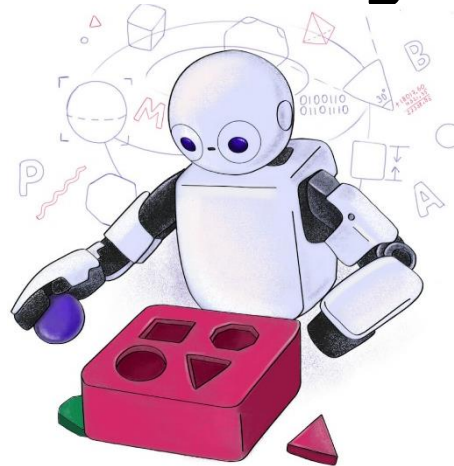


# T319 - Introdução ao Aprendizado de Máquina: *Introdução*



***Inatel***

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

# A disciplina

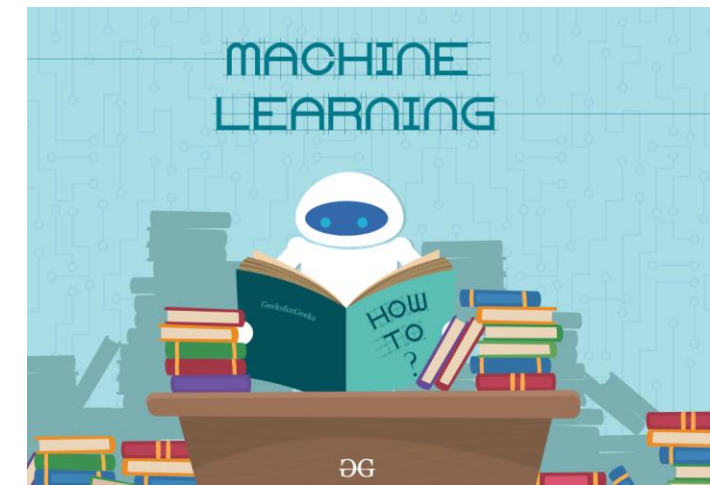
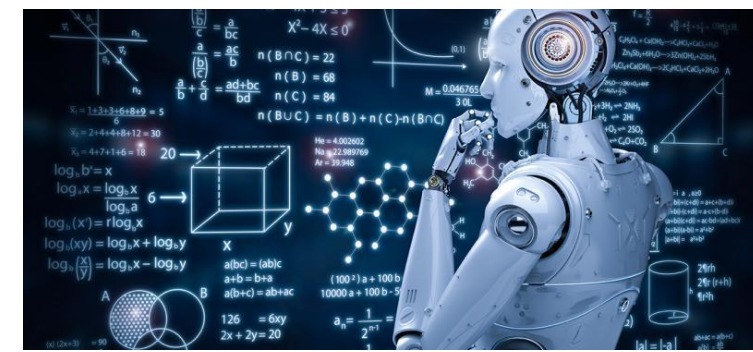
- ***Introdução*** ao aprendizado de máquina.
- Curso introdutório onde veremos os ***conceitos básicos*** de funcionamento de alguns ***algoritmos de aprendizado de máquina*** ou do Inglês, ***machine learning*** (ML).
- O curso será dividido em duas partes: T319 e T320.
- O curso terá sempre uma parte ***expositiva*** e outra ***prática*** para fixação dos conceitos introduzidos.
  - Quizzes e exercícios envolvendo o uso dos algoritmos discutidos.
- Não nos aprofundaremos nos conceitos matemáticos envolvidos.
- Porém, precisamos conhecer Python e alguns conceitos de cálculo, álgebra linear e estatística.

# Cronograma

Aula	Data	Dia	Horário	Atividade
1	5/8/2022	Sexta-feira	21:30 às 23:10	Introdução ao Aprendizado de Máquina
2	19/8/2022			Introdução ao Aprendizado de Máquina
3	2/9/2022			Introdução ao Aprendizado de Máquina
4	16/9/2022			Introdução ao Aprendizado de Máquina
5	30/9/2022			Introdução ao Aprendizado de Máquina
6	14/10/2022			Introdução ao Aprendizado de Máquina
7	28/10/2022			Introdução ao Aprendizado de Máquina
8	11/11/2022			<b>Avaliação Presencial</b>
9	25/11/2022			Introdução ao Aprendizado de Máquina
10	9/12/2022			Introdução ao Aprendizado de Máquina

# Objetivo do curso

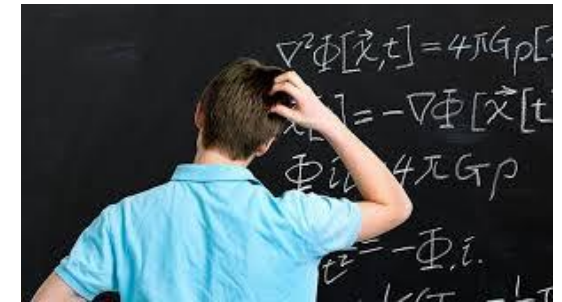
- O objetivo principal do curso é apresentar
  - os conceitos fundamentais da teoria do aprendizado de máquina.
  - um conjunto de ferramentas (ou seja, algoritmos) de aprendizado de máquina para solução de problemas.
- Ao final do curso vocês devem ser capazes de
  - Entender e discutir sobre os principais algoritmos de ML.
  - Compreender a terminologia utilizada na área.
  - Aplicar algoritmos de ML para a resolução de problemas.
  - Analisar e entender novos algoritmos de ML.
  - Criar seus próprios projetos.



