

# Inteligência Artificial (IA)

---

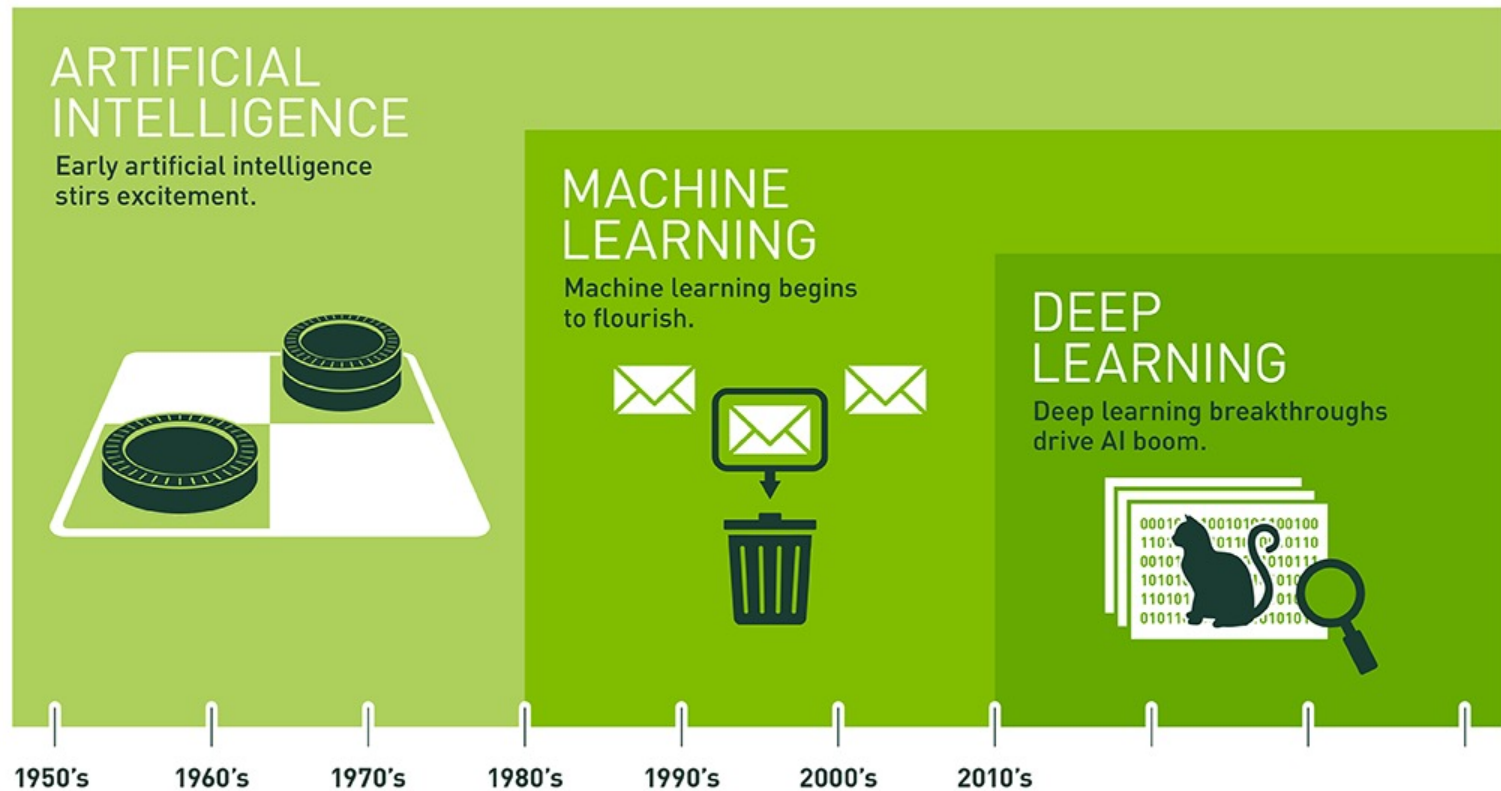
## Aprendizagem de Máquina

Prof. Julio Cesar Soares dos Reis

# Sumário

- Conceitos básicos
- Tipos de aprendizagem
- Abordagens tradicionais

# Introdução



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Fonte: <https://developer.nvidia.com/deep-learning>

# Inteligência Artificial

- ... **inteligência** similar à humana exibida por mecanismos ou software ...
- formas de criar máquinas e softwares que pareçam ter **inteligência** humana

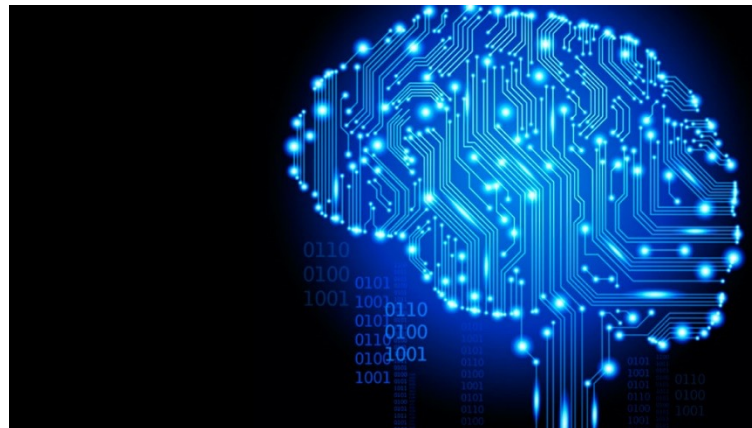


# Aprendizagem de Máquina

- É um subcampo da inteligência artificial dedicado à investigação dos sistemas computacionais capazes de **aprender** e melhorar com a experiência, sem ter sido explicitamente programado para tal fim.
- Um algoritmo parametrizado, capaz de encontrar o conjunto de parâmetros que melhor se aproxima do comportamento desejado.

# Aprendizado Profundo

- “Sub área” da aprendizagem de máquina onde os modelos são inspirados em como o cérebro humano trabalha, expressando este trabalho matematicamente. Parâmetros que definem os modelos matemáticos são **aprendidos** automaticamente através de dados.



# Aprendizagem de Máquina

- Os seguintes passos definem o que é aprendizagem?
  - Memorização de algo
  - Observação e exploração
  - Desenvolvimento de habilidades motoras e cognitivas por meio da prática
  - Organização de novos conhecimentos em representações gerais e efetivas

# O que é Aprendizagem?

*"Aprendizagem denota mudanças em um sistema que são adaptativas no sentido de que elas capacitam o sistema a executar sua(s) tarefa(s) mais eficiente e efetivamente da próxima vez, sobre o mesmo conjunto de exemplos disponíveis."*

*Herbert Simon*



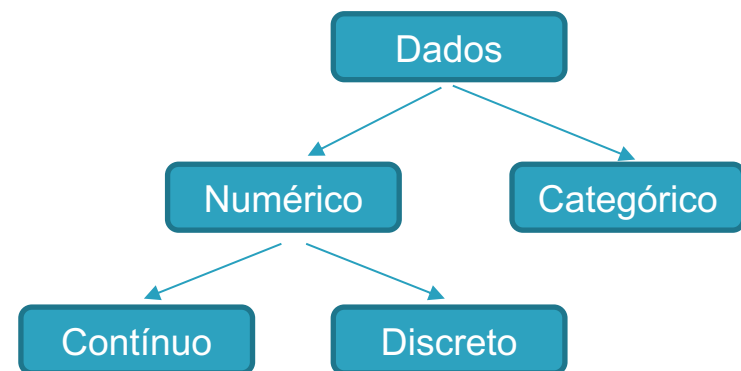
# O que é Aprendizagem?

*“A computer program is said to learn from **experience E** with respect to some **task T** and some performance **measure P**, if its performance on T, as measured by P, improves with experience E.”*

*Tom Mitchell, Carnegie Mellon University*

# Terminologia Geral

HR Information		Contact	
Position	Salary	Office	Extn.
Accountant	\$162,700	Tokyo	5407
Chief Executive Officer (CEO)	\$1,200,000	London	5797
Junior Technical Author	\$86,000	San Francisco	1562
Software Engineer	\$132,000	London	2558
Software Engineer	\$206,850	San Francisco	1314
Integration Specialist	\$372,000	New York	4804
Software Engineer	\$163,500	London	6222
Pre-Sales Support	\$106,450	New York	8330
Sales Assistant	\$145,600	New York	3990
Senior Javascript Developer	\$433,060	Edinburgh	6224



- **Variáveis independentes**, "X", preditoras, "Causa"
- **Variáveis dependentes**, "Y", resposta, efeito

# Tipos de Variáveis

- **Categórica:** contêm um número finito de categorias ou grupos distintos. Os dados categóricos podem não ter uma ordem lógica (exemplo: método de pagamento).
- **Discreta:** variáveis numéricas que têm um número contável de valores entre quaisquer dois valores (exemplo o número de reclamações de clientes)
- **Contínua:** variáveis numéricas que têm um número infinito de valores entre dois valores quaisquer. Uma variável contínua pode ser numérica ou de data/hora. (exemplo: o comprimento de uma peça ou a data e hora em que um pagamento é recebido).

