

# Como implementar/testar métodos e modelos



Na nossa disciplina, já aprendemos sobre os seguintes métodos:

- Regressão Linear
- Regressão Logística
- Redes Neurais (aprendizado profundo)

Levando em conta uma perspectiva mais geral, falaremos nessa aula sobre **dicas de como implementar/testar de forma eficiente algoritmos de aprendizado de máquina**.

Suponha que você está implementando uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico:

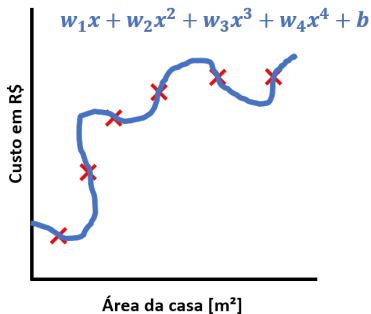
$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados
- Excluir características (testar modelo mais simples)
- Incluir características (testar modelo mais complexo)
- Adicionar características polinomiais ao modelo  $(x_1^2, x_2^2, x_1 x_2, \dots)$
- Aumentar  $\lambda$
- Reduzir  $\lambda$

**Pergunta:** Qual dessas alternativas devemos testar primeiro?



### Perguntas

- Qual é o valor da função custo associado a esse modelo?
- Trata-se de um bom modelo para esses dados?

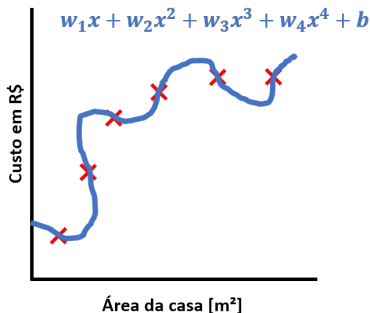
**Observação:** Quando o modelo leva em conta mais do que uma característica, é mais difícil fazermos essa análise graficamente. Como proceder então?

# Ávaliando a qualidade de um modelo

Podemos **testar quantitativamente** a qualidade do nosso modelo fazendo o seguinte:

- 1 Alimente seu modelo com um conjunto de dados que ele não “viu” durante seu treinamento
- 2 Utilize alguma métrica quantitativa para verificar sua performance.

Como você acha que o modelo abaixo se sairia nesse teste?



## Avaliando a qualidade de um modelo: Regra 70/30

Área da casa [m <sup>2</sup> ]	Custo em R\$	
32	51.000	Dados de treinamento
149	265.000	
78	110.000	
36	49.000	
132	280.000	
60	120.000	Dados de teste
59	99.000	
129	279.000	
62	124.000	
220	315.000	

Use cerca de 70% dos dados para treinar os parâmetros do seu modelo

Use cerca de 30% dos dados para validar o modelo treinado

“Um bom modelo é capaz de performar bem mesmo quando aplicado sobre o conjunto de dados de teste (dados que ele não teve acesso durante seu treinamento).”

Notação para dados de treinamento:

$$\begin{array}{c} (x^{(1)}, y^{(1)}) \\ \dots \\ (x^{(m_{trein})}, y^{(m_{trein})}) \end{array}$$

→ conjunto de dados de treinamento contendo  $m_{trein}$  amostras.

Notação para dados de teste:

$$\begin{array}{c} (x_{teste}^{(1)}, y_{teste}^{(1)}) \\ \dots \\ (x_{teste}^{(m_{teste})}, y_{teste}^{(m_{teste})}) \end{array}$$

→ conjunto de dados de teste contendo  $m_{teste}$  amostras.

