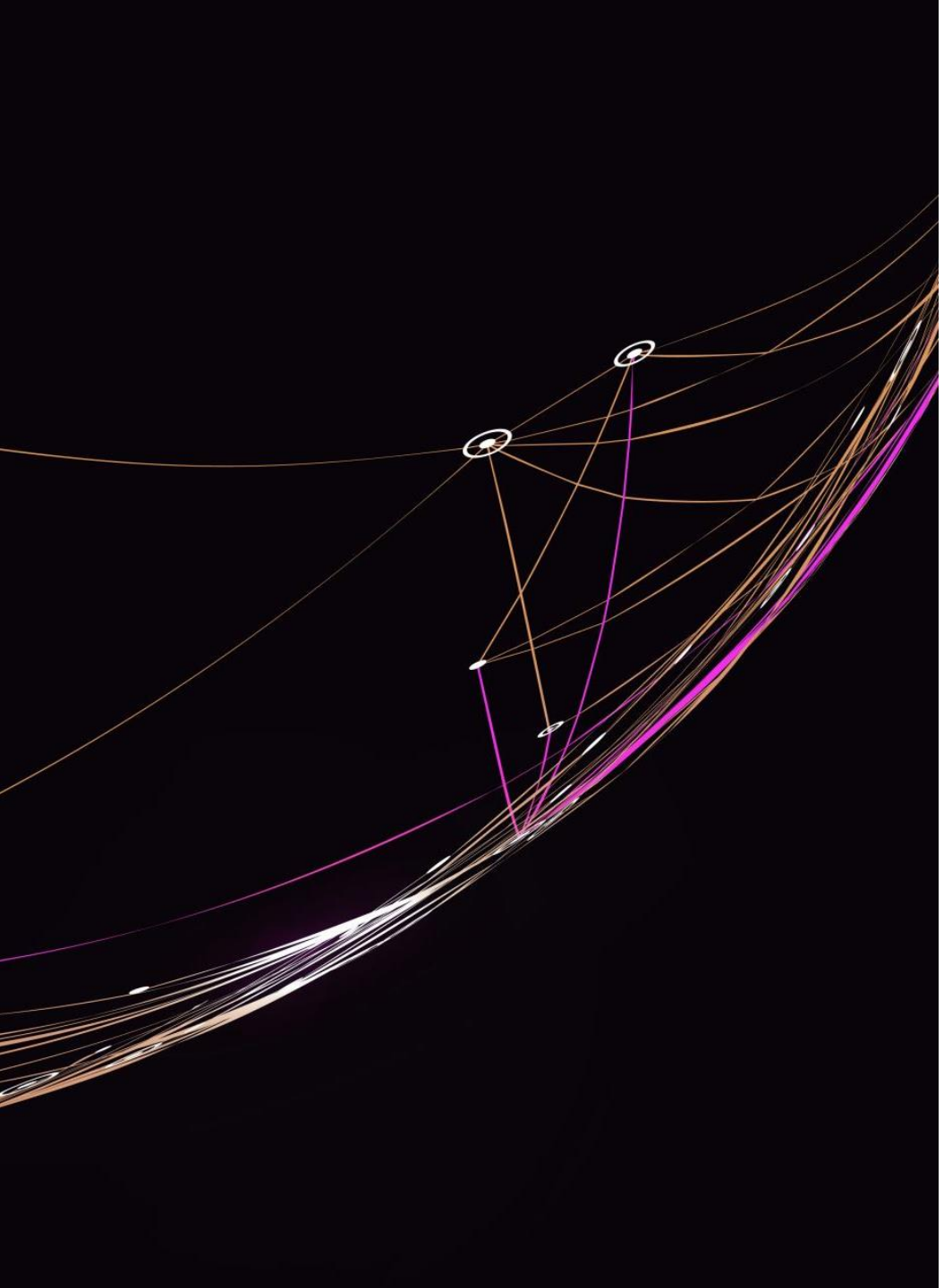


ADMINISTRAÇÃO

IBM0112 DATA MINING

Naïve Bayes

Cassius Figueiredo

O classificador Naïve Bayes

- Utiliza as descobertas de Thomas Bayes para realizar predições.
- O termo “naïve” (ingênuo) diz respeito à forma como o algoritmo analisa as características de uma base de dados, **assumindo que as variáveis são independentes entre si**.
- **Também pressupõe que as variáveis sejam todas igualmente importantes para o resultado**. Em cenários em que isso não ocorre, essa técnica deixa de ser uma boa opção.

