

# Machine Learning

## Aula 1

Prof. Thiago A. N. De Andrade

Universidade Federal de Santa Maria  
Departamento de Estatística

2025-08-16

# Aviso aos estudantes

- Este é um material novo e atualizado, elaborado especialmente para nosso curso **Machine Learning - UFSM 2025.2**. Entretanto, **não se configura em conteúdo original**. É apenas uma compilação resumida de conteúdos presentes nas referências citadas. Em resumo: é indispensável consultar as referências indicadas.
- As imagens não são autorais e os respectivos créditos são reservados aos autores.
- Este material foi integralmente produzido em R Markdown, utilizando o pacote xaringan, que possibilita a criação de apresentações **ninja**.

# Introdução ao Machine Learning

- O que é Machine Learning ?
- Tipos de aprendizado
- Exemplo de problema comum para estatísticos
- As duas culturas

O que é *Machine Learning* ?



Alan Turing (1912-1954)

- Contribuições:  
Matemática,  
Computação, Criptografia  
e Filosofia.
- Reconhecido como um  
dos **pais da ciência da  
computação teórica** e da  
**inteligência artificial**.
- Em *Computing Machinery  
and Intelligence* (1950): "*I  
PROPOSE to consider the  
question, 'Can machines  
think?'*"

# Jogo de Imitação

O jogo original, descrito por Turing em 1950, envolve três participantes:

- **Jogador A:** um homem, cujo objetivo é levar o interrogador a tomar a decisão errada.
- **Jogador B:** uma mulher, que tenta ajudar o interrogador a acertar.
- **Jogador C:** o interrogador, que não vê os outros participantes e se comunica apenas por mensagens escritas.
- Fazendo perguntas a A e B, o interrogador tenta determinar qual deles é o homem e qual é a mulher.

- Turing propôs substituir o participante A por uma **máquina**.
- O objetivo passa a ser avaliar se o interrogador consegue distinguir a máquina de um humano, reformulando a questão *"As máquinas podem pensar?"* para *"Uma máquina pode imitar um humano a ponto de enganar o interrogador?"*.

-- We now ask the question, 'What will happen when a machine takes the part of A in this game?' Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, 'Can machines think?'

VOL. LIX. No. 236.]

[October, 1950

M I N D  
A QUARTERLY REVIEW  
OF  
PSYCHOLOGY AND PHILOSOPHY



I.—COMPUTING MACHINERY AND  
INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can

Downloaded from <https://academic.oup.com/mind/article/LIX/236/433/986238> by gi



Arthur Lee Samuel (1901-1990)

- Cientista da computação estadunidense, pioneiro nos campos dos jogos de computador, inteligência artificial e aprendizado de máquina.
- Desenvolveu o primeiro programa de auto-aprendizagem do mundo.
- Membro sênior da comunidade *TeX*. Escreveu um manual *TeX* em 1983.

-- "Campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados (Frase atribuída a Arthur Lee Samuel.)"

Nos papers:

-- "Programar computadores para aprender com a experiência deve eventualmente eliminar a necessidade de grande parte desse esforço de programação detalhada. (Arthur Samuel, 1959)"

-- "Um computador pode ser programado de modo que aprenda a jogar um jogo de damas melhor do que aquele que pode ser jogado pela pessoa que escreveu o programa. (Arthur Samuel, 1959)"

A. L. Samuel

## Some Studies in Machine Learning Using the Game of Checkers

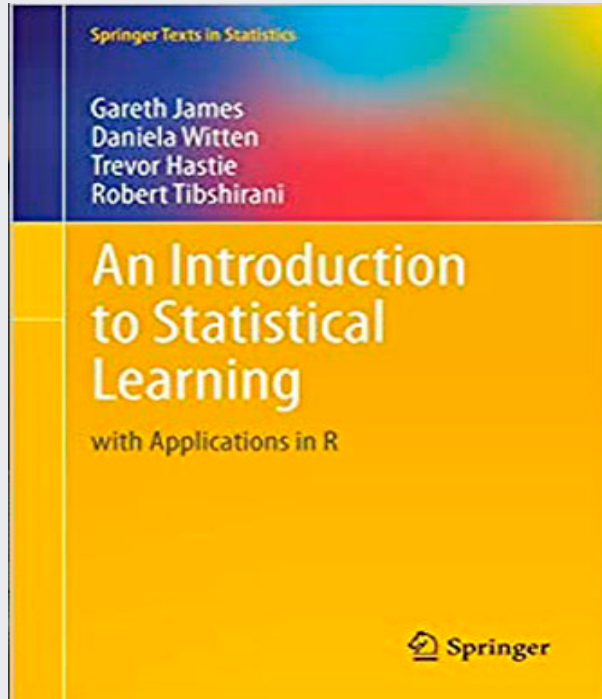
**Abstract:** Two machine-learning procedures have been investigated in some detail using the game of checkers. Enough work has been done to verify the fact that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this in a remarkably short period of time (8 or 10 hours of machine-playing time) when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of machine learning verified by these experiments are, of course, applicable to many other situations.

