

**MBA  
USP  
ESALQ**

**ÁRVORES, REDES E ENSEMBLE  
MODELS II**

João F. Serrajordia R. de Mello

\*A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados é do professor.

**Proibida a reprodução, total ou parcial, sem autorização. Lei nº 9610/98**

Você vai precisar de...



## Preparativos

- Abrir o R
- Importar as bibliotecas
- Algo para fazer suas anotações

# Recapitulando

rapidamente

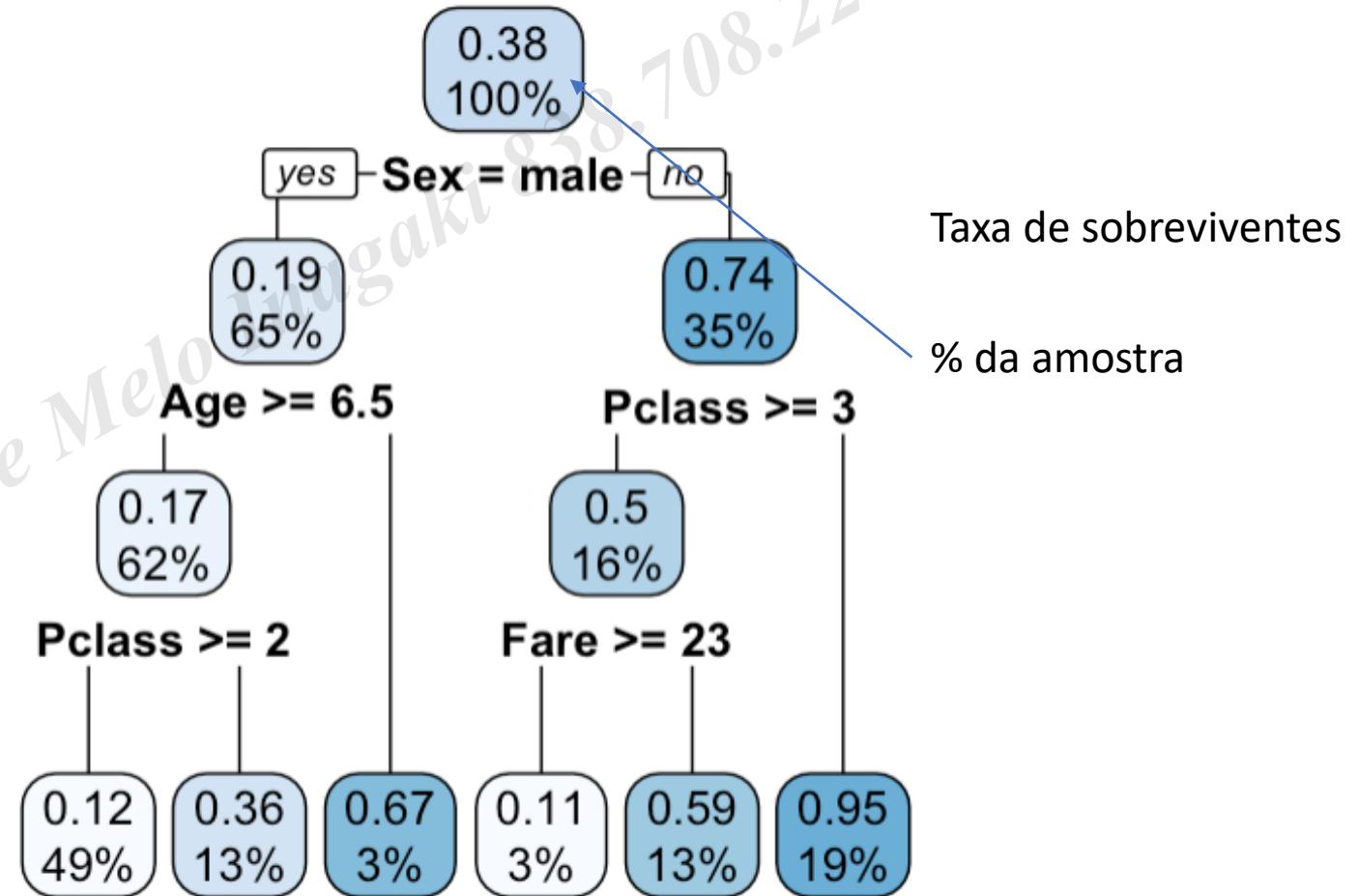
# Vimos conceitos...



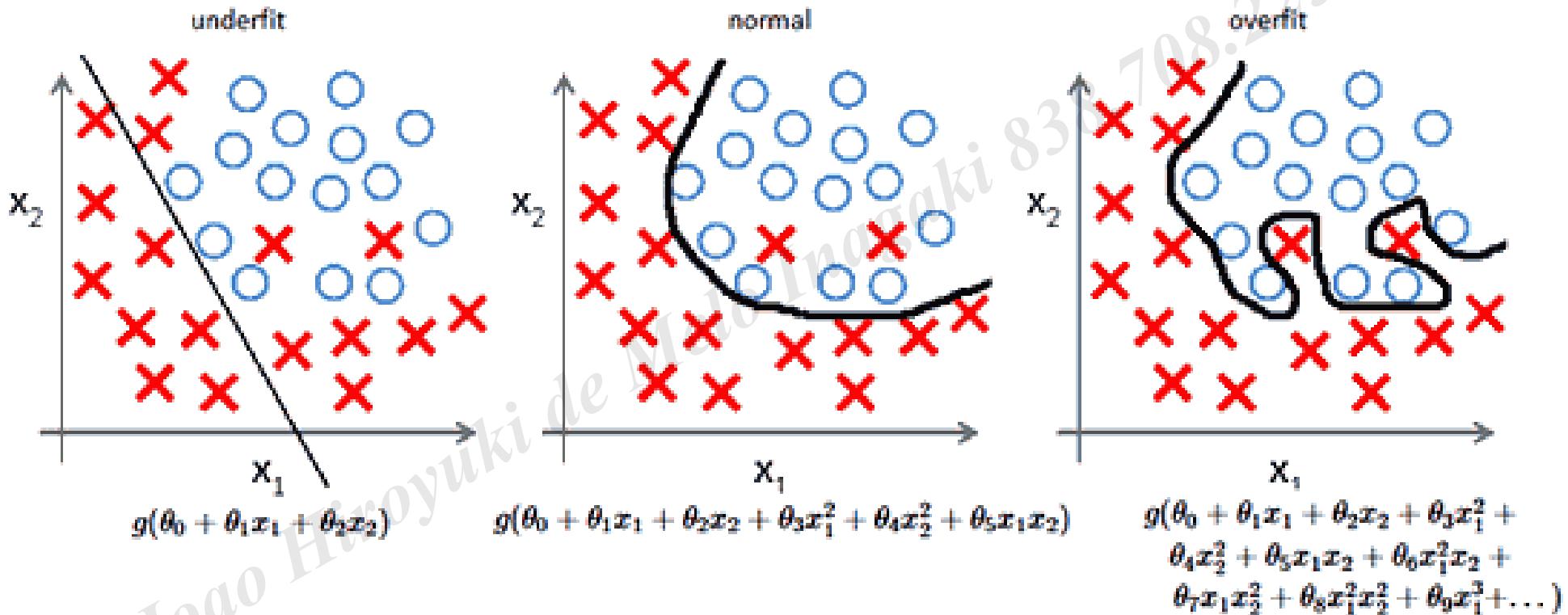
- Machine learning
- IA
- Paradigma machinelârnico
- Nomenclaturas

# Vimos o conceito de árvores

Vamos classificar este rico senhor:

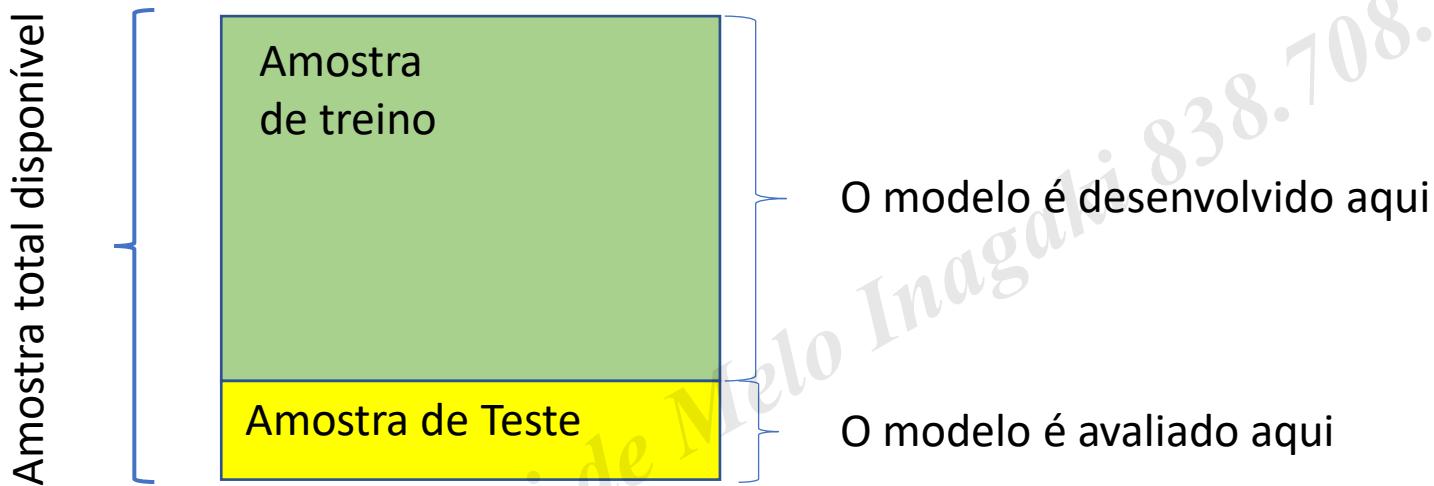


# Falamos sobre overfitting



<http://mlwiki.org/index.php/Overfitting>

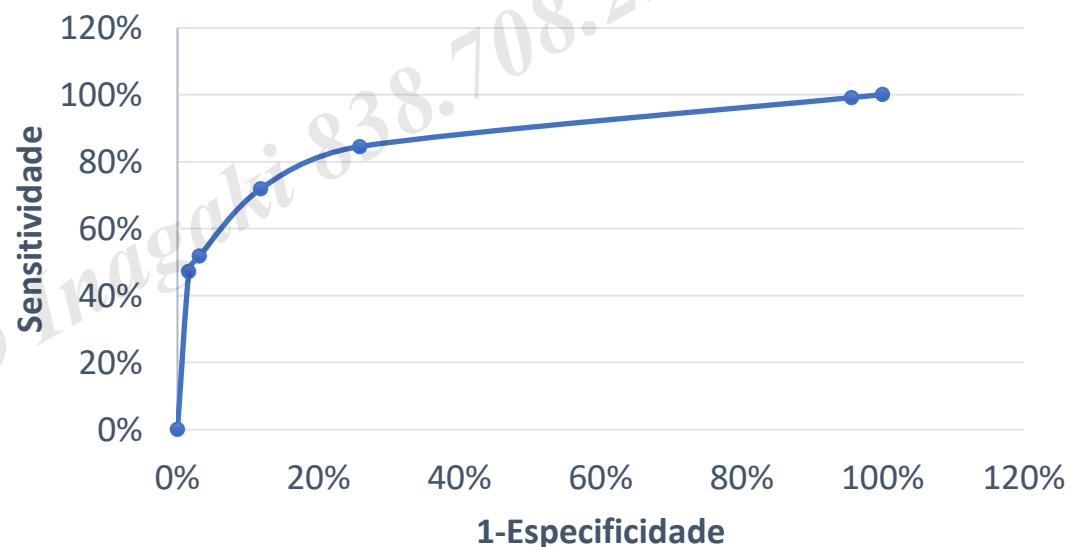
# Vimos Cross validation



A estratégia mais simples é dividir a base em treino e teste.  
Desenvolvemos o modelo na base de treino e avaliamos na base de teste.

# Avaliamos a árvore

Corte	1-Especificidade	Sensibilidade
0% - 11,1%	100%	100%
11,1% - 11,5%	96%	99%
11,5% - 35,8%	26%	85%
35,8% - 58,9%	12%	72%
58,9% - 66,7%	3%	52%
66,7% - 94,7%	2%	47%
94,7% - 100%	0%	0%



A curva ROC é um gráfico de dispersão de 1-Especificidade no eixo x por Sensibilidade no eixo y, obtidos para cada possível ponto de corte do classificador.

# Exercício

E tá lembrado da  
atividade??

Bora lá??

Camón, dongle?



AREM2\_exercicios\_para\_casa\_da\_aula\_anterior.R

# Agenda



Exercícios

Árvores de regressão

Técnicas de validação cruzada

*Bagging – Random Forest*

*Boosting – Gradient Boosting*

*Grid Search CV*



## Árvores de regressão

\*A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados é do professor.  
Proibida a reprodução, total ou parcial, sem autorização. Lei nº 9610/98

