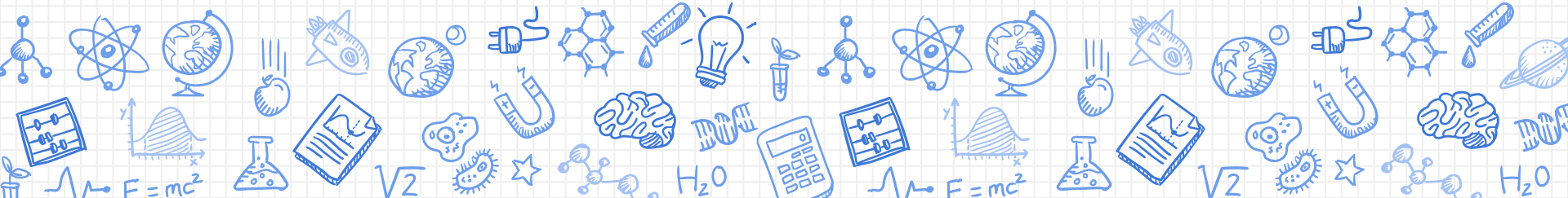
Análise de Dados e Inferência Estatística com Python

Aula de hoje!

- ✘ Introdução
- ✘ Naive Bayes
- ✘ Modelo

Introdução



Introdução

É uma técnica de classificação baseada no teorema de Bayes com uma suposição de independência entre os preditores.

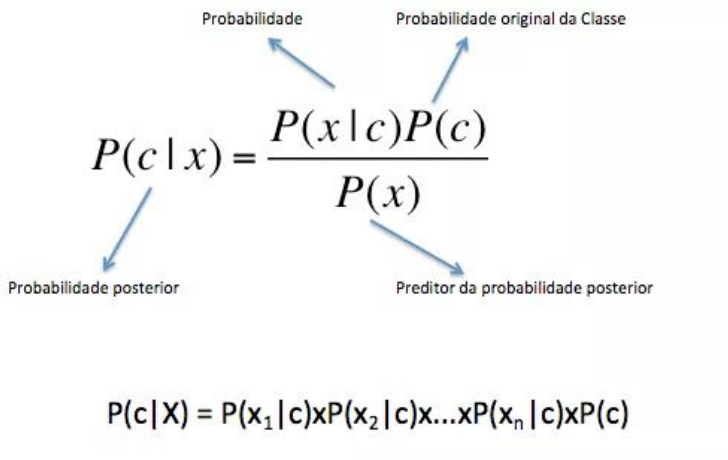
Em termos simples, um classificador Naive Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro recurso.

Por exemplo, um fruto pode ser considerado como uma maçã se é vermelho, redondo, e tiver cerca de 3 polegadas de diâmetro. Mesmo que esses recursos dependam uns dos outros ou da existência de outras características, todas estas propriedades contribuem de forma independente para a probabilidade de que este fruto é uma maçã e é por isso que é conhecido como 'Naive' (ingênuo).

Naive Bayes

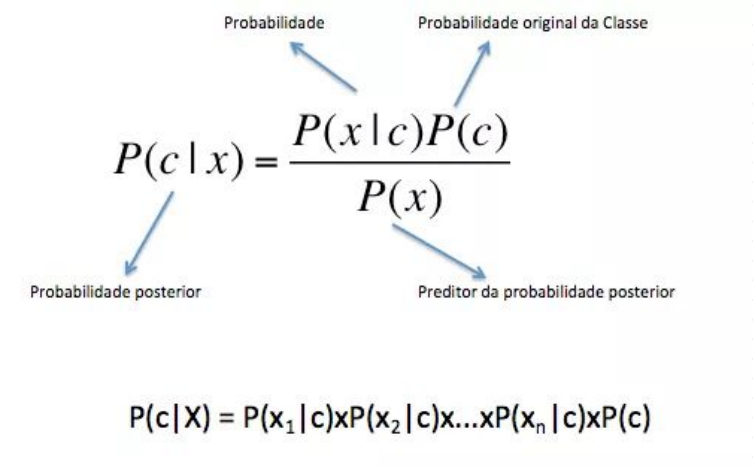
O modelo Naive Bayes é fácil de construir e particularmente útil para grandes volumes de dados. Além de simples, Naive Bayes é conhecido por ganhar de métodos de classificação altamente sofisticados.

