



**UNIVERSIDADE FEDERAL DE MINAS GERAIS (UFMG)**  
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

NICOLAS VON DOLINGER MOREIRA ROCHA (2022035571)

**TRABALHO PRÁTICO 2**  
**INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL**

BELO HORIZONTE

2024

## TRABALHO PRÁTICO 2

Nicolas von Dolinger Moreira Rocha<sup>1</sup>

### 1. Introdução

Este trabalho prático, aborda a aplicação de técnicas de aprendizado supervisionado e por reforço para resolver problemas práticos no contexto da aviação e controle de sistemas.

Na primeira parte, focamos em aprendizado supervisionado, desenvolvendo modelos de regressão para prever a deflexão dos ailerons de uma aeronave com base em dados históricos. A análise inclui a avaliação de diferentes modelos de regressão e a comparação de métricas como MAE, MSE e RMSE, além da regularização com Ridge Regression e a implementação de XGBoost. Na segunda parte, tratamos de um problema de classificação, discretizando variáveis e criando modelos como Regressão Logística, Naive Bayes, Árvores de Decisão, K-NN e XGBoost, para garantir maior conforto durante o voo. Na terceira e última parte, implementamos o algoritmo de Q-learning para controlar o pêndulo invertido no ambiente CartPole, investigando parâmetros e visualizando resultados do treinamento.

Este trabalho visa não apenas desenvolver modelos de aprendizado de máquina eficazes, mas também entender as teorias subjacentes e aplicar os conceitos em problemas reais.

Link para o notebook:

<https://colab.research.google.com/drive/18bkLcdb3ymS4Wk-lw0hGWZ6LKzHv7Zfc?usp=sharing>

### 2. Perguntas e Respostas

#### 2.1) Aprendizado Supervisionado - Regressão

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

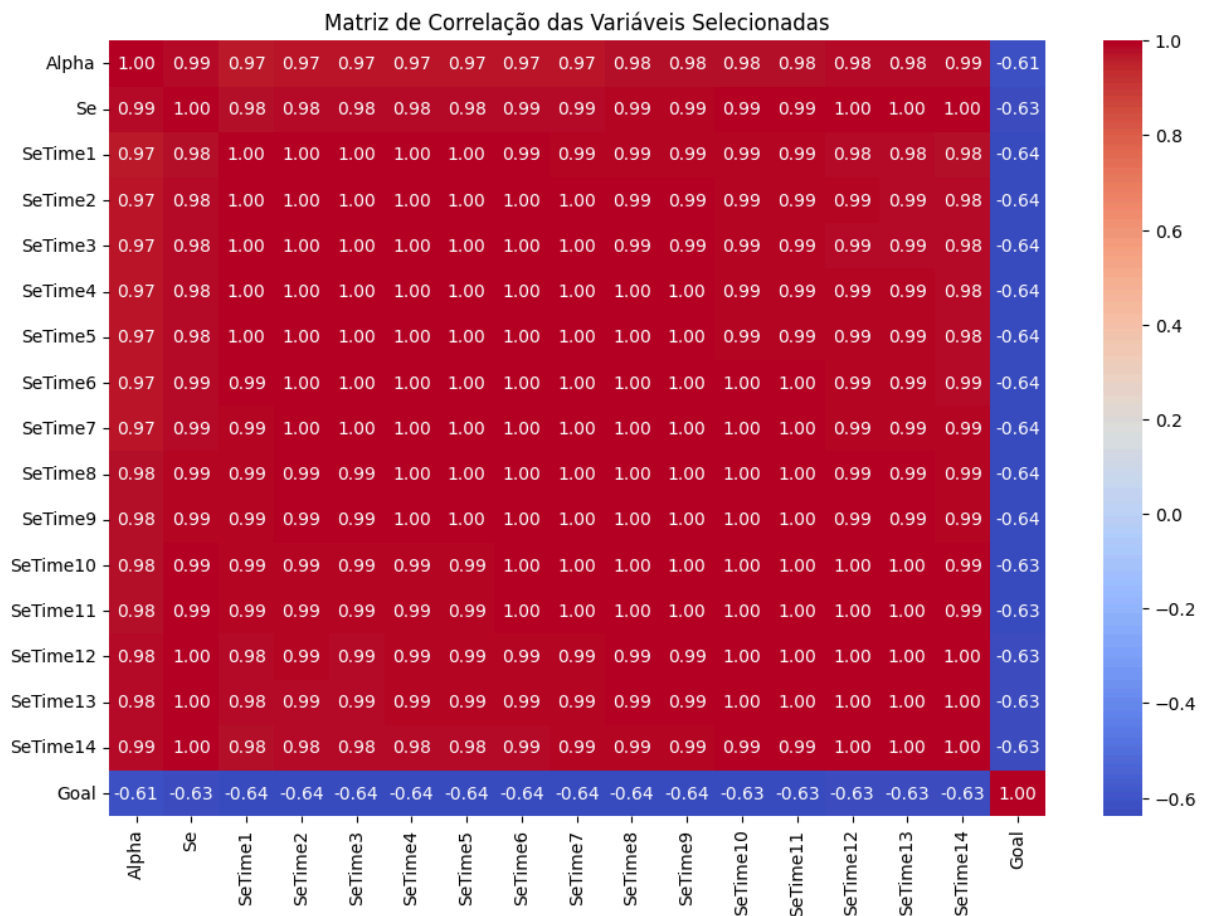
A contribuição dos atributos Alpha, Se e [SeTime1: SeTime14] para um modelo pode não ser significativa em conjunto por várias razões. Vamos examinar algumas delas:

---

<sup>1</sup> Graduando em Sistemas de Informação pela Universidade Federal de Minas Gerais (UFMG). Product Owner na iJunior, empresa júnior do Departamento de Ciência da Computação da UFMG. Monitor de Programação e Desenvolvimento de Software II.

1. Colinearidade: Quando várias variáveis são altamente correlacionadas entre si, uma ou mais podem não adicionar valor adicional ao modelo, já que elas contêm informações redundantes. Por exemplo, SeTime1, SeTime2, ..., SeTime14 podem estar fortemente correlacionadas entre si e com Se, reduzindo sua individualidade informativa.
2. Relevância: Alpha (ângulo de ataque) e Se (deflexão da superfície de controle) podem não ser diretamente relevantes para prever a deflexão do aileron (Goal) quando comparados com variáveis mais diretamente ligadas à dinâmica de voo, como ClimbRate, Sgz, P, Q, etc.
3. Ruído: As variáveis SeTime1, SeTime2, ..., SeTime14 podem introduzir mais ruído do que sinal se não forem bem representativas das mudanças relevantes para a previsão da deflexão do aileron.
4. Sobreajuste: Incluir um grande número de variáveis pode levar ao sobreajuste, onde o modelo se ajusta muito bem aos dados de treinamento, mas não generaliza bem para novos dados. Diminuir o número de variáveis pode melhorar a capacidade de generalização do modelo.

Para ilustrar isso, vamos fazer uma análise básica de colinearidade e correlação entre essas variáveis.



Existe uma alta correlação entre Alpha, Se e os tempos sequenciais, isso pode explicar porque eles não contribuem significativamente para o modelo quando usados juntos.

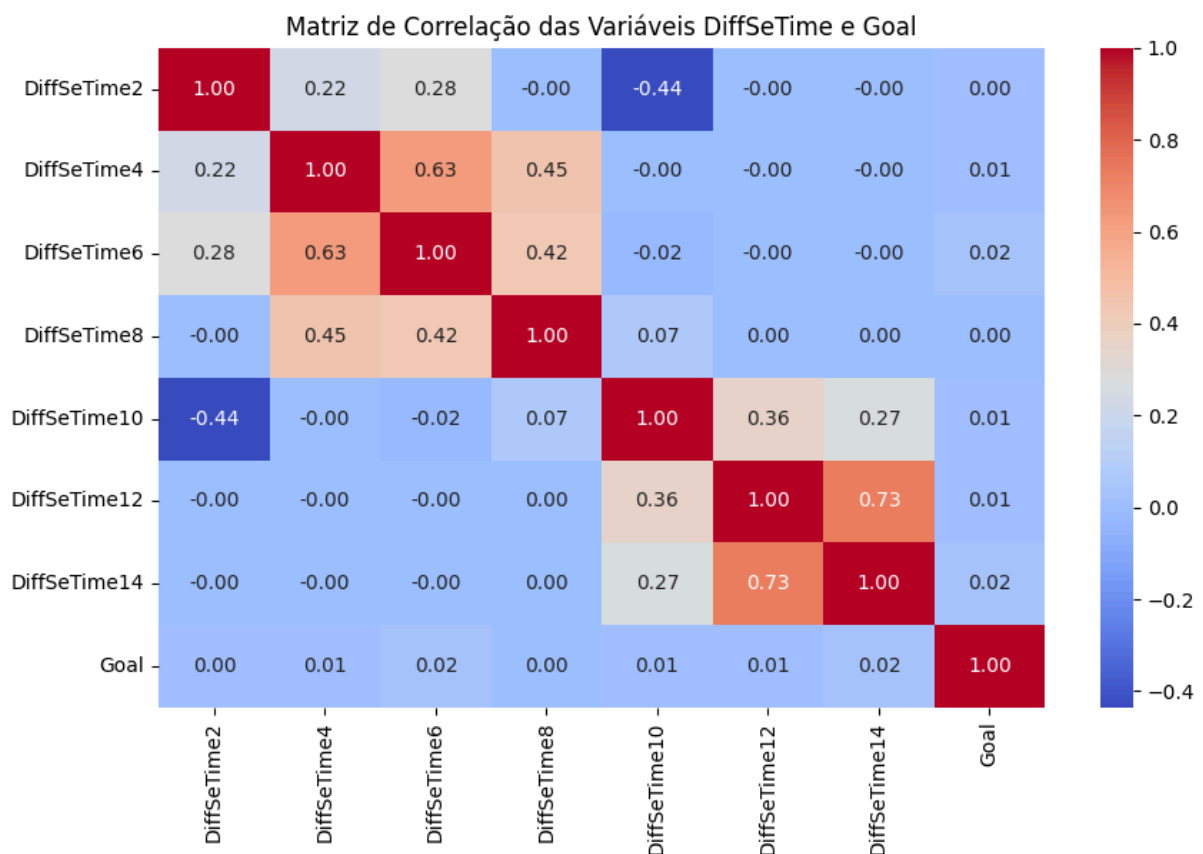
**ii. Porque os atributos DiffSeTime[2i] para i de 1 a 7 tendem a não contribuir para o modelo?**

Os atributos DiffSeTime[2i] para i de 1 a 7 (ou seja, DiffSeTime2, DiffSeTime4, DiffSeTime6, DiffSeTime8, DiffSeTime10, DiffSeTime12, DiffSeTime14) podem não contribuir significativamente para o modelo por algumas razões possíveis, como:

1. Colinearidade: Esses atributos podem estar altamente correlacionados com outros atributos ou entre si, adicionando pouca informação nova ao modelo.
2. Relevância: Eles podem não ser diretamente relevantes para a variável alvo (Goal) comparado a outras variáveis.
3. Ruído: Podem adicionar mais ruído do que sinal ao modelo.
4. Multicolinearidade: A presença de múltiplas variáveis altamente correlacionadas pode prejudicar o desempenho do modelo.

