# Muito além de “treinar e testar”

## A Matemática por trás do aprendizado de máquinas

Thiago Rodrigo Ramos

28 de Março, 2025

# Começando a viagem


Um drink no inferno, 1996.
Tarantino e George Clooney fugindo da polícia.

# Muito além de treinar e testar?

**Problema de aprendizado supervisionado:**

- Espaço de entrada: $\mathcal{X} \subset \mathbb{R}^p$.
- Espaço de saída: $\mathcal{Y} \subset \mathbb{R}$.
- Variáveis aleatórias $(X, Y) \sim \mathcal{D}$, com distribuição desconhecida $\mathcal{D}$ sobre $\mathcal{X} \times \mathcal{Y}$
- Conjunto de dados de treino: $\{(X_i, Y_i)\}_{i=1}^n \sim \mathcal{D}^n$, assumimos i.i.d.

# O que significa i.i.d.?

**i.i.d. = independent and identically distributed**

- **Identically distributed:** todas as amostras $(X_i, Y_i)$ seguem a mesma distribuição $\mathcal{D}$.
  - $\mathbb{P}_{(X_i,Y_i)} = \mathcal{D}$ para todo $i$
- **Independentes:** o valor de uma amostra não influencia as outras.
  - $\mathbb{P}((X_1, Y_1), \ldots, (X_n, Y_n)) = \prod_{i=1}^{n} \mathbb{P}(X_i, Y_i)$

**Exemplo:**

- Cada amostra é uma imagem de um animal, com $X_i$ sendo os pixels da imagem e $y_i \in \{\text{gato}, \text{cachorro}\}$. Nossos dados tem a mesma distribuição!
- Suponha que coletamos $n$ imagens aleatórias — então, i.i.d.!
- Mas se tirarmos várias fotos do mesmo cachorro em sequência, os dados não são mais independentes!

# Exemplo: Variáveis contínuas (Regressão)

**Tarefa:** Estimar a **renda mensal** de uma pessoa com base em características individuais.

| Idade | Nível de escolaridade | Horas de trabalho | Renda (R$) |
|---|---|---|---|
| 25 | Ensino médio | 40 | 2500 |
| 45 | Pós-graduação | 50 | 9000 |
| 30 | Graduação | 35 | 4500 |
| 40 | Ensino médio | 30 | 3200 |

- Cada linha representa uma amostra $(X_i, y_i)$
- $X_i$ contém atributos como idade, escolaridade e carga horária
- $y_i \in \mathbb{R}$ representa a renda: uma variável contínua

**Esse tipo de problema é chamado de *regressão*.**

# Exemplo: Variáveis discretas (Classificação)

**Tarefa:** Prever se uma pessoa possui uma determinada doença com base em exames clínicos.

| Idade | IMC | Pressão arterial | Doença? ($y$) |
|---|---|---|---|
| 52 | 28.3 | Alta | 1 |
| 33 | 22.1 | Normal | 0 |
| 45 | 30.0 | Alta | 1 |
| 27 | 19.5 | Normal | 0 |

- Atributos $X_i$: idade, índice de massa corporal (IMC), pressão arterial
- Resposta $y_i \in \{0, 1\}$ indica ausência (0) ou presença (1) da doença
- Esse é um típico problema de **classificação binária**

# Nosso objetivo

**Dado um conjunto de dados $\{(X_i, Y_i)\}_{i=1}^n$, queremos encontrar uma função $f$ tal que:**

$$f(X_i) \approx Y_i \quad \text{para todo } i$$

- Essa função $f$ é escolhida dentro de uma família de funções possíveis, chamada de **hipóteses**, denotada por $\mathcal{H}$
- Mas o que exatamente significa "$f(X) \approx Y$"?

