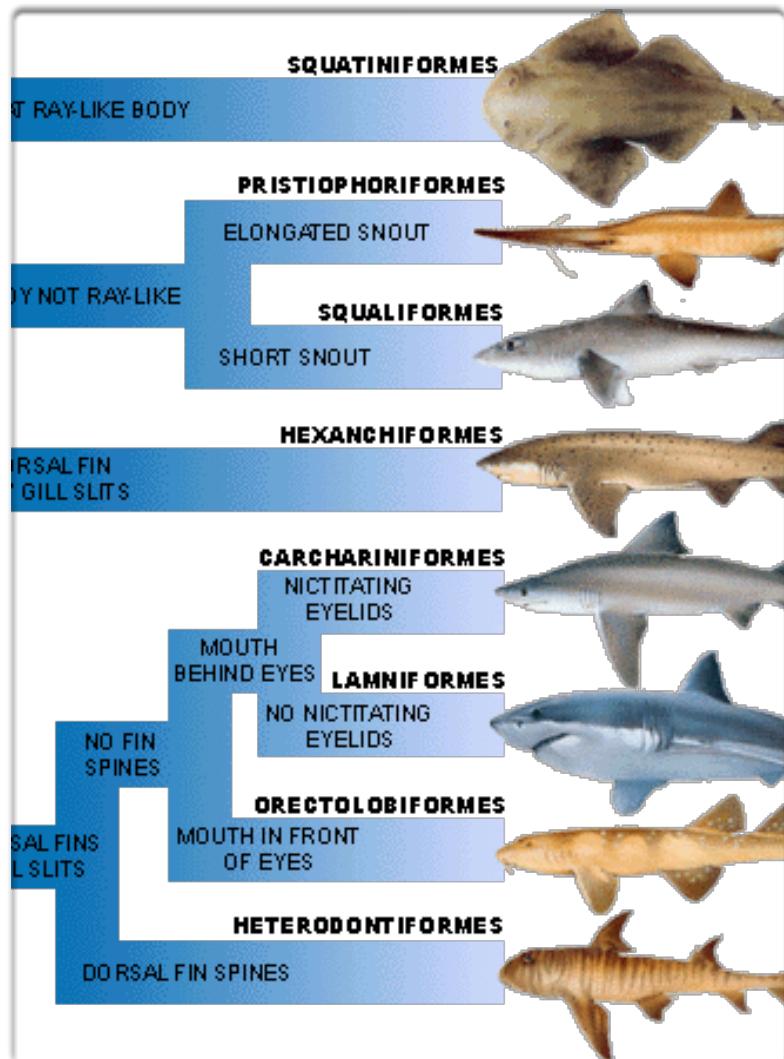


Introdução a Ciencia de Dados

Hitoshi Nagano, Ph.D.

MÉTRICAS DE PERFORMANCE



para classificação

Predições

	Gold	Pred
cliente0	0	0
cliente1	1	1
cliente2	0	0
cliente3	1	1
cliente4	1	0
cliente5	0	0
cliente6	0	0
cliente7	1	0
cliente8	1	1
cliente9	0	1

Matriz de Confusão

		Pred	
		0	1
Gold	0	0	
	1	1	

Predições

	Gold	Pred
cliente0	0	0
cliente1	1	1
cliente2	0	0
cliente3	1	1
cliente4	1	0
cliente5	0	0
cliente6	0	0
cliente7	1	0
cliente8	1	1
cliente9	0	1

Matriz de Confusão

		Pred	
		0	1
Gold	0	verdadeiro negativo	falso positivo
	1	falso negativo	verdadeiro positivo

Predições

	Gold	Pred
cliente0	0	0
cliente1	1	1
cliente2	0	0
cliente3	1	1
cliente4	1	0
cliente5	0	0
cliente6	0	0
cliente7	1	0
cliente8	1	1
cliente9	0	1

Matriz de Confusão

		Pred	
		0	1
Gold	0	0	4
	1	1	3

Matriz de Confusão e métricas

	Gold	Pred
cliente0	0	0
cliente1	1	1
cliente2	0	0
cliente3	1	1
cliente4	1	0
cliente5	0	0
cliente6	0	0
cliente7	1	0
cliente8	1	1
cliente9	0	1

$$\text{acurácia} = \frac{\text{labels corretos}}{\text{todas obs}}$$

$$\text{precisão} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos Positivos}}$$

$$\text{recall} = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos Negativos}}$$

$$\text{f1-score} = \frac{2}{\frac{1}{\text{precisão}} + \frac{1}{\text{recall}}}$$



para regressão

Erro Quadrático Médio

	Gold	Pred	Erro ao quad.
cliente0	3	3.2	0.04
cliente1	1	0.1	0.81
cliente2	-2	-1.9	0.01
cliente3	0.3	-0.1	0.16
cliente4	0.6	0.7	0.01
MSE			0.21

$$MSE = \frac{1}{n} \sum_{i=0}^{n-1} (\hat{y}^{(i)} - y^{(i)})^2$$

Erro Quadrático Médio

	Gold	Pred	Erro ao quad.
cliente0	3	3.2	0.04
cliente1	1	0.1	0.81
cliente2	-2	-1.9	0.01
cliente3	0.3	-0.1	0.16
cliente4	0.6	0.7	0.01
MSE			0.21

$$MSE = \frac{1}{n} \sum_{i=0}^{n-1} (\hat{y}^{(i)} - y^{(i)})^2$$

mean square error

$$MSE = \frac{1}{n} \sum_{i=0}^{n-1} (\hat{y}^{(i)} - y^{(i)})^2$$

root mean square error

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\hat{y}^{(i)} - y^{(i)})^2}$$

$$\bar{y} = \frac{1}{n} \sum_{i=0}^{n-1} y^{(i)}$$



$$SS_{res} = \sum_{i=0}^{n-1} (\hat{y}^{(i)} - y^{(i)})^2$$

$$SS_{tot} = \sum_{i=0}^{n-1} (y^{(i)} - \bar{y})^2$$



*coefficient of
determination*

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

```
labels:  
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 ]  
pred:  
[ 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 ]
```

Esse accuracy é alto, vários '0' foram previstos como '0'. Mas a precisão é zero. De fato, não há verdadeiro positivos, ou seja, não há nenhum que '1' que foi corretamente previsto.

