

UNIVERSIDADE FEDERAL DO PARANÁ

DAVID REKSIDLER JÚNIOR

MEMORIAL DE PROJETOS: AAAAAAAAAAAAAAAAAAAAAA

CURITIBA

2025

DAVID REKSIDLER JÚNIOR

MEMORIAL DE PROJETOS: AAAAAAAAAAAAAAAAAAAAAA

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

Orientadora: Prof^a Dr^a Rafaela Mantovani Fontana

CURITIBA

2025

DEIXAR ESTA PÁGINA EM BRANCO PARA COLOCAR A IMAGEM DO TERMO DE APROVAÇÃO ASSINADO QUE SERÁ FORNECIDO APÓS A AVALIAÇÃO DO MEMORIAL.

AGRADECIMENTOS

AAAAAAAAAAAAAAAAAAAAA

*“Even his most cogent moments were tinged with delusion. On the other hand,
his most delirious dreams touched on deep realities.”*
(Mad Merlin - J. Robert King)

RESUMO

AAAAAAAAAAAAAAAAAAAAA

Palavras-chaves: A; B;C.

ABSTRACT

BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Key-words: D; E; F.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Rotas Lugoj-Bucharest	12
FIGURA 2 – Distâncias em linha reta para a cidade de Bucharest	12
FIGURA 3 – Rede Neural Artificial exercício 4	13
FIGURA 4 – Árvore final	17
FIGURA 5 – Tangente Hiperbólica	18
FIGURA 6 – Distribuição de carros por marca	26
FIGURA 7 – Distribuição de carros por engrenagem (marcha)	27
FIGURA 8 – Evolução da média de preço dos carros ao longo dos meses de 2022	28
FIGURA 9 – Distribuição da média de preço dos carros por marca e tipo de engrenagem	29
FIGURA 10 – Distribuição da média de preço dos carros por marca e tipo de combustível	30
FIGURA 11 – Distribuição dos preços dos carros	31
FIGURA 12 – Mapa de Correlação das Variáveis Numéricas	32
FIGURA 13 – Boxplot Idade das esposas	49
FIGURA 14 – Boxplot Idade dos maridos	49
FIGURA 15 – Histograma Idade das esposas	50
FIGURA 16 – Histograma Idade dos maridos	50
FIGURA 17 – Gráfico da distribuição F	55
FIGURA 18 – Mapa de correlação dados clínicos	75

SUMÁRIO

1	PARECER TÉCNICO	9
	REFERÊNCIAS	10
	APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL . .	11
	APÊNDICE 2 – LINGUAGEM DE PROGRAMAÇÃO APLICADA . .	20
	APÊNDICE 3 – LINGUAGEM R	38
	APÊNDICE 4 – ESTATÍSTICA APLICADA I	47
	APÊNDICE 5 – ESTATÍSTICA APLICADA II	57
	APÊNDICE 6 – ARQUITETURA DE DADOS	65
	APÊNDICE 7 – ASPECTOS FILOSÓFICOS E ÉTICOS DA IA . . .	78
	APÊNDICE 8 – APRENDIZADO DE MÁQUINA	80
	APÊNDICE 9 – DEEP LEARNING	81
9.0.1	Classificação de Imagens (CNN)	81
9.0.2	Detector de SPAM (RNN)	81
9.0.3	Gerador de Dígitos Fake (GAN)	81
9.0.4	Tradutor de Textos (Transformer)	81
	APÊNDICE 10 – BIG DATA	82
	APÊNDICE 11 – VISÃO COMPUTACIONAL	83
11.0.1	Extração de Características	83
11.0.2	Redes Neurais	84
	APÊNDICE 12 – GESTÃO DE PROJETOS DE IA	85
12.0.1	Objetivo	85
12.0.2	Orientações adicionais	85
	APÊNDICE 13 – FRAMEWORKS DE IA	87
13.0.1	Classificação (RNA)	87
13.0.2	Regressão (RNA)	87
13.0.3	Sistemas de Recomendação	89
13.0.4	Deepdream	89
	APÊNDICE 14 – VISUALIZAÇÃO DE DADOS E STORYTELLING .	91
	APÊNDICE 15 – TÓPICOS EM IA	92
15.0.1	Algoritmo Genético	92
15.0.2	Compare a representação de dois modelos vetoriais	93

1 PARECER TÉCNICO

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa ABNTeX2 (2013)

REFERÊNCIAS

ABNTEX2. **Modelo Canônico de Trabalho Acadêmico com abnTeX2**. [S.l.], 2013. Disponível em: <http://abntex2.googlecode.com/>. Citado 1 vez na página 9.

MIT. **How ChatGPT Works: A Non-Technical Primer**. [S.l.: s.n.], 2023. <https://mitsloanedtech.mit.edu/ai/basics/how-chatgpt-works-a-non-technical-primer/>. Accessed: 25/02/2024. Citado 1 vez na página 15.

OPENAI. **What is ChatGPT?** [S.l.: s.n.], 2024. <https://help.openai.com/en/articles/6783457-what-is-chatgpt>. Accessed: 25/02/2024. Citado 1 vez na página 15.

RUBY, M. **How ChatGPT Works: The Model Behind The Bot**. [S.l.: s.n.], 2023. <https://towardsdatascience.com/how-chatgpt-works-the-models-behind-the-bot-1ce5fca96286>. Accessed: 25/02/2024. Citado 1 vez na página 15.

APÊNDICE 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

A - ENUNCIADO

1 ChatGPT

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

2 Busca Heurística

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

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

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

NÃO É NECESSÁRIO IMPLEMENTAR O ALGORITMO.

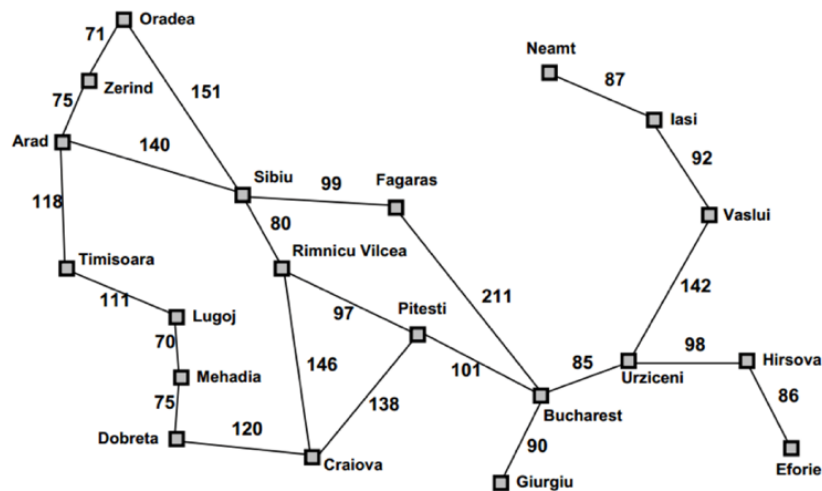


FIGURA 1 – Rotas Lugoj-Bucharest

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

FIGURA 2 – Distâncias em linha reta para a cidade de Bucharest

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.

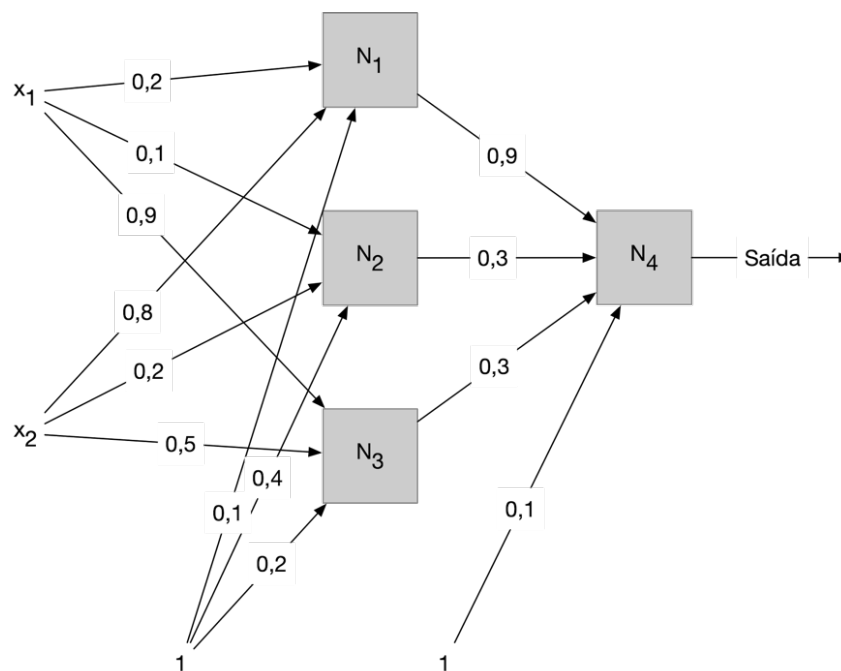


FIGURA 3 – Rede Neural Artificial exercício 4

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

Dada a entrada $x_1 = -3$, $x_2 = 1$, dê:

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

B - RESOLUÇÃO

1 ChatGPT

a) Pergunte ao ChatGPT o que é Inteligência Artificial e cole aqui o resultado.

Inteligência Artificial (IA) refere-se à capacidade de um sistema ou programa de computador realizar tarefas que normalmente exigiriam inteligência humana. Essas tarefas incluem raciocínio, aprendizado, percepção visual, reconhecimento de fala, compreensão de linguagem natural, tomada de decisões e resolução de problemas.

Os sistemas de IA são projetados para imitar certos aspectos da inteligência humana, utilizando algoritmos e modelos matemáticos para processar dados, aprender padrões e tomar decisões. Existem dois tipos principais de IA:

1. IA Fraca (ou Estreita): Refere-se a sistemas de IA especializados em tarefas específicas. Eles são limitados ao escopo de suas programações e não têm a capacidade de realizar tarefas além do que foram projetados para fazer. Exemplos incluem reconhecimento de voz, chatbots e carros autônomos.
2. IA Forte (ou Geral): Envolve sistemas que têm a capacidade de realizar qualquer tarefa cognitiva que um ser humano pode fazer. Este nível de IA ainda não foi totalmente alcançado e é um campo de pesquisa em desenvolvimento.

Algoritmos de aprendizado de máquina (Machine Learning) são frequentemente usados na implementação de sistemas de IA. Eles permitem que a IA melhore seu desempenho em uma tarefa específica à medida que é exposta a mais dados, sem ser explicitamente programada.

A Inteligência Artificial tem aplicações em uma variedade de setores, como medicina, finanças, automação industrial, reconhecimento de padrões, entretenimento e muito mais. É uma área de rápido avanço e continua a desempenhar um papel significativo na transformação de várias indústrias.

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

A definição dada pelo ChatGPT menciona que sistemas de Inteligência Artificial são projetados para "imitar certos aspectos da inteligência humana", o que parece se assemelhar mais à definição de Pensar como Humanos ou Agir como Humanos.

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

O ChatGPT pode ser entendido como uma extrapolação de uma classe de modelos de aprendizagem de máquina chamados Large Language Models (LLMs), que são modelos de Processamento de Linguagem Natural. Esse tipo de modelo consegue processar grandes quantidades de texto e inferir relações entre palavras dentro do texto. A capacidade dos LLMs cresce conforme aumenta o tamanho e variedade de parâmetros da base de dados.(Ruby, 2023)

Segundo o que é disponibilizado pelo OpenAI(OpenAI, 2024), ele é baseado na arquitetura GPT (Generative Pre-trained Transformer), um modelo Transformer, que é uma rede neural capaz de aprender o contexto dado e gerar um novo texto a partir disso. A rede foi otimizada para usar um método de treinamento chamado Reinforcement Learning with Human Feedback (RLHF), que usa demonstrações humanas para guiar seu comportamento, de forma a alcançar o desejável.

