

# Regressão

---

Advanced Institute for Artificial Intelligence

<https://advancedinstitute.ai>

## Regressão: O que é?

- Tenta **prever** valores numéricos diretamente **a partir de atributos** de um novo exemplo.

## Exemplos

- **Prever** a temperatura de amanhã **a partir** das condições atmosféricas.
- **Estimar** o preço de uma casa **a partir** de seu tamanho.

# Mais Formalmente.....

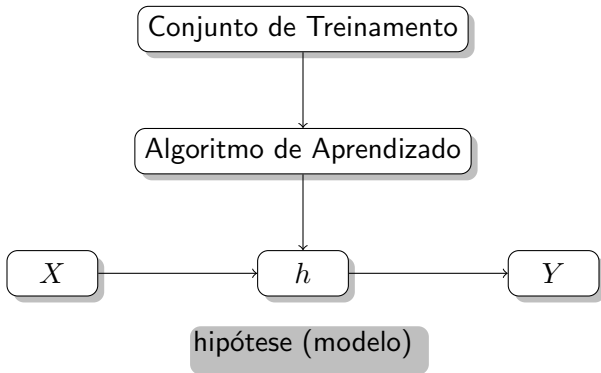
## Definição do problema

- Prever uma variável quantitativa  $Y \in \mathbb{R}$  (*resposta*)
- A partir de variáveis preditoras  $X_1, \dots, X_n \in \mathbb{R}$
- **Objetivo:** Encontrar o modelo  $h$ :

$$Y = h(X_1, \dots, X_n)$$

## Estratégia:

- Utilizar um conjunto de exemplos (*dataset*) onde a resposta correta é conhecida para "aprender" um modelo.

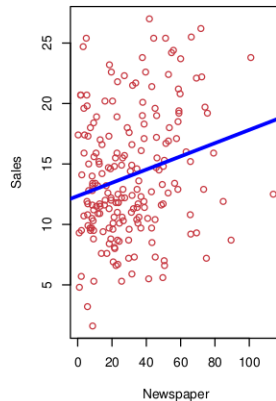
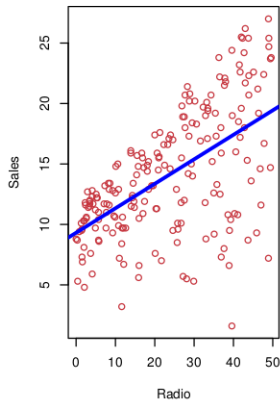
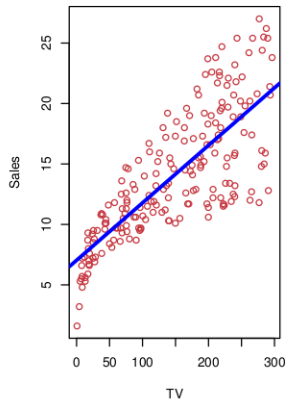


## Idealmente, o algoritmo para aprender o modelo deve:

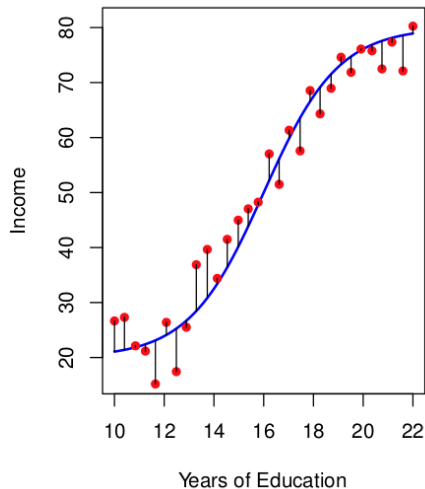
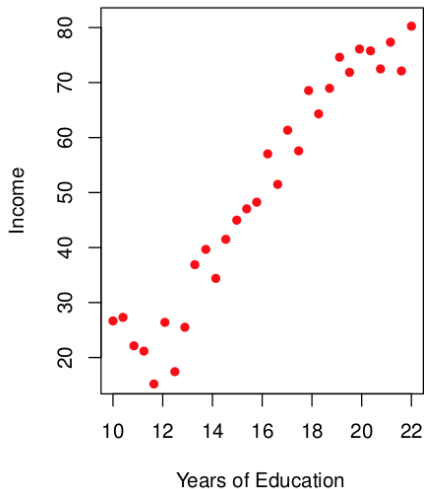
- Ser capaz de reconstruir o fenômeno modelado com maior precisão possível
- Requerer o mínimo possível de dados para o aprendizado
- Representar o modelo da maneira mais simples o possível (Navalha de Occam)

- Não há uma "resposta correta" para todos os problemas
- Existem muitos tipos de modelos (modelos lineares, árvores, redes neurais, etc.)

