

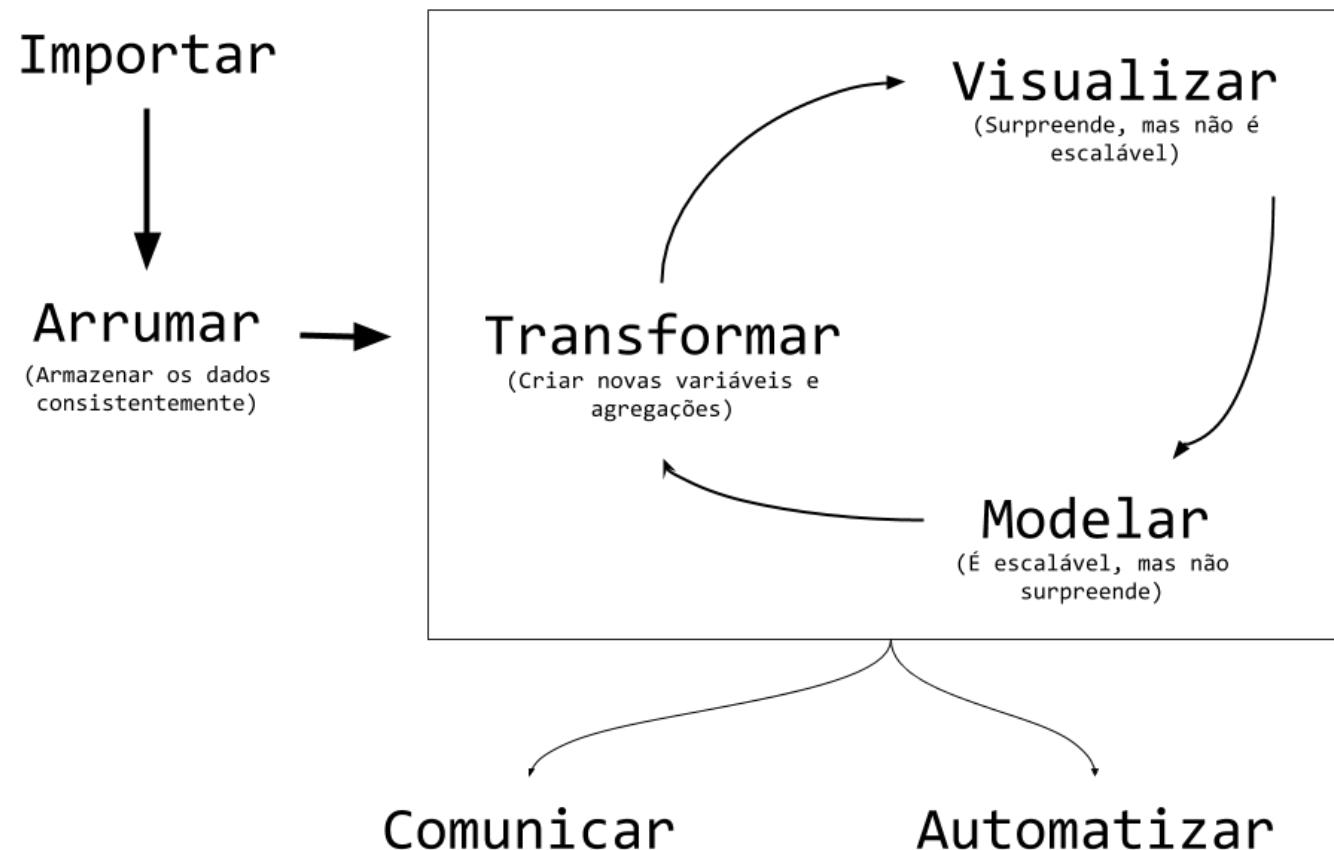
Introdução ao Machine Learning

Definição e Estratégias

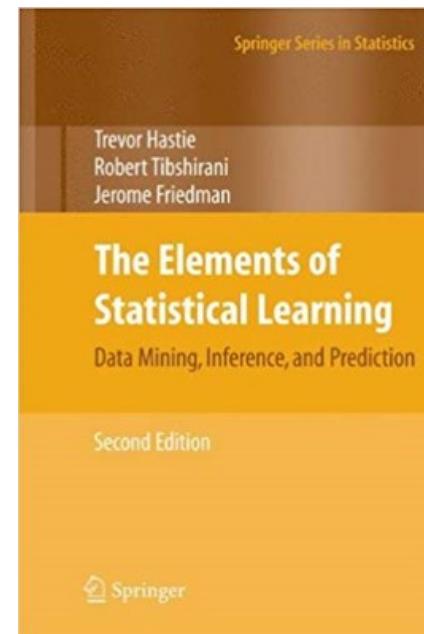
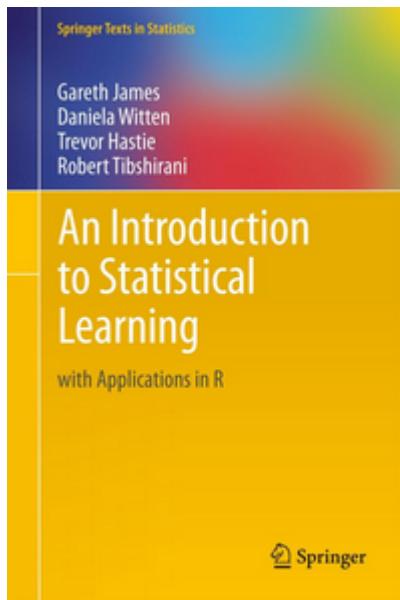


September de 2022

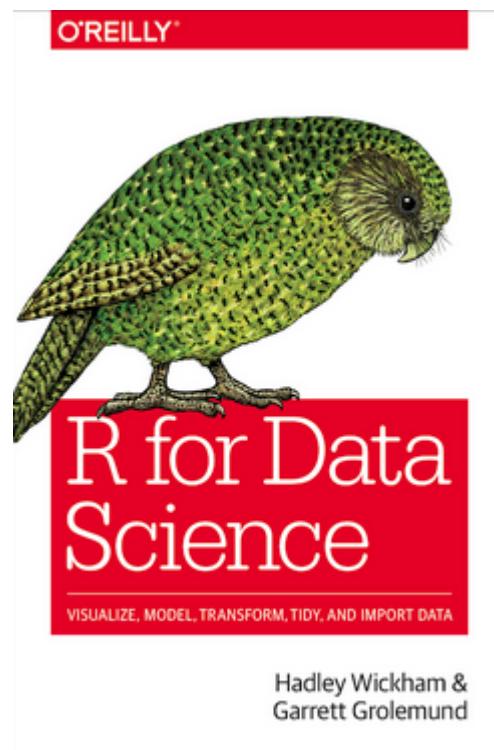
Ciência de dados



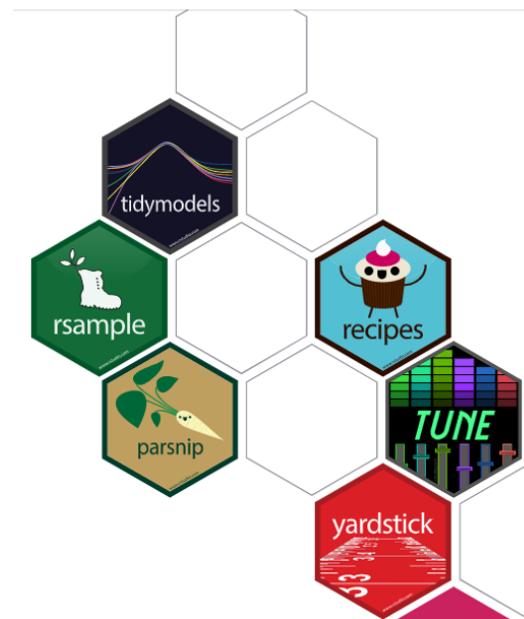
Referências



Referências



Tidymodels



Referências

- Feature Engineering and Selection: A Practical Approach for Predictive Models
- Aprendizado De Máquina
- Forecasting: Principles and Practice

Introdução

O que é Machine Learning?

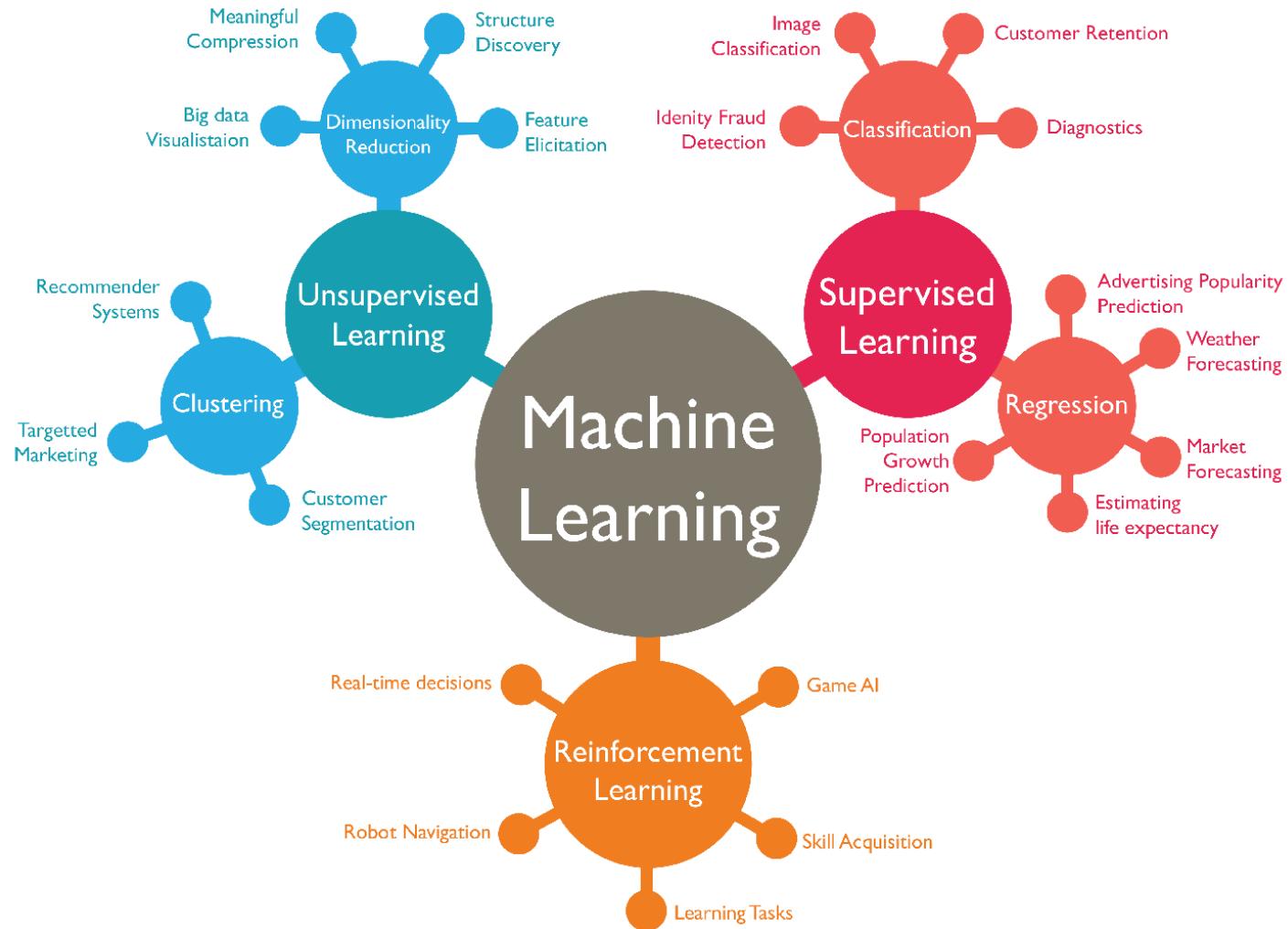
- Termo criado por Arthur Samuel, em 1959



- 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).

Exemplos

- Previsão de churn
- Previsão de inadimplência
- Previsão de demanda
- Previsão de preço
- Previsão meteorológica
- Diagnóstico em imagem médica
- Carro autônomo
- Projeção da taxa de desemprego
- Teste A/B
- Teste clínico
- Eficácia de vacinas
- Impactos de políticas públicas
- Impactos de campanha publicitária
- Curvas epidemiológicas
- Projeção do PIB
- ...

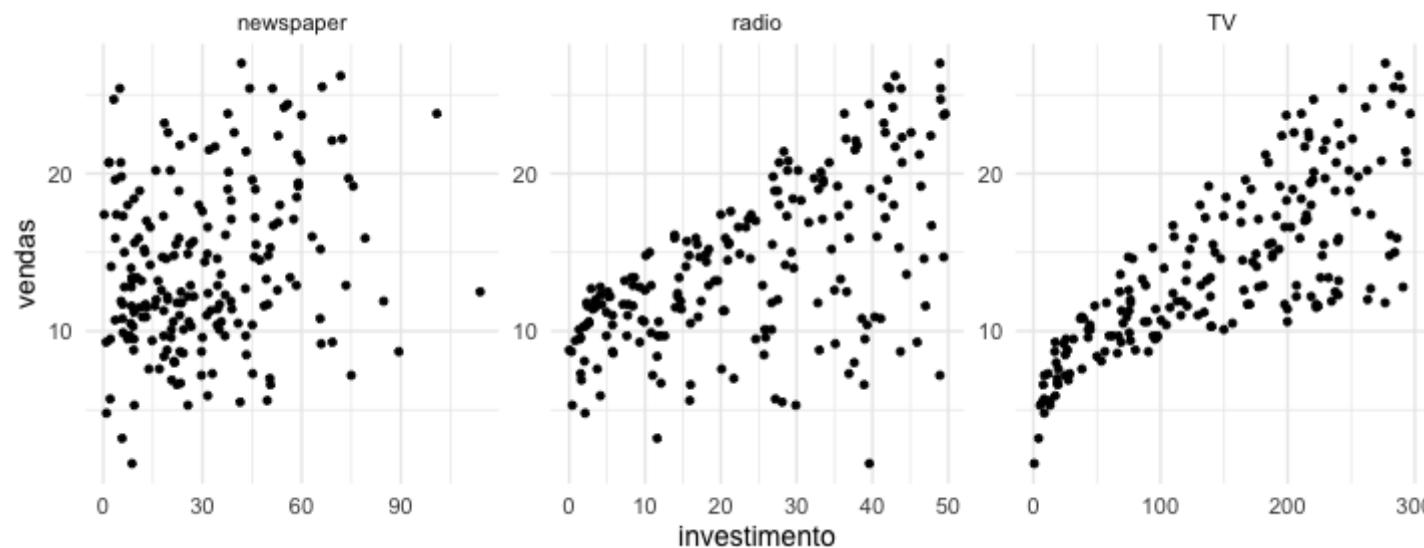


fonte: business2community

Motivação

Somos consultores e fomos contratados para dar conselhos para uma empresa aumentar as suas vendas.