**Pergunta:** O que acontecerá se você decidir usar uma regra do tipo 80/20?

Pergunta:

Como avaliar a qualidade de um modelo de regressão que foi treinado a partir da minimização da função custo

$$J(\vec{w}, b) = \frac{1}{2m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m_{\text{train}}} \sum_{j=1}^n w_j^2 \quad ?$$



## Pergunta:

Como avaliar a qualidade de um modelo de regressão que foi treinado a partir da minimização da função custo

$$J(\vec{w}, b) = \frac{1}{2m_{trein}} \sum_{i=1}^{m_{trein}} \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m_{trein}} \sum_{j=1}^n w_j^2 \quad ?$$

## Resposta:

Verifique como o modelo  $f_{\vec{w}, b}(\vec{x})$  se sai ao ser aplicado ao conjunto de dados de teste:

$$J_{teste}(\vec{w}, b) = \frac{1}{2m_{teste}} \sum_{i=1}^{m_{teste}} \left( f_{\vec{w}, b}(\vec{x}_{teste}^{(i)}) - y_{teste}^{(i)} \right)^2$$

## Pergunta:

Como avaliar a qualidade de um modelo de regressão que foi treinado a partir da minimização da função custo

$$J(\vec{w}, b) = \frac{1}{2m_{trein}} \sum_{i=1}^{m_{trein}} \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m_{trein}} \sum_{j=1}^n w_j^2 \quad ?$$

## Resposta:

Verifique como o modelo  $f_{\vec{w}, b}(\vec{x})$  se sai ao ser aplicado ao conjunto de dados de teste:

$$J_{teste}(\vec{w}, b) = \frac{1}{2m_{teste}} \sum_{i=1}^{m_{teste}} \left( f_{\vec{w}, b}(\vec{x}_{teste}^{(i)}) - y_{teste}^{(i)} \right)^2$$

## Pergunta:

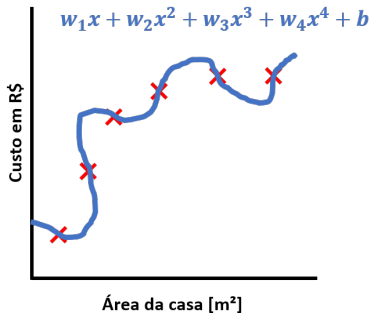
Seja

$$J_{trein}(\vec{w}, b) = \frac{1}{2m_{trein}} \sum_{i=1}^{m_{trein}} \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2$$

Quem naturalmente tenderá a ser maior,  $J_{teste}(\vec{w}, b)$  ou  $J_{trein}(\vec{w}, b)$  ?

## Exemplo

Suponha que no gráfico abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



Pergunta:

Teremos

- A)  $J_{teste}(\vec{w}, b) \gg J_{trein}(\vec{w}, b)$  ?
- B)  $J_{teste}(\vec{w}, b) \ll J_{trein}(\vec{w}, b)$  ?

OBS: Trata-se de um indicativo de **overfitting**.

## Exemplo (problema de classificação)

### Pergunta:

Como avaliar a qualidade de um modelo de classificação que foi treinado a partir da minimização da função

$$J(\vec{w}, b) = -\frac{1}{m_{trein}} \sum_{i=1}^{m_{trein}} \left[ y^{(i)} \log \left( f_{\vec{w}, b}(\vec{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left( 1 - f_{\vec{w}, b}(\vec{x}^{(i)}) \right) \right] \\ + \frac{\lambda}{2m_{trein}} \sum_{j=1}^n w_j^2$$

### Resposta:

Verifique como o modelo  $f_{\vec{w}, b}(\vec{x})$  se sai ao ser aplicado ao conjunto de dados de teste:

$$J_{teste}(\vec{w}, b) = -\frac{1}{m_{teste}} \sum_{i=1}^{m_{teste}} \left[ y_{teste}^{(i)} \log \left( f_{\vec{w}, b}(\vec{x}_{teste}^{(i)}) \right) + (1 - y_{teste}^{(i)}) \log \left( 1 - f_{\vec{w}, b}(\vec{x}_{teste}^{(i)}) \right) \right]$$

Seja

$$J_{trein}(\vec{w}, b) = -\frac{1}{m_{trein}} \sum_{i=1}^{m_{trein}} \left[ y^{(i)} \log \left( f_{\vec{w}, b}(\vec{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left( 1 - f_{\vec{w}, b}(\vec{x}^{(i)}) \right) \right]$$

Quem naturalmente tenderá a ser maior,  $J_{teste}(\vec{w}, b)$  ou  $J_{trein}(\vec{w}, b)$  ?