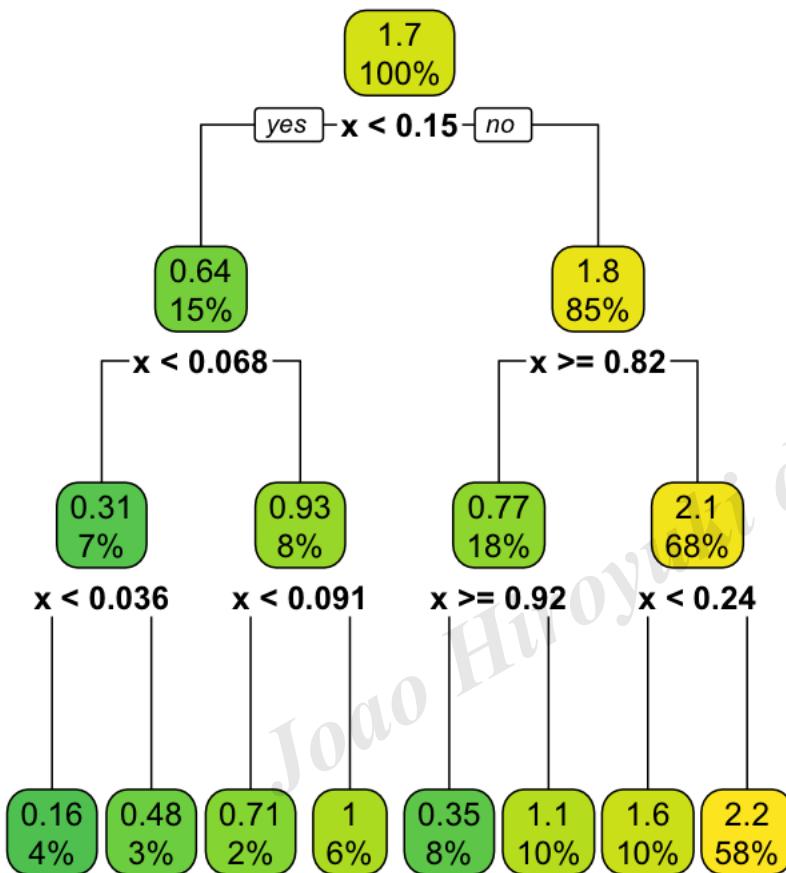
# Árvores de regressão

São muito semelhantes a árvores de classificação

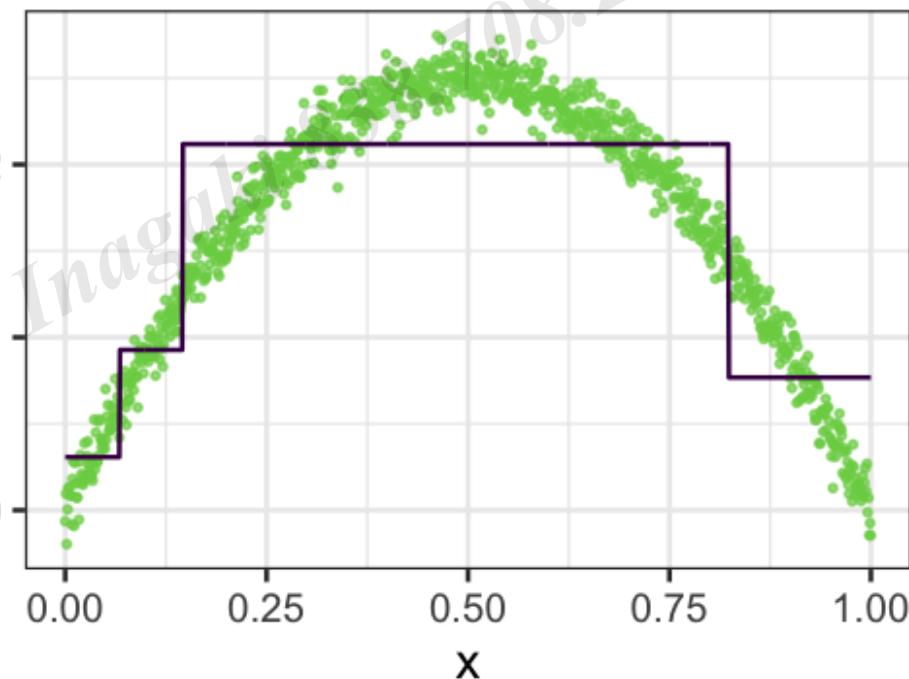
O que muda é o critério de impureza

$$SQE = \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

# Árvores de regressão



Valores observados vs esperados



Dado: — Esperado — Observado

# E a impureza?

$$SSE = \sum_{i} (y_{observed} - y_{predicted})^2$$

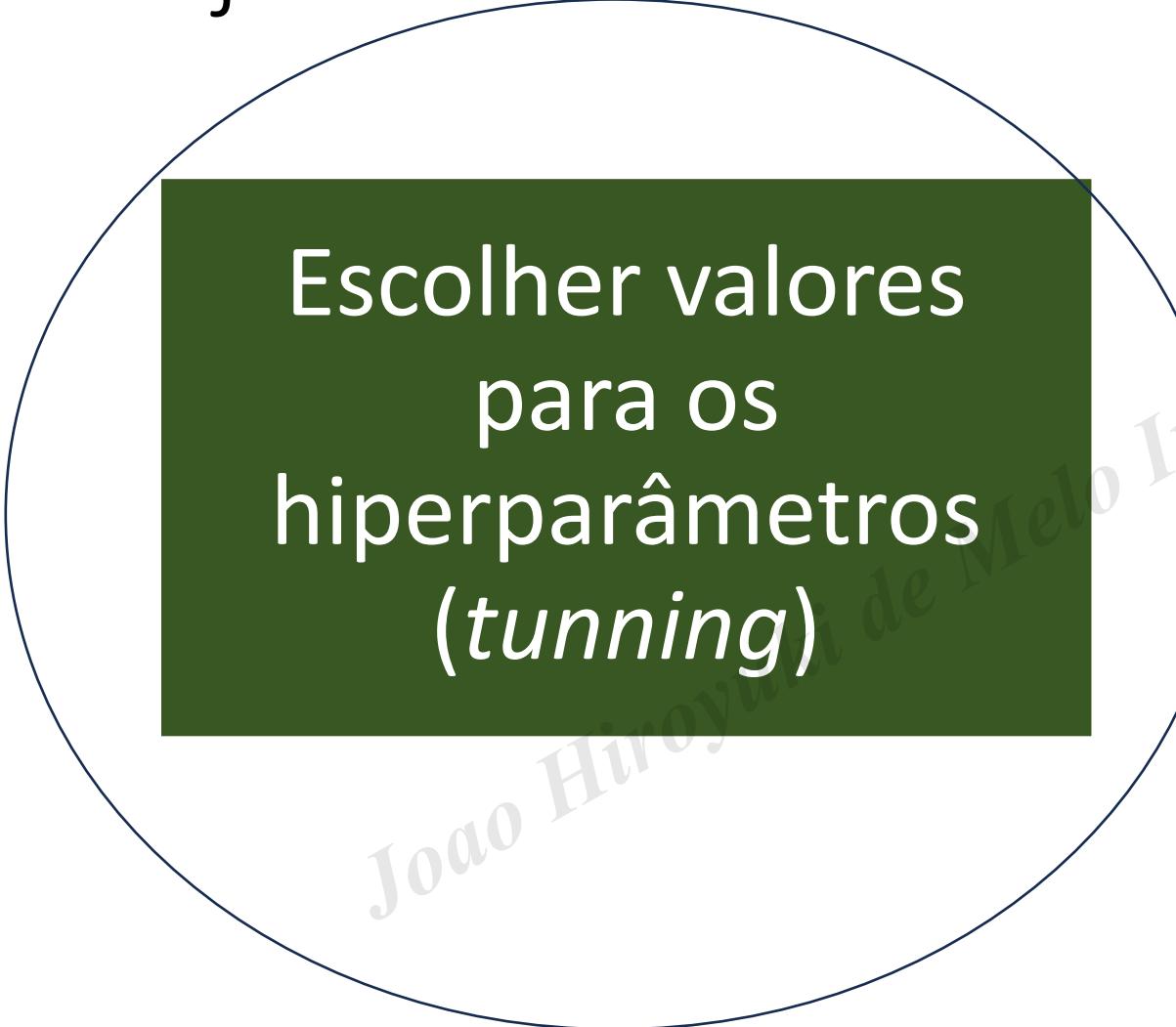


AREM2\_script01\_Regression\_Tree.R

# Cross validation



# Objetivos



Escolher valores  
para os  
hiperparâmetros  
*(tunning)*

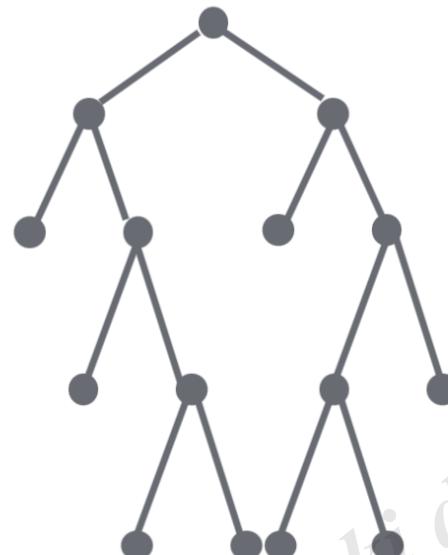
Ter expectativa  
melhor da  
qualidade (e.g. AUC)  
em outras bases

# Ideia básica

1. Dividir a base em treino e teste
2. Rodar o modelo com uma configuração de hiperparâmetros na base de treino
3. Avaliar o modelo na base de testes
4. Alterar a configuração e repetir os passos 2 e 3

Será que uma escolha particular da base de treino e teste pode favorecer uma determinada configuração?

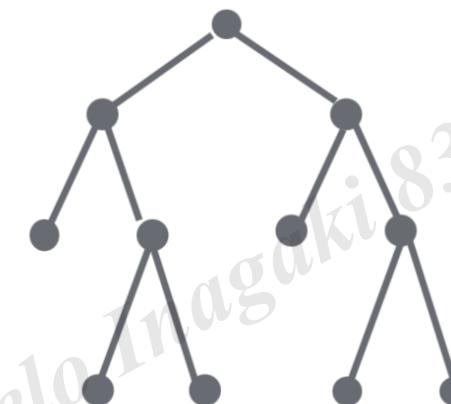
# Poda da árvore (*Pruning*)



Acurácia

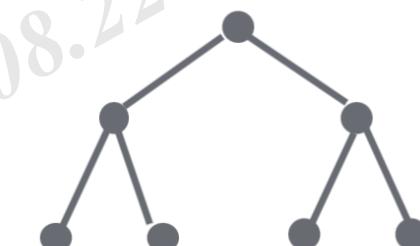
Base de treino: 95%  
Base de validação: 40%

Amostra de treino



Base de treino: 70%  
Base de validação: 60%

Amostra de validação



Base de treino: 65%  
Base de validação: 64%

# Estratégias de cross validation

Escolher parâmetros do modelo com uma base de validação ainda pode propiciar overfitting.

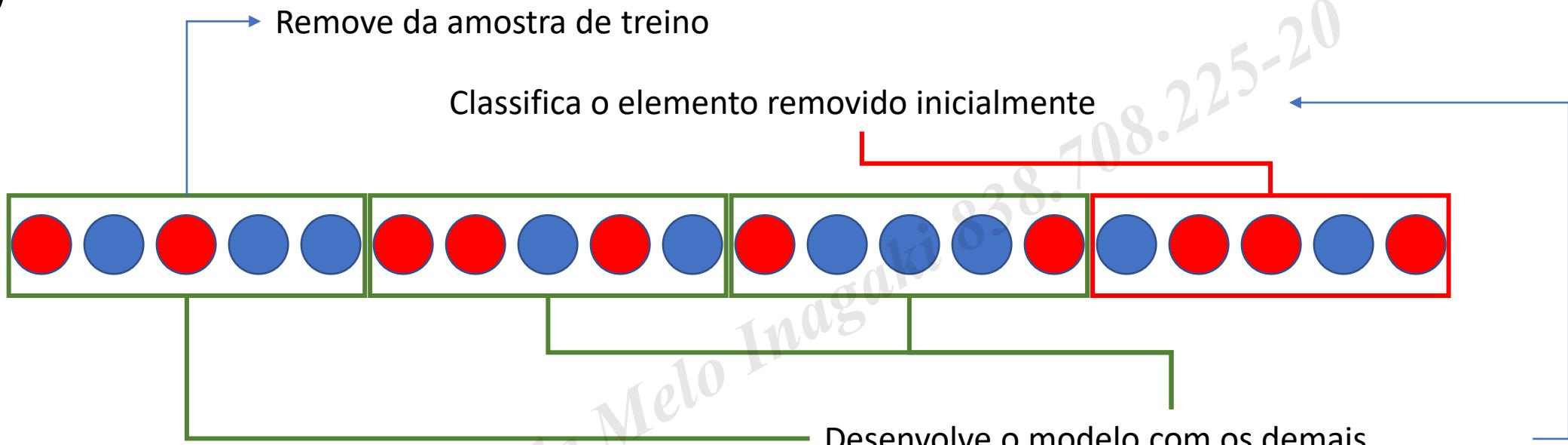
Há diversas técnicas de validação cruzada para se evitar esse efeito. No momento vou mencionar uma técnica clássica: dividir a base em Treino, Validação e Teste