**Para isso, precisamos formalizar uma noção de erro — uma *função de perda*.**

# Função de perda: quantificando o erro

**Dado $f : \mathcal{X} \to \mathcal{Y}$, definimos uma *função de perda* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ que mede o erro da predição:**

$$\ell(f(X), Y)$$

**Exemplos:**

- Regressão: $\ell(f(X), Y) = (f(X) - Y)^2$ (erro quadrático)
- Classificação binária: $\ell(f(X), Y) = \mathbf{1}\{f(X) \neq Y\}$ (perda 0-1)

**Risco teórico (esperado):**

$$L(f) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(f(X), Y)]$$

**Risco empírico:**

$$\hat{L}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), Y_i)$$

**Queremos encontrar $f$ que minimize $L(f)$ — mas só temos acesso a $\hat{L}_n(f)$.**

# Escolher uma boa função $f$

**Temos:** um conjunto de dados $\{(X_i, Y_i)\}_{i=1}^{n}$.

- Nosso objetivo é encontrar uma função $f$ tal que $f(X) \approx Y$.
- Para isso, escolhemos $f$ com base nos dados disponíveis.

**Mas isso levanta duas questões fundamentais:**

1. **Como escolher $f$ com base nos dados?**
2. **Depois de escolher $f$, como saber se fizemos uma boa escolha?**

