TRABALHO PRÁTICO 2

IA

Lucas Cassio Costa

Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brasil

---

1 - Introdução

A segurança e o conforto dos passageiros durante voos transatlânticos são prioridades essenciais na aviação moderna. Recentemente, uma equipe de cientistas enfrentou o desafio de melhorar o desempenho do piloto automático ajustando os ailerons de forma mais precisa e eficiente, reduzindo oscilações perigosas. Para alcançar esse objetivo, optou-se pelo desenvolvimento de um modelo de aprendizado de máquina baseado em dados históricos.

Este trabalho documenta a análise e implementação desse modelo, utilizando o conjunto de dados ailerons.csv. A abordagem inclui desde uma análise exploratória detalhada até a implementação de modelos de regressão e classificação, além da aplicação de técnicas avançadas de aprendizado por reforço.

As perguntas apresentadas serão respondidas de maneira estruturada e detalhada ao longo deste documento. Cada seção será dedicada a explorar e analisar aspectos específicos do problema.

2 - Aprendizado Supervisionado - Regressão

I. Porque os atributos Alpha, Se e [SeTime1 : SeTime14] não contribuem em conjunto para um modelo?

Os atributos SeTime, Alpha e Se são altamente correlacionados entre si, o que pode causar redundância e dificuldade em identificar contribuições independentes para o modelo. A alta colinearidade pode fazer com que alguns atributos pareçam menos importantes devido à dificuldade em distinguir seus efeitos individuais.

II. Porque os atributos DiffSeTime[2i] para i de 1 a 7 tendem a não contribuir para o modelo?

Os atributos DiffSeTime[2i] tendem a não contribuir significativamente para o modelo por:

1. **Baixa Correlação com Goal**: A correlação muito baixa entre esses atributos e a variável alvo indica que eles não fornecem informações relevantes para a predição.
2. **Coeficientes Baixos em Relação aos Outros Atributos**: Embora alguns coeficientes sejam relativamente altos, a baixa correlação sugere que esses atributos não estão oferecendo uma contribuição substancial em comparação com outros atributos que têm maior correlação com Goal.

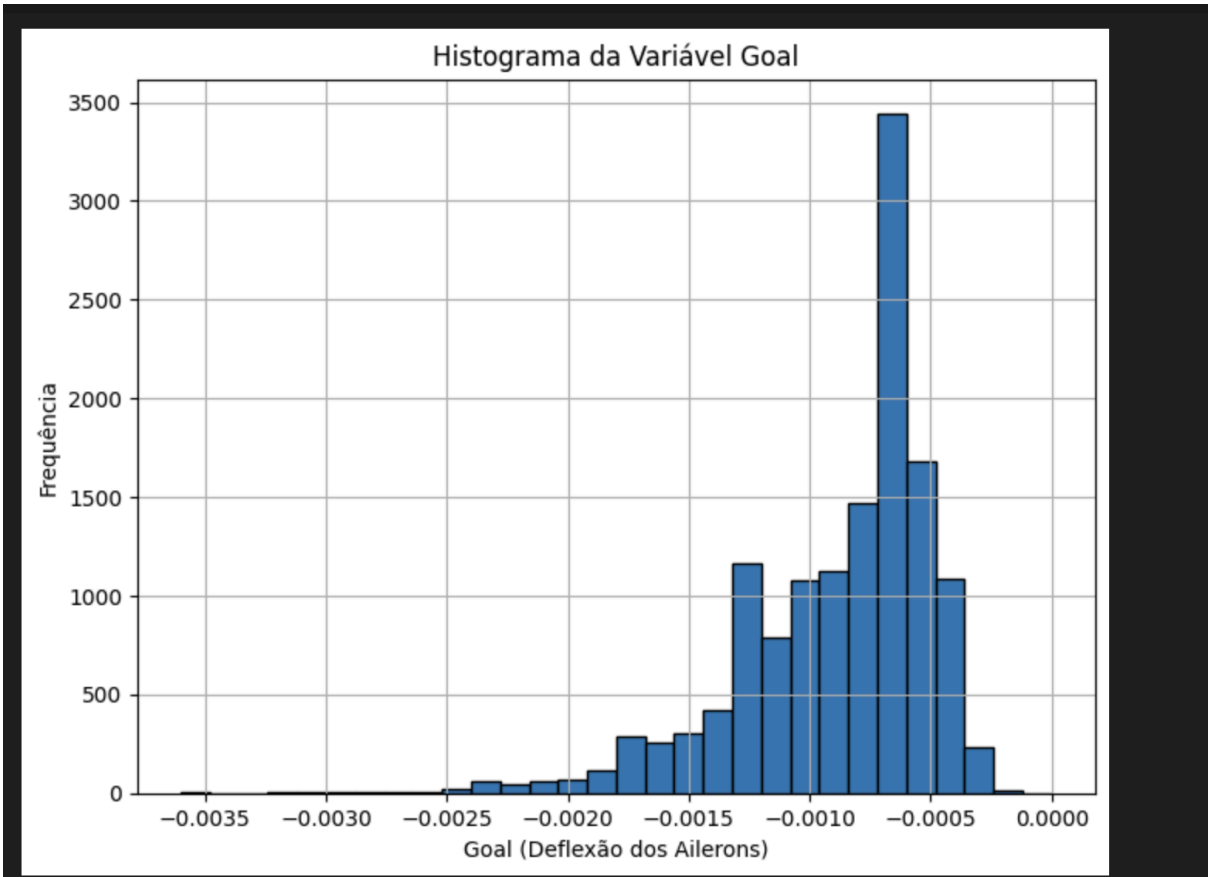
III. Quais os atributos mais promissores para um modelo de predição Linear de Goal?

Para identificar os atributos mais promissores para um modelo de predição linear de Goal, analisamos as correlações entre Goal e os diferentes atributos disponíveis:

- **AbsRoll (0.703023):** Apresenta uma correlação positiva forte com Goal. Isso indica que à medida que AbsRoll aumenta, o valor de Goal tende a aumentar, sugerindo uma relação direta e significativa entre esses dois atributos.
- **Se (-0.627451) e Alpha (-0.609222):** Têm correlações negativas fortes com Goal. Isso significa que conforme Se e Alpha aumentam, o valor de Goal tende a diminuir. Esses atributos mostram uma influência inversa importante na predição de Goal.
- **P (-0.317066), Q (-0.396222) e CurPitch (-0.313684):** Apresentam correlações negativas moderadas com Goal. Isso indica que esses atributos também têm impacto na direção de Goal, contribuindo para uma diminuição no seu valor conforme aumentam.

