

# TP555 - Inteligência Artificial e Machine Learning: *Introdução*



***Inatel***

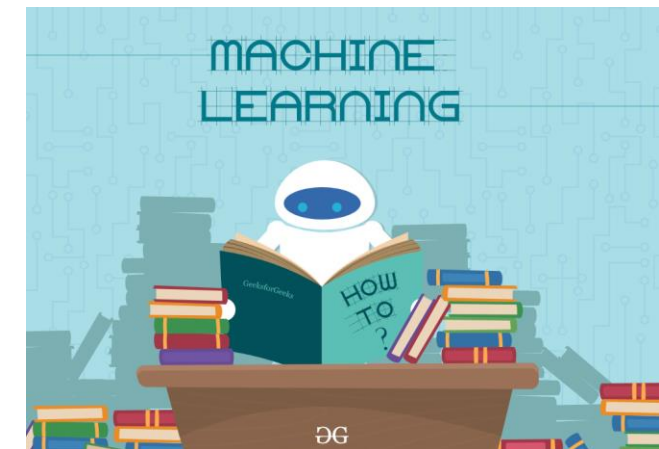
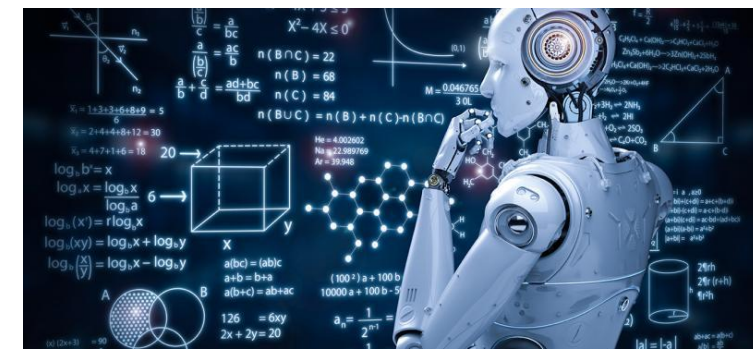
Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

# A disciplina

- Introdução ao aprendizado de máquina.
- Como o próprio nome diz, é um curso introdutório onde veremos os ***conceitos básicos*** de funcionamento de vários ***algoritmos de aprendizado de máquina*** ou do inglês, ***machine learning*** (ML).
- Não nos aprofundaremos nos conceitos matemáticos envolvidos, mas focaremos no uso e aplicação prática dos algoritmos.
- Porém, é necessário que vocês conheçam Python, Telecomunicações e alguns conceitos básicos de álgebra linear, cálculo e estatística.

# Objetivo do curso

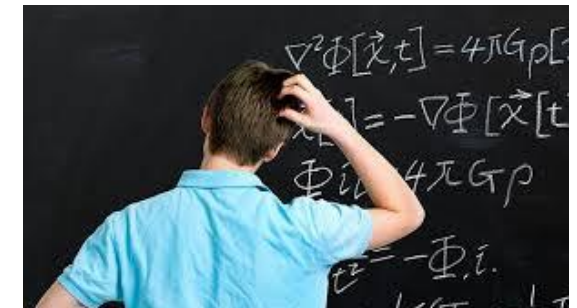
- O objetivo principal do curso é apresentar à vocês
  - os conceitos fundamentais da teoria por trás do ML.
  - um conjunto de ferramentas (ou seja, algoritmos) de ML.
- Ao final do curso vocês devem ser capazes de
  - Entender e discutir sobre os principais algoritmos de ML.
  - Compreender a terminologia utilizada na área.
  - Aplicar algoritmos de ML para a resolução de problemas.
  - Analisar e entender novos algoritmos de ML.
  - Criar seus próprios projetos.
  - Compreender artigos científicos que utilizam ML.
  - Realizar pesquisa que envolva ML e discutir os resultados obtidos.



# Avaliação do Curso



- 2 Provas
  - Cada uma valendo 35% da nota (datas ainda serão definidas).
  - Devem ser resolvidas individualmente durante a aula.
- 1 Projeto Prático
  - 30% da nota.
  - Pode ser feito em grupo de no máximo 2 alunos.
  - Tema deve ser escolhido por vocês.
  - Vejam descrição detalhada na tarefa do Teams.
  - Entrega da descrição até 31/03.
- Avaliação parcial do projeto prático ocorrerá ao final do segundo mês de curso.
  - **Entregável:** relatório com a evolução do projeto, como, por exemplo, a reprodução dos resultados.



