

TP555 - Inteligência Artificial e Machine Learning: *Introdução*



Inatel

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

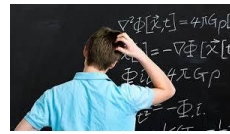
Objetivos do Curso

- Apresentar os **principais algoritmos de aprendizado de máquina** ou do inglês, machine learning (ML).
- Portanto, o foco principal do curso é em **Machine Learning**, o qual dará a vocês um conjunto de ferramentas úteis para suas pesquisas.
- 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.
 - Compreender artigos científicos que utilizam ML.
 - Realizar pesquisa que envolva ML e discutir os resultados obtidos.



Avaliação do Curso

- 2 Provas
 - Cada prova valerá 35% da nota (datas ainda serão definidas).
- Listas de Exercícios
 - 10% da nota.
 - Listas serão disponibilizadas ao final de cada tópico.
 - [Devem ser comitadas no GitHub e entregues no MS Teams.](#)
- Projeto Prático
 - 20% da nota.
 - Tema será escolhido por vocês.
 - [Link para o documento do projeto.](#)



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/lucros.



Referências:

[1] Matti Latva-aho ad Kari Leppänen (editors),
KEY DRIVERS AND RESEARCH CHALLENGES FOR 6G UBIQUITOUS WIRELESS

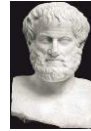
INTELLIGENCE, 6G Flagship, University of Oulu, Oulu,
disponível online em:

<http://jultika.oulu.fi/files/isbn9789526223544.pdf>

[2] Mostafa Zaman Chowdhury, Md. Shahjalal, Shakil Ahmed, Yeong Min Jang, “6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions”,
<https://arxiv.org/ftp/arxiv/papers/1909/1909.11315.pdf>

Um Pouco de História

- A ideia de criar uma máquina pensante ou uma criatura artificial que imitasse as habilidades humanas é muito antiga, remontando à Grécia antiga.
- **Mitologia grega:** Talos, um autômato (máquina que realiza ações que lembram humanos ou animais) criado para proteger a princesa Europa de Creta de invasores.
- **300 A.C.:** Aristóteles almejava substituir a mão-de-obra escrava por objetos autônomos.
- **1642:** Pascal constrói a primeira calculadora mecânica do mundo, a *La pascaline*.
- **1822:** Charles Babbage cria os primeiros computadores mecânicos, as *máquinas diferencial e analítica*, se tornando o “pai dos computadores”.
- **1837:** Ada Lovelace escreve o primeiro programa de computador para a *máquina analítica* de Babbage, se tornando a primeira “programadora” conhecida.



- A ideia de construir uma máquina pensante ou uma criatura artificial que imitasse as habilidades humanas é muito antiga, remontando à Grécia antiga.
- Na mitologia grega, Talos, era um autômato (máquinas que realizam ações que lembram humanos ou animais) criado para proteger a princesa Europa de Creta.
- A ideia de IA como a conhecemos hoje só foi possível após o surgimento dos computadores modernos, ou seja, após a Segunda Guerra Mundial.
- A área de pesquisa em IA surgiu em 1956 em um workshop no Dartmouth College nos EUA onde o termo IA foi cunhado por John McCarthy.
- IA tem como objetivo criar máquinas que *imitem* nossa capacidade mental para uma determinada tarefa. Porém, esta *imitação* não será perfeita dado que somos construídos de matéria viva e até o

momento não fomos capazes de criar nada semelhante ao nosso cérebro. Essa imitação é, portanto, apenas uma aproximação. É por isso que em IA fala-se da criação de máquinas que são modelos de nossa capacidade de aprender, raciocinar, enxergar, falar, etc.

- Embora o GPS fosse capaz de resolver problemas simples, como o problema das torres de Hanói que poderia ser *suficientemente formalizado*, ele não poderia resolver todos os problemas do mundo real, porque a busca facilmente se perderia em uma explosão combinatória de estados intermediários. **Problemas que podem ser formalizados.**

Artificial Intelligence (AI) - the broad discipline of creating intelligent machines.

Machine Learning (ML) - refers to systems that can learn from experience.

Difference: Artificial Intelligence is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”.

On the other hand, Machine Learning is a current application of AI based around the idea that machines should learn for themselves by only having access to data. In ***Machine learning***, algorithms acquire the knowledge or skill through experience. Machine learning relies on big data sets to find common patterns.

In summary, machine learning uses the experience to learn a pattern (induced learning). AI uses the experience to acquire knowledge/skill and also how to apply that knowledge for new environments.

First of all, **leave the terms artificial and machine aside**. We will try to understand the words intelligence and learning one-by-one. Think of how one can solve a given (mathematical and not real life) problem. There are two possibilities - *either*

Method 1: he/she/it has solved such a problem before and has ***learned*** how to solve such kind of problems (learned the pattern) *or*

Method 2: (it is first time he/she/it faced this type of problem and) he/she/it is **actually intelligent** to think of possibilities of how to solve the problem based on what he/she/it **knows**, right?

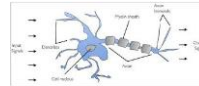
The former is called **learning** and the latter is called **knowledge** (along with intelligence). So, in short, to solve a problem, one needs either of these two. Hence, **learning is a part of intelligence**.

AI is a broader concept (umbrella term) to create intelligent machines that can simulate human thinking capability and behavior, whereas, machine learning is an application or subset of AI that allows machines to learn from data without being programmed explicitly.

Codebreaker: alguém que decodifica mensagens encriptadas sem a chave de criptografia.

Um Pouco de História (continuação)

- **1847:** George Boole cria a lógica booleana, lançando as bases para a era da informação.
- **1943:** Alan Turing e um grupo de codebreakers constroem Colossus, o primeiro computador programável, digital e eletrônico do mundo.
- **1945:** ENIAC, o primeiro computador de uso-geral é construído.
- **1943:** McCulloch e Pitts propõem um modelo matemático para o funcionamento de um neurônio. Lançando as bases para a criação das redes neurais artificiais.
- **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 (*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.



Colossus foi um conjunto de computadores desenvolvidos por codebreakers britânicos nos anos de 1943 a 1945 para ajudar na análise de mensagens criptografadas.

Assia ao filme: The Imitation Game

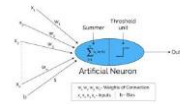
Embora o ENIAC tenha sido projetado e usado principalmente para calcular tabelas de tiro de artilharia para o Laboratório de Pesquisa Balística do Exército dos Estados Unidos (que mais tarde se tornou parte do Laboratório de Pesquisa do Exército), seu primeiro programa foi um estudo da viabilidade de armas termonucleares.

“Pela primeira vez no mundo, computador engana pessoas e passa no teste de Turing”,
<https://canaltech.com.br/ciencia/Pela-primeira-vez-na-mundo-computador-engana-pessoas-e-passa-no-teste-de-Turing/>

Um Pouco de História (continuação)



- **1957:** Newell e Simon criam o *Solucionador de Problemas Gerais*, um programa para resolução de problemas (e.g., provas de teoremas, problemas de geometria, jogo de xadrez, etc.)
- **1958:** John McCarthy cria a linguagem *Lisp*, que foi a linguagem para desenvolvimento de IA dominante até os anos 90.
- **1958:** Frank Rosenblatt cria a primeira rede neural artificial, chamada de *Perceptron*.
- **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: redes neurais retornam a popularidade e grandes avanços em algoritmos e aplicações de ML.



1986 – Computadores se tornam mais poderosos devido a lei de Moore e pesquisadores re-descobrem as redes neurais.

Um Pouco de História (continuação)

- **1997:** O supercomputador da IBM, chamado de *DeepBlue*, vence o campeão mundial de xadrez, Garry Kasparov.
- **2002:** O primeiro aspirador de pó robótico, chamado de *Roomba*, é lançado pela *iRobot*.
- **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.