# Uma primeira ideia

**Dado o conjunto de dados $\{(X_i, Y_i)\}_{i=1}^n$, uma ideia natural seria:**

1. Usar **todo o conjunto** para encontrar a função $f$
2. Avaliar o desempenho de $f$ no **mesmo conjunto**

**Será que isso é uma boa ideia?**

*Vamos ver um exemplo...*

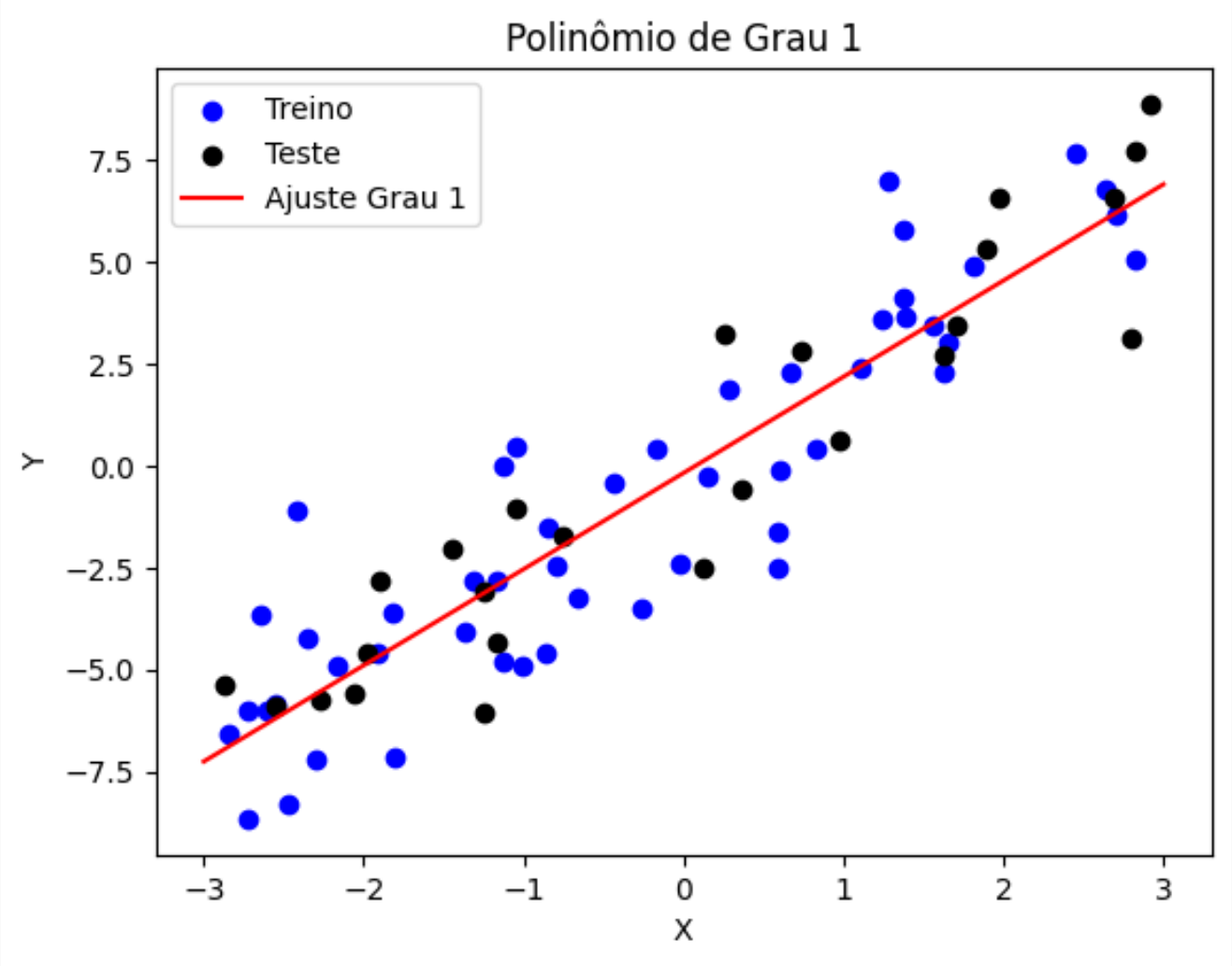
# Um exemplo: Regressão com ruído

**Suponha que os dados foram gerados por:**

$$Y = aX + \text{erro}, \quad a \in \mathbb{R}$$

