

Introdução aos Fundamentos de Machine Learning

github.com/kvpergentino/data-science-explained/

1. Definições Essenciais: Inteligência Artificial e Aprendizado de Máquina.....	1
2. Paradigmas do Aprendizado de Máquina.....	1
3. Técnicas e Aplicações Práticas.....	3
4. O Ciclo de Vida de um Projeto de Machine Learning.....	5
5. Conceitos Fundamentais de Programação para Machine Learning.....	6
6. Ferramentas e Ecossistema Tecnológico.....	8
7. Fluxos de Trabalho em Machine Learning.....	9
Resumo.....	10
Glossário Geral de ML e IA.....	11

1. Definições Essenciais: Inteligência Artificial e Aprendizado de Máquina

A **Inteligência Artificial (IA)** é um vasto campo da ciência da computação dedicado à criação de sistemas capazes de realizar tarefas que, tradicionalmente, exigiriam inteligência humana. Isso inclui atividades como o raciocínio, a resolução de problemas, a compreensão da linguagem e a percepção visual.

Dentro deste campo abrangente, encontra-se o **Aprendizado de Máquina (Machine Learning - ML)**, um subcampo específico que se concentra no desenvolvimento de algoritmos que permitem aos computadores aprender diretamente a partir de dados. Em vez de serem explicitamente programados para executar uma tarefa, os sistemas de ML identificam padrões e relações nos dados fornecidos e utilizam esse conhecimento para tomar decisões ou fazer previsões sobre novos dados. Campos como o **Aprendizado Profundo (Deep Learning)**, que utiliza redes neurais complexas, são especializações dentro do próprio Machine Learning.

2. Paradigmas do Aprendizado de Máquina

Os modelos de Machine Learning aprendem por meio de diferentes abordagens, ou paradigmas. Os quatro principais são:

- **Aprendizado Supervisionado:** Nesta abordagem, o modelo é treinado com um conjunto de dados previamente rotulado, onde cada exemplo de entrada está associado a uma saída correta. O objetivo é que o modelo aprenda a mapear as entradas para as saídas, para que possa prever o rótulo de dados futuros e desconhecidos.
 - **Exemplo:** Treinar um modelo com milhares de imagens de e-mails, cada uma rotulada como "spam" ou "não spam", para que ele aprenda a classificar novos e-mails.

- **Aprendizado Não Supervisionado:** Diferente do supervisionado, este paradigma utiliza dados não rotulados. O objetivo do modelo é descobrir estruturas, padrões ou agrupamentos ocultos nos dados por conta própria, sem uma "resposta" pré-definida.
 - **Exemplo:** Analisar o comportamento de compra de clientes de uma loja para identificar segmentos de mercado distintos (clusters) com base em seus hábitos.
- **Aprendizado Semi-supervisionado:** Esta é uma abordagem híbrida que utiliza um conjunto de dados contendo uma pequena quantidade de dados rotulados e uma grande quantidade de dados não rotulados. O modelo utiliza os dados rotulados para iniciar seu aprendizado e, em seguida, aprimora seu desempenho de forma iterativa ao incorporar os padrões encontrados nos dados não rotulados.
 - **Exemplo:** Em um vasto acervo de documentos, um pequeno subconjunto é classificado manualmente por tópico. O modelo usa essa base para classificar o restante, refinando-se progressivamente.
- **Aprendizado por Reforço:** Neste paradigma, um "agente" de IA aprende a tomar decisões interagindo com um ambiente. O agente recebe recompensas ou penalidades com base nas ações que executa, e seu objetivo é aprender uma política de ações que maximize a recompensa total ao longo do tempo.
 - **Exemplo:** Treinar um sistema de inteligência artificial para jogar xadrez, onde ele é recompensado por movimentos que levam à vitória e penalizado por aqueles que resultam em derrota.

Tipos de Aprendizado de Máquina

Uma visão geral dos principais paradigmas de Machine Learning.