Inatel

Inteligência + Artificial (IA)



- Vocês já refletiram sobre a definição destas duas palavras?
- **Inteligência:** são as **habilidades mentais** de **conhecer, compreender, aprender, resolver problemas e adaptar-se**.
 - **Latim** -> prefixo *inter*: "entre" + sufixo *legere*: "escolha" => entre escolhas.
 - **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.

Habilidades mentais desenvolvidas de forma não natural pela mão do homem.

Cópias de habilidades mentais humanas criadas de forma não natural pelo homem.



- **Inteligência:** é um conjunto que forma todas as **características intelectuais** de um indivíduo, ou seja, a faculdade de conhecer, compreender e aprender. Capacidade de compreender e resolver novos problemas e conflitos e de adaptar-se a novas situações. É a capacidade de alguém para lógica, abstração, memorização, compreensão, autoconhecimento, comunicação, aprendizado, controle emocional, planejamento e resolução de problemas.
- **Conhecer:** perceber e incorporar à memória (algo); ficar sabendo.
- Etimologicamente, a palavra **inteligência** se originou a partir do latim *intelligentia*, oriundo de *intelligere*, em que o prefixo *inter* significa "entre", e *legere* quer dizer "escolha". Assim sendo, o significado original deste termo faz referência a **capacidade de escolha** de um indivíduo entre as várias possibilidades ou opções que lhe são apresentadas.

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 objetivos específicos por meio de adaptação flexível." (Andreas Kaplan).
- **Definição:** "Ciência e engenharia de produzir máquinas inteligentes." (John McCarthy).
- **Objetivo:** Criar máquinas que *imitem* nossa capacidade mental para uma determinada tarefa.
- Essa *imitação* é apenas uma aproximação pois ainda não conseguimos criar matéria viva.
- É por isso que em IA fala-se da criação de máquinas que são **modelos aproximados** de nossa capacidade de aprender, raciocinar, enxergar, falar, ouvir, etc.

- **Inteligência:** é a capacidade de alguém para lógica, abstração, memorização, compreensão, autoconhecimento, comunicação, aprendizado, controle emocional, planejamento e resolução de problemas, etc. Etimologicamente, a palavra "inteligência" se originou a partir do latim *intelligentia*, oriundo de *intelligere*, em que o prefixo *inter* significa "entre", e *legere* quer dizer "escolha". Assim sendo, o significado original deste termo faz referência a capacidade de escolha de um indivíduo entre as várias possibilidades ou opções que lhe são apresentadas.
- **Artificial:** feito com arte, produzido pela mão do homem, não pela natureza.
- **Objetivo:** Criar máquinas que *imitem* nossa capacidade mental para uma determinada tarefa.
- Porém, esta *imitação* não será perfeita dado que somos construídos de matéria viva e até o momento não fomos capazes de criar nada semelhante ao nosso cérebro.
- Essa *imitação* é, portanto, apenas uma aproximação. É por isso que em IA fala-se da criação de máquinas que são *modelos* de nossa capacidade de aprender, raciocinar, enxergar, falar, ouvir, etc.

First of all, **leave the terms artificial and machine aside**. We will try to understand the words intelligence and learning one-by-one. Think of how one can solve a given

(mathematical and not real life) problem. There are two possibilities - *either*

Method 1: he/she/it has solved such a problem before and has ***learned*** how to solve such problems *or*

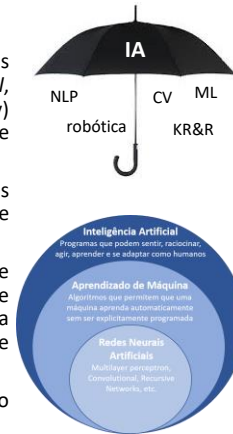
Method 2: (it is first time he/she/it faced this type of problem and) he/she/it is ***actually intelligent*** to think of possibilities of how to solve the problem based on what he/she/it ***knows***, right?

The former is called **learning** and the latter is called **knowledge** (along with intelligence). So, in short, to solve a problem, one needs either of these two. Hence, **learning is a part of intelligence.**

AI is a broader concept to create intelligent machines that can simulate human thinking capability and behavior, whereas, machine learning is an application or subset of AI that allows machines to learn from data without being programmed explicitly.

Inteligência Artificial

- IA é uma área muito ampla que **engloba** várias aplicações (ou subáreas ou ainda objetivos) tais como (i) *processamento de linguagem natural*, (ii) *representação do conhecimento*, (iii) *raciocínio automatizado*, (iv) *visão computacional*, (v) *robótica*, e (vi) *aprendizado de máquina* que por sua vez engloba redes neurais artificiais, etc.
- Aprendizado** é apenas um dos aspectos da inteligência humana (outros são: lógica, abstração, memorização, planejamento, resolução de problemas, etc.)
- Exemplo:** Um problema matemático pode ser solucionado (i) caso ele tenha sido resolvido antes e sua solução **aprendida** ou (ii) caso ele nunca tenha sido visto antes mas um indivíduo com inteligência imagina/pensa possíveis maneiras de como resolve-lo baseado no que ele **sabe (conhecimento)**.
- IA utiliza a **experiência** para adquirir **conhecimento** e também como aplicar esse conhecimento a problemas desconhecidos.



Processamento de linguagem natural: criação automática de resumos, tradução de textos em uma língua para outra, reconhecimento de fala, etc.

Representação do conhecimento: lida com problemas de como criar e armazenar conhecimento do mundo. É um campo da inteligência artificial que se concentra no design de representações computacionais que capturam informações sobre o mundo.

Raciocínio automatizado: utiliza o conhecimento armazenado por computadores para resolver problemas complexos, como por exemplo, provar teoremas matemáticos.

Visão computacional: desenvolvimento de máquinas inteligentes que obtém informação de imagens. Exemplos: reconhecimento de faces, controle de qualidade em indústrias, carros autônomos, etc.

Robótica: lida com o design (projeto), construção e operação de robôs que repliquem ações humanas. Exemplos: robôs militares, industriais, médicos, etc.

Aprendizado de máquina: lida com o design e construção de máquinas que executam uma tarefa específica sem terem sido explicitamente programadas para isso. Exemplos: algoritmo de recomendações do netflix, carro autômato, etc. Esses algoritmos aprendem através da experiência.

- Inteligência:** é a capacidade de alguém para lógica, abstração, memorização, compreensão, autoconhecimento, comunicação, aprendizado, controle emocional, planejamento e resolução de problemas. Etimologicamente, a palavra "inteligência" se originou a partir do latim *intelligentia*, oriundo de *intelligere*, em que o prefixo *inter* significa "entre", e *legere* quer dizer "escolha". Assim sendo, o significado original deste termo faz referência a capacidade de escolha de um indivíduo entre as várias possibilidades ou opções que lhe são apresentadas.
- Artificial:** feito com arte, produzido pela mão do homem, não pela natureza.

- **Objetivo:** Criar máquinas que *imitem* nossa capacidade mental para uma determinada tarefa.
- Porém, esta *imitação* não será perfeita dado que somos construídos de matéria viva e até o momento não fomos capazes de criar nada semelhante ao nosso cérebro.
- Essa *imitação* é, portanto, apenas uma aproximação. É por isso que em IA fala-se da criação de máquinas que são *modelos* de nossa capacidade de aprender, raciocinar, enxergar, falar, ouvir, etc.

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, 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 respondendo 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.).



Recomendação de anúncios/produtos: IA extrai conhecimento através do comportamento dos clientes e com isso consegue recomendar produtos específicos/customizados para os clientes.