# Critérios de avaliação



- Um (1) trabalho com peso de 85%.
  - Envolvendo questões teóricas e/ou práticas.
  - Uma parte do trabalho será feita **presencialmente**.
- Atividades (quizzes e laboratórios) com peso de 15%.
  - Podem sempre ser entregues até a próxima aula.
  - Laboratórios podem ser resolvidos em grupo, mas entregas devem ser individuais.
  - Exercícios serão atribuídos através do MS Teams.
- **Frequência**
  - Gerada automaticamente pelo MS Teams.
  - Por favor, acompanhem a frequência através do portal.



# Motivação

- **Emprego:** grandes companhias usam IA em seus produtos e/ou soluções internas para resolver os mais diversos tipos de problemas e assim aumentarem sua eficiência e consequentemente os lucros.



- **Pesquisa:** já se prevê que IA terá um papel importante no desenvolvimento da próxima geração de redes móveis e sem-fio (e.g., 6G).

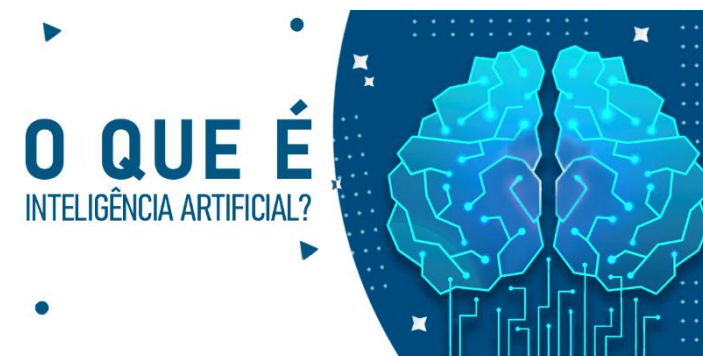




# Inteligência Artificial



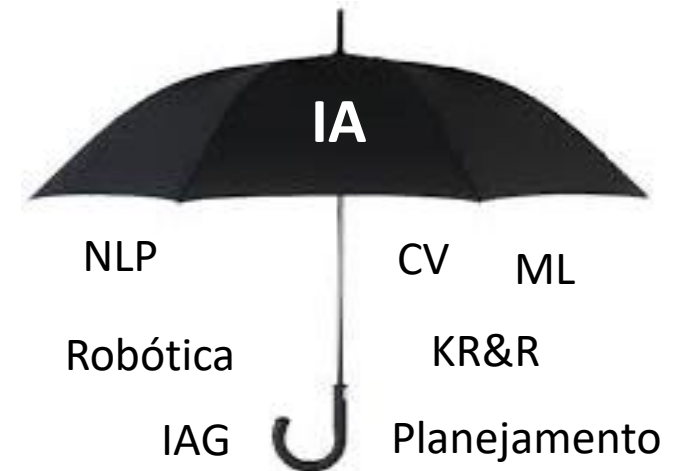
- **Definição:** Capacidade de uma máquina de interpretar corretamente estímulos vindos do ambiente, aprender com eles e usar o conhecimento adquirido para realizar tarefas.
- **Objetivo:** Criar máquinas que *imitem a inteligência humana* para realizar tarefas e que se *aprimoram com base nas informações que coletam*.
- Porém, ensinar as máquinas a pensar não é uma tarefa tão simples.



# Inteligência Artificial

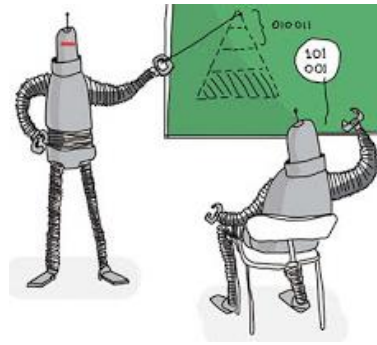
- Para conseguirmos criar uma máquina inteligente, precisamos antes dividir o problema em problemas menores (subáreas):
  - Processamento de linguagem natural.
    - ✓ Geração e compreensão automática de linguagens naturais.
  - Representação do conhecimento.
    - ✓ Criação e armazenamento de conhecimento do mundo real.
  - Raciocínio automatizado.
    - ✓ Resolução de problemas complexos a partir de conhecimento adquirido.
  - Planejamento.
    - ✓ Criação de planos que permitam que uma máquina execute uma tarefa.
  - Visão computacional.
    - ✓ Extração de informações de imagens e vídeos.
  - Robótica.
    - ✓ Projeto, construção e operação de robôs que repliquem ações humanas.
  - Aprendizado de máquina.
    - ✓ Criação de máquinas que aprendem através de exemplos.
  - Inteligência artificial geral.
    - ✓ Criação de máquinas que solucionem qualquer tipo de problema. É a meta final da IA.

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





# Foco do curso



- Como vimos, IA é um área muito ampla, e, portanto, focaremos no estudo de algoritmos de ***Aprendizado de Máquina*** (do inglês, *Machine Learning* - ML).
- **Por quê?**
  - ***Caixa de ferramentas***: ML oferece ferramentas importantes para a análise e solução eficiente de vários problemas em várias áreas.
  - ***Redução de complexidade e custo***: vários procedimentos e processos em várias áreas que apresentam desempenho ótimo na teoria não são utilizados na prática, pois possuem complexidade computacional e/ou custo proibitivos.
  - ***Oportunidades***: existem muitos empregos na área de análise, ciência e engenharia de dados, além de pesquisas inovadoras para a solução de problemas com ML.

