



Regressão linear

Fatec 2025

Aprendizado de máquina

“Aprendizado de Máquina é a ciência (e a arte) de se programar computadores para que eles **aprendam a partir dos dados.**”

[Aurélien Géron, 2017]





“Área de estudo que dá aos computadores a habilidade de **aprender** sem que sejam explicitamente programados”

[Arthur Samuel, 1959]

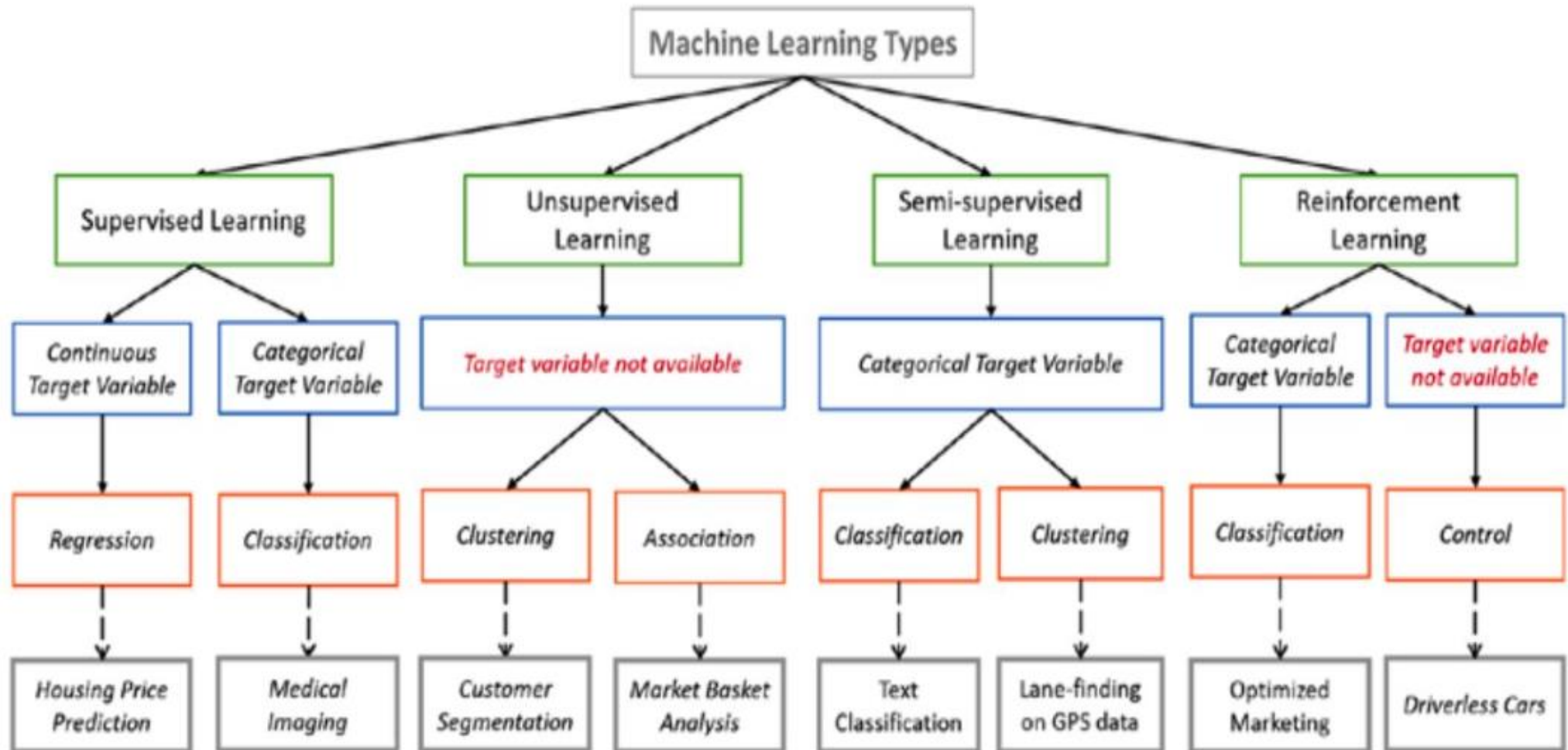
“Um programa de computador **aprende a partir de uma experiência E** com respeito a uma tarefa **T** e uma métrica de performance **P**, se a sua performance em **T**, medida por **P**, melhora com a experiência **E.**”

[Tom Mitchell, 1997]