Conclusão: Para um modelo de predição linear de Goal, os atributos mais promissores são aqueles com correlações significativas com Goal. Portanto, AbsRoll é crucial devido à sua forte correlação positiva, enquanto Se e Alpha são igualmente importantes devido às suas correlações negativas substanciais. Variáveis como P, Q e CurPitch, embora apresentem correlações negativas moderadas, também são relevantes e devem ser consideradas na construção do modelo para capturar de forma mais precisa as variações em Goal.

IV. Faça um histograma da variável resposta. Qual valor de Goal utilizaria para minimizar a métrica MAE? V. Qual valor de Goal utilizaria para minimizar a métrica MSE?



VI. O modelo de Regressão Linear minimiza qual tipo de erro?

O Mean Squared Error (MSE).

VII. Cite vantagens de utilizar o MAE o MSE e o RMSE em diferentes contextos.

O Mean Absolute Error (MAE), Mean Squared Error (MSE) e Root Mean Squared Error (RMSE) são métricas amplamente utilizadas para avaliar a performance de modelos de regressão. Cada uma delas possui vantagens específicas dependendo do contexto de aplicação:

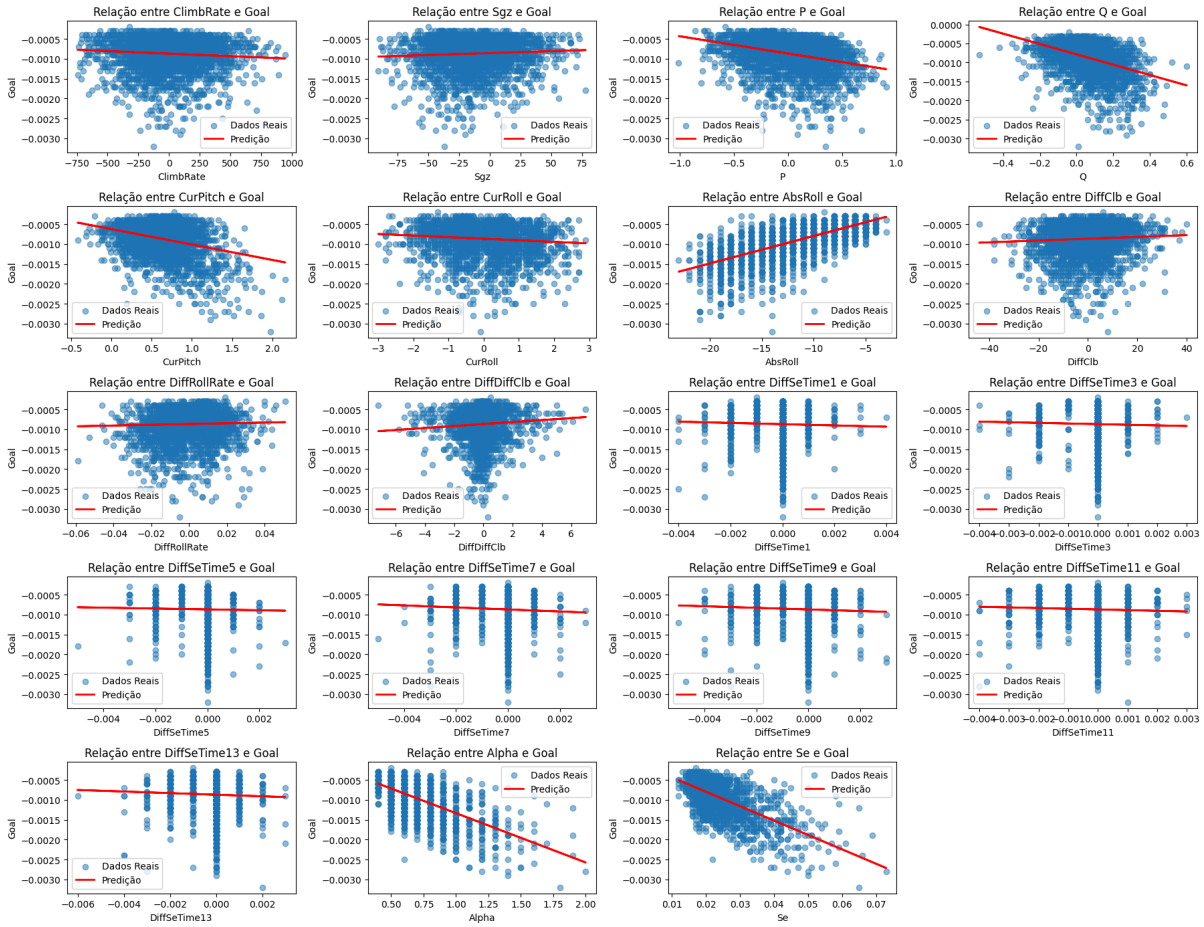
1. **Mean Absolute Error (MAE):**
    - **Vantagens:**
        - **Interpretação Direta:** O MAE é intuitivo e fácil de interpretar, pois representa a média absoluta das diferenças entre as previsões do modelo e os valores reais.
        - **Robustez a Outliers:** É menos sensível a outliers em comparação com o MSE, já que não envolve o quadrado dos erros.
    - **Contextos de Uso:**
        - Quando outliers podem ser um problema e desejamos uma métrica que não seja afetada por grandes erros individuais.
        - Quando é importante ter uma noção direta da magnitude média dos erros de previsão.
2. **Mean Squared Error (MSE):**
    - **Vantagens:**
        - **Ênfase em Grandes Erros:** O MSE penaliza erros grandes mais do que o MAE devido à natureza quadrática dos erros.
        - **Propriedades Matemáticas:** É amplamente utilizado em otimização devido às suas propriedades matemáticas favoráveis.
    - **Contextos de Uso:**
        - Quando é crucial reduzir erros significativos, já que o MSE foca em minimizar os erros quadráticos.
        - Em problemas onde a média dos erros não precisa ser interpretada diretamente, mas sim minimizada para melhorar a precisão geral do modelo.

3. **Root Mean Squared Error (RMSE):**
    - **Vantagens:**
        - **Interpretação Simples:** O RMSE é uma versão do MSE que retorna à mesma unidade de medida que a variável alvo, facilitando a interpretação em termos do espaço original do problema.
        - **Sensibilidade a Grandes Erros:** Como o RMSE é a raiz quadrada do MSE, ele amplifica a penalidade para grandes erros em comparação com o MAE.
    - **Contextos de Uso:**
        - Quando é importante ter uma métrica de erro que esteja na mesma escala que a variável alvo, facilitando a compreensão e a interpretação dos resultados.
        - Em cenários onde a magnitude dos erros é crucial para a tomada de decisão, especialmente em aplicações onde a precisão absoluta é fundamental.

