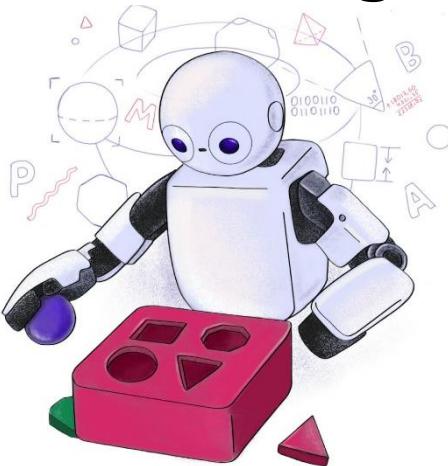


# T319 - Introdução ao Aprendizado de Máquina: *Introdução*



# Inatel

Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

# A disciplina

- *Introdução* ao aprendizado de máquina.
- Curso onde veremos os **conceitos fundamentais** do funcionamento de alguns **algoritmos de aprendizado de máquina** ou em inglês, **machine learning** (ML).
- O curso é dividido em duas partes: T319 e T320.
- O curso terá sempre uma parte **expositiva** e outra **prática** para fixação dos conceitos introduzidos.
  - Quizzes e exercícios envolvendo o uso dos conceitos discutidos em sala de aula.
- Nós não nos aprofundaremos nos conceitos matemáticos envolvidos.
- Porém, vocês precisam conhecer o básico de **Python** e alguns conceitos simples de **cálculo, álgebra linear e estatística**.

# Cronograma

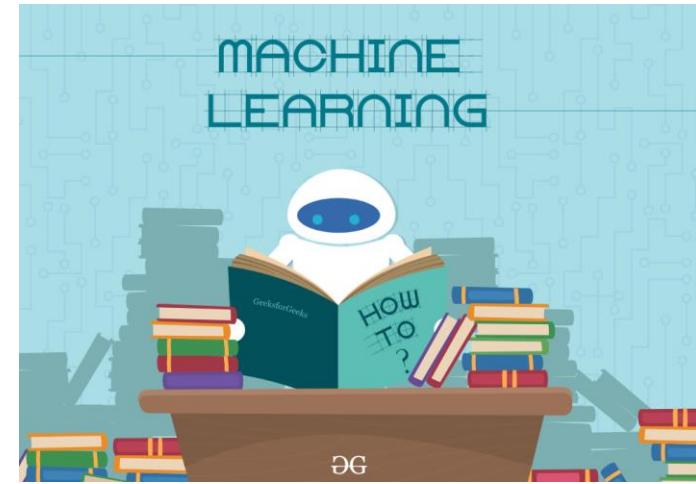
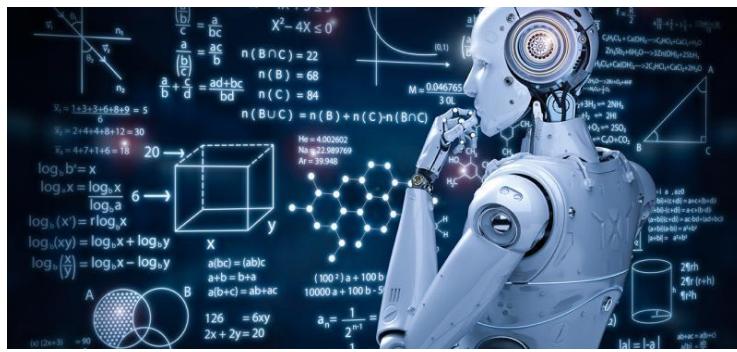
- Aulas **quinzenais**, começando sempre às 19:30.

Aula	Data	Dia	Horário	Atividade
1	13/2/2026			Introdução ao Aprendizado de Máquina
2	27/2/2026			Introdução ao Aprendizado de Máquina
3	13/3/2026			Introdução ao Aprendizado de Máquina
4	27/3/2026			Introdução ao Aprendizado de Máquina
5	10/4/2026			Introdução ao Aprendizado de Máquina
6	24/4/2026	Sexta-feira	19:30 às 21:10	Introdução ao Aprendizado de Máquina
7	8/5/2026			Introdução ao Aprendizado de Máquina
8	<b>22/5/2026*</b>			<b>Introdução ao Aprendizado de Máquina</b>
9	<b>5/6/2026</b>			<b>Avaliação Presencial (Sala I-??)</b>
10	19/6/2026			Introdução ao Aprendizado de Máquina
11	<b>3/7/2026</b>			<b>NP3</b>

\*Feriado (reposição assíncrona)

# Objetivo do curso

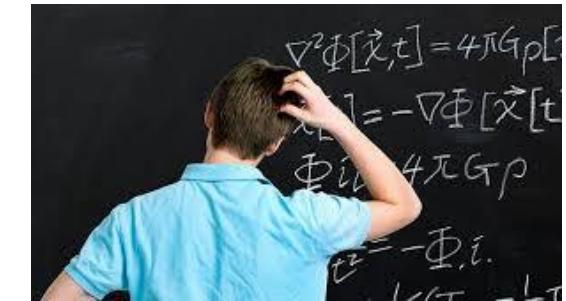
- O objetivo principal do curso é apresentar
  - os **conceitos fundamentais** da teoria do aprendizado de máquina.
  - um **conjunto de ferramentas** (i.e., algoritmos, técnicas, métricas) de aprendizado de máquina para solução de diversos tipos de problemas.
- Ao final do curso vocês devem ser capazes de
  - Entender e discutir os principais algoritmos de ML.
  - Compreender a terminologia utilizada na área.
  - Compreender o funcionamento de novos algoritmos de ML.
  - Aplicar algoritmos de ML para a resolução de problemas.



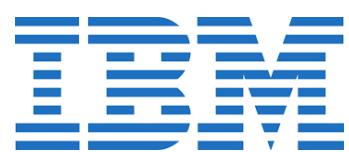
MACHINE  
LEARNING

# Critérios de avaliação

- **Um (1) trabalho** em grupo com peso de 85%.
  - Envolvendo questões práticas e/ou teóricas.
  - Uma parte do trabalho será feita **presencialmente**.
- Atividades (**quizzes e laboratórios**) com peso de 15%.
  - Podem sempre ser entregues até o fim de semana da próxima aula.
  - As atividades podem ser resolvidas em grupo com no máximo 3 alunos.
  - As atividades serão atribuídas e entregues corrigidas através do MS Teams.
- Extra: 10% da nota da FETIN (**mas precisa envolver IA**).
- **Frequência**
  - Gerada automaticamente pelo MS Teams.
  - Por favor, acompanhem suas frequências através do portal.

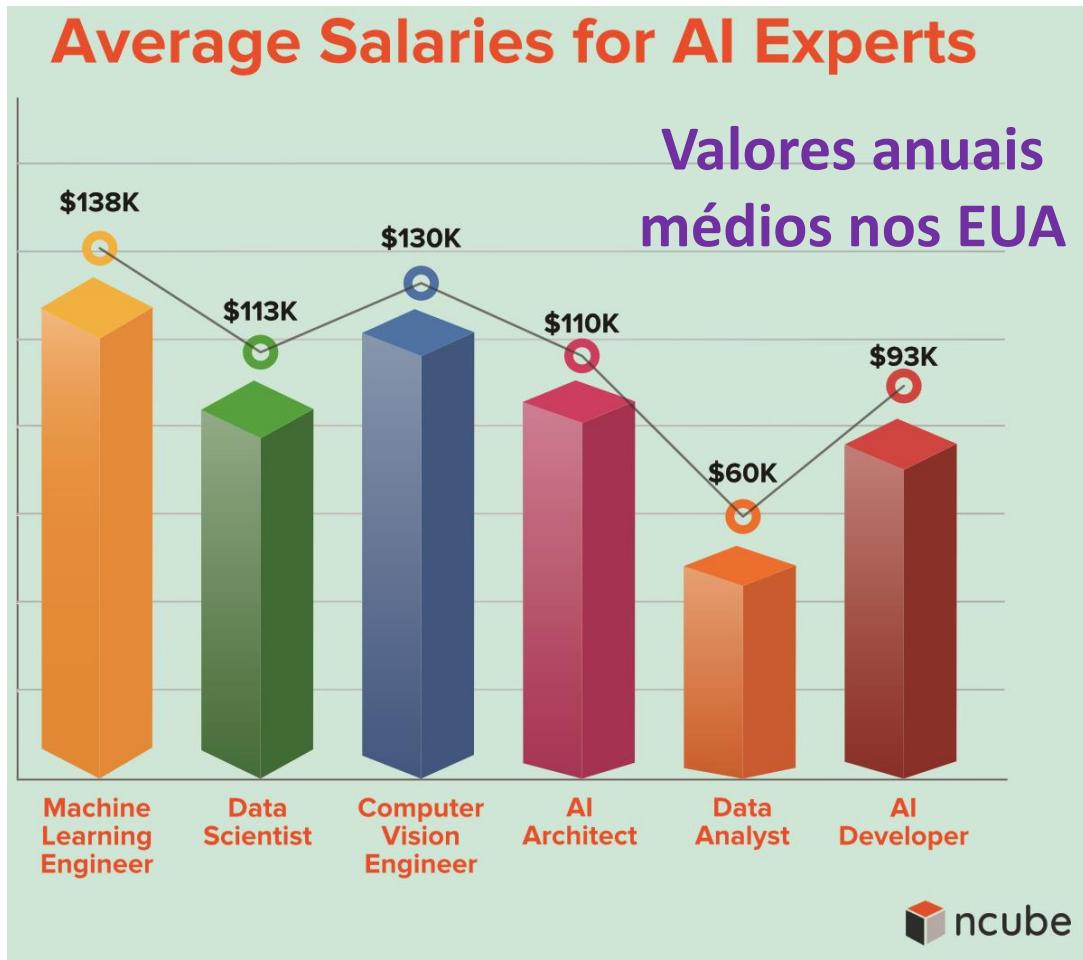


# Motivação



- **Emprego:** grandes empresas têm usado IA para
  - Automatizar tarefas repetitivas e demoradas.
  - Processar grandes volumes de dados para extração de *insights*.
  - Criar experiências personalizadas para seus clientes.
- Tudo isso para aumentar a eficiência dos processos, reduzir custos e, consequentemente, aumentar os lucros.
- Além das empresas de **tecnologia**, IA é empregada em empresas de **telecom, saúde, finanças, automação** e muitas outras.

