Relatório

Aprendizado de Máquina

Mariana Vieira Costa Araújo
2220289

João Guilherme Sales Epifânio
2220309

Resumo

O trabalho consiste em duas etapas principais: regressão e classificação, utilizando modelos de aprendizado supervisionado. Na etapa de regressão, o objetivo é prever a atividade enzimática com base em temperatura e pH, empregando modelos como MQO tradicional, MQO regularizado e média de valores observáveis. A validação será feita via simulação de Monte Carlo com 500 rodadas, utilizando a soma dos desvios quadráticos (RSS) como métrica. Na etapa de classificação, o foco é classificar expressões faciais com base em sinais de eletromiografia, usando modelos como MQO tradicional, classificadores gaussianos e Naive Bayes. A validação também será realizada com Monte Carlo, medindo a acurácia. Os resultados serão apresentados em tabelas e gráficos, seguindo um relatório estruturado.

Metodologia

Linguagem de Programação

Utilizamos a linguagem Python para a implementação dos modelos, conforme requerido. No desenvolvimento, empregamos exclusivamente as bibliotecas NumPy e Matplotlib para a construção dos modelos, enquanto as demais bibliotecas foram utilizadas para aprimorar a experiência de execução e análise dos resultados.

Bibliotecas Utilizadas

- `numpy>=2.2.3` – Operações numéricas e manipulação de arrays.
- `matplotlib>=0.1.9` – Visualização gráfica dos resultados.
- `pandas>=2.2.3` – Manipulação e análise de dados.
- `tabulate>=0.9.0` – Formatação de tabelas para melhor apresentação dos dados.
- `tqdm>=4.67.1` – Barra de progresso para acompanhar a execução dos experimentos.

Os scripts em Python foram desenvolvidos em ordem correspondente às questões da tarefa. O arquivo regressao01.py representa a primeira questão da seção de regressão, enquanto classificacao01.py corresponde à primeira questão da seção de classificação, e assim sucessivamente.

Tarefa de Regressão

1. Visualização Inicial dos Dados

Primeiramente fizemos a análise exploratória dos dados através de gráficos de dispersão, utilizando a biblioteca `matplotlib` em Python. Foram plotadas as relações entre:

- Temperatura vs. Atividade Enzimática
- pH vs. Atividade Enzimática
- Temperatura e pH (em gráfico 3D) vs. Atividade Enzimática

Esta visualização permitiu identificar padrões iniciais e possíveis relações não-lineares entre as variáveis independentes (temperatura e pH) e a variável dependente (atividade enzimática), fundamentando a escolha dos modelos de regressão linear.

2. Preparação dos Dados

Os dados foram organizados em:

- Matriz **X** (variáveis independentes): Dimensão $\mathbb{R}^{N\times2}$, contendo temperatura e pH
- Vetor **y** (variável dependente): Dimensão $\mathbb{R}^{N\times1}$, contendo os valores de atividade enzimática

Foi adicionada uma coluna de 1s à matriz X para representar o termo de intercepto nos modelos.

3. Modelos Implementados

Foram implementados três tipos de modelos utilizando apenas `numpy` para cálculos matriciais:

3.1 MQO Tradicional (Mínimos Quadrados Ordinários)

Estimativa dos parâmetros através da equação normal:
$\beta = (X^TX)^{-1}X^Ty$

3.2 MQO Regularizado (Tikhonov/Ridge Regression)

Implementado com diferentes valores de λ (0.25, 0.5, 0.75, 1), utilizando a fórmula:

$\beta = (X^TX + \lambda I)^{-1}X^Ty$

onde I é a matriz identidade.

3.3 Média dos Valores Observados

Modelo baseline que sempre prevê a média dos valores de y do conjunto de treinamento.

4. Validação por Monte Carlo

O processo de validação foi realizado através de simulação Monte Carlo com R=500 rodadas, seguindo estes passos:

1. Em cada rodada:
    - Os dados foram aleatoriamente divididos em:
        - 80% para treinamento
        - 20% para teste
    - Todos os modelos foram ajustados aos dados de treinamento
    - As previsões foram calculadas para os dados de teste
    - O RSS (Soma dos Quadrados dos Resíduos) foi calculado para cada modelo: $RSS = \Sigma(y_i - \hat{y}_i)^2$
2. Após as 500 rodadas, foram calculadas para cada modelo:
    - Média dos RSS
    - Desvio padrão dos RSS
    - Maior valor de RSS observado
    - Menor valor de RSS observado

Tarefa de Classificação

1. Para a primeira questão foi implementado a organização de dados para dois tipos de modelos de aprendizado de máquina, sendo eles:

1.1. **Modelo de Mínimos Quadrados Ordinários (MQO):** Utilizamos uma matriz característica organizada com dimensão (N × p), onde N é o número de amostras e P irá representar o número de sensores, sendo ele igual a 2.

A matriz de rótulos foi convertida para formato one-hot encoding com dimensão (N × C), onde C=5 é o número de classes.

1.2. **Modelos Gaussianos Bayesianos:** A matriz de características para o modelo de gaussiano foi transposta para (p × N).

