MBA USP ESALQ

Outros modelos de Machine Learning I

João F. Serrajordia R. de Mello

Apresentação

João Fernando Serrajordia Rocha de Mello – (Juka)

Trajetória profissional

Modelagem de crédito em grandes bancos
Telecom
Desenvolvimento de modelos / Validação de modelos

Docência em ciência de dados
Consultoria em ciência de dados
Outsourcing executivo

Acadêmico

IME USP

BACHAREL EM ESTATÍSTICA

MESTRE EM ESTATÍSTICA

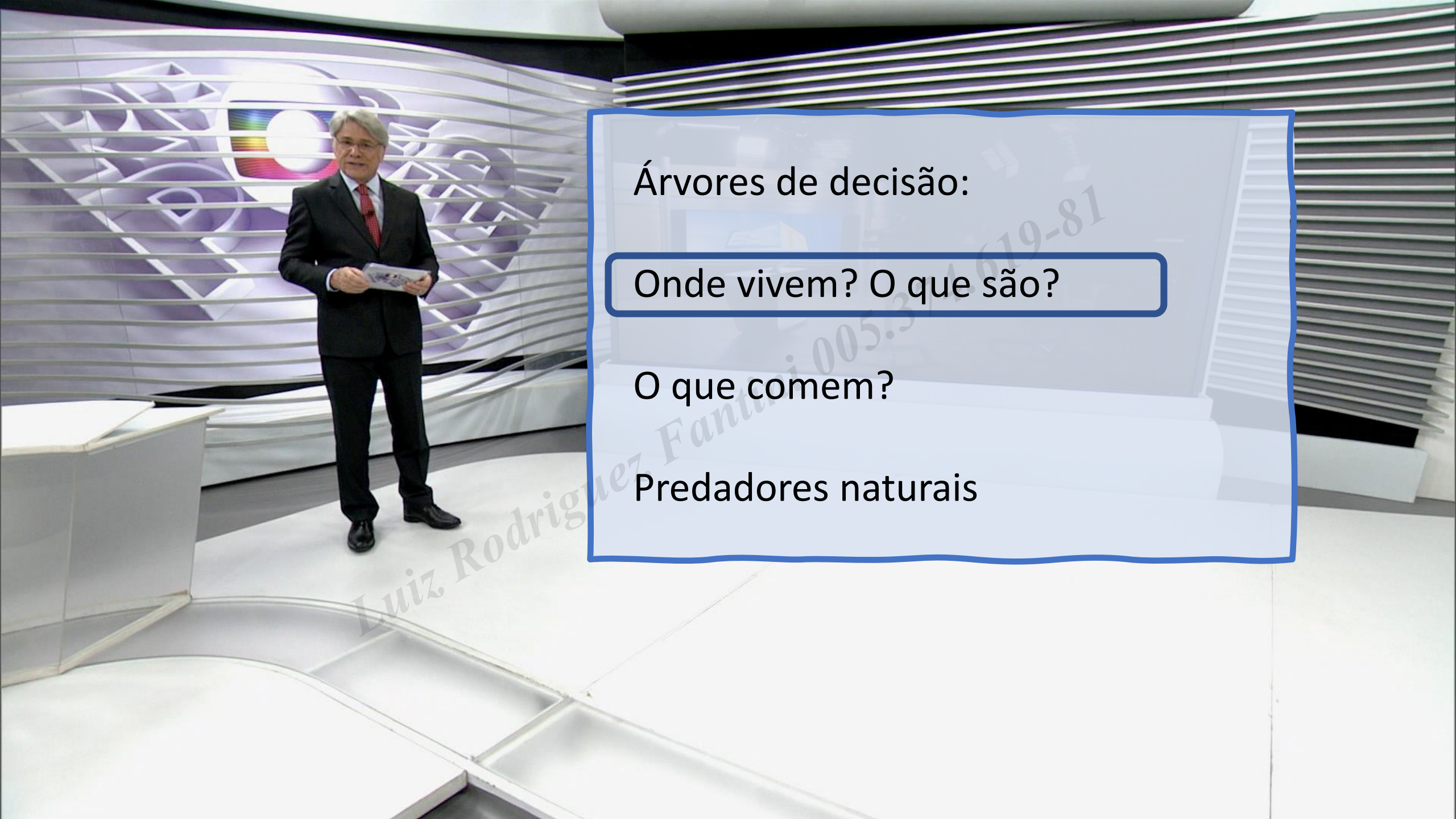
MBA USP ESALQ

Árvores de decisão

Você vai precisar de...

