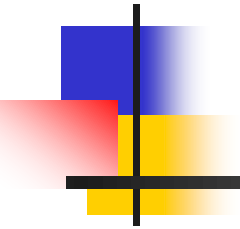


Regressão Bayesiana

Teorema de Bayes

Resultado do teste	Diagnóstico		Total
	Doente	Não Doente	
Positivo	265	47	**312**
Negativo	11	50	**61**
Total	**276**	**97**	**373**

$$s = P(+|D) = \frac{265}{276} = 0{,}96 \quad \text{ou} \quad 96\%$$

$$e = P(-|S) == \frac{50}{97} = 0{,}515 \quad \text{ou} \quad 51{,}5\%$$

Teorema de Bayes

Acertos :

- Entre os positivos $$P(D|+) = \frac{P(D \cap +)}{P(+)}$$

Valor de Predição Positiva (VPP)

- Entre os negativos $$P(S|-) = \frac{P(S \cap -)}{P(-)}$$

Valor de Predição Negativa (VPN)

Teorema de Bayes

$$P(D|+) = \frac{P(D \cap +)}{P(+)} = \frac{P(D) \times P(+|D)}{P[(+ \cap D) \cup (+ \cap S)]}$$

$$P(D|+) = \frac{P(D) \times P(+|D)}{P(D) \times P(+|D) + P(S) \times P(+|S)}$$

Regra de Bayes

Teorema de Bayes

Probabilidade *a priori*

"Verossimilhança"

$$P(S|-) = \frac{P(S) \times P(-|S)}{P(S) \times P(-|S) + P(D) \times P(-|D)}$$

Probabilidade *a posteriori*

A regra de Bayes mostra como alterar as probabilidades a priori tendo em conta novas evidências de forma a obter probabilidades a posteriori.

Regressão Linear

- Como já vimos o modelo assume que a variável de resposta (y) é uma combinação linear de pesos multiplicada por um conjunto de variáveis preditoras (x). A fórmula completa também inclui um termo de erro para explicar o ruído de amostragem aleatória. Por exemplo, se temos dois preditores, a equação é:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Dados : $\mathbf{y} = (y_1, y_2, ..., y_n)$

$\mathbf{x1} = (x1_1, x1_2, ..., x1_n)$

$\mathbf{x2} = (x2_1, x2_2, ..., x2_n)$

Parâmetros: $\boldsymbol{\theta} = (\alpha, \beta_1, \beta_2, \sigma^2)$

- O que obtemos da regressão linear freqüentista é uma única estimativa para os parâmetros do modelo com base apenas nos dados de treinamento. Nosso modelo é completamente informado pelos dados: nessa visão, tudo o que precisamos saber para o nosso modelo é codificado nos dados de treinamento que temos disponíveis.

Regressão Linear Bayesiana

- No ponto de vista bayesiano, formulamos regressão linear usando distribuições de probabilidade em vez de estimativas pontuais. A resposta, y, não é estimada como um valor único, mas é assumida como sendo derivada de uma distribuição de probabilidade.

- O modelo estatístico não é mais somente $P(y|\theta)$ e sim $P(y, \theta)$, a distribuição conjunta dos dados y e dos parâmetros θ

- As estimativas para θ não serão somente valores, mas sim uma distribuição de probabilidades.

Regressão Linear Bayesiana

- Como obter $P(\theta|\mathbf{y})$?

$$P(\theta|y) = \frac{P(\theta,y)}{P(y)}$$

  - Pela Regra de Bayes

Verossimilhança **Probabilidade *a priori***

$$P(\theta|y) = \frac{P(\theta,y)}{P(y)} = \frac{P(y|\theta) \times P(\theta)}{P(y)}$$

Probabilidade *a posteriori*

Regressão Linear Bayesiana

- $P(\theta)$ expressa a incerteza sobre θ antes de observarmos os dados y que dependem dele (a priori) .

- $P(\theta|y)$ expressa a incerteza sobre θ depois de observarmos os dados y que dependem dele (a posteriori).

