

TP555 - Inteligência Artificial e Machine Learning: *Introdução*



Inatel

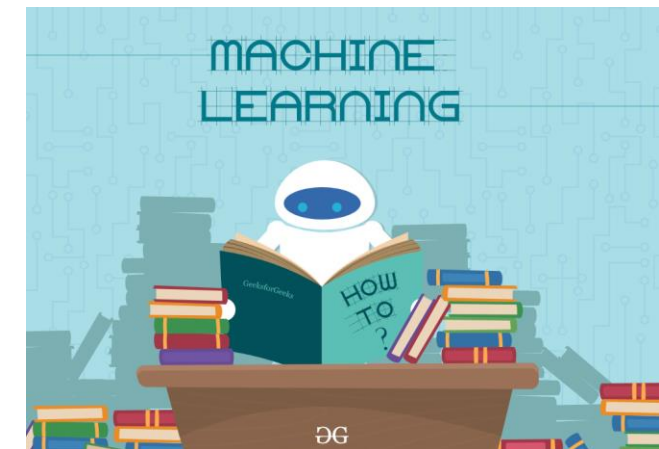
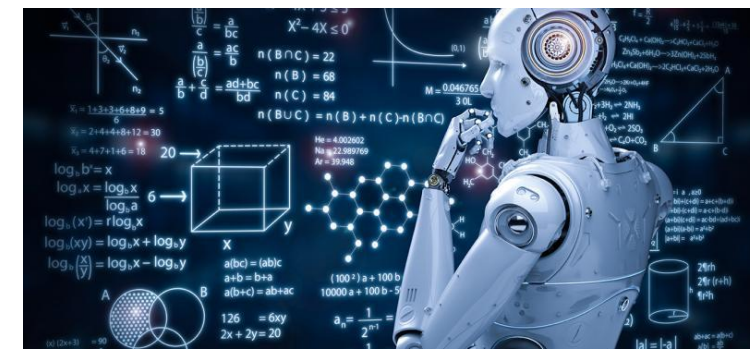
Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

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 nós aprofundaremos nos conceitos matemáticos envolvidos, mas focaremos na aplicação prática dos algoritmos.
- Porém, precisamos conhecer Python e alguns conceitos básicos de álgebra linear e estatística.

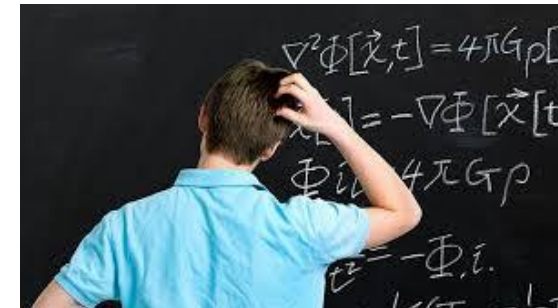
Objetivo do curso

- 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 prova valerá 40% da nota (datas ainda serão definidas).
- Projeto Prático
 - 20% da nota.
 - Pode ser feito em grupo.
 - Tema será escolhido por vocês.
- Listas de Exercícios
 - **Nota extra:** até 5 pontos na média final.
 - As listas serão disponibilizadas ao final de cada tópico.
 - [Devem ser comitadas no GitHub e entregues no MS Teams.](#)

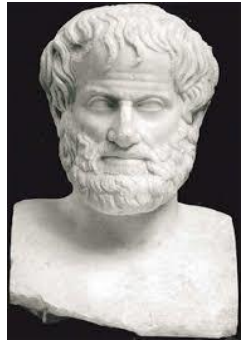


Motivação

- **Pesquisa:** algoritmos de ML tem um grande potencial para solucionar problemas em várias áreas tais como a de jogos, transportes, segurança, saúde, tecnologia, varejo, finanças, etc. Portanto, o emprego de ML em problemas destas áreas pode gerar pesquisas inovadoras.
 - Já se prevê que ML terá um papel importante no desenvolvimento da próxima geração de redes móveis e sem-fio (e.g., 6G).
- **Emprego:** grandes companhias (e.g., Google, Facebook, Amazon, etc.) usam ML para resolver os mais diversos tipos de problemas e assim aumentarem sua eficiência e, conseqüentemente, seus lucros.

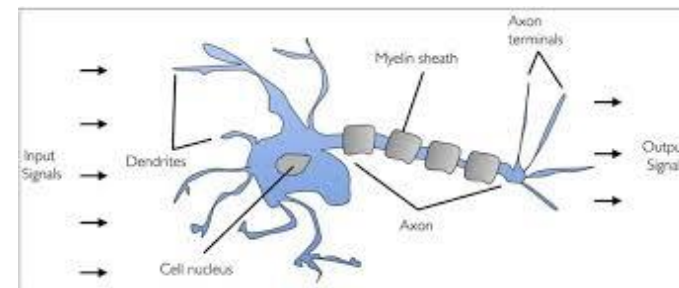
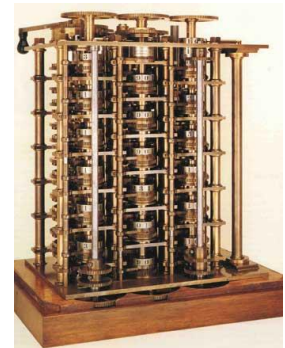


Um pouco de história



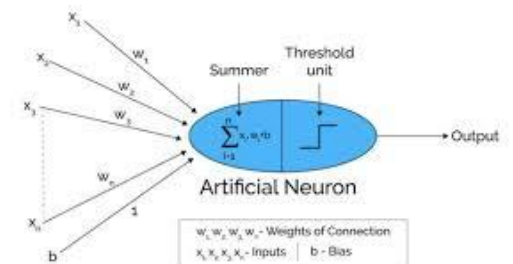
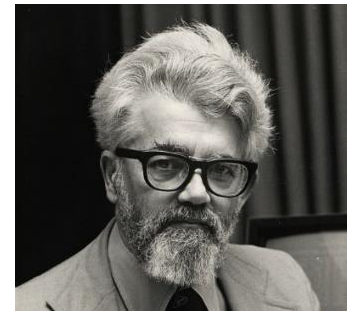
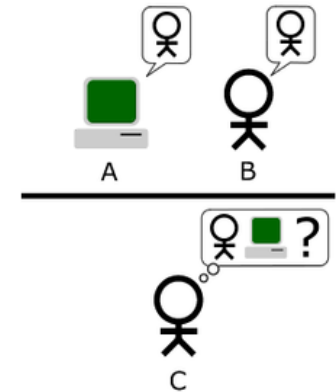
A ideia de criar uma máquina pensante que imitasse as habilidades humanas é muito antiga, remontando à Grécia antiga.

- **300 A.C.:** Aristóteles almejava substituir a mão-de-obra escrava por objetos autônomos.
- **1822:** Charles Babbage cria os primeiros computadores mecânicos, as *máquinas diferencial* e *analítica*, se tornando o “pai dos computadores”.
- **1943:** McCulloch e Pitts propõem um modelo matemático para o funcionamento de um neurônio, lançando as bases para a criação das redes neurais artificiais.



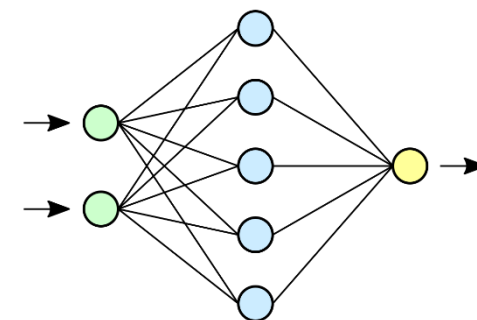
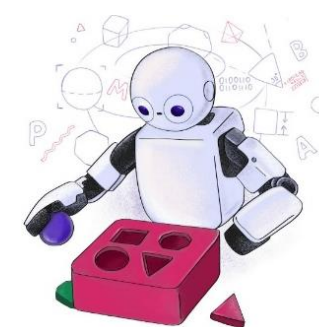
Um pouco de história (continuação)

- **1950:** Alan Turing cria o “*teste de Turing*”, que testa a capacidade de uma máquina em exibir comportamento inteligente indistinguível ao de um ser humano.
- **1952:** Arthur Samuel cria o primeiro programa de *auto-aprendizagem* (que mais tarde seria chamado de *aprendizado de máquina*) do mundo, o *Checkers-Player*, sendo a primeira demonstração do conceito de inteligência artificial.
- **1956:** Surgimento da área de pesquisa em IA em um workshop no Dartmouth College nos EUA onde o termo IA foi cunhado por John McCarthy.
- **1958:** Frank Rosenblatt cria a primeira rede neural artificial, chamada de *Perceptron*.