Para verificar a importância desses atributos, podemos fazer uma análise de correlação e também usar um método de seleção de características. Neste código, calculamos a matriz de correlação para as variáveis DiffSeTime[2i] e a variável alvo Goal para visualizar as correlações. Além disso, utilizamos a técnica de Recursive Feature Elimination (RFE) para verificar a importância relativa de todas as variáveis no dataset, incluindo os DiffSeTime.

Isso permitirá identificar se as variáveis DiffSeTime[2i] são menos importantes e se elas não estão contribuindo significativamente para o modelo devido à colinearidade, irrelevância, ou outros fatores.



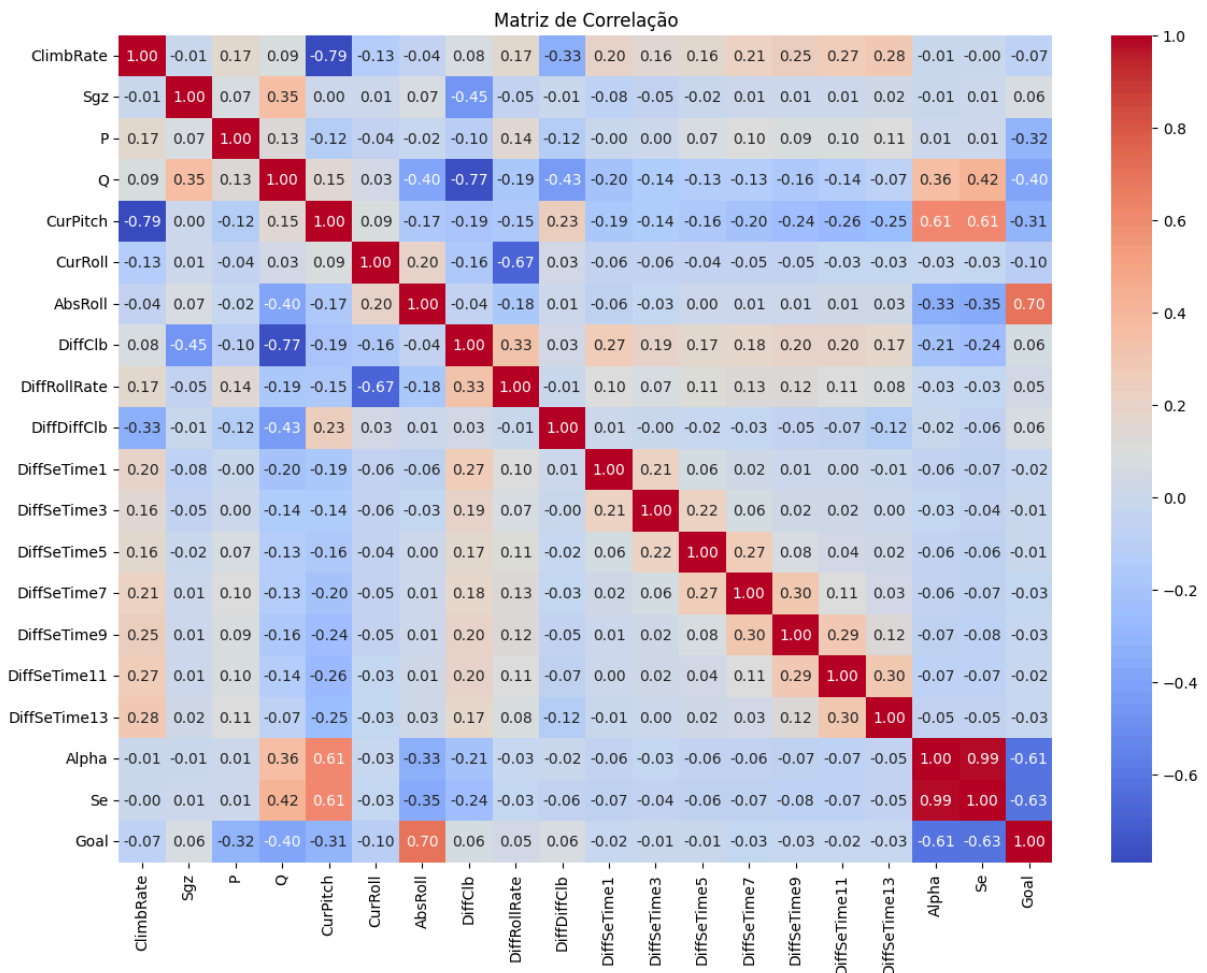
**iii. A partir desta questão utilize apenas as variáveis em instrucoes.txt. Observando uma matriz de correlação e pair\_plots. Quais os atributos mais promissores para um modelo de predição Linear de Goal?**

Para determinar quais atributos são mais promissores para um modelo de predição linear da variável alvo (Goal), podemos analisar a matriz de correlação e visualizar

pair plots para identificar correlações fortes entre as variáveis independentes e a variável dependente.

**Matriz de Correlação:** Na matriz de correlação, estamos interessados em encontrar as variáveis que têm uma alta correlação positiva ou negativa com a variável Goal. As variáveis com coeficientes de correlação próximos a 1 ou -1 são as mais promissoras para um modelo de predição linear, pois indicam uma relação linear forte com a variável alvo.

**Pair Plots:** Os pair plots permitem visualizar as relações entre pares de variáveis, incluindo a variável alvo Goal. Eles ajudam a identificar relações lineares, tendências e padrões que podem não ser óbvios na matriz de correlação.



Resultado: AbsRoll, Alpha, Se.

Considere um modelo de predição pontual da deflexão dos ailerons (Goal) e na AUSÊNCIA de qualquer atributo para IV e V:

**iv e v. Faça um histograma da variável resposta. Qual valor de Goal utilizaria para minimizar a métrica MAE? Qual valor de Goal utilizaria para minimizar a métrica MSE?**

Para determinar o valor de Goal que minimiza as métricas MAE (Mean Absolute Error) e MSE (Mean Squared Error) na ausência de qualquer atributo, seguiremos o passo a passo:

1. Plotar o histograma da variável Goal para entender sua distribuição.
2. Calcular o valor de Goal que minimiza MAE e MSE usando a média e a mediana da distribuição. Em geral:
  - MAE: O valor que minimiza MAE é a mediana da variável resposta.
  - MSE: O valor que minimiza MSE é a média da variável resposta.

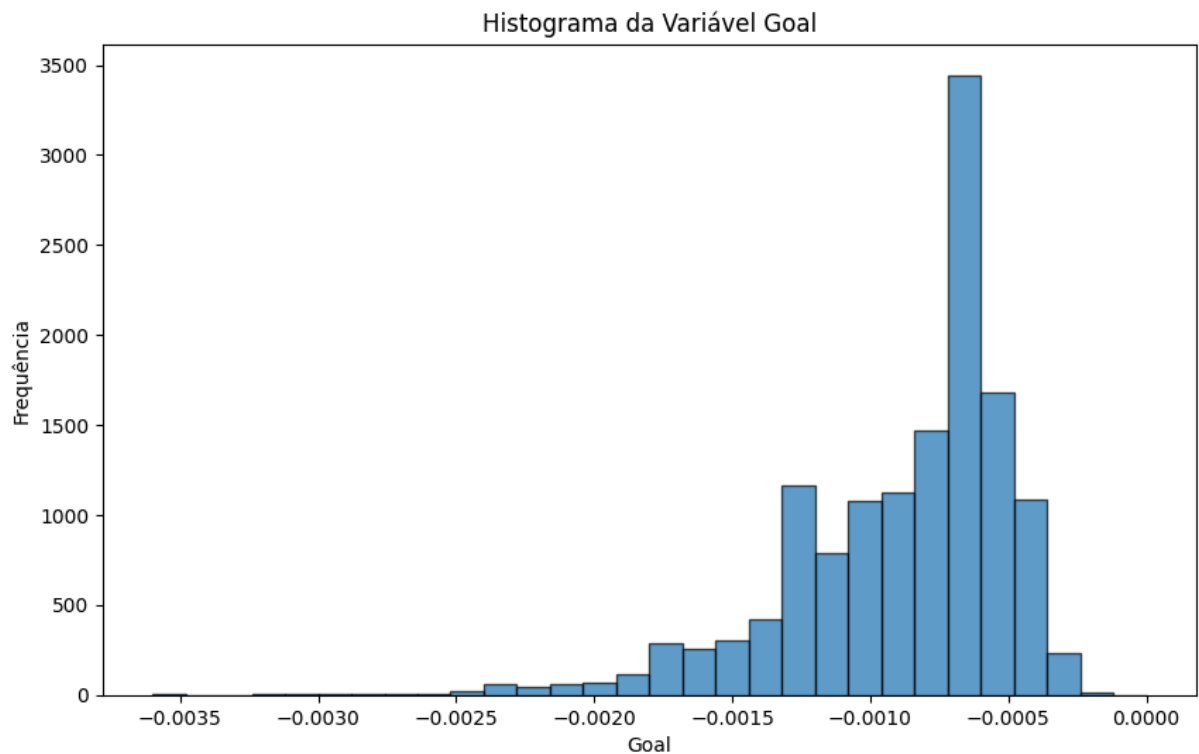
**Resposta:**

Valor de Goal que minimiza MAE: -0.0008

Valor de Goal que minimiza MSE: -0.0008718690909090909

MAE para a mediana: 0.0003034327272727273

MSE para a média: 1.6629083377190083e-07



#### vi. O modelo de Regressão Linear minimiza qual tipo de erro?

O modelo de regressão linear minimiza o erro quadrático médio (MSE, Mean Squared Error).

- **Objetivo:** O objetivo principal da regressão linear é encontrar a linha (ou plano, no caso de múltiplas variáveis) que melhor se ajusta aos dados. "Melhor se ajustar" é definido como a minimização da soma dos erros quadráticos.
- **Erro Quadrático:** O erro quadrático é a diferença entre o valor observado e o valor predito, elevada ao quadrado. Isso assegura que os erros maiores tenham um impacto maior na minimização total.
- **Função Objetivo:** Na regressão linear, a função objetivo é minimizar a soma dos quadrados dos erros (resíduos).

Ao minimizar o MSE, o modelo de regressão linear ajusta os parâmetros (coeficientes) para que a soma dos quadrados das diferenças entre os valores preditos e os valores reais seja a menor possível. Isso ajuda a garantir que o modelo faça previsões o mais próximo possível dos valores reais.



MAE (Mean Absolute Error): mede a média dos erros absolutos entre os valores preditos e os valores reais. É menos sensível a grandes discrepâncias do que o MSE, pois não eleva os erros ao quadrado.

Enquanto o MSE é a métrica que a regressão linear minimiza diretamente, o MAE pode ser usado como uma métrica alternativa para avaliar o desempenho do modelo, especialmente quando se deseja reduzir o impacto de valores extremos ou outliers.