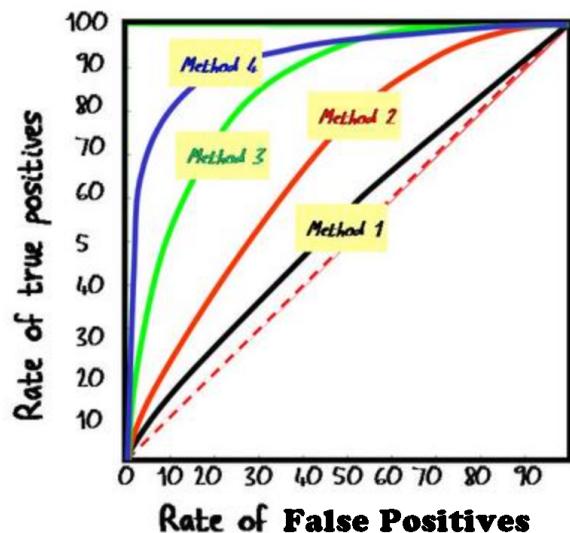
Isso acontece porque a quantidade de '0' é muito grande, muito maior que a de '1'. Assim, existe uma boa chance dos '0' terem sido classificados como '0' somente porque a sua quantidade no label dos dados de teste é alta, e não porque o algoritmo é bom.

Dizemos que a distribuição é desbalanceada (muitos 0 e poucos 1). E nesses casos, accuracy não é uma boa métrica.

Importante: Muitas vezes a meta é identificar 1's. Identificar 0's é um objetivo menor.

- precision, recall e F1 dependem de um limiar de decisão...
- ... então, qual limiar escolher?
- Existe uma métrica que independentemente de um limiar?

ROC CURVE EXAMPLES



- The best classification has the largest area under the curve.
- Too sensitive to errors in the "gold standard" classification.

FPR =

$$\frac{\text{Falsos positivos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}}$$

condição negativa

TPR =

$$\frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}}$$

condição positiva

AREA UNDER THE ROC CURVE

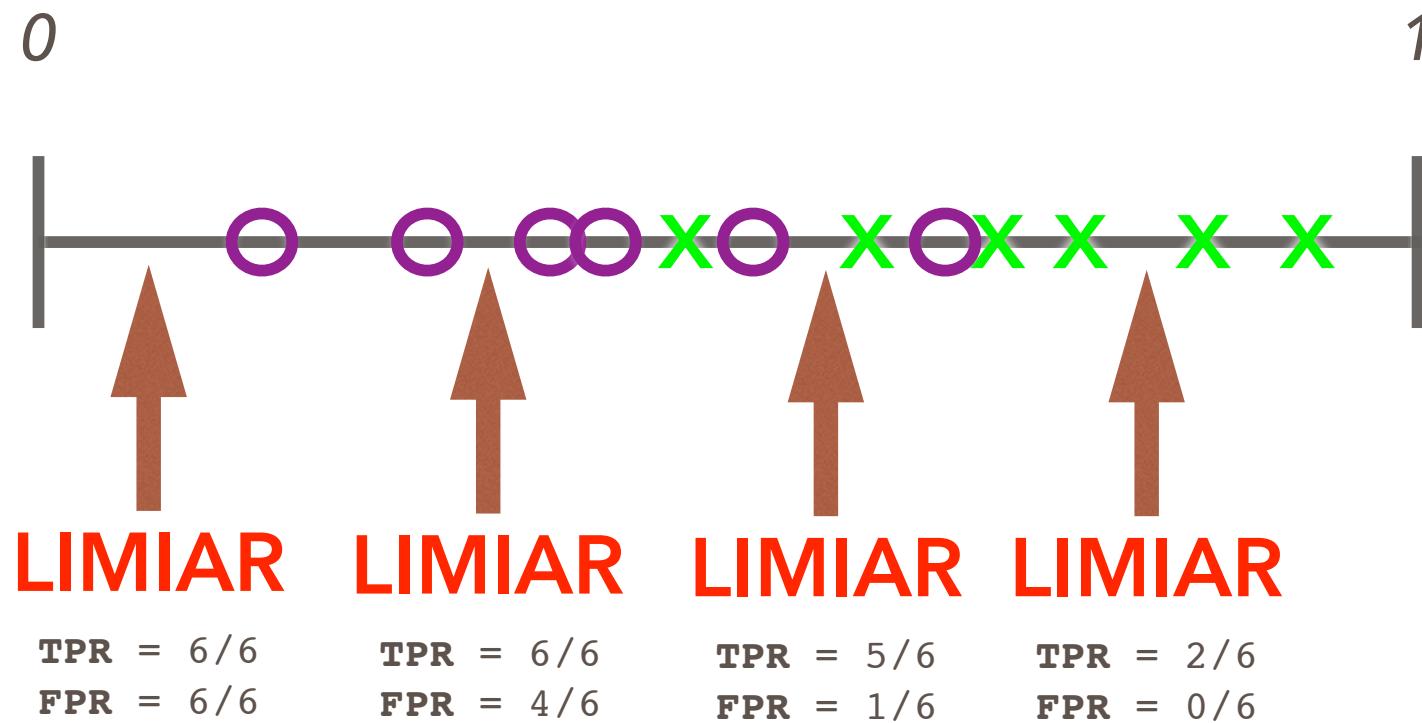
AUC ou AUC_ROC

$$FPR = \frac{\text{Falsos positivos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}}$$