Segundo Rama Ramakrishnan, professor do MIT, (MIT, 2023), o modelo precedente, GPT-3, foi treinado com cerca de 30 bilhões de frases retiradas de livros e da internet, para uma única tarefa: prever a palavra seguinte em uma frase, dadas as palavras usadas anteriormente. Ele calcula uma tabela de probabilidades para possíveis palavras a serem usadas e usa a que tem a melhor probabilidade, quase como um sistema de *autocomplete*. Para formar frases completas, a cada palavra escolhida, ele adiciona ela à frase e refaz o processo de cálculo das probabilidades para escolher a próxima palavra. Já a versão 3.5 foi treinada para seguir instruções dadas por humanos, utilizando uma base de dados contendo pares de exemplos de instruções e respostas de alta qualidade para tais instruções. O modelo treinado a partir dessa base de dados foi então utilizado para gerar múltiplas respostas para cada uma das instruções e essas respostas foram categorizadas por humanos de mais útil a menos útil. A partir desses novos dados, foi possível treinar um "modelo de recompensa", que avalia a qualidade das respostas geradas pelo GPT-3.5 e retorna uma "nota"de avaliação para o GPT-3.5, assim ele passa por um *fine tuning* para melhorar suas respostas com *reinforcement learning*. Na transição do GPT-3.5 para o ChatGPT foi usado um processo similar, mas desta vez utilizando conversas inteiras para o treinamento.

Ele é treinado usando textos escritos por humanos, incluindo conversações, de forma a conseguir imitar o estilo de comunicação humano). Assim, de acordo com os dados utilizados para treinamento, ele pode, inclusive, produzir textos com conteúdo enviesado.

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

De acordo com o comportamento, que se baseia em aprender e imitar como humanos se comunicam, ele parece se encaixar na definição da abordagem Agir como Humanos. Essa conclusão pode ser atingida considerando-se que o ChatGPT encontra padrões nos textos, que são dados produzidos (majoritariamente) por humanos, e com isso aprende contextos, mas não é totalmente consciente sobre o seu conteúdo, e não usa um processo cognitivo, apenas os sintetiza conforme foram fornecidos a ele. Como exemplo, se o ChatGPT fosse treinado com textos com um viés racista, ele poderia reproduzir textos racistas, já que não tem a capacidade de ponderar se racismo é correto ou não de um ponto de vista racional, ou seja, ele não toma decisões racionais baseadas no contexto que é apresentado a ele, apenas busca imitar a forma como humanos se comunicam.

2 Busca Heurística

a) 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.

a) Deve-se mostrar todos os passos e regras aplicadas, no mesmo formato apresentado nas aulas e nas práticas.

$$R1 : p \Rightarrow q$$

$$R2 : q \Rightarrow r$$

$$R3 : \neg r \vee p$$

$$R4 : r \Rightarrow p$$

Equivalência Implicação, R3

$$R5 : q \Rightarrow p$$

SH, R2, R4

$$R6 : p \Rightarrow q \wedge q \Rightarrow p$$

CONJ, R1, R5

$$R7 : q \Leftrightarrow p$$

BICOND, R6

4 Redes Neurais Artificiais

$$x1=-3, x2=1$$

N1, N2, N3: função de ativação linear (mantém o valor)

N4: função de ativação tangente (produz valores no intervalo $[-1,1]$)

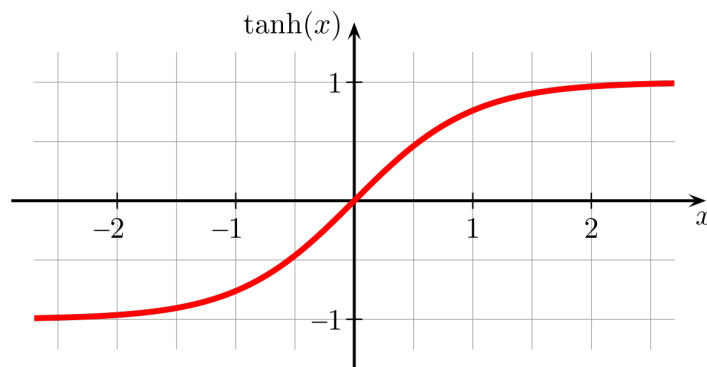


FIGURA 5 – Tangente Hiperbólica

a) Valor de saída do neurônio N1

$$(-3) \times 0.2 + 1 \times 0.8 + 1 \times 0.1 = 0.3$$

b) Valor de saída do neurônio N2

$$(-3) \times 0.1 + 1 \times 0.2 + 1 \times 0.4 = 0.3$$

c) Valor de saída do neurônio N3

$$(-3) \times 0.9 + 1 \times 0.5 + 1 \times 0.2 = -2$$

d) Valor de saída da rede como um todo

$$\tanh(0.3 \times 0.9 + 0.3 \times 0.3 + (-2) \times 0.3 + 1 \times 0.1) = \tanh(0.27 + 0.09 - 0.6 + 0.1) = -0.13909 \approx -0.14$$

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 do ano de referência
month_of_reference	O preço médio corresponde a um mês específico, pois a FIPE atualiza mensalmente
fipe_code	Código único da FIPE
authentication	Código único de autenticação para consulta FIPE
brand	Marca do carro
model	Modelo do carro
fuel	Tipo de combustível
gear	Tipo de engrenagem
engine_size	Tamanho do motor em centímetros cúbicos
year_model	Ano do modelo (pode ser diferente do ano de fabricação)
avg_price	Preço médio do carro em reais

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

1 Análise Exploratória dos dados

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

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

2 Visualização dos dados

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

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

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

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

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

B - RESOLUÇÃO

1 Análise Exploratória dos dados

a. Carregue a base de dados **media_precos_carros_brasil.csv**

```
# Para rápida instalação das bibliotecas necessárias, rodar o script abaixo  
# no CLI:  
# pip install -r requirements.txt  
  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt
```

```

import numpy as np

import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import RandomizedSearchCV

# Métricas de avaliação dos modelos
from sklearn.metrics import mean_squared_error, mean_absolute_error,
    r2_score

dados = pd.read_csv('precos_carros_brasil.csv')
dados.head()

```

year_ref	month	fipe_code	auth	brand	model	fuel	gear	engine	year	price
2021.0	January	004001-0	cfzictzfwrcp	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2002.0	9162.0
2021.0	January	004001-0	cdqwxwpw3y2p	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2001.0	8832.0
2021.0	January	004001-0	cb1t3xwwj1xp	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Gasoline	manual	1	2000.0	8388.0
2021.0	January	004001-0	cb9gct6j65r0	GM - Chevrolet	Corsa Wind 1.0 MPFI / EFI 2p	Alcohol	manual	1	2000.0	8453.0
2021.0	January	004003-7	g15wg0gbz1fx	GM - Chevrolet	Corsa Pick-Up GL/ Champ 1.6 MPFI / EFI	Gasoline	manual	1.6	2001.0	12525.0

b. Verifique se há valores faltantes nos dados. Caso haja, escolha uma tratativa para resolver o problema de valores faltantes

```

# Excluindo as linhas sem dados (característica de importação de .CSV)
dados = dados.dropna(how='all')

antes = dados.shape

# Verificando se existem valores faltantes nos dados
dados.isna().any()

```

year_of_reference	False
month_of_reference	False
fipe_code	False
authentication	False
brand	False
model	False
fuel	False
gear	False
engine_size	False


```
year_model          False
avg_price_brl       False
dtype: bool
```

Não há valores faltantes.

c. Verifique se há dados duplicados nos dados

```
# Verificando se temos valores duplicados
dados.duplicated().sum()
```

```
2
```

```
# Removendo valores duplicados
dados.drop_duplicates(inplace=True)

depois = dados.shape

# Verifica diferença após a normalização dos dados
diff_linhas = depois[0] - antes[0]
diff_colunas = depois[1] - antes[1]
print("Linhas:\n Antes {} x Depois {} = Diff {}".format(antes[0], depois[0],
    diff_linhas))
print("Colunas:\n Antes {} x Depois {} = Diff {}".format(antes[1], depois[1],
    diff_colunas))
```

```
Linhas:
  Antes 202297 x Depois 202295 = Diff -2
Colunas:
  Antes 11 x Depois 11 = Diff 0
```

2 linhas duplicadas excluídas.

d. Crie duas categorias, para separar colunas numéricas e categóricas. Imprima o resumo de informações das variáveis numéricas e categóricas (estatística descritiva dos dados)

```
# Criando categorias para separar colunas numéricas e categóricas: facilita
a AED
numericas_cols = [col for col in dados.columns if dados[col].dtype != 'object'
    ]
categorias_cols = [col for col in dados.columns if dados[col].dtype == '
    object']
```

```
# Resumo das variáveis numéricas – Imprime alguns valores de medidas
de tendências centrais
dados[numericas_cols].describe()
```

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

e. Imprima a contagem de valores por modelo (model) e marca do carro (brand)

```
# Contagem de valores por categoria de 'Modelo'
dados['model'].value_counts()
```

```
Palio Week. Adv/Adv TRYON 1.8 mpi Flex      425
Focus 1.6 S/SE/SE Plus Flex 8V/16V 5p      425
Focus 2.0 16V/SE/SE Plus Flex 5p Aut.      400
Saveiro 1.6 Mi/ 1.6 Mi Total Flex 8V        400
Corvette 5.7/ 6.0, 6.2 Targa/Stingray      375
...
STEPWAY Zen Flex 1.0 12V Mec.                2
Saveiro Robust 1.6 Total Flex 16V CD         2
Saveiro Robust 1.6 Total Flex 16V           2
Gol Last Edition 1.0 Flex 12V 5p            2
Polo Track 1.0 Flex 12V 5p                  2
Name: model, Length: 2112, dtype: int64
```

```
# Contagem de valores por categoria de 'Marca'
dados['brand'].value_counts()
```

```
Fiat      44962
VW - Volkswagen  44312
GM - Chevrolet  38590
Ford      33150
Renault   29191
Nissan    12090
Name: brand, dtype: int64
```

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

Na Análise Exploratória dos Dados pudemos observar que o conjunto de dados possui 11 colunas e nenhuma delas possui dados faltantes. Ao observarmos valores duplicados, apenas 2 linhas estavam duplicadas. Das 11 colunas, apenas 3 são numéricas. Porém, as colunas brand e model podem ser facilmente convertidas para numéricas.

2 Visualização dos dados

a. Gere um gráfico da distribuição da quantidade de carros por marca

```
# Gráfico da distribuição por marca
plt.figure(figsize=(20,10))
plt.bar(dados['brand'].unique(), dados['brand'].value_counts())
plt.title('Distribuição de carros por marca')
plt.ylabel('Total de carros')
plt.xlabel('Marca')
```

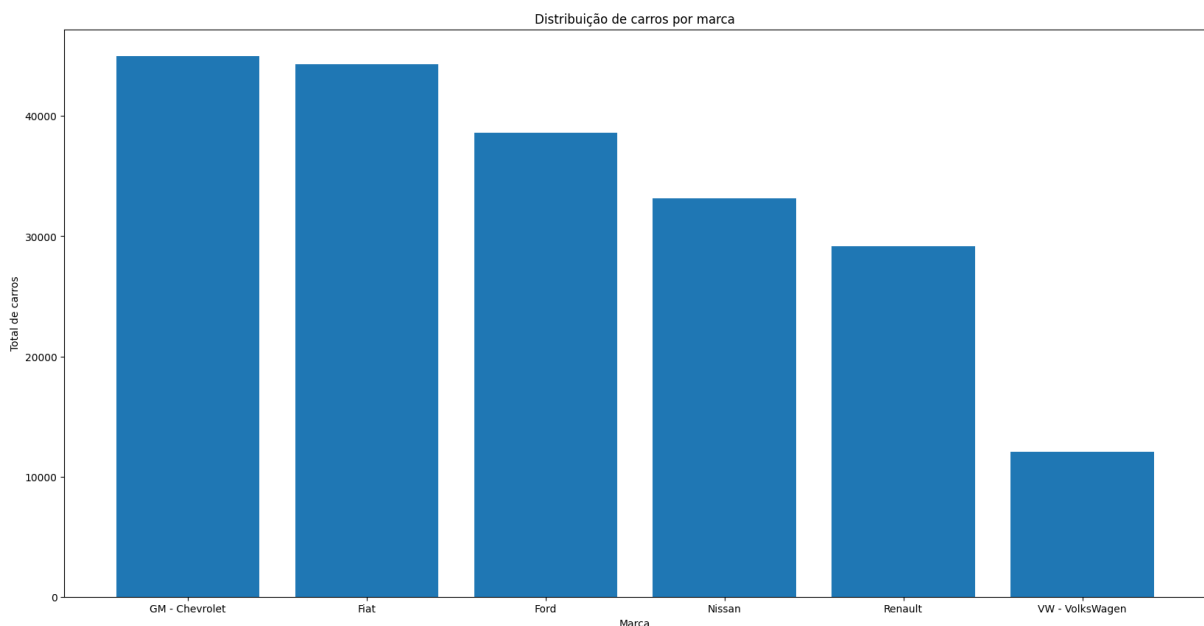


FIGURA 6 – Distribuição de carros por marca

b. Gere um gráfico da distribuição da quantidade de carros por tipo de engrenagem do carro

```
# Gráfico da distribuição por engrenagem (tipo de marcha)
plt.figure(figsize=(20,10))
plt.bar(dados['gear'].unique(), dados['gear'].value_counts())
plt.title('Distribuição de carros por engrenagem (marcha)')
```

```
plt.ylabel('Total de carros')
plt.xlabel('Tipo de engrenagem (marcha)')
```