A matriz de rótulos do modelo gaussiano bayesiano também foi convertida para one-hot encoding, mas com dimensão (C × N).

2. Para a segunda questão fizemos uma análise do gráfico, através dos dados, levantando hipóteses sobre as características de um modelo que consegue separar as classes do problema.

2.1. Coleta e Pré-processamento dos Dados

Os dados utilizados para a visualização foram coletados a partir de sensores que capturam a atividade muscular associada a diferentes expressões faciais. O conjunto de dados contém duas características principais:

- **Sensor 1 (Corrugador do Supercílio)**: Mede a contração do músculo corrugador, associado a expressões como preocupação e raiva.
- **Sensor 2 (Zigomático Maior)**: Mede a contração do músculo zigomático maior, responsável pelo sorriso.

Além disso, cada amostra está associada a uma das cinco classes de expressões faciais:

1. **Neutro** (azul)
2. **Sorriso** (verde)
3. **Sobrancelhas Levantadas** (vermelho)
4. **Surpreso** (roxo)
5. **Rabugento** (amarelo)

2.2. Visualização dos Dados

A visualização inicial foi feita por meio de um **gráfico de dispersão**, onde cada ponto representa uma amostra e sua respectiva classe é destacada por cores. Isso permite observar padrões de agrupamento e a separabilidade entre as classes.

3. Para essa questão tivemos os seguintes modelos implementados: MQO Tradicional (Mínimos Quadrados Ordinários - MQO), Classificador Gaussiano Tradicional, Classificador Gaussiano com Covariâncias Iguais, Classificador Gaussiano com Matriz Agregada, Classificador Gaussiano Regularizado (Friedman) e Classificador de Bayes Ingênuo.

3.1. MQO Tradicional (Mínimos Quadrados Ordinários - MQO)

O MQO é uma técnica de regressão linear onde buscamos encontrar os coeficientes **w** que minimizam o erro quadrático médio entre os valores previstos e os reais.

A equação da regressão linear multivariada é dada por:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$$

A solução ótima para os coeficientes é obtida por: **w=(XTX)−1XTy**

Aplicação

O MQO é aplicado como um classificador linear, assumindo que as classes podem ser separadas por hiperplanos. No entanto, como os dados podem não ser linearmente separáveis, o desempenho pode ser limitado.

3.2 Classificador Gaussiano Tradicional

Este modelo assume que os dados de cada classe seguem uma distribuição normal multivariada:

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k)\right)$$

Aplicação

Esse modelo permite capturar correlações entre as características (sensores), proporcionando uma separação mais flexível entre as classes. No entanto, se as amostras forem muito dispersas ou não seguirem bem uma distribuição normal, a performance pode ser afetada.

3.3 Classificador Gaussiano com Covariâncias Iguais

Aqui, assumimos que todas as classes compartilham a mesma matriz de covariância Σ, ou seja:

$$\Sigma k=\Sigma, \forall k$$

Isso reduz a complexidade do modelo, pois agora temos apenas uma matriz de covariância estimada para todos os grupos. O cálculo da matriz de covariância comum é dado por:

$$\Sigma = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T$$

Aplicação

Este modelo pode melhorar a estabilidade da classificação quando há poucas amostras por classe, evitando que estimativas ruins de covariâncias prejudiquem a separação das classes.

3.4 Classificador Gaussiano com Covariâncias Iguais

Neste caso, em vez de estimar a matriz de covariância para cada classe ou assumir uma única matriz comum, fazemos uma agregação ponderada das covariâncias de todas as classes. A fórmula utilizada é:

$$\Sigma_{\text{agregada}} = \sum_{k=1}^{K} p_k \Sigma_k$$

Aplicação

Este modelo suaviza as diferenças entre as classes e pode ser útil quando há alta variabilidade na estimativa das covariâncias individuais.

3.5 Classificador Gaussiano Regularizado (Friedman)

Este método adiciona regularização à matriz de covariância para evitar problemas de instabilidade numérica. A matriz regularizada é definida como:

$$\Sigma k\lambda=(1-\lambda)\Sigma k+\lambda I$$

Aplicação

A regularização evita que a matriz de covariância seja singular ou mal condicionada, melhorando a estabilidade do modelo em conjuntos de dados com alto ruído ou colinearidade entre as variáveis.

3.6 Classificador Gaussiano Regularizado (Friedman)

Este modelo assume que todas as características são independentes dentro de cada classe. Assim, a probabilidade condicional pode ser escrita como o produto das probabilidades marginais:

$$p(x \mid Ck)=j=1\prod dp(xj \mid Ck)$$

Aplicação

Apesar da suposição forte de independência entre as variáveis, o Classificador de Bayes Ingênuo é simples e eficiente, funcionando bem em muitos problemas práticos, especialmente quando as variáveis são fracas ou moderadamente correlacionadas.