$$TPR = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}}$$

O condição negativa

X condição positiva



AREA UNDER THE ROC CURVE

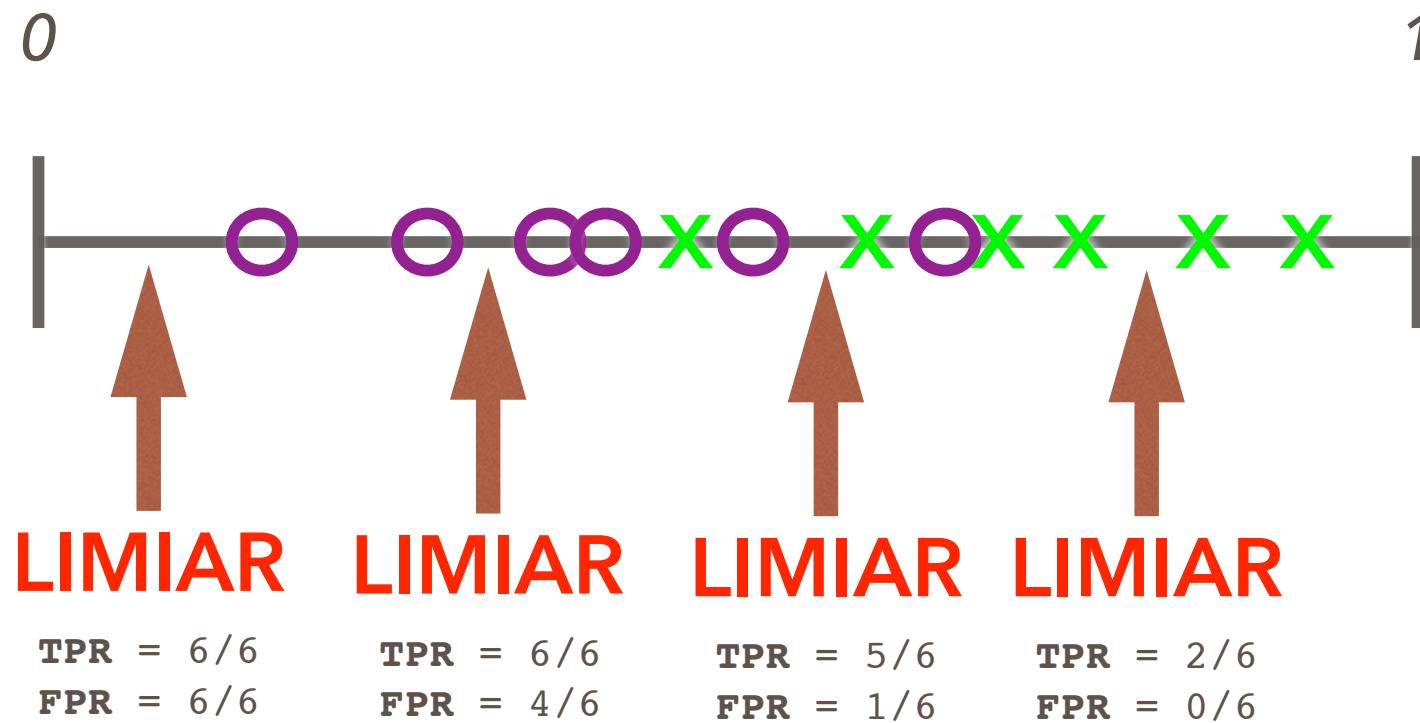
AUC ou AUC_ROC

$$FPR = \frac{\text{Falsos positivos}}{\text{Verdadeiros negativos} + \text{Falsos positivos}}$$

$$TPR = \frac{\text{Verdadeiros positivos}}{\text{Verdadeiros positivos} + \text{Falsos negativos}}$$

