# Processamento de Linguagem Natural

## Naive Bayes

Yuri Malheiros (yuri@ci.ufpb.br)

# Detecção de Spam

## Inbox

SPAM?

X

X

X

# Detecção de Spam

- Dado um e-mail, queremos saber qual a probabilidade dele ser Spam

$$P(Spam \mid Email) = ?$$

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

# Naive Bayes

- Dado um exemplo definido pelas características: [$x_1$, $x_2$, ..., $x_n$]
- Dado um conjunto de possíveis classes C
- O classificador retorna a classe **c** que tenha a maior probabilidade dadas as características do exemplo de teste:

$$R = argmax_{(c \in C)} P(c \mid x_1, x_2, \ldots, x_n)$$

# Naive Bayes

- Dado um exemplo definido pelas características: [$x_1$, $x_2$, ..., $x_n$]
- Dado um conjunto de possíveis classes C
- O classificador retorna a classe **c** que tenha a maior probabilidade dadas as características do exemplo de teste:

$$P(Spam \mid Email) = ?$$

$$P(NaoSpam \mid Email) = ?$$

# Detecção de Spam

- Como calcular $P(Spam \mid Email) = ?$
- Precisamos saber a probabilidade de ser Spam dado que temos um determinado e-mail
- Assim, o conteúdo deste e-mail precisa aparecer múltiplas vezes no dataset
- Quantos possíveis e-mails eu posso ter?

# Naive Bayes

- Qual o problema nessa abordagem?
  - Precisaríamos de um dataset imenso, onde os mais variados e-mails teriam que aparecer múltiplas vezes
  - Isso é inviável

# Naive Bayes

- O Naive Bayes utiliza o teorema de Bayes para nos ajudar no cálculo da probabilidade:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

# Naive Bayes

- Aplicando o Teorema de Bayes ao nosso exemplo:

$$P(Spam \mid Email) = \frac{P(Email \mid Spam)P(Spam)}{P(Email)}$$

# Naive Bayes

- Como calcular $P(Email \mid Spam) = ?$
- Precisamos saber a probabilidade do e-mail dado que temos um Spam
- Agora temos um problema mais simples, pois só existem duas condições: Spam e Não Spam

# Naive Bayes

- De forma geral, no Naive Bayes, temos:

$$P(c \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid c)P(c)}{P(x_1, x_2, \ldots, x_n)}$$

# Naive Bayes

- Substituindo na equação para encontrar R, temos:

$$R = argmax_{(c \in C)} \frac{P(x_1, x_2, \ldots, x_n \mid c) P(c)}{P(x_1, x_2, \ldots, x_n)}$$

# Naive Bayes

- Para classificação de Spam, vamos precisar calcular a probabilidade duas vezes
- Uma para Spam e outra para Não Spam
- A classificação será a classe que resultar na maior probabilidade

# Naive Bayes

$$P(Spam \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid Spam)P(Spam)}{P(x_1, x_2, \ldots, x_n)}$$

$$P(NaoSpam \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid NaoSpam)P(NaoSpam)}{P(x_1, x_2, \ldots, x_n)}$$

# Naive Bayes

$$P(Spam \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid Spam)P(Spam)}{P(x_1, x_2, \ldots, x_n)}$$

constante

$$P(NaoSpam \mid x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n \mid NaoSpam)P(NaoSpam)}{P(x_1, x_2, \ldots, x_n)}$$

constante

# Naive Bayes

$$P(Spam \mid x_1, x_2, \ldots, x_n) = P(x_1, x_2, \ldots, x_n \mid Spam)P(Spam)$$

$$P(NaoSpam \mid x_1, x_2, \ldots, x_n) = P(x_1, x_2, \ldots, x_n \mid NaoSpam)P(NaoSpam)$$

# Naive Bayes

- Substituindo na equação para encontrar R, temos:

$$R = argmax_{(c \in C)} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

# Detecção de Spam

- Qual a probabilidade de uma mensagem ser Spam?
- P(Spam) = ?

# Detecção de Spam

- Qual a probabilidade de uma mensagem ser Spam?
- P(Spam) = ?