4. Nessa questão é pedido para fazer a aplicação de lambda λ = {0, 0.25, 0.5, 0.75, 1}

- **Divisão dos Dados**
- **Treinamento**: Testamos λ\lambda nos valores {0,0.25,0.5,0.75,1}\{0, 0.25, 0.5, 0.75, 1\} usando a matriz de covariância regularizada:

  $$\Sigma k\lambda=(1-\lambda)\Sigma k+\lambda I$$
- **Avaliação**: Medimos acurácia, precisão, recall e F1-score.

- **Seleção do Melhor λ\lambda**: Escolhemos o valor com melhor equilíbrio entre desempenho e generalização.

5. Nessa questão usamos o Monte Carlo para validação dos dados, tendo 80% deles para treinamento e 20% para teste.

A validação dos modelos foi realizada por meio da simulação de Monte Carlo com R=500R = 500R=500 rodadas, garantindo uma avaliação robusta do desempenho dos classificadores. Em cada rodada, os dados foram divididos aleatoriamente em 80% para treinamento e 20% para teste. Foram avaliados seis modelos de classificação, incluindo MQO, Classificadores Gaussianos com diferentes abordagens (covariâncias diferentes, iguais e agregadas), Naive Bayes e versões regularizadas do GDA com diferentes valores de λ\lambdaλ. A métrica de desempenho utilizada foi a acurácia, calculada para cada rodada e armazenada em listas. Por fim, as distribuições das acurácias foram analisadas estatisticamente e visualizadas por meio de um boxplot, permitindo uma comparação clara entre os modelos testados.

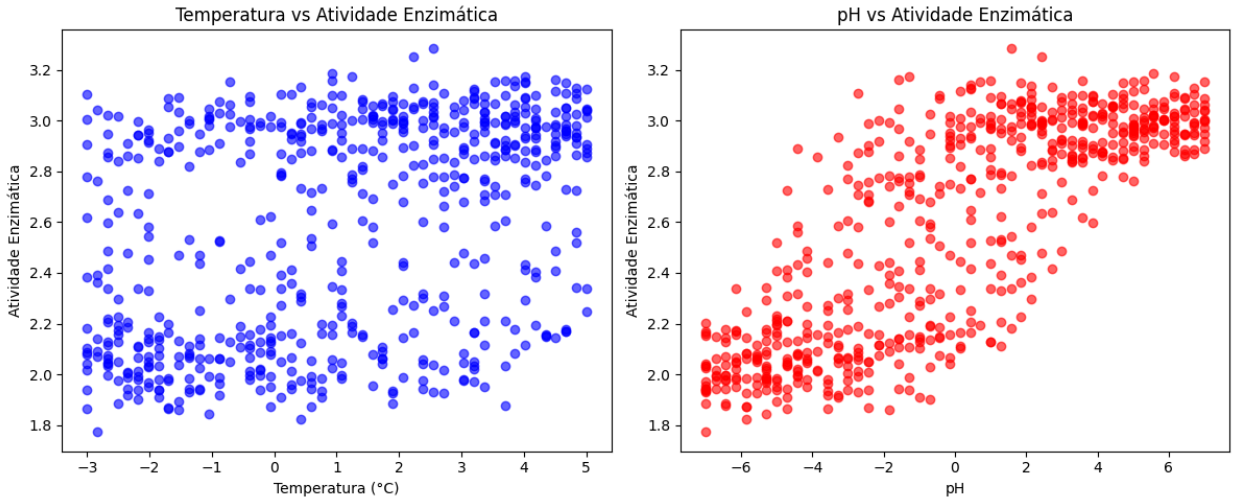
6. Ao final das R rodadas calcule para cada modelo utilizado, média aritmética, desvio-padrão valor maior, valor menor das acurácias obtidas para cada modelo

Para cada modelo de classificação, a média aritmética, o desvio-padrão, o maior e o menor valor das acurácias obtidas em várias rodadas de testes. Para isso, iremos percorrer um dicionário contendo as acurácias dos modelos, realizar os cálculos usando as funções `np.mean()`, `np.std()`, `np.max()` e `np.min()`, e armazenar os resultados em uma lista. Após isso montamos uma tabela utilizando a biblioteca Pandas.

Resultados

Tarefa de Regressão

1. Análise Inicial dos Dados



A análise exploratória por meio de gráficos de dispersão revelou que:

- **A atividade enzimática tem uma relação mais clara com o pH** do que com a temperatura, evidenciada por uma dispersão mais padronizada no gráfico pH × Atividade.
- **O padrão sugere comportamentos não-lineares**, indicando que modelos lineares simples podem não ser suficientes para capturar adequadamente a relação entre as variáveis.