## Exemplo (problema de classificação)

Alternativamente, em problemas de classificação, também é muito comum usarmos a **taxa de acerto** (ou a **taxa de erro**) para avaliarmos a qualidade de um modelo.

Considerando a utilização da taxa de erro, podemos definir:

$$J_{teste}(\vec{w}, b) = \frac{\text{quantidade de amostras em que } \hat{y} \neq y_{teste}}{m_{teste}} \times 100\%$$

$$J_{trein}(\vec{w}, b) = \frac{\text{quantidade de amostras em que } \hat{y} \neq y_{trein}}{m_{trein}} \times 100\%$$

Utilizando a taxa de erro, quanto maior  $J_{teste}(\vec{w}, b)$ , pior o modelo. Caso tivéssemos utilizado a taxa de acerto como métrica, seria o contrário.

Suponha que você calculou  $J_{teste}(\vec{w}, b)$  para 10 modelos diferentes:

$$1) f_{\vec{w}, b}(\vec{x}) = w_1x + b \rightarrow J_{teste}^{<1>}(\vec{w}, b)$$

$$2) f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + b \rightarrow J_{teste}^{<2>}(\vec{w}, b)$$

$$3) f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + b \rightarrow J_{teste}^{<3>}(\vec{w}, b)$$

$$\vdots$$

$$10) f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + \dots + w_{10}x^{10} + b \rightarrow J_{teste}^{<10>}(\vec{w}, b)$$

Suponha que  $J_{teste}^{<5>}(\vec{w}, b)$  seja o menor erro dentre todos  $J_{teste}^{<d>}(\vec{w}, b)$ ,  $d = 1, \dots, 10$ .

Pergunta:

É razoável eleger o quinto modelo como o melhor modelo?

Resposta:

Sim, porém... (continua no próximo slide)

- A partir do momento que você utilizou o conjunto de dados de teste para refinar algum hiperparâmetro do seu projeto (ordem do polinômio, número de camadas de uma rede neural, taxa de aprendizado, etc), esse conjunto de dados não pode mais ser chamado de conjunto de dados teste. É melhor chamar esse conjunto de dados de:

**Conjunto de dados de validação ( $val$ ) = dados de desenvolvimento = dados de validação cruzada ( $cv$ )**

- A partir do momento que você utilizou o conjunto de dados de teste para refinar algum hiperparâmetro do seu projeto (ordem do polinômio, número de camadas de uma rede neural, taxa de aprendizado, etc), esse conjunto de dados não pode mais ser chamado de conjunto de dados teste. É melhor chamar esse conjunto de dados de:

**Conjunto de dados de validação ( $val$ ) = dados de desenvolvimento = dados de validação cruzada ( $cv$ )**

- Ou seja,  $J_{teste}^{<5>}(\vec{w}, b)$  deixa de ser uma estimativa justa para o erro de generalização verdadeiro, já que pelo menos 1 hiperparâmetro está sendo selecionado com base nesse valor. Por isso,  $J_{teste}^{<5>}(\vec{w}, b)$  nesse caso deve ser chamado de  $J_{val}^{<5>}(\vec{w}, b)$ , sendo que  $J_{val}^{<5>}(\vec{w}, b)$  tenderá a ser menor que o erro de generalização verdadeiro.



- A partir do momento que você utilizou o conjunto de dados de teste para refinar algum hiperparâmetro do seu projeto (ordem do polinômio, número de camadas de uma rede neural, taxa de aprendizado, etc), esse conjunto de dados não pode mais ser chamado de conjunto de dados teste. É melhor chamar esse conjunto de dados de:

**Conjunto de dados de validação ( $val$ ) = dados de desenvolvimento = dados de validação cruzada ( $cv$ )**

- Ou seja,  $J_{teste}^{<5>}(\vec{w}, b)$  deixa de ser uma estimativa justa para o erro de generalização verdadeiro, já que pelo menos 1 hiperparâmetro está sendo selecionado com base nesse valor. Por isso,  $J_{teste}^{<5>}(\vec{w}, b)$  nesse caso deve ser chamado de  $J_{val}^{<5>}(\vec{w}, b)$ , sendo que  $J_{val}^{<5>}(\vec{w}, b)$  tenderá a ser menor que o erro de generalização verdadeiro.
- Isso nos leva à ideia de dividirmos o conjunto inicial de dados em 3 partes, ao invés de apenas duas (continua no próximo slide...)

