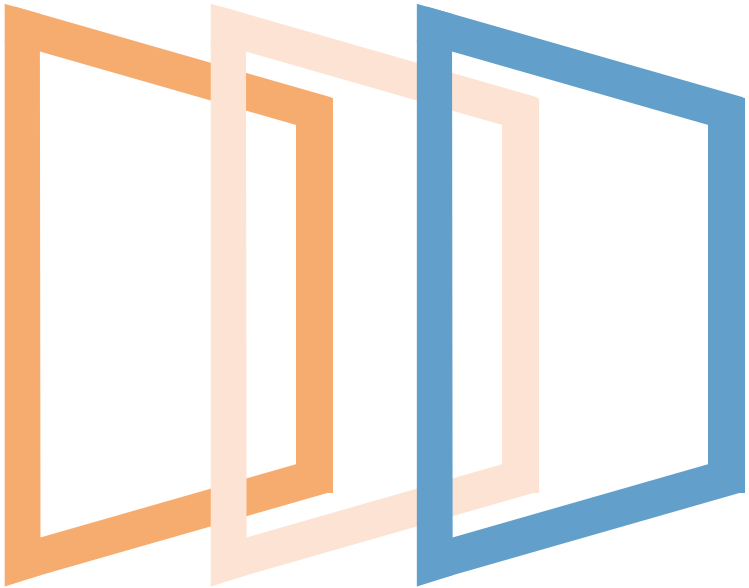
Trilhando Caminhos em Ciência de Dados



Thaís Ratis

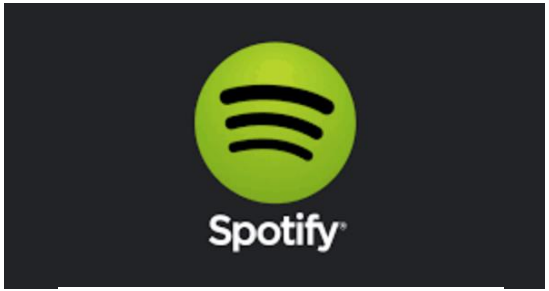
Práticas Tecnológicas, 02.09.2023

minsait

An Indra company

Conteúdo programático

1. O que é IA?
2. O que é ML?
3. Aprendizado supervisionado
4. Aprendizado não supervisionado

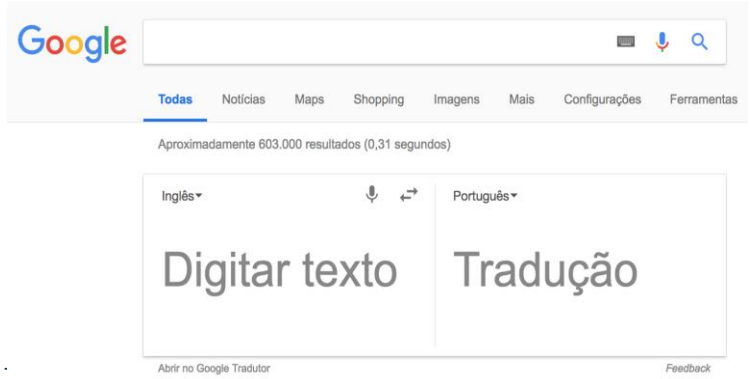

Spotify


NETFLIX


Patrocinado
Criar anúncio
Wine.com.br
wine.com.br
Já conhece a Wine? Não? Confira já nossos rótulos! Os melhores vinhos a um clique de dist...
AMARO
AMARO
ww.amaro.com
As últimas tendências da vez a um clique do seu guarda-roupas! Entrega expressa, em até 6x.

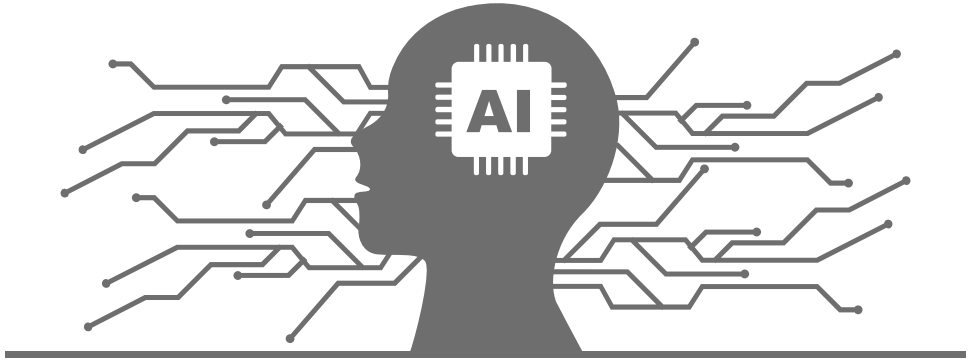

Google
Brasil
eu ja vi
eu ja vivo enjoado
eu ja vi a sarça ardente
eu ja vi o diabo
eu ja vi um anjo
eu ja viajei por todo o universo
eu ja virei seu fã
eu ja vi esse filme
eu ja vivi um sonho
eu ja vi tudo
eu ja vi o proprio lobo na pele de um cordeiro
Pesquisa Google
Estou com sorte


Who is this?
Who is this?
Who is this?


Google
Todas
Notícias
Maps
Shopping
Imagens
Mais
Configurações
Ferramentas
Aproximadamente 603.000 resultados (0,31 segundos)
Inglês
Português
Digitar texto
Tradução
Abrir no Google Tradutor
Feedback


Messages
Mail
What can I help you with?

O que é IA?



01

O que é inteligencia artificial?

“A inteligência artificial (IA) é um campo de estudo da ciência da computação que se concentra no desenvolvimento de sistemas e programas capazes de executar tarefas que normalmente exigiriam inteligência humana. A IA visa criar máquinas que possam simular certos aspectos da inteligência humana, como aprendizado, raciocínio, resolução de problemas, reconhecimento de padrões, compreensão da linguagem natural e tomada de decisões.” (ChatGPT)

Inteligência Artificial

Pensando como Humano

Pensando Racionalmente

Agindo como Humano

Agindo Racionalmente

Pensando como Humano

Introspecção

Ato pelo qual o sujeito observa os conteúdos de seus próprios estados mentais, tomando consciência deles. Dentre estes conteúdos, destacam-se as crenças, memórias, intenções, emoções e pensamentos em geral.

Intuição

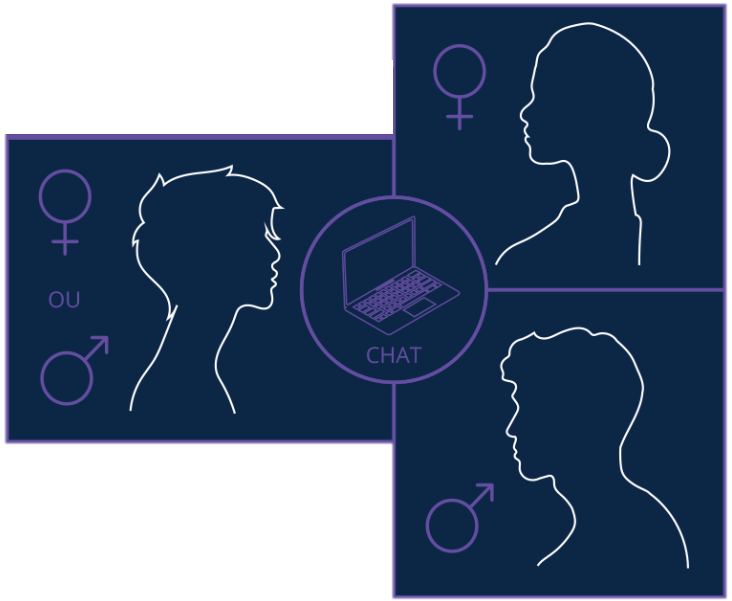
Em psicologia, é o processo pelo qual os humanos passam, mesmo que involuntariamente e inconscientemente, para chegar a uma conclusão sobre algo

Pensando como Humano

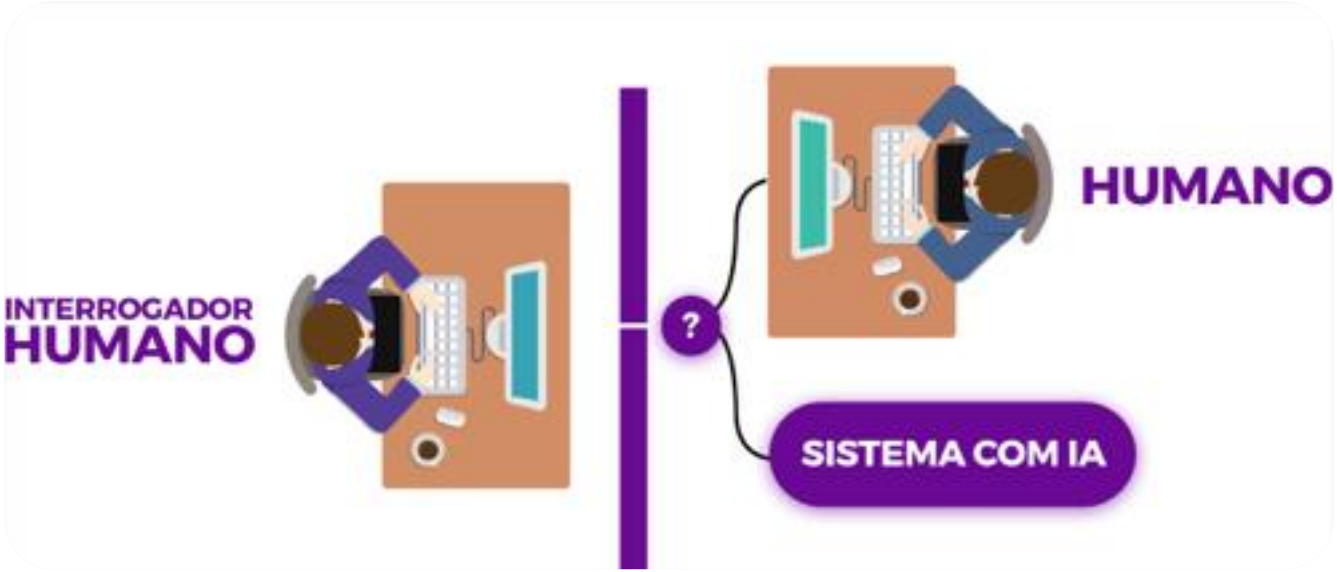
Aquisição de Conhecimento

Como seres humanos, temos a capacidade de adquirir conhecimento, aprender com experiências passadas, compreender conceitos abstratos, tomar decisões com base em informações limitadas e adaptar-nos a diferentes situações. A inteligência artificial busca replicar esses aspectos da inteligência humana por meio do desenvolvimento de algoritmos, modelos e sistemas computacionais.

Agindo como Humano



Agindo como Humano



Agindo como Humano

1. **Compreensão e resposta à linguagem natural**: A IA pode ser projetada para entender e responder à linguagem humana de maneira semelhante a um ser humano. Isso inclui habilidades como processamento de texto, reconhecimento de fala e geração de linguagem natural.
2. **Reconhecimento facial e emocional**: A IA pode ser treinada para reconhecer expressões faciais e emoções humanas. Isso permite que ela interprete e responda aos sinais emocionais das pessoas, tornando a interação mais natural e empática.
3. **Aprendizado e adaptação**: A IA pode ser programada para aprender com dados e experiências passadas e se adaptar a novas situações. Algoritmos de aprendizado de máquina e redes neurais podem permitir que a IA melhore seu desempenho e tome decisões com base em informações disponíveis.
4. **Tomada de decisões e raciocínio**: A IA pode ser projetada para tomar decisões com base em informações limitadas, assim como os humanos. Algoritmos de tomada de decisão e sistemas especialistas podem permitir que a IA analise dados, avalie diferentes opções e escolha a melhor ação a ser tomada.

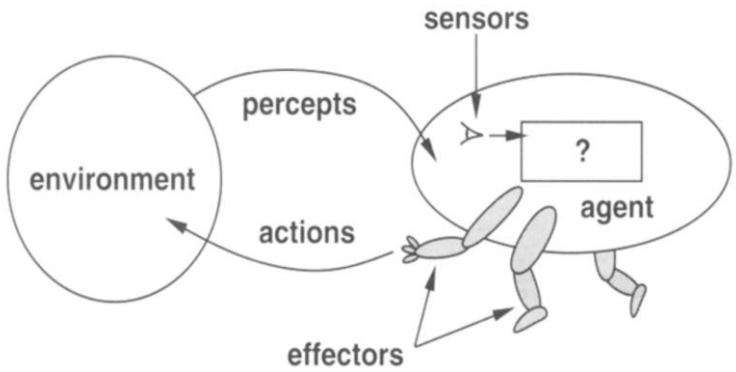
Pensando Racionalmente

- O filósofo grego Aristóteles foi um dos primeiros a codificar o “pensamento correto”.
- Silogismo – forneceu padrões para estruturas de argumentos que resultam em conclusões corretas a partir de premissas corretas.
- “Sócrates é um homem; Todo homem é mortal; portanto, Sócrates é mortal.”