## Volume de vendas em função da verba de publicidade em diferentes meios



## Renda em função da escolaridade

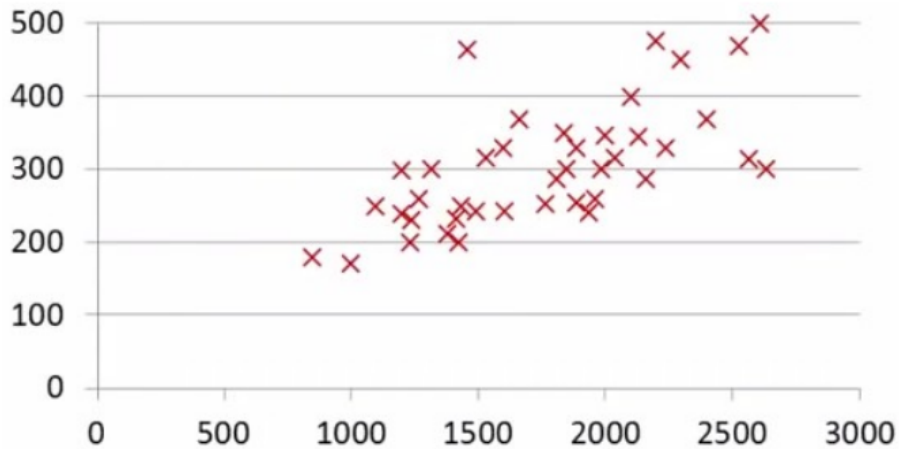




## O Conjunto de treinamento pode ser visualizado como uma tabela

Tamanho em pé <sup>2</sup>	Preço (\$) em 1000's
2104	460
1416	232
1534	315
852	178
...	...

Table: Preço de habitação por tamanho em Portland (OR)



# Regressão Linear Univariada

---

## Caso: Apenas um atributo

### Hipótese

- variável de resposta tem uma relação linear com os atributos.

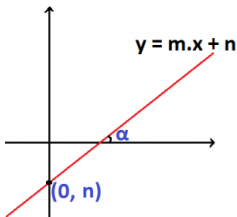
$$Y = \theta_0 + \theta_1 + \epsilon$$

*h* é representado como uma reta:

$$h(\theta) = \theta_0 + \theta_1 x$$

## Equação da Reta:

$$y = mx + n$$



- $m$  = coeficiente angular
- $n$  = coeficiente linear

$$y = mx + n$$

$$h(\theta) = \theta_0 + \theta_1 x$$

**Objetivo: Achar melhor reta ( $\theta$ ) de acordo com os dados de treinamento**

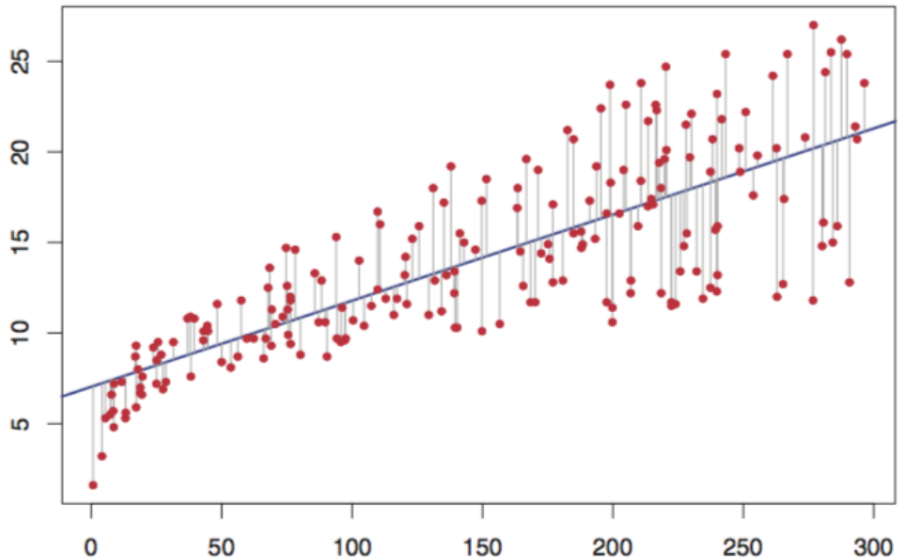
## E o que seria a melhor reta?

- Encontrar a reta  $h$  que passe o mais próximo possível de todos os pontos

### Resíduo

- Diferença entre o valor  $y$  real e a estimativa  $\hat{y} = h_{\theta}(x)$

$$\epsilon = y^i - \hat{y}$$





- Uma maneira de calcular  $\theta_0$  e  $\theta_1$  é se basear na soma do quadrado dos resíduos (RSS - *residual sum os squares*)

$$J(\theta) = \sum_{i=1}^n \epsilon_i^2$$

- Uma maneira de calcular  $\theta_0$  e  $\theta_1$  é se basear na soma do quadrado dos resíduos (RSS - *residual sum os squares*)

$$J(\theta) = \sum_{i=1}^n \epsilon_i^2$$

$$J(\theta) = \sum_{i=1}^n (y^i - \hat{y})^2$$

- Uma maneira de calcular  $\theta_0$  e  $\theta_1$  é se basear na soma do quadrado dos resíduos (RSS - *residual sum os squares*)