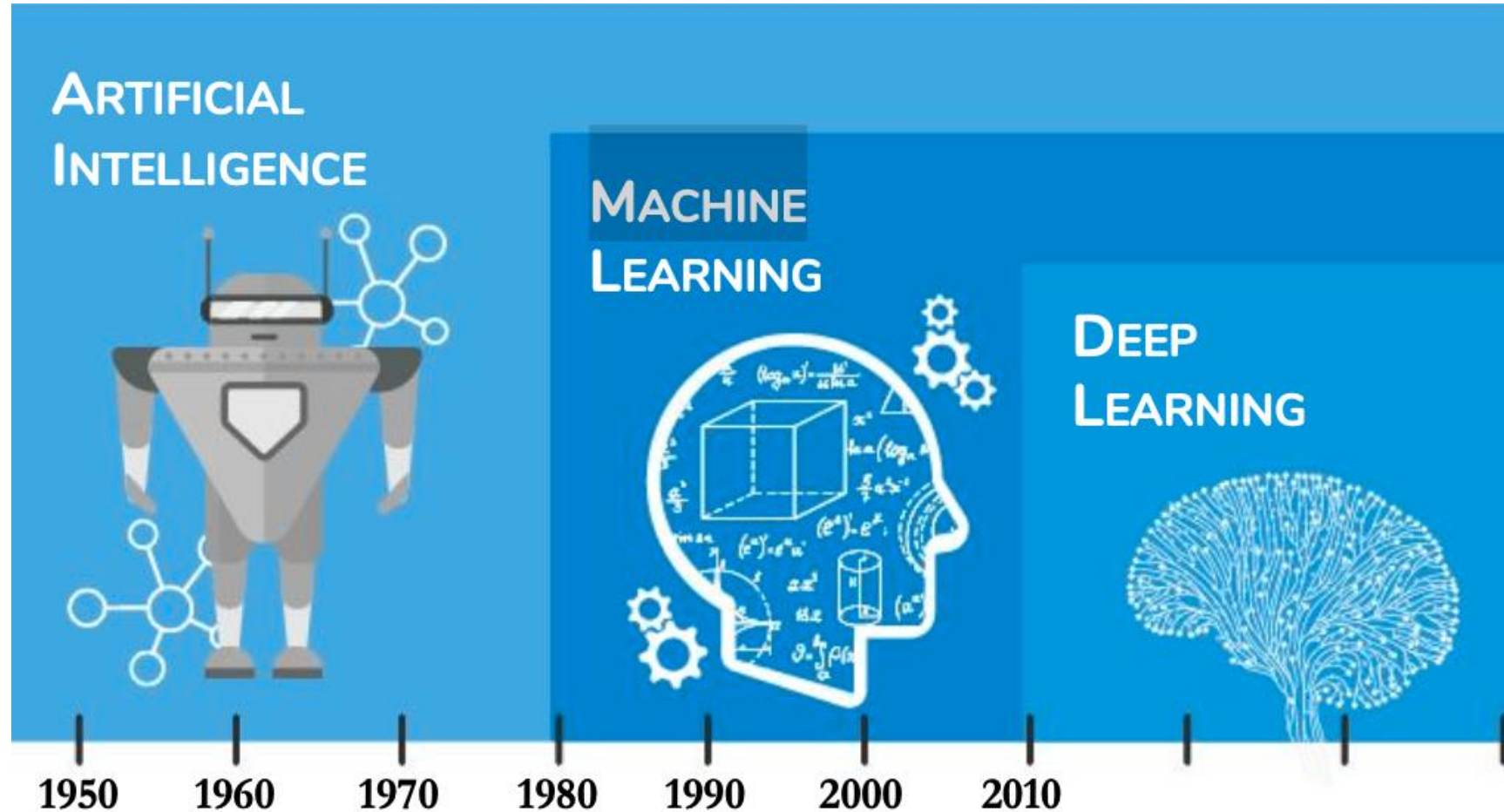
Tipos de aprendizado de máquina

Tipo de Aprendizado	Como funciona	Tem resposta certa nos dados?	Exemplo prático
Supervisionado	Aprende com dados que já têm rótulos (respostas certas)	 Sim	Classificar e-mails como spam ou não
Não supervisionado	Encontra padrões em dados sem rótulos	 Não	Agrupar clientes por comportamento
Semi-supervisionado	Usa poucos dados rotulados e muitos sem rótulo para aprender melhor	 Parcial	Identificar doenças com base em poucos exames rotulados
Por Reforço	Aprende com recompensas ou penalidades após cada ação (tentativa e erro)	 Parcial (por feedback)	Treinar um agente para jogar videogame ou controlar robôs

Tipos de aprendizado de máquina



Contexto



Aprendizado supervisionado

- Classificação é usada para prever valores discretos
- Regressão é usada para prever valores contínuos

Predição de preço de imóveis (Regressão)



\$ 70 000

Predição de preço de imóveis (Regressão)



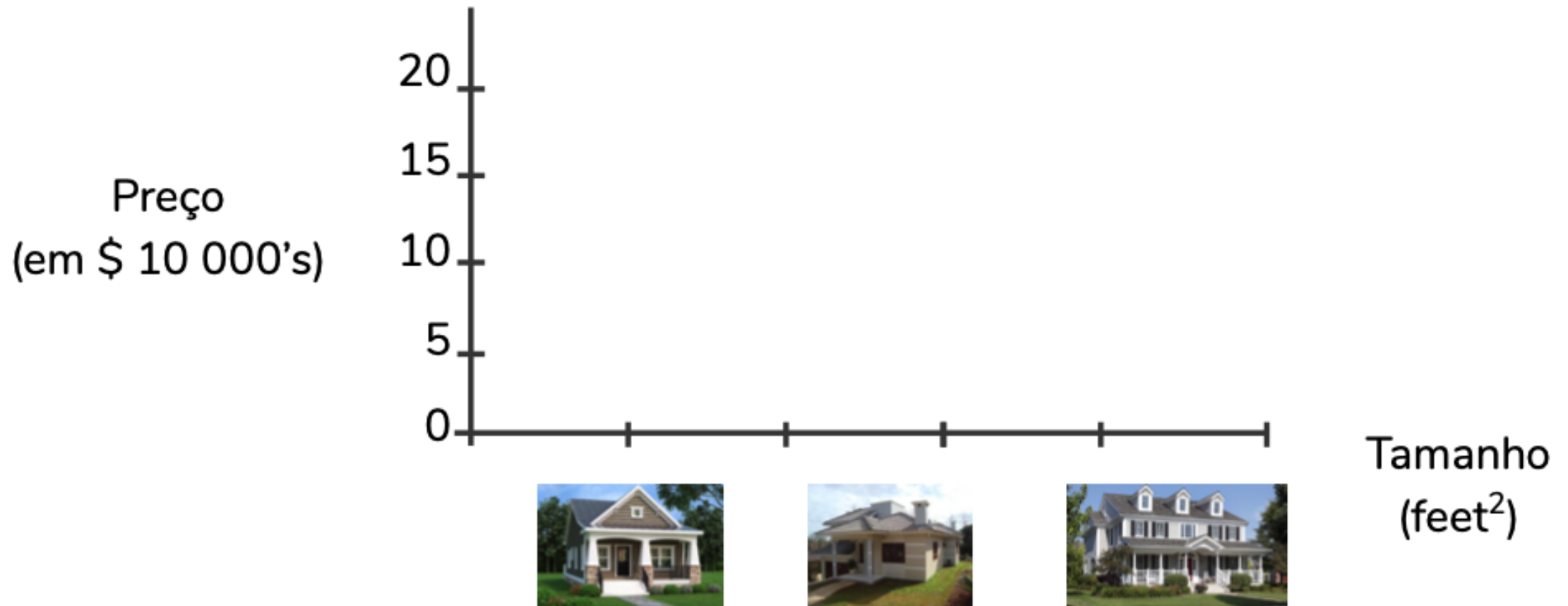
\$ 160 000

Predição de preço de imóveis (Regressão)