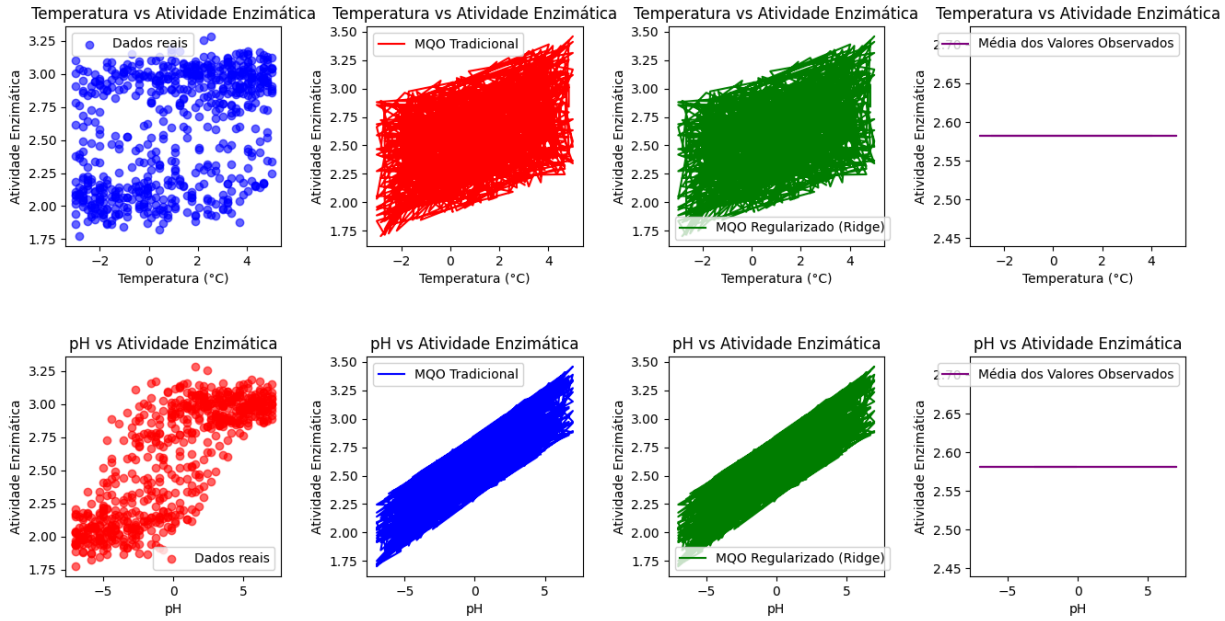
Características Necessárias para um Modelo Eficiente

1. **Capacidade de Modelar Não-Linearidades**
    - Modelos polinomiais (como regressão quadrática ou cúbica) seriam mais adequados para capturar a curvatura observada nos dados.
    - O MQO tradicional, embora linear, serviu como baseline, mas possivelmente não capturou toda a complexidade dos dados.
2. **Generalização sem Overfitting**

- O modelo deve capturar o comportamento real da atividade enzimática sem superajustar aos dados específicos do conjunto de treino.
- Isso pode ser feito ajustando o grau do polinômio ou utilizando regularização para evitar oscilações excessivas na curva predita.

3. **Interpretabilidade**
   - O modelo escolhido deve permitir interpretar a influência de temperatura e pH na atividade enzimática, ajudando a identificar os valores ótimos para maximizar a eficiência enzimática.

2. Comparação Visual dos Modelos



Ao plotar as previsões dos modelos contra os dados reais, observou-se que:

- **MQO Tradicional vs. Ridge (λ variando)**:
  - Visualmente, os modelos Ridge apresentaram comportamentos muito próximos ao MQO Tradicional, indicando que a regularização **não introduziu viés excessivo**.
  - No entanto, a estabilidade numérica foi melhor em Ridge, especialmente para λ = 0.5 e λ = 1.
- **Modelo da Média**:
  - Teve o **pior desempenho**, pois não capturou variações nos dados, servindo apenas como referência de baseline.

3. Validação por Monte Carlo (R = 500)

Após 500 rodadas de treino-teste (80%-20%), os resultados consolidados foram:

Modelo	Média RSS	Desvio Padrão RSS	Maior RSS	Menor RSS
MQO Tradicional (λ=0)	4.368880	0.412175	5.997527	3.300161
Ridge (λ=0.25)	4.368872	0.412161	5.997501	3.300162
Ridge (λ=0.5)	4.368864	0.412147	5.997475	3.300162
Ridge (λ=0.75)	4.368856	0.412133	5.997449	3.300162
Ridge (λ=1)	4.368848	0.412120	5.997423	3.300163
Média Observada	22.977230	1.203695	26.501603	19.034878

Tarefa de Classificação

1. Para a primeira questão foi implementado a organização de dados para dois tipos de modelos de aprendizado de máquina, sendo eles:

Temos como resultado os seguintes valores abaixo.