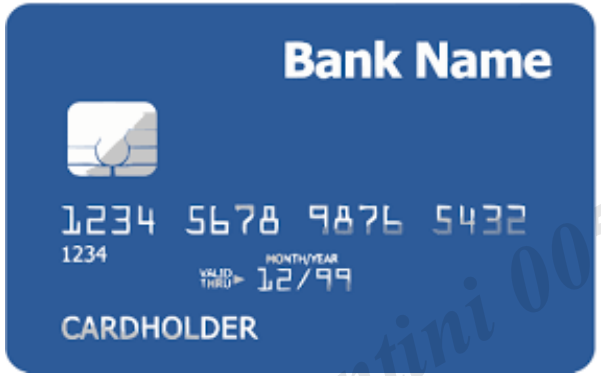
Preparativos

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


Árvores de decisão:
Onde vivem? O que são?
O que comem?
Predadores naturais

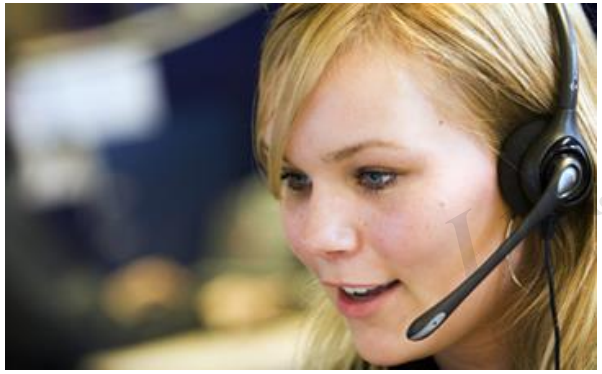
Problemas de preditivos e de classificação

Qual a eficácia de uma vacina?



O cliente vai pagar o empréstimo?

Quanto de petróleo tem no poço?