# Terminologia Geral

← Features →					Label
Position	Experience	Skill	Country	City	Salary (\$)
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

- Cada linha indica uma **observação**
- Cada coluna de atributo (**features**), indica uma característica ou dimensão da observação. **Variável independente.**
- Normalmente existe uma coluna feature que nós chamamos coluna resultado, rótulo (**label**) ou alvo. **Variável Dependente.**

# Tipos de Aprendizagem de Máquina

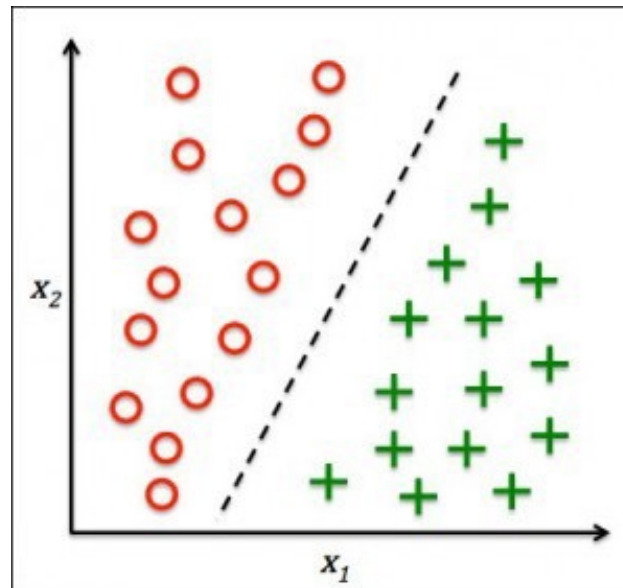


# Aprendizagem Supervisionada

- Dado um conjunto de exemplos que são pares [entrada, saída], um algoritmo deve encontrar (ou aprender) uma *regra* que faça um bom trabalho na predição de qual será a saída para uma nova entrada (ainda não-vista)

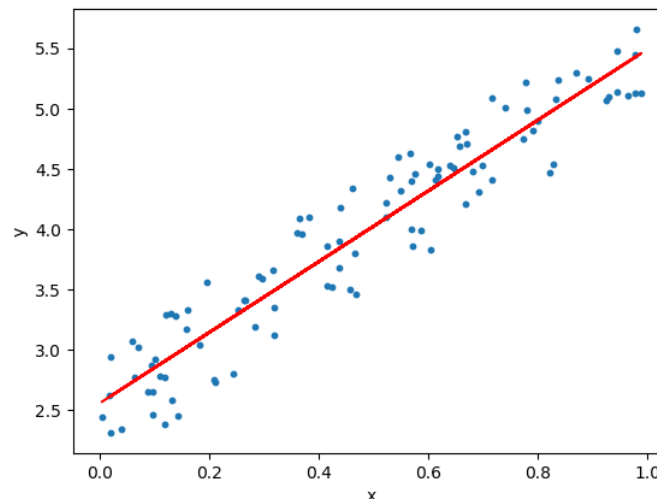
# Aprendizagem Supervisionada

- Classificação:
  - Determinar a que classe (categórica) pertence uma dada observação. Binária ou multinomial.



# Aprendizagem Supervisionada

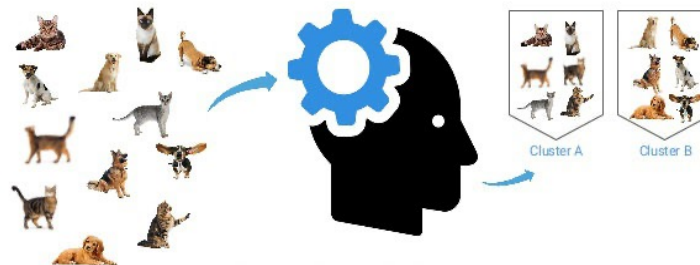
- Regressão:
  - Modelar o relacionamento entre as variáveis independentes e dependentes. A saída, neste caso, é contínua (ou seja, o resultado é um espaço contínuo).





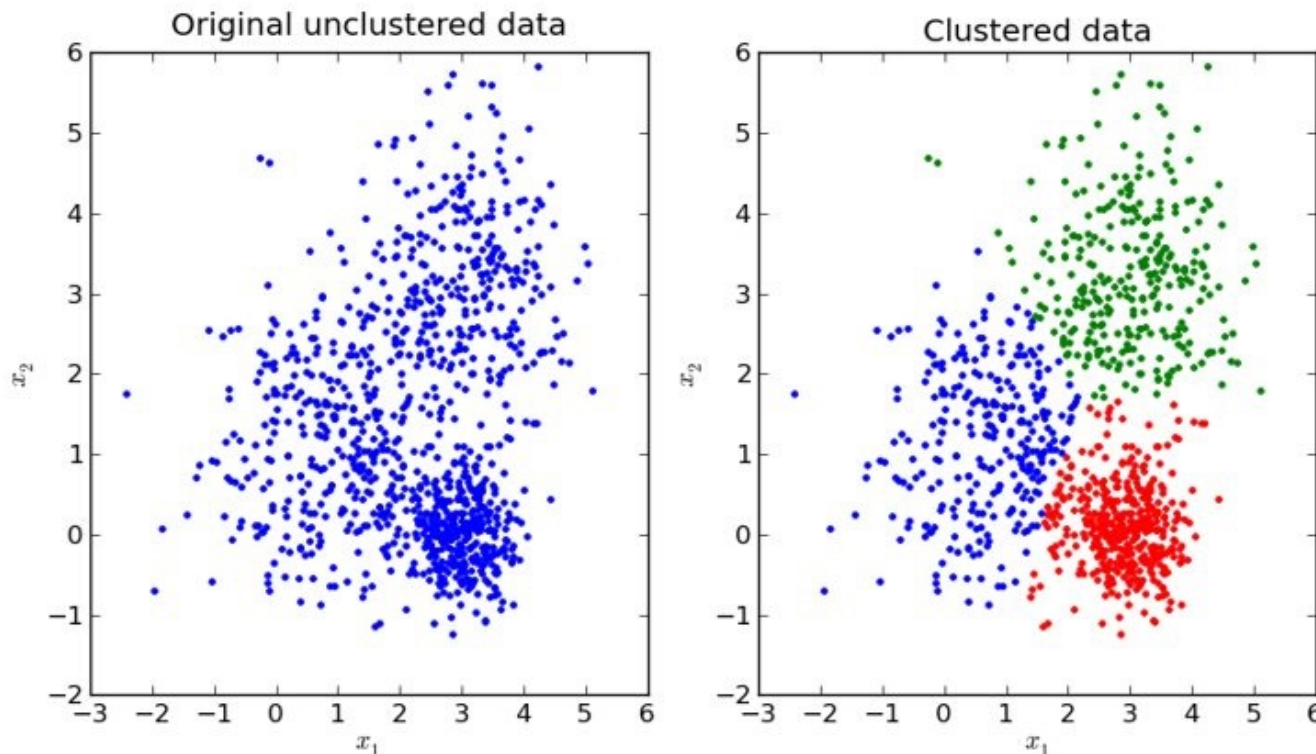
# Aprendizagem Não-supervisionada

- Dado um conjunto de exemplos (apenas a entrada), um algoritmo aprende conceitos apenas extraindo e relacionando características dos exemplos. Objetivo aqui é extrair padrões e correlações significativas dos dados sem a referência de um resultado conhecido (rótulo). O contrário do aprendizado supervisionado.



# Aprendizagem Não-supervisionada

- **Clustering (“agrupamento”)**: Dado um conjunto de exemplos (apenas a entrada), um agente deve agrupá-los de maneira “natural”

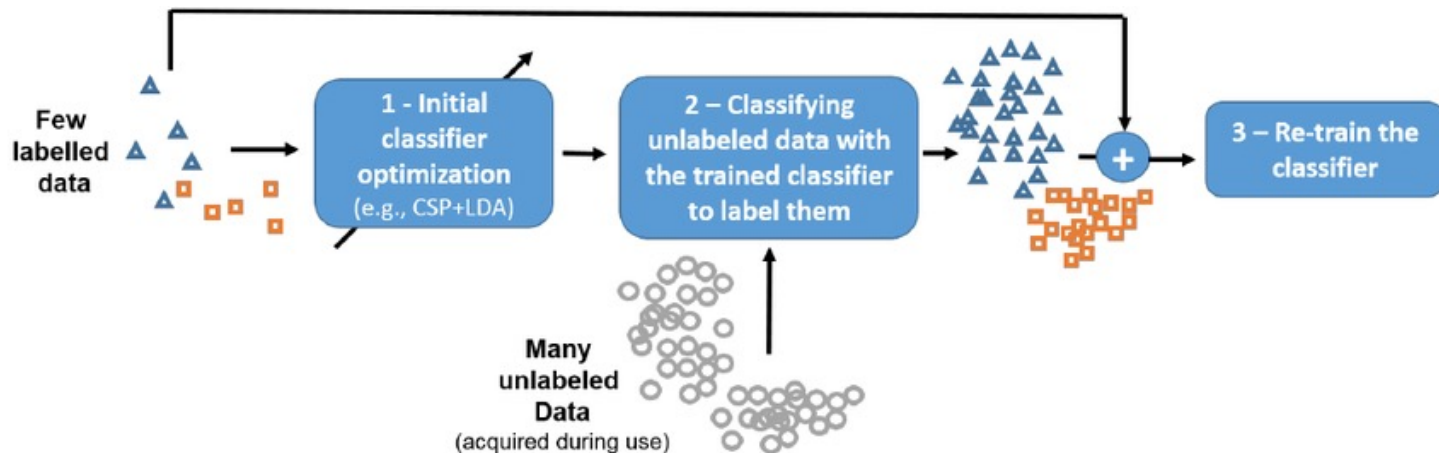


# Aprendizagem Não-supervisionada

- **Regras de associação:** descoberta de relações interessantes entre variáveis (padrões e relacionamentos).
- Exemplo: {leite, pão}  $\rightarrow$  {manteiga}, que indica que se o cliente compra leite e pão, com um determinado grau de certeza, ele também compra manteiga.