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

**MAE (Mean Absolute Error)** é útil quando se deseja uma métrica que seja robusta a outliers e forneça uma interpretação direta do erro médio, pois é a média dos erros absolutos. É apropriado quando o impacto dos erros grandes não deve ser exagerado.

**MSE (Mean Squared Error)** é vantajoso quando se quer penalizar mais fortemente os erros grandes, já que os erros são elevados ao quadrado. Isso ajuda a identificar e mitigar grandes discrepâncias na predição, sendo útil quando se prioriza a redução dos grandes erros.

**RMSE (Root Mean Squared Error)** fornece uma métrica semelhante ao MSE, mas na mesma unidade da variável alvo, o que facilita a interpretação. É útil para problemas onde a magnitude dos erros deve ser considerada em unidades reais e é sensível a outliers, destacando a importância de reduzir grandes erros.

**viii. Elabore os seguintes modelos de regressão:**

Regressão Linear

```
Mean Absolute Error (MAE): 0.00
Mean Squared Error (MSE): 0.00
Root Mean Squared Error (RMSE): 0.00
R^2 Score: 0.83
```

Ridge

	Alpha	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	\
0	0.2	0.000130	3.047462e-08	
1	1.0	0.000130	3.070554e-08	
2	5.0	0.000131	3.103104e-08	
		Root Mean Squared Error (RMSE)	R^2 Score	
0		0.000175	0.825625	
1		0.000175	0.824304	
2		0.000176	0.822441	

## XGboost

	n_estimators	max_depth	learning_rate	Mean Absolute Error (MAE)	\
0	50	4	0.20	0.000124	
1	50	10	1.00	0.000140	
2	50	10	0.20	0.000122	
3	50	10	0.04	0.000134	
4	200	4	1.00	0.000133	
5	200	4	0.20	0.000124	
6	200	4	0.04	0.000123	
7	200	10	0.20	0.000122	
	Mean Squared Error (MSE)		Root Mean Squared Error (RMSE)		R^2 Score
0	2.842438e-08		0.000169		0.837356
1	3.624974e-08		0.000190		0.792580
2	2.764970e-08		0.000166		0.841789
3	3.306033e-08		0.000182		0.810830
4	3.377574e-08		0.000184		0.806736
5	2.842438e-08		0.000169		0.837356
6	2.786184e-08		0.000167		0.840575
7	2.764970e-08		0.000166		0.841789

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

Para determinar quais atributos devem ser descartados com base nos intervalos de confiança dos coeficientes da Regressão Linear, você pode usar a biblioteca statsmodels para ajustar o modelo e calcular os intervalos de confiança dos coeficientes. A ideia é identificar os atributos cujos intervalos de confiança contêm o valor zero, o que sugere que eles não têm um efeito significativo sobre a variável resposta Goal.

Intervalos de Confiança dos Coeficientes:				
	Lower Bound	Upper Bound	Coefficient	Includes Zero
const	3.263756e-04	3.573016e-04	3.418386e-04	False
ClimbRate	1.254342e-07	3.668911e-07	2.461626e-07	False
Sgz	8.252274e-08	3.474094e-07	2.149660e-07	False
P	-4.362747e-04	-4.161312e-04	-4.262029e-04	False
Q	-1.171325e-04	1.194501e-04	1.158787e-06	True
CurPitch	1.466709e-04	4.003732e-04	2.735220e-04	False
CurRoll	-9.052656e-05	-8.228571e-05	-8.640614e-05	False
AbsRoll	5.631329e-05	5.924649e-05	5.777989e-05	False
DiffClb	-4.575397e-06	-2.546736e-06	-3.561067e-06	False
DiffRollRate	2.216136e-03	2.857708e-03	2.536922e-03	False
DiffDiffClb	-6.102582e-06	4.801155e-06	-6.507135e-07	True
DiffSeTime1	-1.006182e-02	-9.272423e-04	-5.494530e-03	False
DiffSeTime3	-6.599556e-03	7.056551e-03	2.284977e-04	True
DiffSeTime5	-1.957210e-02	-6.539687e-03	-1.305589e-02	False
DiffSeTime7	-2.144968e-02	-9.925433e-03	-1.568756e-02	False
DiffSeTime9	-2.162520e-02	-1.113260e-02	-1.637890e-02	False
DiffSeTime11	-1.486290e-02	-4.707473e-03	-9.785189e-03	False
DiffSeTime13	-1.467319e-02	-4.342578e-03	-9.507886e-03	False
Alpha	-2.090311e-04	-1.395975e-05	-1.114954e-04	False
Se	-3.489927e-02	-2.624738e-02	-3.057332e-02	False

### Explicação

- **Preparação dos Dados:** Carrega e filtra o dataset, definindo X (variáveis independentes) e y (variável dependente).
- **Ajuste do Modelo:** Ajusta o modelo de Regressão Linear usando statsmodels e adiciona uma constante para o intercepto.
- **Intervalos de Confiança:** Calcula os intervalos de confiança dos coeficientes e verifica quais intervalos incluem zero. Identifica os atributos cujos coeficientes não são estatisticamente significativos (ou seja, seus intervalos de confiança incluem zero).

Os atributos cujos intervalos de confiança incluem zero são considerados menos significativos e podem ser descartados, pois isso sugere que não há evidências estatísticas suficientes para afirmar que eles têm um efeito significativo sobre a variável Goal.

### x. Compare os resultados da Regressão Linear e sua versão regularizada.

Comparar os resultados de um modelo de Regressão Linear com sua versão regularizada, como a Regressão Ridge, pode fornecer insights sobre a eficácia da

regularização em melhorar a performance do modelo e lidar com problemas como a multicolinearidade.

```
Resultados da Regressão Linear:
Mean Absolute Error (MAE): 0.00
Mean Squared Error (MSE): 0.00
Root Mean Squared Error (RMSE): 0.00
R^2 Score: 0.83

Resultados da Regressão Ridge:
Alpha Mean Absolute Error (MAE) Mean Squared Error (MSE) \
0 0.2 0.000130 3.047462e-08
1 1.0 0.000130 3.070554e-08
2 5.0 0.000131 3.103104e-08

Root Mean Squared Error (RMSE) R^2 Score
0 0.000175 0.825625
1 0.000175 0.824304
2 0.000176 0.822441

Intervalos de Confiança dos Coeficientes da Regressão Linear:
Lower Bound Upper Bound Coefficient Includes Zero
const 3.263756e-04 3.573016e-04 3.418386e-04 False
ClimbRate 1.254342e-07 3.668911e-07 2.461626e-07 False
Sgz 8.252274e-08 3.474094e-07 2.149660e-07 False
P -4.362747e-04 -4.161312e-04 -4.262029e-04 False
Q -1.171325e-04 1.194501e-04 1.158787e-06 True
CurPitch 1.466709e-04 4.003732e-04 2.735220e-04 False
CurRoll -9.052656e-05 -8.228571e-05 -8.640614e-05 False
AbsRoll 5.631329e-05 5.924649e-05 5.777989e-05 False
DiffClb -4.575397e-06 -2.546736e-06 -3.561067e-06 False
DiffRollRate 2.216136e-03 2.857708e-03 2.536922e-03 False
DiffDiffClb -6.102582e-06 4.801155e-06 -6.507135e-07 True
DiffSeTime1 -1.006182e-02 -9.272423e-04 -5.494530e-03 False
DiffSeTime3 -6.599556e-03 7.056551e-03 2.284977e-04 True
DiffSeTime5 -1.957210e-02 -6.539687e-03 -1.305589e-02 False
DiffSeTime7 -2.144968e-02 -9.925433e-03 -1.568756e-02 False
DiffSeTime9 -2.162520e-02 -1.113260e-02 -1.637890e-02 False
DiffSeTime11 -1.486290e-02 -4.707473e-03 -9.785189e-03 False
DiffSeTime13 -1.467319e-02 -4.342578e-03 -9.507886e-03 False
Alpha -2.090311e-04 -1.395975e-05 -1.114954e-04 False
Se -3.489927e-02 -2.624738e-02 -3.057332e-02 False
```

```
Coeficientes da Regressão Ridge (alpha=0.2):
ClimbRate      -3.862921e-07
Sgz            4.376618e-08
P              -4.378470e-04
Q              -4.092699e-04
CurPitch       -3.637496e-04
CurRoll        -9.261589e-05
AbsRoll         5.487947e-05
DiffClb         -7.081293e-06
DiffRollRate    1.941245e-03
DiffDiffClb     -1.328021e-05
DiffSeTime1     -9.483832e-05
DiffSeTime3     -5.201512e-05
DiffSeTime5     -1.774831e-04
DiffSeTime7     -3.147025e-04
DiffSeTime9     -3.588432e-04
DiffSeTime11    -2.773567e-04
DiffSeTime13    -2.290582e-04
Alpha           -4.849758e-04
Se              -8.571719e-04
dtype: float64
```

```
Coeficientes da Regressão Ridge (alpha=1):
ClimbRate      -4.041951e-07
Sgz            6.483230e-08
P              -4.328071e-04
Q              -3.981402e-04
CurPitch       -3.843417e-04
CurRoll        -9.935906e-05
AbsRoll         5.493978e-05
DiffClb         -6.709091e-06
DiffRollRate    1.131417e-03
DiffDiffClb     -1.241504e-05
DiffSeTime1     -2.001586e-05
DiffSeTime3     -1.107828e-05
DiffSeTime5     -3.421318e-05
DiffSeTime7     -6.188637e-05
DiffSeTime9     -7.193201e-05
DiffSeTime11    -5.570512e-05
DiffSeTime13    -4.713507e-05
Alpha           -4.868285e-04
Se              -1.922636e-04
dtype: float64
```

```
Coeficientes da Regressão Ridge (alpha=5):
ClimbRate      -4.193115e-07
Sgz            7.587169e-08
P              -4.262648e-04
Q              -3.014920e-04
CurPitch       -4.019166e-04
CurRoll        -1.057541e-04
AbsRoll         5.603657e-05
DiffClb         -5.602678e-06
DiffRollRate    3.674984e-04
DiffDiffClb     -7.997645e-06
DiffSeTime1     -3.859531e-06
DiffSeTime3     -2.185262e-06
DiffSeTime5     -6.458294e-06
DiffSeTime7     -1.194766e-05
DiffSeTime9     -1.402869e-05
DiffSeTime11    -1.100530e-05
DiffSeTime13    -9.834364e-06
Alpha           -4.722625e-04
Se              -5.391409e-05
dtype: float64
```

## **Explicação**

- Preparação dos Dados: Carrega e filtra o dataset, dividindo-o em conjuntos de treinamento e teste.
- Construção dos Modelos: Ajusta um modelo de Regressão Linear e calcula as métricas de desempenho (MAE, MSE, RMSE,  $R^2$ ). Além disso, ajusta modelos de Regressão Ridge com diferentes valores de alpha e calcula as métricas de desempenho para cada um.
- Comparação dos Resultados: Exibe as métricas de desempenho para o modelo de Regressão Linear e para cada modelo de Regressão Ridge. Os coeficientes e intervalos de confiança da Regressão Linear são apresentados, e compara os coeficientes da Regressão Ridge para diferentes valores de alpha.