# Motivação



- **Salário:** como as empresas têm se apoiado mais e mais em IA para aumentar seus lucros, elas têm tido uma demanda crescente por profissionais com esse conhecimento.
- Os *salários são muito bons*, pois a demanda é alta, *mas não existem tantos profissionais* com essa experiência no mercado.

# Motivação



NOKIA

Qualcomm

ZTE



UNIVERSITY  
OF OULU



- **Pesquisa:** IA já tem sido empregada no 5G e terá um papel fundamental no desenvolvimento da próxima geração de redes móveis e sem fio, o 6G.
- Os algoritmos de IA serão usados nas ***mais diversas camadas*** da rede para auxiliar em tarefas que vão desde a ***estimação de canal, detecção de símbolos de uma modulação, alocação de recursos (e.g., feixes, potência, subportadoras) até segurança cibernética e chatbots para atendimento.***

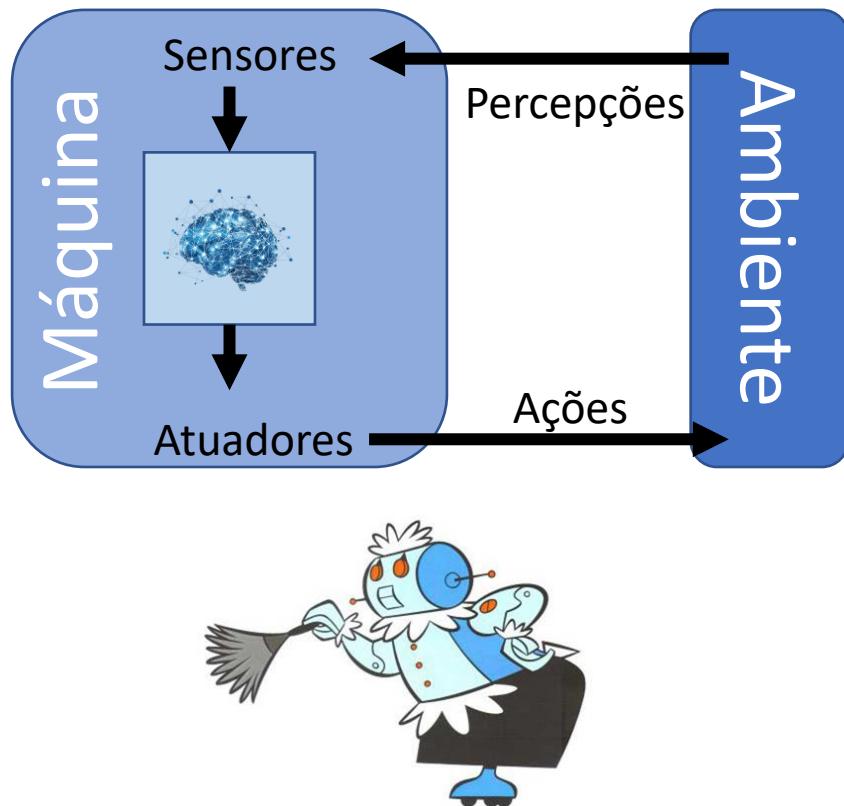
*Mas o que é inteligência  
artificial?*

# O que é inteligência artificial?



- Campo da computação que almeja *criar máquinas que imitem as capacidades humanas*.
- Que capacidades são essas?
  - Aprendizado;
  - Adaptação;
  - Comunicação;
  - Raciocínio e tomada de decisão;
  - Resolução de problemas;
  - Criatividade;
  - Movimento;
  - etc.

# O que é inteligência artificial?



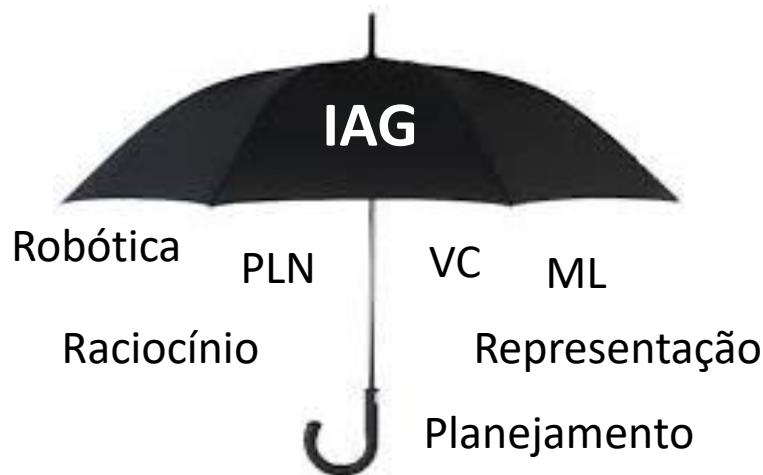
- **Definição formal:**

Capacidade de uma máquina de *receber estímulos* vindos do ambiente, *interpretá-los, aprender* com eles e usar o *conhecimento adquirido* para *tomar decisões e resolver problemas, interagindo* com o ambiente.

- Porém, criar máquinas que **emulem todas** as nossas capacidades de uma única vez, **não é uma tarefa simples**.

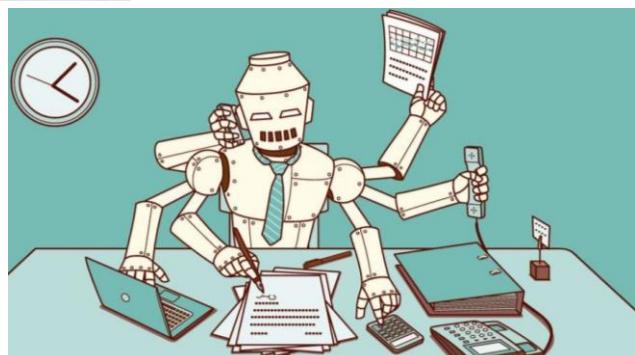
# O problema da inteligência artificial

*IA é uma área muito ampla que engloba várias aplicações (subáreas) diferentes.*



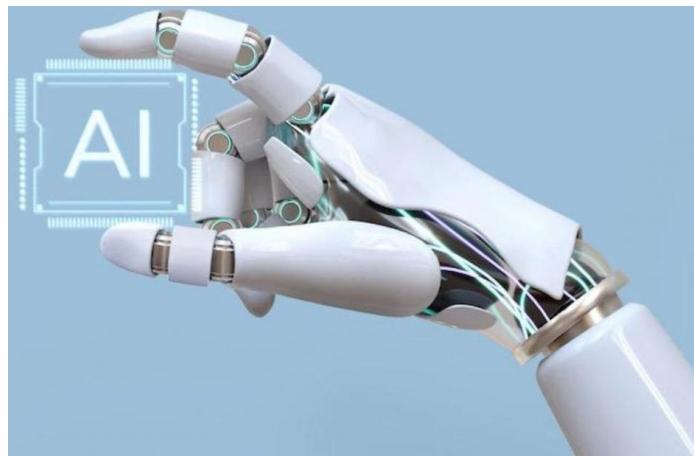
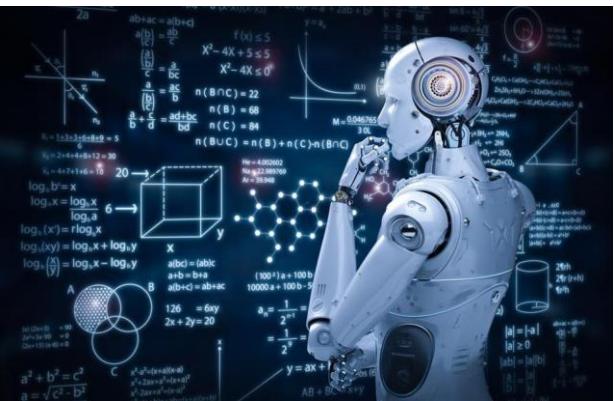
- Portanto, criar máquinas que imitem a inteligência humana é um **problema muito complexo e difícil de ser resolvido de uma só vez**.
- Sendo assim, **divide-se o problema em problemas menores**, chamadas de **subáreas da IA**.

# Subáreas da inteligência artificial



- **Processamento de linguagem natural**
  - Compreensão e interpretação de linguagens humanas.
- **Representação do conhecimento**
  - Extração e armazenamento eficiente de conhecimento do mundo real.
- **Raciocínio automatizado**
  - Resolução de problemas complexos a partir de conhecimento prévio.
- **Planejamento**
  - Criação de planos que permitam que uma máquina execute uma tarefa.

# Subáreas da inteligência artificial



- **Visão computacional**

- Compreensão e interpretação de imagens e vídeos.

- **Robótica**

- Criação de robôs capazes de realizar tarefas físicas e interagir com o ambiente.