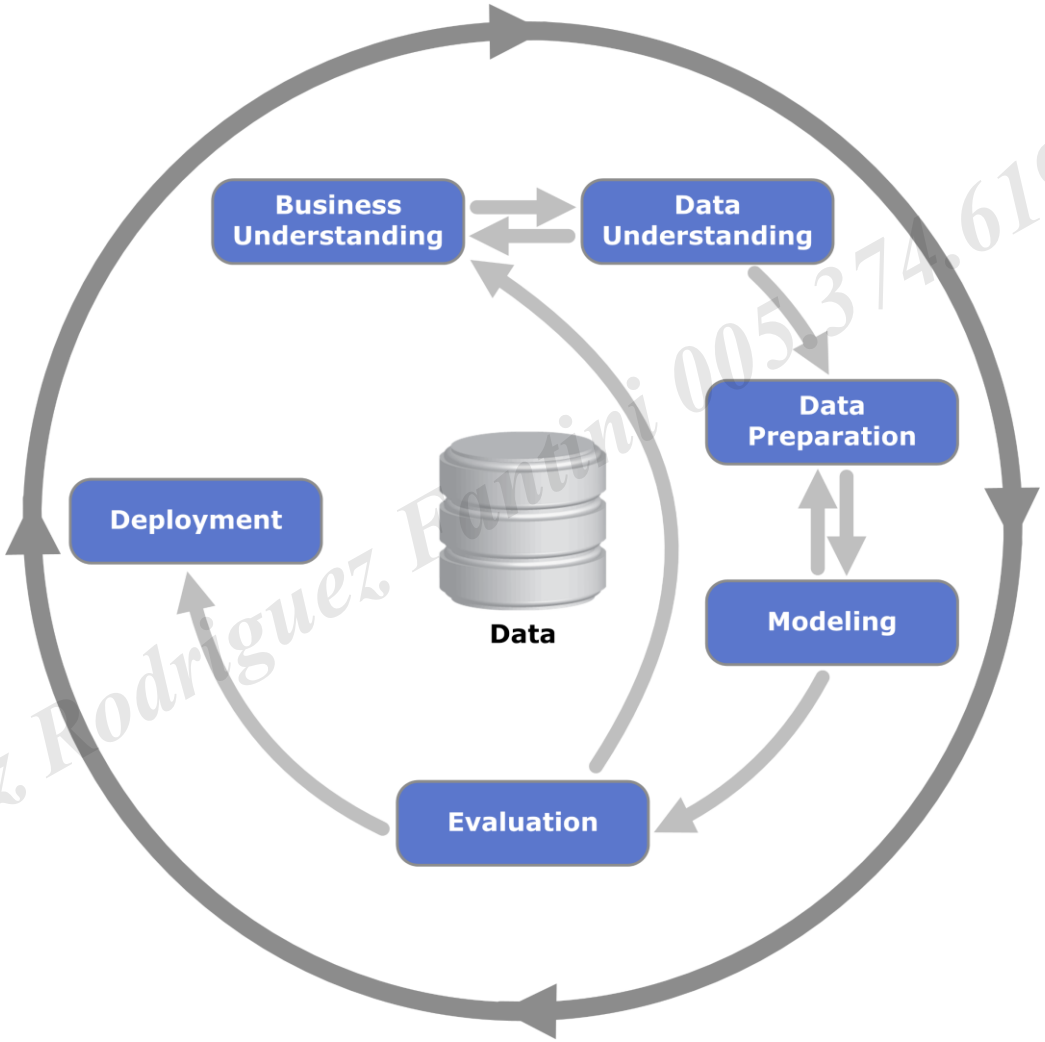


O cliente vai comprar meu produto?

O que a pessoa está fazendo?

Quão ecológico esse veículo é?

