{V1=LegPress, V2=}	=>	{V4=}	0.23076	1
{V1=LegPress, V3=}	=>	{V4=}	0.23076	1
{V2= Gemeos, V3=Bicicleta, V4=Esteira}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2= Gemeos, V3=Bicicleta}	=>	{V4=Esteira}	0.02564	1
{V1=Extensor, V2= Gemeos, V4=Esteira}	=>	{V3=Bicicleta}	0.02564	1
{V1=Afundo, V2=AgachamentoSmith, V3=Gemeos}	=>	{V4=Esteira}	0.02564	1
{V1=Afundo, V3=Gemeos, V4=Esteira}	=>	{V2=AgachamentoSmith}	0.02564	1
{V1=Afundo, V2=AgachamentoSmith, V4=Esteira}	=>	{V3=Gemeos}	0.02564	1
{V1=Extensor, V2=AgachamentoSmith, V3=Gemeos}	=>	{V4=Esteira}	0.02564	1
{V1=Extensor, V3=Gemeos, V4=Esteira}	=>	{V2=AgachamentoSmith}	0.02564	1
{V2=Gemeos, V3=Afundo, V4=Crucifixo}	=>	{V1=LegPress}	0.05128	1
{V1=LegPress, V3=Afundo, V4=Crucifixo}	=>	{V2=Gemeos}	0.05128	1

{V1=LegPress, V2=Gemeos, V4=Crucifixo}	=>	{V3=Afundo}	0.05128	1
{V2=Flexor, V3=Bicicleta, V4=Esteira}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V2=Flexor, V3=Bicicleta}	=>	{V4=Esteira}	0.05128	1
{V1=Extensor, V2=Flexor, V4=Esteira}	=>	{V3=Bicicleta}	0.05128	1
{V2=AgachamentoSmith, V3=Bicicleta, V4=Gemeos}	=>	{V1=Extensor}	0.05128	1
{V1=Extensor, V2=AgachamentoSmith, V4=Gemeos}	=>	{V3=Bicicleta}	0.05128	1
{V1=Extensor, V3=Bicicleta, V4=Gemeos}	=>	{V2=AgachamentoSmith}	0.05128	1
{V1=Gemeos, V2=LegPress, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1
{V2=LegPress, V3=Afundo, V4=Bicicleta}	=>	{V1=Gemeos}	0.02564	1
{V1=Gemeos, V3=Afundo, V4=Bicicleta}	=>	{V2=LegPress}	0.02564	1
{V2=Gemeos, V3=Afundo, V4=Bicicleta}	=>	{V1=LegPress}	0.02564	1
{V1=LegPress, V3=Afundo, V4=Bicicleta}	=>	{V2=Gemeos}	0.02564	1
{V1=LegPress, V2=Gemeos, V4=Bicicleta}	=>	{V3=Afundo}	0.02564	1
{V1=Gemeos, V2=LegPress, V4=Agachamento}	=>	{V3=Afundo}	0.05128	1
{V2=LegPress, V3=Afundo, V4=Agachamento}	=>	{V1=Gemeos}	0.05128	1
{V1=Gemeos, V3=Afundo, V4=Agachamento}	=>	{V2=LegPress}	0.05128	1
{V1=Afundo, V2=, V3=}	=>	{V4=}	0.05128	1
{V1=Afundo, V2=, V4=}	=>	{V3=}	0.05128	1
{V1=Afundo, V3=, V4=}	=>	{V2=}	0.05128	1
{V2=Agachamento, V3=LegPress, V4=Adutor}	=>	{V1=Adutor}	0.07692	1
{V1=Adutor, V3=LegPress, V4=Adutor}	=>	{V2=Agachamento}	0.07692	1
{V1=Adutor, V2=Agachamento, V4=Adutor}	=>	{V3=LegPress}	0.07692	1
{V1=Adutor, V2=Agachamento, V3=LegPress}	=>	{V4=Adutor}	0.07692	1
{V1=LegPress, V2=Gemeos, V3=Agachamento}	=>	{V4=}	0.07692	1
{V2=Gemeos, V3=Agachamento, V4=}	=>	{V1=LegPress}	0.07692	1
{V1=LegPress, V3=Agachamento, V4=}	=>	{V2=Gemeos}	0.07692	1
{V1=LegPress, V2=Gemeos, V4=}	=>	{V3=Agachamento}	0.07692	1
{V1=Gemeos, V2=, V3=}	=>	{V4=}	0.05128	1
{V1=Gemeos, V2=, V4=}	=>	{V3=}	0.05128	1
{V1=Gemeos, V3=, V4=}	=>	{V2=}	0.05128	1
{V2=Gemeos, V3=Bicicleta, V4=Esteira}	=>	{V1=Extensor}	0.02564	1
{V1=Extensor, V2=Gemeos, V3=Bicicleta}	=>	{V4=Esteira}	0.02564	1
{V1=Extensor, V2=Gemeos, V4=Esteira}	=>	{V3=Bicicleta}	0.02564	1
{V2=AgachamentoSmith, V3=Bicicleta, V4=Esteira}	=>	{V1=Extensor}	0.15384	1

{V1=LegPress, V2=, V3=}	=>	{V4=}	0.23076	1
{V1=LegPress, V2=, V4=}	=>	{V3=}	0.23076	1
{V1=LegPress, V3=, V4=}	=>	{V2=}	0.23076	1
{V1=Extensor}	=>	{V3=Bicicleta}	0.30769	9,23077E+1 4
{V4=Esteira}	=>	{V1=Extensor}	0.28205	9,16667E+1 4
{V1=Extensor, V4=Esteira}	=>	{V3=Bicicleta}	0.25641	9,09091E+1 4
{V2=AgachamentoSmith}	=>	{V1=Extensor}	0.23076	9
{V1=Extensor, V2=AgachamentoSmith}	=>	{V3=Bicicleta}	0.20512	8,88889E+1 4
{V2=AgachamentoSmith, V4=Esteira}	=>	{V1=Extensor}	0.17948	875
{V2=Gemeos}	=>	{V1=LegPress}	0.15384	8,57143E+1 4
{V1=Extensor, V2=AgachamentoSmith, V4=Esteira}	=>	{V3=Bicicleta}	0.15384	8,57143E+1 4
{V1=Extensor}	=>	{V4=Esteira}	0.28205	8,46154E+1 4
{V3=Bicicleta}	=>	{V4=Esteira}	0.25641	8,33333E+1 4
{V4=Esteira}	=>	{V3=Bicicleta}	0.25641	8,33333E+1 4
{V1=Extensor, V3=Bicicleta}	=>	{V4=Esteira}	0.25641	8,33333E+1 4
{V4=}	=>	{V2=}	0.33333	8125
{V4=}	=>	{V3=}	0.33333	8125
{V2=AgachamentoSmith}	=>	{V3=Bicicleta}	0.20512	8
{V2=AgachamentoSmith}	=>	{V4=Esteira}	0.20512	8
{V1=LegPress}	=>	{V4=}	0.30769	8
{V1=Extensor, V2=AgachamentoSmith}	=>	{V4=Esteira}	0.17948	7,77778E+1 4
{V4=}	=>	{V1=LegPress}	0.30769	75
{V2=AgachamentoSmith, V3=Bicicleta}	=>	{V4=Esteira}	0.15384	75
{V2=AgachamentoSmith, V4=Esteira}	=>	{V3=Bicicleta}	0.15384	75
{V1=LegPress, V4=}	=>	{V2=}	0.23076	75
{V1=LegPress, V4=}	=>	{V3=}	0.23076	75
{V1=Extensor, V2=AgachamentoSmith, V3=Bicicleta}	=>	{V4=Esteira}	0.15384	75

Colocar a lista de comandos emitidos no RStudio para conseguir os resultados obtidos

```
### Pacotes necessários library(arules)
library(datasets)

##Leitura da database
setwd("D:/GIT/IAA008-AprendizadoMaquina/")

dados <- read.csv2("databases/2 - Musculacao - Dados.csv", header =
F)
View(dados) summary(dados)

set.seed(2034)

rules <- apriori(dados, parameter = list(supp = 0.001, conf =
0.7, minlen = 2)) summary(rules)

options(digits=2)

inspect(sort(rules, by="confidence"))
```

APÊNDICE 8 – DEEP LEARNING

A – ENUNCIADO

1 Classificação de Imagens (CNN)

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

2 Detector de SPAM (RNN)

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

3 Gerador de Dígitos Fake (GAN)

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

4 Tradutor de Textos (Transformer)

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

B – RESOLUÇÃO

1. Classificação de Imagem (CNN)

[<Link para o arquivo jupyter notebook>](#)

```
#Importações Deep Learning

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
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

##Normalização dos dados

x_train, x_test = x_train / 255.0, x_test / 255.0
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)

##Criando a CNN

K = len(set(y_train))
#Estágio 1
i = Input(shape=x_train[0].shape)
x = Conv2D(32, (3,3), strides=2, activation='relu')(i)
x = Conv2D(64, (3,3), strides=2, activation='relu')(x)
x = Conv2D(128, (3,3), strides=2, activation='relu')(x)
#Todas as imagens são do mesmo tamanho, não precisa de Global Pooling
x = Flatten()(x)
#Estágio 2
x = Dropout(0.5)(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.2)(x)
x = Dense(K, activation='softmax')(x)
#Model (lista entrada, lista saída)
model = Model(i, x)
#Relatório do modelo
model.summary()

##Compilar e treinar o modelo

#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)

##Mostrar os gráficos do treino: loss e acurácia
```

```

#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 predição e mostrar Matriz de Confusão

#Efetuar predições na base de teste

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 classificações erradas

# 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]]))

```

2. Detector de SPAM (RNN)

[<Link para o arquivo jupyter notebook>](#)

```

##Importação da biblioteca

# 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

##Carga da base

# 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

##Separação de treino e teste

# 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)

##Tokenização dos dados

# 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)

##Acertando o tamanho (Padding)

# 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)

##Definição do modelo da RNN

# 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)

##Visualização do modelo criado

model.summary()

##Compilar e Treinar o modelo

# Compila e treina o modelo
model.compile(loss="binary_crossentropy", optimizer="adam",
metrics=["accuracy"])
epochs = 5
r = model.fit(data_train, y_train, epochs=epochs, validation_data=(data_test,
y_test))

```



```

##Função de perda e acurácia

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

##Predição de um novo texto

# 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")

```

3. Gerador de Dígitos Fake (GAN)

[<Link para o arquivo jupyter notebook>](#)

```

## Instalar bibliotecas para geração de GIFs

# Para gerar os GIFs
!pip install imageio
!pip install git+https://github.com/tensorflow/docs

##Importações das bibliotecas necessárias

# Importações
import tensorflow as tf
import glob
import imageio
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
from tensorflow.keras import layers
import time

from IPython import display

## Tratamento da base

# Carrega a base de dados

```

```

(train_images, train_labels), (_, _) = tf.keras.datasets.mnist.load_data()

# Normalização
train_images = train_images.reshape(train_images.shape[0], 28, 28,
1).astype('float32')
train_images = (train_images - 127.5) / 127.5 # Normaliza entre [-1, 1]

# Gera o banco em partes e randomiza
BUFFER_SIZE = 60000
BATCH_SIZE = 256

# Cria o dataset (from_tensor_slices)
# shuffle: embaralha os dados
# Combina elementos consecutivos em lotes (batch)
train_dataset =
tf.data.Dataset.from_tensor_slices(train_images).shuffle(BUFFER_SIZE).batch(BATCH_
SIZE)

## Gerador de imagens

# Cria o gerador
def make_generator_model():
    model = tf.keras.Sequential()
    model.add(layers.Dense(7*7*256, use_bias=False, input_shape=(100,)))
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())
    model.add(layers.Reshape((7, 7, 256)))
    assert model.output_shape == (None, 7, 7, 256) # None é o batch size
    model.add(layers.Conv2DTranspose(128, (5, 5), strides=(1, 1), padding='same',
use_bias=False))
    assert model.output_shape == (None, 7, 7, 128)
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())
    model.add(layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same',
use_bias=False))
    assert model.output_shape == (None, 14, 14, 64)
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())
    model.add(layers.Conv2DTranspose(1, (5, 5), strides=(2, 2), padding='same',
use_bias=False, activation='tanh'))
    assert model.output_shape == (None, 28, 28, 1)
    return model

## Teste do Gerador

# 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')

## Criação do discriminador

# Cria o DISCRIMINADOR
def make_discriminator_model():
    model = tf.keras.Sequential()

```

```

        model.add(layers.Conv2D(64, (5, 5), strides=(2,2), padding='same',
input_shape=[28, 28, 1]))
        model.add(layers.LeakyReLU())
        model.add(layers.Dropout(0.3))

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

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

## Teste do Discriminador

discriminator = make_discriminator_model()
decision = discriminator(generated_image)
print (decision)

## Funções de perda - Pré-step treinamento

# Definição das funções de perda
cross_entropy = tf.keras.losses.BinaryCrossentropy(from_logits=True)

# Função de perda do discriminador
def discriminator_loss(real_output, fake_output):
    real_loss = cross_entropy(tf.ones_like(real_output), real_output)
    fake_loss = cross_entropy(tf.zeros_like(fake_output), fake_output)
    total_loss = real_loss + fake_loss
    return total_loss

# Função de perda do gerador
def generator_loss(fake_output):
    return cross_entropy(tf.ones_like(fake_output), fake_output)

## Criação dos otimizadores

# Cria os otimizadores para o gerador e discriminador
generator_optimizer = tf.keras.optimizers.Adam(1e-4)
discriminator_optimizer = tf.keras.optimizers.Adam(1e-4)

## Checkpoints

# Cria checkpoints para salvar modelos ao longo do tempo
# Úteis em tarefas longas, para se recuperar em um desligamento
checkpoint_dir = './training_checkpoints'
checkpoint_prefix = os.path.join(checkpoint_dir, "ckpt")
checkpoint = tf.train.Checkpoint(generator_optimizer=generator_optimizer,
                                discriminator_optimizer=discriminator_optimizer,
                                generator=generator,
                                discriminator=discriminator)

# Configura o loop de treinamento
EPOCHS = 100
noise_dim = 100
num_examples_to_generate = 16

# Seed
seed = tf.random.normal([num_examples_to_generate, noise_dim])

```

```

## Verificação dos gradientes

def f(x,y):
    return 3*x**2 + 2*y**2

x,y = tf.Variable(5.), tf.Variable(3.)
with tf.GradientTape() as tape:
    z = f(x,y)
gradients = tape.gradient(z, [x,y])
print(gradients)

## Passo de Treinamento

# Função que faz um passo de treinamento
# É uma 'tf.function', que compila essa função
@tf.function
def train_step(images):
    noise = tf.random.normal([BATCH_SIZE, noise_dim])

    with tf.GradientTape() as gen_tape, tf.GradientTape() as disc_tape:
        generated_images = generator(noise, training=True)

        real_output = discriminator(images, training=True)
        fake_output = discriminator(generated_images, training=True)

        gen_loss = generator_loss(fake_output)
        disc_loss = discriminator_loss(real_output, fake_output)

        gradients_of_generator = gen_tape.gradient(gen_loss,
generator.trainable_variables)
        gradients_of_discriminator = disc_tape.gradient(disc_loss,
discriminator.trainable_variables)

        generator_optimizer.apply_gradients(zip(gradients_of_generator,
generator.trainable_variables))
        discriminator_optimizer.apply_gradients(zip(gradients_of_discriminator,
discriminator.trainable_variables))

## Loop de treinamento

# Treinamento completo/laço
def train (dataset, epochs):
    for epoch in range(epochs):
        start = time.time()

        for image_batch in dataset:
            train_step(image_batch)

            # Produz imagens do GIF
            display.clear_output(wait=True)
            generate_and_save_images(generator, epoch + 1, seed)

        # Salva o modelo a cada 15 épocas
        if (epoch + 1) % 15 == 0:
            checkpoint.save(file_prefix = checkpoint_prefix)

        print ('Time for epoch {} is {} sec'.format(epoch + 1, time.time()-start))