O condição negativa

X condição positiva

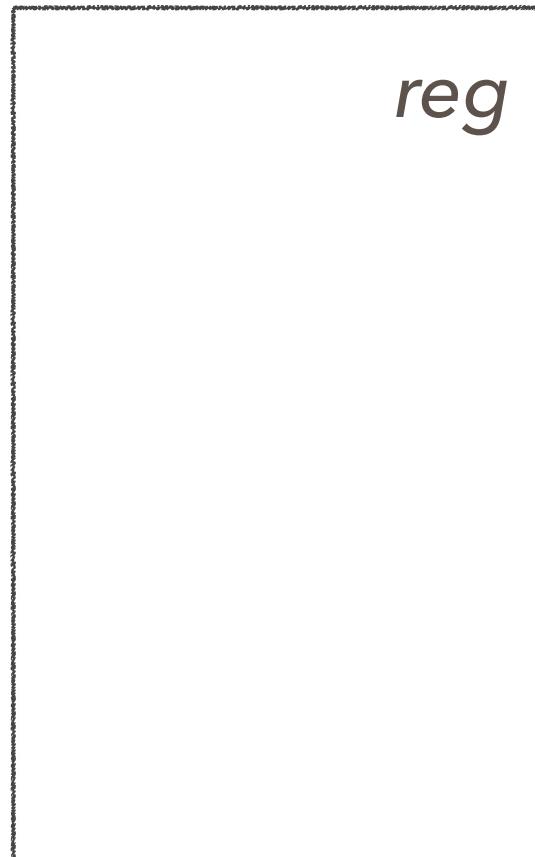


TREINAMENTO E PREDIÇÃO COM SKLEARN

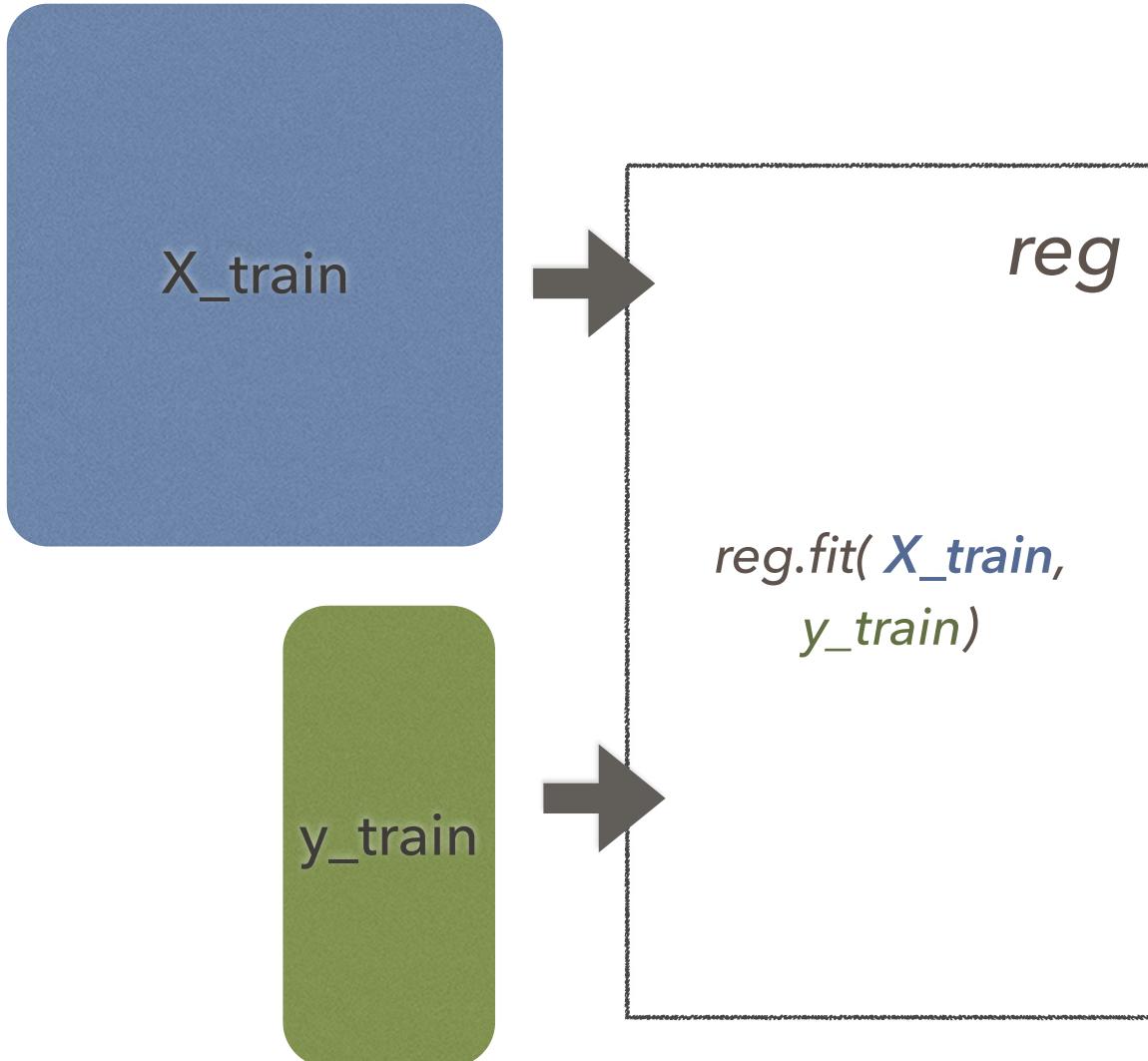


