

Aplicações estatísticas

Introdução

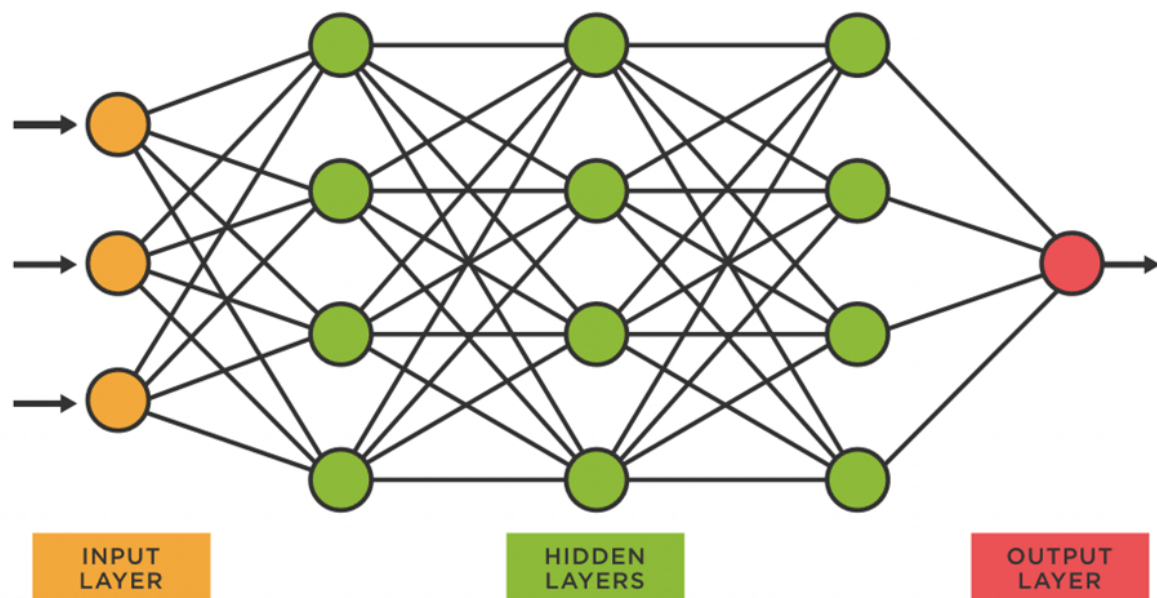
Como já vimos, temos inúmeras aplicações da Estatística no dia a dia e em casos de mercado. Em especial, existe uma gama de aplicações estatísticas em Machine Learning. Neste módulo, veremos 3 aplicações interessantes da Estatística e aprendizado de Máquina.

Objetivos da aula

- Introdução às redes Bayesianas
- MAB Thompson Sampling
- Introdução ao modelo de Markov

Resumo

Redes Bayesianas são um tipo de redes neurais, onde as variáveis aleatórias são os nós, e os arcos identificam as relações entre as variáveis. Elas são grafos que representam relações de probabilidade condicional, ou seja, a ocorrência de certas variáveis depende do estado de outra. Daí, para cada variável (nó) A que possui como pais (nós da camada anterior que se ligam a A) B_1, \dots, B_n , existe uma probabilidade condicional $P(A|B_1, \dots, B_n)$.



Fonte: https://www.tibco.com/sites/tibco/files/media_entity/2021-05/neutral-network-diagram.svg (acesso em 21/12/2022)

Um caso particular de uma rede bayesiana é o classificador Naive Bayes. Vamos ver como ele funciona!

Naive Bayes

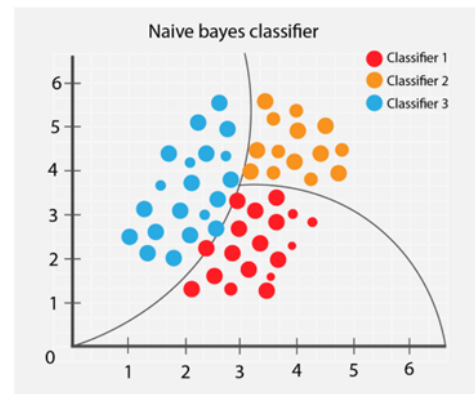


In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

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

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$



Fonte: https://miro.medium.com/max/1200/1*39U1Ln3tSdFqsfQy6ndxOA.png (acesso em 21/12/2022)

O termo “naive” quer dizer ingênuo em inglês, e representa uma característica importante do algoritmo: ele assume que as features são independentes entre si. Vamos ver um exemplo para melhor entendimento. Abaixo temos dados de pessoas que visitaram uma concessionária de veículos e, baseado na cor dos carros, decidiu se comprariam ou não o carro.