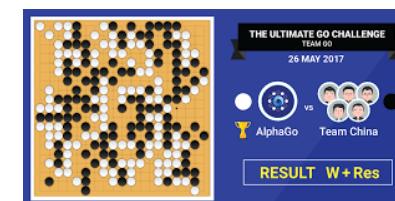
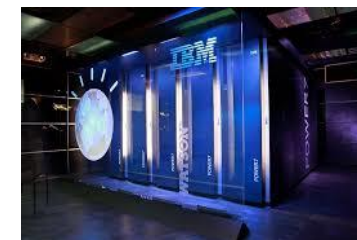
Um pouco de história (continuação)

- **1959:** Arthur Samuel cunha o termo *Aprendizado de Máquina* enquanto trabalhava na IBM.
- **1970-1980:** Devido a limitação dos computadores da época, IA atravessou alguns ***invernos***, i.e., interesse e financiamentos na área diminuíram drasticamente.
- **1986:** Ascensão do *aprendizado de máquina*: com o surgimento de computadores mais poderosos, as redes neurais retornam à popularidade com o algoritmo de ***backpropagation*** e ocorrem grandes avanços em algoritmos e aplicações de ML.
- **1997:** O supercomputador da IBM, chamado de *DeepBlue*, vence o campeão mundial de xadrez, Garry Kasparov.



Um pouco de história (continuação)

- **2009:** O Google constrói o primeiro carro autônomo que dirige em áreas urbanas.
- **2011:** Outro supercomputador da IBM, chamado *Watson*, vence o show de perguntas e respostas *Jeopardy!*.
- **2011-2014:** Surgem assistentes pessoais, tais como *Siri*, *Google Now*, *Alexa* e *Cortana*, que utilizam reconhecimento de fala para responder questões e realizar tarefas simples.
- **2016:** O programa conhecido como *AlphaGo*, da empresa *DeepMind*, derrota o então 18 vezes campeão mundial de *Go*, Lee Sedol.
- **2018:** Universidades de todo o mundo começam a oferecer cursos de AI e ML.



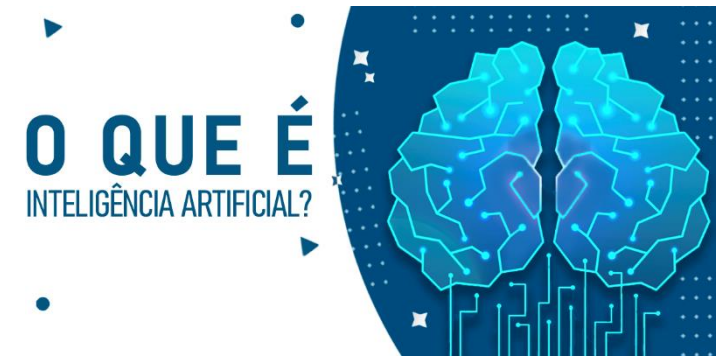
Inteligência + Artificial (IA)



- Vocês já pensaram sobre a definição destas duas palavras?
- **Inteligência:** conjunto das *habilidades mentais* de *conhecer, compreender, aprender, resolver problemas e adaptar-se*.
 - Surgiu do Latim como união do prefixo *inter*: “entre” + sufixo *legere*: “escolha” => “entre escolhas”.
 - Ou seja, inteligência é a *capacidade de escolha* de um indivíduo entre as várias possibilidades ou opções que lhe são apresentadas.
- **Artificial:** produzido pela mão do homem, não pela natureza, geralmente como uma cópia de algo natural.

Conjunto de habilidades mentais desenvolvidas de forma não natural pelo homem.

Definições e objetivo da IA

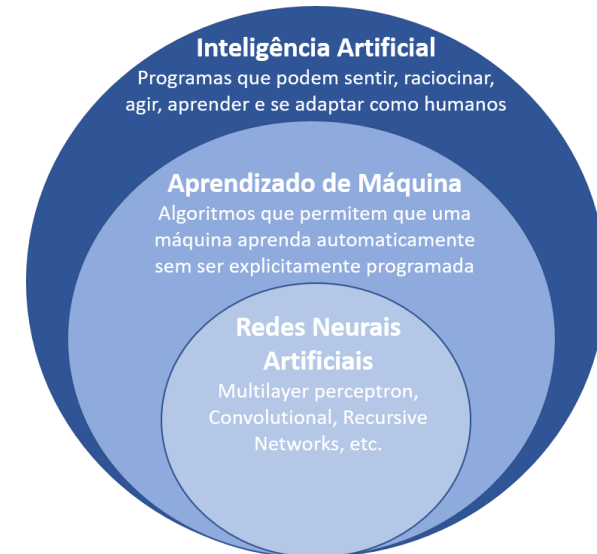
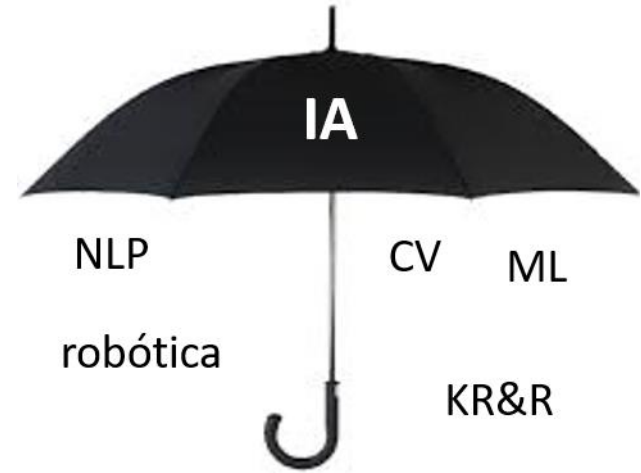


- **Definição:** “Capacidade de um sistema de interpretar corretamente dados externos (vindos do ambiente), aprender com esses dados e usá-los para atingir tarefas e objetivos específicos por meio de adaptação flexível.” (Andreas Kaplan).
- **Objetivo:** Criar máquinas que **imitem** nossas *habilidades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



Inteligência Artificial

- IA é uma área muito ampla que **engloba** várias **aplicações** (ou sub-áreas ou objetivos) tais como
 - processamento de linguagem natural,
 - representação do conhecimento,
 - raciocínio automatizado,
 - planejamento,
 - visão computacional,
 - robótica,
 - aprendizado de máquina, que por sua vez engloba redes neurais artificiais, deep learning, etc. e
 - inteligência artificial geral.

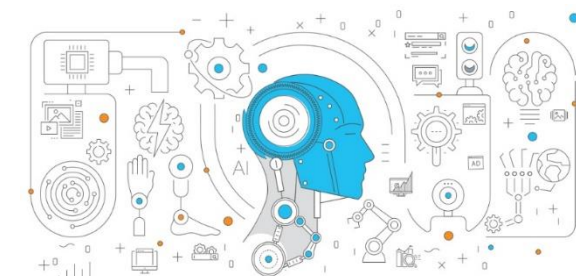


Algumas aplicações de IA



Algumas áreas onde IA é aplicada são:

- **Transporte:** veículos terrestres e aéreos autônomos, previsão do tráfego, etc.
- **Negócios:** recomendação de anúncios, produtos e conteúdos (e.g., netflix), chatbots para atendimento ao cliente, etc.
- **Educação:** pontuação automatizada de fala em testes de Inglês, correção de provas, chatbots para realização de matrículas, dúvidas, etc.
- **Clima:** previsão do tempo (temperatura, chuva, furacões, etc.).
- **Medicina:** detecção e/ou previsão de doenças (câncer, Alzheimer, pneumonia, COVID-19, etc.), chatbots que auxiliam no agendamento de consultas e respondem perguntas referentes a uma doença, descoberta de novas drogas, etc.
- **Finanças:** detecção de fraudes com cartão de crédito, previsão do comportamento do mercado de ações, etc.
- **Tecnologia:** filtros AntiSpam, “motores” de busca como o do Google, reconhecimento de fala, conversão de texto/fala e fala/texto, assistentes pessoais on-line (e.g., *Siri*, *Alexa*, etc.), tradução de textos.



Algumas Aplicações em Comunicações Digitais

- **Estimação de canal e detecção de sinais**

- Ex.: IA estima o canal e detecta o símbolo transmitido conjuntamente, sem a necessidade de estimar o canal primeiro e depois detectar o símbolo transmitido.

- **Aprendizado de Ponta-a-Ponta de Sistemas de Comunicação**

- Ex.: IA 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.: IA é utilizada para prever bloqueios e realizar handovers proativos.

