

INTELIGÊNCIA ARTIFICIAL

APRENDIZADO DE MÁQUINA - INTRODUÇÃO

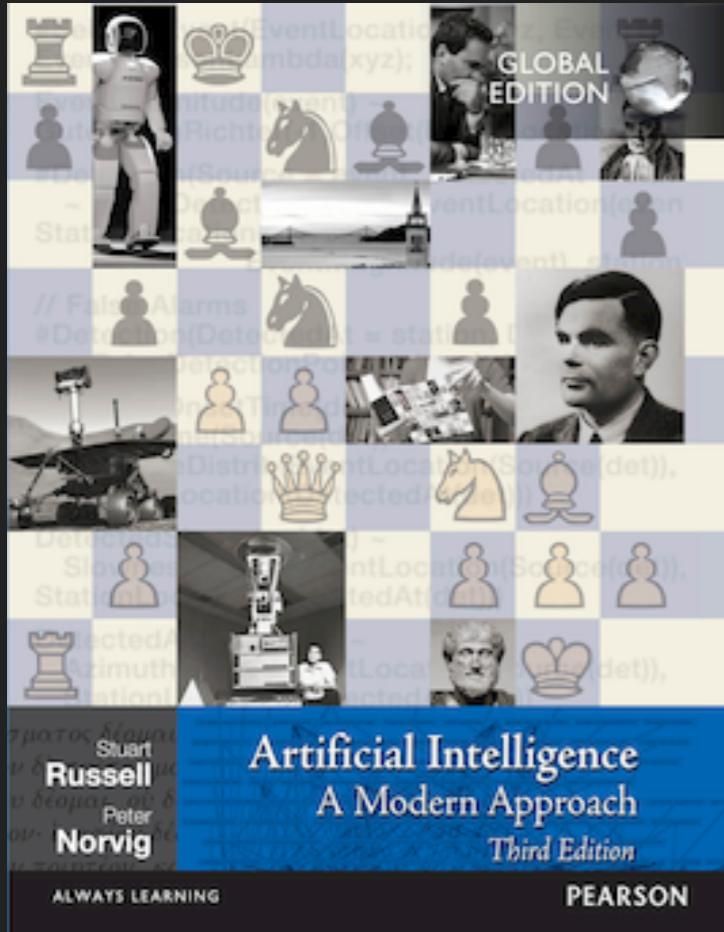
Lucas Baggio Figueira [@lucasfigueira]

MATERIAL DE ESTUDO

Artificial Intelligence: A modern approach

Cap. 18

MATERIAL DE ESTUDO



Artificial Intelligence: A modern approach

Cap. 18

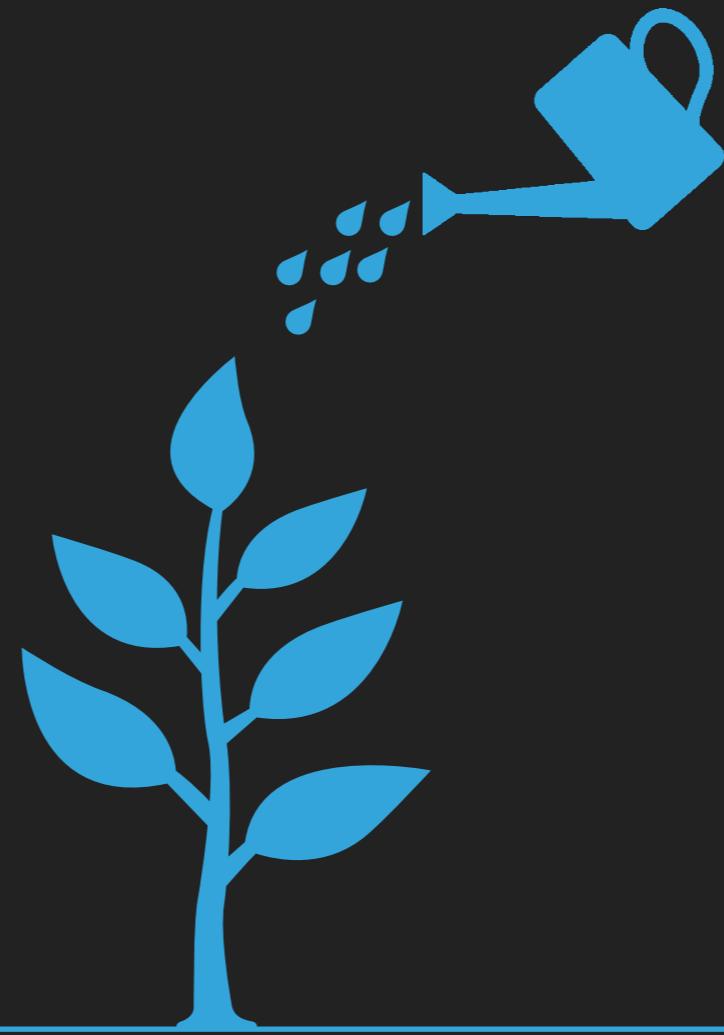
SUMÁRIO

- ▶ Agentes que aprendem
- ▶ Aprendizado Indutivo
- ▶ Aprendizado de Árvores de Decisão
- ▶ Medindo o desempenho do aprendizado

**QUALQUER AGENTE PODE SER
MELHORADO APRENDENDO ATRAVÉS
DOS DADOS**



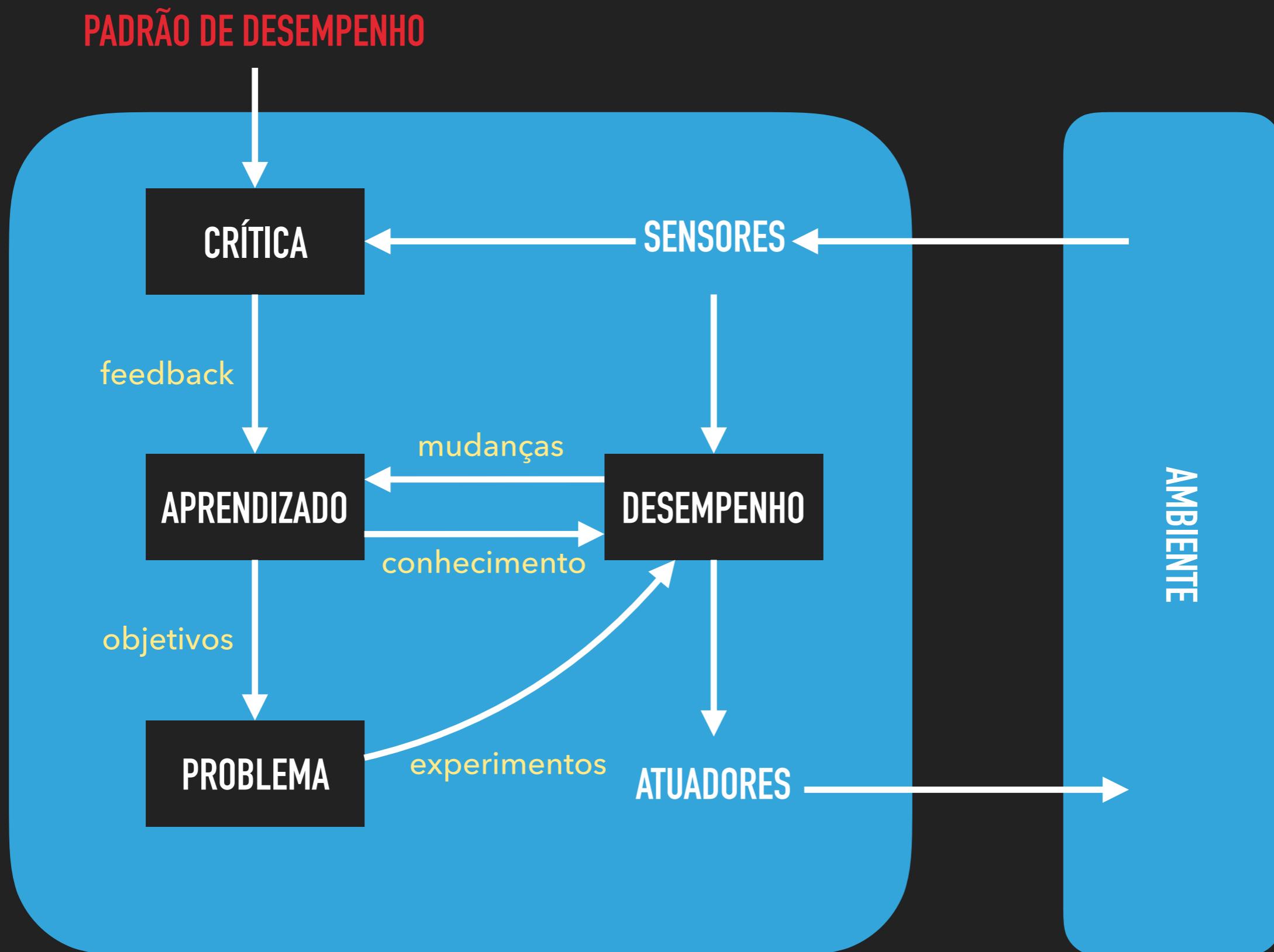
**QUALQUER AGENTE PODE SER
MELHORADO APRENDENDO ATRAVÉS
DOS DADOS**



AGENTE INTELIGENTE QUE APRENDE

Lucas Baggio Figueira [*@lucasfigueira*]

AGENTE INTELIGENTE QUE APRENDE



PROJETAR O APRENDIZADO DEPENDE DE 4 FATORES:

PROJETAR O APRENDIZADO DEPENDE DE 4 FATORES:

- ▶ Qual **componente** será aprendido?

PROJETAR O APRENDIZADO DEPENDE DE 4 FATORES:

- ▶ Qual **componente** será aprendido?
- ▶ Qual o **conhecimento prévio** existente?

PROJETAR O APRENDIZADO DEPENDE DE 4 FATORES:

- ▶ Qual **componente** será aprendido?
- ▶ Qual o **conhecimento prévio** existente?
- ▶ Qual **representação** é usada pelo componente e pelos dados?

PROJETAR O APRENDIZADO DEPENDE DE 4 FATORES:

- ▶ Qual **componente** será aprendido?
- ▶ Qual o **conhecimento prévio** existente?
- ▶ Qual **representação** é usada pelo componente e pelos dados?
- ▶ Qual o **feedback** está disponível para o aprendizado?

COMPONENTE A SER APRENDIDO

AVALIAR QUAL O MELHOR
ESTADO DISPONÍVEL PARA
BUSCA

COMPONENTE A SER
APRENDIDO

AVALIAR QUAL O MELHOR
ESTADO DISPONÍVEL PARA
BUSCA

INFERIR INFORMAÇÃO
RELEVANTE A PARTIR DA
SEQUENCIA DE PERCEPÇÃO

COMPONENTE A SER
APRENDIDO

AVALIAR QUAL O MELHOR
ESTADO DISPONÍVEL PARA
BUSCA

INFERIR INFORMAÇÃO
RELEVANTE A PARTIR DA
SEQUENCIA DE PERCEPÇÃO

COMPONENTE A SER APRENDIDO

INFORMAÇÃO SOBRE COMO O
MUNDO EVOLUI A PARTIR DAS
AÇÕES

AVALIAR QUAL O MELHOR
ESTADO DISPONÍVEL PARA
BUSCA

INFERIR INFORMAÇÃO
RELEVANTE A PARTIR DA
SEQUENCIA DE PERCEPÇÃO

COMPONENTE A SER APRENDIDO

INFORMAÇÃO SOBRE COMO O
MUNDO EVOLUI A PARTIR DAS
AÇÕES

IDENTIFICAR SITUAÇÕES QUE
MAXIMIZAM O RETORNO AO
AGENTE

REPRESENTAÇÃO E CONHECIMENTO PRÉVIO

ÁRVORES DE BUSCA

REPRESENTAÇÃO E
CONHECIMENTO PRÉVIO

ÁRVORES DE BUSCA

REPRESENTAÇÃO DO ESTADO/NÓ

REPRESENTAÇÃO E
CONHECIMENTO PRÉVIO

REPRESENTAÇÃO DO ESTADO/NÓ

ÁRVORES DE BUSCA

REPRESENTAÇÃO E
CONHECIMENTO PRÉVIO

APRENDIZADO POR INDUÇÃO

REPRESENTAÇÃO DO ESTADO/NÓ

ÁRVORES DE BUSCA

REPRESENTAÇÃO E
CONHECIMENTO PRÉVIO

APRENDIZADO POR INDUÇÃO

APRENDIZADO POR DEDUÇÃO

FEEDBACK E MÉTODOS DE APRENDIZADO

Lucas Baggio Figueira [*@lucasfigueira*]

APRENDIZADO SUPERVISIONADO



FEEDBACK E MÉTODOS DE
APRENDIZADO

APRENDIZADO SUPERVISIONADO

APRENDIZADO NÃO-SUPERVISIONADO

FEEDBACK E MÉTODOS DE APRENDIZADO

APRENDIZADO NÃO-SUPERVISIONADO

APRENDIZADO SUPERVISIONADO

FEEDBACK E MÉTODOS DE APRENDIZADO

APRENDIZADO POR REFORÇO

APRENDIZADO NÃO-SUPERVISIONADO

APRENDIZADO SUPERVISIONADO

FEEDBACK E MÉTODOS DE APRENDIZADO

APRENDIZADO SEMI-SUPERVISIONADO

APRENDIZADO POR REFORÇO

APRENDIZADO NÃO-SUPERVISIONADO

APRENDIZADO SUPERVISIONADO

FEEDBACK E MÉTODOS DE APRENDIZADO

APRENDIZADO POR REFORÇO

APRENDIZADO SEMI-SUPERVISIONADO

“Um programa de computador aprende a partir da experiência E com relação à uma tarefa T e desempenho P, se seu desempenho na tarefa T, medido por P, melhora a partir da experiência E”

Tom Mitchel (1997)

T, tarefa

E, experiência

P, desempenho (performance)

APRENDIZADO SUPERVISIONADO

APRENDIZADO SUPERVISIONADO



APRENDIZADO SUPERVISIONADO

Seja um **conjunto de treinamento** contendo N pares entrada-saída

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

onde cada y_i foi gerado por uma função desconhecida $y = f(x)$

f

Descobrir a função h que aproxima a função verdadeira

HIPÓTESE

**APRENDER É PROCURAR NO
ESPAÇO DE HIPÓTESES AQUELA QUE MELHOR GENERALIZA
A FUNÇÃO f .**

**A HIPÓTESE h GENERALIZA BEM SE ELA MAPEIA
CORRETAMENTE y PARA NOVOS VALORES DE x .**

A NATUREZA DE γ DETERMINA O SE O PROBLEMA É:

A NATUREZA DE γ DETERMINA O SE O PROBLEMA É:

CLASSIFICAÇÃO

QUANDO γ DEFINE CATEGORIAS

A NATUREZA DE \hat{y} DETERMINA O SE O PROBLEMA É:

CLASSIFICAÇÃO

QUANDO \hat{y} DEFINE CATEGORIAS

REGRESSÃO

QUANDO \hat{y} REPRESENTA UMA FAIXA
CONTÍNUA DE VALORES

A NATUREZA DE \hat{Y} DETERMINA O SE O PROBLEMA É:

CLASSIFICAÇÃO

QUANDO \hat{Y} DEFINE CATEGORIAS

- IDENTIFICAR GÊNEROS A PARTIR DE ROSTOS
- INDICAR SE UM PACIENTE TEM, OU NÃO, UMA DADA DOENÇA.
- CATEGORIZAR CARTEIRAS DE INVESTIMENTOS EM BOAS OU RUINS.

REGRESSÃO

QUANDO \hat{Y} REPRESENTA UMA FAIXA CONTÍNUA DE VALORES

A NATUREZA DE \hat{Y} DETERMINA O SE O PROBLEMA É:

CLASSIFICAÇÃO

QUANDO Y DEFINE CATEGORIAS

- IDENTIFICAR GÊNEROS A PARTIR DE ROSTOS
- INDICAR SE UM PACIENTE TEM, OU NÃO, UMA DADA DOENÇA.
- CATEGORIZAR CARTEIRAS DE INVESTIMENTOS EM BOAS OU RUINS.