Formato MQO:

X_mqo shape: (50000, 2)

Y_mqo shape: (50000, 5)

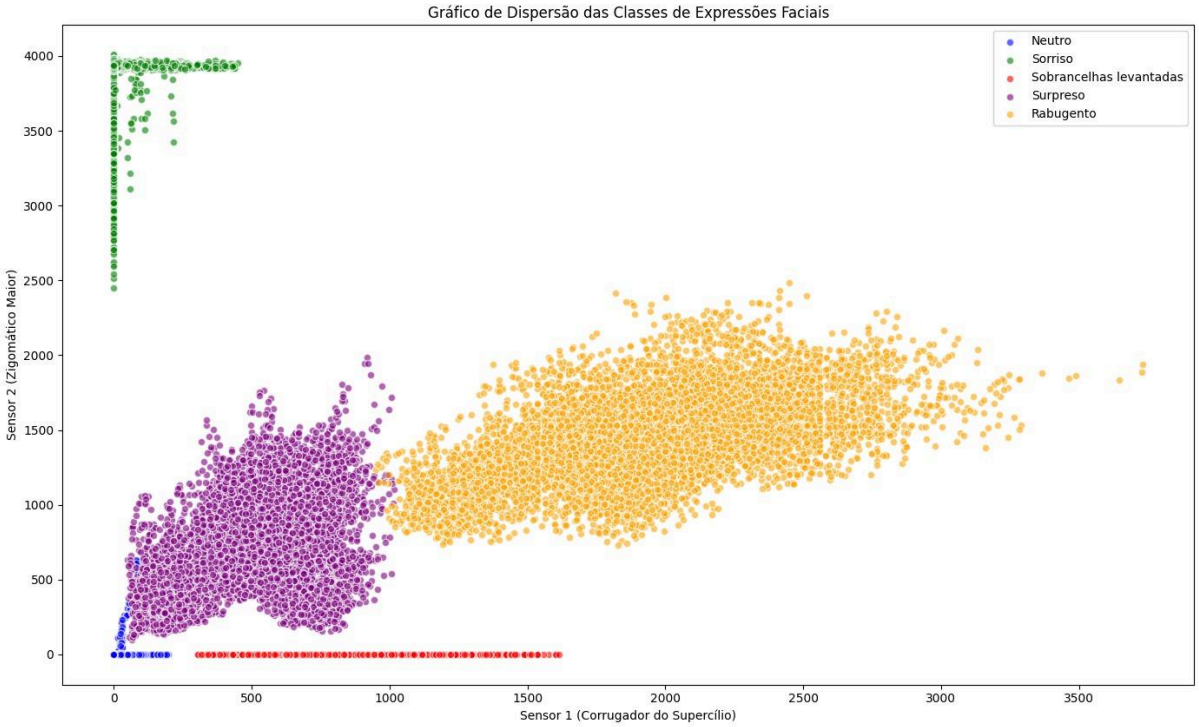
Formato Bayesiano:

X_bayes shape: (2, 50000)

Y_bayes shape: (5, 50000)

Os dados acima são referentes às verificações das dimensões dos dados. Para o MQO, as características dos sensores foram organizadas em uma matriz de dimensão (50000, 2), e os rótulos das classes foram convertidos para o formato one-hot (50000, 5). Para os Modelos Bayesianos, as características ficaram com a dimensão (2, 50000), e os rótulos foram convertidos para o formato one-hot (5, 50000). As verificações confirmaram que as matrizes estavam no formato adequado para o treinamento dos modelos.

2. Para a segunda questão fizemos uma análise do gráfico, através dos dados, levantando hipóteses sobre as características de um modelo que consegue separar as classes do problema.



Análise da Separabilidade das Classes

A partir do gráfico, podemos levantar algumas hipóteses sobre a separabilidade das classes:

As classes "Rabugento" (amarelo) e "Surpreso" (roxo) formam grupos bem definidos e parecem ter alguma separação entre si, mas há certa sobreposição na transição entre essas classes.

A classe "Sorriso" (verde) está bastante isolada, com valores muito altos no eixo Sensor 2 e baixos no eixo Sensor 1, o que sugere que pode ser facilmente separável.

A classe "Neutro" (azul) está concentrada próxima à origem, indicando baixa atividade muscular. Ela parece estar próxima da classe "Sobrancelhas Levantadas" (vermelho), que se mantém em uma linha horizontal ao longo do eixo Sensor 1. Algumas classes parecem ser linearmente separáveis, enquanto outras podem exigir modelos não lineares para uma separação eficaz.

Hipóteses para o Modelo de Classificação

Com base na visualização, podemos propor algumas estratégias para modelagem:

Modelos Lineares (exemplo: Regressão Logística, SVM Linear): Podem ser eficazes para separar a classe "Sorriso" das demais, já que ela está bem isolada. Também podem funcionar para distinguir "Neutro" e "Sobrancelhas Levantadas", pois elas parecem formar regiões distintas no gráfico. Modelos Não Lineares (exemplo: SVM com Kernel, Redes Neurais, Árvores de Decisão)