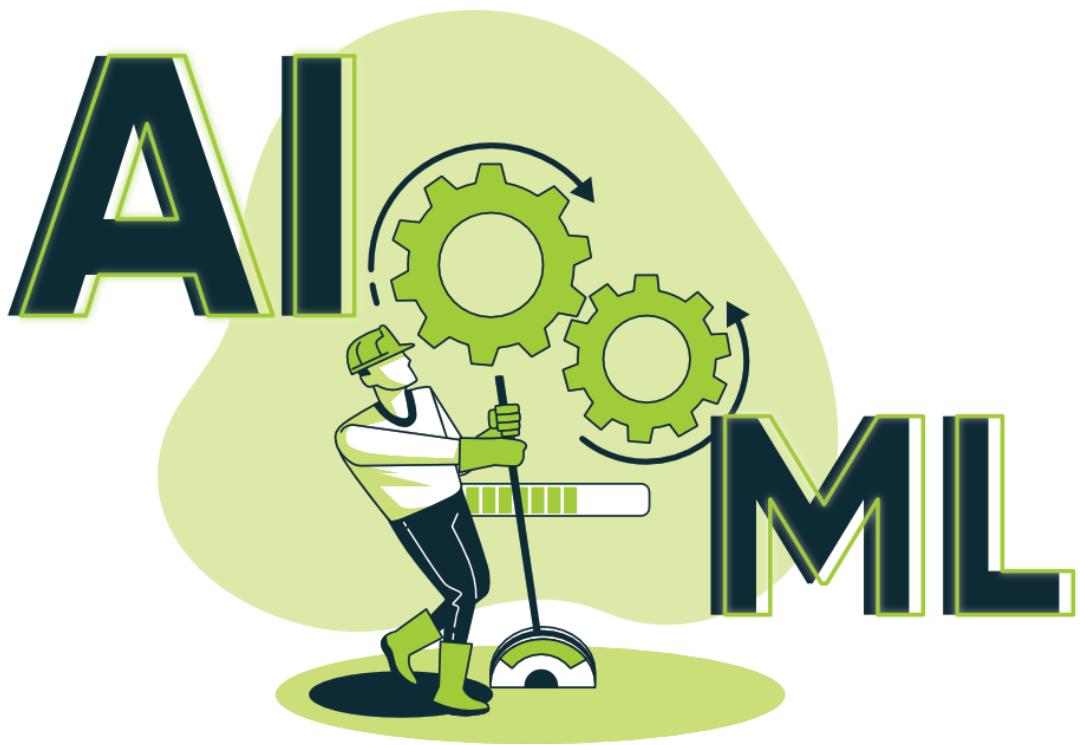
- **Aprendizado de máquina**

- Criação de máquinas que aprendem através de experiências prévias.

- **Inteligência artificial geral (IAG)**

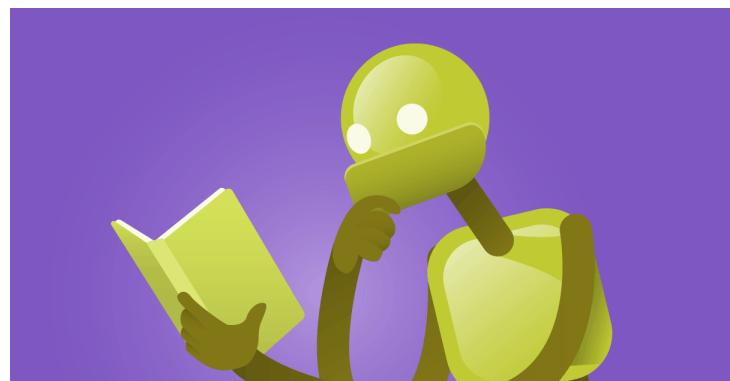
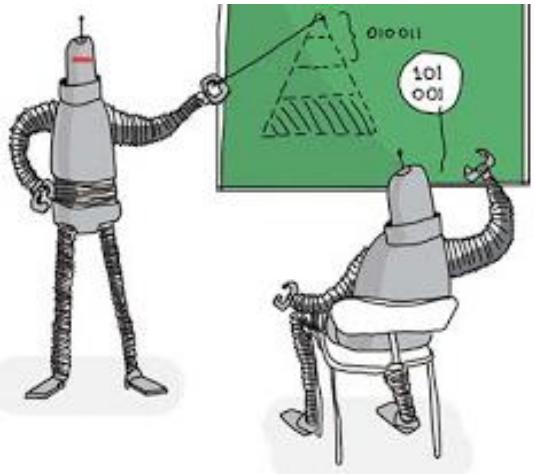
- Criação de máquinas que se comportem como seres humanos. É a meta final da IA.

# Subáreas da inteligência artificial



- Como vimos, IA é um área muito ampla e não teríamos tempo para ver todas as subáreas.
- Além disso, ML é a base da IA moderna.
  - **Exemplos:** Redes convolucionais (imagens), *transformers* (ChatGPT), modelos de difusão (geração de dados sintéticos).
- Assim, focaremos no estudo dos fundamentos do ***aprendizado de máquina***.

# Por quê ML?



- **Caixa de ferramentas:** ML oferece **ferramentas para a análise e solução de problemas** em várias áreas de forma eficiente.
- **Redução de complexidade e custo:** ML pode **reduzir o custo e a complexidade computacional** de métodos muitas vezes com desempenho ótimo na teoria, mas inviáveis na prática devido a sua alta complexidade e/ou custo.
- **Oportunidades:** existem muitos **empregos** na área de análise, ciência e engenharia de dados, além de **pesquisas inovadoras**, que usam ML para a solução de problemas em diversas áreas.

*OK, mas o que é o aprendizado de máquina?*

# O que é o aprendizado de máquina?

- É uma das subáreas da inteligência artificial.
- O termo foi cunhado em 1959, pelo cientista da computação Arthur Samuel, que o definiu como o

*“Campo de estudo que dá aos computadores a habilidade de **aprender sem serem explicitamente programados.**”*



# O que é o aprendizado de máquina?

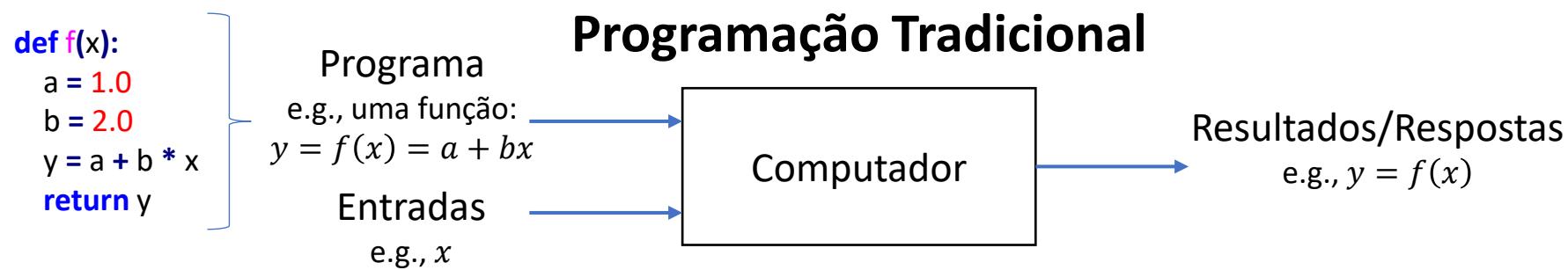


- Mas como eles aprendem?
  - Através de *experiências prévias*, induz-se conhecimento nas máquinas.
- Algoritmos de ML são *orientados a dados*, i.e., eles *aprendem (através de treinamento)* uma *solução geral* a partir de *conjuntos de dados* fornecidos a eles.

*Conjuntos de dados = experiências prévias*

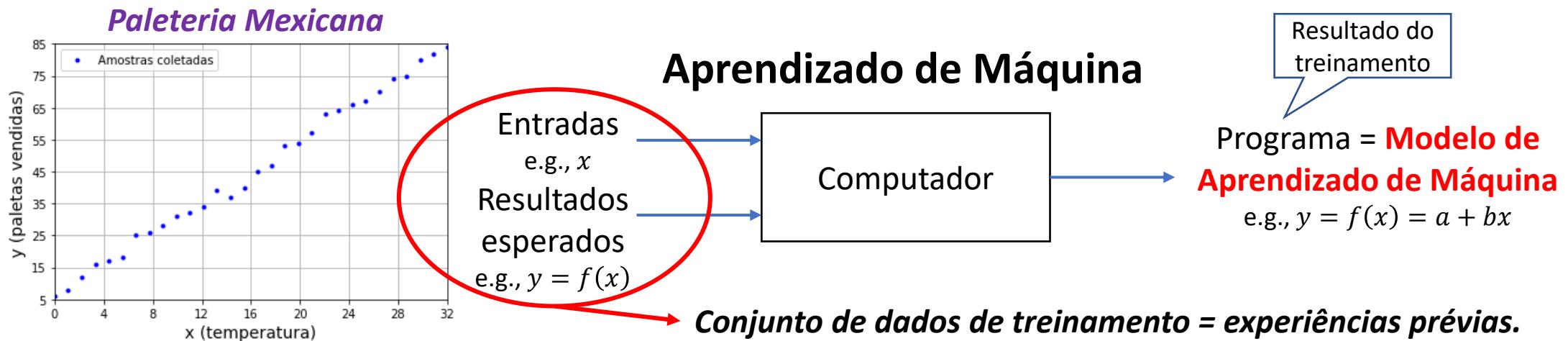
# Paradigma da programação tradicional

- Para entendermos o que é o aprendizado de máquina, vamos fazer um *paralelo com a programação tradicional*.
- Na programação tradicional, o *programador cria as regras* (i.e., programa) *que mapeiam as entradas,  $x$ , nas saídas,  $y = f(x)$* .
- Imaginem um problema onde a *solução geral* é a equação de uma reta,  $f(x)$ .



