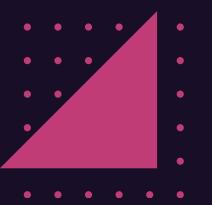
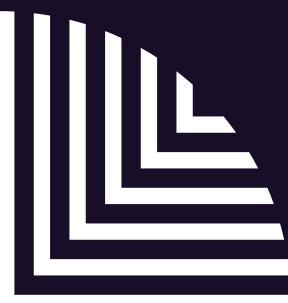
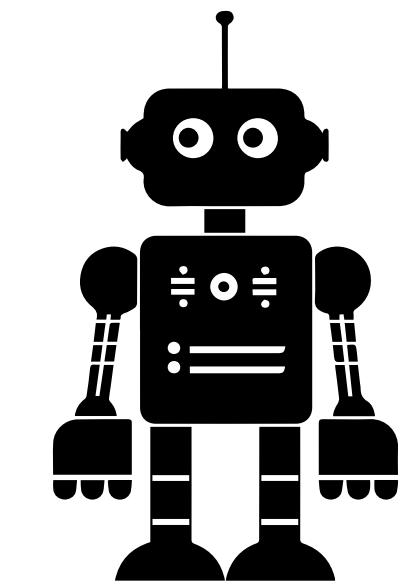


Introdução à Inteligência Artificial Aplicada à Saúde

| Bruno Almeida Silva
| Houemakou Rimaud



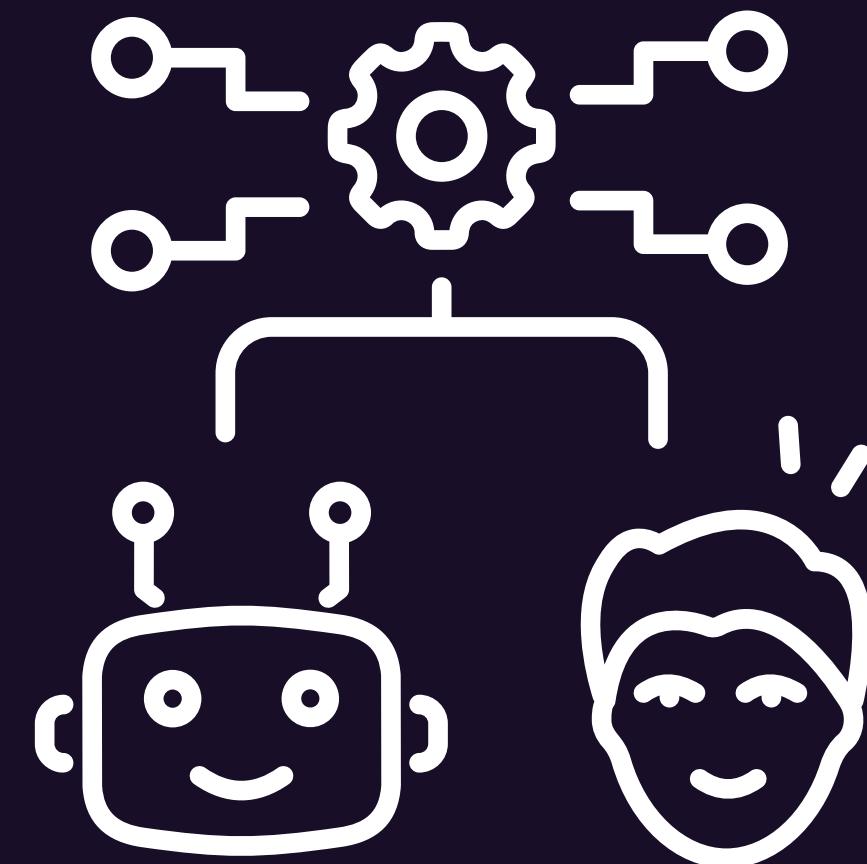
Histórico da Inteligência Artificial (IA)



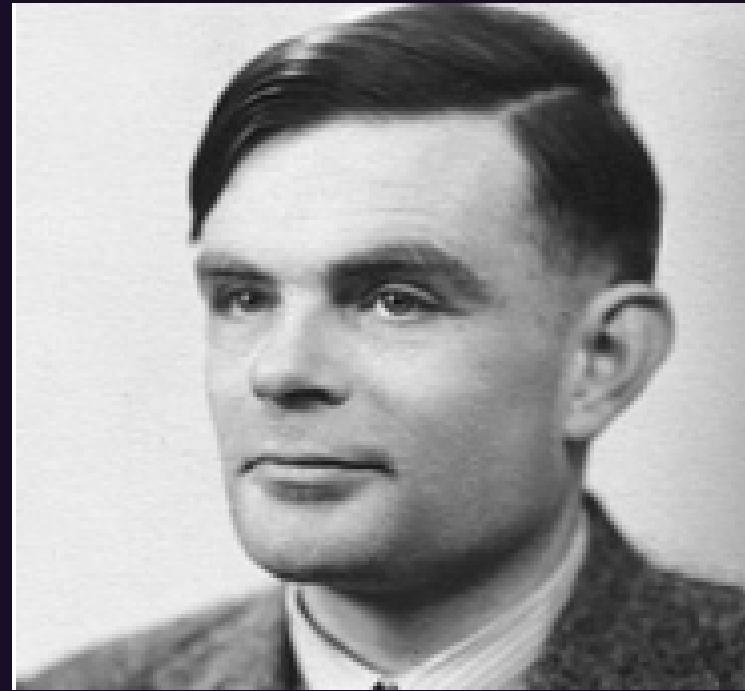
Alan Turing
Pai da Computação

Matemático, lógico, criptoanalista e cientista da computação britânico. Desempenhou um papel importante na criação do computador moderno.

- Em 1950 Alan Turing desenvolveu um método para avaliarmos se tivemos sucesso em criar uma “Inteligência Artificial”, ou seja, se a modelagem de algum comportamento inteligente foi bem-sucedida.
- Esse método foi nomeado **teste de Turing**
- **OBJETIVO:** Avaliar se uma máquina exibe comportamento inteligente.
- **METODOLOGIA:** um jogador humano (em azul) entra em uma conversa, em linguagem natural, com outro ser humano (em laranja) e com uma máquina projetada para produzir respostas indistinguíveis de outro ser humano. Todos os participantes estão separados um dos outros. Se o juiz (em azul) não for capaz de distinguir a máquina do humano, diz-se que a máquina passou no teste, isto é, a máquina é dotada de inteligência artificial



Histórico da Inteligência Artificial (IA)



Alan Turing
Pai da Computação

Matemático, lógico, criptoanalista e cientista da computação britânico. Desempenhou um papel importante na criação do computador moderno.

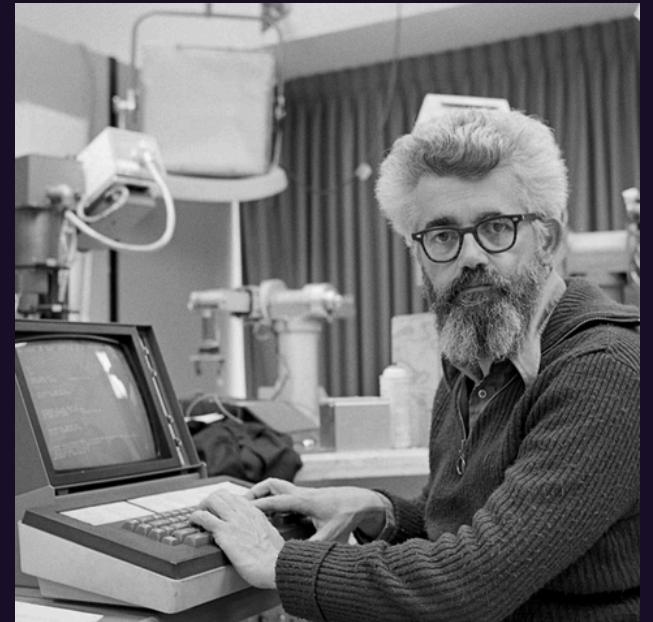
Para passar no teste de Turing, o computador precisaria de diversas habilidades:

- Processamento de linguagem natural: se comunicar com a linguagem humana
- Representação do conhecimento para armazenar o que conhece ou escuta
- Raciocínio lógico para responder às questões e traçar novas conclusões
- Aprendizado de máquina para se adaptar a novas circunstâncias (extrapolar)

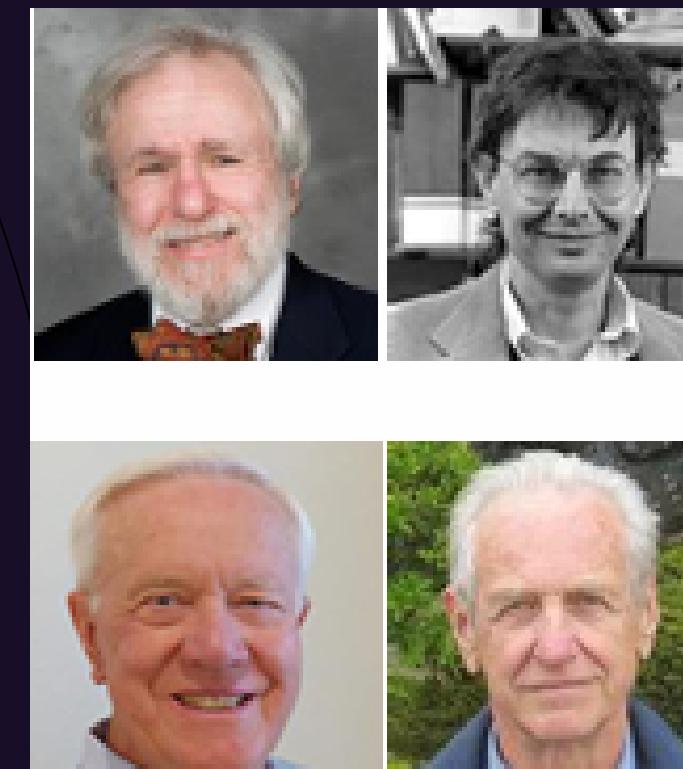
Alguns ainda incluiriam a interação com o ambiente:

- Visão computacional e reconhecimento de voz para perceber o mundo.
- Robótica para manipular objetos e se movimentar. Essas são as principais áreas da Inteligência Artificial

Histórico da Inteligência Artificial (IA)



Feigenbaum e colaboradores definiram IA como o ramo da Ciência da Computação dedicado a desenvolver sistemas computacionais inteligentes, ou seja, que exibem características que nós associamos à inteligência no comportamento humano



Kurzweil define IA como a arte de criar máquinas que executam funções que requerem inteligência quando executadas por pessoas.



1956

Termo “Artificial Intelligence” foi introduzido por John McCarthy durante o Seminário de Dartmouth.



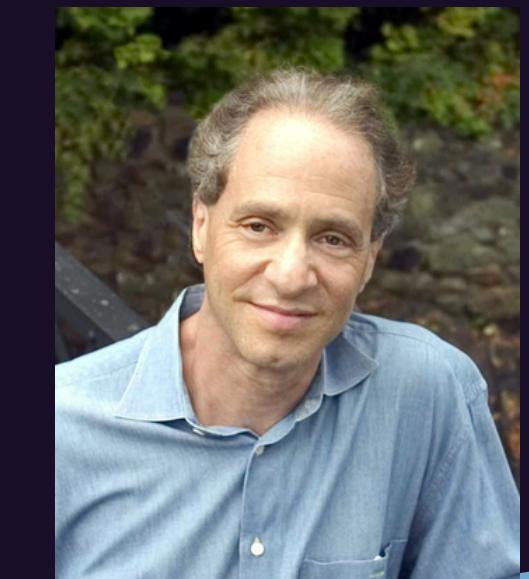
1981



1982

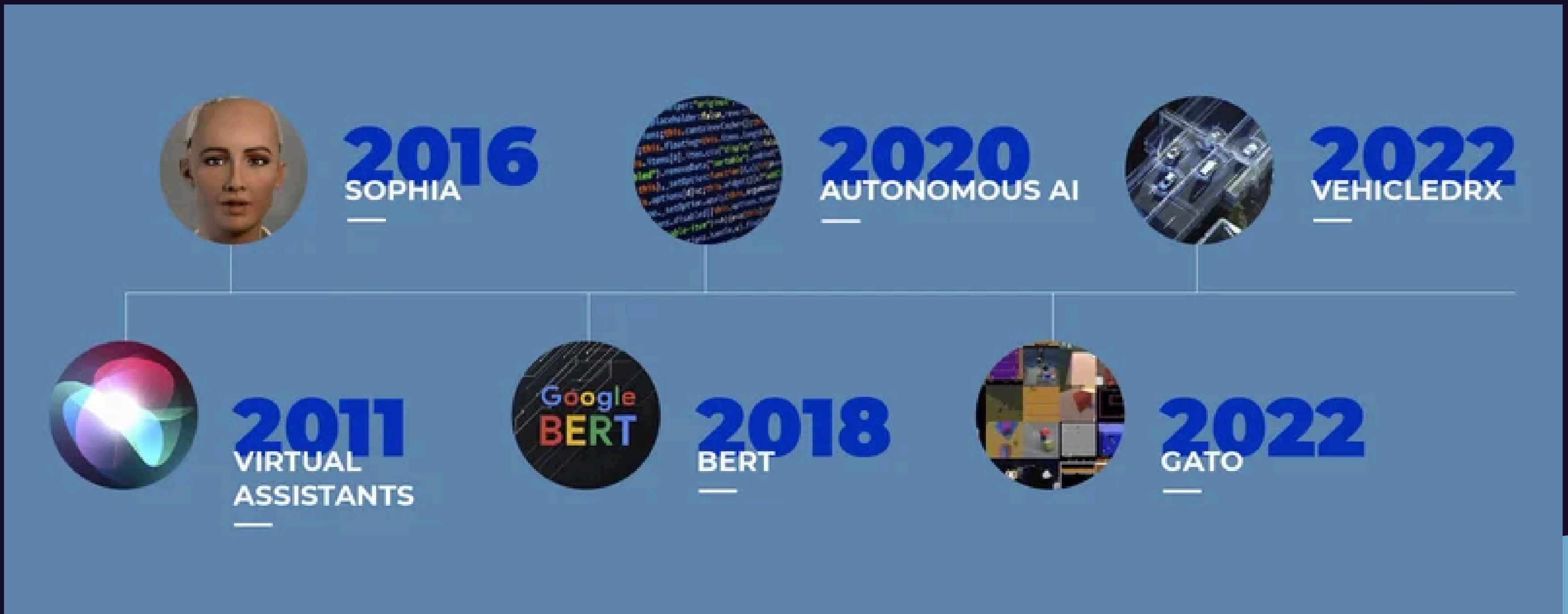


1956



Charniak & McDermott definiram IA como sendo o estudo de faculdades mentais através de modelos computacionais. Nilsson & Genesereth a definiram como o estudo do comportamento inteligente 1987.

