Universidade de Fortaleza

Curso: Ciencia da computação	Disciplina: M. A. Desempenho	Código/Turma: 36
Professor/a: Lyndainês		Data: 05-05-2025
Aluno/a: Pedro Souto Martins		Matrícula:2320443

1- Análise estatística

Existem 200 entradas no dataset, com uma variável dependente e 7 outras variáveis explicativas

1.1 Tratamento de dados faltantes

Existem dados faltantes nas colunas: latencia_ms, armazenamento_tb, tipo_hd e tipo_processador.

1.1.1 Tratamento para as variáveis categóricas

Para as variáveis categóricas os dados faltantes foram preenchidos de forma probabilística, se baseando na frequência de cada tipo possível para cada variável. Por exemplo, tipo_processador pode ser AMD, Intel ou Apple Silicon, usando a distribuição calculamos a porcentagem da chance do dado faltante ser cada um dos tipos de processador, e nos baseamos nisso para preencher os dados faltantes.

Após esse procedimento apenas sobram 20 entradas com algum dado faltante.

1.1.2 Tratamento para variáveis numéricas

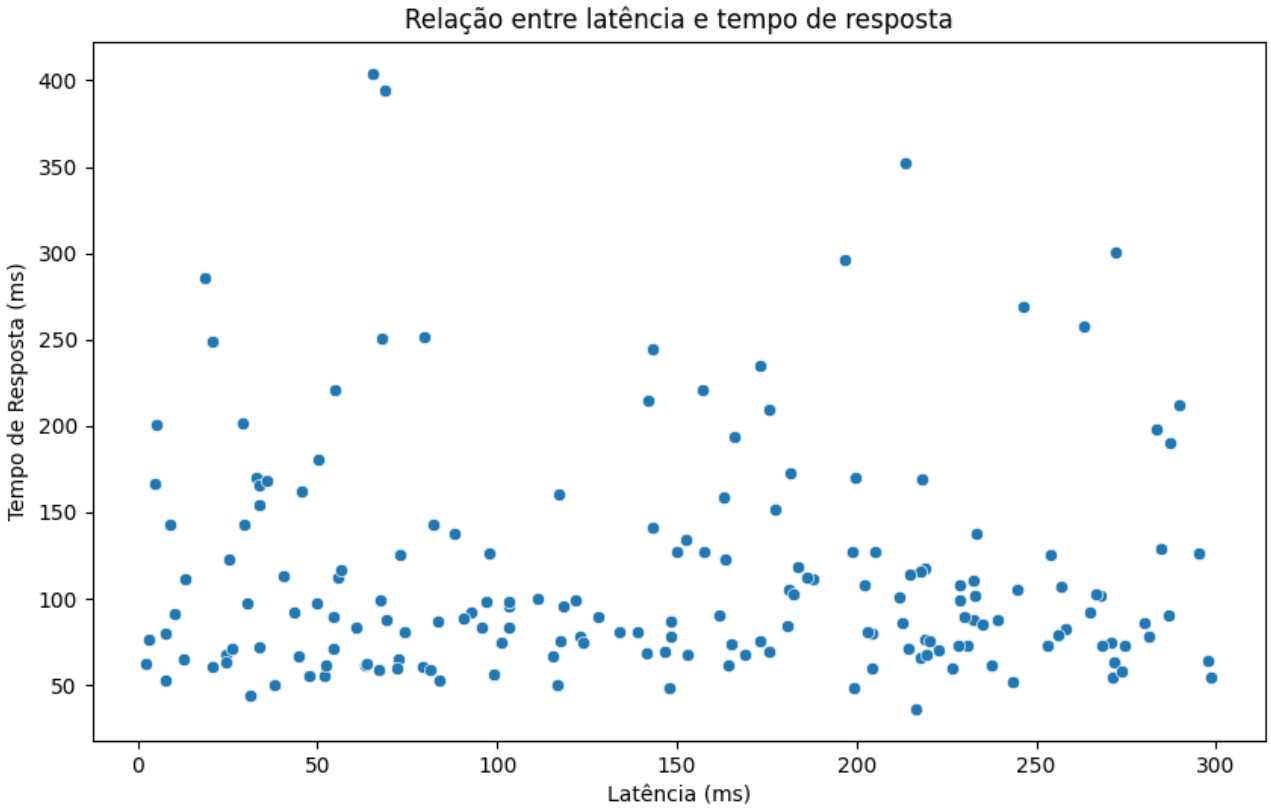
Após preencher todas variáveis categóricas faltantes, foram removidas todas entradas que está faltando o dado de latencia_ms. Essa escolha foi feita pois 20 entradas é apenas 10% do nosso dataset inteiro.