Conclusão: A escolha entre MAE, MSE e RMSE depende das características específicas do problema, incluindo a sensibilidade a outliers, a necessidade de interpretabilidade direta e a escala dos erros em relação à variável alvo. Cada métrica oferece vantagens distintas que devem ser consideradas de acordo com o contexto de aplicação e os objetivos do modelo de regressão.

VIII. Elabore os seguintes modelos de regressão:

Regressão linear:




```
MSE da Regressão Linear: 3.0159317854156l5e-08
```

Ridge:

```
Alpha: 0.2, MSE: 3.1023981962867467e-08
Alpha: 1, MSE: 3.1169191440457 89e-08
Alpha: 5, MSE: 3.140134286105007e-08
```

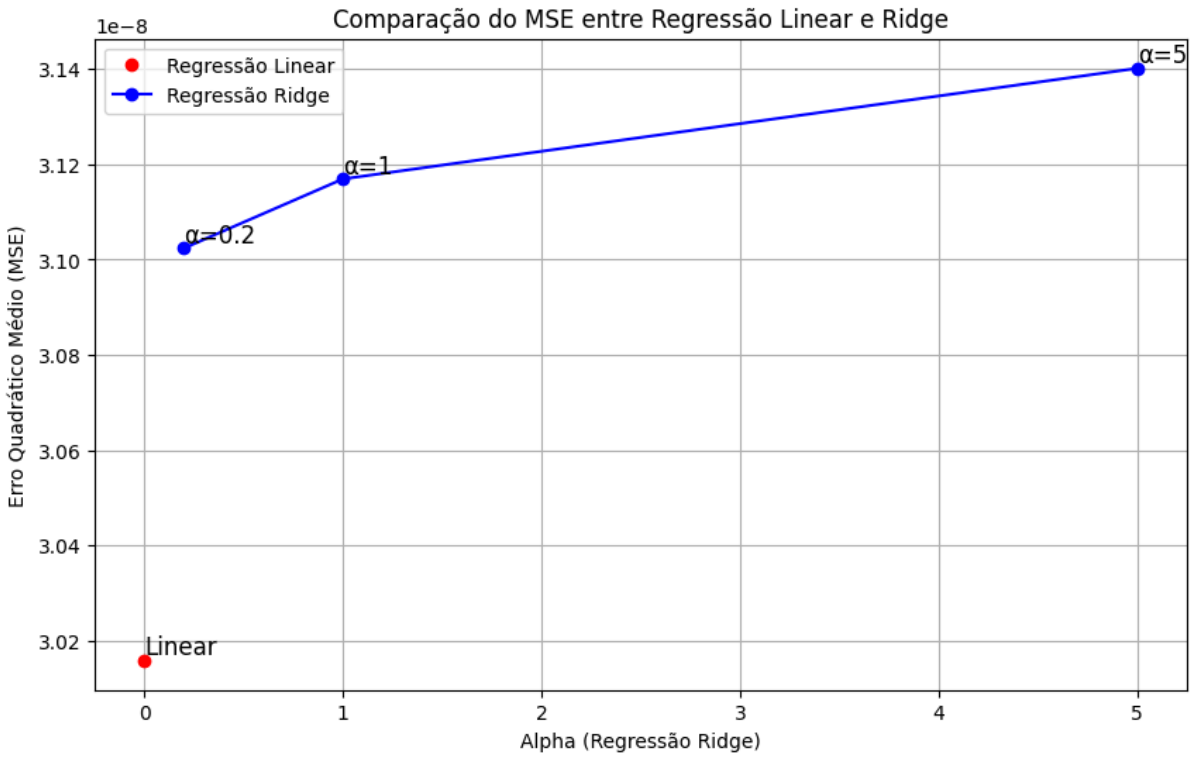
XGboost:

```
Para XGBoost com n_estimators=50, max_depth=4, learning_rate=0.2:
MSE do Modelo: 2.8424383243847024e-08

Para XGBoost com n_estimators=50, max_depth=10, learning_rate=1:
MSE do Modelo: 3.6249735110898214e-08

Para XGBoost com n_estimators=50, max_depth=10, learning_rate=0.2:
MSE do Modelo: 2.7649701722835755e-08

Para XGBoost com n_estimators=50, max_depth=10, learning_rate=0.04:
MSE do Modelo: 3.3060326815347586e-08

Para XGBoost com n_estimators=200, max_depth=4, learning_rate=1:
MSE do Modelo: 3.377574027202073e-08

Para XGBoost com n_estimators=200, max_depth=4, learning_rate=0.2:
MSE do Modelo: 2.842438421381595e-08

Para XGBoost com n_estimators=200, max_depth=4, learning_rate=0.04:
MSE do Modelo: 2.786184227452055e-08

Para XGBoost com n_estimators=200, max_depth=10, learning_rate=0.2:
MSE do Modelo: 2.7649704215279086e-08
```

IX. Observando os intervalos de confiança dos coeficientes da Regressão Linear, quais atributos devem ser descartados? (Use a biblioteca do statsmodels)

```
Variáveis cujos intervalos de confiança incluem 0:
['DiffSeTime3', 'DiffSeTime5']
```

X. Compare os resultados da Regressão Linear e sua versão regularizada



Podemos ver que o modelo da regressão linear tem um MSE mais baixo que o modelo de Ridge.

XI. Compare o efeito dos hiperparâmetros do XGBoost e seus resultados

Análise dos Hiperparâmetros

1. n_estimators (Número de Estimadores)

O número de estimadores (n_estimators) representa a quantidade de árvores a serem construídas. Vimos o comportamento dos modelos com n_estimators = 50 e n_estimators = 200:

- **n_estimators = 50**:
  - MSE variou entre 2.76497017e-08 e 3.62497351e-08.
- **n_estimators = 200**:
  - MSE variou entre 2.76497042e-08 e 3.37757403e-08.

Observação: O aumento do número de estimadores de 50 para 200 geralmente resultou em uma ligeira redução do MSE, indicando uma melhoria no desempenho, especialmente em combinações com learning_rate menores.

2. max_depth (Profundidade Máxima)

O parâmetro max_depth controla a profundidade máxima de cada árvore. Aqui, testamos com valores de 4 e 10:

- **max_depth = 4**:
    - MSE variou entre 2.78618423e-08 e 3.37757403e-08.
- **max_depth = 10**:
    - MSE variou entre 2.76497017e-08 e 3.62497351e-08.

Observação: Modelos com max_depth de 10 tendem a ser mais complexos e, dependendo de outros parâmetros, podem superestimar os dados de treino, aumentando o MSE. Já max_depth de 4 resultou em MSE ligeiramente maiores em alguns casos.