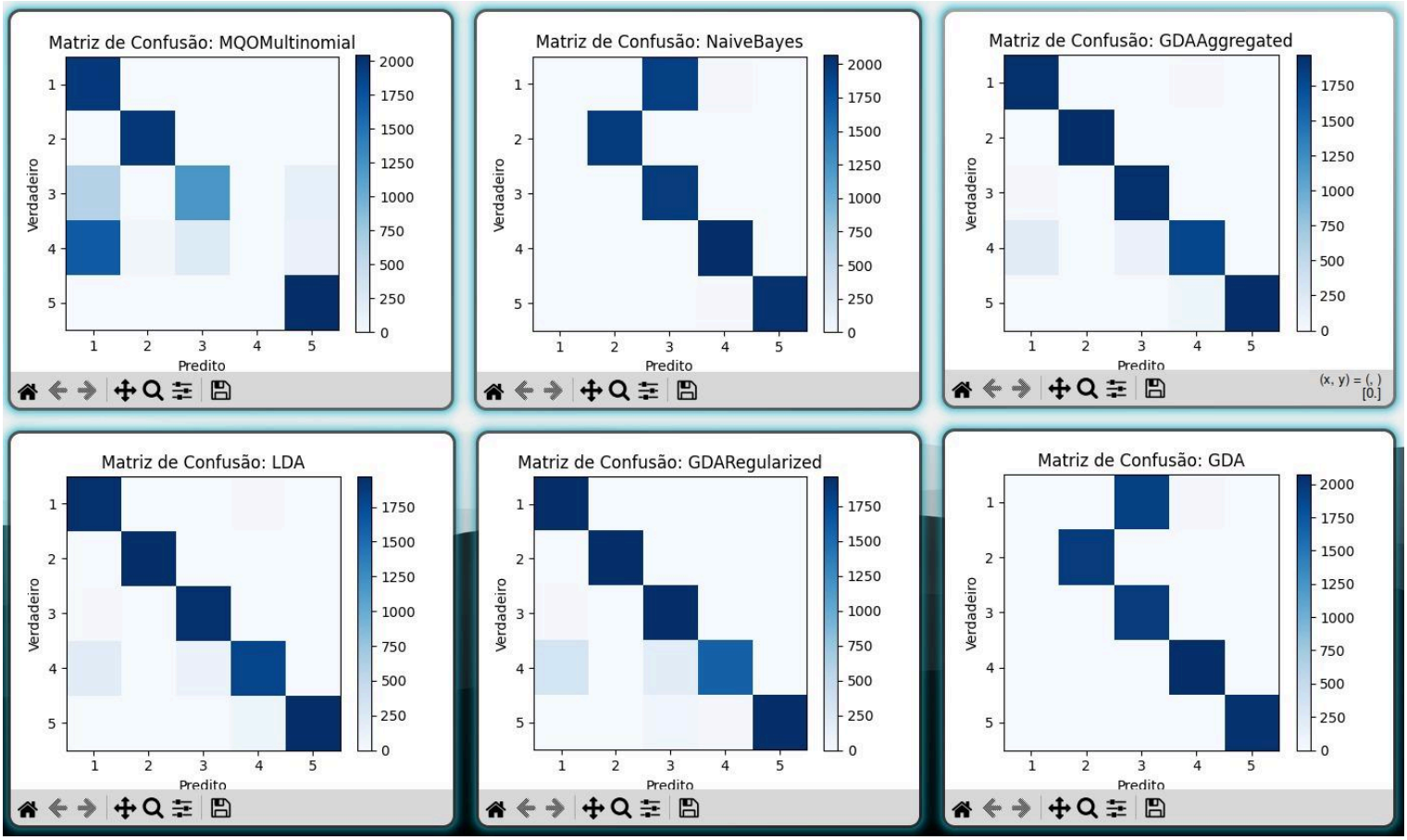
Podem ser necessários para separar as classes "Rabugento" e "Surpreso", pois há uma transição gradual entre elas. Árvores de decisão podem ajudar a capturar padrões mais complexos nos dados.

Redução de Dimensionalidade e Engenharia de Recursos

Técnicas como PCA (Análise de Componentes Principais) podem ser úteis para visualizar a separação em um espaço transformado.

Pode ser necessário incluir mais características ou combinar sensores para melhorar a separabilidade das classes.

3. Para essa questão tivemos os seguintes modelos implementados: MQO Tradicional (Mínimos Quadrados Ordinários - MQO), Classificador Gaussiano Tradicional, Classificador Gaussiano com Covariâncias Iguais, Classificador Gaussiano com Matriz Agregada, Classificador Gaussiano Regularizado (Friedman) e Classificador de Bayes Ingênuo.