Uma alternativa seria preencher todos os dados faltantes de latencia_ms pela a mediana da variável, mas foi optado pela escolha mais simples de remover.

1.2 Observações iniciais sobre os dados

1.2.1 Relação entre latencia e tempo de resposta

A hipótese inicial é que tempo de resposta sempre deve ser maior que a latencia, já que em teoria o tempo de resposta já contem a latencia, porém isso não é verdade para os dados.



A existência de muitos pontos onde latência é maior que o tempo de resposta levanta duas hipóteses: 1. a coluna latência não tem relação com a coluna tempo de resposta; 2. todos pontos onde latência é maior que tempo de resposta devem ser ignorados por terem medidas erradas.

A hipótese 2. vai ser ignorada já que caso removermos todos os pontos que latência é maior que tempo de resposta teremos uma perda significativa no tamanho do dataset.

A hipótese 1. vai ser analisada na etapa 3 com um modelo de regressão que será feito excluindo variáveis.

1.2.2 Relação entre sistemas operacionais Mac e processadores Apple Silicon

Foi confirmado que toda entrada do dataset onde o sistema operacional é Mac o tipo de processador é Apple Sillicon.

2. Modelo e diagnóstico

Vamos criar um modelo de regressão linear com a variável dependente sendo o tempo de resposta.

2.1 Ajustes das variáveis categóricas

Temos 3 variáveis categóricas no dataset: tipo de processador, sistema operacional e tipo de hd.

Tipo de processador tem 3 opções: AMD, Apple Sillicon e Intel

Sistema operacional tem 3 opções: Mac, Linux, Windows

Tipo de hd tem 2 opções: HDD e SSD

Para trabalhar com uma regressão linear vamos transformar essas categorias em números. É possível converter todas essas 3 categorias em outras 4 categorias binárias: is_amd, is_windows, is_mac, tipo_hd.

tipo_hd é 0 quando é HDD e 1 quando é SSD.

Cada uma dessas outras 3 novas variáveis vão ser ou 0 ou 1, e apenas com essas 3 variáveis é possível destinguir todas opções possíveis. Por exemplo, se processador é intel, sistema operacional é linux e tipo de hd é SSD: is_amd = 0, is_windows = 0, is_mac = 0.

2.2 Resultado da regressão linear inicial

Foi usado o modelo OLS com a biblioteca statsmodel com python para fazer a regressão linear.

Valor $R^2 = 0.722$
Valor R^2 ajustado = 0.708

Coeficientes:

intercepto	249.2691
cpu_cores	-12.9233
ram_gb	-1.4148
latencia_ms	-0.0305
armazenamento_tb	0.8122
tipo_hd	0.2089
is_amd	-3.4681
is_windows	10.2311
is_mac	2.6440