1) criar o objeto

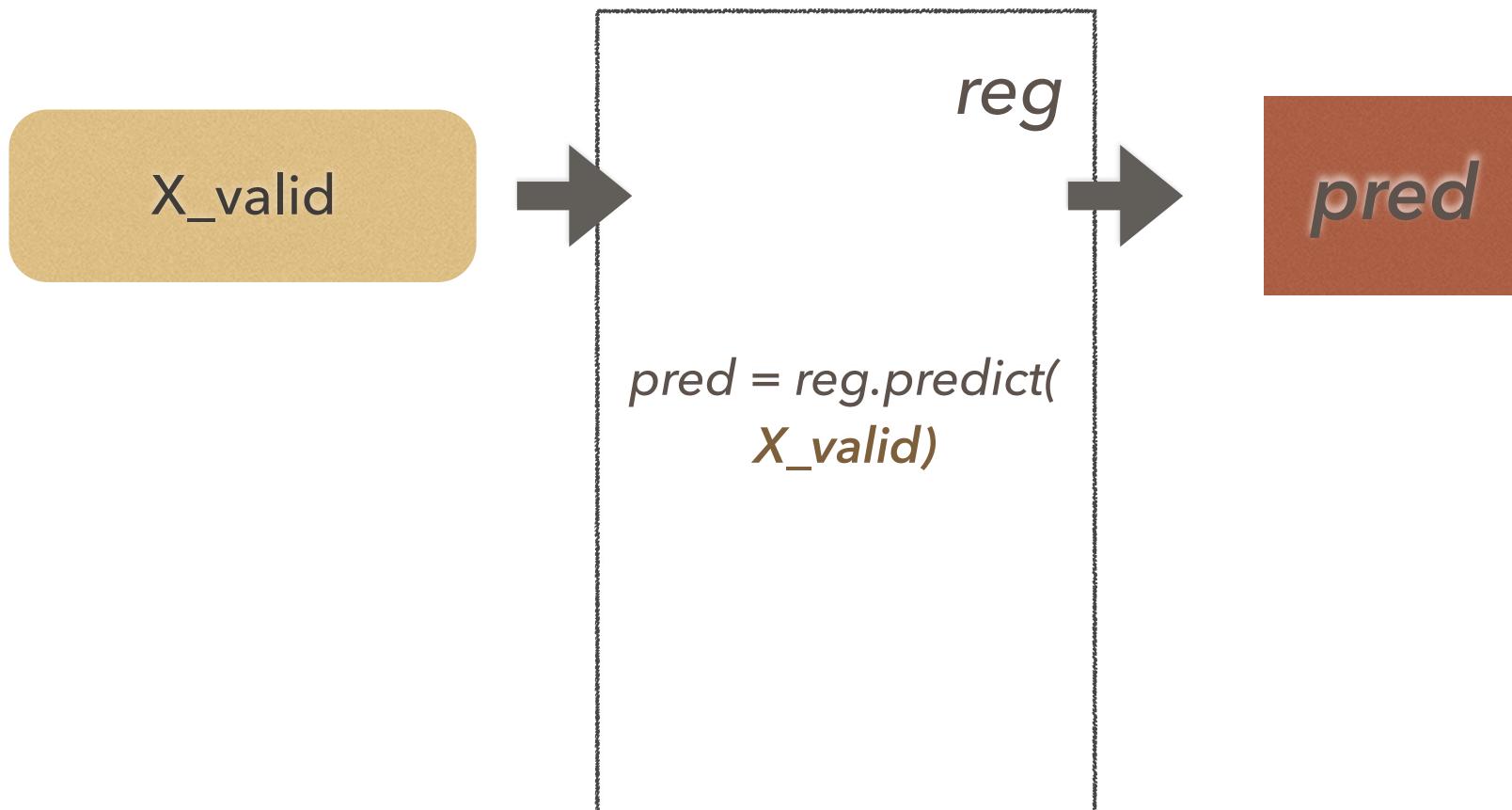
reg = LinearRegression()



2) treinar (fit)



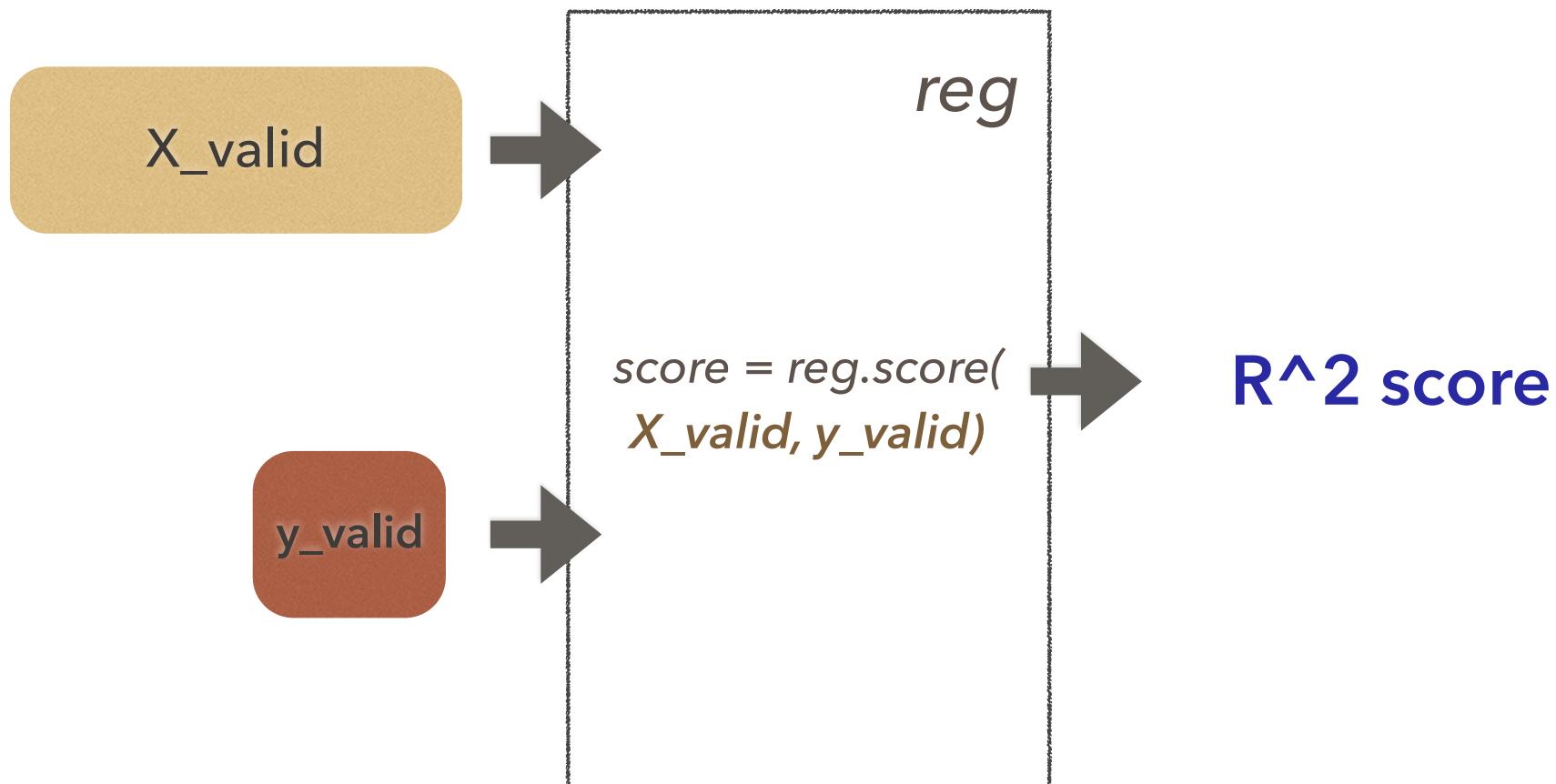
3) predizer (predict)