- **Alocação de Recursos**

- Ex.: IA é empregada em redes IoT de baixa potência para entender o comportamento de dispositivos vizinhos e assim realizar a alocação de recursos com o intuito de evitar colisões, aumentando a vida útil das baterias e a taxa de dados úteis (*goodput*).

- **Rádio-sobre-Fibra**

- Ex.: IA é utilizada para equalizar e detectar conjuntamente sinais transmitidos através de canais sem-fio e óticos, considerando conjuntamente os efeitos de ambos os canais.

- **Pré-distorção**

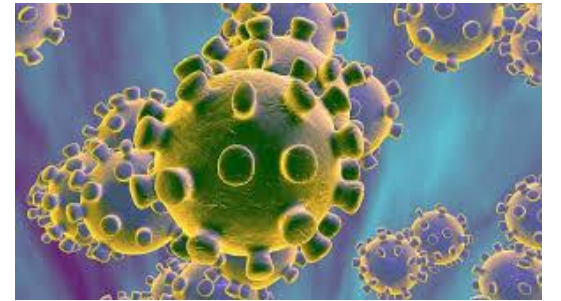
- Ex.: IA é utilizada para aproximar uma equação que aumente a região linear de um amplificador de potência.

- **E vários outros tópicos.**

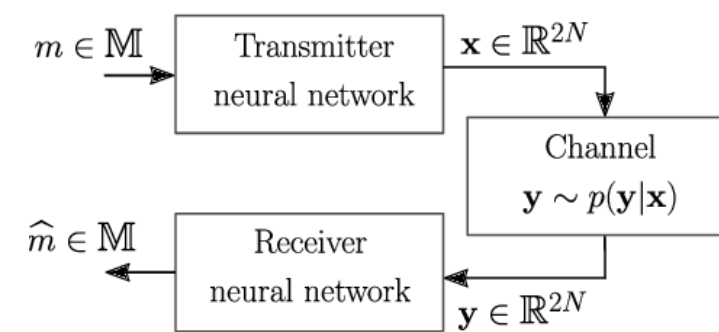
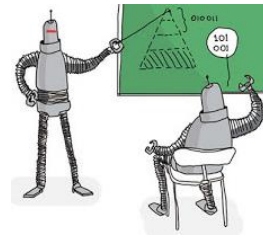


Exemplos interessantes do uso de IA

- **Predição de surtos de doenças infecciosas:** IA previu em Dezembro de 2019 o surto de coronavírus e onde ele apareceria em seguida.
- **Ajuda no diagnóstico de doenças:** IA tem sido usada para ajudar médicos a diagnosticarem o novo coronavírus através de imagens de raio-X e tomografias.
- **Criação de audios, vídeos e fotos falsas:** IA foi usada para criar um vídeo viral utilizando trecho de “A Usurpadora” para falsificar uma conversa entre Lula e Bolsonaro.
- **Criação de novos medicamentos:** IA foi utilizada para criar uma nova droga capaz de combater o transtorno obsessivo-compulsivo (TOC). Encurtou o tempo para encontrar um novo medicamento de 4 para menos de 1 ano.

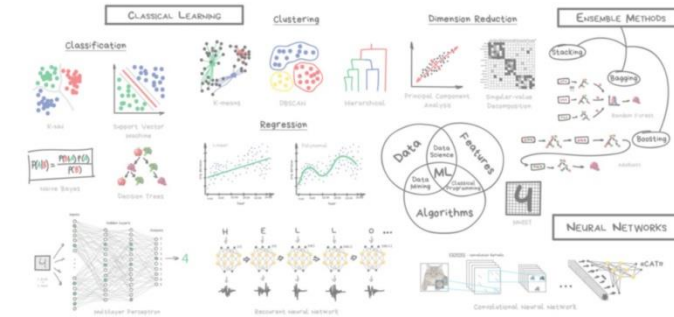


Foco do Curso



- Como vimos, IA é um termo muito amplo, abrangendo várias sub-áreas ou objetivos, que é 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 solução e análise eficiente de vários problemas de telecomunicações e de outras áreas.
 - ***Exemplo: Controle e otimização de parâmetros de sistemas complexos:*** o número de parâmetros que um sistema 5G (PHY, MAC, etc.) precisa controlar e otimizar é muito grande e crescerá mais com o 6G.
 - ❑ Imagine se pudéssemos otimizar ***globalmente*** todos os parâmetros do sistema apenas utilizando dados reais coletados da rede?
 - ❑ Mais ainda, imagine se pudéssemos treinar um modelo que ***aprendesse*** um sistema de transmissão e recepção ótimo para um dado canal?
 - ***Redução de complexidade e custo:*** vários algoritmos de telecomunicações que apresentam desempenho ótimo não são utilizados na prática por possuírem complexidade computacional e/ou custos proibitivos.
 - ***Oportunidades:*** Pesquisas sobre a aplicação de aprendizado de máquina a problemas de telecomunicações ainda estão em fase de exploração. Além de muitos empregos na área de análise de dados.

Mas então, o que é ML?

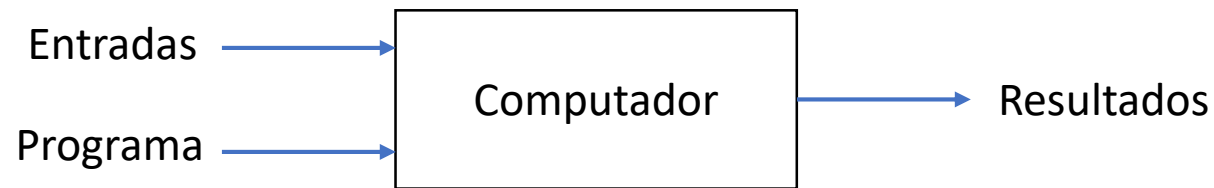


- É uma sub-área ou objetivo da inteligência artificial.
- O termo foi cunhado em 1959, por Arthur Samuel, que o definiu como o “*campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados*”.
- Uma outra definição interessante feita por Jojo John Moolayil é “*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.
- Algoritmos de ML são **orientados a dados**, ou seja, eles aprendem automaticamente um padrão geral (i.e., generalizar) a partir de grandes volumes de dados (i.e., exemplos).
- **Exemplo**: filtro de spam do Gmail.

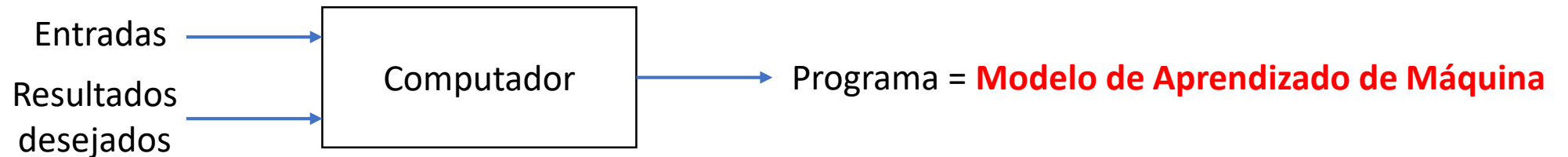
O que é o Aprendizado de Máquina?

- “... sem ser explicitamente programada.”

Programação Tradicional



Aprendizado de Máquina



Por que ML se tornou tão difundido?

Alguns dos principais motivos são:

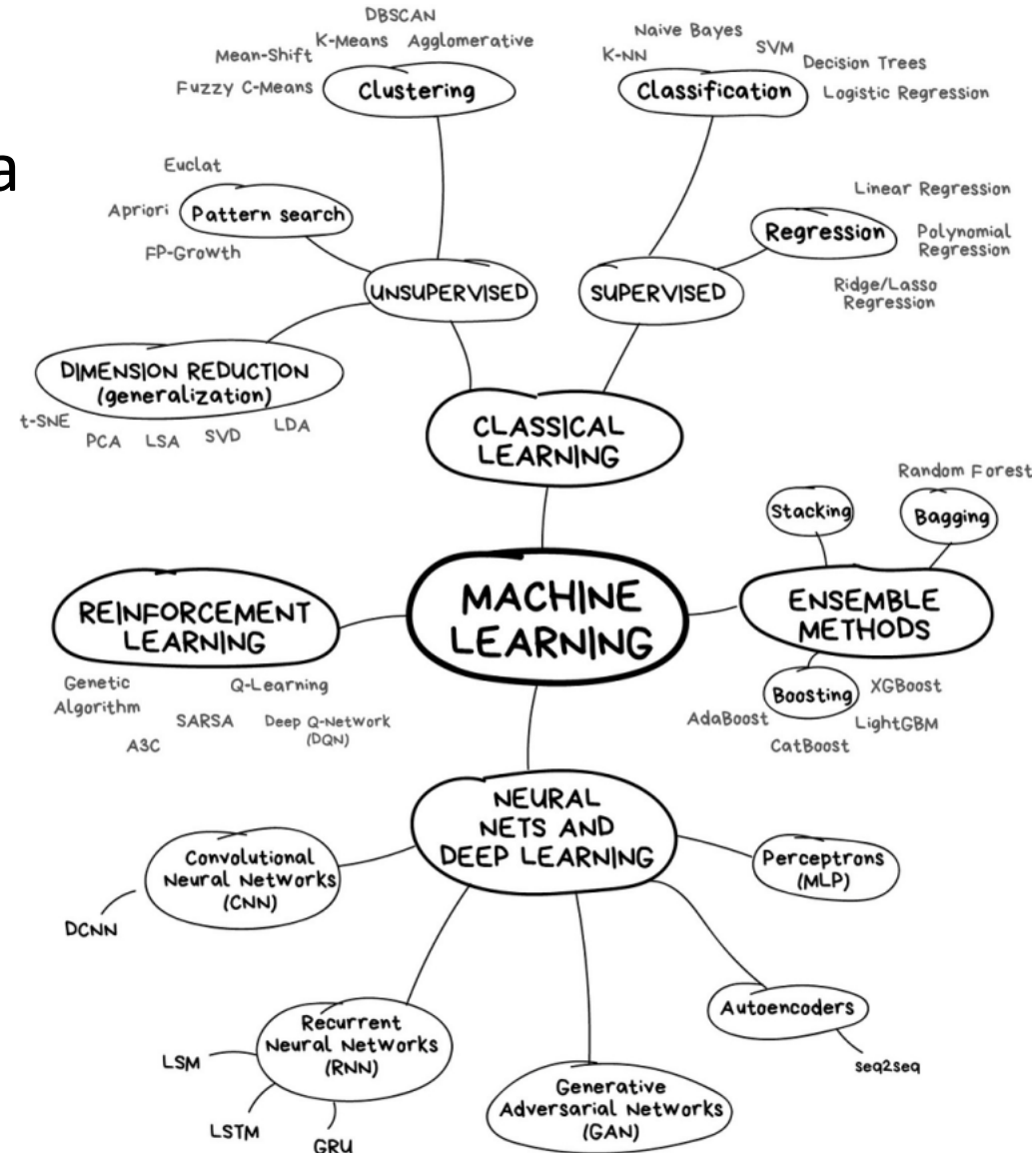
- Vivemos na era da informação. Nessa era, um volume sem precedentes de dados (de tera a petabytes) está disponível, impossibilitando sua análise por nós seres humanos.
- Porém, para modelos de ML isso não é um problema e sim uma solução, pois quanto mais dados melhor será o aprendizado.
- Hoje em dia, dados são preciosíssimos e a extração de novas informações (úteis) vale ouro.
- O surgimento de recursos computacionais poderosos tais como GPUs, FPGAs e CPUs com múltiplos cores.
- Surgimento de novas e eficientes estratégias/técnicas de treinamento (i.e., aprendizagem), e.g., deep-learning, reinforment-learning, etc.
- Existência de frameworks e bibliotecas poderosas que facilitam o desenvolvimento de soluções com ML.



Tipos de Aprendizado de Máquina

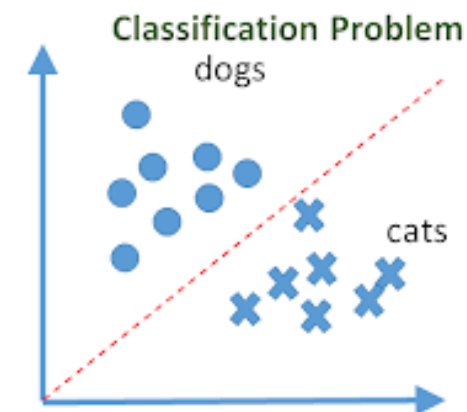
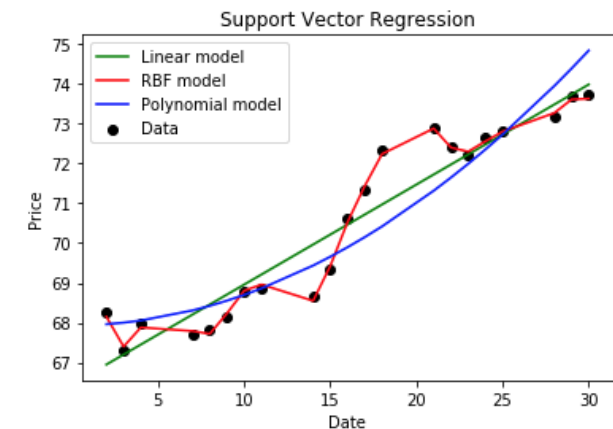
- Os algoritmos de aprendizado de máquina podem ser agrupados nas seguintes categorias:

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



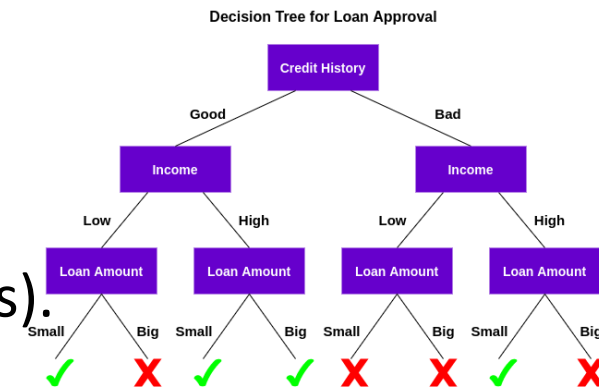
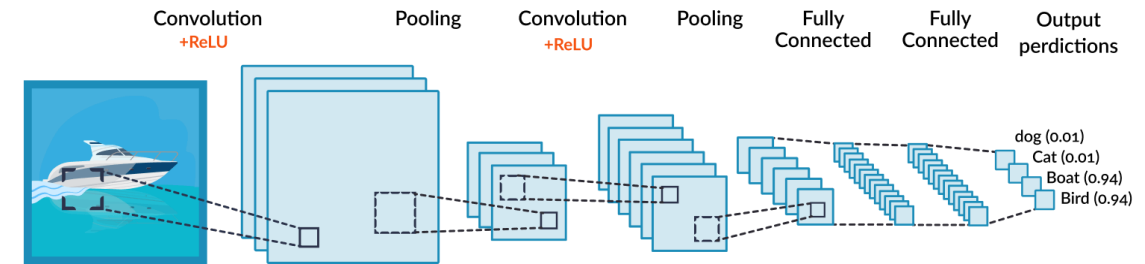
Aprendizado Supervisionado

- No aprendizado supervisionado a máquina sabe o que aprender, ou seja, além das entradas ela tem acesso às respostas esperadas.
- Neste tipo de aprendizado, os dados ou exemplos de treinamento incluem os **atributos**, x , que são a entrada do algoritmo de ML e as **soluções desejadas**, y , (i.e., as respostas corretas), chamadas de **rótulos** (ou *labels*, do Inglês).
- **Tarefa:** os modelos supervisionados de ML devem **aprender** uma **função** que mapeie as entradas x nas saídas y , ou seja, $y = f(x)$.
- Esse tipo de aprendizado pode ser 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.