Obtivemos o seguinte banco de dados

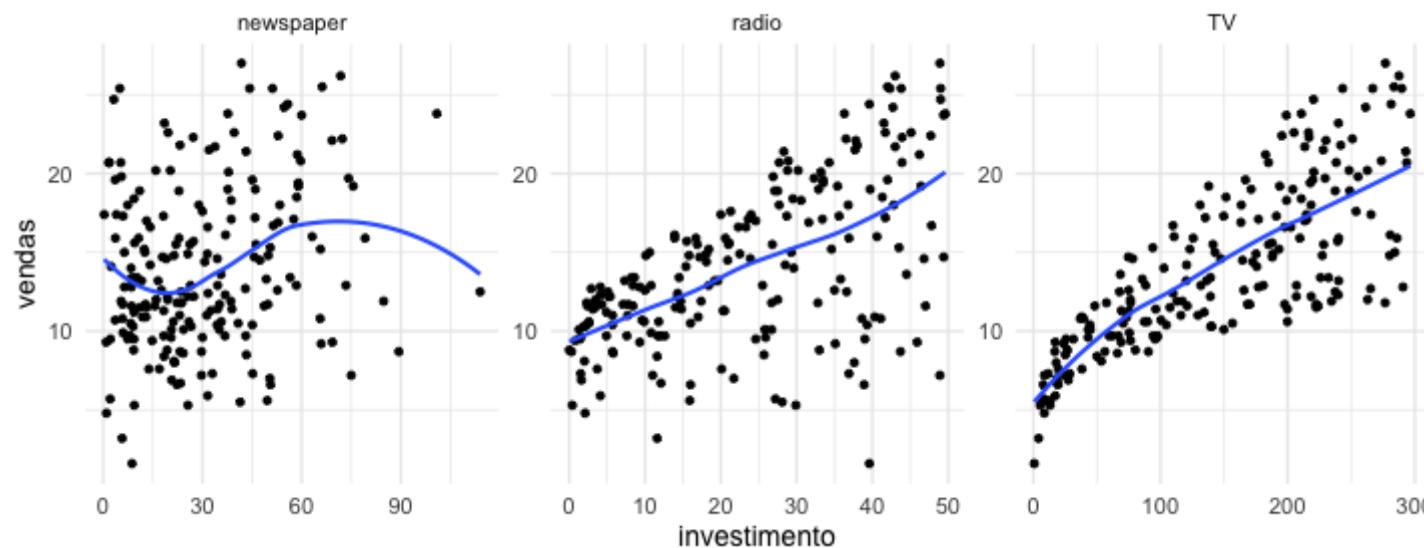


- PERGUNTA: Quantas vendas terão se eu investir X? Em qual mídia eu escolho alocar meu orçamento?

Motivação

Somos consultores e fomos contratados para dar conselhos para uma empresa aumentar as suas vendas.

Obtivemos o seguinte banco de dados

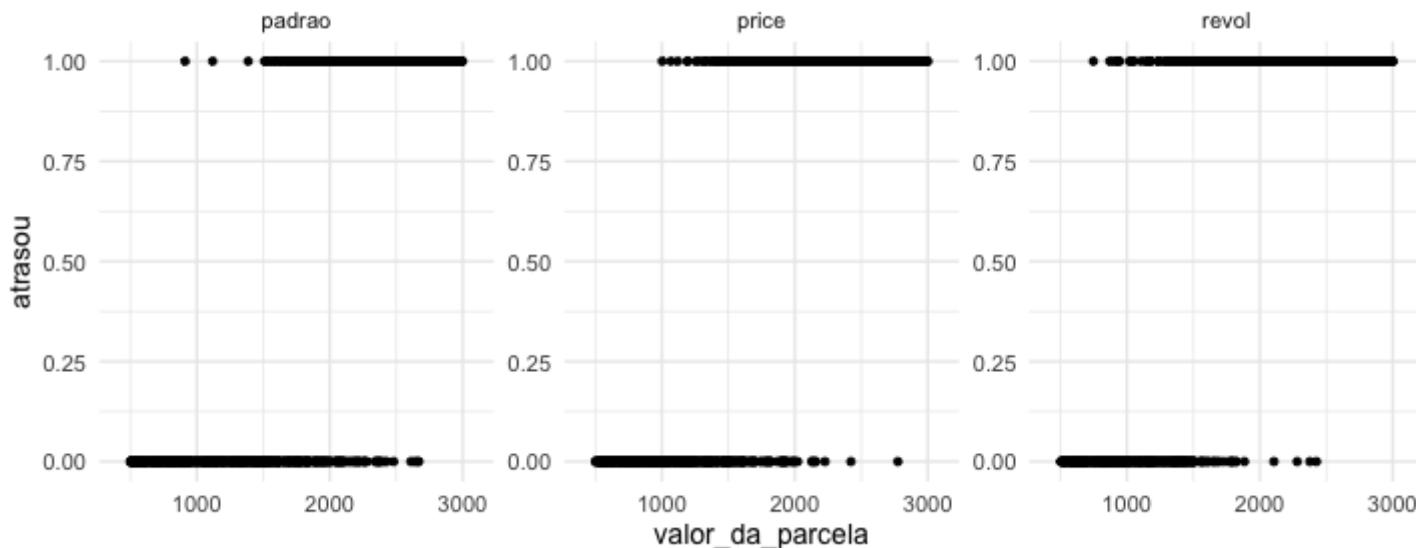


- PERGUNTA: Quantas vendas terão se eu investir X? Em qual mídia eu escolho alocar meu orçamento?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados

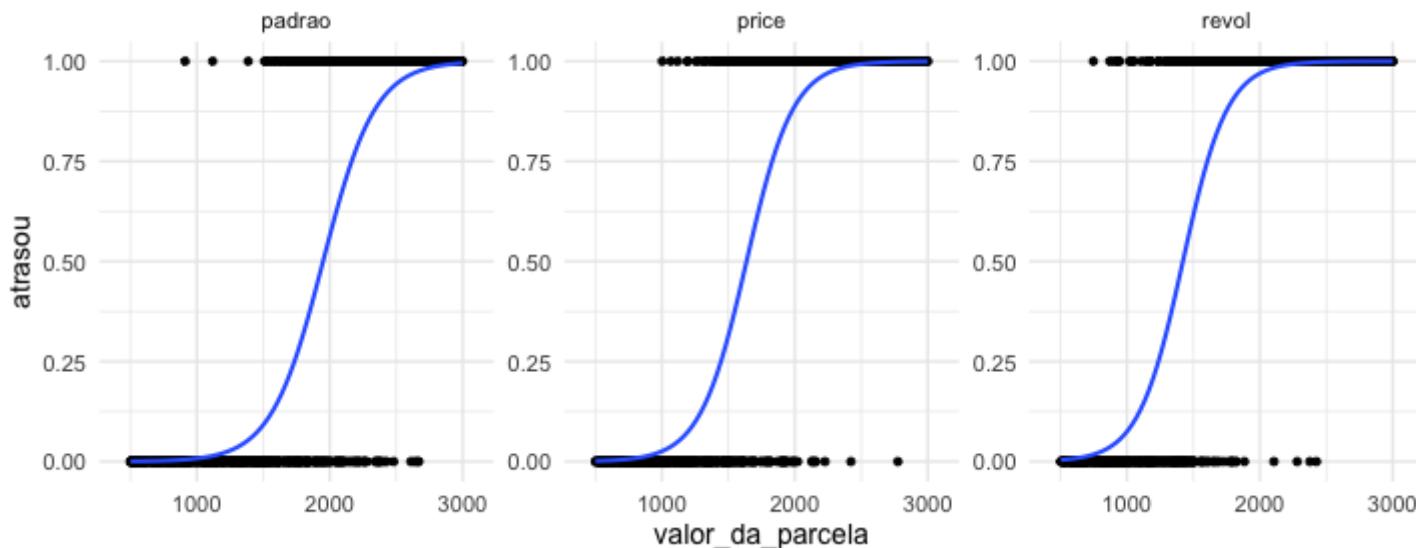


- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados

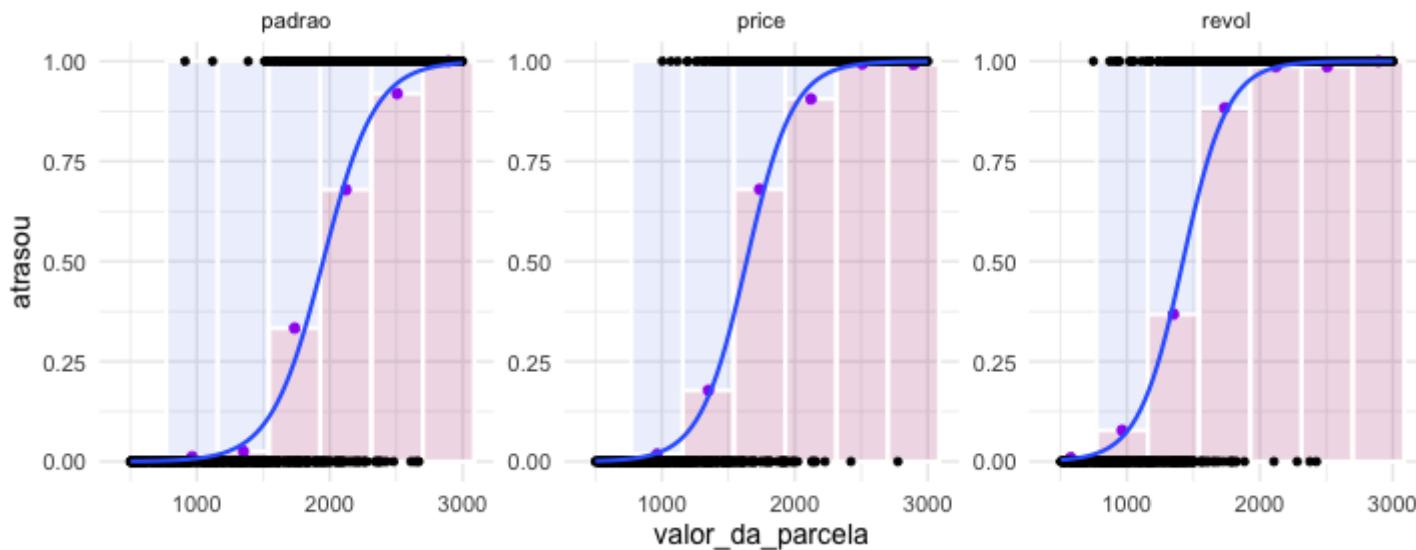


- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados

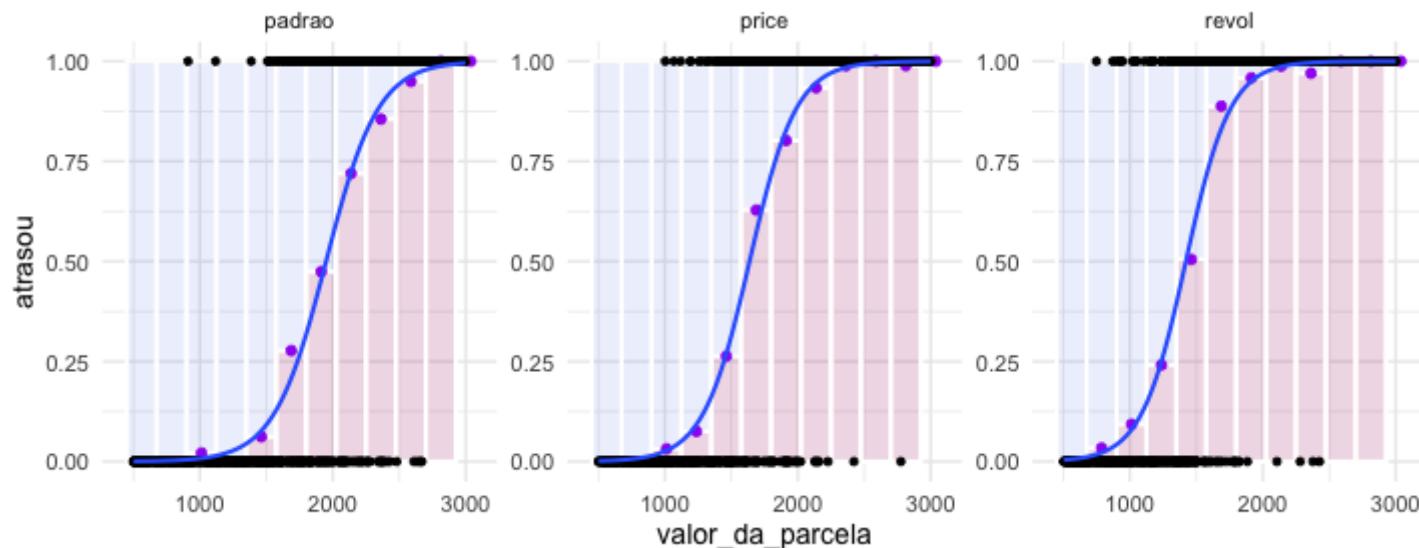


- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados

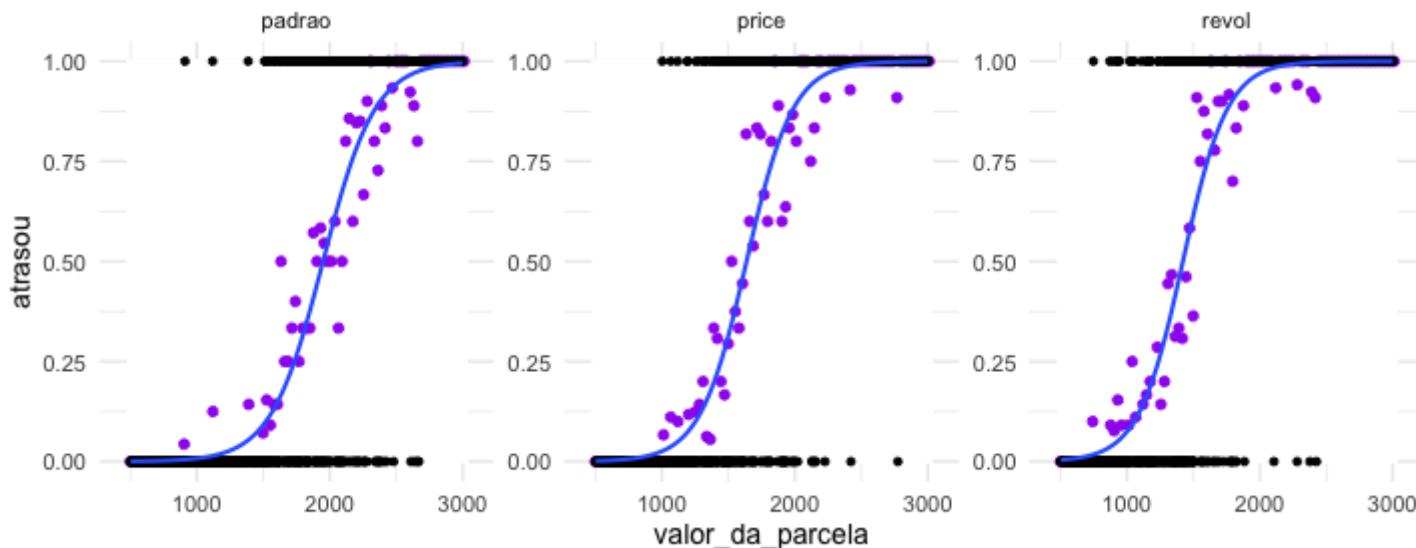


- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados

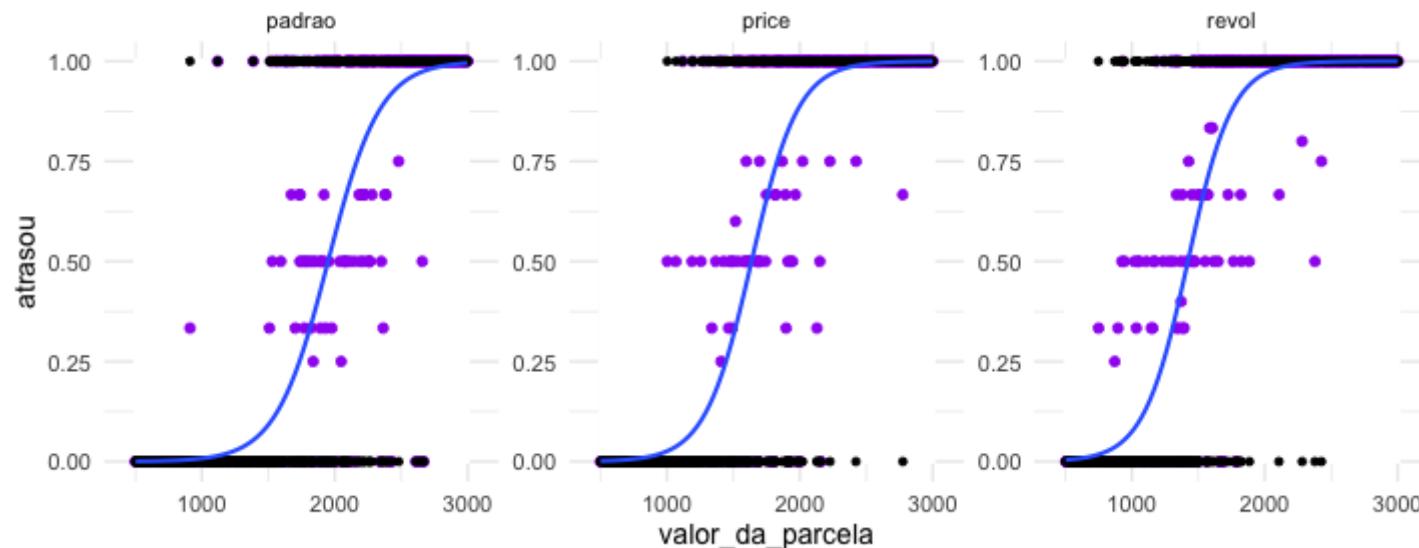


- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Motivação - outro exemplo

Somos da área de inadimplência e precisamos agir para assessorar clientes em situação iminente de atraso.

Obtivemos o seguinte banco de dados



- PERGUNTA: Qual a probabilidade do contrato 123 atrasar a próxima fatura no mês que vem?

Machine Learning

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

$$y \approx f(x)$$

Nos exemplos:

$$\text{vendas} = f(\text{midia}, \text{investimento})$$

$$\text{inadimplência} = f(\text{valordaparcela}, \text{tipodecontrato})$$

Modo - Regressão e Classificação

Existem dois principais tipos de problemas em Machine Learning:

Regressão

Y é uma variável contínua.

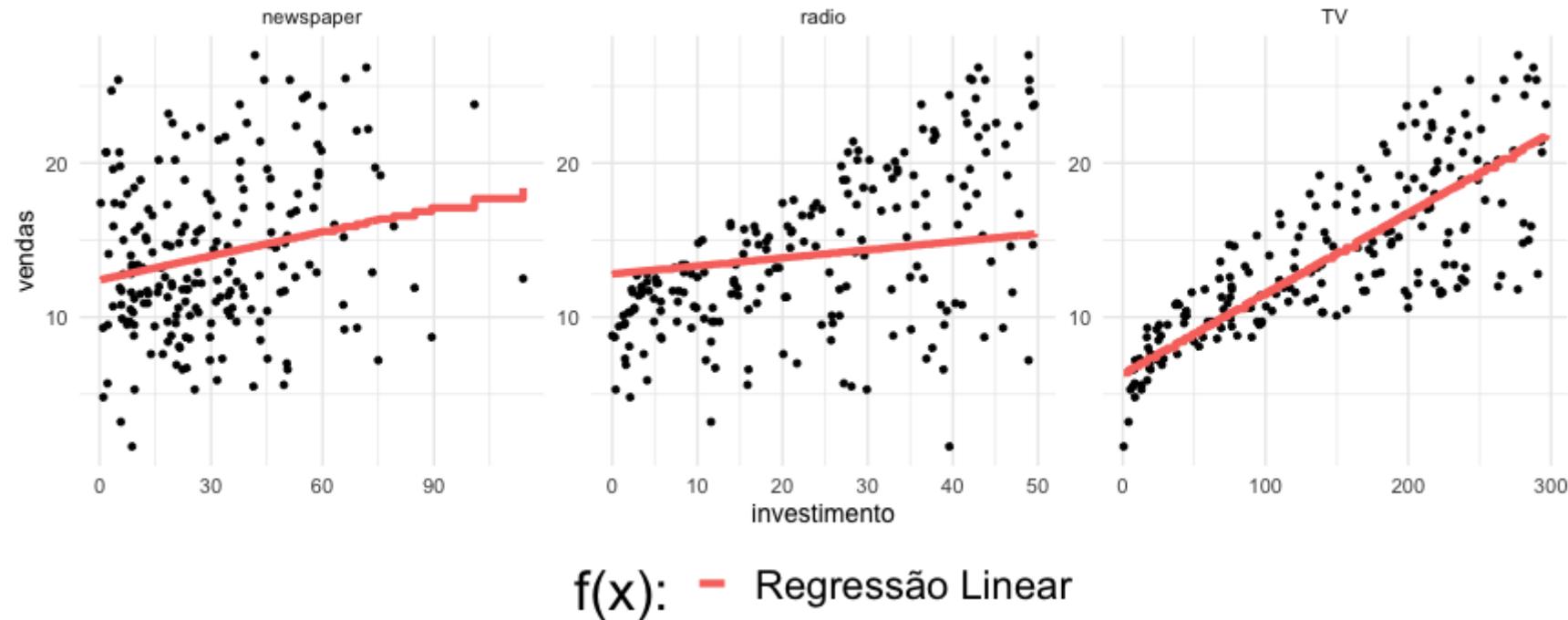
- Volume de vendas
- Peso
- Temperatura
- Valor de Ações

Classificação

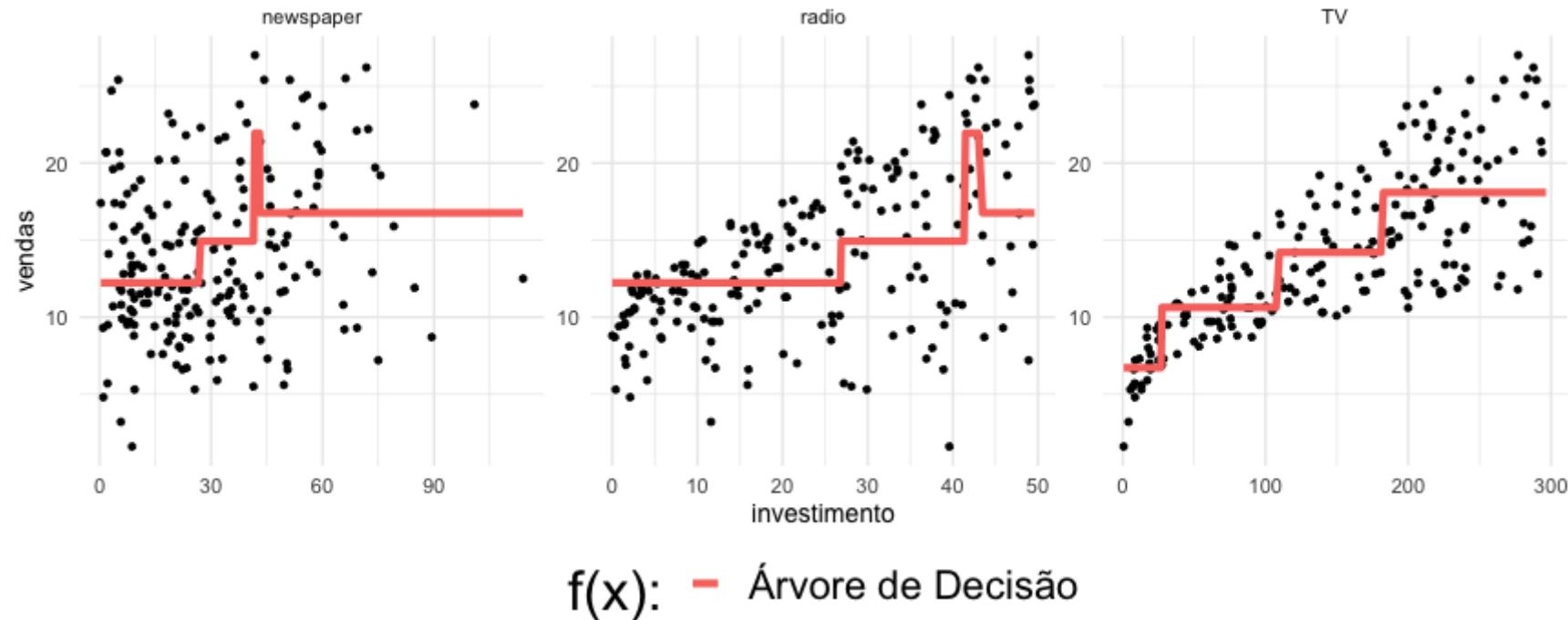
Y é uma variável categórica.

- Fraude/Não Fraude
- Pegou em dia/Não pagou
- Cancelou assinatura/Não cancelou
- Gato/Cachorro/Cavalo/Outro

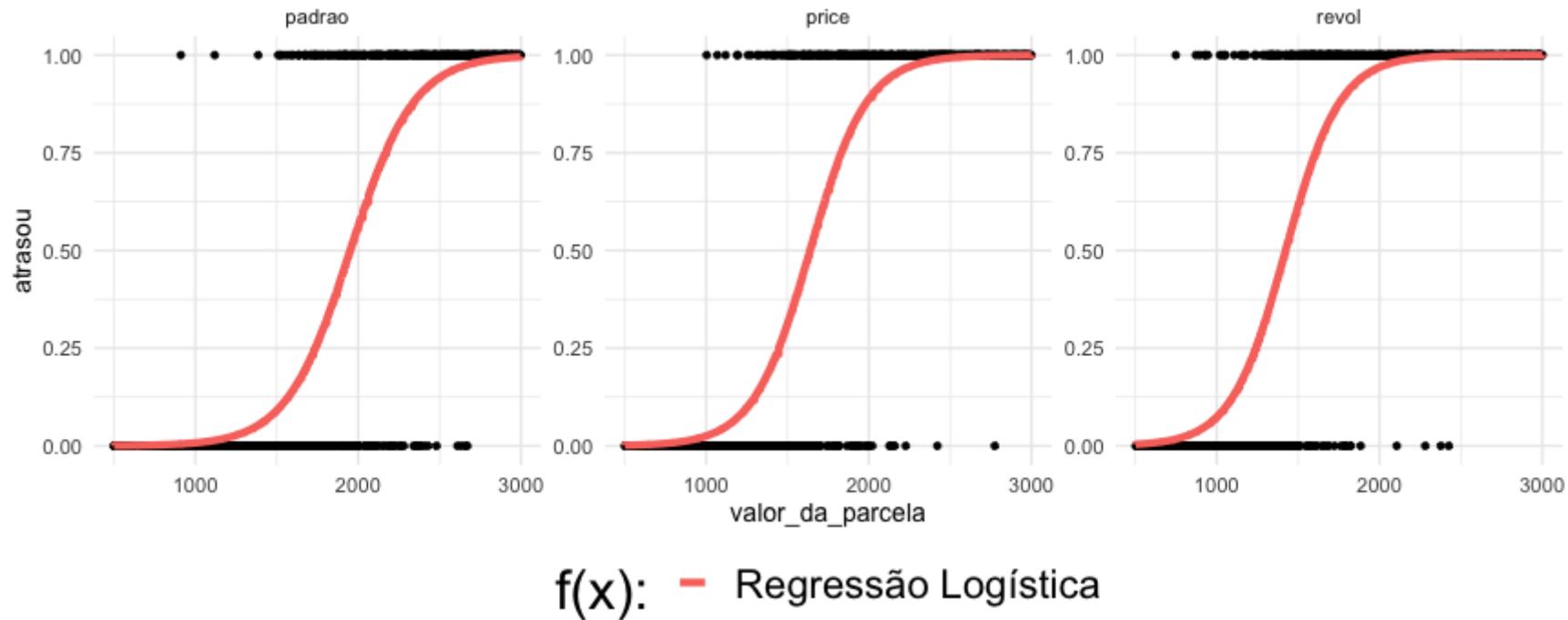
Exemplos de $f(x)$



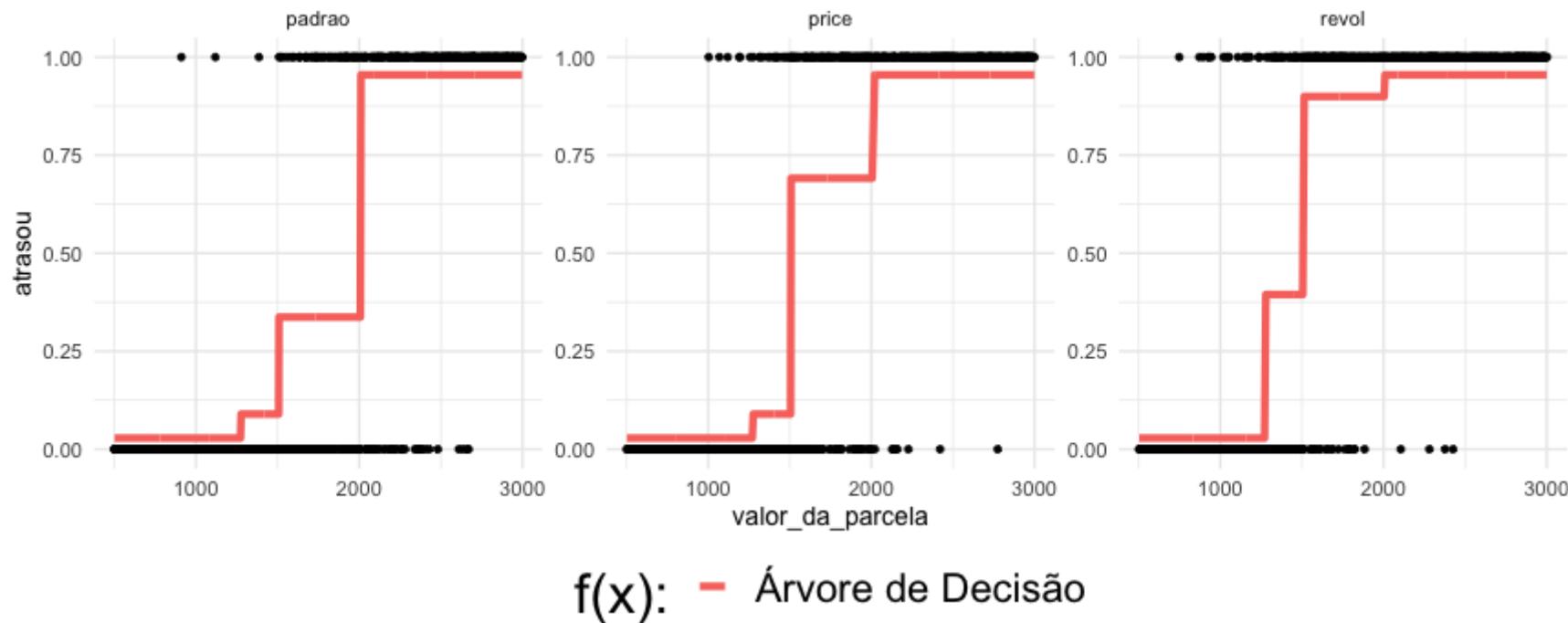
Exemplos de $f(x)$



Exemplos de $f(x)$



Exemplos de $f(x)$



Definições e Nomenclaturas

A tabela por trás (do excel, do sql, etc.)

midia	investimento	vendas
TV	220.3	24.7
newspaper	25.6	5.3
newspaper	38.7	18.3
radio	42.3	25.4
radio	43.9	22.3
TV	139.5	10.3
radio	11.0	7.2
radio	1.6	6.9

Definições e Nomenclaturas

- X_1, X_2, \dots, X_p : variáveis explicativas (ou variáveis independentes ou *features* ou preditores).
- $\mathbf{X} = X_1, X_2, \dots, X_p$: conjunto de todas as *features*.
- \mathbf{Y} : variável resposta (ou variável dependente ou *target*).
- $\hat{\mathbf{Y}}$: valor **esperado** (ou predição ou estimado ou *fitted*).
- $f(\mathbf{X})$ também é conhecida também como "Modelo" ou "Hipótese".