Histórico da Inteligência Artificial (IA)



INTELIGÊNCIA ARTIFICIAL



IA é um ramo multidisciplinar da Ciência da Computação que se encontra com áreas como Neurociência, Lógica, Psicologia, Biologia, Engenharia e outras.

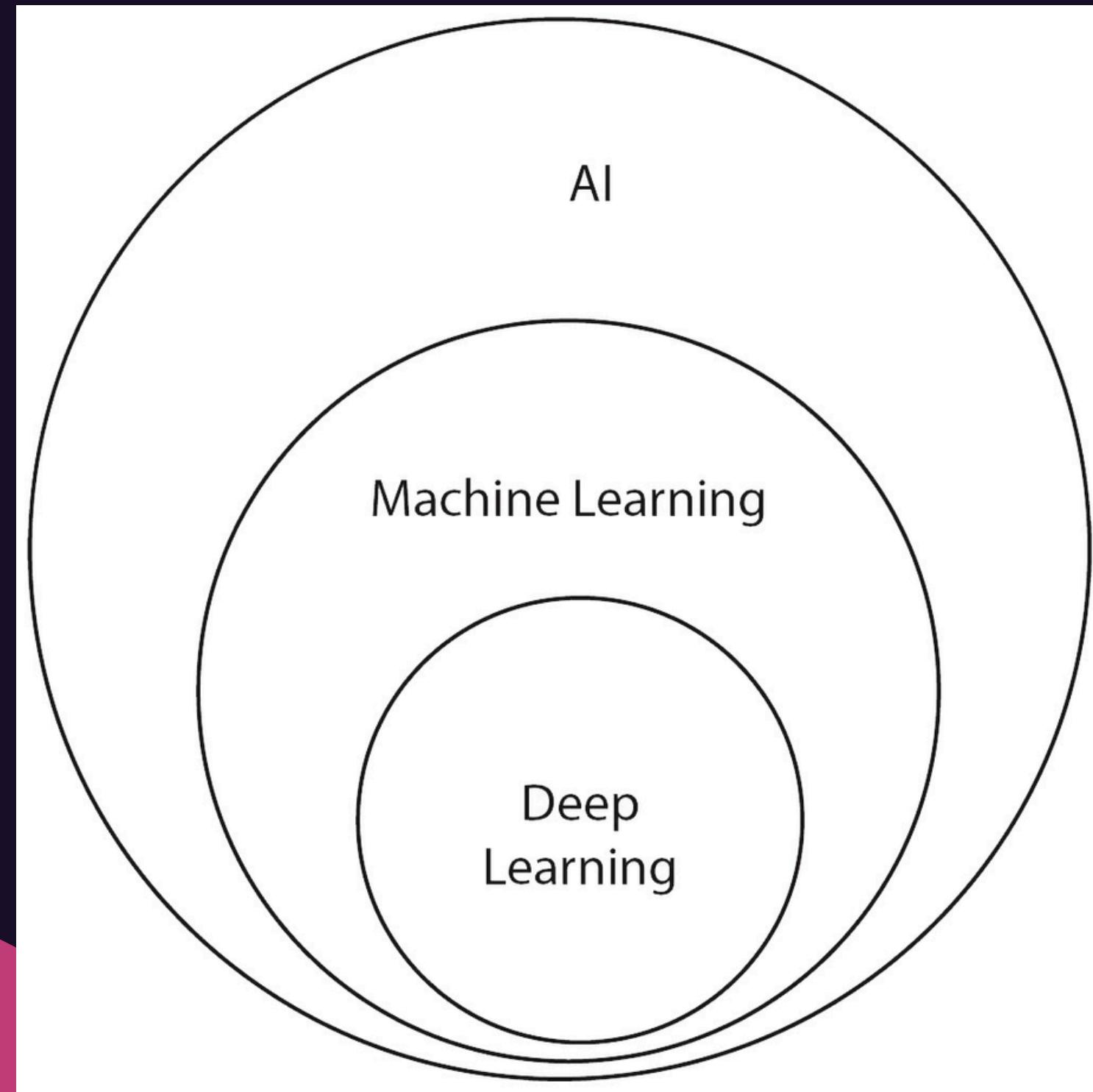
A IA se preocupa em desenvolver sistemas computacionais inteligentes, isto é, sistemas que exibem **características que nós associamos à inteligência no comportamento humano.**

- Compreensão da linguagem;
- Aprendizado;
- Armazenamento de conhecimento;
- Raciocínio;
- Resolução de problemas etc.

Porém, a IA não está limitada a imitar a capacidade humana: em alguns casos, ela é capaz de nos superar.



INTELIGÊNCIA ARTIFICIAL



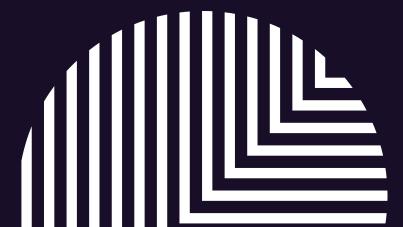


APRENDIZADO DE MÁQUINA

Aprendizado é a capacidade de se adaptar, modificar e melhorar seu comportamento e suas respostas, sendo portanto uma das propriedades mais importantes dos seres ditos inteligentes, sejam eles humanos ou não.

Aprendizado de máquina é o estudo e construção de algoritmos que podem aprender a partir de dados e fazer previsões.

Reconhecimento de padrões



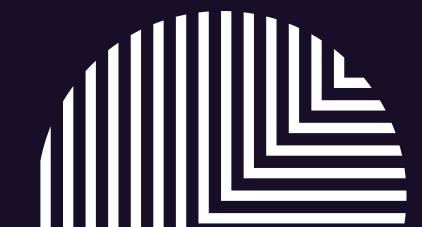
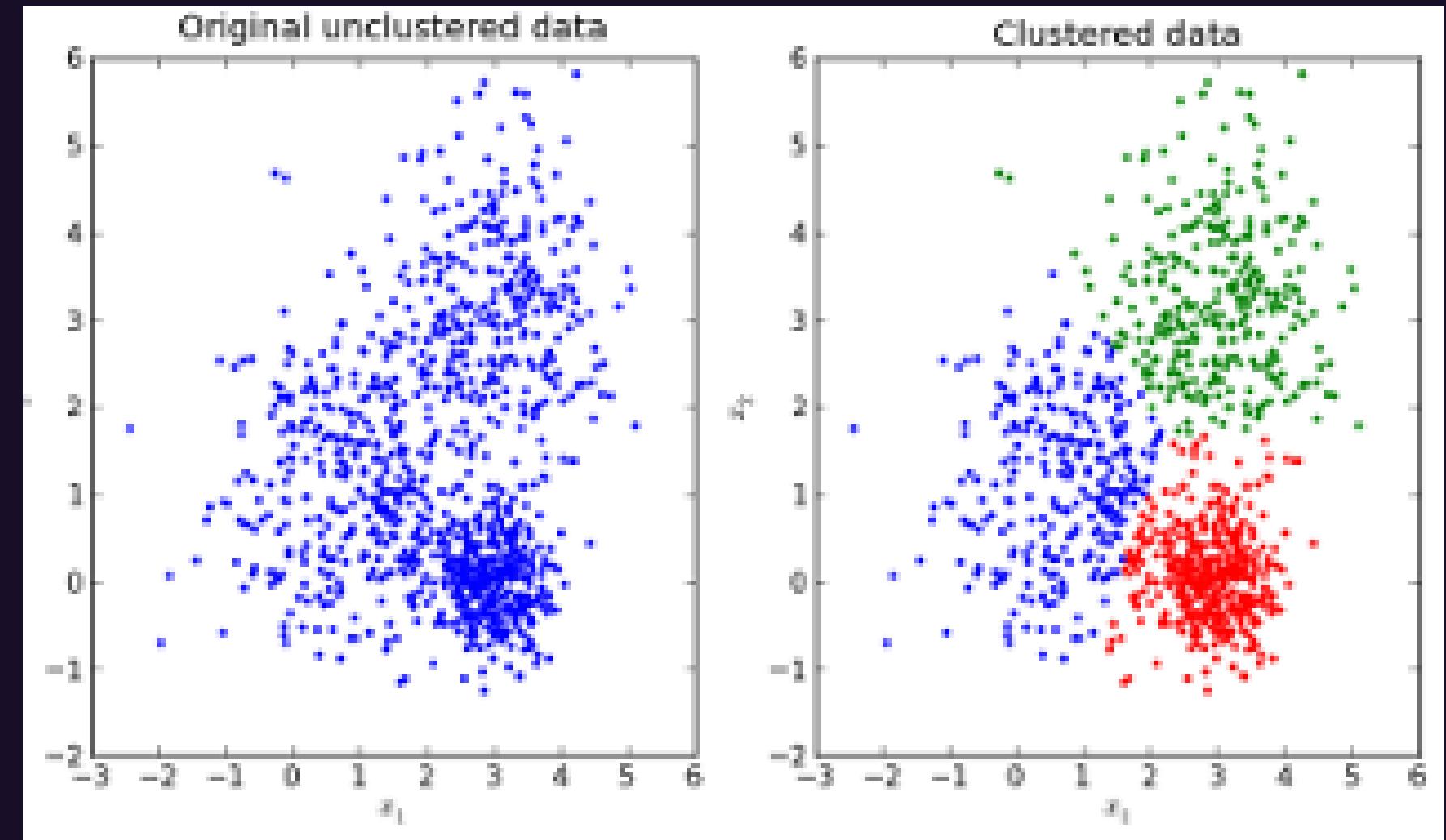


TIPOS DE APRENDIZADO DE MÁQUINA

Aprendizado não supervisionado: análise do comportamento dos dados para encontrar padrões

Aprendizado por reforço: aprende a partir de experiências, interações com o ambiente (causa e efeito/punição e recompensa)

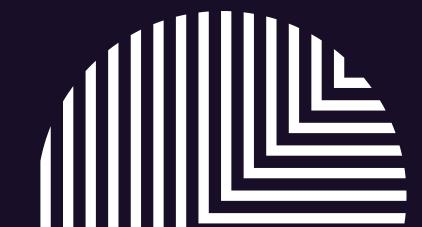
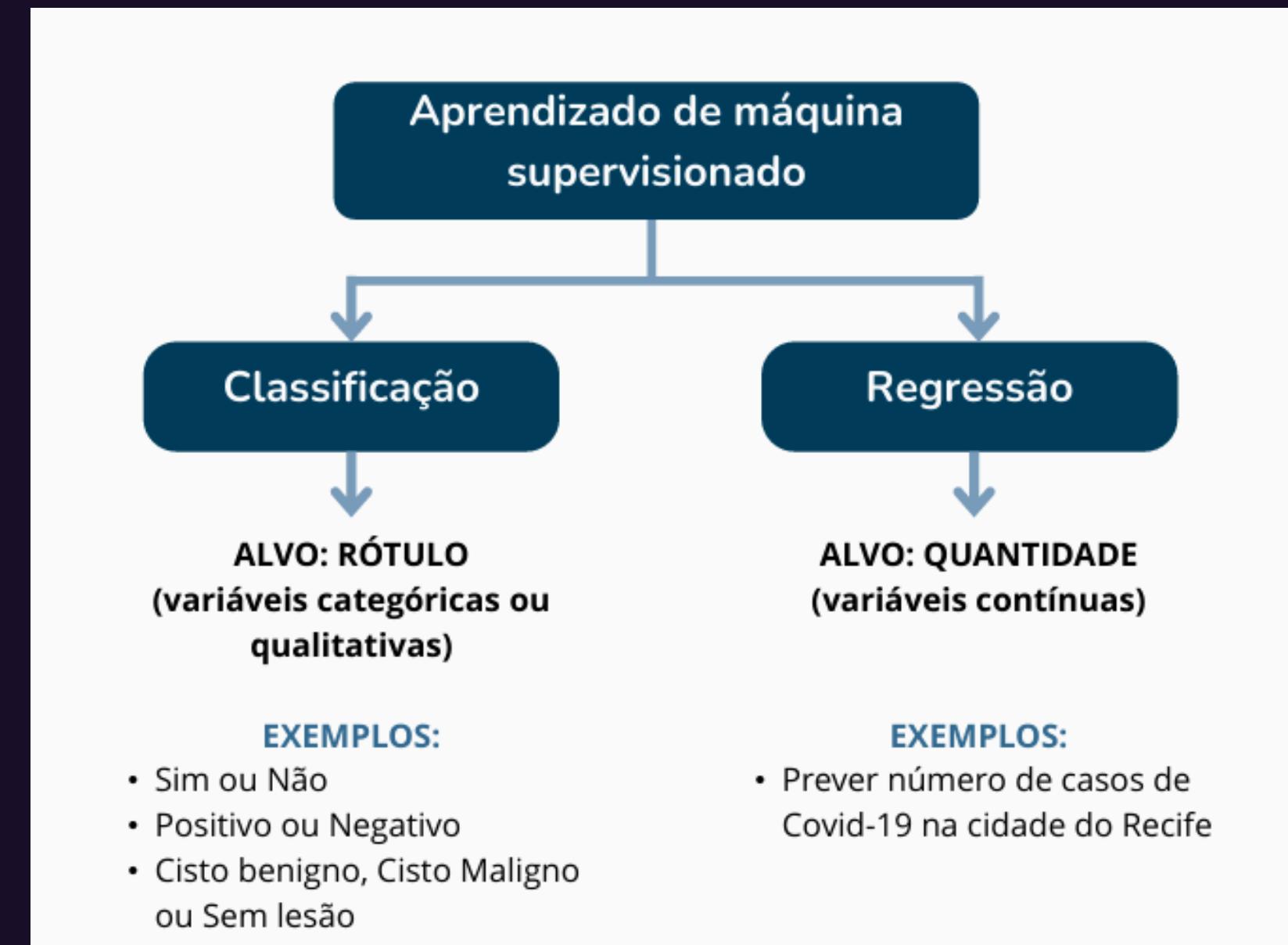
Aprendizado supervisionado: entradas e saídas fornecidas