CRISP-DM

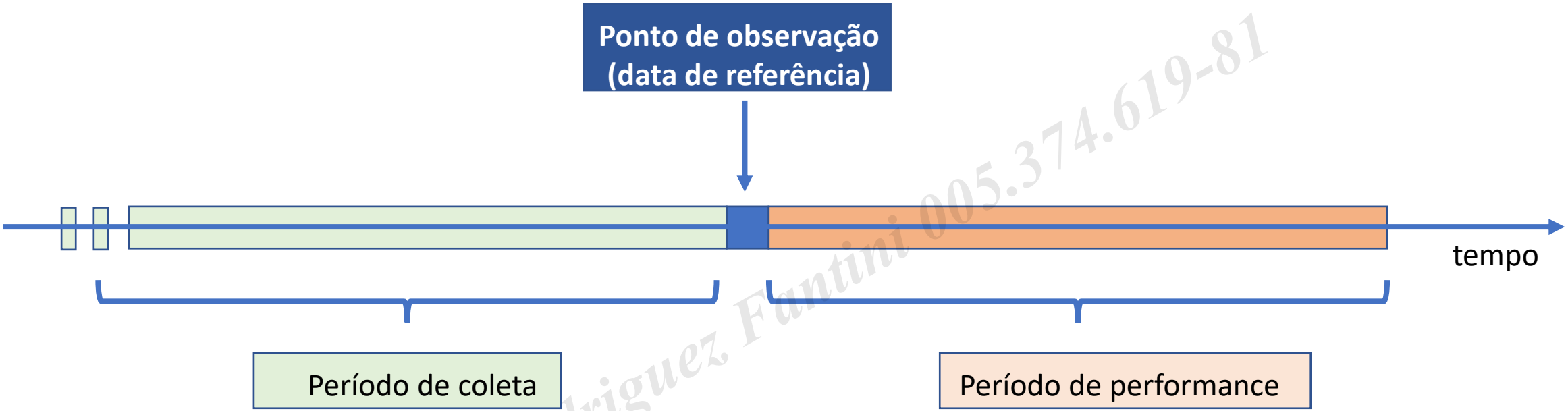


Fonte: https://www.the-modeling-agency.com/crisp-dm.pdf

Modelos preditivos

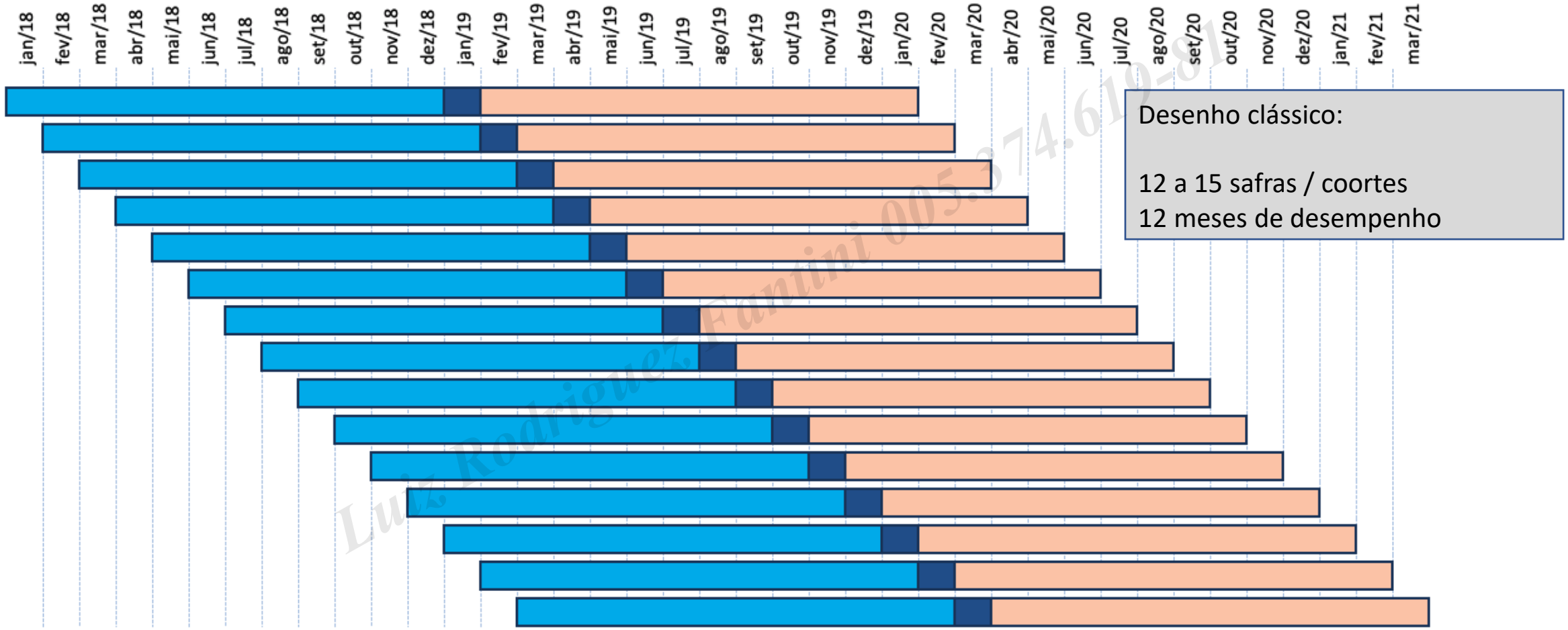
Como é isso?

Desenho de safra (ou coorte)



Exemplo de desenho amostral para modelo preditivo

Desenho do modelo



Desenho clássico:

12 a 15 safras / coortes
12 meses de desempenho

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

Resposta contínua

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

Resposta discreta

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

Estamos aqui!

Classificação dos algoritmos

Métodos Machinelârnicos

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

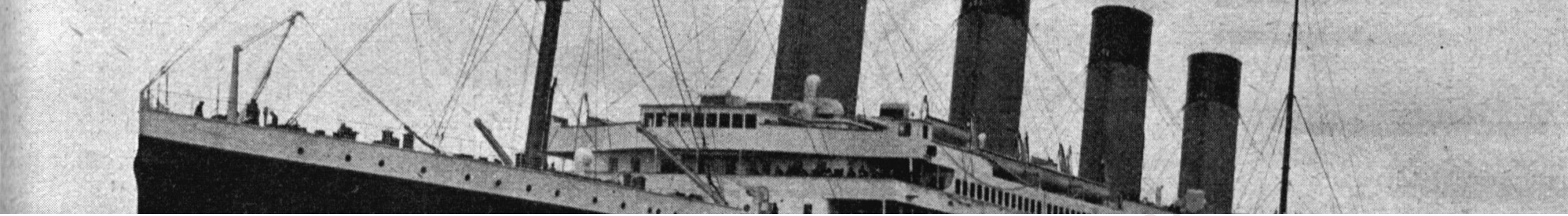
Métodos Machinelârnico-estatísticos

- Regressão
- GLM
- GLMM
- ANOVA

Estamos aqui!