Cor do Carro	Comprar
Azul	Sim
Preto	Não
Branco	Não
Preto	Sim
Preto	Sim
Branco	Não
Azul	Sim
Branco	Não

Branco	Sim
Azul	Não
Preto	Não
Branco	Não
Preto	Sim
Preto	Sim
Branco	Não
Azul	Sim
Preto	Sim

Fonte: autoral

Daí, usando essas informações, vamos calcular a quantidade de vezes que cada cor de carro foi comprada ou não e, após isso, vamos descobrir a probabilidade de um carro ter sido comprado ou não e a probabilidade de cada cor ter sido mostrada para os clientes.

Cor do Carro	Não	Sim	
Azul	1	3	25,00%
Branco	4	1	31,25%
Preto	2	5	43,75%
	43,75%	56,25%	

Fonte: autoral

Por exemplo, a probabilidade de um carro mostrado ter sido comprado foi de 56,25% e a probabilidade de um carro branco ter sido mostrado foi de 31,25%.

O próximo passo será usar o teorema de Bayes para calcular a probabilidade a posteriori de um carro ser vendido dado sua cor. Vamos usar a cor Azul como exemplo:

$$P(\text{Sim} \mid \text{Azul}) = P(\text{Azul} \mid \text{Sim}) * P(\text{Sim}) / P(\text{Azul})$$

Usando a tabela de probabilidade, temos que:

$$P(\text{Azul} \mid \text{Sim}) = 3/9 = 33\%$$

$$P(\text{Sim}) = 56,25\%$$

$$P(\text{Azul}) = 25\%$$

Daí, o resultado seria

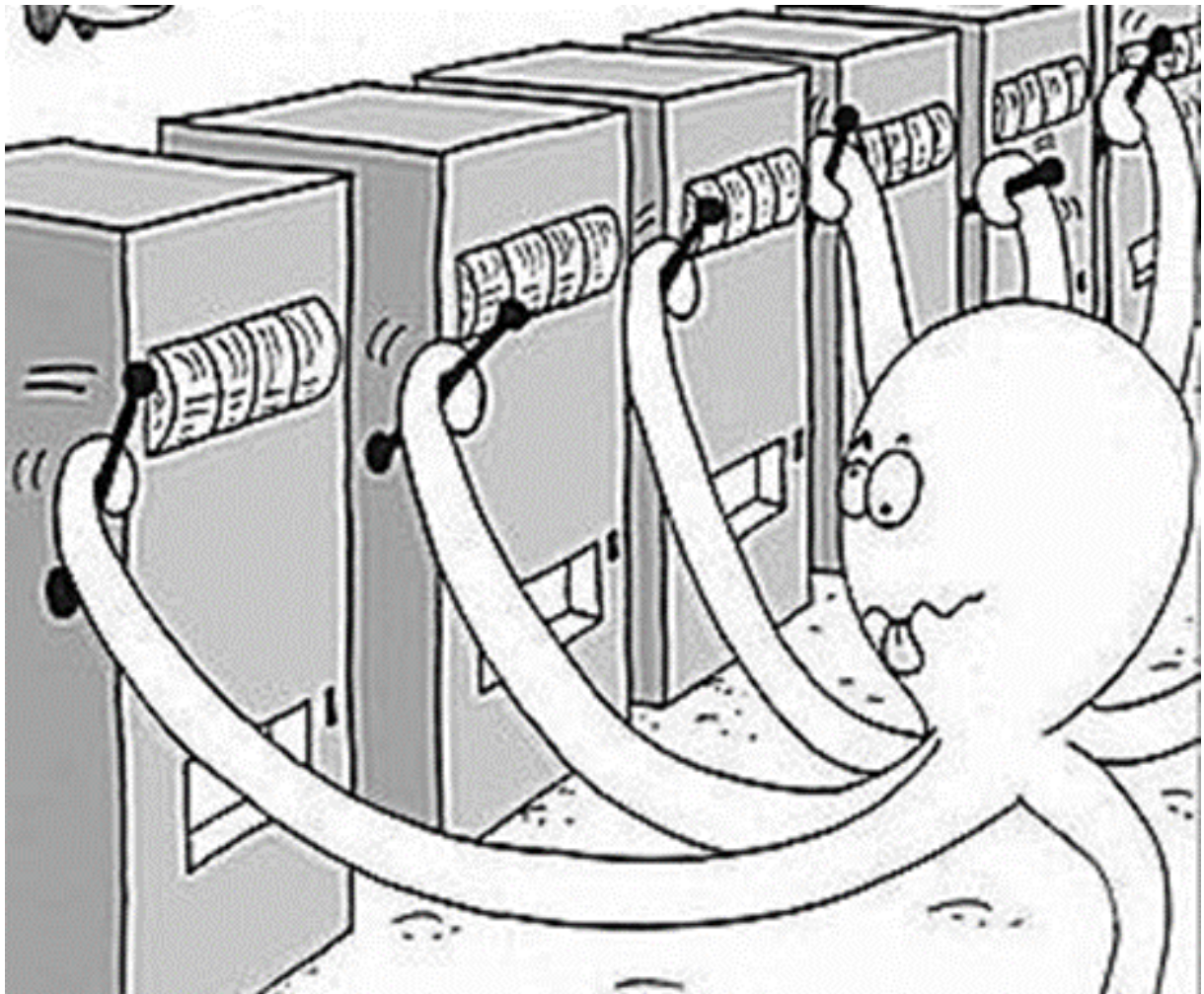
$$P(\text{Sim} \mid \text{Azul}) = 33\% * 56,25\% / 25\% = 74,25\%$$