No exemplo:

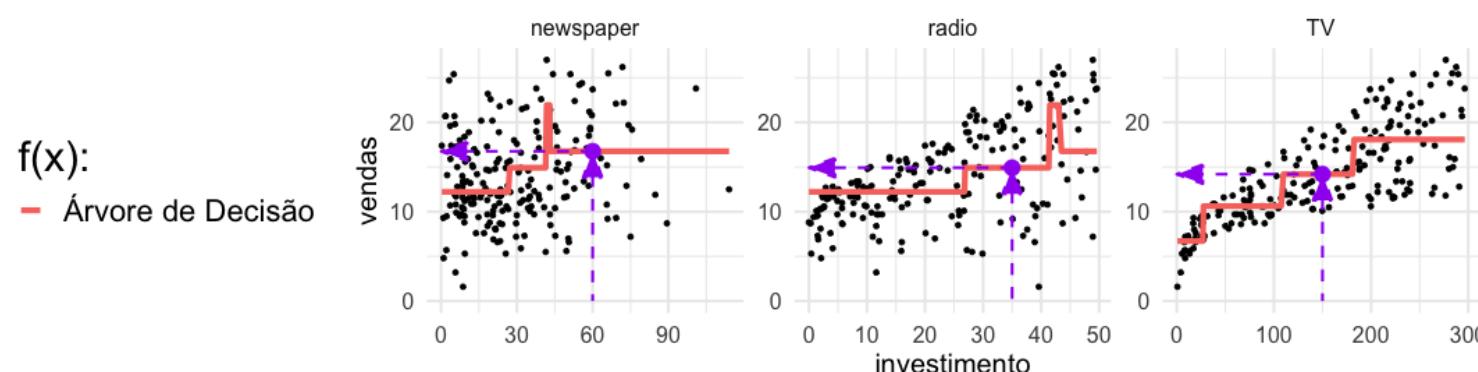
- X_1 : **midia** - indicador de se a propaganda é para jornal, rádio, ou TV.
- X_2 : **investimento** - valor do orçamento
- \mathbf{Y} : **vendas** - qtd vendida

Definições e Nomenclaturas

Observado *versus* Esperado

- Y é um valor **observado** (ou verdade ou *truth*)
- \hat{Y} é um valor **esperado** (ou predição ou estimado ou *fitted*).
- $Y - \hat{Y}$ é o resíduo (ou erro)

Por definição, $\hat{Y} = f(x)$ que é o valor que a função f retorna.

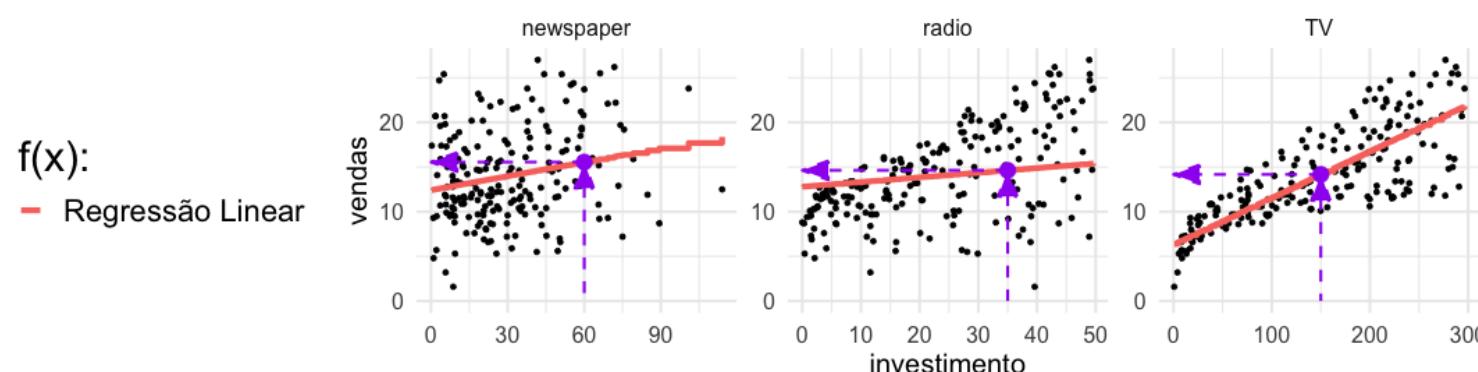


Definições e Nomenclaturas

Observado *versus* Esperado

- Y é um valor **observado** (ou verdade ou *truth*)
- \hat{Y} é um valor **esperado** (ou predição ou estimado ou *fitted*).
- $Y - \hat{Y}$ é o resíduo (ou erro)

Por definição, $\hat{Y} = f(x)$ que é o valor que a função f retorna.



Definições e Nomenclaturas

A tabela por trás depois das previsões

midia	investimento	vendas	arvore	regressao_linear
TV	220.3	24.7	18.1	17.8
newspaper	25.6	5.3	12.2	13.8
newspaper	38.7	18.3	14.9	14.4
radio	42.3	25.4	21.9	15.0
radio	43.9	22.3	16.8	15.1
TV	139.5	10.3	14.2	13.6
radio	11.0	7.2	12.2	13.4
radio	1.6	6.9	12.2	12.9

Outro Exemplo: Classificação

A tabela por trás (do excel, do sql, etc.)

tipo_de_contrato	valor_da.Parcela	atrasou
padrao	2692	1
revol	1245	0
price	2369	1
revol	1571	1
padrao	2349	1
revol	1652	1
price	2840	1
revol	924	0

Outro Exemplo: Classificação

- X_1, X_2, \dots, X_p : variáveis explicativas (ou variáveis independentes ou *features* ou preditores).
- $\mathbf{X} = X_1, X_2, \dots, X_p$: conjunto de todas as *features*.
- \mathbf{Y} : variável resposta (ou variável dependente ou *target*).
- $\hat{\mathbf{Y}}$: valor **esperado** (ou predição ou score ou *fitted*).
- $f(\mathbf{X})$ também é conhecida também como "Modelo" ou "Hipótese".

No exemplo:

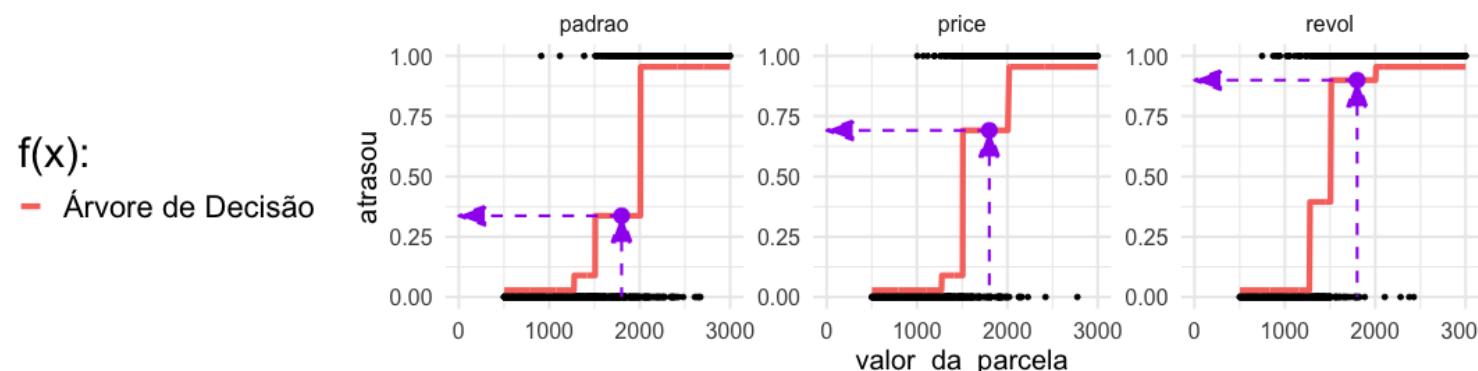
- X_1 : `tipo_de_contrato` - flags de se o contrato é padrao, price, ou revol.
- X_2 : `valor_da_parcela` - Valor da parcela do financiamento.
- \mathbf{Y} : `atrasou` - indicador de atraso maior que 30 dias na parcela.

Outro Exemplo: Classificação

Observado *versus* Esperado

- Y é um valor **observado** (ou rótulo ou target ou verdade ou *truth*)
- \hat{Y} é um valor **esperado** (ou score ou probabilidade predita).
- $\log(\hat{Y})$ ou $\log(1-\hat{Y})$ é o resíduo (ou erro)

Por definição, $\hat{Y} = f(x)$ que é o valor que a função f retorna.

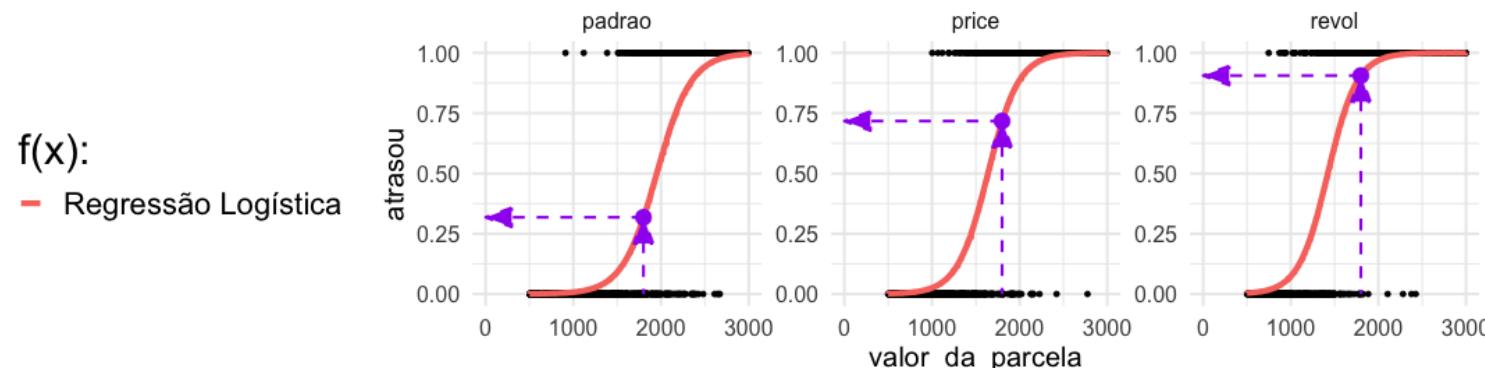


Outro Exemplo: Classificação

Observado *versus* Esperado

- Y é um valor **observado** (ou rótulo ou target ou verdade ou *truth*)
- \hat{Y} é um valor **esperado** (ou score ou probabilidade predita).
- $\log(\hat{Y})$ ou $\log(1-\hat{Y})$ é o resíduo (ou erro)

Por definição, $\hat{Y} = f(x)$ que é o valor que a função f retorna.

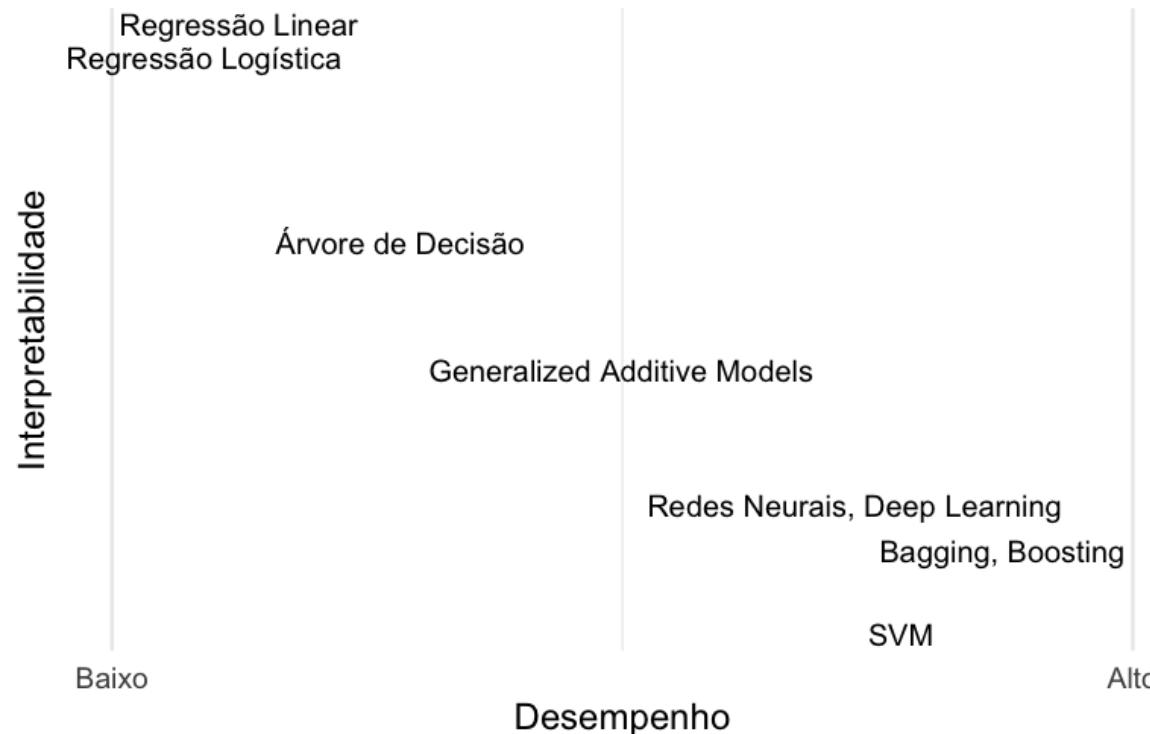


Outro Exemplo: Classificação

A tabela por trás depois das previsões

tipo_de_contrato	valor_da.Parcela	atrasou	arvore	regressao_logistica
padrao	2692	1	0.95	0.98
revol	1245	0	0.03	0.26
price	2369	1	0.95	0.98
revol	1571	1	0.90	0.71
padrao	2349	1	0.95	0.88
revol	1652	1	0.90	0.80
price	2840	1	0.95	1.00
revol	924	0	0.03	0.05

Desempenho vs Interpretabilidade da $f(x)$



Características importantes: interprabilidade, custo computacional e poder preditivo.