Nosso problema: classificar sobreviventes

Imagem: https://commons.wikimedia.org/wiki/File:Sea_Trials_of_RMS_Titanic,_2nd_of_April_1912.jpg



Reflexões sobre a base de dados

População

- ~ 2.200 pessoas
- ~ 1.300 passageiros
- Mais de 1.500 mortos

Amostra

- 891 pessoas
- 549 não sobreviventes
- 342 sobreviventes

Objetivos do algoritmo

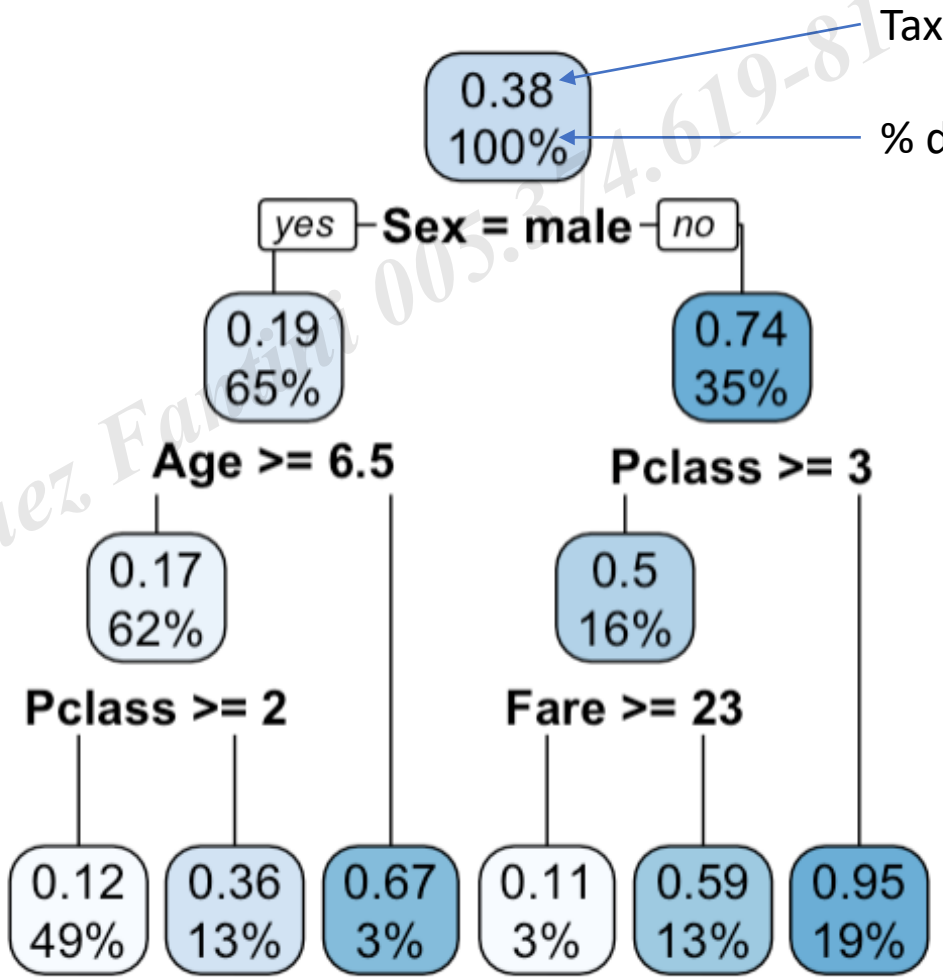
- Classificar da melhor forma possível a variável resposta
  - ... Através de segmentações
  - ... Usando as variáveis explicativas
- Obter insights
  - ... Das relações entre a variável resposta e as explicativas
  - ... Explorar interações

OMML1_script01-Primeiro_contato_com_arvores.R

O que é uma árvore de decisão?