$$P(\text{Não} \mid \text{Azul}) = 100\% - 74,25\% = 25,75\%$$

Daí, o algoritmo classificaria um carro azul dentro da classe Sim, pois tem uma maior probabilidade de ocorrência.

O algoritmo Thompson Sampling é um algoritmo de reinforcement learning do tipo MAB (Multi Armed Bandit). Um MAB tenta resolver um problema onde você possui várias opções de escolha e tem como objetivo maximizar a sua recompensa obtida numa escolha vencedora. Surgiu com o caça-níquel e hoje utilizamos em muitas aplicações de mercado.

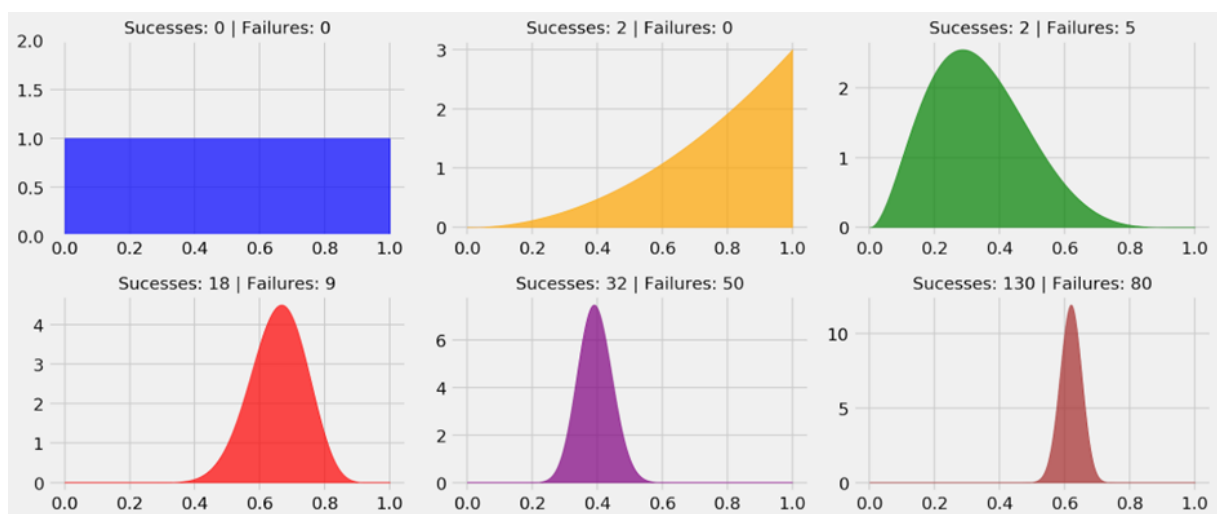
A ideia é que, uma vez identificada a opção que te dá recompensas mais vezes, você passa a majoritariamente escolher essa opção, até que uma outra opção se mostre melhor.



Fonte: <https://blogs.mathworks.com/images/loren/2016/multiarmedbandit.jpg> (acesso em 21/12/2022)

O Thompson Sampling é um MAB que utiliza estatística Bayesiana, numa combinação de uma priori com distribuição de Bernoulli e uma posteriori com distribuição Beta(a , b). Nesse caso,

- a = número de vezes que se obteve uma recompensa na escolha (sucessos) + 1
- b = número de vezes que não se obteve uma recompensa na escolha (fracassos) + 1



Fonte: https://gdmarmarola.github.io/assets/img/posts/ts_for_mab_cover.jpg (acesso em 21/12/2022)