- Nosso objetivo é encontrar uma função $f$ que capture essa relação subjacente
- Não sabemos que a relação verdadeira é linear
- Tudo o que temos são os pares $(X_i, Y_i)$

# Um exemplo: Regressão com ruído



# Um exemplo: Regressão com ruído

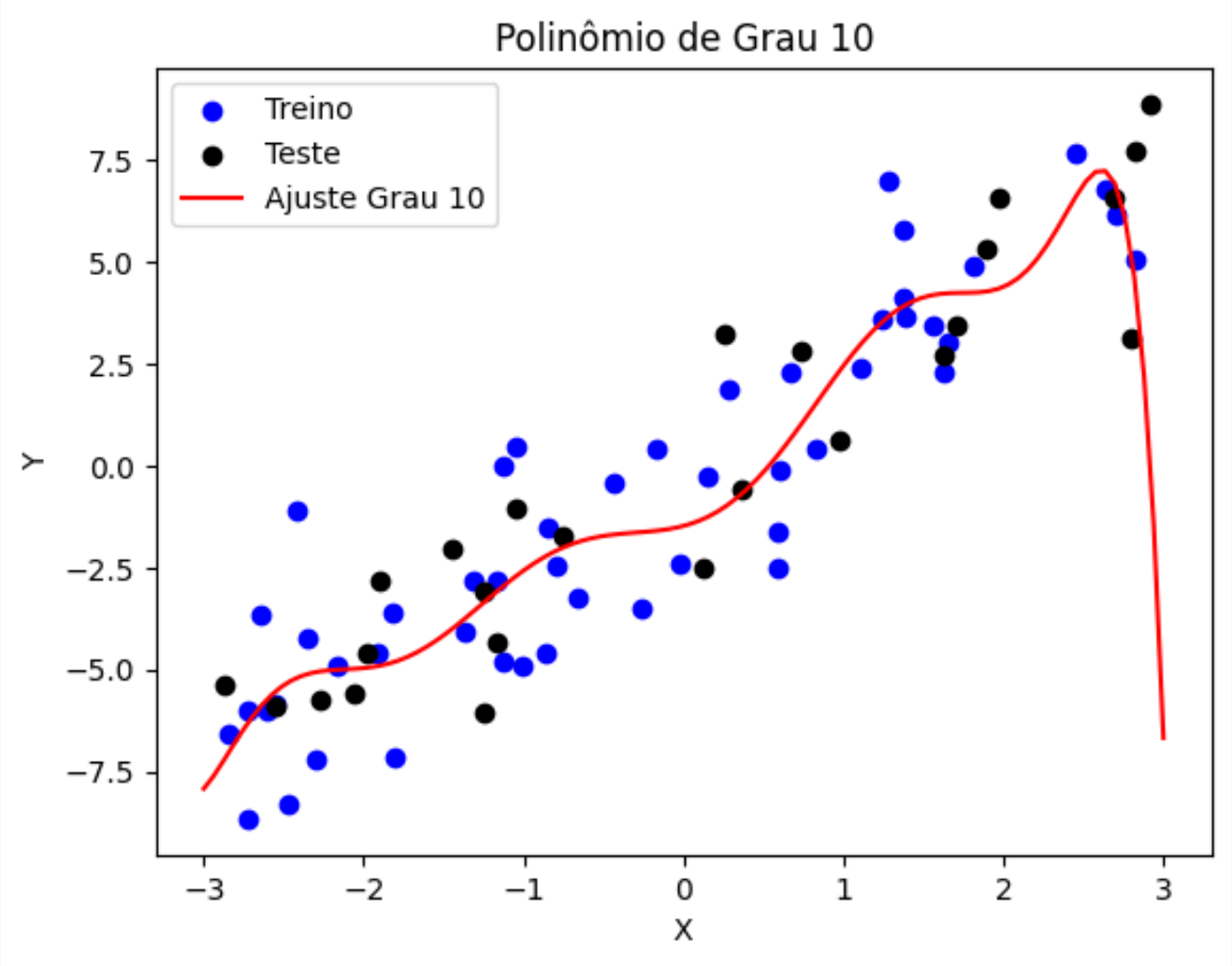
**Ideia:** vamos escolher a função $f$ que minimiza o erro *no próprio conjunto de dados*.