```

```

# Gera depois da epoca final
display.clear_output(wait=True)
generate_and_save_images(generator, epochs, seed)

## Salvamento de Imagens

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

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

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

## Treinamento

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

## Criação do GIF

# Criar m GIF

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)
        image = imageio.imread(filename)
        writer.append_data(image)

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

```

4. Tradutor de Textos (Transformer)

[<Link para o arquivo jupyter notebook>](#)

```

## Importação da biblioteca

!pip uninstall tensorflow
!pip install tensorflow==2.15.0
!pip install tensorflow datasets
!pip install -U tensorflow-text==2.15.0

```

```

import collections
import logging
import os
import pathlib
import string
import sys
import time

import numpy as np
import matplotlib.pyplot as plt

import tensorflow as tf
import tensorflow_datasets as tfds
import tensorflow_text as text

logging.getLogger('tensorflow').setLevel(logging.ERROR) # suppress warnings

## Carrega 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']

## Verifica o dataset

# Verifica o dataset
for pt_examples, en_examples in train_examples.batch(3).take(1):
    for pt in pt_examples.numpy():
        print(pt.decode('utf-8'))

    print()

    for en in en_examples.numpy():
        print(en.decode('utf-8'))

## Tokenização e Destokenização

# Tokenização e Destokenização do texto

model_name = "ted_hrlr_translate_pt_en_converter"

tf.keras.utils.get_file(f"{model_name}.zip",
f"https://storage.googleapis.com/download.tensorflow.org/models/{model_name}.zip",
cache_dir='.', cache_subdir='', extract=True)

# Tem 2 tokenizers: um pt outr em en
# tokenizers .en tokeniza e destokeniza
tokenizers = tf.saved_model.load(model_name)

## Pipeline de Entrada

# Pipeline de entrada
# Codificar/tokenizar lotes de texto puro

def tokenize_pairs(pt, en):
    pt = tokenizers.pt.tokenize(pt)
    # Converte ragged (irregular, tam variável) para dense
    # Faz padding com zeros.
    pt = pt.to_tensor()
```

```

    en = tokenizers.en.tokenize(en)
    # ragged -> dense
    en = en.to_tensor()

    return pt, en

# Pipeline simples: processa, embaralha, agrupa os dados, prefetch
# Datasets de entrada terminam com prefetch

BUFFER_SIZE = 20000
BATCH_SIZE = 64

def make_batches(ds):
    return (
        ds
        .cache()
        .shuffle(BUFFER_SIZE)
        .batch(BATCH_SIZE)
        .map(tokenize_pairs, num_parallel_calls=tf.data.AUTOTUNE)
        .prefetch(tf.data.AUTOTUNE))

train_batches = make_batches(train_examples)
val_batches = make_batches(val_examples)

## Codificação posicional

# Codificação posicional

def get_angles(pos, i, d_model):
    angle_rates = 1 / np.power(10000, (2 * (i//2)) / np.float32(d_model))
    return pos * angle_rates

def positional_encoding(position, d_model):
    angle_rads = get_angles(np.arange(position)[:, np.newaxis],
                             np.arange(d_model)[np.newaxis, :],
                             d_model)

    # sin em índices pares no array; 2i
    angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])

    # cos em índices ímpares no array; 2i+1
    angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])

    # newaxis, aumenta a dimensão [] -> [ [] ]
    pos_encoding = angle_rads[np.newaxis, ...]
    return tf.cast(pos_encoding, dtype=tf.float32)

# 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))

```



```

## Cria a Máscara de 0s e 1s

# Cria uma mascara de 0 e 1, 0 para quando há valor e 1 quando não há
def create_padding_mask(seq):
    seq = tf.cast(tf.math.equal(seq, 0), tf.float32)
    # Add extradimensions to add the padding
    # to the attention logits
    return seq[:, tf.newaxis, tf.newaxis, :] # (batch_size, 1, 1, seq_len)

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

## Função de atenção

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

    #Converte matmul qk para float32
    dk = tf.cast(tf.shape(k)[-1], tf.float32)
    # divide por sqrt (d_k)
    scaled_attention_logits = matmul_qk / tf.math.sqrt(dk)

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

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

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

    return output, attention_weights

## Atenção Multi-cabeças

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

        assert d_model % self.num_heads == 0

        self.depth = d_model // self.num_heads

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

    def split_heads(self, x, batch_size):

```

```

    # Separa a última dimensão em (num_heads, depth). Transpões o resultado para o
    shape (batch_size, num_heads, self.depth)
    x = tf.reshape(x, (batch_size, -1, self.num_heads, self.depth))
    return tf.transpose(x, perm=[0, 2, 1, 3])

def call(self, v, k, q, mask):
    batch_size = tf.shape(q)[0]

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

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

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

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

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

    return output, attention_weights

## Cria rede feed-forward pontual
def point_wise_feed_forward_network(d_model, dff):
    return tf.keras.Sequential([
        tf.keras.layers.Dense(dff, activation='relu'), # (batch_size, seq_len, dff)
        tf.keras.layers.Dense(d_model) # (batch_size, seq_len, d_model)
    ])

## Camada do Codificador
class EncoderLayer(tf.keras.layers.Layer):
    def __init__(self, d_model, num_heads, dff, rate=0.1):
        super(EncoderLayer, self).__init__()

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

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

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

    def call(self, x, training, mask):
        attn_output, _ = self.mha(x, x, x, mask) # (batch_size, input_seq_len, d_model)

```

```

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

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

    return out2

## Camada do Decodificador

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

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

        self.ffn = point_wise_feed_forward_network(d_model, dff)

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

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

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

        # (batch_size, target_seq_len, d_model)
        attn1, att_weights_block1 = self.mha1(x, x, x, look_ahead_mask)
        attn1 = self.dropout1(attn1, training=training)
        out1 = self.layernorm1(attn1 + x)

        # (batch_size, target_seq_len, d_model)
        attn2, att_weights_block2 = self.mha2(enc_output, enc_output, out1,
padding_mask)
        attn2 = self.dropout2(attn2, training=training)
        out2 = self.layernorm2(attn2 + out1) # (batch_size, target_seq_len, d_model)

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

        return out3, att_weights_block1, att_weights_block2

## Encoder completo

class Encoder(tf.keras.layers.Layer):
    def __init__(self, num_layers, d_model, num_heads, dff, input_vocab_size,
maximum_position_encoding, rate=0.1):
        super(Encoder, self).__init__()

        self.d_model = d_model

```

```

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

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

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

        return x # (batch_size, target_seq_len, d_model)

## Decoder completo

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

        self.d_model = d_model
        self.num_layers = num_layers

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

        self.dec_layers = [DecoderLayer(d_model, num_heads, dff, rate)
for _ in range(num_layers)]

        self.dropout = tf.keras.layers.Dropout(rate)

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

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

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

        for i in range(self.num_layers):
            x, block1, block2 = self.dec_layers[i](x, enc_output, training,
look_ahead_mask, padding_mask)

            attention_weights[f'decoder_layer{i+1}_block1'.format(i+1)] = block1
            attention_weights[f'decoder_layer{i+1}_block2'.format(i+1)] = block2

        # x.shape == (batch_size, target_seq_len, d_model)
        return x, attention_weights

```

```
## Transformer Completo
```

```
class Transformer(tf.keras.Model):
    def __init__(self, num_layers, d_model, num_heads, dff, input_vocab_size,
target_vocab_size, pe_input, pe_target, rate=0.1):
        super().__init__()

        self.encoder = Encoder(num_layers, d_model, num_heads, dff, input_vocab_size,
pe_input, rate)
        self.decoder = Decoder(num_layers, d_model, num_heads, dff, target_vocab_size,
pe_target, rate)
        self.final_layer = tf.keras.layers.Dense(target_vocab_size)

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

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

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

        dec_output, attention_weights = self.decoder(tar, enc_output, training,
look_ahead_mask, dec_padding_mask)

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

        return final_output, attention_weights

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

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

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

        return enc_padding_mask, look_ahead_mask, dec_padding_mask
```

```
## Hiperparâmetros
```

```
# Hiperparâmetros
num_layers = 4
d_model = 128
dff = 512
num_heads = 8
dropout_rate = 0.1
```

```
## Otimizador
```

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

        self.d_model = d_model
        self.d_model = tf.cast(self.d_model, tf.float32)
        self.warmup_steps = warmup_steps

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

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

```
## Função de perda e métrica de Acurácia (mascarados)
```

```
loss_object = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True,
reduction='none')
```

```
def loss_function(real, pred):
    mask = tf.math.logical_not(tf.math.equal(real, 0))
    loss_ = loss_object(real, pred)

    mask = tf.cast(mask, dtype=loss_.dtype)
    loss_ *= mask

    return tf.reduce_sum(loss_)/tf.reduce_sum(mask)

def accuracy_function(real, pred):
    accuracies = tf.equal(real, tf.argmax(pred, axis=2))
    mask = tf.math.logical_not(tf.math.equal(real, 0))
    accuracies = tf.math.logical_and(mask, accuracies)
    accuracies = tf.cast(accuracies, dtype=tf.float32)
    mask = tf.cast(mask, dtype=tf.float32)
    return tf.reduce_sum(accuracies)/tf.reduce_sum(mask)
```

```
trains_loss = tf.keras.metrics.Mean(name='train_loss')
trains_accuracy = tf.keras.metrics.Mean(name='train_accuracy')
```

```
## Treinamento
```

```
transformer = Transformer(num_layers, d_model, num_heads, dff,
tokenizers.pt.get_vocab_size().numpy(), tokenizers.en.get_vocab_size().numpy(),
pe_input=1000, pe_target=1000, rate=dropout_rate)
```

```
## Checkpoint
```

```
#Checkpoint
checkpoint_path = "./checkpoints/train"
ckpt = tf.train.Checkpoint(transformer=transformer, optimizer=optimizer)
```

```

ckpt_manager = tf.train.CheckpointManager(ckpt, checkpoint_path, max_to_keep=5)

ckpt_manager = tf.train.CheckpointManager(ckpt, checkpoint_path, max_to_keep=5)

# if a checkpoint exists, restore the latest checkpoint
if ckpt_manager.latest_checkpoint:
    ckpt.restore(ckpt_manager.latest_checkpoint)
    print ('Latest checkpoint restored!!')

## Processo de treinamento

EPOCHS = 20

train_step_signature = [
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
    tf.TensorSpec(shape=(None, None), dtype=tf.int64),
]

@tf.function(input_signature=train_step_signature)
def train_step(inp, tar):
    tar_inp = tar[:, :-1]
    tar_real = tar[:, 1:]
    with tf.GradientTape() as tape:
        predictions, _ = transformer([inp, tar_inp], training = True)
        loss = loss_function(tar_real, predictions)

    gradients = tape.gradient(loss, transformer.trainable_variables)
    optimizer.apply_gradients(zip(gradients, transformer.trainable_variables))

    trains_loss(loss)
    trains_accuracy(accuracy_function(tar_real, predictions))

for epoch in range(EPOCHS):
    start = time.time()
    trains_loss.reset_states()
    trains_accuracy.reset_states()
    # imp -> portuguese, tar _> english
    for (batch, (inp, tar)) in enumerate(train_batches):
        train_step(inp, tar)
        if batch % 50 == 0:
            print (f'Epoch {epoch + 1} Batch {batch} Loss {trains_loss.result():.4f}
Accuracy {trains_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 {trains_loss.result():.4f} Accuracy
{trains_accuracy.result():.4f}')

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

## Tradutor

class Translator(tf.Module):
    def __init__(self, tokenizers, transformer):
        self.tokenizers = tokenizers
        self.transformer = transformer

```

```

def __call__(self, sentence, max_length=20):
    # input sentence in portuguese, hence adding the start and end token
    assert isinstance(sentence, tf.Tensor)
    if len(sentence.shape) == 0:
        sentence = sentence[tf.newaxis]
    sentence = self.tokenizers.pt.tokenize(sentence).to_tensor()
    encoder_input = sentence

    # as the target is english, the first token to the transformer should be the
    english start token.
    start_end = self.tokenizers.en.tokenize([''])[0]
    start = start_end[0][tf.newaxis]
    end = start_end[1][tf.newaxis]

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

    for i in tf.range(max_length):
        output = tf.transpose(output_array.stack())
        predictions, _ = self.transformer([encoder_input, output], training=False)
        # select the last token from the seq_len dimension
        predictions = predictions[:, -1:, :] # (batch_size, 1, vocab_size)
        predicted_id = tf.argmax(predictions, axis=-1)
        output_array = output_array.write(i+1, predicted_id[0])
        if predicted_id == end:
            break

    output = tf.transpose(output_array.stack())
    # output.shape (1, tokens)
    text = tokenizers.en.detokenize(output)[0]
    tokens = tokenizers.en.lookup(output)[0]

    _, attention_weights = self.transformer([encoder_input, output[:, :-1]],
training=False)

    return text, tokens, attention_weights

## Efetuando uma tradução

# Efetuando uma tradução com o Transformer

translator = Translator(tokenizers, transformer)
sentence = "Eu gostaria de saber ."
translated_text, translated_tokens, attention_weights =
translator(tf.constant(sentence))

print(translated_text)

```


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

Estudo de Caso: **Análise de Dados de Mobilidade Urbana com Dados Abertos de Transporte Público**

Descrição

Este estudo de caso visa propor uma arquitetura de Big Data para análise de padrões de mobilidade urbana em Curitiba utilizando dados públicos. Os dados disponíveis incluem e não se limitam a: frota de veículos, linhas existentes, itinerários, equipamentos públicos, terminais de transporte, ciclovias, ciclorrotas, sistema viário, arruamento, quadras e dados populacionais do Censo IBGE.

Entre os objetivos finais desejados em uma eventual implantação da arquitetura estão, a possibilidade de prever congestionamentos, identificar a necessidade de novas linhas de transporte público, propor rotas otimizadas para ciclovias, propor novas localizações de terminais de ônibus. Visualizações interativas em dashboards seriam utilizadas para facilitar a interpretação dos resultados, objetivando melhorar a eficiência do transporte público e a qualidade de vida dos cidadãos.

Ingestão e Processamento de Dados

Entre os dados relevantes para a análise estariam: a relação de linhas, itinerários, frotas e horários do transporte público, que seriam disponibilizados mensalmente pelo portal de dados abertos da Prefeitura de Curitiba; além disso, dados geográficos de arruamento, zoneamento, eixos de ruas e ciclovias da cidade, fornecidos em formato vetorizado de pontos, linhas e polígonos (SHP), também seriam disponibilizados pela Prefeitura, porém sem periodicidade definida. Adicionalmente, os dados no formato GTFS seriam acessíveis via Webservice.

Para extrair informações de um arquivo com extensão “.json” e processá-las em um servidor Hadoop, primeiro seria necessário garantir que o Hadoop está instalado e configurado, incluindo o HDFS (Hadoop Distributed File System) e o YARN para gerenciamento de recursos.