Tipo de Aprendizagem	Entrada de Dados	Objetivo
Supervisionado	Dados Rotulados	Prever Resultados
Não Supervisionado	Dados Não Rotulados	Descobrir Padrões
Semi-supervisionado	Maioria não rotulada	Aproveitar estruturas
Por Reforço	Sem dados iniciais	Maximizar Recompensa

3. Técnicas e Aplicações Práticas

A escolha da técnica de ML adequada depende da natureza do problema, do tipo de dados e do resultado desejado. As principais técnicas são:

Classificação: Técnica supervisionada que prevê uma categoria ou classe discreta. É utilizada para responder a perguntas como "qual tipo?" ou "a qual grupo pertence?".

- Aplicação: Diagnóstico médico, como determinar se uma célula de um tumor é benigna ou maligna.

Regressão: Técnica supervisionada que prevê um valor numérico contínuo. É usada para responder a perguntas como "quanto?" ou "qual o valor?".

- Aplicação: Previsão do preço de um imóvel com base em suas características (área, número de quartos, localização).

Clusterização (Agrupamento): Técnica não supervisionada que agrupa pontos de dados similares em clusters. Itens no mesmo cluster são mais semelhantes entre si do que com aqueles em outros clusters.

- Aplicação: Segmentação de clientes para campanhas de marketing personalizadas.

Associação: Técnica não supervisionada que identifica itens ou eventos que ocorrem frequentemente juntos em um conjunto de dados.

- Aplicação: Análise de cesta de compras em um supermercado para descobrir produtos frequentemente comprados juntos (ex: pão e manteiga).

Deteção de Anomalias: Identifica instâncias raras ou eventos que se desviam significativamente da maioria dos dados.

- Aplicação: Deteção de transações fraudulentas de cartão de crédito, que se desviam do padrão de gastos de um cliente.

Mineração de Sequências: Técnica focada em descobrir padrões temporais ou sequenciais para prever o próximo evento provável em uma série.

- Aplicação: Análise de *clickstream* em um website para entender a jornada do usuário e prever sua próxima ação, ou otimizar o layout do site.

Redução de Dimensionalidade: Reduz o número de variáveis (ou *features*) em um conjunto de dados, mantendo a maior parte da informação relevante. É útil para simplificar modelos e otimizar o armazenamento e o processamento.

- Aplicação: Compressão de imagens ou pré-processamento de dados genômicos com milhares de variáveis.

Sistemas de Recomendação: Preveem a preferência ou avaliação que um usuário daria a um item, com base em seus dados e nos dados de usuários com gostos semelhantes.

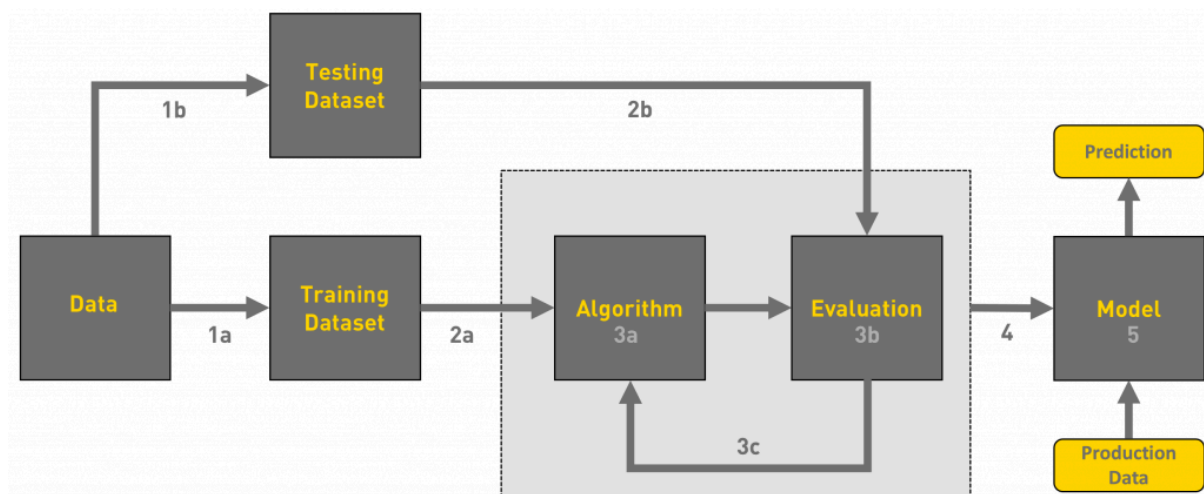
- Aplicação: Recomendação de filmes, livros ou produtos em plataformas de e-commerce e streaming.

