

data

Curso de Machine Learning

Glauco Fleury Corrêa de Moraes,
@GlauqueraFleury



Pré-Requisitos



Pré-Requisitos

- Conhecimento básico na linguagem Python
- Conhecimento básico de álgebra linear
 - Vetores, matrizes, e suas operações.

Aos que se sentirem um pouco inseguros com seu conhecimento a respeito desses tópicos, não se preocupem: basta perguntar e nós responderemos :)



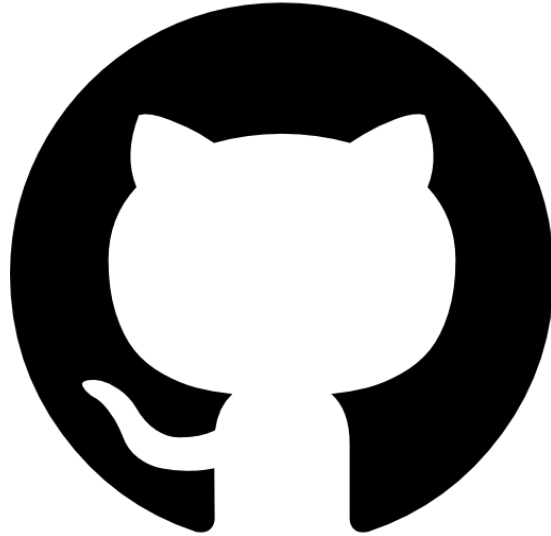


GitHub



GitHub

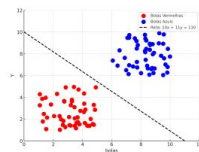
- <https://github.com/icmc-data/Curso-de-Machine-Learning-2025>



Material

- <https://github.com/icmc-data/Curso-de-Machine-Learning-2025>

Uma simples reta é capaz de separar perfeitamente o nosso conjunto de dados entre bolas vermelhas e bolas azuis, como mostrado a baixo:



Sendo assim, tudo que está a baixo dessa reta será classificado como bola vermelha e tudo que está a cima será classificado como bola azul.

Se o conjunto de dados é linearmente separável o nosso objetivo será encontrar qual é a reta que faz a melhor separação entre as classes.

Vetor de características (Feature Vector X)

Cada exemplo do nosso conjunto de dados será representado por um vetor de características (feature vector) X . Tal vetor irá conter as características do nosso exemplo. No exemplo das bolas o feature vector tem dimensionalidade 2: $X = \{x_1, x_2\}$, sendo x_1 a coordenada do ponto no eixo horizontal e x_2 a coordenada do ponto no eixo vertical.

2.1 Classificadores lineares que passam pela origem

Em um classificador binário, a decision boundary é a reta (\mathbb{R}^2), plano (\mathbb{R}^3) ou hiperplano ($\mathbb{R}^n, n > 3$) que separa a parte que será positiva (bolas azuis) da parte que será negativa (bolas vermelhas).

Inicialmente podemos começar a definir o conjunto de todos os classificadores lineares que passam pela origem por questão didática antes de generalizar para todos os casos.

Para o \mathbb{R}^2 , com o feature vector sendo $X = \{x_1, x_2\}$, o conjunto de todas as retas que passam pela origem são:

$$\theta_1 x_1 + \theta_2 x_2 = 0$$

Onde $\theta = \{\theta_1, \theta_2\}$ é chamado de vetor de parâmetros. Esse vetor de parâmetros é o vetor que o nosso algoritmo de machine learning deve encontrar baseado nos dados para fazer as classificações.

De uma forma mais generalizada, podemos dizer que o conjunto de todos os classificadores lineares que passam pela origem é o produto escalar entre os vetores theta e X:

$$\theta \cdot X = 0$$

Para o valor $\theta = (1, 1)$, a reta seria $x_1 = -x_2$, dando:





Presença



Presença

- Linktree: Presente na bio do nosso instagram
- Presença ficará disponível até 1 hora antes da próxima aula
- É necessário 70% de presença para obter o certificado



Presença e Github

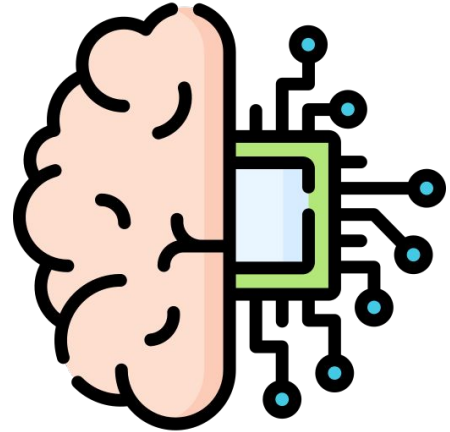


Introdução à área



O que é Machine Learning?

- Criação de modelos que aprendam a partir de dados (superar barreiras antigas da computação)
- Identificação de padrões pelas máquinas
- Decisões automáticas

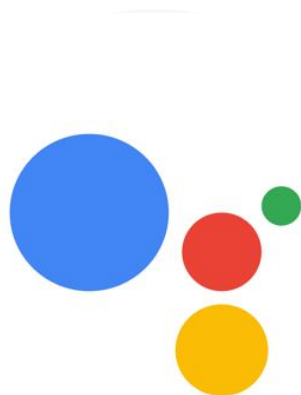


Qual a importância de Machine Learning?