# Mas então, o que é ML?



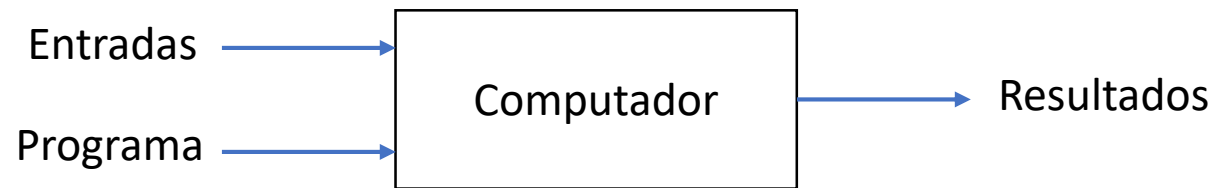
- É uma das subáreas da inteligência artificial.
- O termo foi cunhado em 1959, pelo cientista da computação Arthur Samuel, que o definiu como o  
*“Campo de estudo que dá aos computadores a habilidade de **aprender sem serem explicitamente programados**”.*
- Algoritmos de ML são **orientados a dados**, ou seja, eles **aprendem automaticamente** uma **solução geral** a partir de **conjuntos de dados** fornecidos a eles.



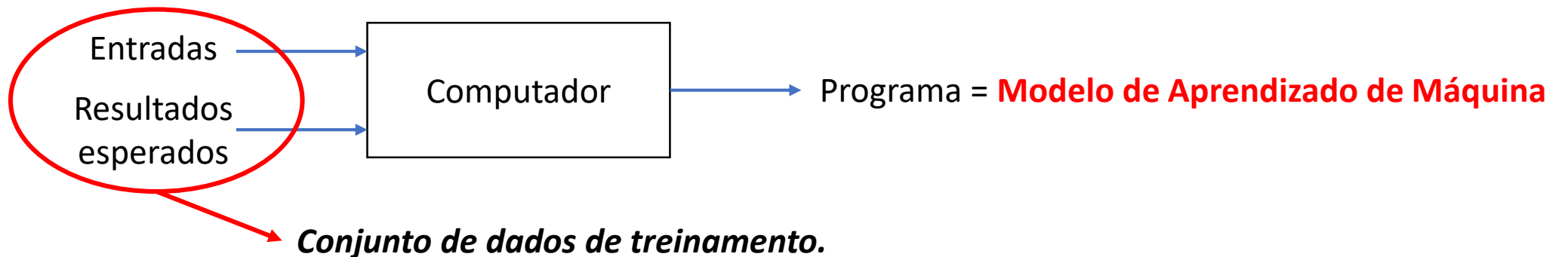
# O que é o Aprendizado de Máquina?

- “... *aprender sem serem explicitamente programados.*”

## Programação Tradicional

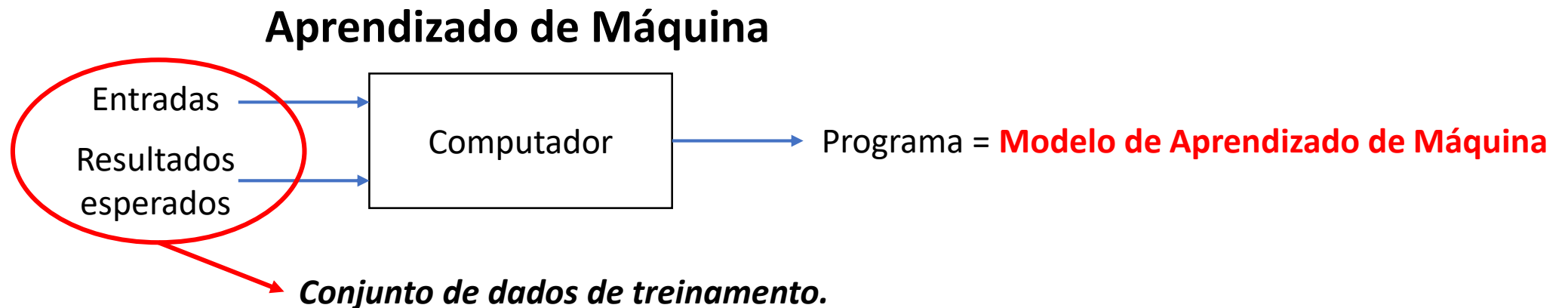


## Aprendizado de Máquina

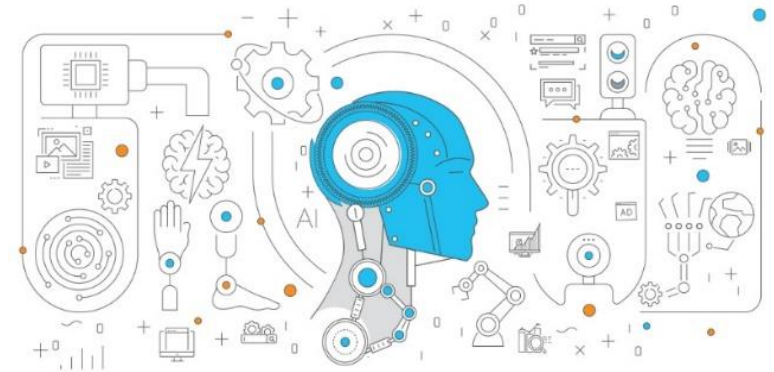


# O que é o Aprendizado de Máquina?

- Através de **treinamento** com um **conjunto de dados** (entradas e saídas esperadas), o **algoritmo** de ML **aprende** um **modelo** que **reproduz os resultados esperados** e, o mais importante, **generaliza** para **entradas não vistas durante o treinamento**.