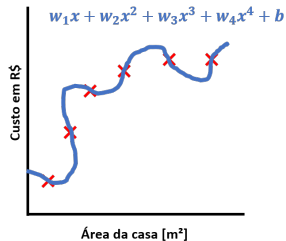
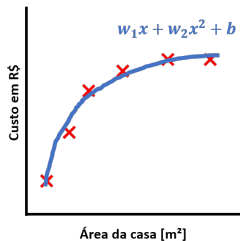
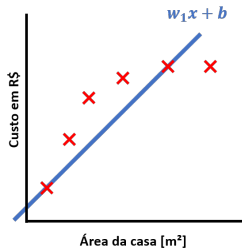
Área da casa [m²]	Custo em R\$	
32	51.000	Dados de treinamento Use cerca de 60% dos dados para treinar os parâmetros do seu modelo ( $w$ e $b$ )
149	265.000	
78	110.000	
36	49.000	
132	280.000	
60	120.000	Dados de validação Use cerca de 20% dos dados para selecionar hiperparâmetros (número de neurônios em cada camada, número de camadas, parâmetro de regularização $\lambda$ , $\alpha$ etc)
59	99.000	
129	279.000	
62	124.000	Dados de teste Use cerca de 20% dos dados para testar seu único modelo final. Assim, você obterá uma estimativa justa para o erro de generalização
220	315.000	

## Observação final:

O conjunto de dados teste serve tão-somente para fornecer uma estimativa justa para o erro de generalização verdadeiro (ou taxa de acerto verdadeira, caso seja o caso). Ele **nunca** deve ser tomado como base para selecionar qualquer tipo de parâmetro ou hiperparâmetro. Caso ele venha ser usado para este fim, ele deixa de ser o conjunto de dados de teste.

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



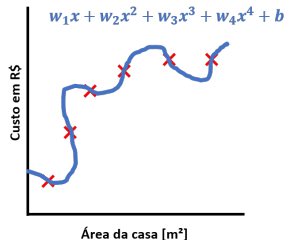
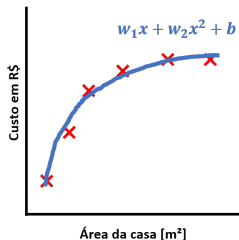
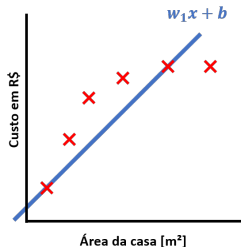
## Pergunta

Olhando apenas para o **gráfico da esquerda**, escolha uma das opções abaixo.

- 1  $J_{trein}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{trein}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{trein}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{trein}$  é grande e  $J_{cv}$  será grande.

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



## Pergunta

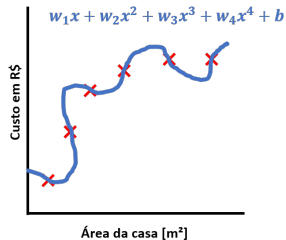
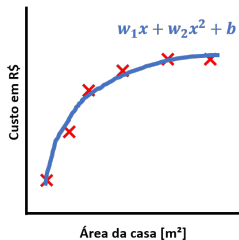
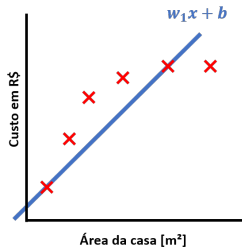
Olhando apenas para o **gráfico da esquerda**, escolha uma das opções abaixo.

- 1  $J_{trein}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{trein}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{trein}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{trein}$  é grande e  $J_{cv}$  será grande.