Aprendizado de Máquina Supervisionado

Aplicado quando os dados a serem avaliados são classificados por meio de um rótulo e com isso o algoritmo consegue fazer previsões e procurar por padrões nos dados fornecidos





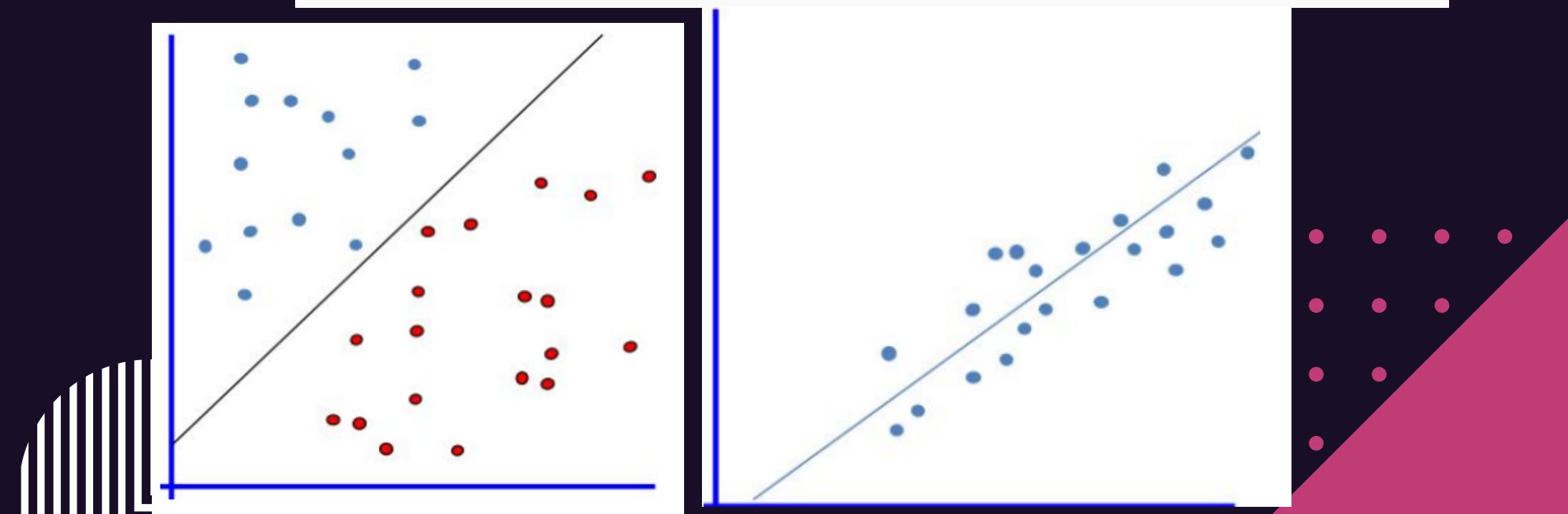
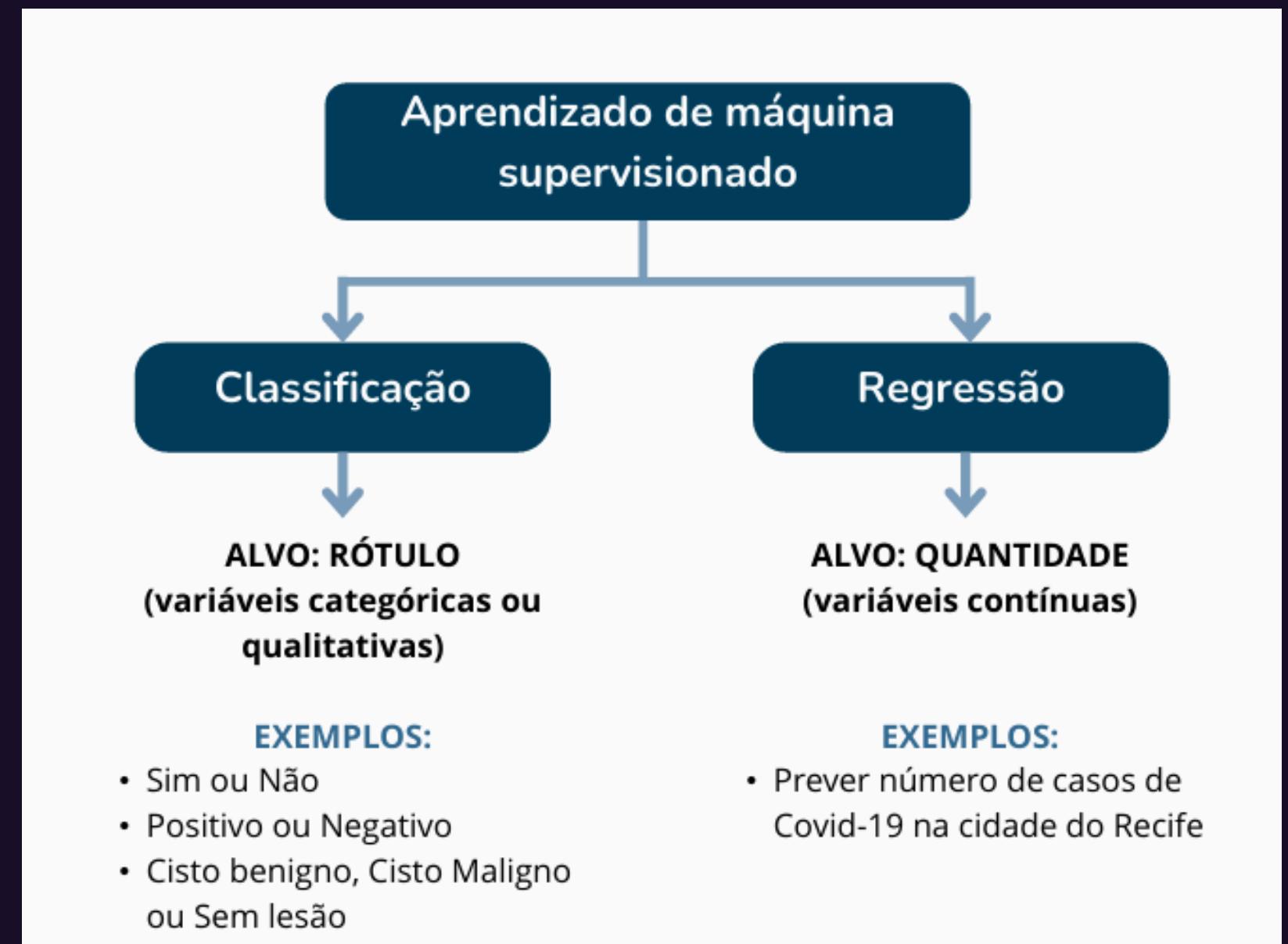
APRENDIZADO DE MAQUINA SUPERVISIONADO

• Métodos Paramétricos

- Assumem que a distribuição dos dados é conhecida (distribuição normal por exemplo)
- Em muitos casos não se tem conhecimento da distribuição

• Métodos Não-Paramétricos

- Não consideram essa hipótese
- Um exemplo é o k-NN (k Nearest Neighbor)

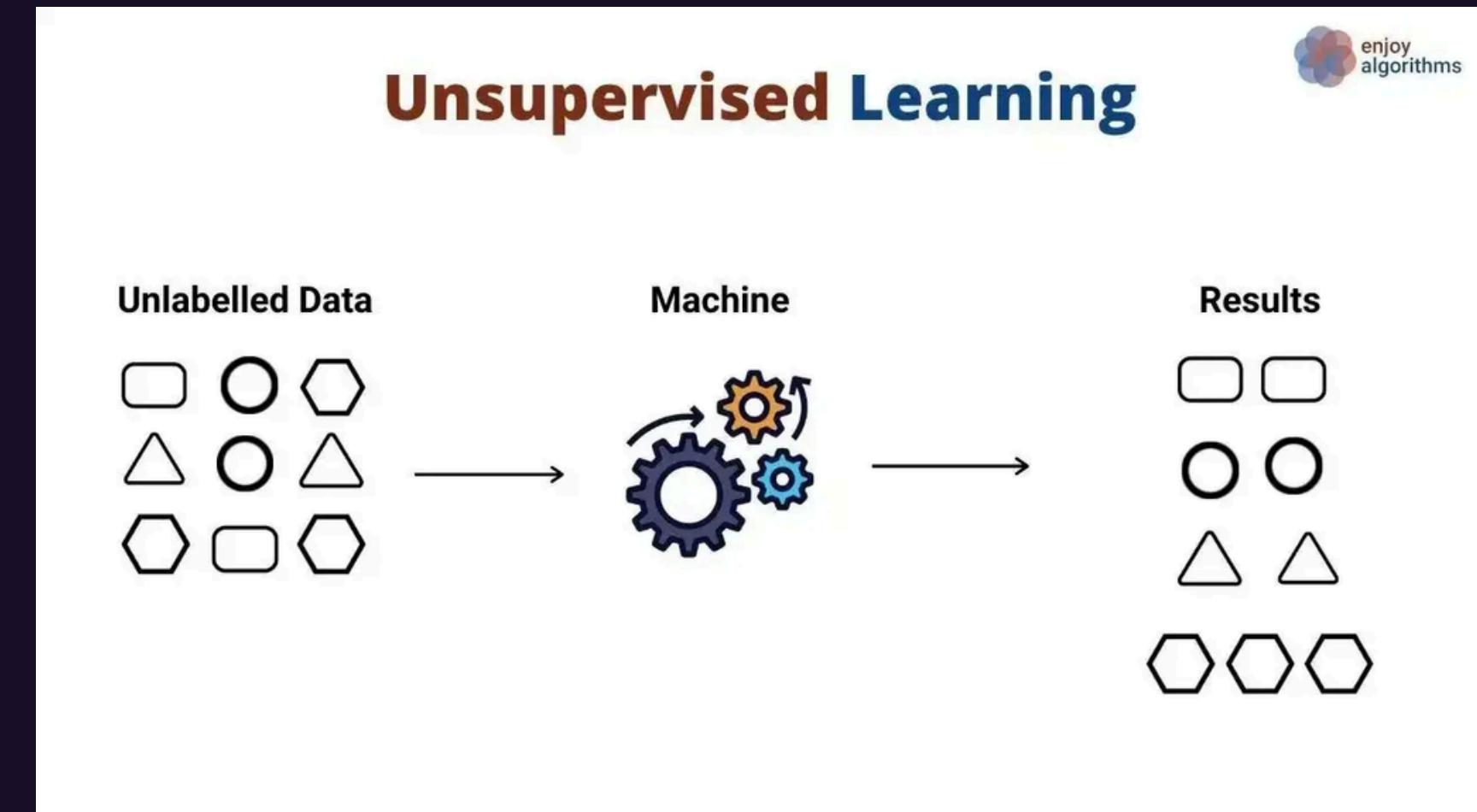




APRENDIZADO DE MAQUINA NÃO SUPERVISIONADO

Utiliza algoritmos de aprendizado de máquina (ML) para analisar e agrupar conjuntos de dados sem rótulos. Esses algoritmos descobrem padrões ocultos ou agrupamentos de dados sem a necessidade de intervenção humana.

- Não tem “crítico” ou “professor” externo, apenas os dados de entrada.
- Tem-se um conjunto de exemplos mas não se conhece as categorias envolvidas.
- Busca extrair as propriedades estatisticamente relevantes



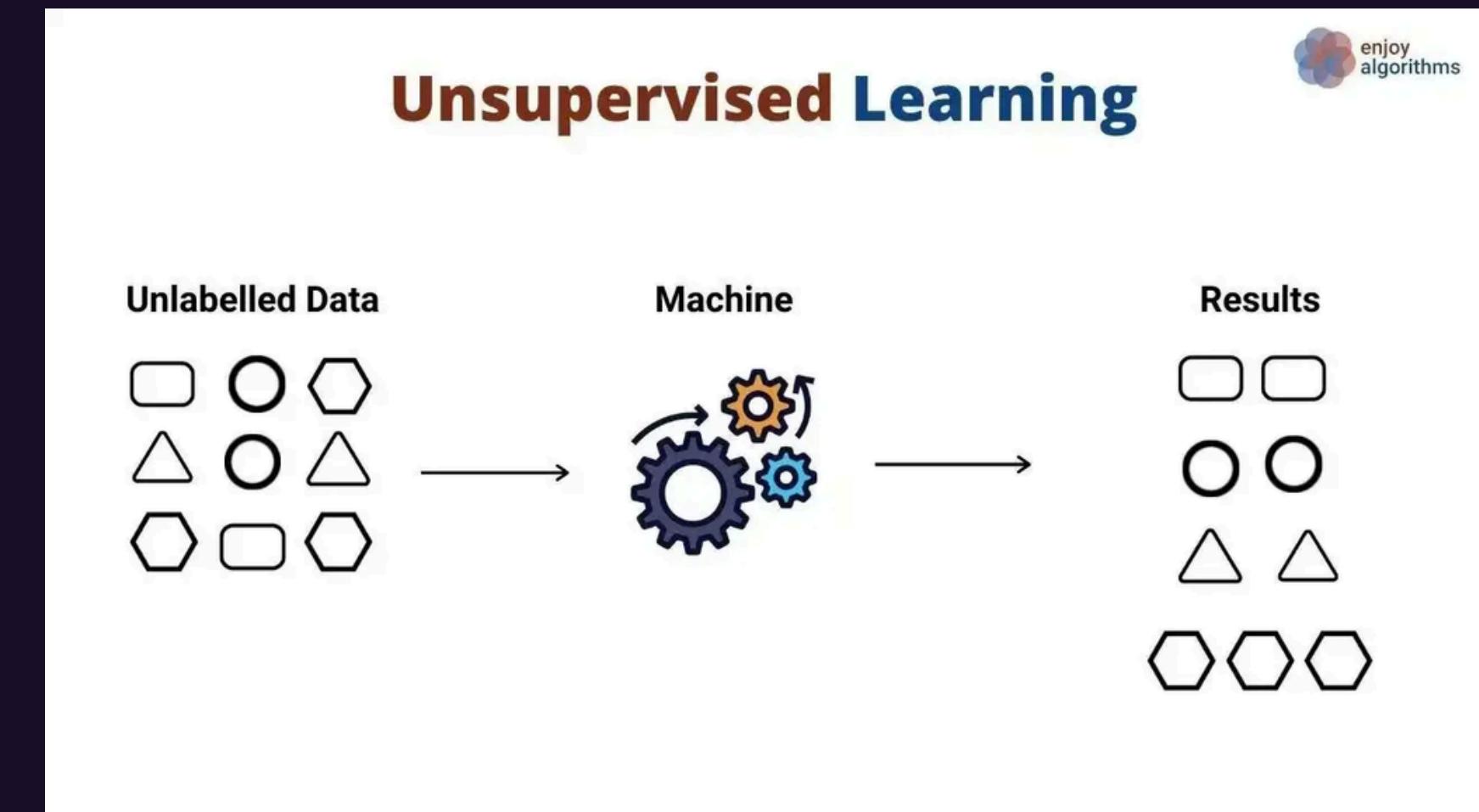


APRENDIZADO DE MAQUINA NÃO SUPERVISIONADO

Clustering

- Organização dos objetos similares (em algum aspecto) em grupos
- Descobre categorias automaticamente