## **Interpretação dos Resultados**

- Regressão Linear: Pode ser mais suscetível a problemas como multicolinearidade e overfitting, especialmente se o número de variáveis é grande em relação ao número de amostras.
- Regressão Ridge: A regularização pode reduzir a variância dos coeficientes, melhorando a generalização do modelo e mitigando problemas de multicolinearidade. Comparar as métricas de desempenho pode mostrar se a regularização ajuda a melhorar a precisão do modelo.
- Coeficientes: Os coeficientes da Regressão Ridge serão geralmente menores em magnitude comparados aos da Regressão Linear, e a magnitude dos coeficientes pode variar com diferentes valores de alpha. Isso reflete o efeito da regularização no modelo.

Com o resultado do código, é possível perceber que os valores das duas regressões são parecidos, e que apenas o  $R^2$  da regressão linear é melhor, o que indica um melhor desempenho nesse conjunto de dados.

## **xi. Compare o efeito dos hiperparâmetros do XGBoost e seus resultados.**

Para comparar o efeito dos hiperparâmetros do XGBoost e seus resultados, é necessário ajustar o modelo com diferentes combinações de hiperparâmetros e avaliar o desempenho de cada configuração. Os principais hiperparâmetros a serem analisados são:

- `n_estimators`: Número de árvores na floresta.
- `max_depth`: Profundidade máxima de cada árvore.
- `learning_rate`: Taxa de aprendizado (ou taxa de atualização dos pesos).

#### Passos para Comparação

- Preparar os Dados: Carregar e dividir o dataset.
- Construir e Avaliar os Modelos: Ajustar o modelo XGBoost com diferentes combinações de `n_estimators`, `max_depth`, e `learning_rate`.
- Comparar os Resultados: Avaliar e comparar o desempenho de cada combinação.

	<code>n_estimators</code>	<code>max_depth</code>	<code>learning_rate</code>	Mean Absolute Error (MAE)	\
0	50	4	0.20	0.000124	
1	50	10	1.00	0.000140	
2	50	10	0.20	0.000122	
3	50	10	0.04	0.000134	
4	200	4	1.00	0.000133	
5	200	4	0.20	0.000124	
6	200	4	0.04	0.000123	
7	200	10	0.20	0.000122	
	Mean Squared Error (MSE)		Root Mean Squared Error (RMSE)		R^2 Score
0	2.842438e-08		0.000169		0.837356
1	3.624974e-08		0.000190		0.792580
2	2.764970e-08		0.000166		0.841789
3	3.306033e-08		0.000182		0.810830
4	3.377574e-08		0.000184		0.806736
5	2.842438e-08		0.000169		0.837356
6	2.786184e-08		0.000167		0.840575
7	2.764970e-08		0.000166		0.841789

#### Interpretação dos Resultados

- `n_estimators`: O número de árvores no modelo. Um valor maior pode melhorar a performance, mas também pode aumentar o tempo de treinamento e risco de overfitting.
- `max_depth`: A profundidade máxima de cada árvore. Árvores mais profundas podem modelar relações mais complexas, mas também podem causar overfitting.
- `learning_rate`: A taxa de aprendizado controla a contribuição de cada árvore para a predição final. Valores menores tornam o treinamento mais robusto, mas requerem mais árvores para convergir.

Nesse sentido, com base nos resultados apresentados, é possível perceber que a taxa de aprendizado e “max\_depth” contribuem positivamente para o resultado.

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

Para comparar os diferentes modelos de regressão na predição da deflexão dos ailerons, você pode usar métricas de avaliação como MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error) e  $R^2$  (coeficiente de determinação).

```
Resultados da Regressão Linear:
Mean Absolute Error (MAE): 0.00
Mean Squared Error (MSE): 0.00
Root Mean Squared Error (RMSE): 0.00
R^2 Score: 0.83

Resultados da Regressão Ridge:
Alpha Mean Absolute Error (MAE) Mean Squared Error (MSE) \
0 0.2 0.000130 3.047462e-08
1 1.0 0.000130 3.070554e-08
2 5.0 0.000131 3.103104e-08

Root Mean Squared Error (RMSE) R^2 Score
0 0.000175 0.825625
1 0.000175 0.824304
2 0.000176 0.822441

Resultados do XGBoost:
n_estimators max_depth learning_rate Mean Absolute Error (MAE) \
0 50 4 0.20 0.000124
1 50 10 1.00 0.000140
2 50 10 0.20 0.000122
3 50 10 0.04 0.000134
4 200 4 1.00 0.000133
5 200 4 0.20 0.000124
6 200 4 0.04 0.000123
7 200 10 0.20 0.000122

Mean Squared Error (MSE) Root Mean Squared Error (RMSE) R^2 Score
0 2.842438e-08 0.000169 0.837356
1 3.624974e-08 0.000190 0.792580
2 2.764970e-08 0.000166 0.841789
3 3.306033e-08 0.000182 0.810830
4 3.377574e-08 0.000184 0.806736
5 2.842438e-08 0.000169 0.837356
6 2.786184e-08 0.000167 0.840575
7 2.764970e-08 0.000166 0.841789
```

**Interpretação dos Resultados**

1. Regressão Linear: Oferece uma linha base para comparação com modelos mais complexos. O desempenho é medido por MAE, MSE, RMSE e  $R^2$ .



2. Regressão Ridge: A regularização ajuda a controlar overfitting e multicolinearidade.
3. XGBoost: Um modelo baseado em árvores que pode capturar interações complexas entre variáveis.

Comparar essas métricas para os diferentes modelos ajudará a entender qual modelo oferece a melhor performance na predição da deflexão dos ailerons e qual configuração de hiperparâmetros otimiza o desempenho. Aparentemente, todos eles demonstraram resultados similares.

## 2.2) Aprendizado Supervisionado - Classificação

### i. Elabore os seguintes modelos de classificação:

Regressão Logística

```

Acurácia: 0.67
Matriz de Confusão:
[[ 0  0  1  17  0  0]
 [ 0  0  0  52  0  0]
 [ 0  0  0 198  2  2]
 [ 0  0  0 417 272 10]
 [ 0  0  0  70 1384 59]
 [ 0  0  0  1  225 40]]
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	0.55	0.60	0.57	699
4.0	0.73	0.91	0.82	1513
5.0	0.36	0.15	0.21	266
accuracy			0.67	2750
macro avg	0.27	0.28	0.27	2750
weighted avg	0.58	0.67	0.61	2750

Naive Bayes

```

Confusion Matrix:
[[2750]]
Classification Report:

```

	precision	recall	f1-score	support
5.0	1.00	1.00	1.00	2750
accuracy			1.00	2750
macro avg	1.00	1.00	1.00	2750
weighted avg	1.00	1.00	1.00	2750

## Árvores de Decisão

Resultados para max\_depth=7:

Confusion Matrix:

```
[[ 3  8  7  0  0  0]
 [ 2 11 35  4  0  0]
 [ 0  9 115 78  0  0]
 [ 0  0  63 448 188  0]
 [ 0  0  2 152 1268 91]
 [ 0  0  0  1 147 118]]
```

Classification Report:

	precision	recall	f1-score	support
0.0	0.60	0.17	0.26	18
1.0	0.39	0.21	0.28	52
2.0	0.52	0.57	0.54	202
3.0	0.66	0.64	0.65	699
4.0	0.79	0.84	0.81	1513
5.0	0.56	0.44	0.50	266
accuracy			0.71	2750
macro avg	0.59	0.48	0.51	2750
weighted avg	0.71	0.71	0.71	2750

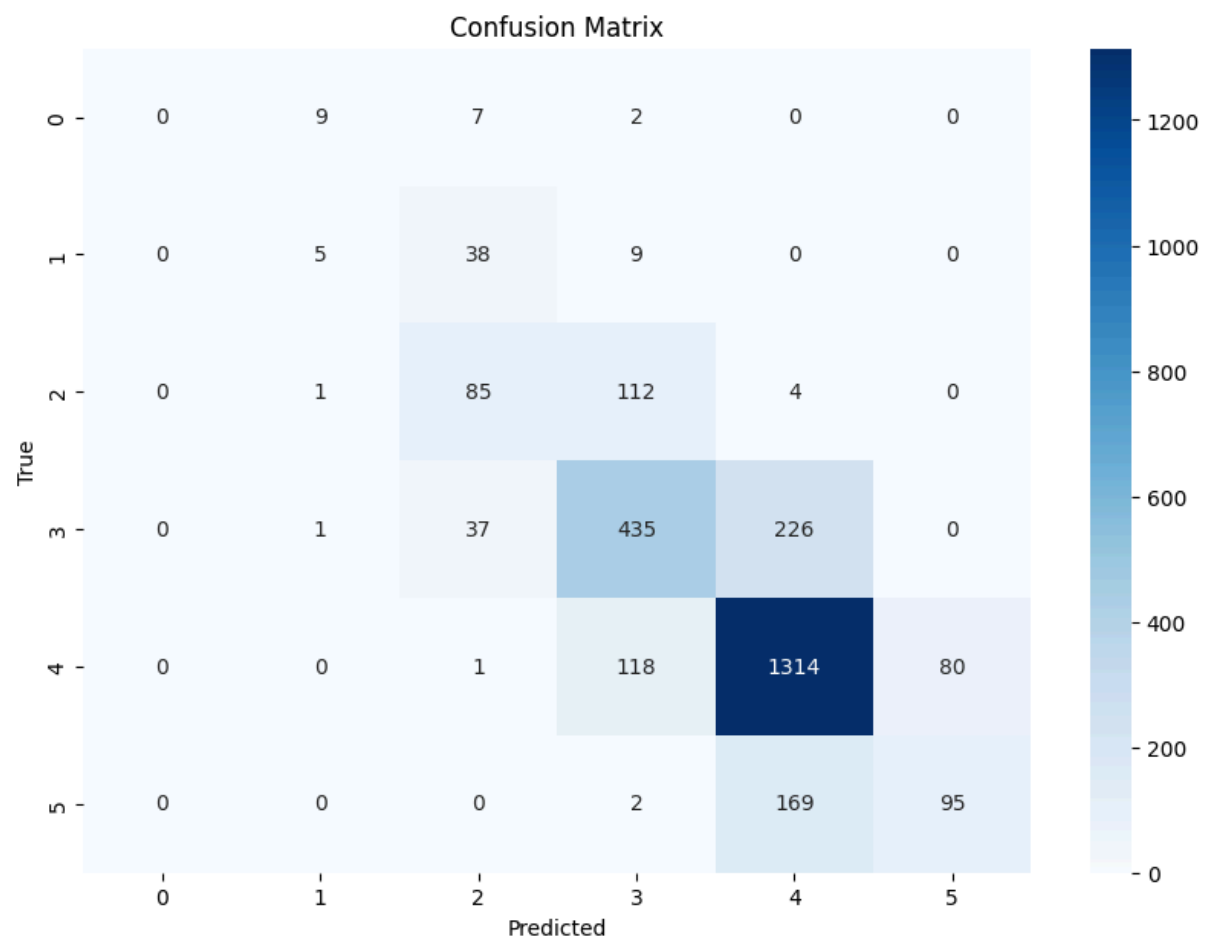
Resultados para max\_depth=10:

Confusion Matrix:

```
[[ 6  8  4  0  0  0]
 [ 1 13 32  6  0  0]
 [ 1 18 103 76  4  0]
 [ 1  2  70 448 176  2]
 [ 0  0  5 152 1279 77]
 [ 0  0  0  3 147 116]]
```

Classification Report:

	precision	recall	f1-score	support
0.0	0.67	0.33	0.44	18
1.0	0.32	0.25	0.28	52
2.0	0.48	0.51	0.50	202
3.0	0.65	0.64	0.65	699
4.0	0.80	0.85	0.82	1513
5.0	0.59	0.44	0.50	266
accuracy			0.71	2750
macro avg	0.59	0.50	0.53	2750
weighted avg	0.71	0.71	0.71	2750



K-NN

n_neighbors	Accuracy	Classification Report
0	3 0.486909	{'0.0': {'precision': 0.0, 'recall': 0.0, 'f1-...
1	5 0.539273	{'0.0': {'precision': 0.3333333333333333, 'rec...
2	7 0.555636	{'0.0': {'precision': 0.0, 'recall': 0.0, 'f1-...

XGBoost

```

n_estimators: 200, max_depth: 4, learning_rate: 0.04
Accuracy: 0.7484
Confusion Matrix:
[[ 4  4  4  0  0  0]
 [ 3  6 36  2  0  0]
 [ 0  9 96 86  2  0]
 [ 0  1 34 465 189  0]
 [ 0  0  4 127 1374 38]
 [ 0  0  0  1 152 113]]
Classification Report:
              precision    recall  f1-score   support

0.0          0.57      0.33      0.42         12
1.0          0.30      0.13      0.18         47
2.0          0.55      0.50      0.52        193
3.0          0.68      0.67      0.68        689
4.0          0.80      0.89      0.84       1543
5.0          0.75      0.42      0.54        266

accuracy          0.75       2750
macro avg         0.61       0.49      0.53       2750
weighted avg      0.74       0.75      0.74       2750

n_estimators: 200, max_depth: 10, learning_rate: 0.2
Accuracy: 0.7585
Confusion Matrix:
[[ 3  7  2  0  0  0]
 [ 2 15 28  2  0  0]
 [ 1 10 95 84  3  0]
 [ 0  2 42 481 164  0]
 [ 0  0  5 132 1358 48]
 [ 0  0  0  1 131 134]]
Classification Report:
              precision    recall  f1-score   support

0.0          0.50      0.25      0.33         12
1.0          0.44      0.32      0.37         47
2.0          0.55      0.49      0.52        193
3.0          0.69      0.70      0.69        689
4.0          0.82      0.88      0.85       1543
5.0          0.74      0.50      0.60        266