**Problema:**

- A função que minimiza perfeitamente o erro no treino pode simplesmente **interpolar os dados**, ajustando-se exatamente a cada ponto observado.
- Isso pode resultar em uma função extremamente complexa e irregular.
- Embora o erro no treino seja zero, essa função pode se sair muito mal em novos dados!

**Ou seja:** minimizar apenas o erro no treino *não garante boa generalização*.

# Um exemplo: Regressão com ruído



# Separação em treino e teste

**Solução prática: dividir os dados em dois conjuntos distintos:**

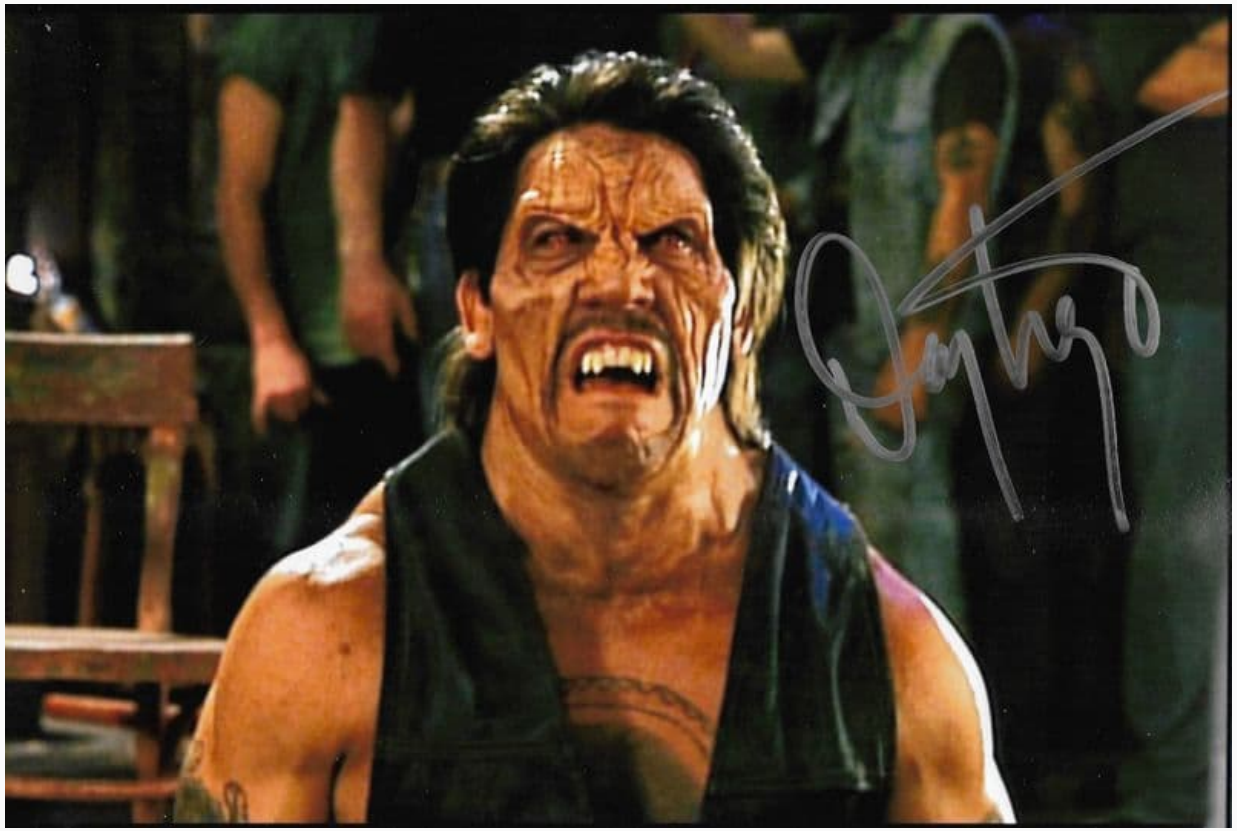
- **Conjunto de treino:** usado para encontrar uma boa função $f$
- **Conjunto de teste:** usado exclusivamente para avaliar a performance de $f$