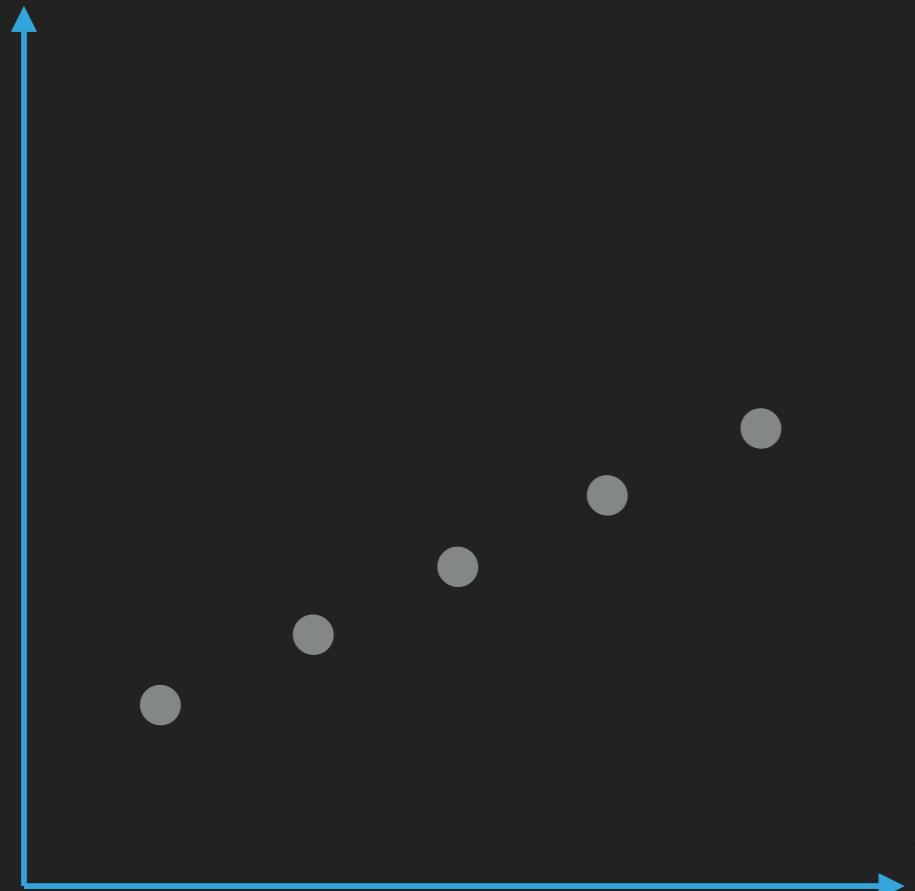
REGRESSÃO

QUANDO Y REPRESENTA UMA FAIXA CONTÍNUA DE VALORES

- ESTIMAR O PREÇO DE VENDA DE UM IMÓVEL
- DETERMINAR O VOLUME DE MATÉRIA PRIMA NECESSÁRIA

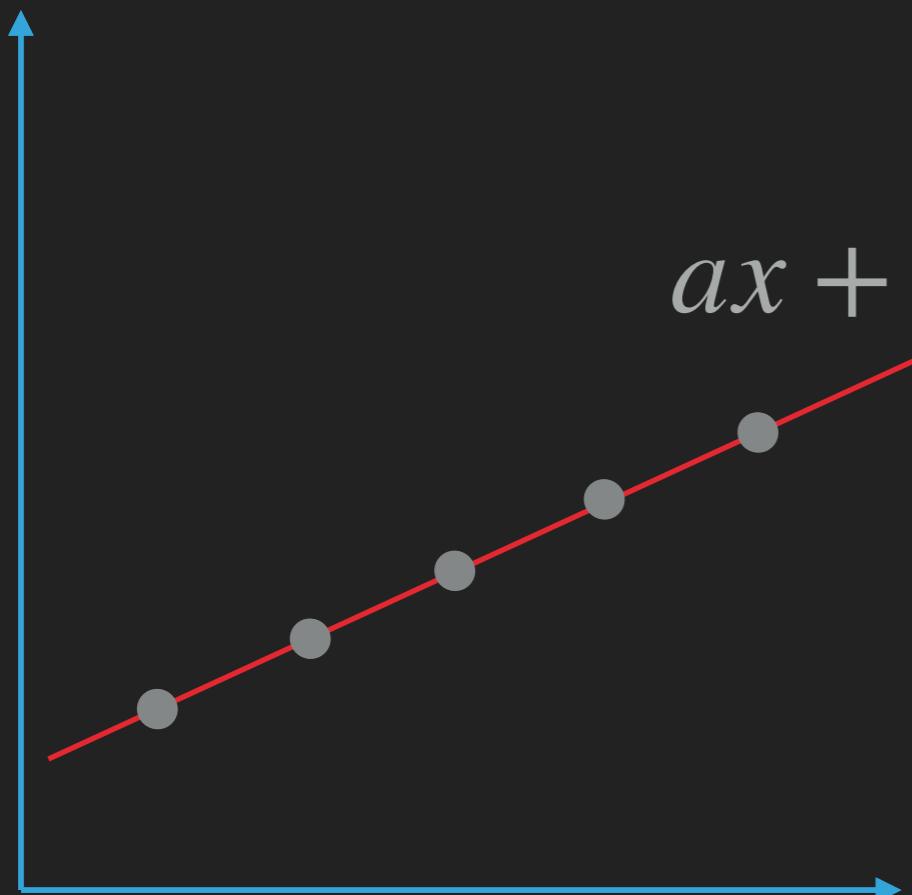
EXEMPLO: FITTING

$f(x)$



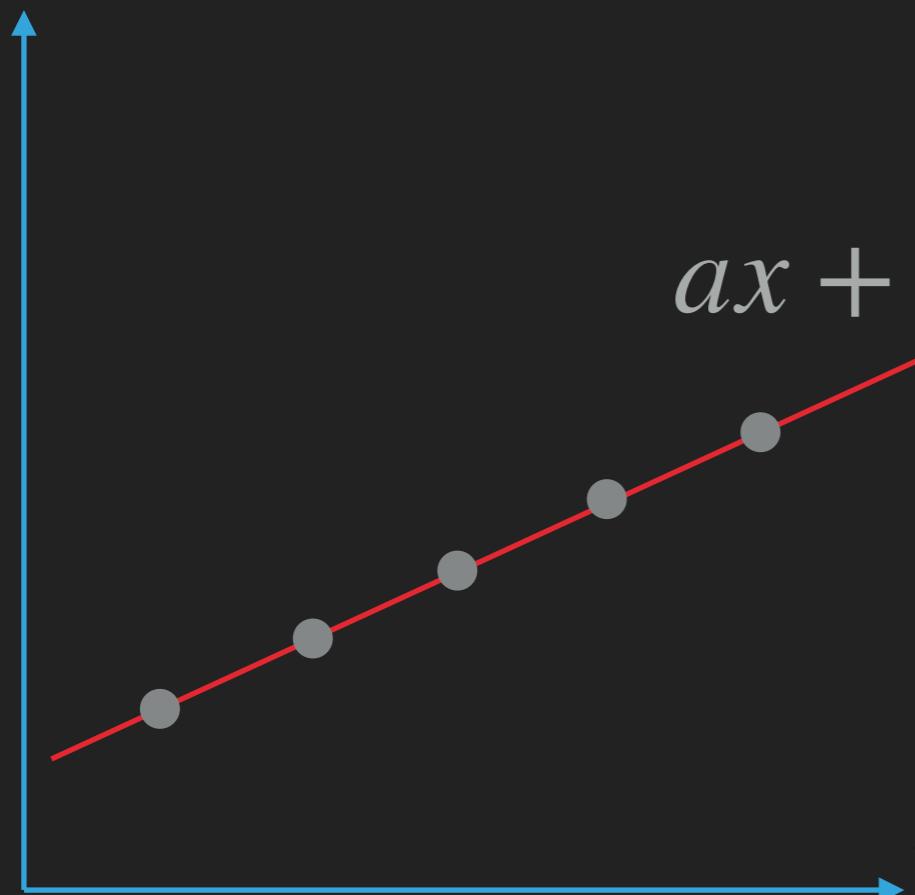
EXEMPLO: FITTING

$f(x)$

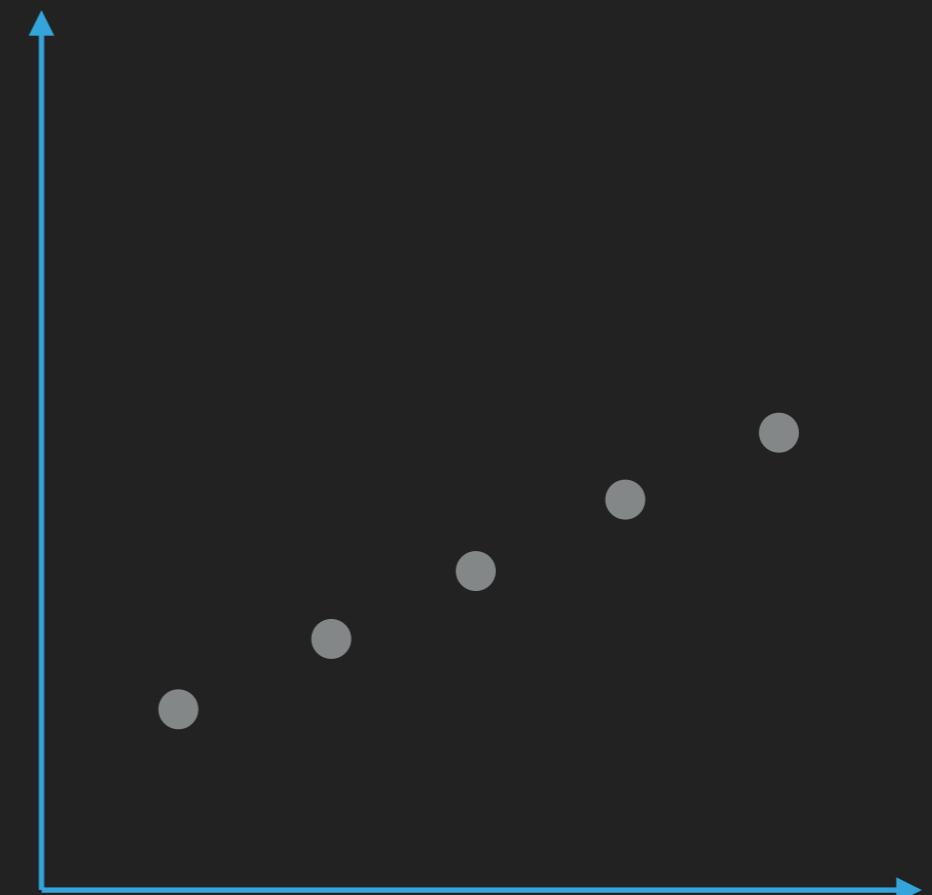


EXEMPLO: FITTING

$f(x)$

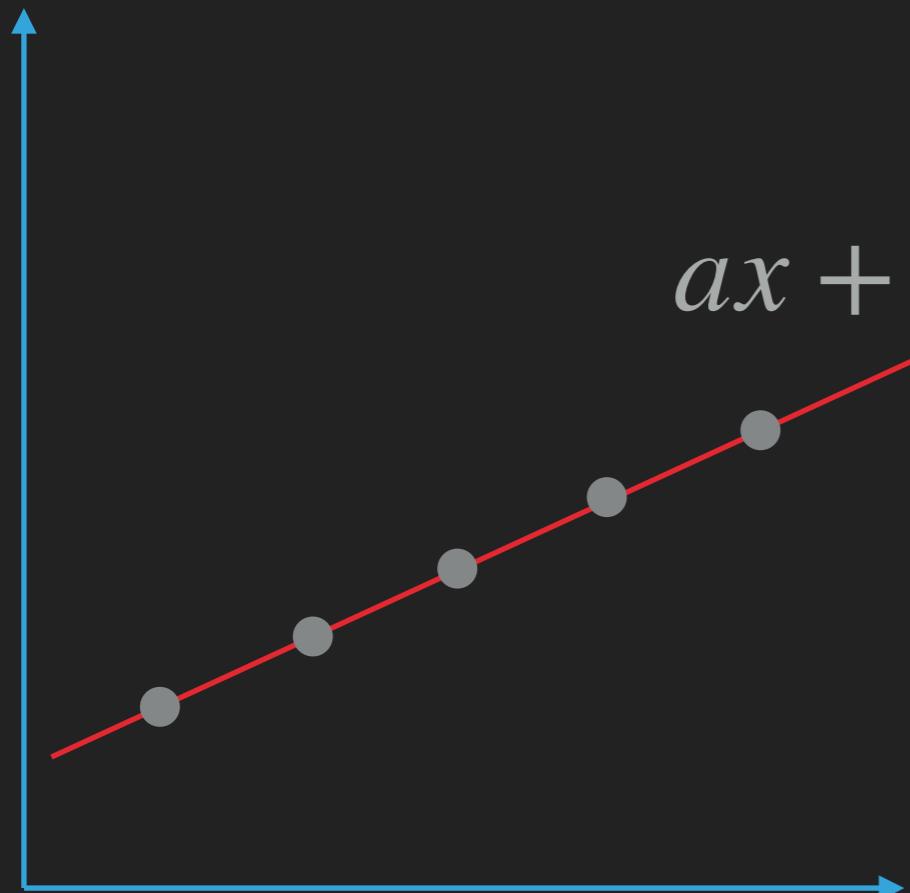


$f(x)$

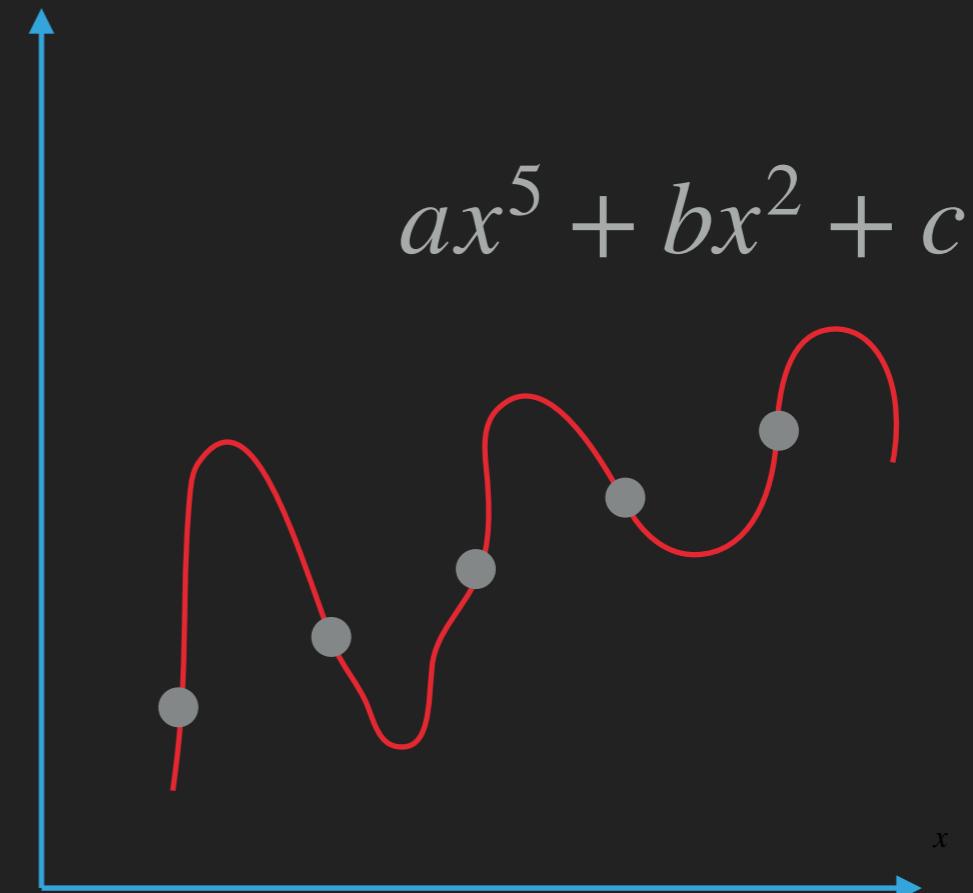


EXEMPLO: FITTING

$f(x)$