$$J(\theta) = \sum_{i=1}^n \epsilon_i^2$$

$$J(\theta) = \sum_{i=1}^n (y^i - \hat{y})^2$$

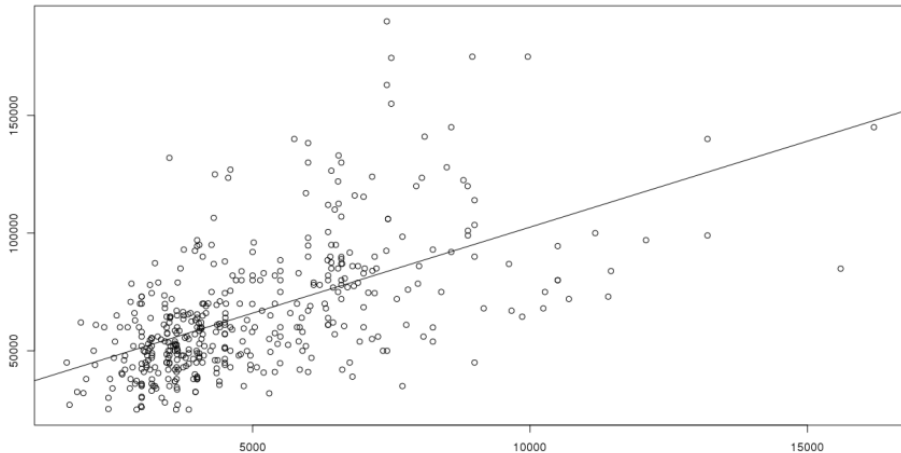
$$J(\theta) = \sum_{i=1}^n (y^i - (\theta_0 + \theta_1 x^i))^2$$

$\theta_0$  e  $\theta_1$  devem ser escolhidos de modo a minimizar o RSS. Solução:  
Método dos mínimos quadrados (Least Squares)

$$\bar{x} = \frac{1}{n} \sum_j x^j \quad \bar{y} = \frac{1}{n} \sum_j y^j$$

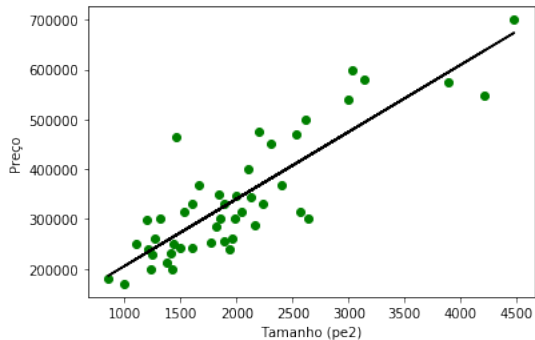
$$\theta_1 = \frac{\sum_j (x^j - \bar{x})(y^j - \bar{y})}{\sum_i (x^j - \bar{x})^2} \quad \theta_0 = \bar{y} - \theta_1 \bar{x}$$

# Como saber se o resultado foi bom?

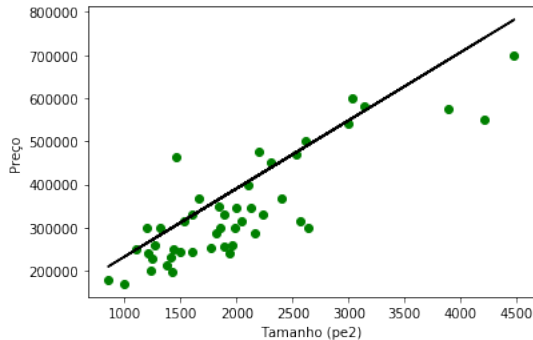


# Avaliação da Regressão

## O próprio RSS pode ser utilizado



(a) Modelo 1:  $RSS = 1.93 \times 10^{11}$

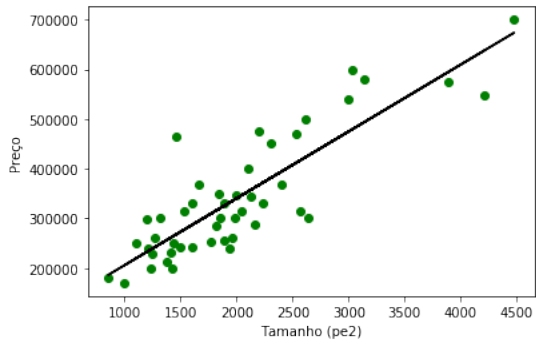


(b) Modelo 1:  $RSS = 3.28 \times 10^{11}$

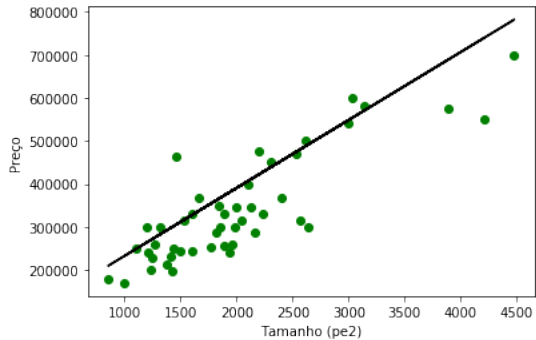
- Mede a proporção da variabilidade de  $Y$  que pode ser explicada por  $X$ .

$$TSS = \sum_j (y^j - \bar{y})^2 \quad RSS = \sum_j (y^j - \hat{y})^2$$

$$R^2 = \frac{TSS - RSS}{TSS}$$



(c) Modelo 1:  $R^2 = 0.63$



(d) Modelo 1:  $R^2 = 0.54$



# Regressão Linear Multivariada

---

- Na maior parte dos problemas práticos, utilizar apenas um atributo **não é o suficiente** para estimar a resposta
- Neste caso, a Regressão Linear deve estimar um Hiperplano como modelo  $h$ .