**Ideia:** o teste simula dados "nunca vistos", imitando o comportamento de $f$ em novos exemplos.

# Fim!

Obrigado!

# Começando a viagem (de verdade)



Tarantino e George Clooney fugindo da polícia e lutando contra vampiros num bar

# Alta dimensão

**Em aprendizado de máquinas, é comum lidarmos com dados em *alta dimensão*:**

$$X \in \mathbb{R}^p \quad \text{com } p \gg 1$$

- Isso acontece em muitas aplicações: imagens, textos, genômica, sensores, etc.
- Em alta dimensão, nossa intuição geométrica pode falhar completamente!
- Precisamos de novas ferramentas matemáticas: **probabilidade e geometria em alta dimensão**.

**Pergunta para vocês:**

*O que acontece com o volume de uma bola unitária conforme a dimensão cresce?*

# Alta dimensão

**Seja $B_p = \{x \in \mathbb{R}^p : \|x\|_2 \leq 1\}$ a bola unitária em $\mathbb{R}^p$**

**Volume da bola unitária:**

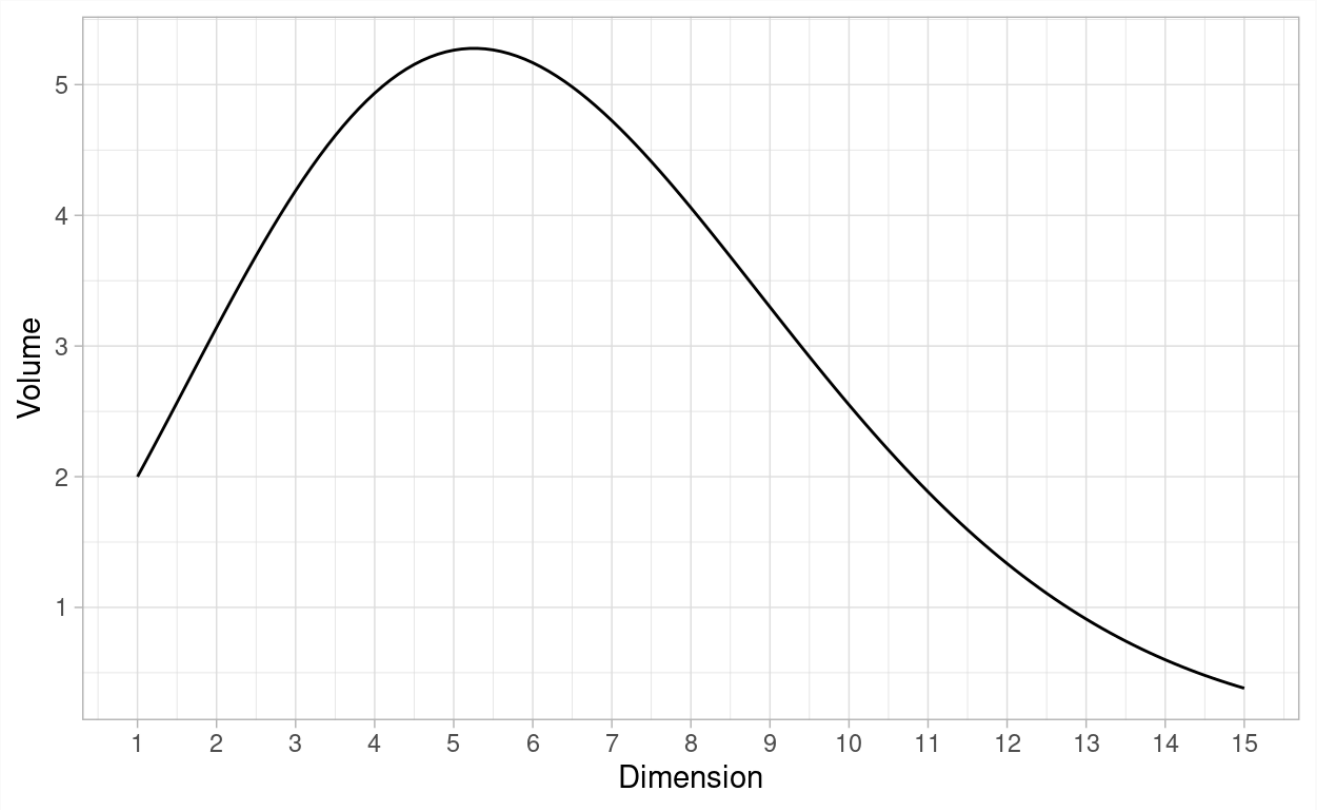
$$V_p = \frac{\pi^{p/2}}{\Gamma\left(\frac{p}{2} + 1\right)}$$