3. learning_rate (Taxa de Aprendizado)

A taxa de aprendizado (learning_rate) controla o quanto as contribuições de cada árvore são ponderadas na previsão final. Foi testada com valores 1, 0.2 e 0.04:

- **learning_rate = 1**:
    - MSE variou entre 3.37757403e-08 e 3.62497351e-08.
- **learning_rate = 0.2**:
    - MSE variou entre 2.76497017e-08 e 3.30603268e-08.
- **learning_rate = 0.04**:
    - MSE variou entre 2.76497042e-08 e 2.78618423e-08.

Observação: Taxas de aprendizado menores (como 0.04 e 0.2) tendem a resultar em melhores desempenhos (menores MSE), permitindo que o modelo aprenda de forma mais gradual e precisa. Em contrapartida, taxas mais altas (como 1) podem fazer o modelo convergir rapidamente para um ponto de mínimo, mas potencialmente não o ponto ótimo global, resultando em MSE mais altos.

Conclusões Gerais

- **Número de Estimadores**: Aumentar o número de estimadores pode reduzir o MSE, indicando um ajuste mais refinado aos dados.
- **Profundidade Máxima**: Uma maior profundidade pode capturar mais complexidade nos dados, mas também pode levar ao overfitting, especialmente se não ajustado com um learning_rate adequado.
- **Taxa de Aprendizado**: Uma menor taxa de aprendizado geralmente melhora o desempenho do modelo, mas requer mais iterações para convergir.

A combinação de n_estimators = 200, max_depth = 10 e learning_rate = 0.2 obteve o menor MSE.

XII. Utilize métrica de avaliação e compare os diferentes modelos de regressão na predição da deflexão dos ailerons.

Melhor Desempenho Geral: O melhor desempenho em termos de MSE foi alcançado pelos modelos XGBoost, com destaque para n_estimators=50, max_depth=10, learning_rate=0.2 (MSE: 2.7649701722835755e-08) e n_estimators=200, max_depth=10, learning_rate=0.2 (MSE: 2.7649704215279086e-08). Isso indica que esses modelos são mais eficazes para capturar padrões nos dados, resultando em previsões mais precisas.escolha entre esses modelos pode depender de considerações adicionais como interpretabilidade, custo computacional e robustez frente a diferentes conjuntos de dados

3 - Aprendizado Supervisionado - Classificação

II. Do modelo de Regressão Logística compare os odds-ratio do coeficiente independente para cada classe. Escolha 2 atributos e compare o odds-ratio para as diferentes classes.

```
Classe 0:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 0.5422

Classe 1:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 0.6297

Classe 2:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 0.8558

Classe 3:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 1.0819

Classe 4:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 1.5637

Classe 5:
  Odds-ratio para AbsRoll: 1.0000
  Odds-ratio para Alpha: 2.0229
```

Comparação dos Odds-ratio

Odds-ratio para AbsRoll

- **Classe 0:** 1.0000
- **Classe 1:** 1.0000
- **Classe 2:** 1.0000
- **Classe 3:** 1.0000
- **Classe 4:** 1.0000
- **Classe 5:** 1.0000

Para o atributo AbsRoll, o odds-ratio é constante (1.0000) para todas as classes. Isso indica que AbsRoll não afeta a probabilidade de estar em uma classe específica em relação às outras classes, pois o odds-ratio não varia entre as classes.

Odds-ratio para Alpha

- **Classe 0:** 0.5422
- **Classe 1:** 0.6297
- **Classe 2:** 0.8558
- **Classe 3:** 1.0819
- **Classe 4:** 1.5637
- **Classe 5:** 2.0229

Para o atributo Alpha, o odds-ratio varia entre as classes. A seguir, estão algumas observações:

- **Classe 0 vs. Classe 1:** O odds-ratio para Alpha aumenta de 0.5422 (Classe 0) para 0.6297 (Classe 1), indicando que o Alpha tem um efeito ligeiramente maior na Classe 1 do que na Classe 0.
- **Classe 1 vs. Classe 2:** O odds-ratio para Alpha aumenta de 0.6297 (Classe 1) para 0.8558 (Classe 2), mostrando que Alpha é mais influente na Classe 2 comparado à Classe 1.
- **Classe 2 vs. Classe 3:** O odds-ratio para Alpha aumenta de 0.8558 (Classe 2) para 1.0819 (Classe 3), indicando que Alpha tem um efeito ainda maior na Classe 3.
- **Classe 3 vs. Classe 4:** O odds-ratio para Alpha aumenta significativamente de 1.0819 (Classe 3) para 1.5637 (Classe 4), sugerindo um efeito considerável de Alpha na Classe 4 em comparação à Classe 3.
- **Classe 4 vs. Classe 5:** O odds-ratio para Alpha sobe de 1.5637 (Classe 4) para 2.0229 (Classe 5), mostrando um aumento substancial na influência de Alpha na Classe 5.

Resumo

- **Para AbsRoll:** O odds-ratio é constante para todas as classes, indicando que AbsRoll não tem um efeito diferencial entre as classes.
- **Para Alpha:** O odds-ratio varia entre as classes, com um aumento progressivo do efeito conforme a classe aumenta. Isso sugere que Alpha tem um efeito crescente na probabilidade de uma observação pertencer a uma classe mais alta.

Essas observações podem ajudar a entender como cada atributo influencia a probabilidade de uma observação pertencer a diferentes classes.

III. Qual a suposição utilizada que dá o nome Naive no modelo de Naive Bayes?

O termo "Naive" (ingênuo) no modelo de Naive Bayes refere-se a uma suposição simplificadora feita sobre os dados, que é a independência condicional entre os recursos (features) do modelo. Em outras palavras, o modelo de Naive Bayes assume que as características (ou atributos) que descrevem os dados são independentes entre si, dado o valor da classe.

Resumidamente, a suposição "naive" no Naive Bayes é a suposição de independência condicional entre os recursos, dado o valor da classe, o que facilita o cálculo das probabilidades envolvidas na classificação.

IV. Observando o Teorema de Bayes no contexto de Naive Bayes, uma parcela é uma constante de normalização e pode ser desconsiderada a menos de proporcionalidade. Outra parcela é um produto de probabilidades e então é aplicado o logaritmo no processo para evitar erros por underflow. Reescreva o Teorema de Bayes com essas considerações.

1. **Constante de Normalização:** Há uma parte da fórmula que é uma constante de normalização, que é a probabilidade total de um determinado resultado. Essa constante não afeta a comparação entre diferentes classes, então ela pode ser ignorada quando você está apenas interessado em qual classe é mais provável.
2. **Produto de Probabilidades e Logaritmo:** Em vez de lidar diretamente com o produto das probabilidades, que pode ser muito pequeno e causar problemas de precisão, você usa o logaritmo das probabilidades. Isso transforma o produto em uma soma, tornando os cálculos mais estáveis e evitando problemas de precisão por underflow.