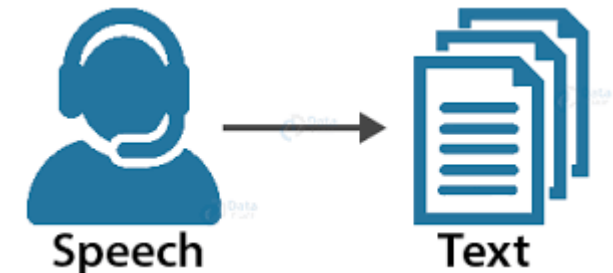
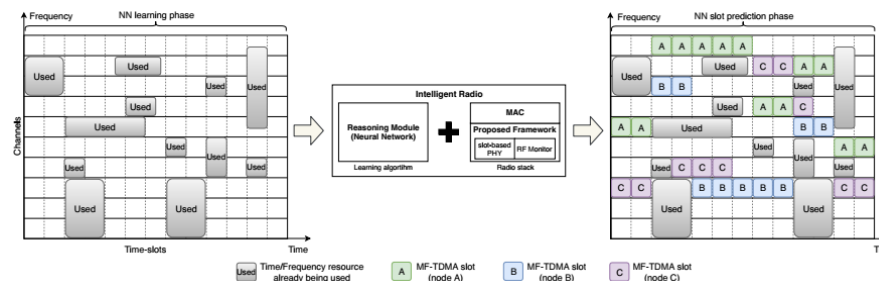
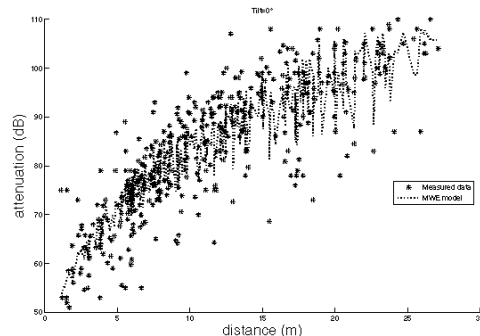


Principais Algoritmos para Aprendizado Supervisionado

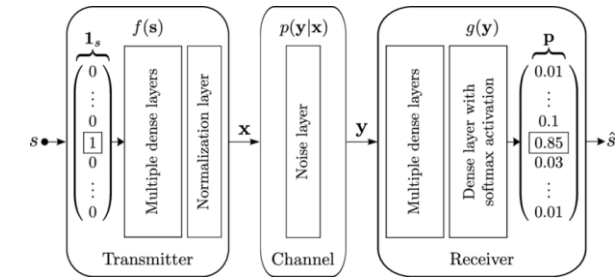
- Regressão Linear.
- Regressão Logística.
- k vizinhos mais próximos (*k-nearest neighbors* - k-NN).
- Árvores de Decisão (*Decision Trees* - DT).
- Florestas Aleatórias (*Random Forests* - RF).
- Máquinas de Vetores de Suporte (*Support Vector Machines* - SVMs).
- Redes Neurais Artificiais
 - Alguns tipos podem ser não-supervisionados, e.g., auto-codificadores. Outros tipos podem ser semi-supervisionados, como as redes de crenças profundas (do Inglês, *Deep Belief Networks*).



Exemplos: predição de perda por caminho e da ocupação do espectro.

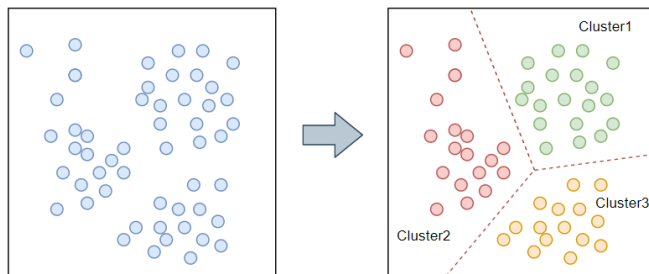


Aprendizado Não-Supervisionado



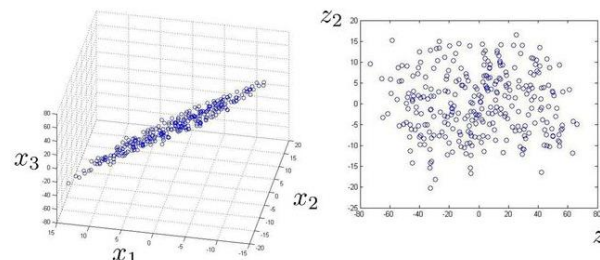
- Neste tipo de aprendizado, as máquinas não são informadas sobre o que aprender. Elas só recebem os exemplos de treinamento, ou seja, os atributos, x .
- Neste caso, os algoritmos ***aprendem/descobrem padrões*** (muitas vezes ocultos) presentes nos dados de entrada sem a presença de rótulos, y .
- **Tarefa:** 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

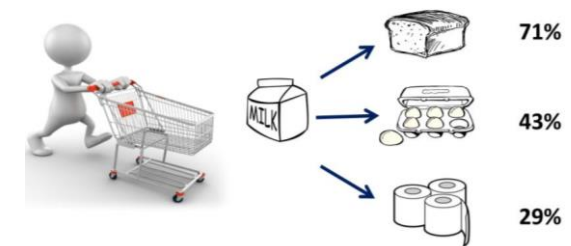
Reduce data from 3D to 2D



Detecção de Anomalias



Regras de associação

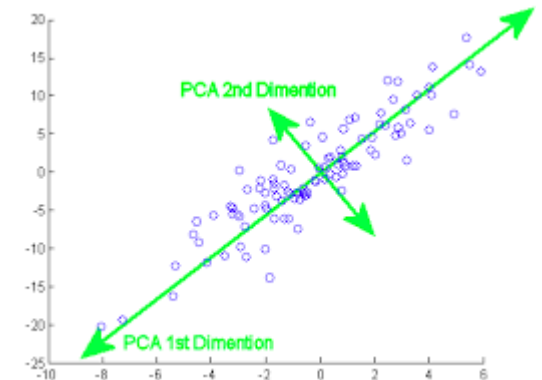
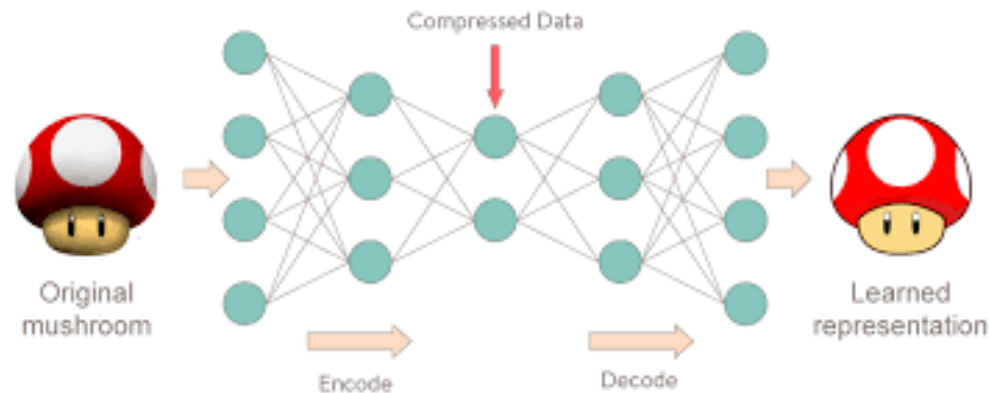
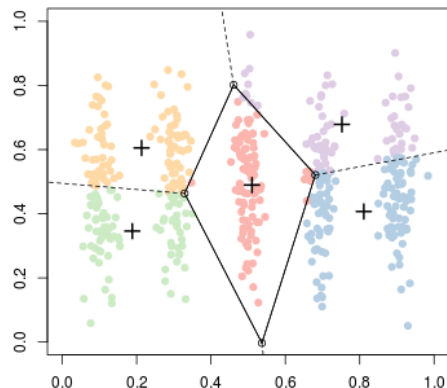


Of transactions that included milk:

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

Principais Algoritmos para Aprendizado Não-Supervisionado

- k-médias (*k-means*).
 - Particiona os atributos em k clusters (ou grupos) distintos com base na distância ao centroide de um cluster.
- 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 tem 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 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.?
- **Clustering** agrupa a quantidade massiva de textos e usamos apenas os exemplos mais representativos de cada **cluster** (quantidade bem menor de textos) para **rotular**.
- Esses exemplos rotulados são usados para treinar um **classificador**.
- Após, o **classificador** classifica automaticamente todos os textos.



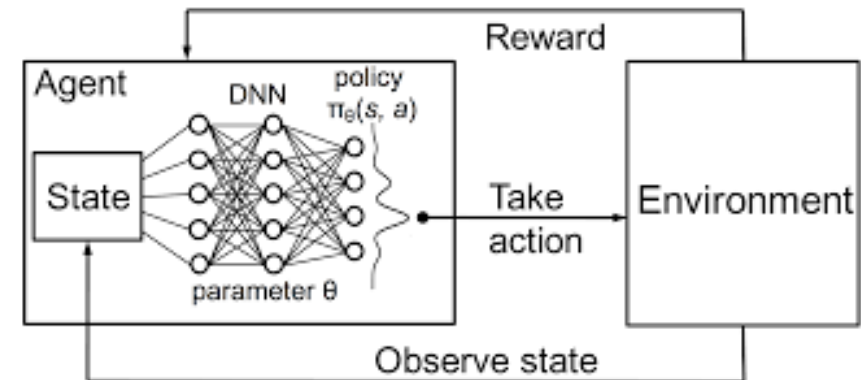
Aprendizado Por Reforço



- Abordagem totalmente diferente das anteriores pois ***não se tem exemplos de treinamento***, sejam eles rotulados ou não.
- O algoritmo de aprendizagem por reforço, chamado de ***agente*** nesse contexto, deve aprender 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 ***recompensas*** (ou ***reforços***) 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, o ***estado*** do ***ambiente***.
- Uma ***política*** é uma função que mapeia os ***estados*** do ***ambiente*** em ***ações*** que o ***agente*** deve tomar para maximizar as recompensas.