**Conclusão importante:** Um modelo com alto viés (high bias), não apresenta uma boa performance nem mesmo para os dados de estimação (o modelo subestima até mesmo esses dados)...

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



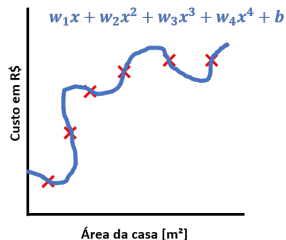
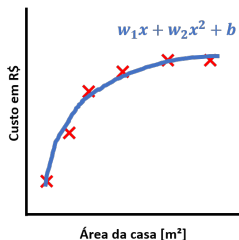
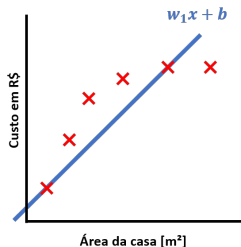
## Pergunta

Olhando apenas para o gráfico da direita, escolha uma das opções abaixo.

- 1  $J_{train}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{train}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{train}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{train}$  é grande e  $J_{cv}$  será grande.

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



## Pergunta

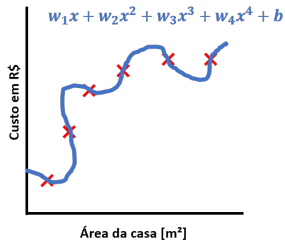
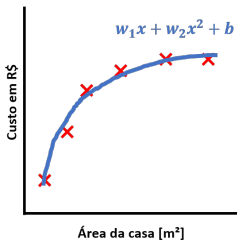
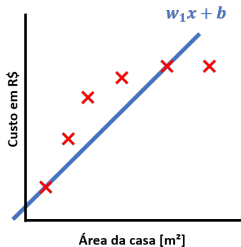
Olhando apenas para o gráfico da direita, escolha uma das opções abaixo.

- 1  $J_{trein}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{trein}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{trein}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{trein}$  é grande e  $J_{cv}$  será grande.

**Conclusão importante:** Um modelo com elevada variância (high variance), apresenta uma boa performance para dados que ele já viu, mas sua performance deteriora consideravelmente para dados não vistos anteriormente...

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



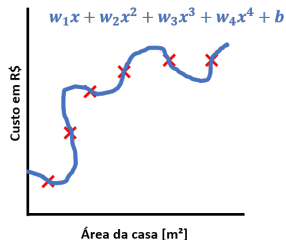
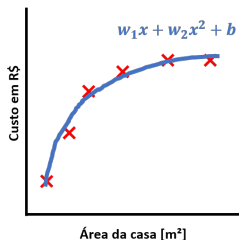
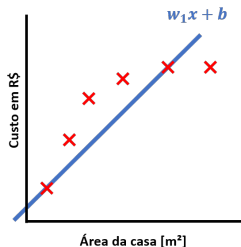
## Pergunta

Olhando apenas para o **gráfico do meio**, escolha uma das opções abaixo.

- 1  $J_{train}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{train}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{train}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{train}$  é grande e  $J_{cv}$  será grande.

# Exemplo

Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.



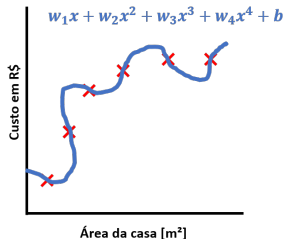
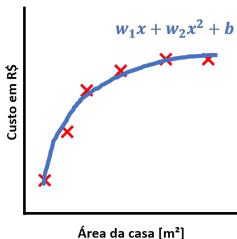
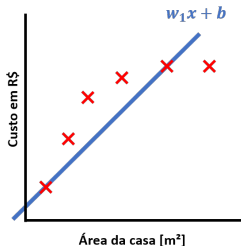
## Pergunta

Olhando apenas para o **gráfico do meio**, escolha uma das opções abaixo.