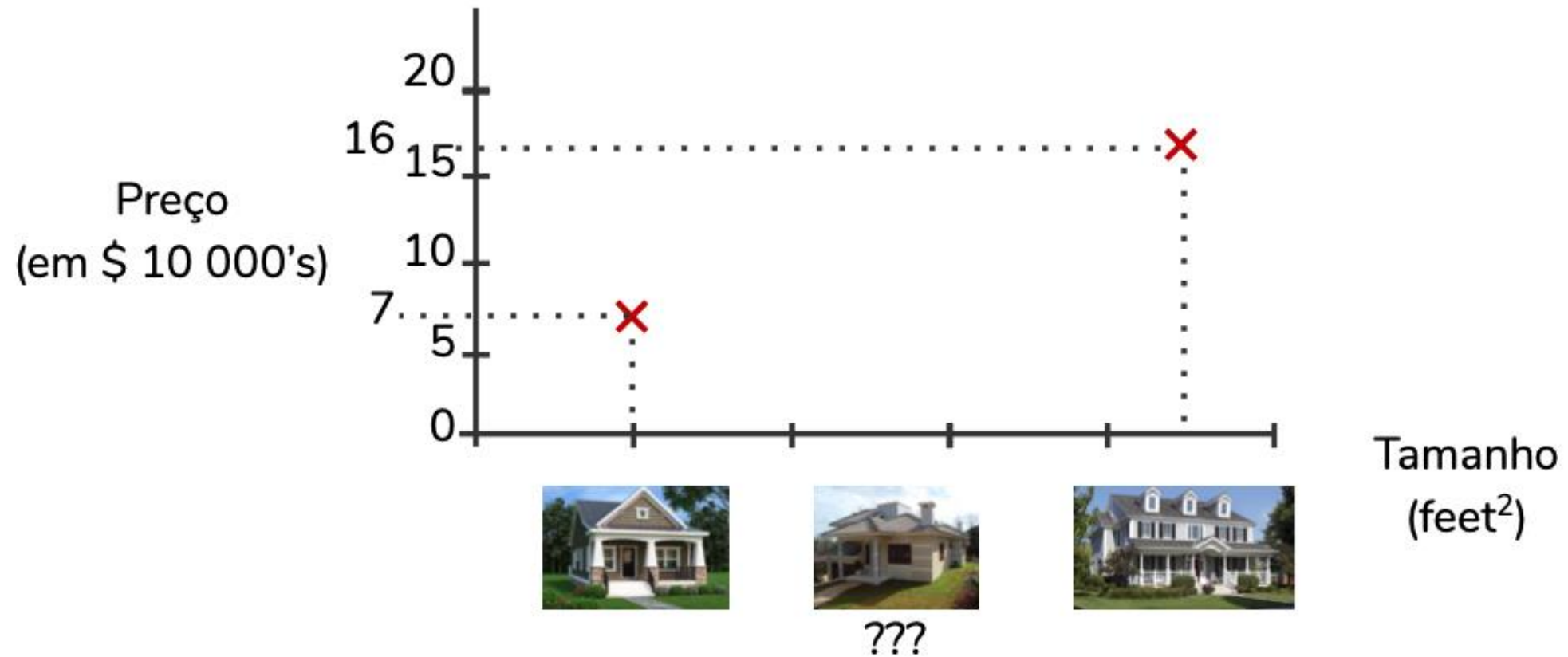


???

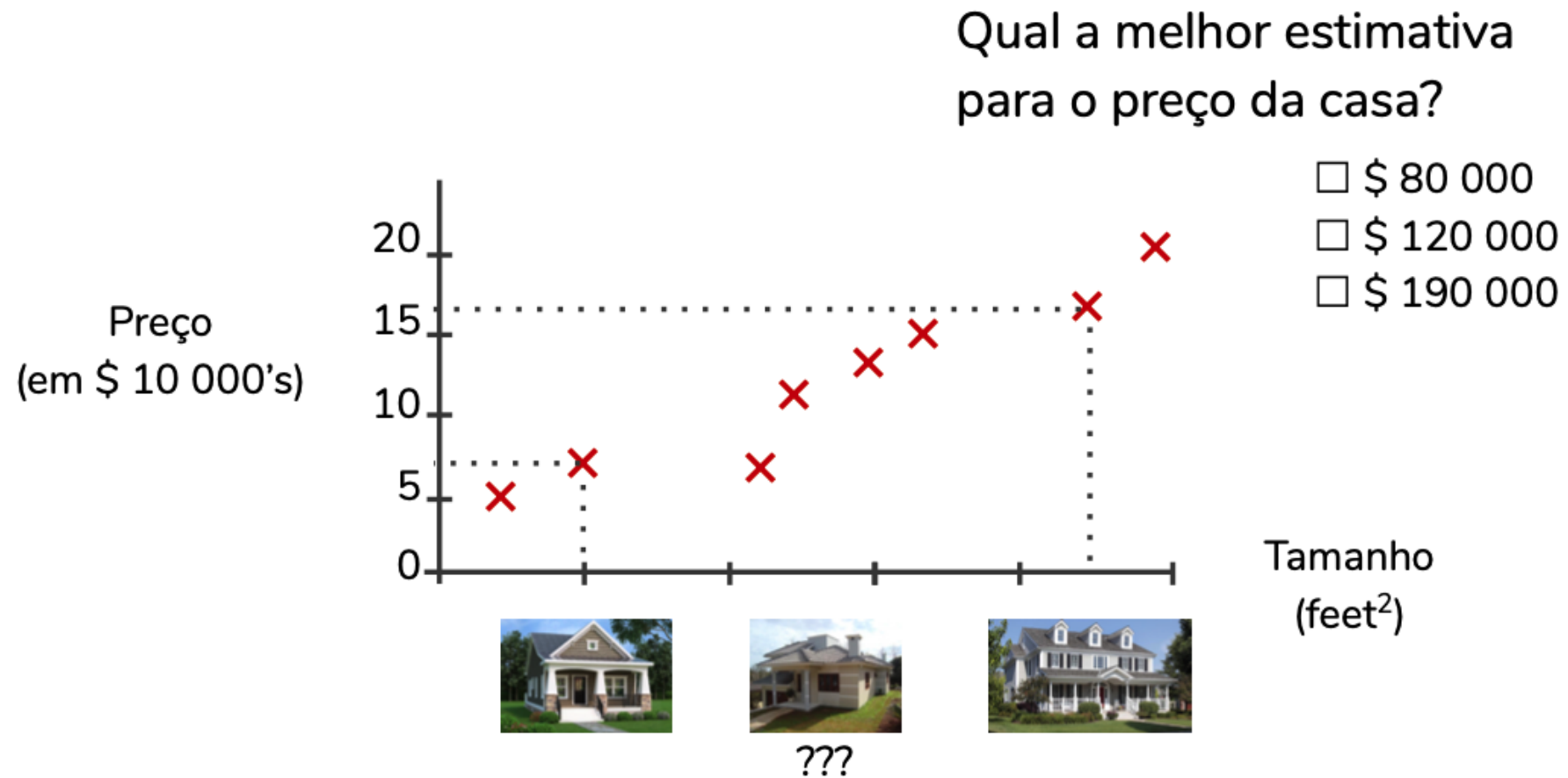
Predição de preço de imóveis (Regressão)