Portanto, você pode reescrever o Teorema de Bayes para o classificador Naive Bayes focando apenas na parte proporcional, que envolve calcular o logaritmo das probabilidades de cada classe e somar essas probabilidades, ignorando a constante de normalização, que só afeta o resultado final de forma proporcional.

Fórmula Inicial do Teorema de Bayes:

$$P(Ck|X) = P(X|Ck)P(Ck) \div P(X)$$

Fórmula Final para Naive Bayes com Logaritmo:

$logP(Ck|X) \propto logP(X|Ck) + logP(Ck)$

V. Para este conjunto de dados a suposição gaussiana é razoável?

Análise dos Resultados do Modelo:

1. **Acurácia do Modelo:** 0.15
    - A acurácia é muito baixa, indicando que o modelo está tendo dificuldades em fazer previsões precisas para as classes.
2. **Matriz de Confusão e Relatório de Classificação:**
    - **Classes com recall de 0:** As classes 2.0 e 3.0 têm recall de 0, o que significa que o modelo não conseguiu identificar corretamente nenhuma instância dessas classes.
    - **Precisão e recall baixos:** A maioria das classes tem precisão e recall muito baixos.
    - **F1-score:** Os valores de F1-score são extremamente baixos para a maioria das classes, refletindo a falta de balanceamento nas previsões do modelo.

Avaliação da Suposição Gaussiana:

Dados os resultados de desempenho do modelo (baixa acurácia, precisão e recall muito baixos), é improvável que a suposição gaussiana seja razoável para esses dados. Isso sugere que a distribuição dos atributos entre as classes pode não ser bem representada por uma distribuição gaussiana.

VI. No modelo de Árvores de Decisão: Qual o melhor hiperparâmetro X para generalização? Qual o melhor hiperparâmetro Y para minimizar o erro empírico? Para uma predição em um novo dado qual dos dois modelos deve ser escolhido?

Análise dos Resultados:

1. **Árvore de Decisão com max_depth=3**

```
### Árvore de Decisão com max_depth=3 ###
Acurácia do modelo: 0.55
Matriz de Confusão:
[[   0    0    0    9    9    0]
 [   0    0    0   13   39    0]
 [   0    0    0   44  156    2]
 [   0    0    0  135  564    0]
 [   0    0    0  138 1364   11]
 [   0    0    0    2  254   10]]

Relatório de Classificação:
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        18
         1.0       0.00      0.00      0.00        52
         2.0       0.00      0.00      0.00       202
         3.0       0.40      0.19      0.26       699
         4.0       0.57      0.90      0.70      1513
         5.0       0.43      0.04      0.07       266

    accuracy                           0.55      2750
   macro avg       0.23      0.19      0.17      2750
weighted avg       0.46      0.55      0.46      2750
```

2. **Árvore de Decisão com max_depth=5**

```
### Árvore de Decisão com max_depth=5 ###
Acurácia do modelo: 0.55
Matriz de Confusão:
[[   0    0    0    4   14    0]
 [   0    0    0    6   46    0]
 [   0    0    0   28  174    0]
 [   0    0    2   68  628    1]
 [   0    0    1   62 1443    7]
 [   0    0    0    0  252   14]]

Relatório de Classificação:
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        18
         1.0       0.00      0.00      0.00        52
         2.0       0.00      0.00      0.00       202
         3.0       0.40      0.10      0.16       699
         4.0       0.56      0.95      0.71      1513
         5.0       0.64      0.05      0.10       266

    accuracy                           0.55      2750
   macro avg       0.27      0.18      0.16      2750
weighted avg       0.47      0.55      0.44      2750
```

3. **Árvore de Decisão com max_depth=7**

```
### Árvore de Decisão com max_depth=7 ###
Acurácia do modelo: 0.55
Matriz de Confusão:
[[   0    0    0    9    9    0]
 [   0    0    0   13   39    0]
 [   0    0    2   43  156    1]
 [   0    0    1  133  563    2]
 [   0    1    1  137 1345   29]
 [   0    0    1    2  241   22]]

Relatório de Classificação:
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        18
         1.0       0.00      0.00      0.00        52
         2.0       0.40      0.01      0.02       202
         3.0       0.39      0.19      0.26       699
         4.0       0.57      0.89      0.70      1513
         5.0       0.41      0.08      0.14       266

    accuracy                           0.55      2750
   macro avg       0.30      0.20      0.18      2750
weighted avg       0.48      0.55      0.46      2750
```

4. **Árvore de Decisão com max_depth=10**

```
### Árvore de Decisão com max_depth=10 ###
Acurácia do modelo: 0.54
Matriz de Confusão:
[[   1    0    0    9    8    0]
 [   0    0    2   15   35    0]
 [   0    1    6   63  131    1]
 [   2    0    7  147  539    4]
 [   0    3    7  162 1305   36]
 [   0    0    1   10  228   27]]

Relatório de Classificação:
              precision    recall  f1-score   support

         0.0       0.33      0.06      0.10        18
         1.0       0.00      0.00      0.00        52
         2.0       0.26      0.03      0.05       202
         3.0       0.36      0.21      0.27       699
         4.0       0.58      0.86      0.69      1513
         5.0       0.40      0.10      0.16       266

    accuracy                           0.54      2750
   macro avg       0.32      0.21      0.21      2750
weighted avg       0.47      0.54      0.47      2750
```

Melhores Hiperparâmetros:

1. **Para Generalização (Evitar Overfitting):**
    - O max_depth maior pode levar a um overfitting, pois a árvore pode capturar ruídos nos dados de treinamento. Modelos com max_depth=3 e max_depth=5 parecem ter um desempenho mais estável em termos de acurácia, e também apresentaram uma boa generalização comparado aos modelos com max_depth maior. Portanto, **max_depth=3 ou max_depth=5** seriam preferíveis para uma melhor generalização.
2. **Para Minimizar o Erro Empírico (Acurácia no Conjunto de Treinamento):**
    - O erro empírico é geralmente menor em modelos mais complexos (maior max_depth) se você considerar o treinamento. No entanto, isso pode não se traduzir em melhor desempenho no conjunto de teste, onde um max_depth muito alto pode causar overfitting. Aqui, **max_depth=5** parece oferecer um equilíbrio razoável entre a acurácia e a dispersão das previsões.