O Teorema de Bayes fornece uma forma de calcular a probabilidade posterior P (C | X) a partir de P (C), P (x) e P (X | c). Veja a equação abaixo:



Naive Bayes

- P (c | x) é a probabilidade posterior da classe (c, alvo) dada preditor (x, atributos).
- P (c) é a probabilidade original da classe.
- P (x | c) é a probabilidade que representa a probabilidade de preditor dada a classe.
- P (x) é a probabilidade original do preditor.



Como o algoritmo Naive Bayes funciona?

Vamos entender isso usando um exemplo. Temos um conjunto de dados de treinamento de clima e da correspondente variável-alvo 'Play' (sugerindo possibilidades de jogar). Agora, precisamos classificar se os jogadores vão jogar ou não com base na condição meteorológica. Vamos seguir os passos abaixo para realizar a operação.

Passo 1: Converter o conjunto de dados em uma tabela de frequência.

Passo 2: Criar tabela de Probabilidade ao encontrar as probabilidades de tempo Nublado = 0,29 e probabilidade de jogar = 0,64.

Como o algoritmo Naive Bayes funciona?

TEMPO	"PLAY"
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 frequência		
Clima	Não	Sim
Nublado	0	4
Sol	3	2
Chuva	2	3
Total	5	9

Tabela de probabilidade				
Clima	Não	Sim		
Nublado	0	4	=4/14	0,29
Sol	3	2	=5/14	0,36
Chuva	2	3	=5/14	0,36
Total	5	9		
	=5/14	=9/14		
	0,36	0,64		

Como o algoritmo Naive Bayes funciona?

Passo 3: Agora, use a equação Bayesiana Naive para calcular a probabilidade posterior para cada classe. A classe com maior probabilidade posterior é o resultado da previsão.

Problema: Os jogadores irão jogar se o tempo está ensolarado. Esta afirmação está correta?

Como o algoritmo Naive Bayes funciona?

Podemos resolver isso usando o método discutido acima de probabilidade posterior.

P (Sim |Ensolarado) = P (Ensolarado | Sim) * P (Sim) / P (Ensolarado)

Aqui temos P (Ensolarado | Sim) = 3/9 = 0,33, P (Ensolarado) = 5/14 = 0,36, P (Sim) = 9/14 = 0,64

Agora, P (Sim | Ensolarado) = 0,33 * 0,64 / 0,36 = 0,60, que tem maior probabilidade.

Naive Bayes usa um método similar para prever a probabilidade de classe diferente com base em vários atributos. Este algoritmo é usado principalmente em classificação de texto e com os problemas que têm múltiplas classes.

Quais são os prós e contras de Naive Bayes?

Prós:

- É fácil e rápido para prever o conjunto de dados da classe de teste. Também tem um bom desempenho na previsão de classes múltiplas.
- Quando a suposição de independência prevalece, um classificador Naive Bayes tem melhor desempenho em comparação com outros modelos como regressão logística, e você precisa de menos dados de treinamento.
- O desempenho é bom em caso de variáveis categóricas de entrada comparada com variáveis numéricas. Para variáveis numéricas, assume-se a distribuição normal (curva de sino, que é uma suposição forte).

Quais são os prós e contras de Naive Bayes?

Contras:

- Se a variável categórica tem uma categoria (no conjunto de dados de teste) que não foi observada no conjunto de dados de treinamento, então o modelo irá atribuir uma probabilidade de 0 (zero) e não será capaz de fazer uma previsão. Isso é muitas vezes conhecido como “Zero Frequency”. Para resolver isso, podemos usar a técnica de alisamento. Uma das técnicas mais simples de alisamento é a chamada estimativa de Laplace.
- Por outro lado, Naive Bayes é também conhecido como um mau estimador, por isso, as probabilidades calculadas não devem ser levadas muito a sério.
- Outra limitação do Naive Bayes é a suposição de preditores independentes. Na vida real, é quase impossível ter um conjunto de indicadores que sejam completamente independentes.