Amostra de treino

Amostra de validação

Amostra de teste

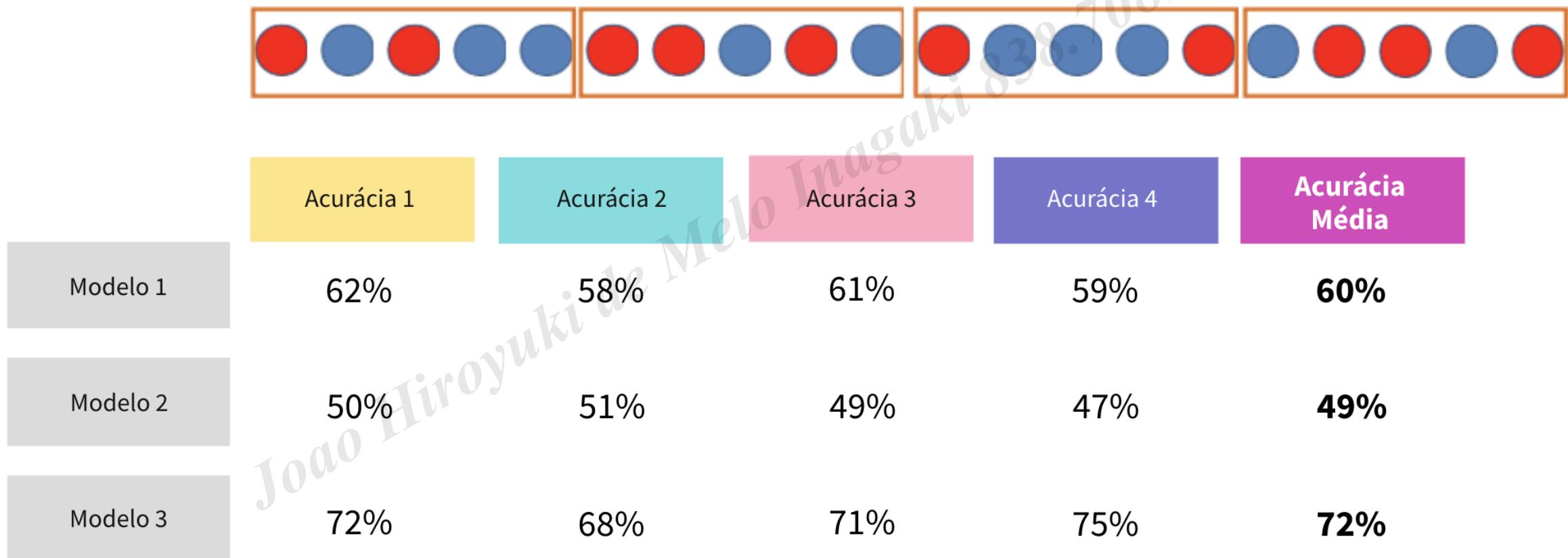
# K-fold



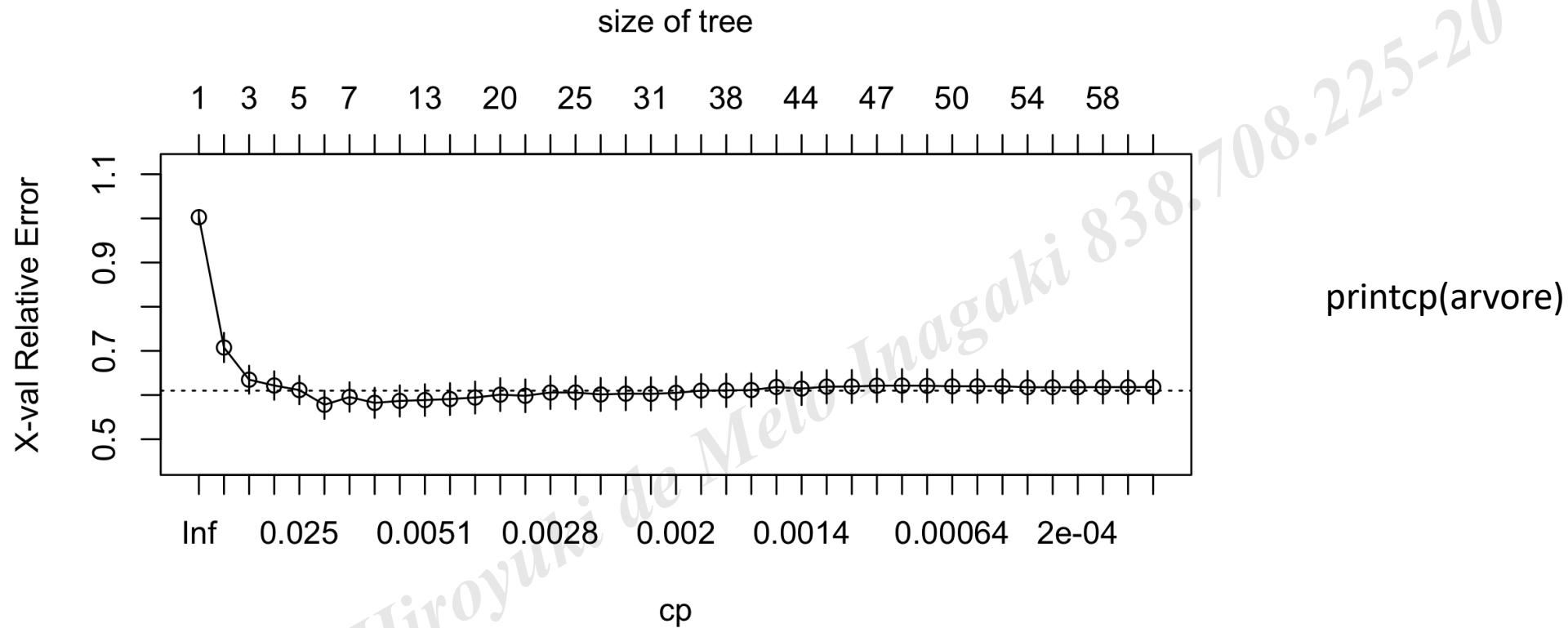
- Dividimos a base em k sub-amostras
- Para cada sub-amostra:
  - Removemos a sub-amostra como validação
  - Treinamos o modelo com as observações restantes
  - Utilizamos este modelo para classificar a sub-amostra removida
  - Avaliamos a métrica de desempenho do modelo
- Calculamos a média das métricas de desempenho do modelo

# *K-fold*

**Tipicamente, fazemos o mesmo para variações do modelo para otimizar hiperparâmetros.**



# Post-prunning com crossvalidation

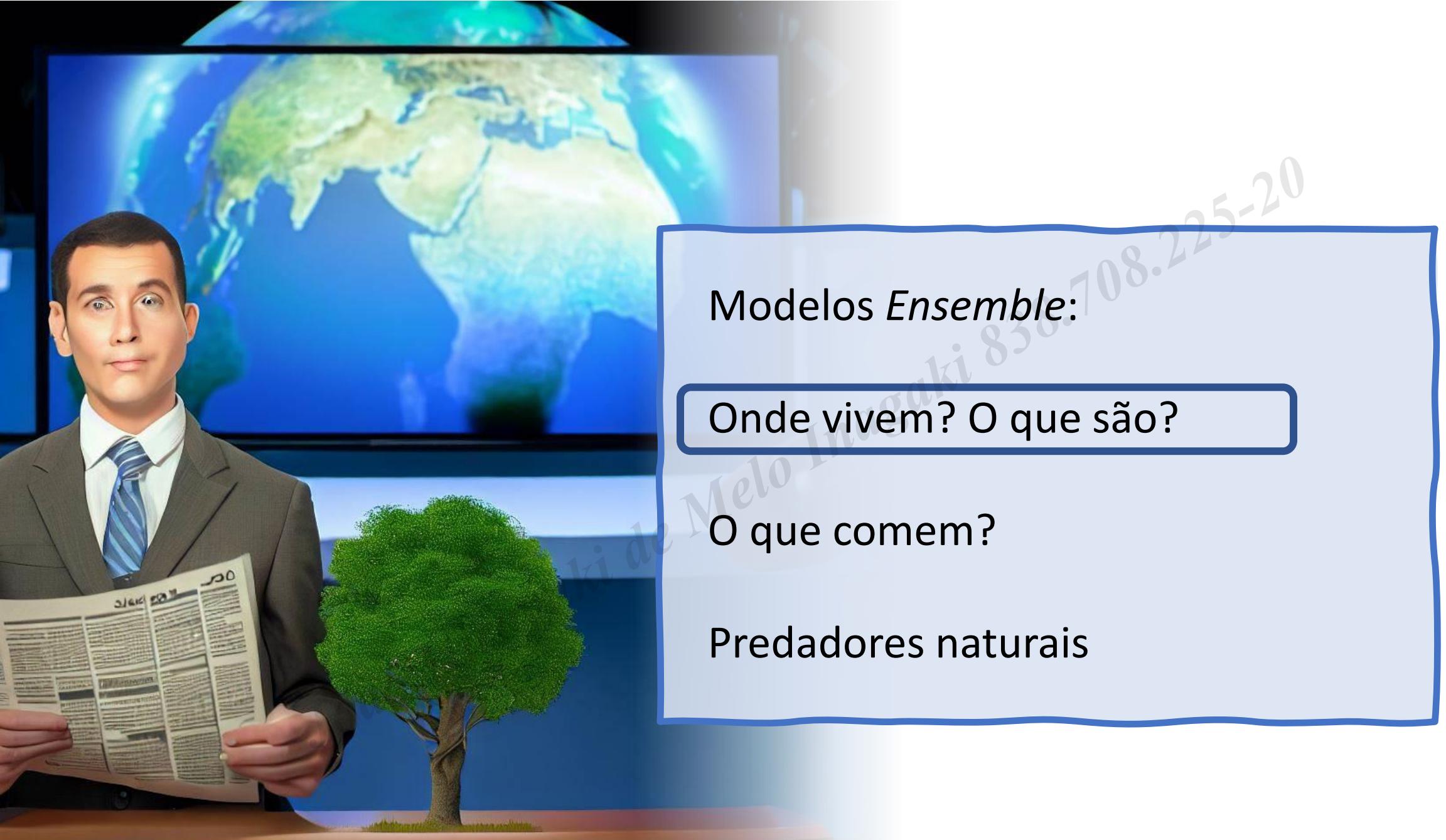


O pacote rpart faz a poda da árvore realizando um k-fold para otimizar o CP (complexity path), um parâmetro que summariza a complexidade da árvore. Isso é feito com um *k-fold*.

# K-fold no *rpart*

```
14 arvore <- rpart::rpart(Survived ~  
15   Pclass +  
16   Sex + Age +  
17   SibSp +  
18   Parch +  
19   Fare +  
20   Embarked,  
21   data=treino,  
22   method='class',  
23   xval=5,  
24   control = rpart.control(cp = 0,  
25     minsplit = 5,  
26     maxdepth = 30)
```

O “K” do K-fold



Modelos *Ensemble*:

Onde vivem? O que são?

O que comem?

Predadores naturais

# Problemas de preditivos e de classificação



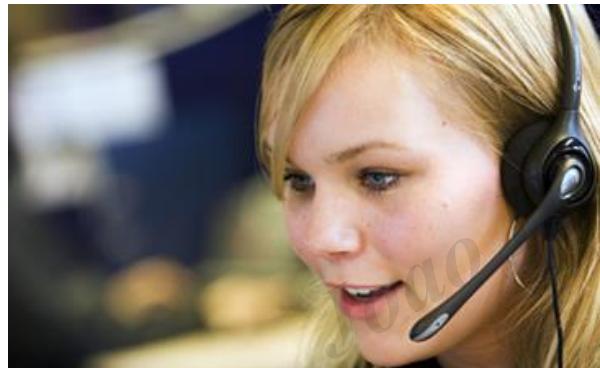
Qual a eficácia de uma vacina?



O cliente vai pagar o empréstimo?



Quanto de petróleo tem no poço?



O cliente vai comprar meu produto?

