Documentação do Projeto: Teorema de Bayes - Análise de Probabilidade de Spam dado Phishing

1. Introdução

Este projeto implementa o **Teorema de Bayes** para calcular a probabilidade de um e-mail ser **spam**, dado que ele foi identificado como um e-mail de **phishing**. A análise é crucial para sistemas de segurança de e-mail, permitindo uma compreensão mais aprofundada da relação entre essas duas categorias de ameaças. O objetivo é fornecer uma ferramenta clara e explicativa para demonstrar o cálculo bayesiano passo a passo.

2. O Que é o Teorema de Bayes?

O Teorema de Bayes é uma fórmula matemática usada para calcular a **probabilidade condicional** de um evento. Ele descreve como atualizar a probabilidade de uma hipótese com base em novas evidências. A fórmula geral é:

$P(A|B)=P(B)P(B|A)\cdot P(A)$

Onde:

- P(A|B): Probabilidade de A acontecer dado que B aconteceu (probabilidade a posteriori).
- P(B|A): Probabilidade de B acontecer dado que A aconteceu (verossimilhança).
- P(A): Probabilidade de A acontecer (probabilidade a priori).
- P(B): Probabilidade de B acontecer.

No contexto deste projeto, estamos interessados em:

- **A**: O evento de um e-mail ser **Spam**.
- **B**: O evento de um e-mail ser **Phishing**.

Portanto, a fórmula se traduz para:

$P(Spam|Phishing)=P(Phishing)P(Phishing|Spam)\cdot P(Spam)$

Onde P(Phishing) é calculado como:

$P(Phishing)=P(Phishing|Spam)\cdot P(Spam)+P(Phishing|\neg Spam)\cdot P(\neg Spam)$

3. Estrutura do Código

O código Python é composto por uma função principal e a execução do cálculo com dados específicos.

3.1. Função teorema_de_bayes

Python

```
def teorema_de_bayes(p_a, p_b_dado_a, p_b_dado_nao_a):
    p_nao_a = 1 - p_a
    numerador = p_b_dado_a * p_a
```

```
    denominador = (p_b_dado_a * p_a) + (p_b_dado_nao_a * p_nao_a)
    p_a_dado_b = numerador / denominador

    explicacao = [
        f"Passo 1: Calcular P(¬A) = 1 - P(A) = {p_nao_a:.6f}",
        f"Passo 2: Calcular Numerador = P(B|A) * P(A) = {numerador:.6f}",
        f"Passo 3: Calcular Denominador = P(B|A)*P(A) + P(B|¬A)*P(¬A) = {denominador:.6f}",
        f"Passo 4: Calcular P(A|B) = Numerador / Denominador = {p_a_dado_b:.6f}",
    ]

    for linha in explicacao:
        print(linha)
    return p_a_dado_b, explicacao
```

Parâmetros de Entrada:

- p_a (P(A)): Probabilidade a priori do evento A (neste caso, P(Spam)).
- p_b_dado_a (P(B|A)): Probabilidade do evento B dado o evento A (neste caso, P(Phishing | Spam)).
- p_b_dado_nao_a (P(B|¬A)): Probabilidade do evento B dado o não evento A (neste caso, P(Phishing | ¬Spam)).

Cálculos Realizados:

1. **p_nao_a**: Calcula a probabilidade de ¬A (não spam) como 1−P(A).
2. **numerador**: Calcula o numerador da fórmula de Bayes: P(B|A)·P(A).
3. **denominador**: Calcula o denominador da fórmula de Bayes, que é a probabilidade total de B: P(B|A)·P(A)+P(B|¬A)·P(¬A).
4. **p_a_dado_b**: Calcula a probabilidade condicional final P(A|B) dividindo o numerador pelo denominador.

Saída: A função retorna dois valores:

- A probabilidade P(A|B) calculada.
- Uma lista de strings (explicacao) detalhando cada passo do cálculo, formatado para facilitar a compreensão.

3.2. Execução Principal

Python

```
print("==== Teorema de Bayes: Probabilidade de Spam dado Phishing ====\n")
```

```
# Dados com base em fontes reais:
p_a = 0.468  # P(Spam)
p_b_dado_a = 0.01825  # P(Phishing | Spam)
p_b_dado_nao_a = 0.0001  # P(Phishing | ¬Spam)

# Cálculo
resultado, passos = teorema_de_bayes(p_a, p_b_dado_a, p_b_dado_nao_a)
resultado_str = f"\nResultado final: P(Spam | Phishing) = {resultado:.6f} ({resultado * 100:.2f}%)"
print(resultado_str)
```

Nesta seção, os dados de entrada são definidos com base em **fontes reais** (mencionadas na seção 4). Esses valores são passados para a função teorema_de_bayes, e o resultado é exibido no console, incluindo o valor percentual para melhor interpretação.