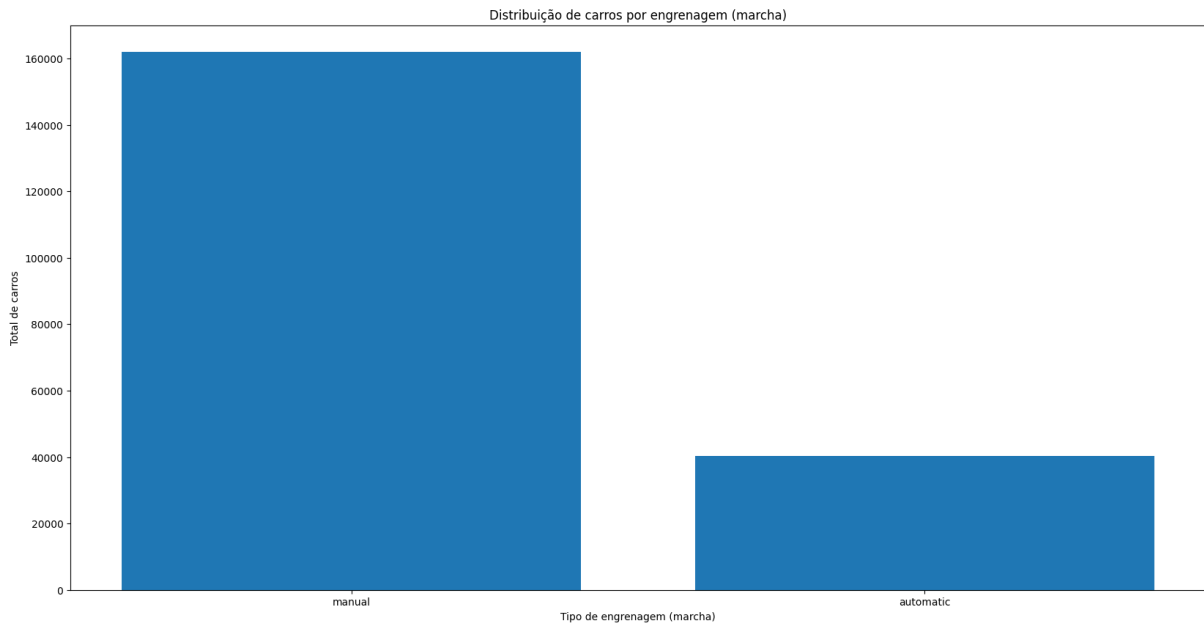


FIGURA 7 – Distribuição de carros por engrenagem (marcha)

c. Gere um gráfico da evolução da média de preço dos carros ao longo dos meses de 2022 (variável de tempo no eixo X)

```
# limitando para somente os dados de 2022
dados_2022 = dados[dados['year_of_reference'] == 2022]

# verificando os completude dos meses
dados_2022['month_of_reference'].drop_duplicates()

# Calculando a média de preços por mês
media_precos_mes = dados_2022.groupby(['year_of_reference',
    'month_of_reference'])['avg_price_brl'].mean().round(0)
df_media_precos_mes = media_precos_mes.reset_index(name='Médio de
    Preço')

# Reordenando o dataframe para melhor exibição no gráfico
df_media_precos_mes.insert(0, "index", [3,7,11,1,0,6,5,2,4,10,9,8], True)
df_media_precos_mes = df_media_precos_mes.sort_values(by=['index'])
df_media_precos_mes.head(12)
```

Index	Order	Year	Month	Average Price (BRL)
4	0	2022.0	January	54840.0
3	1	2022.0	February	55825.0
7	2	2022.0	March	56849.0
0	3	2022.0	April	57150.0
8	4	2022.0	May	57800.0
6	5	2022.0	June	58066.0
5	6	2022.0	July	57894.0
1	7	2022.0	August	57924.0
11	8	2022.0	September	58199.0
10	9	2022.0	October	58227.0
9	10	2022.0	November	58216.0
2	11	2022.0	December	57997.0

```
# Visualizando a média salarial por ano
plt.figure(figsize=(20,10))
sns.barplot(data=df_media_precos_mes, x='month_of_reference', y='Médio
de Preço', hue='year_of_reference')
plt.ylim(50000, 60000) # limitação do eixo Y para melhor observação da
variação de valores
plt.xticks(rotation=45);
```

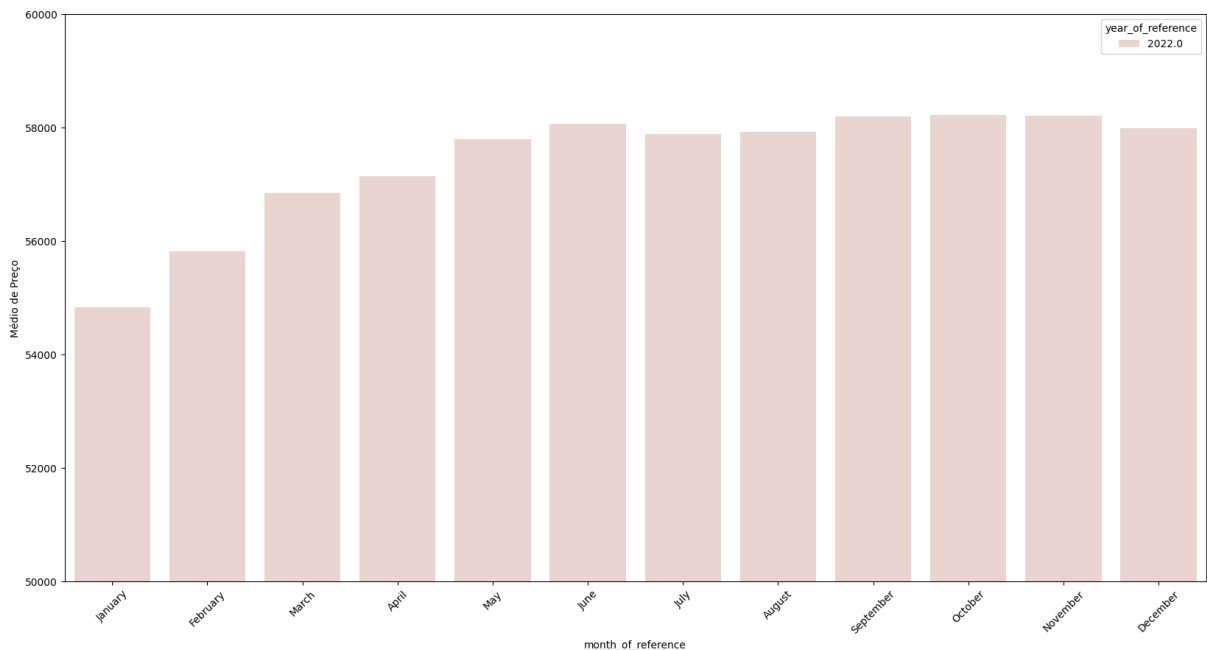


FIGURA 8 – Evolução da média de preço dos carros ao longo dos meses de 2022

d. Gere um gráfico da distribuição da média de preço dos carros por marca e tipo de engrenagem

```
print(dados["gear"].unique()) # verificar os valores da coluna gear
```

```
['manual', 'automatic']
```

```
plt.figure(figsize=(20,10))
sns.barplot(x='brand', y='avg_price_brl', hue='gear', data=dados, hue_order
            =['manual', 'automatic'])
```

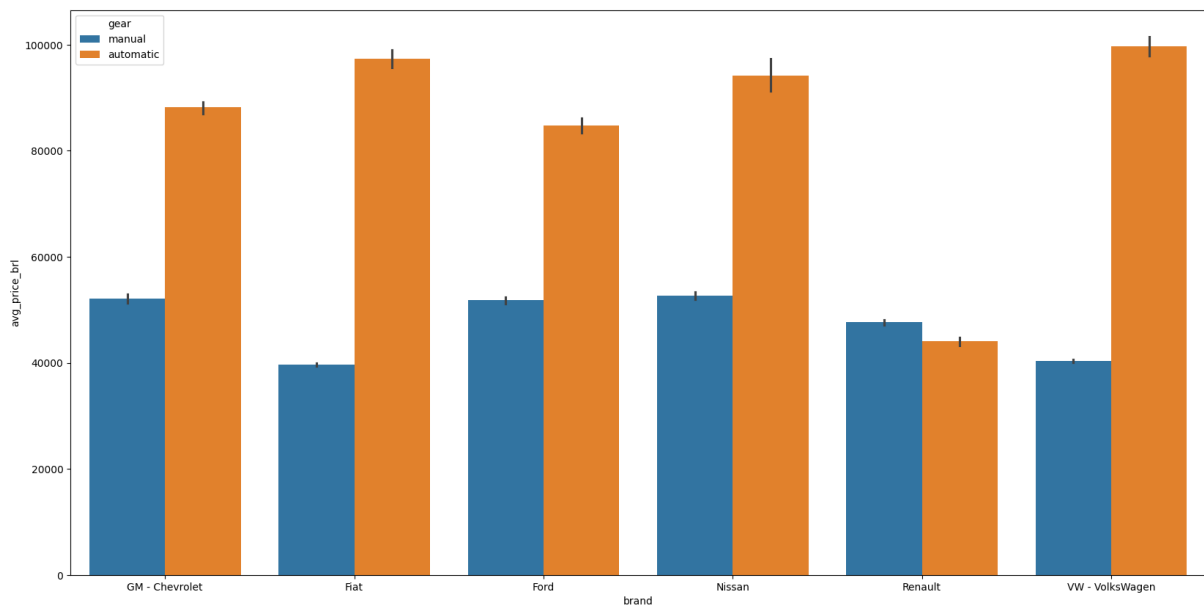


FIGURA 9 – Distribuição da média de preço dos carros por marca e tipo de engrenagem

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

O preço médio de carros com engrenagem (marcha) automática é superior aos com engrenagem manual em 5 das 6 marcas. Somente a Renault tem preços médios com ligeira superioridade nos carros com engrenagem manual. Nas marcas Fiat e Volkswagen os carro com engrenagem automática tem preço médio maiores do que o dobro dos carros com engrenagem manual.

f. Gere um gráfico da distribuição da média de preço dos carros por marca e tipo de combustível

```
print(dados["fuel"].unique()) # verificar os valores da coluna fuel
```

```
['Gasoline', 'Alcohol', 'Diesel']
```

```
plt.figure(figsize=(20,10))
sns.barplot(x='brand', y='avg_price_brl', hue='fuel', data=dados, hue_order
            =['Diesel', 'Gasoline', 'Alcohol'])
```