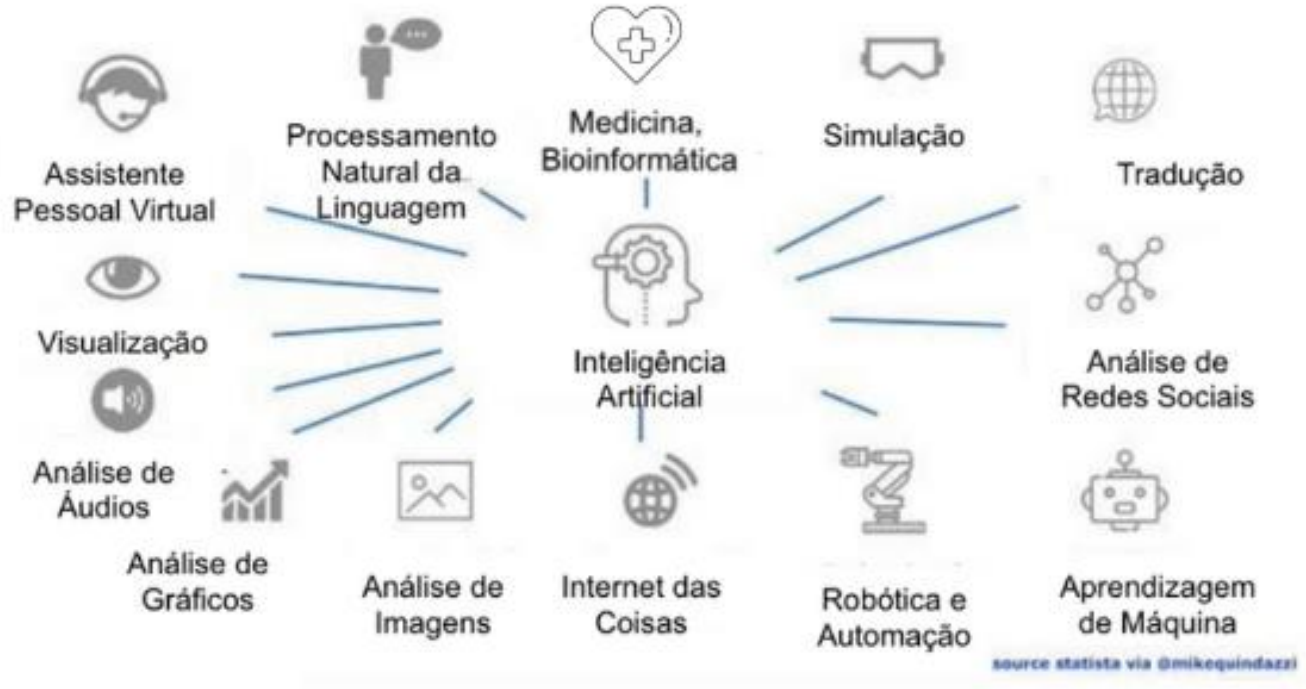
Agindo Racionalmente

Agir racionalmente significa agir de modo a alcançar seus objetivos, dada suas crenças.

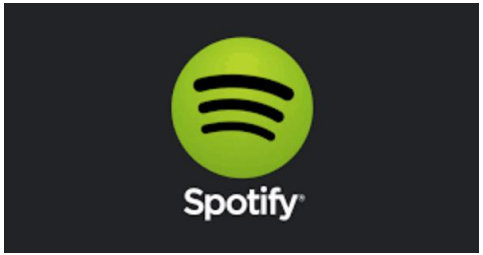
IA pode ser definida como o estudo de "agentes inteligentes": qualquer dispositivo que percebe seu ambiente através de seus sensores, e promove ações através de seus atuadores, de modo que maximizem suas chances de sucesso em algum objetivo.



Aplicações


Assistente Pessoal Virtual
Processamento Natural da Linguagem
Medicina, Bioinformática
Simulação
Tradução
Visualização
Inteligência Artificial
Análise de Redes Sociais
Análise de Áudios
Análise de Gráficos
Análise de Imagens
Internet das Coisas
Robótica e Automação
Aprendizagem de Máquina
source statista via @mikequindazzi

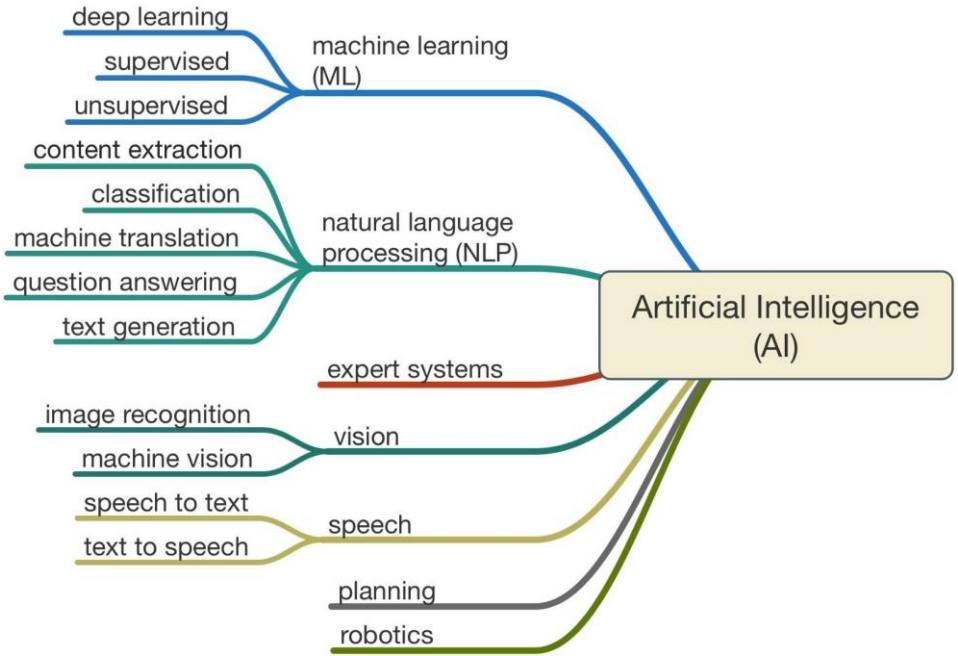
Aplicabilidade das bigtechs - Recomendação de produtos/conteúdo



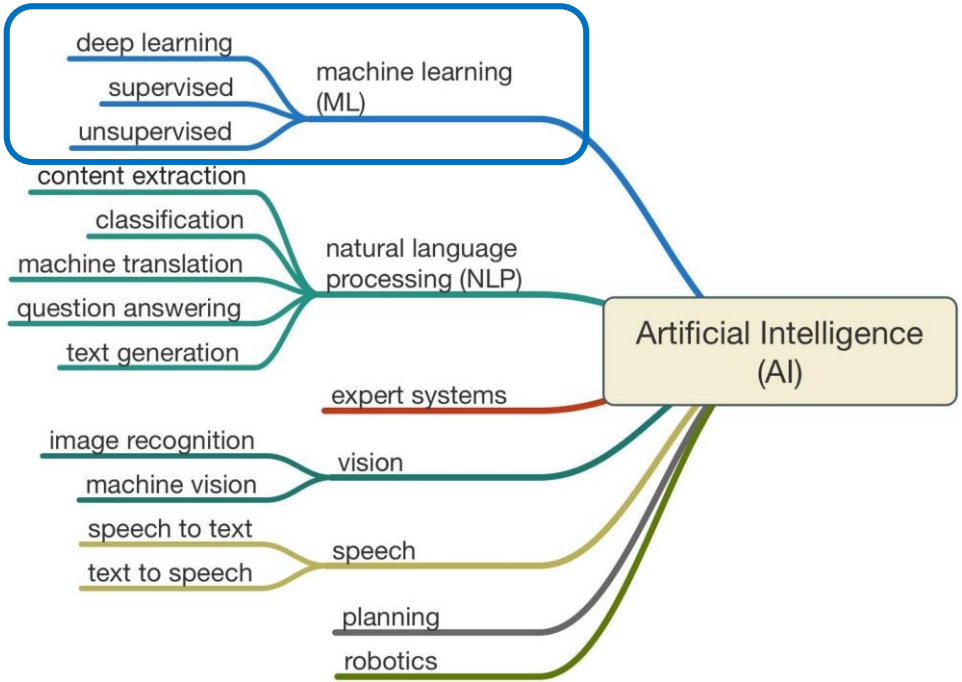
Aplicabilidade das bigtechs - Veículos completamente autônomos

Aplicabilidade das bigtechs - Large Language Models

Principais subáreas da IA



Principais subáreas da IA



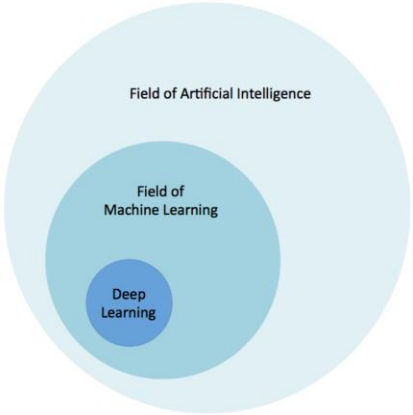
O que é ML?


PATTERN RECOGNITION
AI
ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
AUTOMATION
NEURAL NETWORKS
100110
DATA MINING
ALGORITHM
PROBLEM SOLVING

02

O que é Machine Learning

Machine Learning (Aprendizado de Máquina) é um subcampo da inteligência artificial que envolve o desenvolvimento de algoritmos e técnicas que permitem que os sistemas computacionais aprendam e melhorem automaticamente a partir dos dados, sem serem explicitamente programados para realizar tarefas específicas. (ChatGPT)



Vantagens de utilizar Machine Learning

Capacidade de lidar com grandes volumes de dados

Personalização e recomendações precisas

Automação de tarefas complexas

Detecção de padrões e insights valiosos

Melhoria contínua do desempenho

Previsão e análise de dados

Funcionamento do Machine Learning

Em vez de seguir instruções detalhadas, os algoritmos de Machine Learning são projetados para analisar e interpretar padrões nos dados de entrada, a fim de fazer previsões, tomar decisões ou realizar tarefas específicas. Eles aprendem com exemplos e experiências anteriores, ajustando seus modelos e parâmetros de acordo com os dados disponíveis.

Tipos Machine Learning

- Aprendizagem supervisionada;
- Aprendizagem não supervisionada;
- Aprendizagem semi-supervisionada;
- Aprendizagem por reforço.

Princípio do Machine Learning

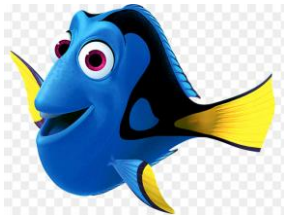
Melhorar a realização de uma tarefa a partir da experiência.

- Melhorar a realização da tarefa T.
- Em relação a uma medida de desempenho P.
- Baseada na experiência E.

Aprendizagem Supervisionada

Peixe

Boi



Peixe

Boi

?

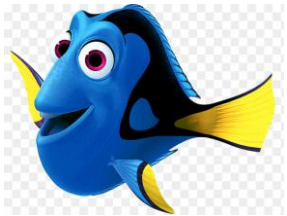
Peixe

Boi

Aprendizagem Supervisionada

Peixe

Boi



Peixe

Boi

Peixe

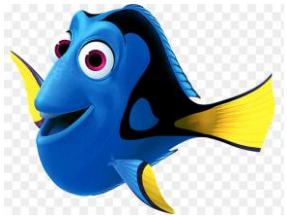
Peixe

Boi

Aprendizagem Supervisionada

Peixe

Boi



Peixe

Boi

?

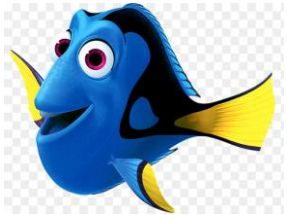
Peixe

Boi

Aprendizagem Supervisionada

Peixe

Boi



Peixe

Boi

Boi

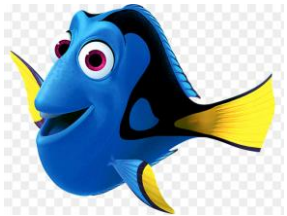
Peixe

Boi

Aprendizagem Supervisionada

Peixe

Boi



Peixe

Boi

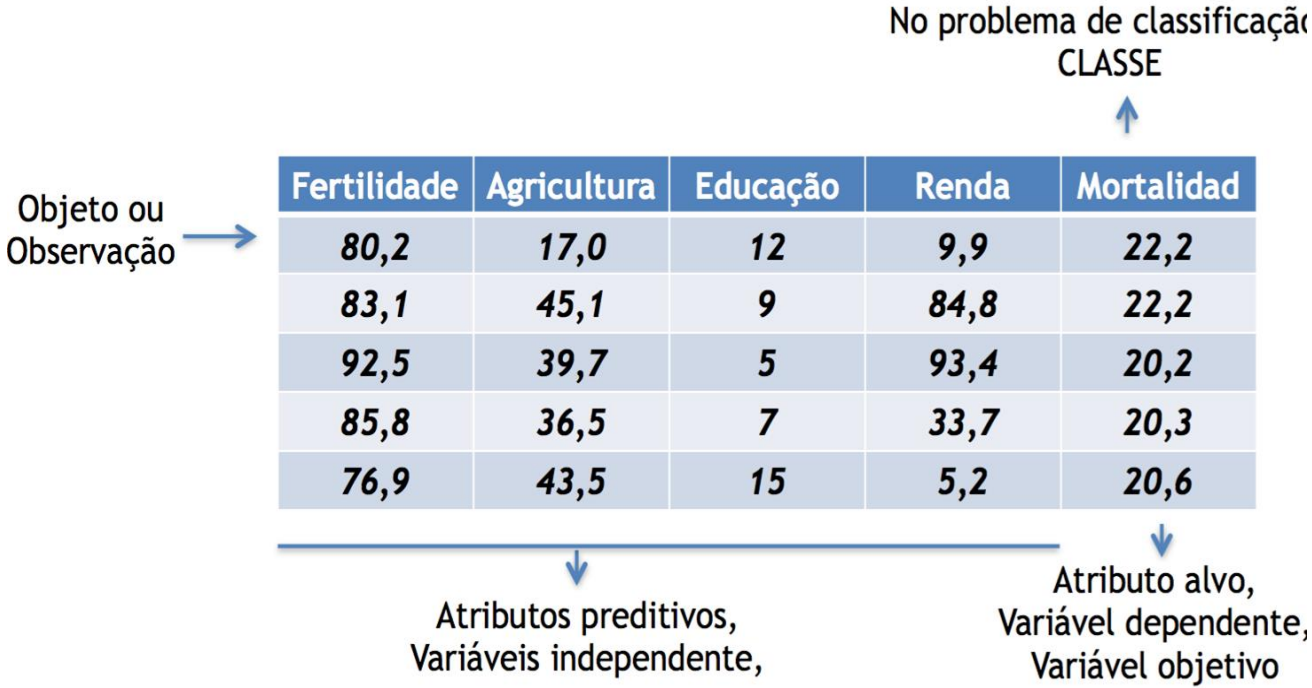
Peixe

Boi

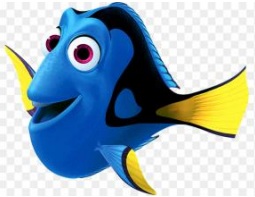
Aprendizagem Supervisionada - Classificação

