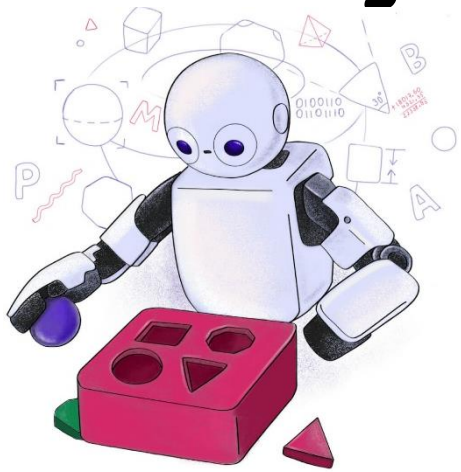
T319 - Introdução ao Aprendizado de Máquina: *Introdução*



Inatel

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

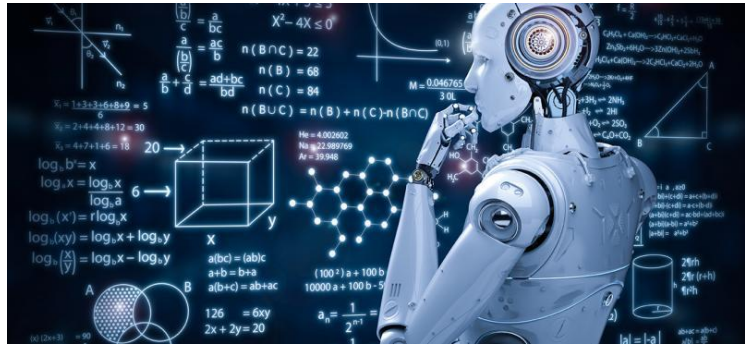
A disciplina

- ***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 nós aprofundaremos nos conceitos matemáticos envolvidos.
- Porém, precisamos conhecer Python e alguns conceitos de cálculo, álgebra linear e estatística.

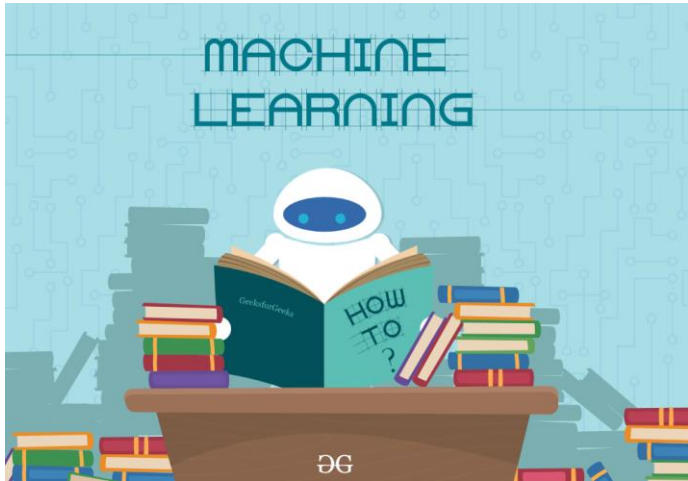
Cronograma

Aula	Data	Dia	Horário	Atividade
1	18/02/2022	Sexta-feira	21:30 às 23:10	Introdução ao Aprendizado de Máquina
2	4/3/2022			Introdução ao Aprendizado de Máquina
3	18/3/2022			Introdução ao Aprendizado de Máquina
4	1/4/2022			Introdução ao Aprendizado de Máquina
5	15/4/2022			Introdução ao Aprendizado de Máquina
6	29/4/2022			Introdução ao Aprendizado de Máquina
7	13/5/2022			Introdução ao Aprendizado de Máquina
8	27/5/2022			Introdução ao Aprendizado de Máquina
9	10/6/2022			Introdução ao Aprendizado de Máquina
10	24/6/2022			Projeto Final

Objetivo do curso

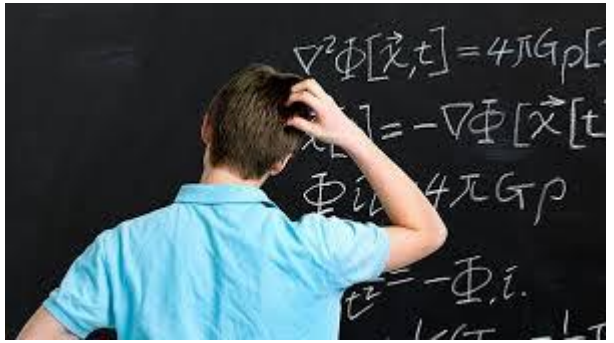


- O objetivo principal do curso é apresentar
  - os conceitos fundamentais da teoria do aprendizado de máquina.
  - um conjunto de ferramentas (ou seja, algoritmos) de aprendizado de máquina para solução de problemas.
- Ao final do curso vocês devem ser capazes de
  - Entender e discutir sobre os principais 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

- 1 trabalho com peso de 85%.
  - Envolvendo questões teóricas e/ou práticas.
- Atividades (quizzes e laboratórios) com peso de 15%.
  - Podem sempre ser entregues até a próxima aula.
  - Podem ser resolvidos em grupo, mas entregas devem ser individuais.
  - Exercícios serão atribuídos através de tarefas do MS Teams.
- **Frequência**
  - Gerada automaticamente pelo MS Teams.
  - Por favor, acompanhem suas presenças no portal.



Motivação

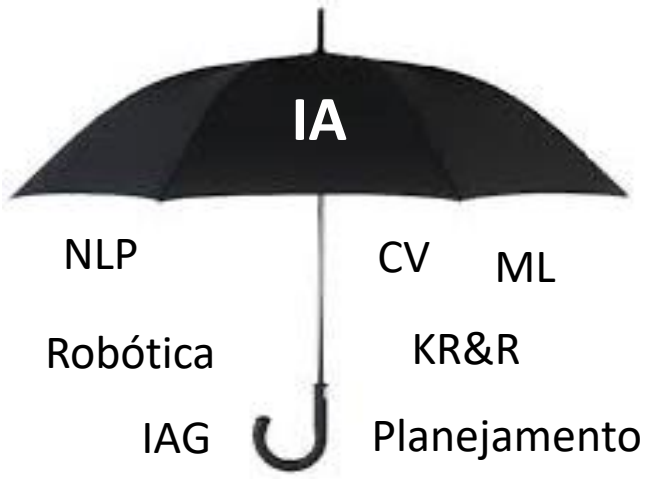
- **Emprego**: grandes companhias usam ML em seus produtos e/ou soluções internas para resolver os mais diversos tipos de problemas e assim aumentarem sua eficiência e consequentemente os lucros.