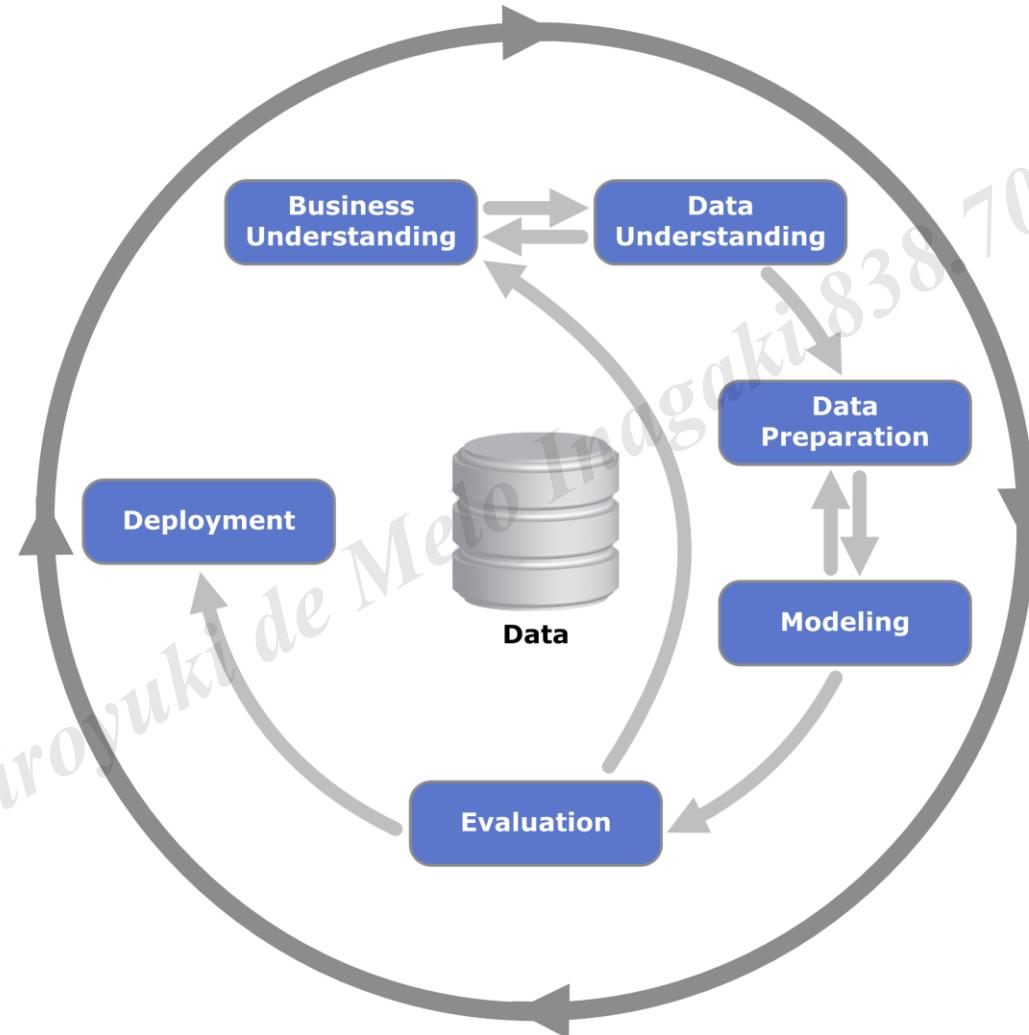


O que a pessoa está fazendo?



Quão ecológico esse veículo é?

# CRISP-DM



Fonte: <https://www.the-modeling-agency.com/crisp-dm.pdf>

\*A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados é do professor.  
Proibida a reprodução, total ou parcial, sem autorização. Lei nº 9610/98

# Classificação dos algoritmos



## Supervisionados

- Regressão
- GLM
- GLMM
- Support vector machines
- Naive Bayes
- K-nearest neighbors
- Redes Neurais
- Decision Trees



## Não supervisionados

- K-Means
- Métodos hierárquicos
- Mistura Gaussiana
- DBScan
- Mini-Batch-K-Means

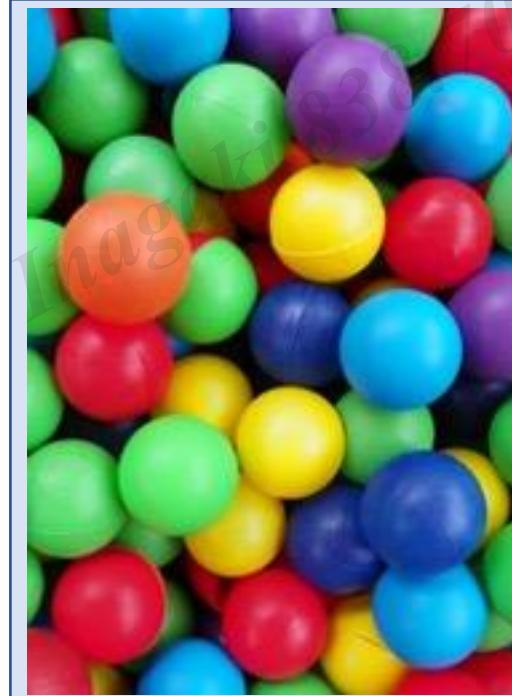
Estamos aqui!

# Classificação dos algoritmos



## Algoritmos de regressão

- Regressão
- GLM
- GLMM
- Support vector machines
- K-nearest neighbors
- Redes Neurais
- Regression Trees

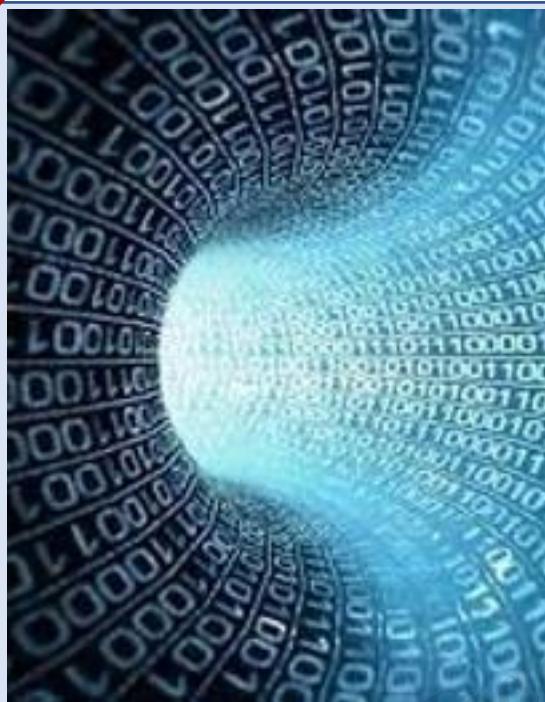


## Algoritmos de classificação

- Regressão logística
- Classification trees
- Redes Neurais
- GLM
- GLMM

Estamos aqui!

# Classificação dos algoritmos



## Paradigma machine learning

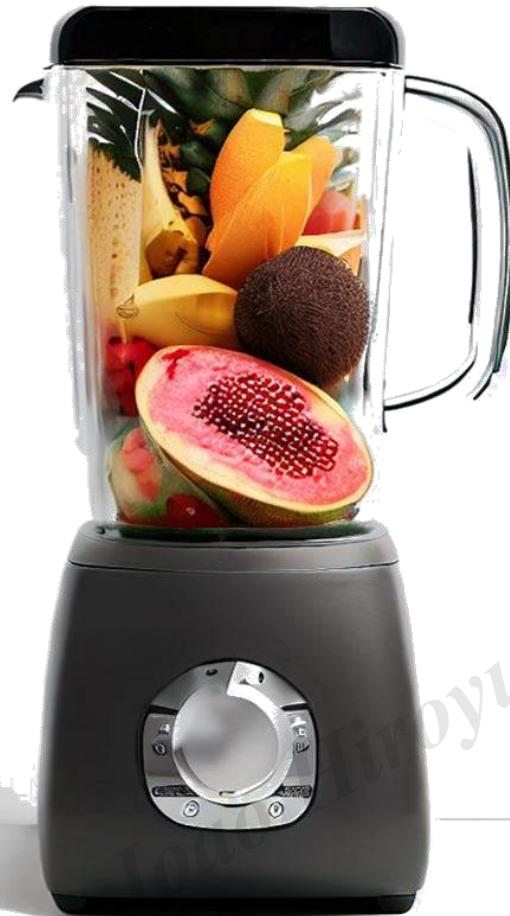
- Árvores de decisão
- Bagging
- Boosting
- K-NN
- Redes Neurais
- Support Vector Machines



## Paradigma estatístico

- Regressão
- GLM
- GLMM
- ANOVA

Estamos aqui!



# Ensemble

Um ensemble é qualquer mistura de modelos já existentes.  
Os principais tipos são:

*Bagging*

*Boosting*

*Stacking*

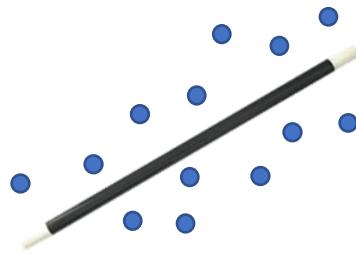
# *Bagging é um aggregation*

Mas o que é um aggregation oras...