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**.
- 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 tabula 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 e muitas vezes sub-ótimas, para um problema.
- **Metaheurísticas** são geralmente aplicadas a problemas para os quais não se conhece um algoritmo eficiente (e.g., problemas NP-completos).
- Características das metaheurísticas:
 - não garantem que uma solução globalmente ótima seja encontrada, mas podem encontrar uma solução suficientemente boa.
 - são estratégias que orientam o processo de busca.
 - 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.
- Otimização 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.

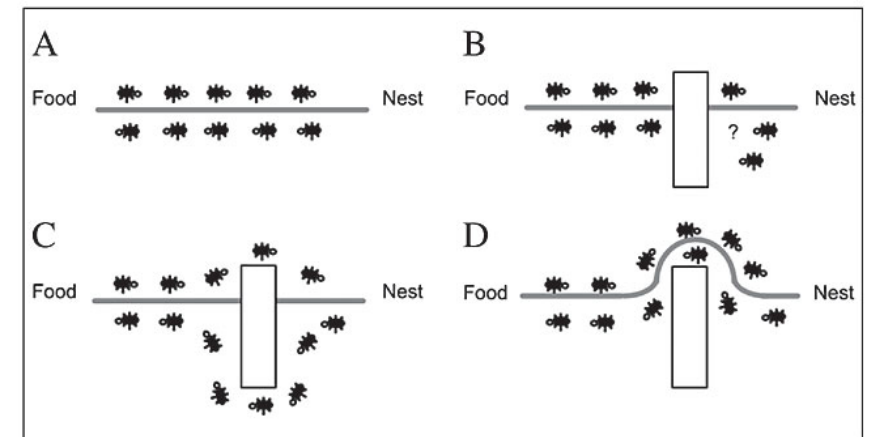
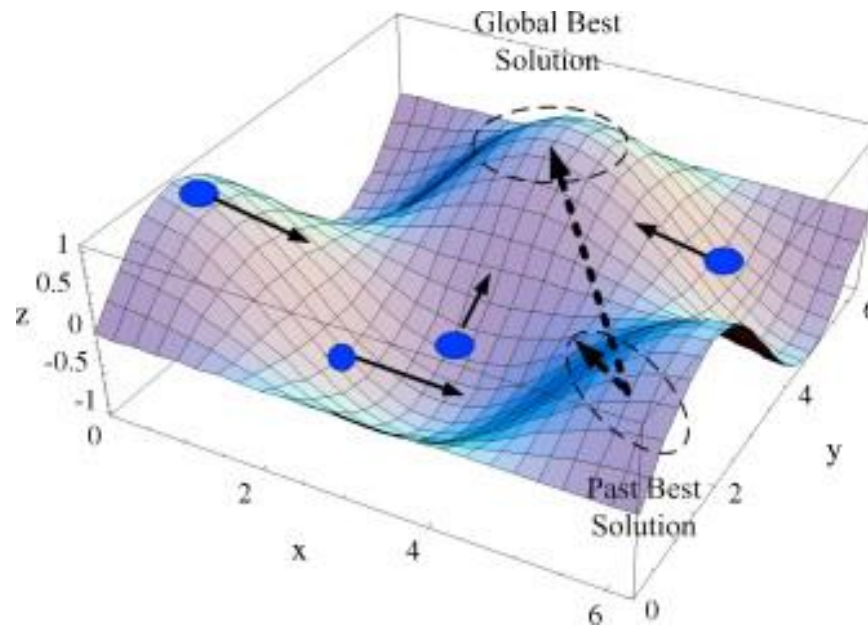
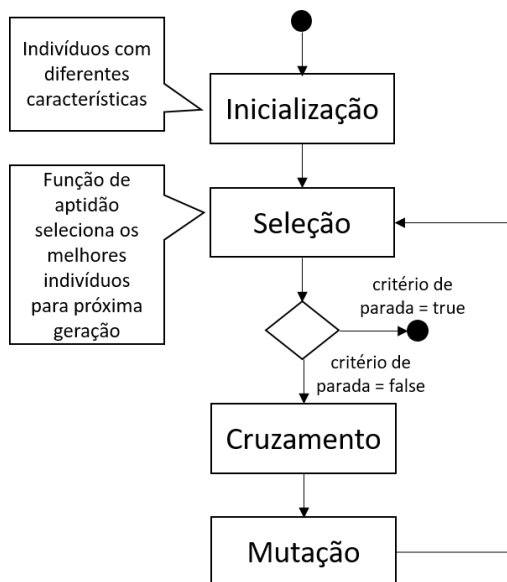


Figure 2. A. Ants in a pheromone trail between nest and food; B. an obstacle interrupts the trail; C. ants find two paths to go around the obstacle; D. a new pheromone trail is formed along the shorter path.

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

Treinamento incremental

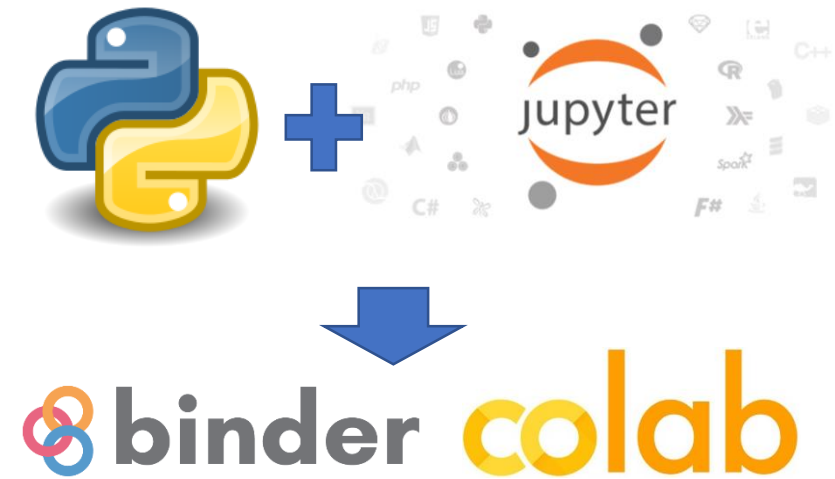
- Neste tipo de treinamento, o algoritmo aprende incrementalmente: exemplos de treinamento são apresentados sequencialmente um-a-um ou em pequenos grupos chamados de mini-batches (ou mini-lotes).
- Cada iteração de treinamento é rápida possibilitando que o sistema aprenda sobre novos dados à medida que eles chegam.
- Ótima opção para casos onde os dados chegam como um fluxo contínuo ou se tem recursos computacionais limitados.
- Entretanto, como não há pré-processamento/análise, 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/analísados, evitando assim, dados corrompidos ou com problemas.
- O treinamento é demorado e utiliza muitos recursos computacionais (e.g., CPU, 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 códigos na nuvem

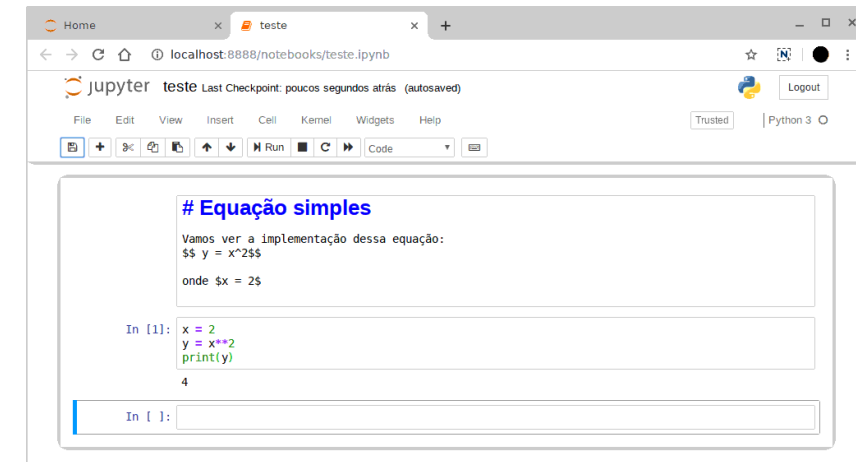
- Durante o curso, usaremos **Python** como linguagem de programação.
 - É uma das linguagens mais fáceis de se aprender.
 - É a linguagem mais usada em aplicações de ML.
 - Possui um rico ecossistema de bibliotecas: SciPy, NumPy, Pandas, Matplotlib, SciKit-Learn, TensorFlow, OpenCV, etc.
 - Além de ser gratuita e open-source.
- Utilizaremos **notebooks Jupyter** para programar.
 - Eles são documentos virtuais usados para criar e compartilhar código juntamente com equações, gráficos e texto.
 - **Notebooks** permitem uma maneira interativa de programar e documentar o código.
- Para executar estes **notebooks**, utilizaremos o **Binder** ou **Google Colaboratory**, que são ambientes computacionais (i.e., servidores) interativos e gratuitos.
- Portanto, não é **mandatório** se instalar nada, apenas ter um navegador web e conexão com a internet.