- Previsão de tendências e diagnósticos
- Otimização de processos industriais
- Tomada de decisão mais eficiente



Exemplos do Dia a Dia



Assistentes virtuais



Recomendação de
filmes



Carros autônomos



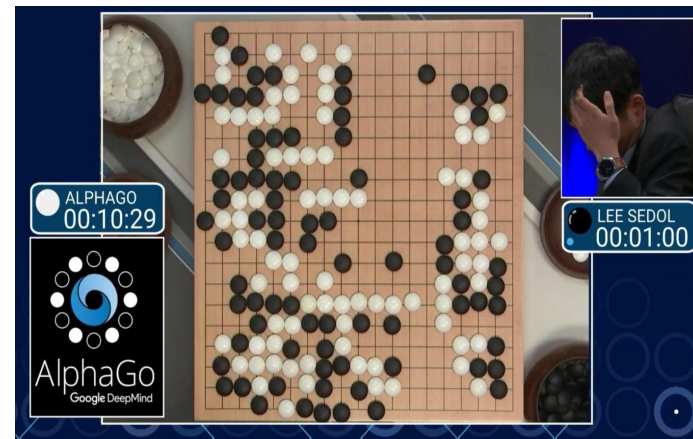
Aplicações fantásticas



Alpha Fold



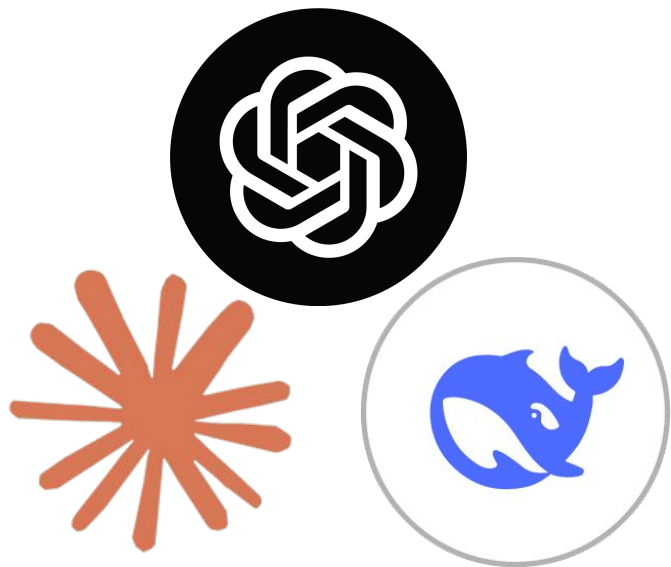
Deep Blue vs
Kasparov



AlphaGo



IA mais famosa: Chat GPT (e seus “amigos”)



ChatBots inteligentes



IAs generativas

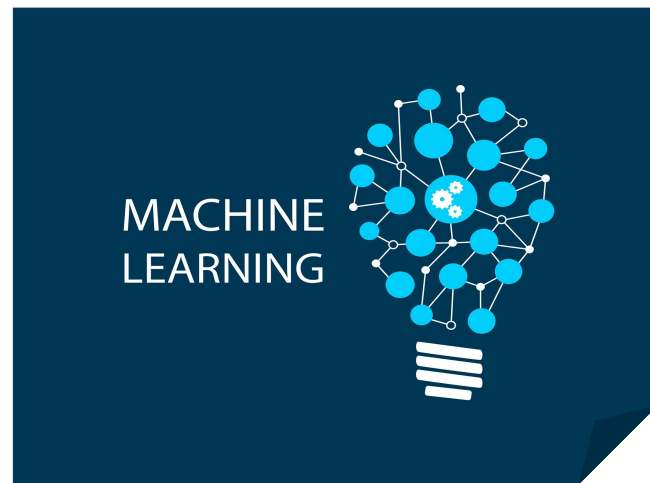
Características das IAs generativas como o GPT

- usos comuns: debuggar, redações, problemas matemáticos, imagens, desenhos, etc.
- muito custosas para treinar (1.8 trilhões de parâmetros no GPT-4); projeto StarGate
- tipos: LLM's, GAN's, Diffusion, etc.