Quantização: atribui valores discretos para um atributo que aceita infinitos valores



METODOLOGIA DE APLICAÇÃO DE MODELO DE ML



- 01** **Preparação e Pré-processamento da base de dado**
- 02** **Extração e seleção de caraterísticas (atributos)**
- 03** **Treinamento, Validação e teste dos models**
- 04** **Aplicação do modelo treinado**



APRENDIZADO DE MAQUINA SUPERVISIONADO

Preparação e Pré-processamento da base de dados

Consiste em algum tipo de preparação dos dados para que eles sejam melhor interpretados. Comumente o pré-processamento é realizado para melhorar a qualidade ou a integridade do dado de entrada.

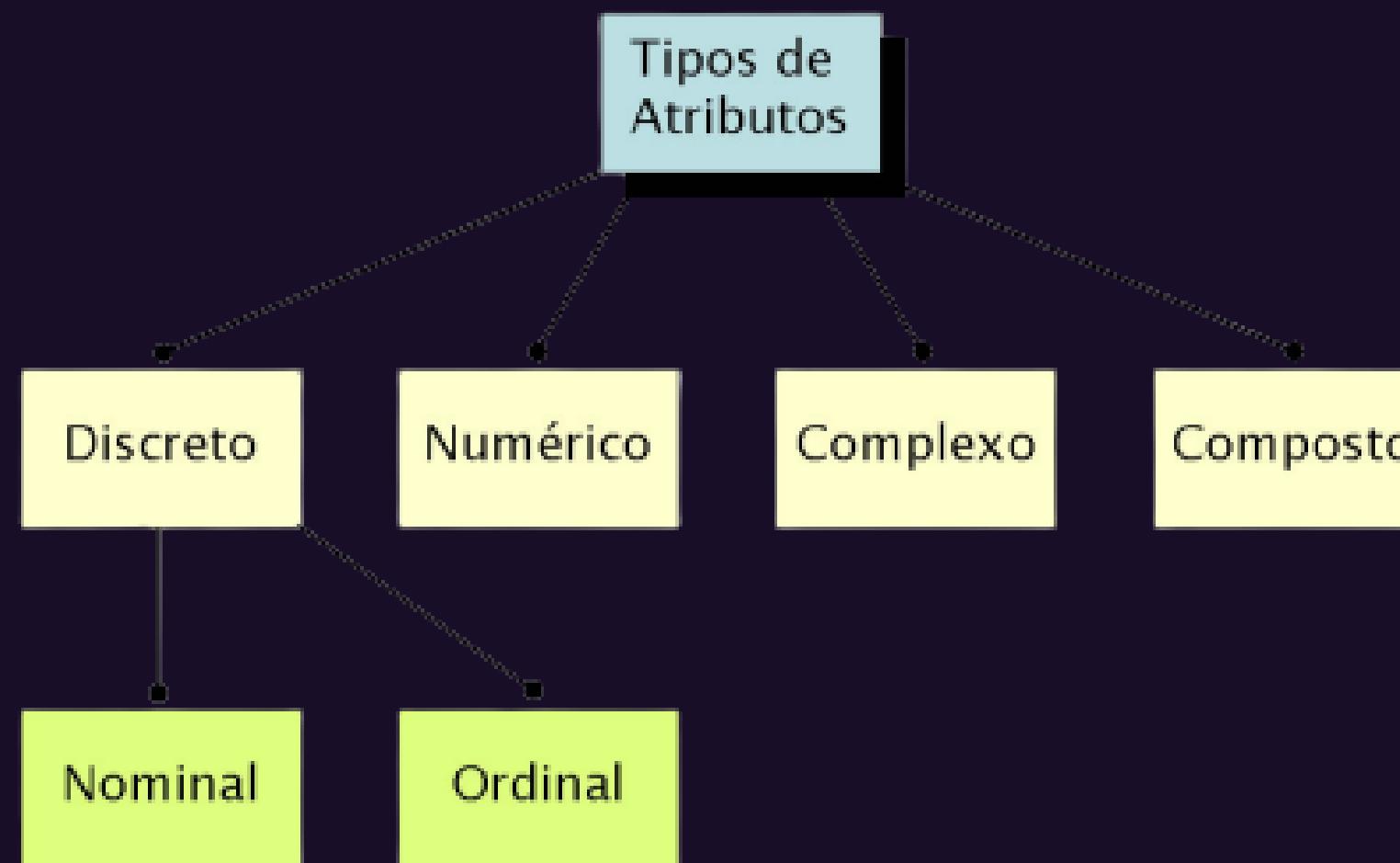
- **Aquisição dos dados (com os metadados)**
- **Limpeza e Ajuste de dados**
- **Transformação e padronização**
- **Redução de Ruidos, Filtragem**
- **Extração de características (features)**





APRENDIZADO DE MAQUINA

Extração e seleção de caraterísticas (atributos)



Columns

ROWS

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Cada coluna (column) representa um atributo do conjunto de dados





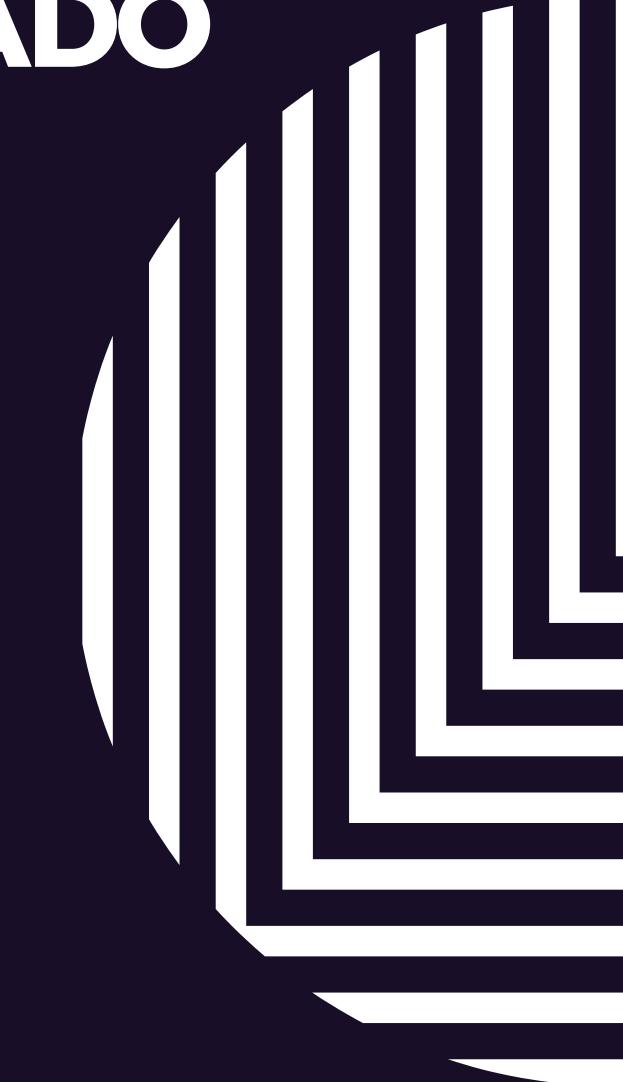
APRENDIZADO DE MAQUINA SUPERVISIONADO

Extração e seleção de características (atributos)

A seleção de atributos, ou Feature Selection, consiste na obtenção de um subconjunto de dados a partir de características consideradas relevantes para descrever aquele dado.

- Ela consiste na redução da dimensionalidade dos dados através da seleção de um subconjunto de atributos mais importantes para o processo de predição ou classificação.
- Reduzir a complexidade e a dimensionalidade do problema.
- **Evitar overfitting dos modelos!**

Existem dois principais extractores bastante utilizados na literatura, são eles: A análise de componentes principais (PCA) e a análise discriminante linear (LDA).





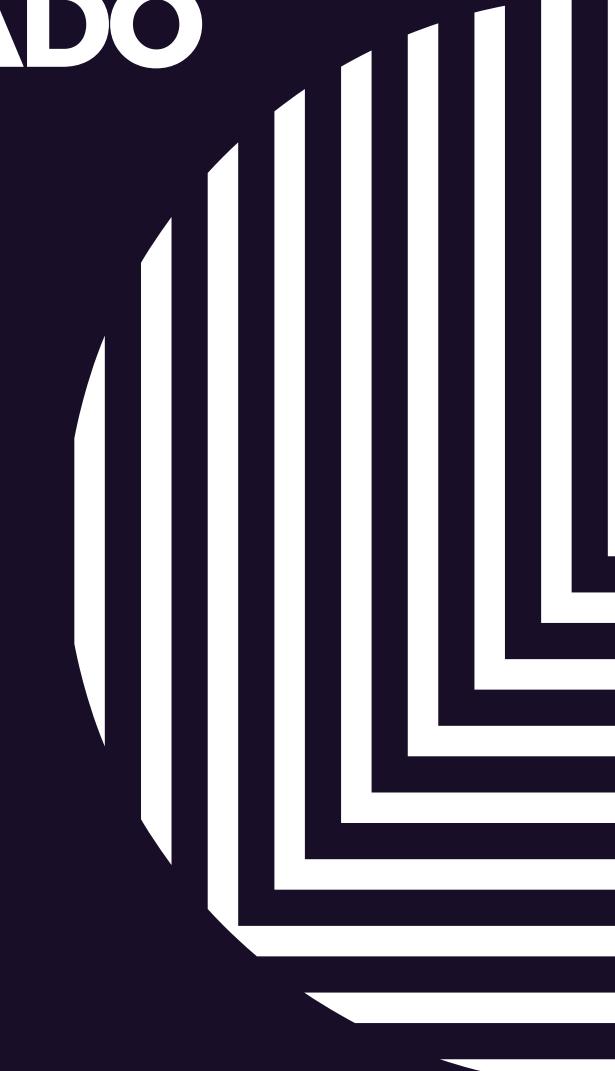
APRENDIZADO DE MAQUINA SUPERVISIONADO

Extração e seleção de caraterísticas (atributos)

O PCA consiste em obter novas variáveis a partir dos atributos iniciais obtendo um pequeno número de componentes principais (combinações lineares) de um conjunto de variáveis.

O objetivo não é apenas reduzir e sim conseguir preservar o máximo possível das informações contidas nas variáveis originais.