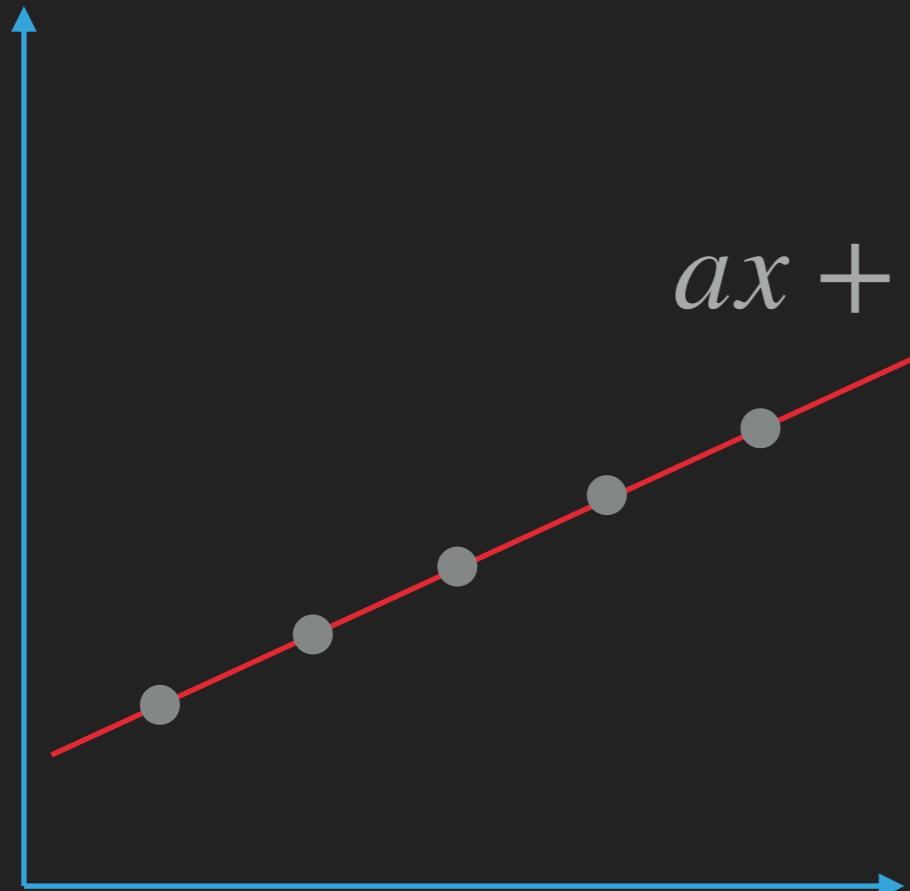
$f(x)$



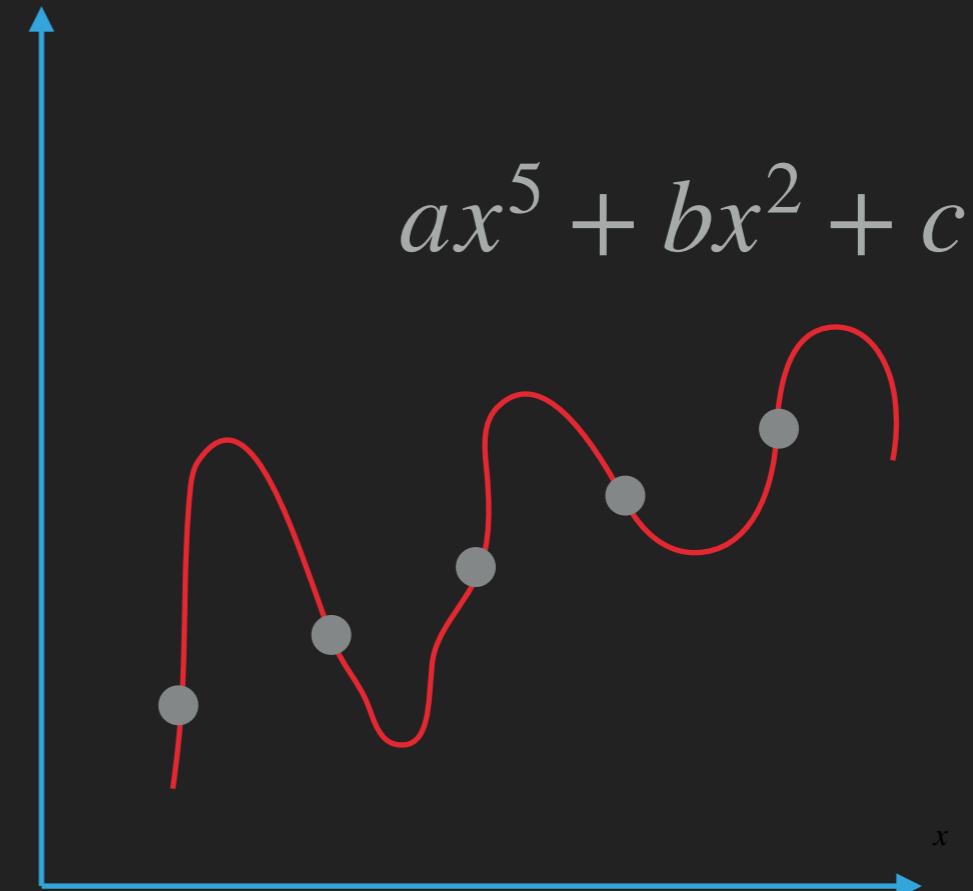
$$ax^5 + bx^2 + c$$

EXEMPLO: FITTING

$f(x)$



$f(x)$



Ambas as funções acima fazem parte do espaço de hipótese.

Qual a mais adequada sendo ambas consistentes?

COMO ESCOLHER ENTRE VÁRIAS HIPÓTESES CONSISTENTES?

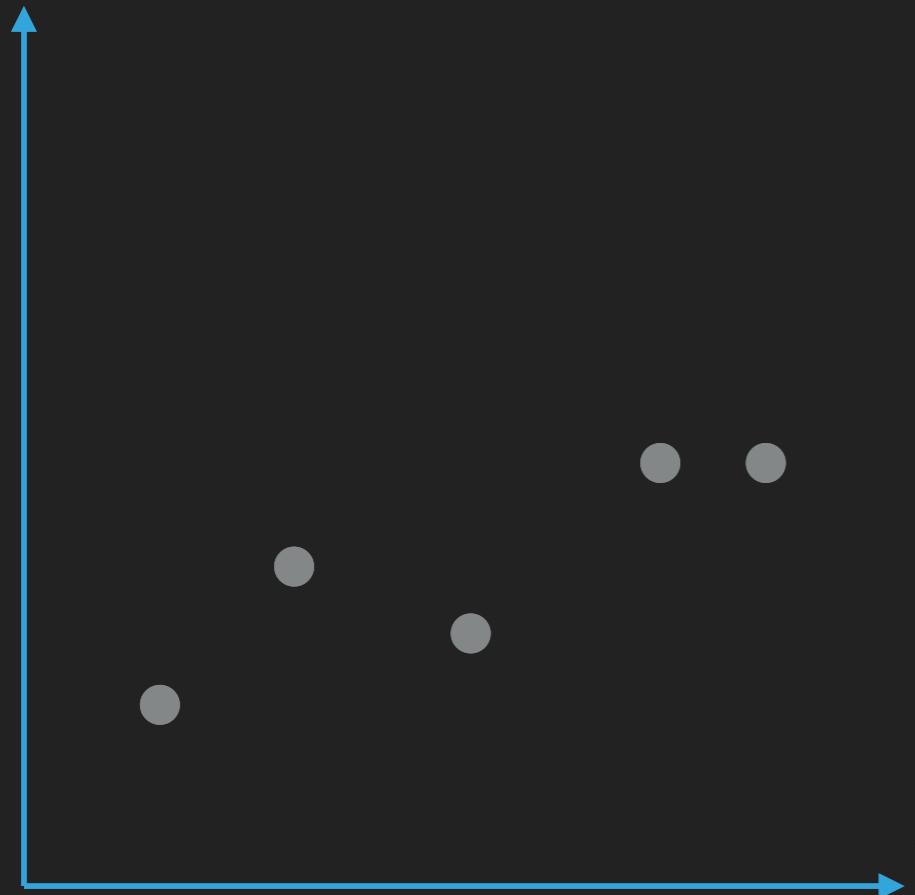


Occam's Razor: No more things should be presumed to exist than are absolutely necessary, i.e., the fewer assumptions an explanation of a phenomenon depends on, the better the explanation.

(William of Occam)