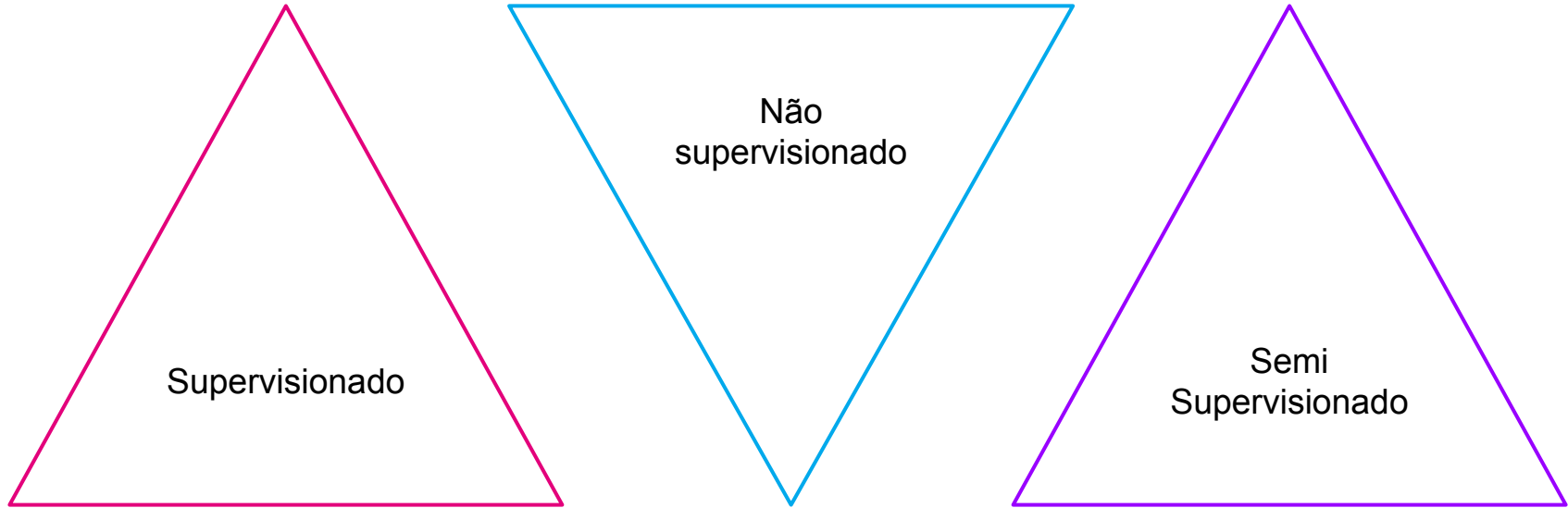


Conceitos Básicos de como Funciona

- Dados e características (features)
- Ajuste de parâmetros contínuo
- Aprender com erros e acertos

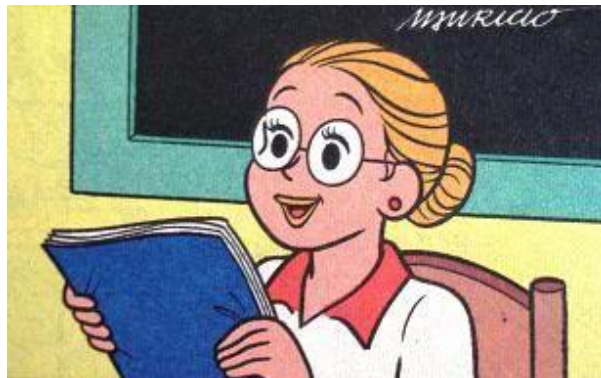


Paradigma Geral: Tipos de Aprendizado

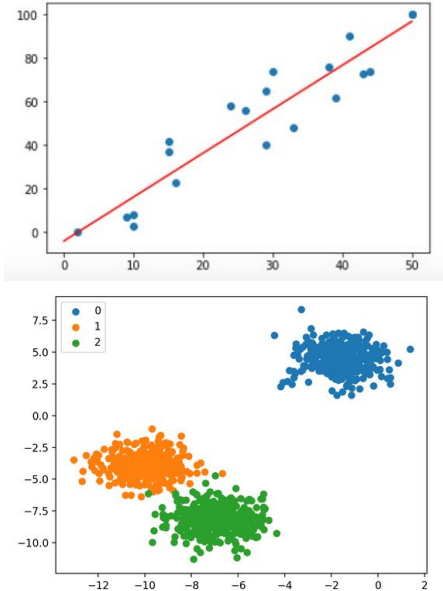


Entrando em mais detalhes

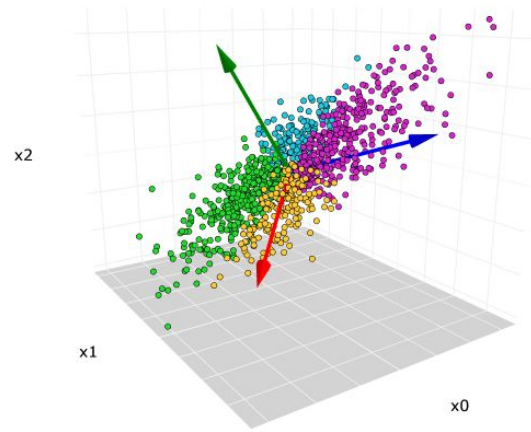
- **Supervisionado:** uso exclusivo de labels
- **Não-Supervisionado:** o modelo descobre sozinho os padrões com base nos dados
- **Semi-Supervisionado:** uma mistura de ambos