### Introduction

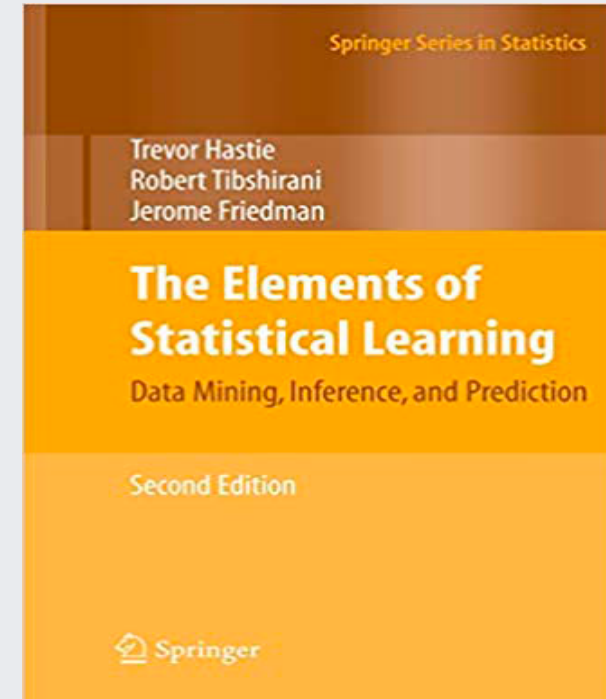
The studies reported here have been concerned with the programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning. While this is not the place to dwell on the importance of machine-learning procedures, or to discourse on the philosophical aspects,<sup>1</sup> there is obviously a very large amount of work, now done by people, which is quite trivial in its demands on the intellect but does, nevertheless, involve some learning. We have at our command computers with

method should lead to the development of general-purpose learning machines. A comparison between the size of the switching nets that can be reasonably constructed or simulated at the present time and the size of the neural nets used by animals, suggests that we have a long way to go before we obtain practical devices.<sup>2</sup> The second procedure requires reprogramming for each new application, but it is capable of realization at the present time. The experiments to be described here were based on this second approach.

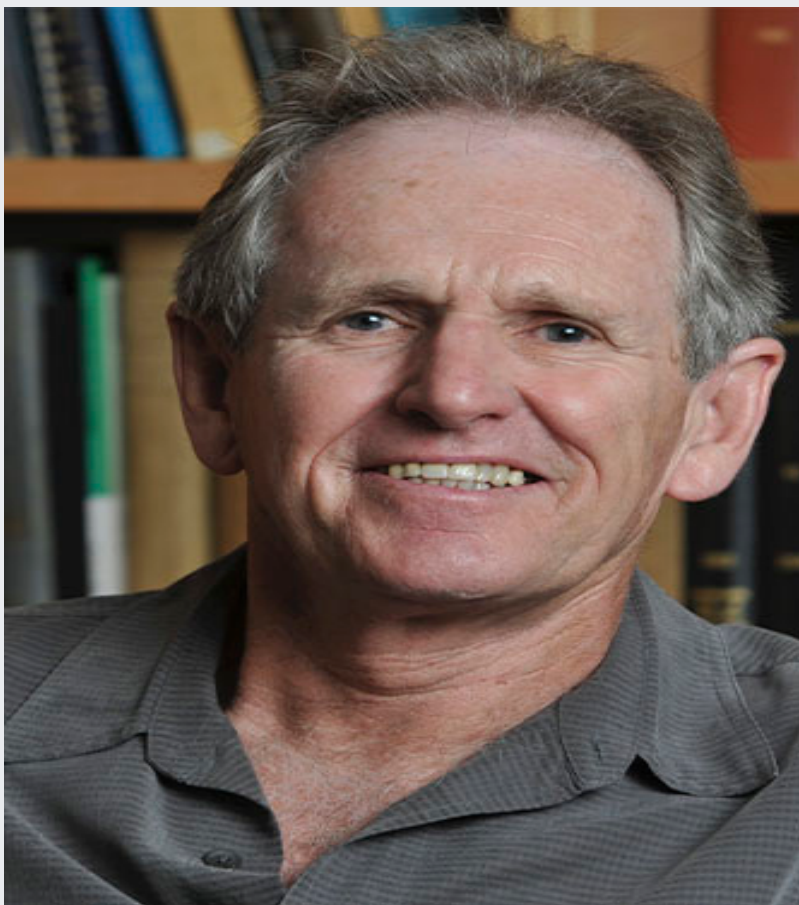
# Popularização do Aprendizado de Máquina na Estatística