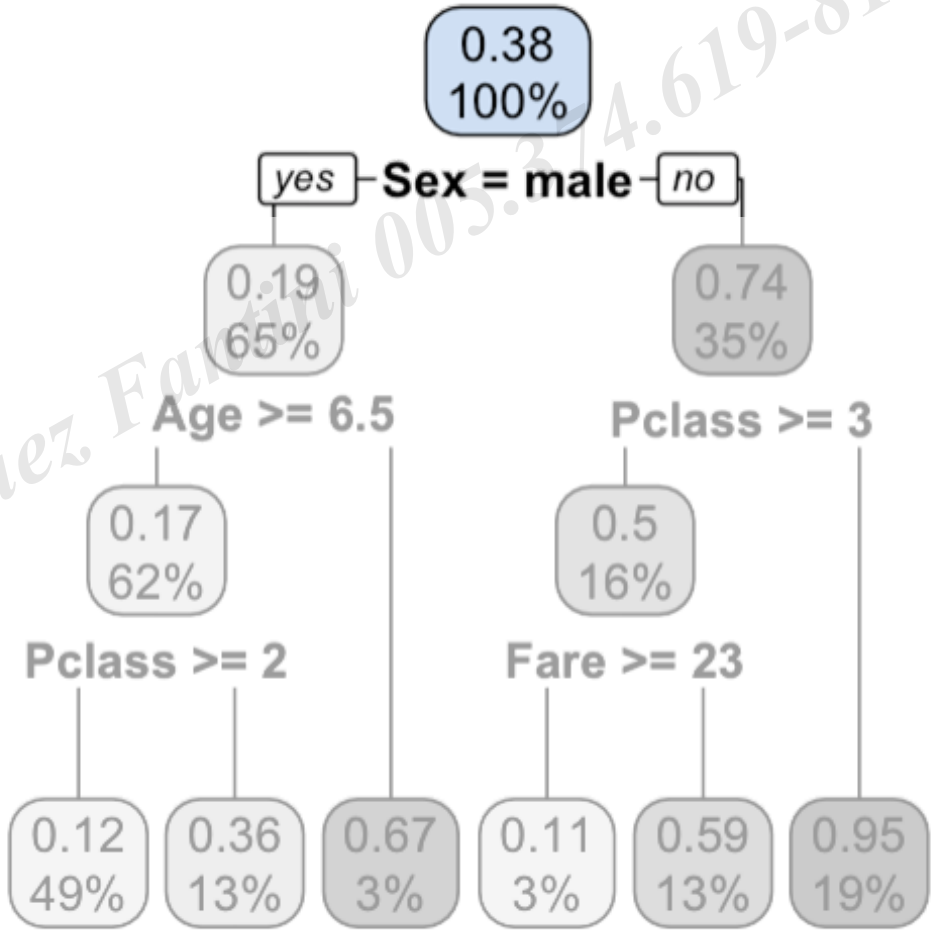
A árvore de decisão é:

Uma sequência de segmentações binárias
Que visa homogeneidade da variável resposta



O que é uma árvore de decisão?

Inicialmente temos 891 passageiros dos quais
342 sobreviveram (38%)
549 não sobreviveram