Objetivo: quanto menor o valor, melhor o modelo

$$\text{mean}(\text{pred} - \text{y_valid})^2$$

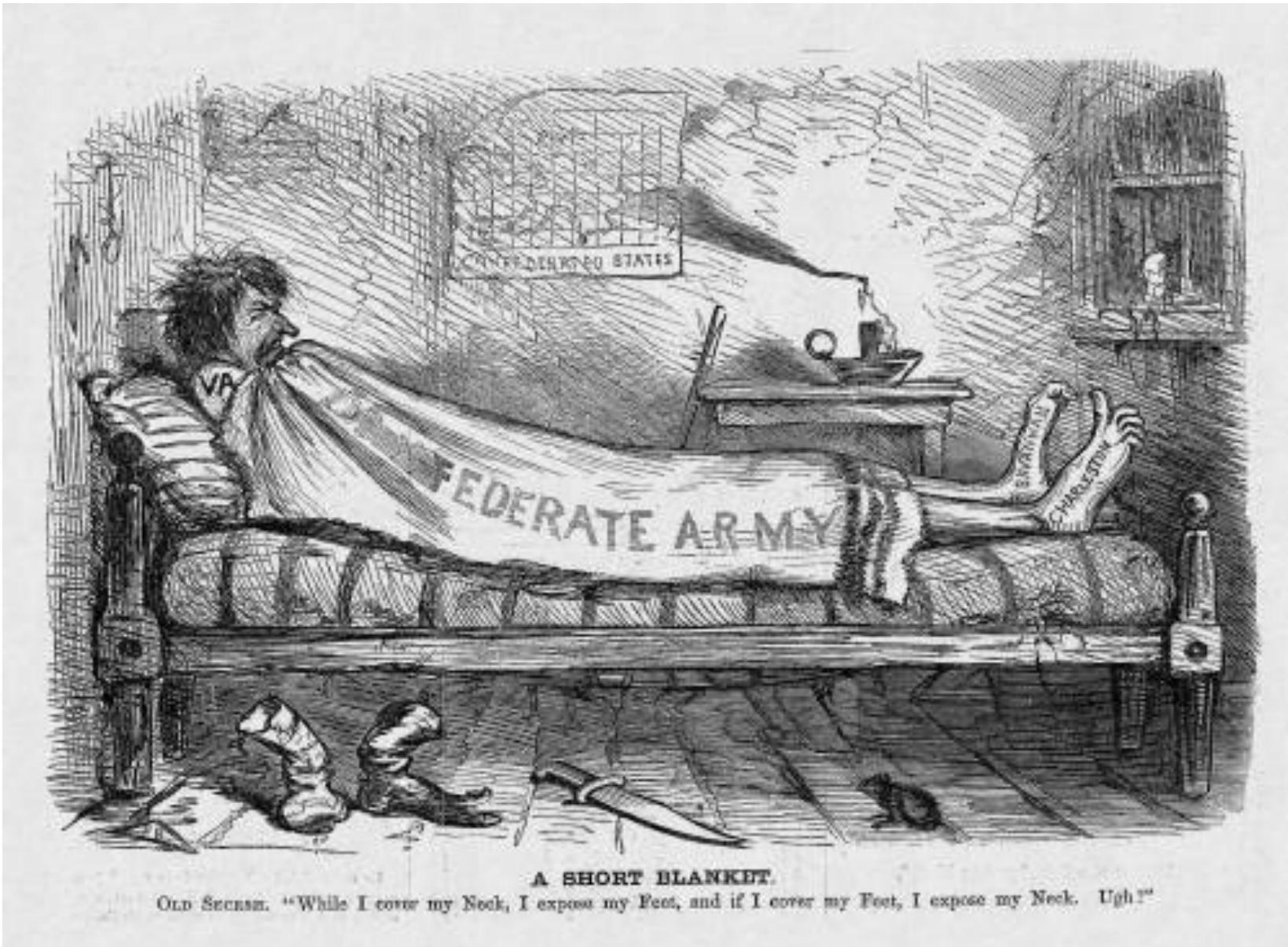
4) validando o modelo (score)