# Exemplos de aplicações de ML



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



# Principais motivos da difusão do ML

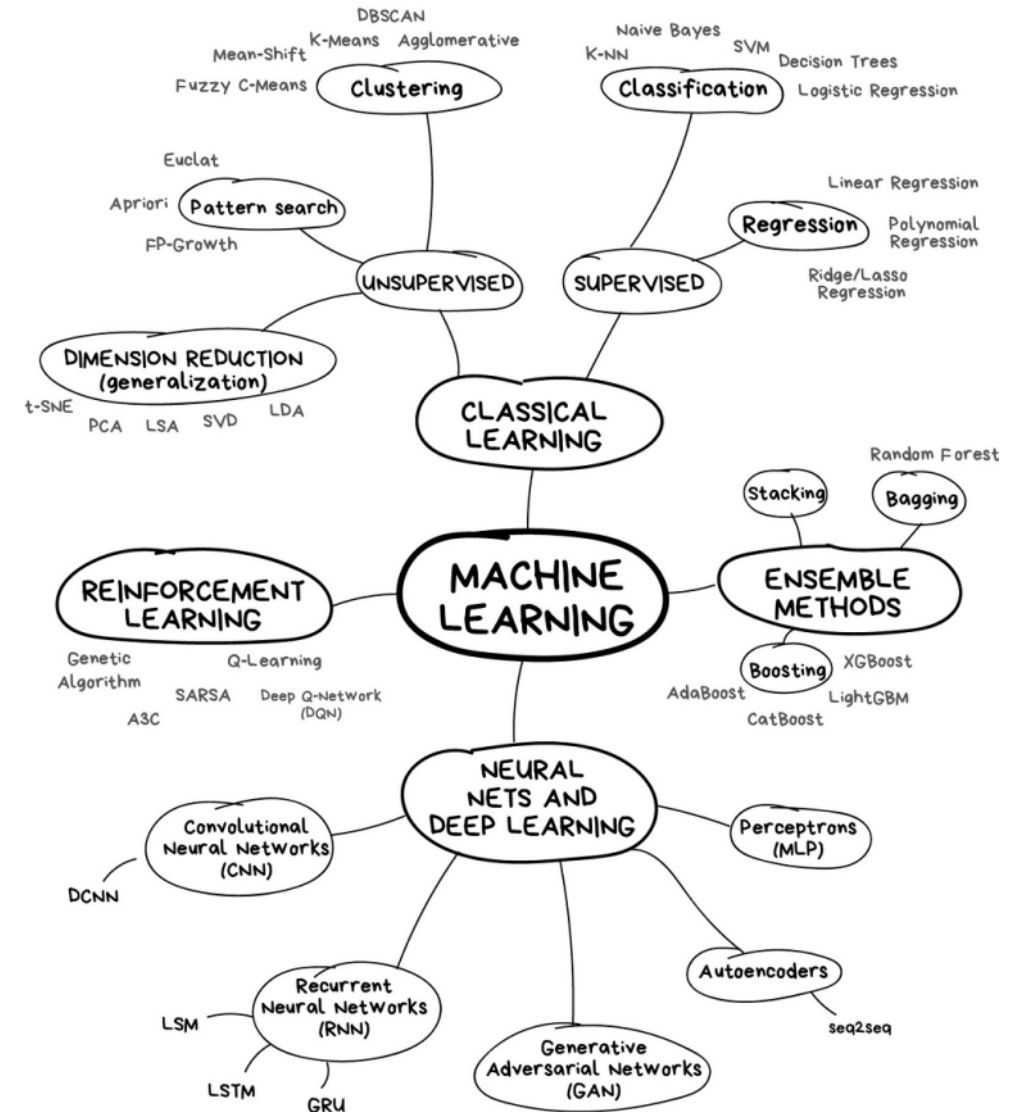
- Possibilidade de analisar e extrair informações úteis de enormes volumes dados (de tera a petabytes) disponíveis atualmente, o que seria impossível para nós.
- A extração de informações úteis a partir de dados vale ouro, pois têm grande potencial para aumentar o lucro das empresas.
- O surgimento de recursos computacionais poderosos tais como GPUs, FPGAs e CPUs com múltiplos cores.
- Surgimento de novas e eficientes estratégias de aprendizagem, e.g., *deep-learning*, *deep reinforment-learning*, *generative adversarial learning*, etc.
- Criação de *frameworks* e bibliotecas poderosas que facilitam o desenvolvimento de soluções com ML.



# Tipos de Aprendizado de Máquina

Dependendo do tipo de aprendizado realizado pelos algoritmos, eles podem ser agrupados da seguinte forma:

- Supervisionado
- Não-Supervisionado
- Semi-Supervisionado
- Por Reforço
- Metaheurístico





# Aprendizado Supervisionado

- No aprendizado supervisionado, o algoritmo tem acesso às saídas esperadas,  $y$ , chamadas de **rótulos** (ou *labels*, do 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,  $x$ , e sua saída correspondente,  $y$ .
- **Objetivo:** os algoritmos supervisionados de ML devem **aprender** uma **função** 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: experiência vs. salário.
  - **Classificação:** o rótulo,  $y$ , pertence a um **conjunto finito** de valores, i.e., conjunto finito de classes. Exemplo: filtro de spam.