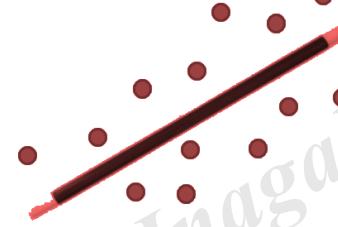
# *Ensemble - aggregation*



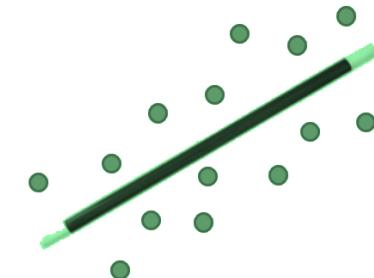
Modelo 1



Modelo 2



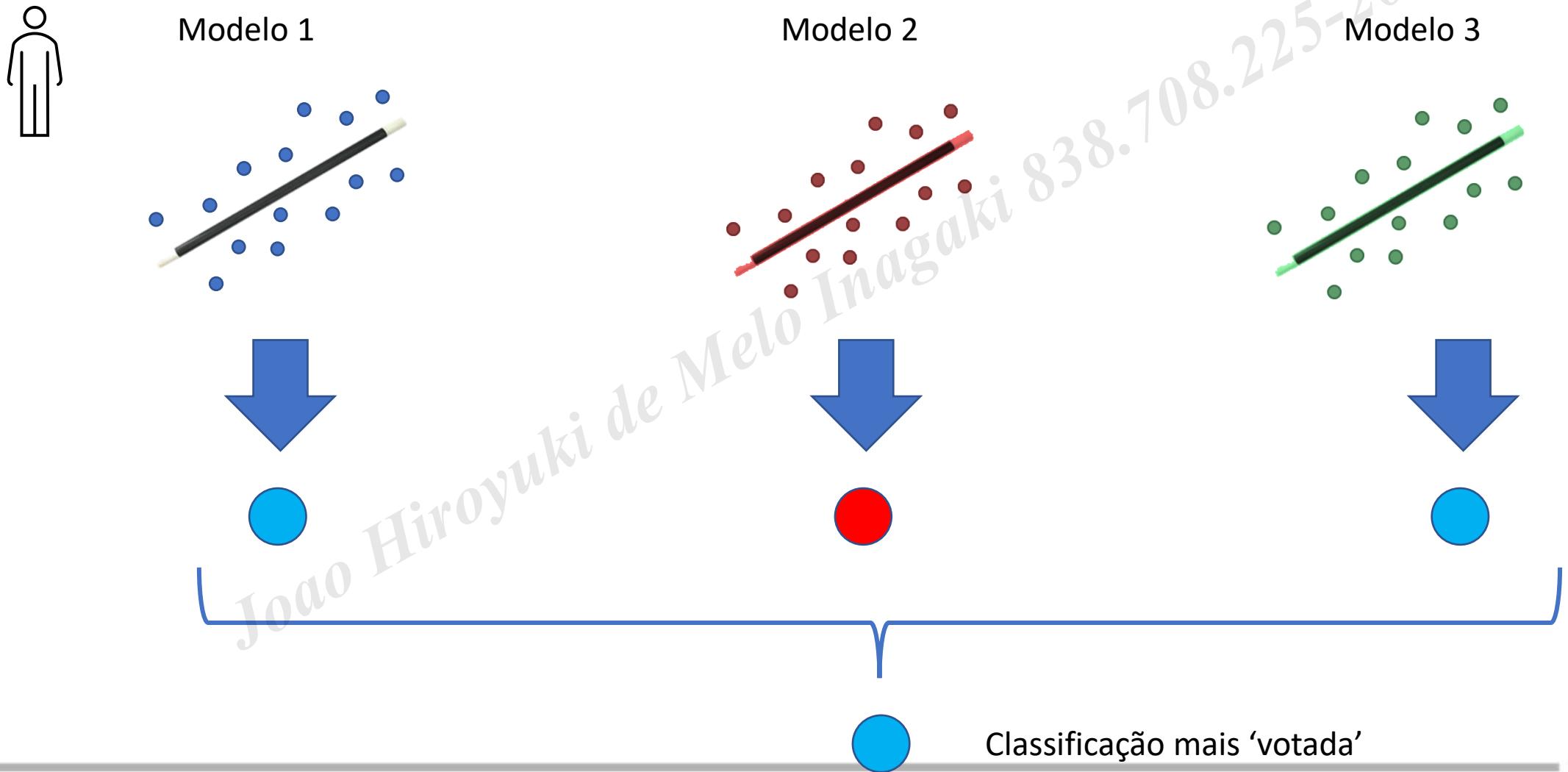
Modelo 3



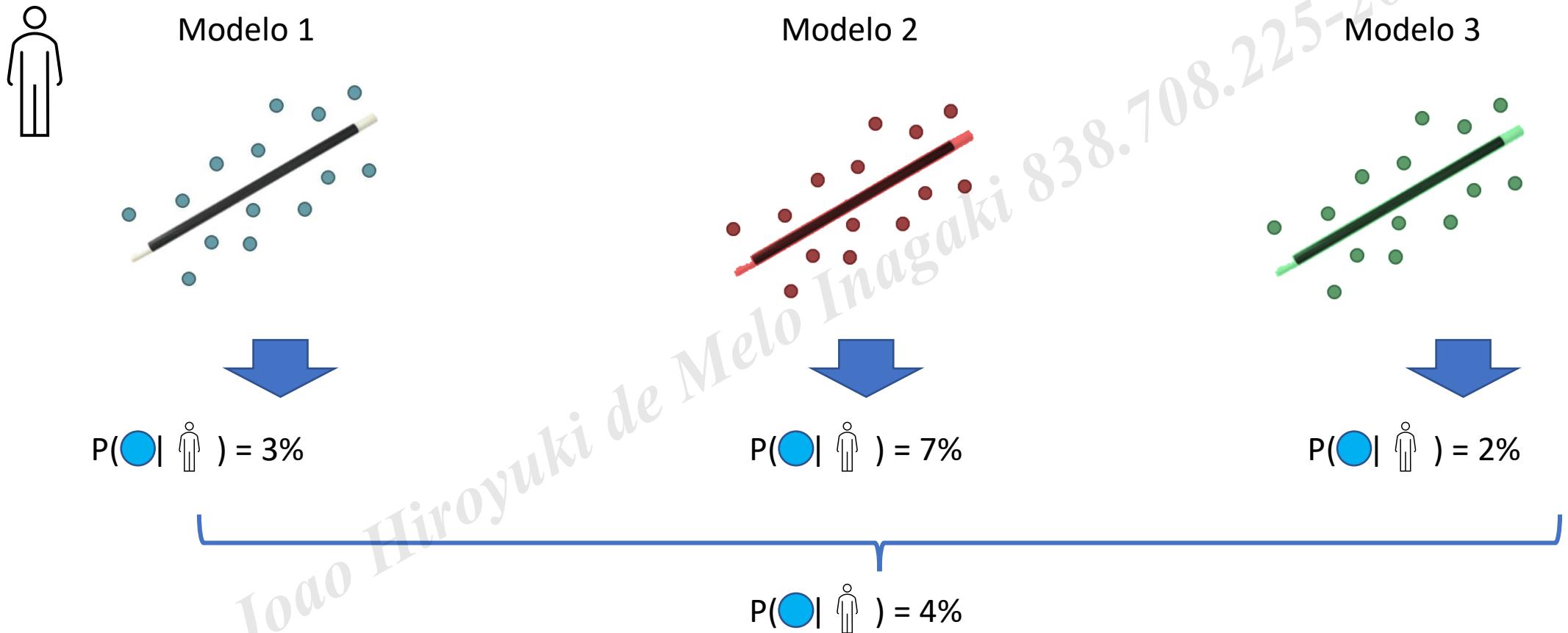
Um *aggregation* consiste em uma combinação (em geral uma média simples) das previsões de dois ou mais modelos previamente construídos.

Objetivo: ainda que cada modelo seja um “*weak learner*”, a combinação pode ser um “*Strong learner*” ou um preditor melhor que cada um dos integrantes.

# *Ensemble – Hard Voting*



# *Ensemble - aggregation*



Um método de agregação simples mas poderoso consiste em obter a média de várias previsões.

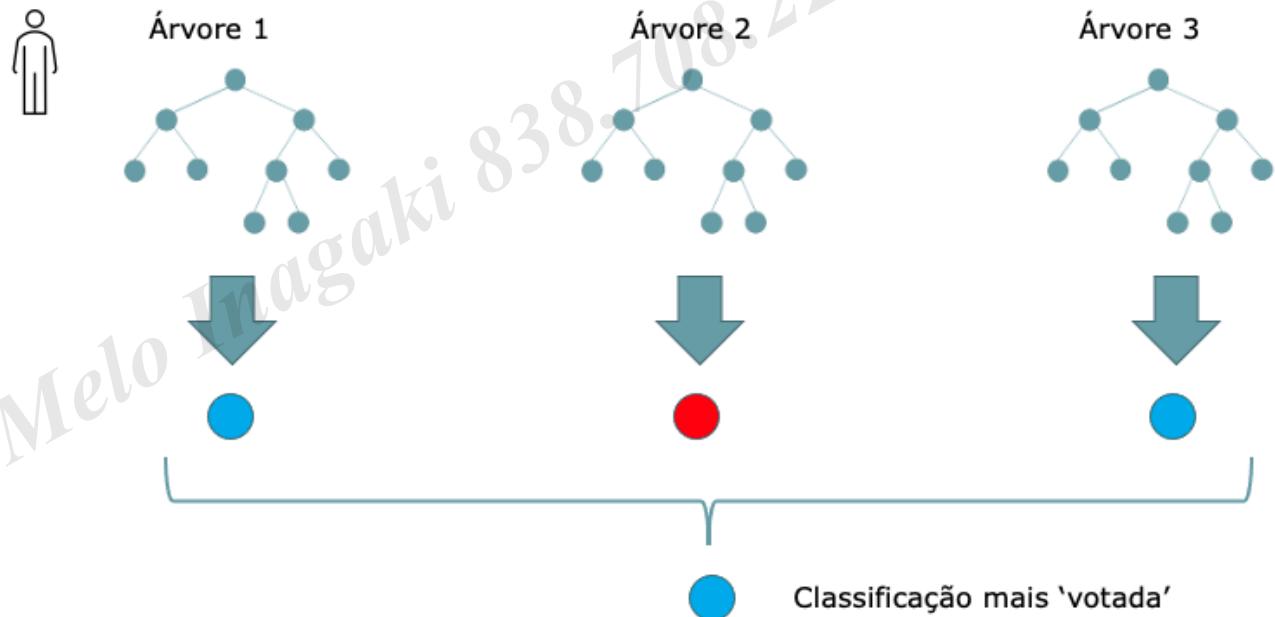
# *Ensemble - aggregation*

Queremos agregar modelos que sejam:

*Úteis*

*Mirem no mesmo alvo*

*Diferentes*



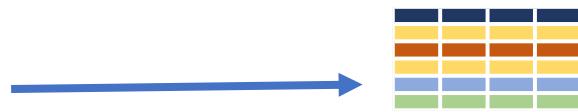
Queremos preditores diferentes, mas que “apontem” para a mesma variável resposta. Uma ideia seria gerar preditores com alguma ‘perturbação’ aleatória.