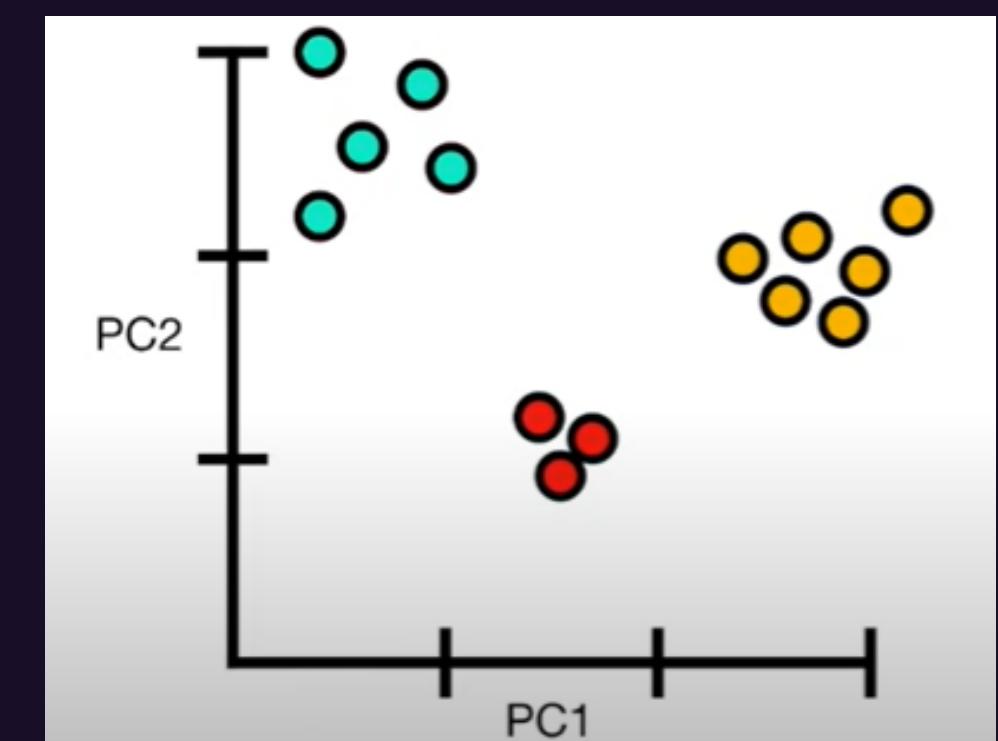
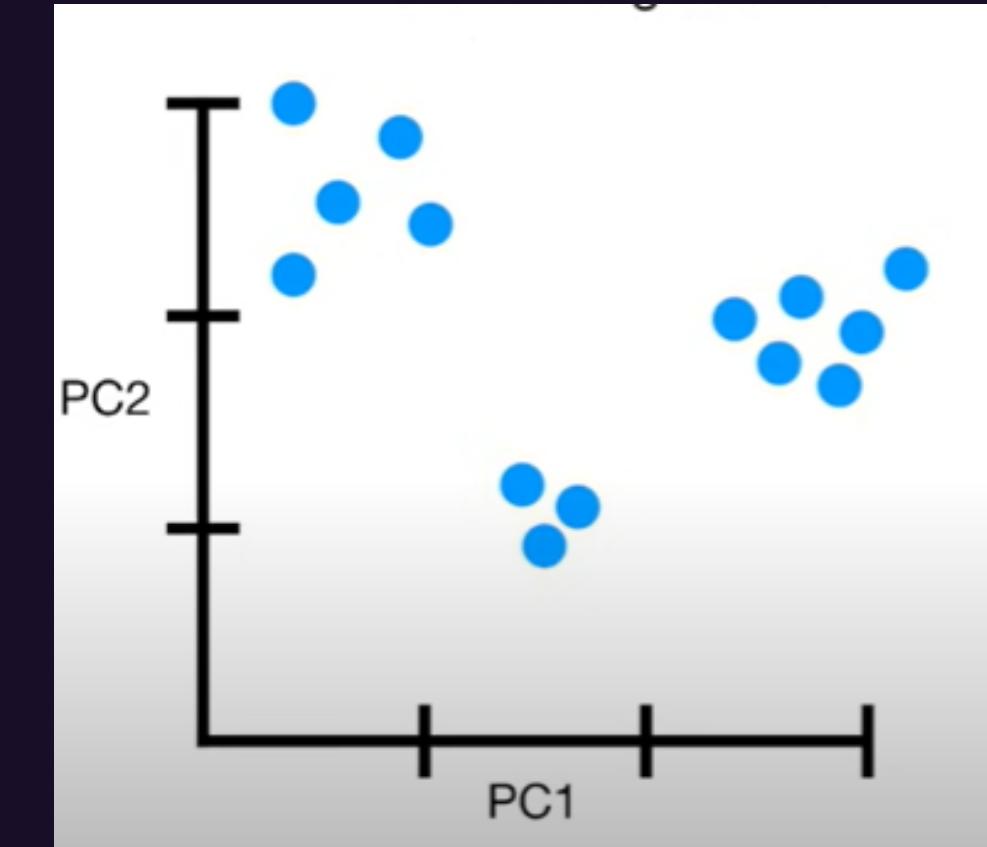
- Media de todos os valores dados
- Normalização ou padronização de dados
- Calculo da matriz de covarianca
- Extração dos autovetores e autovalores desta matriz
- É escolhido os k autovetores com maior quantidades de info associadas, ou seja, o autovetor com maior autovalor associado.
- Monta-se a matriz de transformações baseada nos autovetores selecionados previamente.



APRENDIZADO DE MAQUINA SUPERVISIONADO

Extração e seleção de caraterísticas (atributos)

	Cell1	Cell2	Cell3	Cell4	...
Gene1	3	0.25	2.8	0.1	...
Gene2	2.9	0.8	2.2	1.8	...
Gene3	2.2	1	1.5	3.2	...
Gene4	2	1.4	2	0.3	...
Gene5	1.3	1.6	1.6	0	...
Gene6	1.5	2	2.1	3	...
Gene7	1.1	2.2	1.2	2.8	...
Gene8	1	2.7	0.9	0.3	...
Gene9	0.4	3	0.6	0.1	...





APRENDIZADO DE MAQUINA SUPERVISIONADO

Extração e seleção de caraterísticas (atributos)

O LDA gera um novo conjunto de dados de menor dimensionalidade que representa as classes dos dados originais, minimizando a dispersão entre os registros da mesma classe e maximizando a distância entre as classes.

- Inicialmente é calculada a média de todos os valores dados.
- Todo o conjunto de dados é normalizado.
- Após a normalização, é calculado a matriz de covariância.
- Em seguida, é calculada a matriz de covariância conjunta.
- É calculada a inversa da matriz de covariância conjunta. Aplicar a função discriminante.
- Atribuir um objeto a um grupo que maximize a função discriminante.





Treinamento, Validação e teste dos models

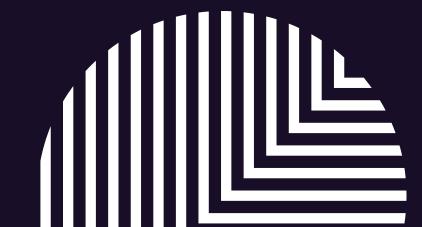
Inicialmente, o modelo irá compreender os dados e reconhecer os padrões associados.

Reconhecimento de padrões:

- Identificação e classificação de padrões em dados
- Atribuir um rótulo (ou classe) para uma certa amostra ou valor de entrada

Aplicações no dia a dia:

- Reconhecimento de faces, de fala, escrita
- Compreensão da linguagem
- Diferenciar objectos
- Saber que uma fruta está madura ou não



Treinamento, Validação e teste dos models

Reconhecimento de padrões:

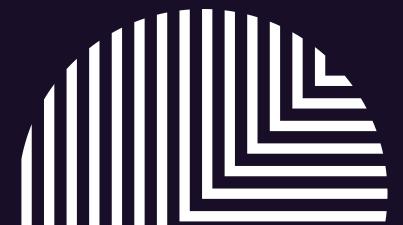
- Identificação e classificação de padrões em dados
 - Atribuir um rótulo (ou classe) para uma certa amostra ou valor de entrada

 - Anote as características entre os dois peixes
 - Comprimento
 - Lividez
 - Largura
 - Número e forma das espinhas
 - Posição da boca
 - Coloração
- ↓
- **Extração de fatores (Redução de dados pela medição de certas “características” ou “propriedades”)**
 - **Classificação:**



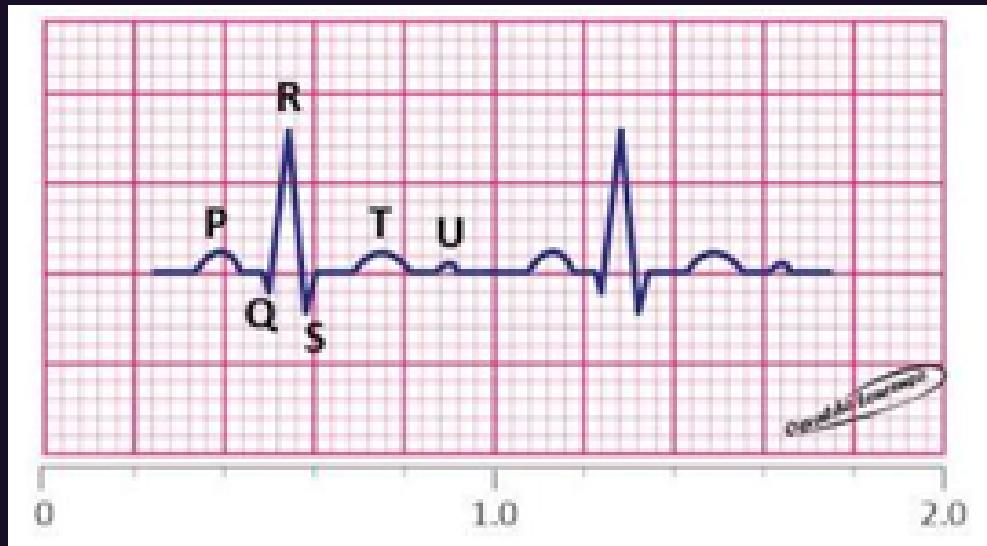
•

..
..
..

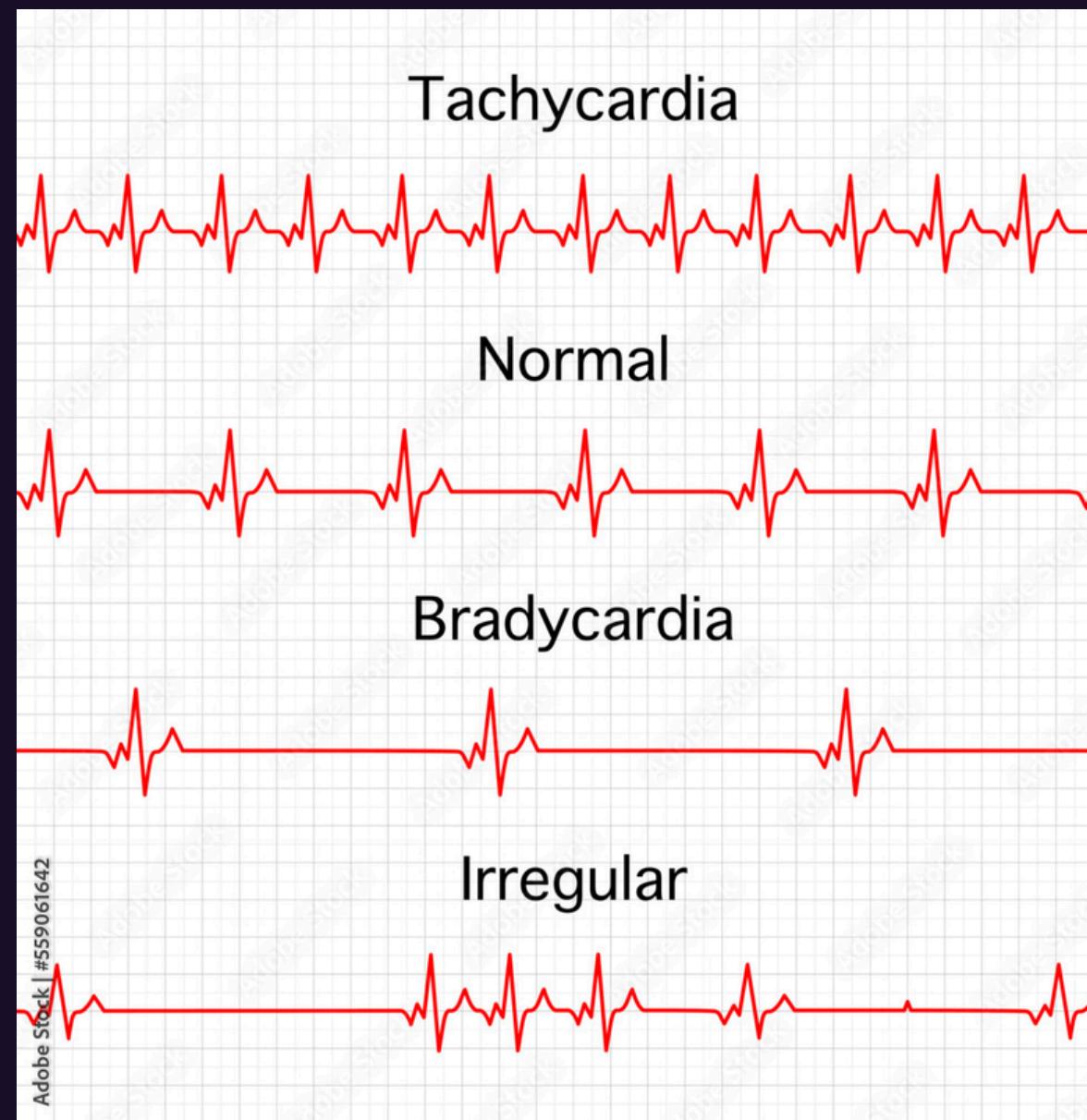


Treinamento, Validação e teste dos models

Reconhecimento de padrões:



Padrão ECG



- O sinal de ECG possui padrões característicos.
 - O médico olha os traçados (ondas P, QRS, T) e reconhece padrões de forma, duração e ritmo.
 - Reconhecimento é baseado em repetição de padrões rítmicos.
- Exemplo: um QRS muito largo sugere bloqueio de ramo; ondas P irregulares sugerem fibrilação atrial.
- Monitores multiparamétricos emitem alertas quando o sinal está fora da normalidade.

• • •
• • •
• •



Treinamento, Validação e teste dos models

Divisão Percentual x Validação Cruzada Balanceamento de Classes

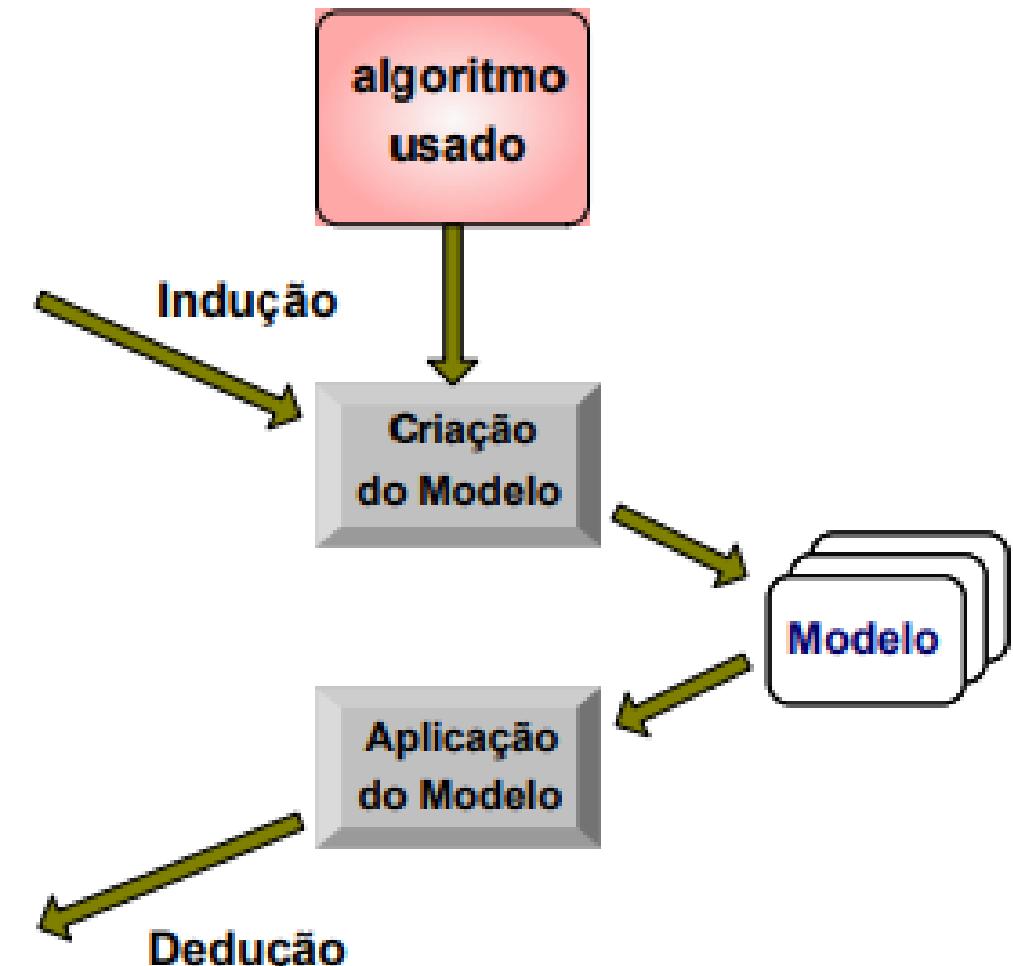
- **Treinamento:**
 - Processo em que os neurônios aprendem;
 - Mapeiam entradas nas saídas com um erro aceitável.
- **Teste**
 - Testar o modelo com dados que não participam do treinamento;
 - Saídas são calculadas para os novos dados e comparados com as saídas esperadas.