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

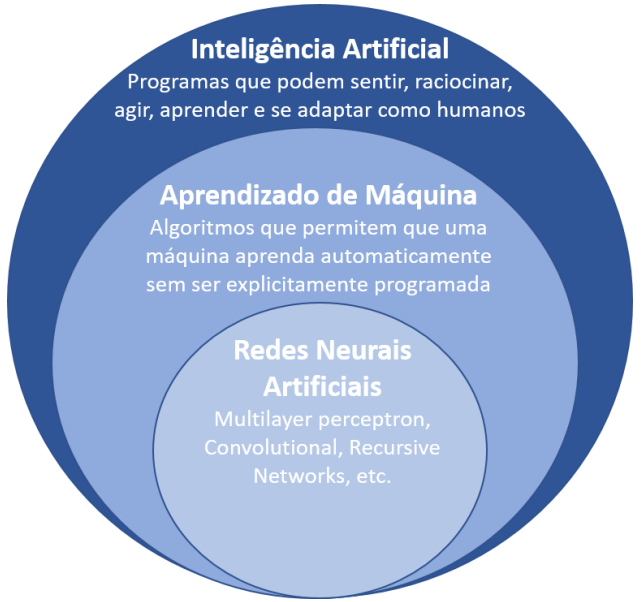
Definição e objetivo da Inteligência Artificial

- **Definição**: "*Capacidade de um sistema de interpretar corretamente dados externos (estímulos vindos do ambiente), aprender com esses dados e usá-los para atingir tarefas e objetivos específicos por meio de adaptação flexível*." (*Andreas Kaplan*).
- **Objetivo**: Criar máquinas que ***imitem*** nossas ***habilidades mentais***, ou seja, criar máquinas que são ***modelos aproximados*** de nossas habilidades de enxergar, falar, ouvir, sentir, aprender, raciocinar, resolver problemas, etc.
- IA utiliza a ***experiência*** para adquirir ***conhecimento*** e também como aplicar esse ***conhecimento*** a problemas desconhecidos.

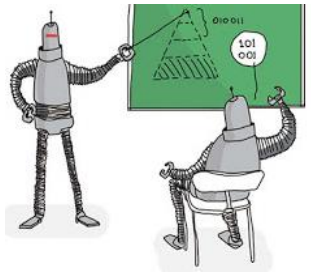
Inteligência Artificial

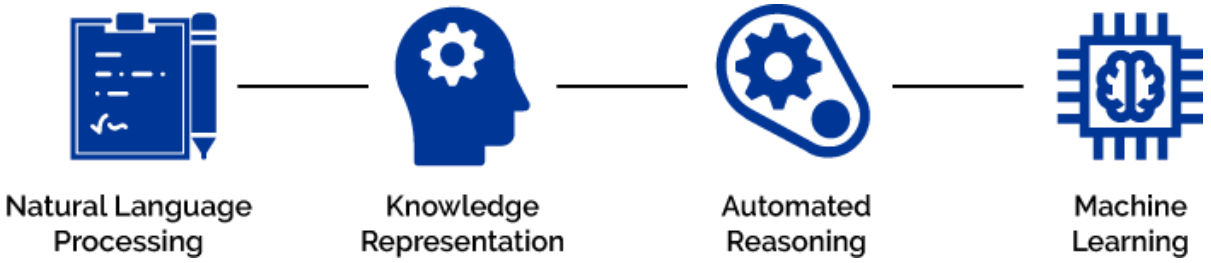


- IA é uma área muito ampla que ***engloba*** várias aplicações (ou sub-áreas) tais como
  - Processamento de linguagem natural.
    - ✓ Geração e compreensão automática de línguas humanas naturais.
  - Representação do conhecimento.
    - ✓ Criação e armazenamento de conhecimento do mundo real.
  - 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.



Foco do curso





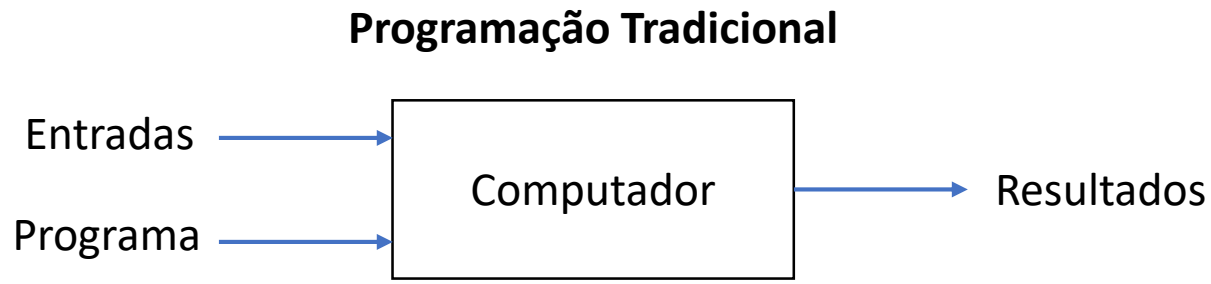
- Como vimos, IA é um termo muito amplo, abrangendo várias sub-áreas, usado para designar máquinas capazes executar ***tarefas*** de forma inteligente.
- **Foco do curso**: estudo dos principais algoritmos de ***Aprendizado de Máquina***.
- **Por quê?**
  - ***Caixa de ferramentas***: ML oferece ferramentas 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.

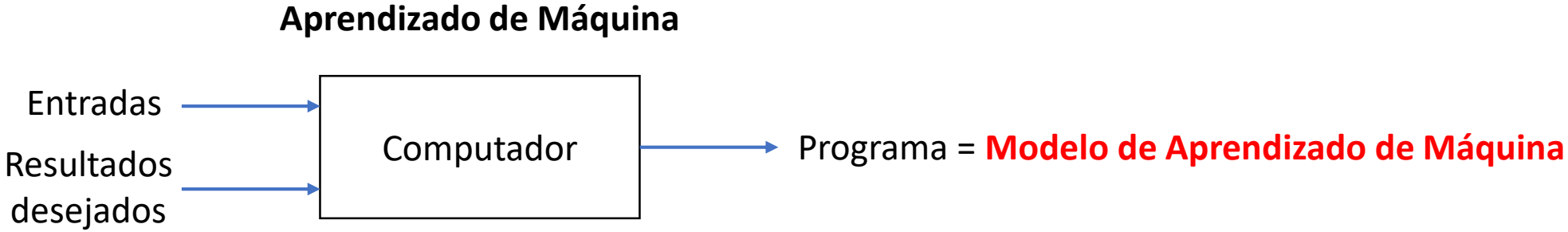
Mas então, o que é ML?

- É uma sub-área 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***".
- Uma outra definição interessante feita por Jojo John Moolayil é "*Aprendizado de máquina é o processo de* ***induzir*** *inteligência em uma máquina sem que ela seja explicitamente programada*".
  - **Indução**: aprender um modelo ou padrão geral ***a partir de exemplos corretos e incorretos***.
- Portanto, podemos dizer que algoritmos de ML são ***orientados a dados***, ou seja, eles aprendem automaticamente um ***padrão geral*** a partir de ***grandes volumes de dados***.

O que é o Aprendizado de Máquina?