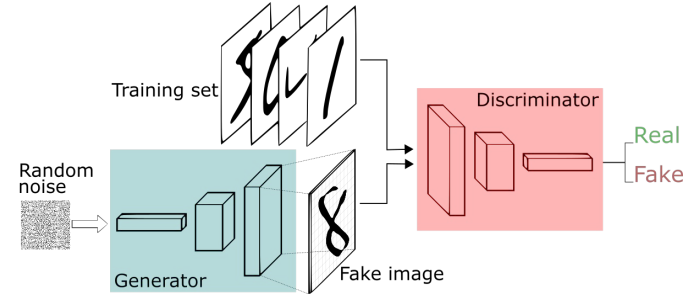
Exemplos



Supervised:
Regressão e
Classificação



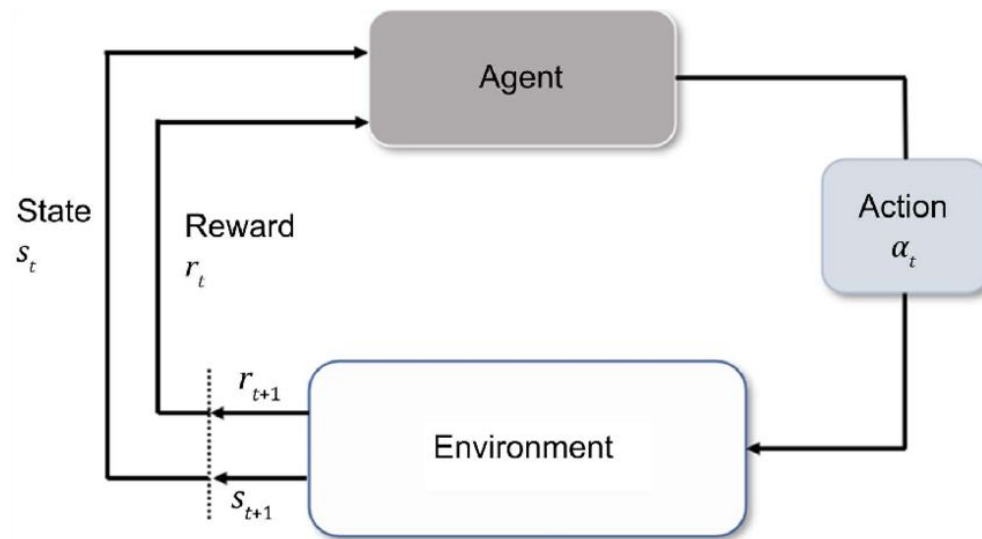
Unsupervised:
PCA



Semi-Supervised:
GAN



Reinforcement Learning: um pouco fora do paradigma



- Para compreender como os modelos mais avançados de IA funcionam na atualidade, precisamos ter uma noção incipiente de como seus predecessores eram
- Começaremos a partir de modelos de aprendizado supervisionado: exploraremos classificadores lineares!





Classificação



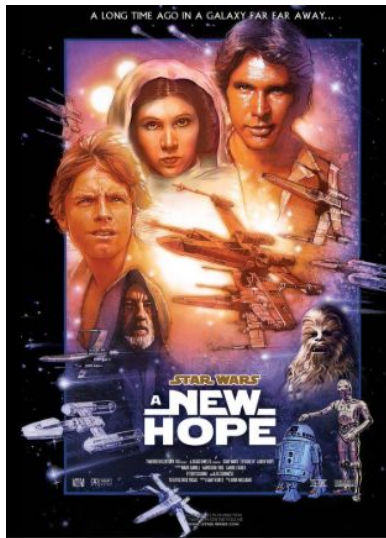
Classificação

- Cada dado está relacionado a uma classe
- Exemplo:
 - Classificar se um filme é bom ou ruim

Como caracterizar um filme? Em outras palavras, quais “features” escolhemos para analisar gostos e compor nosso vetor, que representará os nossos dados?



Como caracterizar um Dado



(1, 1977, 9.3)



(1, 2023, 9.7)



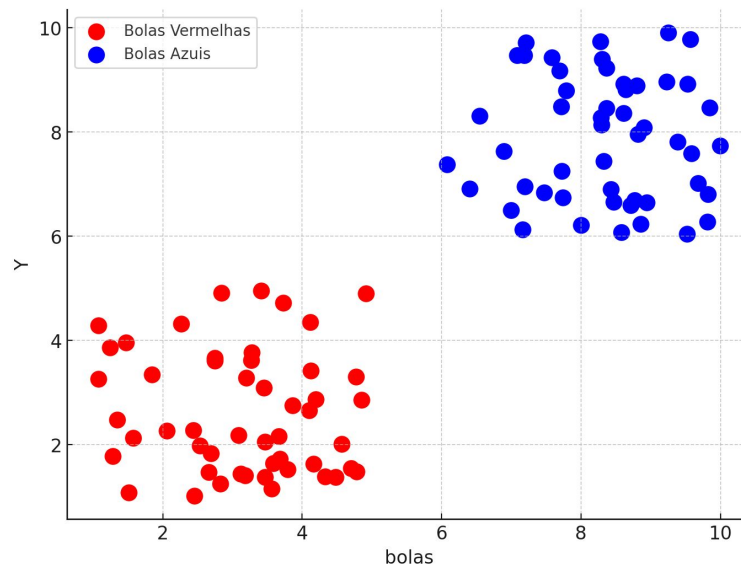
(0, 2017, 3.4)

exemplo: ([ganhou oscar], [ano de lançamento], [avaliação do imdb])