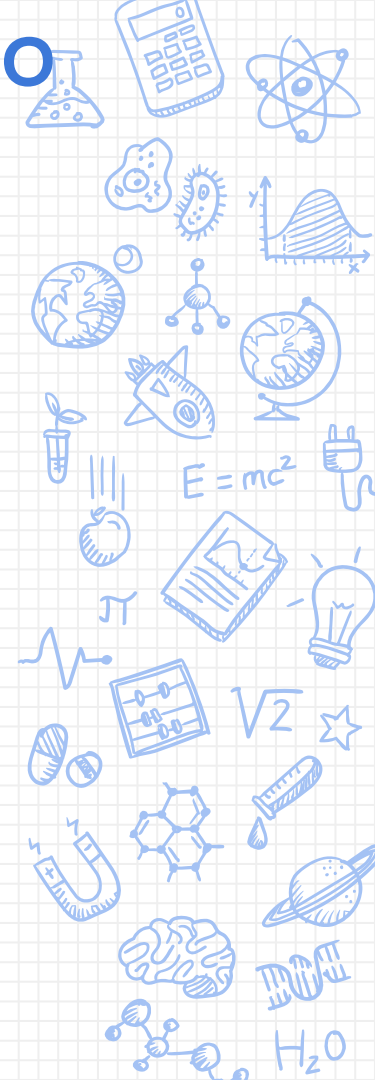
4 Aplicações do Algoritmo Naive Bayes

- Previsões em tempo real: Naive Bayes é um classificador de aprendizagem voraz e, com certeza, rápido. Assim, pode ser usado para fazer previsões em tempo real.
- Previsões multi-classes: Este algoritmo também é conhecido pela funcionalidade de previsão multi-classes. Aqui podemos prever a probabilidade de múltiplas classes das variáveis-alvo.
- Classificação de textos/Filtragem de *spam*/Análise de sentimento: classificadores Naive Bayes, utilizados principalmente em classificação de textos (devido a um melhor resultado em problemas de classes múltiplas e regra de independência), têm maior taxa de sucesso em comparação com outros algoritmos. Como resultado, é amplamente utilizado na filtragem de *spam* (identificar *spam*) e Análise de Sentimento (em análise de mídia social, para identificar sentimentos positivos e negativos dos clientes).
- Sistema de Recomendação: o classificador e a filtragem colaborativa Naive Bayes em conjunto constroem um sistema de recomendação que utiliza técnicas de *machine learning* e mineração de dados para filtrar a informação invisível e prever se um usuário gostaria de um determinado recurso ou não.

Passos para construir um modelo básico Naive Bayes em Python

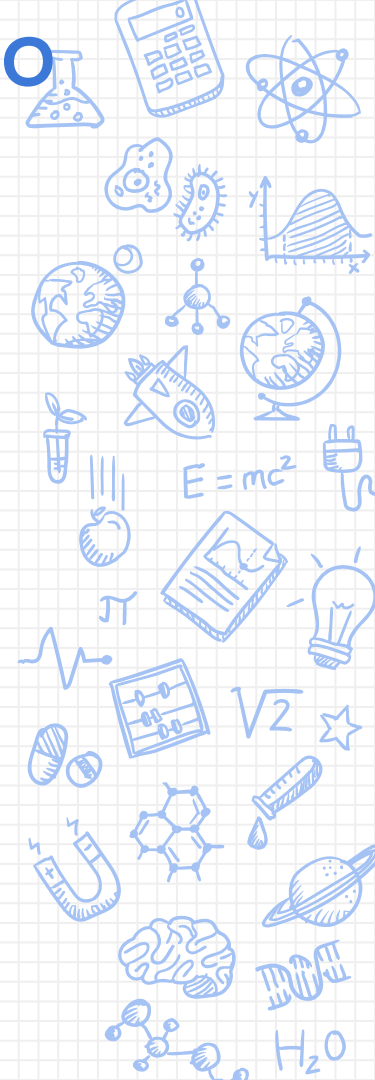
Mais uma vez, o s*cikit learn* (biblioteca python) vai ajudar a construir um modelo Naive Bayes em Python. Existem três tipos de modelo Naive Bayes sob a biblioteca do *scikit learn*:

Gaussian: É usado na classificação e assume uma distribuição normal.