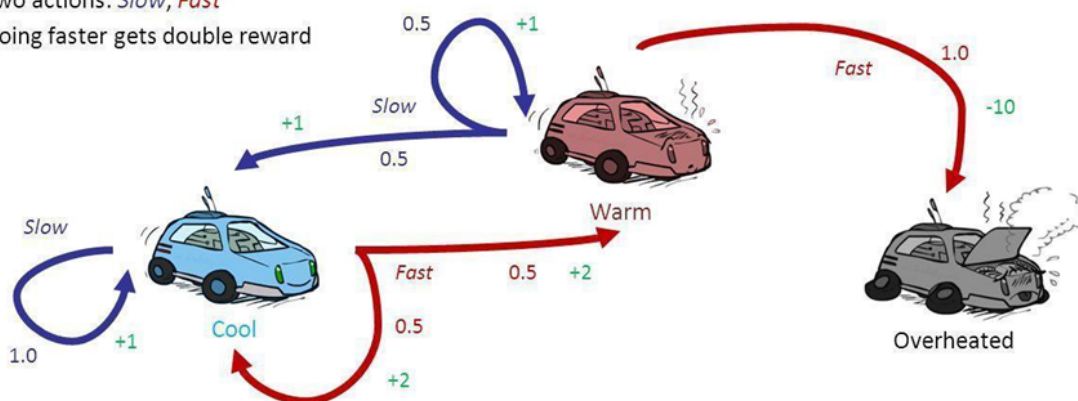
Portanto, cada opção vai ter sua própria distribuição Beta com seus próprios a e b. A partir daí, nós calculamos uma amostra de tamanho 1 de cada distribuição e a que apresentar o maior resultado será a opção escolhida.

As opções com uma taxa de sucesso maior vão ter uma maior probabilidade de terem amostras maiores e, portanto, serão mais escolhidas.

Uma cadeia de Markov é um modelo que explica uma sequência de eventos onde o próximo evento só depende da ocorrência atual, ou o estado atual. Da mesma forma, a ocorrência atual só dependeu da última ocorrência, e não de toda a sequência anterior de eventos ocorridos. A cada ida de um estado a outro há uma probabilidade embutida.

Example: Racing

- A robot car wants to travel far, quickly
- Three states: *Cool*, *Warm*, *Overheated*
- Two actions: *Slow*, *Fast*
- Going faster gets double reward

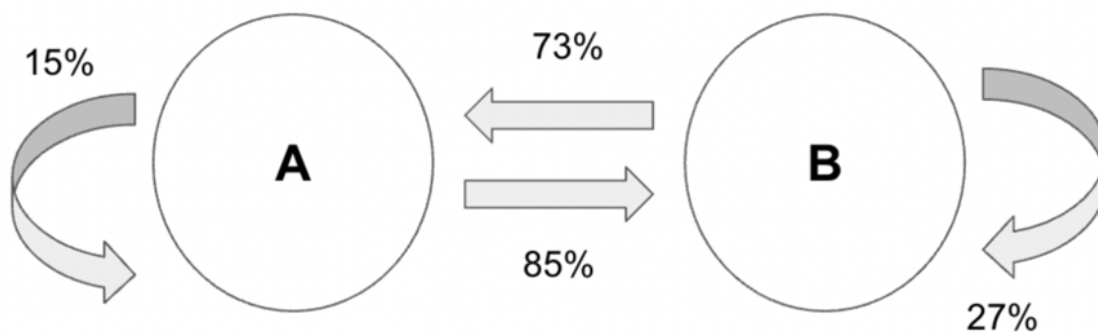


Fonte: https://miro.medium.com/max/1400/1*Uh11rrUKKsHLLRmmv0ss2w.jpeg (acesso em 21/12/2022)

Por exemplo, vamos dizer que você trabalha num escritório no centro de Nova York. Todos os dias você sai para almoçar e precisa decidir entre dois restaurantes (A e B) qual vai ser o escolhido do dia, seguindo a seguinte regra:

- Se comeu no restaurante A hoje, então amanhã você tem 85% de chance de comer no restaurante B
- Se comeu no restaurante B hoje, então amanhã você tem 73% de chance de comer no restaurante A

Assim ficaria o grafo dos estados desse modelo de Markov



Fonte: autoral

Como aplicar na prática o que aprendeu

Identifique algum problema de classificação com dados que você tem, como o visto no exemplo da videoaula, e tente aplicar o algoritmo Naive Bayes. Veja o quanto ele acerta para os seus dados!

Conteúdo bônus

Tópicos avançados

Nos últimos anos, muitas redes neurais têm utilizado otimização bayesiana para achar os melhores hiperparâmetros de cada modelo. É o caso do [Prophet](#), um projeto open source do Facebook para previsão de séries temporais.

Caro estudante, você consegue acessar os códigos utilizados na disciplina no link a seguir: <https://github.com/FaculdadeDescomplica/Statistics-for-Data-Science>

Referência Bibliográfica

Cadeias de Markov com Python. Disponível em: <<https://leandrocruvinel.medium.com/cadeias-de-markov-com-python-ef27b3f21fc7>>. (Acesso em 21/12/2022)

Vooo – Insights. Disponível em: <<https://www.vooo.pro/insights/6-passos-faceis-para-aprender-o-algoritmo-naive-bayes-com-o-codigo-em-python/>>. (Acesso em 21/12/2022)