Hipótese

$$Y = h_{\theta}(X) = \theta_0 + \theta_1 X_1 + \cdots + \theta_n X_n$$

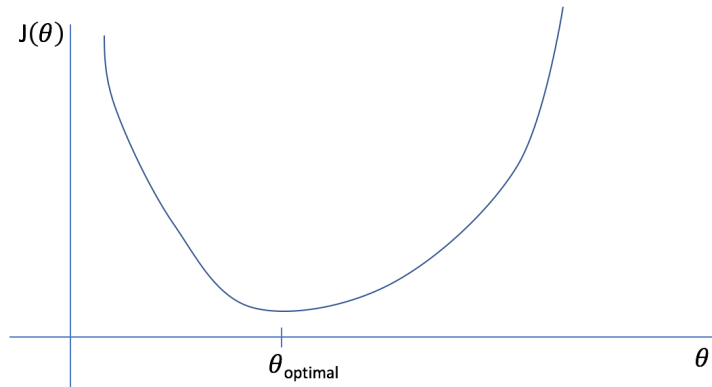
Tamanho em pé <sup>2</sup>	número de quartos	Preço (\$) em 1000's
2104	3	460
1416	3	232
1534	3	315
852	2	178
...		...

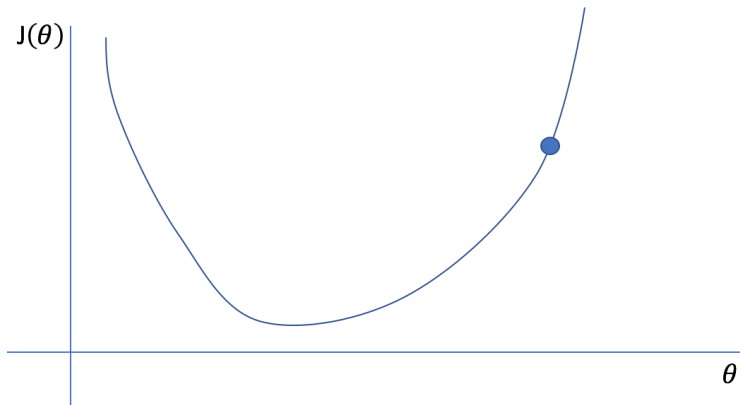
- Assim como no caso univariado, valores de  $\theta$  devem ser escolhidos baseados no conjunto de treinamento
- O método dos mínimos quadrados também funciona para o caso multivariado
- Outra possibilidade de realizar o aprendizado é através do método de **Descida do Gradiente** (*Gradient Descent*)

- Partindo da função de custo de **Erro Quadrático Médio**

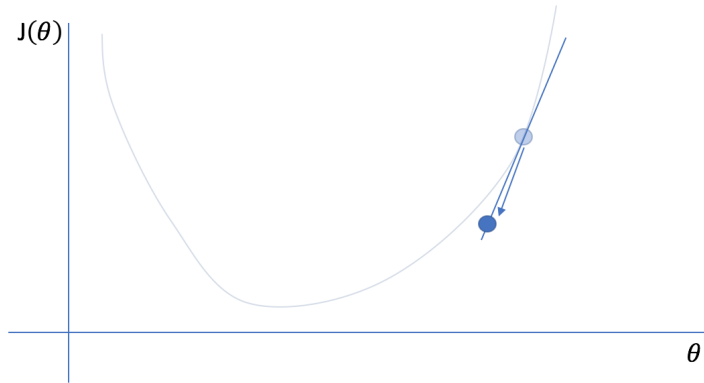
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^n (y^i - h_{\theta}(\mathbf{x}^i))^2$$

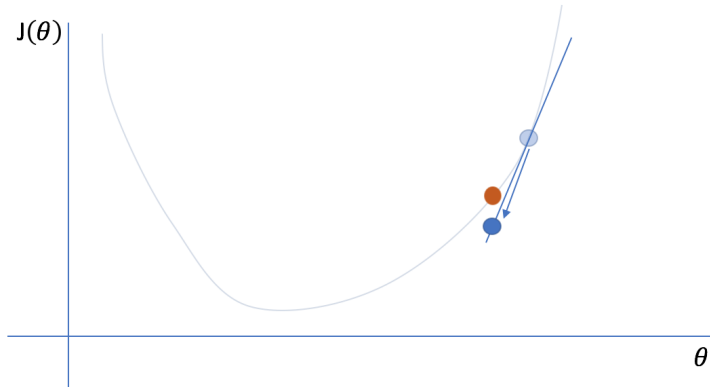
- Definir parâmetros  $\theta$  que minimizem  $J$