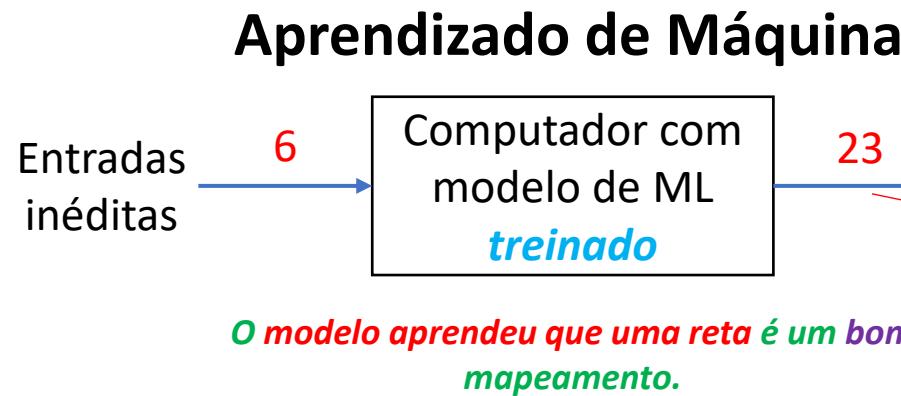
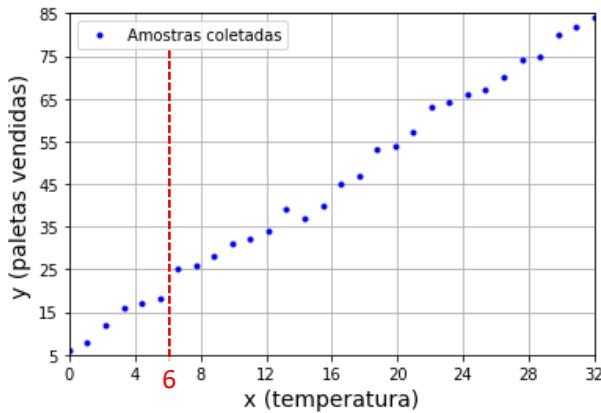
# O paradigma do aprendizado de máquina

- “... *aprender sem serem explicitamente programados.*”
- Esse trecho pode ser entendido se reorganizarmos a figura anterior.
- No ML, nós fornecemos as **entradas** e as **respostas esperadas** ao computador e deixamos que ele **aprenda**, através de **treinamento**, um **modelo** (i.e., as regras) que **mapeie** as entradas nas respostas esperadas.

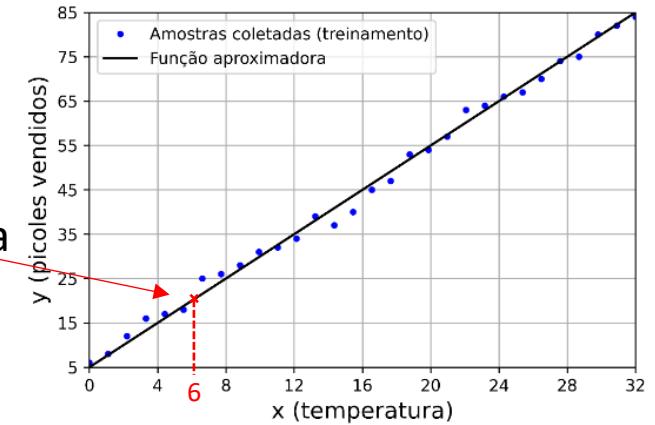


# Generalização

- Porém, não basta que o algoritmo de ML aprenda um modelo que faça um bom mapeamento *apenas para os dados do conjunto de treinamento*.
- O **algoritmo** de ML deve *treinar* um *modelo* que *aprenda* uma *solução geral*, ou seja, que *generalize para entradas não vistas durante o treinamento*.



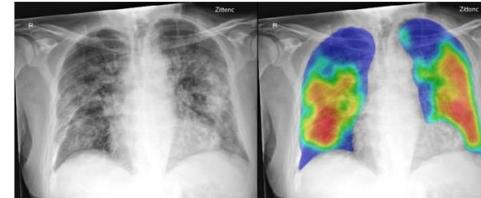
Qual é a estimativa de paletas vendidas quando a temperatura é de  $\approx 6$  graus (valor não visto durante o treinamento)?



A partir do mapeamento aprendido (i.e., reta), o modelo gera como saída o valor 23, que é coerente com o restante dos dados.

# Exemplos de aplicações de ML em várias áreas

- **Transporte:** veículos autônomos.
- **Negócios:** recomendação de produtos e conteúdos (e.g., amazon e netflix).
- **Educação:** pontuação automatizada de fala em testes de Inglês.
- **Saúde:** detecção e diagnóstico de doenças (câncer, Alzheimer, pneumonia, COVID-19, etc.).
- **Finanças:** detecção de fraudes com cartão de crédito.
- **Tecnologia:** assistentes pessoais (e.g., chatGPT, *Siri*, *Alexa*, *Cortana*, etc.).



# Principais motivos da difusão do ML

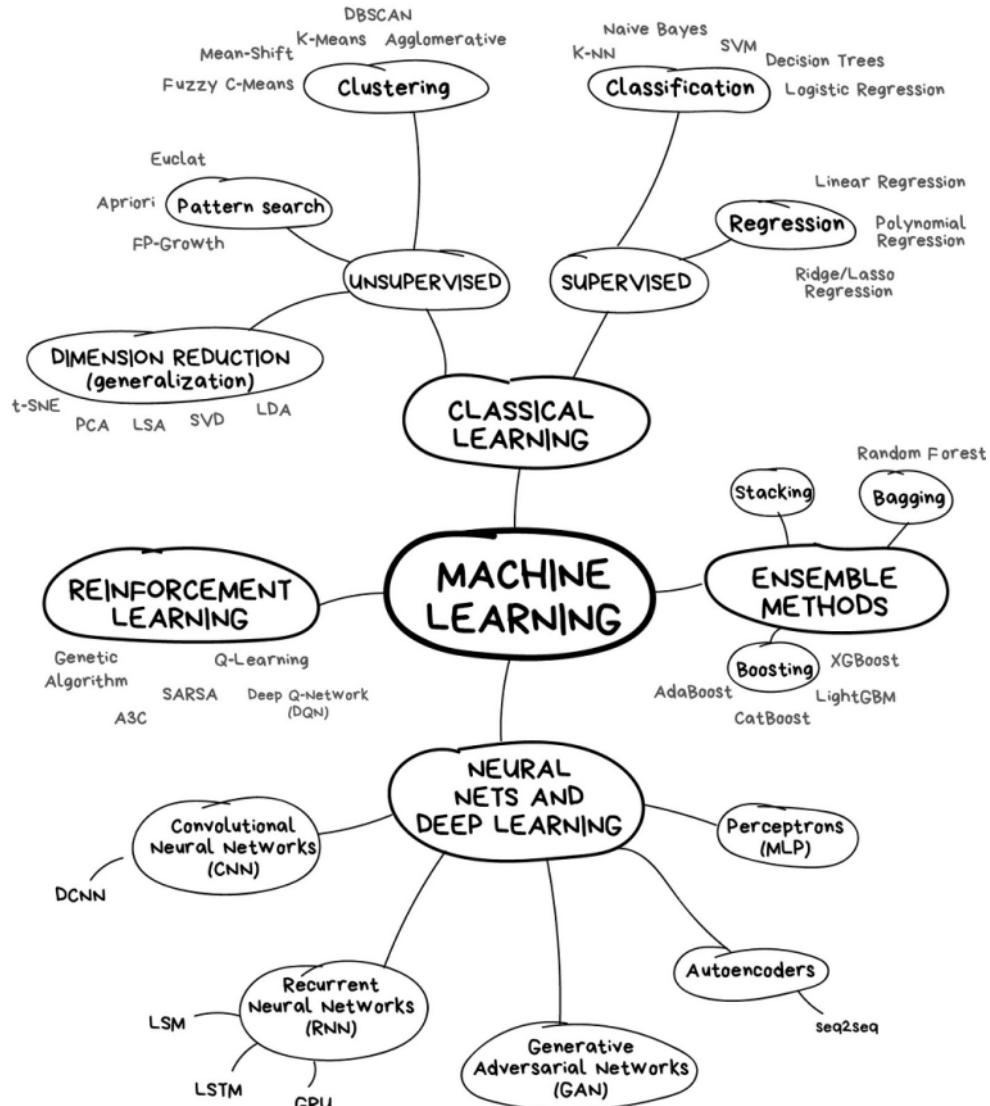
- Grandes volumes de dados disponíveis: geramos centenas de *terabytes* por dia.
- Possibilidade de *extrair informações úteis do enorme volume de dados* disponível atualmente, algo que seria impossível para nós humanos.
  - Isso *vale ouro*, pois tem grande potencial para *aumentar o lucro das empresas*.
- Surgimento de *recursos computacionais poderosos* tais como GPUs, FPGAs e CPUs com múltiplos cores.
- Surgimento de *novas estratégias de aprendizagem*, e.g., *deep-learning*, *deep reinforcement-learning*, modelos generativos, *transformers* etc.
- Disponibilidade de *bibliotecas que facilitam o desenvolvimento* de soluções com ML.
  - Tensorflow, PyTorch, Scikit-learn, etc.



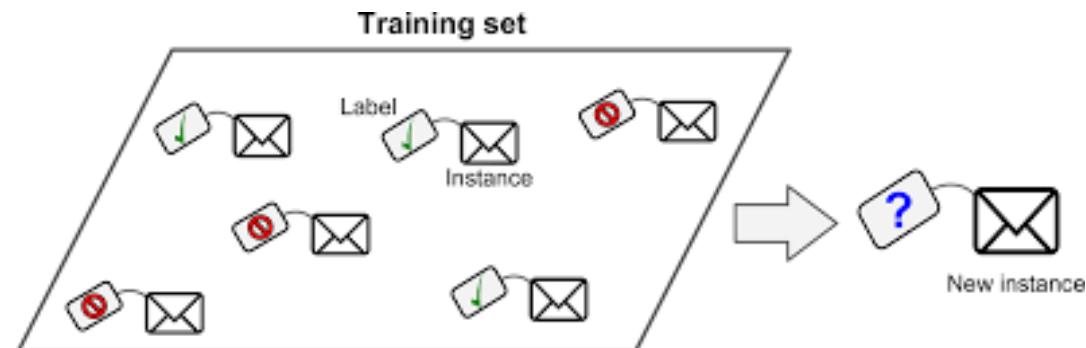
# Paradigmas de aprendizado de máquina

Os algoritmos de aprendizado de máquina podem ser *agrupados* de acordo com o *tipo de aprendizado que realizam*:

- Supervisionado
- Não-Supervisionado
- Por Reforço
- Metaheurístico



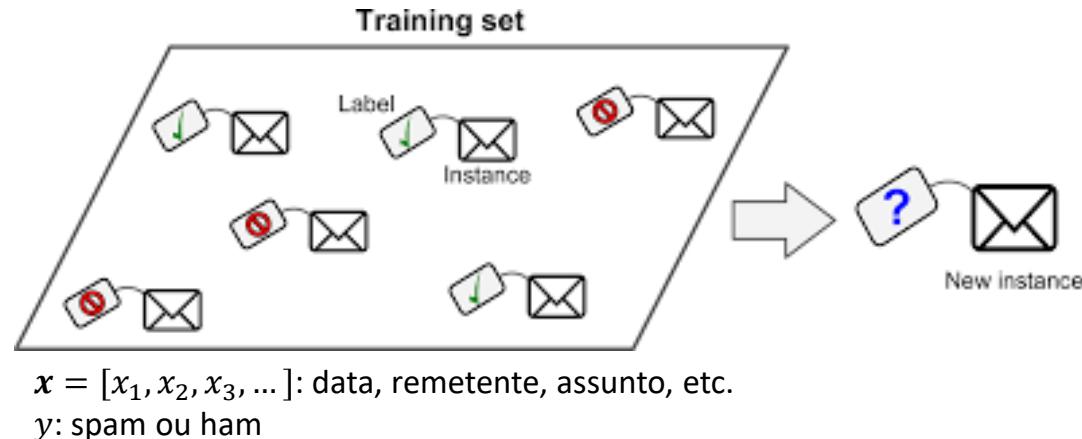
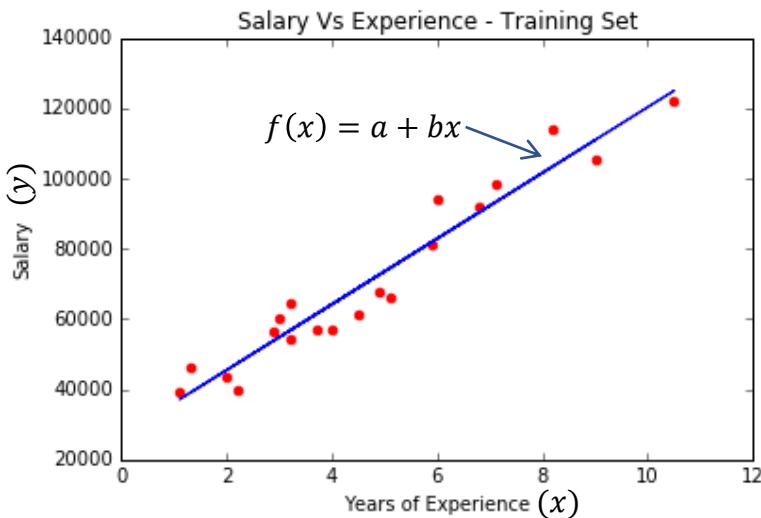
# Aprendizado supervisionado



$x = [x_1, x_2, x_3, \dots]$ : data, remetente, assunto, etc.  
 $y$ : spam ou ham

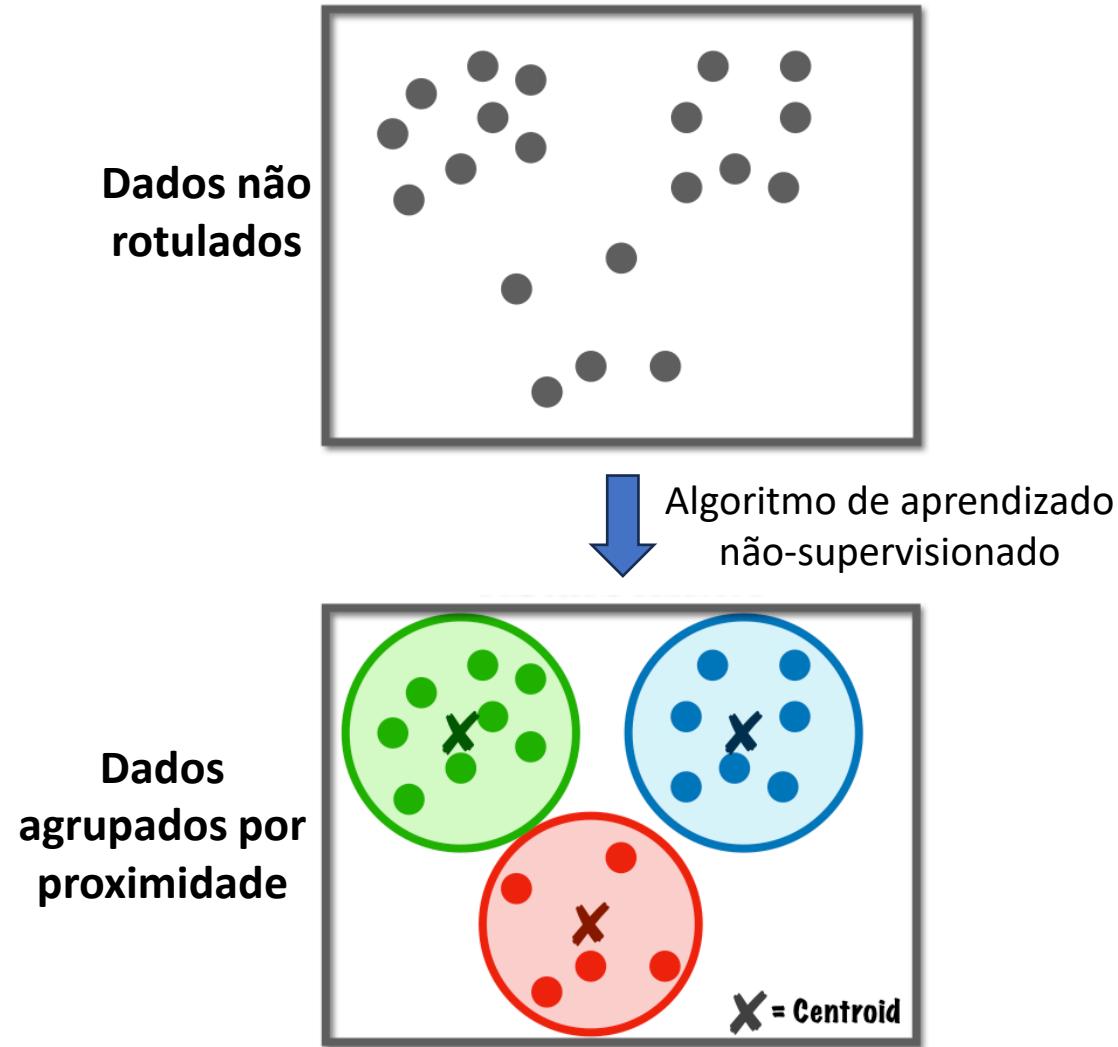
- No aprendizado supervisionado, o **algoritmo de ML tem acesso às saídas esperadas**,  $y$ , chamadas de **rótulos** (ou *labels*, em inglês), **para o conjunto de valores de entrada**, chamados de **atributos**,  $x$ .
- Em outras palavras, cada **exemplo de treinamento** é composto pelos valores de entrada,  $x$  (**atributos**), e sua saída correspondente,  $y$  (**rótulo**).

# Aprendizado supervisionado



- **Objetivo:** os algoritmos *supervisionados* de ML devem *aprender* uma *função* (i.e., o modelo) que *mapeie* as entradas  $x$  nas saídas esperadas,  $y$ , ou seja,  $y = f(x)$ .
- Esse tipo de aprendizado é dividido em *problemas* de *regressão* e *classificação*.
  - **Regressão:** o rótulo,  $y$ , pertence a um *conjunto infinito* de valores, i.e., números reais.
    - **Exemplo:** anos de experiência versus salário.
  - **Classificação:** o rótulo,  $y$ , pertence a um *conjunto finito e discreto* de valores, i.e., número de possíveis classes.
    - **Exemplos:** filtro de spam, classificação de imagens.

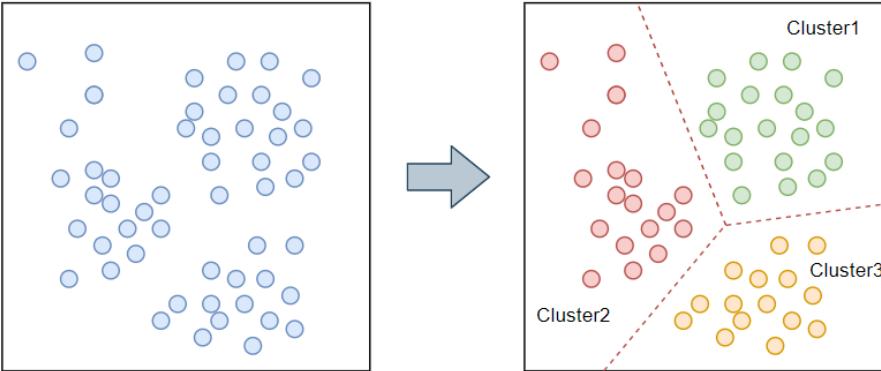
# Aprendizado não-supervisionado



- Neste tipo de aprendizado, os algoritmos de ML **não têm acesso às saídas esperadas**,  $y$ .
- Os algoritmos só têm acesso aos **atributos**,  $x$ .
- **Objetivo:** os algoritmos devem *aprender/descobrir* padrões, muitas vezes ocultos, presentes nos dados se baseando apenas, por exemplo, na **similaridade** entre os **atributos**,  $x$ , ou seja, **sem a presença de rótulos**.