Tamanho (P)	Largura (P)	Tamanho (S)	Largura (S)	Espécie
5,1	*3,5*	*1,4*	*0,2*	*Setosa*
4,9	*3,0*	*1,4*	*0,2*	*Setosa*
7,0	*3,2*	*4,7*	*1,4*	*Versicolor*
6,4	*3,2*	*4,5*	*1,5*	*Versicolor*
6,3	*3,3*	*6,0*	*2,5*	*Virginica*
5,8	*2,7*	*5,1*	*1,9*	*Virginica*

Aprendizagem Supervisionada - Regressão

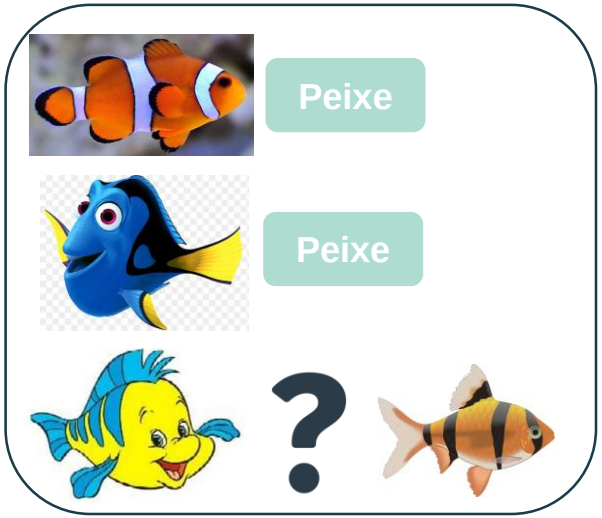


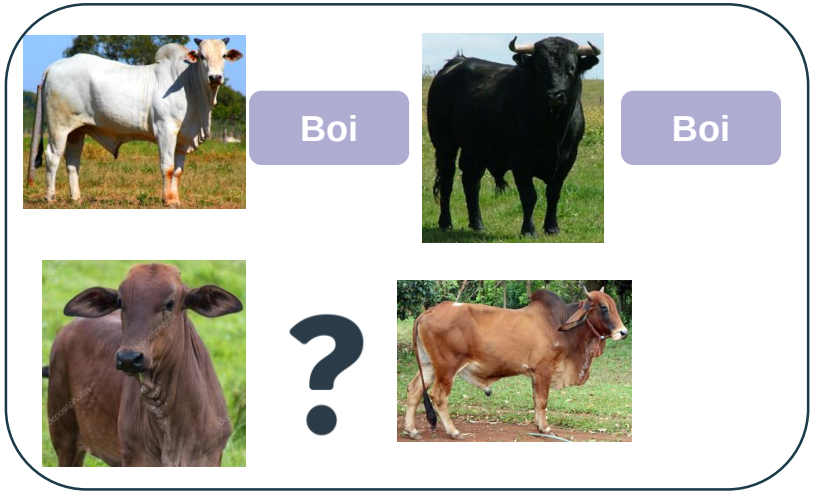
Aprendizagem Não Supervisionada



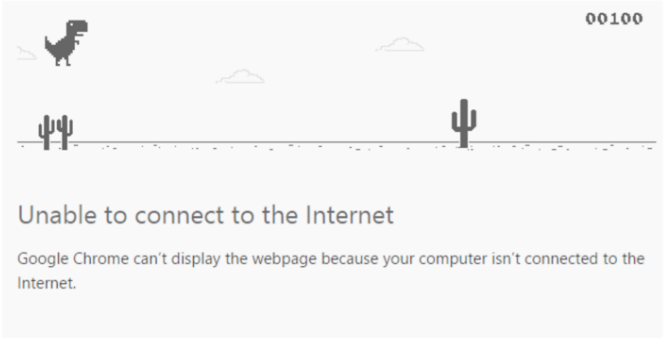
Aprendizagem Não Supervisionada

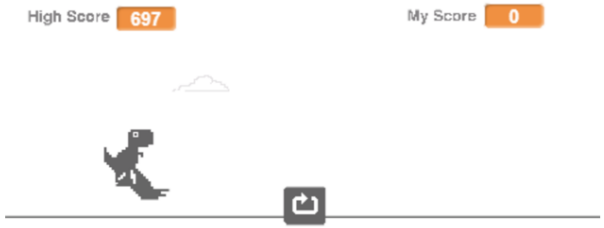
Aprendizagem Semi Supervisionada



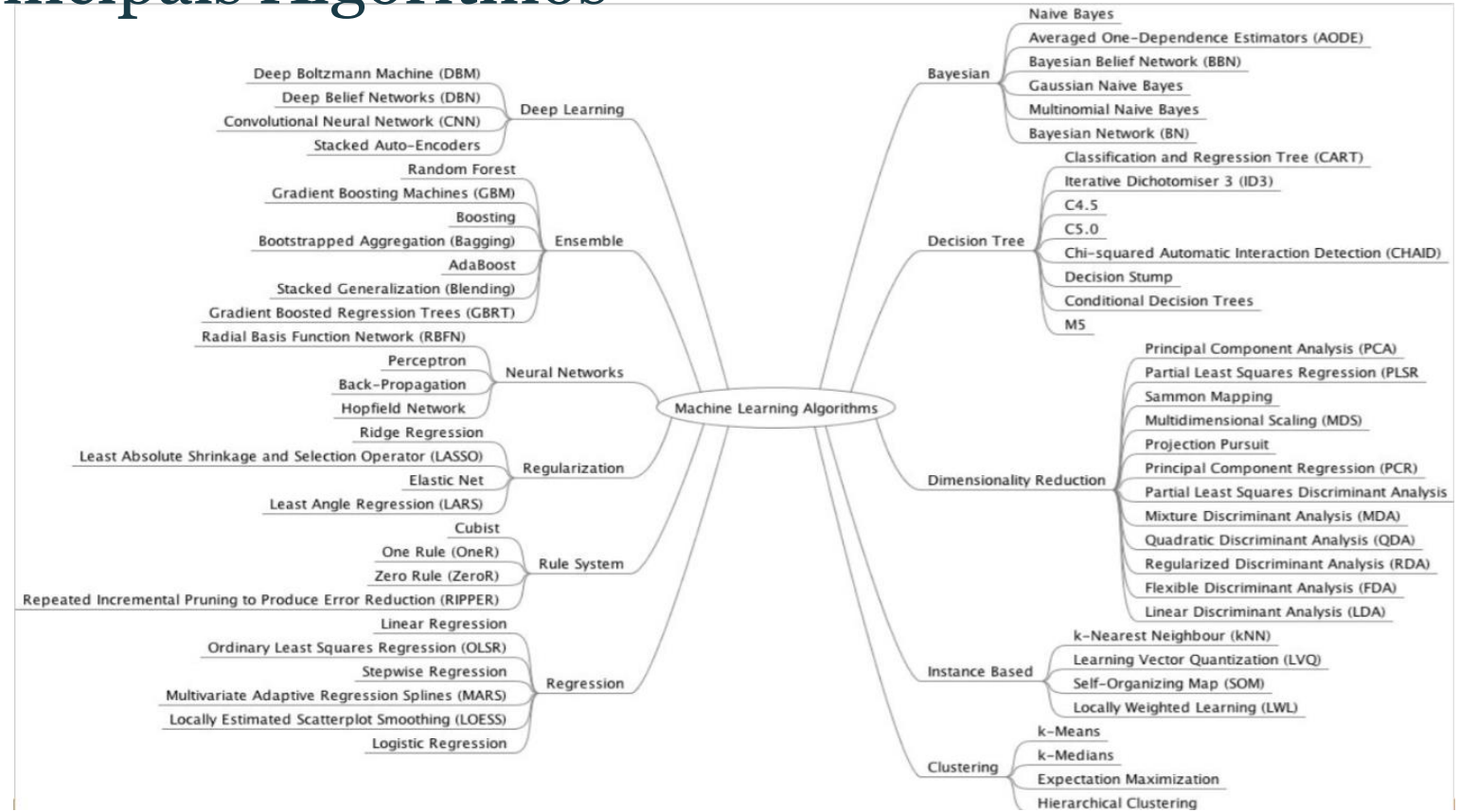


Aprendizagem Por Reforço





Principais Algoritmos



Supervisionado

03

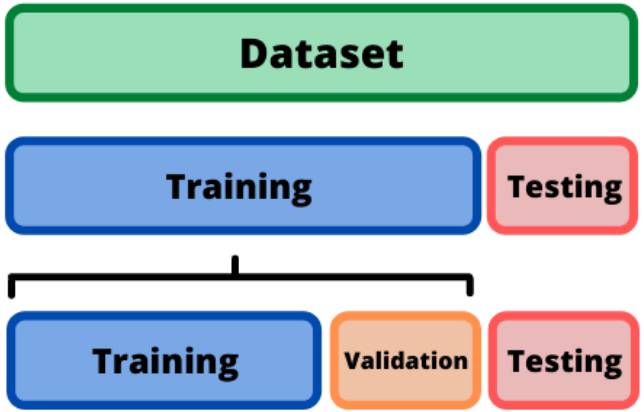
Divisão dos dados

3.1

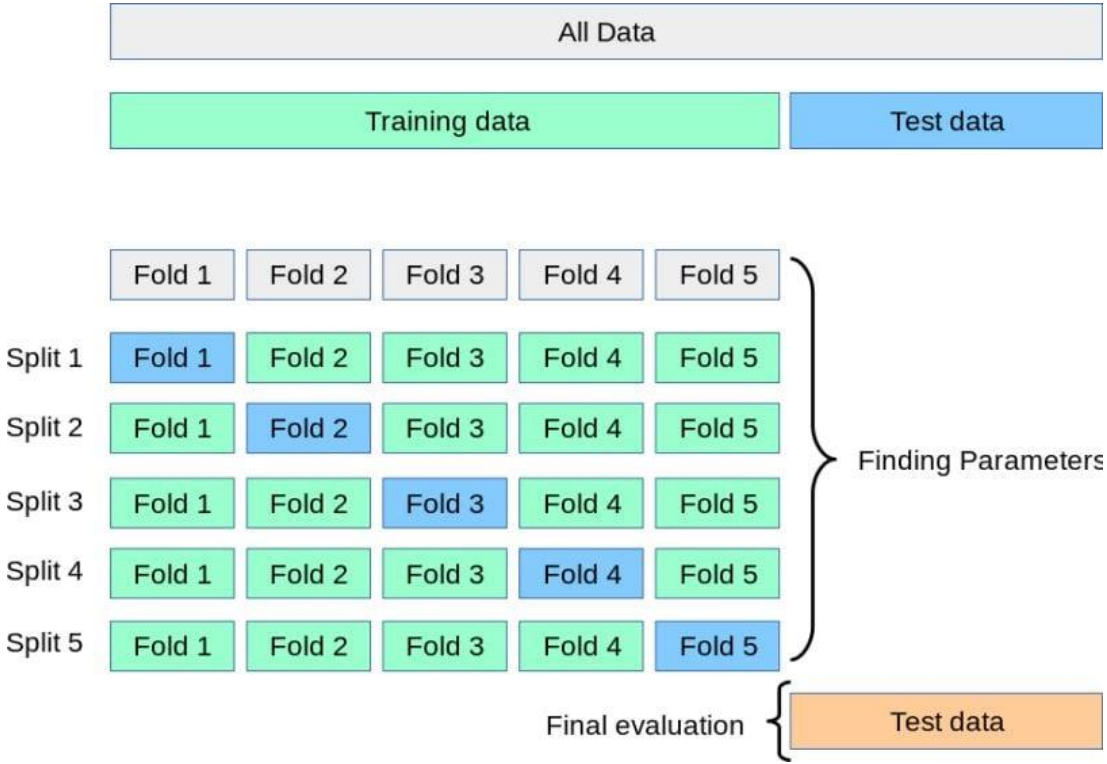
Divisão dos dados

Refere-se ao processo de dividir os dados em conjuntos de treinamento, validação e teste.

- Treinamento é usado para treinar o modelo;
- Validação é usado para ajustar os hiperparâmetros do modelo;
- Teste é usado para avaliar o desempenho do modelo em dados nunca antes vistos.

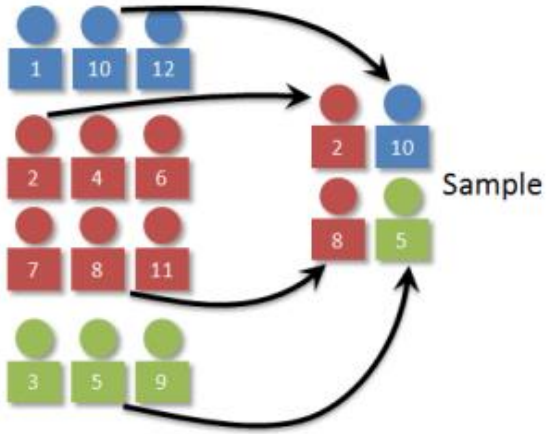


Divisão dos dados



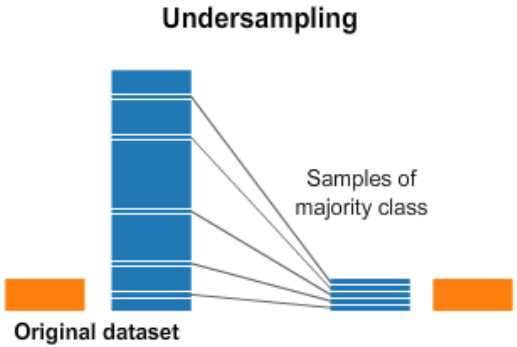
Divisão dos dados - Amostragem

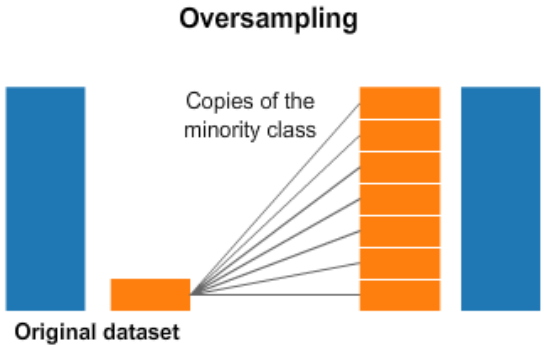
É importante garantir que os dados sejam divididos de forma aleatória e estratificada (mantem o mesmo número de objetos para cada classe, proporcional ao conjunto original), a fim de evitar a introdução de viés nos modelos.