- Temos 8 mensagens e 3 são Spam, então:
- P(Spam) = 3/8

# Detecção de Spam

- Qual a probabilidade de uma mensagem não ser Spam?
- P(NãoSpam) = ?

# Detecção de Spam

- Qual a probabilidade de uma mensagem não ser Spam?
- P(NãoSpam) = ?

- Temos 8 mensagens e 5 não são Spam, então:
- P(NãoSpam) = 5/8

# Detecção de Spam

- Dada uma mensagem que é de uma determinada classe, qual a probabilidade de aparecer a palavra “secret”?

- P(“secret”|Spam) = ?
- P(“secret”|NãoSpam) = ?

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

9 palavras

3 "secret"

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

15 palavras

1 "secret"

# Detecção de Spam

- Calculando P(“palavra”|c):
  - Contamos quantas vezes “palavra" aparece na classe c
  - Contamos quantas palavras existem na classe c
  - Dividimos o primeiro valor pelo segundo

# Detecção de Spam

- Dada uma mensagem que é de uma determinada classe, qual a probabilidade de aparecer a palavra “secret”?

- P(“secret”|Spam) = 3/9
- P(“secret”|NãoSpam) = 1/15

# Detecção de Spam

- Qual a probabilidade da mensagem “sport” ser Spam?

- P(Spam | "sport") = P(“sport" | Spam)P(Spam)

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

# Detecção de Spam

- Qual a probabilidade da mensagem “sport” ser Spam?

- P(Spam | "sport") = P(“sport" | Spam)P(Spam)
- P(Spam | "sport") = 1/9 * 3/8 = 0,0416

# Detecção de Spam

- Qual a probabilidade da mensagem “sport” ser NaoSpam?

- P(NaoSpam | "sport") = P(“sport" | NaoSpam)P(NaoSpam)

# Detecção de Spam

- Qual a probabilidade da mensagem “sport” ser NaoSpam?

- P(NaoSpam | "sport") = P(“sport" | NaoSpam)P(NaoSpam)
- P(NaoSpam | "sport") = 5/15 * 5/8 = 0,208

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser Spam?

- P(Spam | “secret”, “is”, “secret”) = ?

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser Spam?

- P(Spam | “secret”, “is”, “secret”) = P(“secret”|Spam)P(“is”|Spam)P(“secret”|Spam)P(Spam)

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser Spam?

- P(Spam | “secret”, “is”, “secret”) = P(“secret”|Spam)P(“is”|Spam)P(“secret”|Spam)P(Spam)
- P(Spam | “secret”, “is”, “secret”) = 3/9 * 1/9 * 3/9 * 3/8 = 0,00462

# Naive Bayes

- No cálculo anterior, consideramos que cada palavra ocorre de forma independente
- Na teoria isso não é verdade, as palavras interagem e dependem umas das outras
- Para simplificar e tornar viável, consideramos que eles são independentes
- Essa é uma suposição ingênua (naive), por isso o nome do algoritmo

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser NãoSpam?

- P(NaoSpam | “secret”, “is”, “secret”) = ?

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser NãoSpam?

- P(NãoSpam | “secret”, “is”, “secret”) = P(“secret”|NaoSpam)P(“is”|NaoSpam)P(“secret”|NaoSpam)P(NaoSpam)

# Detecção de Spam

- Qual a probabilidade da mensagem “secret is secret” ser NãoSpam?

- P(NãoSpam | “secret”, “is”, “secret”) = P(“secret”|NaoSpam)P(“is”|NaoSpam)P(“secret”|NaoSpam)P(NaoSpam)
- P(NaoSpam | “secret”, “is”, “secret”) = 1/15 * 1/15 * 1/15 * 5/8 = 0,00018

# Detecção de Spam

- P(Spam | “secret”, “is”, “secret”) = 0,00462
- P(NaoSpam | “secret”, “is”, “secret”) = 0,00018

# Detecção de Spam

- **P(Spam | “secret”, “is”, “secret”) = 0,00462**
- P(NaoSpam | “secret”, “is”, “secret”) = 0,00018

# Detecção de Spam

- Qual a probabilidade da mensagem “today is secret” ser Spam?

- P(Spam | “today”, “is”, “secret”) = ?

# Detecção de Spam

- Qual a probabilidade da mensagem “today is secret” ser Spam?