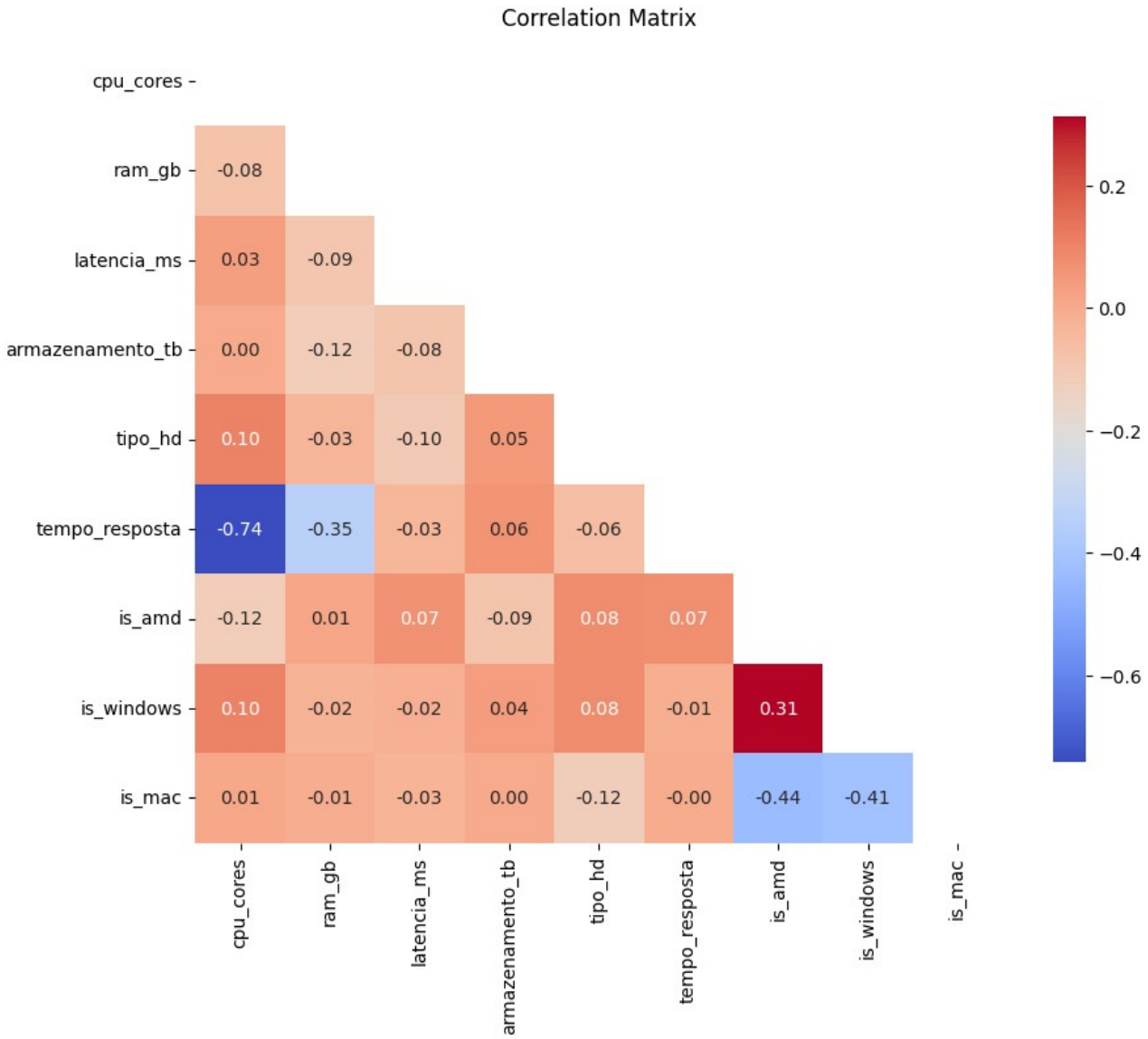
2.3 Diagnóstico de multicolinearidade

Foi usado o método variance_inflation_factor da biblioteca statsmodel para o teste de multicolinearidade. Esse método um valor VIF para cada variável e caso esse valor seja maior que 5 é provável que essa variável seja explicável por outras variáveis do modelo.

Resultado:

1 cpu_cores 1.053444
2 ram_gb 1.035126
3 latencia_ms 1.034791
4 armazenamento_tb 1.025794
5 tipo_hd 1.034463
6 is_amd 1.307646
7 is_windows 1.273223
8 is_mac 1.385501

Também podemos visualizar os dados com uma matriz de correlação e ver se não existem valores absurdos.



Pelos resultados não existe multicolinearidade no modelo.