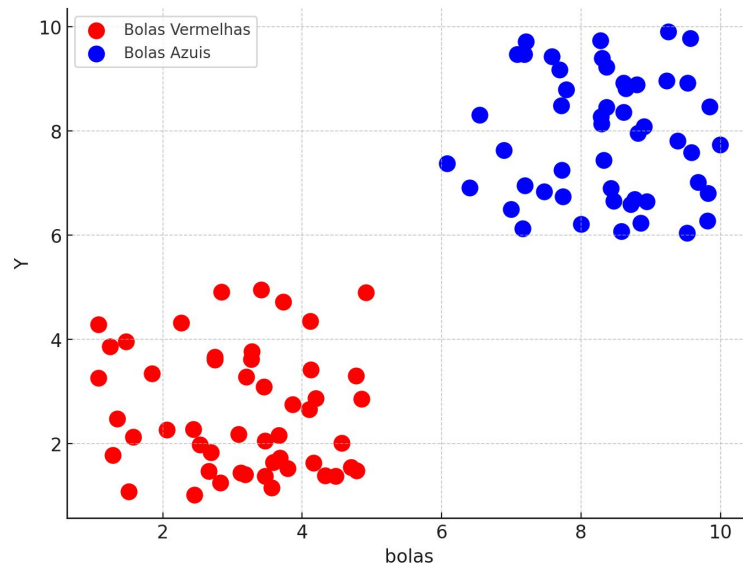
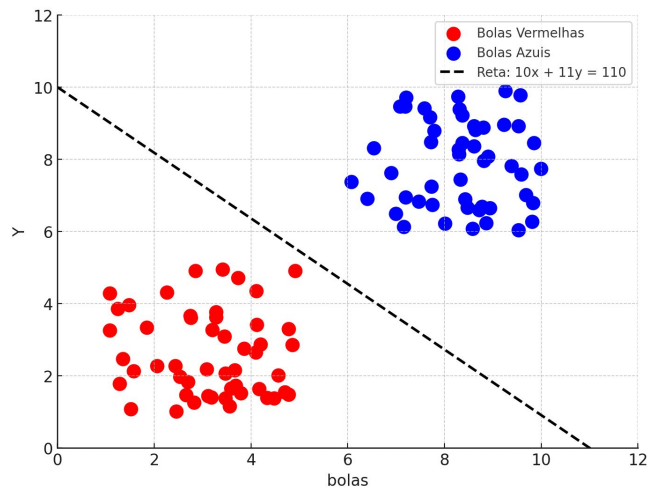
Classificador Linear

- Classifica entre dois tipos de dados
- Dados linearmente separáveis



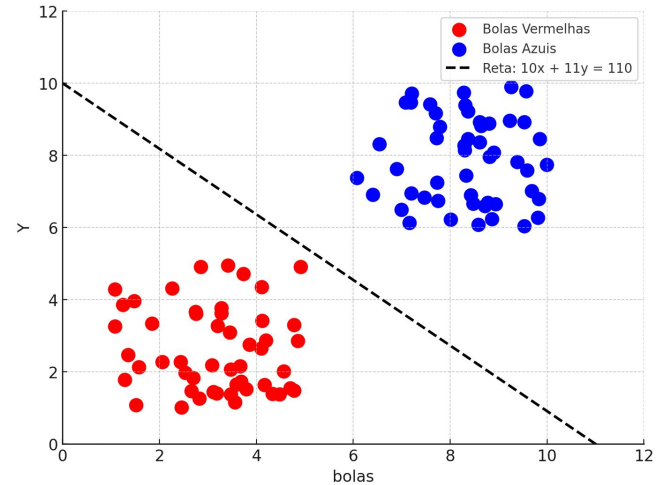
Classificador Linear

- Classifica entre dois tipos de dados
- Dados linearmente separáveis

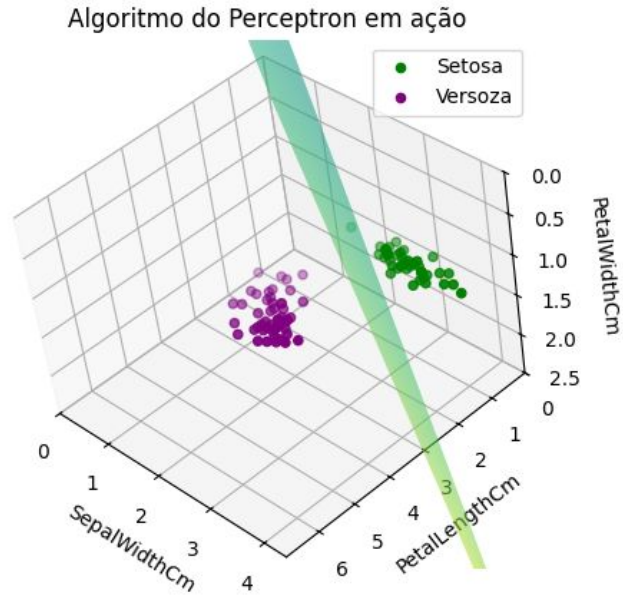


Classificador Linear

- Objetivo: encontrar a função linear
- A dimensão do vetor de característica dita se será uma reta (\mathbb{R}^2), ou um hiperplano (\mathbb{R}^N)
- Iremos trabalhar com \mathbb{R}^2 , mas toda a teoria se aplica para \mathbb{R}^N (hiperplanos)