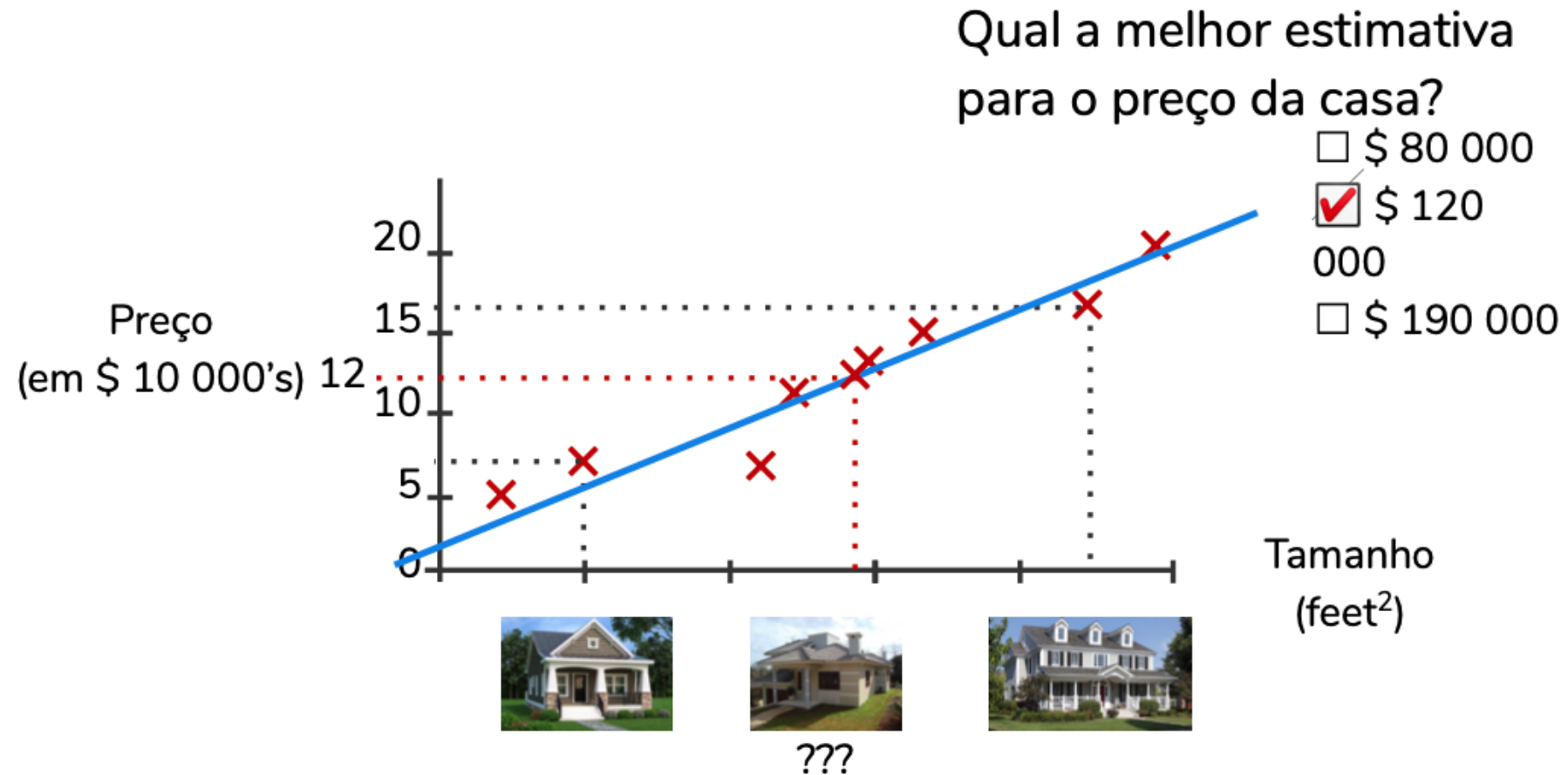
Predição de preço de imóveis (Regressão)



Predição de preço de imóveis (Regressão)



Predição de preço de imóveis (Regressão)