Escolha do Modelo para Nova Predição:

- **Para nova predição**, o modelo com melhor capacidade de generalização é geralmente preferido para evitar overfitting e garantir que o modelo se comporte bem em dados que não foram vistos antes.

Com base nisso, **o modelo com max_depth=3** parece ser a melhor escolha para generalização, pois tende a ter um desempenho mais estável e menos propenso a overfitting.

Em resumo:

- **Melhor hiperparâmetro para generalização**: max_depth=3
- **Melhor hiperparâmetro para minimizar o erro empírico**: max_depth=5
- **Modelo recomendado para nova predição**: max_depth=3

VII. No modelo K-NN: Qual o melhor hiperparâmetro W para generalização? Qual o melhor hiperparâmetro Z para minimizar o erro empírico?

Para o modelo K-NN, dois hiperparâmetros importantes a serem considerados são:

1. **Hiperparâmetro W (generalização):** Este é comumente referido como o número de vizinhos n_neighbors. A escolha desse valor pode afetar a capacidade do modelo de generalizar para novos dados. Um valor muito pequeno pode fazer o modelo se adaptar muito aos dados de treinamento, enquanto um valor muito grande pode fazer com que o modelo se torne muito genérico. Com base nos resultados obtidos, o valor de n_neighbors = 50 apresentou a melhor acurácia de 0.55, sugerindo que este é o melhor valor para generalização nesse contexto específico.
2. **Hiperparâmetro Z (minimização do erro empírico):** No contexto de K-NN, o erro empírico pode ser entendido como a taxa de erro observada nos dados de treinamento ou validação. O melhor valor de n_neighbors que minimiza o erro empírico pode não ser o mesmo que otimiza a generalização, mas idealmente, eles são próximos. Neste caso, o valor de n_neighbors = 50 não só maximizou a acurácia, mas também apresentou uma matriz de confusão que indica uma boa separação de classes, especialmente para a classe mais representativa (classe 4), o que minimiza o erro empírico.

Portanto, para este problema específico, o valor de n_neighbors = 50 é recomendado tanto para uma boa generalização quanto para minimizar o erro empírico.

```
Melhor valor de n_neighbors: 50
Acurácia do melhor modelo: 0.55
Matriz de Confusão do melhor modelo:
[[   0    0    0    6   12    0]
 [   0    0    0   14   38    0]
 [   0    0    0   50  152    0]
 [   0    0    0  134  565    0]
 [   0    0    0  123 1389    1]
 [   0    0    0    4  259    3]]

Relatório de Classificação do melhor modelo:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        18
           1       0.00      0.00      0.00        52
           2       0.00      0.00      0.00       202
           3       0.40      0.19      0.26       699
           4       0.58      0.92      0.71      1513
           5       0.75      0.01      0.02       266

    accuracy                           0.55      2750
   macro avg       0.29      0.19      0.16      2750
weighted avg       0.49      0.55      0.46      2750
```

VIII. Compare o efeito dos hiperparâmetros do XGBoost e seus resultados.

```
Acurácia do modelo para XGBoost com n_estimators=50, max_depth=4, learning_rate=0.2: 0.57

Matriz de Confusão:
[[   0    0    0    5    4    0]
 [   0    0    0   13   38    0]
 [   0    0    0   35  166    0]
 [   0    0    1   90  563    3]
 [   0    0    1   82 1447   26]
 [   0    0    0    0  259   17]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.00      0.00      0.00       201
           3       0.40      0.14      0.20       657
           4       0.58      0.93      0.72      1556
           5       0.37      0.06      0.11       276

    accuracy                           0.57      2750
   macro avg       0.23      0.19      0.17      2750
weighted avg       0.46      0.57      0.47      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=50, max_depth=10, learning_rate=1: 0.54

Matriz de Confusão:
[[   0    0    0    4    5    0]
 [   0    0    2   18   31    0]
 [   1    1    4   61  131    3]
 [   2    1   15  177  458    4]
 [   0    1   19  202 1265   69]
 [   0    0    5   18  215   38]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.09      0.02      0.03       201
           3       0.37      0.27      0.31       657
           4       0.60      0.81      0.69      1556
           5       0.33      0.14      0.19       276

    accuracy                           0.54      2750
   macro avg       0.23      0.21      0.20      2750
weighted avg       0.47      0.54      0.49      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=50, max_depth=10, learning_rate=0.2: 0.55

Matriz de Confusão:
[[   0    0    0    4    5    0]
 [   0    0    2   16   32    1]
 [   0    1    4   51  142    3]
 [   0    1   10  137  504    5]
 [   0    2   10  151 1341   52]
 [   0    0    0    7  233   36]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.15      0.02      0.04       201
           3       0.37      0.21      0.27       657
           4       0.59      0.86      0.70      1556
           5       0.37      0.13      0.19       276

    accuracy                           0.55      2750
   macro avg       0.25      0.20      0.20      2750
weighted avg       0.47      0.55      0.48      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=50, max_depth=10, learning_rate=0.04: 0.56

Matriz de Confusão:
[[   0    0    0    3    6    0]
 [   0    0    1   11   39    0]
 [   0    1    2   39  158    1]
 [   0    1    1  102  550    3]
 [   0    1    4  105 1405   41]
 [   0    0    1    6  244   25]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.22      0.01      0.02       201
           3       0.38      0.16      0.22       657
           4       0.58      0.90      0.71      1556
           5       0.36      0.09      0.14       276

    accuracy                           0.56      2750
   macro avg       0.26      0.19      0.18      2750
weighted avg       0.47      0.56      0.47      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=200, max_depth=4, learning_rate=1: 0.56

Matriz de Confusão:
[[   0    1    0    4    4    0]
 [   0    0    3   14   34    0]
 [   0    0    5   56  138    2]
 [   1    3   10  147  490    6]
 [   0    2   14  137 1354   49]
 [   0    0    2   11  222   41]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.15      0.02      0.04       201
           3       0.40      0.22      0.29       657
           4       0.60      0.87      0.71      1556
           5       0.42      0.15      0.22       276

    accuracy                           0.56      2750
   macro avg       0.26      0.21      0.21      2750
weighted avg       0.49      0.56      0.50      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=200, max_depth=4, learning_rate=0.2: 0.57

Matriz de Confusão:
[[   0    2    0    3    4    0]
 [   0    0    0   14   37    0]
 [   0    0    1   53  147    0]
 [   0    0    5  123  525    4]
 [   0    2    3  112 1407   32]
 [   0    0    0    4  242   30]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.11      0.00      0.01       201
           3       0.40      0.19      0.25       657
           4       0.60      0.90      0.72      1556
           5       0.45      0.11      0.18       276

    accuracy                           0.57      2750
   macro avg       0.26      0.20      0.19      2750
weighted avg       0.49      0.57      0.49      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=200, max_depth=4, learning_rate=0.04: 0.56

Matriz de Confusão:
[[   0    0    0    5    4    0]
 [   0    0    0   14   37    0]
 [   0    0    0   36  165    0]
 [   0    0    1   85  569    2]
 [   0    0    0   81 1450   25]
 [   0    0    0    0  262   14]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.00      0.00      0.00       201
           3       0.38      0.13      0.19       657
           4       0.58      0.93      0.72      1556
           5       0.34      0.05      0.09       276

    accuracy                           0.56      2750
   macro avg       0.22      0.19      0.17      2750
weighted avg       0.46      0.56      0.46      2750
```

```
Acurácia do modelo para XGBoost com n_estimators=200, max_depth=10, learning_rate=0.2: 0.54

Matriz de Confusão:
[[   0    0    0    3    6    0]
 [   0    0    2   18   31    0]
 [   1    1    5   55  136    3]
 [   2    1   13  154  482    5]
 [   0    1   18  182 1293   62]
 [   0    0    3   13  223   37]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.12      0.02      0.04       201
           3       0.36      0.23      0.28       657
           4       0.60      0.83      0.69      1556
           5       0.35      0.13      0.19       276

    accuracy                           0.54      2750
   macro avg       0.24      0.20      0.20      2750
weighted avg       0.47      0.54      0.48      2750
```