**An Introduction to Statistical Learning**



**The Elements of Statistical Learning**

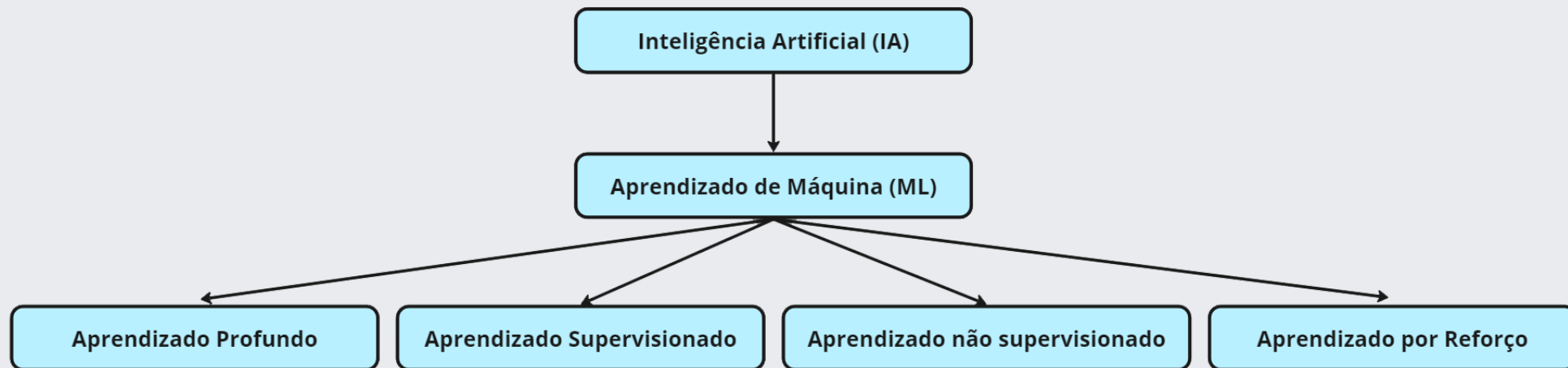


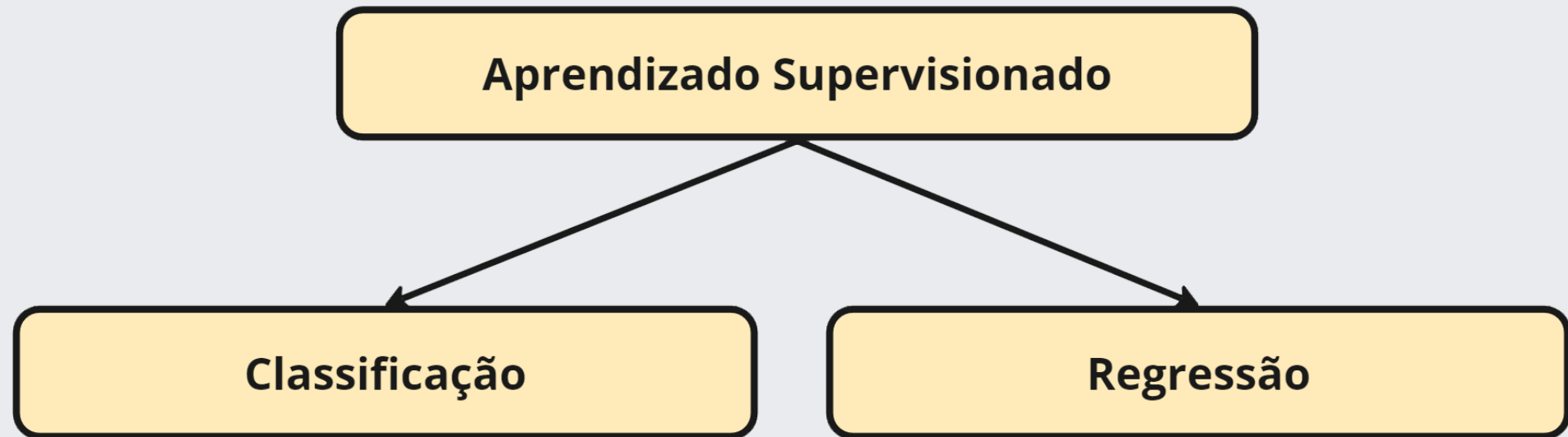
Trevor Hastie

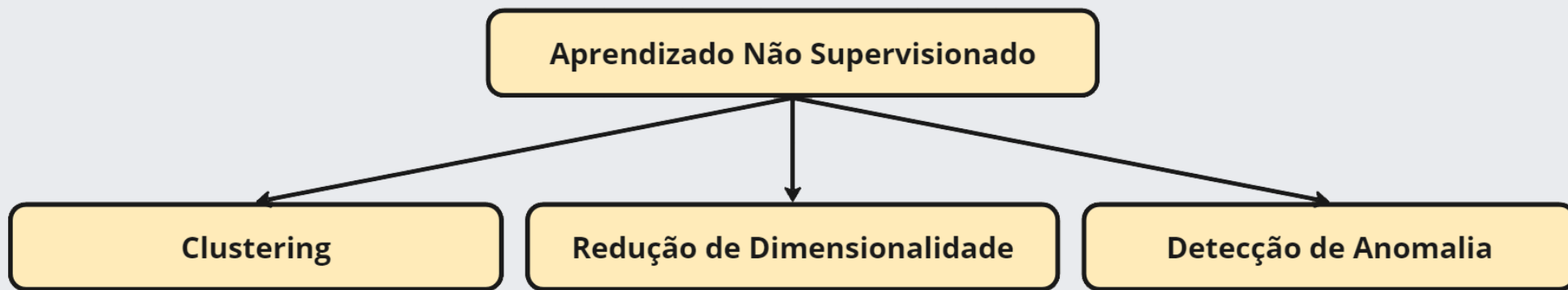


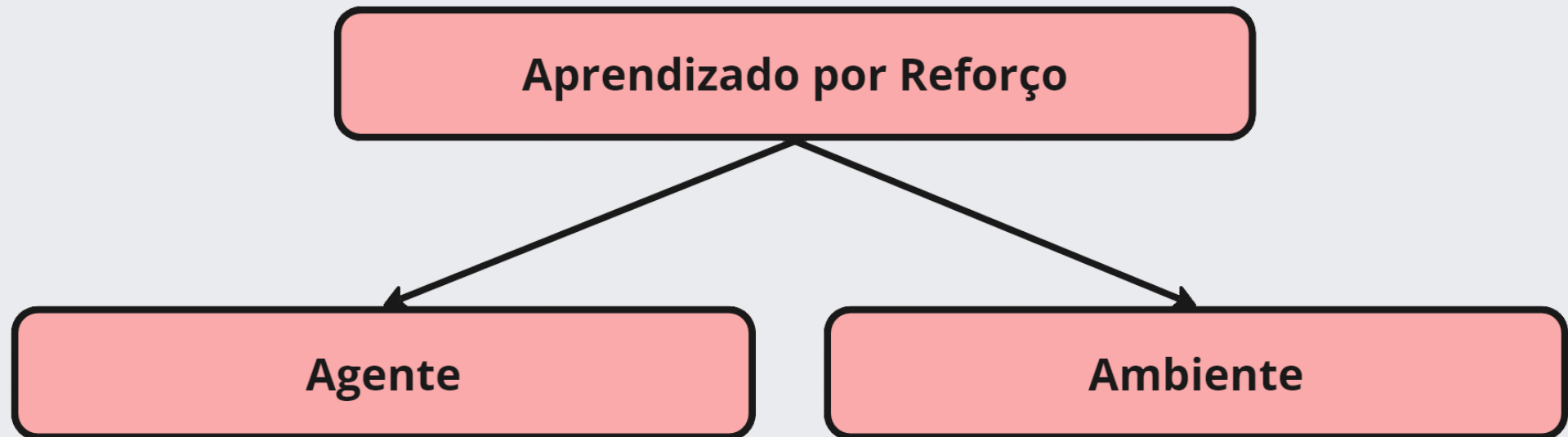
Robert Tibshirani

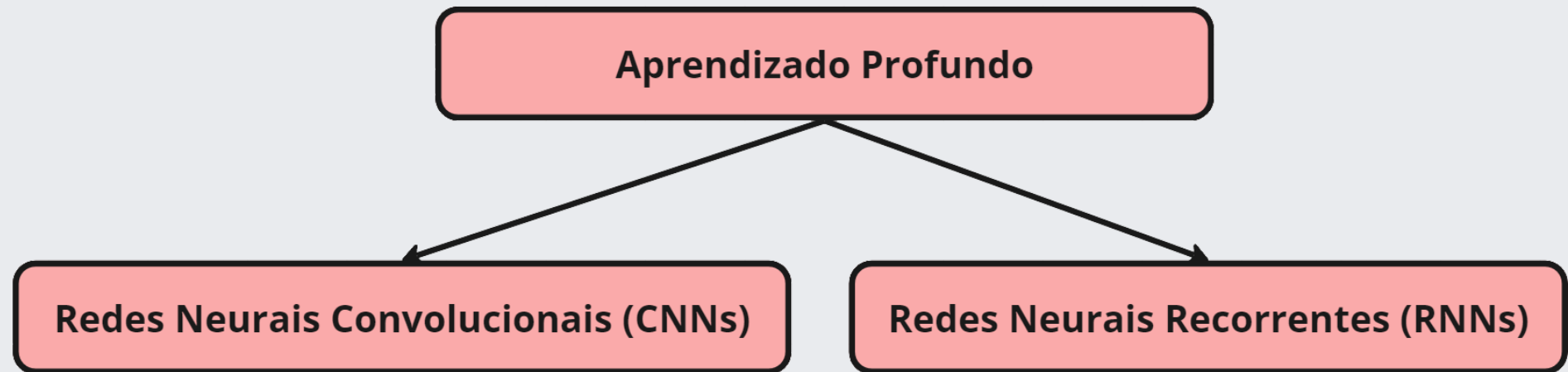
# Tipos de aprendizaje

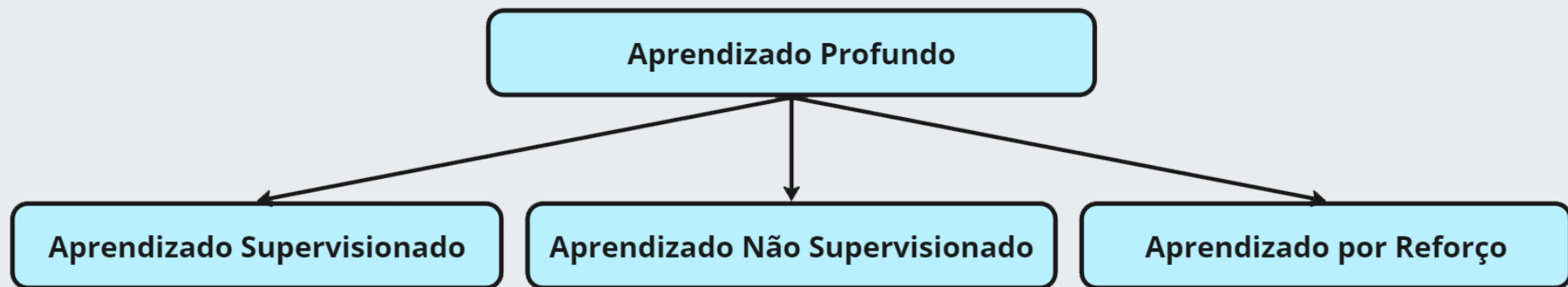












# Aprendizado Supervisionado

# Características

- Utiliza um conjunto de **dados rotulados** para treinar o modelo.
- O objetivo é **prever** ou **classificar** novas observações com base nos padrões aprendidos durante o treinamento.

## Tipos de Problemas

- **Classificação**: previsão de rótulos **categóricos** (e.g., *spam* ou *não-spam*).
- **Regressão**: previsão de valores **contínuos** (e.g., preço estimado de imóveis).

# Dados de Treinamento

- Compostos por pares **entrada-saída**.