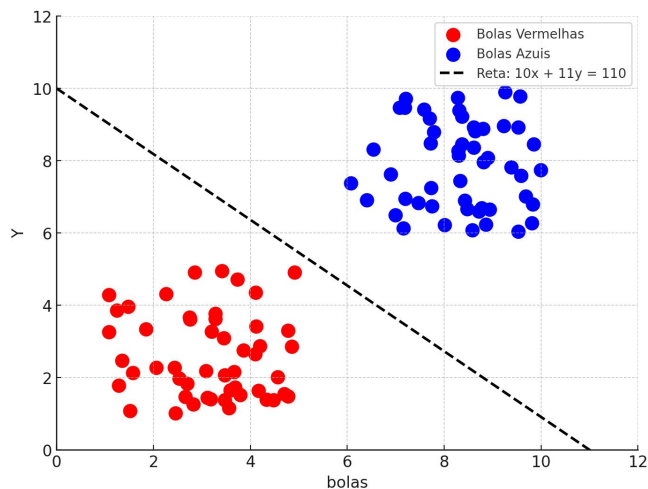


Classificador Linear: exemplo 3D



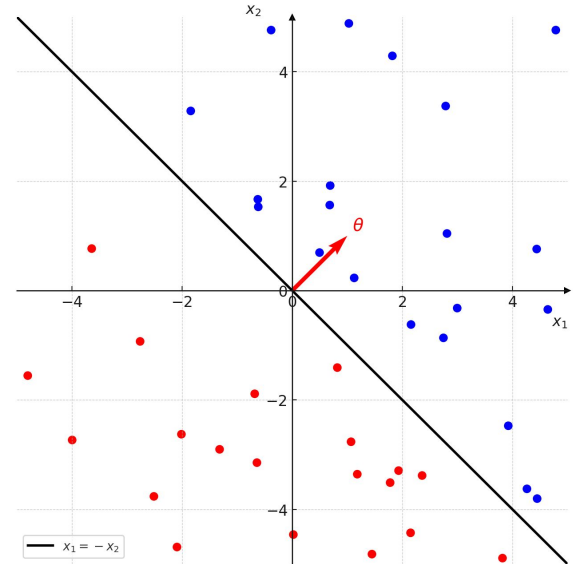
Decision Boundary

- A reta, ou hiperplano é o que chamamos de Decision Boundary
- Ele delimita os pontos positivos (azuis) dos negativos (vermelhos)



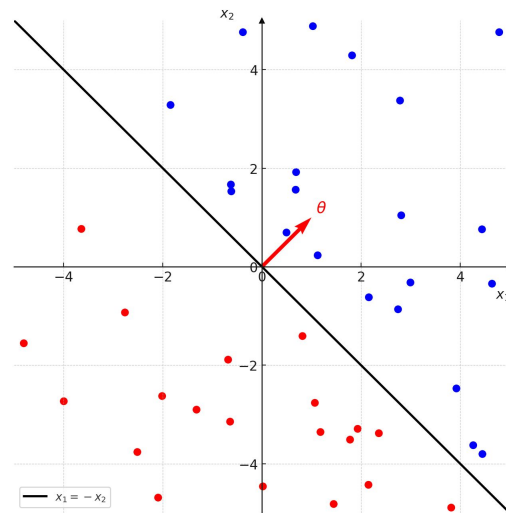
Vetor de parâmetros

- O vetor de parâmetros descreve a decision boundary
- Feature vector genérico: $\vec{x} = \{x_1, x_2\}$



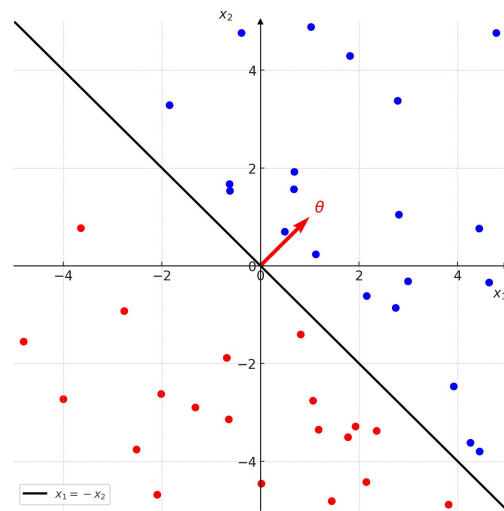
Vetor de parâmetros

- O vetor de parâmetros descreve a decision boundary
- Feature vector genérico: $\vec{x} = \{x_1, x_2\}$
- Vetor de parâmetros: $\vec{\theta} = \{1, 1\}$ [exemplo]



Vetor de parâmetros

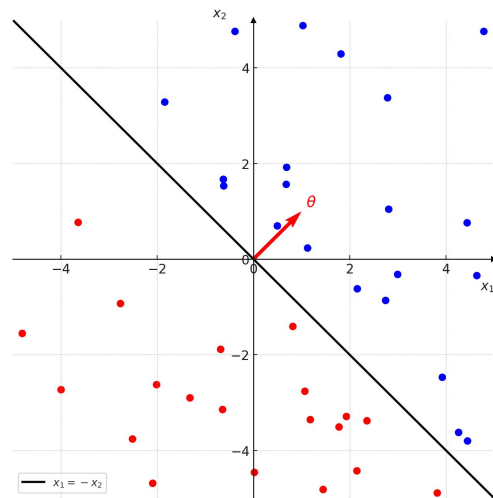
- O vetor de parâmetros descreve a decision boundary
- Feature vector genérico: $\vec{x} = \{x_1, x_2\}$
- Vetor de parâmetros: $\vec{\theta} = \{1, 1\}$ (exemplo)
- Isso define a reta: $1 \cdot x_1 + 1 \cdot x_2 = 0$