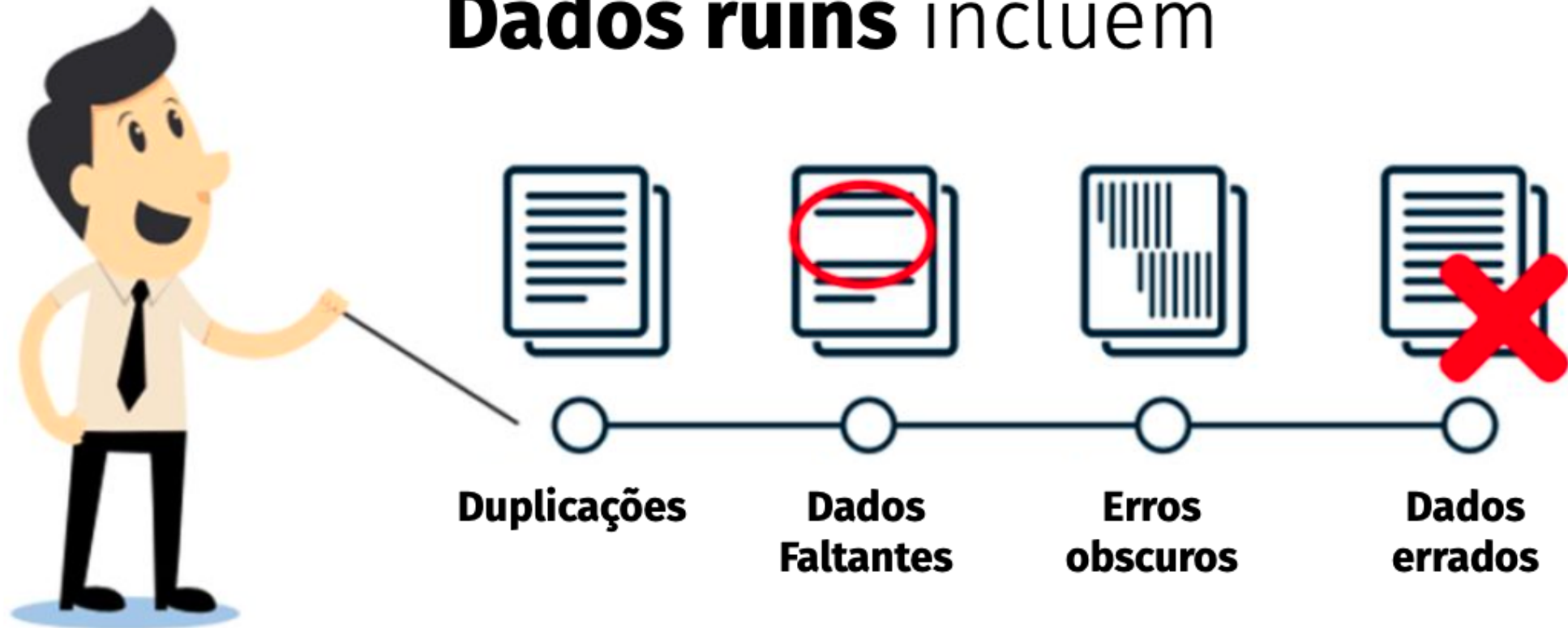


Principais desafios

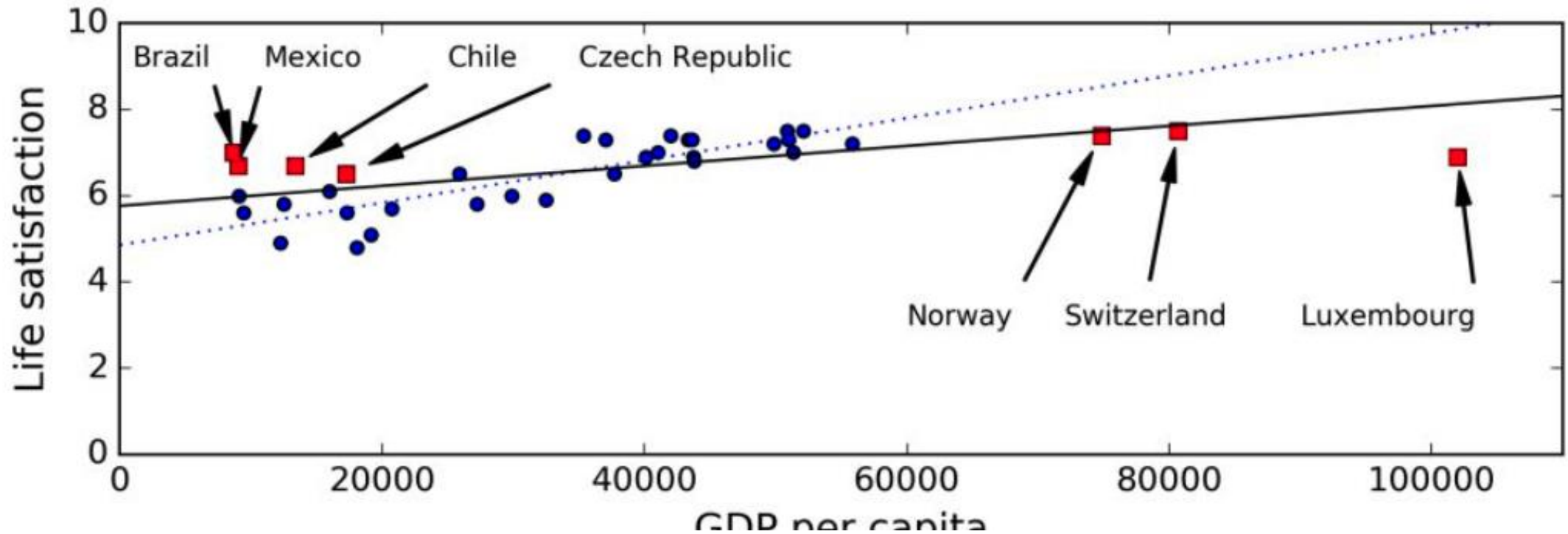
- “bad Data”
 - Quantidade insuficiente de dados de treinamento
 - Dados de treinamento não representativos
 - Dados de baixa qualidade
 - Features irrelevantes
- Bad algorithm
 - Modelo muito especializado (overfitting)
 - Modelo muito genérico (underfitting)
 - Modelo preconceituosos e injustos(unfair)

Dados ruins

Dados ruins incluem



Dados não representativos



Para generalizar bem, é essencial que os dados de treinamento representem os novos casos para os quais você deseja generalizar.