- Exigem quantidade suficiente de dados rotulados para que o modelo generalize bem.

# Modelos Comuns

- Regressão Linear e Regressão Logística
- Árvores de Decisão e Florestas Aleatórias
- Redes Neurais Artificiais

# Avaliação

- **Classificação:** acurácia, precisão (*precision*), revocação (*recall*), F1-score.
- **Regressão:** erro quadrático médio (MSE), erro absoluto médio (MAE), coeficiente de determinação ( $R^2$ ).

# Aplicações

- Diagnóstico médico (e.g., classificação de doenças)
- Reconhecimento de voz
- Análise de crédito (e.g., previsão de inadimplência)

# Aprendizado Não Supervisionado

# Características

- Utiliza **dados não rotulados** para identificar padrões e relações.
- O objetivo é explorar a **estrutura subjacente** dos dados, sem rótulos predefinidos.

# Tipos de Problemas

- **Agrupamento (Clusterização)**
- **Redução de Dimensionalidade**
- **Detecção de Anomalias**

# Dados de Treinamento

- Compostos apenas por **variáveis de entrada** (sem rótulos de saída).
- Podem ser aplicados a dados brutos, sem necessidade de pré-classificação.

# Modelos Comuns

## Agrupamento (Clusterização)

- **Métodos Hierárquicos:**
  - Clusterização hierárquica aglomerativa (HAC)
  - Clusterização hierárquica divisiva (DHC)
- **Métodos Baseados em Centróides:**
  - *k-means*
  - *k-medoids*

- **Métodos Baseados em Distribuições:**
  - Modelos de Mistura Gaussiana (GMM)
  - Latent Dirichlet Allocation (LDA)

## Redução de Dimensionalidade

- **PCA (Análise de Componentes Principais):** técnica linear que projeta dados em componentes principais.
- **t-SNE (t-Distributed Stochastic Neighbor Embedding):** técnica não linear para preservar a estrutura local dos dados em espaços de menor dimensão.