**Alguns valores:**

- $p = 1$: $V_1 = 2$ (um intervalo de comprimento 2)
- $p = 2$: $V_2 = \pi \approx 3.14$ (área do círculo unitário)
- $p = 3$: $V_3 = \frac{4}{3}\pi \approx 4.19$ (volume da esfera)

**E depois disso?**

# Alta dimensão




- O volume cresce até certo ponto e depois **cai rapidamente para zero**
- Em altas dimensões, quase todo o volume da bola se concentra próximo da borda!

# Alta dimensão

**Considere dois vetores $u, v \in \mathbb{R}^p$ com entradas geradas aleatoriamente.**

**Pergunta:** Qual o ângulo entre esses vetores?

- Em dimensões baixas (como $p = 2$ ou $3$), os vetores podem formar qualquer ângulo
- Mas em alta dimensão, algo curioso acontece...

**Fato:** se $p \to \infty$, o ângulo entre vetores aleatórios se concentra em $\frac{\pi}{2}$

$$\text{Se } u, v \sim \text{uniforme na esfera}, \quad \cos\theta = \frac{\langle u, v \rangle}{\|u\|\|v\|} \approx 0$$

**Ou seja:** vetores aleatórios em alta dimensão são **quase ortogonais**!

# Como formalizar isso?

**Vimos fenômenos geométricos surpreendentes:**

- Volume se concentra na borda da bola
- Vetores aleatórios são quase ortogonais

**Mas isso tem tudo a ver com aprendizado de máquinas:**

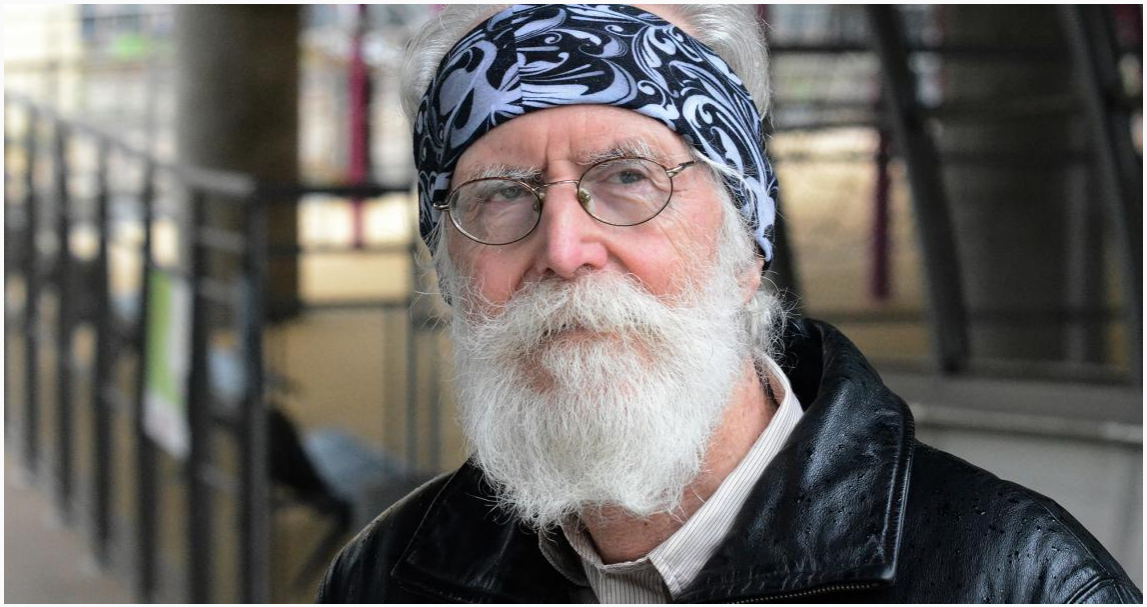
- Nosso objetivo é minimizar o **erro esperado** $L(f)$
- Mas só temos acesso ao **erro empírico** $\hat{L}_n(f)$
- A pergunta central é: **quão perto está** $\hat{L}_n(f)$ **de** $L(f)$**?**

A resposta é dada pela teoria de Concentração de medida!

*Ela garante que, com alta probabilidade, o erro no treino se* ***concentra*** *perto erro verdadeiro.*

# Michel Talagrand

**Michel Pierre Talagrand** é um dos principais nomes no estudo de **concentração de medida** e suas aplicações em probabilidade, análise funcional e física matemática. Em **2024**, recebeu o **Prêmio Abel**.



# Desigualdade de Hoeffding

**Cenário:** queremos entender quão perto a média empírica está da esperança verdadeira.

**Seja $Z_1, \ldots, Z_n$ variáveis aleatórias i.i.d. com valores em $[a, b]$ e:**

$$\bar{Z}_n = \frac{1}{n} \sum_{i=1}^{n} Z_i$$

**A desigualdade de Hoeffding garante que:**

$$\mathbb{P}\left(|\bar{Z}_n - \mathbb{E}[Z]| \geq \varepsilon\right) \leq 2 \exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

**Perceba que essa garantia depende de n, ou seja, não é assintótica.**

# Desigualdade de Hoeffding

**Aplicação direta em aprendizado:**

- Fixe uma função $f \in \mathcal{H}$
- Defina $Z_i = \ell(f(X_i), Y_i)$, ou seja, a perda em cada exemplo
- Então:

$$\hat{L}_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i) = \bar{Z}_n \quad \text{e} \quad L(f) = \mathbb{E}[\ell(f(X), Y)] = \mathbb{E}[Z]$$

- Logo, Hoeffding nos dá uma cota sobre:

$$\mathbb{P}\left(|\hat{L}_n(f) - L(f)| \geq \varepsilon\right)$$

*O erro empírico é uma boa aproximação do erro verdadeiro, com alta probabilidade!*

# E se temos várias hipóteses?

**Até agora:** aplicamos Hoeffding para uma única função $f \in \mathcal{H}$ fixa.

**Mas na prática, não escolhemos $f$ de antemão — nós escolhemos $f$ com base nos dados!**

**Suponha agora que temos um conjunto finito de hipóteses:**

$$\mathcal{H} = \{f_1, f_2, \ldots, f_M\}$$

**Queremos garantir que, para *todas* as $f \in \mathcal{H}$, o erro empírico está próximo do erro esperado:**

$$\sup_{f \in \mathcal{H}} \left| \hat{L}_n(f) - L(f) \right| \leq \varepsilon$$

# E se temos várias hipóteses?

**Solução:** aplicar Hoeffding + união de eventos

$$\mathbb{P}\left(\exists f \in \mathcal{H} \text{ tal que } |\hat{L}_n(f) - L(f)| \geq \varepsilon\right) \leq 2M \exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