# Motivação

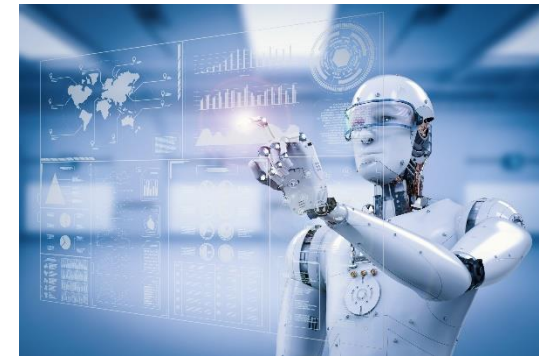
- **Emprego:** grandes companhias usam IA em seus produtos ou soluções internas para resolver os mais diversos tipos de problemas e, assim, aumentarem sua eficiência e, conseqüentemente, seus lucros.



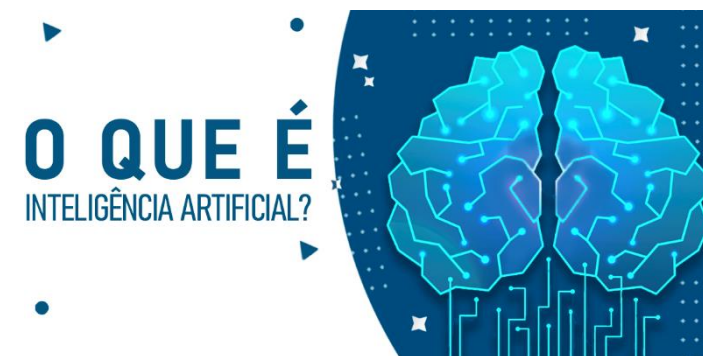
- **Pesquisa:** já se prevê que IA terá um papel importante no desenvolvimento da próxima geração de redes móveis e sem fio (i.e., 6G).



# Inteligência Artificial



- **Definição:** Capacidade de uma máquina de ***interpretar 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 como nós não é uma tarefa tão simples.





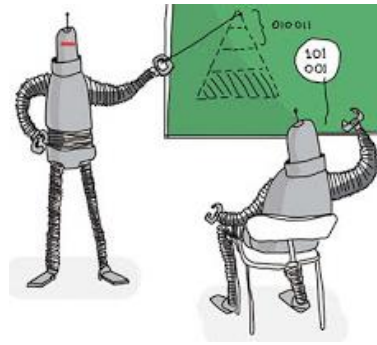
# Inteligência Artificial

- Para criar máquinas que simulem a inteligência humana, divide-se 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 (i.e., experiências prévias).
  - Inteligência artificial geral.
    - ✓ Criação de máquinas que solucionem qualquer tipo de problema. É a meta final da área de IA.

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



# Foco do curso



- Como vimos, IA é uma área muito ampla, e, portanto, focaremos no estudo de algoritmos de ***Aprendizado de Máquina*** (*Machine Learning - ML*).
- Por quê?
  - ***Caixa de ferramentas***: ML oferece ***ferramentas importantes para a análise e solução eficiente de vários problemas*** em várias áreas, incluindo telecomunicações.
  - ***Redução da complexidade e custo***: vários procedimentos e processos de 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*** que usem ML para a solução de problemas.



# Mas então, o que é ML?



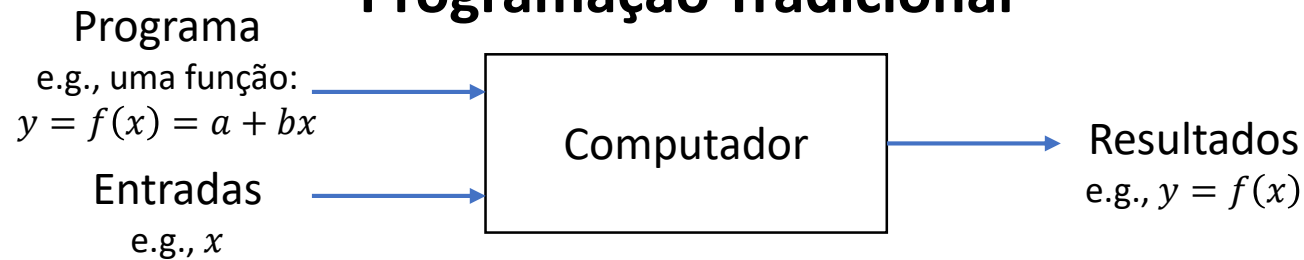
- É uma das subáreas da inteligência artificial.
- O termo foi cunhado em 1959, pelo cientista da computação Arthur Samuel, que o definiu como o  
*“Campo de estudo que dá aos computadores a habilidade de **aprender sem serem explicitamente programados.**”*
- Mas como eles aprendem?
  - Através de **experiências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