DIVIDINDO OS DADOS

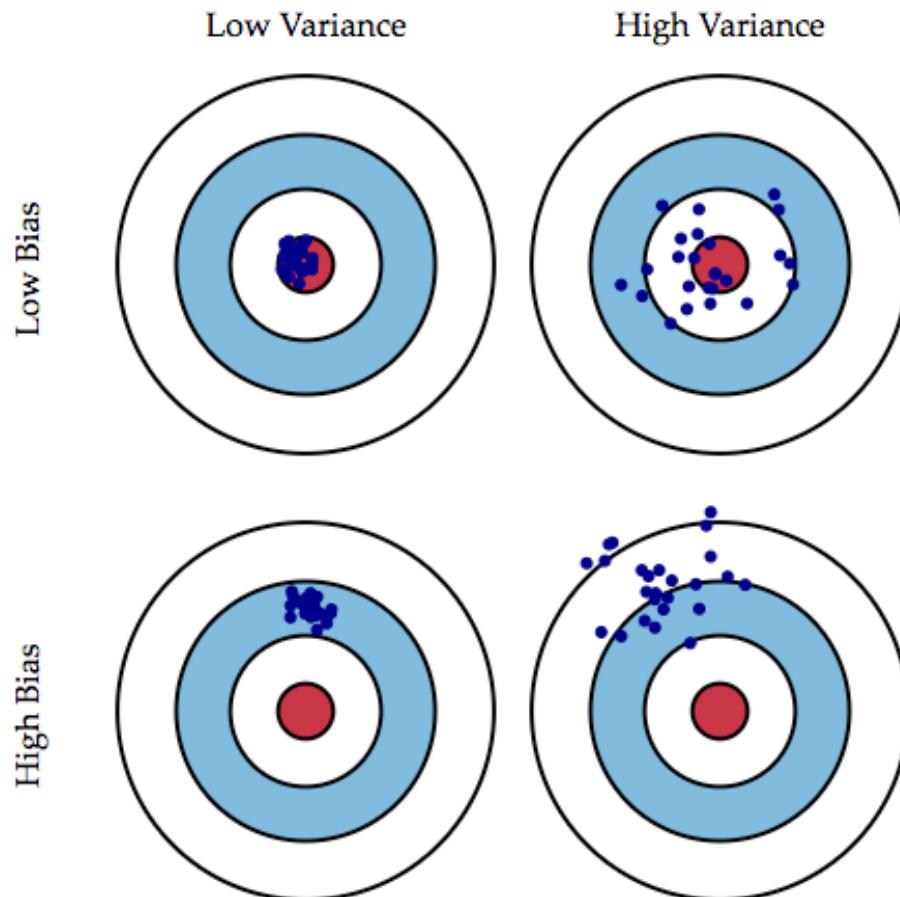


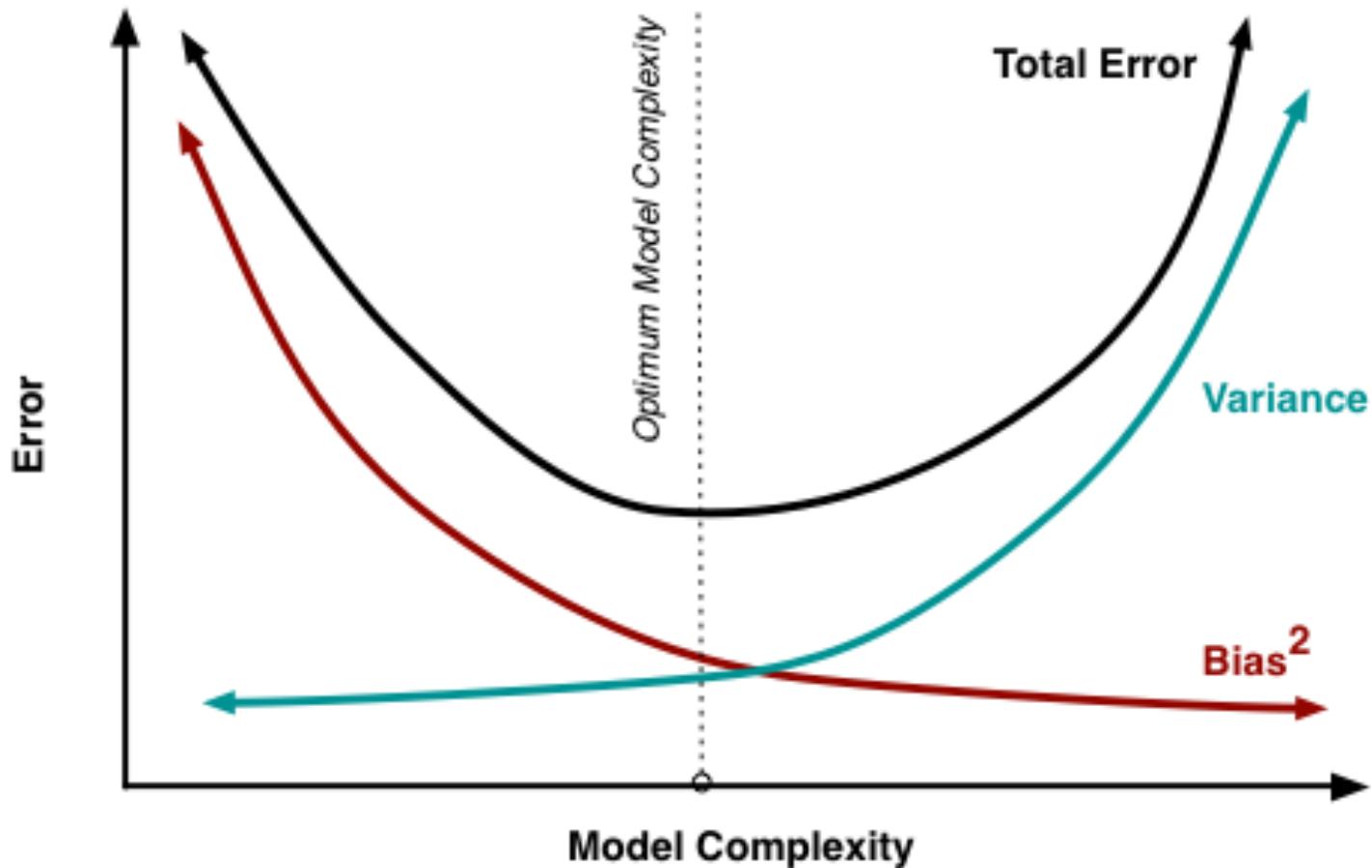
A Short Blanket



BIAS VARIANCE DILEMMA

FGV MANAGEMENT





bias alto



pouca atenção aos dados
oversimplified
erros altos no treinamento
normalmente poucas features