Por que ajustar uma f ?

- Predição
- Inferência

Predição

Em muitas situações X está disponível facilmente mas, Y não é fácil de descobrir. (Ou mesmo não é possível descobrí-lo). Queremos que $\hat{Y} = \hat{f}(X)$ seja uma boa estimativa (preveja bem o futuro). Neste caso não estamos interessados em como é a estrutura \hat{f} desde que ela apresente previsões boas para Y .

Por exemplo:

- Meu cliente vai atrasar a fatura no mês que vem?

Por que ajustar uma f?

- Predição
- Inferência

Inferência

Em inferência estamos mais interessados em entender a relação entre as variáveis explicativas X e a variável resposta Y .

Por exemplo:

- A dose da droga é eficaz para o tratamento da doença X até quanto?
- **Quanto que é** o impacto nas vendas para cada real investido em TV?

Neste material focaremos em **predição**.

Por que ajustar uma f?

Ordenação priorização	Média o valor importa	Variância intervalo de confiança	Explicação inferência de parâmetros
			
Quais são os 10 mais prováveis?	Qual a probabilidade?	Qual minha confiança nessa estimativa?	Essa probabilidade está associada a qual fator?
Qual é o mais caro?	Quanto é o preço estimado?	Quanto esse preço pode variar?	O que pode fazer esse preço variar?
Quais imagens mais parecem gatos?	É provável que a imagem seja de um gato?	Essa probabilidade é informativa?	Quais formas são características de gato?
Aplicações: <ul style="list-style-type: none">• Cobrança• Campanha publicitária• Seleção de currículos	Aplicações: <ul style="list-style-type: none">• Reconhecimento Facial• Fraude• Crédito	Aplicações: <ul style="list-style-type: none">• Diagnóstico Médico• Linha de produção• Pesquisa de Opinião	Aplicações: <ul style="list-style-type: none">• Impactos Regulatórios• Ensaios clínicos• Teste A/B

Métricas - "Melhor $f(x)$ " segundo o quê?

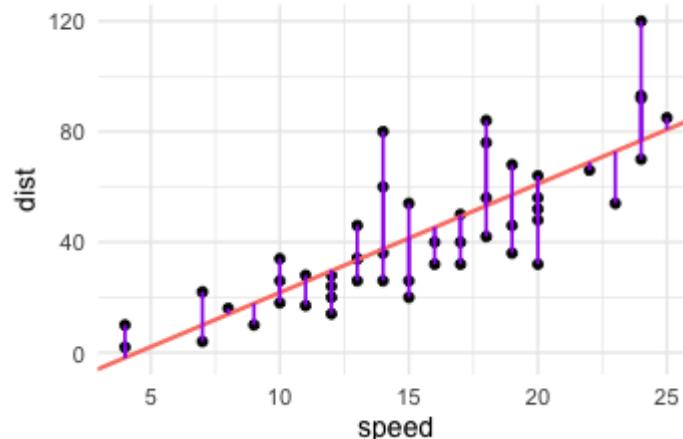
Queremos a $f(x)$ que **erre menos**.

Exemplo de **métrica** de erro: **Root Mean Squared Error**.

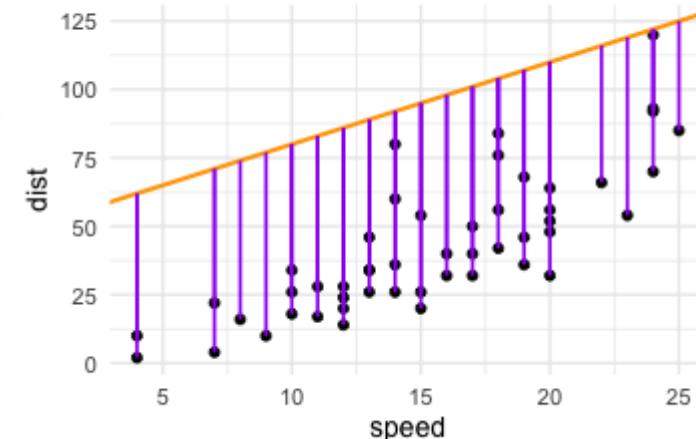
$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2}$$

Os segmentos azuis são os resíduos (ou o quanto o modelo errou naqueles pontos).

Resíduos da Melhor Reta



Resíduos da Reta Escolhida a Mão



Métricas - "Melhor $f(x)$ " segundo o quê?

Queremos a $f(x)$ que **erre menos**.

Exemplo de métrica de erro: **Root Mean Squared Error**.

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2}$$

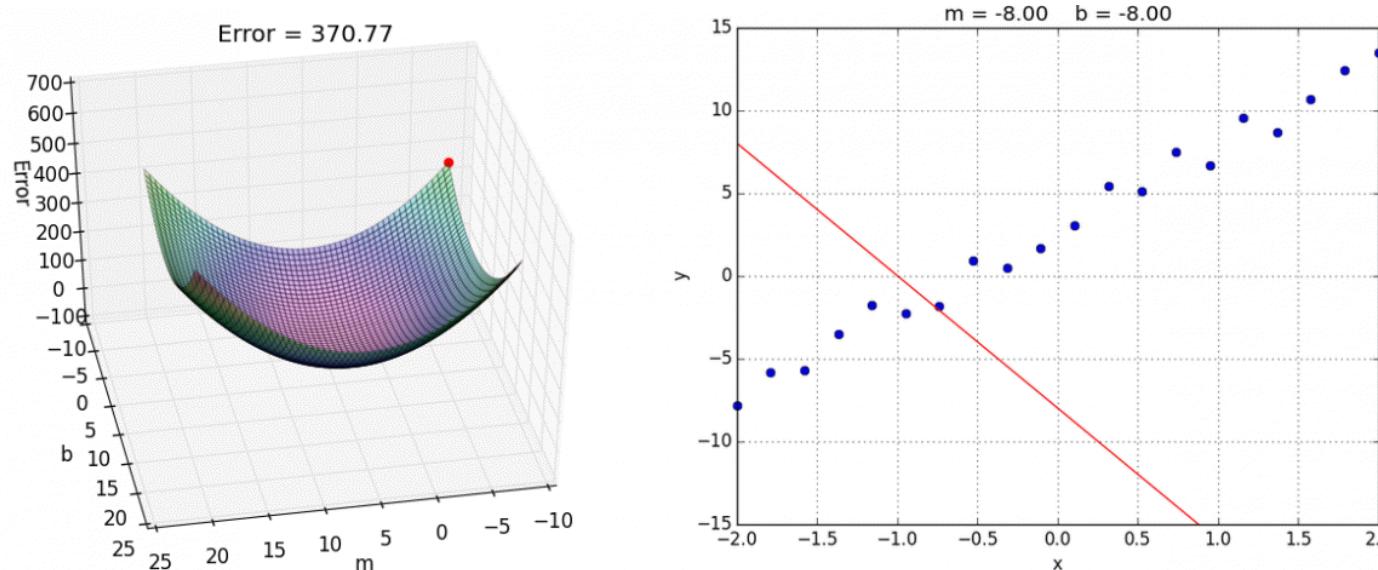
Ou seja, nosso **objetivo** é

Encontrar $f(x)$ que nos retorne o ~menor~ RMSE.

Métricas - "Melhor $f(x)$ " segundo o quê?

Queremos a reta que **erre menos**.

Exemplo: Modelo de regressão linear $f(x) = \beta_0 + \beta_1 x$.



Fonte: https://alykhantejani.github.io/images/gradient_descent_line_graph.gif

Métricas - "Melhor $f(x)$ " segundo o quê?

Queremos a $f(x)$ que **erre menos**.

Exemplo de métrica de erro: **Root Mean Squared Error**.

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2}$$

MAE: Mean Absolute Error

$$MAE = \frac{1}{N} \sum |y_i - \hat{y}_i|$$

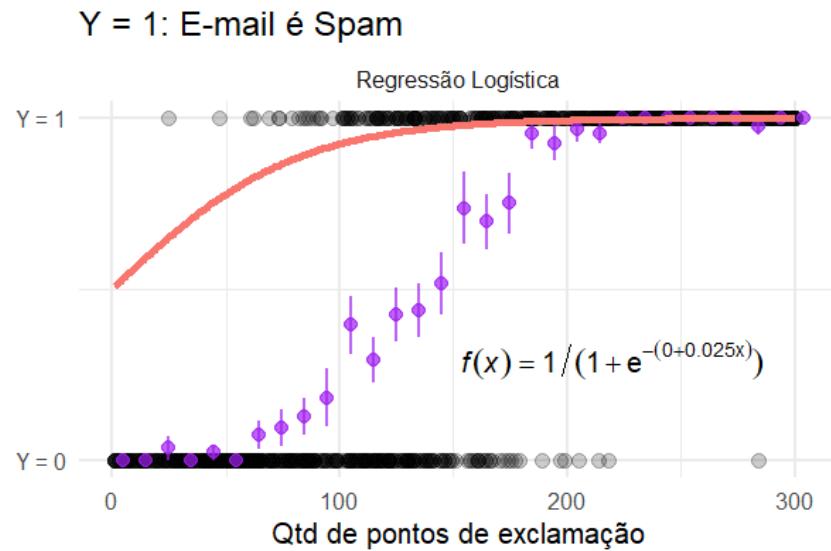
R2: R-squared

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

Métricas - "Melhor f(x)" segundo o quê?

Na classificação a estratégia é a mesma. Queremos a curva que **erre menos**.

Exemplo: Modelo de regressão logística $f(x) = \frac{1}{1+e^{-(\beta_0+\beta_1x)}}$.



Métrica de Erro da Logística:

$$D = \frac{-1}{N} \sum [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

Em que

$$\hat{y}_i = f(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

Métricas

Métricas: para medir o quanto a $f(x)$ está errando as previsões.

Regressão

Y é uma variável contínua.

- RMSE
- R2
- MAE
- MAPE ...

lista de métricas no yardstick

Classificação

Y é uma variável categórica.

- Deviance (Cross-Entropy)
- Acurácia
- AUROC
- Precision/Recall
- F1
- Kappa ...

Regressão Linear

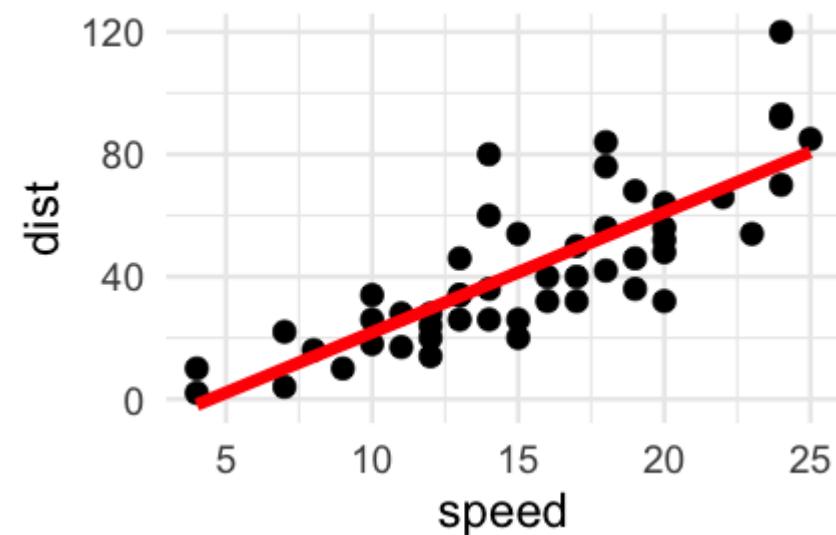
Regressão Linear Simples

$$y = \beta_0 + \beta_1 x$$

Exemplo:

$$dist = \beta_0 + \beta_1 speed$$

```
### No R:  
linear_reg() %>%  
  fit(dist ~ speed, data=cars)
```

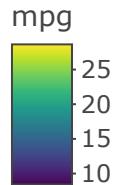


Ver [ISL](#) página 61 (Simple Linear Regression).

Regressão Linear

Regressão Linear Múltipla

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$



Exemplo:

$$mpg = \beta_0 + \beta_1 wt + \beta_2 disp$$

```
### No R:  
linear_reg() %>%  
  fit(mpg ~ wt + disp, data=mtcars)
```

Fonte: sthda.com

Regressão Linear - "Melhor Reta"

Queremos a reta que **erre menos**.

Uma métrica de erro: RMSE

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{N} \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 speed))^2}$$

Ou seja, nosso é **encontrar os $\hat{\beta}'s$ que nos retorno o ~menor~ RMSE.**

IMPORTANTE!

o RMSE é **Métrica** que a regressão usa como **Função de Custo**.

- **Função de Custo - Métrica** usada para encontrar os melhores parâmetros.

Qual o valor ótimo para β_0 e β_1 ?

No nosso exemplo, a nossa **HIPÓTESE** é de que

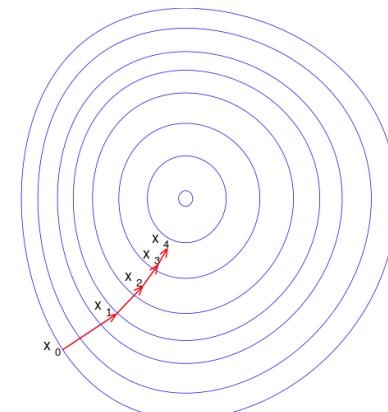
$$dist = \beta_0 + \beta_1 speed$$

Então podemos escrever o RMSE

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{N} \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 speed))^2}$$

Método mais utilizado para otimizar modelos com parâmetros: **Gradient Descent**

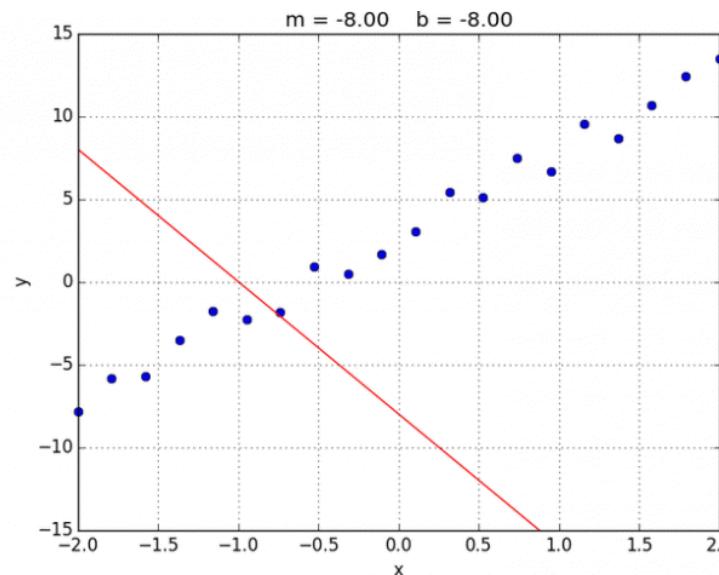
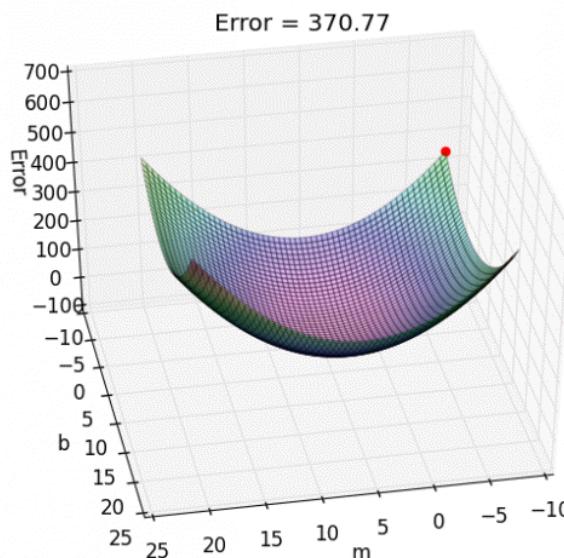
Ver [Wikipedia do Gradient Descent](#)



Regressão Linear - "Melhor Reta"

Queremos a reta que **erre menos**.