Divisão dos dados - Balanceamento dos dados

Refere-se ao processo de equilibrar a distribuição de classes nas amostras de dados. Em muitos casos, os dados podem estar desbalanceados, ou seja, uma classe pode estar sub-representada em comparação com outras. Isso pode levar a modelos de aprendizado de máquina tendenciosos, que favorecem a classe majoritária. Para evitar esse problema, é necessário equilibrar a distribuição de classes, o que pode ser feito por meio de técnicas como *oversampling*, *undersampling* e geração sintética de dados.





Divisão dos dados - Balanceamento dos dados

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
4	4340	Claudia	21	F	52	Uniformes	37.6	1	PE	Doente
5	2301	Ana	22	F	72	Inexistentes	58.0	3	RJ	Doente
2	4039	Luiz	49	M	92	Espalhadas	38.0	2	RS	Doente
0	4201	Joao	28	M	79	Concentradas	38.0	2	SP	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

Métricas de avaliação de modelos

3.2

Métricas de avaliação de modelos

Métricas que refletem a qualidade de um modelo, portanto se forem mal escolhidas, será impossível avaliar se o modelo de fato está atendendo os requisitos necessários. Ademais, também são utilizadas para realizar comparação de modelos.

Métricas de avaliação de modelos de classificação

3.2.1

Métricas de avaliação de modelos de classificação

Um modelo de classificação binária tem como objetivo decidir em qual classe uma nova observação pertence dentre duas classes possíveis. Em geral as duas classes, denominadas de positiva (P) e negativa (N), indicam a ocorrência ou não de um determinado evento. Um exemplo seria classificar se um determinado paciente possui uma determinada doença (positivo) ou não (negativo).

A avaliação de um modelo de classificação é feita a partir da comparação entre as classes preditas pelo modelo e as classes verdadeiras de cada exemplo. Todas as métricas de classificação têm como objetivo comum medir quão distante o modelo está da classificação perfeita, porém fazem isto de formas diferentes.

Matriz de confusão

Uma maneira simples de se representar os resultados de um método de classificação de dados é através da chamada matriz de confusão.

Matriz de confusão		**Classe predita**	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

diegomariano.com

Matriz de confusão - Exemplo

Um programa de predição de chuva foi usado durante 100 dias. Dos 100 dias, o programa disse que iria chover em 55 e que não iria chover nos outros 45 dias. Entretanto, após os 100 dias, percebemos que choveu em 50 e não choveu nos outros 50 dias. Vamos observar a matriz de confusão dos resultados do nosso programa

- **VP = 40:** o programa disse que em 40 dos 100 dias **iria chover** e realmente **choveu**.
- **FP = 15:** o programa disse que em 15 dos 100 dias **iria chover**, mas **não choveu**.
- **FN = 10:** o programa disse que em 10 dos 100 dias **não iria chover**, mas **choveu**.
- **VN = 35:** o programa disse que em 35 dos 100 dias **não iria chover** e **realmente não choveu.**

Matriz de confusão		**O que o programa disse:**	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Matriz de confusão - Exemplo

Total de predições realizadas em cada classe: $pred_p = VP + FP$ $pred_n = VN + FN$

Total de valores reais: $real_p = VP + FN$ $real_n = VN + FP$

Erros = FP + FN

Acertos = VP + VN

Matriz de confusão		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Acurácia

A **acurácia** é considerada uma das métricas mais simples e importantes. Ela avalia simplesmente o percentual de acertos, ou seja, ela pode ser obtida pela razão entre a quantidade de acertos e o total de entradas:

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP}$$

Matriz de confusão		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Sensibilidade

Sensibilidade (também conhecida como *recall* ou revocação). Essa métrica avalia a capacidade do método de detectar com sucesso resultados classificados como positivos.

$$sensibilidade = \frac{VP}{VP + FN}$$

Matriz de confusão		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Especificidade

Especificidade avalia a capacidade do método de detectar resultados negativos.

$$especificidade = \frac{VN}{VN + FP}$$

Matriz de confusão		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Precisão

A **precisão** é uma métrica que avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos.

$$precisão = \frac{VP}{VP + FP}$$

Matriz de confusão		Classe predita	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

F1-score

Média harmônica calculada com base na precisão e na sensibilidade.

$$f1 = 2 * \frac{precisão * sensibilidade}{precisão + sensibilidade}$$

Matriz de confusão		**Classe predita**	
		Positiva	**Negativa**
Classe original	**Positiva**	VP	FN
	Negativa	FP	VN

Matriz de confusão		**O que o programa disse:**	
		Vai chover	**Não vai chover**
O que aconteceu de verdade:	**Choveu**	40	10
	Não choveu	15	35

diegomariano.com

Métricas de avaliação de modelos de regressão

3.2.2

R2 score

Medida estatística que representa a proporção da variância para uma variável dependente que é explicada por uma variável independente em um modelo de regressão. Enquanto a correlação explica a força da relação entre uma variável independente e uma variável dependente, o R-quadrado explica até que ponto a variância de uma variável explica a variância da segunda variável. Portanto, se o R2 de um modelo for 0,50, aproximadamente metade da variação observada pode ser explicada pelas entradas do modelo, logo R2 Score, normalmente, está entre 0 e 1, quanto mais próximo de 1, melhor o ajuste da regressão.

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}$$

Erro médio (ME)

Média da diferença do realizado e do previsto.

Previsto	Realizado	Dif.
3,34	3,00	-0,34
4,18	4,00	-0,18
3,00	3,00	0
2,99	3,00	0,01
4,51	4,50	-0,01
5,18	4,00	-1,18
8,18	4,50	-3,68

$$ME = \sum_{I=1}^{N} \frac{p_i - t_i}{n}$$

$$ME = \frac{-5,38}{7} = -0,76$$

Erro médio absoluto (MAE)

Média da diferença absoluta do realizado e do previsto.

Previsto	Realizado	Dif. Absoluta
3,34	3,00	|0,34|
4,18	4,00	|0,18|
3,00	3,00	|0|
2,99	3,00	|0,01|
4,51	4,50	|0,01|
5,18	4,00	|1,18|
8,18	4,50	|3,68|
		5,4

$$MAE = \sum_{I=1}^{N} \frac{|p_i - t_i|}{n}$$

$$MAE = \frac{5,4}{7} = 0,77$$

Erro médio quadrático (RMSE)

O desvio da amostra da diferença entre o previsto e o realizado.

Previsto	Realizado	Dif. ao Quad.
3,34	3,00	0,1156
4,18	4,00	0,0324
3,00	3,00	0
2,99	3,00	1E-04
4,51	4,50	1E-04
5,18	4,00	1,3924
8,18	4,50	13,5424

$$RMSE = \sqrt{\frac{\sum_{I=1}^{N}(p_i - t_i)^2}{N}}$$

$$RMSE = \sqrt{\frac{15{,}083}{7}}$$

$$RMSE = 1{,}46$$

Erro de porcentagem absoluta média (MAPE)

Diferença absoluta percentual de erro.

Previsto	Realizado	Erro % abs.
3,34	3,00	0,1133333
4,18	4,00	0,045
3,00	3,00	0
2,99	3,00	0,0033333
4,51	4,50	0,0022222
5,18	4,00	0,295
8,18	4,50	0,8177778

$$MAPE = \frac{\sum_{I=1}^{N} \frac{|p_i - t_i|}{|t_i|}}{N} \times 100$$

$$MAPE = \frac{1{,}2766667}{7} \times 100$$

MAPE = 18%

Regressão

3·3

Regressão

Para o funcionamento do modelo de regressão serão fornecidas informações na forma de variáveis atributos e o modelo estimará o valor da variável resposta usando dados de referência durante o treinamento. Lembrando que, para o caso da Regressão, o tipo de resposta esperado na saída será um valor contínuo representativo de acordo com a variável resposta. Especificamente quando se trata de **Regressões Lineares**, a inferência feita sobre a relação entre as variáveis é que pode ser descrita por uma **equação de reta**.

Regressão linear

Na regressão linear simples, tem-se um conjunto de dados formado por um único atributo X e a variável resposta Y. O modelo vai procurar estabelecer a melhor equação de reta que descreva o conjunto de dados, ou seja define a equação como:

$$Y \approx \beta_0 + \beta_1 X$$

Regressão linear

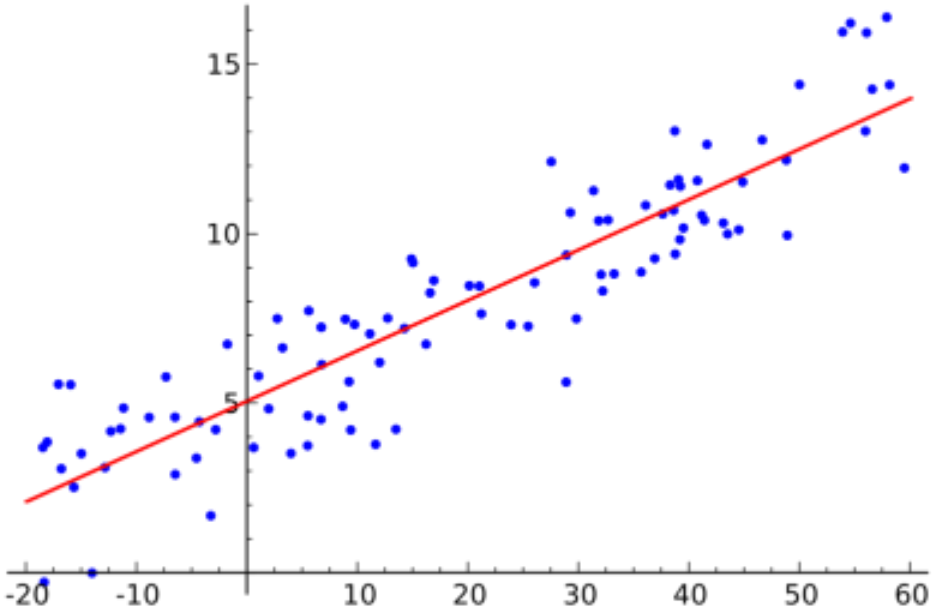
Uma forma para estimar os valores de $\beta 0$ e $\beta 1$ é a partir dos **valores médios** para X e Y, onde $\beta 0$ é o **coeficiente independente** (onde a equação de reta vai cortar o eixo Y no gráfico) e o $\beta 1$ é o **coeficiente angular** desta reta (indicando a inclinação da reta a ser ajustada). Assim os valores dos coeficientes também serão um valor médio da forma $\beta 0$ e $\beta 1$, dados pelas seguintes equações:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_{xx}} = \frac{Covar(x,y)}{Var(x)}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- Covariância indica o grau de interdependência entre duas variáveis;
- Variância é uma medida de o quão disperso estão os dados, ou seja o quão distante está cada valor desse conjunto do valor médio.

Regressão linear



imgur.com

Implementação

```python
def linear_regression(x, y):
    # Definir as médias
    mean_x, mean_y = np.mean(x), np.mean(y)

    # Calcular a covariância e variância
    S_xy = 0
    S_xx = 0

    #Laço para o somatório
    for i in range(0, len(x)):
        # termo covariância
        S_xy += (x[i] - mean_x)*(y[i] - mean_y)
        # termo variância
        S_xx += (x[i] - mean_x)**2

    # Calcular os coeficientes de regressão
    beta_1 = S_xy / S_xx
    beta_0 = mean_y - beta_1*mean_x

    # Retorna os coeficientes
    return beta_0, beta_1
```

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_{xx}} = \frac{Covar(x,y)}{Var(x)}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

```python
from sklearn.linear_model import LinearRegression

# Instanciar o modelo
model - LinearRegression()

# Treinamento do Modelo
model.fit(X_train, y_train)

# Gerar novas predições
y_pred = model.predict(X_test)
```

Algoritmo KNN

3·4

Vizinhos mais próximos

- Classifica um novo objeto com base nos exemplos do conjunto de treinamento que são próximos a ele;
- Pode ser utilizado tanto para classificação quanto para regressão;
- Tem variações definidas (principalmente) pelo número de vizinhos considerados.

1- Vizinho mais próximo

- 1-NN, 1-Nearest neighbour;
- Calcula distância entre cada 2 pontos;
- A métrica mais usual é a distância Euclidiana.

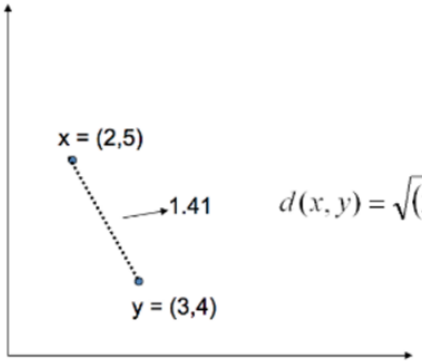
$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Exemplo - Distância Euclidiana

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

PODEM SER USADAS QUAISQUER MÉTRICAS QUE REPRESENTEM SIMILARIDADE ENTRE DOIS OBJETOS, COMO CORRELACÃO, POR EXEMPLO!!