accuracy          0.76       2750
macro avg         0.62       0.52      0.56       2750
weighted avg      0.75       0.76      0.75       2750

```

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.

	ClimbRate	Sgz	P	Q	CurPitch	CurRoll	AbsRoll	\
0.0	1.000249	0.987944	1.237644	1.140306	1.001416	0.821453	1.101692	
1.0	1.000466	0.990467	1.853112	1.407866	1.371957	1.206045	0.902386	
2.0	1.000657	1.003677	6.133259	1.829111	1.431983	1.742856	0.683682	
3.0	1.000487	1.002673	5.886531	0.894140	1.197605	1.580015	0.764594	
4.0	0.998585	1.006645	0.462547	0.505365	0.178644	0.908514	1.066860	
5.0	0.999558	1.008784	0.026109	0.753644	2.375778	0.403459	1.803662	
	DiffClb	DiffRollRate	DiffDiffClb	DiffSeTime1	DiffSeTime3	\		
0.0	1.042652	1.005852	0.908811	1.000039	1.000134			
1.0	0.960191	0.996318	1.035605	1.000087	1.000182			
2.0	0.999609	0.992857	0.913794	1.000222	1.000238			
3.0	0.989117	1.003973	0.985183	1.002146	0.999961			
4.0	0.978335	1.000015	1.025417	1.000668	1.001045			
5.0	1.032613	1.001043	1.150978	0.996845	0.998443			
	DiffSeTime5	DiffSeTime7	DiffSeTime9	DiffSeTime11	DiffSeTime13	\		
0.0	1.000085	1.000019	0.999984	1.000116	1.000065			
1.0	0.999892	0.999934	1.000083	1.000270	1.000655			
2.0	1.000318	1.001289	1.000989	1.000332	1.000129			
3.0	1.001016	1.000870	1.000942	1.001349	1.001225			
4.0	0.999449	0.999682	0.999934	0.999056	0.999200			
5.0	0.999242	0.998209	0.998071	0.998878	0.998728			
	Alpha	Se						
0.0	1.042200	1.001958						
1.0	1.428113	1.014477						
2.0	1.564577	1.017743						
3.0	1.133456	1.003304						
4.0	0.157231	0.934688						
5.0	2.409616	1.030792						
	ClimbRate	P						
0.0	1.000249	1.237644						
1.0	1.000466	1.853112						
2.0	1.000657	6.133259						
3.0	1.000487	5.886531						
4.0	0.998585	0.462547						
5.0	0.999558	0.026109						

## Interpretação dos Resultados

### Odds-Ratio

- Objetivo: Indica a multiplicação dos odds de uma classe em comparação com a classe base para uma unidade de aumento no atributo.
- Coeficientes Positivos: Um coeficiente positivo indica que, à medida que o valor do atributo aumenta, a probabilidade da classe aumenta em comparação com a classe base.
- Coeficientes Negativos: Um coeficiente negativo indica que, à medida que o valor do atributo aumenta, a probabilidade da classe diminui em comparação com a classe base.

## Análise

**ClimbRate:** Para o atributo ClimbRate, os odds-ratios são muito próximos de 1 para todas as classes, indicando que este atributo tem um impacto muito pequeno na probabilidade das diferentes classes. Isso sugere que mudanças na taxa de subida não afetam significativamente a classificação dos ailerons.

**P (Taxa de rotação):** Para o atributo P, os odds-ratios variam muito mais entre as classes:

- **Classes 0.0 a 3.0:** O aumento na taxa de rotação está associado a um aumento significativo nas probabilidades dessas classes. Em particular, para as classes 2.0 e 3.0, os odds-ratios são bastante elevados (6.133259 e 5.886531), indicando uma forte influência da taxa de rotação nessas classificações.
- **Classes 4.0 e 5.0:** O aumento na taxa de rotação está associado a uma diminuição significativa nas probabilidades dessas classes, com odds-ratios muito baixos (0.462547 e 0.026109). Isso sugere que altas taxas de rotação tornam essas classes muito menos prováveis.

## Conclusão

- **Significância dos Atributos:** O atributo ClimbRate tem um impacto muito pequeno na classificação, com odds-ratios próximos de 1. Por outro lado, o atributo P tem um impacto significativo, com odds-ratios variando amplamente entre as classes, mostrando uma forte influência na probabilidade de certas classificações.
- **Contexto de Aplicação:** Em um contexto prático, a taxa de rotação (P) deve ser cuidadosamente monitorada e controlada para evitar classes indesejadas, enquanto a taxa de subida (ClimbRate) pode ser menos crítica para a classificação dos ailerons.

## Observações

Nem todos os atributos terão um impacto significativo em todas as classes. O valor do odds-ratio próximo de 1 indica que o atributo tem pouco ou nenhum efeito na probabilidade da classe. Em aplicações práticas, a interpretação dos odds-ratios deve

considerar o contexto operacional e a importância relativa dos atributos para a segurança e conforto do voo.

Este método fornece uma análise detalhada dos efeitos dos atributos selecionados nas probabilidades de classificação das diferentes classes, ajudando a entender melhor os fatores que influenciam a deflexão dos ailerons.

### **iii. Qual a suposição utilizada que dá o nome Naive no modelo de Naive Bayes?**

A suposição que dá o nome "Naive" ao modelo de Naive Bayes é a independência condicional. Especificamente, o modelo assume que todos os atributos (ou características) de entrada são mutuamente independentes, dados a classe.

#### **Suposição de Independência Condicional**

- Independência Condicional: Naive Bayes assume que a presença (ou ausência) de uma determinada característica de uma classe não está relacionada com a presença (ou ausência) de qualquer outra característica. Isto significa que, para calcular a probabilidade de uma amostra pertencer a uma determinada classe, o modelo considera cada atributo de forma independente.

#### **Porque essa Suposição é "Ingênua"**

Na prática, os atributos de um problema real muitas vezes não são completamente independentes. Por exemplo, em um problema de diagnóstico médico, os sintomas de uma doença podem estar correlacionados entre si. Apesar da suposição ser frequentemente irrealista, ela simplifica significativamente os cálculos e a implementação do modelo, tornando o Naive Bayes computacionalmente eficiente e fácil de interpretar.

Curiosamente, mesmo com a suposição de independência, o modelo Naive Bayes frequentemente proporciona bons resultados na prática, especialmente em contextos de classificação de texto e sistemas de recomendação.

#### **Conclusão**

A suposição de independência condicional é a base do "naive" no Naive Bayes. Embora esta suposição seja frequentemente violada em problemas do mundo real, a simplicidade do modelo e sua capacidade de fornecer boas previsões em muitos casos

práticos justificam seu uso contínuo e sua popularidade em diversas aplicações de aprendizado de máquina.

**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.**

**Resposta:** No contexto do modelo Naive Bayes, o Teorema de Bayes é ajustado para aproveitar a suposição de independência condicional dos atributos e lidar com problemas de computação, como o underflow. Aqui estão os passos para reescrever o Teorema de Bayes com essas considerações:

- **Teorema de Bayes**

O Teorema de Bayes pode ser expresso como:

$$P(A | B) = \frac{P(B | A) \cdot P(A)}{P(B)}$$

1.  $P(A|B)$  é a probabilidade posterior da classe A dado o vetor de atributos B.
2.  $P(B|A)$  é a probabilidade condicional dos atributos B dado a classe A.
3.  $P(A)$  é a probabilidade a priori da classe A.
4.  $P(B)$  é a probabilidade marginal dos atributos B.

- **Ajuste para o Modelo Naive Bayes**

5. Supressão da Constante de Normalização: A constante  $P(B)$  não é dependente da classe A e não afeta a comparação entre diferentes classes. Assim, no contexto de classificação, podemos desprezar  $P(B)$  e focar na parte proporcional.
6. Produto de Probabilidades e Logaritmo: Para evitar problemas de underflow quando se trabalha com produtos de probabilidades pequenas, aplicamos o logaritmo. A função logarítmica transforma o produto de probabilidades em uma soma de logaritmos.



- **Independência Condicional**

Com a suposição de independência condicional, a probabilidade condicional  $P(B|A)$  é decomposta como um produto das probabilidades de cada atributo dado a classe.

- **Fórmula Final**

Incorporando todas essas considerações, o logaritmo da probabilidade posterior  $P(A|B)$  é:

$$\log P(A|B) \propto \sum \log P(B_i|A) + \log P(A) \quad \log P(A|B) \propto \sum \log P(B_i|A) + \log P(A)$$

Finalmente, para a classificação, escolhemos a classe  $A$  que maximiza a probabilidade posterior  $P(A|B)$ . Em termos de logaritmos, isso equivale a maximizar:

$$\log P(A|B) = \sum \log P(B_i|A) + \log P(A) \quad \log P(A|B) = \sum \log P(B_i|A) + \log P(A)$$

- **Conclusão**

Este ajuste ao Teorema de Bayes para o modelo Naive Bayes, com a exclusão da constante de normalização e a aplicação do logaritmo, facilita o cálculo e evita problemas de underflow, além de manter a abordagem baseada na proporcionalidade das probabilidades.

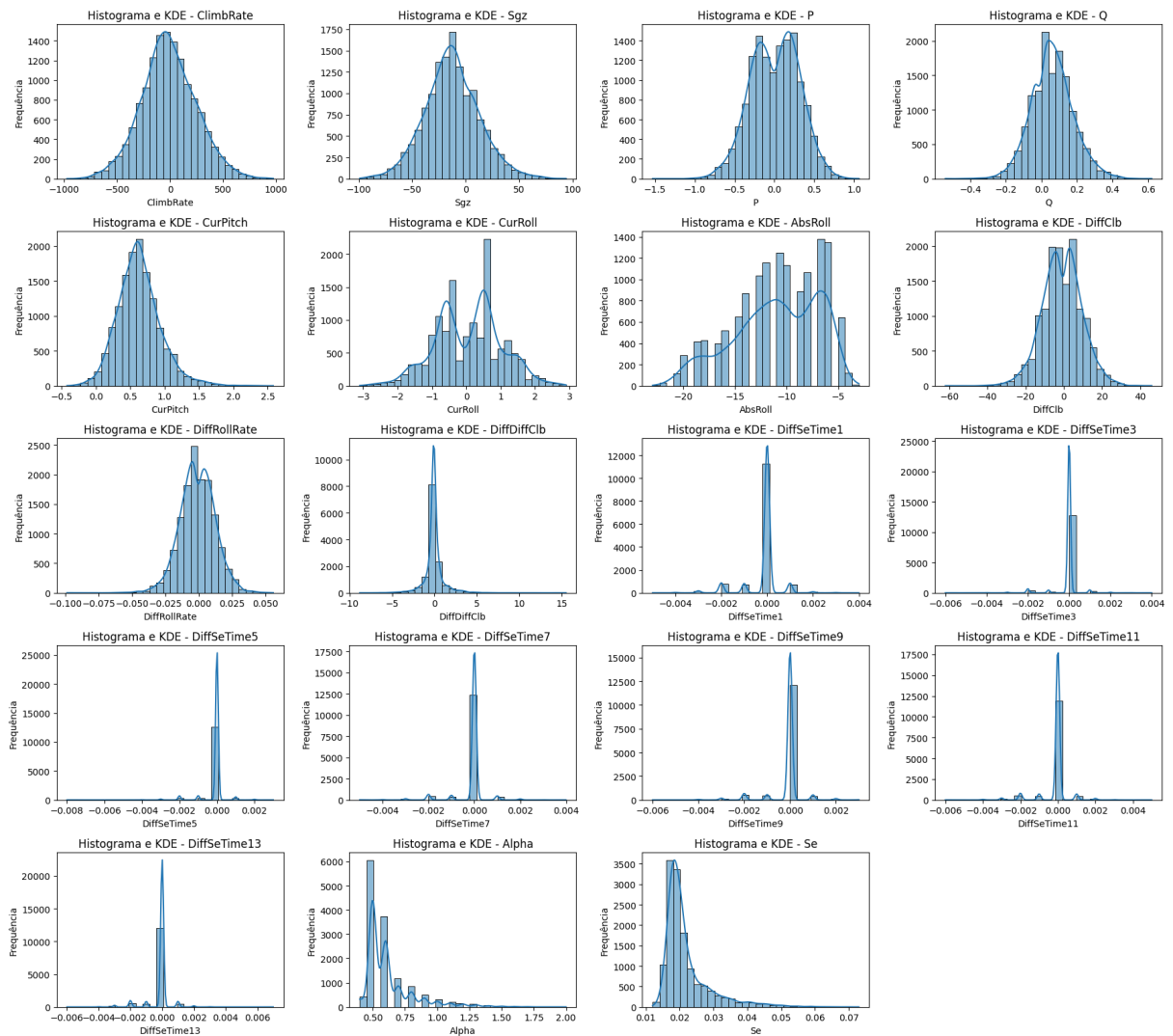
#### **v. Para este conjunto de dados a suposição gaussiana é razoável?**

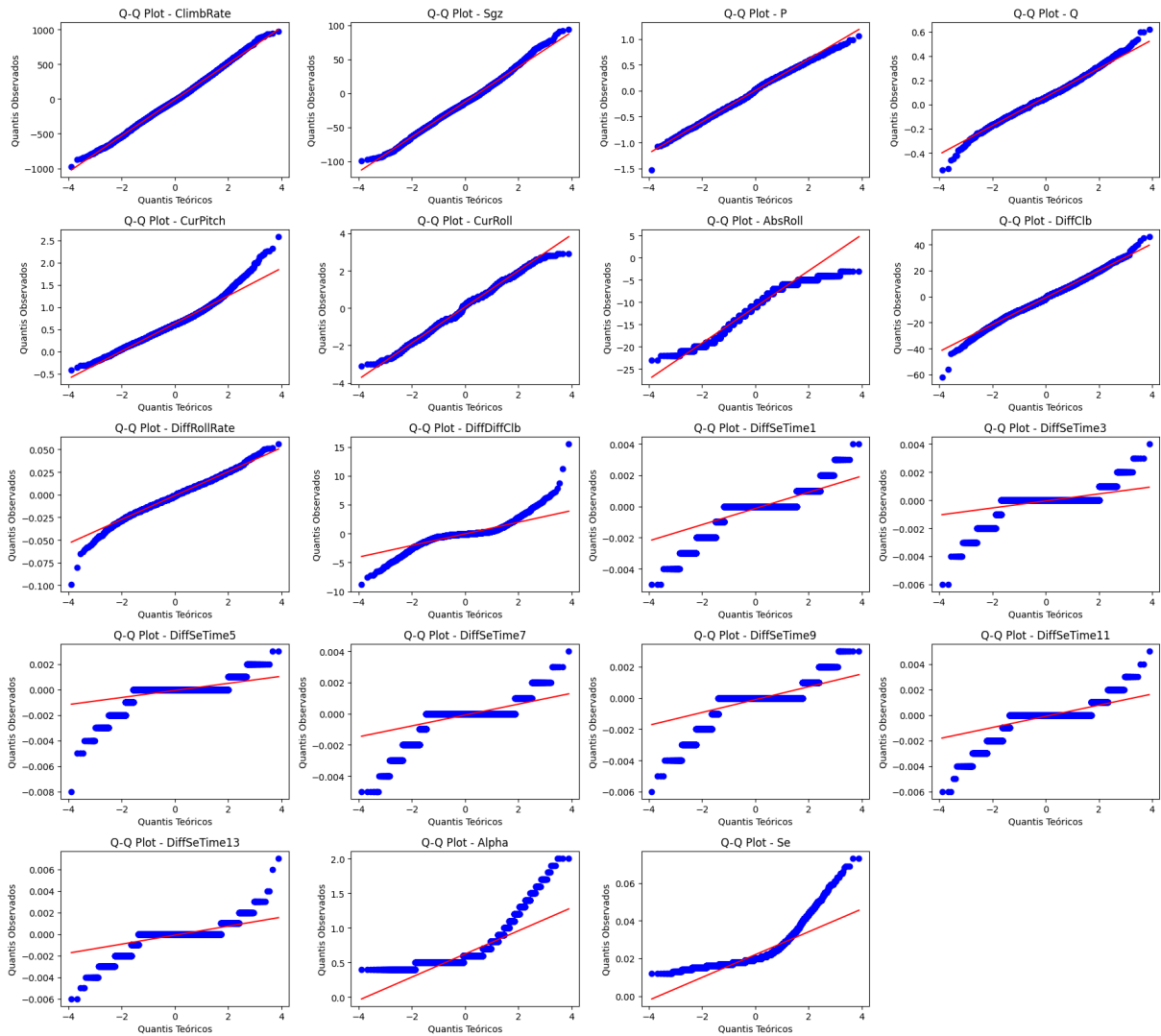
Para determinar se a suposição gaussiana é razoável para um conjunto de dados, precisamos avaliar se as variáveis seguem uma distribuição aproximadamente normal. No contexto de modelos de Naive Bayes, a suposição gaussiana se refere ao Naive Bayes Gaussiano, que assume que as características contínuas seguem uma distribuição normal (gaussiana) condicionada à classe.

- **Passos para Avaliar a Suposição Gaussiana**

1. Visualização dos Dados: Plotar histogramas e gráficos de densidade para as variáveis contínuas e verificar visualmente se se aproximam de uma distribuição normal.

2. Testes Estatísticos: Aplicar testes estatísticos para verificar a normalidade das variáveis. Testes comuns incluem o teste de Shapiro-Wilk e o teste de Kolmogorov-Smirnov.
3. Q-Q Plots: Utilizar gráficos quantil-quantil (Q-Q plots) para comparar a distribuição dos dados com uma distribuição normal.





De acordo com os resultados obtidos, visualmente é possível afirmar que alguns parâmetros dentro desse conjunto, a suposição de Gauss é razoável.

**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?**

No contexto de modelos de Árvores de Decisão, escolher o melhor hiperparâmetro para generalização e para minimizar o erro empírico envolve diferentes aspectos de ajuste e avaliação.

### 1. Hiperparâmetro para Generalização

Hiperparâmetro X: max\_depth

`max_depth` define a profundidade máxima da árvore de decisão. Limitar a profundidade ajuda a evitar o sobreajuste (overfitting) ao controlar a complexidade do modelo. O objetivo é encontrar a profundidade que maximiza a capacidade do modelo de generalizar para novos dados sem se ajustar excessivamente aos dados de treinamento.

É necessário realizar uma busca por validação cruzada para diferentes profundidades e escolher aquela que apresenta o melhor desempenho em dados de validação. Profundidades muito baixas podem subajustar o modelo (underfitting), enquanto profundidades muito altas podem levar ao sobreajuste.

## 2. **Hiperparâmetro para Minimizar o Erro Empírico**

Hiperparâmetro Y: `min_samples_split` ou `min_samples_leaf`

`min_samples_split` é o número mínimo de amostras necessárias para dividir um nó interno. Menores valores permitem divisões mais finas e detalhadas, enquanto valores maiores resultam em divisões mais gerais.

`min_samples_leaf` é o número mínimo de amostras que deve estar presente em um nó folha. Controla a profundidade da árvore e evita folhas com poucas amostras, ajudando a evitar sobreajuste.

O objetivo é minimizar o erro empírico, que é o erro calculado no conjunto de treinamento. Ajustar esses hiperparâmetros pode ajudar a modelar os dados mais detalhadamente, reduzindo o erro empírico. Para escolher, é necessário usar uma busca por validação cruzada para testar diferentes valores dos hiperparâmetros e escolher aqueles que minimizam o erro empírico.

## 3. **Comparação dos Modelos**

Modelo que Maximiza a Generalização: Focado em encontrar o melhor `max_depth` para evitar o sobreajuste e garantir que o modelo se generalize bem para novos dados. Esse modelo geralmente terá melhor desempenho em um conjunto de validação ou teste, indicando que foi bem ajustado para não se sobreajustar aos dados de treinamento.

Modelo que Minimiza o Erro Empírico: Focado em encontrar o melhor `min_samples_split` ou `min_samples_leaf` para minimizar o erro nos dados de treinamento. Esse modelo pode ter um desempenho muito bom no conjunto de treinamento, mas pode não se generalizar bem para novos dados se for muito específico.

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

	n_neighbors	Test Accuracy	Train Accuracy	Test F1 Score	Train F1 Score
0	3	0.486909	0.726636	0.470289	0.717733
1	5	0.539273	0.677273	0.496927	0.645555
2	7	0.555636	0.656091	0.499917	0.611494
3	9	0.557455	0.647636	0.490435	0.592399
4	11	0.564727	0.635727	0.488036	0.570387
5	13	0.567273	0.626273	0.483198	0.553215
6	15	0.564727	0.619364	0.475725	0.541328
7	17	0.568364	0.612273	0.472961	0.528680
8	19	0.567636	0.607182	0.466246	0.518355
9	21	0.564364	0.607364	0.460779	0.513914
10	23	0.568000	0.603182	0.460922	0.506493
11	25	0.568364	0.599818	0.456471	0.498967
12	27	0.562909	0.596364	0.446202	0.491700
13	29	0.565818	0.592727	0.446702	0.484692
14	31	0.564727	0.590909	0.441555	0.479280

Melhor hiperparâmetro W para generalização: 17.0  
 Melhor hiperparâmetro Z para minimizar o erro empírico: 3.0

**viii. Compare o efeito dos hiperparâmetros do XGBoost e seus resultados.**

Resultados do XGBoost:							
	n_estimators	max_depth	learning_rate	Accuracy	Precision	Recall	\
0	50	4	0.20	0.750182	0.741033	0.750182	
1	50	10	1.00	0.743273	0.735919	0.743273	
2	50	10	0.20	0.758182	0.750804	0.758182	
3	50	10	0.04	0.748727	0.740375	0.748727	
4	200	4	1.00	0.732000	0.724881	0.732000	
5	200	4	0.20	0.763273	0.756408	0.763273	
6	200	4	0.04	0.748364	0.738808	0.748364	
7	200	10	0.20	0.758545	0.751984	0.758545	

	F1 Score	Confusion Matrix
0	0.740197	[[4, 5, 3, 0, 0, 0], [3, 7, 34, 3, 0, 0], [0, ...
1	0.737569	[[2, 8, 2, 0, 0, 0], [2, 16, 26, 3, 0, 0], [1, ...
2	0.750672	[[5, 5, 2, 0, 0, 0], [2, 12, 31, 2, 0, 0], [1, ...
3	0.740428	[[4, 4, 3, 0, 1, 0], [2, 12, 31, 2, 0, 0], [1, ...
4	0.727435	[[4, 5, 3, 0, 0, 0], [3, 13, 29, 2, 0, 0], [1, ...
5	0.756898	[[3, 6, 2, 1, 0, 0], [3, 16, 26, 2, 0, 0], [1, ...
6	0.737085	[[4, 4, 4, 0, 0, 0], [3, 6, 36, 2, 0, 0], [0, ...
7	0.752080	[[3, 7, 2, 0, 0, 0], [2, 15, 28, 2, 0, 0], [1, ...

**Comparação dos Efeitos dos Hiperparâmetros**

### Efeito do `n_estimators`

- **50 estimadores:** Modelos com 50 estimadores tendem a apresentar uma acurácia moderada a alta, com valores variando entre 0.743273 e 0.758182.
- **200 estimadores:** Modelos com 200 estimadores mostram uma ligeira melhora na acurácia, especialmente quando combinados com uma `learning_rate` de 0.20, atingindo até 0.763273 de acurácia.

### Efeito do `max_depth`

- **Profundidade 4:** A precisão e recall são consistentemente boas, mas o melhor desempenho (acurácia de 0.763273) foi observado com `n_estimators` de 200 e `learning_rate` de 0.20.
- **Profundidade 10:** A profundidade aumentada permite capturar padrões mais complexos, resultando em uma acurácia ligeiramente superior em alguns casos, mas não sempre. O melhor desempenho para esta profundidade foi uma acurácia de 0.758545.

### Efeito da `learning_rate`

- **Taxa de aprendizado 1.00:** Resultados mostram uma acurácia um pouco menor e menos estável. Exemplo: `n_estimators` de 200 e `max_depth` de 4 resultaram em uma acurácia de 0.732000.
- **Taxa de aprendizado 0.20:** Proporcionou os melhores resultados gerais. Exemplo: `n_estimators` de 200 e `max_depth` de 4 resultaram na maior acurácia de 0.763273.
- **Taxa de aprendizado 0.04:** Resultou em uma acurácia menor em comparação com 0.20, mas ainda razoavelmente boa. Exemplo: `n_estimators` de 50 e `max_depth` de 10 resultaram em uma acurácia de 0.748727.

### Conclusão

- **Melhor combinação:** `n_estimators` de 200, `max_depth` de 4 e `learning_rate` de 0.20 mostrou o melhor desempenho geral com uma acurácia de 0.763273.
- **Impacto dos hiperparâmetros:**

Aumentar o número de estimadores (`n_estimators`) tende a melhorar a acurácia, especialmente com uma `learning_rate` moderada (0.20). Aumentar a

profundidade (`max_depth`) pode capturar padrões mais complexos, mas deve ser balanceado com a `learning_rate` para evitar overfitting. A `learning_rate` de 0.20 parece ser um bom ponto de equilíbrio entre desempenho e estabilidade.