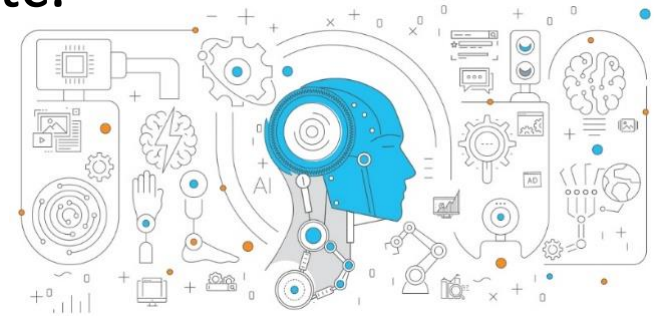
- *"... **aprender sem serem explicitamente programados**."*





Exemplos de aplicações de ML

- **Transporte**: veículos autônomos, predição do tráfego, etc.
- **Negócios**: recomendação de produtos e conteúdos (e.g., amazon e netflix), chatbots para atendimento ao cliente, etc.
- **Educação**: pontuação automatizada de fala em testes de Inglês, correção de provas, etc.
- **Clima**: previsão do tempo (temperatura, chuva, furacões, etc.).
- **Medicina**: detecção e/ou predição de doenças (câncer, Alzheimer, pneumonia, COVID-19, etc.), descoberta de novas drogas, etc.
- **Finanças**: detecção de fraudes com cartão de crédito, etc.
- **Tecnologia**: filtros anti-spam, assistentes pessoais (e.g., *Siri*, *Alexa*, etc.), tradutores em tempo-real.



Por que ML se tornou tão difundido?

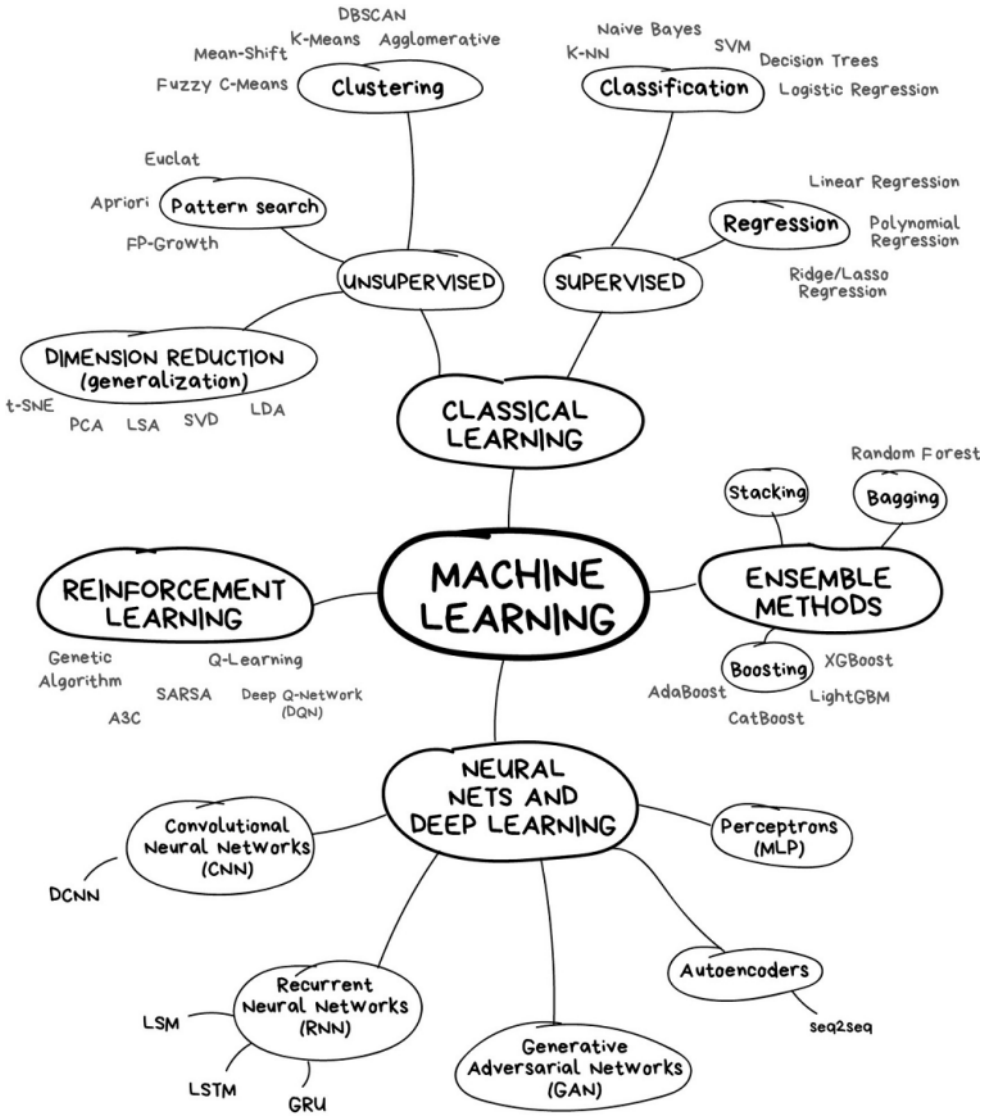
Alguns dos principais motivos são:

- Vivemos na era da informação e, portanto, um volume sem precedentes de dados (de tera a petabytes) está disponível, impossibilitando sua análise por nós seres humanos.
- Porém, para modelos de ML isso não é um problema e sim uma solução, pois quanto mais dados melhor será o aprendizado.
- Além disso, a extração de informações úteis a partir dos dados gerados por empresas vale ouro, pois têm grande potencial para aumentar seus lucros.
- 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*, etc.
- Existência de frameworks e bibliotecas poderosas que facilitam o desenvolvimento de soluções com ML.

Tipos de Aprendizado de Máquina

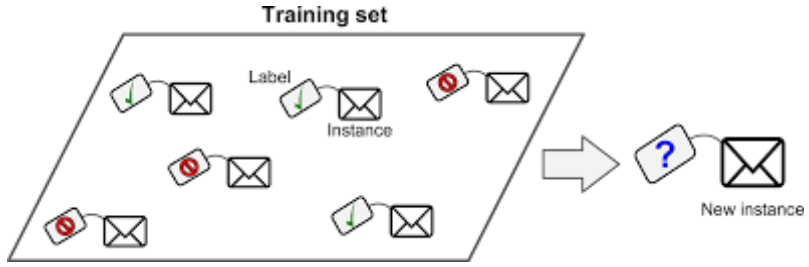
Dependendo do tipo de aprendizado realizado pelos algoritmos, eles podem ser agrupados da seguinte forma:

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

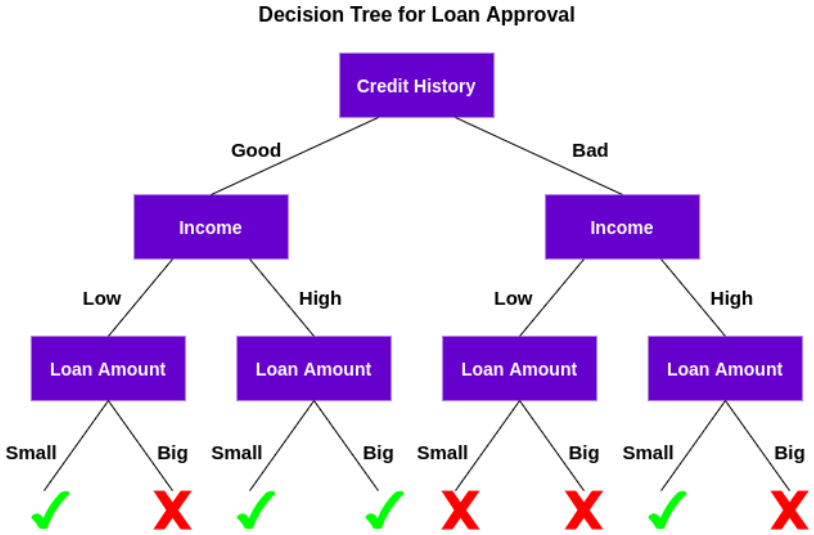


Aprendizado Supervisionado

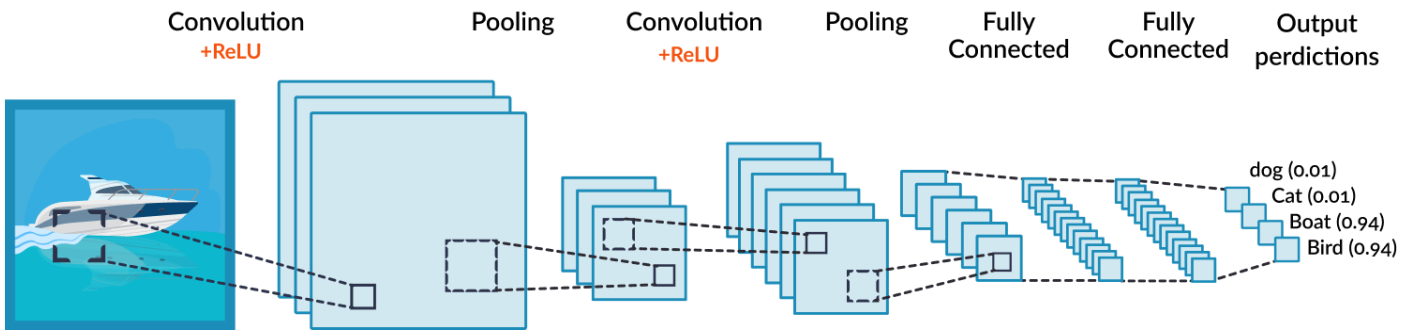
- No aprendizado supervisionado a máquina sabe o que aprender, ou seja, ela tem acesso às respostas esperadas.
- Neste tipo de aprendizado, os dados ou exemplos de treinamento incluem os ***atributos***, $\boldsymbol{x}$, que são a entrada do algoritmo de ML e as ***respostas esperadas***, y, para cada entrada, chamadas de ***rótulos*** (ou *labels*, do Inglês).
- **Objetivo**: os modelos supervisionados de ML devem ***aprender*** uma função que mapeie as entradas $\boldsymbol{x}$ nas saídas y, ou seja, $y = f(\boldsymbol{x})$.
- 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.
  - **Classificação**: o rótulo, y, pertence a um ***conjunto finito*** de valores, i.e., conjunto finito de classes.

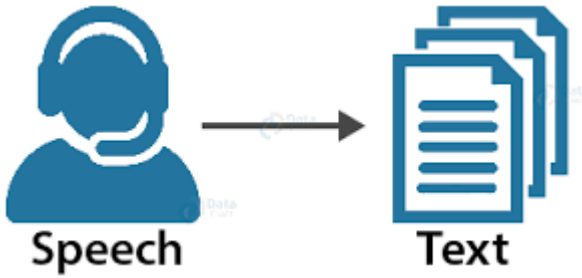


Principais Algoritmos para Aprendizado Supervisionado



- Regressão linear.
- Regressão logística.
- Arvores de Decisão (*Decision Trees*).
- Florestas Aleatórias (*Random Forests*).
- k vizinhos mais próximos (*k-nearest neighbors* - k-NN).
- Máquinas de Vetores de Suporte (*Support Vector Machines* - SVMs).
- Redes Neurais Artificiais.

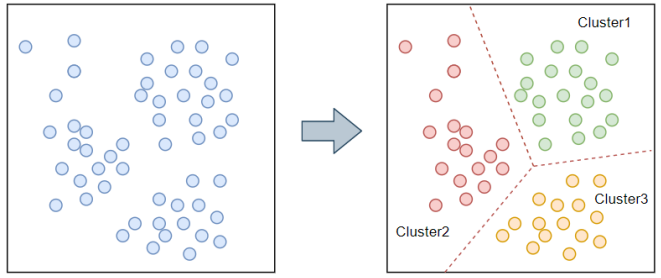




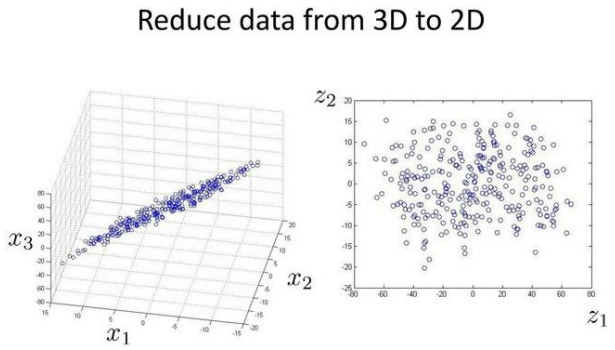
Aprendizado Não-Supervisionado

- Neste tipo de aprendizado, as máquinas não são informadas sobre o que aprender. Elas só recebem os exemplos de treinamento, $\boldsymbol{x}$.
- Neste caso, os algoritmos ***aprendem/descobrem padrões*** (muitas vezes ocultos) presentes nos dados de entrada sem a presença de rótulos.
- **Objetivo**: os modelos devem ***aprender/descobrir*** padrões 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