O classificador Naïve Bayes - Vantagens

- Não necessita de muitos dados para o processo de treinamento.
- Implementação e uso simples.
- Facilmente escalável (paralelizável) para muitos dados e preditores.
- Aceita dados contínuos ou discretos.
- Não é sensível a outliers.
- Muito utilizado para predições em tempo real por entregar uma resposta rápida.

O classificador Naïve Bayes - Desvantagens

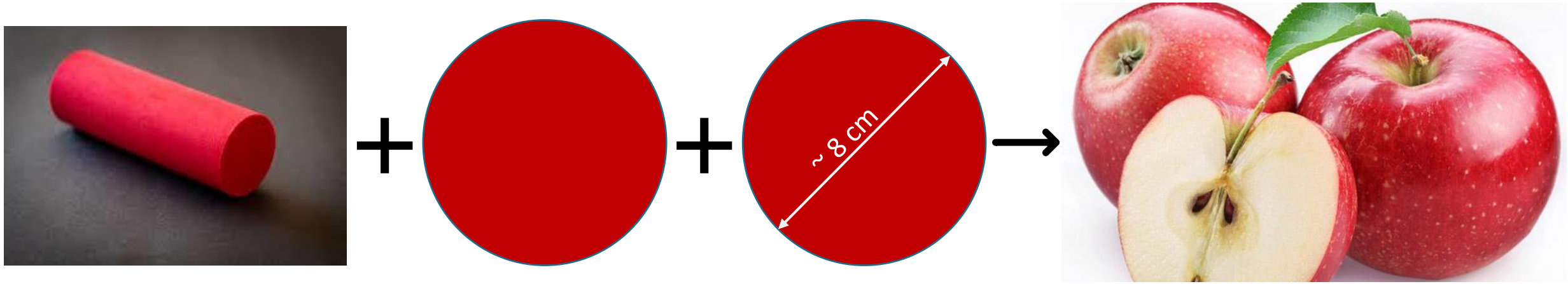
- Assume que as variáveis do problema são independentes (ou não-correlacionadas), o que raramente acontece em problemas reais. Dificulta a sua utilização ampla, mas pode ser útil como algoritmo de referência (benchmark).
- A abordagem de cálculo ("ingênua") torna difícil o uso direto dos valores em si, não sendo indicado para problemas de regressão ou de estimação de probabilidades.
- Sofre do problema de probabilidade zero ("*zero-probability problem*"). Para funcionar em classes-alvo que não ocorrem no conjunto de treinamento necessita de ajustes, tais como a suavização de Laplace, em muitos casos já incluída como parâmetro nas implementações computacionais.

Cenários de uso

- Classificadores em geral.
  - Dignósticos
  - Spam
- Sistemas de recomendação, de forma isolada ou em conjunto com outros algoritmos (ensembles).
- Análise de sentimentos.
- Natural Language Processing (NLP).

Como funciona?

- Um classificador Naïve Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outra característica.