Tid	Atrib1	Atrib2	Atrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	80K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Conjunto de treinamento

Tid	Atrib1	Atrib2	Atrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de teste

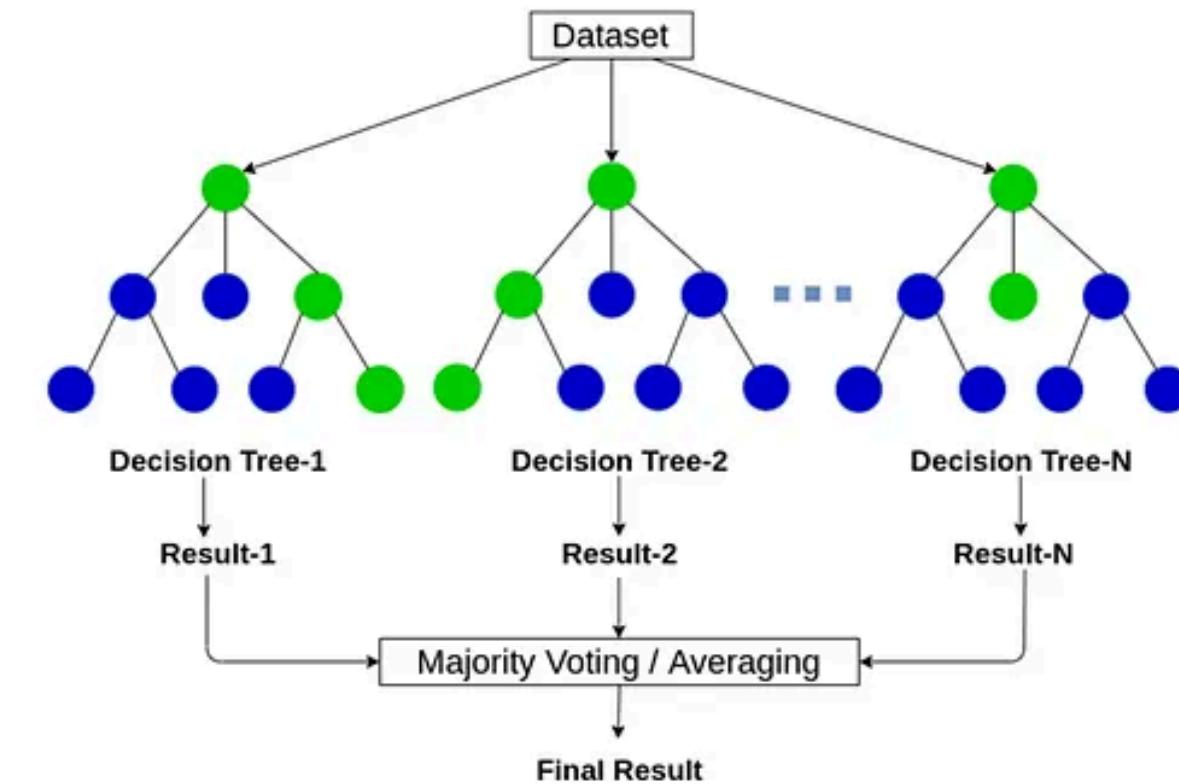


Treinamento, Validação e teste dos models

Árvores de decisão

- **Métodos de aprendizado de máquina supervisionados e não paramétricos, muito utilizados em tarefas de classificação e regressão;**
- **Árvores, em computação, são estruturas de dados formadas por um conjunto de elementos que armazenam informações chamados nós.**
- **Toda árvore possui um nó raiz, com o maior nível hierárquico (o ponto de partida) e ligações para outros elementos, denominados nós filhos;**
- **Esses filhos podem possuir seus próprios filhos:- Quando possui: nó de decisão;- Quando não possui: nó folha ou nó de término.**

Random Forest





Treinamento, Validação e teste dos modelos

Árvores de decisão

- **Amostragem Bootstrap**
- **Random Feature Selection**
- **Crescimento das Árvores**
- **Aggregação (Majority Voting / Averaging)**

Paciente	Idade	Febre	Tosse	Dor no corpo	Contato	Congestão	Diagnóstico (Gripe?)
P1	Adulto	Sim	Sim	Sim	Sim	Sim	SIM
P2	Jovem	Não	Não	Não	Não	Não	NÃO
P3	Idoso	Sim	Sim	Sim	Não	Não	SIM
P4	Adulto	Não	Sim	Não	Sim	Sim	SIM
P5	Jovem	Sim	Não	Sim	Sim	Não	SIM
P6	Idoso	Não	Não	Não	Não	Não	NÃO
P7	Adulto	Sim	Sim	Não	Não	Sim	SIM
P8	Jovem	Não	Sim	Sim	Sim	Sim	SIM

APRENDIZADO DE MAQUINA NÃO SUPERVISIONADO

Dados Complexos e Multidimensionais Encontrando a Ordem no Caos



Em biologia e ciência de dados, frequentemente temos tabelas com dezenas ou centenas de variáveis (colunas) para cada amostra. É como estar em uma floresta densa: é difícil enxergar os padrões gerais porque há muitas árvores (variáveis) na sua frente.

Como podemos simplificar essa complexidade sem perder a informação mais importante? Como podemos encontrar grupos naturais de amostras que se parecem entre si?

Usaremos duas técnicas poderosas:

- **PCA (Análise de Componentes Principais): Para simplificar a "floresta".**
- **Clusterização (K-Means): Para encontrar os "grupos de árvores" semelhantes.**

	GC_Content	AT_Content	Num_A	Num_T	Num_C	Num_G	kmer_3_freq	Mutation_Flag
PLE_1	-0.02370282	0.02370282	-0.68342679	0.71118072	-1.3805294513	1.3590162	1.69526343	-0.9931898
PLE_2	-1.01132046	1.01132046	0.47466198	0.71118072	-0.6907245956	-0.4887285	-0.23881335	1.0065212
PLE_3	0.17382070	-0.17382070	0.24304423	-0.44684260	1.1487550198	-0.9506646	-0.69912363	1.0065212
PLE_4	0.96391481	-0.96391481	0.70627974	-1.83647057	-0.4607896436	1.5899842	-0.55600195	-0.9931898
PLE_5	-0.81379693	0.81379693	-0.22019128	1.17439004	-0.0009197398	-0.9506646	1.04541363	-0.9931898
PLE_6	-1.01132046	1.01132046	0.93789749	0.24797139	0.2290152121	-1.4126008	-0.20399997	1.0065212
PLE_7	2.34657950	-2.34657950	-2.99960433	0.24797139	2.2984297794	0.4351439	0.08611155	1.0065212
PLE_8	-0.81379693	0.81379693	0.24304423	0.71118072	-0.4607896436	-0.4887285	-0.49411149	-0.9931898
PLE_9	-1.60389104	1.60389104	1.63275075	0.24797139	-1.3805 -0.4607896436	35	1.49798759	1.0065212
PLE_10	1.75400892	-1.75400892	-1.60989780	-0.44684260	1.3786899717	0.6661119	0.22536507	1.0065212
PLE_11	-0.22122635	0.22122635	-1.60989780	1.86920403	-0.6907245956	0.4351439	-1.43020465	-0.9931898
PLE_12	-1.20884399	1.20884399	1.16951525	0.24797139	0.2290152121	-1.6435688	-0.83837716	-0.9931898
PLE_13	-0.02370282	0.02370282	-0.68342679	0.71118072	0.2290152121	-0.2577604	-1.62361233	-0.9931898
PLE_14	-0.22122635	0.22122635	0.47466198	-0.21523793	-0.6907245956	0.4351439	-1.09367529	1.0065212
PLE_15	-0.02370282	0.02370282	0.01142648	0.01636673	-0.2308546917	0.2041758	0.84426964	-0.9931898
PLE_16	0.37134423	-0.37134423	0.24304423	-0.67844726	-0.2308546917	0.6661119	-0.09182352	-0.9931898
PLE_17	2.34657950	-2.34657950	-0.68342679	-2.06807523	0.9188200679	1.8209523	1.30457992	-0.9931898

APRENDIZADO DE MAQUINA NÃO SUPERVISIONADO

PCA: Criando um "Resumo" Inteligente dos Seus Dados

PCA é uma técnica de redução de dimensionalidade. Ela transforma um grande número de variáveis em um número menor de "variáveis artificiais" chamadas Componentes Principais (PCs).

- **Imagine que você quer representar uma cadeira 3D em uma folha de papel 2D. A melhor maneira é projetar sua sombra na parede. Você perde uma dimensão, mas a sombra ainda captura a forma principal da cadeira. A PCA faz exatamente isso com os dados: ela projeta a "sombra" mais informativa dos seus dados multidimensionais em um espaço com menos dimensões.**
- **Objetivo: Simplificar os dados para facilitar a visualização e identificar as principais fontes de variação.**



Como a PCA Funciona?

- **Componente Principal 1 (PC1):** A PCA encontra a "melhor linha" que pode ser traçada através dos seus dados, de forma a capturar a maior quantidade de variação possível. Esta linha é o PC1. Ele é o "resumo" mais importante dos seus dados.
- **Componente Principal 2 (PC2):** Em seguida, a PCA encontra uma segunda linha, perpendicular à primeira, que captura a maior parte da variação restante. Este é o PC2.
- **Resultado:** Em vez de analisar 10 ou 20 variáveis, agora podemos olhar para apenas 2 (PC1 e PC2) e ainda assim ter uma ótima ideia de como nossas amostras estão distribuídas.

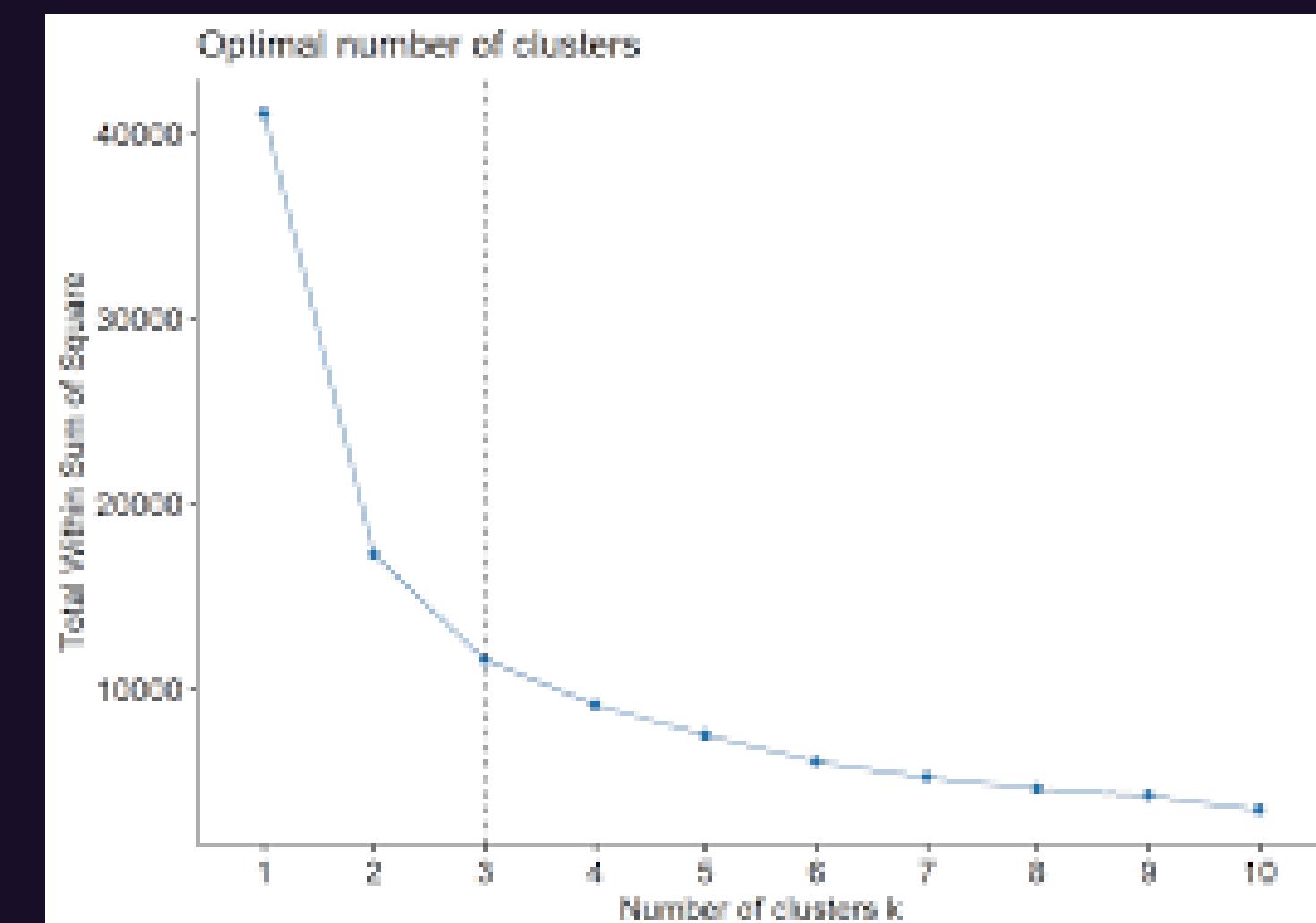
O que é a Análise de Cluster (Clusterização)?