“Faculdade particular usa robô para corrigir provas e dar nota aos alunos”,
<https://cartacampinas.com.br/2020/04/faculdade-particular-usa-robo-para-corriger-provas-e-dar-nota-aos-alunos/>

“Artificial Intelligence May Be Key to Better Weather Forecasts”,
<https://eos.org/opinions/artificial-intelligence-may-be-key-to-better-we>




“AI improves fraud detection, prediction and prevention”,
<https://www.ibm.com/analytics/fraud-prediction-weather-forecasts>

“Inteligência artificial ajuda em diagnóstico da covid-19 no Brasil”,
<https://www.correiobraziliense.com.br/app/noticia/ciencia-e-saude/2020/08/04/interna-ciencia-saude,878353/inteligencia-artificial-ajuda-em-diagnostico-da-covid-19-no-brasil.shtml>

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 Sistemas de Comunicação de Ponta-a-Ponta**
 - 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.**

Fonte: <https://mlc.committees.comsoc.org/research-library/>

- **Estimação de canal e detecção de sinais**
 - Algoritmos de IA são treinados para realizar a estimação de canal e detecção de símbolos conjuntamente. Em alguns exemplos, os algoritmos são treinados para estimar o canal sem o envio de pilotos, aumentando a eficiência espectral do sistema, que se transforma em maior velocidade para o usuário.
 - **Vantagens:** As tarefas de estimação de canal e detecção envolvem a estimação dos vários canais (um por antena) e a inversão de matrizes, e por exemplo, em sistemas MIMO com um número massivo de antenas isso pode levar a um alto custo e alta complexidade computacional. O uso de IA nesta tarefa tem o potencial reduzir custos e complexidade computacional. Além disso, a estimação de canal envolve a transmissão de pilotos, que consomem recursos que poderiam ser utilizados para transmissão de dados. IA tem também o potencial de aumentar a eficiência do sistema através do envio de um menor número de ou nenhum pilotos.
- **Aprendizado de Sistemas de Comunicação de Ponta-a-Ponta**
 - Os blocos de processamento em sistemas de comunicação (transmissor e receptor) são desenvolvidos separadamente e otimizados com premissas (suposições) e objetivos diferentes, tornando a otimização global do sistema muito difícil. AI pode ser usado para modelar e otimizar conjuntamente os blocos de processamento do transmissor (codificação de canal, modulação, MIMO, etc) e do receptor (estimação, detecção, demodulação, decodificação de canal).
 - **Vantagens:** maior performance do sistema dado que ele é otimizado conjuntamente.
- **Alocação de Recursos**

- Dispositivos IoT de baixa potência (e.g., LoRa, Zigbee, 6LoWPAN) não utilizam protocolos de contenção, o que acarreta em colisões, consequentemente diminuindo a performance da rede e drenando as baterias dos dispositivos devido ao maior número de retransmissões. IA pode aprender como (onde-> canal, quando-> periodicidade e por quanto tempo) os outros dispositivos estão acessando o meio e consequentemente evitar colisões com uma certa probabilidade.
- **Vantagens:** aumento da performance da rede e da vida útil das baterias devido ao menor número de retransmissões.
- **Rádio-sobre-Fibra**
 - A transmissão analógica de sinais banda base para BS usando rádio-sobre-fibra pode reduzir os custos de conversores quando comparado com soluções digitais tais como o CPRI, entretanto, o canal composto Radio/Fibra introduz vários tipos de artefatos e distorções ao sinal transmitido que precisam ser mitigadas. AI pode ser utilizada para equalizar e detectar sinais conjuntamente.
 - **Vantagens:** Menor custo e complexidade computacional.
- **Comunicações em Ondas Milimétricas**
 - A comunicação na faixa de ondas milimétricas é predominantemente feita através de links com visada direta (LoS – Line of Sight) e portanto é facilmente susceptível a bloqueios. Um dispositivo móvel quando tem seu link com uma BS bloqueado inicia o que é chamado de processo de handover (procura por outra BS), isso leva a desconexão de sua sessão de comunicação, o que impacta na confiabilidade e aumenta a latência. IA pode ser usada para prever que um link irá em breve ficar bloqueado, o que possibilita que a BS servindo o dispositivo faça um handover proativo (em background) deste dispositivo para outra BS.
 - **Vantagens:** A predição e handover proativo garantem alta confiabilidade e baixa latência nos sistemas móveis.
- **Pré-distorção**
 - Amplificadores de potencia são inerentemente não-lineares quando operam em regimes de alta potência, o que introduz distorção dentro da banda desejada e produtos de intermodulação fora da banda desejada (ou seja, sinal é “*espirrado*” pra fora da banda).

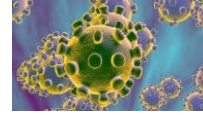
- O dispositivo de pré-distorção introduz uma “*distorção inversa*” na entrada do amplificador, o que cancela, em teoria, qualquer não-linearidade que o amplificador possa apresentar.
- IA tem o potencial para aprender uma curva pré-distorção capaz de mitigar os efeitos não-lineares dos amplificadores de RF.
- **Vantagens:** O emprego de IA tem potencial para diminuir os custos e complexidade computacional dos dispositivos de pré-distorção.

Fonte: <https://mlc.committees.comsoc.org/research-library/>

Goodput: é o throughput no nível de aplicação de um sistema de comunicação; isto é, o número de bits de informação úteis entregues pelo sistema a um determinado destino por unidade de tempo.

Exemplos interessantes de uso da IA


- **Predição de surtos de doenças infecciosas:** IA previu em Dezembro de 2019 o novo surto de corona vírus 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 vídeos/fotos falsas:** IA foi usada para criar 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 novos medicamentos de 4 para menos de 1 ano.

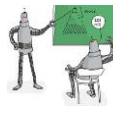


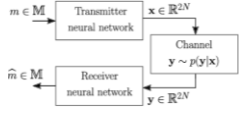
- **Predição de surtos de doenças infecciosas:** IA ajudou uma startup a prever em Dezembro de 2019 o novo surto de corona vírus e onde ele apareceria em seguida.
 - Eles utilizaram processamento de linguagem natural e aprendizado de máquina para processar informações e notícias publicadas em redes sociais, sites e jornais em mais de 30 idiomas, além de informações sobre passagens aéreas.
- **Criação de vídeos/fotos falsas:** Redes Generativas Adversariais e Autoencoders foram usados no vídeo viral que utilizou trecho de “A Usurpadora” para falsificar uma conversa entre Lula e Bolsonaro.
- **Criação de novos medicamentos:** IA foi utilizada para criar uma droga capaz de combater o transtorno obsessivo-compulsivo (TOC).
 - Uma Rede Neural (professor) é inicialmente treinada para distinguir entre compostos químicos que funcionam no sistema imunológico humano e compostos inúteis.
 - Na sequência, outro modelo (aluno) aprende a gerar diferentes combinações de compostos.
 - Em seguida, o aluno gera trilhões de diferentes combinações de compostos e

os apresenta ao professor, que separa as combinações que podem resultar em novos medicamentos.

- **Neste caso,** IA ajudou a encurtar o tempo necessário para se encontrar novos medicamentos de 4 para menos de 1 ano.











Foco do Curso

- IA é um termo genérico (*guarda-chuva*) usado para designar máquinas capazes executar **tarefas** de forma inteligente.
- IA é dividida em vários objetivos ou sub-áreas, sendo as principais: **planejamento, representação do conhecimento e raciocínio, aprendizado de máquina, processamento de linguagem natural, visão computacional, robótica e inteligência artificial geral.**
- Foco do curso:** estudo dos principais algoritmos de **Aprendizado de Máquina. Por quê?**
 - ML oferece ferramentas importantes para a solução eficiente de vários problemas de telecomunicações (pós-graduação stricto sensu em telecomunicações).
 - 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 você pudesse otimizar **globalmente** todos os parâmetros do sistema apenas utilizando dados reais coletados da rede?
 - Mais ainda, imagine se você pudesse treinar um modelo que **aprendesse** um sistema de transmissão e recepção ótimo para um dado canal?
 - Redução de complexidade/custo:** vários algoritmos de comunicações que apresentam desempenho ótimo não são utilizados na prática pois possuem complexidade computacional e/ou custo proibitivos.
 - Oportunidades:** Pesquisas sobre a aplicação de aprendizado de máquina a problemas de comunicação ainda estão em fase de exploração.

AI is a broad concept of machines being able to carry out tasks smartly while machine learning (ML) is a current (probably the most popular) application/sub-area of AI that enables machines to learn from large amounts of data and act accordingly without being explicitly programmed.