- De posse de $P(\theta|y)$, podemos examinar qualquer aspecto de θ (média, variância, percentis, probabilidade de assumir determinados valores, etc.) ("Full Posterior Distribution")

Passos para estimação

- Escolher um modelo probabilístico para P(y| θ) – a função de verossimilhança;
- Escolher um modelo probabilístico para P(θ) – a distribuição a priori ;
- Aplicar a regra de Bayes e calcular P(θ |y).

Exemplo : modelo Beta-binomial

- Experimento para estimar proporção de cura com uma nova terapia em bovinos:

n = 16 animais

yi = 1, se o animal for curado

0, caso contrário. i = 1,2,3, ... , 16

y é o total de animais curados (y1 + y2 + ... +y16)

θ é a proporção de cura $(0 \leq \theta \leq 1)$

para $P(\theta)$ – a distribuição a priori ;

Aplicar a regra de Bayes e calcular $P(\theta \mid y)$.

Modelo para $P(y|\theta)$: $y \sim$ Binomial $(16, \theta)$

$$P(y \mid \theta) = \binom{16}{y} \theta^{y} (1-\theta)^{(16-y)}$$

Exemplo : modelo Beta-binomial

Modelo para P(θ) : $\theta \sim$ **Beta** (α , β)

hiperparâmetros

Cálculo da *posteriori* P(θ|**y**)

$$P(\theta \mid y) = \frac{P(y \mid \theta)\, P(\theta)}{P(y)} = \frac{P(y \mid \theta)\, P(\theta)}{\int_0^1 P(y,\theta) d\theta} = \frac{P(y \mid \theta)\, P(\theta)}{\int_0^1 P(y \mid \theta)\, P(\theta) d\theta}$$

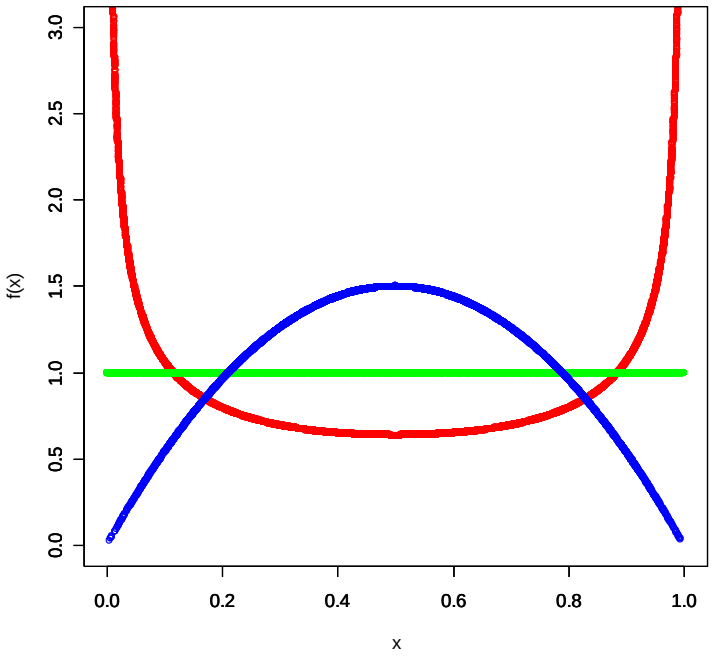
$$P(\theta \mid y) \propto \theta^{y+\alpha-1}(1-\theta)^{16-y+\beta-1}$$