*Quanto maior o número de hipóteses $M$, mais difícil garantir generalização!*

# E quando o número de hipóteses é infinito?

**Problema:** na prática, o conjunto de hipóteses $\mathcal{H}$ é geralmente **infinito**!

- Ex: todos os classificadores lineares no plano
- Ex: árvores de decisão de profundidade arbitrária
- Ex: redes neurais com pesos reais

**Nesse caso, o número $M$ é infinito e a cota anterior não nos diz nada.**

## Como então controlar a complexidade de $\mathcal{H}$?

*Precisamos de uma medida mais sutil de complexidade: a **dimensão VC**.*

# O que é a dimensão VC?

**A dimensão VC (Vapnik–Chervonenkis)** é uma ferramenta combinatória para medir a complexidade de um conjunto de hipóteses $\mathcal{H}$.

- Em vez de contar quantas funções existem, ela mede **o poder de separação de** $\mathcal{H}$.
- Intuitivamente: *quantos padrões distintos de classificação um modelo consegue realizar?*
- Permite obter cotas de generalização mesmo quando $\mathcal{H}$ é infinito!

**Alguns exemplos clássicos:**

- Classificadores lineares em $\mathbb{R}^2$ (retas): VC-dimensão = 3
- Classificadores lineares em $\mathbb{R}^p$: VC-dimensão = $p + 1$
- Árvores de decisão de profundidade $d$: VC-dimensão $\leq 2^d$
- Conjuntos de todas as funções de $\mathbb{R}$ para $\{0, 1\}$: VC-dimensão = infinito

**Conclusão:** a dimensão VC nos ajuda a entender quão "flexível" (ou perigosa!) é uma classe de modelos.

# Teorema de generalização via dimensão VC

**Teorema:** Seja $\mathcal{H}$ uma classe de hipóteses com dimensão VC $d$. Então, para toda distribuição $\mathcal{D}$ e todo $\delta \in (0,1)$, com probabilidade pelo menos $1-\delta$ sobre a escolha da amostra $S \sim \mathcal{D}^n$, temos:

$$\left|\hat{L}_n(f) - L(f)\right| \leq \frac{4 + \sqrt{d \log\left(\frac{2en}{d}\right)}}{\sqrt{2n\delta}}$$

**Onde:**

- $\hat{L}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), Y_i)$ é o erro empírico
- $L(f) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\ell(f(X), Y)]$ é o erro esperado
- $n$ é o número de amostras
- $d$ é a dimensão VC de $\mathcal{H}$
- $\delta$ controla a confiança da desigualdade

*Mesmo para $\mathcal{H}$ infinita, podemos controlar a generalização se $d$ for pequeno.*

# Provando a generalização de modelos

**A teoria de generalização baseada em dimensão VC é aplicável a vários modelos clássicos:**

- **Classificadores lineares em $\mathbb{R}^p$:**

$$\mathcal{H} = \{x \mapsto \text{sign}(\langle w, x \rangle + b)\} \quad \Rightarrow \quad \text{VC-dim} = p + 1$$

- **Polinômios de grau $d$ em $\mathbb{R}$:**

$$\mathcal{H} = \{x \mapsto \text{sign}(a_0 + a_1 x + \cdots + a_d x^d)\} \quad \Rightarrow \quad \text{VC-dim} = d + 1$$

- **Árvores de decisão de profundidade $d$:**

$$\text{VC-dim} \leq 2^d$$

- **Redes neurais com funções de ativação lineares por partes (ReLU):**

VC-dim depende do número de parâmetros (pesos e biases)

**Esses resultados ajudam a entender a capacidade dos modelos e como ela afeta a generalização.**

# Conclusão: Muito além de treinar e testar

- Aprendizado de máquinas não é só código, bibliotecas e "testar modelos"
- Por trás de algoritmos modernos existe uma teoria profunda, elegante e surpreendentemente útil
- Conceitos como concentração de medida, dimensão VC, complexidade de hipótese e generalização formam a base matemática do campo

**Muita gente acha que "não existe teoria" em aprendizado de máquina...**

*...assim como muita gente acha que não existem seres das trevas em bares ancestrais.*

# Estão errados nos dois casos!

# Fim!

Tarantino e George Clooney fugindo da polícia e lutando contra vampiros num bar que na verdade era um templo profano