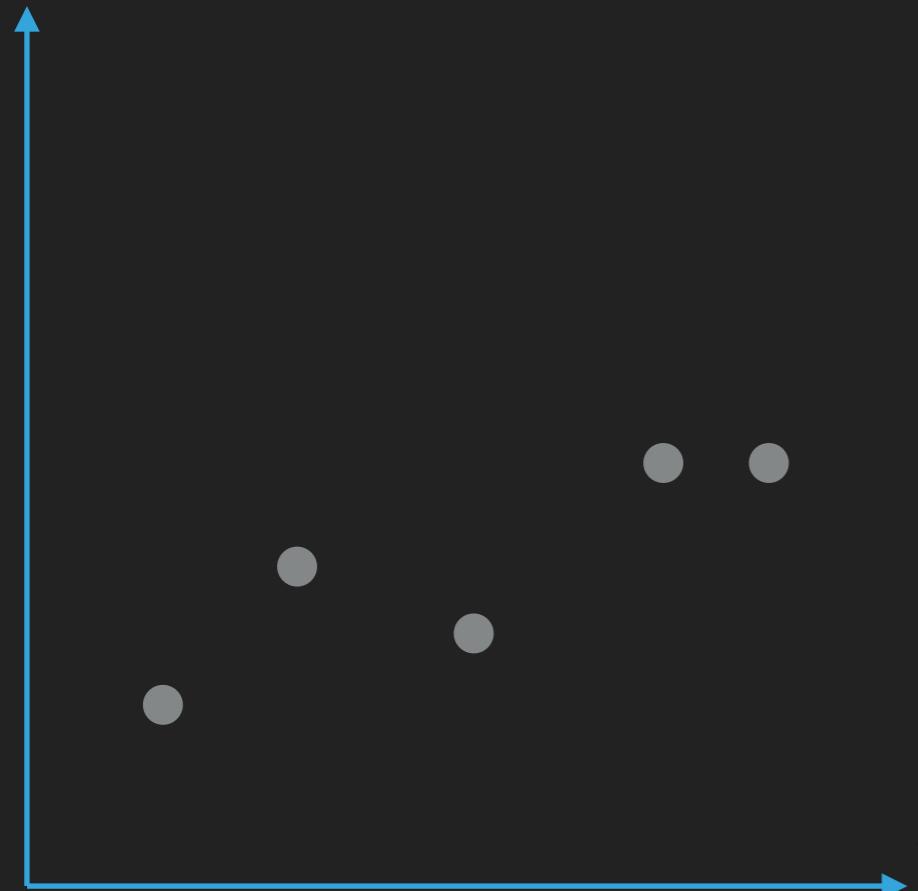
EXEMPLO: FITTING

$f(x)$

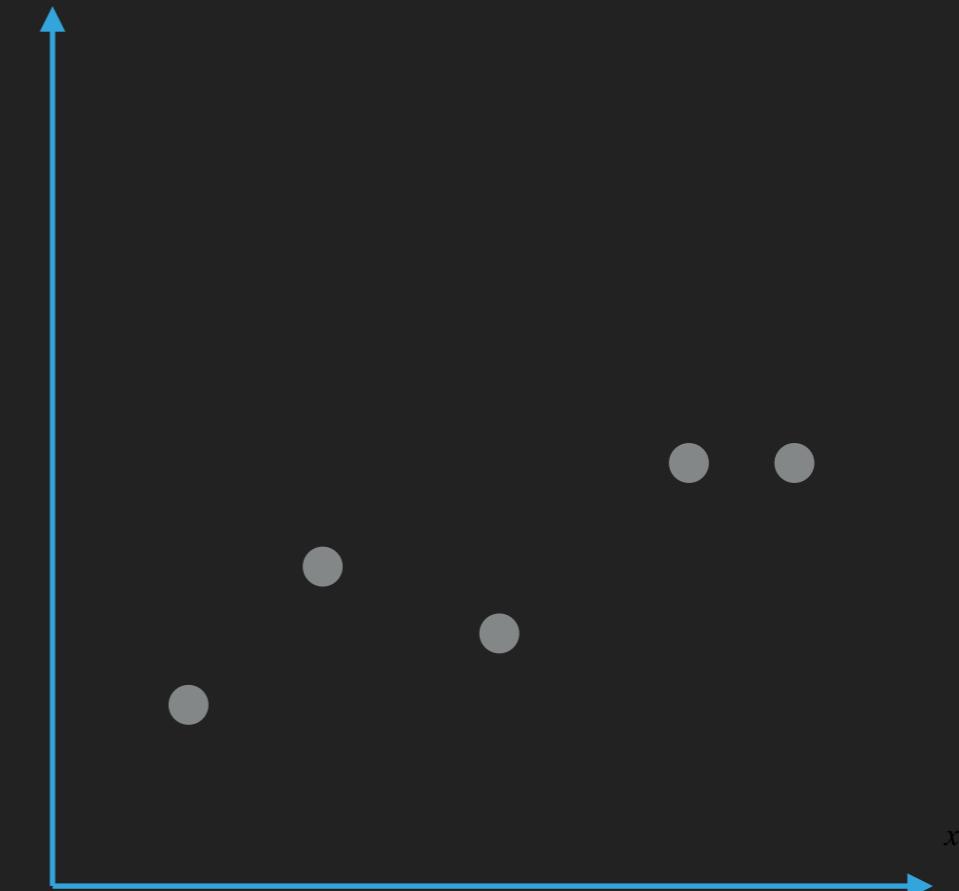


EXEMPLO: FITTING

$f(x)$

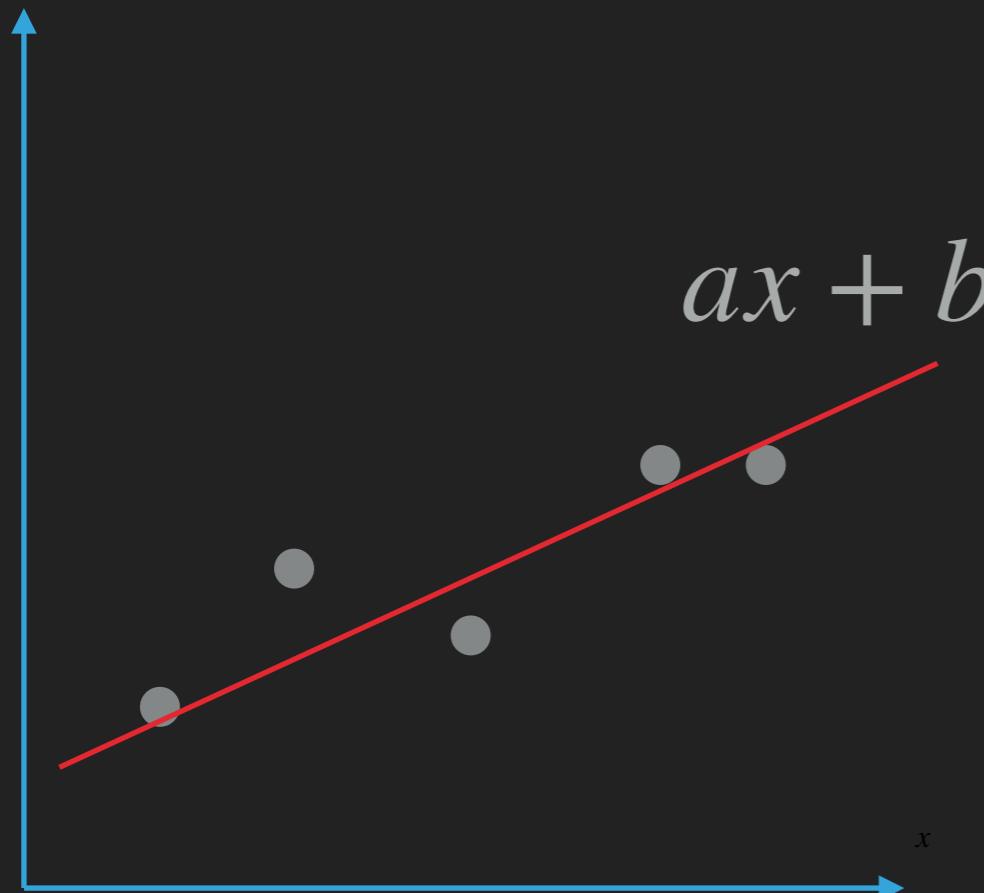


$f(x)$

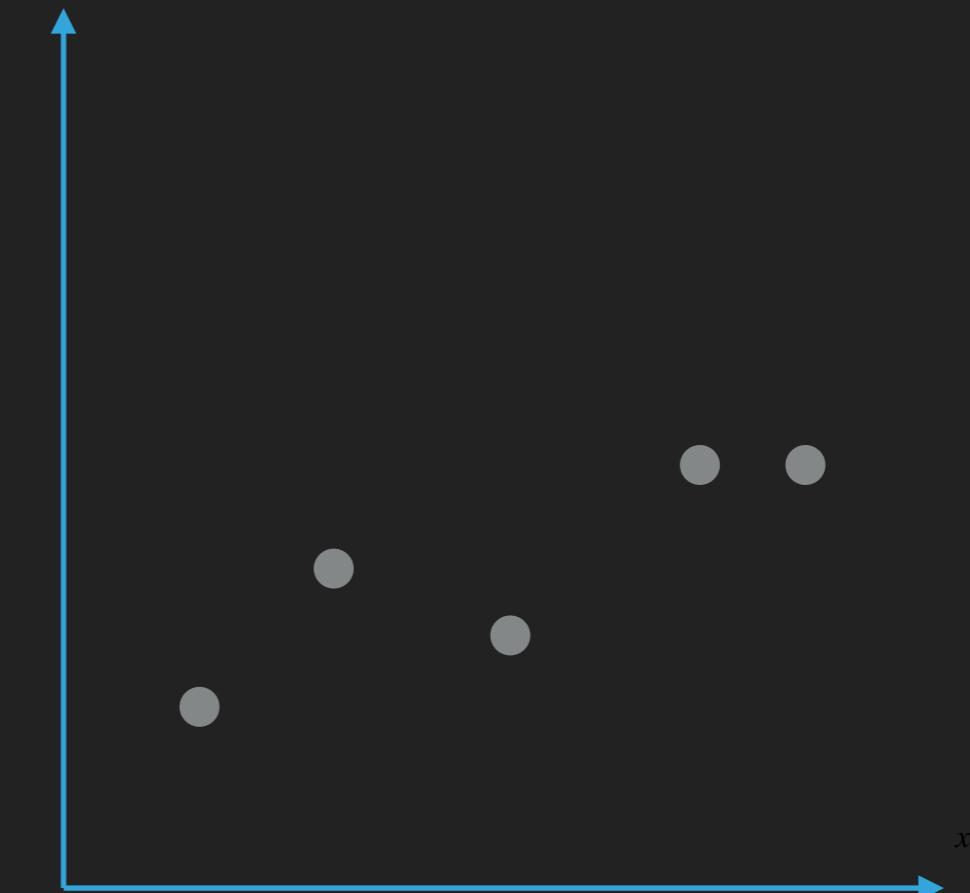


EXEMPLO: FITTING

$f(x)$

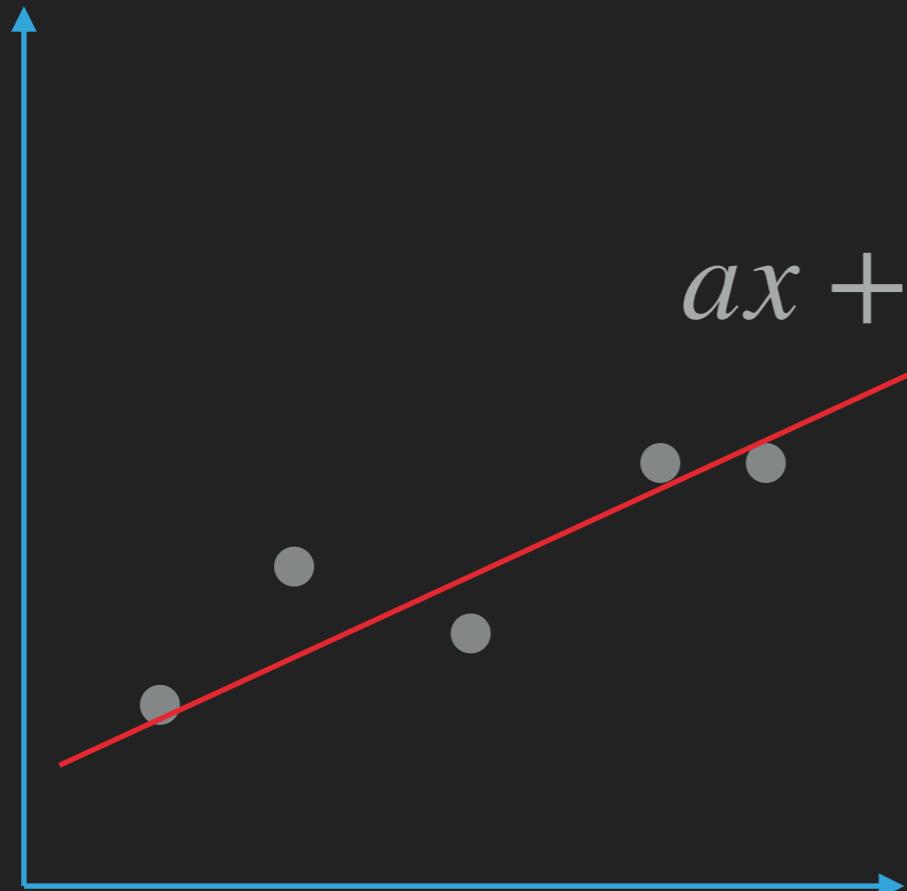


$f(x)$

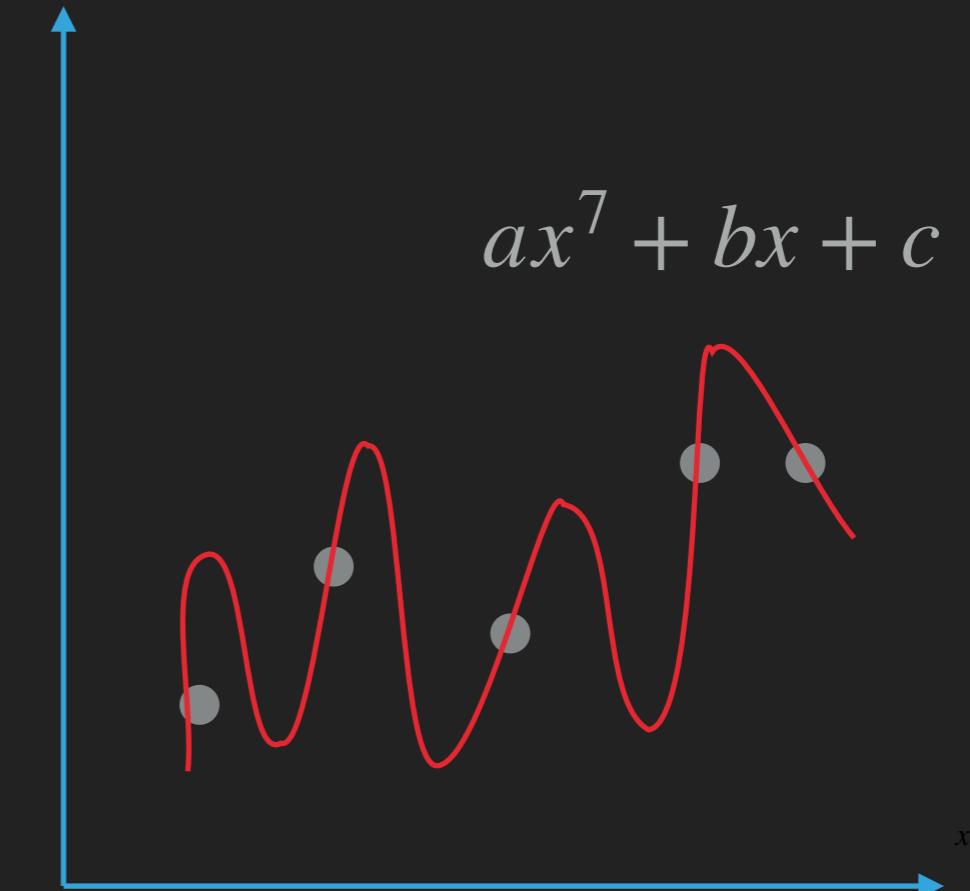


EXEMPLO: FITTING

$f(x)$

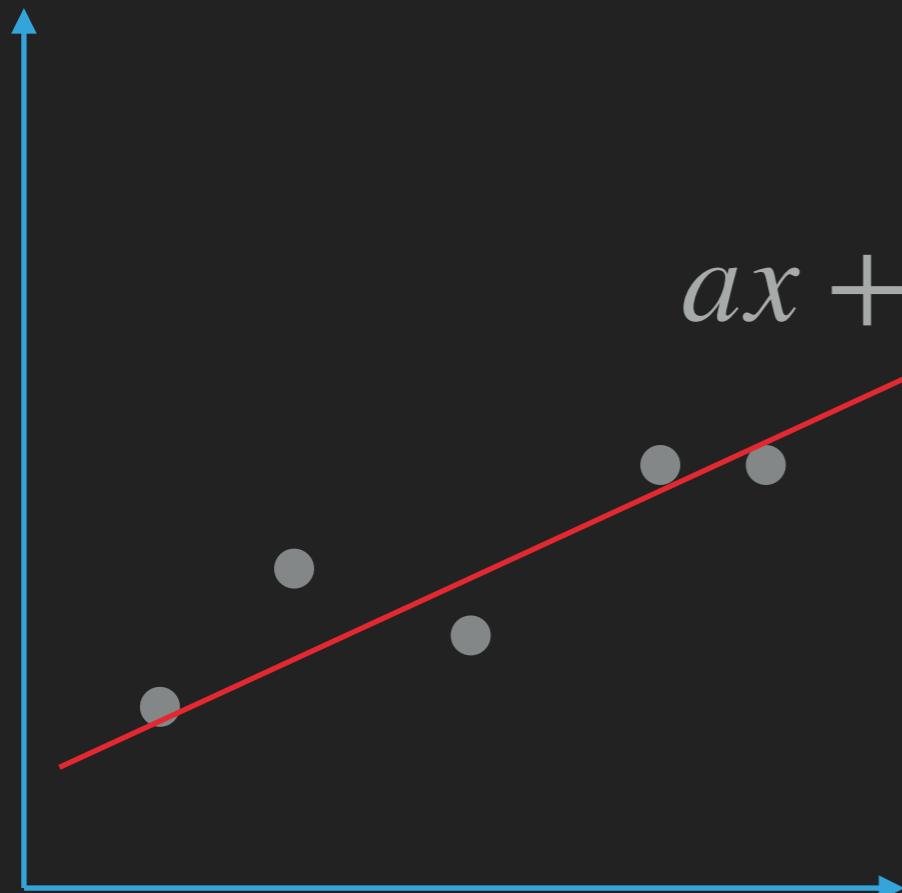


$f(x)$

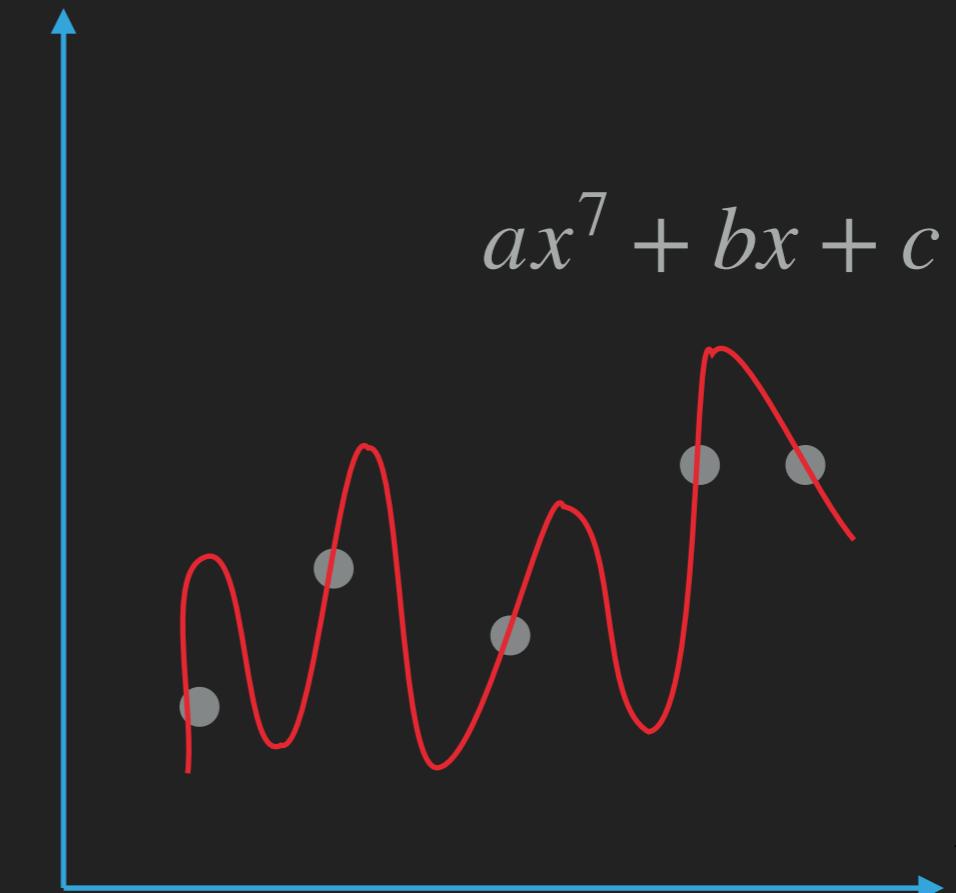


EXEMPLO: FITTING

$f(x)$



$f(x)$



Ambas as funções acima fazem parte do espaço de hipótese.

Qual é a mais adequada sendo apenas uma consistente?

O ESPAÇO DE HIPÓTESES
DEVE SER REALIZÁVEL.

O ESPAÇO DE HIPÓTESES
DEVE SER REALIZÁVEL.