- 1  $J_{trein}$  é pequeno e  $J_{cv}$  será pequeno.
- 2  $J_{trein}$  é pequeno e  $J_{cv}$  será grande.
- 3  $J_{trein}$  é grande e  $J_{cv}$  será pequeno.
- 4  $J_{trein}$  é grande e  $J_{cv}$  será grande.

**Conclusão importante:** Um modelo adequado apresenta uma boa performance tanto para os dados de estimação como também para dados não vistos anteriormente...



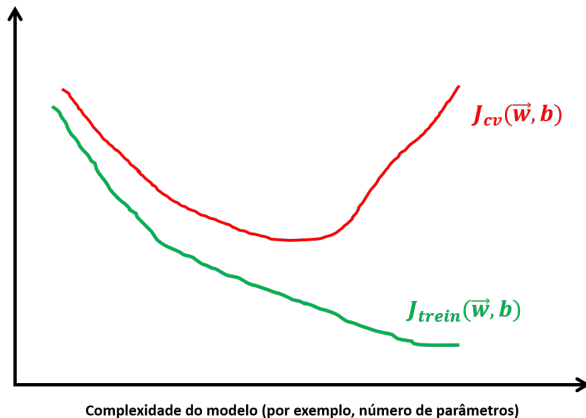


## Pergunta

Analisando os três modelos acima, qual modelo possui maior ordem? Ou seja, é mais complexo?

Com base na resposta para essa pergunta e levando em conta o que já estudamos, podemos perceber o seguinte: (continua no próximo slide)

## Diagnosticando underfitting e overfitting usando $J_{trein}$ e $J_{cv}$



Pergunta

Onde queremos estar nessa figura?

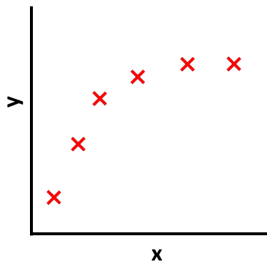
Regra prática:

Teste desde modelos mais simples até modelos altamente complexos. Selecione aquele que resulta no menor  $J_{cv}$ .

## Como a regularização afeta o desempenho do seu modelo?

Suponha que você deseja ajustar um modelo do tipo  $f_{\vec{w}, b}(\vec{x}^{(i)}) = w_1x + w_2x^2 + \dots + w_nx^n + b$  para os dados abaixo usando Regressão Linear com regularização:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left( f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$



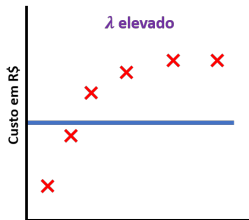
Pergunta:

Supondo, por exemplo,  $n = 8$ . Qual será o modelo resultante caso você escolha  $\lambda = 1000000$ ?

# Como a regularização afeta o desempenho do seu modelo?

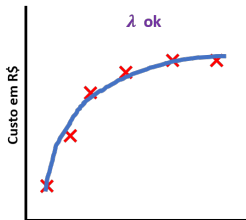
Suponha que nos gráficos abaixo encontram-se representados apenas  $(x^{(i)}, y^{(i)}) \rightarrow$  dados de treinamento.

$$f = w_1x + w_2x^2 + \dots + w_8x^8 + b$$



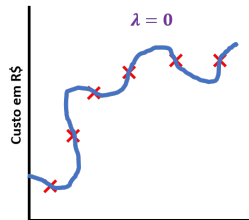
Área da casa [m²]

$J_{trein}(\bar{w}, b)$  e  $J_{cv}(\bar{w}, b)$  elevados  
(alto viés)



Área da casa [m²]

$J_{trein}(\bar{w}, b)$  e  $J_{cv}(\bar{w}, b)$   
pequenos



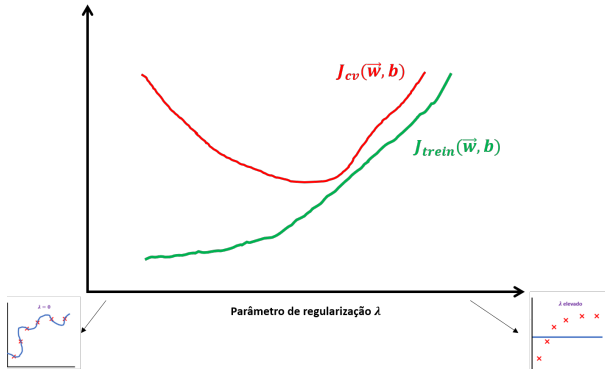
Área da casa [m²]

$J_{trein}(\bar{w}, b) \ll J_{cv}(\bar{w}, b)$  (alta  
variância)

## Pergunta

Analisando os três modelos acima, quando  $\lambda = 0$  teremos um modelo mais simples ou mais complexo?