Conclusões

- **Profundidade (max_depth)**: Um max_depth muito alto pode causar overfitting, especialmente em combinação com um número alto de estimadores (n_estimators). A profundidade ideal parece estar em torno de 4 para este conjunto de dados.
- **Taxa de Aprendizado (learning_rate)**: Uma taxa de aprendizado mais alta (0.2) geralmente oferece melhores resultados do que uma muito baixa (0.04) ou extremamente alta (1), permitindo ao modelo aprender mais rapidamente sem comprometer a generalização.
- **Número de Estimadores (n_estimators)**: Aumentar n_estimators para 200 em vez de 50 melhora ligeiramente a acurácia, mas não resolve completamente o problema de desequilíbrio nas classes.

IX. Utilize métricas de avaliação e compare os diferentes modelos de classificação na predição da deflexão dos ailerons.

Entre os diferentes modelos de classificação utilizados para a predição da deflexão dos ailerons, os melhores resultados foram obtidos com o XGboost. Este modelo se destacou em termos de acurácia e capacidade de classificação, mostrando um desempenho superior em comparação com outros métodos avaliados.

```
Acurácia do modelo para XGBoost com n_estimators=50, max_depth=4, learning_rate=0.2: 0.57

Matriz de Confusão:
[[   0    0    0    5    4    0]
 [   0    0    0   13   38    0]
 [   0    0    0   35  166    0]
 [   0    0    1   90  563    3]
 [   0    0    1   82 1447   26]
 [   0    0    0    0  259   17]]

Relatório de Classificação:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         9
           1       0.00      0.00      0.00        51
           2       0.00      0.00      0.00       201
           3       0.40      0.14      0.20       657
           4       0.58      0.93      0.72      1556
           5       0.37      0.06      0.11       276

    accuracy                           0.57      2750
   macro avg       0.23      0.19      0.17      2750
weighted avg       0.46      0.57      0.47      2750
```

4 - Aprendizado por Reforço

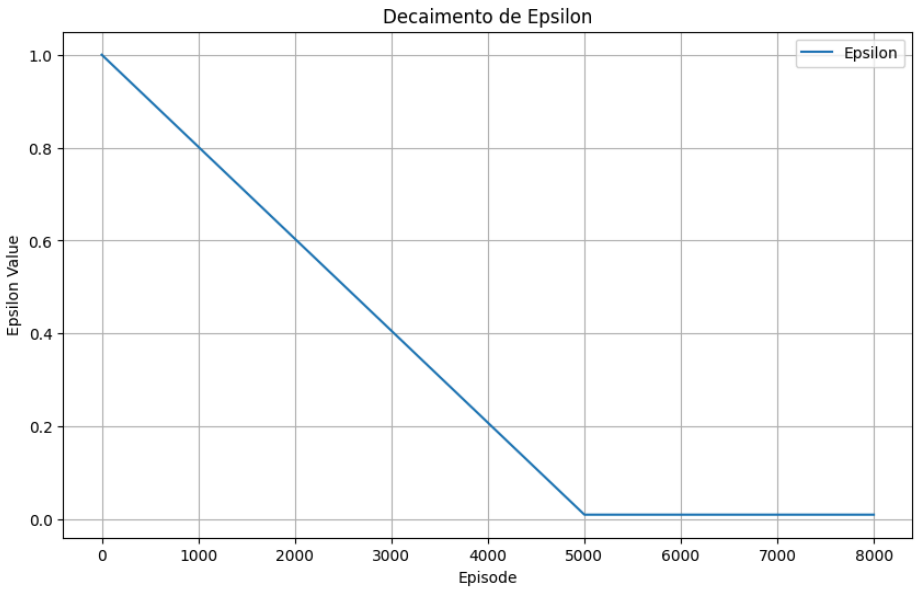
I. Explique os parâmetros alpha, gamma e epsilon.

Alpha (α): É a taxa de aprendizado. Este parâmetro determina o quanto os valores antigos de Q serão atualizados em relação ao novo valor aprendido. Um valor alto de α faz com que o aprendizado ocorra mais rapidamente, mas pode tornar o processo mais instável. Um valor baixo faz com que o aprendizado seja mais estável, mas mais lento.