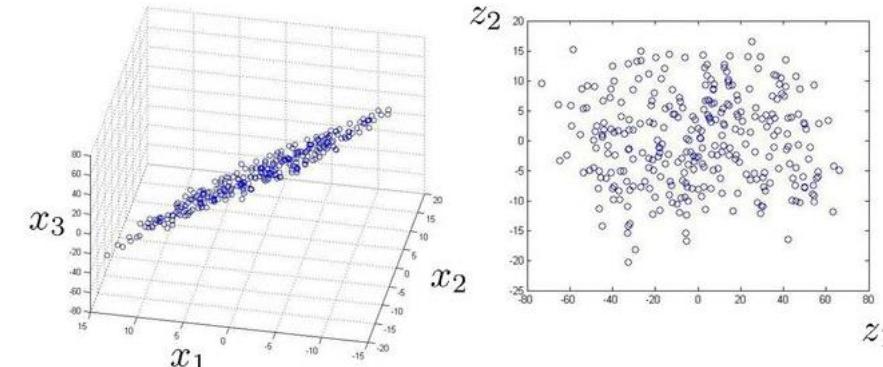
# Aprendizado não-supervisionado

Clusterização

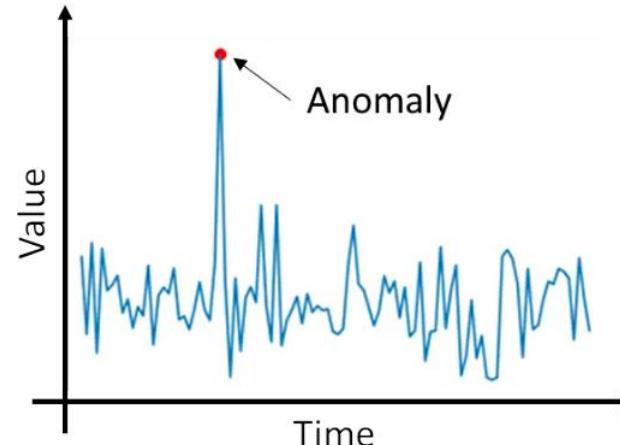


Redução de dimensionalidade

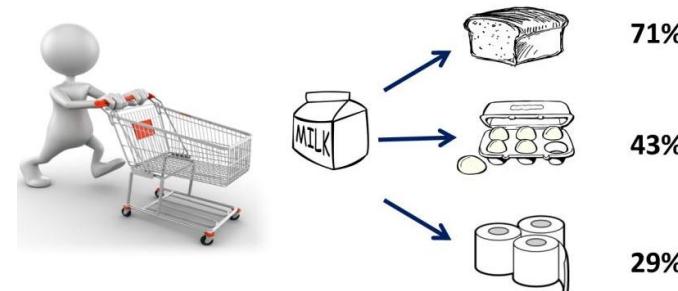
Reduce data from 3D to 2D



Detecção de Anomalias



Regras de associação

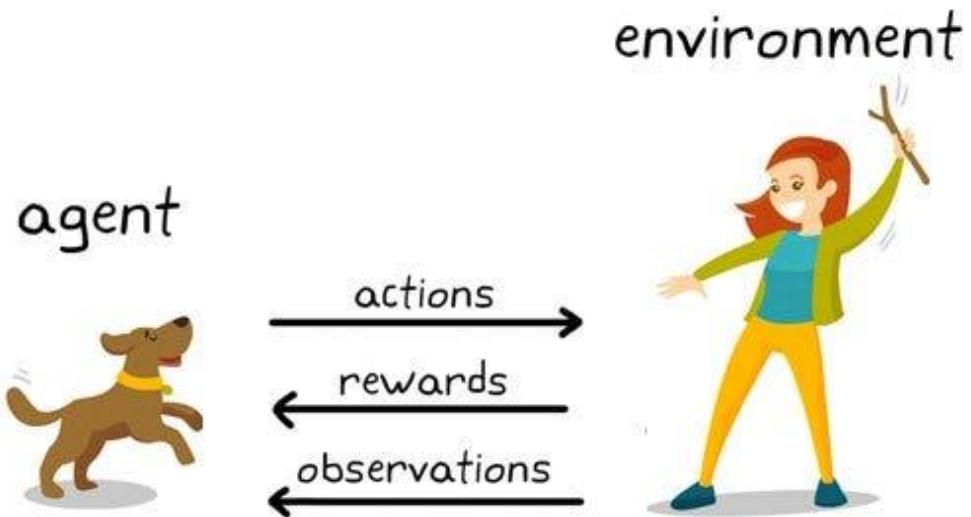


Of transactions that included milk:

- 71% included bread
- 43% included eggs
- 29% included toilet paper

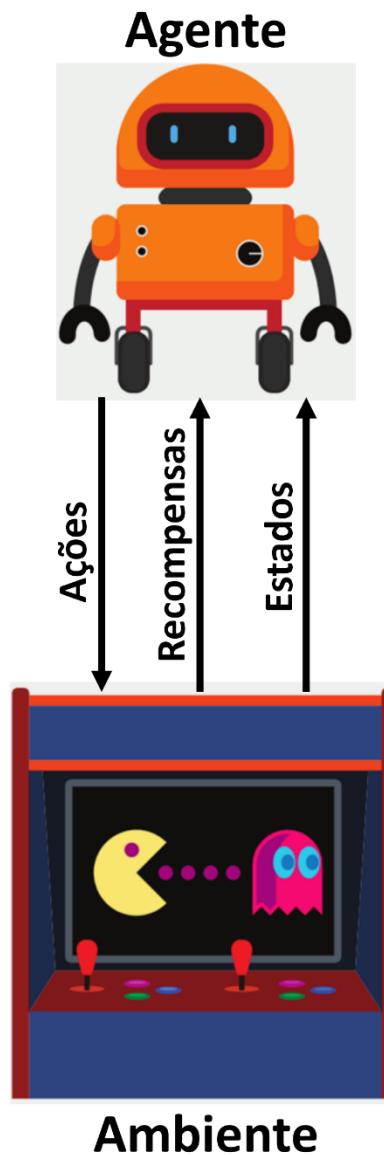
- Esses algoritmos de ML tratam problemas de **clusterização, detecção de anomalias (outliers), redução de dimensionalidade, e aprendizado de regras de associação.**

# Aprendizado por reforço



- Abordagem de aprendizado totalmente diferente das anteriores, pois **não temos exemplos de treinamento**, sejam eles rotulados ou não.
- O algoritmo de aprendizado por reforço, chamado de **agente** nesse contexto, aprende como se comportar em um **ambiente** através de interações do tipo **tentativa e erro**.
- O **agente** observa o **estado** do **ambiente**, seleciona, executa uma **ação** e recebe um **reforço positivo ou negativo** em consequência da **ação** tomada.

# Aprendizado por reforço

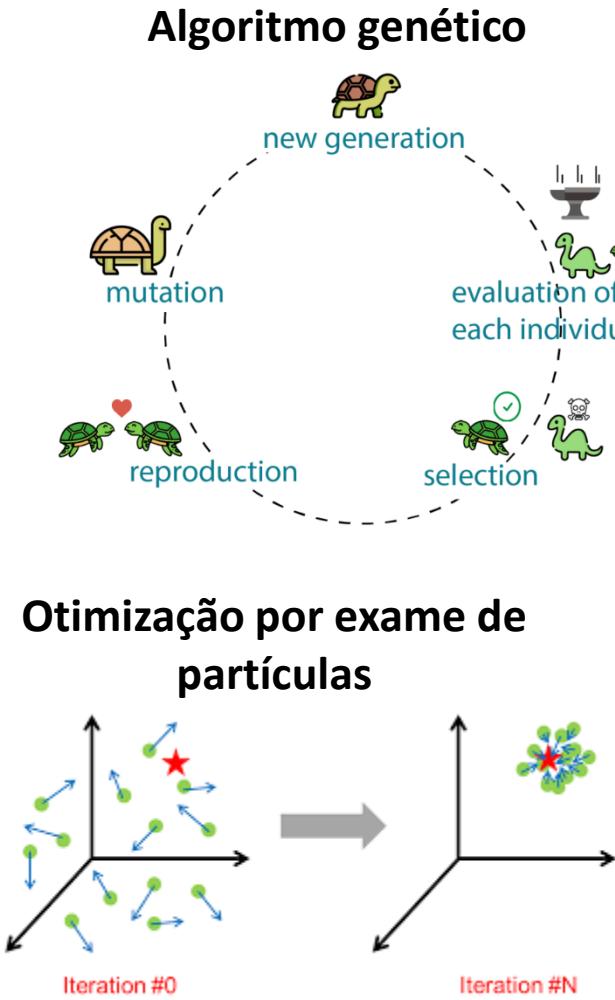


- Seguindo estes passos, o agente aprende por si só qual a melhor **estratégia**, chamada de **política**, para obter a **maior recompensa possível ao longo do tempo**.
- Uma **política** define qual **ação** o **agente** deve escolher quando o **ambiente** estiver em um determinado **estado**.
- Portanto, a **política** é uma **função que mapeia os estados do ambiente em ações** que o **agente** deve tomar para **maximizar as recompensas**.

# Aprendizado metaheurístico

- **Metaheurísticas** são **estratégias gerais de otimização** usadas para encontrar boas soluções (muitas vezes próximas da ótima, i.e., **subótimas**) para **problemas complexos**, especialmente quando:
  - O espaço de busca é muito grande
  - O problema é NP-difícil
    - Problemas para os quais não existe algoritmo rápido que os resolva exatamente para todos os casos.
  - Não existe solução exata viável computacionalmente
  - O modelo matemático é não linear ou não diferenciável
- Elas não garantem a solução ótima, mas geralmente produzem **boas soluções em tempo razoável**.