4. O Ciclo de Vida de um Projeto de Machine Learning

O desenvolvimento de uma solução de Machine Learning segue um processo estruturado e iterativo, conhecido como ciclo de vida do modelo:

1. **Definição do Problema:** Compreender a necessidade de negócio e traduzi-la em um problema de ML bem definido (ex: "precisamos classificar e-mails" ou "queremos prever as vendas do próximo trimestre").
2. **Coleta e Preparação de Dados:** Esta fase envolve a extração, transformação e carregamento de dados, processo conhecido como **ETL (Extract, Transform, Load)**. Os dados são coletados de diversas fontes, limpos (tratamento de valores ausentes, remoção de duplicatas) e transformados em um formato adequado para a modelagem.
3. **Desenvolvimento do Modelo:** Inclui a seleção do algoritmo, o treinamento do modelo com os dados de treino e a avaliação de seu desempenho com dados de teste.
4. **Avaliação e Ajuste Fino:** O desempenho do modelo é medido com métricas apropriadas. Se necessário, seus hiperparâmetros são ajustados (**hyperparameter tuning**) para otimizar os resultados.
5. **Implantação (Deployment):** O modelo final é integrado a um ambiente de produção onde pode receber novos dados e fornecer previsões em tempo real ou em lote.

É fundamental entender que este ciclo é **iterativo**. Um modelo em produção pode ter seu desempenho degradado com o tempo, exigindo um retorno a etapas anteriores para coletar novos dados, refinar o problema ou treinar um novo modelo.



5. Conceitos Fundamentais de Programação para Machine Learning

Para compreender como as ferramentas de Machine Learning funcionam, é essencial entender a estrutura e a organização do código. Os conceitos a seguir descrevem como os componentes de software são construídos e interagem.

Linguagem de Programação

É o sistema formal utilizado para escrever instruções que um computador pode executar. Cada linguagem possui uma sintaxe (regras gramaticais) e semântica (significado) próprias.

- **No contexto de ML:** O **Python** é a linguagem predominante devido à sua sintaxe legível e, crucialmente, ao seu vasto conjunto de ferramentas especializadas (bibliotecas e frameworks).

Algoritmo

É uma sequência finita e bem definida de passos lógicos ou matemáticos, projetada para resolver um problema ou executar uma tarefa. Um algoritmo é a teoria ou o método por trás de uma operação.

- **No contexto de ML:** Os modelos de aprendizado são baseados em algoritmos. Por exemplo, "Regressão Linear" é um algoritmo que define o método matemático para encontrar a melhor relação linear entre variáveis.

Módulo

Em Python, um módulo é a unidade fundamental de organização de código. Trata-se de um único arquivo (com extensão `.py`) que agrupa um conjunto de funções, classes e variáveis relacionadas. Sua finalidade é organizar o código de forma lógica e permitir sua reutilização em diferentes partes de um projeto.

Biblioteca

Uma biblioteca é uma coleção de módulos que fornecem código pré-escrito e reutilizável para realizar um conjunto de tarefas relacionadas. O objetivo principal de uma biblioteca é abstrair a complexidade: ela oferece funcionalidades prontas para uso, permitindo que o desenvolvedor as utilize sem precisar conhecer os detalhes de sua implementação interna.

- **No contexto de ML:** Bibliotecas são indispensáveis. Em vez de um cientista de dados programar um algoritmo de classificação do zero, ele simplesmente importa e utiliza a implementação robusta e otimizada da biblioteca **Scikit-learn**. Da mesma forma, a biblioteca **Pandas** oferece estruturas de dados e funções para a manipulação de dados tabulares, acelerando drasticamente a fase de preparação de dados.