O ESPAÇO DE HIPÓTESES CONTÉM PELO MENOS UMA
HIPÓTESE QUE É CONSISTENTE COM A FUNÇÃO VERDADEIRA



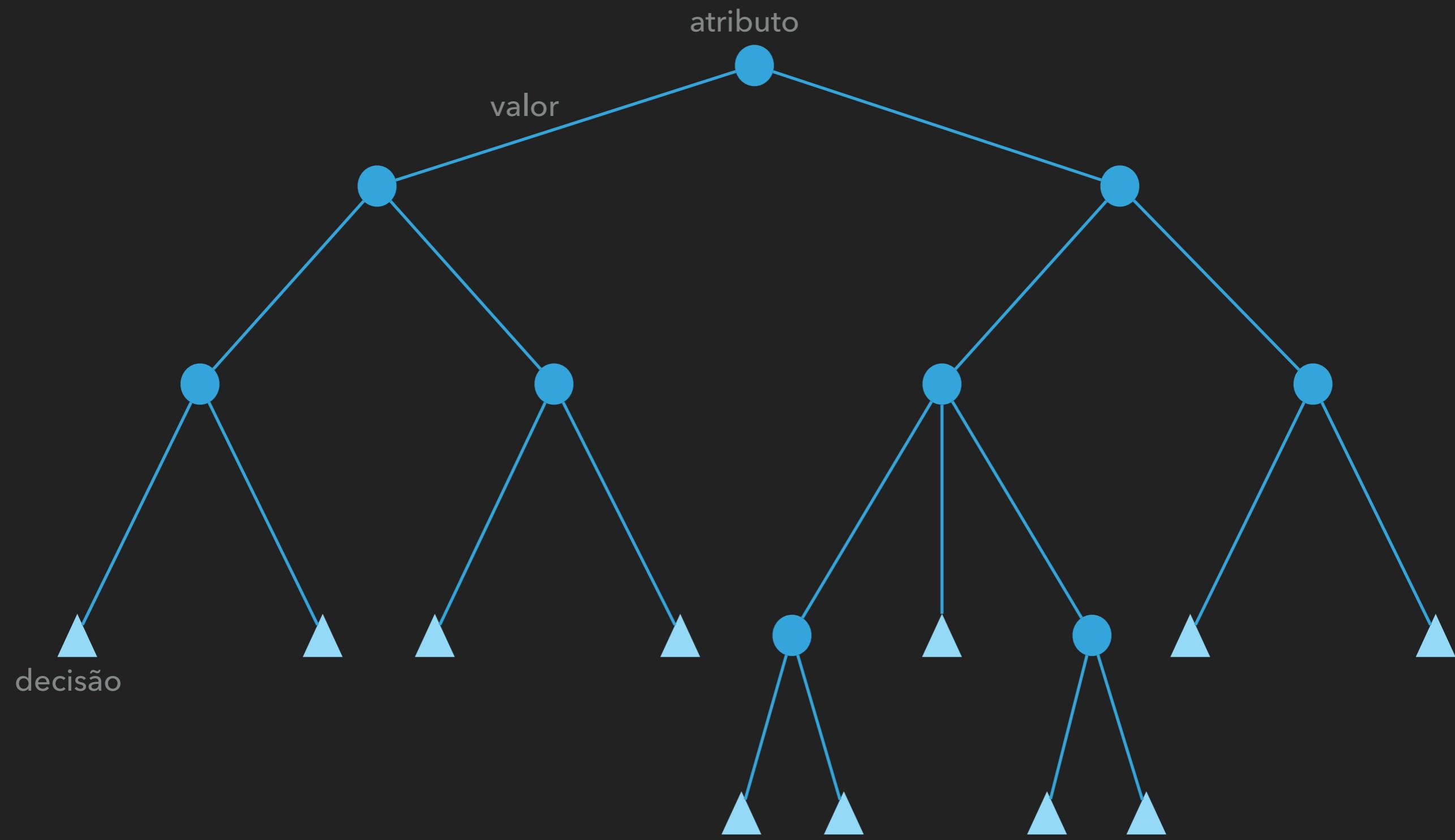
APRENDIZADO DE MÁQUINA

APRENDIZADO DE ÁRVORES DE DECISÃO

Lucas Baggio Figueira [@lucasfigueira]

APRENDIZADO (INDUÇÃO) DE ÁRVORES DE DECISÃO

- ▶ Forma mais simples de aprendizado.
- ▶ Representa uma função $y = f(\mathbf{x})$
- ▶ Vetor de entrada: $\mathbf{x}_i = (A_0, A_1, \dots, A_n)$
 - ▶ Atributos:
 - ▶ Contínuos: $A_i = [0.0 - 99.9]$
 - ▶ Discretos: $A_i = (a, b, c)$
 - ▶ Saída: y (valor singular categórico ou contínuo).



ÁRVORES DE DECISÃO

- Exemplo: construir uma árvore de decisão para determinar quando esperar uma mesa num dado restaurante.



ÁRVORES DE DECISÃO

- Exemplo: construir uma árvore de decisão para determinar se é bom ou ruim um dado restaurante.

QUAL É A ENTRADA (X)?

QUAL É A SAÍDA (Y)?



ATRIBUTOS DE ENTRADA (X)

- ▶ Alternate: existe outro restaurante próximo.
- ▶ Bar: o restaurante possui um bar para espera.
- ▶ Fri/Sat: é sexta ou sábado.
- ▶ Hungry: estamos com fome.
- ▶ Patrons: quantas pessoas estão no restaurante.
- ▶ Price: ticket médio (\$, \$\$, \$\$\$).
- ▶ Raining: chove lá fora.
- ▶ Reservation: fizemos reserva.
- ▶ Type: tipo de culinária.
- ▶ WaitEstimate: espera estimada pelo maitre.

BASE DE DADOS - MAPEAMENTO X/Y

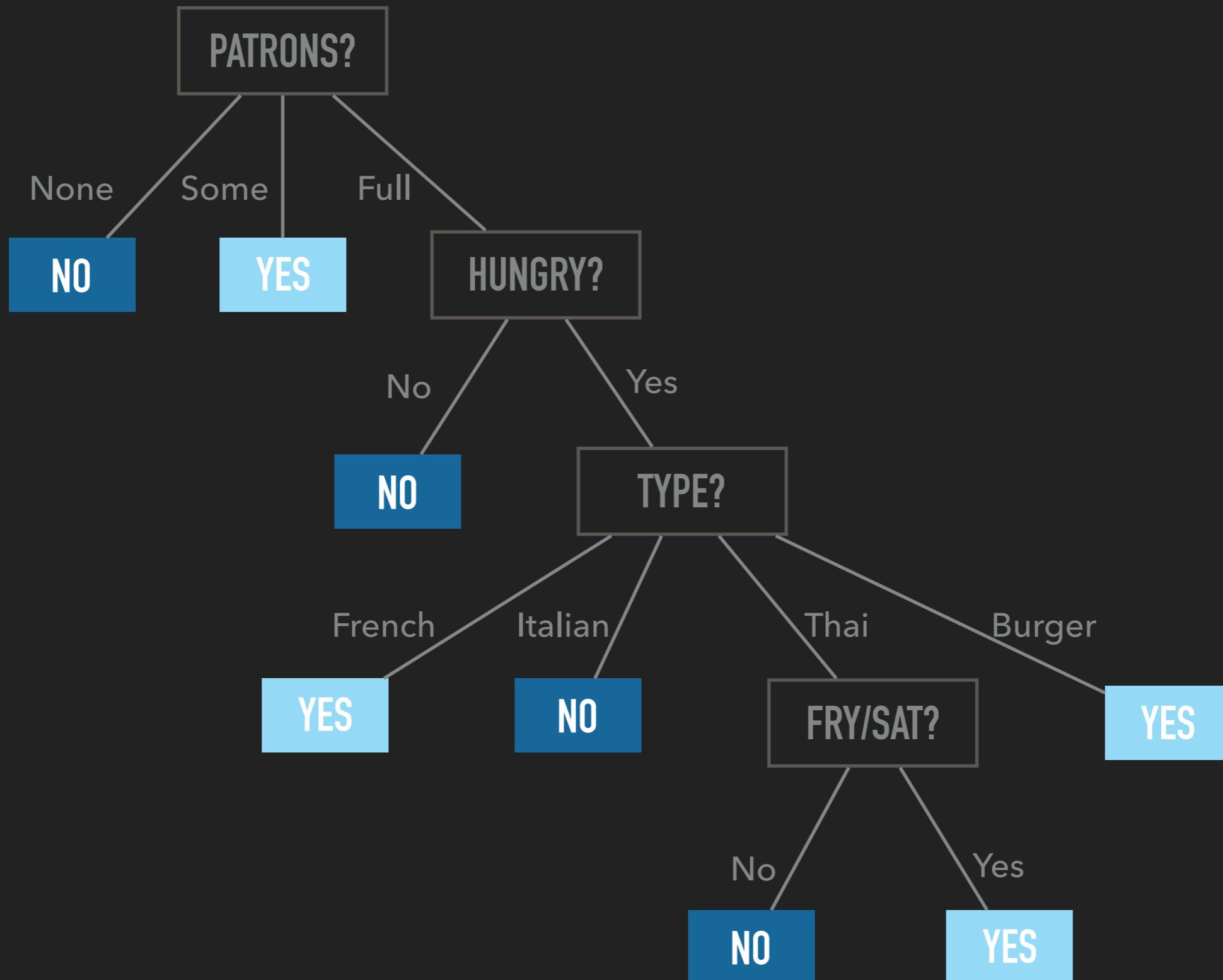
BASE DE DADOS - MAPEAMENTO X/Y

| Example | Input Attributes | | | | | | | | | | Goal <i>WillWait</i> |
|-----------------------|------------------|------------|------------|------------|------------|--------------|-------------|------------|-------------|------------|-------------------------|
| | <i>Alt</i> | <i>Bar</i> | <i>Fri</i> | <i>Hun</i> | <i>Pat</i> | <i>Price</i> | <i>Rain</i> | <i>Res</i> | <i>Type</i> | <i>Est</i> | |
| x₁ | Yes | No | No | Yes | Some | \$\$\$ | No | Yes | French | 0–10 | $y_1 = \text{Yes}$ |
| x₂ | Yes | No | No | Yes | Full | \$ | No | No | Thai | 30–60 | $y_2 = \text{No}$ |
| x₃ | No | Yes | No | No | Some | \$ | No | No | Burger | 0–10 | $y_3 = \text{Yes}$ |
| x₄ | Yes | No | Yes | Yes | Full | \$ | Yes | No | Thai | 10–30 | $y_4 = \text{Yes}$ |
| x₅ | Yes | No | Yes | No | Full | \$\$\$ | No | Yes | French | >60 | $y_5 = \text{No}$ |
| x₆ | No | Yes | No | Yes | Some | \$\$ | Yes | Yes | Italian | 0–10 | $y_6 = \text{Yes}$ |
| x₇ | No | Yes | No | No | None | \$ | Yes | No | Burger | 0–10 | $y_7 = \text{No}$ |
| x₈ | No | No | No | Yes | Some | \$\$ | Yes | Yes | Thai | 0–10 | $y_8 = \text{Yes}$ |
| x₉ | No | Yes | Yes | No | Full | \$ | Yes | No | Burger | >60 | $y_9 = \text{No}$ |
| x₁₀ | Yes | Yes | Yes | Yes | Full | \$\$\$ | No | Yes | Italian | 10–30 | $y_{10} = \text{No}$ |
| x₁₁ | No | No | No | No | None | \$ | No | No | Thai | 0–10 | $y_{11} = \text{No}$ |
| x₁₂ | Yes | Yes | Yes | Yes | Full | \$ | No | No | Burger | 30–60 | $y_{12} = \text{Yes}$ |

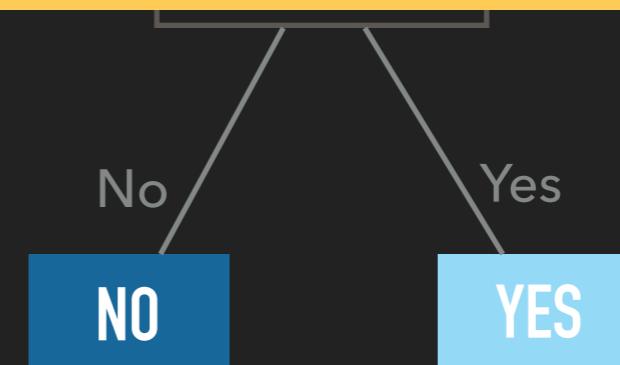
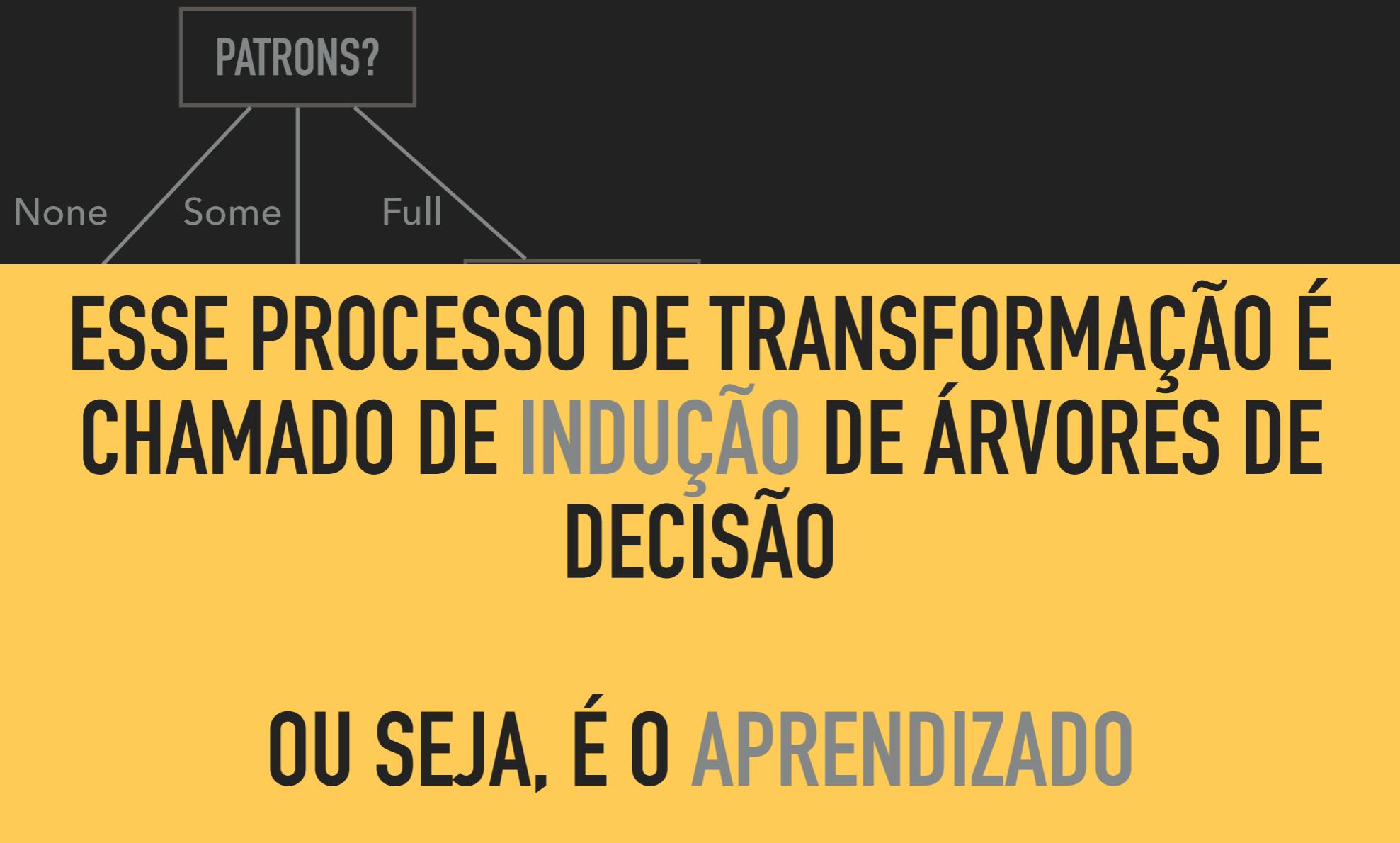
COMO TRANSFORMAR?

| Example | Input Attributes | | | | | | | | | | Goal |
|-------------------|------------------|-----|-----|-----|------|--------|------|-----|---------|-------|-----------------------|
| | Alt | Bar | Fri | Hun | Pat | Price | Rain | Res | Type | Est | |
| \mathbf{x}_1 | Yes | No | No | Yes | Some | \$\$\$ | No | Yes | French | 0–10 | $y_1 = \text{Yes}$ |
| \mathbf{x}_2 | Yes | No | No | Yes | Full | \$ | No | No | Thai | 30–60 | $y_2 = \text{No}$ |
| \mathbf{x}_3 | No | Yes | No | No | Some | \$ | No | No | Burger | 0–10 | $y_3 = \text{Yes}$ |
| \mathbf{x}_4 | Yes | No | Yes | Yes | Full | \$ | Yes | No | Thai | 10–30 | $y_4 = \text{Yes}$ |
| \mathbf{x}_5 | Yes | No | Yes | No | Full | \$\$\$ | No | Yes | French | >60 | $y_5 = \text{No}$ |
| \mathbf{x}_6 | No | Yes | No | Yes | Some | \$\$ | Yes | Yes | Italian | 0–10 | $y_6 = \text{Yes}$ |
| \mathbf{x}_7 | No | Yes | No | No | None | \$ | Yes | No | Burger | 0–10 | $y_7 = \text{No}$ |
| \mathbf{x}_8 | No | No | No | Yes | Some | \$\$ | Yes | Yes | Thai | 0–10 | $y_8 = \text{Yes}$ |
| \mathbf{x}_9 | No | Yes | Yes | No | Full | \$ | Yes | No | Burger | >60 | $y_9 = \text{No}$ |
| \mathbf{x}_{10} | Yes | Yes | Yes | Yes | Full | \$\$\$ | No | Yes | Italian | 10–30 | $y_{10} = \text{No}$ |
| \mathbf{x}_{11} | No | No | No | No | None | \$ | No | No | Thai | 0–10 | $y_{11} = \text{No}$ |
| \mathbf{x}_{12} | Yes | Yes | Yes | Yes | Full | \$ | No | No | Burger | 30–60 | $y_{12} = \text{Yes}$ |

COMO TRANSFORMAR?