Vetor de parâmetros

- O vetor de parâmetros descreve a decision boundary
- Feature vector genérico: $\vec{x} = \{x_1, x_2\}$
- Vetor de parâmetros: $\vec{\theta} = \{1, 1\}$ [exemplo]
- Isso define a reta: $1 \cdot x_1 + 1 \cdot x_2 = 0$
- Podemos escrevê-la como: $\theta_1 x_1 + \theta_2 x_2 = 0$

$$\vec{\theta} \cdot \vec{x} = 0$$



Classificadores Lineares que passam pela origem

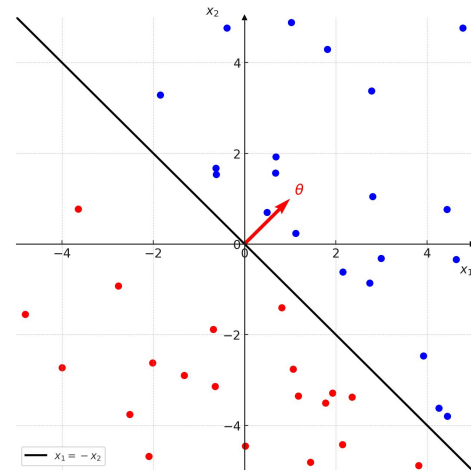
- Conjunto de todos os classificadores lineares que passam pela origem: $\vec{\theta} \cdot \vec{x} = 0$; o produto escalar decidirá as classes

- Bolas azuis:** $\vec{\theta} \cdot \vec{x} > 0$

$$\vec{x} = \{1, 2\}, \vec{\theta} = \{1, 1\}: \vec{\theta} \cdot \vec{x} = 3$$

- Bolas vermelhas:** $\vec{\theta} \cdot \vec{x} < 0$

$$\vec{x} = \{-2, 1\}, \vec{\theta} = \{1, 1\}: \vec{\theta} \cdot \vec{x} = -1$$



A projeção entre o vetor de parâmetros e o nosso dado

decidirá a qual classe ele pertence: isso ocorre via o dot product





Perceptron



Algoritmo Perceptron

- Ideia:
 - Inicializar o vetor de parâmetros como 0: $\vec{\theta} = 0$
 - Iterar sobre o conjunto de dados fazendo operações, ajustando $\vec{\theta}$ sempre que um resultado for mal classificado
 - No final das iterações teremos construído a decision boundary



Loss Function: noções gerais

$$L(\vec{x}_i, y_i) = y_i(\theta \cdot \vec{x}_i) \leq 0$$

$$\begin{cases} y_i > 0 \text{ and } \theta \cdot x_i > 0 : L(y_i, x_i) > 0 \text{ (acerto)} \\ y_i > 0 \text{ and } \theta \cdot x_i < 0 : L(y_i, x_i) < 0 \text{ (erro)} \\ y_i < 0 \text{ and } \theta \cdot x_i > 0 : L(y_i, x_i) < 0 \text{ (erro)} \\ y_i < 0 \text{ and } \theta \cdot x_i < 0 : L(y_i, x_i) > 0 \text{ (acerto)} \end{cases}$$

o “y” é a nossa label: +1 ou -1 (classificação binária)



Algoritmo Perceptron

$\theta = 0$

for i in range (n)

 if $y_i(\theta \cdot X_i) \leq 0$

Atualizar θ



Atualização $\vec{\theta}$.

- $\vec{\theta}_{novo} = \vec{\theta}_{antigo} + \vec{x}_1 y_1$
- $\vec{\theta}_{novo} = 0 + \vec{x}_1 y_1, \vec{\theta}_{novo} = \vec{x}_1 y_1$ (1ª atualização)
- Caso o exemplo x_1 seja visto novamente:
 - $y_1(\vec{\theta}\vec{x}_1) = y_1([\vec{x}_1 y_1]\vec{x}_1) = (y_1)^2(\vec{x}_1)^2 = 1(\vec{x}_1)^2$
 - Valor sempre positivo (acerto)



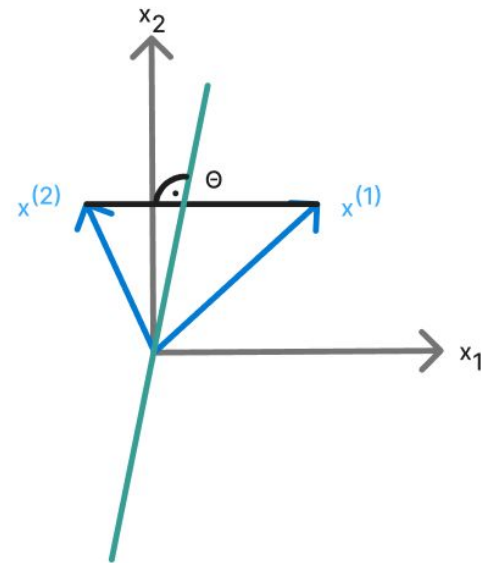
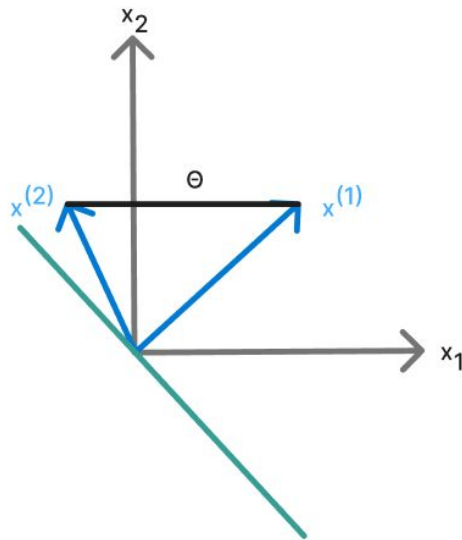
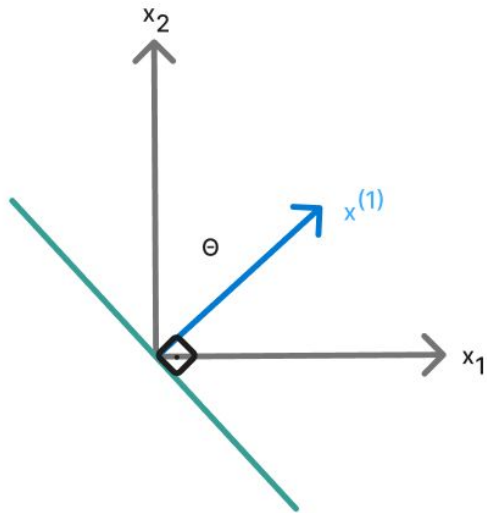
Algoritmo Perceptron

