Oficina 6

Contextualização do problema: O objetivo deste exercício é construir um modelo de classificação capaz de analisar e-mails e determinar se eles são **spam** ou **não spam** com base na ocorrência de palavras específicas em cada mensagem. O dataset contém as palavras que aparecem nos e-mails, e com essas informações, devemos classificar os e-mails de forma eficiente.

Escolha do Algoritmo: Naive Bayes

Justificativa técnica: O **Naive Bayes** é uma excelente escolha para esse tipo de tarefa, pois é um algoritmo probabilístico que trabalha muito bem com dados textuais. Ele calcula a probabilidade de um e-mail ser de uma classe (spam ou não spam) com base nas palavras que aparecem nele. A técnica é chamada de "naive" (ingênuo) porque assume que a presença de uma palavra é independente das outras, o que simplifica o cálculo. Para ilustrar como o Naive Bayes funciona, considere a tabela a seguir, que contém dados sobre alguns e-mails e as palavras que aparecem neles:

E-mail	Classe	Promoção	Importante
“Promoção única”	Spam	Sim	Não
“Promoção incrível”	Spam	Sim	Não
“Importante reunião”	Não Spam	Não	Sim
“Muito importante”	Não Spam	Não	Sim

A partir dessa tabela, o Naive Bayes calculará as probabilidades para determinar se um novo e-mail é **Spam** ou **Não Spam**.

Passo 1: Calcular as Probabilidades A Priori

Primeiro, calculamos as probabilidades a priori das classes **Spam** e **Não Spam**. Com base no dataset fornecido, temos:

P(Spam) = 2/4 = 0.5
P(Não Spam) = 2/4 = 0

Ou seja, a probabilidade de um e-mail ser **Spam** ou **Não Spam** é igual, 50% para cada classe.

Passo 2: Calcular as Probabilidades Condicionais

Agora, vamos calcular as probabilidades condicionais para cada palavra presente nos e-mails. Começamos com a palavra "Promoção":

P(Promoção | Spam) = 2/2 = 1.0 -> Ou seja, temos a palavra “Promoção” aparecendo 2 vezes como Spam, onde temos duas ocorrências de emails como spam no dataset.

P(Promoção | Não Spam) = 0/2 = 0.0

Em seguida, para a palavra "Importante":

P(Importante | Spam) = 0/2 = 0.0
P(Importante | não Spam) = 2/2 = 1.0

Passo 3: Aplicar a Suavização de Laplace

A suavização de Laplace é usada para evitar o problema de contagem zero, o que pode acontecer quando uma palavra não aparece em uma determinada classe. Para isso, aplicamos o seguinte ajuste:

Para α=1 (suavização de Laplace):
P(Promoção | Spam) = (2+1) (2+2) = 0.75
P(Importante | Spam) = (0+1) (2+2) = 0.25

P(Promoção | Não Spam) = (0+1) (2+2) = 0.25
P(Importante | não Spam) = (2+1) (2+2) = 0.75

Agora, com as probabilidades ajustadas, podemos calcular a probabilidade de um e-mail ser **Spam** ou **Não Spam** dado um conjunto de palavras, como "Promoção Importante".

Passo 4: Calcular as Probabilidades para um Novo E-mail

Vamos calcular as probabilidades de um novo e-mail, que contém as palavras "Promoção" e "Importante". Usando as probabilidades ajustadas:

- **Para Spam**:
  P(Spam | X) = 0.75 * 0.25 * 0.5 = 0.09375

- **Para Não Spam**

  P(Não Spam | X) = 0.25 * 0.75 * 0.5 = 0.09375

Passo 5: Interpretação dos Resultados

Neste caso, como as probabilidades para **Spam** e **Não Spam** são iguais (0.09375 para ambos), o modelo não consegue classificar esse e-mail de forma definitiva apenas com as informações fornecidas. Em situações como essa, seria necessário utilizar mais informações ou palavras adicionais para resolver o empate e fazer uma classificação mais precisa.

Adequação do problema: Além do algoritmo Naive Bayes ser perfeito para problemas que envolvem mineração de textos, ele tbm é fácil de implementar e não exige muitos recursos computacionais, o que é ideal para sistemas de e-mail, além disso, funciona bem com textos curtos e variáveis, como e-mails, pois calcula a probabilidade de cada palavra ser associada a um rótulo específico (spam ou não spam).