4. Dados Utilizados e Fontes

Os dados de probabilidade utilizados neste projeto são baseados em estatísticas reais sobre spam e phishing, conforme as seguintes fontes:

- **P(Spam) = 0.468 (46.8%)**: Representa a probabilidade geral de um e-mail ser spam.
    - Fonte: Mailmodo (2025): https://www.mailmodo.com/guides/email-spam-statistics/
- **P(Phishing | Spam) = 0.01825 (1.825%)**: Representa a probabilidade de um e-mail ser phishing dado que ele é spam. Este valor foi derivado de uma interpretação que sugere que uma pequena parcela do spam é de fato phishing.
    - Fontes: AAG IT (2025): https://aag-it.com/the-latest-phishing-statistics/ e Astra Security (2025): https://www.getastra.com/blog/security-audit/phishing-attack-statistics/
- **P(Phishing | ¬Spam) = 0.0001 (0.01%)**: Representa a probabilidade de um e-mail ser phishing dado que ele *não* é spam. Este valor é consideravelmente menor, refletindo que a maioria dos e-mails de phishing são categorizados como spam.
    - Fontes: AAG IT (2025): https://aag-it.com/the-latest-phishing-statistics/ e Astra Security (2025): https://www.getastra.com/blog/security-audit/phishing-attack-statistics/

É importante notar que, embora os valores sejam baseados em dados reais, o cenário exato e a intersecção de "spam" e "phishing" podem variar dependendo da metodologia de classificação e das fontes.

5. Geração de Arquivo de Saída

O script também gera um arquivo de texto chamado resultado_bayes.txt, que contém todos os detalhes da execução:

Python

```
with open("resultado_bayes.txt", "w", encoding="utf-8") as f:
    f.write("==== Teorema de Bayes: Probabilidade de Spam dado Phishing ====\n\n")
    f.write("Entradas:\n")
    f.write(f"- P(Spam) = {p_a}\n")
    f.write(f"- P(Phishing | Spam) = {p_b_dado_a}\n")
    f.write(f"- P(Phishing | ¬Spam) = {p_b_dado_nao_a}\n\n")

    f.write("Etapas do cálculo:\n")
    for passo in passos:
        f.write(passo + "\n")

    f.write(resultado_str + "\n\n")

    f.write("=== Fontes ===\n")
    for fonte in fontes:
        f.write("- " + fonte + "\n")
```

Este arquivo serve como um registro completo da análise, incluindo:

- O título do projeto.
- As probabilidades de entrada (P(Spam), P(Phishing | Spam), P(Phishing | ¬Spam)).
- Os passos detalhados do cálculo.
- O resultado final (P(Spam|Phishing)).
- As fontes dos dados.

6. Resultados e Análise

Ao executar o código com os dados fornecidos, o resultado esperado para P(Spam|Phishing) é de aproximadamente **0.976510 (97.65%)**.

Este resultado indica que, se um e-mail for classificado como **phishing**, a probabilidade de ele também ser **spam** é extremamente alta. Isso valida a intuição de que a grande maioria dos ataques de phishing ocorre por meio de e-mails de spam.

A análise do denominador (P(Phishing)) é crucial:

- A contribuição de P(Phishing|Spam)·P(Spam) é significativa, pois, embora P(Phishing|Spam) seja relativamente baixa, P(Spam) é alta.
- A contribuição de P(Phishing|¬Spam)·P(¬Spam) é muito menor, já que P(Phishing|¬Spam) é extremamente baixa.

Isso demonstra como o Teorema de Bayes permite que, mesmo com uma baixa probabilidade de phishing dentro de spam, a alta prevalência de spam em geral leva a uma probabilidade posterior muito alta de que um e-mail de phishing seja, de fato, spam.

7. Conclusão

Este projeto demonstra uma aplicação prática do Teorema de Bayes para entender a relação entre spam e phishing em sistemas de segurança de e-mail. A implementação clara e a documentação detalhada dos passos de cálculo e das fontes de dados tornam este projeto uma ferramenta valiosa para compreender as probabilidades condicionais no contexto de segurança cibernética. A alta probabilidade de um e-mail de phishing ser spam (97.65%) reforça a importância das defesas anti-spam como uma primeira linha de proteção contra ataques de phishing.