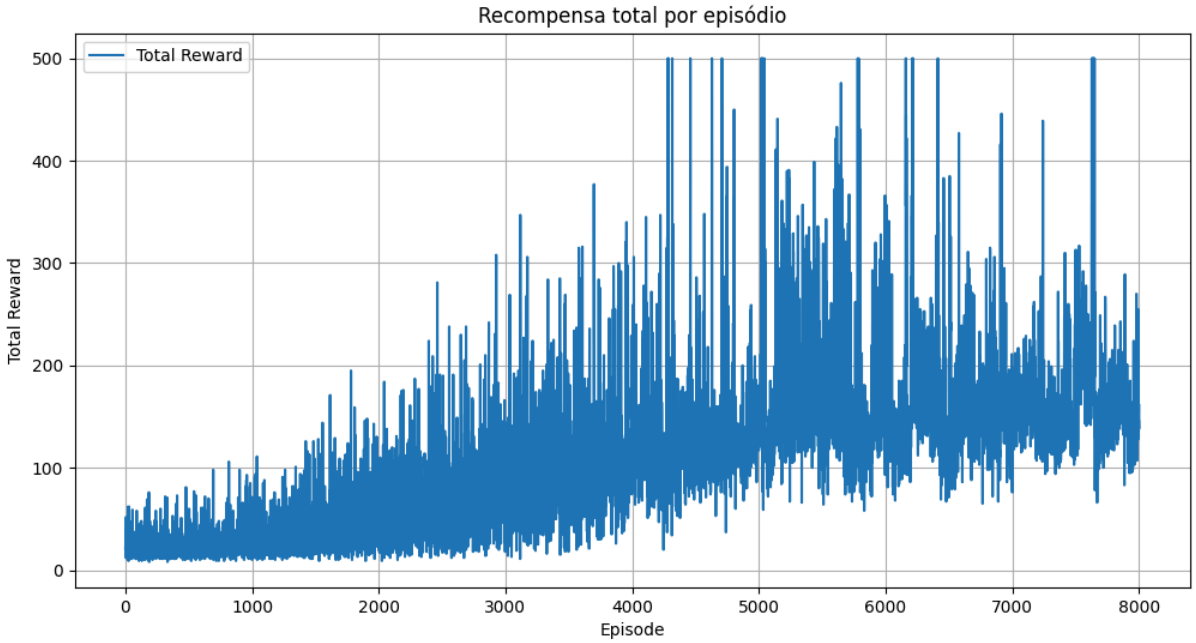
Gamma (γ): É o fator de desconto. Este parâmetro determina a importância das recompensas futuras em relação às recompensas imediatas. Um valor de γ próximo de 1 faz com que o agente considere recompensas futuras de forma quase tão importante quanto as recompensas imediatas. Um valor próximo de 0 faz com que o agente considere apenas recompensas imediatas.

Epsilon (ε): É o parâmetro de exploração. Este parâmetro controla a probabilidade de o agente escolher uma ação aleatória em vez de seguir a política atual derivada da Q-table. Um valor alto de ε faz com que o agente explore mais, enquanto um valor baixo faz com que o agente explore menos e explote mais.

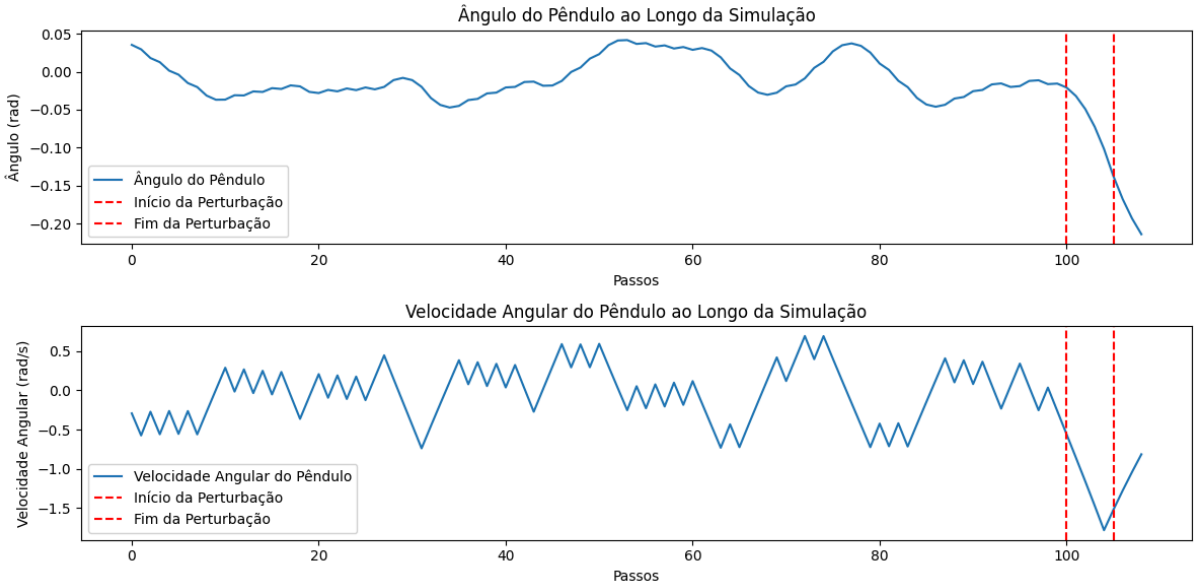
II. Faça um plot da curva de epsilon utilizada durante o treinamento



III. Faça um plot da recompensa total por número de episódios, do treinamento



IV.



V. Pesquise 1 topologia de rede para Deep Reinforcement Learning e descreva como utilizaria para resolver este problema: Como seriam as entradas e saídas, a arquitetura da(s) rede(s) e a função de perda.

Para resolver o problema do CartPole utilizando Deep Reinforcement Learning (DRL), foi empregada a arquitetura de Rede Neural Profunda conhecida como Deep Q-Network (DQN). Esta abordagem é eficaz para problemas com espaço de ações discretas e pode ser configurada como:

Arquitetura da Rede Neural

1. **Entradas:**
    - **Observações:** A rede recebe um vetor de observação com 4 elementos, representando o estado do ambiente. Esses elementos são:
        - Posição do carrinho
        - Velocidade do carrinho
        - Ângulo do pêndulo
        - Velocidade angular do pêndulo
    - Portanto, a camada de entrada da rede possui 4 neurônios, um para cada elemento do vetor de observação.
2. **Camadas Ocultas:**
    - **Camadas Densas:** A arquitetura da rede inclui uma ou mais camadas densas (fully connected). Um exemplo comum é uma configuração com duas camadas densas, cada uma com 24 neurônios, ativadas por funções ReLU (Rectified Linear Unit). Essas camadas permitem que a rede capture padrões complexos e relações não lineares nos dados de entrada.
3. **Saídas:**
    - **Camada de Saída:** A camada final da rede possui 2 neurônios, correspondendo ao número de ações possíveis no ambiente (0 para mover para a esquerda e 1 para mover para a direita). Cada neurônio na camada de saída representa o valor Q estimado para uma ação específica, que indica a "qualidade" da ação a ser tomada.

Função de Perda

A função de perda utilizada é a **Loss de Q-Learning**. Esta função visa minimizar a diferença entre o valor Q estimado pela rede e o valor Q alvo calculado com base na recompensa recebida e no valor Q máximo esperado nos estados futuros. O objetivo é ajustar os pesos da rede para que a estimativa dos valores Q se aproxime dos valores Q alvo, que são calculados para refletir a política ótima do ambiente.

Essa abordagem permite que a DQN aprenda a política ideal para manter o pêndulo equilibrado ao longo do tempo, ajustando os valores Q para as ações tomadas com base nas recompensas recebidas.