Em seguida, mover o arquivo JSON para o HDFS usando o comando “hadoop fs -put /caminho/para/arquivo.json /diretorio/destino/no/HDFS”. Após o arquivo estar no HDFS, seria possível processá-lo utilizando ferramentas como Apache Hive ou Apache Pig.

No Hive, é possível criar uma tabela externa que aponta para o arquivo JSON e usa a SerDe (Serializer/Deserializer) específica para JSON. Por exemplo, a criação da tabela poderia ser feita com o seguinte comando SQL: `CREATE EXTERNAL TABLE json_table (key1 STRING, key2 STRING, key3 INT) ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' LOCATION '/user/hadoop/input/';`, onde *json_table* é o nome da tabela e *key1*, *key2*, *key3* são campos que correspondem às chaves no arquivo JSON. Após criar a tabela, seria possível executar consultas SQL sobre esses dados, como `SELECT key1, key2 FROM json_table WHERE key3 > 10;`.

Alternativamente, no Apache Pig, poderia ser escrito um script para carregar e processar o JSON. Um exemplo seria: `data = LOAD '/user/hadoop/input/dados.json' USING JsonLoader('key1:chararray, key2:chararray, key3:int');` para carregar o arquivo, seguido de comandos para filtrar e projetar os dados. Após o processamento, os resultados podem ser armazenados de volta no HDFS, e para visualizá-los, poderia usar o comando `hadoop fs -cat /user/hadoop/output/part-00000`.

Uma outra possível abordagem, seria escrever um job MapReduce para processar o JSON diretamente, utilizando bibliotecas como `org.json` ou `Jackson` no código Java. O processamento final pode ser acessado no HDFS com o comando `hadoop fs -cat`, que exibirá os resultados após a execução das consultas ou jobs.

Para extrair informações de um arquivo `.dbf` e processá-las em um servidor Hadoop, o primeiro passo é converter o arquivo para um formato compatível, como CSV ou JSON, já que o Hadoop não suporta diretamente arquivos `.dbf`. A conversão pode ser feita utilizando a linguagem Python com a biblioteca `dbfread`, que permite ler arquivos `.dbf` e convertê-los para CSV. Primeiro, deve-se instalar a biblioteca com o comando `pip install dbfread`. Depois, utilizar um script Python para realizar a conversão, que carrega o arquivo `.dbf` e escreve seu conteúdo em um arquivo CSV. Após a conversão, o arquivo CSV pode ser carregado no HDFS (Hadoop Distributed File System) com o comando `hadoop fs -put /caminho/para/saida.csv /diretorio/destino/no/HDFS`, onde é especificado o caminho do arquivo convertido e o diretório de destino no HDFS.

Com o arquivo no HDFS, ele pode ser processado usando ferramentas como Apache Hive ou Apache Pig. No Apache Hive, você cria uma tabela que aponta para o arquivo CSV com a seguinte instrução SQL: `CREATE EXTERNAL TABLE dbf_table (coluna1 STRING, coluna2 INT, coluna3 DOUBLE) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE LOCATION '/user/hadoop/input/';`. Isso define uma tabela externa baseada no CSV que está no HDFS, permitindo que se execute consultas SQL diretamente nos dados, como `SELECT coluna1, coluna2 FROM dbf_table WHERE coluna3 > 100;`.

Alternativamente, no Apache Pig, se pode carregar o arquivo CSV com o comando `LOAD` e processar os dados com operações como filtro e projeção. Por exemplo, após carregar o arquivo com `LOAD '/user/hadoop/input/saida.csv' USING PigStorage(',') AS (coluna1:chararray, coluna2:int, coluna3:double);`, se pode aplicar um filtro para selecionar registros com um valor maior que 100 em `coluna3` e armazenar o resultado de volta no HDFS.

Por fim, os dados processados podem ser acessados no HDFS utilizando o comando ``hadoop fs -cat /user/hadoop/output/part-00000``, que exibe os resultados diretamente no terminal. Se preferir, também seria possível desenvolver um job MapReduce personalizado para processar o CSV, utilizando bibliotecas específicas para leitura e manipulação de arquivos de texto no Hadoop.

Armazenamento e análise

Para o armazenamento dos dados de maneira distribuída e escalável, utilizaríamos o HDFS. O HBase seria empregado para gerenciar dados que demandariam acesso rápido e aleatório, sendo uma solução NoSQL construída sobre o HDFS.

O processamento e a transformação dos dados seriam feitos com o Apache Spark, aproveitando o Spark MLlib para aplicar algoritmos de machine learning e gerar modelos preditivos. Para a criação de um data warehouse sobre o HDFS, utilizaríamos o Apache Hive, que permitiria consultas semelhantes a SQL (HiveQL) para análise de grandes volumes de dados. A integração entre Hive e Spark possibilitaria otimizar o desempenho das consultas e análises.

Vs do Big Data

VOLUME

É necessário garantir o volume, pois se trata de uma grande quantidade de dados, pode-se estimar gigas de informações, dependendo do tamanho da cidade e da quantidade de dispositivos que enviarão as informações. Com o HDFS do Hadoop, é possível armazenar grande quantidade de dados, distribuídos em pequenas partes entre servidores, garantindo o V de Volume dos Big Datas.

VARIEDADE

Devido a quantidade de fontes de informações, e preciso garantir a variedade, pois são diversos dispositivos enviando seus dados, pode ser em diversos formatos, dependendo do dispositivo, então, é necessário garantir o recebimento de todos os tipos de dados. Com o Hive, HBase ou Cassandra, é possível armazenar uma grande variedade de dados, sendo possível armazenar arquivos, inserir informações no SQL ou mesmo enviar arquivos logs, json e afins. Garantindo assim o V de Variedade do Big Data.

VERACIDADE

Se tratando de dados que serão usados para gerar informações para a população, é necessário garantir a veracidade dos dados. O modelo a ser usado para prever, identificar e propor requer dados confiáveis. A partir do Hadoop, pode-se garantir a disponibilidade dos dados, a segurança, o monitoramento e auditoria das informações, garantindo assim o V de Veracidade do Big Data.

VALOR

Os dados devem ser relevantes para o modelo proposto, pois dados sem valor não são úteis. Após o processamento inicial com Map-Reduce, os dados são armazenados no HBase, garantindo sua

integridade para uso com Drill ou Mahout. Além disso, a coleta de dados deve ser planejada para assegurar o V de Valor do Big Data.

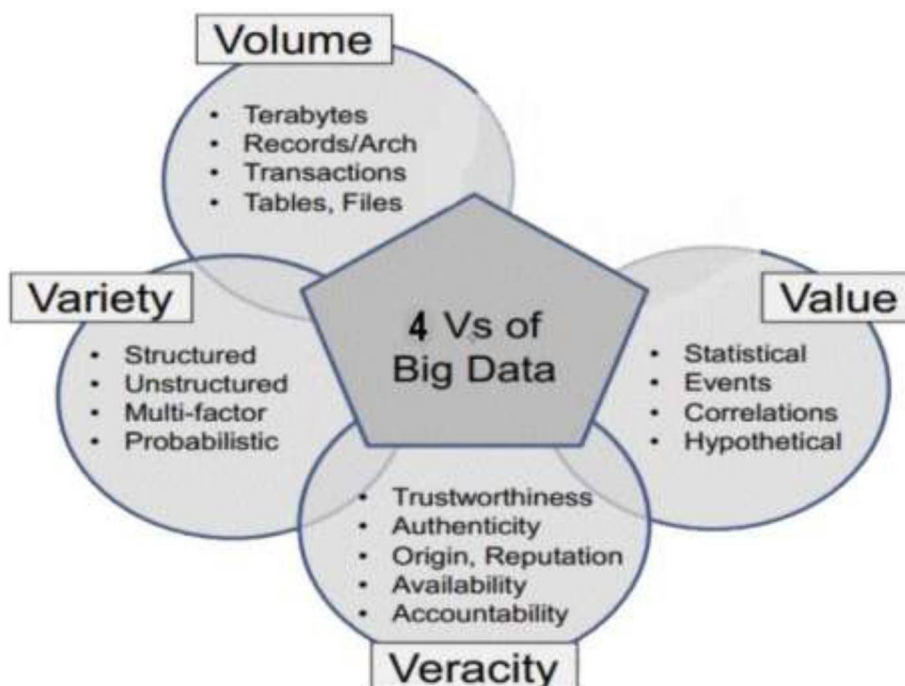


Figura 1 – Vs de Big Data necessários para o projeto

Visualização dos Dados

Os dados podem ser disponibilizados em tabelas relacionais com as principais consultas já configuradas e automatizadas no Power BI, cujos dashboards interativos facilitam a compreensão e análise dos profissionais das diversas áreas que poderiam fazer uso das informações mesmo que não possuam profundo conhecimento de ferramentas de análise de dados, assim democratizando as informações obtidas gerando resultados de maior valor.

Arquitetura Proposta

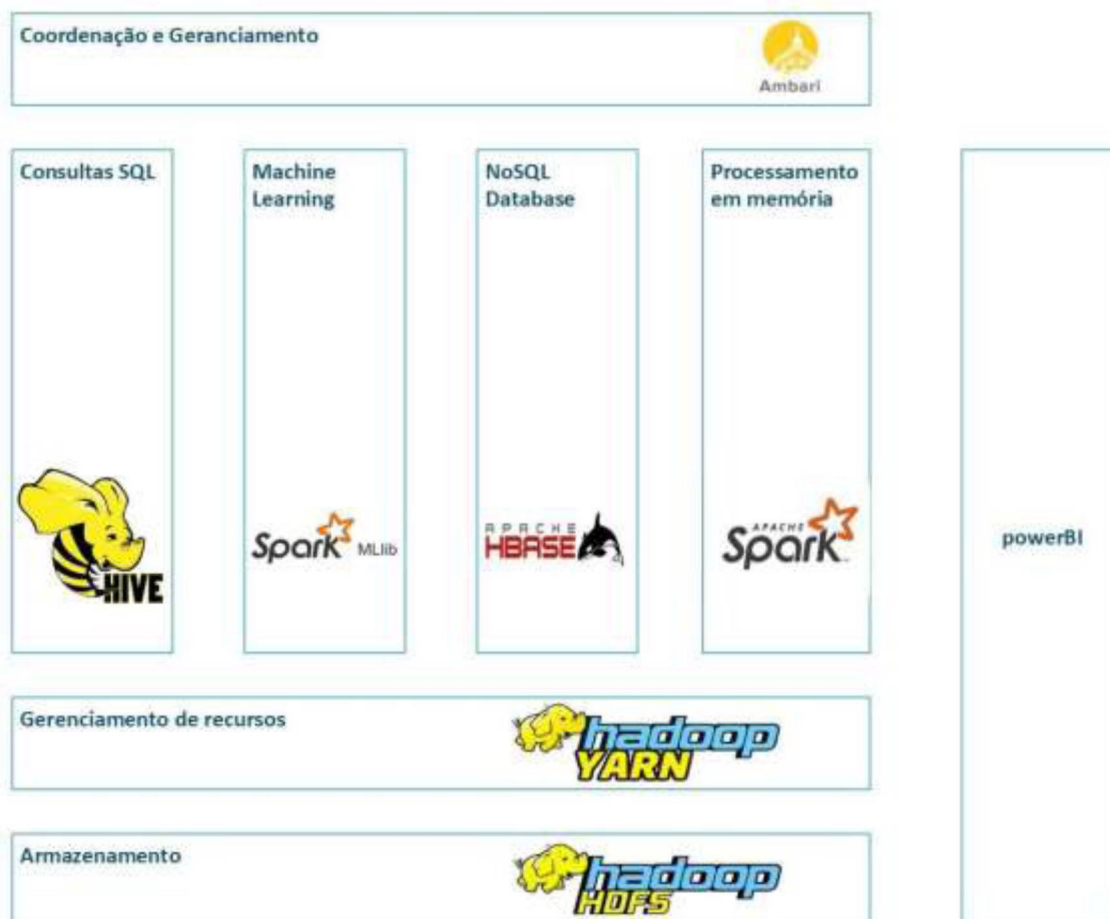


Figura 2 – Proposta de arquitetura

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.
- Aplique 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:

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

B – RESOLUÇÃO

De acordo com os treinamentos e testes executados nesse trabalho, obtivemos os seguintes resultados:

1- Extração de Características:

Base de treino	Sensibilidade	Especificidade	F1-Score
RF - LBP	0.766060	0.789186	0.757797
RF - CNN	0.841730	0.850291	0.842091
SVM - LBP	0.786877	0.826087	0.790327
SVM - CNN	0.831258	0.846108	0.83201
RNA - LBP	0.74122	0.779888	0.743243
RNA - CNN	0.825351	0.830564	0.824516

Base de Teste	Sensibilidade	Especificidade	F1-Score
RF - LBP	0.689026	0.697747	0.691186
RF - CNN	0.794087	0.795062	0.786184
SVM - LBP	0.683718	0.721638	0.691717
SVM - CNN	0.74934	0.74087	0.736752
RNA - LBP	0.631903	0.646659	0.634835
RNA - CNN	0.7555	0.753941	0.751216

O modelo escolhido na extração de características foi o Random Forest CNN que foi melhor em todas as métricas tanto na base de treino como da de testes.

2- Redes Neurais:

Redes Neurais - Base de Teste	Sensibilidade	Especificidade	F1-Score
VGG16 com D.A.	60.86%	62.07%	59.20%
VGG16 sem D.A.	48.97%	44.83%	44.28%
ResNet50 com D.A.	80.74%	75.86%	74.07%
ResNet50 sem D.A.	80.38%	78.45%	76.36%

O modelo escolhido dentre as redes neurais foi o Resnet50 sem data augmentation, pois performou melhor em Especificidade e em F1-Score

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

MARLON MATEUS PRUDENTE DE OLIVEIRA

ESTUDO DE CASO: IMPLICAÇÕES ÉTICAS DO USO DO CHATGPT

Artigo apresentado como requisito parcial à conclusão da disciplina de Aspectos Filosóficos e Éticos da IA, do curso de Especialização em Inteligência Artificial Aplicada Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Professor: Prof. Dr. Marcos Wachowicz

CURITIBA

2024

RESUMO

Este trabalho aborda os diversos impactos e implicações éticas do uso do ChatGPT em diferentes contextos contemporâneos. Inicialmente, discute-se como a inteligência artificial está reconfigurando processos educacionais, substituindo métodos tradicionais de pesquisa e redação por ferramentas automatizadas, o que desafia os educadores a adaptarem suas metodologias de avaliação para garantir a autenticidade do aprendizado. Além da educação, são exploradas as potenciais melhorias no diagnóstico médico com a integração da IA, destacando-se a necessidade de aperfeiçoamento dos algoritmos e medidas robustas de segurança de dados para assegurar sua implementação ética. No setor jurídico, o artigo examina como o ChatGPT pode aumentar a produtividade na elaboração de documentos legais, embora enfrente desafios como a garantia de precisão e a proteção da privacidade. Por fim, no entretenimento, são levantadas questões sobre o uso ético de dados na criação de conteúdos gerados por IA, enfatizando a necessidade de regulamentação e compensação adequada para o trabalho produzido por essas ferramentas. Em síntese, a IA representa um avanço significativo com potencial transformador, desde que seu uso seja orientado por princípios éticos que preservem a integridade e responsabilidade humana.

Palavras-chave: ChatGPT. Inteligência artificial. Educação. Saúde. Sistema legal. Entretenimento. Ética. Adaptação educacional. Diagnóstico médico. Privacidade de dados.