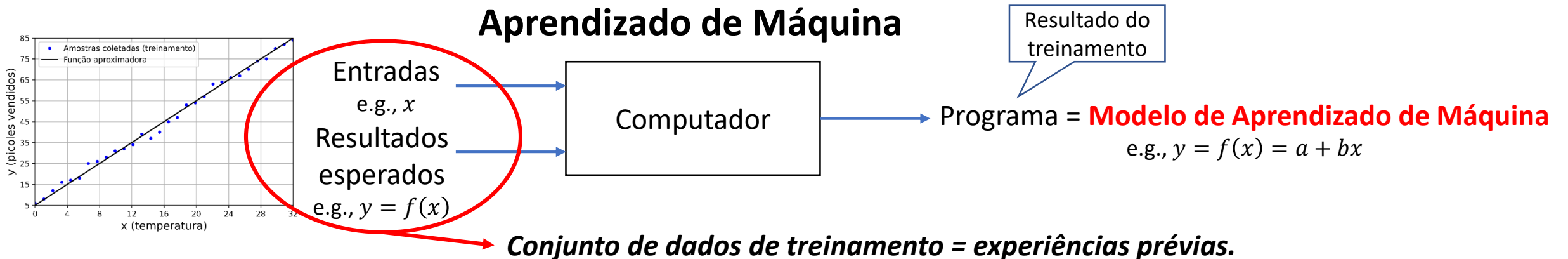
# O que é o Aprendizado de Máquina?

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

## Programação Tradicional

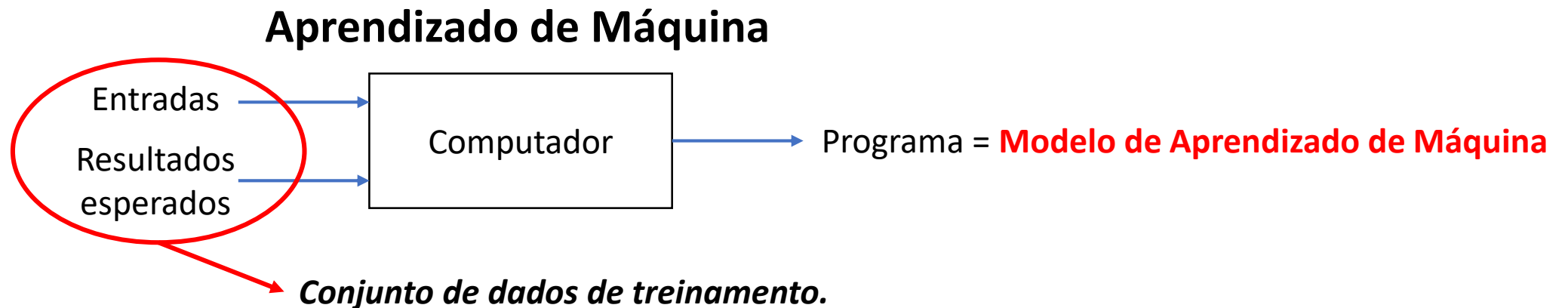


## Aprendizado de Máquina

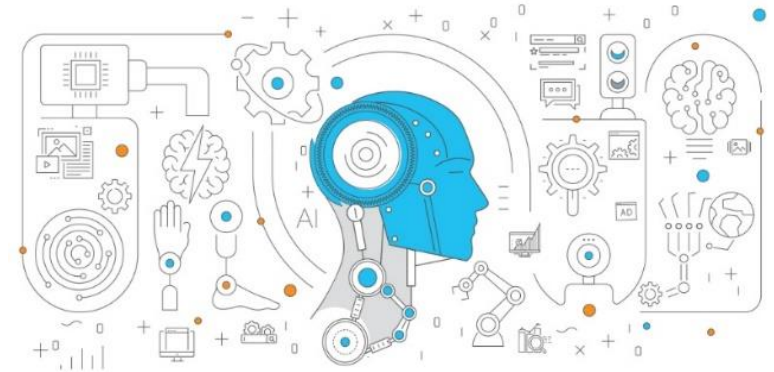


# O que é o Aprendizado de Máquina?

- 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), chatbots.
- **Educação:** pontuação automatizada de fala em testes de Inglês.
- **Medicina:** detecção e diagnóstico de doenças como câncer, Alzheimer, pneumonia, COVID-19, etc.
- **Finanças:** detecção de fraudes com cartão de crédito.
- **Tecnologia:** assistentes pessoais (e.g., *Siri, Alexa, Cortana*, etc.).