Dados com baixa qualidade

Obviamente, se os seus dados de treinamento estão cheios de erros, outliers e ruído, o sistema terá dificuldade em detectar os padrões presentes e, provavelmente, não irá ter um bom desempenho

Features(características) irrelevantes

Uma parte crítica para o sucesso de um projeto de Aprendizado de Máquina é obter um bom conjunto de características (features) para se realizar o treinamento

Isso é chamado de **engenharia de características** e envolve:

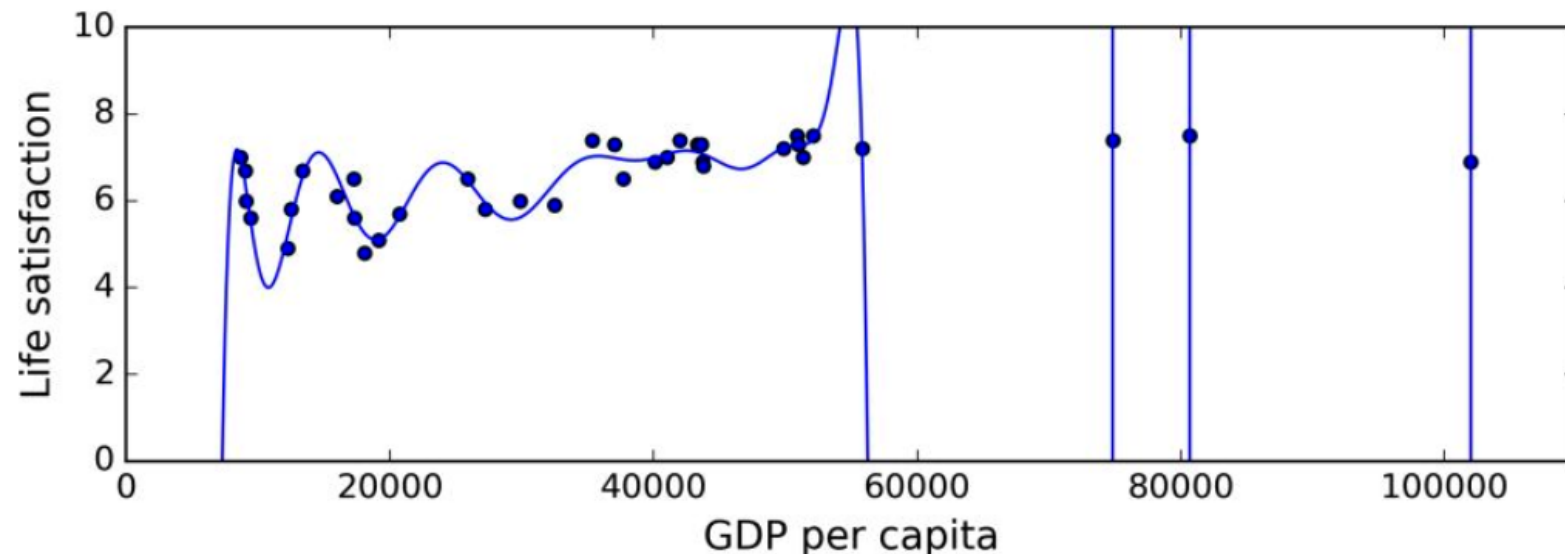
- Seleção de características: o processo de selecionar as features mais úteis entre as existentes para realizar o Treinamento
- Extração de características: combiner features existentes para gerar features mais úteis/discriminativas

Principais desafios

- “bad Data”
 - Quantidade insuficiente de dados de treinamento
 - Dados de treinamento não representativos
 - Dados de baixa qualidade
 - Features irrelevantes
- Bad algorithm
 - Modelo muito especializado (overfitting)
 - Modelo muito genérico (underfitting)
 - Modelo preconceituosos e injustos(unfair)

Overfitting

- Generalizar ao extremo é algo que nós humanos fazemos com bastante frequência, e infelizmente as máquinas podem cair na mesma armadilha se não formos cuidadosos
- Overfitting significa que o modelo tem um bom desempenho nos dados de treinamento, mas não generaliza.



Overfitting

Overfitting acontece quando o modelo é **muito complexo** em relação à quantidade e ao ruído dos dados de treinamento

As possíveis opções para lidar com overfit são:

- simplificar o modelo selecionando um com menos parâmetros, reduzindo o número de atributos nos dados de treinamento ou limitando o modelo,
- obter mais dados de treinamento,
- reduzir o ruído nos dados de treinamento