AI is the ability to acquire and apply knowledge, and perform naturally like a human. For example, planning, problem solving, reasoning, learning, etc.

- Planejamento:** (determinar ações necessárias para se atingir um objetivo) tem como objetivo refletir e decidir quais ações devem ser tomadas por uma máquina, antecipando resultados esperados. Exemplo: planejamento de uma operação de resgate após um desastre natural.
- Representação do conhecimento e raciocínio:** (armazenar informação de forma eficaz e de rápido acesso. Baseado em informações conhecidas ser capaz de determinar novas informações) tem como objetivo representar informações sobre o

mundo de uma forma que uma máquina possa utiliza-las para resolver tarefas complexas tais como diagnosticar uma condição médica ou conversar com alguém através de linguagem natural.

- **Aprendizado de máquina:** tem como objetivo fazer com que uma máquina aprenda sem ser explicitamente programada através de uma grande quantidade de dados. Algoritmos de ML tem como objetivo aprender com seus erros e fazer previsões sobre dados. Exemplo, filtragem de spam, motores de busca como o do Google, diagnóstico de doenças, etc.
- **Processamento de linguagem natural:** tem como objetivo a **geração e compreensão automática de línguas naturais**. Exemplos: reconhecimento de voz, transcrição de voz em texto e vice versa,
- **Visão computacional:** tem como objetivo a **construção de máquinas que obtém entendimento de alto-nível a partir de imagens ou vídeos**. Exemplo: processamento de imagens para diagnósticos médicos. Inspeção e controle de qualidade.
- **Robótica:** (interação com o mundo) tem como objetivo o design/projeto, construção, operação e uso de robôs, bem como sistemas de computador para controle, feedback sensorial e processamento de informações. Exemplos: robôs domésticos, drones autônomos, robôs industriais, robôs militares (drones, tanques, resgate).
- **Inteligência artificial geral:** também conhecida como IA forte ou IA completa. Tem como objetivo desenvolver máquinas que percebam o ambiente ao seu redor e que executem ações que maximizem sua chance de atingir com sucesso seus objetivos. Ou seja, tem como objetivo criar máquinas que executem toda a gama de habilidades cognitivas humanas.
- Foco do curso: estudo dos principais algoritmos de **Aprendizado de Máquina**. Por quê?
 - ML oferece ferramentas importantes para a solução eficiente de vários problemas de Telecomunicações.
 - Pós-graduação em Telecomunicações.
 - Sistemas de comunicações são estudados e desenvolvidos baseando-se em modelos matemáticos (muitas vezes simplificados) que não capturam com precisão os efeitos dos sistemas e canais do mundo real (HW: intermodulação, desbalanceamento de quadratura, perdas por quantização, etc. e o canal: desvanecimento, multi-percursos, interferência, etc.).

- Modelos de canal, imperfeições do hardware, interferência, etc. são muito complicados na prática devido à natureza dinâmica dos canais de comunicação sem fio.
 - **Controle e otimização de parâmetros:** Além disto, o número de parâmetros (modulação, vetores de beamforming, taxas de código, larguras de banda, canais, potência de transmissão, seleção da melhor rota) que um sistema de próxima geração precisa controlar e otimizar continua a crescer.
 - Algoritmos de aprendizado de máquina podem modelar correlações e estimar parâmetros (sub)ótimos do sistema.
 - **Redução de complexidade/custo:** Complexidade e/ou custo de algoritmos de comunicações ótimos as vezes é proibitivo. Por exemplo, algoritmos de otimização como o de controle de potência em sistemas (massive) MIMO podem exibir alta complexidade, impedindo sua utilização/implementação em sistema de tempo real. Dirty Paper Coding (DPC) que é a técnica ótima de pré-codificação em sistemas MIMO, possui alta complexidade computacional e de implementação e portanto, não é implementado em sistemas reais. Linearização dos canais de fibra-ótica. Pré-distorção de amplificadores de RF.
 - **Oportunidades:** Pesquisas sobre a aplicação de aprendizado de máquina em problemas de comunicações móveis e sem fio ainda estão em fase de exploração.
- Channel and interference models are extremely complicated in reality due to the dynamic nature of wireless communication channels.

Telecommunications is a discipline traditionally driven by well-established mathematical models. First, while communications' systems have developed largely as a **model-driven** field, the complexities of many emerging communication scenarios raise the need to introduce **data-driven** methods into the design and analysis of mobile networks.

Designers rely on simplified closed-form models that don't accurately or holistically

capture the effects of real-world systems and channels.

Additionally, the degrees of freedom required to operate next-generation communications systems continues to increase. The number of radio and modulation parameters, such as antennas, channels, bands, beams, codes, and bandwidths, represent degrees of freedom that must be controlled by the radio system. Individual communications standards now use many more of these – in part, as an effort to mitigate the growing number of impairments.

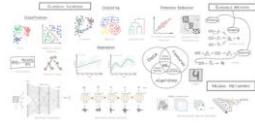
Research in the field of machine learning for wireless communications are still largely in an exploration phase.

Até hoje, sistemas de comunicações são estudados e desenvolvidos baseando-se modelos matemáticos (muitas vezes simplificados) que não capturam com precisão os efeitos dos sistemas e canais do mundo real.

Weak AI is the form of AI where programs are developed to perform specific tasks, for instance, Machine learning, Planning, Computer vision, etc.

ML is a learning technique where machines *learn* from huge data sets.

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 é “Aprendizado de máquina é o processo de **induzir** inteligência em uma máquina sem que ela seja explicitamente programada”, Jojo John Moolayil.
- **Indução**: aprender um modelo ou padrão geral a partir de exemplos.
- Algoritmos de ML são **orientados a dados**, ou seja, eles aprendem automaticamente um modelo/padrão geral (i.e., generalizar) a partir de grandes volumes de dados (i.e., exemplos).
- **Exemplo**: filtro de spam do Gmail.

ML: técnicas e algoritmos orientadas a dados: aprendem automaticamente a partir de grandes volumes de dados.

O aprendizado de máquina pode ser definido como o processo de **induzir** inteligência em uma máquina **sem que ela seja explicitamente programada**.

Por exemplo, o filtro de spam do gmail utiliza aprendizado de máquina para aprender se um email é spam (por exemplo, sinalizados por usuários) e exemplos de emails regulares (não spam, também chamados de “ham”).

Os exemplos que o modelo usa para aprender são chamados de **conjunto de treinamento**. Cada **exemplo de treinamento** é chamado de **instância de treinamento** (ou amostra).

Por que ML se tornou tão importante?

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, 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).
- Existência de frameworks e bibliotecas poderosas que facilitam o desenvolvimento de soluções com ML.



Estamos vivendo 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, quanto mais dados melhor será o aprendizado. Surgimento de recursos computacionais poderosos tais como GPUs, FPGAs, CPUs com múltiplos cores.

Surgimento de novas estratégias de treinamento (i.e., aprendizagem).

Existência de frameworks e bibliotecas que facilitam o desenvolvimento de soluções com ML.

TensorFlow é uma biblioteca de software livre e de código aberto para fluxo de dados e programação diferenciável. É uma biblioteca matemática simbólica e também é usada para aplicativos de aprendizado de máquina, como redes neurais.

Theano é uma biblioteca de computação científica. Foi desenvolvido pela Université de Montréal e está disponível desde 2007.

PyTorch é uma biblioteca de aprendizado de máquina de código aberto baseada na biblioteca Torch usada em aplicações de visão computacional e processamento de linguagem natural.

Scikit-learn é uma biblioteca de aprendizado de máquina de software livre para a linguagem de programação Python.