- **A Grande Ideia:** Clusterização é uma técnica de aprendizado não supervisionado. "Não supervisionado" significa que nós não dizemos ao algoritmo quais são as respostas. Em vez disso, pedimos a ele: "Olhe para estes dados e encontre grupos de amostras que são mais parecidas entre si do que com as outras".
- **Analogia da Cesta de Roupas:** Imagine uma cesta cheia de meias de cores diferentes. A clusterização é o processo de separar as meias em montinhos por cor (um monte de meias pretas, um de brancas, etc.), sem que ninguém te diga quais cores existem.
- **Objetivo:** Identificar subgrupos ou padrões naturais nos dados.

MÉTODO DO COTOVELO (Elbow Method)

FUNÇÃO: `fviz_nbclust(..., method = "wss")`

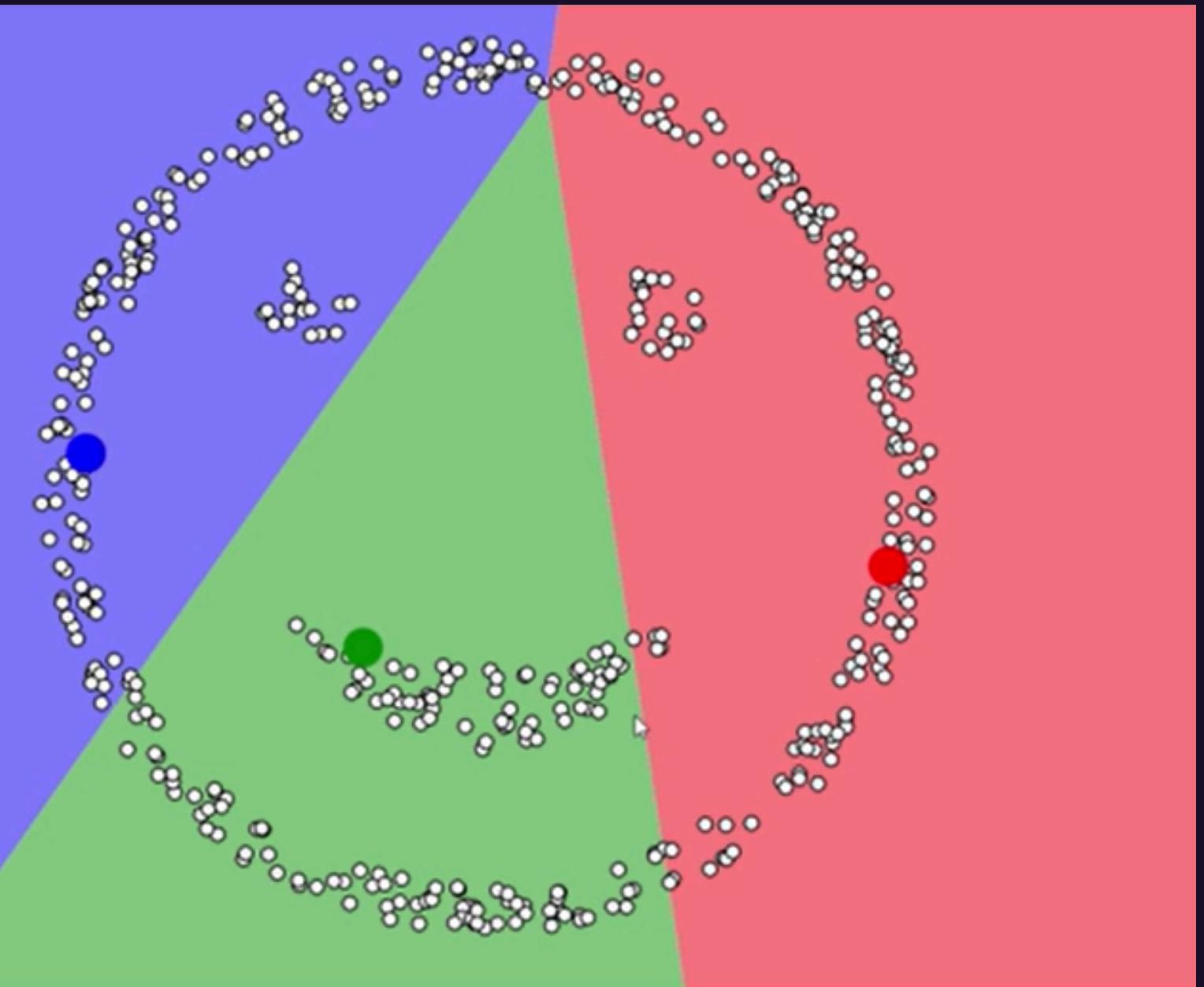
- O que ele mede? A "Soma dos Quadrados Dentro dos Clusters" (em inglês, Within-cluster Sum of Squares - WSS).
- Como funciona a análise? O algoritmo roda o K-Means várias vezes, testando diferentes números de clusters ($K=1, K=2, K=3, \dots, K=10$, etc.). Para cada valor de K , ele calcula o WSS total.
- Como interpretar o gráfico?
 1. A curva sempre desce: É normal que o WSS diminua à medida que você aumenta o número de clusters. Se cada ponto fosse seu próprio cluster, o WSS seria zero.
 2. Procure o "Cotovelo": O número ideal de clusters é o ponto onde a curva começa a achatar, formando um "cotovelo". Esse é o ponto em que adicionar mais um cluster depois do cotovelo não melhora significativamente a compactação dos grupos.



K-Means: A Dança dos Centros

O K-Means é o algoritmo de clusterização mais popular. Ele funciona em passos simples e repetitivos:

- Escolha "K": Primeiro, nós decidimos quantos clusters (grupos) queremos encontrar (por exemplo, K=3).
- Passo Inicial: O algoritmo sorteia K pontos aleatórios no gráfico, chamados "centroides".
- Atribuição: Cada amostra (ponto) é atribuída ao centroide mais próximo. Isso forma os clusters iniciais.
- Atualização: A posição de cada centroide é recalculada para ser o centro exato do seu novo grupo.
- Repetição: Os passos 3 e 4 são repetidos até que os centroides não mudem mais de lugar. Os grupos estão estáveis!



PCA + K-Means: A Dupla Perfeita para Exploração

- **Primeiro, a PCA:** Nós aplicamos a PCA para reduzir a complexidade e o "ruído" dos nossos dados. Isso nos permite visualizar as amostras em um gráfico 2D (PC1 vs PC2) e ter uma primeira intuição se existem grupos.
- **Depois, o K-Means:** Aplicamos o K-Means nos dados (geralmente nos dados normalizados originais) para que o algoritmo identifique matematicamente os clusters.
- **A Grande Finalização:** Nós plotamos o resultado do K-Means (colorindo os pontos por cluster) no gráfico da PCA. Se os grupos coloridos estiverem bem separados no gráfico, isso nos dá grande confiança de que os clusters encontrados são reais e significativos.

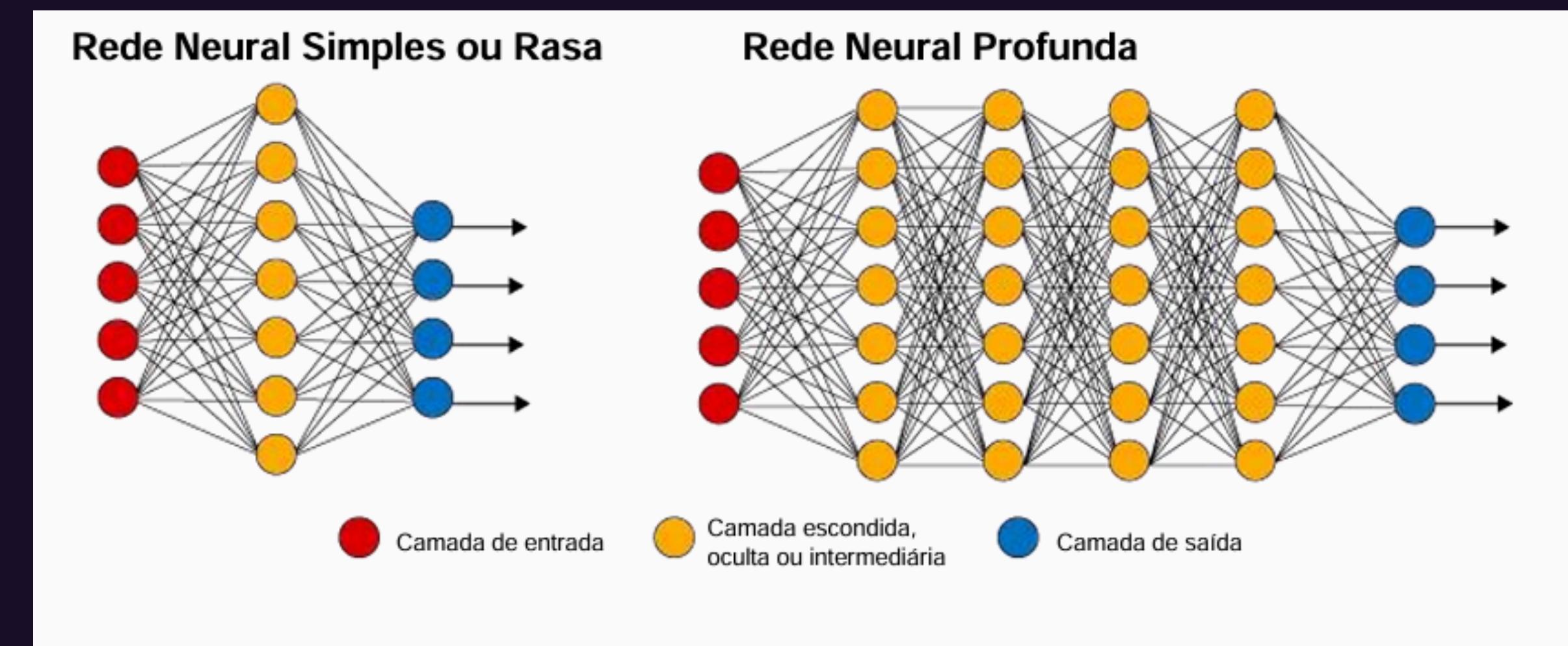


DEEP LEARNING

DeepLearning (também conhecida como aprendizado estruturado profundo, aprendizado hierárquico ou aprendizado de máquina profundo) é um ramo de aprendizado de máquina (Machine Learning) baseado em um conjunto de algoritmos que tentam modelar abstrações de alto nível de dados usando Redes Neurais Artificiais Profundas, ou seja, com várias camadas de processamento e compostas de diversas transformações lineares e não lineares.

Técnicas para ensinar uma
máquina a tomar decisões.

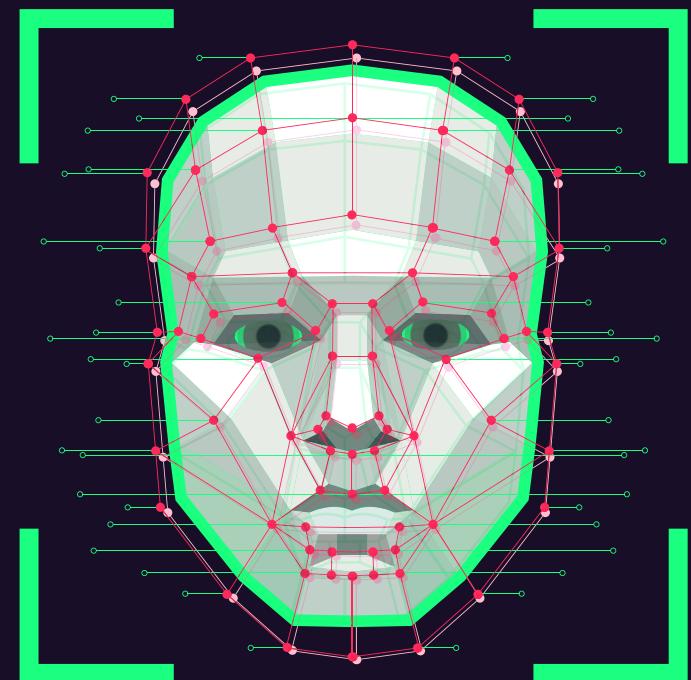
Redes Neurais Convolucionais
Redes adversariais generativa