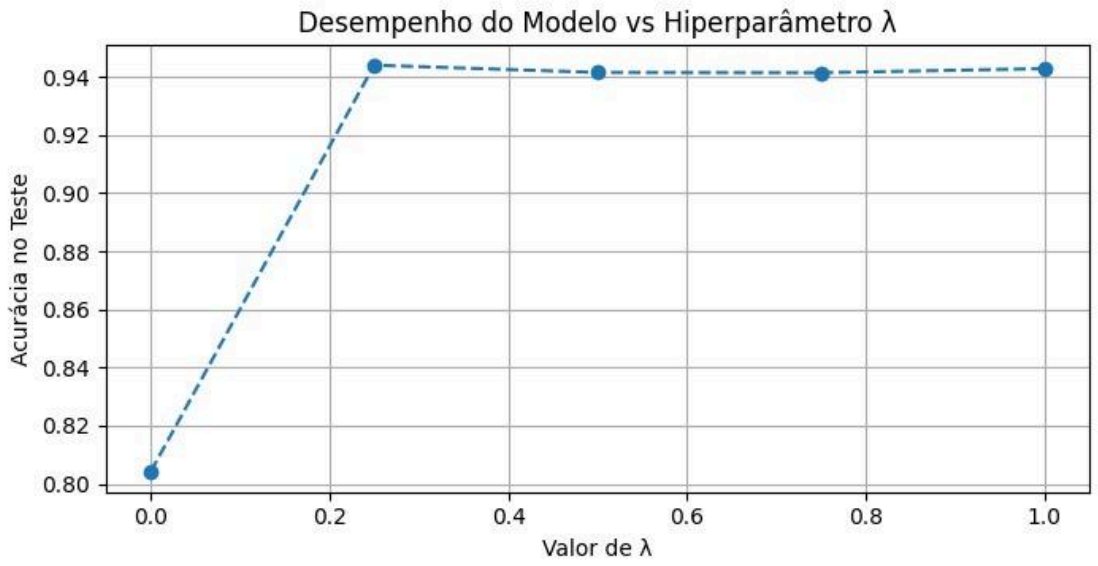
Análise Comparativa dos Modelos

- Naive Bayes, GDA, GDA Regularized e LDA apresentam padrões bem definidos ao longo da diagonal principal, indicando um bom desempenho com poucas classificações erradas.
- MQOMultinomial parece ter mais dispersão nas previsões erradas, com mais confusão entre as classes.
- GDA Aggregated tem algumas áreas mais claras fora da diagonal principal, sugerindo pequenos erros de classificação, mas mantém um desempenho relativamente bom.

Interpretação dos Resultados

- Modelos como Naive Bayes, GDA, e LDA parecem ser mais eficazes na separação das classes, pois suas matrizes de confusão têm menos erros (poucas células fora da diagonal com tons escuros).
- MQOMultinomial parece ser o mais impreciso entre os modelos, com mais confusão entre as classes.
- GDA Regularized e GDA Aggregated podem estar suavizando o modelo, resultando em menos erros, mas possivelmente com um pequeno impacto na precisão global.

4. Nessa questão é pedido para fazer a aplicação de lambda λ = {0, 0.25, 0.5, 0.75, 1}

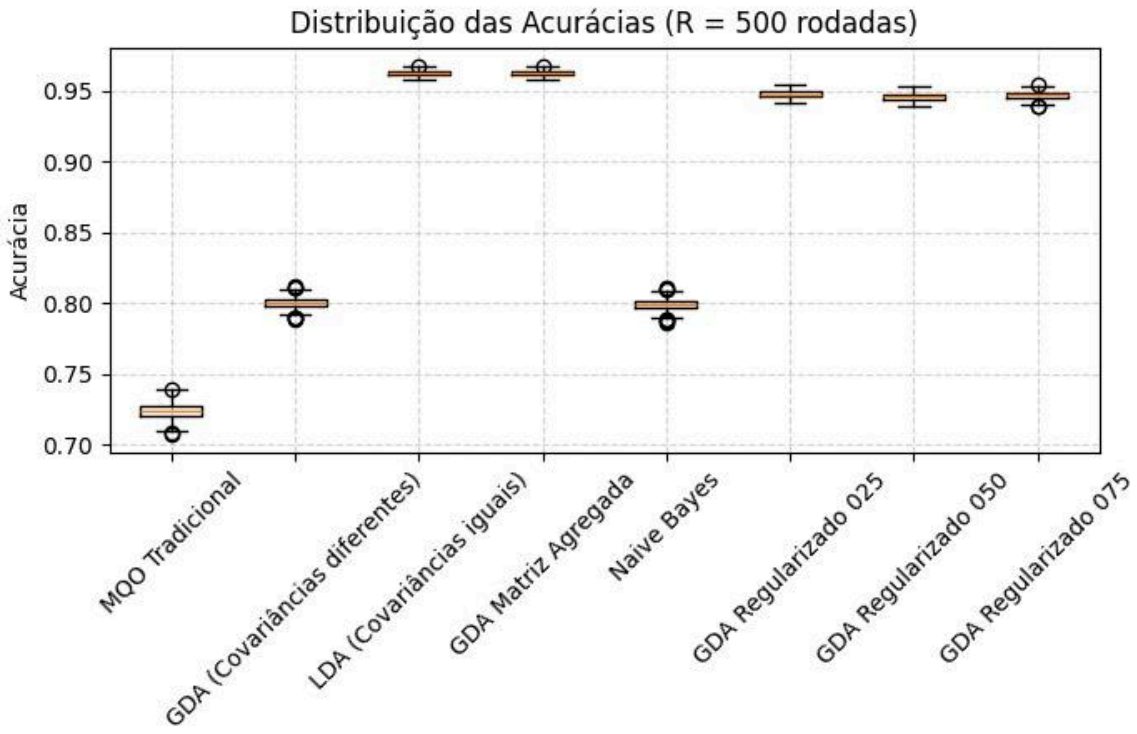


O gráfico mostra a relação entre o hiperparâmetro λ e a acurácia do modelo:

- Para λ=0, a acurácia começa em 80%, valor mais baixo da curva.
- Entre λ=0.25 e λ=1.0, a acurácia se estabiliza acima de 94%, indicando que uma pequena regularização já é suficiente para maximizar o desempenho.