## Como a regularização afeta o desempenho do seu modelo?



### Pergunta

Onde queremos estar nessa figura?

### Regra prática:

Teste diferentes valores para  $\lambda$  (0; 0.01; 0.02; 0.04; 0.08; ...; 10). Selecione aquele que resulta no menor  $J_{cv}$ .

## Exemplo Numérico: Classificação de tumor maligno ou benigno

Suponha as seguintes taxas de erro:

- $J_{train} = 10.8 \%$
- $J_{cv} = 14.8 \%$
- Performance humana média = 10.6 %

Suponha as seguintes taxas de erro:

- $J_{train} = 10.8 \%$
- $J_{cv} = 14.8 \%$
- Performance humana média = 10.6 %

**Pergunta:**

- O modelo encontra-se subestimando os dados de treinamento? Em outras palavras, temos um problema de alto viés?

**Resposta:**

- Não, afinal, para os dados de treinamento o modelo alcança praticamente um nível humano de performance.

## Exemplo Numérico: Classificação de tumor maligno ou benigno

Suponha as seguintes taxas de erro:

- $J_{train} = 10.8 \%$
- $J_{cv} = 14.8 \%$
- Performance humana média = 10.6 %

**Pergunta:**

- O modelo encontra-se subestimando os dados de treinamento? Em outras palavras, temos um problema de alto viés?

**Resposta:**

- Não, afinal, para os dados de treinamento o modelo alcança praticamente um nível humano de performance.

**Pergunta:**

- O modelo encontra-se sobrestimando os dados de treinamento? Em outras palavras, temos um problema de alta variância?

**Resposta:**

- Podemos dizer que sim, afinal  $J_{cv}$  é significativamente maior que  $J_{train}$ .



No exemplo do slide anterior, nós usamos a **performance humana** como **performance de referência**. Entretanto, dependendo da aplicação, existem outras maneiras para se estabelecer tal referência, quais sejam:

- Comparar a performance do seu algoritmo com outros existentes na literatura
- Usar da sua experiência no assunto para definir um valor razoável de referência

Suponha as seguintes taxas de erro:

- $J_{train} = 16.8 \%$
- $J_{cv} = 17.2 \%$
- Performance de referência = 10.1 %

Suponha as seguintes taxas de erro:

- $J_{train} = 16.8 \%$
- $J_{cv} = 17.2 \%$
- Performance de referência = 10.1 %

**Pergunta:**

- O modelo encontra-se sobrestimando os dados de treinamento? Em outras palavras, temos um problema de alta variância?

**Resposta:**

- Não, afinal  $J_{cv}$  é próximo de  $J_{train}$ .

Suponha as seguintes taxas de erro:

- $J_{train} = 16.8 \%$
- $J_{cv} = 17.2 \%$
- Performance de referência = 10.1 %

**Pergunta:**

- O modelo encontra-se sobrestimando os dados de treinamento? Em outras palavras, temos um problema de alta variância?

**Resposta:**

- Não, afinal  $J_{cv}$  é próximo de  $J_{train}$ .

**Pergunta:**

- O modelo encontra-se subestimando os dados de treinamento? Em outras palavras, temos um problema de alto viés?