Binder



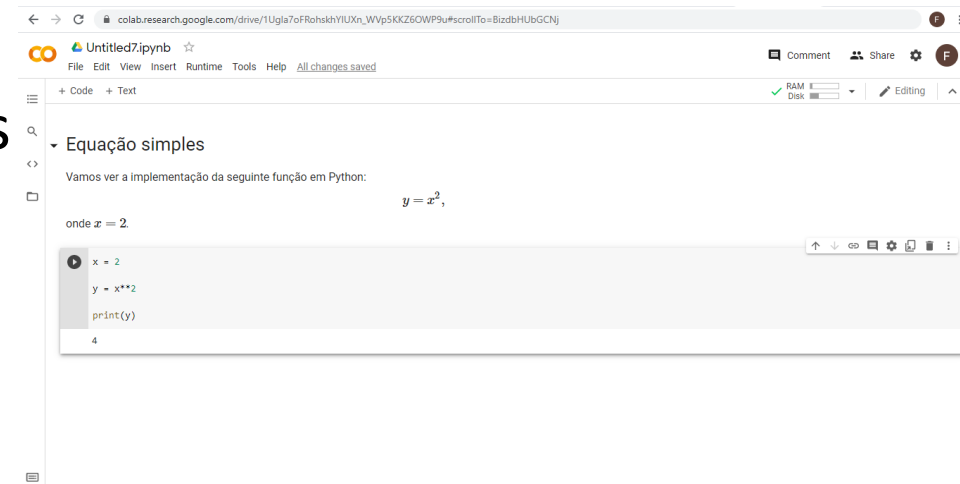
- **Binder**: aplicação web gratuita que permite a criação e edição de **notebooks** em navegadores web.
- Suporta a execução de várias linguagens de programação: Python, C++, C#, PHP, Julia, R, etc.
- Algumas desvantagens do **Jupyter** são:
 - Poucos servidores disponíveis.
 - Depois de algum tempo inativo, a máquina virtual executando seu **notebook** se desconecta e você pode perder seu código.
- URL (através do Jupyter): <https://jupyter.org/>
- A lista de exercícios #0 contém um link para um tutorial de instalação do Jupyter.



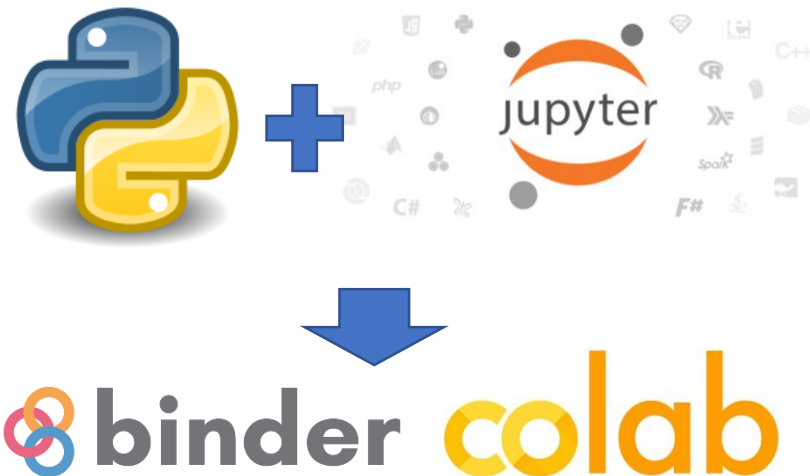
Google Colaboratory (Colab)



- **Colab**: outra aplicação web gratuita, baseada no Jupyter, que permite a criação e edição de *notebooks* 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.
 - Fornece GPUs e TPUs gratuitamente.
 - Compartilhamento de notebooks entre usuários é mais fácil.
 - Notebooks podem ser salvos no seu Google Drive, evitando que você perca seu código.
- URL: <https://colab.research.google.com/>



Exemplos de uso dos notebooks com Binder e Colab



Histogramas

```
import numpy as np
import matplotlib.pyplot as plt

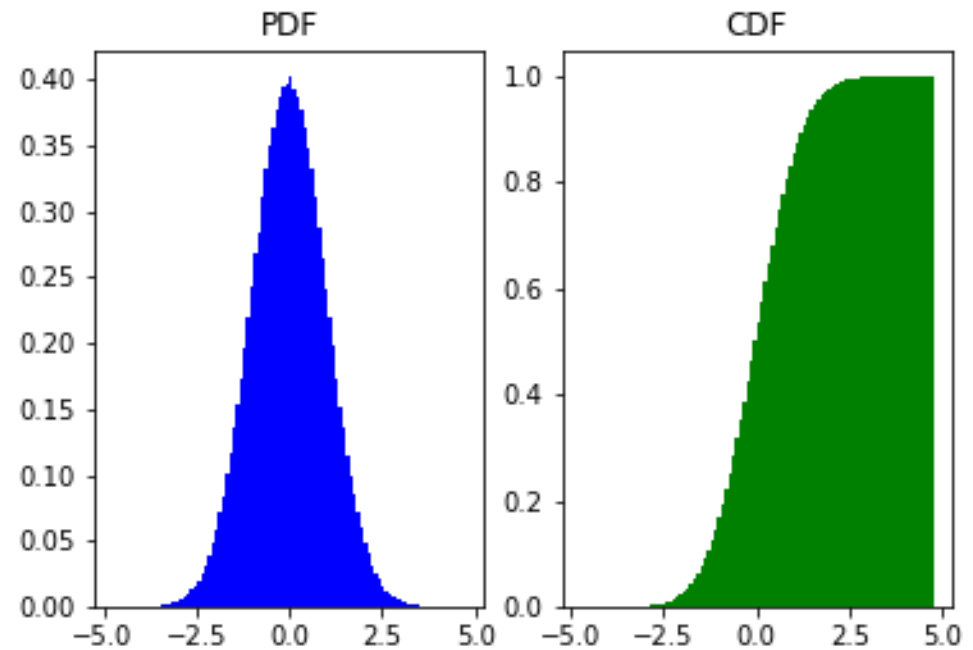
%matplotlib inline

data = np.random.randn(1000000)

# histograma (pdf)
plt.subplot(1, 2, 1)
plt.title('PDF')
plt.hist(data, bins=100, normed=True, color='b')

# CDF empirica
plt.subplot(1, 2, 2)
plt.title('CDF')
plt.hist(data, bins=100, normed=True, color='g', cumulative=True)

plt.savefig('histogram.png') # salva figura em arquivo
```



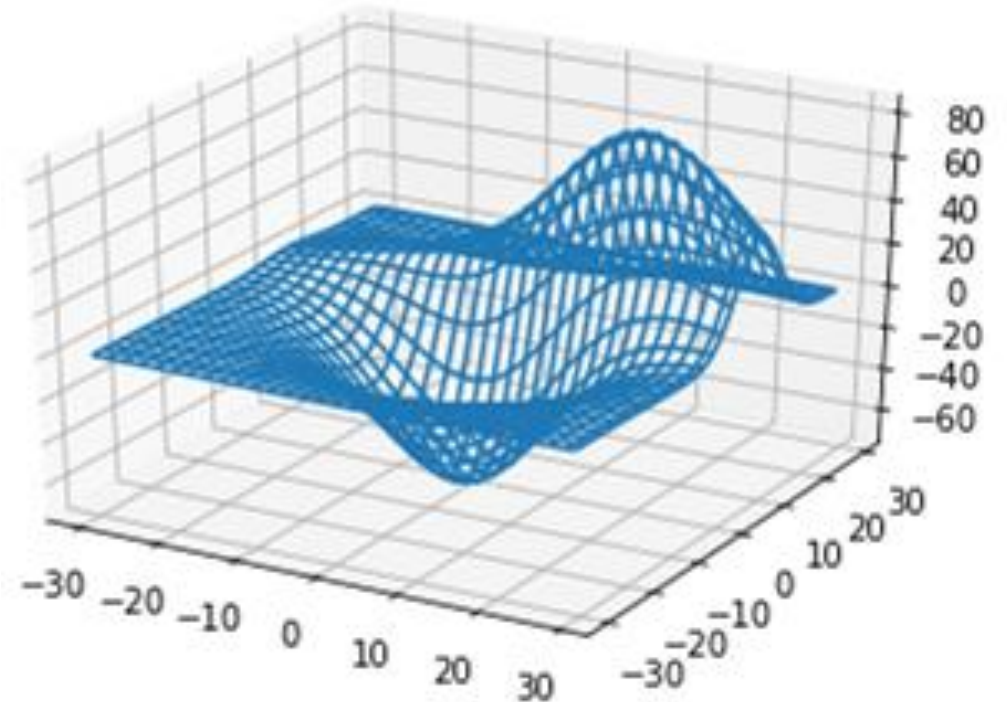
Figuras 3D

```
import matplotlib.pyplot as plt
# facilita visualizacao de figuras 3D
from mpl_toolkits.mplot3d import axes3d # graficos 3D sao habilitados importando axes3d

# para figuras interativas usar "notebook" ao inves de "inline"
%matplotlib notebook

ax = plt.subplot(111, projection='3d')
X, Y, Z = axes3d.get_test_data(0.1)
ax.plot_wireframe(X, Y, Z)

# salva figura em arquivo
plt.savefig('figura3d.png')
```