- 1. $\vec{\theta} = 0$
- 2. for t in range (T) :
- 3. for i in range (n) :
- 4. if $y_i(\vec{\theta} \cdot \vec{x}_i) \leq 0$
- 5. $\vec{\theta} = \vec{\theta} + y_i \vec{x}_i$

T: fazer com que os ajustes prossigam até que o parâmetro vá convergir no resultado ideal (se o dataset é lin.separável)



Intuição geométrica

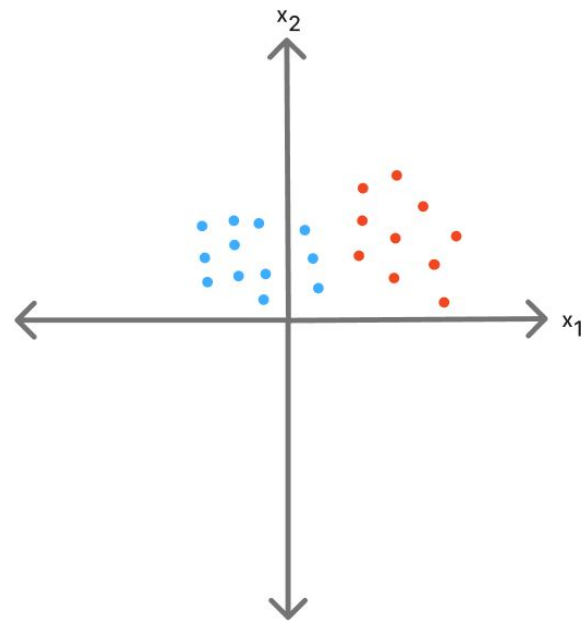


Todos os Classificadores lineares

- Nenhum classificador do formato:

$\vec{\theta} \cdot \vec{x} = 0$ é capaz de separar perfeitamente
o conjunto de dados ao lado

- Deslocar do centro
- Adicionar o bias: θ_0
 - $\vec{\theta} \cdot \vec{x} + \theta_0 = 0$



Noção geométrica do theta zero

- $\vec{\theta} \cdot \vec{x} + \theta_0 = 0$
- $\vec{\theta}$ é um vetor (dimensão de \vec{x})
 - $\vec{\theta}$ altera a direção e orientação do classificador
- θ_0 é um número real
 - θ_0 altera a altura (eixo vertical) do classificador



Perceptron Final

- 1. $\vec{\theta} = 0$
- 2. for t in range (T) :
- 3. for i in range (n) :
- 4. if $y_i(\vec{\theta} \cdot \vec{x}_i) \leq 0$
- 5. $\vec{\theta} = \vec{\theta} + y_i \vec{x}_i$
- 6. $\theta_0 = \theta_0 + y_i$



Hora de mostrar o código!



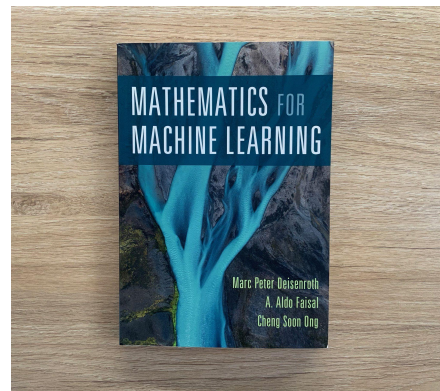
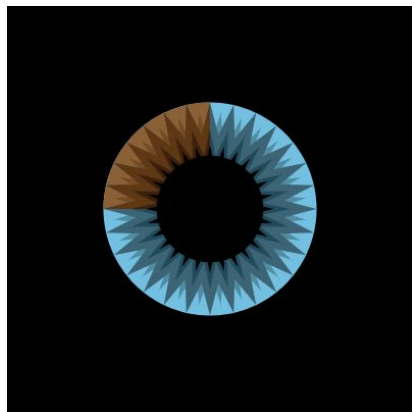
Próxima aula

- Definição e uso de otimização em algoritmos de Machine Learning
- Margin Boundaries: conceito básico para o entendimento de Support Vector Machines



Recomendações interessantes

- Séries do canal 3B1B sobre Álgebra Linear e Cálculo (base matemática)
- Pra quem gosta de ler: Mathematics for Machine Learning (passar por cima da parte matemática); aviso, é um livro longo





data@icmc.usp.br



@data.icmc



/c/DataICMC



/icmc-data



data.icmc.usp.br



obrigado por sua
presença!