Modelo: $y = \beta_0 + \beta_1 x$



Fonte: https://alykhantejani.github.io/images/gradient_descent_line_graph.gif

Depois de estimar...

$$\hat{y} = \hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$$

Exemplo:

$$\hat{dist} = \hat{\beta}_0 + \hat{\beta}_1 speed$$

Colocamos um $\hat{\cdot}$ em cima dos termos para representar "estimativas". Ou seja, \hat{y}_i é uma estimativa de y_i . No nosso exemplo,

- $\hat{\beta}_0$ é uma estimativa de β_0 e vale -17.5.
- $\hat{\beta}_1$ é uma estimativa de β_1 e vale 3.9.
- \hat{dist} é uma estimativa de $dist$ e vale $-17.5 + 3.9 \times speed$.

```
# Exercício: se speed for 15 m/h, quanto que  
# seria a distância dist esperada?
```

Tidymodels

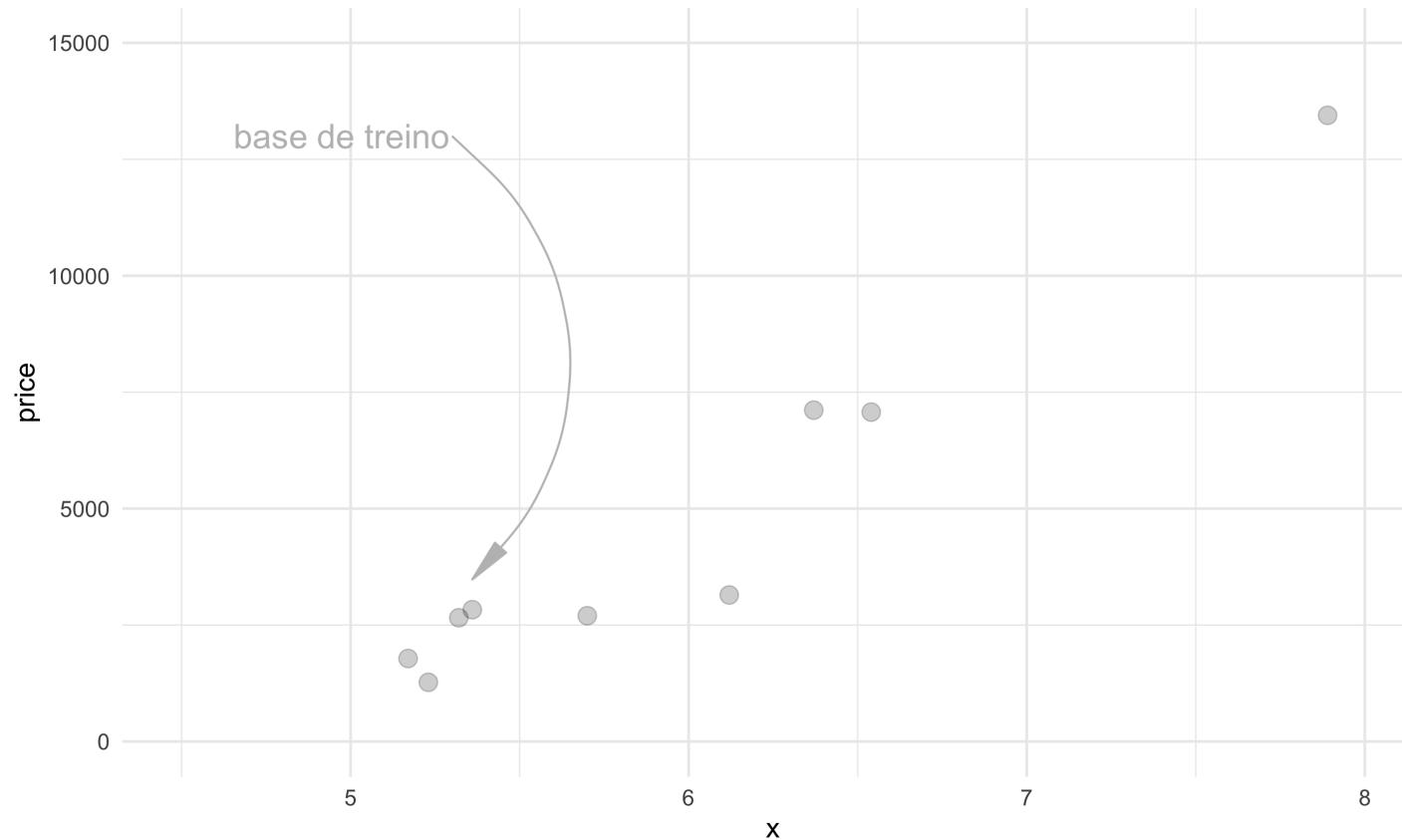
- Conjunto de pacotes/framework para desenvolvimento de modelos preditivos. Muitos tutoriais e guias no [site](#).
- Em desenvolvimento ativo pela RStudio. Possui muitas semelhanças com o 'tidyverse' o que faz com que mais prático.
- Unifica o uso dos modelos já existentes no R. Ele é também extensível: você pode implementar um novo modelo que funcione com o tidymodels.
- Alternativas: [{caret}](#), [{mlr3}](#), [{scikit-learn}](#) (Python), [{PyCaret}](#) (Python). Em geral é fácil migrar de um framework p/ o outro - a parte mais difícil é aprender o fluxo de trabalho de **machine learning**.

Exemplo 01

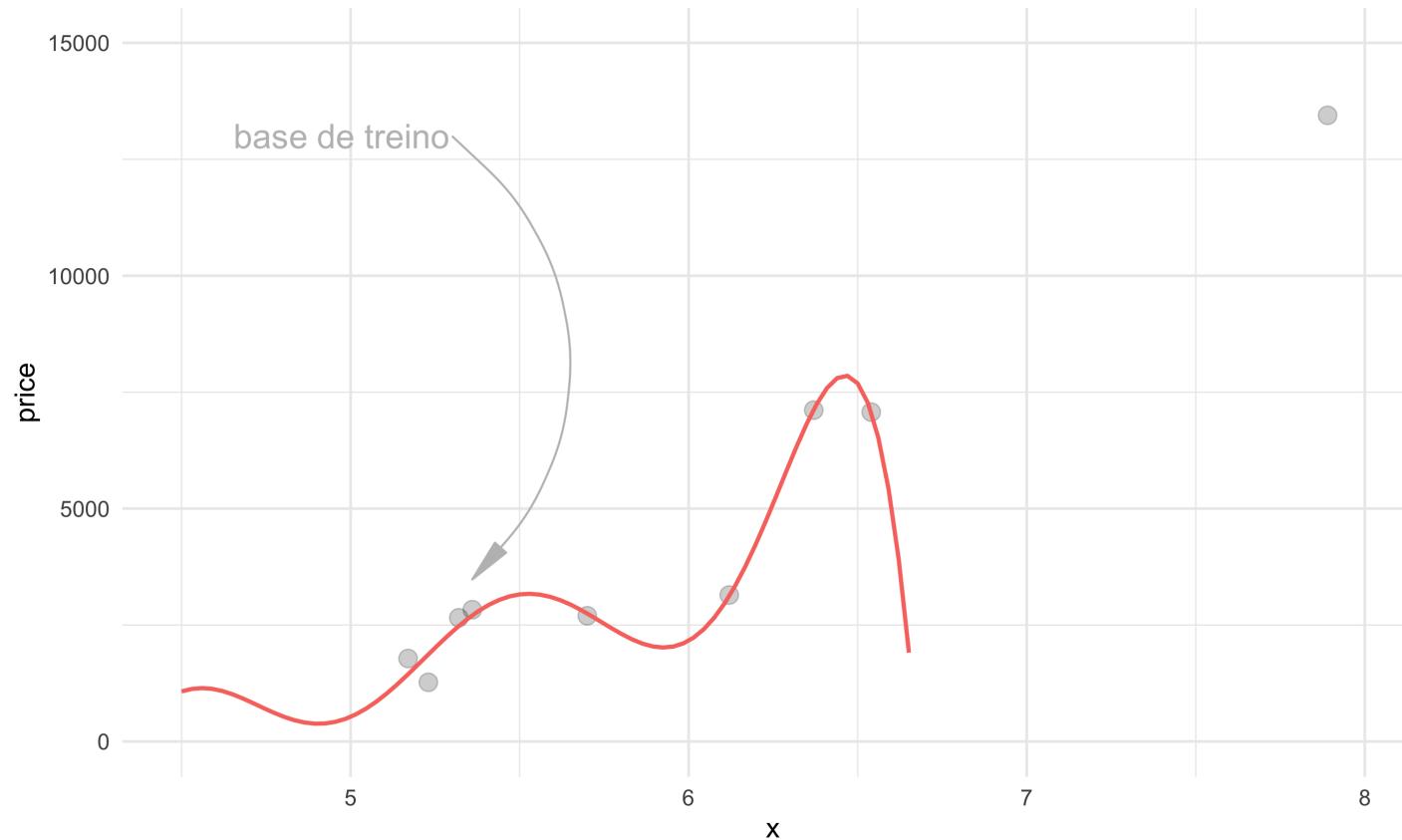
Overfitting (sobreajuste)

- Acontece quando um modelo funciona muito pior quando usado com dados novos quando comparado com a performance nos dados em que foi treinado.
- Uma das principais preocupações quando ajustamos modelos em ML.
- **Solução:** Sempre testar o modelo com dados 'novos'.

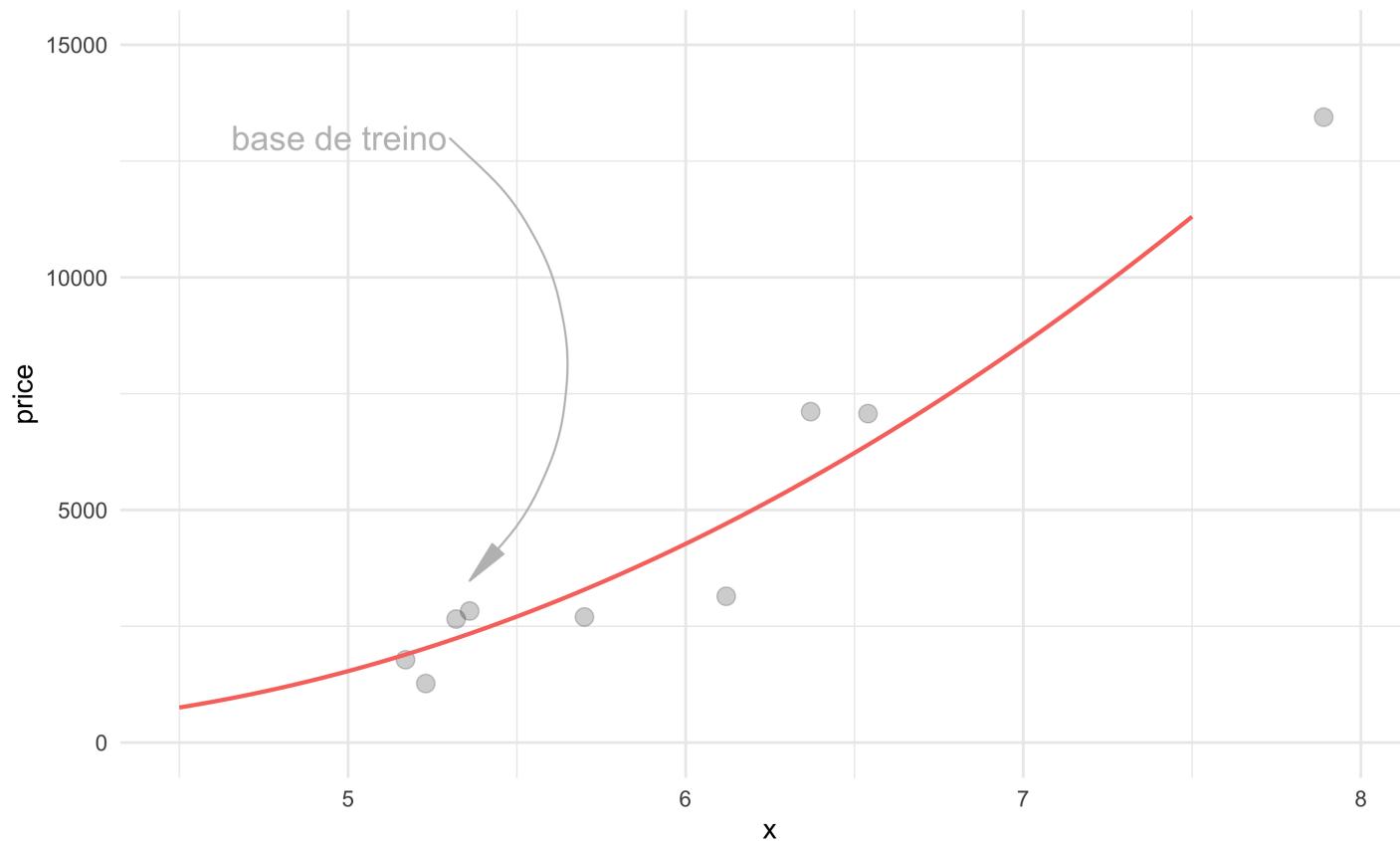
Overfitting (sobreajuste)