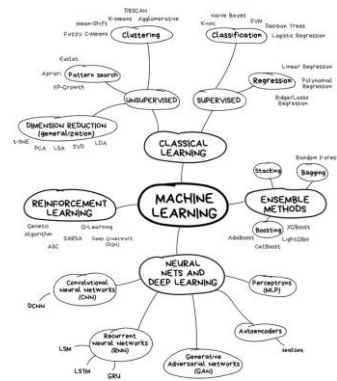
Keras é uma biblioteca de rede neural de código aberto escrita em Python. É capaz de rodar sobre TensorFlow, Microsoft Cognitive Toolkit, R, Theano ou PlaidML. Keras foi projetado para permitir experimentação rápida com redes neurais profundas, ele se concentra em ser fácil de usar, modular e extensível.

Pandas é uma biblioteca de código aberto criada para a linguagem de programação Python para manipulação e análise de dados. Em particular, oferece estruturas e operações de dados para manipulação de tabelas e séries temporais.

Tipos de Aprendizado de Máquina

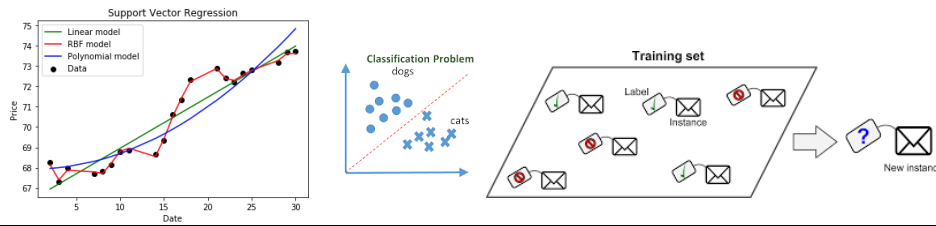
Os algoritmos de aprendizado de máquina podem ser agrupados da seguinte forma:

- Supervisionado
- Não-Supervisionado
- Semi-Supervisionado
- Por Reforço
- Evolutivo



Aprendizado Supervisionado

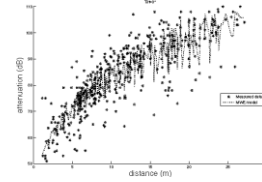
- Os dados de treinamento, chamados de **atributos** ou exemplos de treinamento (ou também características, do Inglês, features) que alimentam o algoritmo de ML incluem as **soluções desejadas** (i.e., as respostas corretas), chamadas de **rótulos** (labels).
- **Resumindo**: Os modelos de ML **aprendem** uma função que mapeie as entradas nas saídas, $y = f(x)$.
- Esse tipo de aprendizado pode ser dividido em problemas de: **Regressão** e **Classificação**.
 - **Classificação**: Rótulo, y , pertence a um **conjunto finito** de valores, i.e., conjunto finito de classes.
 - **Regressão**: Rótulo, y , pertence a um **conjunto infinito** de valores, i.e., números reais.



- Aprendizado supervisionado é o processo de aprendizado quando a máquina sabe o que aprender, ou seja, os **rótulos** são o que a máquina deve aprender.
- No **aprendizado supervisionado**, os dados de treinamento que você alimenta para o algoritmo incluem as soluções desejadas, chamadas de rótulos
- Por exemplo, a algoritmo de ML do filtro de spam, tem como entrada o email (**atributos** são: remetente, assunto, corpo do email, horário recebido) e um **rótulo** dizendo se aquele é ou não um spam.
- Analogia com trabalho, onde você tem alguém supervisionando seu trabalho e dizendo se o que foi feito está ou não correto.
- Exemplos de **regressão**: predição de quando o número de leitos de UTI vão se esgotar devido a uma pandemia, predição do preço de ações, predição do preço de imóveis, no caso de engenharia aproximação da PDF de uma variável aleatória com PDF desconhecida, predição do path-loss.

Principais Algoritmos para Aprendizado Supervisionado

- Regressão Linear e logística.
- k vizinhos mais próximos, k-nearest neighbors (k-NN)
- Árvores de Decisão, do Inglês, Decision Trees
- Florestas Aleatórias, do Inglês, Random Forests
- Máquinas de Vetores de Suporte, do Inglês, 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).




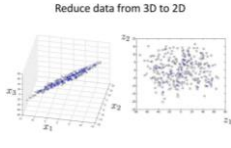
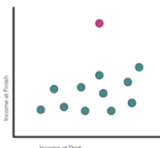

Exemplos: predição de perda por caminho e da ocupação do espectro.

SVM: pode ser usado, por exemplo, para criar um modelo de previsão de perda de caminho (path-loss) para ambientes urbanos usando máquinas de vetores de suporte (SVM) para garantir um nível aceitável de qualidade de serviço (QoS) para usuários de redes móveis e sem fio.

Deep Neural Network (ou Deep Learning): pode ser utilizada para predição da futura ocupação do espectro de frequências por redes vizinhas desconhecidas. Esta predição pode, por exemplo, ser utilizada por schedulers (alocadores de recursos) durante a alocação de recursos com o objetivo de mitigar colisões, encontrar o canal e a potência ótima.

Aprendizado Não-Supervisionado

- As máquinas não são informadas sobre o que aprender. Elas só recebem os exemplos de treinamento.
- Neste caso, os algoritmos ***aprendem/descobrem padrões*** (ocultos) presentes nos dados de entrada sem a presença de rótulos.
- **Resumindo:** Os modelos ***descobrem/aprendem*** padrões desconhecidos.
- Trata problemas de clusterização, redução/aumento de dimensionalidade, detecção de anomalias (outliers) e aprendizado de regras de associação.

- **Clusterização:** tarefa de agrupar automaticamente os dados de entrada (i.e., features) em categorias/classes. A categoria é o rótulo que é determinado para os dados de entrada fornecidos. Ou seja, podemos dizer que a clusterização tem como objetivo fazer com que a máquina aprenda/encontre classes/grupos e rotule as features.
 - **Exemplo de aplicação:** Agrupar/encontrar grupos de clientes que são similares em termos de comportamento ou características para envio de propaganda personalizada.
- **Redução de dimensionalidade:** como o próprio nome já diz, tem por objetivo encontrar formas eficientes de representação das entradas/features, ou seja, aprender vetores de saída com menor dimensão mas que ainda representem bem o vetor de entrada.
 - **Exemplo de aplicação:** Em reconhecimento de faces, as features possuem grandes dimensões (ou seja, grandes vetores) o que dificulta o treinamento de algoritmos de ML, portanto, diminuindo-se a dimensão das features pode-se diminuir o tempo de treinamento sem afetar grandemente a performance do algoritmo utilizado.
- **Aumento de dimensionalidade:** aprender vetores de saída que possuam uma dimensão maior, aumentando a quantidade de informação sobre aquele vetor.
 - **Exemplo de aplicação:** Em sistemas de comunicação digital o aumento de dimensionalidade é utilizado para encontrar *palavras código* (code words) ótimas para um canal sem fio, ou seja, esse tipo de aprendizado encontra um vetor que possui informações adicionais (redundância) que consequentemente aumenta a chances de decodificação bem sucedida da mensagem transmitida.
- **Detecção de Anomalias:** detecta se uma nova observação (i.e., feature) pertence à mesma distribuição que as observações existentes/anteriores. Identificação de

features que levantam suspeitas por serem significativamente diferentes da maioria dos dados.

- **Exemplo de aplicação:** detecção de fraude bancária, defeito estrutural, problemas médicos, erros em um texto, controle de qualidade em fábricas, etc.
- **Regras de associação:** tarefa de associar as entradas (i.e., features), encontrando regras que descrevam grandes porções dos dados.
 - **Exemplo de aplicação:** Localização de itens em supermercados. Por exemplo, o algoritmo pode encontrar uma regra de associação que diz que mais de 70% dos clientes que compram leite também compram pão e portanto, o supermercado pode colocar ambos produtos próximos uns dos outros, aumentando a chance da venda desses produtos.