Esta análise ajuda a identificar as melhores configurações de hiperparâmetros para o modelo XGBoost, otimizando a performance do modelo para a tarefa de classificação específica.

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

Feito no notebook, resultado muito grande para colocar no documento.

### **2.3) Aprendizado por Reforço**

#### **i. Explique os parâmetros alpha, gamma e epsilon.**

Os parâmetros alpha, gamma e epsilon são fundamentais em algoritmos de aprendizado por reforço, especialmente em métodos como o Q-learning. Cada um desses parâmetros controla diferentes aspectos do processo de aprendizado do agente. Aqui está uma explicação detalhada para cada um:

##### **1. Alpha ( $\alpha$ )**

Alpha é a taxa de aprendizado ou learning rate. Ele controla o quanto as atualizações nas estimativas dos valores Q devem impactar os valores atuais durante o treinamento. Em termos simples, alpha determina a velocidade com a qual o agente aprende com novos dados.

- Valor Alto (próximo a 1): O agente aprende rapidamente, pois as atualizações são mais significativas. No entanto, isso pode levar a uma instabilidade, onde as estimativas podem oscilar muito.
- Valor Baixo (próximo a 0): O agente aprende lentamente, pois as atualizações são pequenas. Isso pode levar a um processo de aprendizado mais estável, mas também mais demorado.

##### **2. Gamma ( $\gamma$ )**

Gamma é o fator de desconto. Ele define a importância das recompensas futuras em comparação com as recompensas imediatas. Em outras palavras, gamma determina o quanto o agente valoriza recompensas futuras em relação às recompensas imediatas.

- Valor Alto (próximo a 1): O agente valoriza as recompensas futuras quase tanto quanto as recompensas imediatas. Isso é útil em ambientes onde as recompensas são alcançadas após várias ações.
- Valor Baixo (próximo a 0): O agente valoriza principalmente as recompensas imediatas. Isso é útil em ambientes onde as recompensas são obtidas rapidamente.

O fator de desconto é usado para calcular o valor de longo prazo de uma ação em um estado.

### 3. **Epsilon ( $\epsilon$ )**

Epsilon é o parâmetro de exploração-exploração, que controla a frequência com a qual o agente escolhe ações aleatórias em vez de seguir a política ótima. Esse equilíbrio é crucial para garantir que o agente explore o ambiente adequadamente enquanto tenta maximizar as recompensas.

- Valor Alto (próximo a 1): O agente explora mais e escolhe ações aleatórias com mais frequência. Isso ajuda o agente a descobrir novas estratégias e evitar ficar preso em soluções subótimas.
- Valor Baixo (próximo a 0): O agente explora menos e tende a seguir a política que já aprendeu. Isso maximiza a exploração da política atual, mas pode resultar em não descobrir soluções melhores.

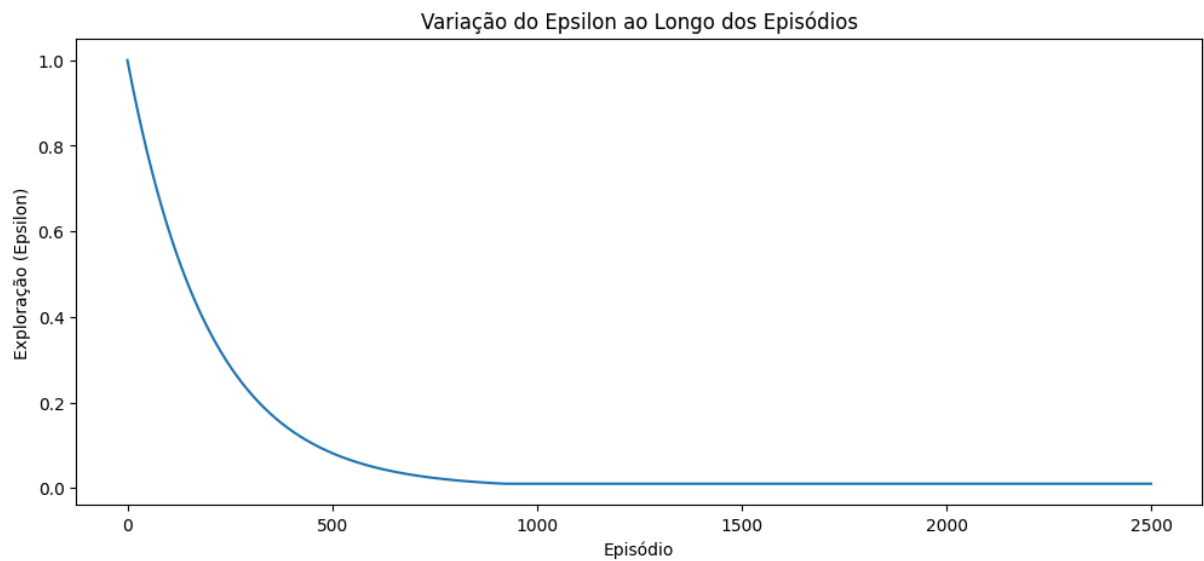
A política de escolha de ação é baseada em epsilon:

- Com probabilidade  $\epsilon$ , escolha uma ação aleatória (exploração).
- Com probabilidade  $1-\epsilon$ , escolha a ação com o maior valor  $Q$  (exploração).

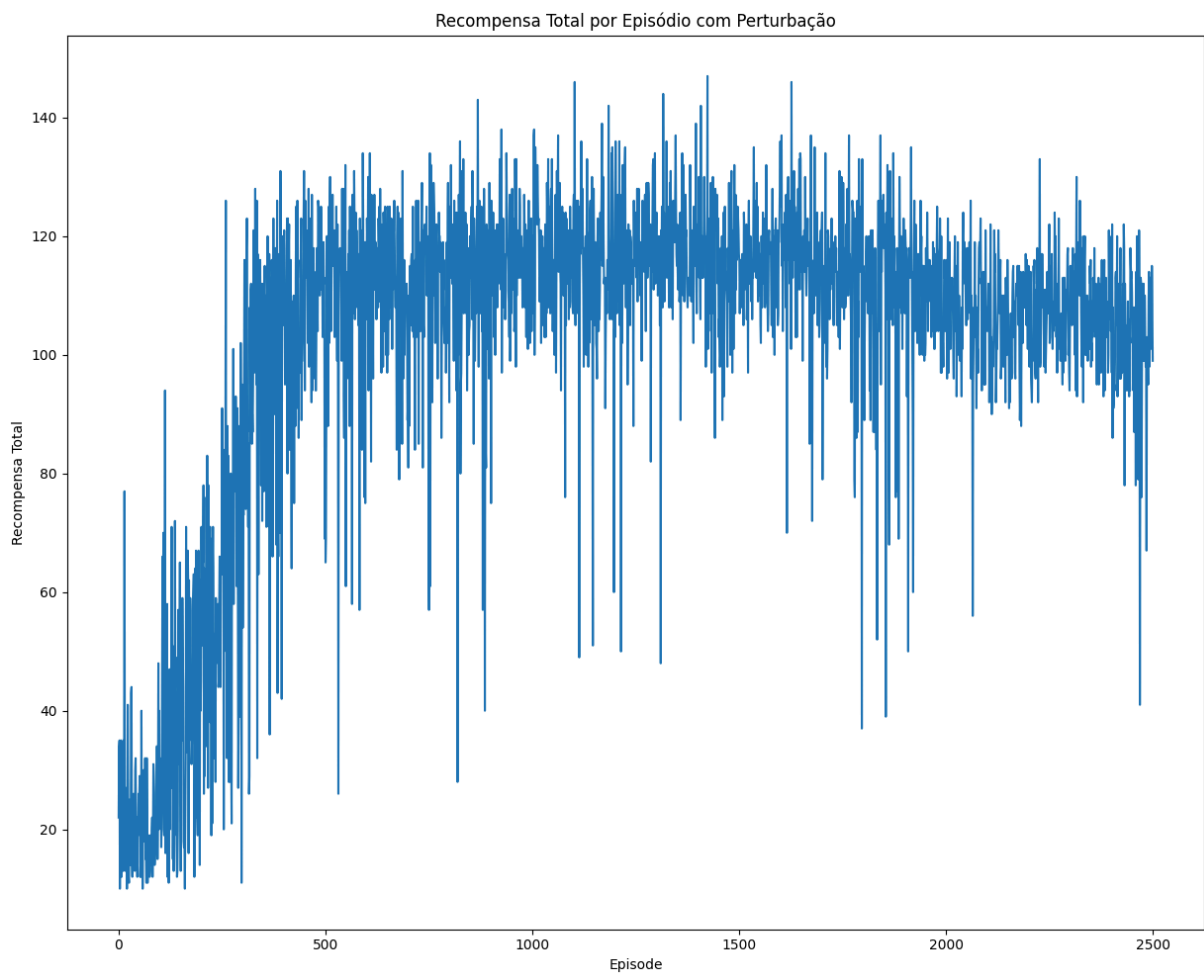
Epsilon geralmente começa alto e diminui ao longo do tempo (decaência de epsilon), permitindo que o agente explore mais no início e, gradualmente, se concentre mais na exploração de ações conhecidas à medida que o treinamento avança.

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





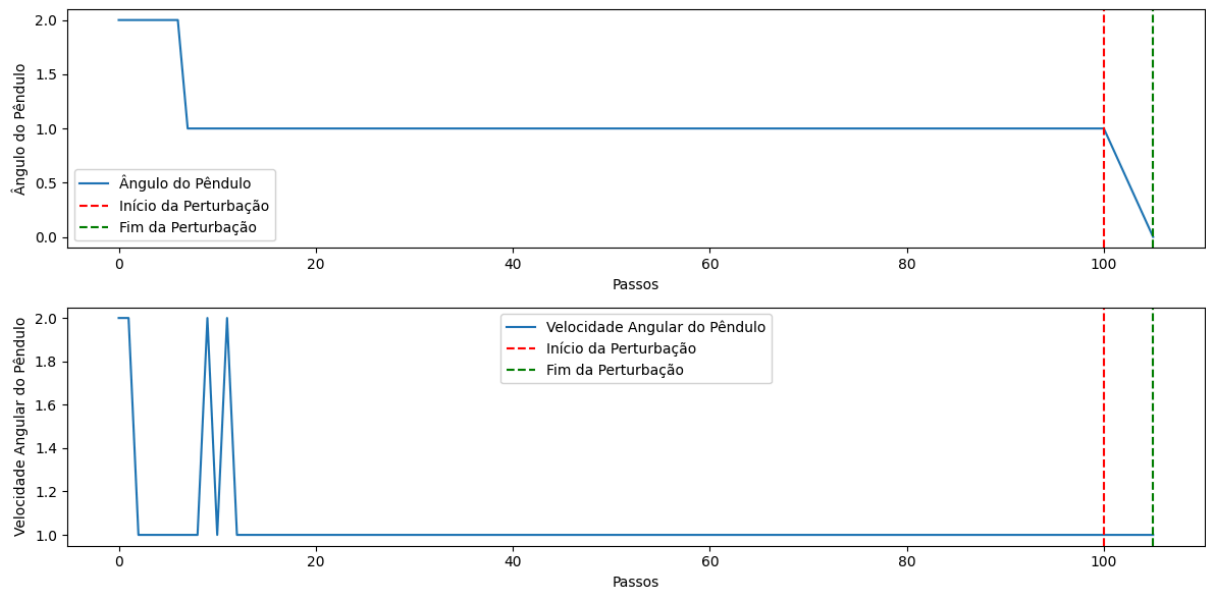
**iii. Faça um plot da recompensa total por número de episódios, do treinamento.**



**iv. Após o treinamento da Q-table, atingindo a estabilidade do CartPole faça o seguinte experimento e mostre os seguintes resultados:**

- Inicie um novo episódio e nos 100 primeiros passos faça o controle ótimo com a Q-table treinada.

- Dê 5 passos na simulação com ação fixa em 1: Carro para direito, simulando uma perturbação.
- Complete os 500 passos utilizando a Q-table treinada novamente.
- Registre os resultados e faça um plot do ângulo e velocidade angular ao longo da simulação. Identifique a incidência da perturbação e o retorno à estabilidade do sistema.



**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 usando Deep Reinforcement Learning (DRL), uma arquitetura comum é a Deep Q-Network (DQN). O DQN é uma extensão do Q-learning tradicional que usa uma rede neural profunda para aproximar a função Q. Abaixo, está uma descrição detalhada de como essa arquitetura pode ser utilizada para o problema do CartPole:

### 1. Entradas e Saídas

As entradas para a rede neural serão os estados observados do ambiente CartPole. Cada estado é representado por um vetor de quatro valores contínuos:

- Posição do carrinho (Cart Position).
- Velocidade do carrinho (Cart Velocity).

- Ângulo do pêndulo (Pole Angle).
- Velocidade angular do pêndulo (Pole Angular Velocity).

As saídas da rede neural serão os valores Q para cada ação possível no estado dado. No caso do CartPole, existem duas ações possíveis:

- Empurrar o carrinho para a esquerda (ação 0).
- Empurrar o carrinho para a direita (ação 1).

Portanto, a rede neural terá uma saída para cada ação, resultando em duas saídas no total.

2. **Arquitetura da Rede Neural:** A arquitetura típica para um DQN aplicado ao problema do CartPole pode ser relativamente simples, pois o problema é de baixa dimensão. Uma rede neural com algumas camadas totalmente conectadas (fully connected layers) é geralmente suficiente. Exemplo de Arquitetura:

- Entrada: Vetor de estado com 4 elementos.
- Camada Oculta 1: 24 neurônios, função de ativação ReLU.
- Camada Oculta 2: 24 neurônios, função de ativação ReLU.
- Saída: 2 neurônios (um para cada ação), sem função de ativação.

3. **Função de Perda:** A função de perda para o DQN é baseada no Bellman Equation. A perda é calculada como o erro quadrático médio (MSE) entre o valor Q estimado pela rede neural e o valor Q alvo calculado a partir da recompensa recebida e do valor Q futuro máximo.

4. **Implementação do Treinamento:** A implementação do treinamento inclui o processo de experiência de replay, onde transições (estado, ação, recompensa, próximo estado) são armazenadas em um buffer de replay e amostradas aleatoriamente para treinar a rede neural. Este processo ajuda a quebrar a correlação entre amostras consecutivas e estabiliza o treinamento.

#### Descrição dos Parâmetros

- Alpha ( $\alpha$ ): Taxa de aprendizado, define o quanto novos valores substituem os antigos durante o processo de atualização.
- Gamma ( $\gamma$ ): Fator de desconto, determina a importância de recompensas futuras em comparação com recompensas imediatas.

- Epsilon ( $\epsilon$ ): Taxa de exploração, define a probabilidade de escolher uma ação aleatória em vez da ação ótima.

### Experimento de Perturbação

Para realizar o experimento de perturbação conforme descrito, o mesmo código pode ser adaptado para incluir os passos específicos da perturbação após os primeiros 100 passos controlados pela Q-table treinada. Este método de Deep Reinforcement Learning pode ser aplicado com sucesso para resolver o problema do CartPole, garantindo um controle eficiente e robusto, mesmo diante de perturbações.

## 6. Conclusão

Neste trabalho prático, foram exploradas diferentes técnicas de aprendizado supervisionado e por reforço para abordar problemas específicos relacionados ao controle de ailerons em aviões e ao controle do pêndulo invertido no ambiente CartPole.

O trabalho consolidou o conhecimento em várias técnicas de aprendizado de máquina, demonstrando a importância de uma análise cuidadosa de dados, da escolha adequada dos modelos e da otimização de parâmetros para resolver problemas complexos em contextos reais. A aplicação dessas técnicas contribui significativamente para a melhoria da segurança e conforto na aviação, além de proporcionar uma base sólida para futuros estudos e implementações na área de inteligência artificial.

## REFERÊNCIAS

Google. Disponível em:

[https://www.gstatic.com/education/formulas2/553212783/en/bayes\\_\\_theorem.svg](https://www.gstatic.com/education/formulas2/553212783/en/bayes__theorem.svg). Acesso em: 30 jul. 2024.