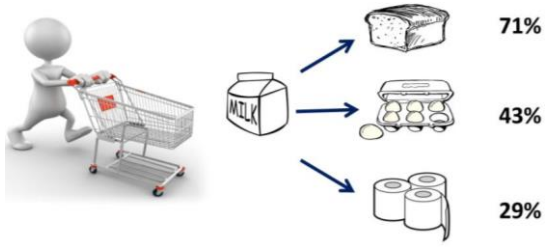


Redução de dimensionalidade



Detecção de Anomalias

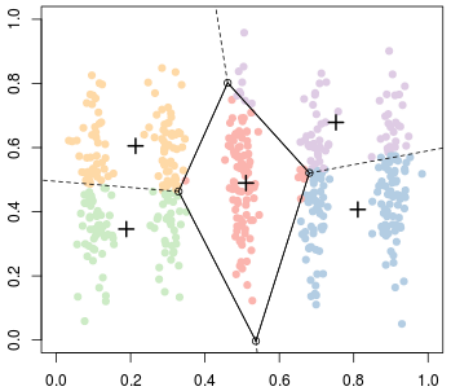
Regras de associação

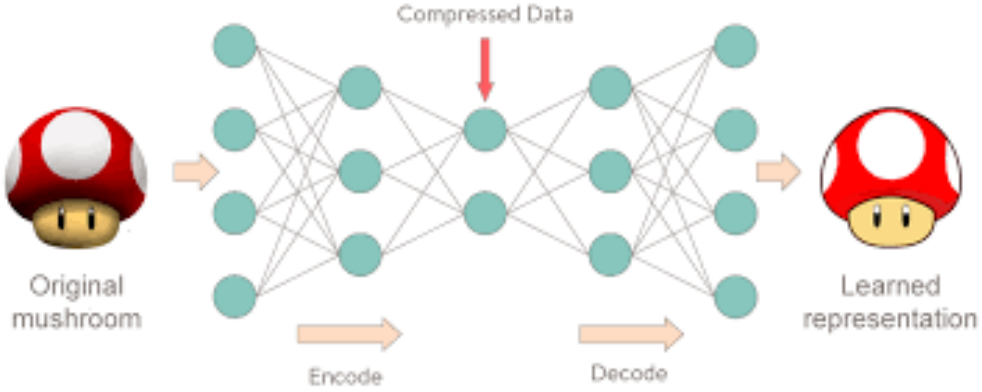


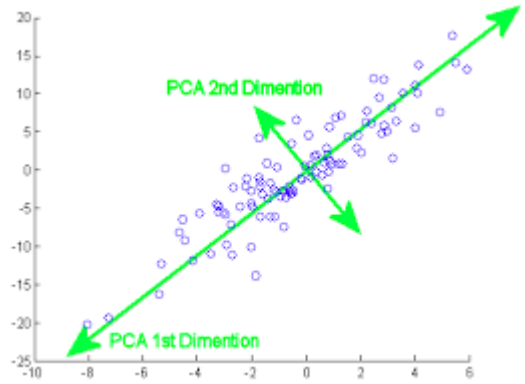
Of transactions that included milk:
- 71% included bread
- 43% included eggs
- 29% included toilet paper

Principais Algoritmos para Aprendizado Não-Supervisionado

- k-médias (*k-means*).
  - Particiona os exemplos em ***k*** clusters (ou grupos) distintos com base em sua distância até os centroides dos clusters.
- Redes Neurais Artificiais, e.g., auto-encoders.
  - Os autoencoders são usados para redução ou aumento de dimensionalidade.
- Análise de Componentes Principais (*Principal Component Analysis* - PCA).
  - Redução de dimensionalidade.





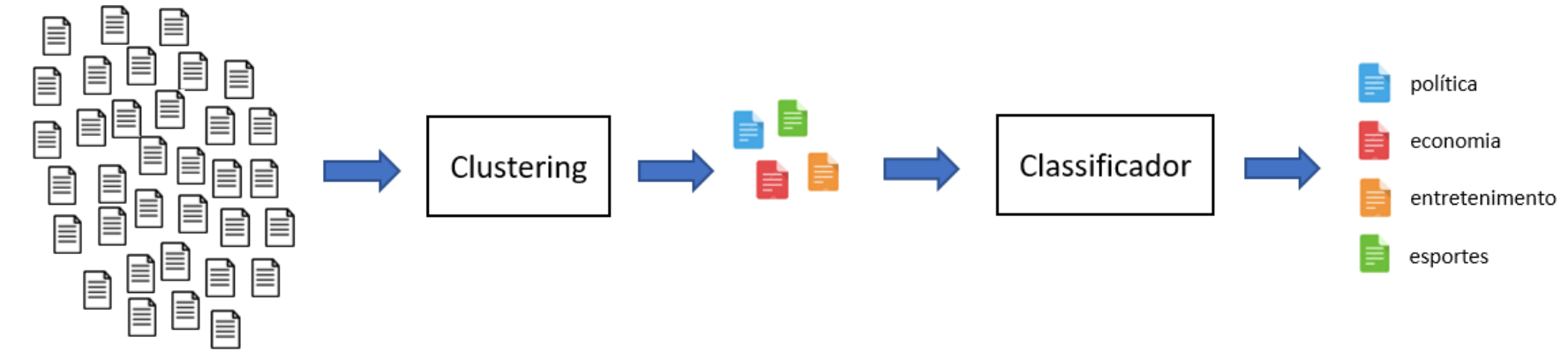


Aprendizado Semi-Supervisionado

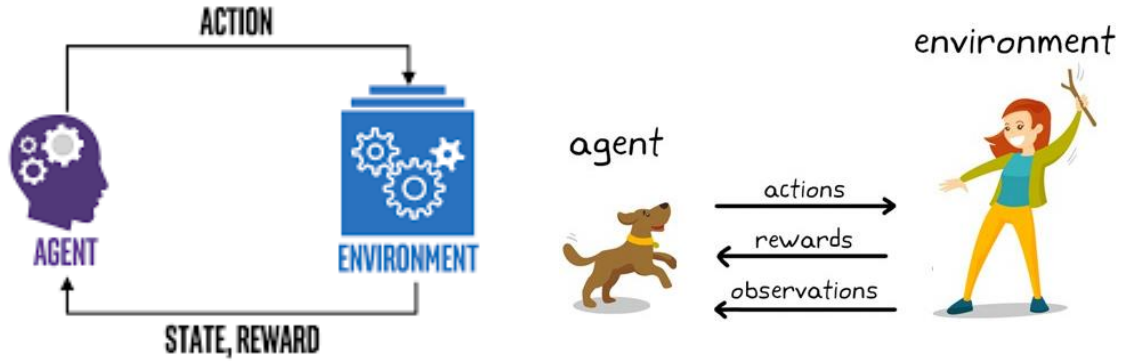
- Neste tipo de aprendizado, as máquinas têm acesso a exemplos com e sem rótulos.
- Geralmente envolve uma ***pequena quantidade de dados*** rotulados e uma ***grande quantidade de dados não-rotulados***.
- É de grande ajuda em casos onde se ter uma grande quantidade de dados rotulados é muito demorado, 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 Semi-Supervisionado

- **Exemplo**: Como ***classificaríamos*** milhões de textos ***não-rotulados*** da internet em categorias como economia, esportes, política, entretenimento, etc.?
- Poderíamos usar ***clustering*** para agrupar a quantidade massiva de textos e usar apenas os exemplos mais representativos de cada ***cluster*** (quantidade bem menor de textos) para ***rotular manualmente***.
- Esses ***exemplos rotulados*** são usados para treinar um ***classificador***.
- Após o treinamento, o ***classificador*** classifica automaticamente todos os textos.



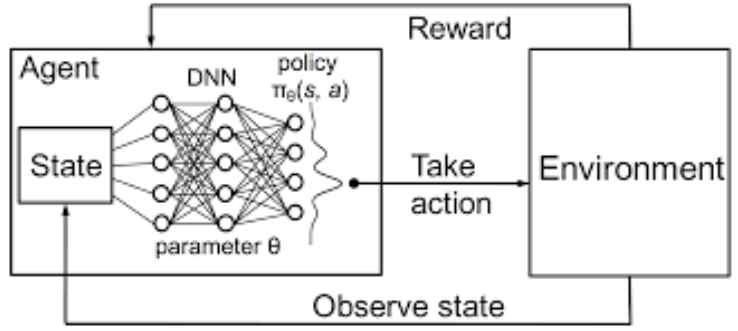
Aprendizado Por Reforço



- Abordagem 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.