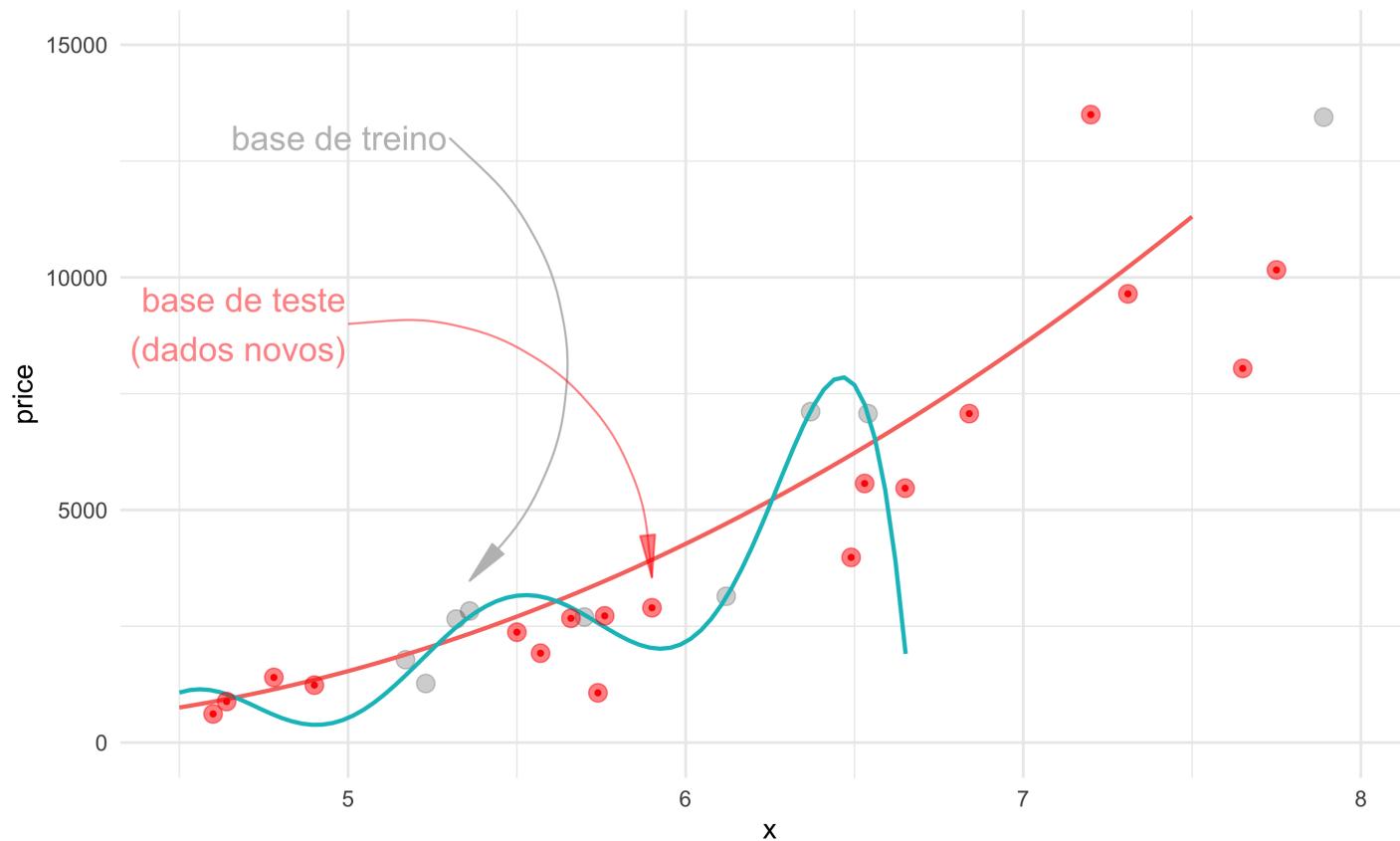
Overfitting (sobreajuste)



Overfitting (sobreajuste)



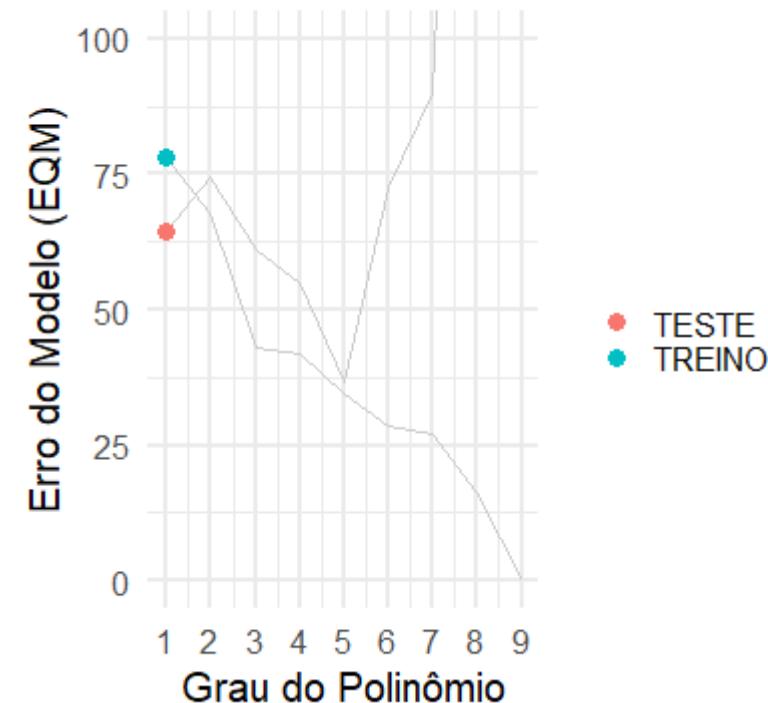
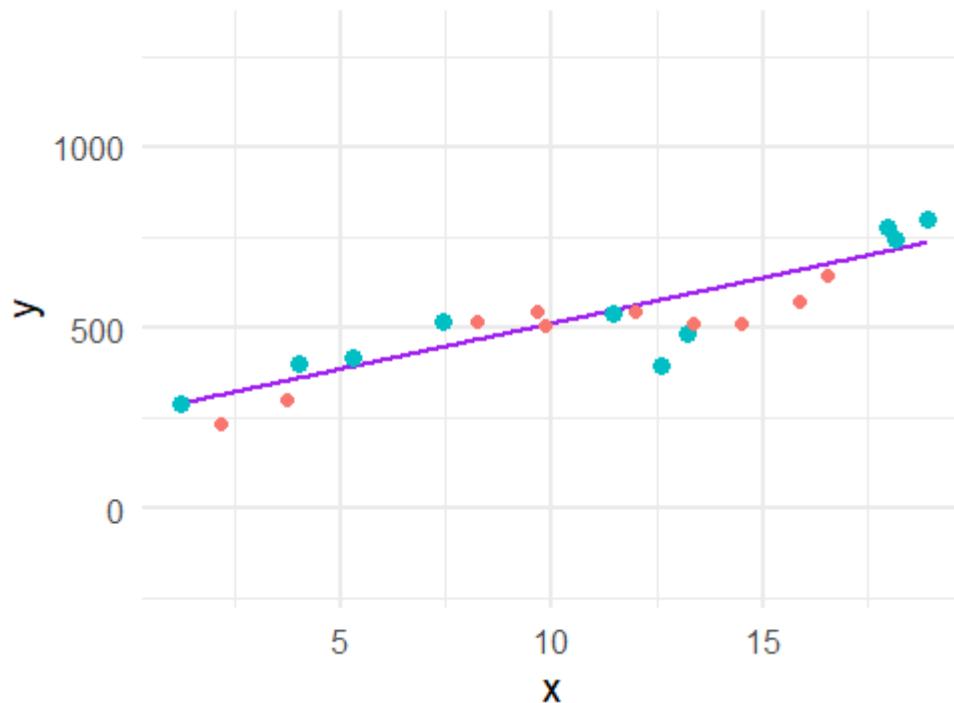
Overfitting (sobreajuste)



Overfitting (sobreajuste)

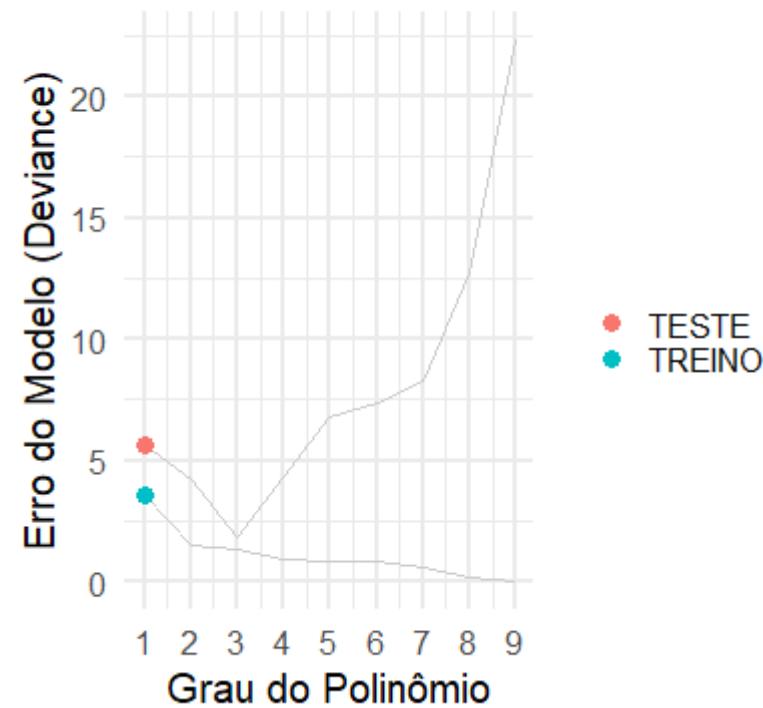
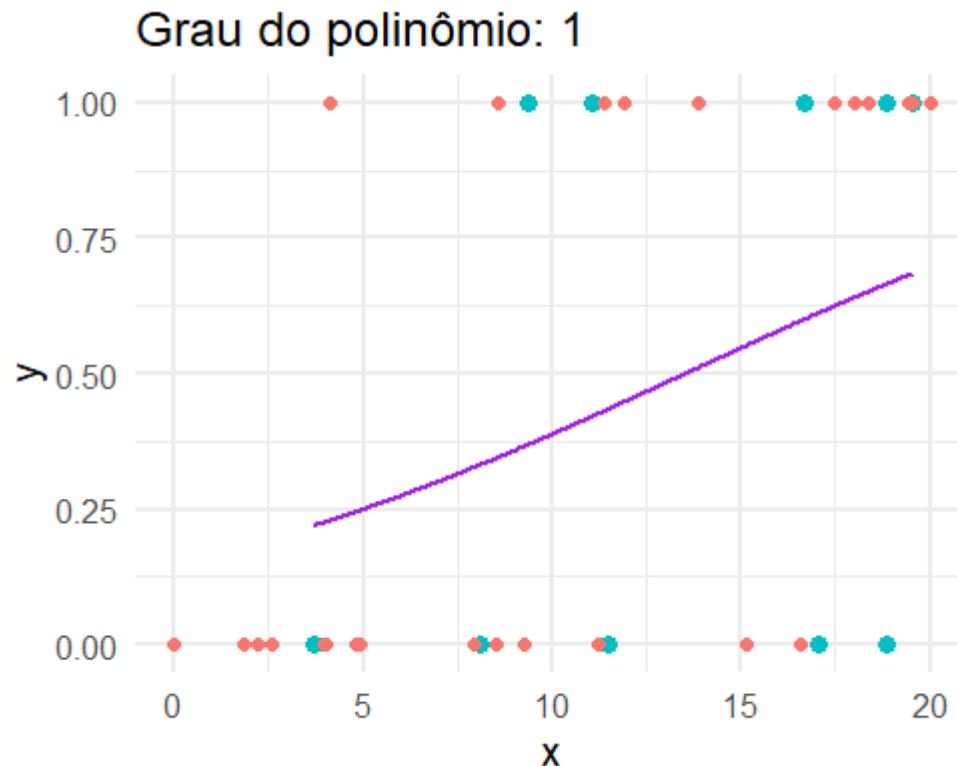
Intuição

Grau do polinômio: 1



Overfitting (sobreajuste)

Intuição



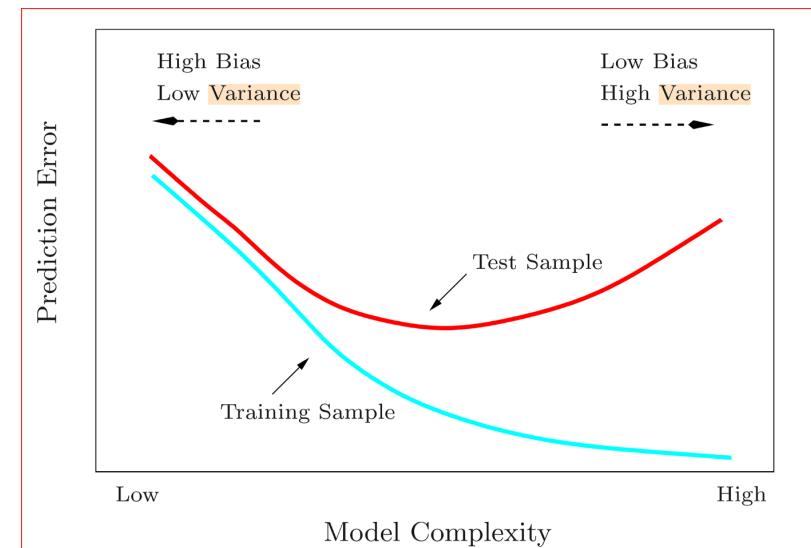
Dados novos vs antigos

- **Base de Treino** (dados antigos): a base de histórico que usamos para ajustar o modelo.
- **Base de Teste** (dados novos): a base que irá simular a chegada de dados novos, "em produção".

```
initial_split(dados, prop=3/4)
```

"Quanto mais complexo for o modelo, menor será o **erro de treino**."

"Porém, o que importa é o **erro de teste**."



Exemplo 02

Dados novos vs antigos

Estratégia

- 1) Separar inicialmente a base de dados em duas: treino e teste.

```
initial_split(dados, prop=3/4) # 3/4 de treino aleatoriamente  
initial_time_split(dados, prop=3/4) # 3/4 de treino respeitando a ordem
```

A base de teste só será tocada quando a modelagem terminar. Ela nunca deverá influenciar nas decisões que tomamos durante o período da modelagem.

palavra-chave: **data leakage** ou **vazamento de informação**

Regularização

Objetivo da Regularização: Oferecer um parâmetro (um valor que podemos mudar) para termos controle sobre a **complexidade** da $f(x)$ e assim evitar o *sobreajuste*.

No exemplo da regressão linear, haverá um valor λ que chamaremos de "hiperparâmetro" da regressão. Iremos chutar diferentes valores de λ até encontrar a melhor $f(x)$.