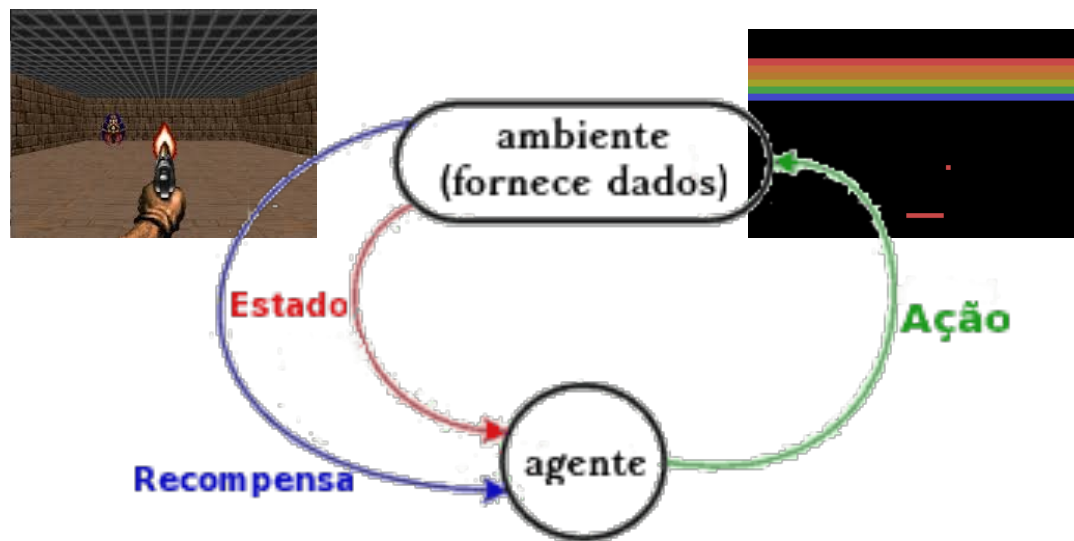
# Aprendizagem Semi-supervisionada

- Técnicas que fazem uso de dados anotados, e não anotados de forma simultânea no seu processo de aprendizagem.
- Exemplo: *hashtags* no Twitter?



# Aprendizagem por Reforço

- Agente cujo processo de aprendizagem se baseia na maximização da recompensa por uma ação. A máquina tenta aprender qual é a melhor ação a ser tomada, dependendo das circunstâncias na qual essa ação será executada.



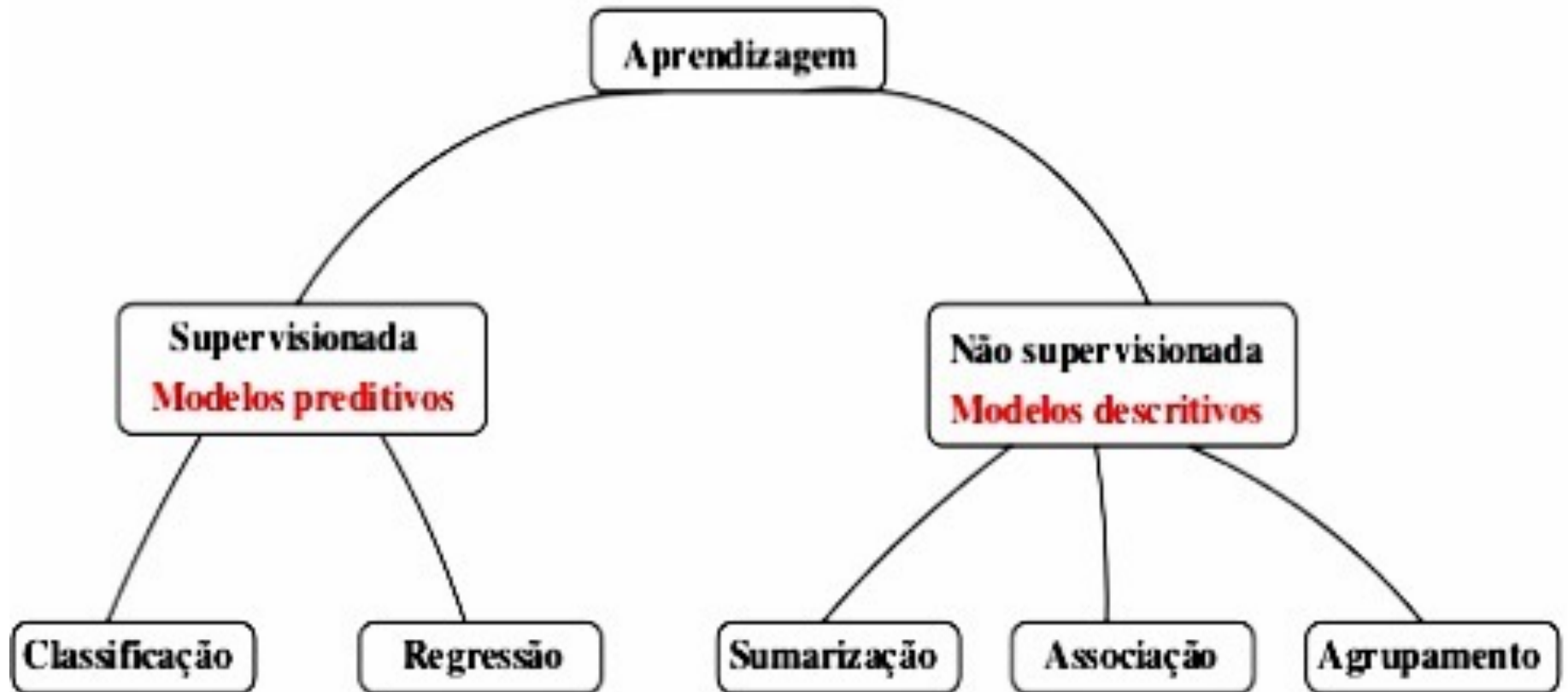
# Aprendizagem por Reforço

- Um exemplo de tarefa de reforço é a de ensinar um robô a encontrar a melhor trajetória entre dois pontos. Algoritmos de aprendizado utilizados nessa tarefa, em geral, punem a passagem por trechos pouco promissores e recompensam a passagem por trechos promissores.

# Tipos de Aprendizagem de Máquina



# Hierarquia do Aprendizado





# Aprendizagem Supervisionada (Regressão)

*O pedaço de pão número 1 foi nutritivo quando o comi.  
O pedaço de pão número 2 foi nutritivo quando o comi.  
O pedaço de pão número 3 foi nutritivo quando o comi.  
...  
O pedaço de pão número 100 foi nutritivo quando o comi.  
Então, todos os pedaços de pão serão nutritivos se eu os comer!!!*



David Hume



Bertrand Russell

*Tem sido argumentado que nós temos razão em saber que o futuro se parecerá com o passado porque o que era futuro tem, constantemente, se tornado passado e tem sempre sido notado que se parece com o passado, de modo que nós, na realidade, temos experiência do futuro, os quais podemos chamar de futuros já passados. Mas esse tipo de argumento empobrece a própria questão... Nós temos experiência dos futuros já passados, mas não dos futuros futuros, e a questão é: os futuros futuros se parecerão com os futuros já passados?*

# Aprendizagem Supervisionada (Regressão)

- Considere a seguinte tabela:

Instância	x	f(x)
1	0	1
2	1	2
3	2	5
4	3	10
5	4	17
6	5	26
7	6	37
8	7	50
9	8	65
10	9	82

Qual a forma analítica (“matemática”) de  $f(x)$ ?

# Aprendizagem Supervisionada (Regressão)

- Considere a seguinte tabela:

Instância	$x$	$f(x)$
1	0	1
2	1	2



E se houver poucos dados?

# Aprendizagem Supervisionada (Regressão)

- Há garantia de que a função é esta?

Instância	x	f(x)
1	0	1
2	1	2
3	2	5
4	3	10
5	4	17
6	5	26
7	6	37
8	7	50
9	8	65
10	9	82

# Aprendizagem Supervisionada (Regressão)

- E se alguém coletar os dados destacados abaixo?