Principais Algoritmos de Aprendizado Por Reforço

- **Q-Learning**
  - Usado para encontrar uma ***política*** ótima de seleção de ***ações*** usando a ***função-Q***.
  - ***Q,*** ou ***valor-Q***, representa a ***qualidade*** de uma dada ***ação*** em um determinado ***estado***.
  - É impraticável se encontrar uma ***política*** quando número de ***estados*** e ***ações*** é grande.
- **Deep Q-Learning**
  - Junção de Deep Learning + Q-Learning.
  - Redes neurais profundas possibilitam que Q-Learning seja aplicado a problemas com número gigantesco de ***estados*** e ***ações***.
  - O Q-Learning tabela a ***função-Q***, já o Deep Q-Learning encontra uma ***função*** que aproxime a ***função-Q***.

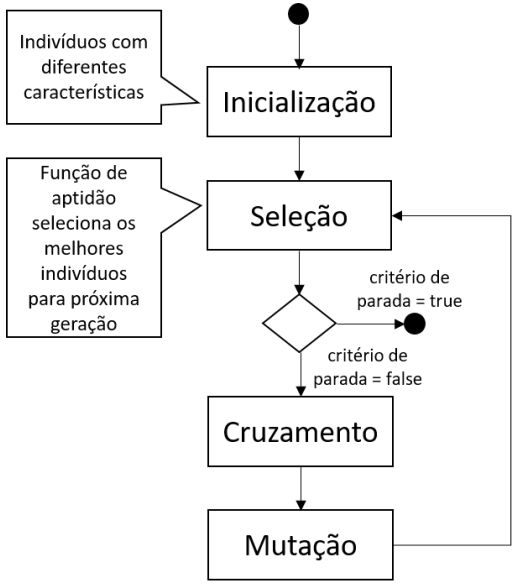


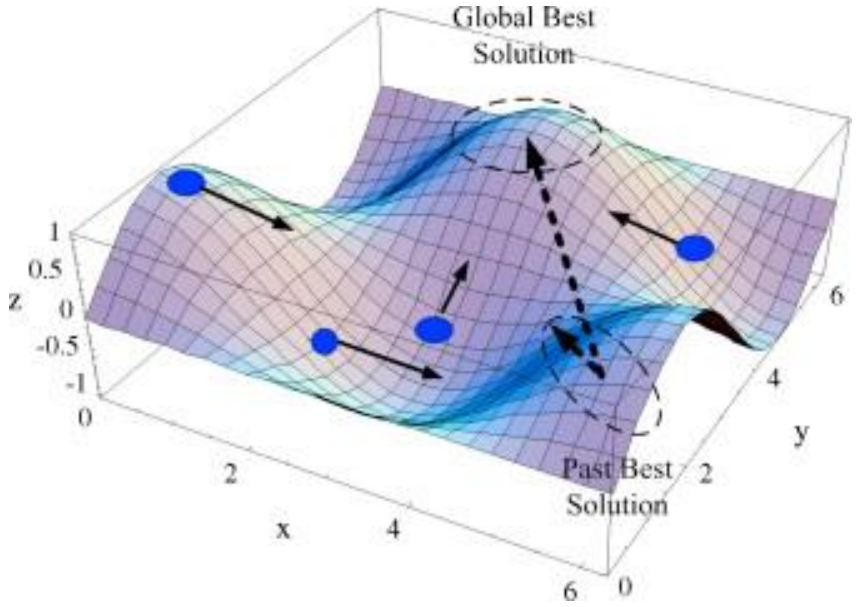
Aprendizado Metaheurístico

- 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.

Principais Algoritmos de Aprendizado Metaheurístico

- Algoritmo Genético (*Genetic Algorithm* - GA).
  - Inspirados pelo processo de seleção natural.
- Optimização por enxame de partículas (*Particle Swarm Optimization* - PSO).
  - Inspirado no comportamento de cardumes de peixes e de bandos de pássaros
- Otimização da colônia de formigas (*Ant Colony Optimization* - ACO).
  - Inspirado no comportamento das formigas ao saírem de sua colônia para encontrar comida.





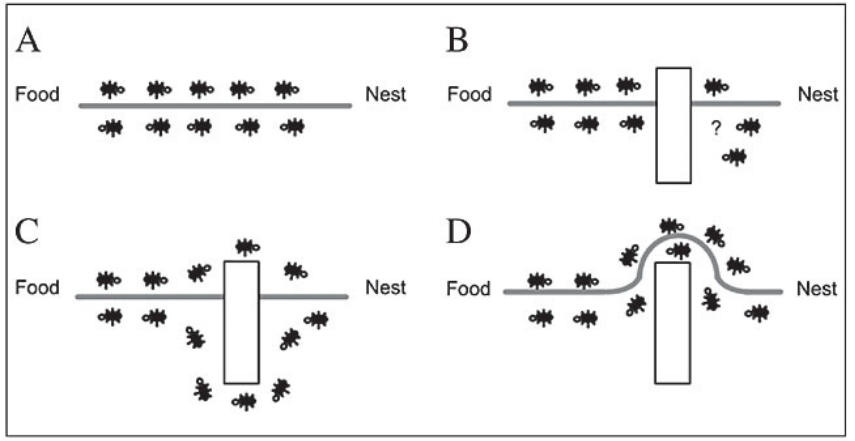


Figure 2. A. Ants in a pheromone trail between nest and food; **B.** an obstacle interrupts the trail; **C.** ants find two paths to go around the obstacle; **D.** a new pheromone trail is formed along the shorter path.

Tipos de Treinamento

Uma outra forma de se classificar algoritmos de ML é com relação se eles podem ser treinados incrementalmente ou não. Assim, os algoritmos podem ser divididos em algoritmos com teinamento:

- **incremental (online)**.
- **em batelada (batch)**.

Treinamento incremental

- Neste tipo de treinamento, o algoritmo ***aprende incrementalmente***:
  - Os exemplos de treinamento são apresentados ***sequencialmente um-a-um*** ou em ***pequenos grupos*** chamados de ***mini-batches***.
- Cada ***iteração de treinamento é rápida*** possibilitando que o sistema aprenda sobre novos dados à medida que eles chegam.
- Ótima opção para casos onde os dados chegam como um ***fluxo contínuo*** ou se tem ***recursos computacionais limitados***.
- Entretanto, como na maioria dos casos não é possível se realizar pré-processamento/análise dos dados, dados corrompidos ou com problemas afetam a performance do sistema.

Treinamento em batelada

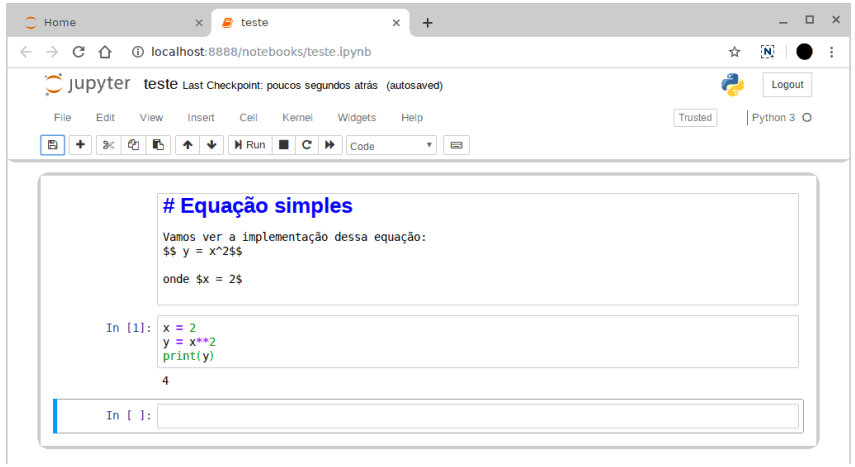
- Neste tipo de treinamento, o algoritmo é ***treinado com todos os exemplos disponíveis***.
- É um tipo de treinamento simples, de fácil implementação e obtém ótimos resultados.
- Dados podem ser pré-processados/analisados, evitando assim, dados corrompidos ou com problemas.
- O treinamento é mais demorado e utiliza mais recursos computacionais (CPU e memória) quando comparado ao treinamento incremental.
- Para treinar com novos exemplos é necessário iniciar o treinamento do zero.
- Se a quantidade de dados do conjunto de treinamento for muito grande pode ser impossível treinar em batelada.