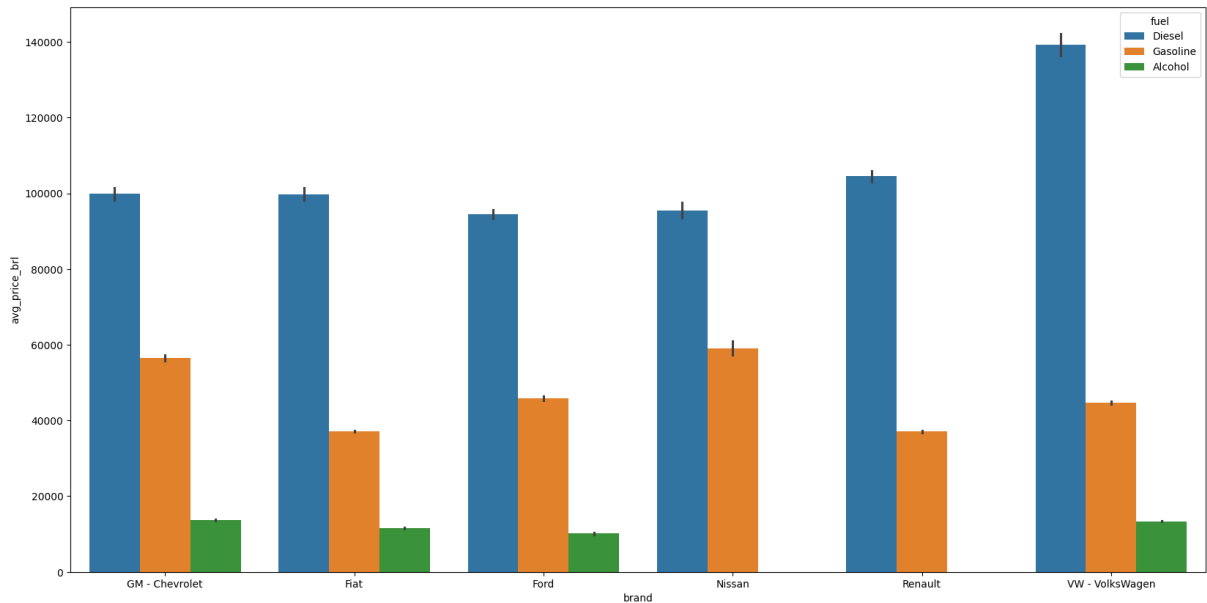


FIGURA 10 – Distribuição da média de preço dos carros por marca e tipo de combustível

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

Carros com o combustível Diesel tem os maiores preços médios em todas as marcas. O que faz sentido haja visto que motores a Diesel são comumente encontrados em carros maiores e de maiores valores. Veículos a Gasolina são os segundos com maiores preços médios em todas as marcas. Vale salientar que, nesta base, os veículos Flex (que podem consumir álcool e gasolina) estão categorizados como Gasolina. Veículos a Álcool tem os menores preços médios nas marcas GM, Fiat, Ford e VM. As marcas Nissan e Renault não apresentaram veículos com combustível álcool.

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

a. Escolha as variáveis numéricas (modelos de Regressão) para serem as variáveis independentes do modelo. A variável target é avg_price. Observação: caso julgue necessário, faça a transformação de variáveis categóricas em variáveis numéricas para inputar no modelo. Indique quais variáveis foram transformadas e como foram transformadas

```
sns.boxplot(dados['avg_price_brl']).set_title("Distribuição dos preços dos carros")
```

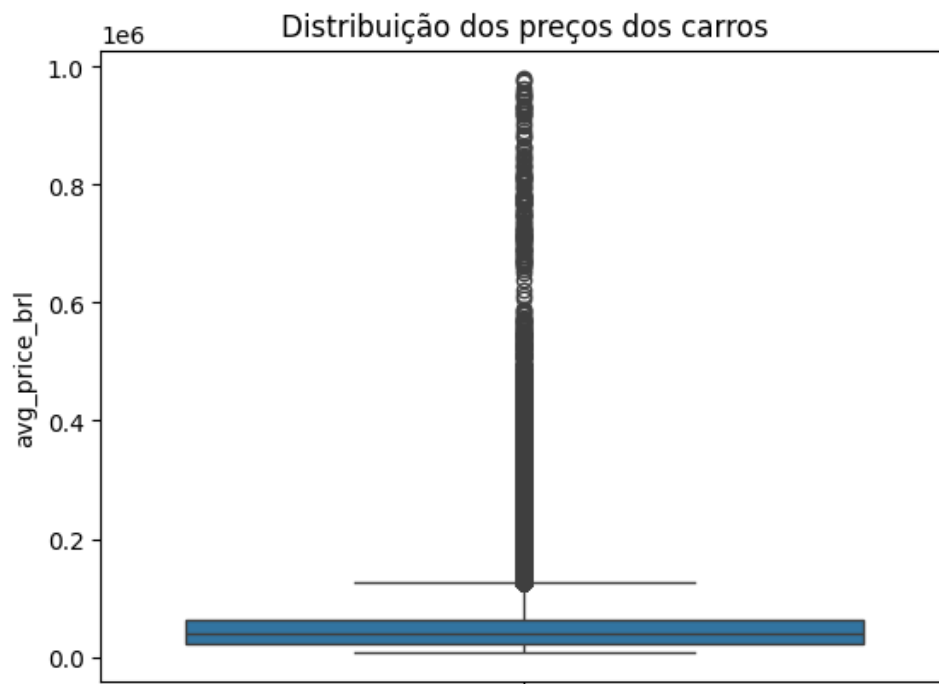


FIGURA 11 – Distribuição dos preços dos carros

```
# Transformação da coluna modelo (model) de categórica para numérica
dados['model'] = LabelEncoder().fit_transform(dados['model'])

# Transformação da coluna engrenagem (gear) de categórica para numérica
dados['gear'] = LabelEncoder().fit_transform(dados['gear'])
dados.head()

# Transformação da coluna combustível (fuel) de categórica para numérica
dados['fuel'] = LabelEncoder().fit_transform(dados['fuel'])
dados.head()
```

ID	Year	Month	FIPE Code	Auth	Brand	Model	Fuel	Gear	Engine	Year Model	Avg. Price (BRL)
0	2021.0	January	004001-0	cfzlcztzfwrcp	GM - Chevrolet	297	2	1	1	2002.0	9162.0
1	2021.0	January	004001-0	cdqwxwpw3y2p	GM - Chevrolet	297	2	1	1	2001.0	8832.0
2	2021.0	January	004001-0	cb1t3xwwj1xp	GM - Chevrolet	297	2	1	1	2000.0	8388.0
3	2021.0	January	004001-0	cb9gct6j65r0	GM - Chevrolet	297	0	1	1	2000.0	8453.0
4	2021.0	January	004003-7	g15wg0gbz1fx	GM - Chevrolet	260	2	1	1,6	2001.0	12525.0

```
# Variável dados_num contém apenas variáveis numéricas de interesse (
    exclui o restante)
dados_num = dados.drop(['year_of_reference', 'month_of_reference', '
    fiipe_code', 'authentication', 'brand', 'engine_size'],axis = 1)

sns.heatmap(dados_num.corr("spearman"), annot = True)
plt.title("Mapa de Correlação das Variáveis Numéricas\n", fontsize = 15)
```



```
plt.show()
```

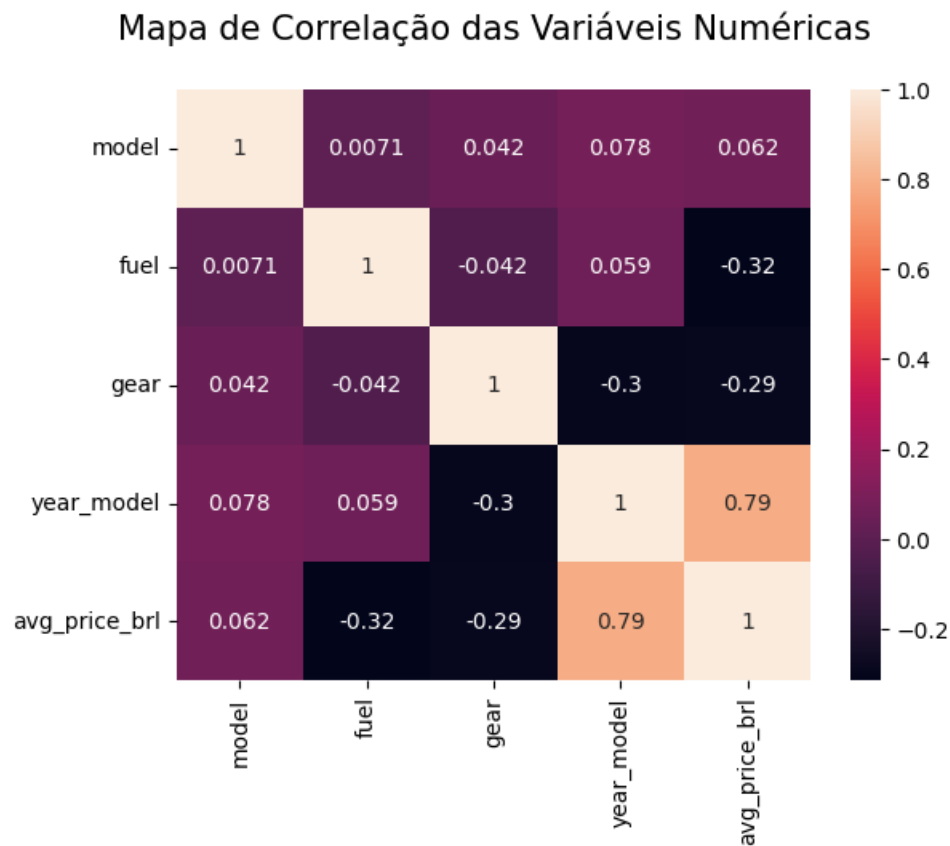


FIGURA 12 – Mapa de Correlação das Variáveis Numéricas

```
# Variável X contém apenas variáveis numéricas de interesse para a análise, excluindo a variável target
X = dados_num.drop(['avg_price_brl'], axis = 1)

Y = dados_num['avg_price_brl']
```

b. Crie partições contendo 75% dos dados para treino e 25% para teste

```
# Divisão: 30% dos dados são de teste e 70% de treinamento
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25,
                                                    random_state = 42)
```

c. Treine modelos RandomForest (biblioteca RandomForestRegressor) e XGBoost (biblioteca XGBRegressor) para predição dos preços dos carros. Observação: caso julgue necessário, mude os parâmetros dos modelos e rode novos modelos. Indique quais parâmetros foram inputados e indique o treinamento de cada modelo

```

#RandomForest
model_rf = RandomForestRegressor()
model_rf.fit(X_train, Y_train)

#RandomForest com parâmetros
n_estimators = [5,20,50,100] # number of trees in the random forest
max_features = ['auto', 'sqrt'] # number of features in consideration at every
split
max_depth = [int(x) for x in np.linspace(10, 120, num = 12)] # maximum
number of levels allowed in each decision tree
min_samples_split = [2, 6, 10] # minimum sample number to split a node
min_samples_leaf = [1, 3, 4] # minimum sample number that can be stored
in a leaf node
bootstrap = [True, False] # method used to sample data points

random_grid = {'n_estimators': n_estimators, 'max_features': max_features,
               'max_depth': max_depth, 'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf, 'bootstrap': bootstrap}

rf = RandomForestRegressor()
rf_random = RandomizedSearchCV(estimator = rf,param_distributions =
                               random_grid,
                               n_iter = 100, cv = 5, verbose=2, random_state=35, n_jobs = -1)

rf_random.fit(X_train, Y_train)

# Utilizando os melhores parâmetros encontrados
model_rf_parametros = RandomForestRegressor(max_depth=70,
      min_samples_leaf=1, min_samples_split=2, n_estimators=20,
      random_state=80)

model_rf_parametros.fit(X_train, Y_train)

#XGBoost
model_xgboost = XGBRegressor()
model_xgboost.fit(X_train, Y_train)

```

d. Grave os valores preditos em variáveis criadas

```

#RandomForest
valores_preditos_rf = model_rf.predict(X_test)
valores_preditos_rf

```

```
#RandomForest com parâmetros
valores_preditos_rf_parametros = model_rf_parametros.predict(X_test)
valores_preditos_rf_parametros
```

```
#XGBoost
valores_preditos_xgboost = model_xgboost.predict(X_test)
valores_preditos_xgboost
```

e. Realize a análise de importância das variáveis para estimar a variável target, para cada modelo treinado

```
#RandomForest
model_rf.feature_importances_
feature_importances_rf = pd.DataFrame(model_rf.feature_importances_,
    index = X_train.columns, columns=['importance']).sort_values('
    importance', ascending = False)
feature_importances_rf
```