Instância	x	f(x)
1	0	1
2	1	2
3	2	5
4	3	10
5	4	17
6	5	26
7	6	37
8	7	50
9	8	65
10	9	82
11	10	102

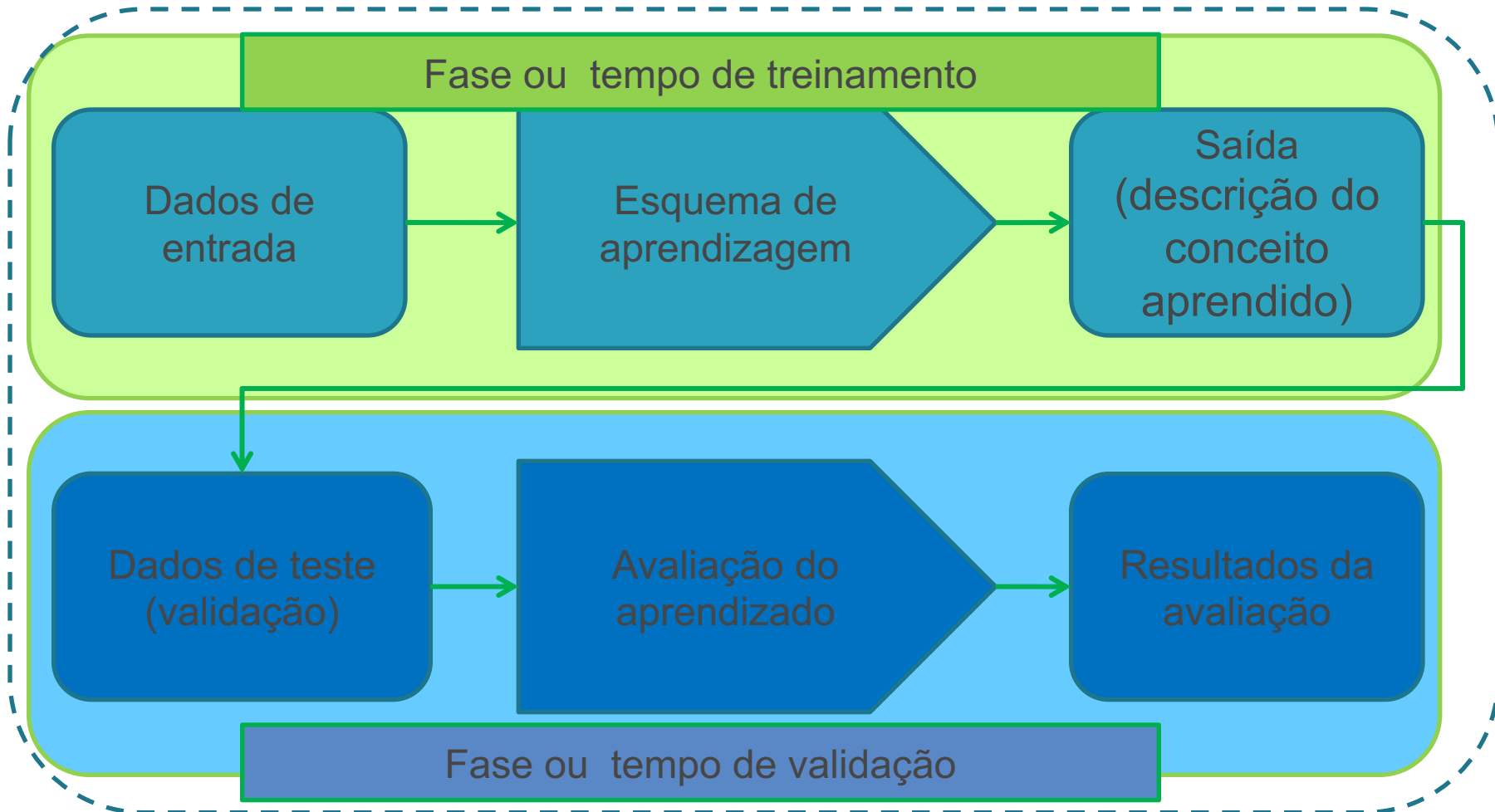
# Conceitos

- Em aprendizagem supervisionada, **o objetivo é encontrar uma *função* (ou regra ou mapeamento) que mapeie as entradas nas saídas**
- Um algoritmo deve aprender a função desconhecida com base em alguns exemplos de entradas com os valores das saídas correspondentes (isto é, os valores da função desconhecida)

# Conceitos

- **Exemplo ou Instância (*Instance*)**
  - Formalmente, um exemplo é um par  $[x, f(x)]$ , no qual  $x$  é a entrada e  $f(x)$  é a saída da função **desconhecida** aplicada a  $x$
- Um **atributo** é uma característica. Pode ter um valor *contínuo* ou *discreto* ou um símbolo com valor **qualitativo (vs. quantitativo)**
- Um exemplo é dito ser composto pelos valores de vários atributos
- Um exemplo pode ser chamado de um **vetor de características** ou ***feature vector***

# Processo Geral do Aprendizado Supervisionado





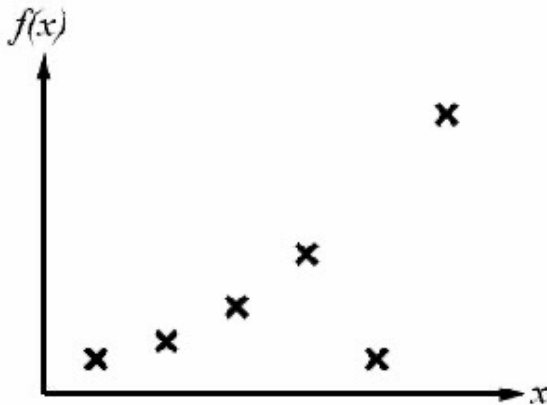
# Conceitos

- **Regressão (Indução):** Dada uma coleção de exemplos de  $f$ , indução é uma maneira de encontrar uma função  $h$  que seja uma “aproximação” de  $f$
- **Hipótese:** A função  $h$  é chamada de uma hipótese
- **Generalização:** É a capacidade de uma hipótese prever corretamente exemplos ainda não-vistos (durante a aprendizagem)
- **Treinamento:** É a fase da aprendizagem na qual um algoritmo é exposto a dados (exemplos) do conceito a ser aprendido. O resultado do treinamento é a hipótese
- **Validação:** É a fase da aprendizagem na qual a hipótese é testada para verificar sua capacidade de generalização

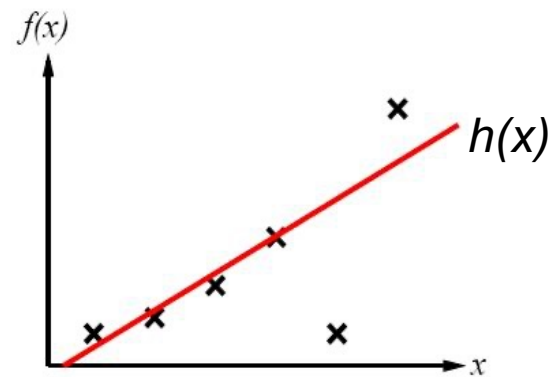
# Exemplo de Indução: Ajuste de uma Função aos Pontos de Dados

- Seja  $f(x)$  uma função de uma variável
- Os exemplos são pares  $[x, f(x)]$ , sendo  $x$  e  $f(x)$  números reais
- O conjunto (a coleção) de exemplos é chamado de **conjunto de treinamento**

Exemplos



Uma aproximação

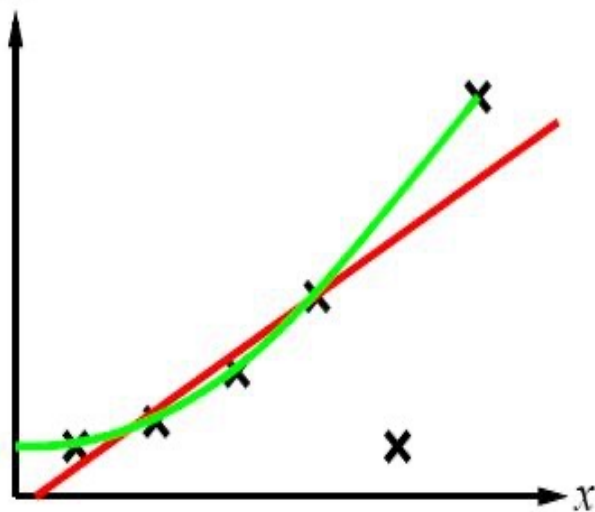


- A função hipótese  $h$  é dita ser **consistente** se ela concorda com  $f$  em **todos** os exemplos do conjunto de treinamento
- $h(x)$  **não** é consistente, neste exemplo

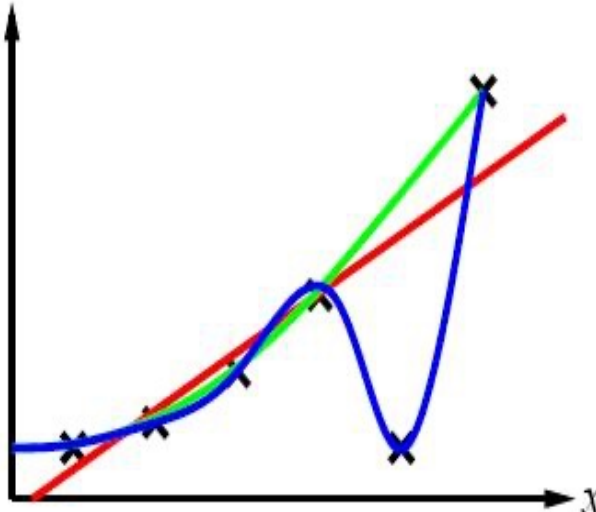
# Exemplo de Indução: Ajuste de uma Função aos Pontos de Dados

- Outras aproximações

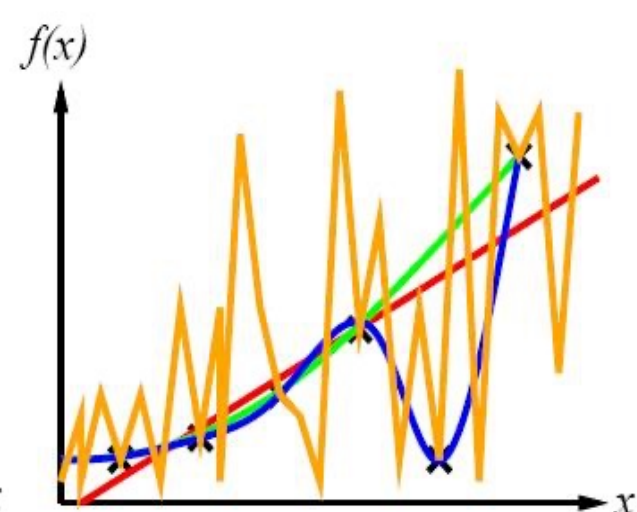
(1)



(2)



(3)



- Lâmina de Ockham (ou Princípio da Parcimônia): maximize a combinação de consistência e simplicidade.