Framework

Um framework é uma estrutura de software mais abrangente e, por vezes, mais rígida que uma biblioteca. Ele fornece uma arquitetura fundamental e um fluxo de trabalho predefinido para a construção de aplicações. A principal diferença conceitual é a "inversão de controle":

- **Com uma biblioteca**, seu código está no controle e *chama* as funções da biblioteca quando necessário.
- **Com um framework**, a estrutura do framework está no controle e *chama* o seu código em pontos específicos. Você preenche as partes customizáveis dentro da arquitetura que o framework impõe.
- **No contexto de ML: TensorFlow e PyTorch** são frameworks de Deep Learning. Eles fornecem a estrutura completa para projetar, treinar e implantar redes neurais complexas, gerenciando operações de baixo nível como cálculos em GPU e diferenciação automática.

API (Application Programming Interface | Interface de Programação de Aplicações)

Uma API é um conjunto de regras, definições e protocolos que especificam como diferentes componentes de software devem interagir. Ela funciona como um "contrato" que define os métodos (funções) disponíveis em uma biblioteca ou serviço, os parâmetros que eles aceitam e o tipo de resposta que retornam.

- **No contexto de ML:** A API da Scikit-learn, por exemplo, é consistente: modelos são treinados com um método `.fit()` e fazem previsões com um método `.predict()`. Essa padronização torna a experimentação com diferentes algoritmos extremamente eficiente.

Ecossistema

Refere-se à rede coletiva de uma linguagem de programação, suas bibliotecas, frameworks, ferramentas de desenvolvimento e as convenções da comunidade. Um ecossistema forte implica que suas ferramentas são bem mantidas, documentadas e, fundamentalmente, interoperáveis.

- **No contexto de ML:** O ecossistema Python é considerado o mais maduro para ciência de dados porque suas principais bibliotecas (NumPy, Pandas, Scikit-learn, Matplotlib, etc.) foram projetadas para funcionar de maneira integrada, permitindo um fluxo de trabalho coeso desde a coleta de dados até a produção do modelo.

6. Ferramentas e Ecossistema Tecnológico

O desenvolvimento em Machine Learning é suportado por um robusto ecossistema de ferramentas, especialmente em torno das linguagens de programação Python e R.

- **Linguagens de Programação:**

- **Python:** A linguagem dominante em ML, devido à sua sintaxe simples e a um vasto conjunto de bibliotecas especializadas.
- **R:** Fortemente utilizada em estatística e pesquisa acadêmica, com excelentes pacotes para análise e visualização de dados.
- **Outras:** Linguagens como Julia, Scala, Java e JavaScript também são utilizadas em contextos específicos, como computação de alta performance e aplicações web.

- **Processamento e Análise de Dados:**

- **Bibliotecas Python: NumPy** é a base para computação numérica, fornecendo estruturas de dados eficientes (arrays). **Pandas** é construída sobre o NumPy e oferece DataFrames, uma estrutura de alto nível para manipulação e análise de dados tabulares.
- **Big Data:** Ferramentas como **Apache Spark** e **Hadoop** são usadas para processar volumes massivos de dados que não cabem na memória de uma única máquina.

- **Visualização de Dados:**

- **Bibliotecas: Matplotlib** é a biblioteca fundamental para a criação de gráficos em Python. **Seaborn** é uma interface de alto nível sobre o Matplotlib, simplificando a criação de gráficos estatísticos.
- **Ferramentas de Business Intelligence (BI):** Plataformas como **Tableau**, **Power BI** e **Looker Studio** permitem a criação de dashboards interativos para a exploração visual dos dados.

- **Frameworks de Machine Learning e Deep Learning:**

- **ML Clássico: Scikit-learn** é a biblioteca padrão em Python para implementar algoritmos clássicos de classificação, regressão e clusterização.
- **Deep Learning: TensorFlow** e **PyTorch** são os principais frameworks para redes neurais complexas, com **Keras** atuando como uma interface de alto nível que simplifica o uso do TensorFlow.