Aprendizado Não-Supervisionado

- **Clusterização:** agrupa automaticamente os dados de entrada em classes. Pode-se dizer que a clusterização tem como objetivo fazer com que a máquina rotule os dados de entradas.
 - **Exemplo de aplicação:** Agrupar de clientes similares em termos de comportamento ou características para envio de propaganda personalizada.
- **Redução da dimensionalidade:** tem por objetivo encontrar representações com menor dimensão das features.
 - **Exemplo de aplicação:** Em reconhecimento de faces, as features possuem grandes dimensões o que dificulta o treinamento de algoritmos de ML, portanto, diminuindo-se a dimensão das features pode-se diminuir o tempo de treinamento sem afetar grandemente a performance do algoritmo utilizado.
- **Aumento da dimensionalidade:** encontrar representações das features que possuam uma dimensão maior, aumentando a quantidade de informação.
 - **Exemplo de aplicação:** Em sistemas de comunicação digital, o aumento de dimensionalidade é utilizado para encontrar mensagens que possuam informações adicionais (i.e., redundância), o que consequentemente aumenta a chances de decodificação da mensagem transmitida.
- **Deteção de Anomalias:** detecta se uma feature pertence à mesma distribuição que as features já conhecidas, com isso, identifica features que levantam suspeitas por serem significativamente diferentes das outras.
 - **Exemplo de aplicação:** deteção de fraude bancária, defeito estrutural, controle de qualidade em fábricas, etc.
- **Regras de associação:** associa as features, encontrando regras que descrevam grandes porções dos dados.
 - **Exemplo de aplicação:** Posicionamento de itens em supermercados. Por exemplo, encontrar uma regra de associação que diz que mais de 70% dos clientes que compram leite também compram pão e portanto, o supermercado pode colocar ambos produtos próximos uns dos outros, aumentando a chance da venda desses produtos.

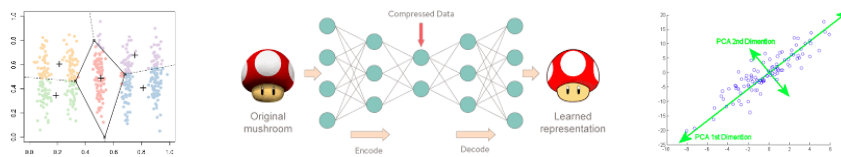
- **Clusterização:** tarefa de agrupar automaticamente os dados de entrada (i.e., features) em categorias/classes. A categoria é o rótulo que é determinado para os dados de entrada fornecidos. Ou seja, podemos dizer que a clusterização tem como objetivo fazer com que a máquina aprenda/encontre classes/grupos e rotule as features.
 - **Exemplo de aplicação:** Agrupar/encontrar grupos de clientes que são similares em termos de comportamento ou características para envio de propaganda personalizada.
- **Redução de dimensionalidade:** como o próprio nome já diz, tem por objetivo encontrar formas eficientes de representação das entradas/features, ou seja, aprender vetores de saída com menor dimensão mas que ainda representem bem o vetor de entrada.
 - **Exemplo de aplicação:** Em reconhecimento de faces, as features possuem grandes dimensões (ou seja, grandes vetores) o que dificulta o treinamento de algoritmos de ML, portanto, diminuindo-se a dimensão das features pode-se diminuir o tempo de treinamento sem afetar grandemente a performance do algoritmo utilizado.
- **Aumento de dimensionalidade:** aprender vetores de saída que possuam uma dimensão maior, aumentando a quantidade de informação sobre aquele vetor.
 - **Exemplo de aplicação:** Em sistemas de comunicação digital o aumento de dimensionalidade é utilizado para encontrar *palavras código (code words)* ótimas para um canal sem fio, ou seja, esse tipo de aprendizado encontra um vetor que possui informações adicionais (redundância) que consequentemente aumenta a chances de decodificação bem sucedida da mensagem transmitida.
- **Deteção de Anomalias:** detecta se uma nova observação (i.e., feature) pertence à mesma distribuição que as observações existentes/anteriores. Identificação de

features que levantam suspeitas por serem significativamente diferentes da maioria dos dados.

- **Exemplo de aplicação:** detecção de fraude bancária, defeito estrutural, problemas médicos, erros em um texto, controle de qualidade em fábricas, etc.
- **Regras de associação:** tarefa de associar as entradas (i.e., features), encontrando regras que descrevam grandes porções dos dados.
 - **Exemplo de aplicação:** Localização de itens em supermercados. Por exemplo, o algoritmo pode encontrar uma regra de associação que diz que mais de 70% dos clientes que compram leite também compram pão e portanto, o supermercado pode colocar ambos produtos próximos uns dos outros, aumentando a chance da venda desses produtos.

Principais Algoritmos para Aprendizado Não-Supervisionado

- **k-médias (k-means)**
 - Particiona as features em k clusters 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.



No aprendizado não-supervisão não se sabe o que a máquina precisa aprender.

Alguns algoritmos não-supervisionados são:

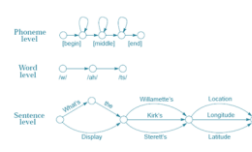
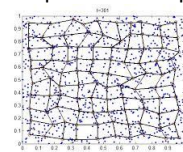
- **k-médias:** particiona os dados em k clusters distintos com base na distância ao centroide de um cluster.
- **Autoencoders:** usado para redução ou para aumento de dimensionalidade.

Exemplos:

- **k-médias:** encontrar grupos de clientes com um mesmo perfil de compra ou encontrar separações ótimas para a decodificação de uma modulação desconhecida.
 - Imagine que você tenha dados de uma modulação mas que você não tenha os labels, com o k-médias você pode encontrar grupos/clusters que representariam cada um dos símbolos da modulação desconhecida.
 - Num supermercado, baseado em várias características do que foi comprado o algoritmo poderia encontrar um cluster/grupo de homens que compram fraldas e que também compram cerveja, e o supermercado poderia colocar essas mercadorias juntas.
- **Auto-encoders:** podem ser usados em comunicações digitais para por exemplo aprender um sistema de codificação e decodificação de canal ótimo para um dado canal sem-fio.
- **PCA:** utilizado na redução da dimensionalidade. Utilizado em problemas onde um número de dimensões é muito grande para ser tratado por um modelo em tempo razoável e com recursos computacionais existentes.

Principais Algoritmos para Aprendizado Não-Supervisionado

- Análise de Componentes Independentes (Independent Component Analysis, ICA)
 - Realiza uma espécie de clusterização baseada em propriedades estatísticas dos sinais envolvidos.
- Mapas Auto-Organizáveis (Self-Organized Map, SOM)
 - Tipo de rede neural usada para aprender a topologia e distribuição dos dados.
- Modelos Ocultos de Markov (Hidden Markov Models, HMM)
 - usa dados observados para recuperar uma sequência de estados.



Alguns algoritmos não-supervisionados são:

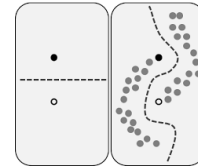
- **Self-organizing maps:** usa redes neurais para aprender a topologia e distribuição dos dados.
- **Hidden Markov Models:** usa dados observados para recuperar uma sequência de estados.

Exemplos:

- ICA: utilizado no “problema do coquetel” onde o algoritmo separa as conversas de várias pessoas falando ao mesmo tempo em um mesmo ambiente.
- SOM: utilizado para criar visualizações de baixa dimensão à partir de dados de alta dimensão.
- HMM: utilizados em reconhecimento de fala para reconhecer a sequência de fonemas e por fim identificar uma palavra.

Aprendizado Semi-Supervisionado

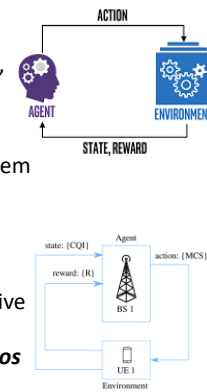
- As máquinas tem acesso a exemplos com e sem rótulos.
- Geralmente envolve uma pequena quantidade de dados rotulados e uma grande quantidade de dados não-rotulados.
- É de grande ajuda em casos onde se ter uma grande quantidade de dados rotulados é muito caro ou complexo.
- Algoritmos de aprendizagem semi-supervisionada são o resultado da combinação de algoritmos supervisionados e não-supervisionados.
- **Exemplo:** Facebook e Instagram recebem várias fotos suas e de conhecidos seus. Em uma várias fotos um algoritmo de **clusterização** agrupa pessoas A, B e C, em outras fotos, pessoas B, D e E. Em determinado momento, você carrega uma foto onde você fornece o nome (label) de algumas dessas pessoas.