Regularização - LASSO

Relembrando o nossa **função de custo** RMSE.

$$RMSE = \sqrt{\frac{1}{N} \sum (y_i - \hat{y}_i)^2} = \sqrt{\frac{1}{N} \sum (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \cdots + \hat{\beta}_p x_{pi}))^2}$$

Regularizar é "não deixar os β' s soltos demais".

$$RMSE_{regularizado} = RMSE + \lambda \sum_{j=1}^p |\beta_j|$$

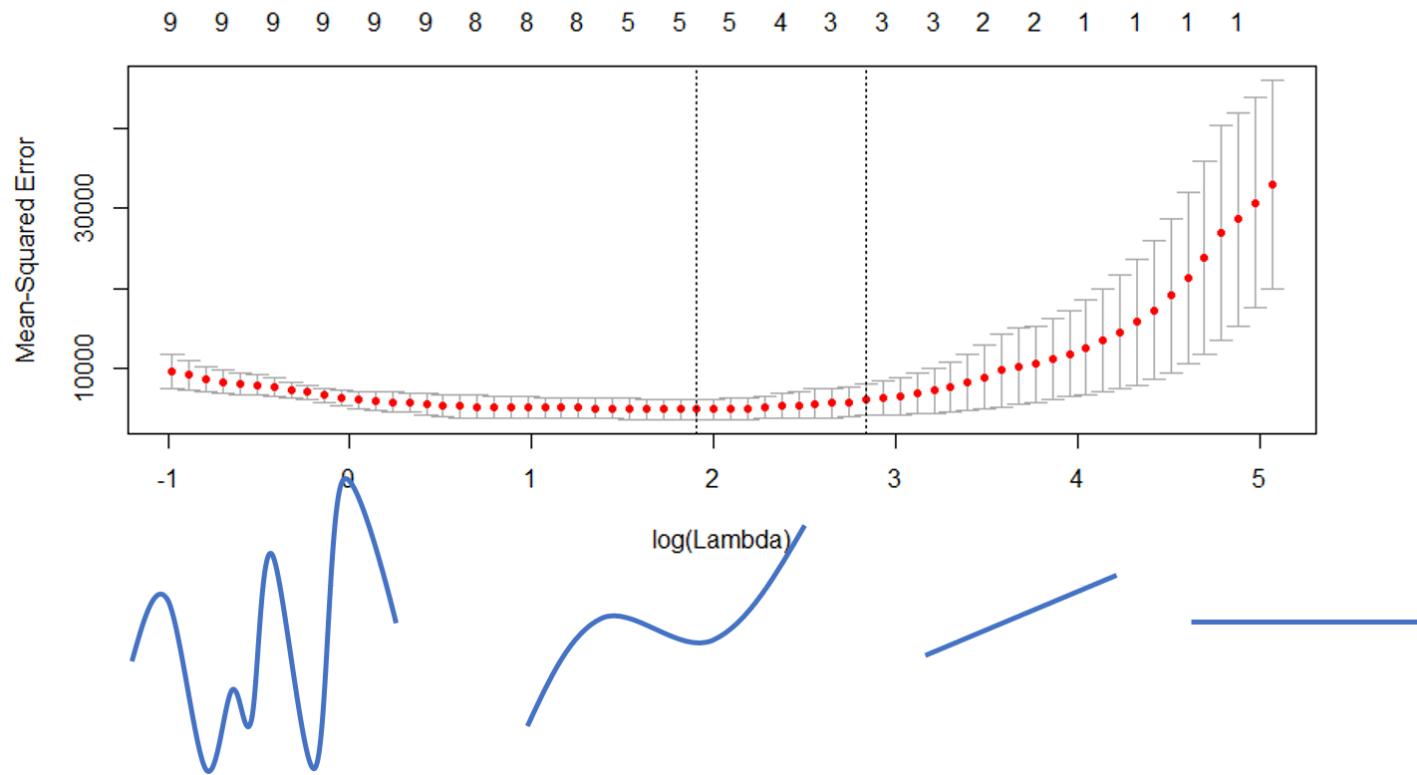
Ou seja, **penalizamos** a função de custo se os β' s forem muito grandes.

PS1: O λ é um **hiperparâmetro** da Regressão Linear.

PS2: Quanto maior o λ , mais penalizamos os β' s por serem grandes.

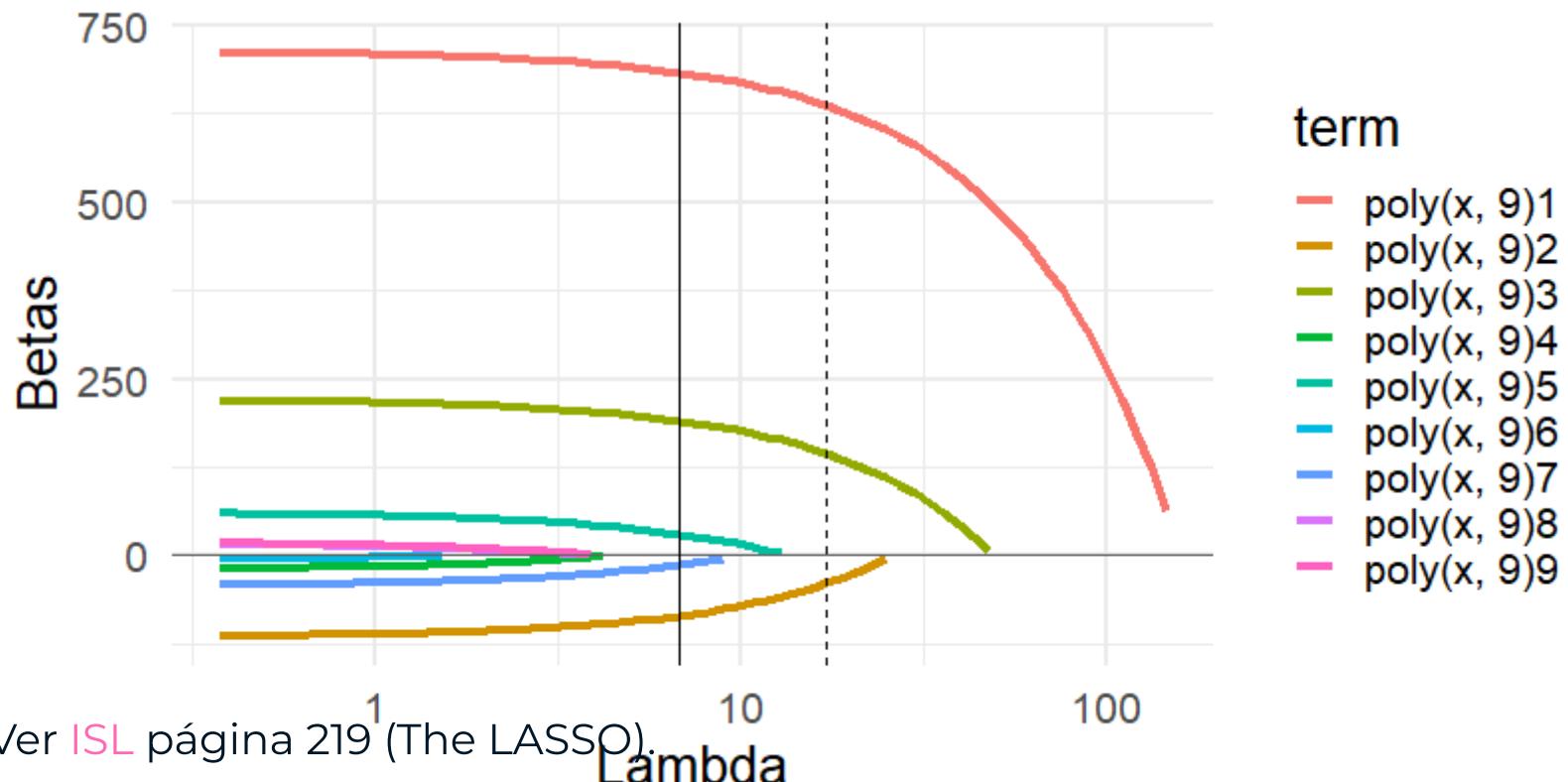
Regularização - LASSO

Vamos testar diversos valores para λ até encontrar o que dá o menor erro de teste.



Regularização - LASSO

Conforme aumentamos o λ , forçamos os β 's a serem cada vez menores.



Hiperparâmetros

São parâmetros que têm que ser definidos antes de ajustar o modelo. Não há como achar o valor ótimo diretamente nas funções de custo. Precisam ser achados **na força bruta**.

Exemplo: **lambda** da penalização do LASSO (**penalty**)

```
linear_reg(penalty = 0.0)
linear_reg(penalty = 0.1)
linear_reg(penalty = 1.0)
linear_reg(penalty = tune())
```

Problema!

Teremos que testar muitos 'lambda's'. Podemos desgastar a base de teste (erro de teste vai ter alta variabilidade). Para isso, inventaram a estratégia de reamostragem que oferece uma estimativa do erro de predição (erro de teste) de forma mais confiável.

Cross-validation (validação cruzada)

O que Validação cruzada faz: estima (muito bem) o erro de predição.

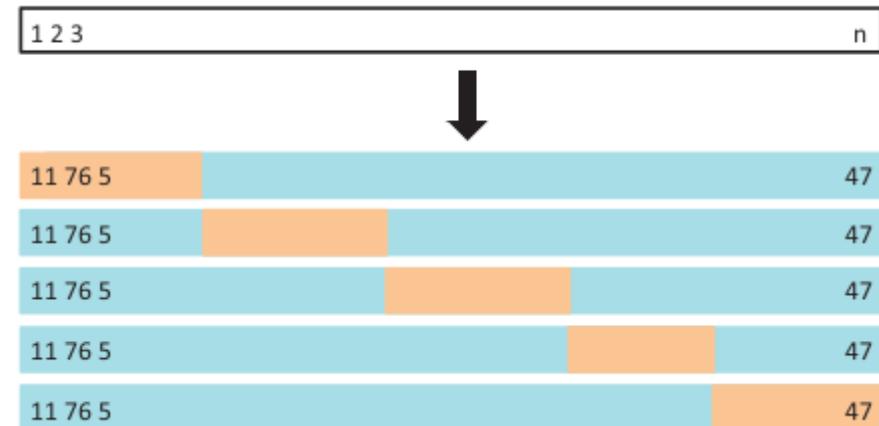
Objetivo da Validação cruzada: encontrar o melhor conjunto de hiperparâmetros.

Estratégia

1) Dividir o banco de dados em K partes.
(Por ex, K = 5 como na figura)

2) Ajustar o mesmo modelo K vezes,
deixar sempre um pedaço de fora para
servir de base de teste.

3) Teremos K valores de erros de teste.
Tira-se a média dos erros.



Cross-validation (validação cruzada)

```
vfold_cv(cars, v = 5)
```

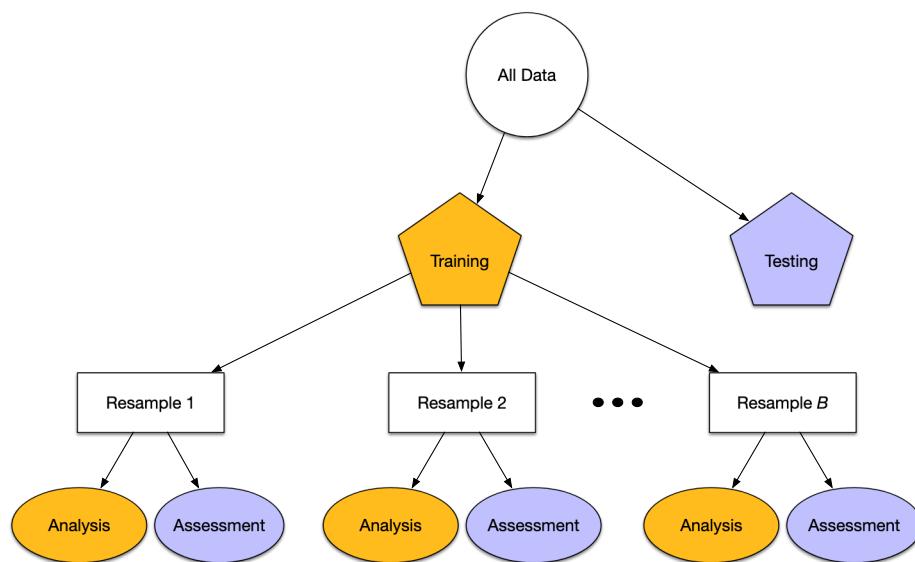
```
## # 5-fold cross-validation
## # A tibble: 5 × 6
##   splits          id    n_treino n_teste regressao rmse_teste
##   <list>        <chr>     <dbl>     <dbl> <list>           <dbl>
## 1 <split [40/10]> Fold1      40       10 <lm>            12.0
## 2 <split [40/10]> Fold2      40       10 <lm>            21.4
## 3 <split [40/10]> Fold3      40       10 <lm>            16.6
## 4 <split [40/10]> Fold4      40       10 <lm>            11.3
## 5 <split [40/10]> Fold5      40       10 <lm>            13.8
```

ERRO DE VALIDAÇÃO CRUZADA:

$$RMSE_{cv} = \frac{1}{5} \sum_{i=1}^5 RMSE_{Fold_i} = 15,1$$

Cross-validation (validação cruzada)

Esquema das divisões de bases:



Fonte: bookdown.org/max/FES/resampling.html

Cross-validation (validação cruzada)

Em pseudo-código:

```
K <- 5

fold <- sample.int(K, nrow(mtcars), replace = TRUE)
for (k in 1:K) {
  train <- mtcars[fold != k,]
  valid <- mtcars[fold == k,]

  # ajusta_modelo(train)
  # metrics(valid)
}
```

Exemplo 03

Regularização - Ridge

No LASSO, usamos o módulo dos betas para fazer a regularização. É possível fazer também usando o quadrado dos coeficientes:

$$RMSE_{Ridge} = RMSE + \lambda \sum_{j=1}^p \beta_j^2$$

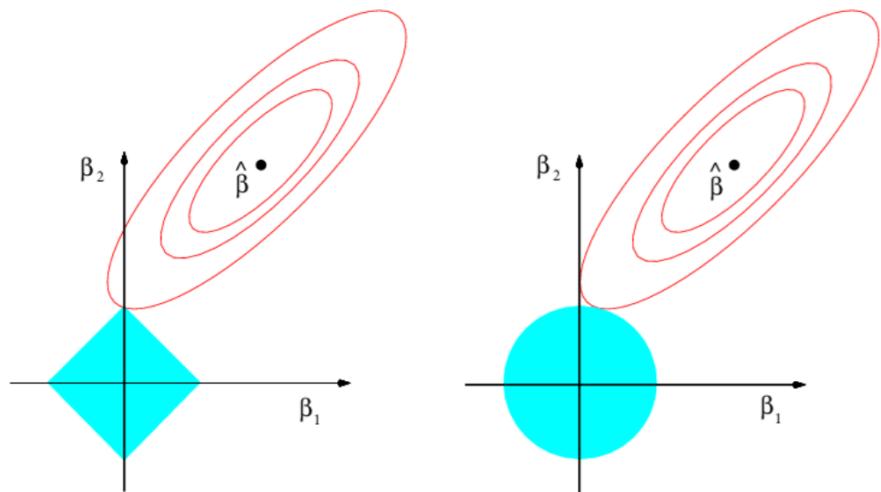
Também é possível misturar os dois

$$RMSE_{regularizado} = RMSE + (\alpha) \times \lambda \sum_{j=1}^p |\beta_j| + (1 - \alpha) \times \lambda \sum_{j=1}^p \beta_j^2$$

Nessa definição α é chamado de 'mixture' (mistura). Quando $\alpha = 1$ temos o LASSO e quando $\alpha = 0$ temos Ridge. O α também pode ser tunado.

Ridge vs LASSO

O LASSO tem uma propriedade muito interessante quando comparada ao Ridge. Por razões matemáticas, ele consegue produzir estimativas esparsas, isto é, alguns coeficientes podem ser exatamente 0.



Fonte: ISLR pag. 224

Exercício 01

Resumo dos conceitos