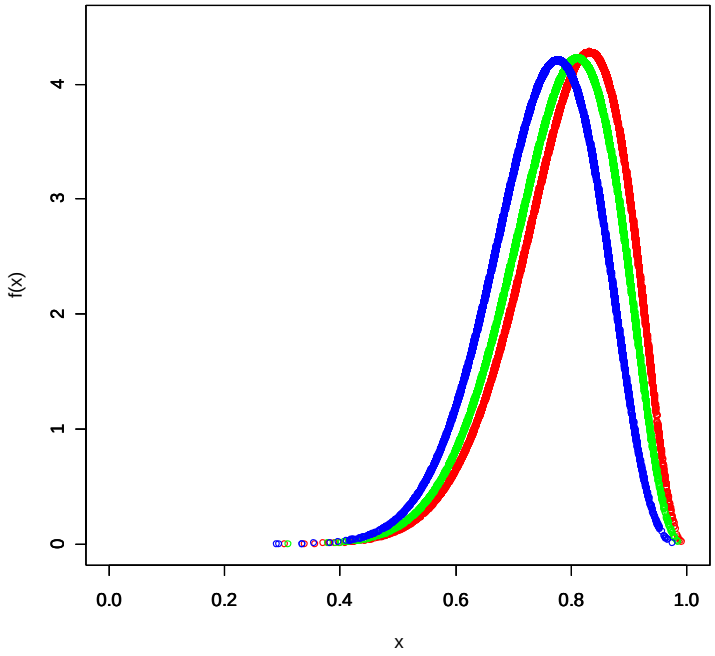
$\theta|y \sim$ **Beta** $(y+\alpha\ ,\ 16-y+\beta)$

Exemplo : modelo Beta-binomial

Suponha que y = 13 (13/16 = 0.8125)
Priori´s : Beta (0.5 , 0.5), Beta (1,1) e Beta (2,2)

Posteriori´s : Beta (13.5 , 3.5), Beta (14,4) e Beta (15,5)





Exemplo : modelo Beta-binomial

Priori	**Quantis *a posteriori***			**Média a *posteriori***
	0.025	**0.500**	**0.975**	
Beta (0.5,0.5)	0.544	0.758	0.909	0.250
Beta (1 , 1)	0.566	0.788	0.932	0.222
Beta (2 , 2)	0.579	0.806	0.944	0.206

Intervalo de Credibilidade de 95%

Cadeia de Markov

Se o cálculo de $P(\theta|\mathbf{y})$ é difícil, vamos simular muitas amostras de $P(\theta|\mathbf{y})$ e usá-las para estimar as características de $\boldsymbol{\theta}$ que nos interessem .

Na regressão bayesiana, a cadeia de Markov é usada para gerar amostras da distribuição a posteriori dos parâmetros do modelo, permitindo estimativas bayesianas robustas e a realização de inferências sobre os parâmetros do modelo. Essas amostras podem então ser usadas para calcular estimativas pontuais, intervalos de credibilidade e realizar comparações entre modelos, entre outras análises estatísticas.

Vantagens

- Incorporação de Incerteza: A capacidade de modelar e quantificar incertezas nos parâmetros do modelo e previsões, o que é útil em situações onde a incerteza é intrínseca aos dados.

- Flexibilidade na Especificação de Priors: A regressão bayesiana permite que os pesquisadores incorporem conhecimento prévio sobre os parâmetros do modelo na forma de distribuições priori, o que pode levar a estimativas mais robustas em conjuntos de dados pequenos.

- Manejo Natural de Dados Escassos: A abordagem bayesiana é particularmente útil quando os dados são escassos, pois permite que as estimativas sejam baseadas tanto nos dados observados quanto nas distribuições priori.

- Incorporação de Informações Contextuais: É possível incorporar informações contextuais relevantes nos modelos bayesianos por meio das distribuições priori, o que pode levar a estimativas mais precisas.

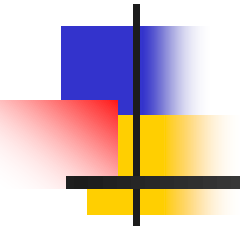
Limitações

- Complexidade Computacional: A inferência bayesiana pode ser computacionalmente intensiva, especialmente para modelos complexos ou grandes conjuntos de dados, o que pode limitar sua escalabilidade.

- Especificação de Priors: A escolha de distribuições priori pode ser subjetiva e influenciar significativamente os resultados do modelo, exigindo cuidado na sua especificação.

- Interpretação de Modelos Complexos: A interpretação de modelos bayesianos complexos pode ser desafiadora devido à presença de múltiplos parâmetros e interações, tornando necessário o uso de técnicas de diagnóstico adequadas.