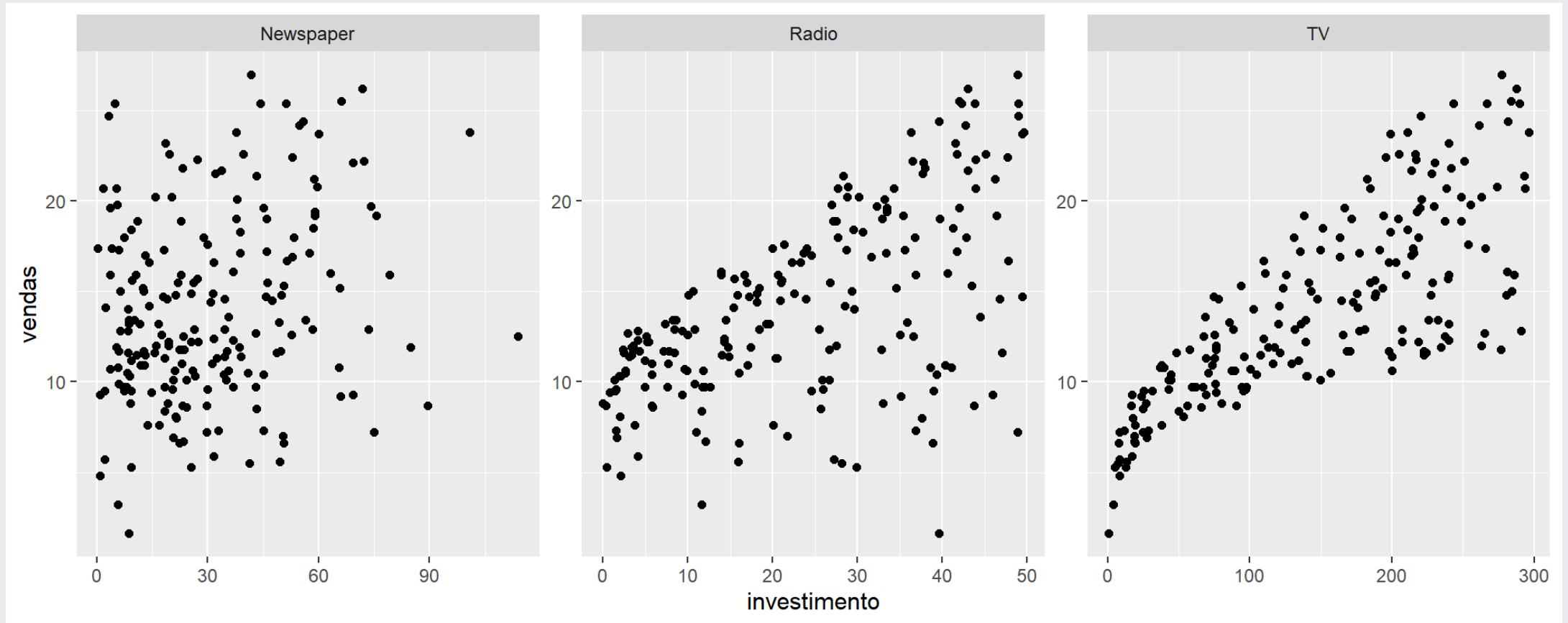
## Detecção de Anomalias

- **Autoencoders:** redes neurais que aprendem representações compactas e detectam anomalias pelo erro de reconstrução.
- **Modelos Probabilísticos:** como Modelos de Mistura Gaussiana (GMM), que modelam a distribuição dos dados para identificar outliers.

# Exemplo de problema comum para estatísticos

-- "(...) O processo de desenvolvimento desses tipos de ferramentas evoluiu em vários campos, como química, ciência da computação, física e estatística, e foi chamado de “aprendizagem de máquina”, “inteligência artificial”, “reconhecimento de padrões”, “mineração de dados”, “análise preditiva” e “descoberta de conhecimento”. Enquanto cada campo aborda o problema usando diferentes perspectivas e conjuntos de ferramentas, o objetivo final é o mesmo: fazer uma previsão precisa. Para este livro, agruparemos esses termos na chamada *modelagem preditiva*." (Kuhn & Johnson, 2013)