# Supervisionado X Não Supervisionado

- Considere um conjunto de dados de entrada (exemplos ou instâncias), que padrões poderiam ser extraídos?

Table 1.3 Weather Data with Some Numeric Attributes				
Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

# Supervisionado X Não Supervisionado

- Regras de associação (associar é “correlacionar” atributos uns aos outros):

IF outlook == sunny AND play == no THEN humidity > 80

IF windy == no AND play == no THEN outlook = sunny AND humidity > 80

- Regras de classificação (classificar é atribuir uma classe para um dado exemplo):

IF outlook == sunny AND humidity > 83 THEN play = no

# Exemplos de Problemas

- Quais são exemplos de problemas supervisionados e não-supervisionados?
  - Um conjunto de fotos de 6 pessoas, mas sem informação sobre quem é em cada foto e precisamos dividir esse conjunto de dados em 6 pilhas, cada pilha representa a foto de 1 indivíduo
  - Um conjunto de fotos com informações sobre o que está nela e você treina um modelo para reconhecer novas fotos
  - Um conjunto de moléculas com informação sobre quais são drogas e você treina um modelo para responder se novas moléculas também são uma droga
  - Há um conjunto de moléculas, parte delas são drogas e outras não, mas você não sabe qual é qual e precisa de um algoritmo para descobrir as drogas

# Exemplos de Problemas

- Quais são exemplos de problemas supervisionados e não-supervisionados?
  - Um conjunto de fotos de 6 pessoas, mas sem informação sobre quem é em cada foto e precisamos dividir esse conjunto de dados em 6 pilhas, cada pilha representa a foto de 1 indivíduo (**não supervisionado**)
  - Um conjunto de fotos com informações sobre o que está nela e você treina um modelo para reconhecer novas fotos (**supervisionado**)
  - Um conjunto de moléculas com informação sobre quais são drogas e você treina um modelo para responder se novas moléculas também são uma droga (**supervisionado**)
  - Há um conjunto de moléculas, parte delas são drogas e outras não, mas você não sabe qual é qual e precisa de um algoritmo para descobrir as drogas (**não-supervisionado**)

# Exemplos de Problemas

- Quais são exemplos de problemas supervisionados e não-supervisionados?
  - Determinar se um email é ou não spam
  - Descobrir padrões nos dados, por exemplo, se estiver chovendo, as pessoas tendem a ficar dentro de casa
  - Dado um filme no Netflix, prever a nota que o usuário atribuirá a um determinado filme
  - Dada uma imagem, determinar quais objetos estão presentes nela (cachorro, gato, computador, prédios, etc)
  - Dada uma lista de clientes e informações sobre eles, identifique grupos de usuários similares
  - Dadas medidas de sensores em uma instalação de fabricação, identifique anomalias, ou seja, que algo está errado



# Exemplos de Problemas

- Quais são exemplos de problemas supervisionados e não-supervisionados?
  - Determinar se um email é ou não spam (**classificação**)
  - Descobrir padrões nos dados, por exemplo, se estiver chovendo, as pessoas tendem a ficar dentro de casa
  - Dado um filme no Netflix, prever a nota que o usuário atribuirá a um determinado filme (**regressão**)
  - Dada uma imagem, determinar quais objetos estão presentes nela (cachorro, gato, computador, prédios, etc) (**classificação**)
  - Dada uma lista de clientes e informações sobre eles, identifique grupos de usuários similares (**clusterização**)
  - Dadas medidas de sensores em uma instalação de fabricação, identifique anomalias, ou seja, que algo está errado (**clusterização**)

# Features

- Em uma tarefa convencional:



Determine o valor de  
uma casa.



Determine se uma casa  
é vendável ou não.

- Determinar as características que influenciam no fenômeno é uma tarefa necessária e factível

## Exemplo: Features em PLN

- Colete o sentimento do indivíduo na seguinte frase:

“A casa é razoável, com bons quartos e uma sala pequena”.

- Encontre todos os adjetivos e suas associações:

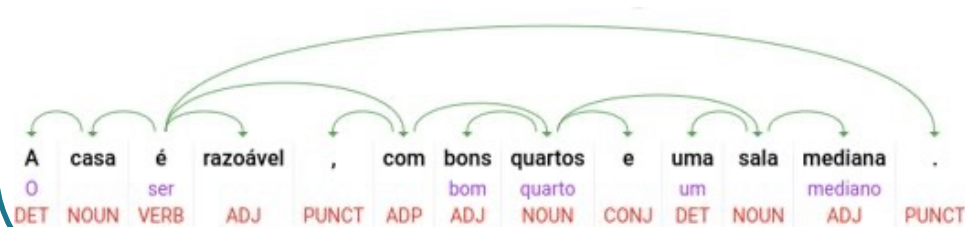
*Não existem features diretamente observáveis, apenas palavras.*

# Features em PLN

- Podem ser tratadas de diversas formas
  - Exemplo: “bag-of-words”

the dog is on the table

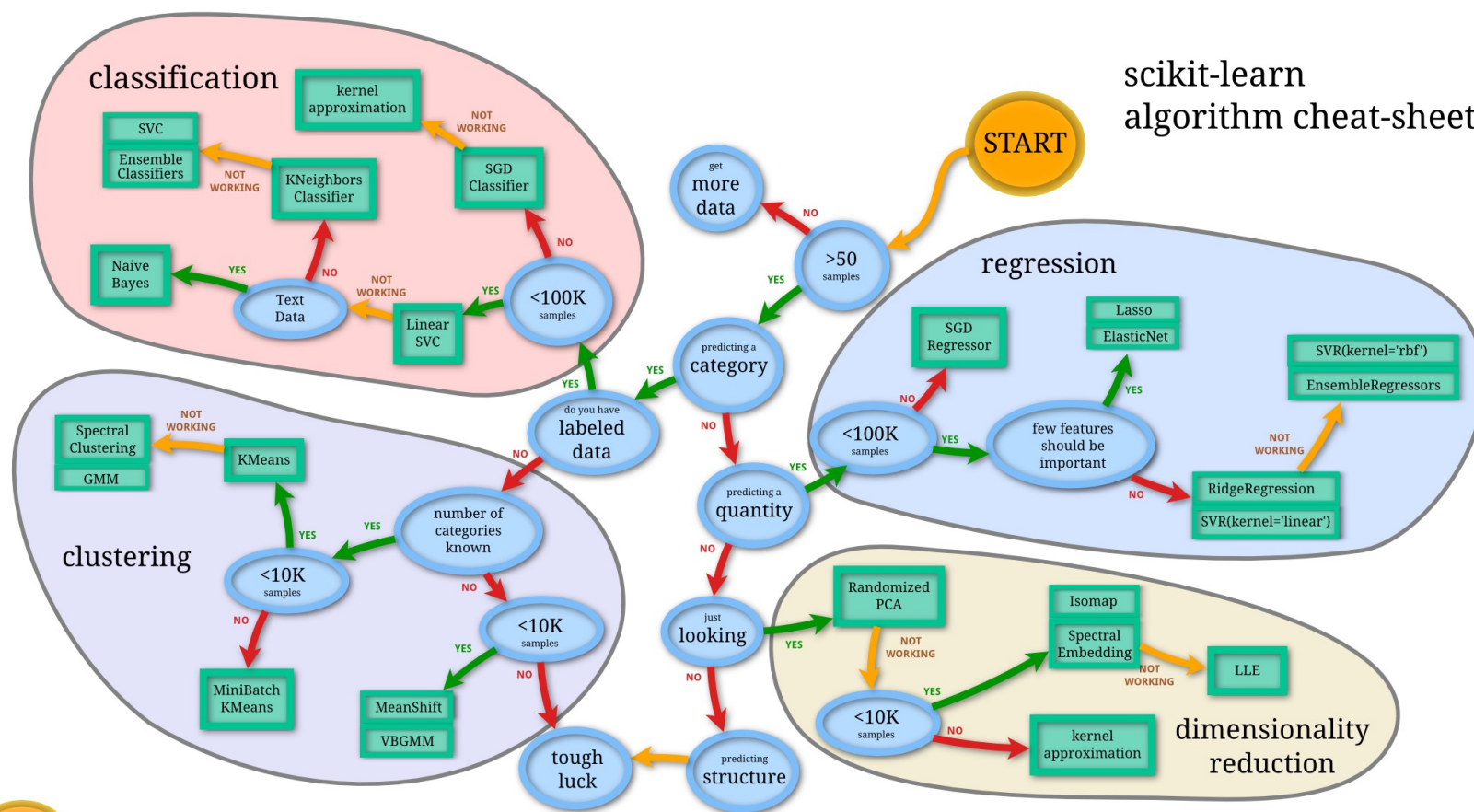
0	0	1	1	0	1	1	1
are	cat	dog	is	now	on	table	the



- Alta dimensionalidade
- Alto custo computacional (memória e processamento)
- Pouca expressividade (o trabalho fica por conta do classificador)
- Outras opções?