7. Fluxos de Trabalho em Machine Learning

Atualmente, coexistem duas abordagens principais para o desenvolvimento de soluções de IA:

A. Fluxo de Trabalho Clássico Este fluxo é centrado na construção de um modelo a partir do zero, utilizando bibliotecas como Scikit-learn.

1. **Pré-processamento:** Os dados são limpos e preparados com Pandas.
2. **Divisão dos Dados:** O conjunto de dados é dividido em subconjuntos de treino e teste (`train_test_split`).
3. **Treinamento:** Um modelo é instanciado (ex: `RandomForestClassifier`) e treinado com os dados de treino usando o método `.fit()`.
4. **Avaliação:** O desempenho do modelo é verificado nos dados de teste usando métricas como acurácia ou matriz de confusão.
5. **Implantação:** O modelo treinado é salvo para ser usado em produção.

B. Fluxo de Trabalho com IA Generativa Este fluxo, impulsionado por modelos de grande escala, foca na adaptação de modelos pré-existentes.

1. **Ponto de Partida:** Inicia-se com um **modelo de fundação (foundation model)**, como os da família GPT, que já foi treinado em uma quantidade massiva de dados.
 2. **Interação e Adaptação:** Em vez de treinar do zero, a interação se dá por meio de técnicas como:
 - **Engenharia de Prompt:** Elaboração de instruções precisas em linguagem natural para guiar a resposta do modelo.
 - **RAG (Retrieval-Augmented Generation):** Conexão do modelo a uma base de dados externa (ex: documentos internos de uma empresa) para que ele possa fornecer respostas baseadas em informações específicas e atualizadas.
 - **Ajuste Fino (Fine-Tuning):** Adaptação do modelo pré-treinado a um domínio ou tarefa específica, utilizando um conjunto de dados menor e mais focado. Técnicas como **PEFT (Parameter-Efficient Fine-Tuning)** permitem fazer isso com grande economia de recursos computacionais.
-

Resumo

- **Conceitos Fundamentais:** A Inteligência Artificial (IA) é o campo amplo que simula a cognição humana. O Aprendizado de Máquina (ML) é o subcampo focado em algoritmos que aprendem a partir de dados, seguindo paradigmas como o aprendizado supervisionado (com dados rotulados), não supervisionado (encontrando padrões ocultos) e por reforço (interagindo com um ambiente).
- **Técnicas e Processos:** A escolha de uma técnica de ML (como Classificação, Regressão ou Clusterização) depende do problema a ser resolvido. Essas técnicas são aplicadas dentro de um ciclo de vida iterativo, que abrange desde a preparação dos dados (ETL) até a implantação e o monitoramento do modelo.
- **Ecossistema de Ferramentas:** A linguagem Python é a base do ecossistema de ML, suportada por bibliotecas cruciais como NumPy (cálculos numéricos), Pandas (manipulação de dados), Scikit-learn (modelos clássicos) e Matplotlib/Seaborn (visualização). Para tarefas de Aprendizado Profundo, os frameworks dominantes são TensorFlow e PyTorch.
- **Fluxos de Trabalho Modernos:** Atualmente, destacam-se duas abordagens principais: o fluxo de trabalho clássico, que envolve treinar modelos a partir do zero, e o fluxo de IA Generativa, que se concentra em adaptar modelos de fundação pré-treinados por meio de Engenharia de Prompt, RAG e técnicas de Ajuste Fino (Fine-Tuning).