Exemplo: O Facebook/Instagram recebe centenas/milhares de fotos suas e de conhecidos/familiares seus. Em uma grande quantidade de fotos um algoritmo de clusterização encontra/agrupa pessoas A, B e C, em outra grande quantidade de fotos pessoas B, D e E. Em determinado momento, você carrega uma foto onde você fornece o nome (label) de algumas desses grupos/pessoas.

Aprendizado Por Reforço

- Abordagem totalmente diferente das anteriores pois não se tem dados de treinamento, sejam eles rotulados ou não.
- O algoritmo de aprendizagem, 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 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, o **estado** do **ambiente**.
- Uma **Política**, denotada por π , é uma função que mapeia os **estados** do **ambiente** em **ações** que o **agente** deve tomar, $\pi(s) = a$.

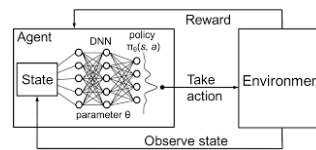
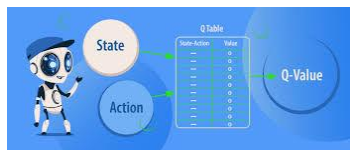


- Aprendizado por reforço pode ser entendido como o problema em que um **agente** deve aprender como se comportar em um **ambiente** através de interações do tipo “tentativa e erro”.
- O **agente** tem como objetivo aprender/construir um modelo do ambiente em que está inserido.
- O algoritmo aprende a escolher **ações** apenas interagindo com o **ambiente**.
- **Problema da aprendizagem por reforço**: Como escolher uma **política** de **ações** que maximize o total de **recompensas** recebidas pelo **agente**.
- A **política** pode ser entendida como sendo a função que mapeia os **estados** do **ambiente** em **ações** que o **agente** deve tomar: $\pi(s) = a$.
- **Exemplos**:
 - Algoritmo muito utilizado em jogos.
 - O programa AlphaGo da Google (DeepMind) utilizou aprendizado por reforço (Deep reinforcement learning) para aprender a jogar Go e derrotar o campeão mundial de Go.
 - Agente jogador de damas
 - **Estados**: as diferentes configurações do tabuleiro.
 - **Ações**: mover uma determinada peça.
 - **Recompensas**: número de capturas de peças vs. número de perdas.
 - Modulação e codificação adaptativas
 - **Estado**: qualidade do canal (CQI) enviado pelo dispositivo móvel (podem haver vários outros estados, por exemplo QoS).
 - **Ação**: modulação e taxa de código.
 - **Recompensa**: uma medida baseado na taxa de erro de bloco (BLER) ou na eficiência espectral (i.e., no número de bits/s/Hz possível).

- Robô aspirador de pó que precisa aprender a limpar uma casa de forma ótima.
- Os algoritmos de aprendizado por reforço tentam encontrar a **política** que mapeia os **estados** do **ambiente** às **ações** que o **agente** deve tomar nesses **estados**.
- Como um algoritmo de aprendizagem aprende a escolher ações apenas interagindo com o ambiente?
 - Muitas vezes, é impraticável o uso de aprendizagem supervisionada.
 - Como obter exemplos do comportamento correto e representativo para qualquer situação?
 - E se o agente for atuar em um ambiente desconhecido?

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 tabela a **função-Q**, já o Deep Q-Learning encontra uma função que aproxime a **função-Q**.



Deep Q-Learning: É a junção de Deep Learning com Q-Learning. As redes neurais profundas possibilitam que Q-Learning seja aplicado a problemas maiores.

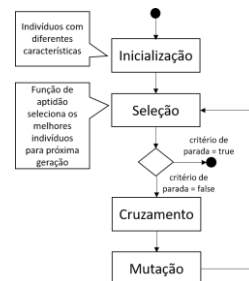
Q-Table é apenas um nome sofisticado para uma tabela de pesquisa (lookup Table) simples, onde calculamos as recompensas futuras máximas esperadas para a ação em cada estado.

Cada posição na tabela Q será a recompensa futura máxima esperada que o agente receberá se executar essa ação nesse estado. Este é um processo iterativo, pois precisamos melhorar o Q-Table a cada iteração.

Uma função de valor Q mostra o quão boa uma determinada ação é, dado um estado, para um agente que segue uma determinada política.

Aprendizado Evolutivo

- Aprendizado baseado na evolução biológica dos seres vivos.
- Novos **indivíduos** com diferentes **características**, pertencentes a uma **população**, são criados através de **cruzamentos** e **mutações** onde apenas os mais aptos são **selecionados** (**seleção natural**) e passam suas **características** para uma nova geração. Com isso a **população** se torna mais apta a cada geração.
- Utilizado para encontrar soluções de problemas de otimização.



Composto por uma família de algoritmos para otimização global inspirada na evolução biológica.

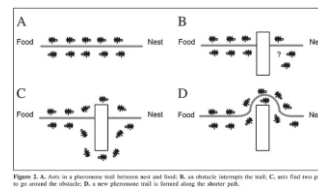
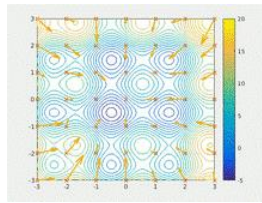
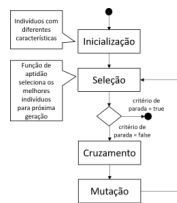
Novos **indivíduos** com diferentes **características** (código genético), pertencentes a uma **população**, são criados através de **cruzamentos** e **mutações** onde apenas os mais aptos são **selecionados** (**seleção natural**) e passam suas **características** para uma nova geração. Com isso a **população** se torna mais apta a cada geração.

No aprendizado evolutivo, um conjunto inicial de soluções candidatas (**indivíduos**) é gerado e atualizado iterativamente. Cada nova **geração** é produzida pela remoção de soluções menos desejadas e pela introdução de pequenas alterações aleatórias. Na terminologia biológica, uma população de soluções está sujeita a seleção natural e mutação. Como resultado, a população evoluirá gradualmente para aumentar a aptidão, neste caso a função de aptidão escolhida do algoritmo.

Tipo de aprendizagem utilizada para encontrar soluções aproximadas em problemas de otimização e busca.

Principais Algoritmos de Aprendizado Evolutivo

- **Algoritmos Genéticos.**
 - 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.



- **Algoritmos Genético:** inspirados pelo processo de seleção natural. São projetados para encontrar, gerar ou selecionar uma heurística (algoritmo de busca parcial) que forneça uma solução suficientemente boa para um problema de otimização, especialmente com informações incompletas ou imperfeitas ou capacidade de computação limitada.
- **Otimização por enxame de partículas:** Inspirado no comportamento de cardumes de peixes e de bandos de pássaros. Resolve um problema de otimização ao ter uma **população** de soluções candidatas, denominadas **partículas**, e movê-las no **espaço de busca** sendo que existe cooperação entre as partículas.
 - Analogia: Imagine que você e vários amigos, todos com rádios, estão tentando localizar o ponto mais baixo de um terreno. Vocês começam a percorrer o terreno e a reportar via rádio a posição atual e a elevação do terreno, até que através de cooperação vocês encontram o ponto mais baixo.
- **Otimização da colônia de formigas:** É inspirado no comportamento das formigas ao saírem de sua colônia para encontrar comida. É uma técnica probabilística para resolver problemas computacionais que pode ser reduzida para encontrar bons caminhos através de grafos.
 - As formigas ao procurarem por alimento deixam feromônios para que outras formigas possam segui-las.