Feature	Importance
model	0.439964
year_model	0.358480
fuel	0.179241
gear	0.022315

```
#RandomForest com parâmetros
model_rf_parametros.feature_importances_
feature_importances_rf_param = pd.DataFrame(model_rf_parametros.
    feature_importances_, index = X_train.columns, columns=['importance'
    ]).sort_values('importance', ascending = False)
feature_importances_rf_param
```

Feature	Importance
model	0.442857
year_model	0.356757
fuel	0.178231
gear	0.022155

```
#XGBoost
model_xgboost.feature_importances_
feature_importances_x = pd.DataFrame(model_xgboost.
    feature_importances_, index = X_train.columns, columns=['importance'
    ]).sort_values('importance', ascending = False)
```

```
feature_importances_x
```

Feature	Importance
fuel	0.699523
year_model	0.160642
model	0.088919
gear	0.050917

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

Tanto no modelo Random Forest, quando no modelo Random Forest com parâmetros, as variáveis mais importantes foram modelo e ano do modelo (model e year_model) com quase 80% de importância. Somando a combustível (fuel) chegamos a 97% de importância. Tornando quase irrelevante a variável engrenagem (gear). Para o modelo XGBoost temos como variável mais importante o combustível (fuel) com aproximadamente 70% e seguido de ano do modelo com 16%. Modelo e engrenagem (model e gear) aparecem com menos de 10% cada.

g. Escolha o melhor modelo com base nas métricas de avaliação MSE, MAE e R²

```
#RandomForest
mse_rf = mean_squared_error(Y_test, valores_preditos_rf)
mae_rf = mean_absolute_error(Y_test, valores_preditos_rf)
r2_rf = r2_score(Y_test, valores_preditos_rf)
mse_rf
mae_rf
r2_rf
```

```
54633997.34991135
4215.705784378993
0.97969945241549
```

```
#RandomForest com parâmetros
mse_rf_param = mean_squared_error(Y_test,
    valores_preditos_rf_parametros)
mae_rf_param = mean_absolute_error(Y_test,
    valores_preditos_rf_parametros)
r2_rf_param = r2_score(Y_test, valores_preditos_rf_parametros)
mse_rf_param
mae_rf_param
r2_rf_param
```

```
54716253.201698475
4216.618281471371
0.9796688883177797
```

```
#XGBoost
mse_x = mean_squared_error(Y_test, valores_preditos_xgboost)
mae_x = mean_absolute_error(Y_test, valores_preditos_xgboost)
r2_x = r2_score(Y_test, valores_preditos_xgboost)
mse_x
mae_x
r2_x
```

```
297039446.66991967
6429.30300193814
0.8896280024509496
```

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

Levando em conta a acurácia resultante da métrica R2, temos os seguintes resultados:

- Random Forest: 98%
- Random Forest com parâmetros: 98%
- XGBoost: 89%

Desta forma, temos um empate entre Random Forest e Random Forest com parâmetros. Para fins de entendimento, trazemos a comparação dos resultados de MSE e MAE (quanto menores, melhores os resultados):

- MSE
 - Random Forest: 54.622.503
 - Random Forest com parâmetros: 54.716.253
- MAE
 - Random Forest: 4215
 - Random Forest com parâmetros: 4217

Como podemos ver, há uma ligeira vantagem no uso do Random Forest, sem parâmetros. Assim sendo, concluímos que o melhor resultado preditivo foi obtido através do modelo Random Forest.

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

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

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

Coefficiente 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)

1. Carregue a base de dados Satellite

```
install.packages("e1071")
install.packages("randomForest")
install.packages("kernlab")
install.packages("mlbench")
install.packages("caret")
install.packages("RSNNS")

sink("./output.txt", append = T)

library(mlbench)
library(caret)
library(RSNNS)

data("Satellite")
df <- Satellite[,17:20]
df$classes <- Satellite$classes
```

2. Crie partições contendo 80% para treino e 20% para teste

```
set.seed(7)
indices <- createDataPartition(df$classes, p=0.8, list=FALSE)
treino <- df[indices, ]
teste <- df[-indices, ]
```

3. Treine modelos RandomForest, SVM e RNA para predição destes dados.

```
print(">> Iniciando treinamento rf...")
rf <- caret::train(classes~., data=treino, method="rf")
predicoes.rf <- predict(rf, teste)
confusion.rf <- caret::confusionMatrix(predicoes.rf, teste$classes)

print(">> Iniciando treinamento svm...")
svm <- caret::train(classes~., data=treino, method="svmRadial")
predicoes.svm <- predict(svm, teste)
confusion.svm <- caret::confusionMatrix(predicoes.svm, teste$classes)

print(">> Iniciando treinamento rna...")
rna <- caret::train(classes~., data=treino, method="nnet")
```

```
predicoes.rna <- predict(rna, teste)
confusion.rna <- caret::confusionMatrix(predicoes.rna, teste$classes)
```

4. Escolha o melhor modelo com base em suas matrizes de confusão.

```
print(">> Iniciando Análises de Matrizes de Confusão...")
print(confusion.rf) #acuracia = 84%
print(confusion.svm) #acuracia = 87%
print(confusion.rna) #acuracia = 80%
```

5. Indique qual modelo dá o melhor o resultado e a métrica utilizada

A métrica escolhida para comparação é a acurácia. A acurácia dos modelos são:

- RandomForest: 84%
- SVM: 87%
- RNA: 80%

O modelo de melhor acurácia é o SVM com 87%.

2 Estimativa de Volumes de Árvores

1. Carregar o arquivo Volumes.csv (<http://www.razer.net.br/datasets/Volumes.csv>)

```
install.packages("mlbench")
install.packages("caret")
install.packages("RSNNS")
install.packages("kernlab")
install.packages("randomForest")

sink("./output.txt", append = T)

library(mlbench)
library(caret)
library(RSNNS)

# load dataset
df <- read.csv("./Volumes.csv", header=TRUE, sep=";")
```

2. Eliminar a coluna NR, que só apresenta um número sequencial

```
df$DAP <- as.numeric(sub(",", ".", df$DAP, fixed = TRUE))
df$HT <- as.numeric(sub(",", ".", df$HT, fixed = TRUE))
df$HP <- as.numeric(sub(",", ".", df$HP, fixed = TRUE))
df$VOL <- as.numeric(sub(",", ".", df$VOL, fixed = TRUE))
df$NR <- NULL
summary(df)

set.seed(7)
```

3. Criar partição de dados: treinamento 80%, teste 20%

```
df[c(1, 2, 3)] <- lapply(df[c(1, 2, 3)], function(x) c(scale(x)))

indices <- createDataPartition(df$VOL, p=0.80, list=FALSE)
traindf <- df[indices,]
testdf <- df[-indices,]

str(df)
```

4. Usando o pacote "caret", treinar os modelos: Random Forest (rf), SVM (svmRadial), Redes Neurais (neuralnet) e o modelo alométrico de SPURR

```
# modelo alométrico de SPURR

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

predictalom <- predict(alom, testdf)
print(predictalom)

train_control <- trainControl(method = "cv", number = 10)

rf <- caret::train(VOL~., data=traindf, method="rf", trainControl=train_control)

svm <- caret::train(VOL~., data=traindf, method="svmRadial", trainControl=train_control)

rna <- caret::train(VOL~., data=traindf, method="nnet", trainControl=train_control)
```

5. Efetue as previsões nos dados de teste

```
predictionsrf <- predict(rf, testdf)
predictionssvm <- predict(svm, testdf)
predictionsrna <- predict(rna, testdf)
```

6. Crie suas próprias funções (UDF) e calcule as seguintes métricas entre a previsão e os dados observados

```
RSQUARE = function(y_actual,y_predict){
  cor(y_actual,y_predict)^2
}

LR_R = RSQUARE(testdf[,4], predictionsrf)
LR_R = RSQUARE(testdf[,4], predictionssvm)
LR_R = RSQUARE(testdf[,4], predictionsrna)
LR_R = RSQUARE(testdf[,4], predictalom)

syx <- function(vol, pred) {
  residual <- vol - pred
  sse <- sum(residual^2)
  n <- length(vol) - 2
  syx <- sqrt(sse / n)
  return(syx)
}

syx_r <- syx(testdf[,4], predictionsrf)
syx_r <- syx(testdf[,4], predictionssvm)
syx_r <- syx(testdf[,4], predictionsrna)
syx_r <- syx(testdf[,4], predictalom)

syx_percentual <- function(vol, pred) {
  syx_v <- syx(vol, pred)
  media_dados <- mean(vol)
  syx_percentual <- (syx_v / media_dados) * 100
  return(syx_percentual)
}

syxp_r <- syx_percentual(testdf[,4], predictionsrf)
syxp_r <- syx_percentual(testdf[,4], predictionssvm)
syxp_r <- syx_percentual(testdf[,4], predictionsrna)
syxp_r <- syx_percentual(testdf[,4], predictalom)
```

7. Escolha o melhor modelo.

Três métricas foram calculadas para os modelos. São elas, Coeficiente de determinação R^2 , Erro padrão da estimativa Syx, e porcentagem de Syx. A seguir os valores obtidos pelas métricas em cada um dos modelos.

R^2 (Quanto mais perto de 1 melhor)

- RandomForest: 0.8589349
- SVM: 0.856365
- NNET: NA
- SSPUR: 0.8546826

SYX (Quanto mais perto de 0 melhor)

- RandomForest: 0.1445527
- SVM: 0.1470481
- NNET: 0.49606
- SSPUR: 0.1574296

SYX% (Quanto mais perto de 0 melhor)

- RandomForest: 11.07658%
- SVM: 11.2678%
- NNET: 38.01139%
- SSPUR: 12.0633%




Como forma de melhorar os resultados, tentamos fazer a normalização dos dados como etapa de pré-processamento. Essa etapa de “dimensionamento” assegura que os dados de todas as colunas estejam dentro do mesmo intervalo de valores, $[-2,2]$. Isso evita que a rede possa acabar “perdendo” *features* com valores muito pequenos, e que seriam relevantes durante o treinamento, em seus cálculos, privilegiando *features* com valores muito mais altos.

Após realizar a normalização e padronização dos dados, com a aplicação da função **scale** nas colunas DAP, HP e HT da base de dados, pudemos observar uma




leve melhora em algumas métricas, mas principalmente, conseguimos um resultado muito mais expressivo para o treinamento da RNA.

Ao mesmo tempo, o resultado do modelo alométrico de SPURR teve uma grande piora na métrica R2, provavelmente porque ele funciona através de uma aproximação à função de volume e a mudança dos valores interfira nos cálculos.




R2

- RandomForest: 0.8601537 
- SVM: 0.856365
- NNET: 0.4843006 
- SSPUR: 0.5317544 

SYX

- RandomForest: 0.1438805 
- SVM: 0.1470481
- NNET: 0.4960265 
- SSPUR: 0.3854788 

SYX%

- RandomForest: 11.02507% 
- SVM: 11.2678%
- NNET: 38.00883% 
- SSPUR: 29.53794% 

Escolhemos o modelo RandomForest como melhor por ter se saído ligeiramente melhor em todas as métricas calculadas.