DISTÂNCIA ENTRE STRINGS PODEM SER MEDIDAS PELA DISTÂNCIA DE HAMMING!!



$$d(x, y) = \sqrt{(2-3)^2 + (5-4)^2} = \sqrt{2} = 1.41$$

- CADA DIMENSÃO REPRESENTA UM ATRIBUTO
- X E Y REPRESENTAM INSTÂNCIAS (OBJETOS)

Implementação Distância Euclidiana

Do princípio (from scratch)

```
def euclidean_distance(point1, point2):
    sum_squared_distance = 0
    for i in range(len(point1)):
        sum_squared_distance += math.pow(point1[i] - point2[i], 2)
    return math.sqrt(sum_squared_distance)
```

```
x = [2,5]
y = [3,4]

distance = euclidean_distance(x,y)

print(distance)
```

```
1.4142135623730951
```

$$d_{L2}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Usando sklearn

```
dist = DistanceMetric.get_metric('euclidean')
X = [[2, 5],
     [3, 4]]

print(dist.pairwise(X))
```

```
[[0.         1.41421356]
 [1.41421356 0.        ]]
```

Usando numpy

```
x = np.array((2,5))
y = np.array((3,4))

dist = np.linalg.norm(x - y)

print(dist)
```

```
1.4142135623730951
```

Distância entre strings

Hamming

Strings de mesmo tamanho

- 1011101 and 1001001 is 2.
- 2173896 and 2233796 is 3.

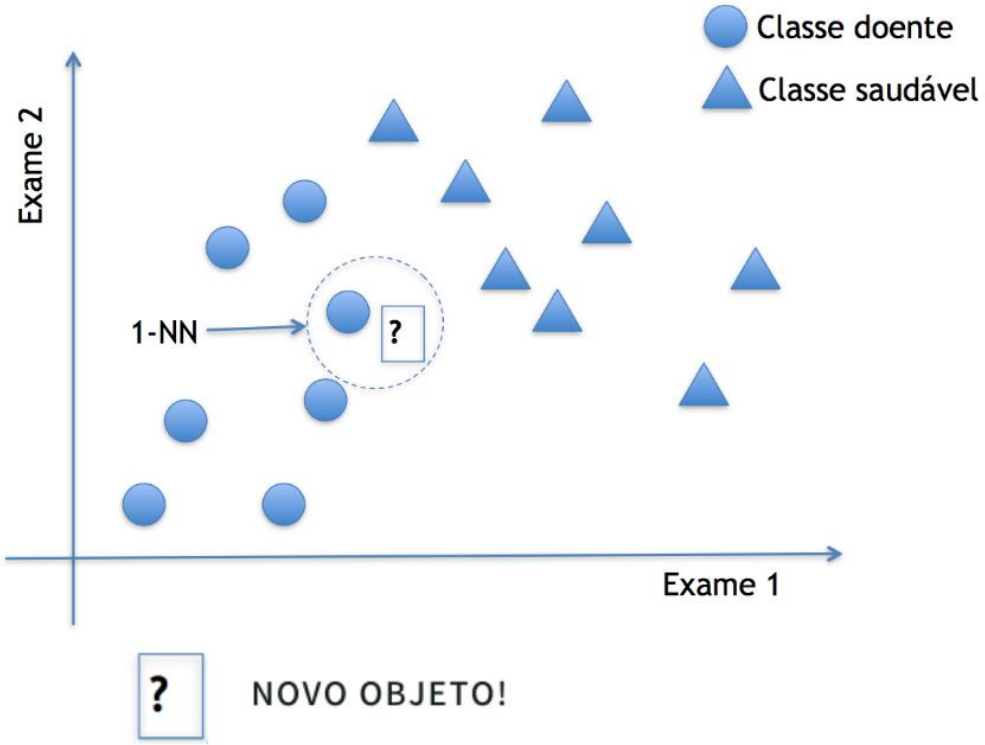
Levenshtein

Strings de tamanhos distintos

H		O	N	D	A	
H	Y	U	N	D	A	I

H	O		N	D	A	
H	Y	U	N	D	A	I

1-NN


Classe doente
Classe saudável
Exame 2
1-NN
?
Exame 1
?
NOVO OBJETO!

Algoritmo

Entrada: Conjunto de treinamento (D); objeto teste (z); conjunto de classes (L)

Saída: A classe de z

PARA CADA objeto y que pertence a D faça:

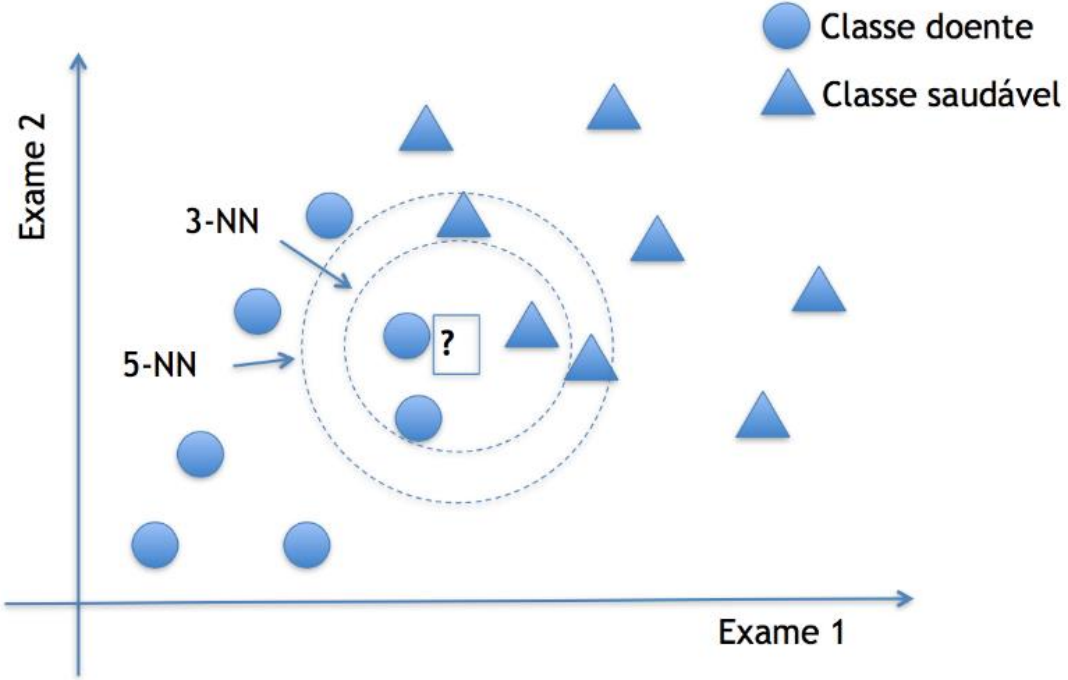
Compute a distância entre z e y

FIM

Fixe k, o número de vizinhos mais próximos na população (z) a ser considerado

A classe de saída do objeto z será a moda, média ou mediana dos rótulos dos k vizinhos do conjunto selecionado como mais próximo

K-NN



Resumo

Em vez de 1 vizinho mais próximo, os k objetos do conjunto de treinamento mais próximo do ponto de teste X;

Classificação:

- O objeto é classificado na classe mais votada: moda;

Regressão:

- Minimizar o erro quadrático: média;
- Minimizar o desvio absoluto: Mediana

Vantagens e Desvantagens

PRÓS:

- O algoritmo é simples e fácil de implementar;
- Não há necessidade de ajustar vários parâmetros;
- Treinamento rápido;
- Ele pode ser usado para classificação, regressão.

CONS:

- A classificação da base de teste é mais lenta e mais custosa em termos de tempo e memória;
- KNN também não é adequado para grandes dados dimensionais

Algoritmo Árvore de decisão

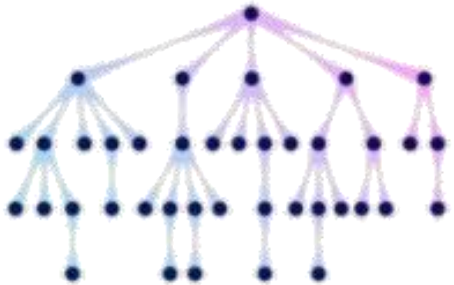
3·5

Árvore de Decisão

O principal objetivo do algoritmo árvore de decisão é encontrar o atributo ou variável independente que melhor realiza a divisão dos dados.

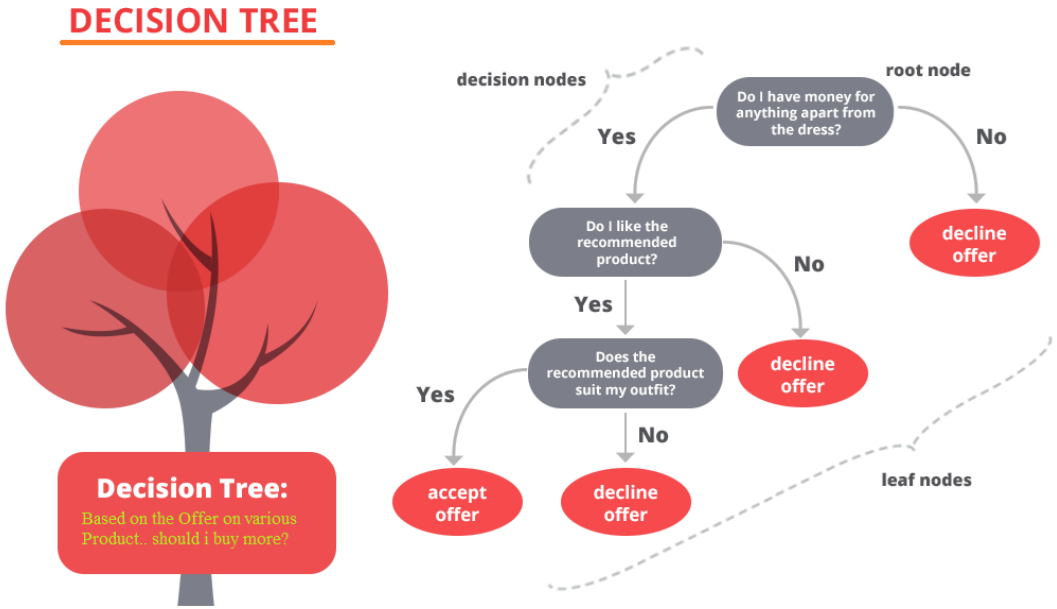
Uma árvore de decisão, utiliza a técnica de dividir para conquistar:

- Um problema complexo é decomposto em subproblemas mais simples;
- Recursivamente a mesma estratégia é aplicada cada subproblema.



Árvore de Decisão

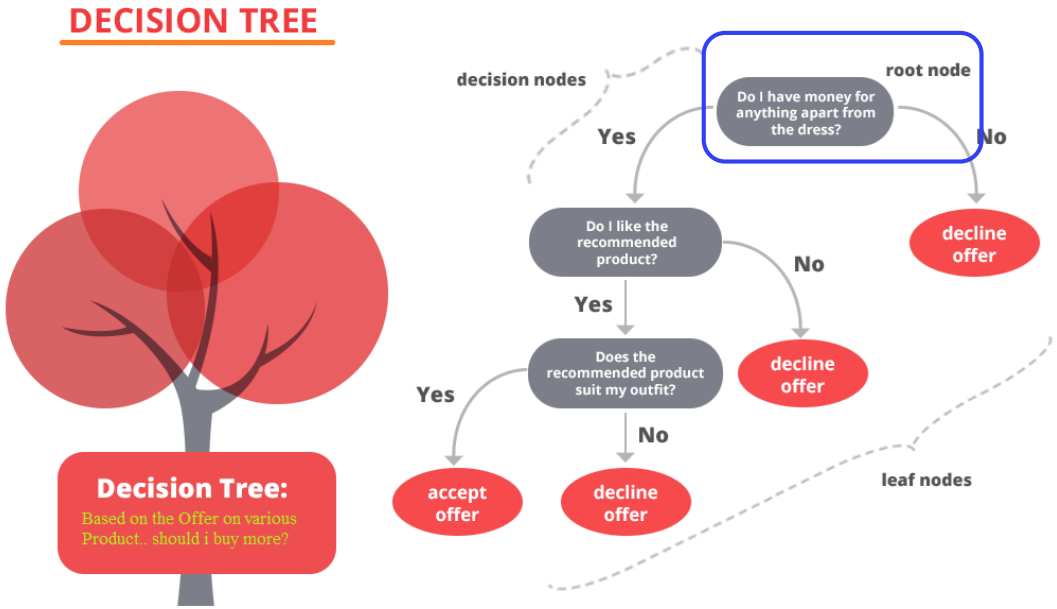
Podemos assimilar o funcionamento deste algoritmo como um fluxograma. As árvores de decisão consistem em nós de decisão interconectados em uma hierarquia, incluindo o nó raiz e os nós folha.



