

AULA 1 – INTRODUÇÃO AO APRENDIZADO DE MÁQUINA (MACHINE LEARNING)

1. Introdução: Programação Tradicional vs. Aprendizado de Máquina

Para compreender como um computador "aprende", é necessário primeiro distinguir esse processo da programação convencional que domina o desenvolvimento de software clássico.

1.1 O Paradigma Tradicional

No desenvolvimento de sistemas convencional, o programador codifica regras explícitas. Se quisermos criar um programa que converta temperaturas, escrevemos a fórmula exata ($F = C \times 1.8 + 32$).

- **Entrada:** Dados + Regras (Código).
- **Processamento:** O computador executa as regras sobre os dados.
- **Saída:** A resposta correta.

1.2 O Paradigma de Aprendizado de Máquina (Machine Learning)

Em problemas complexos (como reconhecimento facial ou detecção de spam), é impossível escrever regras manuais para todas as variações possíveis. Aqui, invertemos o fluxo.

- **Entrada:** Dados + Respostas (Histórico).
- **Processamento:** O computador analisa os dados e as respostas para criar as regras.
- **Saída:** Um Modelo (o programa que contém as regras aprendidas).

Definição Formal: Segundo Tom Mitchell (1997), diz-se que um programa de computador aprende com a experiência **E** em relação a uma classe de tarefas **T** e medida de desempenho **P**, se o seu desempenho em **T**, medido por **P**, melhora com a experiência **E**.

2. A Matéria-Prima: Dados e Atributos

O computador não aprende a partir do "nada". Ele aprende identificando padrões matemáticos em grandes volumes de dados.

2.1 Terminologia Essencial

Imagine uma tabela de banco de dados contendo informações sobre imóveis vendidos.

- **Dataset (Conjunto de Dados):** A tabela inteira.
- **Instância (Exemplo/Amostra):** Cada linha da tabela (um imóvel específico).
- **Features (Atributos/Características):** As colunas usadas para aprender (ex: número de quartos, metragem, bairro).
- **Target (Alvo/Rótulo):** A coluna que queremos prever (ex: preço do imóvel).

2.2 O Processo de Generalização

O objetivo do computador não é memorizar os dados (isso seria apenas um banco de dados), mas sim **generalizar**. Generalizar significa criar uma função matemática capaz de prever corretamente o resultado para dados que o computador nunca viu antes.

3. Tipos de Aprendizado

A forma como o computador aprende depende de como os dados são apresentados a ele. Existem três paradigmas principais:

3.1 Aprendizado Supervisionado

O sistema é treinado com dados "rotulados". É como um professor ensinando um aluno com um gabarito. Mostramos a imagem de um gato e dizemos "isto é um gato".

- **Tarefas comuns:**
 - **Classificação:** O alvo é uma categoria (Sim/Não, Gato/Cachorro, Spam/Não Spam).
 - **Regressão:** O alvo é um valor numérico contínuo (Preço de ações, temperatura, peso).

3.2 Aprendizado Não Supervisionado

O sistema recebe dados sem rótulos (sem respostas). O objetivo é explorar os dados para encontrar estruturas ocultas. O computador tenta "arrumar a bagunça" sozinho.

- **Tarefas comuns:**
 - **Agrupamento (Clustering):** Separar clientes em grupos de consumo baseados em comportamento de compra, sem saber a priori quais são esses grupos.

3.3 Aprendizado por Reforço

O computador (agente) aprende interagindo com um ambiente dinâmico. Ele toma uma ação e recebe um *feedback* (recompensa ou punição). O objetivo é maximizar a recompensa a longo prazo.

- **Exemplos:** Carros autônomos, robôs industriais, IAs de jogos (xadrez, Dota).
-

4. Como o Algoritmo Funciona: O Mecanismo Matemático

Para um técnico em desenvolvimento, é vital entender que "aprendizado" é, na verdade, **otimização matemática**. Vamos analisar como isso ocorre de forma simplificada.

4.1 A Função de Custo (Loss Function)

Quando o computador faz uma previsão durante o treino, ele quase sempre erra no início.

1. O modelo chuta um valor.
2. Compara-se o chute com o valor real.

3. Calcula-se o **Erro** (a diferença entre o chute e o real).

Essa medida de erro é chamada de **Função de Custo**. O objetivo de todo algoritmo de aprendizado de máquina é **minimizar a Função de Custo**.

4.2 O Ajuste de Pesos

Os modelos possuem parâmetros internos (chamados de pesos, denotados geralmente por w).

- Se o erro foi alto, o algoritmo ajusta os pesos matematicamente (aumenta ou diminui um pouco os valores).
- Esse processo se repete milhares de vezes (épocas) até que o erro seja o menor possível.

4.3 Exemplo: Redes Neurais Artificiais

Inspiradas nos neurônios biológicos, são compostas por camadas de nós conectados.

1. **Entrada:** Recebe os números (pixels de uma imagem, por exemplo).
2. **Camadas Ocultas:** Multiplicam a entrada por "pesos" e aplicam funções de ativação.
3. **Saída:** Entrega a probabilidade (ex: 90% de chance de ser um cachorro).

5. O Pipeline de Desenvolvimento

No dia a dia de um desenvolvedor trabalhando com IA, o fluxo de trabalho (pipeline) segue estas etapas rigorosas:

1. **Coleta de Dados:** Obtenção de dados brutos (banco de dados, APIs, sensores).
2. **Pré-processamento:**
 - Limpeza (remover dados nulos).
 - Normalização (colocar todos os números na mesma escala, ex: entre 0 e 1).
 - Conversão (transformar texto em números).
3. **Divisão dos Dados (Split):**
 - **Treino (70-80%):** Usado para o computador aprender.
 - **Teste (20-30%):** Usado para verificar se ele aprendeu mesmo (dados que ele nunca viu).
4. **Treinamento:** O algoritmo processa os dados de treino.
5. **Avaliação:** Mede-se a precisão usando os dados de teste.
6. **Deploy (Implantação):** Integração do modelo treinado ao software final (web, mobile, desktop).

6. Desafios e Ética

O computador aprende o que está nos dados. Se os dados forem enviesados, o aprendizado será enviesado.

- **Overfitting (Sobreaajuste):** Quando o modelo "decora" os dados de treino e não consegue generalizar para novos dados. Funciona bem no treino, mas falha no mundo real.
 - **Viés Algorítmico:** Se treinarmos um sistema de contratação apenas com currículos de homens, o sistema aprenderá a descartar mulheres, reproduzindo preconceitos humanos.
-

7. Referências Bibliográficas

As seguintes obras são fundamentais para o aprofundamento técnico no tema:

1. **RUSSELL, Stuart; NORVIG, Peter.** *Inteligência Artificial: Uma Abordagem Moderna*. 3. ed. Rio de Janeiro: Elsevier, 2013. (A "bíblia" da área).
2. **HAYKIN, Simon.** *Redes Neurais: Princípios e Prática*. 2. ed. Porto Alegre: Bookman, 2001.
3. **FACELI, Katti et al.** *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro: LTC, 2011. (Excelente referência em português).
4. **MITCHELL, Tom M.** *Machine Learning*. New York: McGraw-Hill, 1997.