# Exemplos em Comunicações Digitais



- **Estimação, equalização e detecção conjunta**
  - Ex.: ML estima e equaliza o canal e detecta os símbolos transmitidos conjuntamente, sem a necessidade de realizar estas tarefas individualmente.
- **Aprendizado de Ponta-a-Ponta de Sistemas de Comunicação**
  - Ex.: ML otimiza conjuntamente os blocos de processamento do Tx e Rx (estimação de canal, equalização, codificação/decodificação de canal, modulação/demodulação, etc.)
- **Comunicações em Ondas Milimétricas**
  - Ex.: ML é utilizado para prever bloqueios e realizar *handovers* proativos.
- **Alocação de Recursos**
  - Ex.: ML é empregado em redes IoT de baixa potência e sem contenção para entender o comportamento de dispositivos vizinhos e, assim, realizar a alocação de recursos com o intuito de evitar colisões, aumentando a taxa de dados útil e a vida útil das baterias.
- **Rádio-sobre-Fibra**
  - Ex.: ML é utilizado para estimar, equalizar e detectar conjuntamente sinais transmitidos através de canais sem fio e óticos, considerando os efeitos conjuntos de ambos os canais (e.g., distorção, desvanecimento, ruído, etc.).
- **Pré-distorção**
  - Ex.: ML aproxima uma equação que aumenta a região linear de um amplificador de potência.
- **E vários outros tópicos:** <https://mlc.committees.comsoc.org/research-library/>

# Principais motivos da difusão do ML

- Possibilidade de ***analisar e extrair informações úteis de enormes volumes de 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 tem 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 reinforcement-learning, generative adversarial learning*, etc.
- Disponibilidade de *frameworks* e bibliotecas que facilitam o desenvolvimento de soluções com ML.