ABSTRACT

This paper addresses the various impacts and ethical implications of using ChatGPT in different contemporary contexts. Initially, it discusses how artificial intelligence is reconfiguring educational processes, replacing traditional research and writing methods with automated tools, which challenges educators to adapt their assessment methodologies to ensure the authenticity of learning. In addition to education, the potential improvements in medical diagnosis with AI integration are explored, highlighting the need for algorithm refinement and robust data security measures to ensure ethical implementation. In the legal sector, the article examines how ChatGPT can increase productivity in legal document drafting, although facing challenges such as ensuring accuracy and protecting privacy. Finally, in entertainment, questions are raised about the ethical use of data in creating AI-generated content, emphasizing the need for regulation and appropriate compensation for work produced by these tools. In summary, AI represents a significant advancement with transformative potential, provided its use is guided by ethical principles that preserve human integrity and responsibility.

Keywords: ChatGPT. Artificial intelligence. Education. Health. Legal system. Entertainment. Ethics. Educational adaptation. Medical diagnosis. Data privacy.

1. INTRODUÇÃO

No cenário atual, é possível observar muitos trabalhos acadêmicos, notícias e artigos com a temática dos diversos usos do ChatGPT, muitos deles levantando questões relacionadas a mudanças fundamentais em processos de trabalho e aprendizado, suas vantagens, desvantagens e implicações éticas nas relações humano-máquina.

Na educação, por exemplo, o ChatGPT tem transformado a maneira como os alunos realizam pesquisas e produzem trabalhos, substituindo a necessidade de consultar diversas fontes e elaborar textos de forma independente. Isso apresenta um desafio para os educadores, que precisam encontrar maneiras diferentes de avaliar se os alunos realmente compreendem o conteúdo, como através de apresentações orais e escrita manual. A adaptação das metodologias de ensino e avaliação é crucial para garantir que o aprendizado não seja prejudicado pelo uso da IA.

Além de seus impactos na educação, este trabalho abordará as implicações éticas do uso do ChatGPT em contextos como Saúde, Sistema Legal e Entretenimento. Exploraremos como essa ferramenta pode ser integrada de maneira ética e responsável, contribuindo positivamente para essas áreas sem comprometer a autonomia e a integridade dos usuários.

1.1. CONTEXTO 1: EDUCAÇÃO

Os impactos do ChatGPT na educação são significativos. Anteriormente, para fazer um trabalho escolar, os alunos precisavam realizar suas pesquisas no Google e, em seguida, ler e organizar a pesquisa em um texto. O professor se preocupava principalmente com o "copiar e colar" sem as devidas citações e referências. Mesmo sem dar os devidos créditos às fontes, o professor tinha certeza de que o aluno havia lido e desenvolvido o trabalho, garantindo que ele tivesse aprendido algo nesse esforço.

Com o surgimento do ChatGPT, esse trabalho de consultar diversas fontes para escrever deixa de ser realizado pelo aluno, dificultando assim, a avaliação do aprendizado pelo professor. O professor precisa garantir que o aluno aprendeu o conteúdo ensinado na matéria, e faz isso por meio de avaliações diversas, como provas escritas, trabalhos, seminários, etc. Todos os meios de avaliação possuem falhas. Uma prova escrita, por exemplo, não garante que o aluno que tirou uma nota

baixa não saiba o conteúdo; ele pode não ter entendido a questão ou estar em um dia ruim.

Com a chegada do ChatGPT, alguns tipos de avaliação precisarão ser adaptados para garantir que o aluno absorveu o conteúdo explicado. O ChatGPT é uma ferramenta e não deve ser banida da academia. Deve-se apenas instruir os alunos sobre como utilizá-la da melhor forma. Alguns professores têm adotado mais avaliações manuscritas como forma de driblar o uso do ChatGPT. Karine Karsten, professora doutoranda em Educação na UFPR, diz: “Com a chegada das inteligências artificiais como o ChatGPT, os trabalhos de pesquisa por si só não serão suficientes para avaliar o aluno. Acredito que será necessário complementar o trabalho escrito com uma apresentação oral do que foi feito ou outras formas de complementar as avaliações, como a escrita em sala de um resumo do que havia no trabalho. Assim, mesmo que o trabalho tenha sido desenvolvido utilizando IA, será possível avaliar se o aluno leu e entendeu o que escreveu. Portanto, se o aluno fez ou não o trabalho utilizando IA não seria relevante, desde que ele tenha se apropriado do conteúdo, das competências e habilidades que eram o objetivo ao se propor um trabalho.”

Ronaldo Lemos, professor da Schwarzman College, passou a exigir que seus alunos sempre utilizassem o ChatGPT para os trabalhos. No entanto, em vez de avaliar as respostas, ele passou a avaliar as perguntas (ou "prompts") feitas para a ferramenta.

Na área de humanas, para obter uma resposta mais detalhada, é necessário fazer perguntas detalhadas sobre o tema. Perguntas óbvias geram respostas óbvias. Fazer as perguntas certas está se tornando uma forma de programação, mais acessível a pessoas sem conhecimento técnico, prompts complexos são necessários se o que se espera são resultados complexos nas respostas do ChatGPT.

1.2. CONTEXTO 2: SAÚDE

A IA no diagnóstico médico promete melhorar a precisão, eficiência e velocidade dos diagnósticos, beneficiando pacientes e profissionais de saúde. No entanto, enfrenta desafios como a necessidade de aperfeiçoamento dos algoritmos para melhor precisão, garantia da privacidade dos dados e superação de barreiras culturais para a utilização da ferramenta. A aceitação e uso ético da IA são cruciais,

e, por isso, até a OMS sugere princípios para assegurar o uso responsável, promovendo autonomia, bem-estar, transparência e sustentabilidade da tecnologia.

Esses princípios visam, principalmente, garantir que a IA não substitua completamente o julgamento humano, mas sim seja usada como uma ferramenta de auxílio nos diagnósticos. Eles asseguram que a IA beneficie a saúde pública de maneira segura e eficiente.

O uso da IA ainda é tímido em muitas instituições de saúde, porém, em alguns locais, já se utilizam algoritmos que facilitam o reconhecimento de padrões em imagens médicas, principalmente para exames como tomografias e ressonâncias, além de soluções para áreas administrativas.

Um dos desafios para utilizar eficazmente a IA é fornecer todos os dados necessários para que ela possa fazer qualquer tipo de diagnóstico ou sugestão. Muitos sistemas hospitalares são distribuídos e seus dados não se comunicam entre si. Portanto, criar uma forma centralizada de coleta de dados dos sistemas e disponibilização para uma IA é essencial. Medidas de segurança e privacidade devem ser aplicadas no local centralizado para evitar quaisquer tipos de incidentes que possam vazar dados de pacientes.

Assegurar a segurança dos sistemas que armazenam informações de pacientes envolve a implementação de diversas medidas essenciais. A criptografia transforma os dados em formatos ilegíveis sem a chave adequada, garantindo confidencialidade e integridade. A redundância cria cópias dos dados em vários locais para mitigar riscos de perda e permitir recuperação eficiente em incidentes. A conformidade com LGPD ou GDPR assegura que empresas sigam diretrizes rigorosas, fortalecendo a confiança dos clientes ao proteger a privacidade e os direitos individuais. A detecção e correção de brechas e vulnerabilidades identifica e resolve falhas de segurança, protegendo dados sensíveis e garantindo conformidade com regulamentações de proteção de dados.

Outro fator muito relevante a ser considerado é a possível desumanização da relação médico-paciente proporcionada pelo uso da IA. Equilibrar o uso desta ferramenta com a necessidade de manter a empatia e o cuidado humanizado na relação médico-paciente é de suma importância. A implementação bem-sucedida da IA na medicina deve sempre considerar o aspecto humano, garantindo que o paciente continue sendo o foco central do cuidado, evitando um formato excessivamente mecânico de diagnósticos.

Outro aspecto importante é que o uso da IA no contexto médico, sem uma auditoria ou verificação clara dos algoritmos, pode propagar os chamados vieses de algoritmos que podem gerar discriminações por etnia, gênero, idade, entre outras características. Dessa forma, a supervisão humana dos algoritmos deve fazer parte do processo de implementação e verificação periódica.

Também é relevante discutir a responsabilidade do médico quando obtém um diagnóstico utilizando uma ferramenta de IA. Juridicamente, o profissional tem a palavra final em qualquer decisão de diagnóstico e tratamento. Dessa forma, a argumentação de que uma IA tomou uma má decisão não exime o médico de responsabilização por danos causados a pacientes em determinados tratamentos ou pela falta deles.

1.3. CONTEXTO 3: JUSTIÇA E SISTEMA LEGAL

Na área jurídica, o ChatGPT vem causando uma revolução na produtividade. A elaboração de documentos jurídicos de baixa e média complexidade e o gerenciamento de documentos permitem aos advogados coletar e analisar grandes quantidades de dados jurídicos com rapidez e precisão. Procedimentos que levavam horas ou dias, envolvendo várias pessoas, agora podem ser feitos em questão de minutos ou segundos por uma única pessoa que faz a solicitação correta. As respostas podem ser usadas para apoiar argumentos no tribunal, gerar aconselhamento jurídico econômico ou preparar para o julgamento.

No entanto, toda essa comodidade e praticidade cobra seu preço: o ChatGPT, como outras inteligências artificiais, é passível de “alucinações”, isto é, criar conteúdo incorreto ou indicar jurisprudências inexistentes. A mesma característica que permite respostas tão personalizadas tem a capacidade de fabricar fatos e produzir fontes sem ter acesso a dados suficientes.

Outro problema são os erros que ele comete ao transcrever os artigos das leis. Como essa solução é baseada em um grande conjunto de dados contidos no GPT-3, ela tende a reproduzir as informações e os vieses ali encontrados. Sua base de dados, na versão gratuita, vai até setembro de 2021, então todas as atualizações e mudanças na lei após essa data não foram incluídas, o que pode causar problemas.

Muitos juristas ainda se preocupam com a privacidade de seus dados e de seus clientes, pois as perguntas feitas para o ChatGPT são armazenadas e usadas

em seu treinamento constante, o que pode ocasionar uma quebra de privacidade se algum viés resgatar os dados pessoais que o usuário passou sem perceber.

Para além da especulação que possamos fazer sobre os problemas que o ChatGPT pode causar no meio jurídico, já começam a aparecer os primeiros casos de mau uso dessa ferramenta:

Nos Estados Unidos, o advogado Steven Schwartz, com mais de 30 anos de experiência, fez sua pesquisa jurídica de casos anteriores usando o ChatGPT e se baseou em suas respostas para redigir sua petição sobre um caso contra uma companhia aérea. O problema é que o ChatGPT criou casos de jurisprudência que simplesmente não existiam. O juiz do caso, percebendo o problema, puniu o advogado com uma multa de 5 mil dólares e sanções.

No Brasil, um advogado foi multado em R\$ 2.400 depois de protocolar uma petição redigida pelo programa ChatGPT. Além disso, um juiz federal do TRF da 1ª região, que assinou sentença feita por meio de inteligência artificial, será investigado pelo CNJ. A razão para essa investigação reside no fato de a ferramenta ter criado jurisprudências do STJ que fundamentaram a decisão do magistrado.

1.4. CONTEXTO 4: ENTRETENIMENTO

Primeiro, a implicação de que não é ético usar dados que, mesmo estando disponíveis online, não estejam disponíveis de maneira legal. Isso inclui dados oriundos de pirataria ou que estejam acessíveis apenas mediante compensação monetária. Utilizar tais dados sem a devida autorização ou pagamento pode infringir direitos autorais e éticos.

Segundo, surge a questão de quem deve ser compensado pelo que é produzido por meio do ChatGPT. Se uma tarefa cumprida pela IA atinge um alto nível de qualidade e pode substituir o trabalho humano, quem deveria receber a compensação financeira por esse trabalho? Devemos considerar o papel do desenvolvedor da IA, do usuário que fez o prompt e de qualquer outro colaborador humano no processo, inclusive, os autores do conteúdo usado no treinamento.

Terceiro, é necessário discutir sobre quem recai a responsabilidade pelo uso final do que é gerado pelo ChatGPT para fins ilegais e imorais. Se uma obra gerada pela IA é utilizada de maneira prejudicial, quem deve ser responsabilizado? O criador da IA, o usuário ou ambos?

Um caso real que ilustra esses dilemas é o do ganhador do prêmio de produção artística cuja obra “Théâtre D’opéra Spatial” foi o produto de um híbrido entre o que foi feito por uma IA e o que foi editado posteriormente pelo artista. Este, teve seu pedido de direitos autorais negado, destacando a complexidade de se definir autoria e direitos em criações híbridas entre humanos e IA.

2. METODOLOGIA

O método utilizado neste trabalho foi a revisão de literatura de artigos disponíveis na internet fora do âmbito acadêmico, mas elaborados por profissionais dos diferentes contextos abordados. Esta abordagem foi escolhida para melhor demonstrar a percepção das implicações éticas pelos profissionais que, de alguma forma, se sentem prejudicados ou são a favor do uso da ferramenta nos processos de trabalho.

Nesses artigos, buscamos identificar o cenário atual de cada contexto abordado, avaliar se a percepção é favorável ao uso do ChatGPT e fornecer um panorama de informações suficiente para discussão dos dilemas éticos e propor possíveis soluções.

3. APRESENTAÇÃO DOS RESULTADOS

O quadro abaixo apresenta os resultados observados na síntese dos artigos utilizados como fonte deste trabalho.

QUADRO 1 – Percepções, soluções e principais dilemas éticos de cada contexto

Contexto	Percepção	Dilemas Éticos	Possíveis Soluções
Educação	Favorável com restrições	4. Autenticidade do aprendizado 5. Qualidade da Avaliação 6. Uso adequado da ferramenta 7. Privacidade e segurança	8. Utilizar formas diferentes de avaliação 9. Disponibilização de Treinamento/Cursos 10. Regulamentação
Saúde	Favorável	• Necessidade de aperfeiçoamento dos Algoritmos • Garantia da Privacidade dos Dados • Barreiras Culturais	11. Transparência nos algoritmos usados. 12. Uso da IA como auxílio e não substituição de diagnóstico final

Contexto	Percepção	Dilemas Éticos	Possíveis Soluções
		• Desumanização da Relação Médico-Paciente • Vieses de Algoritmos • Responsabilidade Médica	13. Conscientização da possível agilidade adicional que a IA pode trazer no diagnóstico médico
Justiça e Sistema Legal	Favorável com ressalvas	• Precisão e confiabilidade • Privacidade dos Dados • Responsabilidade Legal	14. Sempre revisar com cautela, textos gerados. 15. Criar e usar chatbots especializados e atualizados para consulta. 16. Regulamentação de boas práticas no uso.
Entretenimento	Crítica com restrições	• Uso de dados ilegais ou não autorizados • Compensação financeira • Responsabilidade pelo uso para fins ilegais ou imorais • Direitos autorais e criação híbrida	17. Inibição do acesso ilegal a dados para treinamento de IAs 18. Definição legal de limites de posse para compensação monetária 19. Maior fiscalização e melhor rastreabilidade de artefatos gerados por IA 20. Definição dos limites de direitos autorais de artefatos gerados por IA que sofrem intervenção humana

FONTE: Autoria própria (2024).

4. CONSIDERAÇÕES FINAIS

Na era atual, comparável a marcos tecnológicos como a Revolução Industrial e a Era dos Computadores, a inteligência artificial (IA) emerge como uma nova fronteira. Assim como ocorreu na Revolução Industrial, surgem debates sobre o impacto da IA no emprego, destacando-se a necessidade de adaptação a novos paradigmas profissionais.