APÊNDICE 4 – ESTATÍSTICA APLICADA I

A - ENUNCIADO

1 Gráficos e tabelas

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

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

- a) **(40 pontos)** (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:

- 1) **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.**
- 2) **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

```
install.packages("car")
install.packages("fdth")
library(car)
library(fdth)

load("salarios.RData")

# Boxplots
Boxplot(salarios$age, data=salarios, id=list(method="y"), ylab="Idade das
esposas")
Boxplot(salarios$husage, data=salarios, id=list(method="y"), ylab="Idade
dos maridos")

# Histogramas
hist(salarios$age, breaks = 5, xlab="Idade das esposas", ylab = "Frequency
")
hist(salarios$husage, breaks = 5, xlab="Idade dos maridos", ylab = "
Frequency")
```

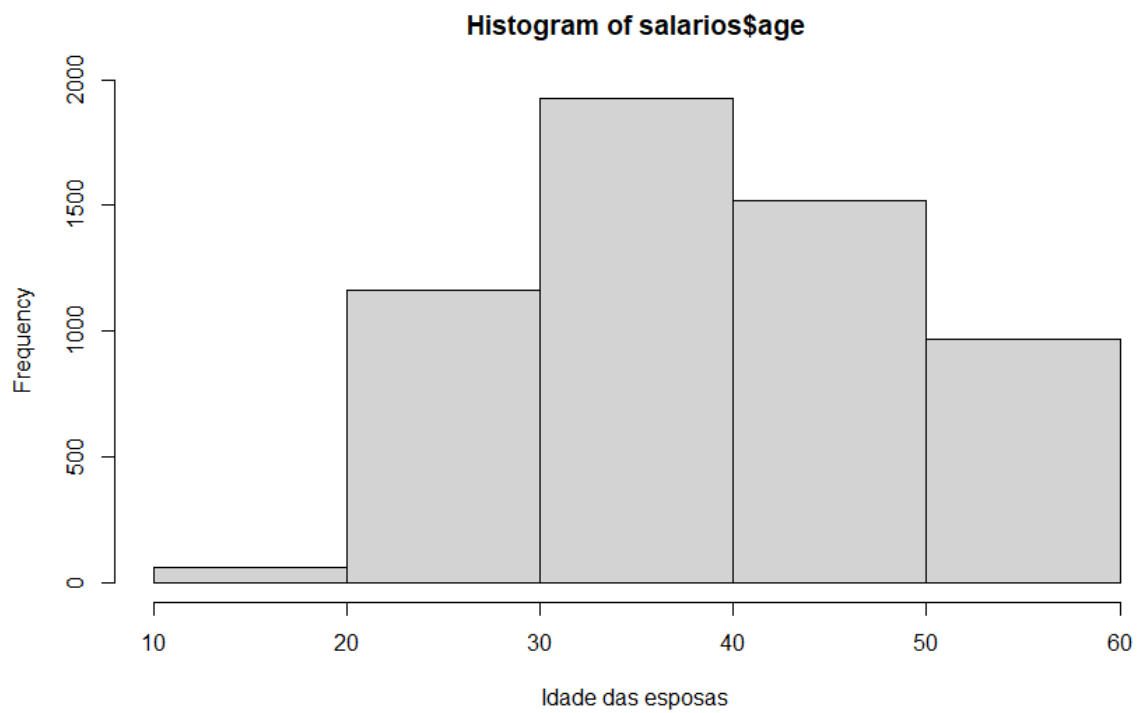



FIGURA 15 – Histograma Idade das esposas



FIGURA 16 – Histogram Idade das maridos

Analisando os gráficos box-plot podemos observar que as medianas das es-

posas e dos maridos parecem iguais, mas os limites superiores e inferiores diferem bastante, assim como a amplitude interquartílica. No caso o gráfico dos maridos possui o limite superior maior que as esposas além de possuir outliers. Já o gráfico das esposas tem maior amplitude interquartílica que o dos maridos.

Analisando os histogramas, observamos que há uma presença maior de esposas e maridos com idades entre 20 e 50 anos. À partir deste recorte a quantidade dos maridos cai drasticamente, porém, se alongando até acima de 80 anos. Já as mulheres, tem presença considerável até 60 anos e nenhum elemento acima desta idade.

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

```
# TABELA DE FREQUENCIA DA VARIÁVEL AGE (IDADE DA ESPOSA)
table_esposas <- fdt(salarios$age)
```

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

```
# TABELA DE FREQUENCIA DA VARIÁVEL HUSAGE (IDADE DO MARIDO)
table_maridos <- fdt(salarios$husage)
```

Class limits	f	rf	rf(%)	cf	cf(%)
[18.81,23.671)	102	0.02	1.81	102	1.81
[23.671,28.531)	466	0.08	8.27	568	10.08
[28.531,33.392)	809	0.14	14.36	1377	24.44
[33.392,38.253)	895	0.16	15.89	2272	40.33
[38.253,43.114)	917	0.16	16.28	3189	56.60
[43.114,47.974)	629	0.11	11.16	3818	67.77

[47.974,52.835)	649	0.12	11.52	4467	79.29
[52.835,57.696)	541	0.10	9.60	5008	88.89
[57.696,62.556)	394	0.07	6.99	5402	95.88
[62.556,67.417)	152	0.03	2.70	5554	98.58
[67.417,72.278)	51	0.01	0.91	5605	99.49
[72.278,77.139)	21	0.00	0.37	5626	99.86
[77.139,81.999)	6	0.00	0.11	5632	99.96
[81.999,86.86)	2	0.00	0.04	5634	100.00

Comparando as tabelas de frequência, podemos observar que as classes de frequências mais populosas para as esposas são de 29 a 35 anos e para os maridos de 33 a 43 anos. E a dispersão de idades é maior entre os maridos.

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

```
# MEDIA
mean(salarios$age) # 39.42758
mean(salarios$husage) # 42.45296

((39.42758/42.45296)-1)*100 # -7.126429

# A media das idades das esposas eh de aproximadamente
# 7% menor do que dos maridos.
```

```
# MEDIANA
median(salarios$age) # 39
median(salarios$husage) # 41

((39/41)-1)*100 # -4.878049

# A mediana das idades das esposas eh de aproximadamente
# 5% menor do que dos maridos.
```

```
# MODA
table(salarios$age)
subset(table(salarios$age),
  table(salarios$age) == max(table(salarios$age))) # 37

table(salarios$husage)
subset(table(salarios$husage),
```

```
table(salarios$husage) == max(table(salarios$husage))) # 44
```

```
# A moda da idade das esposas eh de 37 anos, com 217 pessoas.
# A moda da idade dos maridos eh de 44 anos, com 201 pessoas.
```

```
((37/44)-1)*100 # -15.90909
```

```
# A moda de idade das esposas eh aproximadamente
# 16% menor do que dos maridos.
```

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

```
# VARIANCIA
```

```
var(salarios$age)
```

```
var(salarios$husage)
```

```
# A variancia das idades das esposas eh 99.75234
```

```
# A variancia das idades dos maridos eh 126.0717
```

```
((99.75234/126.0717)-1)*100 # -20.8765
```

```
# A variancia das idades das esposas eh aproximadamente
```

```
# 21% menor do que a variancia das idades dos maridos
```

```
# DESVIO PADRAO
```

```
sd(salarios$age)
```

```
sd(salarios$husage)
```

```
# O desvio padrao das idades das esposas eh 9.98761
```

```
# O desvio padrao das idades dos maridos eh 11.22817
```

```
((9.98761/11.22817)-1)*100 # -11.04864
```

```
# O desvio padrao das idades das esposas eh aproximadamente
```

```
# 11% menor que dos maridos
```

```
# COEFICIENTE DE VARIACAO DAS VARIÁVEIS
```

```
meanE <- mean(salarios$age)
```

```
meanM <- mean(salarios$husage)
```

```
sdE <- sd(salarios$age)
```

```
sdM <- sd(salarios$husage)
```

```

cvM <- (sdM/meanM)*100 # 26.44849
cvM
cvE <- (sdE/meanE)*100 # 25.33153
cvE

# O coeficiente de variacao do rendimento das esposas
# eh de aproximadamente 26% e dos maridos eh de
# aproximadamente 25%.
# Ambos possuem dispersao media.

```

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.

Checagens preliminares para verificar as exigências do teste: 1) Amostras independentes, 2) normalidade e 3) homogeneidade das variâncias entre grupos 4) outliers

- 1) Amostras independentes: Sim, pois os grupos de esposas e maridos provêm de indivíduos distintos.
- 2) Normalidade: Não, utilizando a regra de bolso, como o p-value eh inferior a 0.05, a variavel “idade” NÃO é normalmente distribuída .

```

# Executando teste de normalidade de Kolmogorov–Smirnov
lillie.test(idade$idade)
# p-value = 0.00000000000000022

```

- 3) Homogeneidade das variâncias: Não. Conforme observa-se no teste abaixo, o valor da estatística F calculada é 0.79123. Como esse valor se encontra na região de rejeição de H₀, então rejeitamos a hipótese de que as variâncias são estatisticamente iguais.

```

# Vamos usar o teste F com as seguintes hipoteses:

# H0: As variancias sao estatisticamente iguais(homogeneas)
# HA: As variancias nao sao estatisticamente iguais(homogeneas)

```

```
# Executando o teste F
res.ftest <- var.test(idade ~ group, data = idades)
res.ftest

# Obtendo o valor tabelado da distribuicao F
qf(0.95, 5633, 5633)
# temos F=1,04
# para a outra cauda temos:
1/1.04
# F = 0,96

# Vamos construir o grafico:
dist_f(f = 1.04, deg.f1 = 5633, deg.f2 = 5633)
dist_f(f = 0.96, deg.f1 = 5633, deg.f2 = 5633)
```

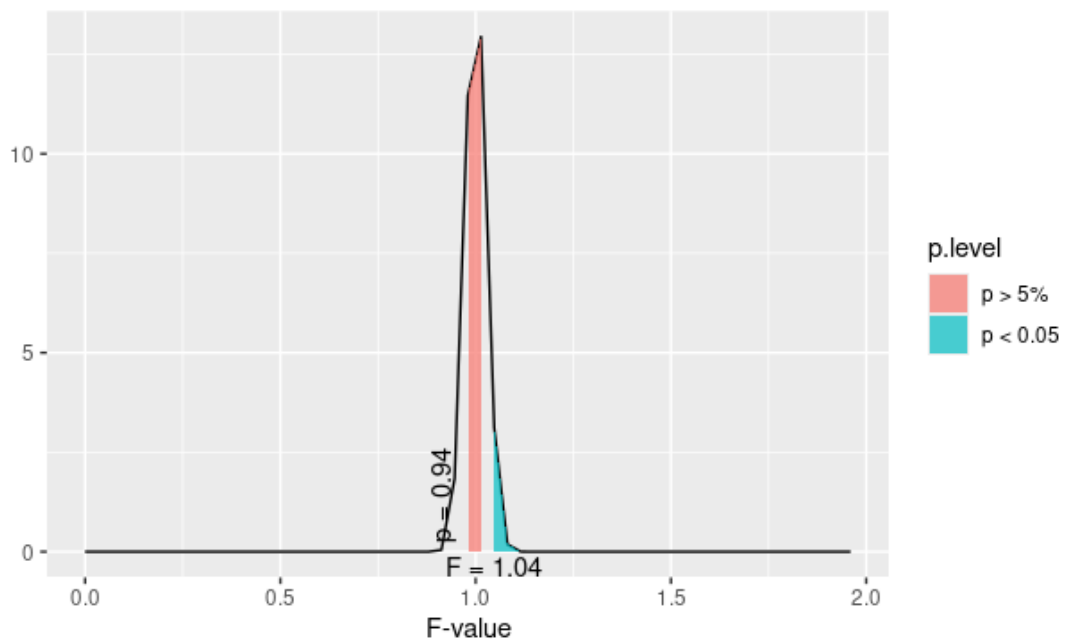


FIGURA 17 – Gráfico da distribuição F

4) Outliers: Sim. Ao observar o boxplot dos maridos percebe-se alguns outliers.

Dada as checagens anteriores de que a distribuição da variável idade NÃO é normalmente distribuída, as variâncias não serem estatisticamente iguais e também haverem outliers, não podemos usar um teste paramétrico para testar os dados, portanto utilizaremos o teste não paramétrico Mann Whitney U Test.

Queremos saber se a idade mediana das esposas difere da idade mediana dos maridos. Portanto, testando se a idade mediana dos maridos é igual a idade mediana das esposas temos as seguintes hipóteses:

- H_0 : A idade mediana dos maridos é estatisticamente igual a idade mediana das esposas
- H_a : A idade mediana dos maridos não é estatisticamente igual a idade mediana das esposas

```
# Executando o teste Mann Whitney U Test
res <- wilcox.test(idade ~ group, data = idades,
                  exact = FALSE, conf.int=TRUE)
res
```

```
Wilcoxon rank sum test with continuity correction

data:  idade by group
W = 13619912, p-value < 0.00000000000000022
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -3.000024 -2.000033
sample estimates:
difference in location
      -2.999966
```

O p-value do teste é 0.00000000000000022, que é menor que o nível de significância 0,05. Podemos concluir que a idade mediana dos maridos é estatisticamente diferente da idade mediana das esposas (rejeitamos H_0). O intervalo de confiança da diferença entre as medianas está entre -3.000024 e -2.000033, com uma mediana de -2.999966.

APÊNDICE 5 – ESTATÍSTICA APLICADA II

A - ENUNCIADO

1 Regressões Ridge, Lasso e ElasticNet

- a) **(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)
husbck = 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

a) 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).