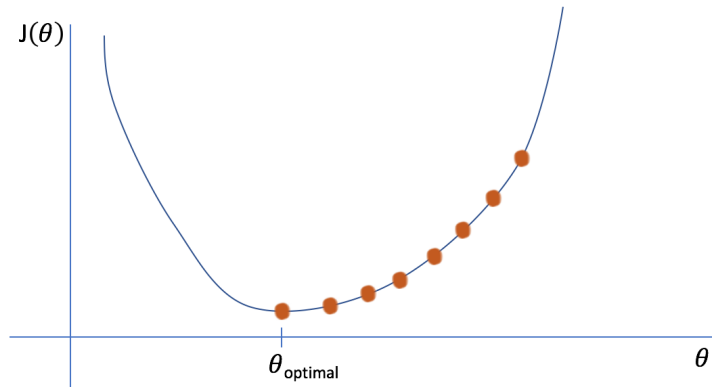






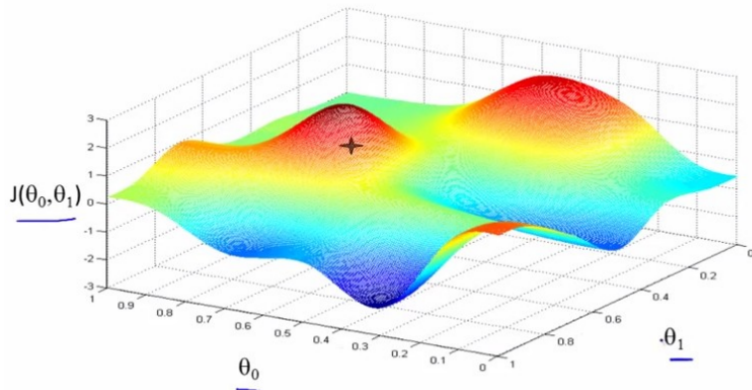


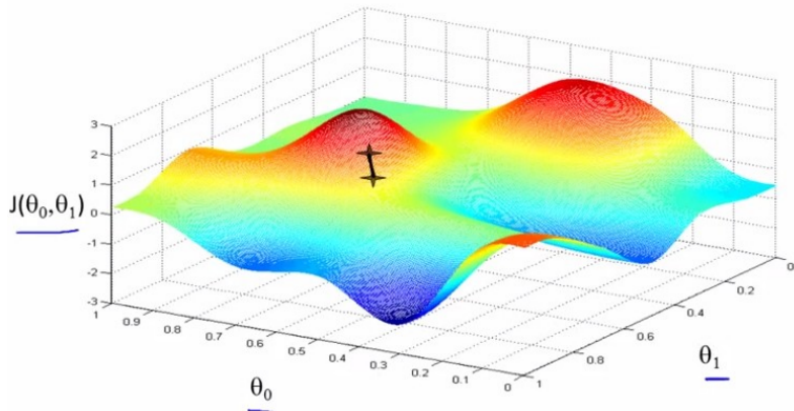


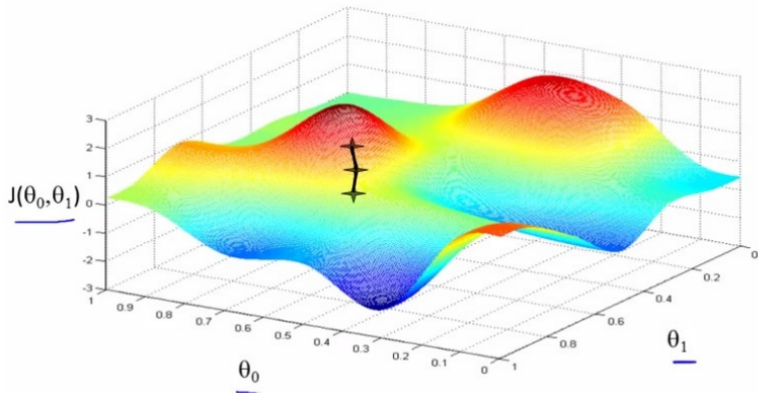


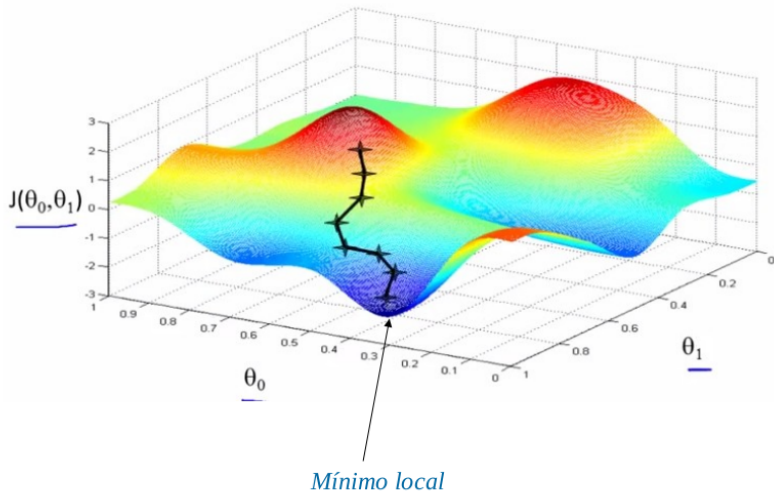
## Algoritmo Gradiente Descendente

- ❶ Iniciar  $\theta$  aleatoriamente
- ❷ Modificar valores de  $\theta$  (seguindo o gradiente), para reduzir  $J$  até que um valor mínimo seja atingido.