COMO TRANSFORMAR?





**OBJETIVO: OBTER A
MENOR ÁRVORE POSSÍVEL
QUE SEJA CONSISTENTE**

INDUÇÃO DE ÁRVORES DE DECISÃO: ALGORITMO GERAL

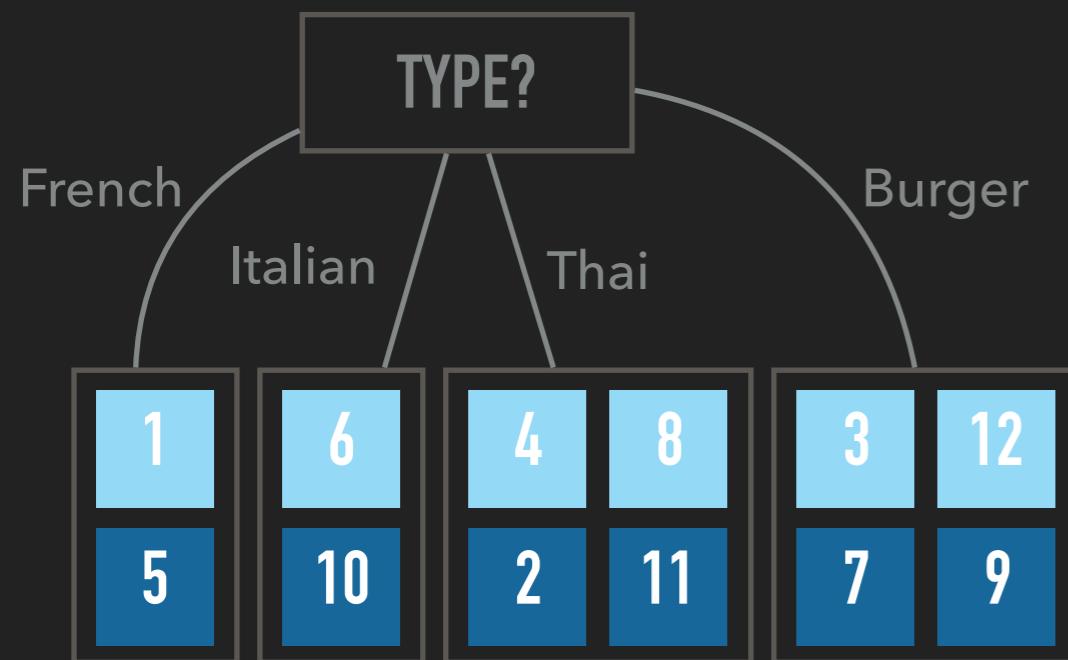
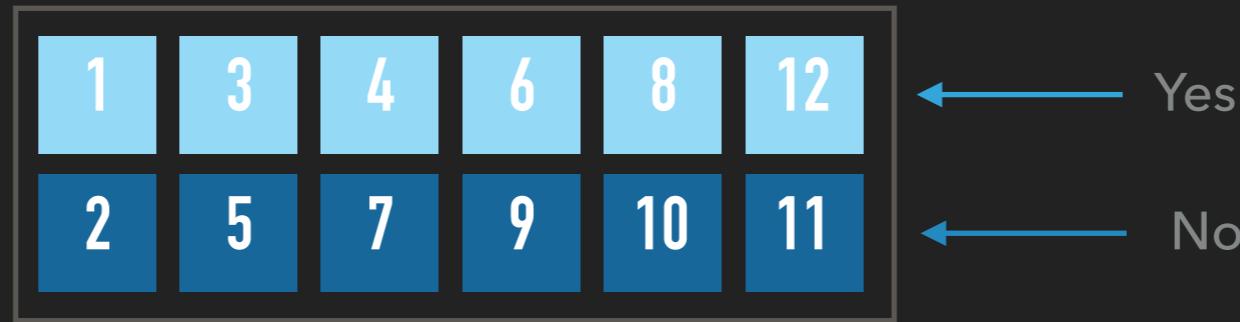
- ▶ algoritmo guloso que adota a estratégia dividir pra conquistar:
- ▶ **sempre testar o atributo mais importante primeiro**
- ▶ dividir o problema em subproblemas menores e tratá-los recursivamente.

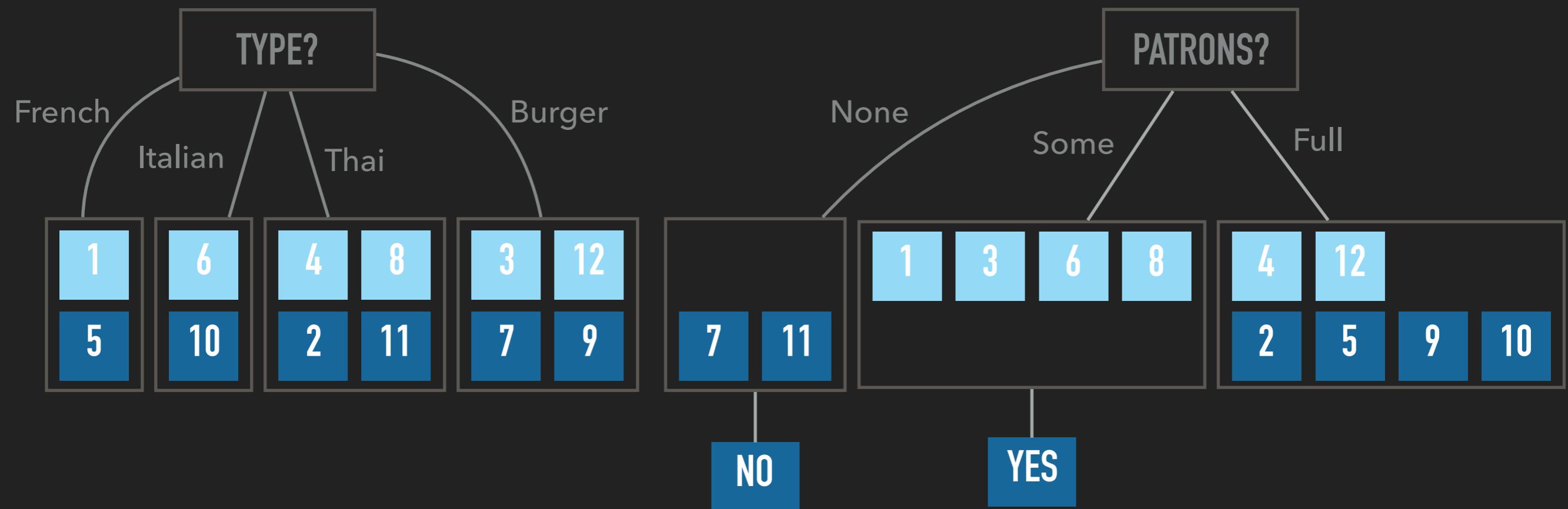
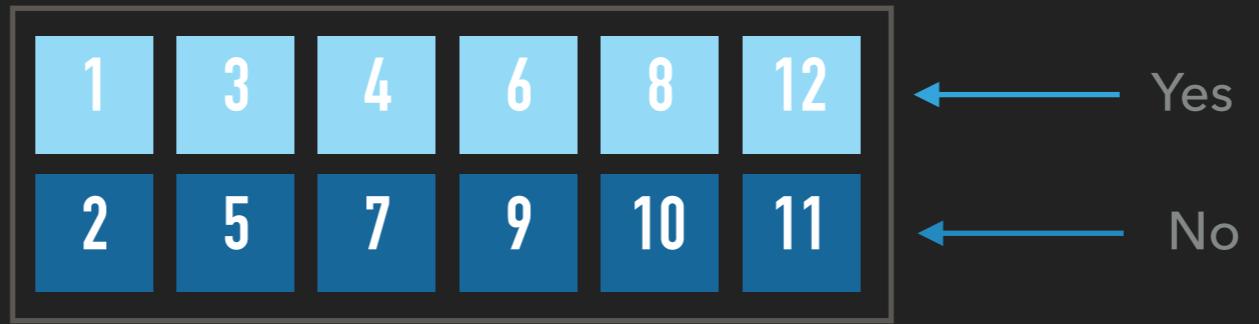
INDUÇÃO DE ÁRVORES DE DECISÃO: ALGORITMO GERAL

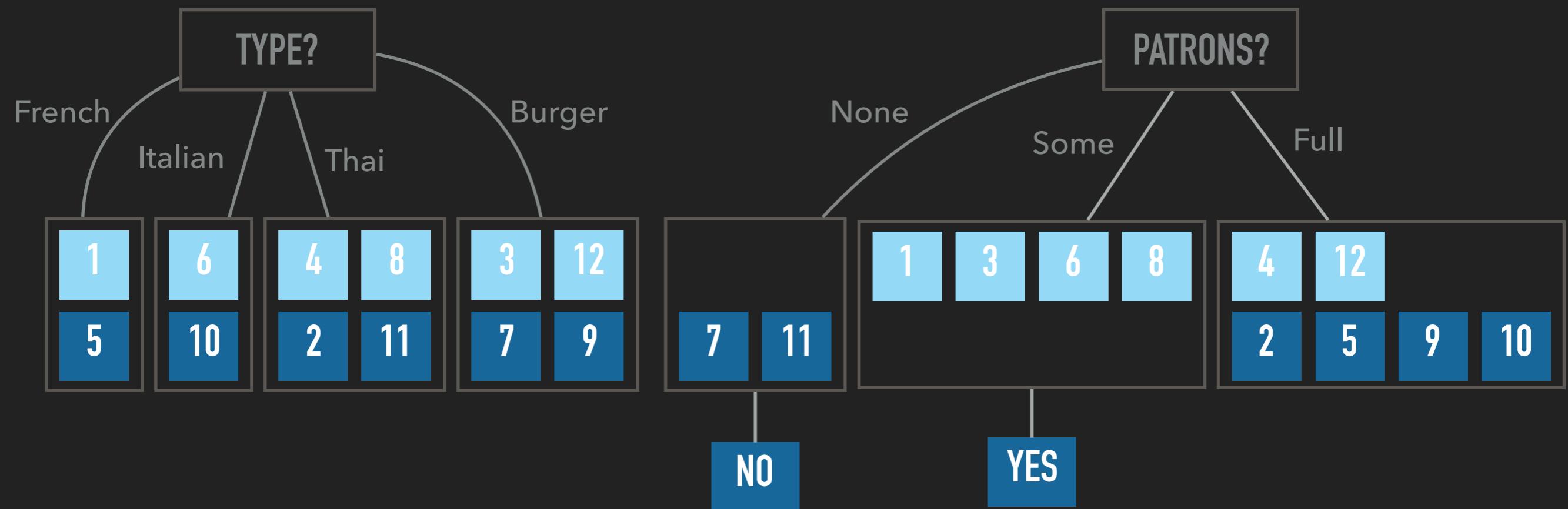
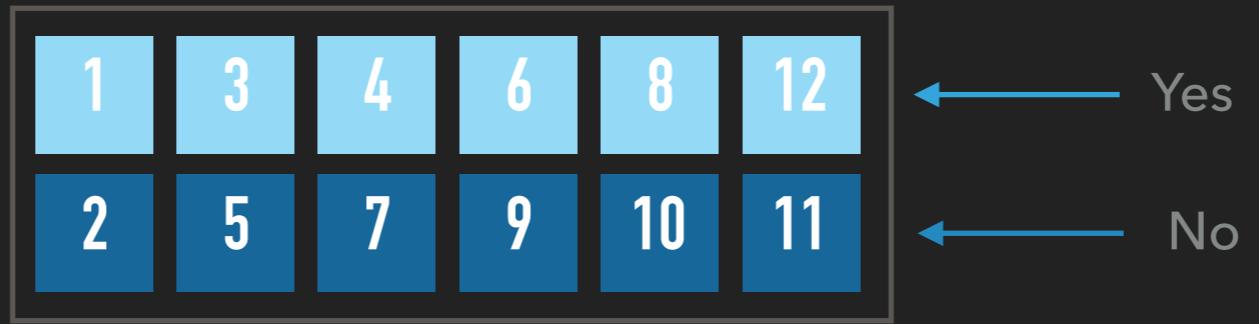
- ▶ algoritmo guloso que adota a estratégia dividir pra conquistar:
- ▶ **sempre testar o atributo mais importante primeiro**
- ▶ dividir o problema em subproblemas menores e tratá-los recursivamente.

O ATRIBUTO MAIS IMPORTANTE É AQUELE QUE MELHOR CLASSIFICA OS DADOS









QUAL É MAIS IMPORTANTE?