# Aprendizado Não-Supervisionado

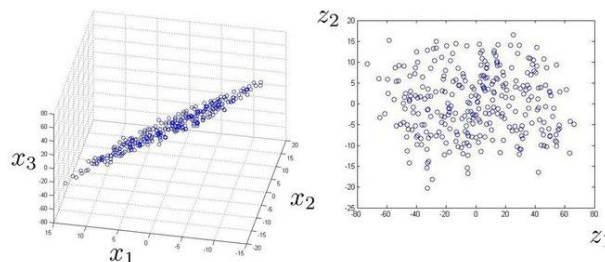
- Neste tipo de aprendizado, os algoritmos não têm acesso às saídas esperadas. Eles só recebem os atributos,  $x$ .
- Neste caso, os algoritmos ***aprendem/descobrem padrões*** (muitas vezes ocultos) presentes nos dados de entrada ***sem a presença de rótulos***.
- **Objetivo:** os algoritmos devem ***aprender/descobrir*** padrões (e.g., similaridades) desconhecidos se baseando apenas nos exemplos de entrada.
- Trata problemas de clusterização, redução de dimensionalidade, detecção de anomalias (*outliers*) e aprendizado de regras de associação.

Clusterização

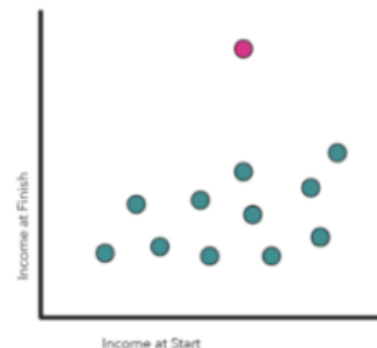


Redução de dimensionalidade

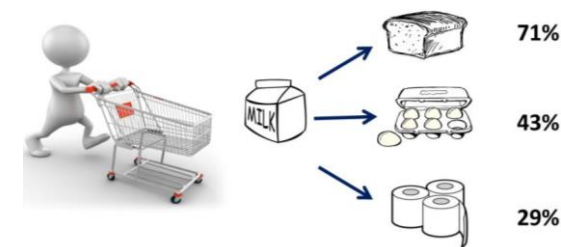
Reduce data from 3D to 2D



Detecção de Anomalias



Regras de associação



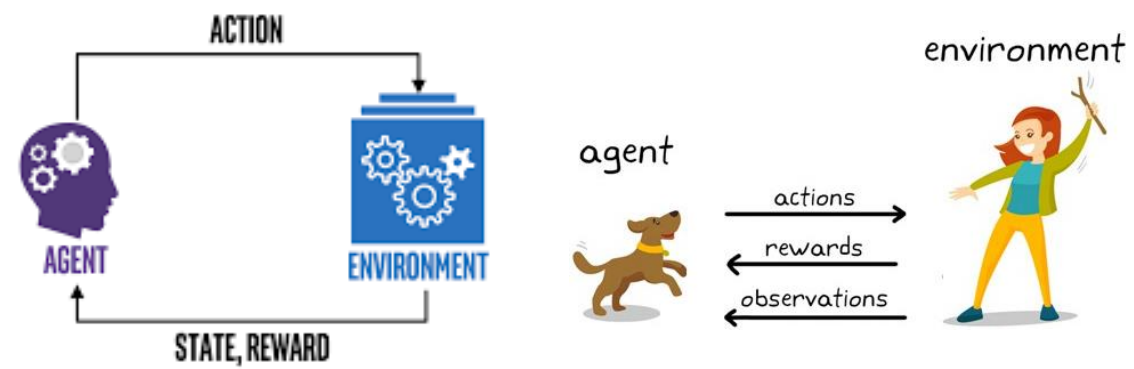
Of transactions that included milk:

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

# Aprendizado Semi-Supervisionado

- Neste tipo de aprendizado, os algoritmos têm acesso a exemplos de treinamento com e sem rótulos.
- Geralmente envolve uma ***pequena quantidade de dados*** rotulados e uma ***grande quantidade de dados não-rotulados***.
- É de grande ajuda em casos onde se ter uma grande quantidade de dados rotulados é muito demorado, caro ou complexo.
- Algoritmos de aprendizagem semi-supervisionada são o resultado da combinação de algoritmos supervisionados e não-supervisionados.
- Uma maneira de realizar aprendizado semi-supervisionado é combinar, por exemplo, algoritmos de ***clustering*** e ***classificação***.