# *Bootstrapping para avaliar a média*



E se ao invés de alterar o algoritmo, alterarmos a base usando o mesmo algoritmo?

# *Bootstrapping para avaliar a média*



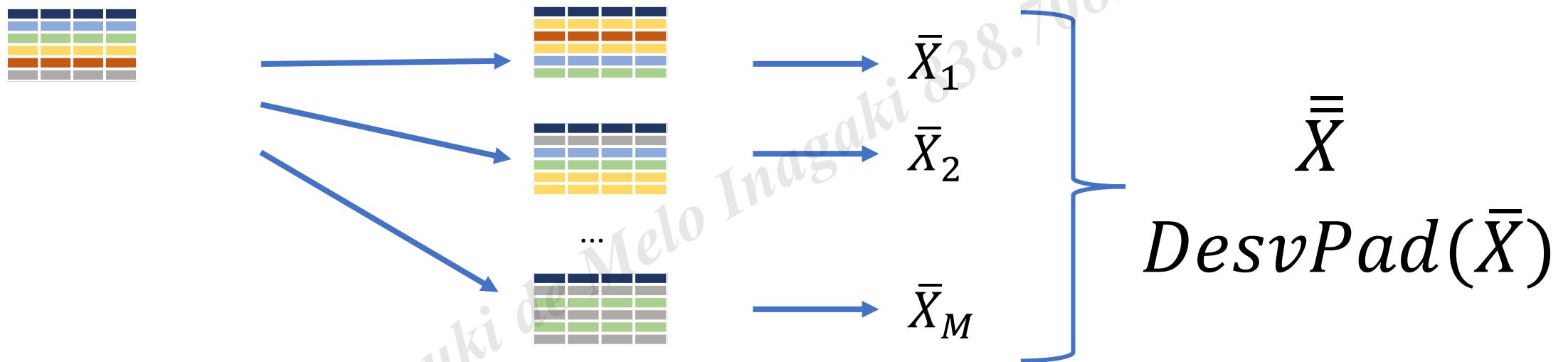
$$\bar{X}_1$$

Temos um conjunto de dados de tamanho N

Queremos estimar o erro padrão de um parâmetro, por exemplo, a média.

- 1) Retirar uma amostra aleatória de tamanho N da base
- 2) Calcular o parâmetro, armazenar a informação

# Bootstrapping para avaliar a média

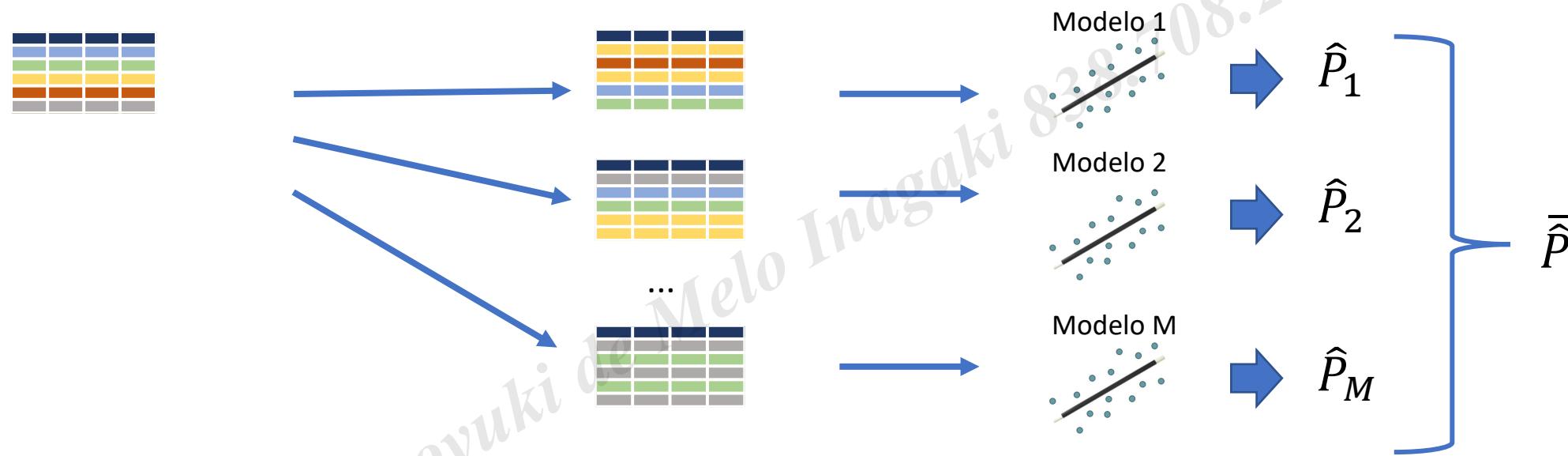


- 3) Repetimos isso  $M$  vezes (digamos...  $M=10.000$  vezes)
- 4) Podemos calcular a média e o erro padrão do estimador



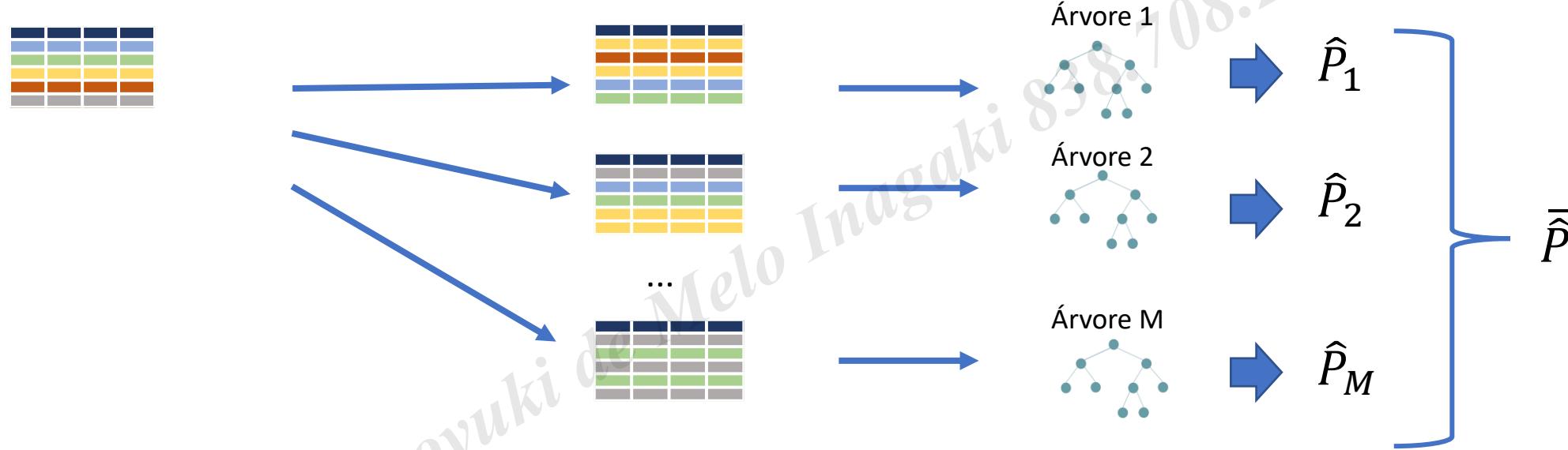
AREM2\_script02\_bootstrapping.R

# Bootstrap – aggregation (bagging)



Bagging é um *agregation* do mesmo *algoritmo* em amostras *bootstrap*

# Bootstrap – aggregation (*bagging*)



O *bagging* com árvores é o famoso *Random Forest*

# RANDOM, FOREST, RANDOM!



Random Forest



OMML2\_script03\_bagging\_v2.R

Joao Hiroyuki de Melo Ingaki 828.108.225-20

# *Bagging e Pasting*

## *Bagging*

1. Retirar uma amostra aleatória **com reposição de tamanho N**
2. Construir o modelo nessa amostra
3. Repetir 1 e 2 M vezes

## *Pasting*

1. Retirar uma amostra aleatória **SEM reposição de tamanho Q<N**
2. Construir o modelo nessa amostra
3. Repetir 1 e 2 M vezes

O bagging mais famoso é *Random Forest*, que é feito com árvores, daí o nome.

# *Características*

## *Bagging*

1. Roda em paralelo
2. Também classifica em paralelo
3. Costuma ter bom desempenho sem grandes ajustes

Se ele fosse um carro, eu diria que é um GMC Hummer H3.

# Perguntas que eu tinha quando aprendi

## *Random Forest*

1. O *default* é fazer 500 árvores?
2. Demora muito para treinar?
3. E para aplicar a regra? Tenho que aplicar tudo isso de regra?  
Demora?
4. O algoritmo guarda tudo isso de árvore?

Se ele fosse um carro, eu diria que é um GMC Hummer H3.

# *Boosting*

Correção sequencial de erros





A variável resposta de uma iteração é o 'erro' da anterior.

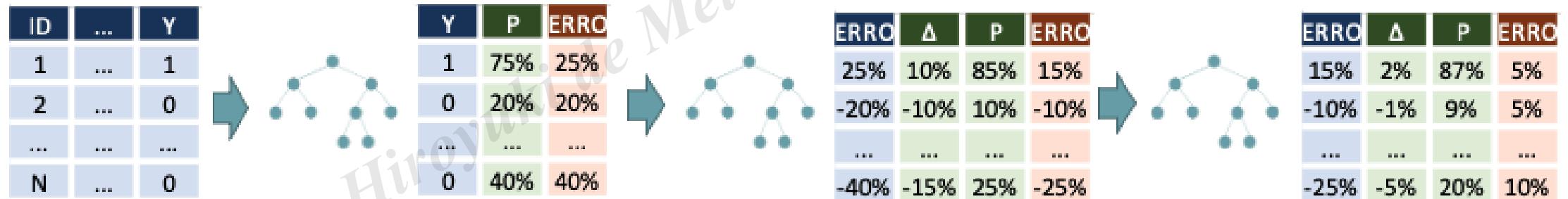
A variável resposta de uma iteração é o 'erro' da anterior.