Ajuste de Curvas com Redes Neurais

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neural_network import MLPRegressor # importa classe MLPRegressor do modulo neural network

%matplotlib inline

x = np.arange(-10, 10, 0.1)

# dados originais
y = 12 + 3 * np.exp(-0.05*x) + 1.4 * np.sin(1.2*x) + 2.1 * np.sin(-2.2*x + 3)

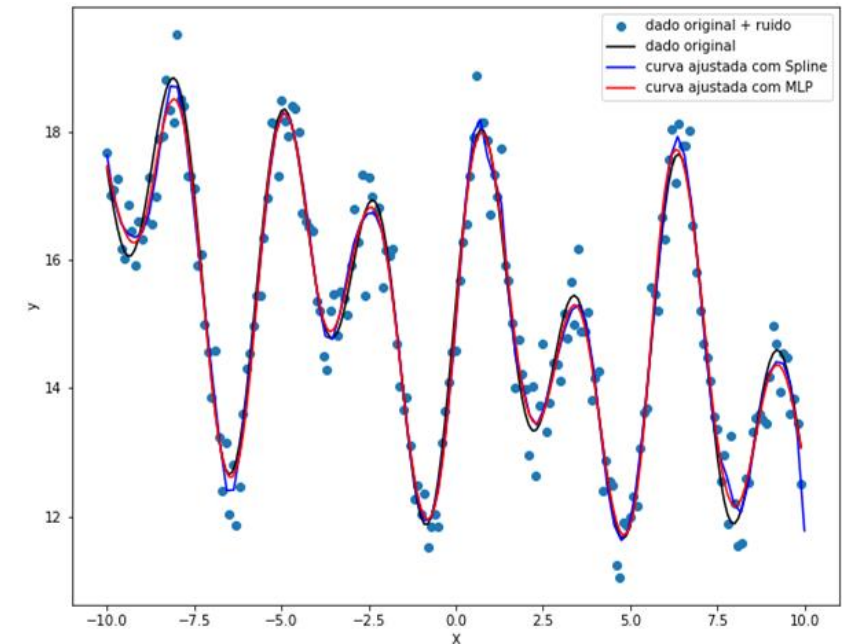
# faz com que o gerador de numeros aleatorios sempre forneça os mesmos valores
np.random.seed(42)

# adicionando ruido aos dados originais
y_noise = y + np.random.normal(0, 0.5, size = len(y))

# trata o ajuste de curva como um problema de regressao e treina um modelo para que se ajuste aos dados.
mlp = MLPRegressor(hidden_layer_sizes=(30,20,10), max_iter=10000, solver='lbfgs', alpha=0.9, activation='tanh')
yfit = mlp.fit(x[:, None], y_noise).predict(x[:, None])

plt.figure(figsize = (10,8))
plt.plot(x, y_noise, 'o', label = 'dados original + ruido')
plt.plot(x, y, 'k', label = 'dados original')
plt.plot(x, yfit, '-r', label = 'curva ajustada com MLP', zorder = 10)
plt.legend()
plt.xlabel('X')
plt.ylabel('y')

# salva figura em arquivo
plt.savefig(mlp_regression.png)
```



Referências

- [1] Athanasios Papoulis and S. Pillai, “Probability - Random Variables and Stochastic Processes,” McGraw Hill Education; 4th ed., 2017.
- [2] 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/1lylIMu1w6POBhrVnw11yqXXy6BjC439j?usp=s_haring

Avisos

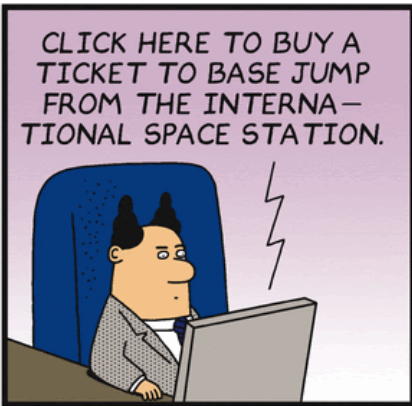
- Lista #0, #1 e descrição do [projeto prático](#) já estão disponíveis no MS Teams.
- Todas as entregas devem ser feitas tanto no GitHub quanto no MS Teams.
 - Meu controle de entregas será via MS Teams e a validação das atividades será feita através do GitHub.
 - Se atentem às datas de entrega no MS Teams.
- Todo material do curso será disponibilizado no MS Teams.
- Horário de Atendimento
 - Todas as Quintas-Feiras das 13:30 às 15:30 via MS Teams.

Perguntas?

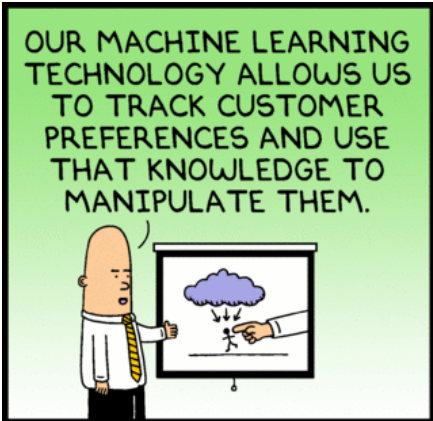
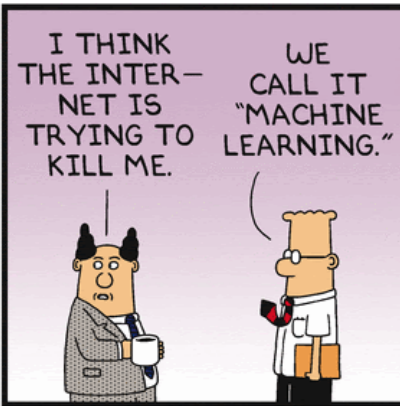
Obrigado!



Dilbert.com DilbertCartoonist@gmail.com



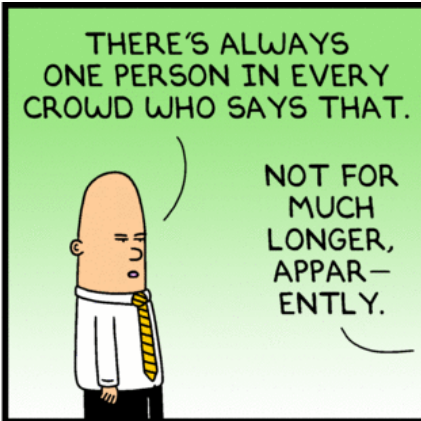
2-2-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick



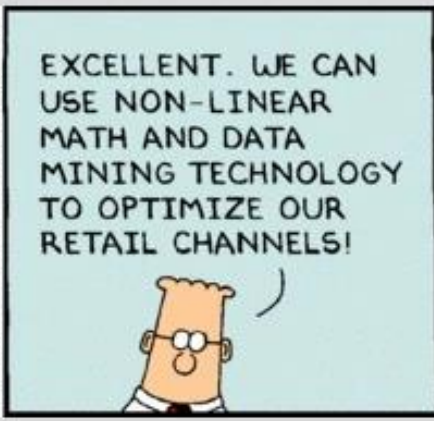
Dilbert.com DilbertCartoonist@gmail.com



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



www.dilbert.com scottadams@aol.com



©2003 United Feature Syndicate, Inc.