# Aprendizado Por Reforço



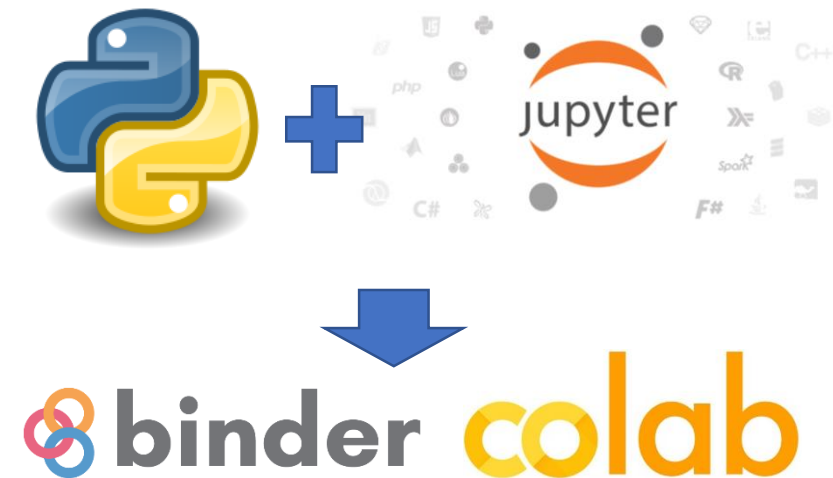
- Abordagem de aprendizado totalmente diferente das anteriores pois ***não temos exemplos de treinamento***, sejam eles rotulados ou não.
- O algoritmo de aprendizado por reforço, chamado de ***agente*** nesse contexto, aprende como se comportar em um ***ambiente*** através de interações do tipo ***tentativa e erro***.
- O ***agente*** observa o ***estado*** do ***ambiente*** em que está inserido, seleciona e executa ***ações*** e recebe uma ***recompensa*** (ou ***reforço***) em consequência das ***ações*** tomadas.
- Seguindo estes passos, o agente deve aprender 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 estiver em uma determinada situação, ou seja, em um ***estado*** do ***ambiente***.
- Portanto, a ***política*** é uma ***função*** que mapeia os ***estados*** do ***ambiente*** em ***ações*** que o ***agente*** deve tomar.

# Aprendizado Metaheurístico

- Uma **metaheurística** é um método **heurístico** usado para resolver de forma **genérica** problemas de otimização.
  - **Heurística** é um método ou processo criado com o objetivo de encontrar soluções, de **forma rápida**, mas muitas vezes **sub-ótimas**, para problemas complexos.
- **Metaheurísticas** são geralmente aplicadas a problemas para os quais não se conhece um algoritmo eficiente ou não se tem uma solução conhecida.
- Características das metaheurísticas:
  - não garantem que uma solução globalmente ótima seja encontrada, mas podem encontrar uma solução suficientemente boa (sub-ótima).
  - são estratégias que orientam o processo de busca através do espaço de soluções.
  - não são específicas do problema, ou seja, são genéricas.
  - funcionam bem mesmo com capacidade de computação limitada.
- São algoritmos inspirados pelo processo de seleção natural (algoritmo genético) ou no comportamento de animais (otimização da colônia de formigas).

# Executando os códigos

- Durante o curso, usaremos **Python** como linguagem de programação.
- Utilizaremos **notebooks Jupyter** para execução de exemplos e resolução dos exercícios práticos.
  - Eles são **documentos virtuais** usados para criar e documentar código.
  - Pode-se adicionar equações, gráficos e texto, além de código.
- Para executá-los, utilizaremos o **Google Colaboratory** ou o **Binder**, que são ambientes computacionais (i.e., servidores) interativos e gratuitos.
- Portanto, **vocês não precisam instalar nada**, apenas terem um navegador web e conexão com a internet.



# Google Colaboratory (Colab)



- **Colab**: aplicação web gratuita que permite a criação e edição de *notebooks Jupyter* em navegadores web.
- É um produto da Google.
- Vantagens:
  - Grande número de servidores.
  - Rápida inicialização e processamento do código.
  - Fornece acesso a GPUs e TPUs gratuitamente.
  - Notebooks podem ser salvos no seu Google Drive, evitando que você perca seu código.
- Desvantagem
  - Por hora, suporta apenas a execução de códigos escritos em Python.
- URL: <https://colab.research.google.com/>



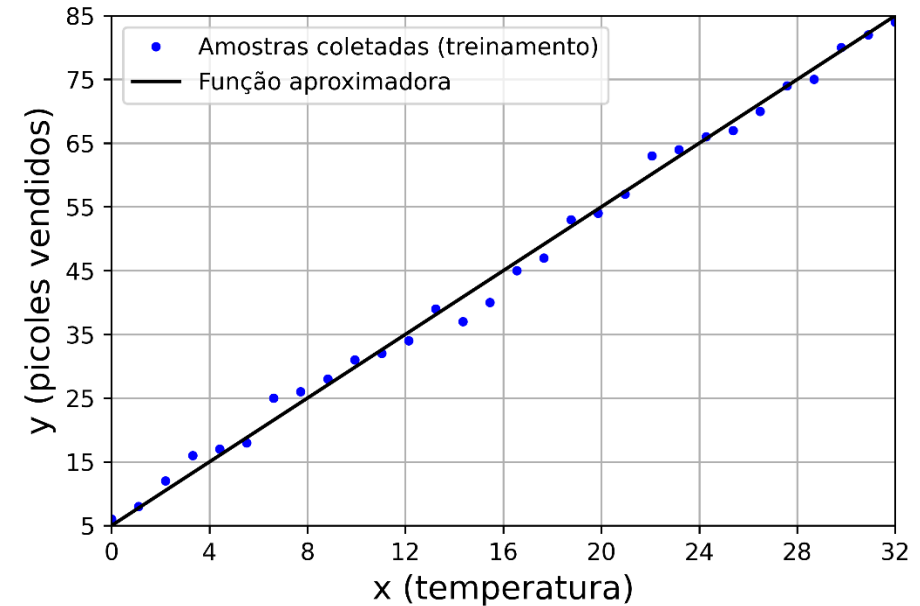
# Binder



- **Binder**: outra aplicação web gratuita que permite a criação e edição de ***notebooks Jupyter*** em navegadores web.
- Vantagem:
  - Suporta a execução de várias linguagens de programação: Python, C++, C#, PHP, Julia, R, etc.
- Desvantagens:
  - Poucos servidores disponíveis.
  - Não é possível salvar os notebooks (e.g., Google Drive).
  - Depois de algum tempo inativo, a máquina virtual executando seu ***notebook*** se desconecta e você pode perder seu código.
- URL (através do Jupyter): <https://jupyter.org/>

# Objetivo

- O objetivo desta primeira parte do curso é encontrar uma **função**, usando aprendizado de máquina, que **aproxime** o comportamento de um **conjunto de amostras** ( $x$  e  $y$ ) da **melhor forma possível**.
- Na maioria dos casos, não conhecemos o mapeamento verdadeiro (muitas vezes ele nem existe) entre  $x$  e  $y$  e nos baseamos apenas em uma métrica para definir se a aproximação é boa.