Glossário Geral de ML e IA

A

- **AI - Inteligência Artificial:** Um campo da ciência da computação que se concentra na criação de máquinas capazes de realizar tarefas que normalmente exigiriam inteligência humana.
 - **Algoritmo:** Uma sequência de passos lógicos e computacionais, finitos e bem definidos, projetada para resolver um problema específico ou executar uma tarefa. Em Machine Learning, define como um modelo aprende a partir dos dados.
 - **API (Application Programming Interface):** Um conjunto de regras e protocolos que especifica como diferentes componentes de software devem interagir. Atua como uma interface que define os métodos (chamadas) e formatos de dados válidos para a comunicação entre sistemas.
 - **Ajuste Fino (Fine-Tuning):** O processo de pegar um modelo de machine learning que já foi treinado em um grande volume de dados (um modelo pré-treinado) e continuar seu treinamento com um conjunto de dados menor e mais específico para uma nova tarefa.
 - **Autoencoders:** Um tipo de rede neural artificial usada para compressão de dados não supervisionada.
-

B

- **Backpropagation:** Um algoritmo fundamental no treinamento de redes neurais que ajusta os pesos do modelo para minimizar o erro.
 - **BERT - Bidirectional Encoder Representations from Transformers:** Um modelo de linguagem pré-treinado do Google que processa palavras em uma frase em ambas as direções para uma compreensão mais profunda do contexto.
 - **Biblioteca (Library):** Uma coleção de código pré-escrito e reutilizável, organizado em módulos, que fornece funções para realizar tarefas específicas, permitindo que desenvolvedores utilizem funcionalidades complexas sem programá-las do zero.
 - **BLEU - Bilingual Evaluation Understudy:** Uma métrica usada para avaliar a qualidade do texto gerado, principalmente em tarefas de tradução automática.
-

C

- **CNNs - Convolutional Neural Networks:** Um tipo de rede neural que se destaca no processamento de dados com uma topologia de grade, como imagens e vídeos.
-

D

- **DataFrame:** Uma estrutura de dados bidimensional, semelhante a uma tabela ou planilha, com colunas de diferentes tipos. É a principal estrutura de dados utilizada na biblioteca Pandas.
 - **Data Science:** Um campo interdisciplinar que utiliza métodos científicos, processos, algoritmos e sistemas para extrair conhecimento e insights de dados.
 - **Decision Trees - Árvores de Decisão:** Um modelo de aprendizado de máquina que usa uma estrutura de árvore para tomar decisões.
 - **Deep Learning:** Um subcampo do aprendizado de máquina que se baseia em redes neurais artificiais com múltiplas camadas.
 - **Diffusion Models - Modelos de Difusão:** Uma classe de modelos generativos que criam dados por meio de um processo de remoção gradual de ruído.
 - **DPO - Direct Preference Optimization:** Uma técnica de ajuste fino para LLMs que usa feedback humano para otimizar diretamente as preferências do modelo.
 - **DQNs - Deep Q-Networks:** Uma abordagem de aprendizado por reforço que usa redes neurais para estimar valores de ação.
-

E

- **Ecossistema (Ecosystem):** No contexto de software, refere-se à rede coletiva de uma linguagem de programação, suas bibliotecas, frameworks, ferramentas e convenções da comunidade, que juntas suportam o desenvolvimento em um domínio específico.
 - **ETL (Extract, Transform, Load):** Processo em computação de dados que envolve a extração de dados de fontes diversas, sua transformação em um formato adequado e o carregamento em um destino final.
-

F

- **Feature Engineering (Engenharia de Atributos):** O processo de usar o conhecimento do domínio dos dados para criar novas variáveis (features) que ajudam os algoritmos de machine learning a funcionar melhor.
 - **Foundation Model (Modelo de Fundação):** Um modelo de IA de grande escala, treinado em uma vasta quantidade de dados, que pode ser adaptado para uma ampla gama de tarefas específicas.
 - **Framework:** Uma estrutura de software que fornece uma arquitetura fundamental e um fluxo de trabalho predefinido para a construção de aplicações. Diferente de uma biblioteca, um framework dita a estrutura geral e chama o código do desenvolvedor em pontos específicos.
-