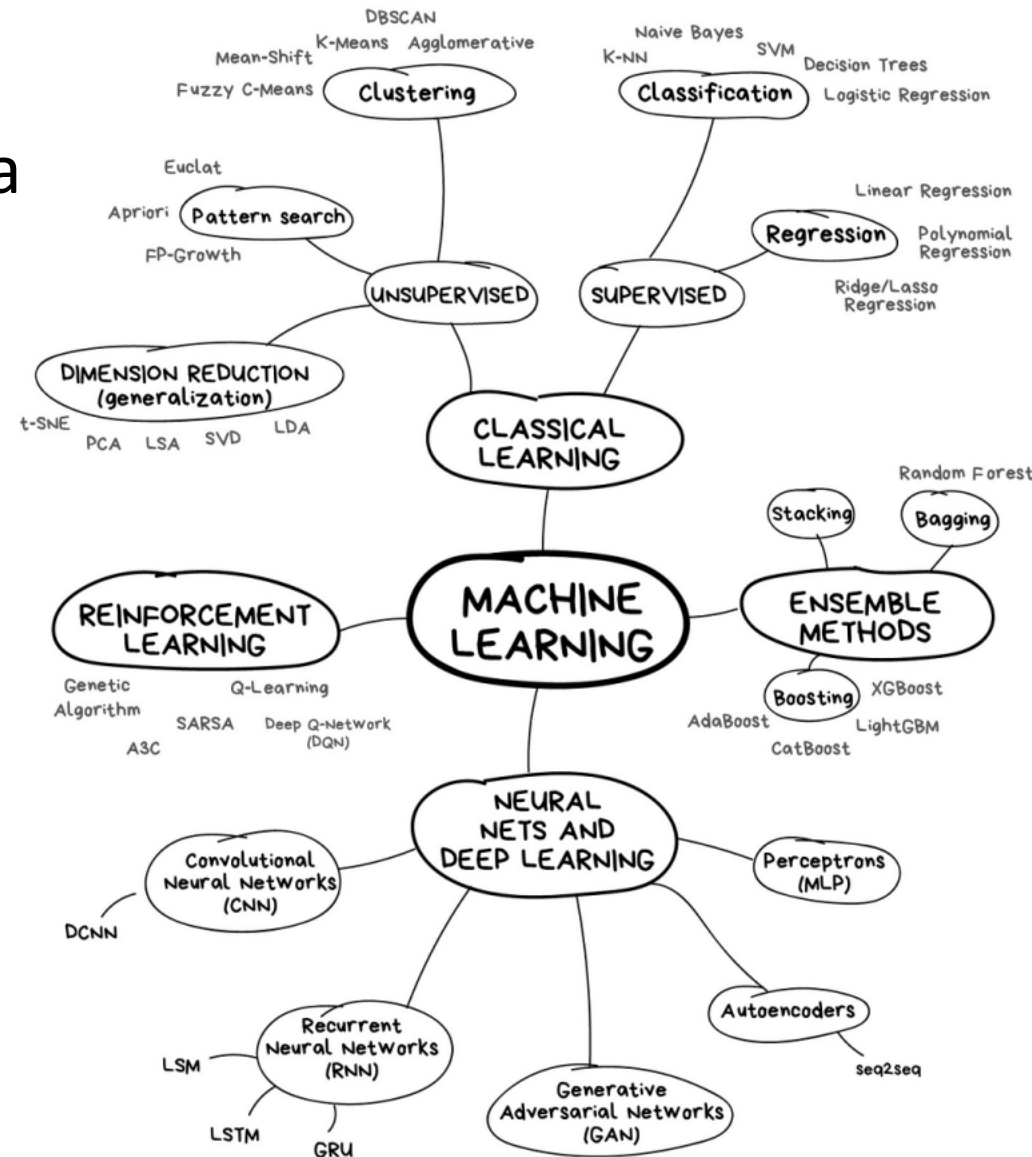




# Tipos de Aprendizado de Máquina

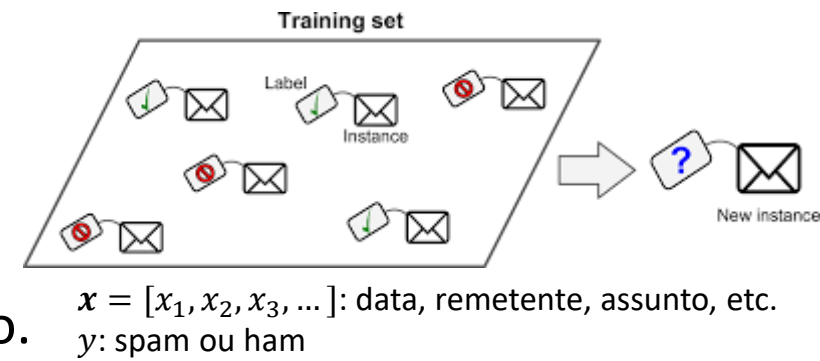
- Os algoritmos de aprendizado de máquina podem ser agrupados de acordo com o ***tipo de (paradigma) aprendizado que realizam:***

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



# Aprendizado Supervisionado

- No aprendizado supervisionado, o **algoritmo de ML 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,  $x$  (atributos), e sua saída correspondente,  $y$  (rótulo).
- **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

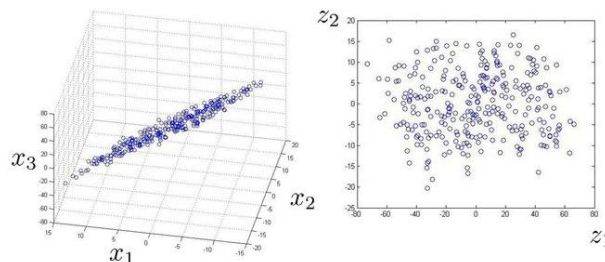
- Neste tipo de aprendizado, os algoritmos não têm acesso às saídas esperadas,  $y$ . Eles só têm acesso aos atributos,  $x$ .
- **Objetivo:** os algoritmos devem ***aprender/descobrir*** padrões, muitas vezes ocultos, presentes nos dados se baseando apenas, por exemplo, na similaridade entre os ***atributos,  $x$***  ou seja, ***sem a presença de rótulos***.
- Os algoritmos tratam problemas de clusterização, redução de dimensionalidade, detecção de anomalias (*outliers*) e aprendizado de regras de associação.