# Principais tipos de metaheurísticas

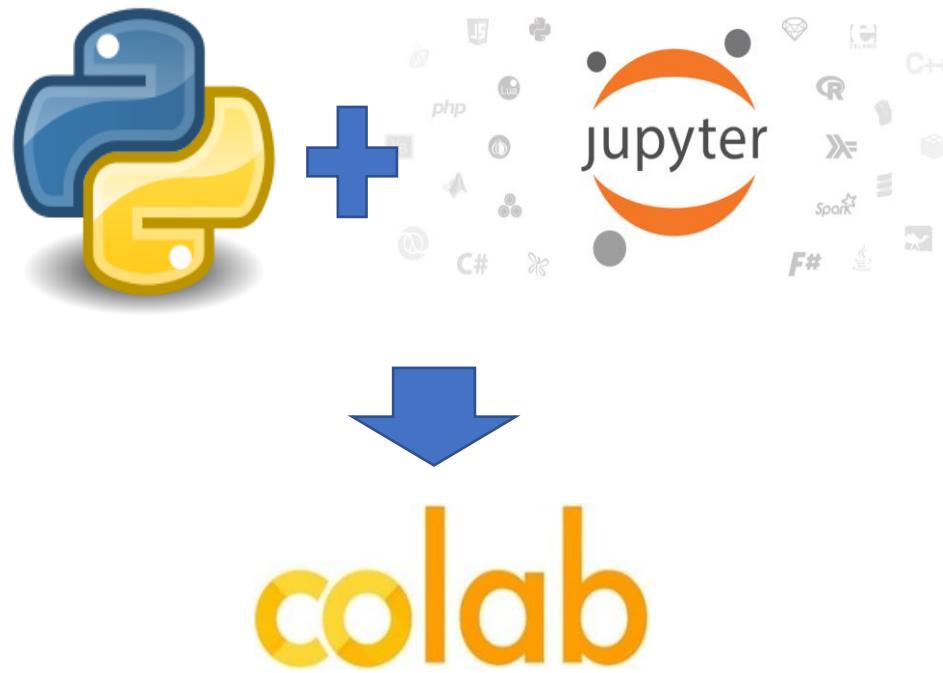


- Inspirados na natureza
  - Otimização por colônia de formigas
    - Baseada no comportamento de formigas encontrando caminhos mínimos.
  - Otimização por exame de partículas
    - Inspirada no movimento coletivo de pássaros ou peixes.
  - Algoritmo genéticos
    - Baseado em evolução natural dos seres vivos.
- Baseadas em busca local
  - Tabu Search
    - Mantém memória para evitar revisitar soluções.

# Características das metaheurísticas

- São **genéricas** (podem ser aplicadas a vários problemas).
  - Não foram projetadas para um único problema específico.
- Trabalham com **busca estocástica**.
  - Exploram o espaço de soluções probabilisticamente.
- Não exigem o cálculo de derivadas como regressores, redes neurais, etc.
- Funcionam bem mesmo em dispositivos com **capacidade computacional limitada** (e.g., dispositivos IoT).
- Equilibram exploração vs. exploração.
  - Explora soluções em novas regiões do espaço de soluções ao mesmo tempo em que se refina as melhores soluções já encontradas.

# Executando códigos



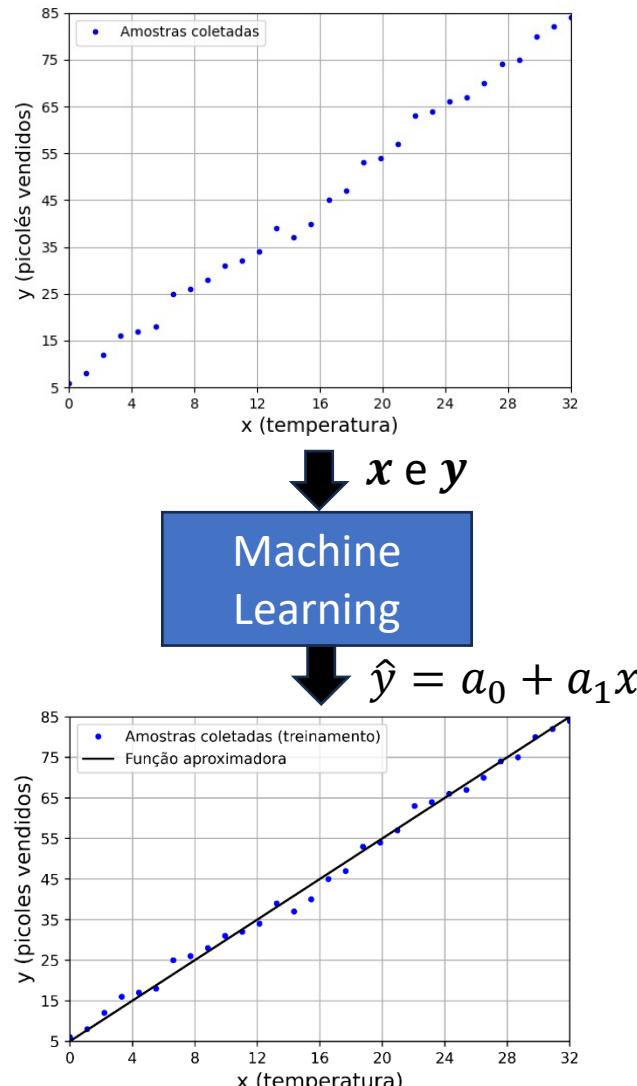
- Durante o curso, usaremos ***Python***.
  - Pois é a linguagem mais utilizada em ML além de ser gratuita e *open-source*.
- Utilizaremos ***notebooks Jupyter*** para execução de exemplos e resolução dos laboratórios.
  - Eles são ***documentos virtuais*** usados para desenvolver e documentar código.
  - Pode-se adicionar equações, gráficos e texto.
- Para executá-los, usaremos o ***Google Colab***.

# Goolge Colaboratory (Colab)



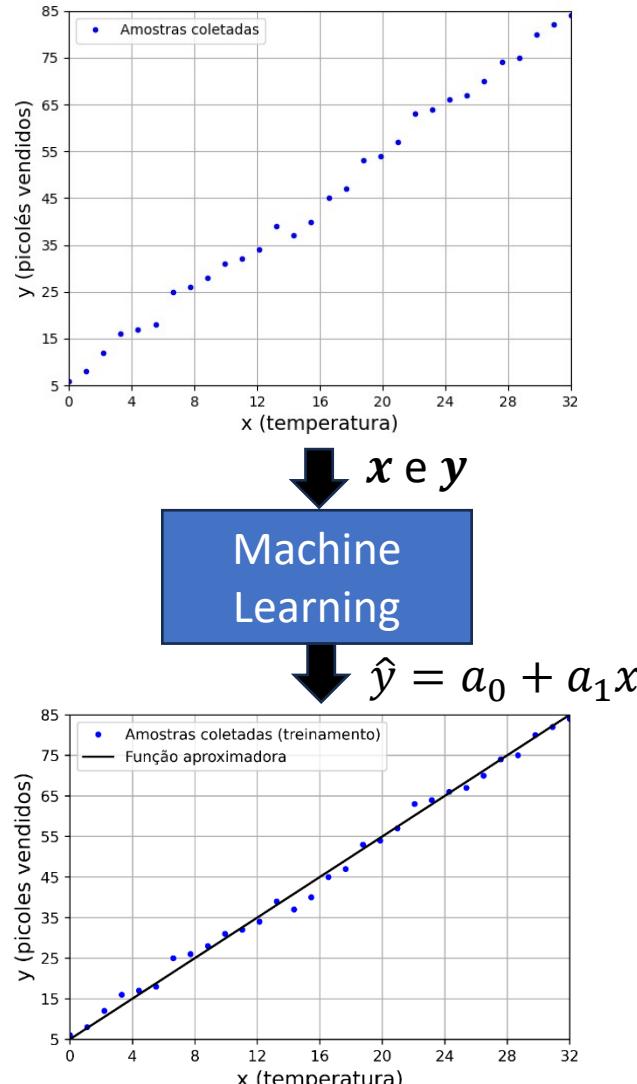
- **Colab** é uma aplicação web gratuita que permite a *criação, edição e execução* de *notebooks Jupyter* em navegadores web.
- É um produto da Google.
- Vantagens:
  - Grande número de servidores.
  - Fornece acesso a GPUs e TPUs gratuitamente.
  - Notebooks podem ser salvos no Google Drive, evitando a perda de códigos.
- URL: <https://colab.research.google.com/>
- **Vocês não precisam instalar nada**, apenas terem conexão com a internet.

# Objetivo do curso



- O **objetivo** desta primeira parte do curso é ensinar a vocês *como encontrar uma função* (i.e., *um modelo*), *usando ML*, que *aproxime* (i.e., *aprenda*) o comportamento geral por trás de um *conjunto de amostras* (*x e y*) da *melhor forma possível*.
  - Queremos um bom mapeamento tanto para amostras conhecidas quanto para inéditas.

# Objetivo do curso



- Mas o que você quer dizer com “**melhor forma possível**”?
  - No sentido da otimização de uma dada métrica.
- Na maioria dos casos, o **mapeamento verdadeiro** entre  $x$  e  $y$  **não é conhecido ou nem mesmo existe** e nós nos baseamos apenas em uma **métrica** para definir se a função de mapeamento aprendida (i.e., o **modelo**) é boa ou não.
  - **Exemplo:** dada a previsão da temperatura média para um dia qualquer, quantos picolés serão vendidos?
    - Não existe uma função na natureza que mapeie temperatura em picolés.

# Referências

- [1] Stuart Russell e Peter Norvig, “*Artificial Intelligence: A Modern Approach*,” Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.
- [2] Aurélien Géron, “*Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*”, 1st ed., O'Reilly Media, 2017.
- [3] Levy Boccato, “Notas de aula do curso Tópicos em Sistemas Inteligentes II - Aprendizado de Máquina” (IA006), disponíveis em [https://www.dca.fee.unicamp.br/~lboccato/ia006\\_2s2019.html](https://www.dca.fee.unicamp.br/~lboccato/ia006_2s2019.html) (2019).
- [4] Joseph Misiti, “*Awesome Machine-Learning*,” on-line data base with several free and/or open-source books (<https://github.com/josephmisiti/awesome-machine-learning>).
- [5] C. M. Bishop, “*Pattern Recognition and Machine Learning*,” Springer, 1st ed., 2006.
- [6] Coleção de livros, <https://tinyurl.com/mp64ksye>

# Avisos

- Toda a nossa comunicação será feita via Teams.
- Todas as aulas serão gravadas e os vídeos ficarão disponíveis na pasta "Recordings" dentro de "Arquivos".
- Todo material do curso está disponível no GitHub:
  - [https://github.com/zz4fap/t319\\_aprendizado\\_de\\_maquina](https://github.com/zz4fap/t319_aprendizado_de_maquina)
- As entregas das atividades (laboratórios e quizzes) devem ser feitas através do Teams.
  - Se atentem às datas e horários de entrega das atividades no Teams.