Passos para construir um modelo básico Naive Bayes em Python

Multinomial: É usado para contagem discreta. Por exemplo, digamos que temos um problema de classificação de texto. Aqui podemos considerar tentativas de Bernoulli, que é um passo além e, em vez de “palavra que ocorre no documento”, temos “contar quantas vezes a palavra ocorre no documento”, você pode pensar nisso como “número de vezes que o número desfecho x_i é observado durante as n tentativas “.

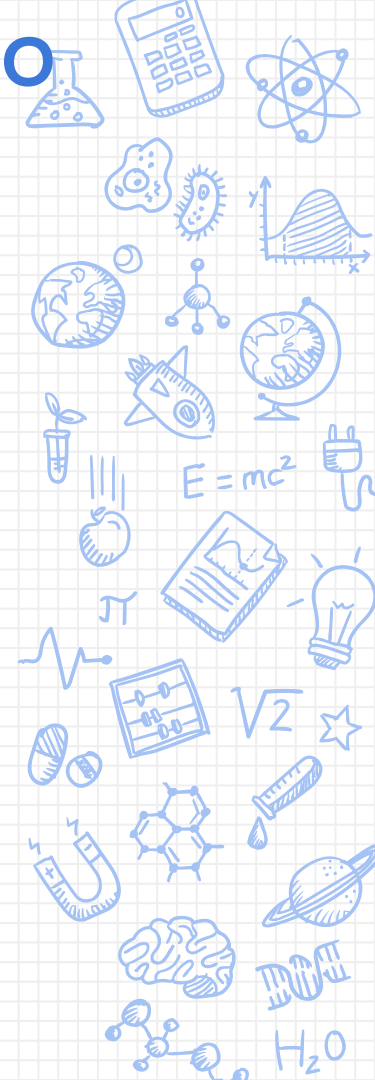


Passos para construir um modelo básico Naive Bayes em Python

Bernoulli: O modelo binomial é útil se os vetores são binários (ou seja zeros e uns). Uma aplicação seria de classificação de texto com um modelo de 'saco de palavras' onde os 1s e 0s são "palavra ocorre no documento" e "palavra não ocorre no documento", respectivamente.

Com base no seu conjunto de dados, você pode escolher qualquer um dos modelos acima discutidos.

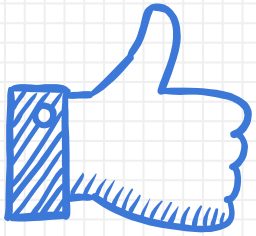
Vamos ver agora um exemplo de modelo de Gauss.



Dicas para melhorar o poder do modelo Naive Bayes

Aqui vão algumas dicas para melhorar o poder do Modelo Naive Bayes:

- Se as funções contínuas não têm distribuição normal, devemos usar a transformação ou métodos diferentes para convertê-las na distribuição normal.
- Se o conjunto de dados de teste tem problema de frequência zero, aplique a técnica de suavização "Laplace Correction" para prever a classe de conjunto de dados de teste.
- Remova características correlacionadas, já que as características altamente correlacionadas são votadas duas vezes no modelo e podem levar a um excesso de importância.
- Classificadores Naive Bayes têm opções limitadas para ajuste de parâmetros como alfa = 1 para alisamento, fit_prior = [Verdade | Falso] para saber a classe de probabilidades anteriores ou não e algumas outras opções. Recomenda-se focar no pré-processamento de dados e seleção de recursos.
- Você poderia pensar em aplicar alguma técnica de combinação de classificador como "ensembling", "bagging" e "boosting", mas na prática esses métodos não ajudariam, pois sua finalidade é reduzir a variância. Naive Bayes não tem variância para minimizar.



Obrigado!