Clusterização

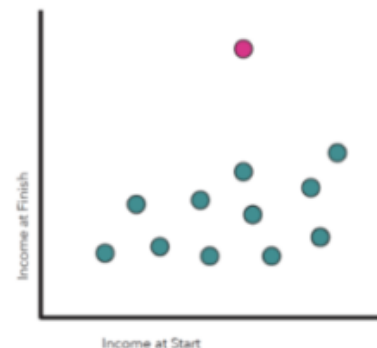


Redução de dimensionalidade

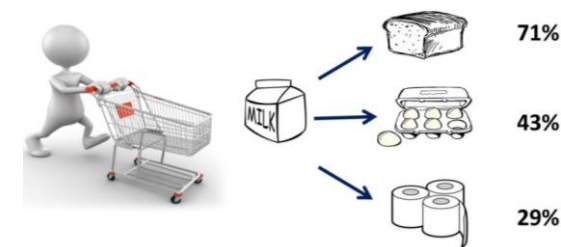
Reduce data from 3D to 2D



Detecção de Anomalias



Regras de associação

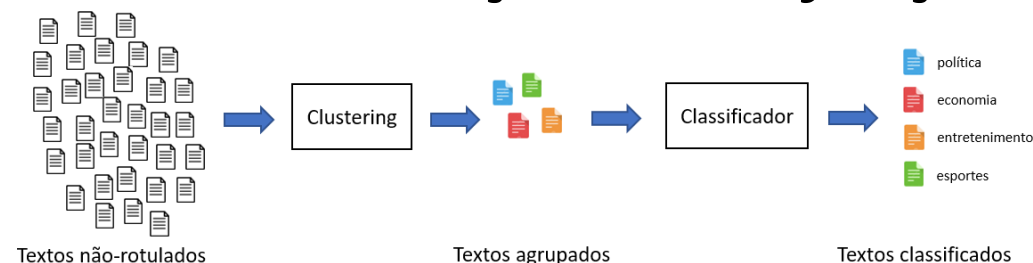


Of transactions that included milk:

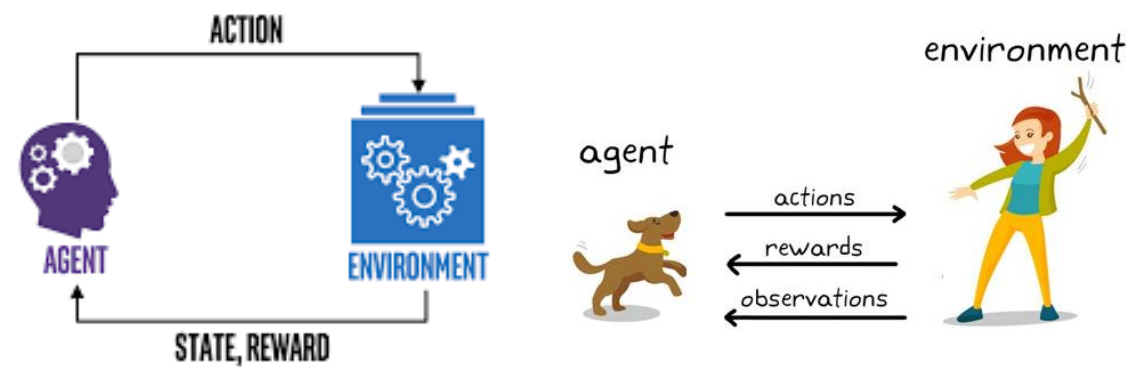
- 71% included bread
- 43% included eggs
- 29% included toilet paper

# 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 ***clusterização*** 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***, seleciona e executa uma ***ação*** e recebe uma ***recompensa*** (ou ***reforço +/-***) em consequência da ***ação*** tomada.
- 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***,  $s$ , do ***ambiente*** em ***ações***,  $a$ , que o ***agente*** deve tomar,  $\pi(s) = a$ .

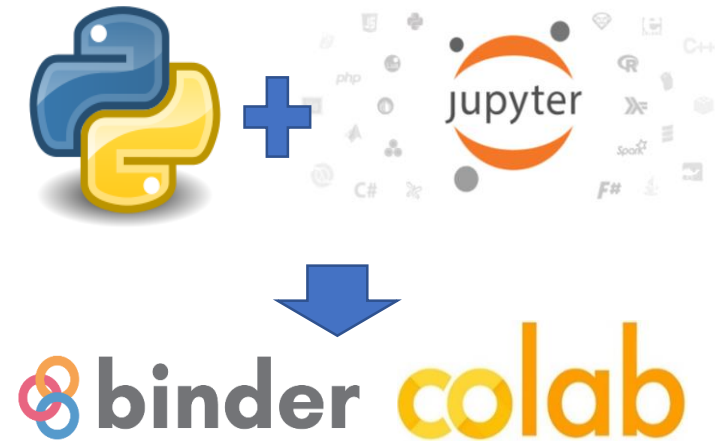
# Aprendizado Metaheurístico

- Uma **metaheurística** é um algoritmo usado para encontrar soluções de **forma rápida e genérica**, mas muitas vezes **sub-ótimas**, para **problemas complexos de otimização**.
- 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 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 em dispositivos com **capacidade computacional limitada** (e.g., dispositivos IoT).
- São algoritmos inspirados pelo **processo de seleção natural** (e.g., algoritmo genético) ou no **comportamento de grupos de animais** (e.g., algoritmo de otimização da colônia de formigas).