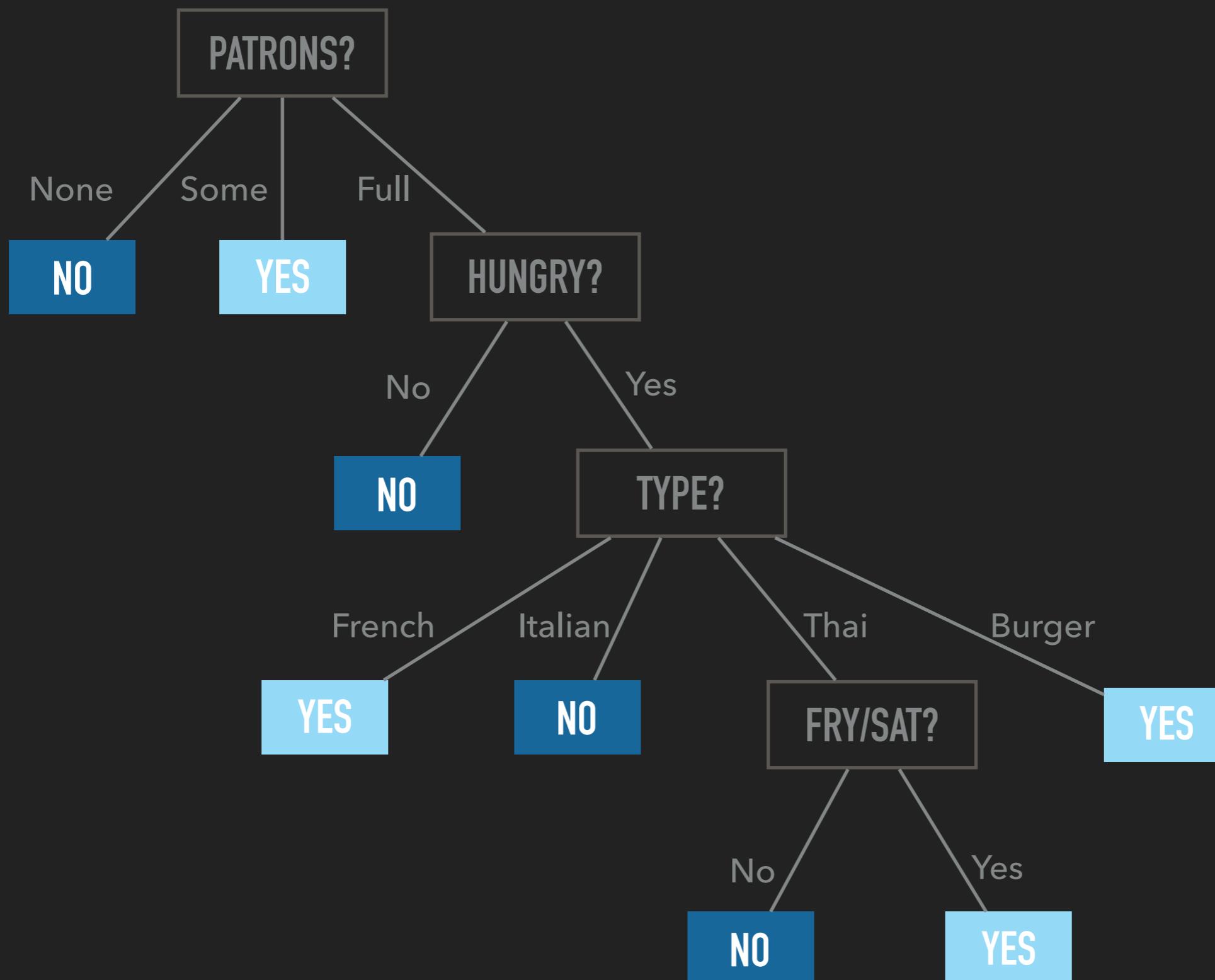
```

function DECISION-TREE-LEARNING(examples, attributes,
                                parent_examples) returns a tree

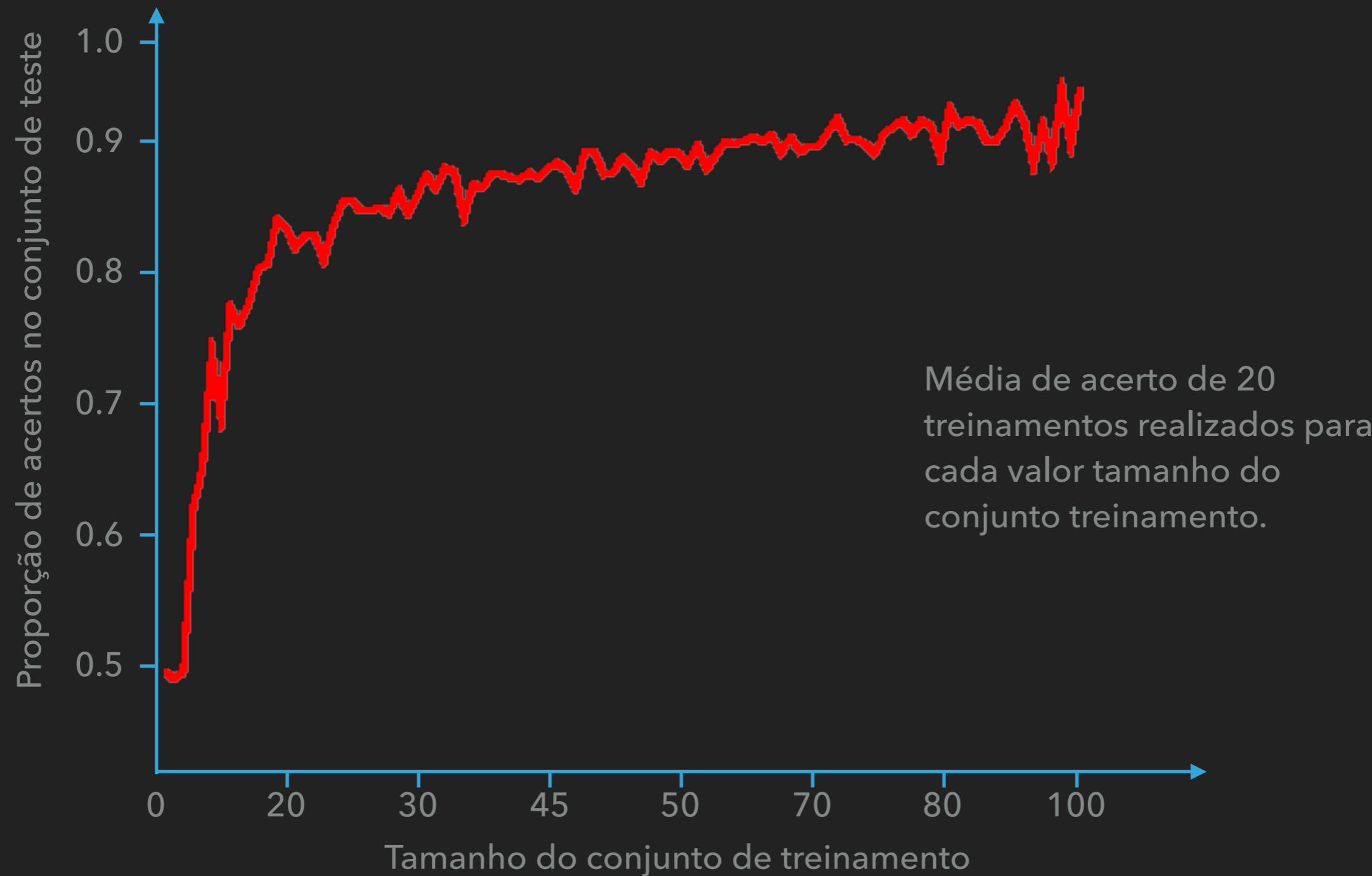
if examples is empty then
    return PLURALITY-VALUE(parent_examples)
else if all examples have the same classification then
    return the classification
else if attributes is empty then
    return PLURALITY-VALUE(examples)
else
    A ← argmaxa ∈ attributes IMPORTANCE(a, examples)
    tree ← a new decision tree with root test A
    for each value vk of A do
        exs ← {e : e ∈ examples and e.A = vk}
        subtree ← DECISION-TREE-LEARNING(
                    exs, attributes - A, examples)
        add a branch to tree with label (A = vk)
                                            and subtree subtree
    return tree

```

ÁRVORE INDUZIDA A PARTIR DO ALGORITMO



CURVA DE APRENDIZADO



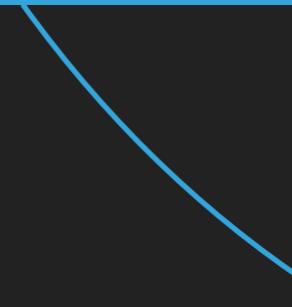
MAS COMO DECIDIR OBJETIVAMENTE
QUAIS ATRIBUTOS CLASSIFICAM MELHOR OS
DADOS?



ENTROPIA

(SHANNON & WEAVER, 1949)

É UMA MEDIDA DE INCERTEZA
SOBRE UMA VARIÁVEL
ALEATÓRIA



ENTROPIA
(SHANNON & WEAVER, 1949)

É UMA MEDIDA DE INCERTEZA
SOBRE UMA VARIÁVEL
ALEATÓRIA

QUANTO MAIOR A ENTROPIA
MAIS INCERTEZA ADVINDA DA
VARIÁVEL

ENTROPIA

(SHANNON & WEAVER, 1949)

É UMA MEDIDA DE INCERTEZA
SOBRE UMA VARIÁVEL
ALEATÓRIA

QUANTO MAIOR A ENTROPIA
MAIS INCERTEZA ADVINDA DA
VARIÁVEL

ENTROPIA

(SHANNON & WEAVER, 1949)

EXEMPLO:

- MOEDA NÃO-VICIADA (ENTROPIA ALTA)
- MOEDA VICIADA (ENTROPIA BAIXA)

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = - \sum_k P(v_k) \log_2 P(v_k)$$

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = - \sum_k P(v_k) \log_2 P(v_k)$$

V - variável aleatória

v_k - um dado valor da variável em questão

$P(v_k)$ - probabilidade de ocorrência do valor na variável em questão

ENTROPIA DAS MOEDAS

$$H(\text{justa}) = - (0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

$$H(\text{viciada}) = - (0.99 \log_2 0.99 + 0.01 \log_2 0.01) \approx 0.08$$



ENTROPIA EM DOMÍNIOS BINÁRIOS

- Em casos da variável Y ser binária temos,

$$B(q) = - (q \log_2 q + (1 - q) \log_2 (1 - q))$$

- Sendo a probabilidade dada como,

$$B(Y) = \left(\frac{p}{p+n} \right)$$

- Exemplo, no caso do restaurante, temos

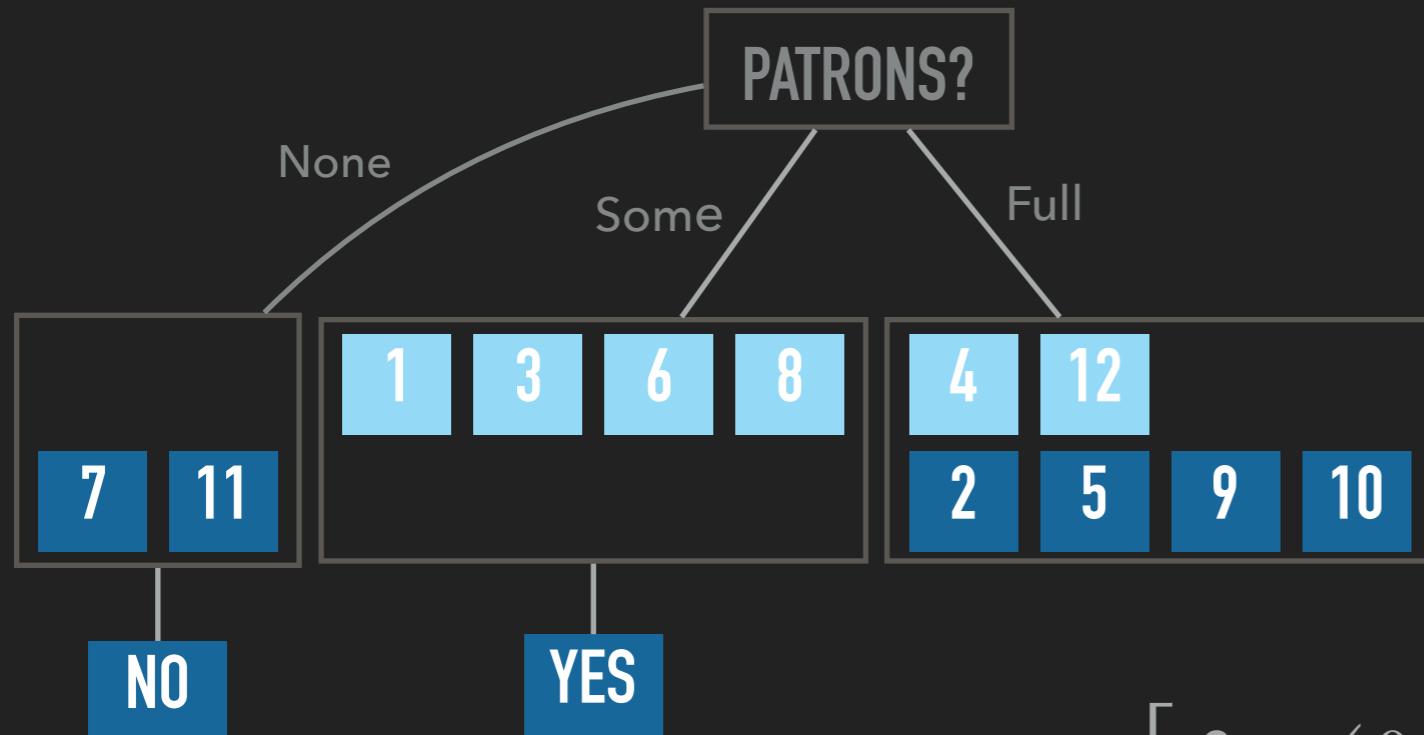
$$B(\text{Wait}) = B(0.5) = 1$$

ENTROPIA EM DOMÍNIOS BINÁRIOS

- ▶ Portanto, para avaliarmos o melhor atributo, temos que avaliar a influência do atributo na entropia de \mathbb{Y} .
- ▶ Ganho de Informação:

$$Gain(A) = B \left(\frac{p}{p+n} \right) - Remainder(A)$$

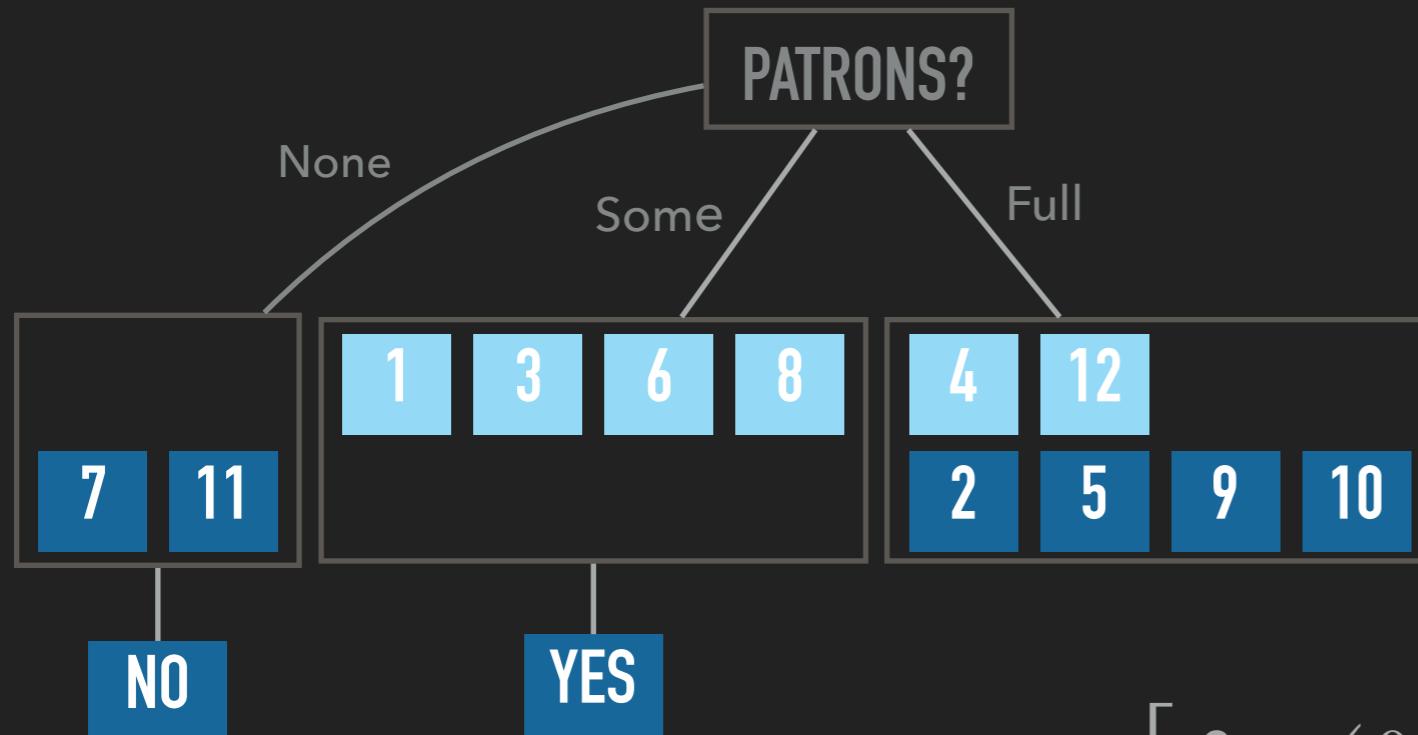
$$Remainder(A) = \sum_{k=1}^d \frac{p_k + n_k}{p+n} B \left(\frac{p_k}{p_k + n_k} \right)$$



NO

YES

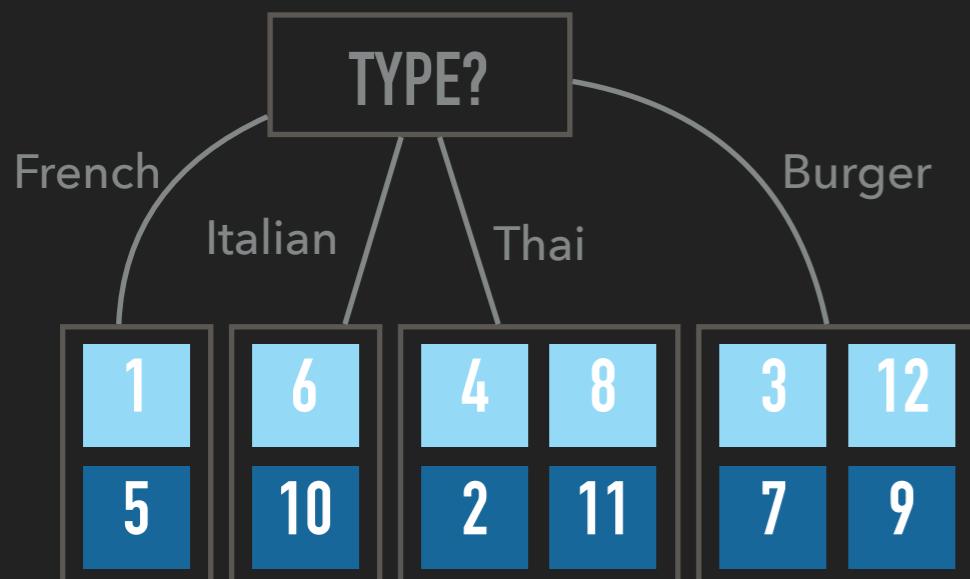
$$Gain(Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right) \right] \approx 0.541$$