Como qualquer tecnologia, a IA possui vantagens e desvantagens. Educadores, alunos e a sociedade em geral devem aprender a utilizar essas novas ferramentas de maneira eficaz.

Na área da saúde, a integração da IA promete melhorias significativas na precisão e eficiência dos diagnósticos. No entanto, é crucial implementar medidas robustas de segurança de dados e garantir conformidade com regulamentações como a LGPD. É essencial também que a IA complemente, e não substitua, o julgamento humano, preservando a empatia na relação médico-paciente.

No meio jurídico, a IA, como o ChatGPT, deve ser utilizada para tarefas de baixa complexidade, auxiliando os juristas. A responsabilidade final e a validação dos resultados devem sempre permanecer com os especialistas, garantindo a qualidade e a precisão das decisões.

No campo do entretenimento, a IA pode ser utilizada ética e criativamente, desde que o acesso aos dados para treinamento seja legal e consentido. Ela complementa, mas não substitui, a criatividade humana na produção de artes gráficas, textos e outros conteúdos.

Em resumo, a IA representa uma evolução tecnológica com potencial transformador em diversas áreas, desde que seja adotada de maneira ética e responsável, preservando sempre o papel central da inteligência humana e sua criatividade.

REFERÊNCIAS

CASSOL, Daniel. **Quais os impactos do ChatGPT e da inteligência artificial na educação?** Disponível em: <<https://www.ifsc.edu.br/web/ifsc-verifica/w/quais-os-impactos-do-chatgpt-e-da-inteligencia-artificial-na-educacao-#:~:text=%22O%20ChatGPT%20p%C3%B5e%20em%20xeque,obter%20com%20uma%20boa%20aula>>. Acesso em: 25 jun. 2024.

LEMONS, Ronaldo. **ChatGPT e a arte de fazer perguntas.** Disponível em <<https://www1.folha.uol.com.br/colunas/ronaldolemons/2023/02/chatgpt-e-a-arte-de-fazer-perguntas.shtml>>. Acesso em: 25 jun. 2024.

MEDICINA DIRETA. **Inteligência artificial no diagnóstico médico.** Disponível em: <<https://medicinadireta.com.br/blog/inteligencia-artificial-no-diagnostico-medico/>>. Acesso em: 02 jun. 2024.

BALERINI, Cristina. **Uso da inteligência artificial na saúde: resultados e desafios.** Disponível em: <<https://www.saudebusiness.com/ti-e-inovacao/uso-da-inteligencia-artificial-na-saude-resultados-e-desafios>>. Acesso em: 17 jun. 2024.

NEURALMED. **Proteção de dados na saúde.** Disponível em: <<https://www.neuralmed.ai/blog/protecao-de-dados-na-saude>>. Acesso em: 01 jun. 2024.

FORTI, Jefferson K.. **A inteligência artificial como uma nova disciplina em medicina.** Disponível em: <<https://academiamedica.com.br/blog/a-inteligencia-artificial-como-uma-nova-disciplina-em-medicina>>. Acesso em: 05 jun. 2024.

RIPLA, Andre. **AI and dehumanization in interactions: Exploring an ethical case.** Disponível em: <<https://www.linkedin.com/pulse/ai-dehumanization-interactions-exploring-ethical-case-ripla-pgcert-bnlye>>. Acesso em: 25 jun. 2024.

ARAUJO, Jailson S.; HORNUNG, Jociane A. **Inteligência artificial e suas implicações éticas.** Disponível em: <<https://revistathemis.tjce.jus.br/THEMIS/article/view/899/pdf>>. Acesso em: 25 jun. 2024.

MELLO, Victor. **Chat GPT na Advocacia.** Disponível em: <<https://www.migalhas.com.br/depeso/385996/chat-gpt-na-advocacia>>. Acesso em: 23 jun. 2024.

MODELO INICIAL. **Chat GPT na advocacia: entenda os riscos e funcionalidades.** Disponível em: <<https://modeloinicial.com.br/artigos/chat-gpt>>. Acesso em: 23 jun. 2024.

Redação. **Lee, Brock & Camargo Advogados restringe uso de ChatGPT e IA externas.** Disponível em: <<https://www.migalhas.com.br/quentes/386994/lee-brock-camargo-advogados-restringe-uso-de-chatgpt-e-ia-externas>>. Acesso em: 23 jun. 2024.

VIEIRA, Rodrigo. **Advogado é multado pelo TSE por usar ChatGPT em petição.** Disponível em: <<https://www.jornaljurid.com.br/noticias/advogado-e-multado-pelo-tse-por-usar-chat-gpt-em-peticao>>. Acesso em: 23 jun. 2024.

FABRO, Clara. **Advogado usa ChatGPT para fazer pesquisa jurídica e é multado em R\$ 2,4 mil.** Disponível em: <<https://www.techtudo.com.br/noticias/2023/07/advogado-usa-chatgpt-para-fazer-pesquisa-juridica-e-e-multado-em-r-24-mil-entenda-edviralizou.ghml>>. Acesso em: 23 jun. 2024.

STRECK, Lenio. **ChatGPT inventando precedente e a terceirização da justiça**. Disponível em: <<https://www.conjur.com.br/2023-nov-16/chatgpt-inventando-precedente-e-a-terceirizacao-da-justica/>>. Acesso em: 23 jun. 2024.

PIGNATI, Giovana. **ChatGPT: escritório de advocacia usa IA para desenvolver contrato**. Disponível em: <https://www.terra.com.br/byte/chatgpt-escritorio-de-advocacia-usa-ia-para-desenvolver-contrato_da814011f6a94389521dc6999824a98421z4mv6x.html>. Acesso em: 23 jun.2024.

REDAÇÃO G1: **Empresa que criou primeiro advogado robô do mundo é processada em ação coletiva**. Disponível em: <<https://g1.globo.com/tecnologia/noticia/2023/03/13/empresa-que-criou-primeiro-advogado-robo-do-mundo-e-processada-em-acao-coletiva.ghtml>>. Acesso em: 23 jun.2024.

CHESTERMAN, Simon. **Good models borrow, great models steal: intellectual property rights and generative AI**. Disponível em: <<https://doi.org/10.1093/polsoc/puae006>>. Acesso em: 18 jun. 2024.

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

Artigo escolhido: Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows

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?
Entender como a automação de aprendizado de máquina está sendo usada na prática atualmente. Através de uma pesquisa qualitativa, apresentar pontos positivos e negativos das ferramentas atuais, e também, apresentar as respectivas funções do ser humano e das ferramentas no processo de Machine Learning.	Necessidade de tornar o Aprendizado de Máquina mais acessível mundialmente, oportunidade de verificar como as ferramentas atuais são utilizadas na prática, identificar o papel do desenvolvedor e das ferramentas de automação no processo de desenvolvimento de IA.	Pesquisa qualitativa, entrevistando 16 pessoas que trabalham com as ferramentas de IA, desde pessoas que utilizam como hobby, até pesquisadores de grandes indústrias.	As ferramentas de automação de ML conseguem abstrair grande parte da complexidade do desenvolvimento de IA, tornando mais fácil o desenvolvimento. Conclui-se também que não é o ideal tirar o humano do processo de desenvolvimento, deixando tudo 100% automatizado, pois é necessário uma visão e acompanhamento humano no processo de desenvolvimento, para garantir confiança, entendimento, e que respeite as questões éticas e legais. O autor defende ferramentas que trabalhe em conjunto com o ser humano no desenvolvimento, não que o substitua.

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_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 (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) Link:

<https://colab.research.google.com/drive/1WpA882AldVOMTq4mmx6x8XZnUSho7xT>

Breve explicação dos resultados:

- Gráficos de perda e de acurácia;

Em um teste com 10 épocas, na sexta época percebemos que os dados de validação estabilizam com perda perto de 0.33 e acurácia perto de 0.88.

Níveis relativamente bons pela quantidade de épocas de treinamento.

- Imagem gerada na seção “Mostrar algumas classificações erradas”, apresentada na aula prática. Nas classificações erradas temos uma bota de cano curto que o modelo classificou como uma sandália.

2 Regressão (RNA)

Link:

<https://colab.research.google.com/drive/1T9iMqagZjfwTwz3B3anBfbjwNpccniU?usp=sharing>

Breve explicação dos resultados:

- Gráficos de avaliação do modelo (loss):

Utilizando o método “Early stop”, verificamos que o modelo estabiliza com valores muito bons na quadragésima oitava época com os dados de validação com perda perto de 0.0078.

- Métricas de avaliação do modelo (pelo menos uma entre MAE, MSE, R^2). A princípio usando os dados brutos, o modelo apresentou um desempenho ruim, depois de normalizarmos os dados de Score obtivemos resultados satisfatórios com:

mse = 0.004

rmse = 0.0686

r2 = 0.8442

3 Sistemas de Recomendação

Link:

https://colab.research.google.com/drive/1xolO6ZAmePNYL8jTc8ftMIGy_BT08zx_r?usp=sharing

Breve explicação dos resultados:

- Gráficos de avaliação do modelo (loss);

Em um teste com 100 épocas, com o parâmetro de perda utilizando os dados de validação ficam entre 8.4 tendendo para 8.6 o que nos leva a inferir que mais épocas de treinamento não vão melhorar o modelo

- Exemplo de recomendação de livro para determinado Usuário.

Recomendação para o usuário 276729:

Livro - 373293062 / 4.967702 *

Nome: High Country Hero (Harlequin Historical Series)

4 Deepdream

Link:

<https://colab.research.google.com/drive/1li2KkCGGlw13LFEUBFtlo4e8J0-froZb>

Breve explicação dos resultados:

- Imagem onírica obtida por Main Loop;

O padrão visual do gatinho gerado pela rede neural se destaca com detalhes interessantes e a neve desaparece completamente

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

Levando o modelo até oitava, verificamos que os padrões oníricos se intensificam e realmente parece mais um “gatinho dos sonhos”

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

Percebemos que tratando a imagem e elevando a uma oitava, o gatinho começa a ficar indistinguível parecendo até ser uma raposa.

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 público-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

The background is a solid purple color with various colorful geometric shapes (circles, ovals, and elongated rectangles) in shades of blue, green, yellow, and red scattered across it.

IAA007

Visualização de Dados e Storytelling

EVOLUÇÃO DO ACESSO A
TECNOLOGIA NOS LARES DE
MINAS GERAIS.

MARLON MATEUS PRUDENTE DE OLIVEIRA

Dados Brutos

7 - Domicílios
Tabela 7.7 - Domicílios particulares permanentes e moradores em domicílios particulares permanentes, por classe de rendimento mensal domiciliar, segundo a existência de microcomputador, o acesso à Internet e o tipo de telefone - Minas Gerais - 2013

Existência de microcomputador, acesso à Internet e tipo de telefone	Domicílios particulares permanentes (1 000 domicílios)				Moradores em domicílios particulares permanentes (1 000 pessoas) (1)			
	Total (2)	Classes de rendimento mensal domiciliar (salário mínimo)			Total (3)	Classes de rendimento mensal domiciliar (salário mínimo)		
		Até 10	Mais de 10 a 20	Mais de 20		Até 10	Mais de 10 a 20	Mais de 20
Total	6 818	6 110	326	119	20 569	18 372	1 078	377
Microcomputador								
Tinham	3 440	2 880	306	116	11 279	9 433	1 020	369
Com acesso à Internet	2 949	2 420	298	113	9 627	7 880	997	361
Não tinham	3 377	3 231	20	4	9 291	8 939	58	8
Telefone								
Tinham	6 270	5 589	325	119	19 331	17 175	1 077	377
Somente celular	3 798	3 574	83	16	11 738	11 091	258	48
Somente fixo convencional	188	177	3	1	379	354	7	2
Celular e fixo convencional	2 284	1 837	239	102	7 214	5 730	813	327
Não tinham	547	522	1	-	1 239	1 197	1	-

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2013.
(1) Exclui os moradores cuja a condição no domicílio era pensionista, empregado doméstico ou parente do empregado doméstico.
(2) Inclusive os domicílios sem declaração de rendimento, sem rendimento ou cujos moradores recebiam somente em benefícios.
(3) Inclusive os moradores em domicílios sem declaração de rendimento, sem rendimento ou cujos moradores recebiam somente em benefícios.

7 - Domicílios

Existência de microcomputador, acesso à Internet e tipo de telefone	Domicílios particulares permanentes (1 000 domicílios)				Moradores em domicílios particulares permanentes (1 000 pessoas) (1)			
	Total (2)	Classes de rendimento mensal domiciliar (salário mínimo)			Total (3)	Classes de rendimento mensal domiciliar (salário mínimo)		
		Até 10	Mais de 10 a 20	Mais de 20		Até 10	Mais de 10 a 20	Mais de 20
Total	6 965	6 328	351	121	20 718	18 692	1 183	368
Microcomputador								
Tinham	3 469	2 941	325	115	11 220	9 462	1 111	356
Com acesso à Internet	2 939	2 438	312	112	9 469	7 795	1 066	352
Não tinham	3 495	3 387	26	5	9 497	9 231	72	12
Telefone								
Tinham	6 538	5 910	351	121	19 854	17 844	1 183	368
Somente celular	4 163	3 965	97	21	12 738	12 128	328	53
Somente fixo convencional	181	170	2	1	323	302	7	1
Celular e fixo convencional	2 195	1 774	252	99	6 793	5 415	848	313
Não tinham	427	418	-	-	864	848	-	-

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2014.

7 - Domicílios

Existência de microcomputador, acesso à Internet e tipo de telefone	Domicílios particulares permanentes (1 000 domicílios)				Moradores em domicílios particulares permanentes (1 000 pessoas) (1)			
	Total (2)	Classes de rendimento mensal domiciliar (salário mínimo)			Total (3)	Classes de rendimento mensal domiciliar (salário mínimo)		
		Até 10	Mais de 10 a 20	Mais de 20		Até 10	Mais de 10 a 20	Mais de 20
Total	7 057	6 499	340	114	20 855	19 168	1 059	373
Microcomputador								
Tinham	3 410	2 934	315	109	10 950	9 438	1 000	359
Com acesso à Internet	2 917	2 459	307	105	9 356	7 894	979	349
Não tinham	3 647	3 565	25	5	9 905	9 730	59	15
Telefone								
Tinham	6 628	6 078	340	114	19 925	18 249	1 059	373
Somente celular	4 264	4 104	87	22	12 951	12 511	254	70
Somente fixo convencional	176	172	1	-	293	287	2	-
Celular e fixo convencional	2 189	1 802	253	91	6 680	5 451	803	303
Não tinham	429	421	-	-	931	919	-	-

Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Trabalho e Rendimento, Pesquisa Nacional por Amostra de Domicílios 2015

(1) Excluídos os moradores cuja a condição no domicílio era pensionista, empregado doméstico ou parente do empregado doméstico

(2) Inclusive os domicílios sem declaração de rendimento, sem rendimento ou cujos moradores recebiam somente em benefícios

(3) Inclusive os moradores em domicílios sem declaração de rendimento, sem rendimento ou cujos moradores recebiam somente em benefícios.



Contexto e público-alvo

- Apresentar os dados do IBGE relacionados aos lares do estado de Minas Gerais nos anos de 2013 à 2015;
- Mostrar a evolução da escolha da população brasileira;
- Apresentação destinada à alunos da matéria de redes e comunicação sem fio de Eng. De Computação.