Fonte: data-notes.co

Árvore de Decisão

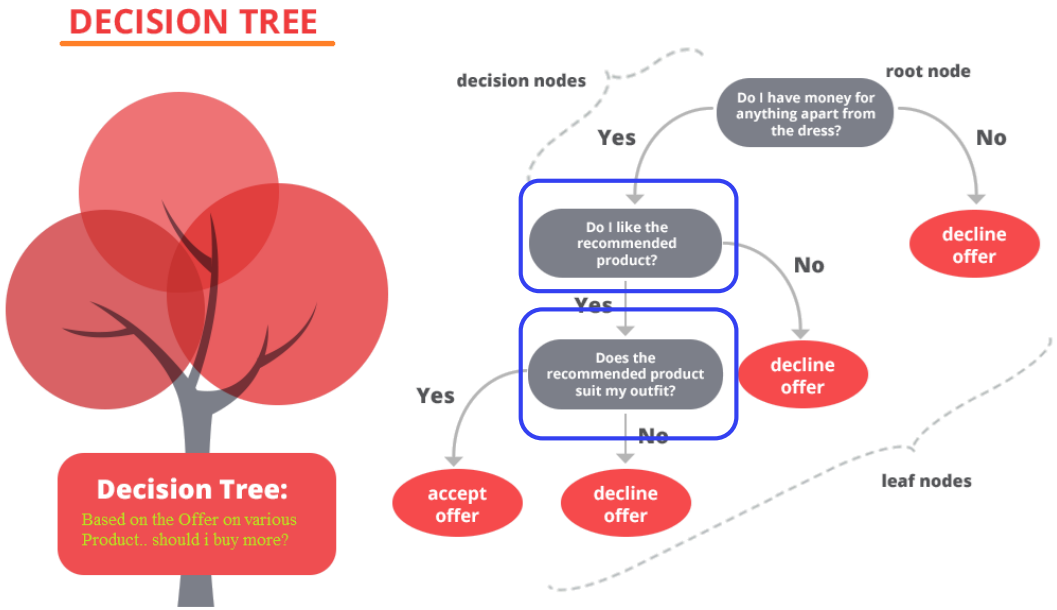
O nó raiz é o mais importante, pois separa o conjunto de dados em partições mais puras.



Fonte: data-notes.co

Árvore de Decisão

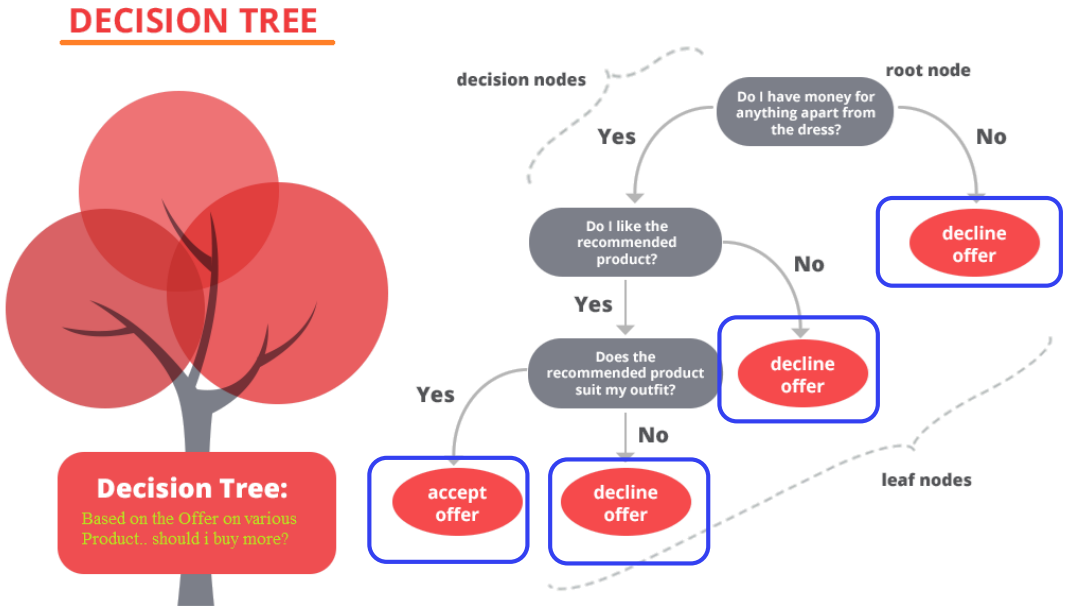
Os nós de decisão são atributos da base de dados que levam às ramificações da árvore.



Fonte: data-notes.co

Árvore de Decisão

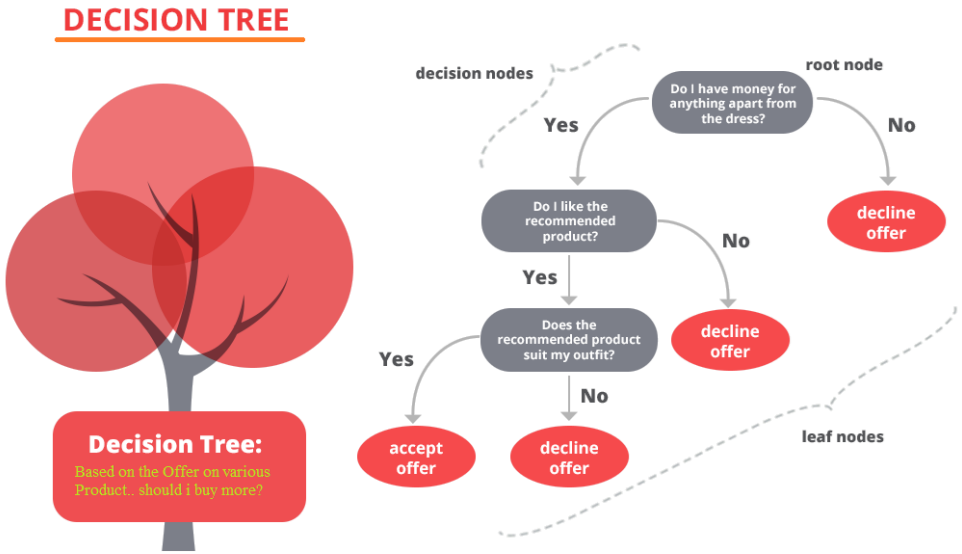
Os nós folha representam as respostas finais do algoritmo, como classes ou valores dependendo do tipo de problema (classificação ou regressão).



Fonte: data-notes.co

Árvore de Decisão

Em cada ligação entre os nós, o algoritmo pergunta acerca de uma condição (“if-else”) gerada fundamentada no conhecimento absorvido dos dados pelo algoritmo e realiza a divisão dos dados com base nas regras ou condições. Por exemplo, se estamos lidando com uma base de pessoas, uma condição que poderia ser gerada seria considerar pessoas com idade maior de 18 anos.



Fonte: data-notes.co

Árvore de Decisão – Critérios para escolha dos atributos

1. Índice de Gini;
2. Entropia;
3. Ganho de informação

Índice de Gini

É uma medida da impureza de um nó. Esta medida quantifica a quantidade de vezes que um elemento escolhido aleatoriamente do conjunto de dados seria rotulado de maneira incorreta se fosse rotulado aleatoriamente de acordo com a distribuição de rótulos do subconjunto. É a maneira mais popular e fácil de dividir uma árvore de decisão e funciona apenas com alvos categóricos, pois faz apenas divisões binárias. Quanto menor a Impureza (Gini), maior a homogeneidade do nó. A Impureza (Gini) de um nó puro (mesma classe) é igual a zero. A fórmula para calcular a Impureza (Gini) é a:

$$gini(R) = \sum p(c|R)(1 - p(c|R))$$

Onde:(p(c|R)) é a probabilidade de um ponto da região R pertencer a classe C.

Entropia

Representa a falta de uniformidade ou uma medida de aleatoriedade nos dados. Quanto mais alta a entropia, mais caótico e misturados estão os dados e quanto menor a entropia, mais uniforme e homogênea está o conjunto de dados. A fórmula para se calcular a entropia é:

$$entropia(R) = -\sum p(c|R)log(p(c|R))$$

Onde: A probabilidade é estimada pela razão entre quantidade de pontos da classe c e o total de pontos em R.

Ganho de informação

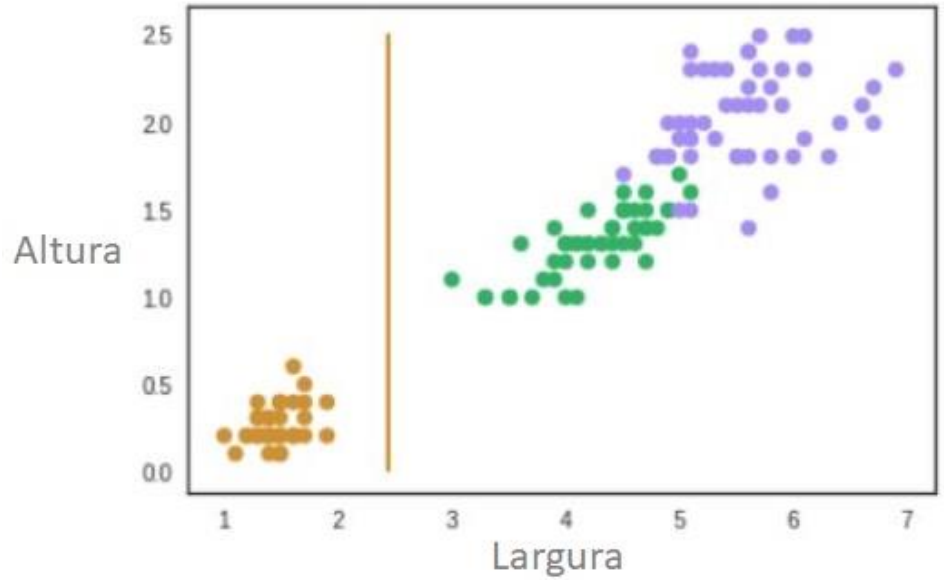
É uma propriedade estatística que mede quão bem um determinado atributo separa os exemplos de treinamento de acordo com sua classificação alvo ou rótulo. Em outras palavras, o ganho de informação representa a informação aprendida sobre os rótulos quando dividimos uma região do espaço em duas sub-regiões de acordo com um critério de divisão como a entropia ou impureza (gini), citadas acima. A fórmula que define o ganho de informação é a:

$$InfoGain(R, R_e, R_d) = H(R) - (|R_e| * H(R_e) + |R_d| * H(R_d))/|R|$$

Onde: H é a impureza da região (gini ou entropia); (R) é a região atual; (Re) é sub-região da esquerda; (Rd) é sub-região da direita |R| é quantidade de exemplos na dada região

Exemplo Prático

Distribuição de Produtos, onde é apresentada a distribuição de produtos de acordo com suas medidas em termos de largura e altura.



Exemplo Prático

- Vamos iniciar calculando (p(c|R)) para cada região e produto:

  Sabendo que temos 50 exemplos de cada produto (Produto_A, Produto_B e Produto_C), a (p(c|R)) para todos os produtos é (50 / 150 ~ 0.33)

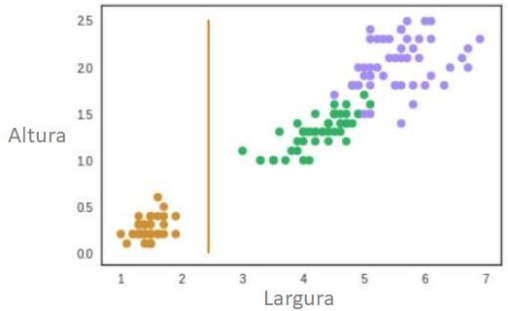
Exemplo Prático

- Vamos iniciar calculando (p(c|R)) para cada região e produto:

  Sabendo que temos 50 exemplos de cada produto (Produto_A, Produto_B e Produto_C), a (p(c|R)) para todos os produtos é (50 / 150 ~ 0.33)

- Calculando para a sub-região da esquerda:

  Temos que (p(Produto_A | Re) = 1.0) (só temos Produto_A nessa região) e (p(Produto_B | Re) = (p(Produto_C | R_e) = 0.0)



Exemplo Prático

- Vamos iniciar calculando (p(c|R)) para cada região e produto:

  Sabendo que temos 50 exemplos de cada produto (Produto_A, Produto_B e Produto_C), a (p(c|R)) para todos os produtos é (50 / 150 ~ 0.33)

- Calculando para a sub-região da esquerda:

  temos que (p(Produto_A | Re) = 1.0) (só temos Produto_A nessa região) e (p(Produto_B | Re) = (p(Produto_C | R_e) = 0.0)

- Calculando para a sub-região da direita:

  Para direita, (p(Produto_A | Rd) = 0.0 e p(Produto_B | Rd) = p(Produto_C | Rd) = 0.5)

Exemplo Prático

Com esses valores em mãos, podemos calcular a entropia e a impureza (gini) de cada região e, por consequência, obter o ganho de informação.

$$entropia(R) = -\sum p(c|R)log(p(c|R)) = -3 * (0.33log(0.33))\ 0.48$$

$$entropia(R_e) = -(1.0log(1.0) + 0.0log(0.0) + 0.0log(0.0)*) = 0$$

$$entropia(R_d) = -(0.0log(0.0) + 0.5log(0.5) + 0.5log(0.5)*)\ 0.30$$

$$InfoGain(R, R_e, R_d) = H(R) - (|R_e| * H(R_e) + |R_d| * H(R_d))/|R|$$

Portanto, o ganho de informação usando entropia como critério de impureza é:

InfoGain = 0.48 - (50 * 0 + 100 * 0.30) / 150 = 0.28

Exemplo Prático

Com esses valores em mãos, podemos calcular a entropia e a impureza (gini) de cada região e, por consequência, obter o ganho de informação.

$$gini(R) = \sum p(c|R)(1 - p(c|R)) = 3 * (0.33 * (1 - 0.33))\ 0.66$$

$$gini(R_e) = (1.0 * (1.0 - 1.0) + 0.0 * (1 - 0.0) + 0.0 * (1 - 0.0)) = 0$$

$$gini(R_d) = (0.0 * (1 - 0.0) + 0.5 * (1 - 0.5) + 0.5 * (1 - 0.5))\ 0.5$$

$$InfoGain(R, R_e, R_d) = H(R) - (|R_e| * H(R_e) + |R_d| * H(R_d))/|R|$$

Portanto, o ganho de informação usando (gini) como critério de impureza é:

InfoGain = 0.66 - (50 * 0 + 100 * 0.50) / 150 = 0.16

Exemplo Prático

Portanto, o ganho de informação usando entropia como critério de impureza é:

InfoGain = 0.48 - (50 * 0 + 100 * 0.30) / 150 = 0.28

Portanto, o ganho de informação usando (gini) como critério de impureza é:

InfoGain = 0.66 - (50 * 0 + 100 * 0.50) / 150 = 0.16

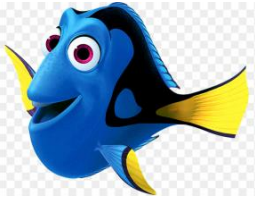
Desta maneira, o ganho de informação considerando o critério de impureza entropia foi maior do que considerando o critério de impureza (gini). Assim, o ponto de corte que retornou o maior ganho de informação e o melhor a ser considerado no exemplo acima seria a entropia, porque torna os ramos da árvore mais homogêneos.