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.

Binder

- **Binder**: aplicação web gratuita que permite a criação e edição de ***notebooks Jupyter*** em navegadores web.
- Suporta a execução de várias linguagens de programação: Python, C++, C#, PHP, Julia, R, etc.
- Algumas desvantagens do **Jupyter** são:
  - 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/



Goolge Colaboratory (Colab)

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

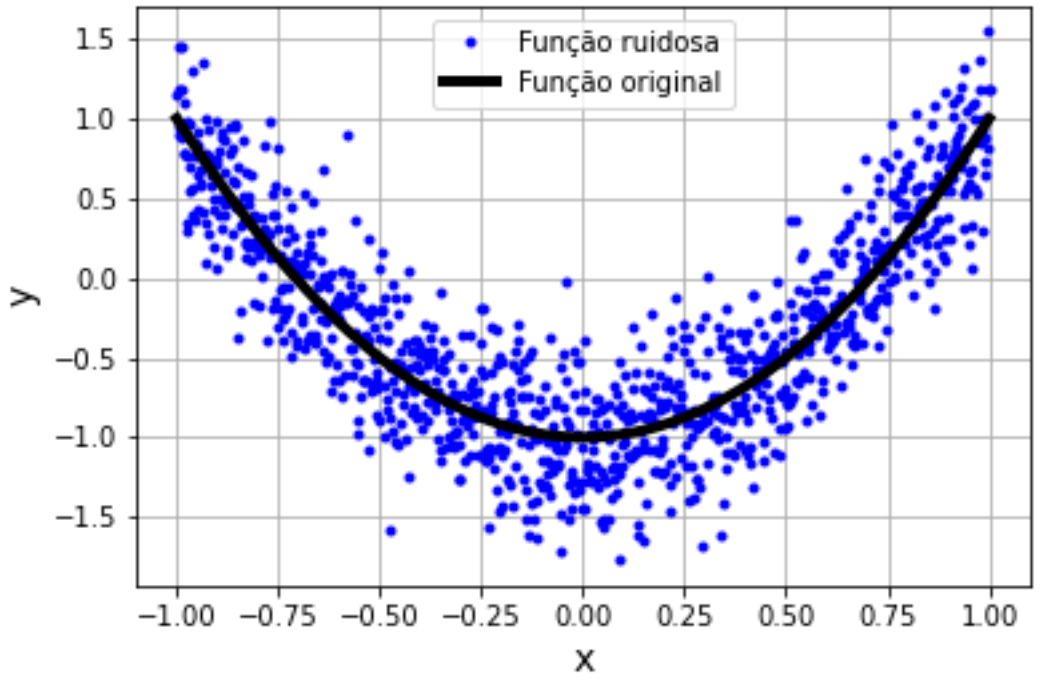
Exemplo de uso dos notebooks com Colab

colab

Figura 2D

```
# Importando as bibliotecas necessárias.
import numpy as np
import matplotlib.pyplot as plt

# Reseta o gerador de sequências pseudo-aleatórias.
np.random.seed(42)

# Define o número de exemplos.
N = 1000

# Vetor coluna com dimensão Nx1, com valores linearmente espaçados entre -1 e 1.
x = np.linspace(-1, 1, N).reshape(N,1)

# Vetor ruído com dimensão Nx1 e variância igual a 0.1.
w = np.sqrt(0.1)*np.random.randn(N,1)

# Função original.
y = -1 + 2*x**2

# Versão ruidosa de y.
y_noisy = y + w

plt.plot(x, y_noisy, '.b', label='Função ruidosa')
plt.plot(x, y, 'k', label='Função original', linewidth=4)
plt.xlabel('x', fontsize=14)
plt.ylabel('y', fontsize=14)
plt.legend()
plt.grid()
# salva figura em arquivo
plt.savefig('figura_2D.png')
# Mostra a figura.
plt.show()
```

Exemplo (colab): Figura_2D.ipynb



Exemplo (binder): Figura_2D.ipynb

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/1IyIIMu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing

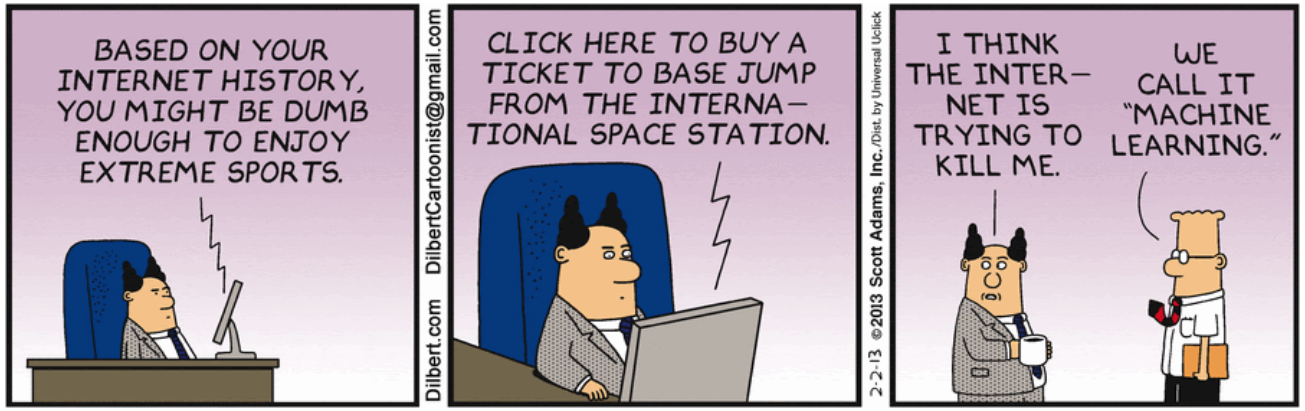
Avisos

- 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 18:30 às 19:30 e sextas-feiras das 15:30 às 16:30.
  - Monitor (Pedro Rezende: ***pedro_rafael@get.inatel.br***): Todas as sextas-feiras das 17:30 às 18:30.
  - Atendimento via MS Teams.