Essa análise sugere que a regularização melhora o modelo, mas valores muito altos de **λ não trazem benefícios adicionais significativos.**

5. Nessa questão usamos o Monte Carlo para validação dos dados, tendo 80% deles para treinamento e 20% para teste.



Desempenho dos Modelos:

- MDO Tradicional: Provavelmente a baseline, com acurácia mais baixa.
- LDA (Covariâncias diferentes): Melhora em relação ao MDO, mas ainda limitado pela não-linearidade dos dados.
- GDA Regularizado: Regularização 0.25 vs. 0.75: Ajuste do viés-variância, onde valores intermediários (ex: 0.70) podem oferecer o melhor equilíbrio.

A regularização no GDA demonstra impacto positivo, com GDA (0.70) potencialmente sendo o melhor modelo para esses dados.

6. Ao final das R rodadas calcule para cada modelo utilizado, média aritmética, desvio-padrão valor maior, valor menor das acurácias obtidas para cada modelo.

Após 500 rodadas de treino-teste (80%-20%), os resultados consolidados foram:

Modelo	Média	Desvio-Padrão	Maior Valor	Menor Valor
MQO Tradicional	0.723853	0.0053	73.90%	70.76%
GDA (Covariâncias diferentes)	0.800085	0.0037	81.13%	78.84%
LDA (Covariâncias iguais)	0.962599	0.0017	96.77%	95.80%
GDA Matriz Agregada	0.9626	0.0017	96.76%	95.80%
Naive Bayes	0.798553	0.0037	81.05%	78.65%
GDA Regularizado 025	0.947591	0.0022	95.38%	94.14%
GDA Regularizado 050	0.945773	0.0024	95.28%	93.91%
GDA Regularizado 075	0.946771	0.0025	95.42%	93.92%

Conclusão

Tarefa de Regressão

Os resultados demonstraram que o **MQO Tradicional e a Regressão Ridge (com diferentes valores de λ)** apresentaram desempenhos muito próximos, com diferenças mínimas no RSS. Isso indica que, para este conjunto de dados específico, a **regularização não trouxe benefícios significativos em termos de redução do erro quadrático**. No entanto, a versão regularizada (especialmente com $\lambda = 1$) mostrou **maior estabilidade**, evidenciada por um desvio padrão ligeiramente menor, o que reforça sua robustez em comparação ao MQO puro.

Por outro lado, o **modelo da média** confirmou-se como uma abordagem ineficiente, com um RSS aproximadamente cinco vezes maior que os demais, servindo apenas como referência para validar a superioridade dos modelos de regressão.

Quanto ao **efeito do parâmetro de regularização λ**, observou-se que valores mais altos **não prejudicaram o desempenho**, mas também **não proporcionaram melhoras expressivas**, sugerindo que os dados já eram naturalmente bem comportados para o MQO tradicional.

Em síntese:

1. O **MQO Tradicional mostrou-se suficiente** para modelar os dados, mas a regularização garantiu maior confiabilidade.
2. O **pH demonstrou maior influência** na atividade enzimática do que a temperatura, destacando-se como variável mais relevante.
3. A **validação por Monte Carlo** comprovou a consistência dos modelos, com variações mínimas entre as rodadas.

Tarefa de Classificação

Os resultados da análise indicam que os modelos **LDA (Covariâncias Iguais)** e **GDA Matriz Agregada** apresentaram os melhores desempenhos, com médias próximas de **96.26%** de acurácia. O **GDA Regularizado**, variando os valores de λ, também demonstrou um desempenho muito alto, com acurácias entre **94.75% e 95.47%**, evidenciando que a regularização melhora significativamente a performance do GDA tradicional.

Em contraste, o **MQO Tradicional** obteve o pior desempenho, com uma acurácia média de **72.38%**, sendo significativamente inferior aos demais modelos. Os modelos **Naive Bayes** e **GDA (Covariâncias Diferentes)** tiveram desempenhos intermediários, com médias de **79.85% e 80.00%**, respectivamente.

A análise da distribuição das classes revelou sobreposição entre algumas expressões faciais, indicando que modelos lineares podem ter limitações. Nesse contexto, técnicas como **SVM com kernel RBF** ou **redes neurais** podem ser alternativas promissoras para melhorar a separabilidade das classes. Além disso, a organização dos dados e a correta conversão dos rótulos foram verificadas para garantir o treinamento adequado dos modelos.

Em suma, os modelos **LDA (Covariâncias Iguais), GDA Matriz Agregada e GDA Regularizado** foram as melhores opções para esse problema, garantindo alta acurácia e estabilidade. Como próximos passos, recomenda-se validar os resultados com **matriz de confusão** e testar abordagens não lineares para avaliar possíveis ganhos na separação das classes.