# Executando códigos

- Durante o curso, usaremos **Python** como linguagem de programação.
  - Fácil de aprender, possui várias bibliotecas, é a linguagem mais utilizada em ML e é *open-source* e gratuita.
- Utilizaremos **notebooks Jupyter** para execução de exemplos e resolução dos exercícios.
  - Eles são **documentos virtuais** usados para **desenvolver 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 interativos e gratuitos executados na nuvem.
- Portanto, **vocês não precisam instalar nada**, apenas terem um navegador web e conexão com a internet.



# Google Colaboratory (Colab)



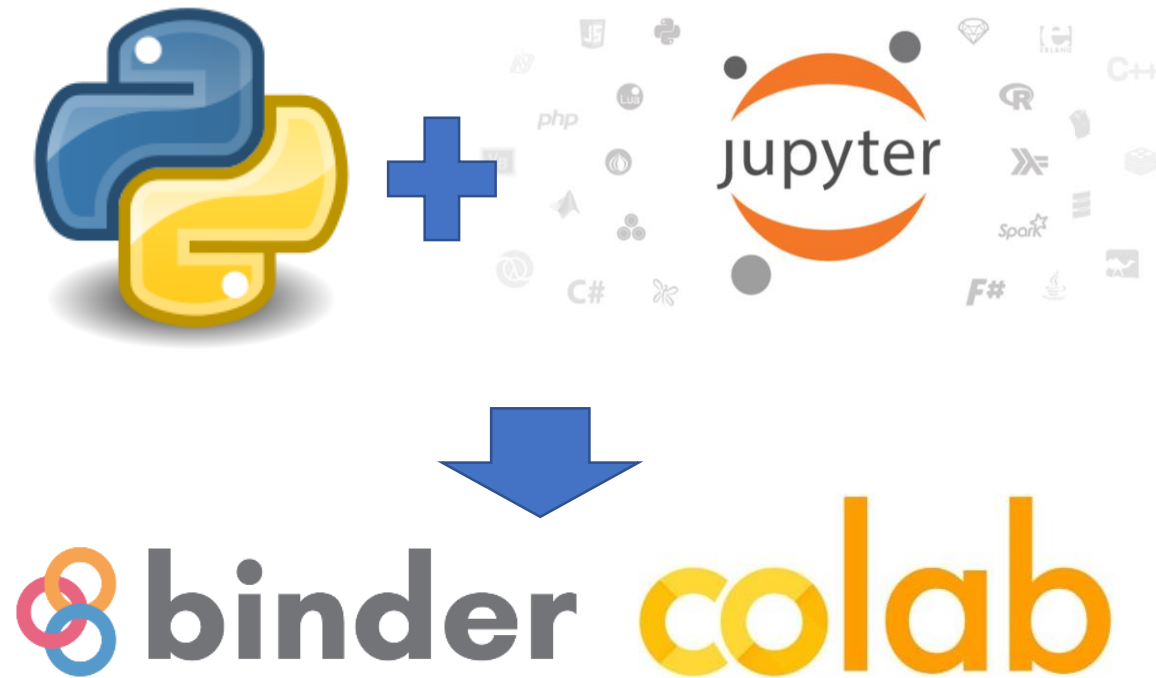
- **Colab**: aplicação web gratuita que permite a criação e edição de *notebooks* *Jupyter* em navegadores web.
- É um produto da Google.
- Vantagens:
  - Grande número de servidores.
  - Rápida inicialização e processamento do código.
  - Fornece acesso a GPUs e TPUs gratuitamente.
  - Notebooks podem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

# Binder



- **Binder:** outra aplicação web gratuita que permite a criação e edição de ***notebooks Jupyter*** em navegadores web.
- Vantagens:
  - Suporta a execução de várias linguagens de programação: Python, C++, C#, PHP, Julia, R, etc.
  - Pode ser instalado localmente.
    - ✓ A lista #0 contém um tutorial para instalação do Anaconda/Jupyter/Binder.
- 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/>