Sobre a visualização de dados

- Foi escolhido apresentar os dados em gráfico de barras, pois é mais fácil para o público comparar os valores apresentados;
- Foi escolhido o Excel, pois os dados do IBGE já vem nesse formato, então foi mais fácil fazer os gráficos pelo próprio Excel.



Telefones e Computadores

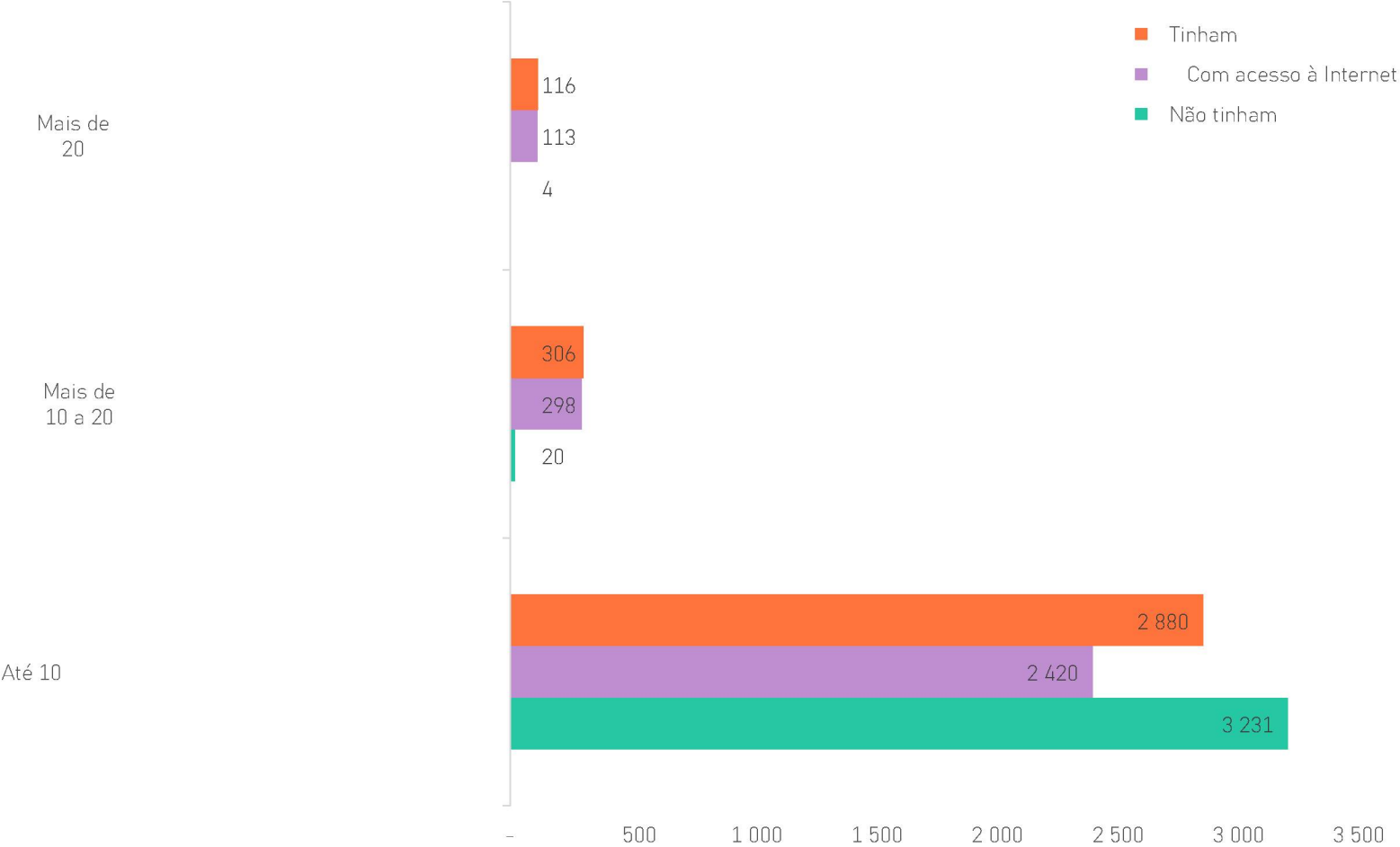
- Em 1877, D. Pedro II instalou as primeiras redes telefônicas no Brasil;
- Já o acesso à Internet, só em 1988 por meio de uma parceria entre a academia e o Laboratório Nacional de Computação Científica (LNCC);
- De lá pra cá, essas tecnologias foram se popularizando e chegando aos lares da população Brasileira.



O que se espera do acesso da população brasileira à Telefones e Computadores?

- Com a popularização dessas tecnologia, é esperado que mais e mais lares tenham acesso à computadores com internet, telefones e celulares.

Microcomputador – Minas Gerais – 2013





2013

- Observando o gráfico com os dados do IBGE relacionados a lares com renda de até 10 salários mínimos, podemos observar que, em 2013, 3.231 M de lares não tinham computador em suas casas, 2.880 M de lares tinham computador, mas sem acesso à internet, e apenas 2.420 M de lares tinham computador com acesso à internet.

Evolução

- É espantoso imaginar lares sem computador, ou mesmo com computador e sem acesso a internet hoje em dia, utilizamos a internet para tudo, desde pagar contas até mesmo fazer compras;
- O gráfico também mostra a diferença de lares com renda de 10 à 20 salários mínimos, e também lares acima de 20 salários mínimos, nessa população, observa-se o inverso, são poucos lares sem computador;
- O salário mínimo em 2013 era de R\$ 678,00, até 10 SM seria equivalente à R\$ 6.780,00
- O trabalho focará nos lares com renda de até 10 salários mínimos, onde possuo lugar de fala.

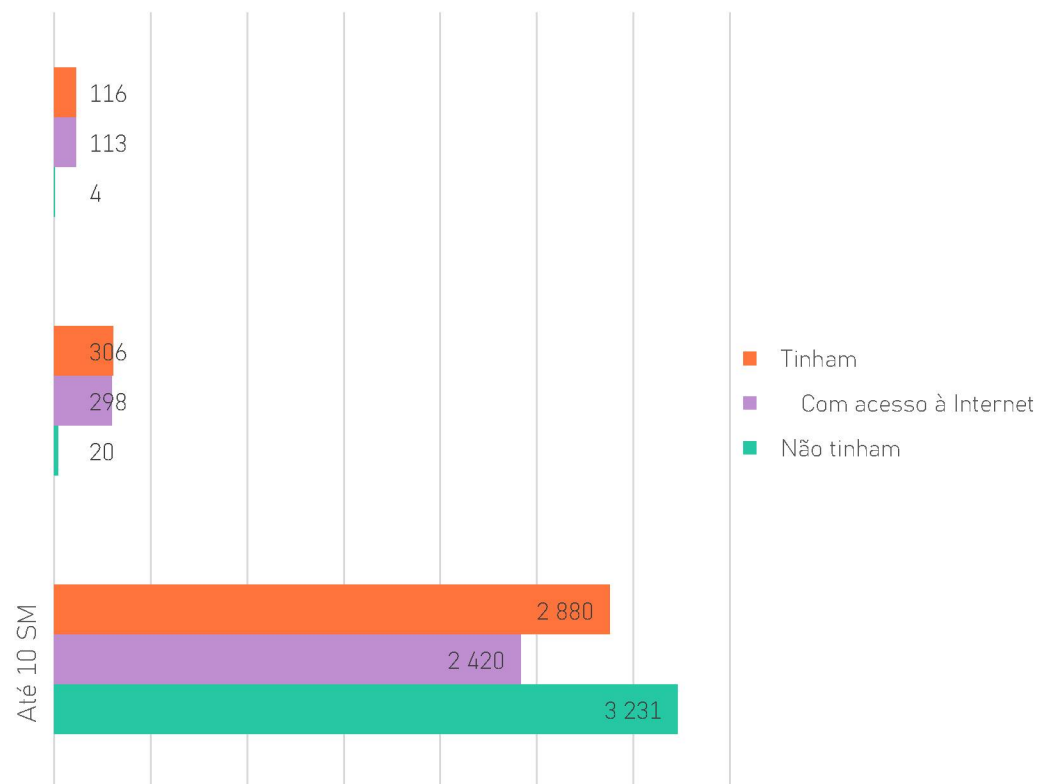


2014

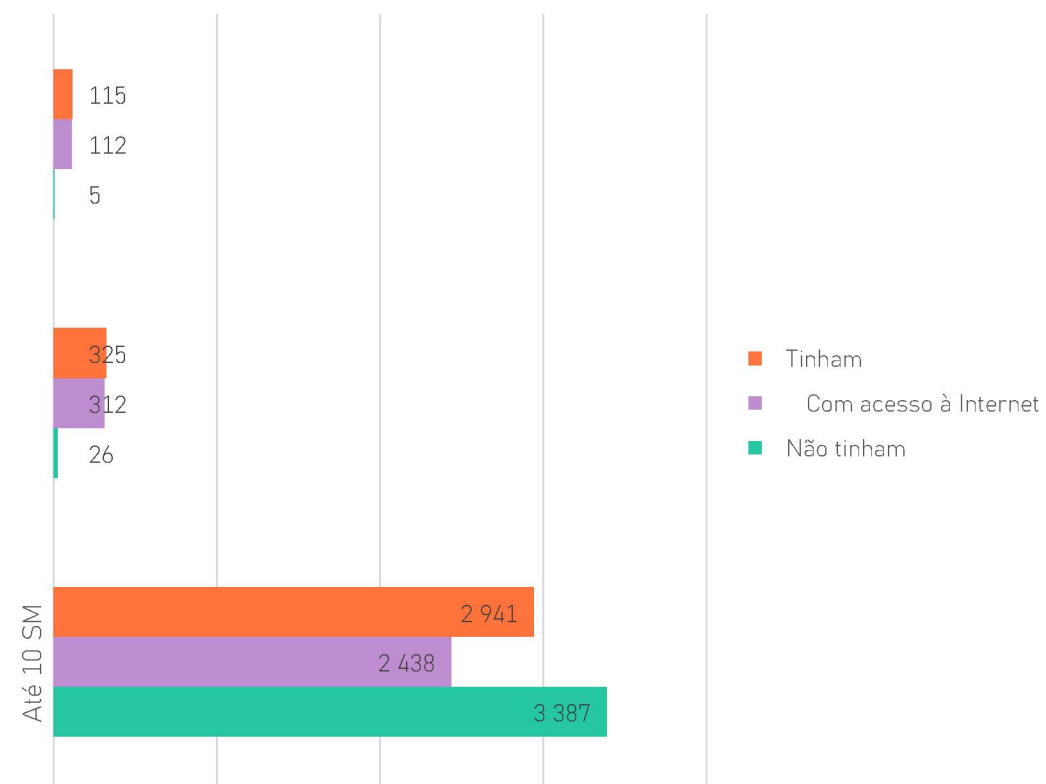
- ¿Qué podemos esperar de los datos de 2014?



Microcomputador – Minas Gerais – 2013



Microcomputador – Minas Gerais – 2014





2013 x 2014

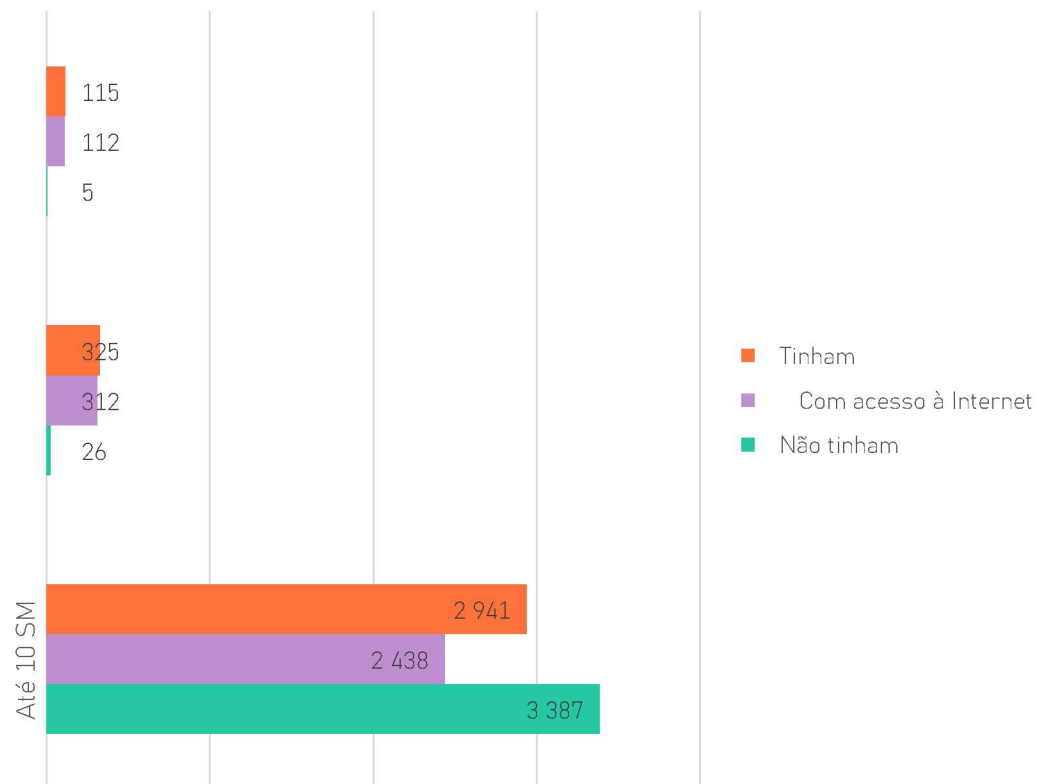
- Pode-se observar que o número de lares com computador diminuiu em 2014;
- O esperado seria a diminuição da quantidade de lares sem computador e um aumento significativo de lares com acesso à computador com internet, mas não foi o observado;
- Houve um aumento do número de lares com computador sem acesso à internet.



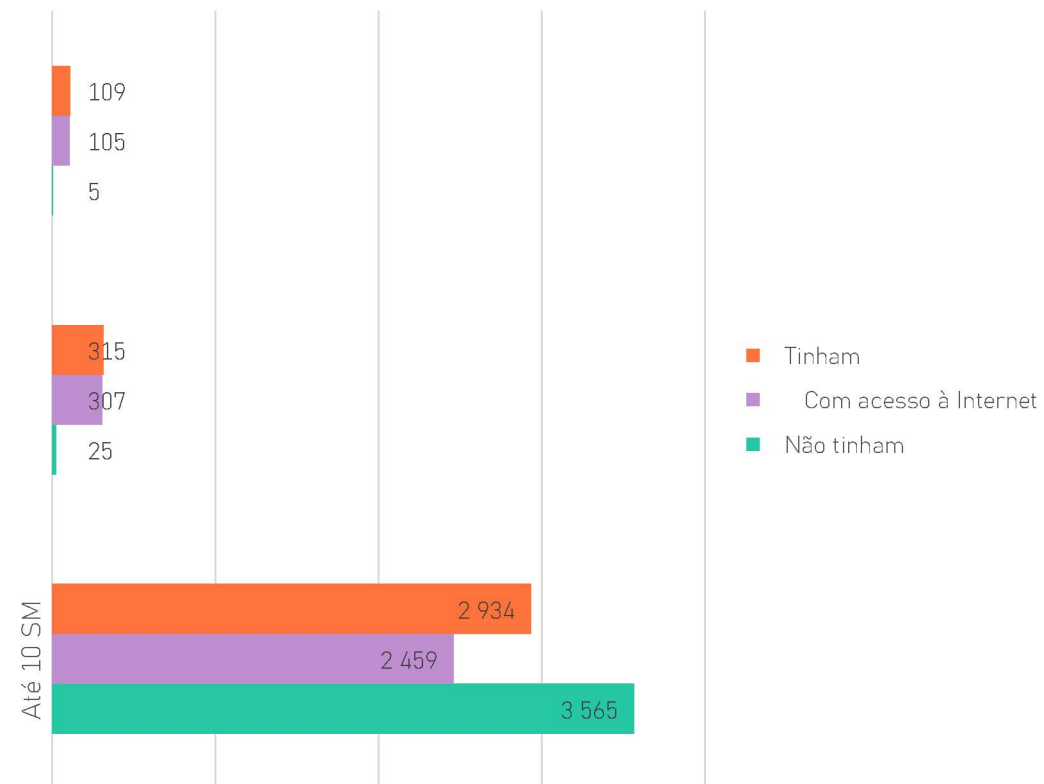
2015

- O que podemos esperar dos dados de 2015?
- Será que essa tendência continuará?