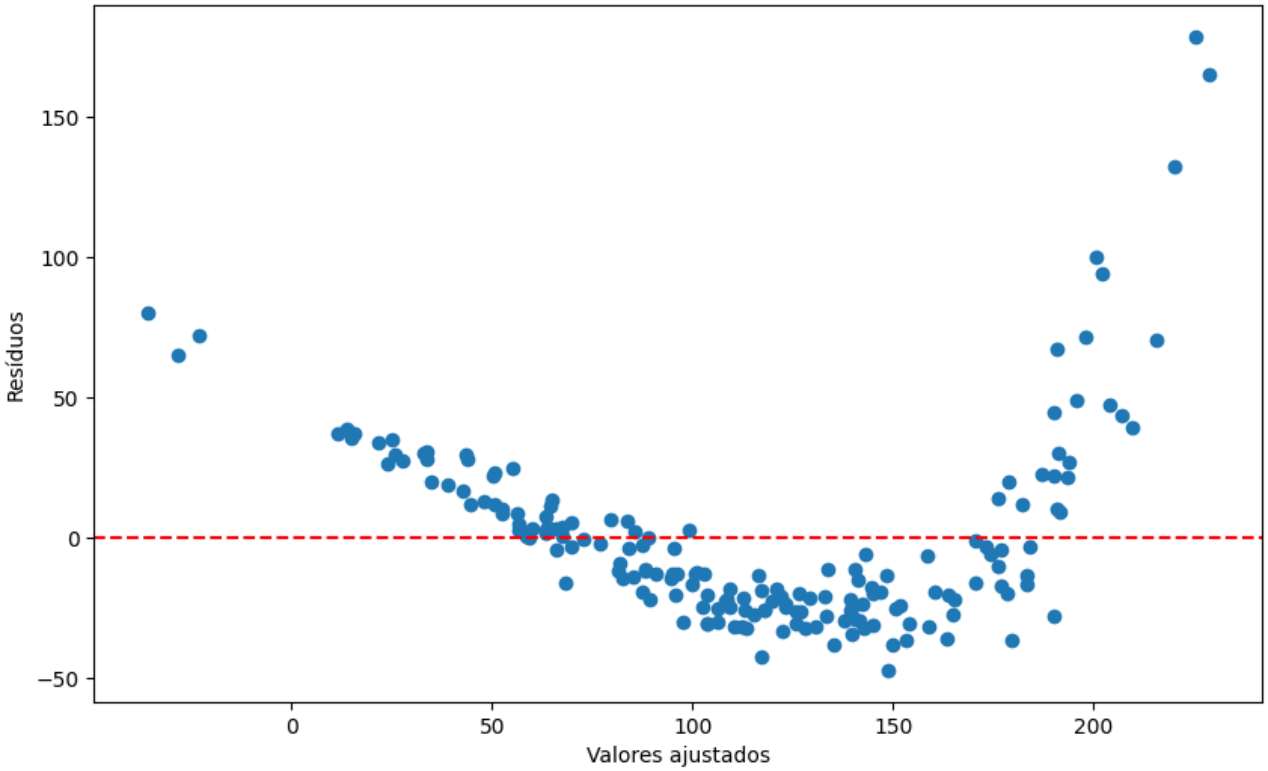
2.4 Diagnóstico de heterocedasticidade

2.4.1 Teste Breusch-Pagan

Foi usado o método het_breuschpagan da biblioteca statsmodels

Resultado:
p value: 0.00212

Visualização do gráfico de resíduos por valores estimados



Conclusão:
Pelo p-value ser menor que 0.05 e pelo gráfico não ter pontos uniformemente distribuídos, rejeita-se a hipótese de homocedasticidade.

2.5 Teste F e teste t

Resultado do p-value do teste F:

8.125846370072553e-49

Conclusão: Rejeita-se a hipótese nula, implicando que as variáveis têm significancia estatística para o modelo.

Resultado do p-value para cada variável:

cpu_cores 0.000
ram_gb 0.000
latencia_ms 0.317
armazenamento_tb 0.842
tipo_hd 0.952
is_amd 0.569
is_windows 0.099
is_mac 0.696

Conclusão: Rejeita-se a hipótese nula para as variáveis cpu_cores e ram_gb, implicando que essas duas variáveis tem significância para o modelo.

3. Comparação com modelos alternativos

Foram criados 7 modelos, o primeiro com apenas 1 variável explicativa, o segundo com 2 variáveis explicativas, e assim em diante. Foi usado o método SequentialFeatureSelector da bibliotéca sklearn para fazer a escolha das variáveis de cada modelo, usando como critério o R^2.

Esses foram os resultados de cada modelo:

Número de variáveis	R^2	R^2 ajustado	F statistic	Variáveis
1	0.55	0.547	217.564	cpu_cores
2	0.714	0.711	220.768	cpu_cores, ram_gb
3	0.714	0.709	146.815	cpu_cores, ram_gb, tipo_hd
4	0.714	0.708	109.972	cpu_cores, ram_gb, tipo_hd, is_mac
5	0.715	0.707	87.476	cpu_cores, ram_gb, armazenamento_tb, tipo_hd, is_mac
6	0.719	0.709	74.019	cpu_cores, ram_gb, armazenamento_tb, tipo_hd, is_windows, is_mac
7	0.721	0.709	63.578	cpu_cores, ram_gb, latencia_ms, armazenamento_tb, tipo_hd, is_windows, is_mac

O modelo com o melhor R^2 ajustado e F statistic foi o modelo com duas variáveis cpu_cores e ram_gb. Usando a tabela acima e os p-values de cada variável, o modelo com duas variáveis cpu_cores e ram_gb seria o modelo recomendado.