# Exemplo de uso dos notebooks Jupyter



[Figura\\_2D.ipynb](#)

# Referências

- [1] Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach,” Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.
- [3] 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.
- [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] Andriy Burkov, “The Hundred-Page Machine-Learning Book,” Andriy Burkov 2019.
- [6] C. M. Bishop, “Pattern Recognition and Machine Learning,” Springer, 1st ed., 2006.
- [7] S. Haykin, “Neural Networks and Learning Machines,” Prentice Hall, 3ª ed., 2008.
- [8] Coleção de livros,  
<https://drive.google.com/drive/folders/1IyIIMu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing>

# Avisos

- Vocês já podem começar a trabalhar nas listas #0 e #1.
- A descrição do projeto prático já está disponível no MS Teams.
  - Se atentem ao prazo de entrega da **proposta** do projeto prático.
- Todo material do curso está disponibilizado no GitHub:
  - <https://github.com/zz4fap/tp555-ml>
- Horário de Atendimento
  - Todas as quintas-feiras das 17:30 às 19:30.
  - Presencialmente ou remotamente.

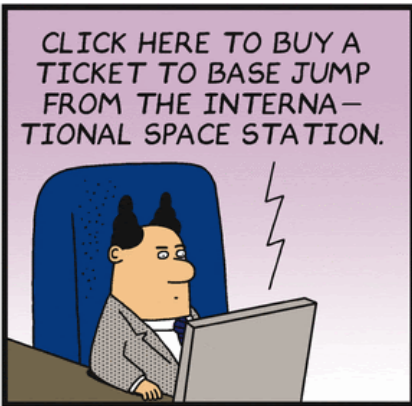


Perguntas?

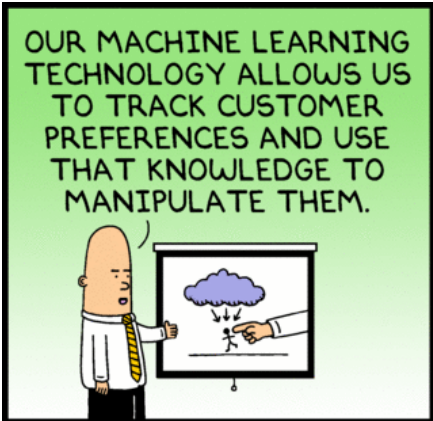
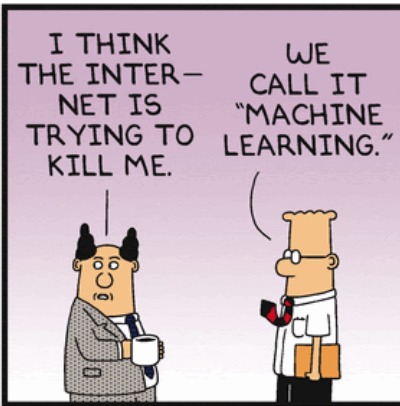
Obrigado!



Dilbert.com DilbertCartoonist@gmail.com



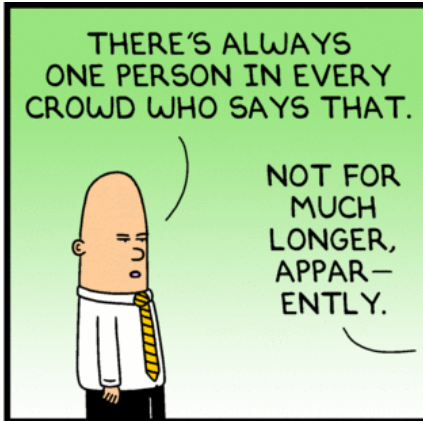
2-2-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick



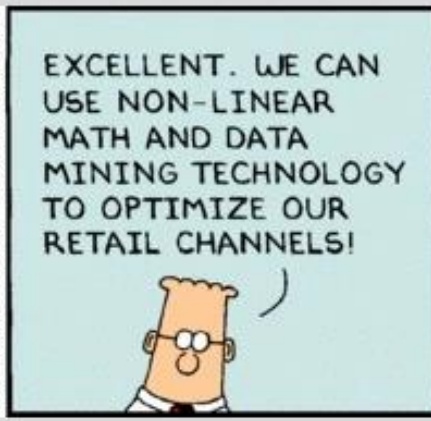
Dilbert.com DilbertCartoonist@gmail.com



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



www.dilbert.com scottadams@aol.com



©2003 United Feature Syndicate, Inc.