PADRONIZAÇÃO DOS DADOS

```
cols = c('husage', 'husearns', 'huseduc', 'hushrs', 'earns', 'age', 'educ', '
  exper', 'lwage')
pre_proc_val <- preProcess(train[,cols], method = c("center", "scale"))

train[,cols] = predict(pre_proc_val, train[,cols])
test[,cols] = predict(pre_proc_val, test[,cols])

cols_reg = c('husage', 'husunion', 'husearns', 'huseduc', 'husblk', 'hushisp',
  'hushrs',
  'kidge6', 'age', 'black', 'educ', 'hispanic', 'union', 'exper', 'kidlt6', 'lwage')

dummies <- dummyVars(lwage~husage+husunion+husearns+huseduc+
  husblk+
  hushisp+hushrs+kidge6+age+black+educ+hispanic+union+exper+kidlt6,
  data = dat[,cols_reg])
train_dummies = predict(dummies, newdata = train[,cols_reg])
test_dummies = predict(dummies, newdata = test[,cols_reg])

x_train = as.matrix(train_dummies)
y_train = train$lwage

x_test = as.matrix(test_dummies)
y_test = test$lwage
```

REGRESSÃO RIDGE

```
lambdas <- 10^seq(5, -5, by = -1)
ridge_lamb <- cv.glmnet(x_train, y_train, alpha = 0, lambda = lambdas,
  nfolds = 10)

best_lambda_ridge <- ridge_lamb$lambda.min
best_lambda_ridge
```

```

ridge_reg = glmnet(x_train, y_train, nlambda = 50, alpha = 0, family = '
    gaussian',
lambda = best_lambda_ridge)
ridge_reg

# Resultado
# (coeficientes)
ridge_reg[["beta"]]

predictions_train_ridge <- predict(ridge_reg, s = best_lambda_ridge, newx
    = x_train)

```

Visualizando o resultado da estimativa dos coeficientes percebemos que nem todas as variáveis explicam a variável dependente. Infelizmente as que explicam tem valor explicativo muito baixo, sendo somente as com certa relevância “husearns”, “husblk”, “hushisp”, “kidge6”, “black”, “educ” e “union”.

```

15 x 1 sparse Matrix of class "dgCMatrix"
      s0
husage    0.02651324
husunion -0.04174920
husearns  0.23671354
huseduc   0.04804650
husblk    0.24319544
hushisp   0.11457810
hushrs    -0.07293064
kidge6    -0.16442536
age        0.04466390
black     -0.29066663
educ       0.30859463
hispanic  -0.09979930
union      0.41614645
exper     -0.01411404
kidlt6    -0.02475760

```

REGRESSÃO LASSO

```

lambdas <- 10^seq(5, -5, by = -.1)

lasso_lamb <- cv.glmnet(x_train, y_train, alpha = 1, lambda = lambdas,
    standardize = TRUE, nfolds = 10)

```

```

best_lambda_lasso <- lasso_lambd$lambda.min
best_lambda_lasso

lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda_lasso,
  lasso,
  standardize = TRUE)

lasso_model[["beta"]]

predictions_train_lasso <- predict(lasso_model, s = best_lambda_lasso,
  newx = x_train)

```

Visualizando o resultado da estimativa dos coeficientes percebemos que nem todas as variáveis explicam a variável dependente. Infelizmente as que explicam tem valor explicativo muito baixo, sendo somente as com certa relevância “husearns”, “husblk”, “hushisp”, “kidge6”, “black”, “educ” e “union”. Enquanto a variável “exper” teve tão pouco valor explicativo que foi a zero.

```

15 x 1 sparse Matrix of class "dgCMatrix"
      s0
usage    0.02406805
union    -0.04144957
husearns  0.23903324
huseduc   0.04539082
husblk    0.26797943
hushisp   0.11329929
hushrs    -0.07369940
kidge6    -0.16534640
age       0.03331493
black     -0.31556379
educ      0.31552507
hispanic  -0.09716331
union     0.41807407
exper     .
kidlt6    -0.02464497

```

REGRESSÃO ELASTICNET

```

train_cont <- trainControl(method = "repeatedcv",
  number = 10,

```

```

        repeats = 5,
        search = "random",
        verboselter = FALSE)

elastic_reg <- train(lwage~husage+husunion+husearns+huseduc+husbck+
  hushisp+
  hushrs+kidge6+age+black+educ+hispanic+union+exper+kidlt6,
  data = train,
  method = "glmnet",
  tuneLength = 20,
  trControl = train_cont)

# O melhor parametro alpha escolhido eh:
#   alpha   lambda
#15  0.8015939  0.01293922
elastic_reg$bestTune

# E os parametros sao:
#elastic_reg[["finalModel"]][["beta"]]

predictions_train_elasticnet <- predict(elastic_reg, x_train)

```

COMPARANDO RESULTADOS

```

eval_results <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))

  # As metricas de performace do modelo:
  data.frame(
    RMSE = RMSE,
    Rsquare = R_square
  )
}

eval_with_params <- function(model, best_lambda = NULL) {
  husage = (40-pre_proc_val[["mean"]][["husage"]])/pre_proc_val[["std"]][["
    husage"]]
  husunion = 0
  husearns = (600-pre_proc_val[["mean"]][["husearns"]])/

```

```

pre_proc_val[["std"]][["husearns"]]
huseduc = (13-pre_proc_val[["mean"]][["huseduc"]])/pre_proc_val[["std"]][["huseduc"]]
husblck = 1
hushisp = 0
hushrs = (40-pre_proc_val[["mean"]][["hushrs"]])/pre_proc_val[["std"]][["hushrs"]]
kidge6 = 1
age = (38-pre_proc_val[["mean"]][["age"]])/pre_proc_val[["std"]][["age"]]
black = 0
educ = (13-pre_proc_val[["mean"]][["educ"]])/pre_proc_val[["std"]][["educ"]]
hispanic = 1
union = 0
exper = (18-pre_proc_val[["mean"]][["exper"]])/pre_proc_val[["std"]][["exper"]]
kidlt6 = 1

our_pred = as.matrix(data.frame(husage=husage,
                                husunion=husunion,
                                husearns=husearns,
                                huseduc=huseduc,
                                husblck=husblck,
                                hushisp=hushisp,
                                hushrs=hushrs,
                                kidge6=kidge6,
                                age=age,
                                black=black,
                                educ=educ,
                                hispanic=hispanic,
                                union=union,
                                exper=exper,
                                kidlt6=kidlt6))

prediction <- predict(model, s = best_lambda, newx = our_pred)

if (is.null(best_lambda)) {
  prediction <- predict(model, our_pred)
}

wage_pred=(prediction*pre_proc_val[["std"]][["lwage"]])+
pre_proc_val[["mean"]][["lwage"]]

```

```

cat("Predicao: ", wage_pred, "\n")
cat("Predicao exp: ", exp(wage_pred), "\n")

return(wage_pred)
}

show_values <- function(wage_pred) {

  # O intervalo de confianca
  n <- nrow(train)
  m <- wage_pred
  s <- pre_proc_val[["std"]][["lwage"]]
  dam <- s/sqrt(n)
  Cllwr <- m + (qnorm(0.025))*dam # intervalo inferior
  Clupr <- m - (qnorm(0.025))*dam # intervalo superior

  cat("Intervalo inferior: ", Cllwr, "\n")
  cat("Intervalo superior: ", Clupr, "\n")

  cat("Intervalo inferior exp: ", exp(Cllwr), "\n")
  cat("Intervalo superior exp: ", exp(Clupr), "\n")
}

# ----- RIDGE ----- #

eval_results(y_train, predictions_train_ridge, train)

predictions_test <- predict(ridge_reg, s = best_lambda_ridge, newx = x_
  test)
eval_results(y_test, predictions_test, test)

predict_our_ridge <- eval_with_params(ridge_reg, best_lambda = best_
  lambda_ridge)

show_values(predict_our_ridge)

# ----- LASSO ----- #

eval_results(y_train, predictions_train_lasso, train)

predictions_test <- predict(lasso_model, s = best_lambda_lasso, newx = x_
  test)

```



```

eval_results(y_test, predictions_test, test)

predict_our_lasso <- eval_with_params(lasso_model, best_lambda = best_
  lambda_lasso)

show_values(predict_our_lasso)

# ----- ELATICNET ----- #

eval_results(y_train, predictions_train_elasticnet, train)
predictions_test <- predict(elastic_reg, x_test)
eval_results(y_test, predictions_test, test)

predict_our_elastic <- eval_with_params(elastic_reg)

show_values(predict_our_elastic)

```

Observando a Tabela 1 abaixo observamos que os valores para as métricas de todos os modelos são muito próximas e também não muito boas, não descartando a hipótese de overfitting ou underfitting. Mesmo assim, a título de comparação observamos que o modelo Lasso se saiu ligeiramente melhor do que os outros para o conjunto de dados de treinamento, enquanto o modelo Ridge se saiu ligeiramente melhor nos dados de teste.

TABELA 1 – Tabela de métricas dos modelos

	Treinamento		Teste	
	RMSE	Rsquare	RMSE	Rsquare
Ridge	0.8485564	0.2796022	0.8488229	0.3004756
Lasso	0.8485375	0.2796343	0.8488541	0.3004242
ElasticNet	0.8495878	0.2778498	0.8490653	0.3000762

Na Tabela 2 pode-se observar os valores preditos pelos modelos para as variáveis especificadas inicialmente.

TABELA 2 – Tabela de Predição e Intervalos de Confiança

	Predição	Intervalos de Confiança
Ridge	8.633904	8.442068 8.8301
Lasso	8.746603	8.552263 8.945359
ElasticNet	8.160874	7.979548 8.34632

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 Construção de Características: Identificador automático de idioma

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.",
]

A “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 portugues:
    pre_padroes.append( [frase, 'português'])

random.shuffle(pre_padroes)

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

```

ID	Frase	Idioma
0	Me gusta cocinar la cena en casa.	espanhol
1	Vamos fazer uma viagem de carro.	português
2	Do you have any recommendations for restaurants?	inglês
3	Quero aprender a tocar violão.	português
4	Hola, ¿cómo estás?	espanhol
...
87	Vamos sair para jantar no sábado.	português
88	Estoy emocionado por el concierto.	espanhol
89	Estoy aprendiendo a tocar el piano.	espanhol
90	Preciso fazer exercícios físicos regularmente.	português
91	Quiero aprender italiano.	espanhol

TABELA 3 – Frases com seus respectivos idiomas

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
from sklearn.feature_extraction.text import CountVectorizer

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 temCaractere(texto, simbolo):
    return int(simbolo in texto)

def gerarNGrams(list_textos, n, max_f):
    vec = CountVectorizer(analyzer='char', ngram_range=(n,n), max_features
        =max_f)
    ngrams = vec.fit_transform(list_textos)
    feature_names = vec.get_feature_names_out()

    return feature_names

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)

    #ESTRATEGIA 1: buscando caracteres especiais
    caracteristica2=temCaractere(texto, 'ç')
    caracteristica3=temCaractere(texto, 'ñ')
    caracteristica12=temCaractere(texto, 'à')
    caracteristica13=temCaractere(texto, 'ón')
    caracteristica14=temCaractere(texto, 'ã')
    caracteristica15=temCaractere(texto, 'ay')
    caracteristica16=temCaractere(texto, 'yo')

    #ESTRATEGIA 2: buscando caracteres iguais sucessivos
    caracteristica4=temCaractere(texto, 'cc')
    caracteristica5=temCaractere(texto, 'oo')
    caracteristica6=temCaractere(texto, 'll')
    caracteristica7=temCaractere(texto, 'ss')