Microcomputador – Minas Gerais – 2014



Microcomputador – Minas Gerais – 2015





2014 x 2015

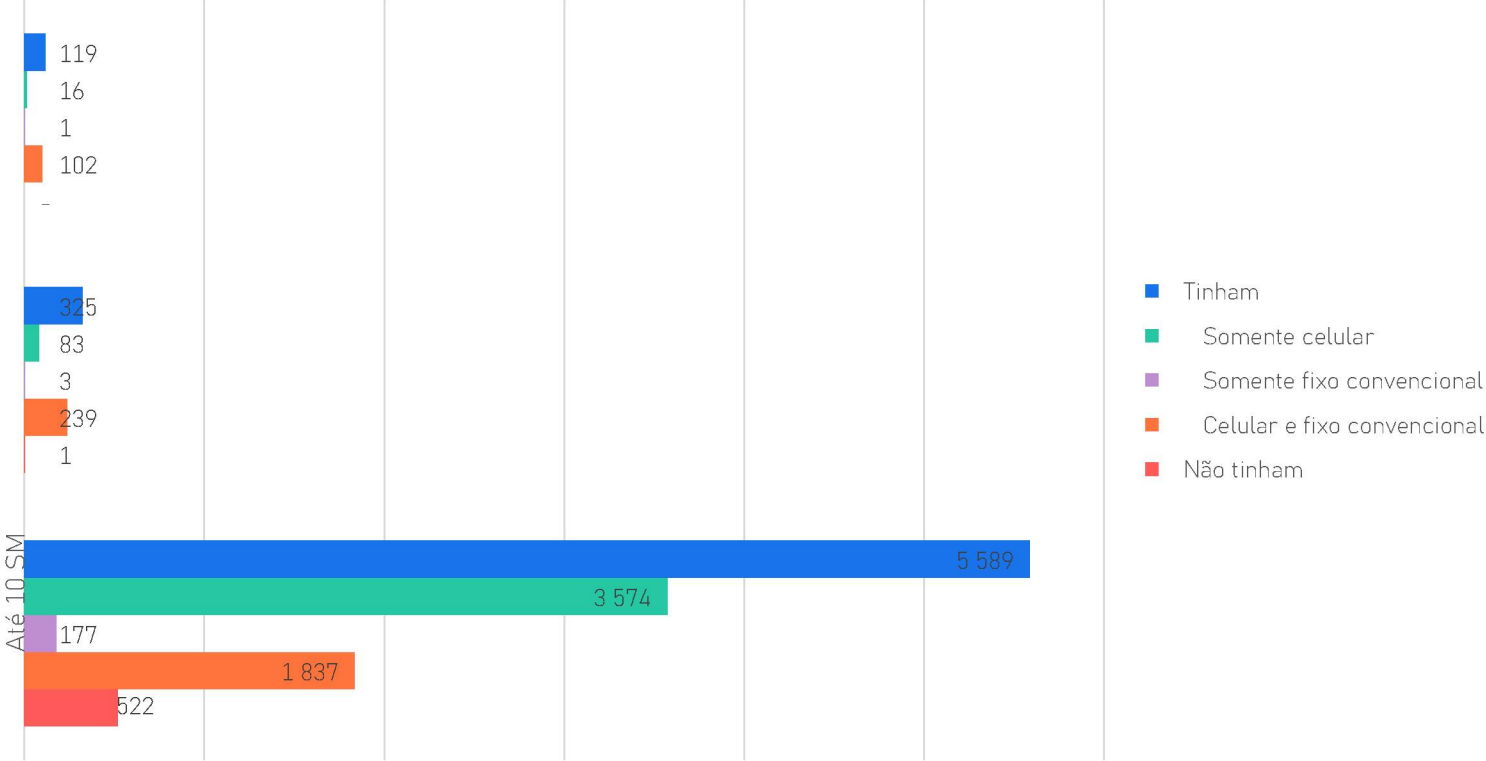
- Podemos observar novamente a mesma tendência.
- O que está ocorrendo nos lares do Brasil? Porque está diminuindo o número de computadores?
- Não era esperado que com o passar dos anos, o computador se tornasse mais e mais popular?



Telefones

- Vamos observar o acesso a telefones em 2013

Telefone – Minas Gerais – 2013





Telefones em 2013

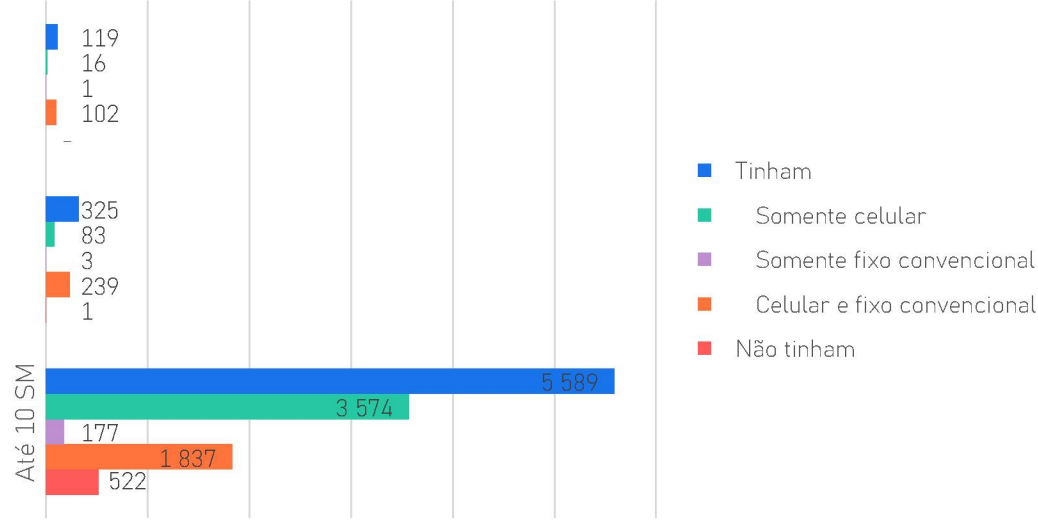
- Observa-se que, proporcionalmente falando, apenas uma parcela dos lares não possuem telefone/celular;
- Se tratando da população acima de 10 salários mínimos, todas as residências possuem telefone/celular;
- Somente uma pequena parte da população possui somente telefone fixo.



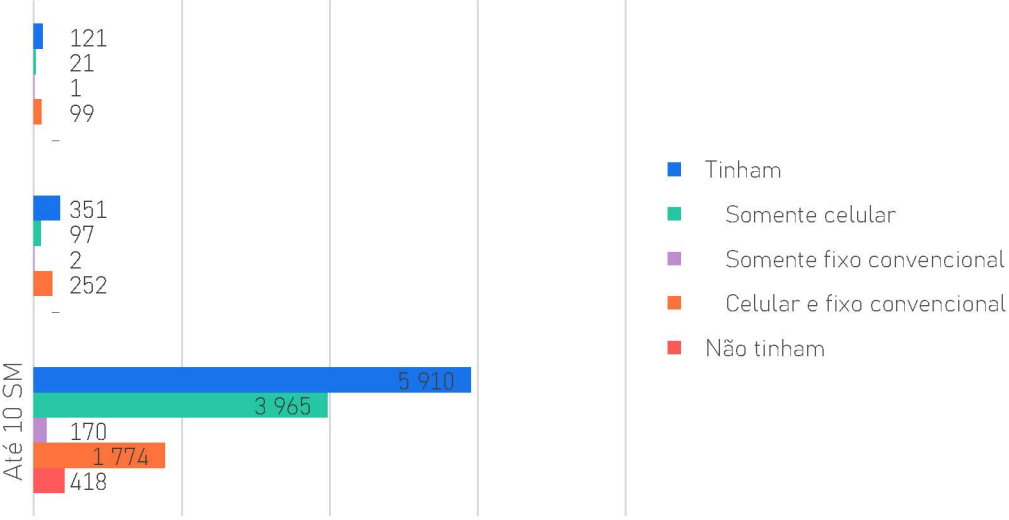
2014

- ¿Qué podemos esperar de 2014?

Telefone – Minas Gerais – 2013



Telefone – Minas Gerais – 2014





2013 x 2014

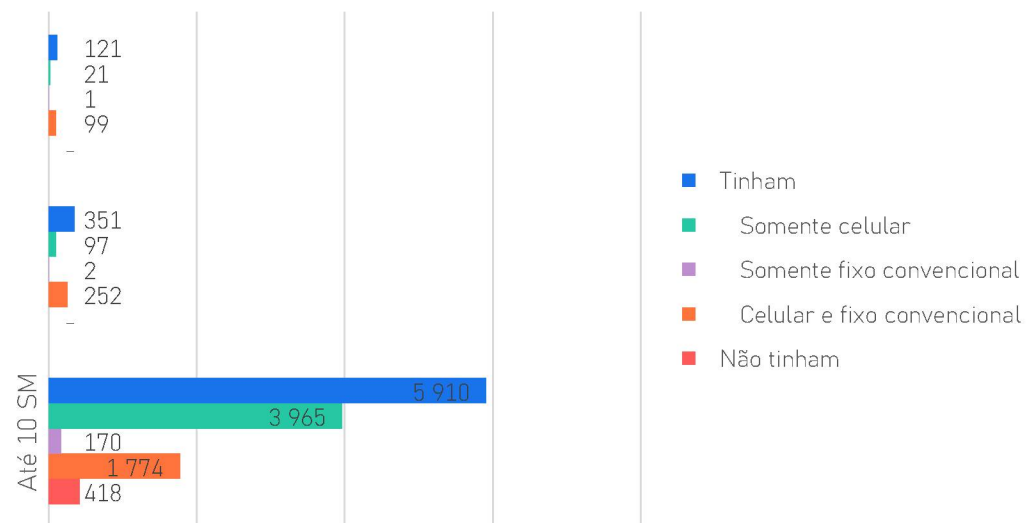
- No caso dos celulares, houve um aumento, como esperávamos, dos celulares por residência, e uma diminuição dos lares que não possuem telefone/celular;
- Diminuiu o número de residências que possuem apenas telefone fixo.



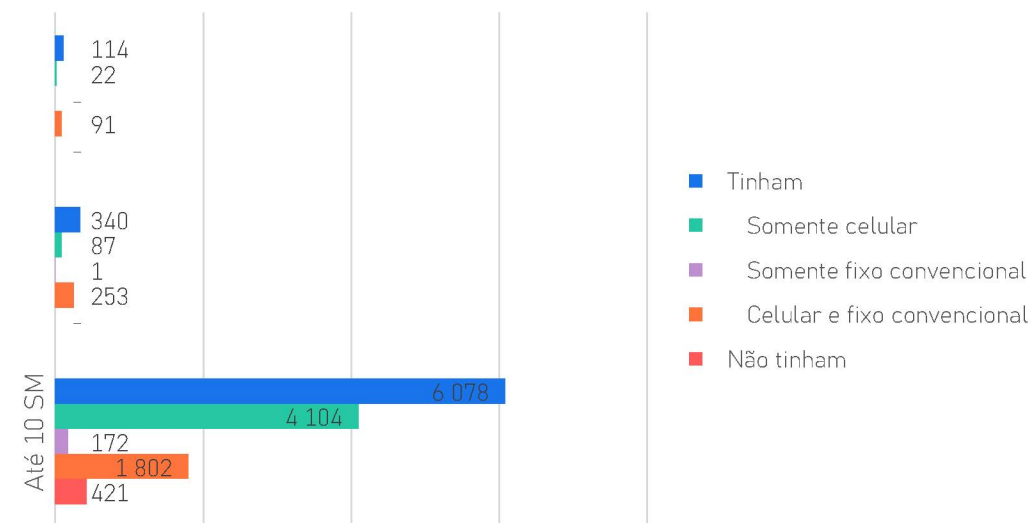
2015

- O que esperar de 2015?
- Será que 2015 seguirá a mesma tendência?

Telefone – Minas Gerais – 2014



Telefone – Minas Gerais – 2015





2014 x 2015

- Houve um aumento não muito significativo dos lares que não possuem telefones/celulares;
- Houve um aumento significativo dos lares que possuem telefones/celulares e somente celulares;
- Proporcionalmente, houve um aumento do número de residências que possuem somente celular.



Conclusão

- Os lares do Brasil entre 2013 à 2015 preferindo adquirir celulares ao invés de computadores;
- Por enquanto, os celulares vem se popularizando bem mais que os computadores nos lares com renda de até 10 salários mínimos;
- Com a chegada dos smartphones, os celulares se tornaram meios de comunicação e pequenos computadores com acesso à internet para os lares brasileiros.

APÊNDICE 15 – TÓPICOS EM INTELIGÊNCIA ARTIFICIAL

A – ENUNCIADO

1) Algoritmo Genético

Problema do Caixeiro Viajante

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

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

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

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

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

Sugestão:

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

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

2) Compare a representação de dois modelos vetoriais

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

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

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

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

B – RESOLUÇÃO

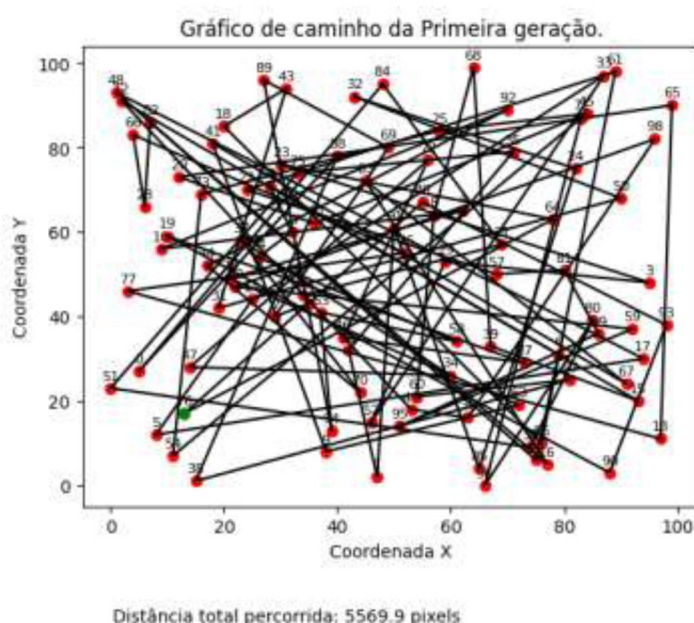
1 – Algoritmo Genético

O Problema do Caixeiro Viajante (PCV) é um problema clássico de otimização combinatória, que busca o menor caminho para visitar um conjunto de cidades. Por ser um problema NP-Completo, sua complexidade aumenta superpolinomialmente com o número de cidades. Neste trabalho, um algoritmo genético foi usado para propor uma solução otimizada em relação a uma solução aleatória, dado que o PCV ainda não possui uma solução ótima para um número muito grande de cidades.