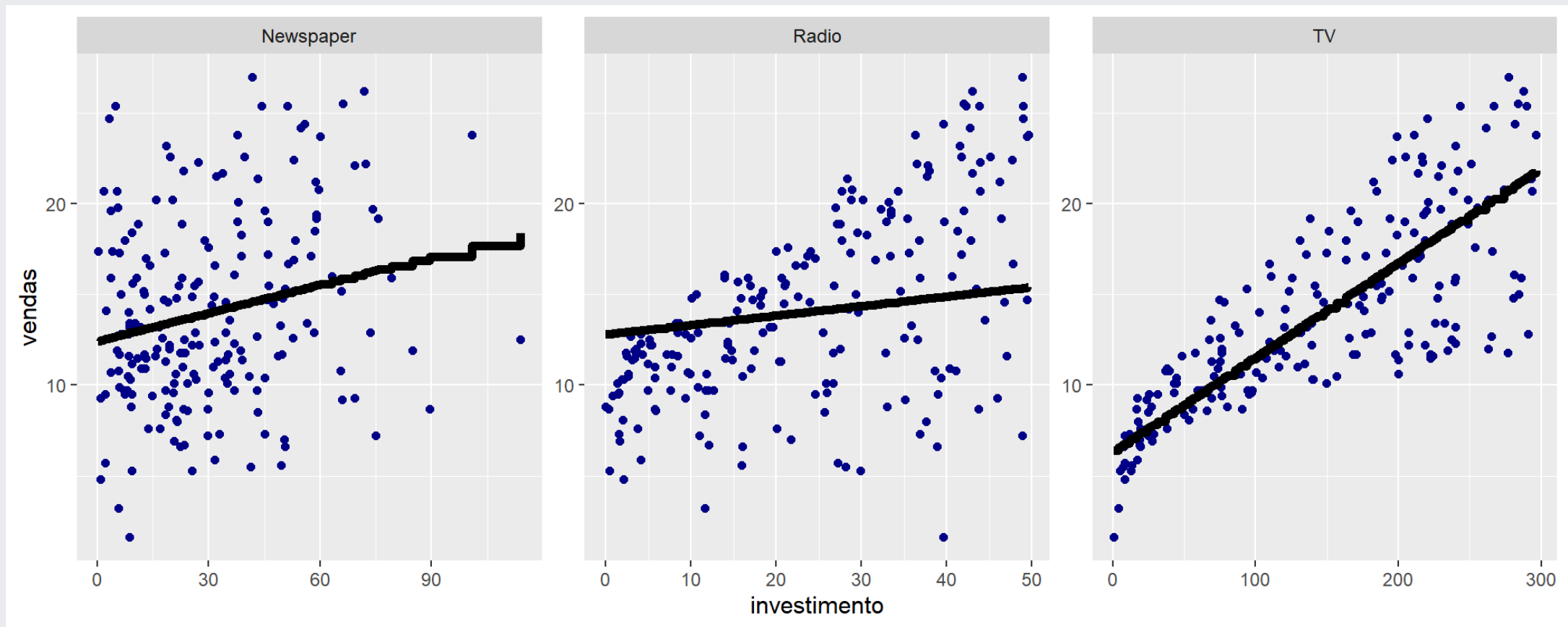
-- "Modelagem preditiva é um framework de análise de dados que visa gerar a estimativa mais precisa possível para uma quantidade ou fenômeno." (Max Kuhn, 2014)



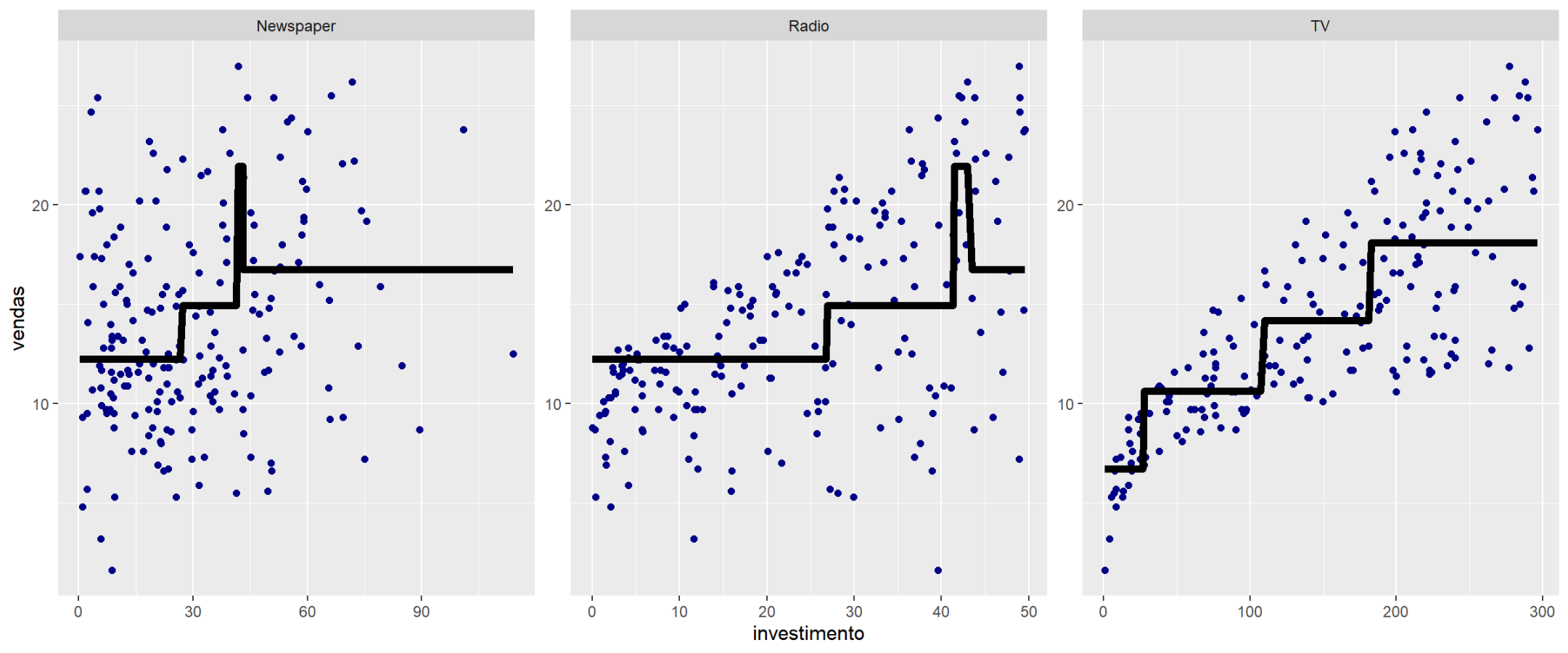
Em ML, queremos encontrar uma função  $f()$  tal que:

$$y \approx f(x)$$

## Exemplo de $f(x)$ : Regressão



# Exemplo de $f(x)$ : Árvore de Decisão



# As duas culturas

# Por que ajustar uma $f(\cdot)$ ?

- **Modelo gerador:**  $y = f(\mathbf{x}) + \varepsilon$ .
- Em essência, a aprendizagem estatística reúne abordagens para **estimar**  $f$ .
- Pergunta-chave: qual o objetivo ao estimarmos  $f$ ? **Predição** ou **Inferência**.

- **Inferência:** deseja-se conhecer a associação entre  $x_1, x_2, \dots, x_p$  e  $y$ .
  - A forma funcional importa; o modelo deve ser **interpretável** para extrair efeitos/associações.
- **Predição:** dados  $x_1, x_2, \dots, x_p$ , estimamos  $\hat{y} = \hat{f}(\mathbf{x})$ .
  - Não é necessário conhecer a forma explícita de  $\hat{f}$ . **Podemos tratar  $\hat{f}$  como caixa-preta, desde que minimize o erro de teste .**

- Dependendo do objetivo (predição, inferência ou ambos), diferentes métodos para estimar  $f$  podem ser apropriados.
- Em geral, há um **trade-off** entre **poder preditivo** e **interpretabilidade**.
- Ex.: modelos lineares facilitam inferências, mas podem perder em acurácia frente a modelos mais flexíveis.
- Ex.: modelos muito flexíveis podem prever bem, mas têm interpretação difícil.

# Como avaliar a $\hat{f}$ obtida?

- Foque em **erro de teste** (desempenho fora da amostra).
- Use **validação cruzada k-fold** (ex.:  $k = 5$  ou  $k = 10$ ).
- Evite avaliar no treino (risco de **overfitting**).

<b>Mídia</b>	<b>Investimento</b>	<b>Vendas</b>	<b>Árvore</b>	<b>Regressão Linear</b>
Newspaper	9.3	11.2	12.2	12.9
Radio	12.0	9.7	12.2	13.4
Radio	5.2	12.2	12.2	13.1
Newspaper	9.3	18.4	12.2	12.9
Newspaper	35.1	10.1	14.9	14.3
Newspaper	71.8	26.2	16.8	16.2
Newspaper	45.7	14.7	16.8	14.8
Newspaper	49.6	5.6	16.8	15.0
Newspaper	56.5	13.4	16.8	15.4
Radio	5.7	10.4	12.2	13.1

# Poder predictivo *versus* Interpretabilidade

Interpretação

Regressão linear

Árvore de decisão

Modelos aditivos generalizados

Redes neurais

Deep learning

Baixo

Alto

Predição

# Referências

- **Some Studies in Machine Learning Using the Game of Checkers**
- **Computing Machinery and Intelligence**
- **Machine Learning: an introduction**
- **An Introduction to Statistical Learning**
- **Aprendizado de máquina: uma abordagem estatística**
- **Materiais Curso R**

Não deixe de entrar em contato comigo para tirar suas dúvidas:  
thiagoan.andrade@gmail.com

Estamos no  @thiagoan.andrade para networking e socializações

# Obrigado!

# Thanks!