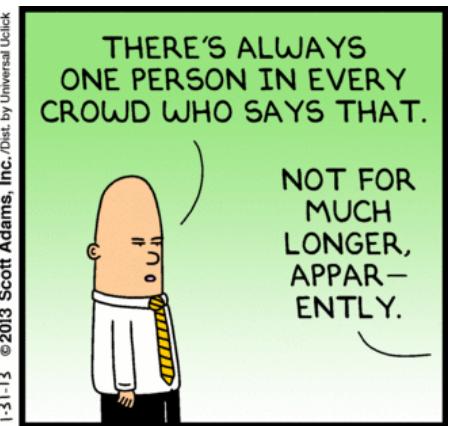
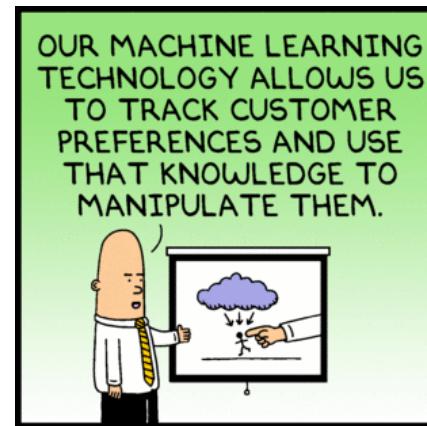
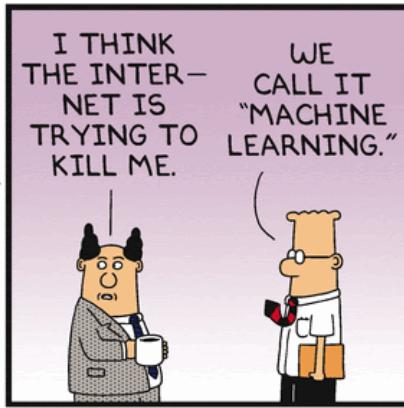
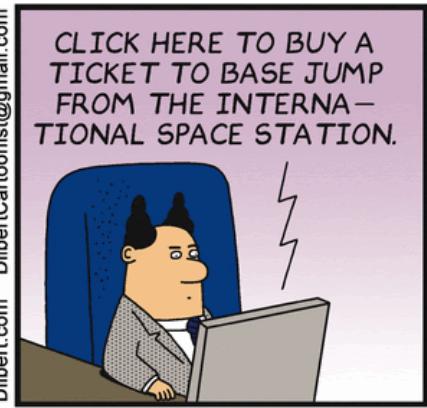
# Avisos

- Vídeos do minicurso curso de Python e de como usar o Colab estão na pasta “*Recordings*” dentro de “Arquivos”.
- Horários de atendimento
  - Professor: sextas-feiras das 18:30 às 19:30.
  - Monitor (Marcus Wilians Gomes Chagas: [marcuswilians@gea.inatel.br](mailto:marcuswilians@gea.inatel.br)): todas as terças-feiras das 17:30 às 19:30.
  - **Atendimento remoto via Teams.**

# Tarefas

- **Quiz:** “*T319 - Quiz - Introdução*” que se encontra no MS Teams.
- **Exercício Prático:** Laboratório #1.
  - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
  - **Vídeo explicando o laboratório #1: Arquivos -> Recordings -> Laboratório #1.**
  - Se atentem aos prazos de entrega.
  - Instruções para resolução e entrega dos laboratórios.

Obrigado!



## **Inteligência Artificial**

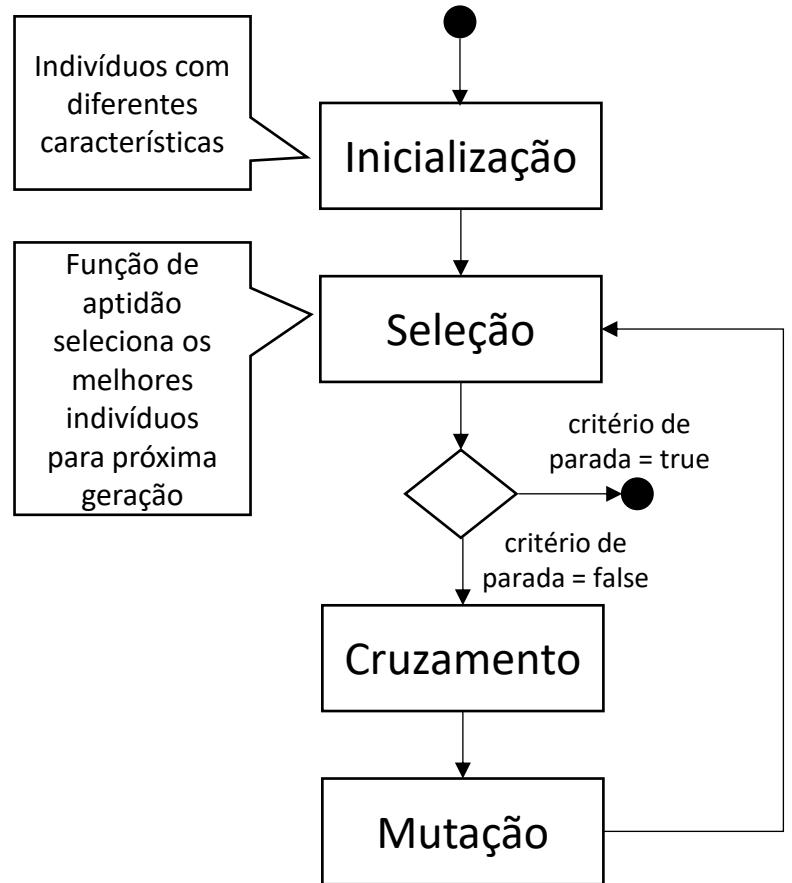
Programas que podem sentir, raciocinar, agir, aprender e se adaptar como humanos

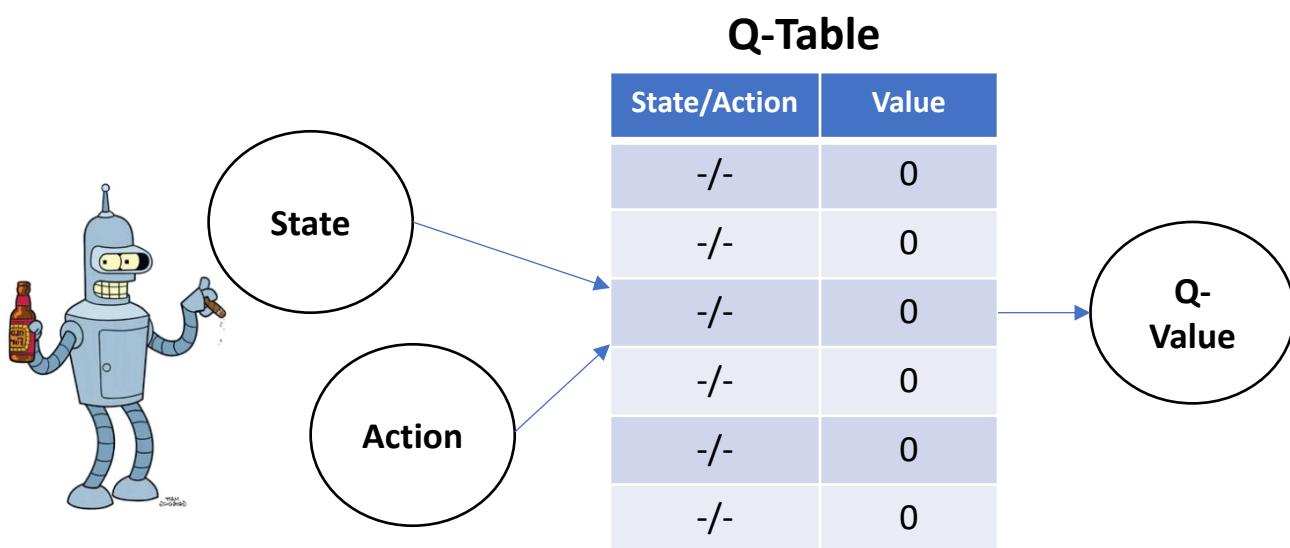
## **Aprendizado de Máquina**

Algoritmos que permitem que uma máquina aprenda automaticamente sem ser explicitamente programada

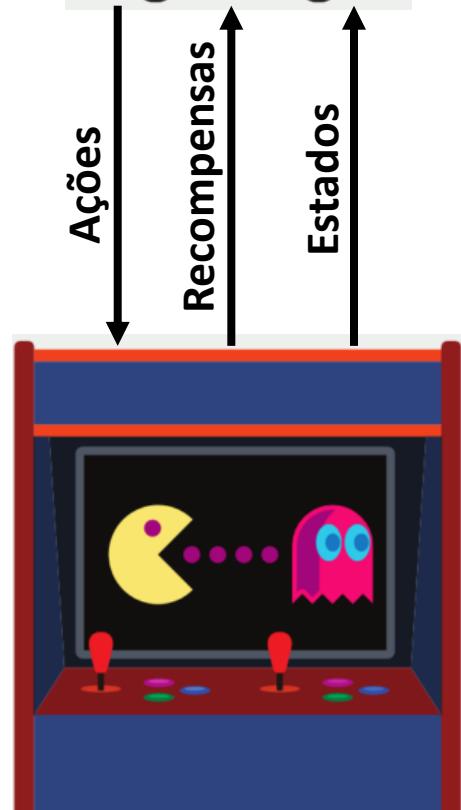
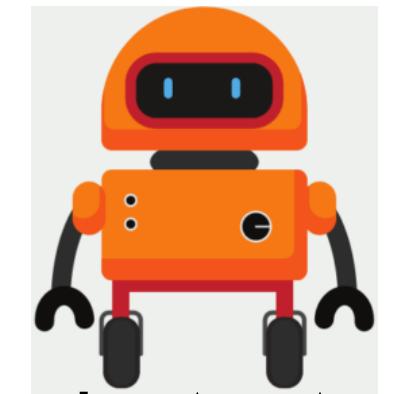
## **Redes Neurais Artificiais**

Multilayer perceptron, Convolutional, Recursive Networks, etc.





**Agente**



**Ambiente**