G

- **GANs - Generative Adversarial Networks:** Um modelo generativo que consiste em duas redes neurais (geradora e discriminadora) que competem para criar dados realistas.
 - **GPT - Generative Pre-trained Transformer:** Um modelo de linguagem baseado na arquitetura Transformer, conhecido por sua capacidade de gerar texto de alta qualidade.
 - **Gradio:** Uma biblioteca Python que permite a criação rápida de interfaces web para modelos de machine learning.
 - **Generative AI - IA Generativa:** Um tipo de inteligência artificial que pode gerar novos conteúdos, como texto, imagens, áudio e vídeo.
 - **Gradient Descent:** Um algoritmo de otimização usado para encontrar o mínimo de uma função, comumente empregado para treinar modelos.
-

H

- **Hugging Face:** Uma plataforma e biblioteca de código aberto que fornece ferramentas para construir, treinar e implantar modelos de ML, especialmente para PNL.
 - **Hyperparameter Tuning (Ajuste de Hiperparâmetros):** O processo de encontrar a combinação ideal de hiperparâmetros (configurações externas ao modelo) para maximizar o desempenho de um modelo de machine learning.
-

K

- **Keras:** Uma biblioteca de rede neural de alto nível, escrita em Python e executada sobre frameworks como TensorFlow.
-

L

- **LangChain:** Uma estrutura que facilita a criação de aplicações baseadas em modelos de linguagem.
 - **Linguagem de Programação (Programming Language):** Um sistema formal de instruções, com regras de sintaxe e semântica, usado para escrever código que um computador pode executar.
 - **Linear Regression - Regressão Linear:** Um algoritmo de ML supervisionado usado para modelar a relação entre variáveis.
 - **LLMs - Large Language Models:** Modelos de linguagem de grande escala treinados em vastas quantidades de dados textuais.
 - **Logistic Regression - Regressão Logística:** Um algoritmo estatístico para prever resultados binários.
 - **Low-Rank Adaptation - LORA:** Uma técnica de ajuste fino eficiente para LLMs que reduz o número de parâmetros a serem treinados.
-

M

- **ML - Machine Learning:** Um subcampo da IA que permite que as máquinas aprendam a partir de dados e melhorem seu desempenho.
 - **MLM - Masked Language Modeling:** Uma técnica de treinamento onde algumas palavras são mascaradas e o modelo precisa prever as palavras originais.
 - **Módulo (Module):** Em Python, um único arquivo contendo código (funções, classes e variáveis). É a unidade básica para organizar e reutilizar código, servindo como bloco de construção para bibliotecas.
 - **Multi-head Attention:** Um mecanismo que permite a um modelo processar diferentes partes da entrada em paralelo.
-

N

- **N-gram:** Uma sequência contígua de n itens de uma amostra de texto.
 - **Neural Networks - Redes Neurais:** Um modelo de computação inspirado no cérebro humano, composto por camadas de nós interconectados.
 - **Next Sentence Prediction - NSP:** Uma tarefa de treinamento para prever se uma sentença segue a outra.
 - **NLP - Natural Language Processing:** Um campo da IA que permite que os computadores entendam, interpretem e gerem a linguagem humana.
 - **NumPy:** Uma biblioteca Python fundamental para computação numérica, especialmente para trabalhar com arrays e matrizes.
-

O

- **One-hot Encoding:** Uma técnica de representação de dados categóricos em que cada valor é transformado em um vetor binário.
-

P

- **Pandas:** Uma biblioteca Python de código aberto para manipulação e análise de dados, oferecendo estruturas de dados como DataFrames.
 - **PEFT - Parameter-Efficient Fine-Tuning:** Uma coleção de técnicas que permite o ajuste fino de modelos grandes de forma mais eficiente.
 - **Positional Encoding:** Uma técnica usada em Transformers para dar ao modelo informações sobre a posição das palavras.
 - **PPO - Proximal Policy Optimization:** Um algoritmo de aprendizado por reforço popular para treinar políticas.
 - **Prompt Engineering:** A arte de criar comandos (prompts) eficazes para modelos de linguagem a fim de obter as respostas desejadas.
 - **PyTorch:** Uma biblioteca de código aberto de aprendizado de máquina usada para aplicações como visão computacional e PNL.
 - **Python:** Uma linguagem de programação de alto nível, amplamente utilizada em ciência de dados e machine learning.
-