Tipos de Treinamento

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

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

Treinamento em Batelada

- Algoritmo não pode ser treinado incrementalmente, sendo treinado com todos os exemplos (i.e., dados de treinamento) disponíveis.
- O treinamento é demorado e utiliza muitos recursos computacionais (e.g., CPU, memória, etc.).
- 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.
- Dados podem ser pré-processados/analísados, evitando assim, dados corrompidos ou com problemas.
- É um tipo de treinamento simples, de fácil implementação e obtém ótimos resultados.

Treinamento Incremental

- 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.
- Como não há pré-processamento/análise, dados corrompidos ou com problemas afetam a performance do sistema.

Python, Jupyter & Colab



O que é Python?

- Linguagem de programação de alto nível, interpretada, de script, multiparadigma (suporta os paradigmas orientado a objetos, imperativo, funcional, procedural e reflexivo).
- Foi lançada por Guido van Rossum em 1991.
- Possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela Python Software Foundation.
- **Curiosidade:** o nome Python é uma homenagem ao grupo de humor britânico, Monty Python.

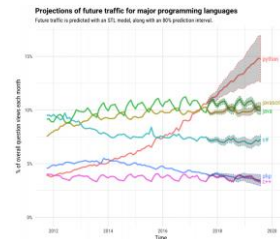


- Alto nível: longe do código de máquina e mais próximo à linguagem humana.
- Interpretada: programa é interpretado/executado por um interpretador e em seguida é executado pelo sistema-operacional/processador. Linguagens interpretadas têm seus códigos fontes transformados em uma linguagem intermediária (específica de cada linguagem), que será interpretada pela máquina virtual da linguagem quando o programa for executado.
- Script: que pode ser utilizada para automatizar pequenas tarefas, porém, python é muito poderosa e vai além da linguagem de script
- Python é tida como uma linguagem multiparadigma, um mesmo programa pode ser feito utilizando paradigmas diferentes ou um único programa pode ser criado utilizando mais de um paradigma de programação
 - Imperativa: comandos que dizem como o processador deve acessar e modificar dados armazenados em memória. O programador diz **como** e **o que** um programa deve realizar.
 - Orientada a objetos: sw baseado na composição e interação entre diversos 'objetos'.
 - Funcional: programação através de funções e métodos que são avaliados como se fossem funções matemáticas.
 - Procedural: baseado no conceito de chamadas a procedimentos. Os procedimentos, são também conhecidos como rotinas, subrotinas, métodos, ou funções
 - Reflexão é a capacidade de um programa observar ou até mesmo modificar sua estrutura ou comportamento.

Por que Python?

Algumas razões são:

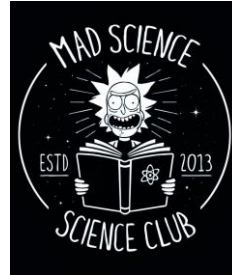
1. É uma das linguagens mais fáceis de se aprender.
2. Python é muito popular e será mais ainda nos próximos anos.
3. Grandes empresas usam python.
4. Python é a linguagem mais usada em aplicações de Machine Learning.
5. Possui um vasto suporte on-line: Tutoriais, Videos e StackOverflow.
6. Python é usada como linguagem educativa para ensino de computação e eletrônica (e.g., Raspberry Pi, LEGO Mindstormsem suas aplicações desenvolvidas em Python).
7. Python é gratuito e open-source.
8. Possui um rico ecossistema de bibliotecas: TensorFlow, OpenCV, Scrappy, Ipython, etc.



- 1 – Com uma taxa de crescimento constante, projeções mostram que Python será a linguagem mais utilizada em 2020, superando até o JavaScript.
- 2 – Google, Uber, Netflix, Facebook, Instagram dentre várias outras utilizam python.
- 3 – Por exemplo, TensorFlow, que é um framework para machine learning, utiliza principalmente python
- 4 – Milhares de tutoriais, vídeos, e temos o StackOverflow dando suporte para questões desde nível básico até níveis avançadíssimos, existe até uma versão de Minecraft que ensina como programar em python.
- 5 – Muitas plataformas que promovem o ensino de computação e eletrônica como Raspberry Pi e LEGO Mindstorms tem suas aplicações desenvolvidas em python.
- 6 – Você pode criar e distribuir sua própria biblioteca, modificar/melhorar bibliotecas já existentes e uma grande suporte da comunidade
- 7 – TensorFlow: Machine Learning, OpenCV: visão computacional (reconhecimento facial, de movimentos, etc.), Scrappy: Data Mining, Ipython: execução de códigos python no web browser.

Programação Científica e Python

- Teve uma rápida adoção pela comunidade científica.
 - Engenharias
 - Biologia
 - Química
 - Física
 - Etc.



Jupyter

- **Jupyter**: aplicação web que permite criar e compartilhar documentos contendo código, equações, visualizações e texto explicativo.
- Suporta execução em dezenas de linguagens de programação: Python, C++, C#, PHP, Julia, R, etc.
- <https://jupyter.org/>
- A Lista #0 contém um link para um tutorial de instalação do Jupyter.
- Pode ser usado para execução de exemplos e exercícios do curso.



Google Colaboratory (Colab)



- **Colab**: outra aplicação web, baseada no Jupyter, que permite criar e compartilhar documentos contendo código, equações, visualizações e texto explicativo.
- É um produto da Google.
- Por hora, suporta apenas a execução de programas em Python.
- Vantagens sobre o Jupyter:
 - Maior número de servidores.
 - Fornece GPUs e TPUs gratuitamente.
 - Compartilhamento de notebooks entre usuários é mais fácil.
- <https://colab.research.google.com/>
- Pode também ser usado para execução de exemplos e exercícios do curso.

Alguns Exemplos

Histogramas

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

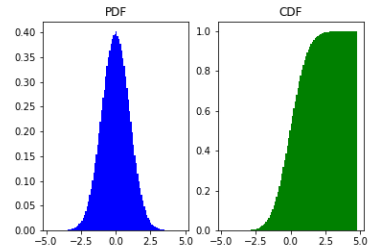
%matplotlib inline

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
```



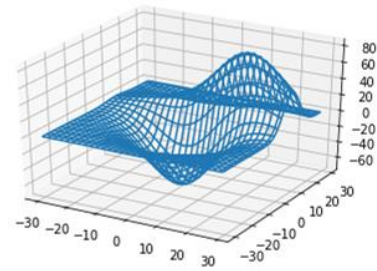
Figuras 3D

```
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)

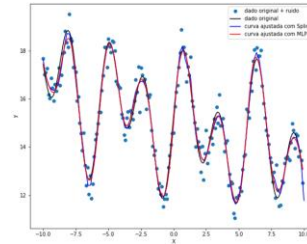
# faz com que o gerador de numeros aleatorios sempre forneça os mesmos valores
np.random.seed(42)

# adicionando ruído 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 = 'dado original + ruído')
plt.plot(x, y, 'x', label = 'dado 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)
```



Outras Bibliotecas Científicas

- Muitas outras possibilidades através do uso de diversas outras bibliotecas.
- Alguns exemplos:
 - **Pandas**: manipulação e análise de dados.
 - **SymPy**: manipulações simbólicas ao estilo *Mathematica*.
 - **AstroPy**: funcionalidades para astrônomos e astrofísicos.
 - **NetworkX**: usada para estudo de grafos e redes
 - etc.



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 pessoal de livros,
https://drive.google.com/drive/folders/1IyII_Mu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing

Avisos

- Lista #0 já está disponível tanto no site quanto no MS Teams.
- Lista #1 será liberada ainda hoje.
- [Projeto Prático](#)
- Entregas devem ser feitas tanto no GitHub quanto no MS Teams.
 - Meu controle de entregas será via MS Teams e a correção será através do GitHub.
 - Se atentem às datas de entrega no MS Teams.
- Todo material do curso será disponibilizado no MS Teams e no site:
 - <https://www.inatel.br/docentes/felipefigueiredo/>
- Horário de Atendimento
 - Todas as Quintas-Feiras das 15:30 às 17:30 via MS Teams.

Perguntas?

Obrigado!