- Dificuldade na Comunicação dos Resultados: A comunicação dos resultados de modelos bayesianos para audiências não técnicas pode ser mais desafiadora devido à natureza probabilística das estimativas



Regressão kNN

Algoritmo K-Nearest Neighbors

- O kNN é um simples e popular algoritmo de aprendizado de máquina que pode ser utilizado para problemas de classificação e regressão.

- Dado um elemento sem rótulo conhecido, seu rótulo será definido por uma análise de seus k vizinhos mais próximos.

- Vizinhos são basicamente os elementos mais próximos (ou semelhantes) a um dado elemento. Existem diferentes métricas para calcular a distância entre pontos.

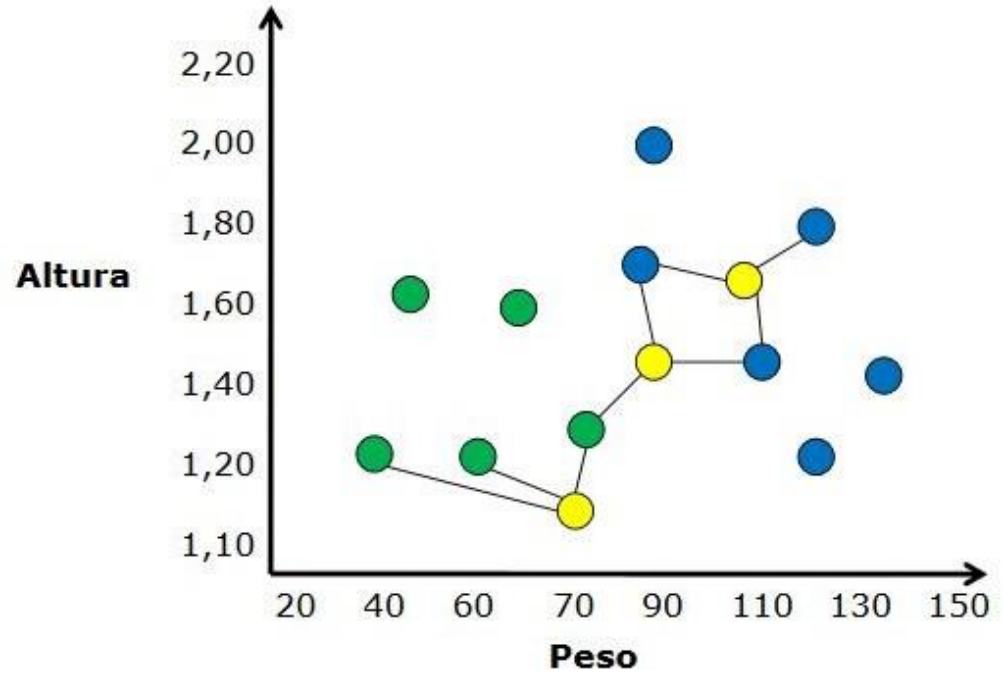
A distância Euclidiana é uma das mais simples e populares;

$$d(a,b) = \sqrt{\sum_{1}^{n}(a_i - b_i)^2}$$

Algoritmo K-Nearest Neighbors

- Uma vez calculada a distância o passo seguinte é encontrar os K-vizinhos mais próximos.

- Os valores dos vizinhos mais próximos ajuda a encontra o valor procurado para o ponto em questão.



Algoritmo K-Nearest Neighbors

- É necessário normalizar os dados para evitar que medidas de distância sejam monopolizadas por um atributo.
- Exemplo:
  - Altura: vai de 1,20 a 2,10
  - Peso vai de 40 a 150
  - Salário vai de 800 a 20.000

Como escolher o valor de k

- k na regressão kNN desempenha um papel crucial na determinação da precisão do modelo.
- Escolher o valor adequado de k é essencial para obter resultados confiáveis e generalizáveis.
- Um k baixo leva a um modelo com baixo viés e alta variância, o que pode levar ao sobreajuste.
- Um k alto resulta em um modelo com alto viés e baixa variância, o que pode levar ao subajuste.
- Deve-se experimentar alguns valores para encontrar o valor de k que gera as previsões mais precisas com o menor número de erros.