# Mapa de Métodos

scikit-learn  
algorithm cheat-sheet



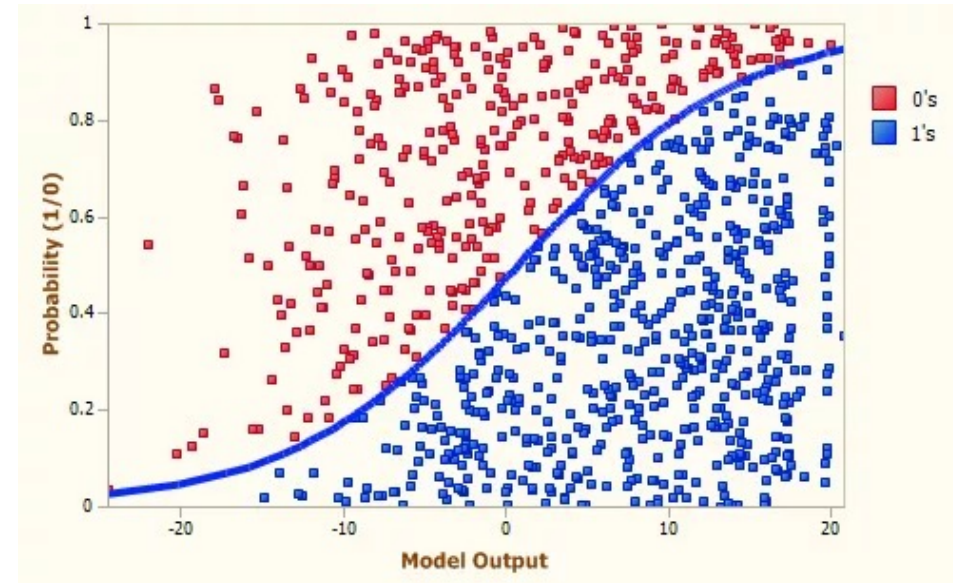
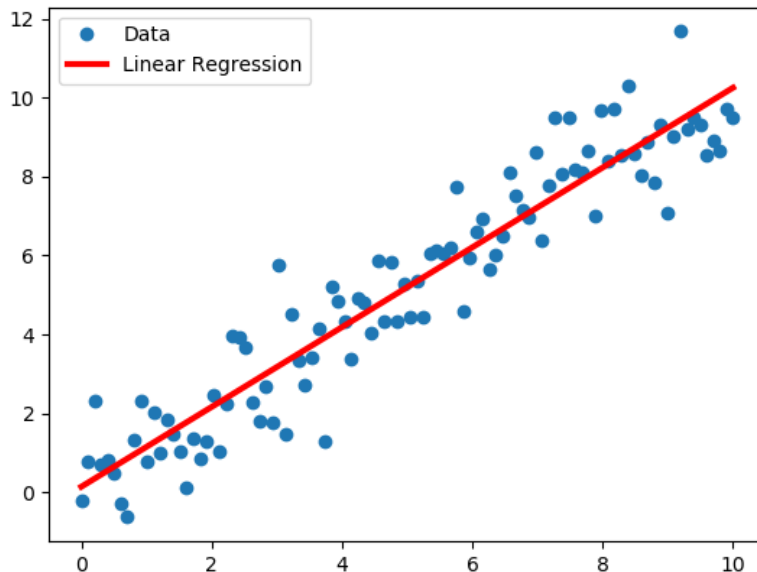
Fonte: [https://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/index.html](https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html)

# Algumas Abordagens Tradicionais

- Regressão linear e logística
- Árvores de decisão
- K-nearest neighbors
- Hidden Markov Model
- K-Means

# Regressão Linear / Logística

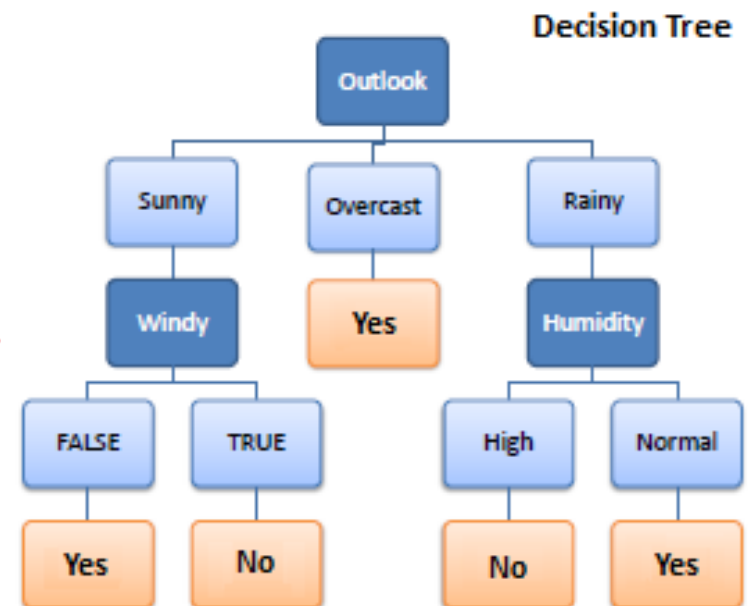
- Produzir, a partir de um conjunto de observações, um modelo que permita a predição de valores tomados por uma variável categórica.



# Árvores de Decisão (*Decision Trees*)

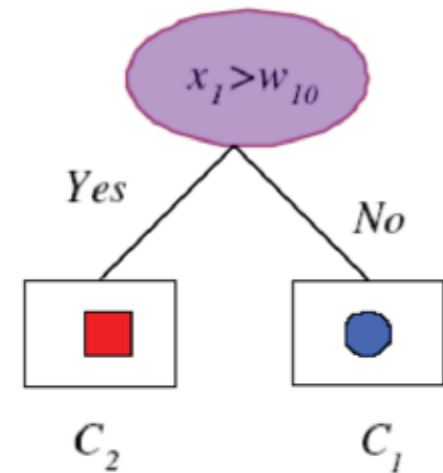
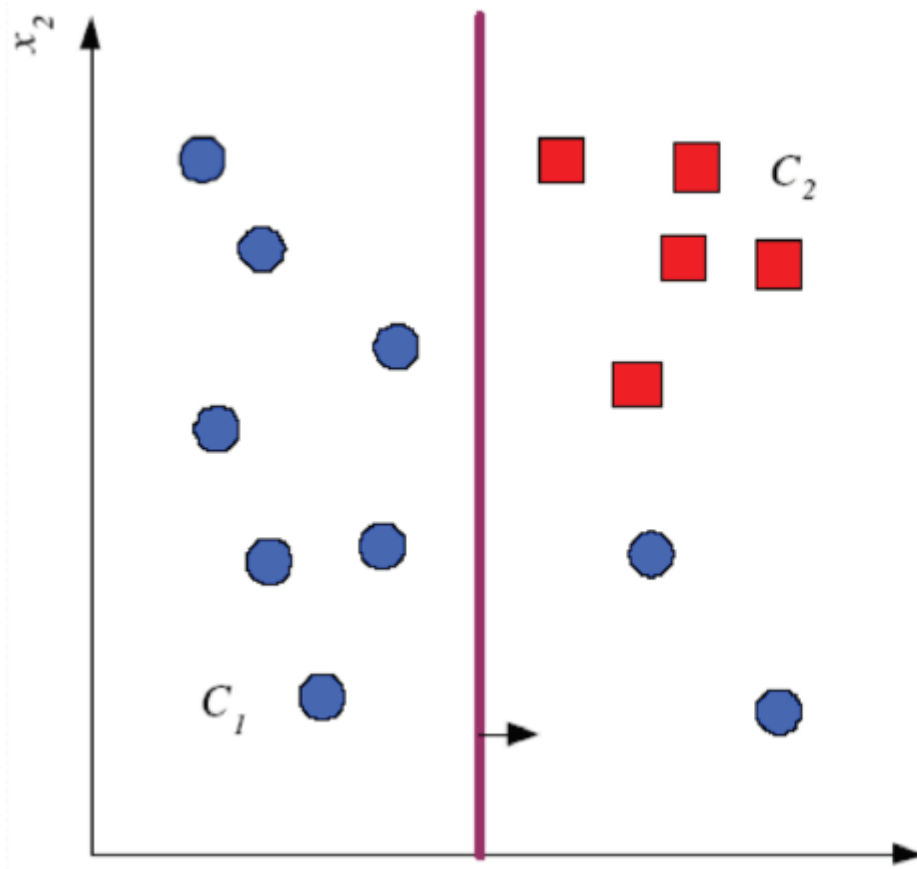
- Mapa de possíveis resultados de uma série de escolhas relacionadas...