Underfitting

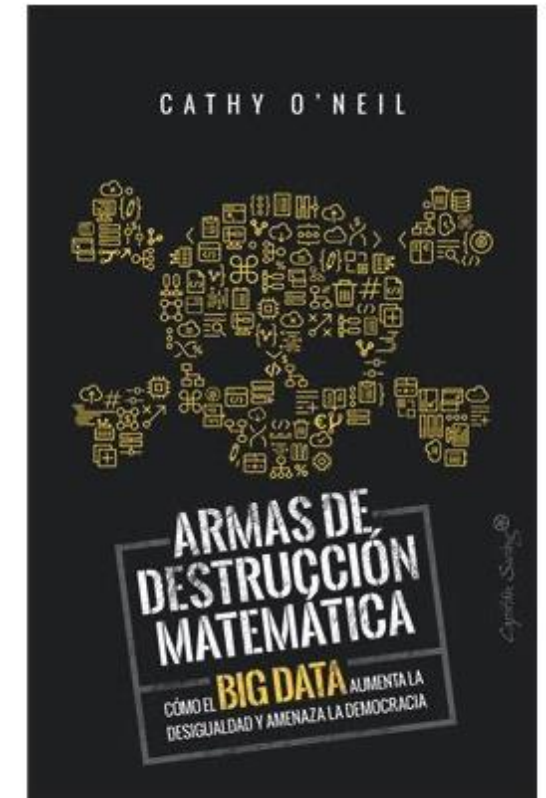
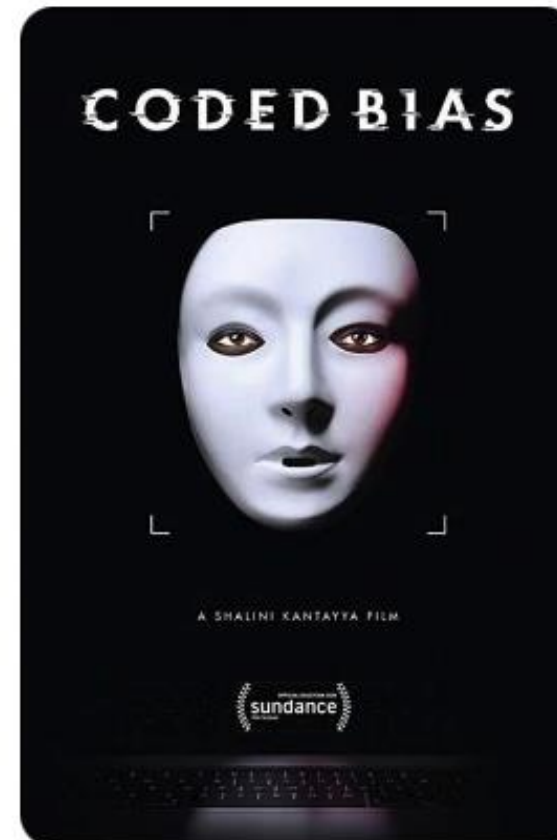
Underfitting é o contrário de overfitting: ocorre quando o modelo é **muito simples** para aprender a estrutura intrínseca dos dados

As principais opções para lidar com underfit são:

- selecionar um modelo mais poderoso, com mais parâmetros,
- fornecer features melhores para o algoritmo de aprendizado (engenharia de características),
- não limitar o aprendizado do modelo.

Modelos preconceituosos

- Devemos desenvolver algoritmos que sejam
 - Justos
 - Explicáveis
 - Auditáveis
 - transparentes



Exemplo de regressão linear

```
# Gerando dados simulados
set.seed(123)
horas_estudo <- runif(30, 1, 10) # gera numeros aleatorios entre 1h e 10h
nota_prova <- 5 + 2 * horas_estudo + rnorm(30, mean = 0, sd = 2) # relação linear com ruído

# Colocando em um data frame
dados <- data.frame(horas_estudo, nota_prova)
print(dados)

# Visualizando os dados
plot(dados$horas_estudo, dados$nota_prova,
     main = "Relação entre Horas de Estudo e Nota",
     xlab = "Horas de Estudo", ylab = "Nota na Prova",
     pch = 19, col = "blue")
```


Exemplo de regressão linear

```
# Ajustando ( criando) o modelo de regressão linear
modelo <- lm(nota_prova ~ horas_estudo, data = dados)
summary(modelo)
```

```
# Visualizando os dados
plot(dados$horas_estudo, dados$nota_prova,
     main = "Relação entre Horas de Estudo e Nota",
     xlab = "Horas de Estudo", ylab = "Nota na Prova",
     pch = 19, col = "blue")
# Adicionando a reta de regressão ao gráfico
abline(modelo, col = "red", lwd = 2)
```

Exemplo de regressão linear

```
# Fazendo uma previsão
novas_horas <- data.frame(horas_estudo = c(2, 5, 8))
previsoes <- predict(modelo, novas_horas) #usa o modelo para criar as previsões com base no modelo
print(cbind(novas_horas, previsoes))
```

Variáveis

Variável Independente (ou explicativa, preditora)

- É a variável que **controlamos ou observamos** para explicar ou prever outra
- Representa a **causa** ou o fator que influencia a outra variável
- Fica do lado direito da fórmula do modelo

◆ **Exemplo:** horas_estudo

Quanto mais horas a pessoa estuda, maior tende a ser sua nota.

Variável Dependente (ou resposta, regressanda)

- É a variável que **queremos prever ou explicar**
- Representa o **efeito** ou o resultado influenciado pela variável independente.
- Fica do lado esquerdo da fórmula do modelo.

◆ **Exemplo:** nota_prova

A nota depende das horas de estudo.

Lm()-Linear Model – Modelo linear

- A função `lm()` ajusta modelos lineares, como **regressão linear simples** ou **múltipla**, usando o método dos **mínimos quadrados**
- Sintaxe:
 - `lm(formula, data, subset, weights, na.action, method = "qr", ...)`

Argumento	Descrição
formula	A equação do modelo. Ex: $y \sim x1 + x2$ significa "y em função de x1 e x2"
data	Data frame onde as variáveis estão definidas
subset	Subconjunto dos dados a ser usado
weights	Pesos para os valores das observações
na.action	Como tratar valores ausentes (NA)
method	Método usado para calcular (geralmente "qr" por padrão)

Retorno:

Um objeto da classe "lm" que contém:

- Coeficientes do modelo (intercepto e inclinações)
- Resíduos
- Valores ajustados (fitted)
- Informações de diagnóstico

Lm - exemplo

```
dados <- data.frame(x = 1:10, y = c(2.1, 2.9, 4.2, 4.8, 5.9, 7.1, 8.0, 9.3, 10.1, 11.2))  
modelo <- lm(y ~ x, data = dados)  
summary(modelo)
```

```
plot(dados$x, dados$y)  
abline(modelo, col="red", lwd=2)
```


Predict() – previsões com modelos

- A função `predict()` **usa um modelo treinado** (como o de `lm()`) para prever novos valores da variável dependente com base em novas entradas
- Sintaxe:
 - `predict(object, newdata, interval = "none", level = 0.95, ...)`

Argumento	Descrição
<code>object</code>	Um modelo criado com <code>lm()</code> (ou outro método)
<code>newdata</code>	Data frame com novos valores das variáveis independentes
<code>interval</code>	Tipo de intervalo: "none" (padrão), "confidence" ou "prediction"
<code>level</code>	Nível de confiança (padrão é 95%)