DEEP LEARNING

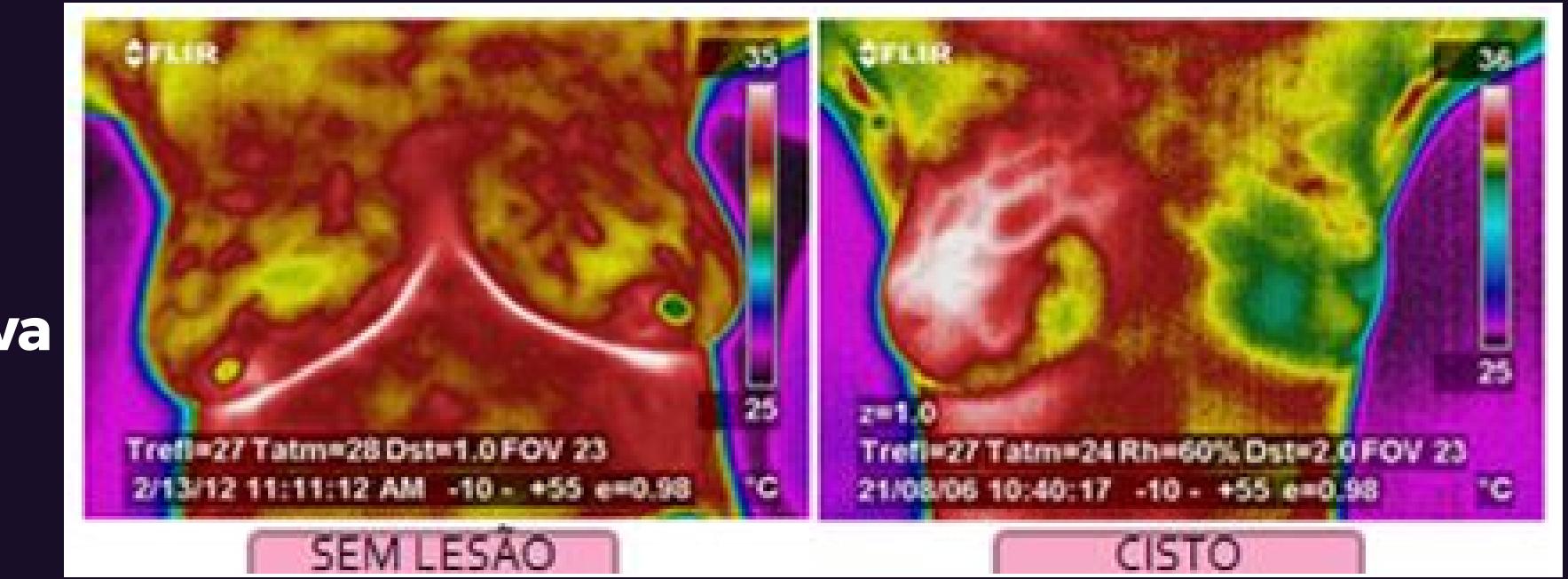
Transferência de Estilo Neural (Neural Style Transfer)





APLICAÇÕES

Diagnóstico e Predição
Monitorização e Gestão de Pacientes
Saúde Populacional e Medicina Preventiva
Descoberta de Medicamentos:





Introdução ao R

Texto, Números e Lógic

A Linguagem R: Vamos Falar com a Máquina

- **O que é um Objeto?** Pense em um objeto como uma caixa na memória do computador. Nós damos um nome (uma etiqueta) para essa caixa e guardamos uma informação (um valor) dentro dela.
- **O que é uma Variável?** É o tipo de informação que guardamos na caixa. O R precisa saber se está lidando com texto, números ou lógica.

Criamos um objeto 'a' para guardar o texto "maçã"

```
a <- "maçã"
```

Criamos um objeto 'b' para guardar o número 2

```
b <- 2
```

character (Texto):

- Sempre dentro de aspas (" " ou ' ').
- Ex: x <- "banana"

numeric (Número):

- Números inteiros ou com casas decimais.
- Ex: y <- 256 ou y <- 4.25

logical (Lógico):

- Apenas dois valores possíveis: TRUE ou FALSE.
- Pode ser abreviado para T ou F.
- Ex: z <- TRUE



Como Guardar Múltiplas Informações?

Às vezes, uma única caixa não é suficiente. Precisamos de "estantes" para organizar nossos objetos. O R tem várias estruturas para isso.

- Hoje vamos ver as mais importantes: vetor, lista, matriz e o famoso data frame.

A Estrutura Mais Fundamental: O Vetor

- Um vetor é um conjunto de variáveis do mesmo tipo.
- É criado com a função `c()` (que significa "combinar" ou "concatenar").

```
# Um vetor de texto (character)
```

```
rhcp_nomes <- c("Antony", "Flea", "John", "Chad")
```

```
# Um vetor de números (numeric)
```

```
rhcp_idade <- c(61, 61, 54, 62) # Idade do Chad corrigida ;)
```

```
# Um vetor de lógicos (logical)
```

```
rhcp_teste <- c(TRUE, TRUE, TRUE, FALSE)
```



Lista, Matriz e Data Frame

Organizando Dados de Formas Diferentes

- **Lista (list):**
 - A "estante" mais flexível. Pode guardar vetores de tipos diferentes.
 - `rhcp_info <- list(nomes = rhcp_nomes, idade = rhcp_idade)`
- **Matriz (matrix):**
 - Uma tabela 2D (linhas e colunas), mas todos os elementos devem ser do mesmo tipo.
Se misturar texto e número, tudo vira texto!
 - `rhcp_matrix <- matrix(..., nrow = 4, ncol = 3)`
- **Data Frame (data.frame):**
 - A estrutura mais importante para análise de dados!
 - É uma tabela 2D, como uma planilha, onde cada coluna pode ter um tipo diferente.
 - `rhcp_df <- data.frame(nomes = rhcp_nomes, idade = rhcp_idade)`



Acessando Dados: Onde Está a Informação?

As "Coordenadas" dos Seus Dados

- Para pegar uma informação específica, usamos operadores de indexação.

1. **\$ (Cifrão)**: A forma mais comum de acessar uma coluna em um `data.frame`.

- `rhcp_df$nomes` # Pega a coluna "nomes" inteira.

•

2. **[] (Colchetes Simples)**: Usado para "fatiar" objetos. Retorna um subconjunto, mantendo a estrutura original (ex: um sub-dataframe).

- `rhcp_idade[2]` # Pega o segundo elemento do vetor.
 - `rhcp_df[2, 3]` # Pega o elemento na linha 2, coluna 3.
 - `rhcp_df[1:3,]` # Pega as linhas de 1 a 3 (e todas as colunas).

•

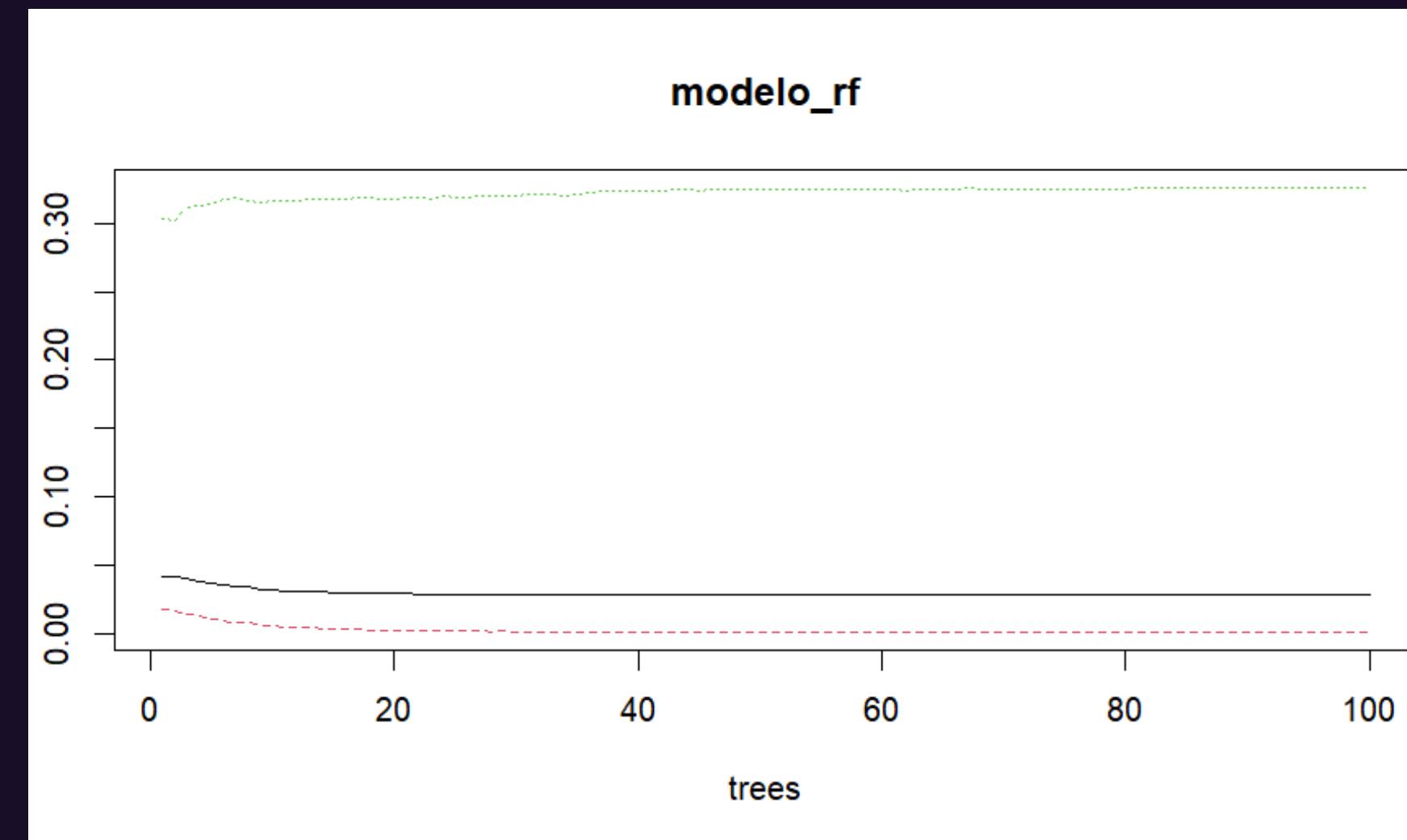
3. **[][] (Colchetes Duplos)**: Usado para extrair um único elemento de uma `list`.

- `rhcp_info[["idade"]]` # Extrai o vetor de idades de dentro da lista.

ANÁLISE DO ERRO

- **Linha Preta (OOB):** Ela representa a taxa de erro geral do modelo. É a média ponderada dos erros das outras duas linhas. Você pode ver que o erro começa um pouco mais alto e rapidamente se estabiliza em um valor baixo, em torno de 2.8% (como vimos no `print(modelo_rf)`).
- **Linha Verde (Erro para a classe "1" - Diabéticos):** Esta linha mostra a taxa de erro do modelo especificamente ao tentar prever os casos positivos (pacientes com diabetes). Esta linha está muito alta, estabilizando em torno de 0.32 (ou 32%).
- **Linha Vermelha (Erro para a classe "0" - Não Diabéticos):** Esta linha mostra a taxa de erro ao prever os casos negativos (pacientes sem diabetes). Ela é extremamente baixa, quase zero (estabilizando em torno de 0.06%). Isso mostra que o modelo é excelente para identificar quem não tem a doença.

Pacote	Função Principal (Analogia)	Etapa no Projeto
<code>tidyverse</code>	A Oficina Completa	Limpeza, Transformação e Visualização
<code>rsample</code>	A Faca de Corte Preciso	Divisão dos Dados (Treino/Teste)
<code>randomForest</code>	O Construtor Especializado	Treinamento do Modelo
<code>caret</code>	O Inspetor de Qualidade	Avaliação da Performance do Modelo



- Acurácia: 97.22%. No geral, o modelo acertou 97.22% de todas as previsões.
- Diagnóstico: Parece um número fantástico, mas é enganoso. A acurácia é alta porque o seu dataset é desbalanceado (tem muito mais "0"s do que "1"s), então o modelo acerta muito só por prever o resultado mais comum ("Não Diabético").
- Sensitivity (Sensibilidade ou Recall): 67.064%. De todas as pessoas que realmente tinham diabetes no conjunto de teste ($1124 + 552 = 1676$ pessoas), o seu modelo conseguiu identificar corretamente apenas 67.1%. Ele está deixando passar quase um terço (32.9%) dos pacientes diabéticos, classificando-os como saudáveis (os 552 Falsos Negativos). Para um modelo de diagnóstico médico, este é o ponto mais importante a ser melhorado.
- Especificidade: 99.984%. O que significa: De todas as pessoas que não tinham diabetes, o modelo identificou corretamente 99.98%, ou seja, o modelo é praticamente perfeito para liberar quem é saudável. Ele raramente comete o erro de diagnosticar um paciente saudável como doente.

- Pos Pred Value (Precisão ou VPP): 99.734%. O que significa: Das vezes que o seu modelo disse que uma pessoa tem diabetes ($1124 + 3 = 1127$ vezes), ele estava correto em 99.7% das vezes. O modelo é extremamente confiável quando dá um diagnóstico positivo. Se ele diz "diabético", a chance de ser verdade é altíssima.
- Balanced Accuracy (Acurácia Balanceada): 83.524%
- O que significa: É a média entre a Sensibilidade e a Especificidade. É uma métrica muito mais justa que a acurácia para datasets desbalanceados. Um valor de 83.5% mostra que o modelo é bom, mas a baixa sensibilidade (67.1%) está puxando a média para baixo

1. O que é a no Information Rate (NIR): 0.9162 (ou 91.62%) ?

- No seu caso: A classe mais comum no seu conjunto de teste é "0" (Não Diabético), que representa 91.62% dos dados. Portanto, um modelo que previsse "0" para todo mundo teria uma acurácia de 91.62%.
- Para que serve? O NIR é a sua linha de base. É o mínimo que o seu modelo precisa superar para ser considerado minimamente útil. Se a acurácia do seu modelo fosse menor ou igual a 91.62%, significaria que ele não aprendeu nada útil e seria melhor simplesmente chutar o resultado mais provável.

- Existe diferença estatística entre o meu modelo e entre o acaso??
 - P-Value [Acc > NIR] : < 2.2e-16
-
- Interpretação: O p-valor (P-Value) é a probabilidade de observar uma acurácia tão alta quanto a sua se, na verdade, seu modelo não fosse melhor que o chute.
 - Seu resultado: Como nosso P foi muito menor que o limiar padrão de 0.05, nós rejeitamos a hipótese de que a melhora foi por acaso (HIPÓTESE NULA)

A LUZ NO FIM DO TÚNEO

- Em conclusão nosso modelo é estatisticamente melhor do que um simples chute. Ele aprendeu padrões úteis nos dados.

Kappa (Cohen's Kappa): 0.7877

Essa métrica mede o quanto o modelo concorda com a realidade, mas depois subtrai a concordância que já era esperada por acaso (devido ao desbalanceamento das classes).

- Escala de Interpretação (aproximada):
- < 0.20: Pobre
- 0.21 - 0.40: Razoável
- 0.41 - 0.60: Moderada
- 0.61 - 0.80: Substancial / Boa
- 0.81 - 1.00: Quase Perfeita



GitHub:

<https://github.com/Bruno-4lmeida/SBM/>