JUPYTER NOTEBOOK
+
ATIVIDADE PRÁTICA

Não Supervisionado

04

Aprendizagem Não Supervisionada

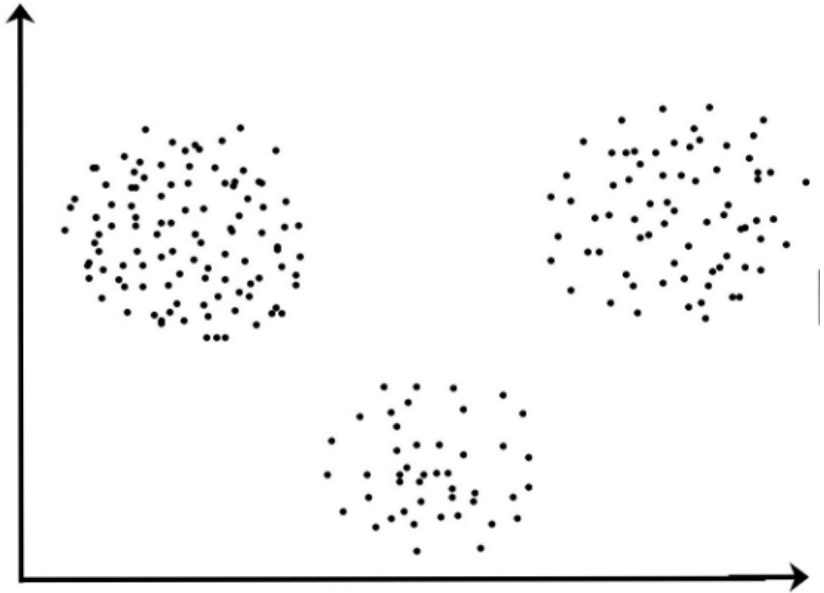


Aprendizagem Não Supervisionada

Os objetos pertencentes a cada cluster compartilham alguma característica.

Cluster: uma coleção de objetos próximos ou que satisfazem alguma relação espacial.

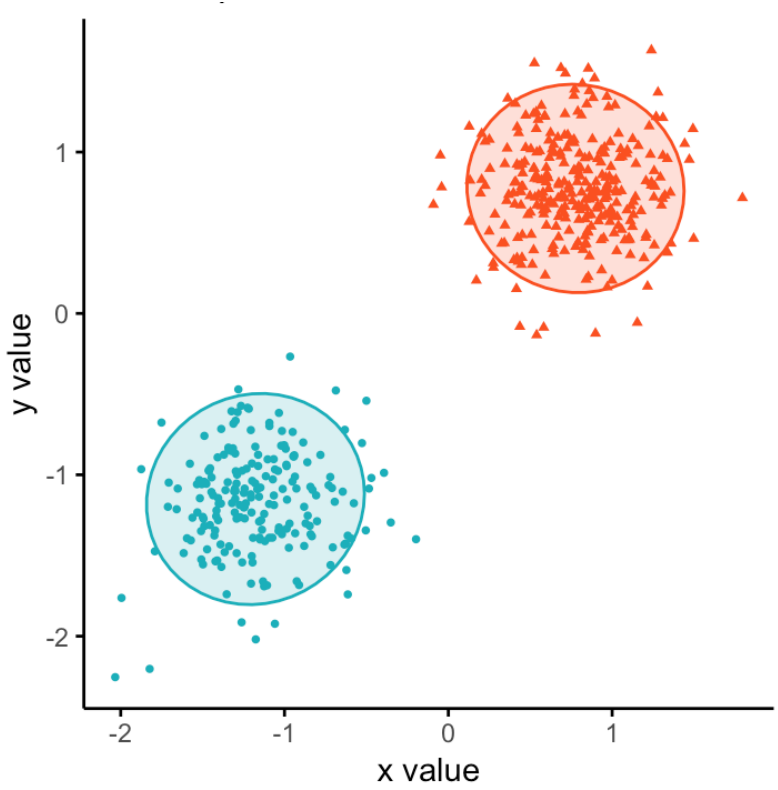
Cluster bem separado: conjunto de pontos tal que qualquer ponto em um determinado cluster está mais próximo (ou é mais similar) dos prontos de seu cluster do que de qualquer ponto de outros clusters.



Aprendizagem Não Supervisionada

Cluster baseado em centro: conjunto de pontos tal que qualquer ponto em um dado cluster está mais próximo (ou é mais similar) ao centro do cluster do que ao centro de qualquer outro cluster.

O centro de um cluster pode ser um centróide, como a média aritmética dos pontos do cluster.



Principais Características Não Supervisionado

4.1

Matriz de Similaridade e Dissimilaridade

Representa a similaridade ou a dissimilaridade entre cada par de objetos.

Cada elemento da matriz S_{nxn}, s_{ij}, é dado pela distância. $d(x_i,x_j)$, ou pela similaridade, $s(x_i,x_j)$, entre os objetos x_i e x_j.

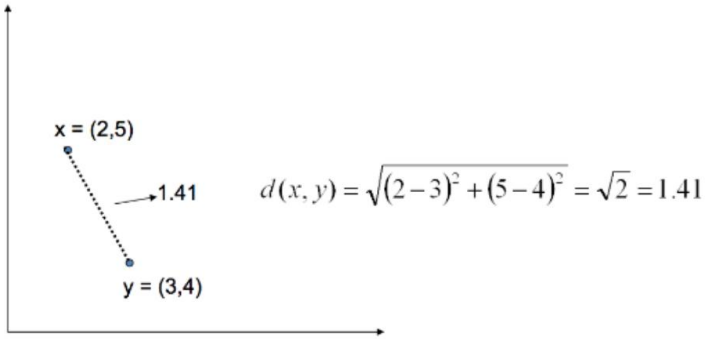
Objeto	Coordenada X1	Coordenada X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Proximidade

Uma das medidas de dissimilaridade mais comum para objetos cujos atributos são todos contínuos é a distância euclidiana;

Uma das medidas de similaridade mais usadas é a correlação.




```
corr = df.corr()
corr.style.background_gradient(cmap='coolwarm')
```

	identificador	idade	peso	temperatura	internacoes
identificador	1	0.48618	0.0620799	-0.319808	-0.817134
idade	0.48618	1	0.560835	-0.191917	-0.512349
peso	0.0620799	0.560835	1	0.0593191	-0.28261
temperatura	-0.319808	-0.191917	0.0593191	1	-0.035277
internacoes	-0.817134	-0.512349	-0.28261	-0.035277	1

Validação

Determina se os clusters são significativos, ou seja, se a solução é representativa para o conjunto de dados analisado.

Determina o número apropriado de clusters para um conjunto de dados, que em geral não é conhecido previamente.

Alguns fatores têm grande influência no desempenho das técnicas de agrupamento:

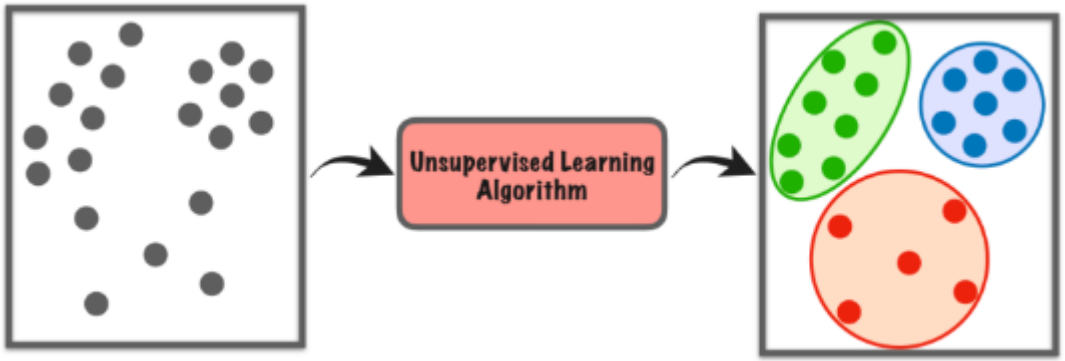
- Estrutura dos clusters (forma, tamanho, número de clusters);
- Presença de *outliers*;
- Grau de sobreposição dos clusters;
- Escolha da medida de similaridade.

K-médias (K-Means)

4.2

K-Means

O K-Means é um dos principais algoritmos de clusterização utilizados em *Machine Learning* pois é um modelo de simples **interpretabilidade** e apresenta uma boa **eficiência computacional**, ou seja, consegue modelar com facilidade utilizando o *K-Means*.



towardsdatascience

K-Means - Algoritmo

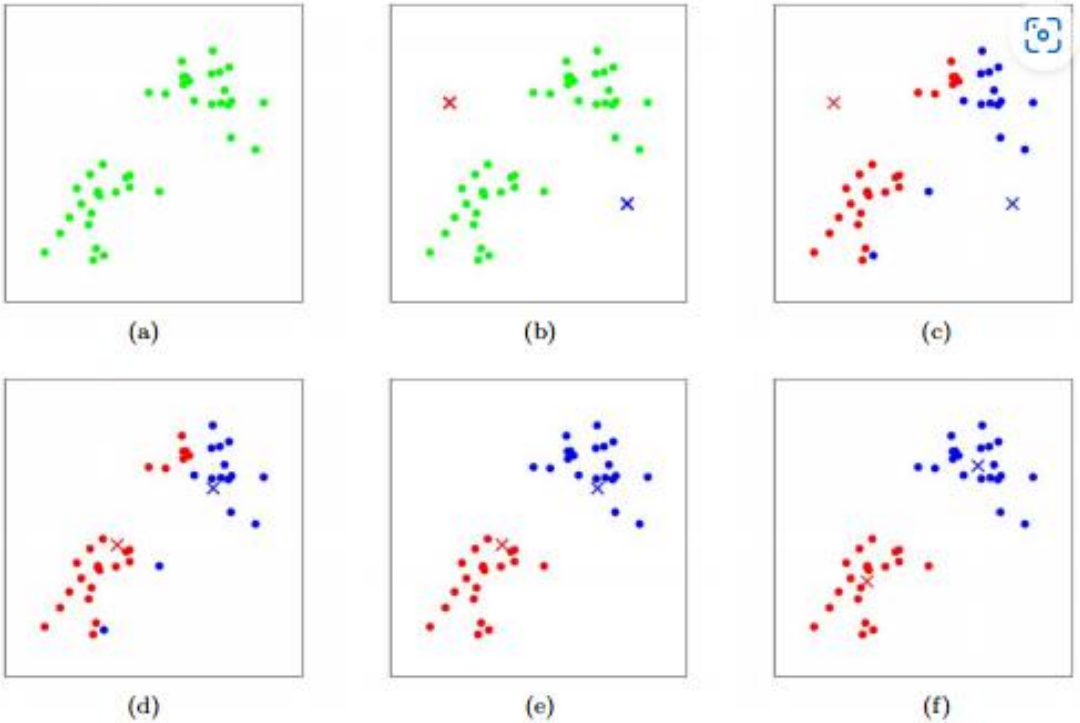
1) **Definição dos Centroides:** Supondo que deve-se separar os dados em *K* grupos (também chamados de *clusters*), para inicializar o modelo define-se *K* centroides iniciais aleatoriamente, que serão utilizados para os cálculos do modelo;

2) **Agrupamentos dos *K* grupos:** Dado os *K* grupos utilizados no modelo, cada uma das observações será associada ao **centroide mais próximo** utilizando de cálculos de distância (no caso, distância euclidiana);

3) **Reposicionamento dos Centroides:** Separados as observações em cada um dos *K* grupos, a partir das observações recalcula-se a **posição dos centroides** como a média da posição das observações dentro de determinado grupo (isto fazendo referência ao nome de *K-Means*);

4) **Processo Iterativo:** Os passos 2 e 3 serão repetidos até que o modelo considere que não houve mais alterações na **posição dos centroides**, levando em considerando uma margem de erro para as variações do posicionamento do centroide, ou mesmo quando o número máximo de iterações é atingido.

K-Means - Funcionamento



stanford.edu

Método do cotovelo

4.2.1

Método do cotovelo

A ideia é executar o KMeans para várias quantidades diferentes de clusters e dizer qual dessas quantidades é o **número ótimo de clusters**.

O que geralmente acontece ao aumentar a quantidade de clusters no KMeans é que as diferenças entre clusters se tornam muito pequenas, e as diferenças das observações intra-clusters vão aumentando. Então é preciso achar um equilíbrio em que as observações que formam cada agrupamento sejam o mais homogêneas possível e que os agrupamentos formados sejam o mais diferentes um dos outros.

Método do cotovelo

Como o KMeans calcula a distância das observações até o centro do agrupamento que ela pertence, o ideal é que essa distância seja a menor viável. Matematicamente falando, nós estamos buscando uma quantidade de agrupamentos em que a soma dos quadrados intra-clusters (ou do inglês *within-clusters sum-of-squares* (wcss)) seja a menor possível, sendo zero o resultado ótimo.