variancia alta



muita atenção aos dados (não generaliza)
overfitting
erros mais altos nos testes do que no treinamento
normalmente muitas features,
tentativas seguidas de minimizar SSE

objetivo



sweet-spot: qtd adequada features
regularização

X_train

y_train

X_valid

y_valid

SEPARANDO OS DADOS

X_train

y_train

X_valid

y_valid

SEPARANDO OS DADOS

X_train

y_train

X_valid

y_valid

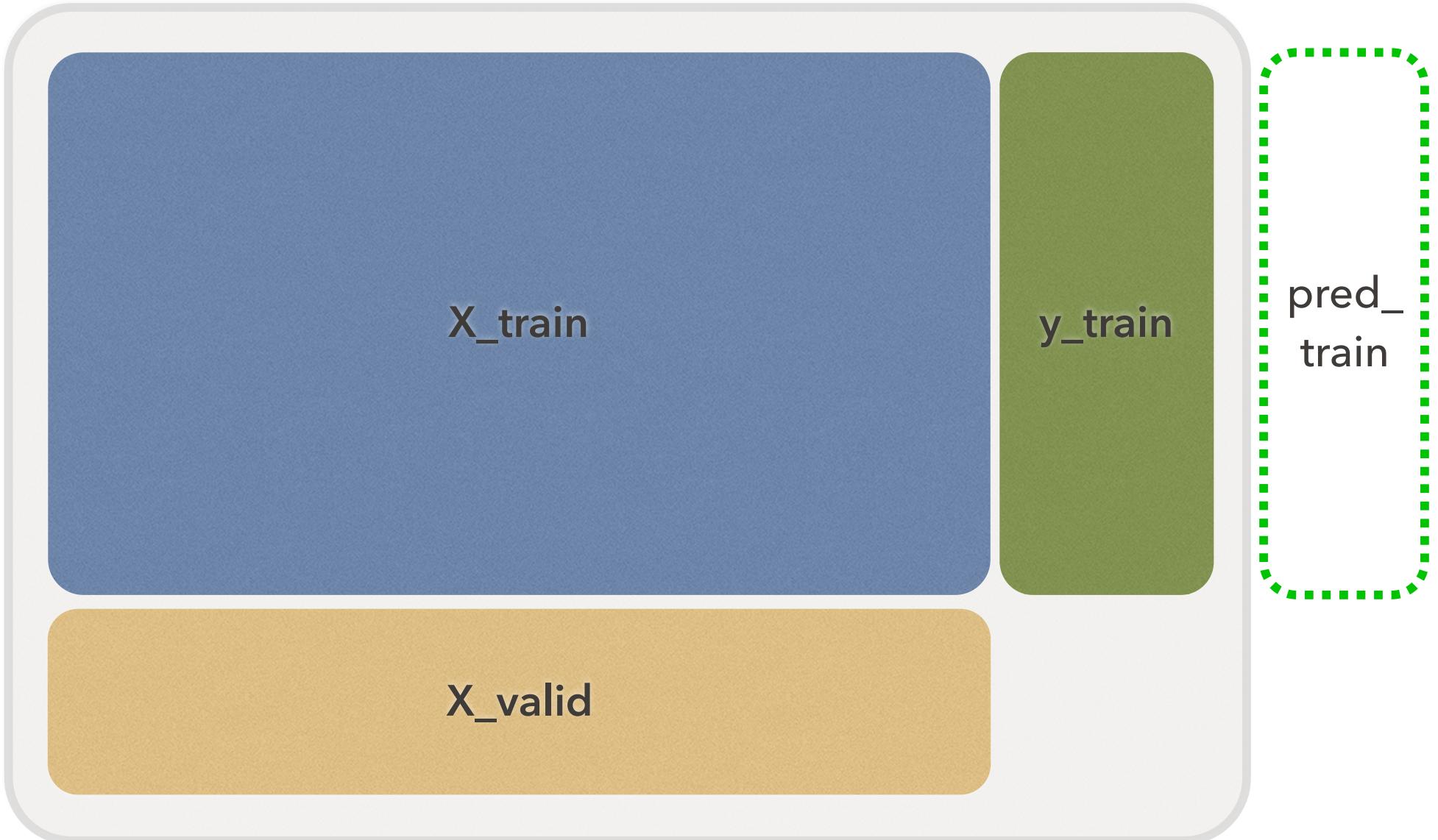
SEPARANDO OS DADOS

X_train

y_train

X_valid

SEPARANDO OS DADOS



SEPARANDO OS DADOS

X_train

y_train

X_valid

 pred

Objetivo: quanto menor o valor, melhor o modelo

$$\text{mean}(\hat{\text{pred}} - \text{y_valid})^2$$

X_train

y_train

X_valid

y_valid

EM VEZ DE...

X_train

y_train

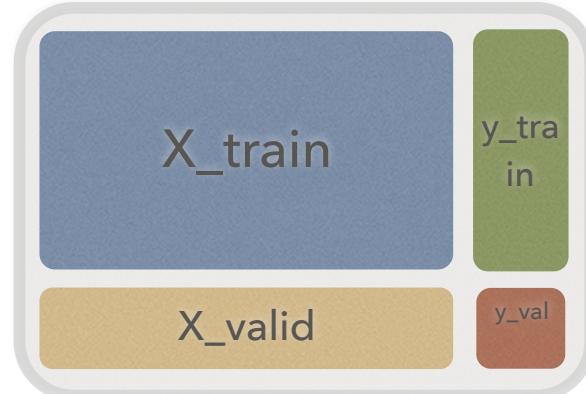
X_valid

y_valid

KFOLD: k splits

para $k = 3$

Fold # 1



Fold # 2



Fold # 3