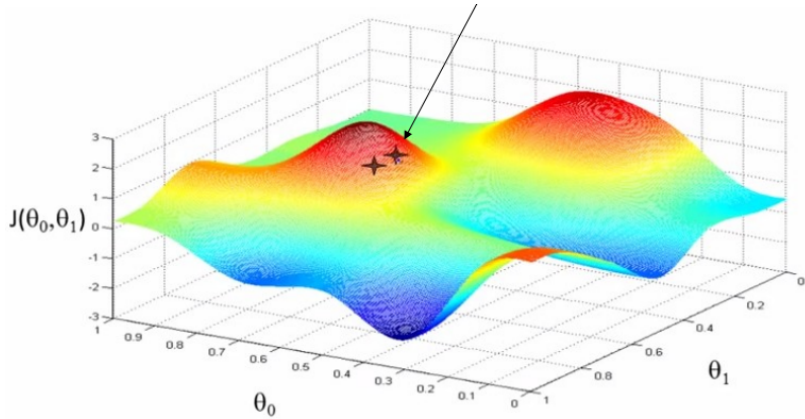


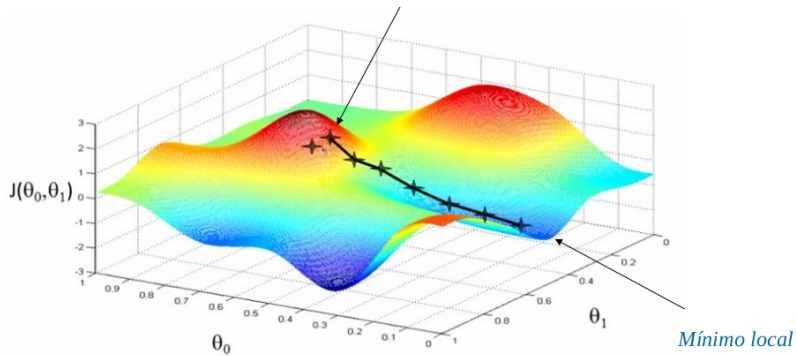












# Como computar a atualização?

- Repetir até a convergência, para cada parâmetro  $\theta$ :

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

- $\alpha$  é a taxa de aprendizado, que controla o "salto" na atualização dos parâmetros

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta)$$

$$i = 0 : \frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum h_{\theta}(\mathbf{x}^i) - y^i$$

$$i = 1 : \frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum (h_{\theta}(\mathbf{x}^i) - y^i) x_1^i$$

# Aplicando Descida do Gradiente

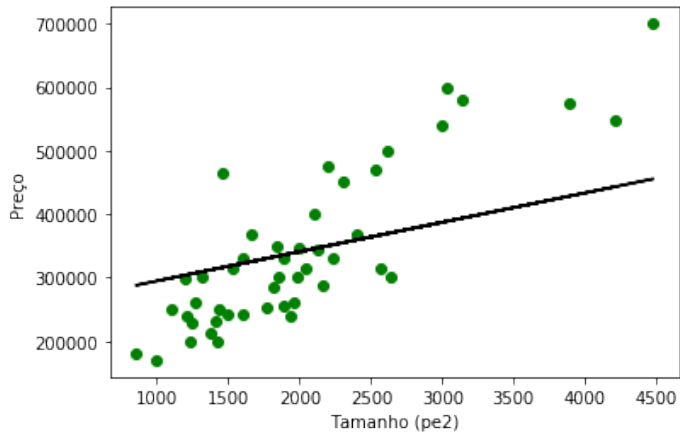


Figure: Repetindo **1** vez o treinamento.  $RSS = 4.2 \times 10^{12}$ ,  $R^2 = -5.76$

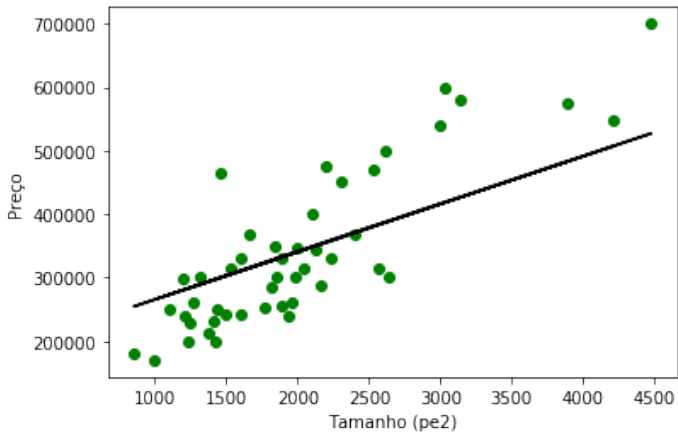


Figure: Repetindo 2 vezes o treinamento.  $RSS = 2.9 \times 10^{11}$ ,  $R^2 = -0.79$

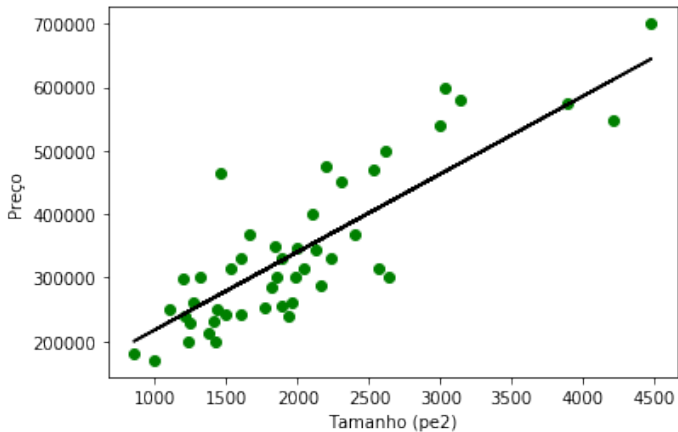


Figure: Repetindo **7** vezes o treinamento.  $RSS = 1.9 \times 10^{11}$ ,  $R^2 = 0.55$

## Vantagens

- Aprendizado eficiente
- Modelo simples de se visualizar e compreender

## Desvantagens

- Muitos problemas reais não são lineares



# Regressão por K Vizinhos mais Próximos

---

# Regressão KNN (K-Nearest Neighbours)

## Hipótese

- $Y$  pode ser estimada se baseando nos  $k$  exemplos mais próximos na base de treinamento.