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

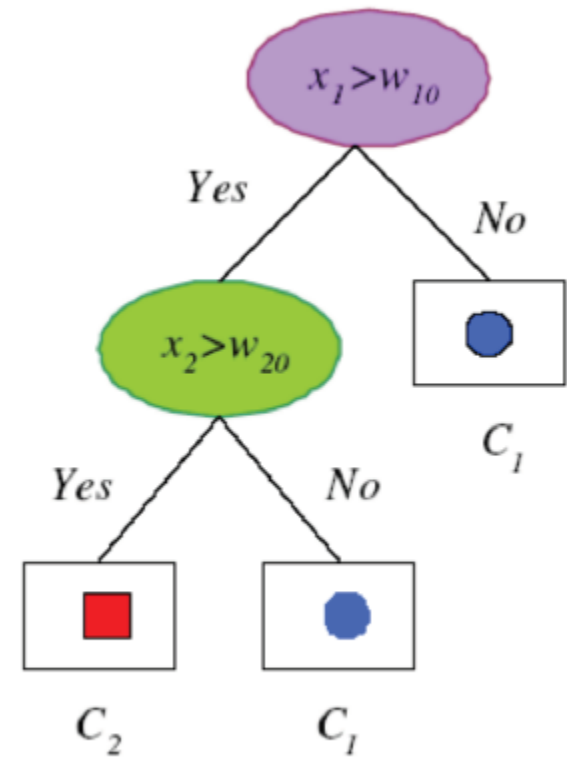
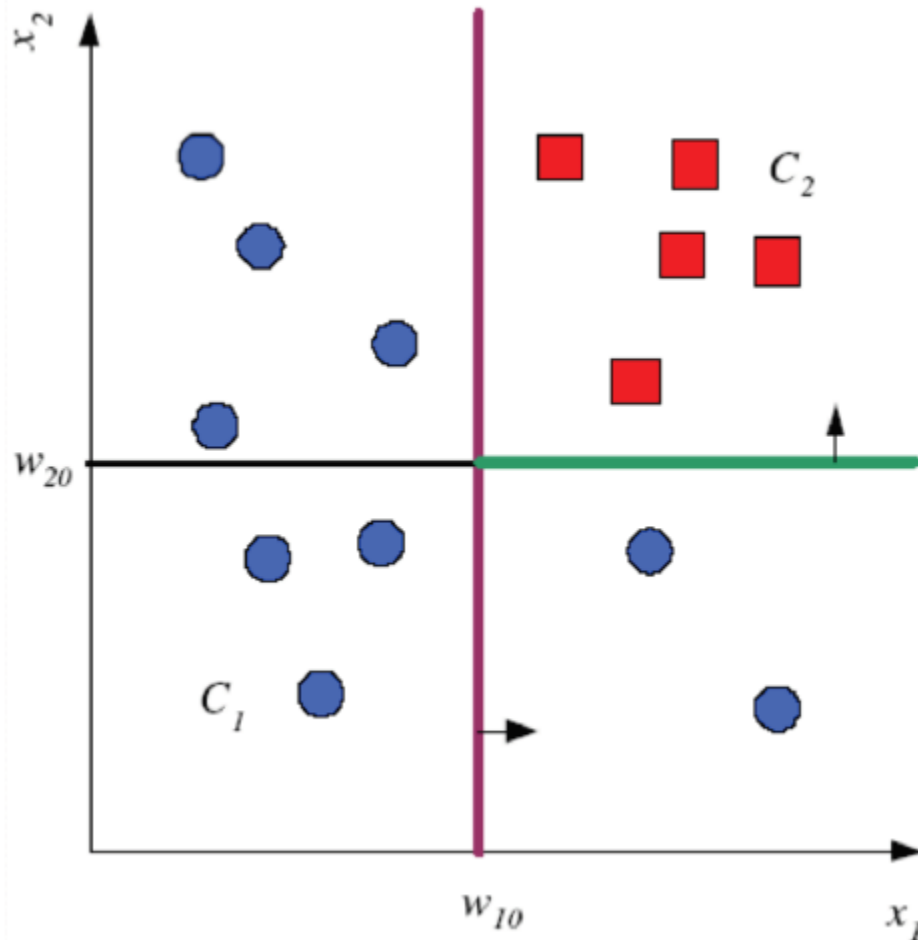




# Árvores de Decisão (*Decision Trees*)

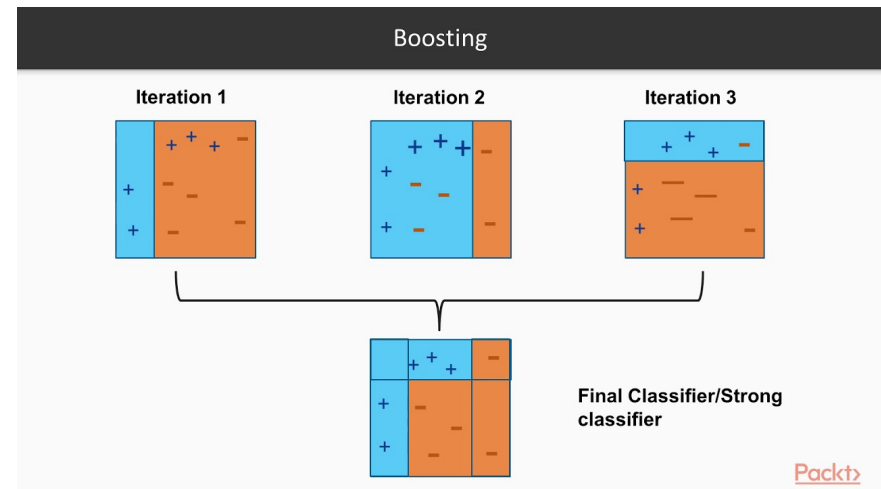
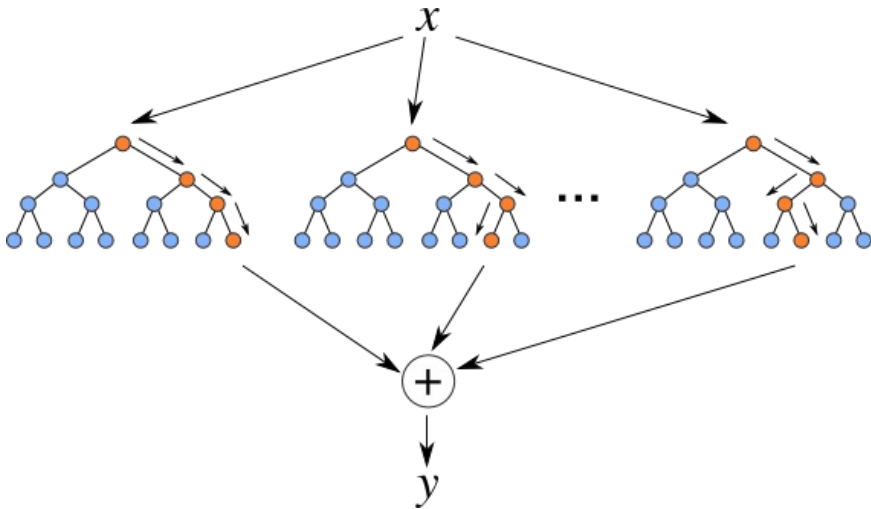


# Árvores de Decisão (*Decision Trees*)



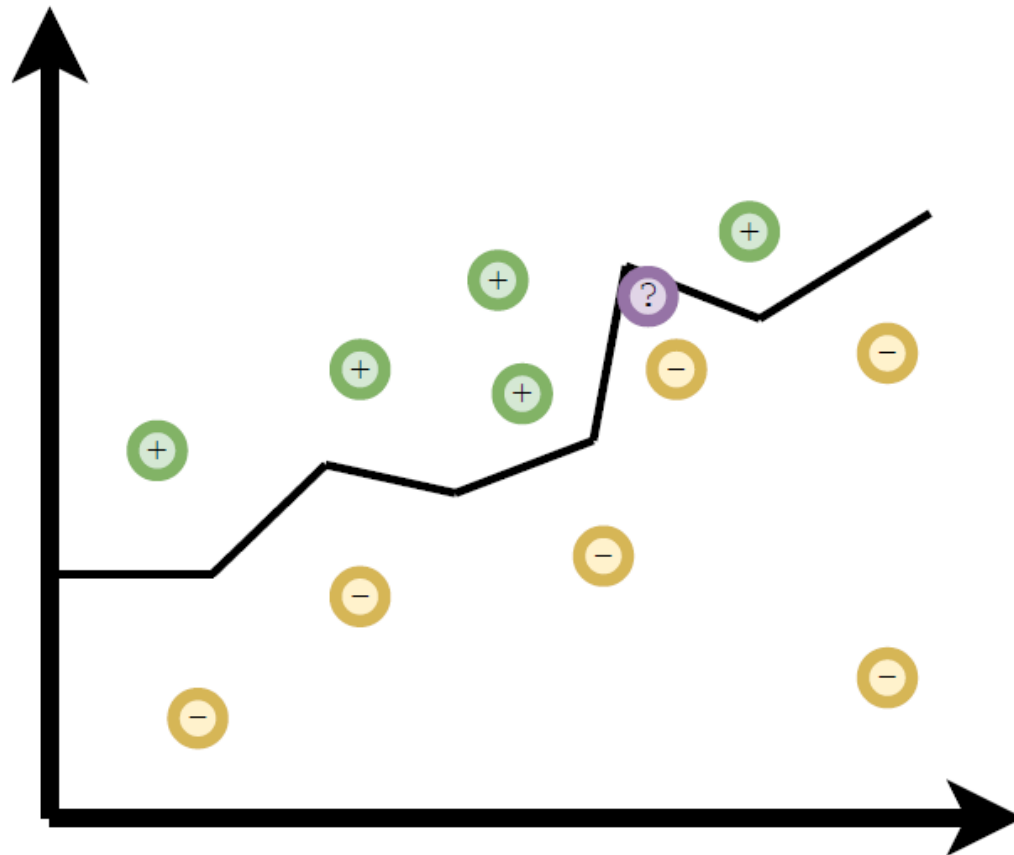
# Random Forests

- *Random* = aleatório (comportamento do algoritmo ao selecionar subconjuntos de features)
- *Forests* = florestas (várias árvores de decisão)