# Boosting

- Os métodos de *boosting* são modelos sequenciais que tentam melhorar o erro do modelo anterior

# Gradient Boosting

- O *Gradiente Boosting* é uma variação baseada em árvores com alguns hiperparâmetros que controlam o algoritmo





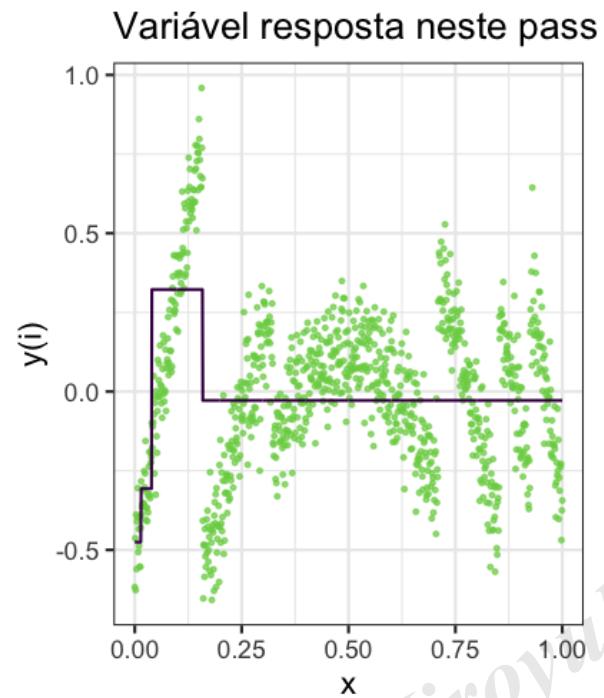
Joao Hiroyuki de Melo Ingaki 828.108.225-20



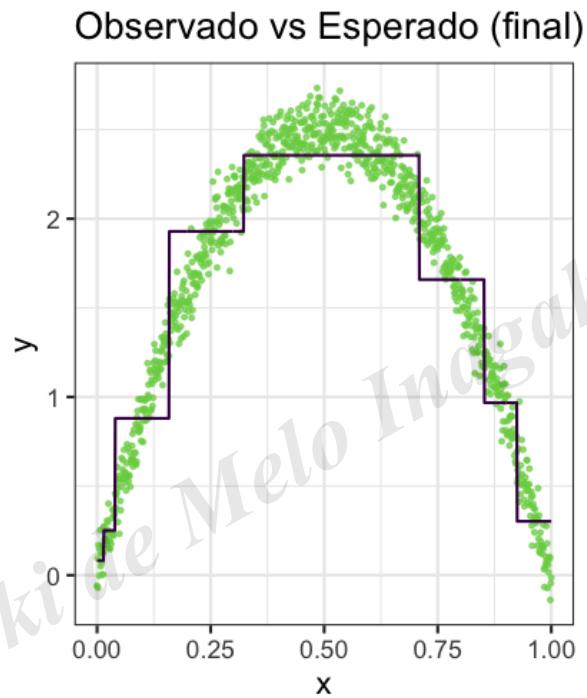
# Learning rate

“Estique a corda demais e  
ela arrebenta, deixe-a  
muito frouxa, e o  
instrumento não toca”

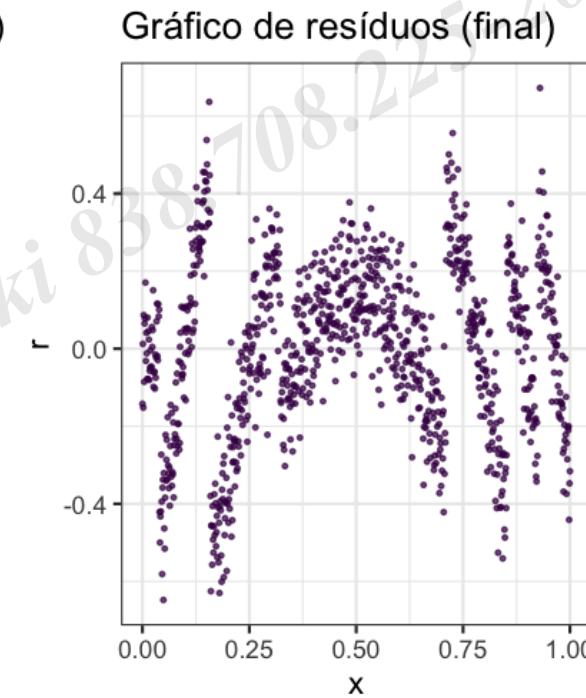
# Learning rate



Dado: — Esperado — Observado



Dado: — Esperado — Observado



Dado: • Resíduo

O Learning Rate diminui o impacto de cada iteração  
costuma demandar mais iterações,  
mas ajuda a alcançar melhores resultados

# XGBoosting

Nome curto para Extreme Gradient Boosting

É uma implementação do Gradient Boosting

Possui interfaces para R e Python

Ficou famosa por ser usada por vencedores de competições

Criado por Tianqi Chen



Joao Hiroyuki de Melo Ingaki 828.108.225-20

# O que fazer com meus novos superpoderes?

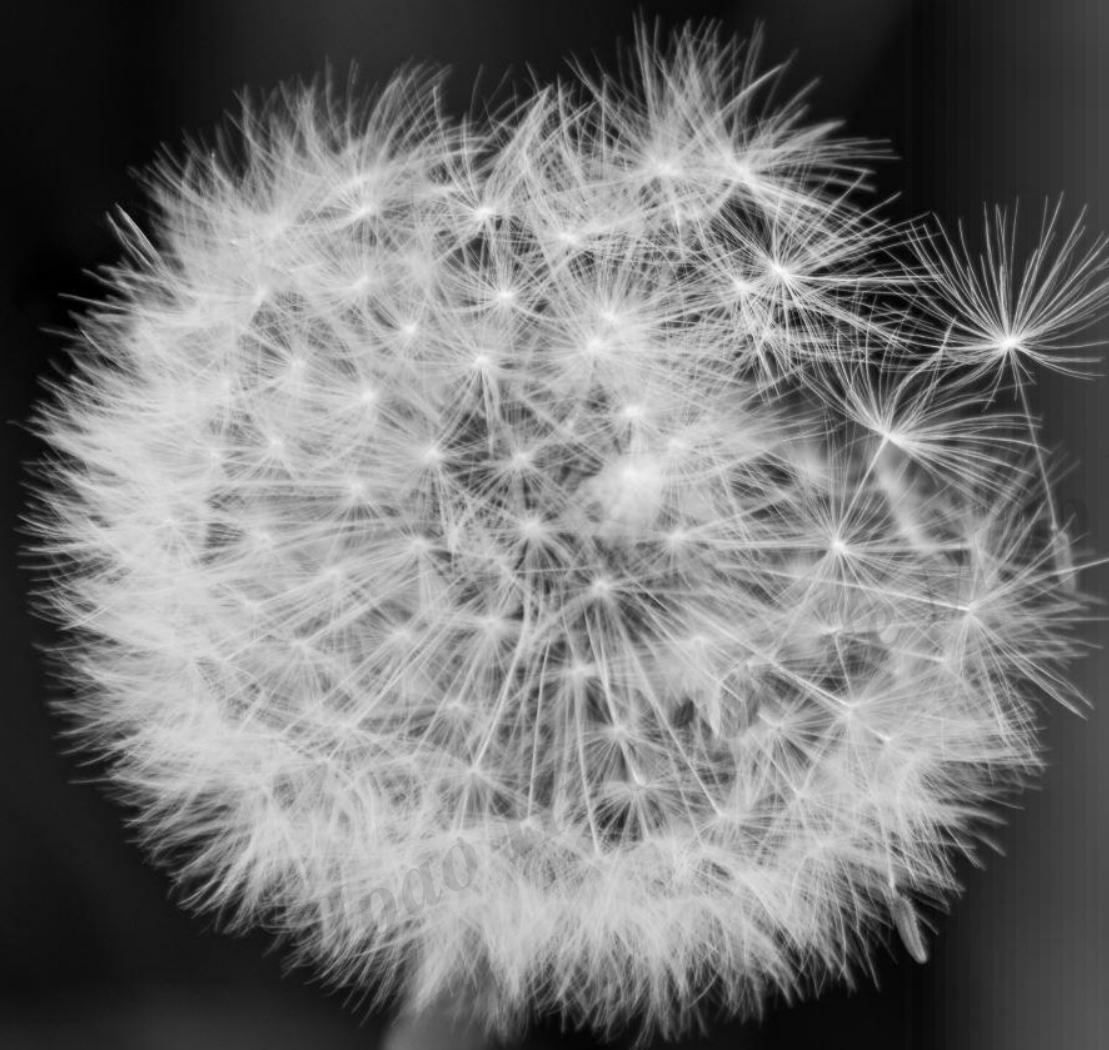
- Sugestões de prática além da aula:
  - Tentar classificar atividade humana por acelerômetro e giroscópio de celular  
<https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones>
  - Identificar doença cardíaca  
<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>



# Conclusões

- Árvores são só o começo
- Há **INFINITAS** formas de combinar modelos, essas são as mais famosas
- Esses modelos são difíceis de se interpretar
- O *cross-validation* ‘entra no lugar’ do *stepwise*
- **PRATIQUE!**





Por hoje é só ;)



[linkedin.com/in/joao-serrajordia](https://linkedin.com/in/joao-serrajordia)

**OBRIGADO!**

[linkedin.com/in/joao-serrajordia/](https://linkedin.com/in/joao-serrajordia/)