Não Linearidade na Regressão kNN

- Diferentemente de modelos lineares, como regressão linear, a regressão kNN pode capturar padrões complexos e não lineares nos dados.
- Devido à sua natureza baseada em vizinhos, a regressão kNN é altamente flexível e não impõe uma estrutura específica aos dados.

- Como a regressão kNN considera os valores de saída dos vizinhos mais próximos para prever o valor de saída de uma instância, ela pode capturar padrões complexos, como curvas e superfícies não lineares.

Vantagens

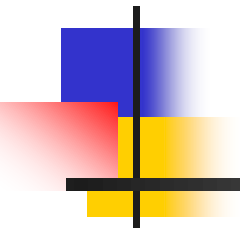
- Simples. O kNN é fácil de implementar devido à sua simplicidade e acurácia. Como tal, é frequentemente um dos primeiros classificadores que um cientista de dados aprenderá.

- Adaptável. Assim que novas amostras de treinamento são adicionadas a seu conjunto de dados, o algoritmo kNN ajusta suas previsões para incluir os novos dados de treinamento.

- Facilmente programável. o kNN requer apenas alguns hiperparâmetros — um valor de k e uma métrica de distância. Isso o torna um algoritmo razoavelmente descomplicado.

Limitações

- Difícil de redimensionar. Como o kNN ocupa muita memória e armazenamento de dados, aumenta as despesas associadas ao armazenamento. Essa dependência da memória também significa que o algoritmo é computacionalmente intensivo, o que, por sua vez, consome muitos recursos.
- A maldição da dimensionalidade. Fenômeno que ocorre na ciência da computação, em que um conjunto fixo de exemplos de treinamento é desafiado por um número crescente de dimensões e pelo aumento inerente de valores de recursos nessas dimensões. Em outras palavras, os dados de treinamento do modelo não conseguem acompanhar a dimensionalidade em evolução do hiperespaço.
- Superajuste. O valor de k terá impacto no comportamento do algoritmo. Isso pode acontecer especialmente quando o valor de k é muito baixo. Valores mais baixos de k podem superajustar os dados, enquanto valores mais altos de k “suavizarão” os valores de previsão porque o algoritmo calcula a média dos valores em uma área maior.



Persistência de Modelos

Definições

- Persistência de modelos refere-se ao processo de salvar modelos treinados em um formato que pode ser armazenado em disco para uso futuro.
- Por que é importante persistir modelos?
  - A persistência de modelos permite que os modelos treinados sejam reutilizados sem a necessidade de retreinamento, economizando tempo e recursos computacionais.
  - É útil para implantar modelos em produção, compartilhar modelos com outras pessoas e reproduzir resultados de experimentos.

- Benefícios da persistência de modelos.
  - Flexibilidade: Os modelos podem ser salvos e carregados em diferentes ambientes ou plataformas.
  - Reprodutibilidade: Permite reproduzir resultados exatos usando os mesmos modelos salvos.
  - Escalabilidade: Facilita o compartilhamento de modelos em grande escala em sistemas distribuídos.

Definições

- Em Python, os modelos podem ser persistidos usando bibliotecas como joblib, pickle, HDF5, entre outros.
- Joblib é uma biblioteca popular para persistência de modelos em Python, especialmente modelos scikit-learn.
  - É eficiente para objetos grandes e suporta paralelização para acelerar o processo de salvamento.
- A biblioteca pickle é uma opção nativa em Python para serialização de objetos.
- É flexível e suporta uma ampla gama de objetos Python, mas pode ser mais lenta para objetos grandes.
- Outros métodos de persistência.
  - HDF5: Adequado para modelos com muitos parâmetros ou dados grandes.
  - JSON: Útil para modelos simples ou estruturas de dados serializáveis.