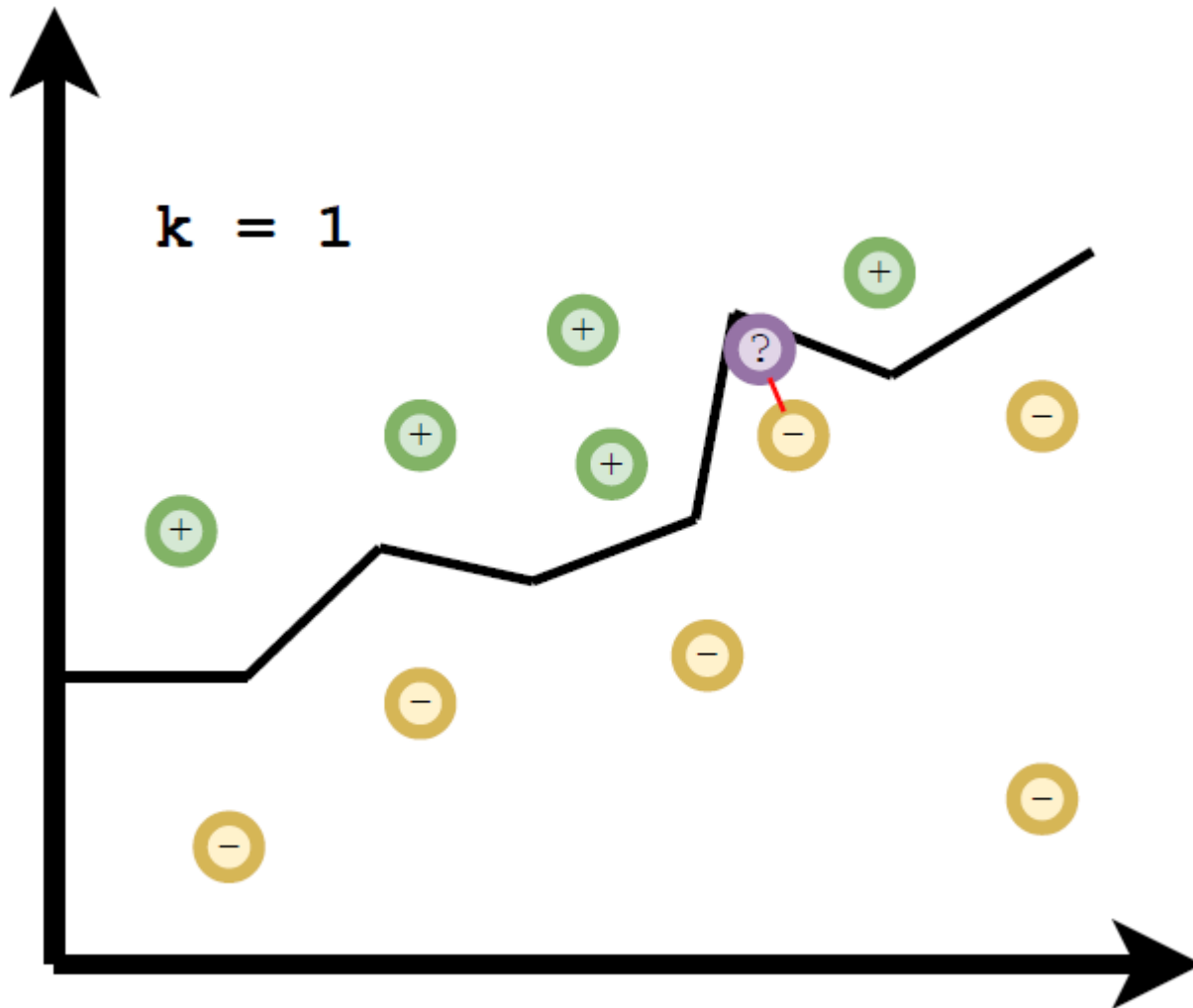


# K-nearest neighbors (KNN)

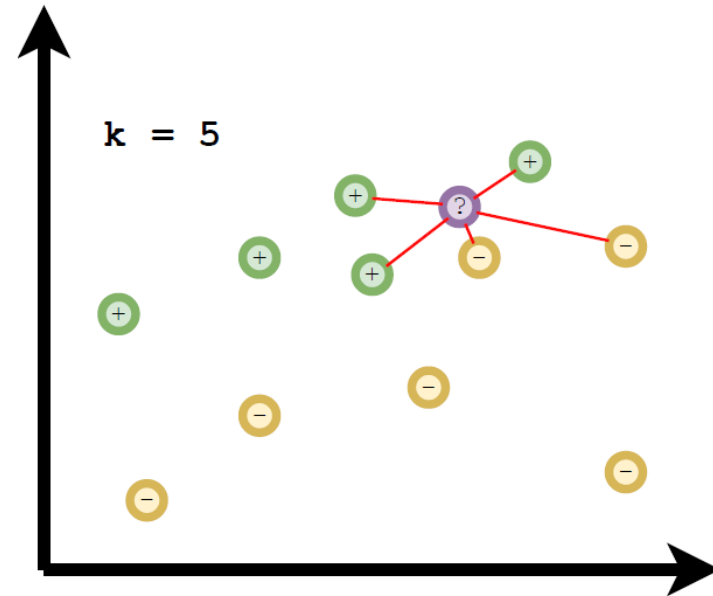
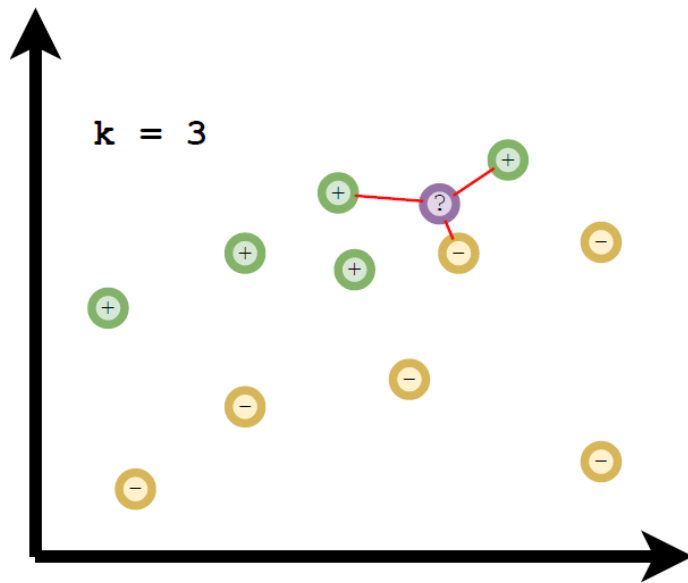
- Baseado “no quão similar” um dado é do outro.



# K-nearest neighbors (KNN)

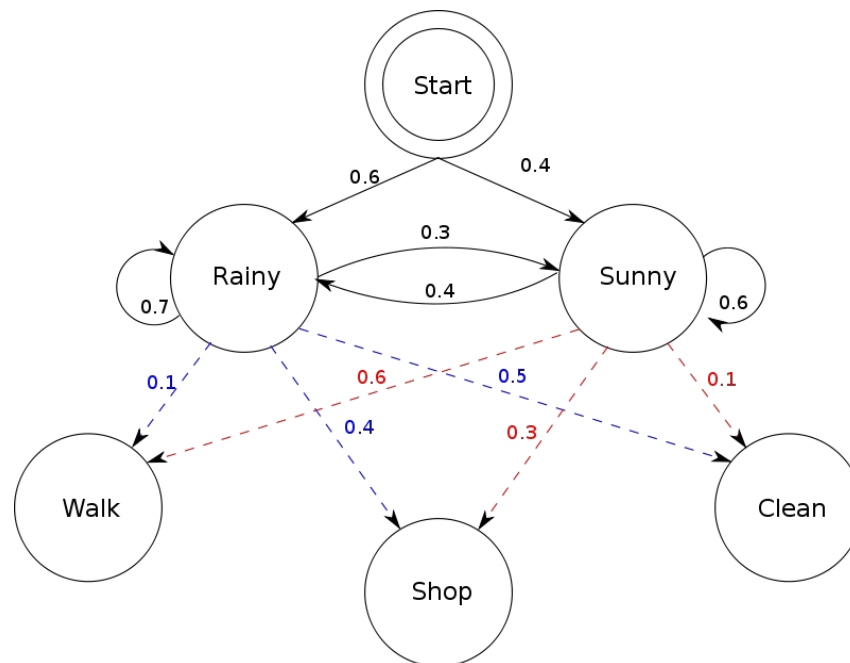


# K-nearest neighbors (KNN)



# Hidden Markov Model

- Propriedade de Markov: Todo futuro depende exclusivamente do estado atual.
- Em PLN é a suposição que a probabilidade de uma palavra depende apenas da probabilidade de uma(s) palavra(s) anterior(es).



# K-Means

- Partição das observações em k grupos onde cada observação pertence ao grupo mais próximo da média