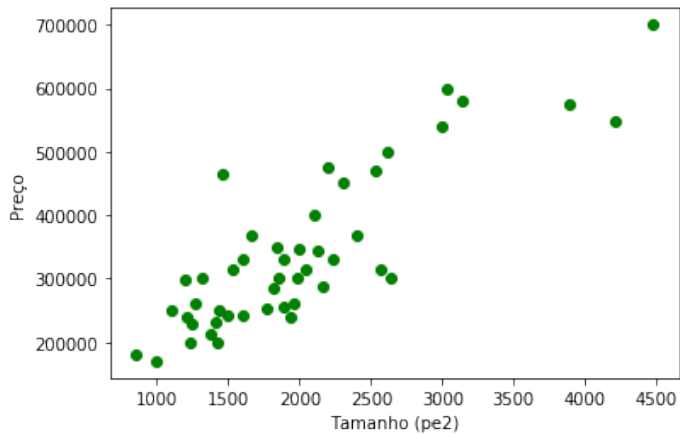
$$h_K(\mathbf{x}) = \sum_{j \in \mathcal{N}_K} w_j(\mathbf{x}, \mathbf{x}^j) y^j$$

- Onde  $K$  é o número de vizinhos,  $\mathcal{N}_K$  é o conjunto de amostras presentes na  $K$ -vizinhança e  $w_j$  é o peso de  $\mathbf{x}$  em relação a  $\mathbf{x}^j$

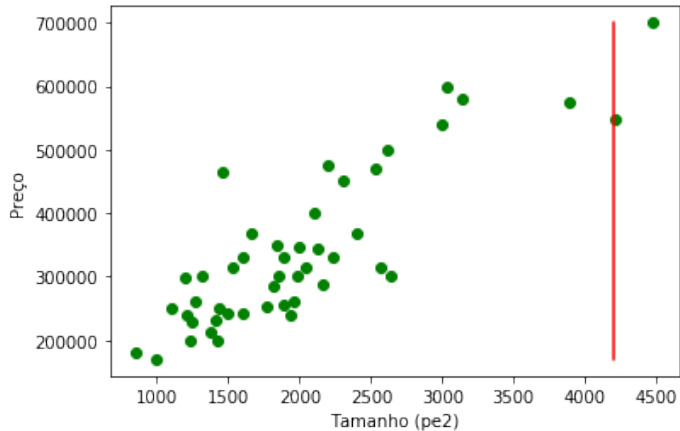
- Assumindo que todas as amostras têm o mesmo peso:

$$h_K(\mathbf{x}) = \frac{1}{K} \sum_{j \in \mathcal{N}_K} y^j$$

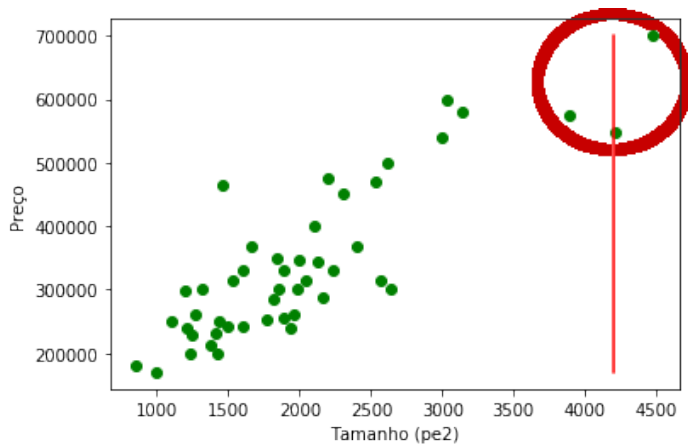
- Não é necessário um processo de treinamento



Imagine que queremos prever o preço de uma casa com tamanho = 4200

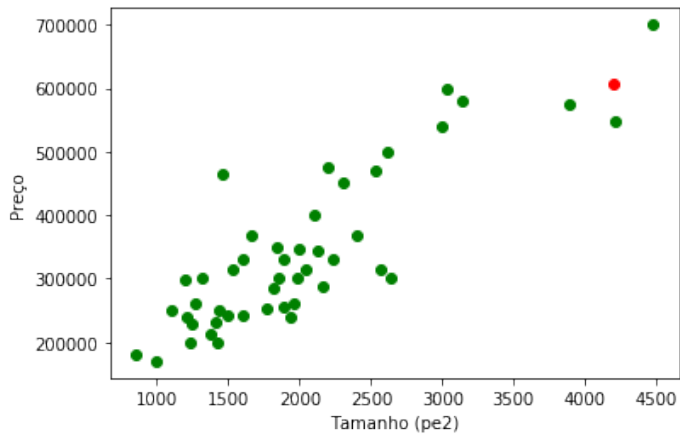


Se  $K = 3$ , devemos encontrar os 3 exemplos mais próximos do valor desejado



**O valor previsto é definido como a média destes valores**

$$Y = \frac{1}{3}(699900 + 573900 + 549000) = 607600$$





# Parâmetros Necessários para KNN

- ①  $K \rightarrow$  número de vizinhos
- ② Uma métrica de distância para encontrar os vizinhos "mais próximos"
- ③ Uma forma de definir o peso para cada exemplo