```

```

caracteristica8=temCaractere(texto, 'mm')
caracteristica9=temCaractere(texto, 'nn')
caracteristica10=temCaractere(texto, 'rr')
caracteristica11=temCaractere(texto, 'ee')

#ESTRATEGIA 3: buscando ocorrencia dos ngrams mais frequentes por
lingua
features_ngrams = []

for n in ngrams_populares_p_lingua:
    features_ngrams.append(int(n in texto))

# acrescenta as suas funcoes no vetor padrao
padrao = [caracteristica1, caracteristica2, caracteristica3,
          caracteristica4, caracteristica5, caracteristica6,
          caracteristica7, caracteristica8, caracteristica9,
          caracteristica10, caracteristica11,
          caracteristica12, caracteristica13, caracteristica14, caracteristica15,
          caracteristica16,
          *features_ngrams,
          frase[1]]

return padrao

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

#GERA 88 1,2,3,4-GRAMS MAIS COMUNS PARA CADA LINGUA
#set para evitar repetidos
ngrams_populares_p_lingua = set()
max_features = 38 # ou 88
for nsize in range(1, 5): # ou 4
    ngrams_populares_p_lingua.update(gerarNGrams(portugues, nsize,
        max_features))
    ngrams_populares_p_lingua.update(gerarNGrams(ingles, nsize,
        max_features))
    ngrams_populares_p_lingua.update(gerarNGrams(espanhol, nsize,

```



```

max_features))

# 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(data=padroes, columns=['TMediaFrase', 'ç', 'ñ', 'cc', '
oo', 'll', 'ss', 'mm', 'nn', 'rr', 'ee', 'à', 'ón', 'ã', 'ay', 'yo', *
ngrams_populares_p_lingua, 'Lingua'])
dados

```

Treinando o modelo com SVM

Separando o conjunto de treinamento do conjunto de testes

```

from sklearn.model_selection import train_test_split
import numpy as np

#from sklearn.metrics import confusion_matrix

vet = np.array(padroes)
classes = vet[:, -1]      # classes = [p[-1] for p in padroes]
#print(classes)
padroes_sem_classe = vet[:, 0:-1]
#print(padroes_sem_classe)
X_train, X_test, y_train, y_test = train_test_split(padroes_sem_classe,
                                                    classes, test_size=0.20, stratify=classes)

```

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))

```

```

Acuracia nos dados de treinamento: 98.63%
[[24  0  0]
 [ 0 24  0]
 [ 1  0 24]]

```

	precision	recall	f1-score	support
espanhol	0.96	1.00	0.98	24
ingles	1.00	1.00	1.00	24
portugues	1.00	0.96	0.98	25
accuracy			0.99	73
macro avg	0.99	0.99	0.99	73
weighted avg	0.99	0.99	0.99	73

```

metricas mais confiaveis
[[6 0 0]
 [0 5 1]
 [4 0 3]]

```

	precision	recall	f1-score	support
espanhol	0.60	1.00	0.75	6
ingles	1.00	0.83	0.91	6
portugues	0.75	0.43	0.55	7

accuracy			0.74	19
macro avg	0.78	0.75	0.73	19
weighted avg	0.78	0.74	0.72	19

2 Melhore uma base de dados ruim

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.preprocessing import scale
from sklearn.preprocessing import minmax_scale
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
df.describe()
```

Stat	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets
count	299.000	299.000	299.000	299.000	299.000	299.000	299.000
mean	60.834	0.431	581.839	0.418	38.084	0.351	263358.029
std	11.895	0.496	970.288	0.494	11.835	0.478	97804.237
min	40.000	0.000	23.000	0.000	14.000	0.000	25100.000
25%	51.000	0.000	116.500	0.000	30.000	0.000	212500.000
50%	60.000	0.000	250.000	0.000	38.000	0.000	262000.000
75%	70.000	1.000	582.000	1.000	45.000	1.000	303500.000
max	95.000	1.000	7861.000	1.000	80.000	1.000	850000.000

Stat	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
count	299.000	299.000	299.000	299.000	299.000	299.000
mean	1.394	136.625	0.649	0.321	130.261	0.321
std	1.035	4.412	0.478	0.468	77.614	0.468
min	0.500	113.000	0.000	0.000	4.000	0.000
25%	0.900	134.000	0.000	0.000	73.000	0.000
50%	1.100	137.000	1.000	0.000	115.000	0.000
75%	1.400	140.000	1.000	1.000	203.000	1.000
max	9.400	148.000	1.000	1.000	285.000	1.000

```
import seaborn as sns
corr = df.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
```

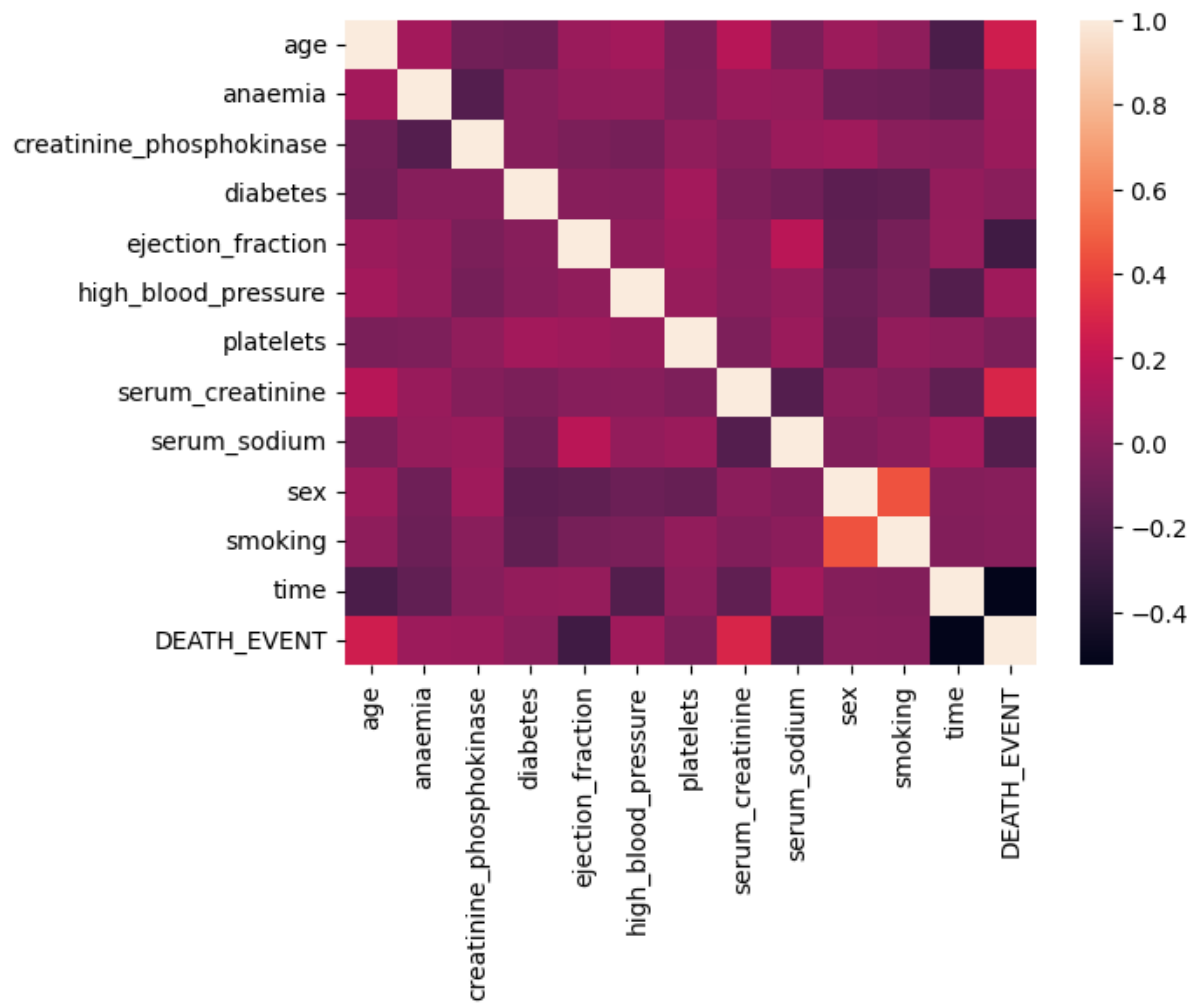


FIGURA 18 – Mapa de correlação dados clínicos

```

X_orig = df.iloc[:, :-1]
y_orig = df['DEATH_EVENT']

X_train_orig, X_test_orig, y_train, y_test = train_test_split(X_orig, y_orig,
    test_size=0.20, stratify=y_orig, random_state=10)

treinador = svm.SVC() #algoritmo escolhido

modelo_orig = treinador.fit(X_train_orig, y_train)

# predição com os mesmos dados usados para treinar
y_pred = modelo_orig.predict(X_train_orig)
cm_orig_train = confusion_matrix(y_train, y_pred)
print('Matriz de confusao – com os dados ORIGINAIS usados no
TREINAMENTO')
#print(cm_orig_train)

```

```

print(classification_report(y_train, y_pred, zero_division=0))

# predição com os mesmos dados usados para testar
y2_pred = modelo_orig.predict(X_test_orig)
cm_orig_test = confusion_matrix(y_test, y2_pred)
print('Matriz de confusao – com os dados ORIGINAIS usados para
      TESTES')
#print(cm_orig_test)
print(classification_report(y_test, y2_pred, zero_division=0))

```

```

Matriz de confusao - com os dados ORIGINAIS usados no TREINAMENTO
      precision    recall  f1-score   support

```

```

      0      0.68      1.00      0.81      162
      1      0.00      0.00      0.00       77

 accuracy      0.68      239
 macro avg      0.34      0.50      0.40      239
weighted avg      0.46      0.68      0.55      239

```

```

Matriz de confusao - com os dados ORIGINAIS usados para TESTES
      precision    recall  f1-score   support

```

```

      0      0.68      1.00      0.81      41
      1      0.00      0.00      0.00      19

 accuracy      0.68      60
 macro avg      0.34      0.50      0.41      60
weighted avg      0.47      0.68      0.55      60

```

```

y = df['DEATH_EVENT']
X = df.drop(['diabetes', 'time', 'DEATH_EVENT'],axis = 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify
=y_orig,random_state=10)

scaler = StandardScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

treinador = svm.SVC() #algoritmo escolhido

```

```

modelo_orig = treinador.fit(X_train_scaled, y_train)

# predição com os mesmos dados usados para treinar
y_pred = modelo_orig.predict(X_train_scaled)
cm_orig_train = confusion_matrix(y_train, y_pred)
print('Matriz de confusao – com os dados escalonados e de alta correlacao
      usados no TREINAMENTO')
#print(cm_orig_train)
print(classification_report(y_train, y_pred, ))

# predição com os mesmos dados usados para testar
y2_pred = modelo_orig.predict(X_test_scaled)
cm_orig_test = confusion_matrix(y_test, y2_pred)
print('Matriz de confusao – com os dados escalonados e de alta correlacao
      usados no TESTE')
#print(cm_orig_test)
print(classification_report(y_test, y2_pred))

```

Matriz de confusao - com os dados escalonados e de alta correlacao usados no TREINAMENTO

	precision	recall	f1-score	support
0	0.83	0.96	0.89	162
1	0.88	0.60	0.71	77
accuracy			0.85	239
macro avg	0.86	0.78	0.80	239
weighted avg	0.85	0.85	0.84	239

Matriz de confusao - com os dados escalonados e de alta correlacao usados no TESTE

	precision	recall	f1-score	support
0	0.78	0.85	0.81	41
1	0.60	0.47	0.53	19
accuracy			0.73	60
macro avg	0.69	0.66	0.67	60
weighted avg	0.72	0.73	0.72	60

APÊNDICE 7 – 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 8 – 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 9 – DEEP LEARNING

A - ENUNCIADO

9.0.1 Classificação de Imagens (CNN)

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

9.0.2 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.

9.0.3 Gerador de Dígitos Fake (GAN)

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

9.0.4 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 10 – 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 11 – VISÃO COMPUTACIONAL

A - ENUNCIADO

11.0.1 Extração de Características

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

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

Tarefas:

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

h) Indique qual modelo dá o melhor o resultado e a métrica utilizada

11.0.2 Redes Neurais

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

Tarefas:

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

B - RESOLUÇÃO

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 12 – GESTÃO DE PROJETOS DE IA

A - ENUNCIADO

12.0.1 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>

12.0.2 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 13 – FRAMEWORKS DE IA

A - ENUNCIADO

13.0.1 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()
```

13.0.2 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)

13.0.3 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).

13.0.4 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

APÊNDICE 15 – TÓPICOS EM IA

A - ENUNCIADO

15.0.1 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”.

15.0.2 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

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.