**Exemplo:** dada a previsão da temperatura para um dia qualquer, quantos picolés serão vendidos?

# Referências

- [1] Stuart Russell and 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] Joseph Misiti, *“Awesome Machine-Learning,”* on-line data base with several free and/or open-source books (<https://github.com/josephmisiti/awesome-machine-learning>).
- [4] Andriy Burkov, *“The Hundred-Page Machine-Learning Book,”* Andriy Burkov 2019.
- [5] C. M. Bishop, *“Pattern Recognition and Machine Learning,”* Springer, 1st ed., 2006.
- [6] S. Haykin, *“Neural Networks and Learning Machines,”* Prentice Hall, 3ª ed., 2008.
- [7] Coleção de livros,  
<https://drive.google.com/drive/folders/1lylIMu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing>

# Avisos

- Entregas de exercícios (laboratórios e quizzes) devem ser feitas através do MS Teams.
  - Se atentem às datas/horários de entrega no MS Teams.
- Todo material do curso será disponibilizado no MS Teams e no GitHub:
  - <https://github.com/zz4fap/t319> aprendizado de maquina
- Horários de Atendimento
  - Professor: quintas-feiras das 18:00 às 19:00 e sextas-feiras das 16:00 às 17:00.
  - Monitor (Maycol Teles: **maycol.teles@ges.inatel.br**): quartas-feiras das 18:30 às 19:30 .
  - Atendimento remoto via MS Teams.

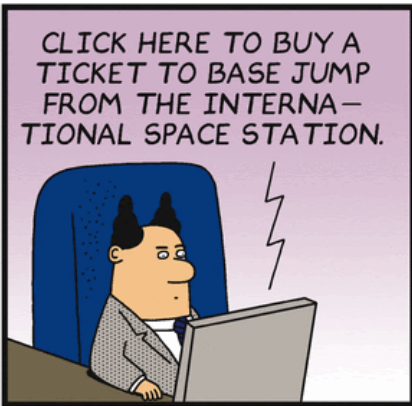
# Tarefas

- **Quiz:** “*T319 - Quiz - Introdução*” que se encontra no MS Teams.
- **Exercício Prático:** [Laboratório #1](#).
  - Pode ser baixado do MS Teams ou do GitHub.
  - Pode ser respondido através do link acima (na nuvem) ou localmente.
  - [Instruções para resolução e entrega dos laboratórios](#).
  - **Laboratórios podem ser feitos em grupo, mas as entregas devem ser individuais.**

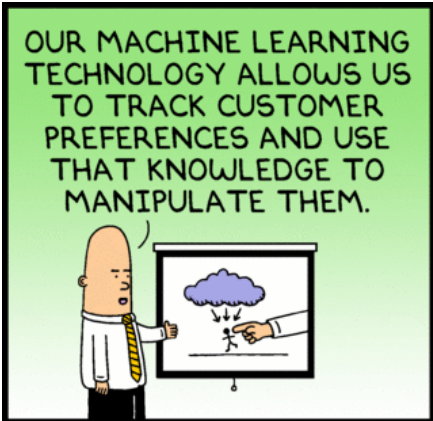
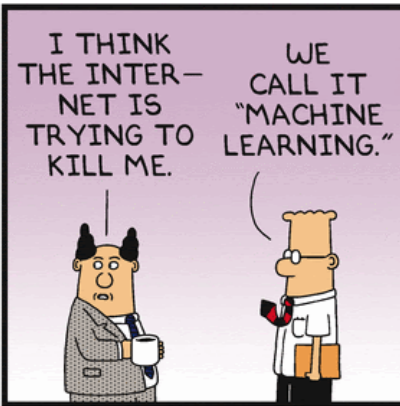
Obrigado!



Dilbert.com DilbertCartoonist@gmail.com



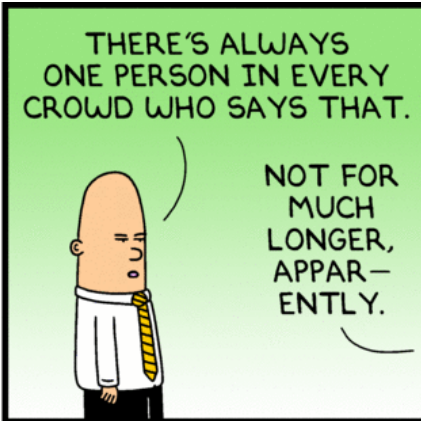
2-2-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick



Dilbert.com DilbertCartoonist@gmail.com



1-31-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick



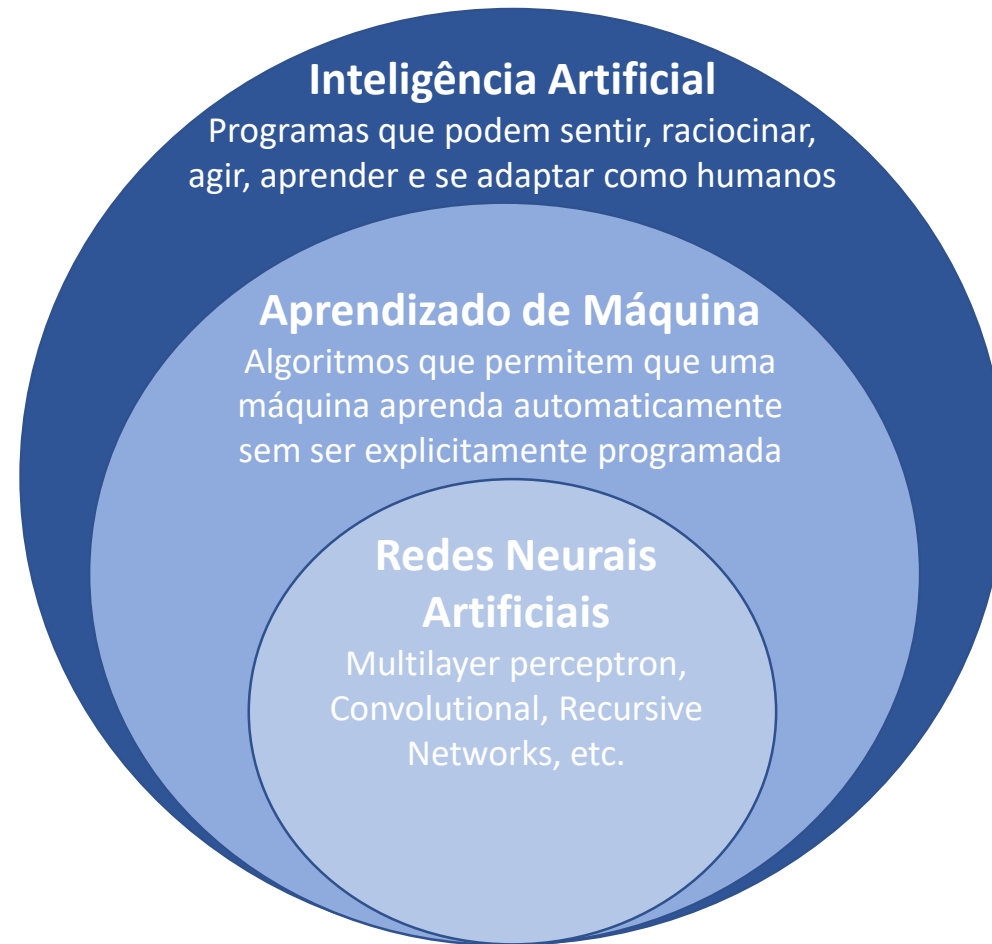
www.dilbert.com scottadams@aol.com

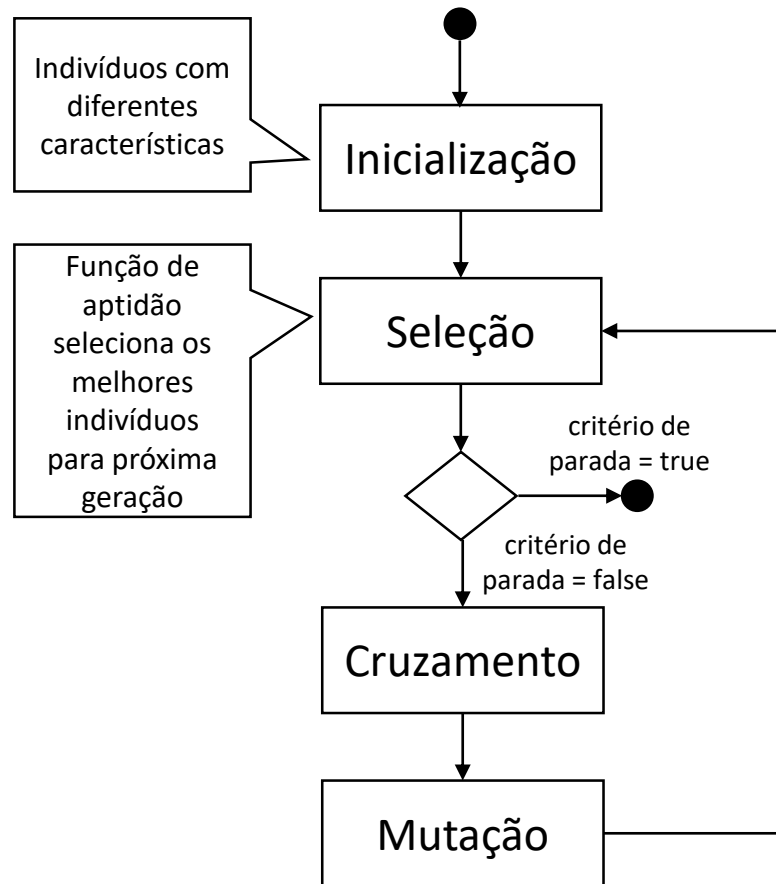


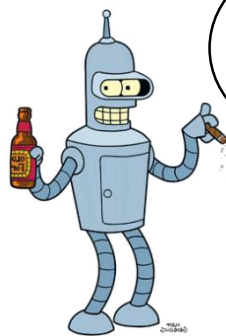
©2003 United Feature Syndicate, Inc.











State

Action

Q-Table

State/Action	Value
-/-	0
-/-	0
-/-	0
-/-	0
-/-	0
-/-	0

Q-Value