# Exemplos de modelos aprendidos com KNN

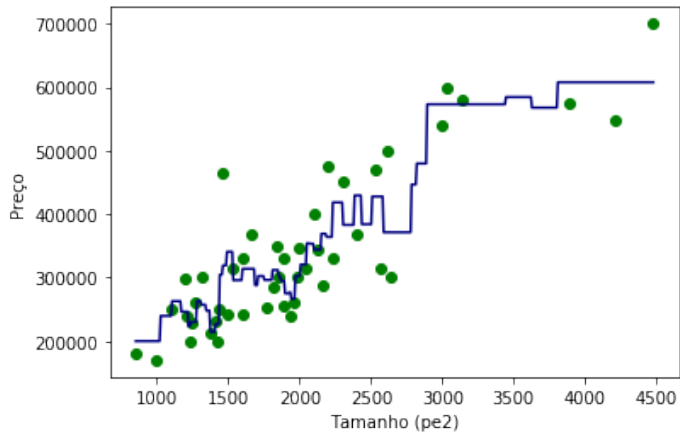


Figure:  $K = 3$

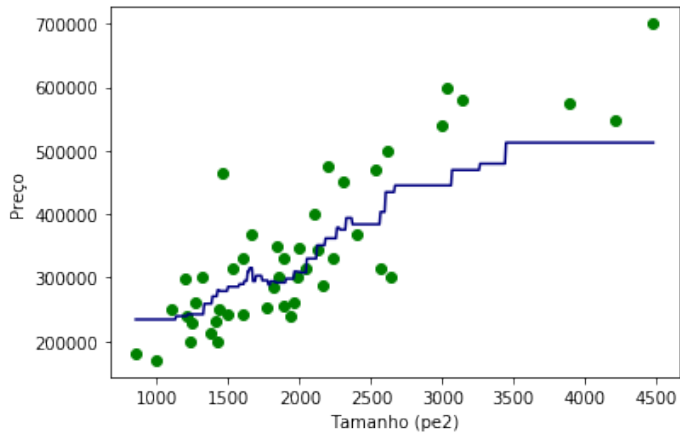


Figure:  $K = 10$

## Vantagens

- O modelo aprendido não precisa ser linear
- Não há fase de treinamento
- Poucos parâmetros a serem definidos

## Desvantagens

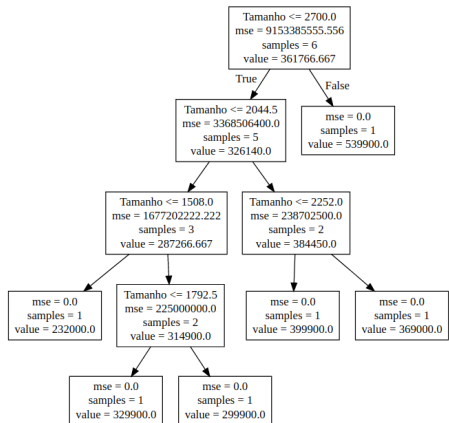
- Processo de inferência muito custoso
- Sensitividade a ruído e escala

# Árvore de Regressão

---

# Árvore de Regressão

- Aprende uma estrutura de árvore que "divide" o conjunto de treinamento de acordo com os valores dos atributos
- Resulta em um modelo fácil de **visualizar** e **interpretar**, muito utilizado em domínios em que uma solução "caixa-preta" é inaceitável



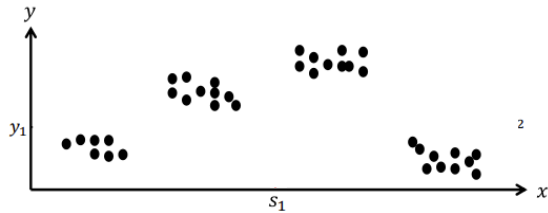
# Como aprender a árvore?

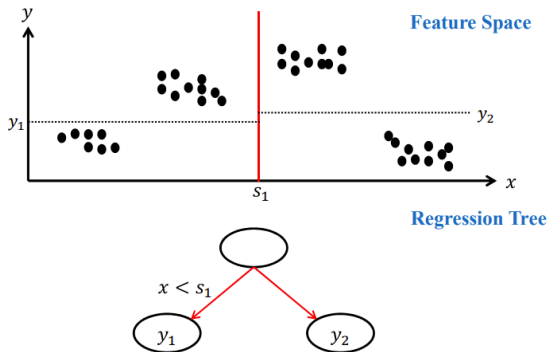
Repetir iterativamente até que um critério de parada seja atingido:

- ❶ Encontrar o melhor valor para particionar cada atributo
- ❷ Selecionar a melhor partição entre as definidas no passo anterior
- ❸ Particionar os dados de treinamento conforme escolhido no passo anterior

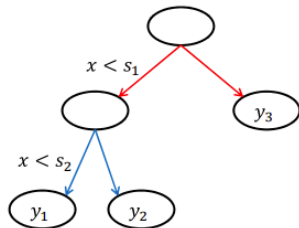
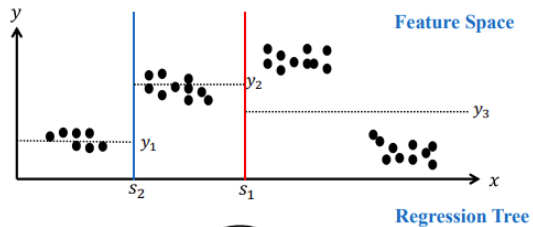


# Particionando a Árvore





A Predição para uma amostra é definida como uma média do valor  $y$  de todos os exemplos no conjunto de treinamento que caem naquela mesma partição.



## Vantagens

- Pouco afetada pela escala dos atributos
- Intuitiva e fácil de se compreender

## Desvantagens

- Pequenas alterações nos dados podem resultar em grandes alterações na árvore resultante
- Valores preditos na regressão não são muito precisos
- Processo de treinamento relativamente custoso