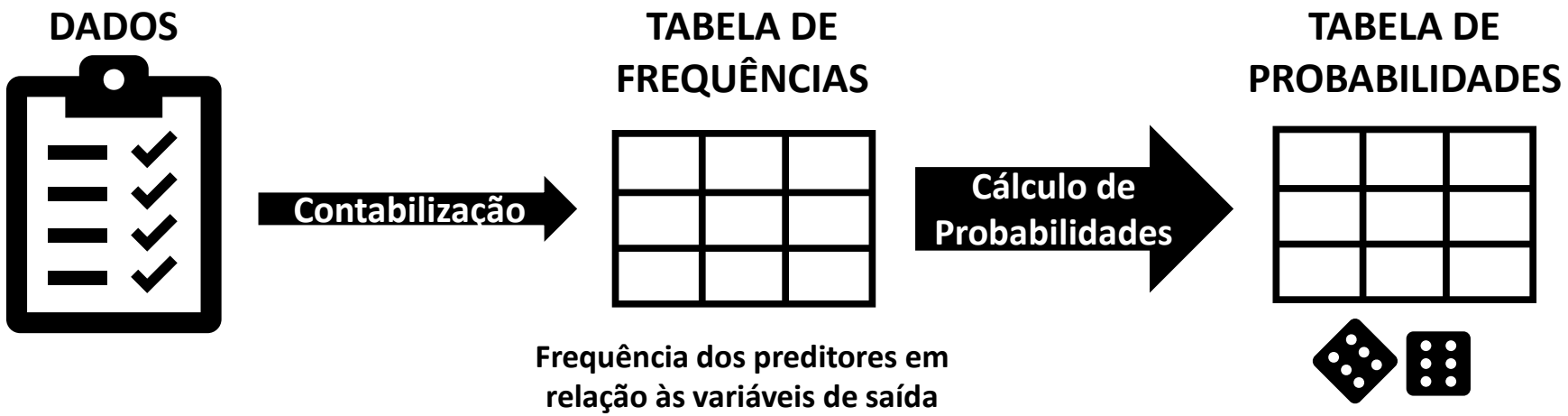


Maçãs!

Todas estas propriedades contribuem de forma independente para que o fruto seja classificado como maçã!

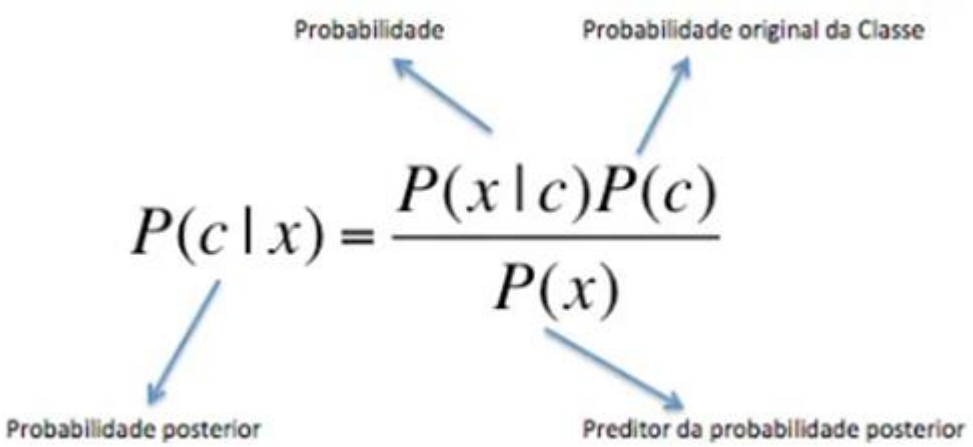
Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Estrutura do classificador Naïve Bayes



Teorema de Bayes

- O Teorema de Bayes nos fornece uma forma de calcular a probabilidade posterior $P(c|x)$, à partir das probabilidades individuais $P(c)$, $P(x)$ e $P(x|c)$.



- $P(c|x)$ é a probabilidade posterior da classe (c, alvo), dado o preditor (x, atributos).
- $P(c)$ é a probabilidade original da classe.
- $P(x|c)$ é a probabilidade do preditor, dada a classe.
- $P(x)$ é a estimação da probabilidade original do preditor.

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Temos um conjunto de dados de treinamento com condições de tempo e a variável-alvo ‘Jogar?’ (sugerindo a possibilidade de um grupo de crianças se divertir do lado de fora).
- Queremos verificar a possibilidade das crianças brincarem ou não, com base na condição de tempo.

DADOS	
Condições	**Jogar?**
Sol	Não
Nublado	Sim
Chuva	Sim
Sol	Sim
Sol	Sim
Nublado	Sim
Chuva	Não
Chuva	Não
Sol	Sim
Chuva	Sim
Sol	Não
Nublado	Sim
Nublado	Sim
Chuva	Não

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Passo 1: converter o conjunto de dados em uma tabela de frequências.

DADOS	
Condições	**Jogar?**
Sol	Não
Nublado	Sim
Chuva	Sim
Sol	Sim
Sol	Sim
Nublado	Sim
Chuva	Não
Chuva	Não
Sol	Sim
Chuva	Sim
Sol	Não
Nublado	Sim
Nublado	Sim
Chuva	Não

TABELA DE FREQUÊNCIAS		
Condições	**Não**	**Sim**
Nublado	0	4
Sol	2	3
Chuva	3	2
Total	**5**	**9**

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Passo 2: criar tabela de probabilidades a partir da tabela de frequências.