**Resposta:**

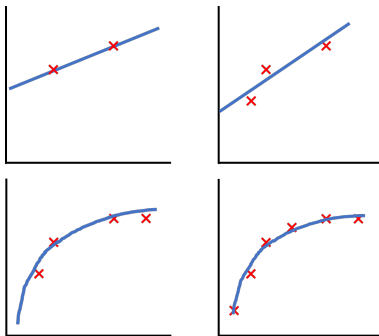
- Podemos dizer que sim, afinal, mesmo para os dados de treinamento o modelo não alcança um valor próximo ao de referência.

Já vimos como a complexidade do modelo pode afetar sua performance. Também fizemos tal análise com base na regularização. Agora buscaremos responder à seguinte pergunta:

**Como o número de amostras afeta o desempenho do seu modelo?**

# Como o número de amostras afeta o desempenho do seu modelo?

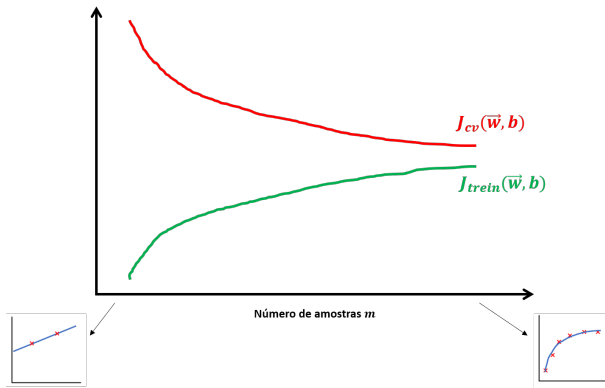
Suponha que você deseja ajustar um modelo do tipo  $f_{\vec{w}, b}(\vec{x}^{(i)}) = w_1x + w_2x^2 + b$  para os dados de treinamento abaixo, onde  $m$  varia.



Observação:

À medida com que mais amostras são utilizadas, o modelo vai conhecendo e se adequando melhor ao comportamento mais fundamental presente nos dados.  $J_{\text{train}}$  tende a aumentar um pouco, porém  $J_{\text{cv}}$  tende a cair.

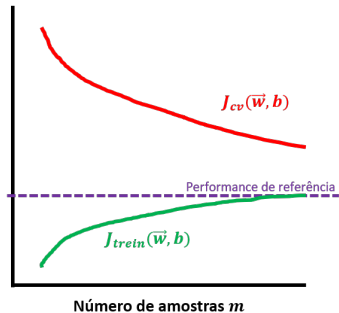
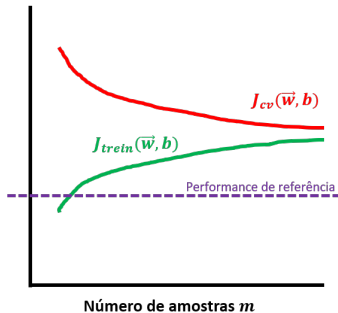
## Como o número de amostras afeta o desempenho do seu modelo?



Pergunta

Onde queremos estar nessa figura?

## Como o número de amostras afeta o desempenho do seu modelo?



### Perguntas

- Qual é o tipo de problema que está provavelmente acontecendo com o modelo da esquerda? Sobrestimação ou Subestimação?
- Qual é o tipo de problema que está provavelmente acontecendo com o modelo da direita? Sobrestimação ou Subestimação?
- Qual dos dois tipos de problema mais se beneficia do aumento do número de amostras?



**De olho no código!**

## De olho no código!

Iremos agora verificar como implementar na prática os conceitos vistos até aqui.

Acesse o Python Notebook usando o QR code ou o link abaixo:



[https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo\\_aula\\_21\\_Dicas\\_para\\_implementar\\_e\\_testar\\_metodos\\_e\\_modelos.ipynb](https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo_aula_21_Dicas_para_implementar_e_testar_metodos_e_modelos.ipynb)

## Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

## Parte 2

- 1 Compare as curvas obtidas por meio do código com as curvas teóricas apresentadas nos slides. Faça uma análise das diferenças e semelhanças.
- 2 Mostre que aumentar o número de amostras  $m$  tende a não resolver problemas de alto viés (subestimação). **Dica:** teste um modelo bem simples para os dados (por exemplo, uma reta).