Tarefas

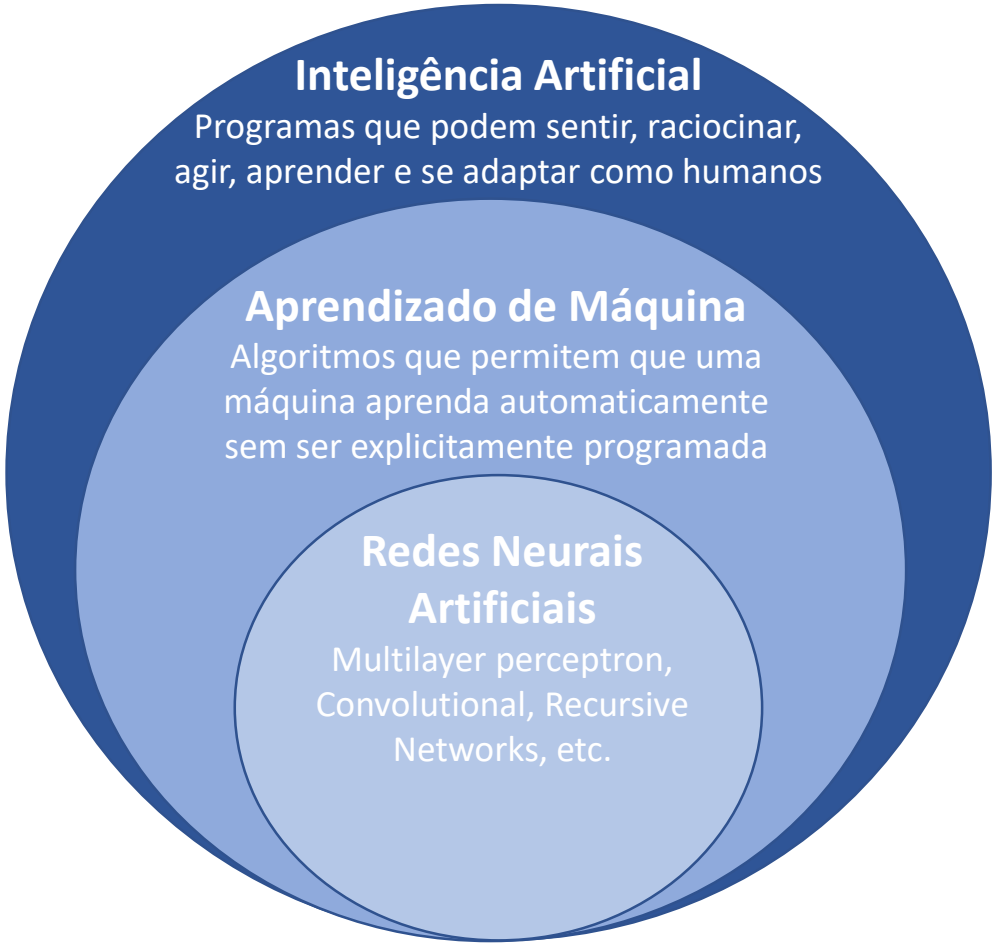
- **Quiz**: “*T319 - Quiz - Introdução*” que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #1**.
  - Pode ser baixado do MS Teams ou do GitHub.
  - Pode ser respondido através do link acima (na nuvem) ou localmente.
  - Instruções para resolução e entrega dos laboratórios.
  - **Laboratórios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

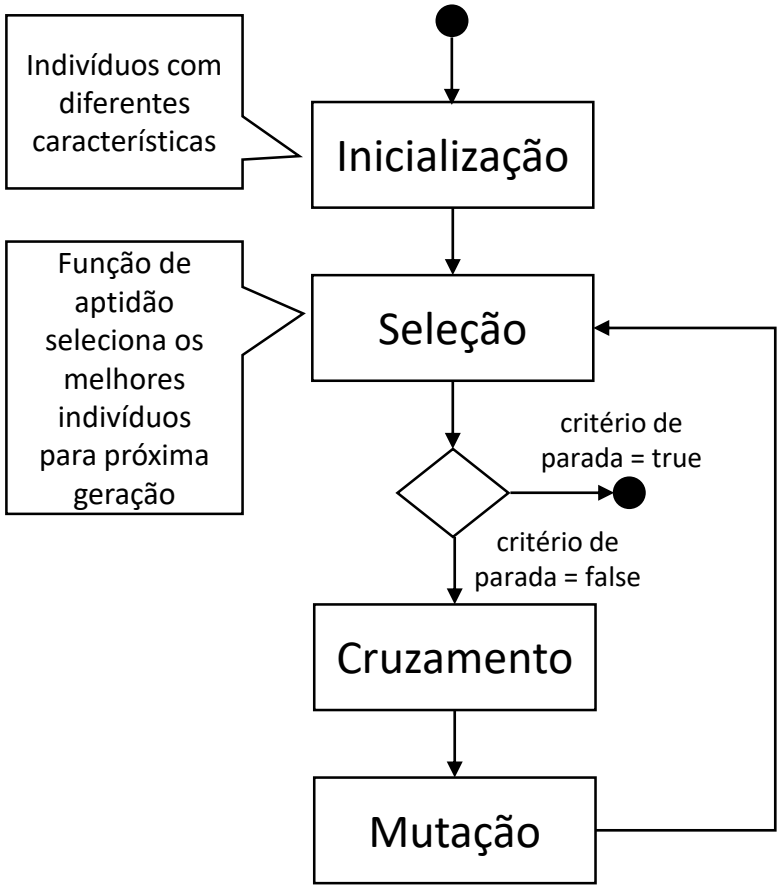
Obrigado!


BASED ON YOUR INTERNET HISTORY, YOU MIGHT BE DUMB ENOUGH TO ENJOY EXTREME SPORTS.
Dilbert.com DilbertCartoonist@gmail.com
CLICK HERE TO BUY A TICKET TO BASE JUMP FROM THE INTERNA-TIONAL SPACE STATION.
2-2-13 ©2013 Scott Adams, Inc. /Dist. by Universal Uclick
I THINK THE INTER-NET IS TRYING TO KILL ME.
WE CALL IT "MACHINE LEARNING."


OUR MACHINE LEARNING TECHNOLOGY ALLOWS US TO TRACK CUSTOMER PREFERENCES AND USE THAT KNOWLEDGE TO MANIPULATE THEM.
Dilbert.com DilbertCartoonist@gmail.com
THAT SEEMS LIKE THE STEP THAT HAPPENS RIGHT BEFORE THE MACHINES TAKE OVER THE EARTH AND ANNI-HILATE ALL HUMANS.
1-31-13 ©2013 Scott Adams, Inc. /Dist. by Universal Uclick
THERE'S ALWAYS ONE PERSON IN EVERY CROWD WHO SAYS THAT.
NOT FOR MUCH LONGER, APPAR-ENTLY.


WE HAVE A GIGANTIC DATABASE FULL OF CUSTOMER BEHAVIOR INFORMATION.
www.dilbert.com scottadams@aol.com
EXCELLENT. WE CAN USE NON-LINEAR MATH AND DATA MINING TECHNOLOGY TO OPTIMIZE OUR RETAIL CHANNELS!
© 2000 United Feature Syndicate, Inc.
IF THAT'S THE SAME THING AS SPAM, WE'RE HAVING A GOOD MEETING HERE.


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.


Indivíduos com diferentes características
Inicialização
Função de aptidão seleciona os melhores indivíduos para próxima geração
Seleção
critério de parada = true
critério de parada = false
Cruzamento
Mutação