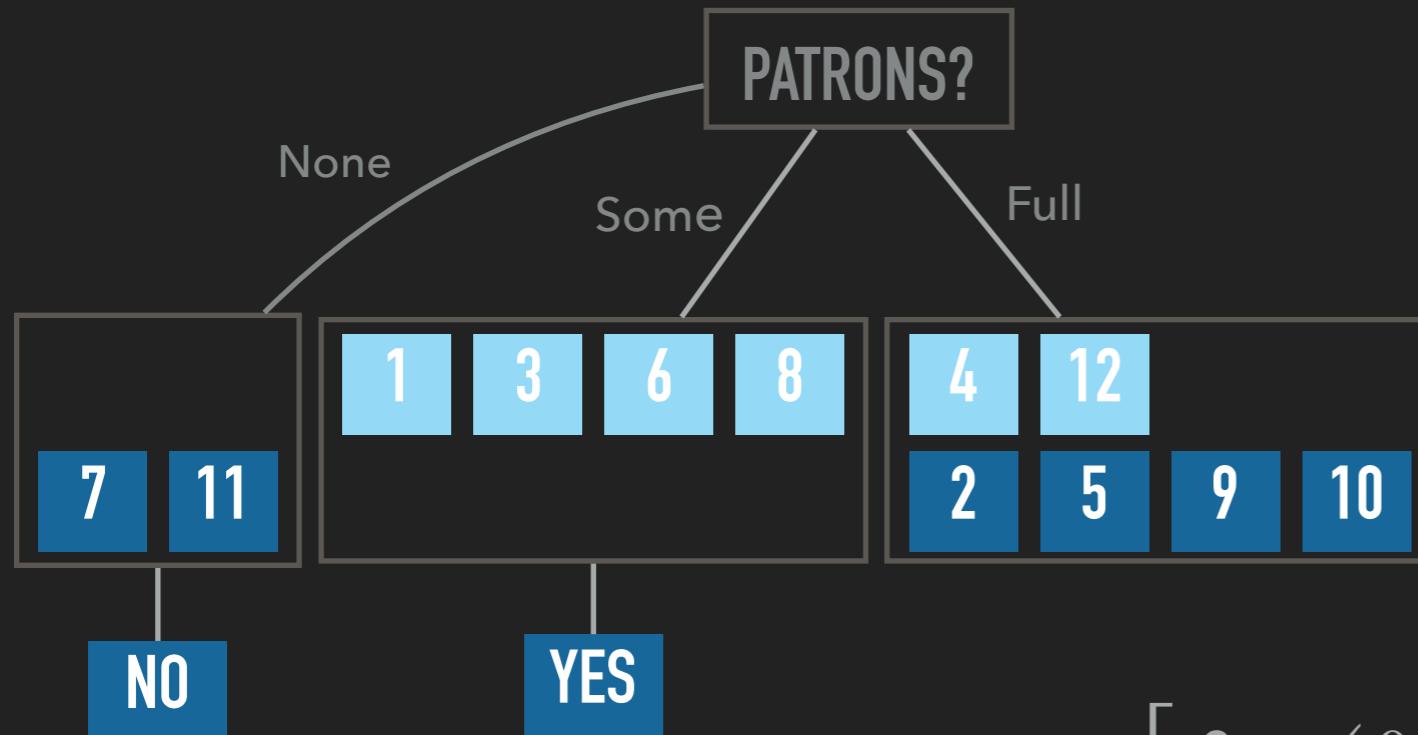
NO

YES

$$Gain(Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right) \right] \approx 0.541$$



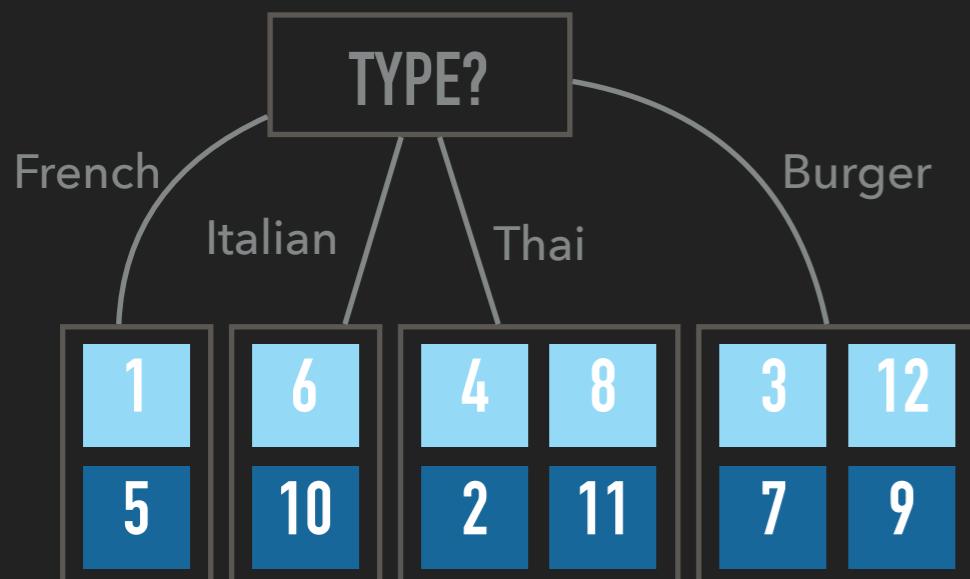
$$Gain(Type) = 1 - \left[\frac{2}{12}B\left(\frac{1}{2}\right) + \frac{2}{12}B\left(\frac{1}{2}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) \right] \approx 0$$



NO

YES

$$Gain(Patrons) = 1 - \left[\frac{2}{12}B\left(\frac{0}{2}\right) + \frac{4}{12}B\left(\frac{4}{4}\right) + \frac{6}{12}B\left(\frac{2}{6}\right) \right] \approx 0.541$$



$$Gain(Type) = 1 - \left[\frac{2}{12}B\left(\frac{1}{2}\right) + \frac{2}{12}B\left(\frac{1}{2}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) + \frac{4}{12}B\left(\frac{2}{4}\right) \right] \approx 0$$

INDUÇÃO DE ÁRVORES DE DECISÃO

- ▶ Existe um problema crônico
 - ▶ Overfitting - super treinamento.
 - ▶ Remédio:
 - ▶ Prunning - Poda
 - ▶ Early Stopping

INDUÇÃO DE ÁRVORES DE DECISÃO

OVERFITTING

INDUÇÃO DE ÁRVORES DE DECISÃO

OVERFITTING

QUANDO O MODELO CLASSIFICA OS DADOS DE
TREINAMENTO EXCESSIVAMENTE BEM.

INDUÇÃO DE ÁRVORES DE DECISÃO

OVERFITTING

QUANDO O MODELO CLASSIFICA OS DADOS DE TREINAMENTO EXCESSIVAMENTE BEM.

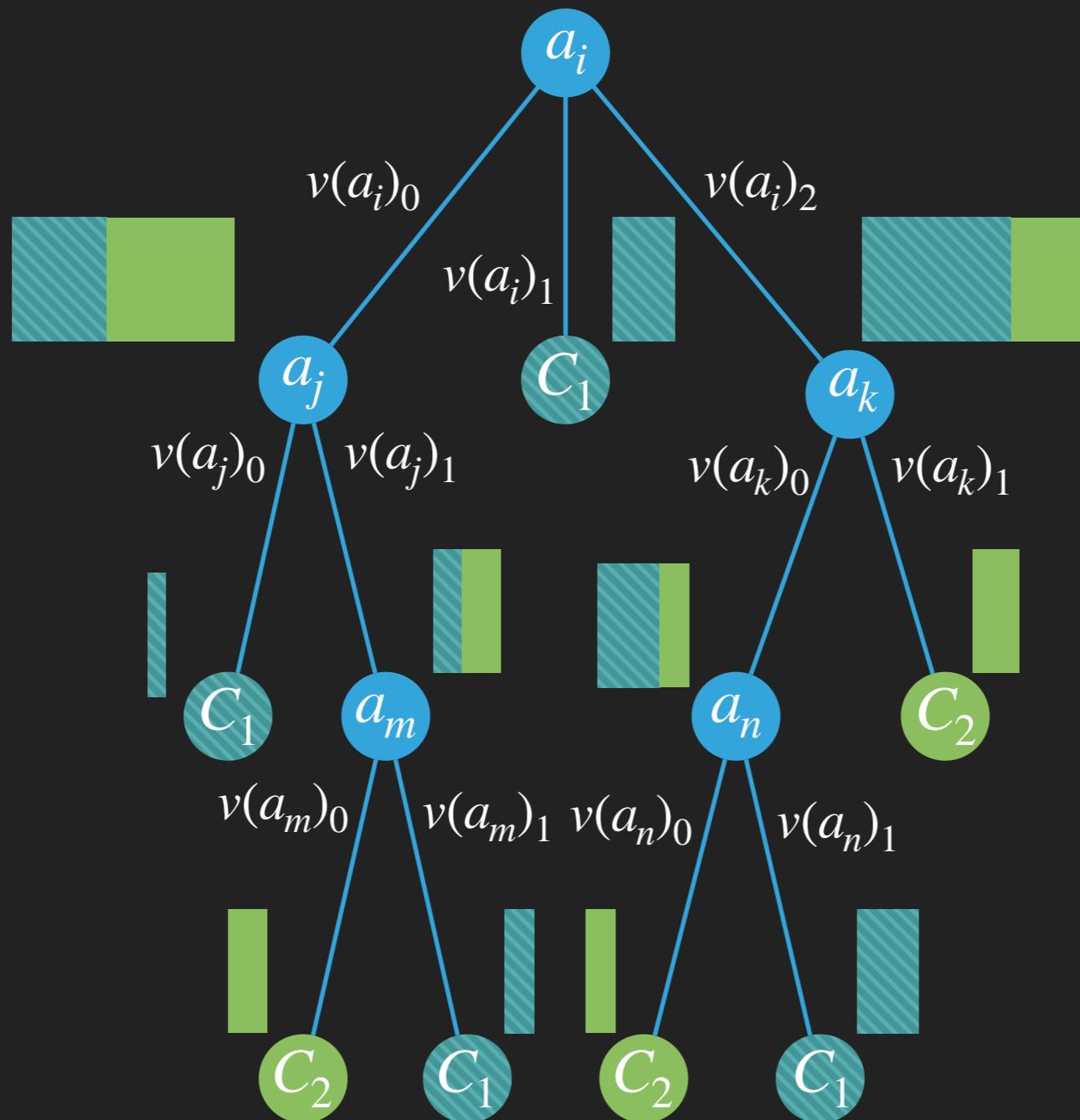


A HIPÓTESE (MODELO) DEIXA DE CAPTURAR AS CARACTERÍSTICAS GERAIS DOS EXEMPLOS E SE APEGA ÀS CARACTERÍSTICAS ESPECÍFICAS DOS MESMOS.

OVERFITTING

CLASSE 1 CLASSE 2

Conjunto de Treinamento

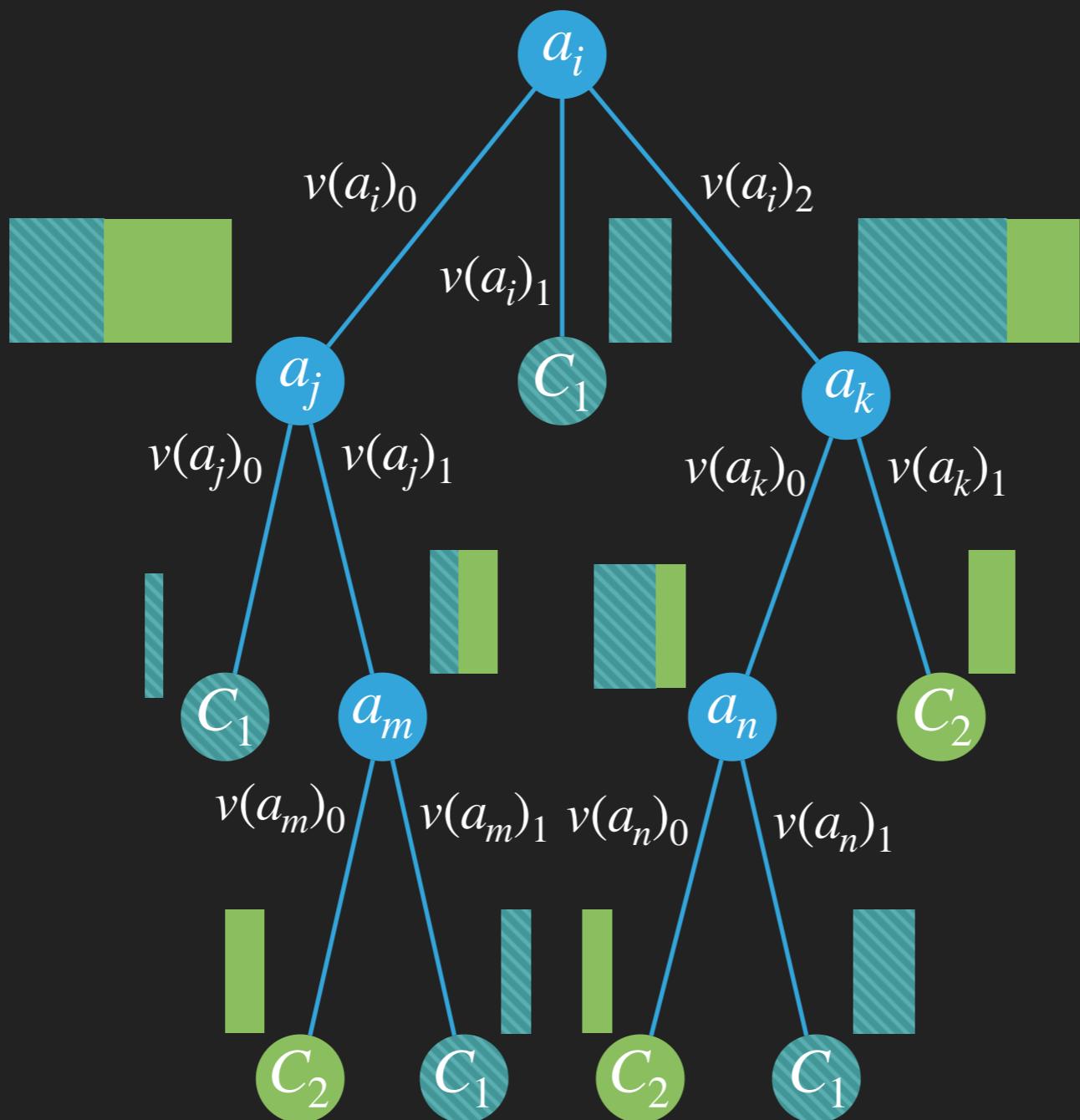


OVERFITTING

CLASSE 1

CLASSE 2

Conjunto de Treinamento



GERAL

GLOBAL

LOCAL

ESPECÍFICO

INDUÇÃO DE ÁRVORES DE DECISÃO

- ▶ Problemas para aplicação:
 - ▶ Atributos com valores ausentes
 - ▶ Atributos multivalorados
 - ▶ Atributos contínuos
 - ▶ Saída (Y) com valores contínuos - regressão.

HANDS ON



Lucas Baggio Figueira [@lucasfigueira]