Q

- **QA bot - Question Answering bot:** Um programa de computador projetado para responder a perguntas em linguagem natural.
 - **Quantized Low-Rank Adaptation - QLORA:** Uma técnica de ajuste fino que combina a quantização com o LoRA para otimizar o uso de memória.
-

R

- **RAG - Retrieval Augmented Generation:** Uma técnica que combina a recuperação de informações com a geração de texto para fornecer respostas mais precisas e informativas.
 - **Recurrent Networks:** O mesmo que RNNs.
 - **ReLU - Rectified Linear Unit:** Uma função de ativação comumente usada em redes neurais.
 - **Reinforcement Learning:** Um tipo de ML em que um agente aprende a tomar decisões para maximizar uma recompensa.
 - **Restricted Boltzmann Machines:** Um tipo de rede neural estocástica que pode aprender e gerar dados.
 - **RLHF - Reinforcement Learning from Human Feedback:** Uma técnica para alinhar modelos de linguagem com preferências humanas.
 - **RNNs - Recurrent Neural Networks:** Um tipo de rede neural adequado para o processamento de dados sequenciais.
-

S

- **Scikit-learn:** Uma biblioteca Python de código aberto para machine learning, que oferece algoritmos para classificação, regressão e agrupamento.
- **SciPy:** Uma biblioteca Python de código aberto usada para computação científica e técnica.
- **Sigmoid:** Uma função de ativação que mapeia qualquer valor de entrada para um valor entre 0 e 1.
- **Softmax regression:** Uma generalização da regressão logística para problemas de classificação multi-classe.

- **SQL - Structured Query Language:** Uma linguagem de programação para gerenciar e manipular bancos de dados relacionais.
 - **Statistical Modeling - Modelagem Estatística:** O uso de métodos estatísticos para construir modelos que descrevem relações entre variáveis.
 - **Supervised Learning - Aprendizado Supervisionado:** Um tipo de ML onde o modelo é treinado em um conjunto de dados com rótulos.
-

T

- **Tanh:** Uma função de ativação que mapeia valores para o intervalo de -1 a 1.
 - **TensorFlow:** Uma plataforma de código aberto para machine learning, desenvolvida pelo Google.
 - **Text-splitting:** Um processo de dividir grandes documentos de texto em partes menores para facilitar o processamento.
 - **Time series data - Dados de séries temporais:** Um tipo de dado sequencial onde os valores são registrados em intervalos de tempo sucessivos.
 - **Tokenization:** O processo de dividir um texto em unidades menores (tokens).
 - **TorchText:** Uma biblioteca em PyTorch para processamento de texto e dados de linguagem natural.
 - **Transformers:** Uma arquitetura de rede neural que revolucionou o PNL por sua capacidade de processar dados sequenciais de forma eficiente.
-

U

- **Unsupervised Deep Learning:** O uso de deep learning em tarefas de aprendizado não supervisionado.
 - **Unsupervised Learning - Aprendizado Não Supervisionado:** Um tipo de ML onde o modelo encontra padrões em um conjunto de dados sem rótulos.
-

V

- **VAEs - Variational Autoencoders:** Um tipo de modelo generativo que pode gerar novos dados semelhantes aos dados de treinamento.

- **Vanishing Gradient - Gradiente Desvanecente:** Um problema em redes neurais onde os gradientes se tornam extremamente pequenos durante o treinamento.
 - **Vector databases - Bancos de dados de vetores:** Bancos de dados especializados para armazenar e pesquisar vetores (representações numéricas de objetos).
-

W

- **Watsonx:** A plataforma de IA da IBM que ajuda as empresas a dimensionar e automatizar o uso de IA.
- **Word Embeddings:** Representações de palavras como vetores numéricos, capturando seu significado e contexto.
- **Word2Vec:** Um modelo popular para criar embeddings de palavras.