TABELA DE FREQUÊNCIAS		
Condições	**Não**	**Sim**
Nublado	0	4
Sol	2	3
Chuva	3	2
Total	**5**	**9**

TABELA DE PROBABILIDADES				
Condições	**Não**	**Sim**		
Nublado	0	4	**= 4/14**	**0,29**
Sol	2	3	**= 5/14**	**0,36**
Chuva	3	2	**= 5/14**	**0,36**
Total	**5**	**9**		
	= 5/14	**= 9/14**		
	0,36	**0,64**		

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Passo 3: use a equação Bayesiana Naïve para calcular a probabilidade posterior para cada classe. A classe com maior probabilidade posterior será definida como o resultado da previsão.

- Problema: as crianças devem sair para brincar se o tempo está ensolarado?

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Usando o método de probabilidade posterior:

$$\mathcal{P}(Sim/Ensolarado) = \frac{\mathcal{P}(Ensolarado/Sim) \cdot \mathcal{P}(Sim)}{\mathcal{P}(Ensolarado)}$$

- Aqui temos:

$$\mathcal{P}(Ensolarado/Sim) = {}^{3}/_{9} = 0{,}33$$

$$\mathcal{P}(Ensolarado) = {}^{5}/_{14} = 0{,}36$$

$$\mathcal{P}(Sim) = {}^{9}/_{14} = 0{,}64$$

TABELA DE PROBABILIDADES				
Condições	**Não**	**Sim**		
Nublado	0	4	**= 4/14**	**0,29**
Sol	2	3	**= 5/14**	**0,36**
Chuva	3	2	**= 5/14**	**0,36**
Total	**5**	**9**		
	= 5/14	**= 9/14**		
	0,36	**0,64**		

- A probabilidade final deste cenário será:

$$\mathcal{P}(Sim/Ensolarado) = \frac{0{,}33 \cdot 0{,}64}{0{,}36} = \mathbf{0,60}$$

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo

- Agora, para o outro cenário em questão:

$$\mathcal{P}(Não/Ensolarado) = \frac{\mathcal{P}(Ensolarado/Não) \cdot \mathcal{P}(Não)}{\mathcal{P}(Ensolarado)}$$

- Aqui temos:

$$\mathcal{P}(Ensolarado/Não) = {}^{2}/_{5} = 0{,}40$$

$$\mathcal{P}(Ensolarado) = {}^{5}/_{14} = 0{,}36$$

$$\mathcal{P}(Não) = {}^{5}/_{14} = 0{,}36$$

TABELA DE PROBABILIDADES				
Condições	**Não**	**Sim**		
Nublado	0	4	**= 4/14**	**0,29**
Sol	2	3	**= 5/14**	**0,36**
Chuva	3	2	**= 5/14**	**0,36**
Total	**5**	**9**		
	= 5/14	**= 9/14**		
	0,36	**0,64**		

- A probabilidade final deste cenário será:

$$\mathcal{P}(Não/Ensolarado) = \frac{0{,}40 \cdot 0{,}36}{0{,}36} = \mathbf{0,40}$$

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Exemplo – Classificação final

0,60

Maior probabilidade posterior!

0,40

Fonte: 6 passos fáceis para aprender o algoritmo Naive Bayes (com o código em Python)

Implementações computacionais

- O scikit-learn, no Python, nos oferece algumas implementações diferentes para o algoritmo Naïve Bayes:
  - **Gaussian Naive Bayes (GaussianNB)**
    - Variáveis contínuas (assumindo distribuição Normal)
  - **Multinomial Naive Bayes (MultinomialNB)**
    - Variável-alvo categórica multinomial
  - **Complement Naive Bayes (ComplementNB)**
    - Similar ao multinomial, porém apropriado para problemas desbalanceados.
  - **Bernoulli Naive Bayes (BernoulliNB)**
    - Variáveis categóricas binárias (dummy)
  - **Categorical Naive Bayes (CategoricalNB)**
    - Variáveis categóricas

Fonte: scikit-learn - Naïve Bayes