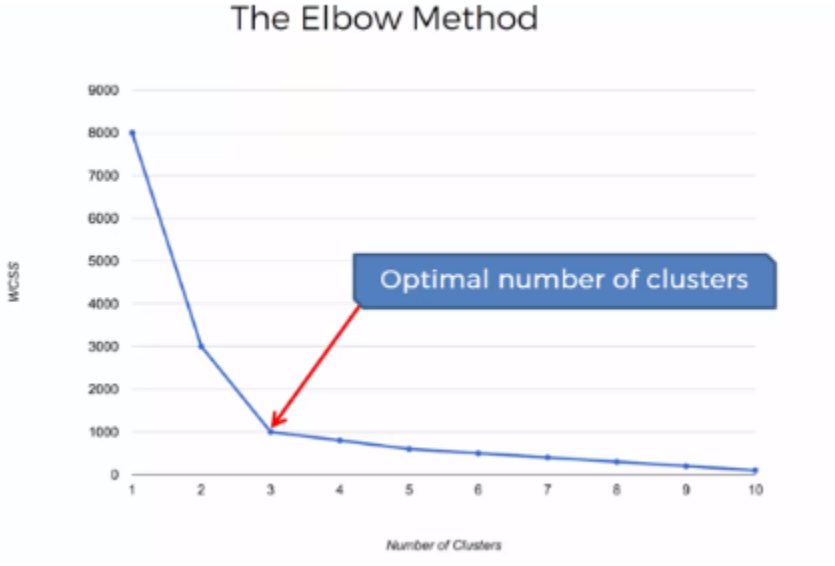
Método do cotovelo

Por exemplo, quando K = 3, a distância de soma (p, c) é a soma da distância dos pontos em um cluster do centróide.

$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Método do cotovelo

Na representação abaixo podemos ver que após 3 não há diminuição significativa no WCSS, então 3 é o melhor aqui. Portanto, há um formato de cotovelo que se forma e geralmente é uma boa ideia escolher o número onde esse cotovelo é formado. Muitas vezes o gráfico não seria tão intuitivo, mas com a prática fica mais fácil.



Método de silhueta

4.2.2

Método de silhueta

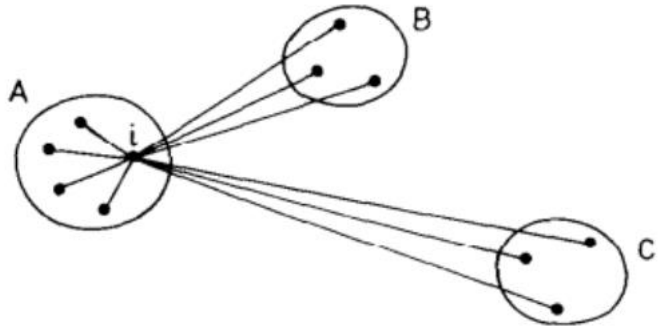
Silhueta refere-se a um método estatístico utilizado para interpretação e validação de consistência dentro dos clusters formados. O valor da silhueta (também chamado de coeficiente de silhueta) é uma medida de quão semelhante um objeto é ao seu próprio cluster (coesão), em comparação com outros clusters (separação). Este valor pode ser calculado com qualquer métrica de distância, como a euclidiana ou a de Manhattan.

Os coeficientes de silhueta variam de -1 a +1. Os valores próximos a +1 indicam que a amostra está longe dos clusters vizinhos, ou seja, indica que o objeto está no grupo que deveria se encontrar e que não deveria ser agrupado aos grupos vizinhos. Um valor 0 indica que a amostra está dentro ou muito perto do limite de decisão entre dois clusters vizinhos. Valores negativos indicam que essas amostras podem ter sido atribuídas ao cluster errado.

Método de silhueta

1. Para cada ponto *i*, calcula-se a distância intra-cluster, denominada de *a(i)*. Esta distância é calculada através da distância média de *i* para todos os outros pontos que foram agrupados dentro deste mesmo *cluster*. Pode-se interpretar *a(i)* como quão bem *i* foi atribuído ao seu grupo (quanto menor o valor, melhor a atribuição). Desta forma, a média de dissimilaridade do ponto *i* para um grupo *c* é definida como a média das distâncias de *i* para todos os pontos em *c*.

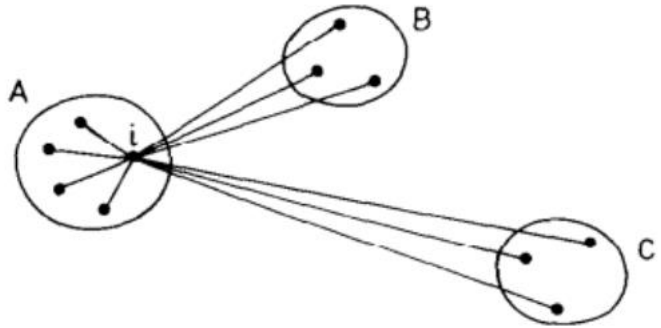
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Método de silhueta

2. Para cada ponto *i*, calcula-se a distância inter-cluster, denominada por *b(i)*. Esta distância é calculada através da distância média de *i* para todos os outros pontos que foram agrupados em *clusters* distintos do ponto *i*. Assim, o valor de *b(i)* será denominado pela menor distância média de *i* para todos os pontos pertencentes a outros grupos, do qual *i* não é um membro. O grupo com essa menor dissimilaridade média é o "grupo vizinho" de *i*.

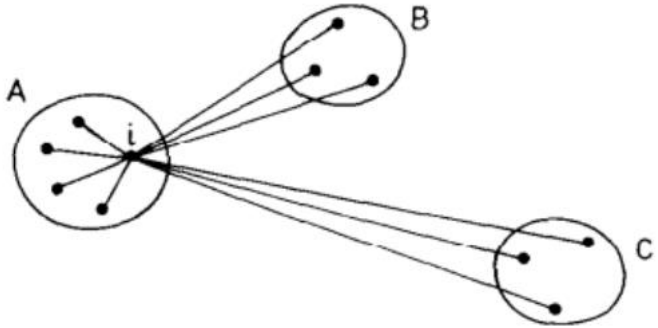
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Método de silhueta

3. Para cada ponto *i*, o coeficiente de silhueta pode ser descrito pela equação abaixo. Ademais, uma ilustração correspondente aos elementos envolvidos no cálculo de *s(i)* pode ser visualizada na figura abaixo.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

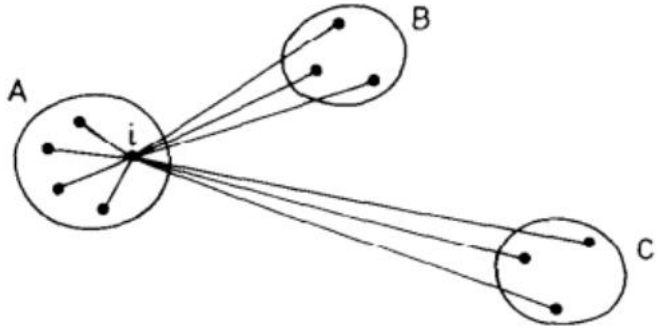


Método de silhueta

3. Para cada ponto *i*, o coeficiente de silhueta pode ser descrito pela equação abaixo. Ademais, uma ilustração correspondente aos elementos envolvidos no cálculo de *s(i)* pode ser visualizada na figura abaixo.

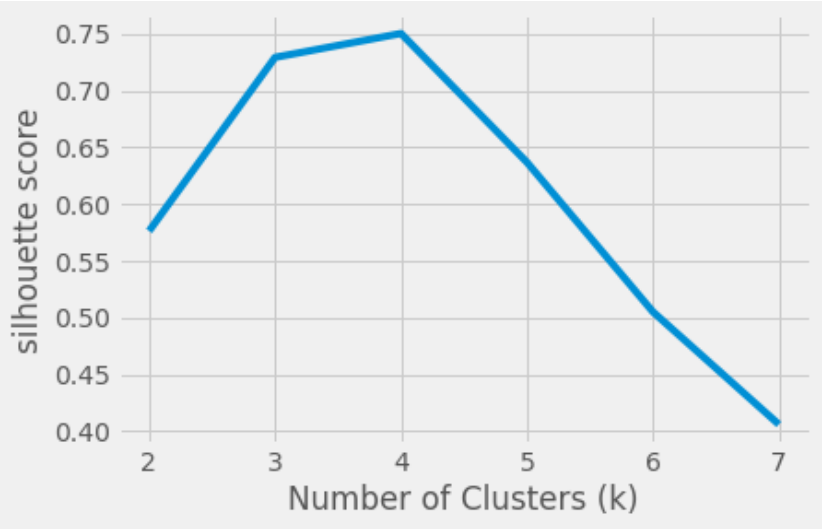
Portanto, o score de um cluster é calculado através da média dos coeficientes de silhueta de todos os pontos pertencentes a este grupo. Assim, para saber o quão bom foi o agrupamento, basta calcular o valor de silhueta do agrupamento total, ou seja, a média dos coeficientes de silhueta de todos os pontos do conjunto de dados.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

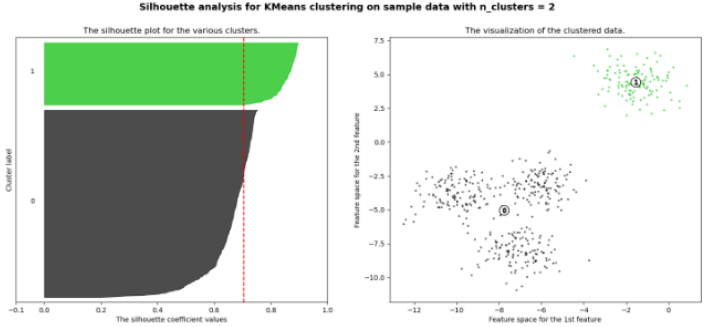


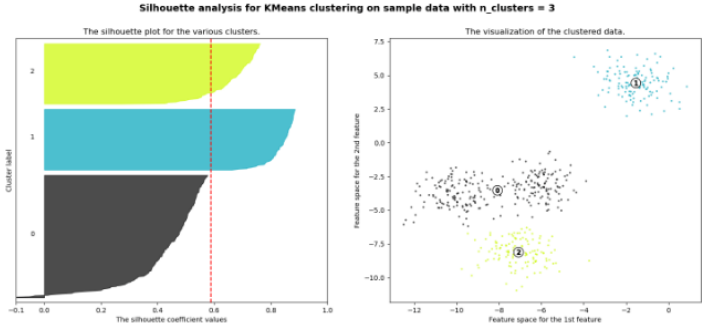
Método de silhueta

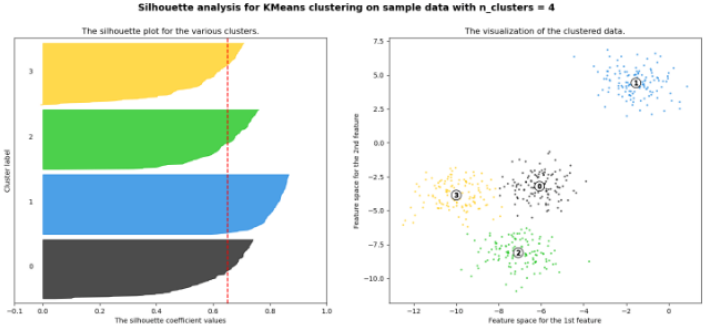
Semelhante ao método anterior, escolhemos um intervalo de valores candidatos de k (número de clusters) e, em seguida, treinamos o agrupamento K-Means para cada um dos valores de k. Para cada modelo de agrupamento k-Means representamos os coeficientes de silhueta em um gráfico e observamos as flutuações de cada cluster.

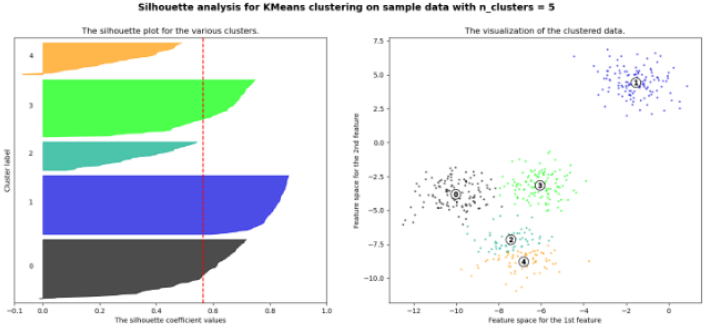


Método de silhueta







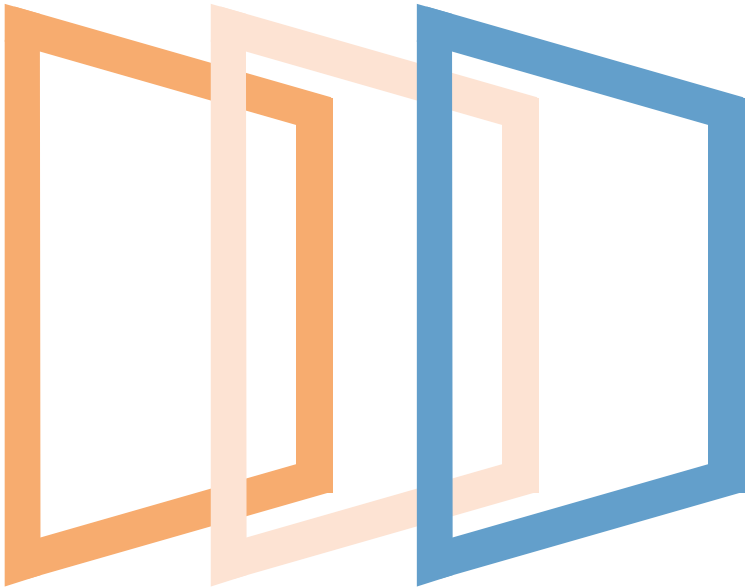


JUPYTER NOTEBOOK
+
ATIVIDADE PRÁTICA

Referências

- Inteligência Artificial - Aulas de Inteligência Artificial (google.com)
- What Really is R2-Score in Linear Regression? | by Benjamin Obi Tayo Ph.D. | Medium
- Métricas de avaliação em machine learning – Diego Mariano
- Supervised vs Unsupervised Learning in 3 Minutes | by Alan Jeffares | Towards Data Science
- CS221 (stanford.edu)
- What is R Squared? R2 Value Meaning and Definition (freecodecamp.org)
- R-Squared: Definition, Calculation Formula, Uses, and Limitations (investopedia.com)

Trilhando Caminhos em Ciência de Dados



Thaís Ratis

Práticas Tecnológicas, 02.09.2023

minsait

An Indra company