- P(Spam | “today”, “is”, “secret”) = P(“today”|Spam)P(“is”|Spam)P(“secret”|Spam)P(Spam)

# Detecção de Spam

## Spam

- offer is secret
- click secret link
- secret sport link

## Não Spam

- play sport today
- went play sport
- secret sport event
- sport is today
- sport costs money

# Detecção de Spam

- Qual a probabilidade da mensagem “today is secret” ser Spam?

- P(Spam | “today”, “is”, “secret”) = P(“today”|Spam)P(“is”|Spam)P(“secret”|Spam)P(Spam)
- P(Spam | “today”, “is”, “secret”) = 0/9 * 1/9 * 3/9 * 3/8 = 0

# Detecção de Spam

- O zero é um problema
- Ele torna todas as outras probabilidades irrelevantes
- Uma única palavra está determinando o resultado da nossa análise
- Problema de superajuste

# Suavização

- Para resolver esse problema vamos utilizar a suavização de Laplace
- Vamos supor que cada palavra aparece uma vez mais nos dados de treinamento
  - Adicionaremos 1 em todas as contagens nos cálculos das probabilidades

| Spam | | | Não Spam | | |
|---|---|---|---|---|---|
| Sem Suavização | | Com suavização | Sem Suavização | | Com suavização |
| offer | 1 | offer | play | 2 | play |
| is | 1 | is | sport | 5 | sport |
| secret | 3 | secret | today | 2 | today |
| click | 1 | click | went | 1 | went |
| link | 2 | link | secret | 1 | secret |
| sport | 1 | sport | event | 1 | event |
| | | | is | 1 | is |
| | | | costs | 1 | costs |
| | | | money | 1 | money |
| Total | 9 | Total | Total | 15 | Total |

| Spam | | | | Não Spam | | | |
|---|---|---|---|---|---|---|---|
| Sem Suavização | | Com suavização | | Sem Suavização | | Com suavização | |
| offer | 1 | offer | 2 | play | 2 | play | 3 |
| is | 1 | is | 2 | sport | 5 | sport | 6 |
| secret | 3 | secret | 4 | today | 2 | today | 3 |
| click | 1 | click | 2 | went | 1 | went | 2 |
| link | 2 | link | 3 | secret | 1 | secret | 2 |
| sport | 1 | sport | 2 | event | 1 | event | 2 |
| | | <desconhecida> | 1 | is | 1 | is | 2 |
| | | | | costs | 1 | costs | 2 |
| | | | | money | 1 | money | 2 |
| | | | | | | <desconhecida> | 1 |
| Total | 9 | Total | 16 | Total | 15 | Total | 25 |

# Suavização

$$P_{Laplace}(palavra|c) = \frac{count(palavra, c) + 1}{N(c) + V(c) + 1}$$

- count(palavra, c) - quantidade de vezes que a palavra aparece na classe c
- N(c) - quantidade de palavras nos exemplos para classe c
- V(c) - quantidade de palavras únicas para a classe c

# Suavização

$$P_{Laplace}(palavra|c) = \frac{count(palavra, c) + 1}{N(c) + V(c) + 1}$$

- Como adicionamos 1 para cada palavra existente, então precisamos aumentar o total de palavras de acordo N(c)
- Também somamos 1 no denominador para representar palavras desconhecidas

# Underflow

- Nos cálculos de probabilidade de uma mensagem multiplicamos as probabilidades de cada palavra
- Como os valores estão entre 0 e 1, após várias multiplicações, podemos ter valores muito pequenos

# Underflow

- Quando o número é tão próximo de zero que o computador não consegue diferenciá-lo do próprio zero, temos Underflow
  - Tente fazer 0.1 elevado a 1000 no Python

# Underflow

- Para evitar o underflow podemos calcular o logaritmo da multiplicação das probabilidades:

$$\log(P(a_1 \mid c)P(a_2 \mid c) \ldots P(a_n \mid c)) = \log(P(a_1 \mid c)) + \log(P(a_2 \mid c)) + \ldots + \log(P(a_n \mid c))$$

# Exemplo

- Vamos fazer um exemplo usando um dataset maior e o Scikit-Learn no Jupyter Notebook