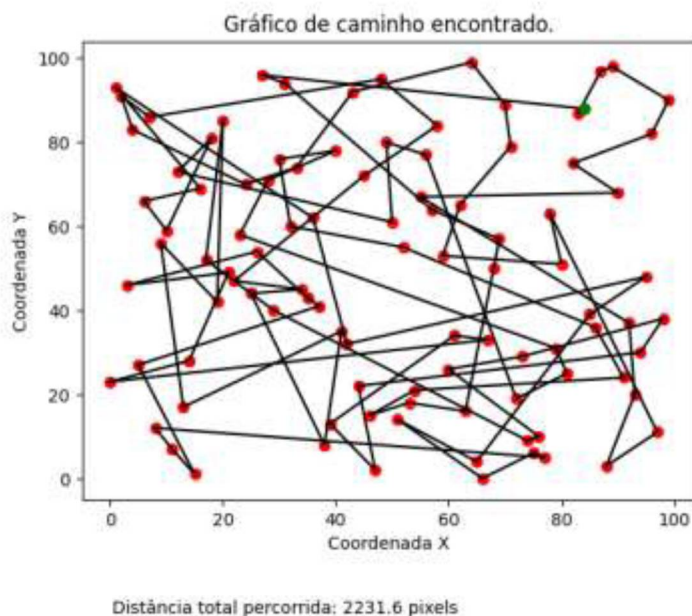
Para o cruzamento, foi utilizada a técnica de Crossover de Ordem (OX) com um ponto de corte, devido à restrição de que os valores das cidades não podem se repetir. Para a mutação, o indivíduo foi dividido em dois segmentos, e os alelos do segundo segmento foram embaralhados.

Na população inicial, foram criados 100 indivíduos aleatoriamente, e o indivíduo com o melhor fitness foi selecionado para a segunda geração, implementando o elitismo. Foram observadas 30.000 gerações, com uma taxa de cruzamento de 90% e uma taxa de mutação de 1%.

Melhor indivíduo da primeira geração:



Melhor indivíduo da última geração:



Os resultados demonstram a eficácia do AG em reduzir a distância total das rotas. Além da observação da menor distância euclidiana, a plotagem das rotas otimizadas revela uma significativa diminuição no número de cruzamentos entre os caminhos percorridos. Essa redução de cruzamentos sugere uma menor complexidade da rota, o que corrobora a diminuição da distância total percorrida.

2 – Comparando a representação de dois modelos vetoriais

Para visualizar a representação vetorial de textos, selecionamos dez fragmentos curtos, abrangendo temas como clima, culinária e animais, com similaridades intencionais para evidenciar o agrupamento vetorial. O pré-processamento incluiu conversão para minúsculas, remoção de pontuação, números e stopwords, seguido de tokenização.

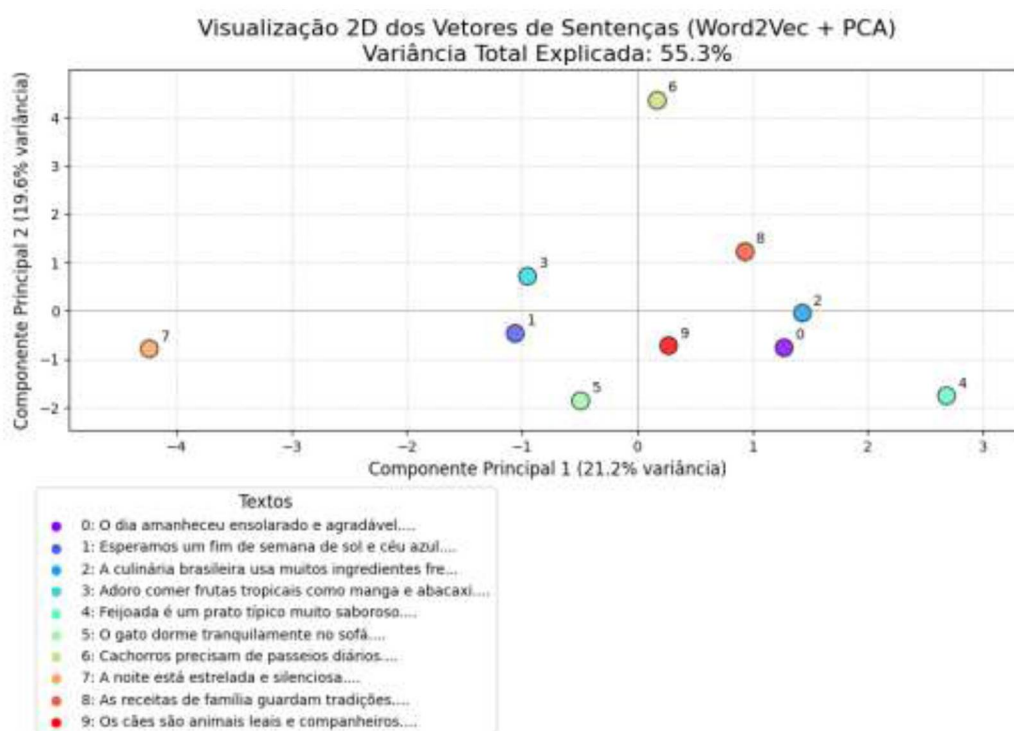
A representação vetorial foi obtida via Word2Vec, treinado com 100 épocas para capturar nuances semânticas em nosso pequeno corpus. A representação de cada sentença foi calculada pela média dos vetores das palavras, resultando em vetores de alta dimensionalidade.

A Análise de Componentes Principais (PCA) foi aplicada para reduzir a dimensionalidade, facilitando a visualização em gráficos 2D. Os resultados, multiplicados por 100 para clareza visual, demonstram o agrupamento de sentenças semanticamente similares, validando a eficácia da representação vetorial e da PCA na identificação de relações textuais.

Frases utilizadas		relações
1	O dia amanheceu ensolarado e agradável.	Clima 1
2	Esperamos um fim de semana de sol e céu azul.	Clima 2 (similar a 1)

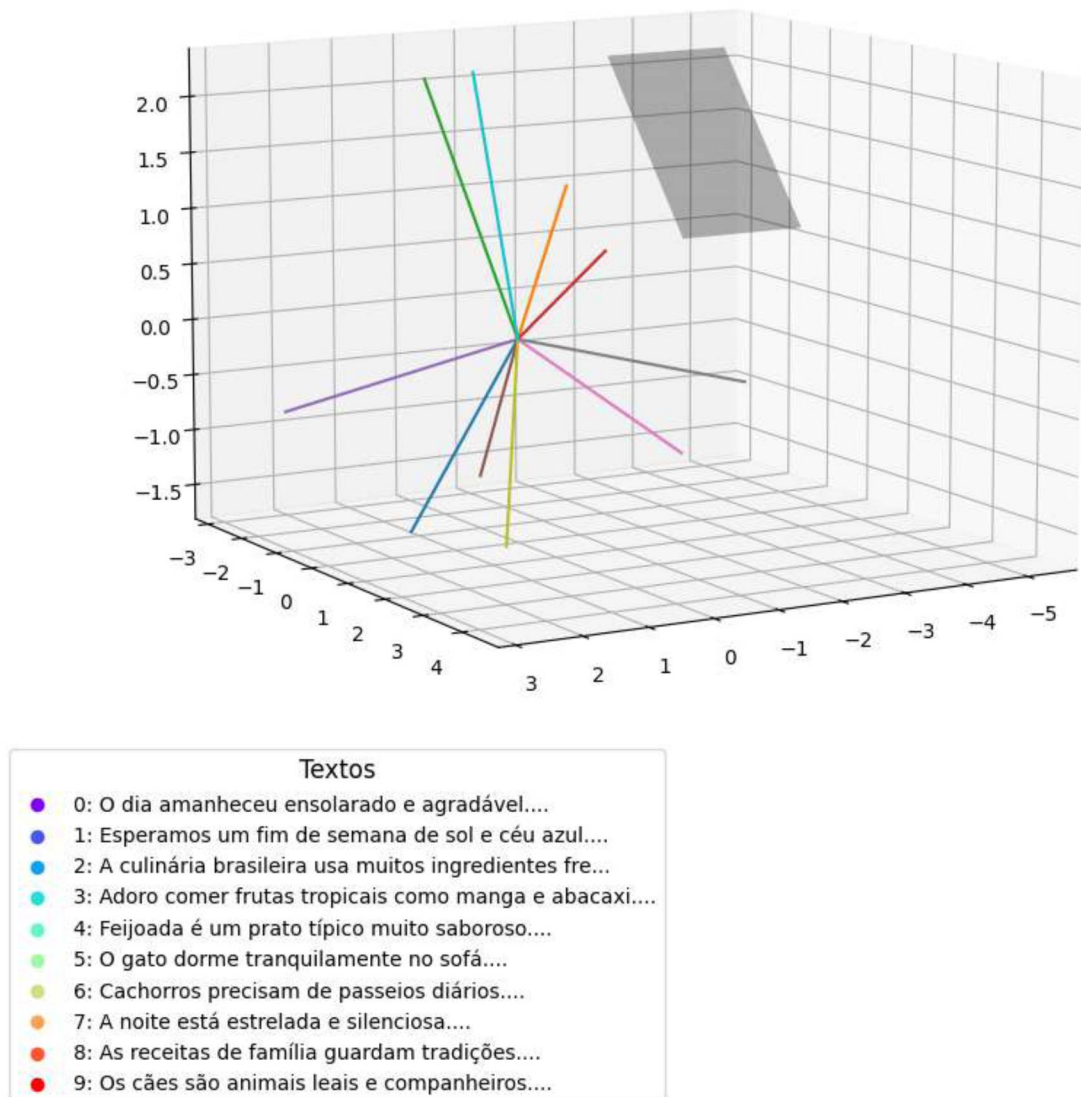
3	A culinária brasileira usa muitos ingredientes frescos.	Culinária 1
4	Adoro comer frutas tropicais como manga e abacaxi.	Culinária/frutas 2
5	Feijoada é um prato típico muito saboroso.	Culinária 3 (similar ao 3)
6	O gato dorme tranquilamente no sofá.	Animais 1
7	Cachorros precisam de passeios diários.	Animais 2
8	A noite está estrelada e silenciosa.	Clima/Noite 3 (ligeiramente similar ao 1/2)
9	As receitas de família guardam tradições.	Culinária 4 (similar ao 3/5)
10	Os cães são animais leais e companheiros.	Animais 3 (similar ao 7)

Visualização dos vetores



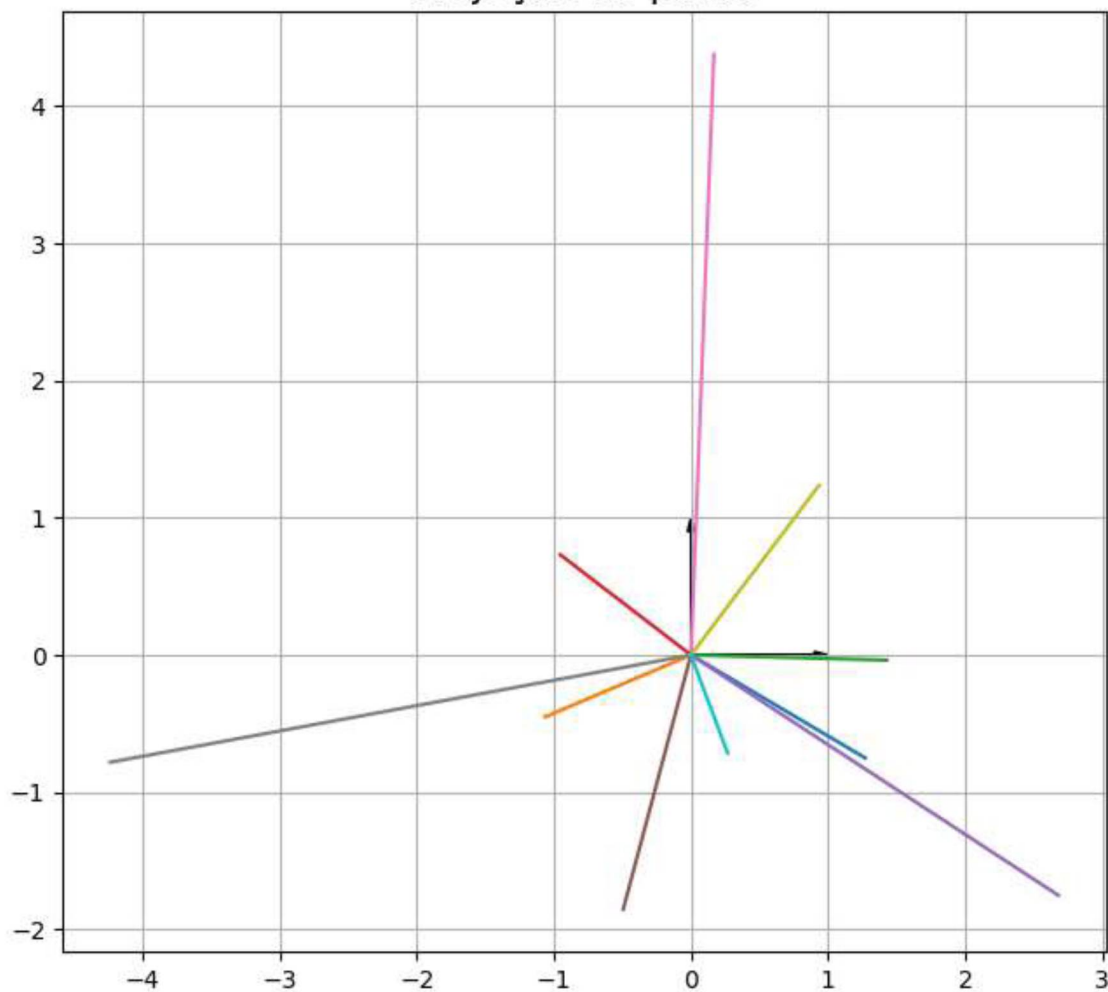
O gráfico acima mostra cada texto original como um ponto no espaço 2D. A posição de cada ponto é determinada pelo vetor da sentença (média dos vetores das palavras) após a redução com PCA. Textos com conteúdo semanticamente similar (segundo o modelo Word2Vec) devem aparecer mais próximos uns dos outros. Com um corpus pequeno, as relações capturadas pelo Word2Vec podem ser fracas ou baseadas em coocorrências simples de palavras.

Vetores e plano calculado para projeção



O gráfico acima representa a visualização dos vetores das sentenças em três dimensões, essa técnica é uma alternativa para gerar uma visualização 2D desses vetores com a dimensionalidade menor e, portanto, mais compreensível para a interpretação humana, como segue abaixo:

Projeção no plano



Textos

- 0: O dia amanheceu ensolarado e agradável....
- 1: Esperamos um fim de semana de sol e céu azul....
- 2: A culinária brasileira usa muitos ingredientes fre...
- 3: Adoro comer frutas tropicais como manga e abacaxi....
- 4: Feijoada é um prato típico muito saboroso....
- 5: O gato dorme tranquilamente no sofá....
- 6: Cachorros precisam de passeios diários....
- 7: A noite está estrelada e silenciosa....
- 8: As receitas de família guardam tradições....
- 9: Os cães são animais leais e companheiros....