Retorno:

Um vetor (ou data frame, se `interval` for usado) com os valores previstos

Predict() – previsões com modelos

```
# 1. Criando os dados manualmente
x <- c(1, 2, 3)           # variável independente
y <- c(2, 4, 5)           # variável dependente

dados <- data.frame(x, y)
```

```
# 2. Ajustando o modelo de regressão linear
modelo <- lm(y ~ x, data = dados)
```

```
# 3. Mostrando a equação do modelo
summary(modelo)
```

```
# 4. Prevendo o valor de y para um novo x
novo_valor <- data.frame(x = 4)
previsao <- predict(modelo, newdata = novo_valor)
```

```
# 5. Exibindo a previsão
print(paste("A previsão de y para x = 4 é:", round(previsao, 2)))
```

Regressão no PowerBi

- Crie a tabela (inserir dados):

horas_estudo, nota_prova

1, 2.5

2, 3.8

3, 5.1

4, 6.4

5, 7.2

6, 8.5

7, 9.1

8, 10.3

9, 10.8

10, 12.0



	Horas de est...	nota_prova	+
1	1	2,5	
2	2	3,8	
3	3	5,1	
4	4	6,4	
5	5	7,2	
6	6	8,5	
7	7	9,1	
8	8	10,3	
9	9	10,8	
10	10	12	
+			

Regressão no Power BI

Criar o gráfico de dispersão com regressão

1. Vá até a aba **Visualizações**.
2. Selecione o visual de “**Gráfico de dispersão**”.
3. Configure:
4. **Eixo X:** horas_estudo
5. **Eixo Y:** nota_prova
6. Clique com o botão direito no gráfico e selecione:
7. **Análise → Linha de tendência → Adicionar**

- Isso adiciona a **reta de regressão linear** sobre os pontos.



Exercício

- Na loja do Power BI, procure outros visuais que calculam regressão

Equação da regressão linear

- Medidas auxiliares:

$Média_X = AVERAGE(Tabela[Hora\ de\ estudo])$

$Média_Y = AVERAGE(Tabela[nota_prova])$

- Covariância:

$Cov_XY =$

$VAR\ media_x = AVERAGE(Tabela[Horas\ de\ estudo])$

$VAR\ media_y = AVERAGE(Tabela[nota_prova])$

$RETURN$

$SUMX($

$Tabela,$

$(Tabela[Horas\ de\ estudo] - media_x) * (Tabela[nota_prova] - media_y)$

$)$

Interpretação:

Valor da Covariância	Interpretação
> 0	X e Y variam na mesma direção
< 0	X e Y variam em direções opostas
≈ 0	Pouca ou nenhuma relação linear entre X e Y

Equação da regressão linear

- Variância:

ar_X =

```
VAR media_x = AVERAGE(Tabela[Horas de estudo])  
RETURN  
SUMX(  
    Tabela,  
    POWER(Tabela[Horas de estudo] - media_x, 2)  
• )
```

Interpretação:

- **Variância = 0** → todos os valores são iguais à média (sem variação).
- **Variância maior** → dados mais espalhados (maior dispersão).
- Quanto maior a variância, **menos consistentes** são os valores da variável.

Equação da regressão linear

- Inclinação (coeficiente angular)

$$\text{Inclinação} = \text{DIVIDE}([\text{Cov_XY}], [\text{Var_X}])$$

- Intercepto (coeficiente linear)

$$\text{Intercepto} = [\text{Média_Y}] - [\text{Inclinação}] * [\text{Média_X}]$$

- Exibindo em um cartão

Equação =

"nota_prova = " &

ROUND([Intercepto], 2) & " + " &

ROUND([Inclinação], 2) & " * Hora de estudo"

Predição

- Predição da nota com base na hora de estudo

Nota_Prevista =

[Intercepto] + [Inclinação] * SELECTEDVALUE(Tabela[Hora de estudo])

R^2 -R quadrado ou Coeficiente de determinação

O R^2 mede o quanto da **variabilidade da variável dependente (Y)** é explicada pelo modelo de regressão

Interpretação:

- $R^2 = 1 \rightarrow$ o modelo explica **100% da variação** dos dados (ajuste perfeito).
- $R^2 = 0 \rightarrow$ o modelo **não explica nada** da variação.
- $0 < R^2 < 1 \rightarrow$ o modelo explica parcialmente a variabilidade dos dados.
- Se nota_prova varia muito entre os alunos, e o modelo com Hora de estudo explica bem essas variações, o R^2 será alto

Calculando R^2

- Média da variável dependente

```
Média_Y = AVERAGE(Tabela[nota_prova])
```

- Soma total dos quadrados

```
SStot =
```

```
VAR media_y = [Média_Y]
```

```
RETURN
```

```
SUMX (
```

```
    Tabela,
```

```
    POWER(Tabela[nota_prova] - media_y, 2)
```

```
)
```

Calculando R^2

- Soma dos quadrados dos resíduos

```
SSres =  
SUMX(  
    Tabela,  
    POWER(Tabela[nota_prova] - [Nota_Prevista], 2)  
)
```

- Calculando o R^2

```
R2 = 1 - DIVIDE([SSres], [SStot])
```

Exercício: Treinamento físico x Desempenho

Treinamento (horas)	Desempenho (pontos)
1.5	15
2.0	18
2.5	20
3.0	22
3.5	24
4.0	27
4.5	28
5.0	30
5.5	32
6.0	33
6.5	35
7.0	37
7.5	39
8.0	40
8.5	42
9.0	43
9.5	44
10.0	45
10.5	47
11.0	48

- Inserir essa tabela no **Power BI**
- Criar as mesmas **medidas DAX**:
 - Média de X e Y
 - Covariância
 - Variância
 - Inclinação e Intercepto
 - Nota prevista
 - R^2 e Correlação
- Plotar os **gráficos comparando valores reais e previstos, além das medidas em cartões**