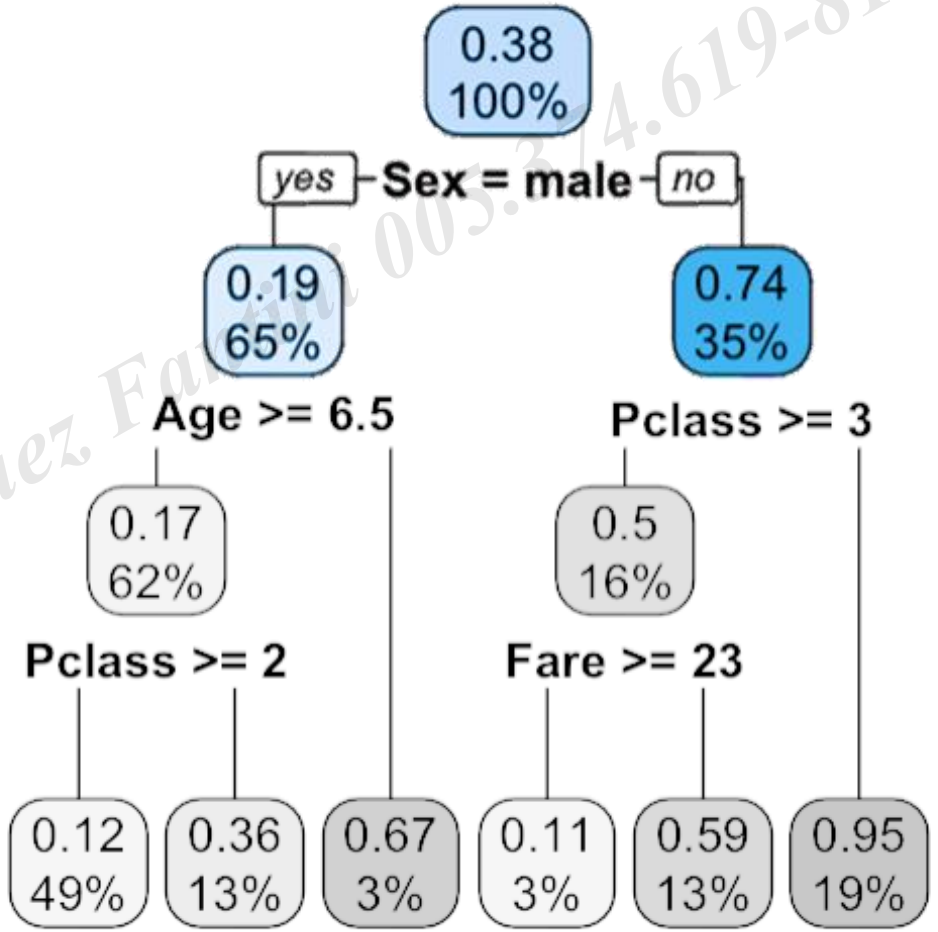


O que é uma árvore de decisão?

Dos 891, podemos segmenta-los em:

577 homens (65%) dos quais
- 109 sobreviveram (19%)
- 468 não sobreviveram

314 mulheres (35%) das quais
- 233 sobreviveram (74%)
- 81 não sobreviveram



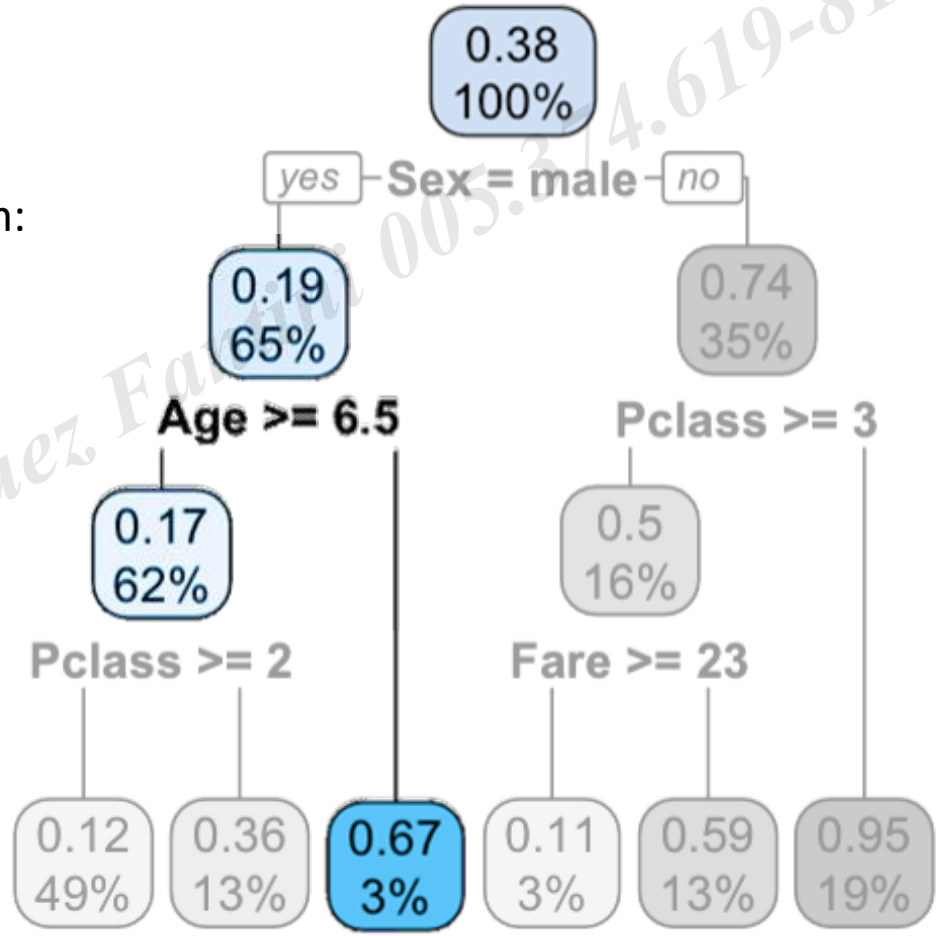
O que é uma árvore de decisão?

Dos 891, podemos segmenta-los em:

577 homens que por sua vez segmentamos em:

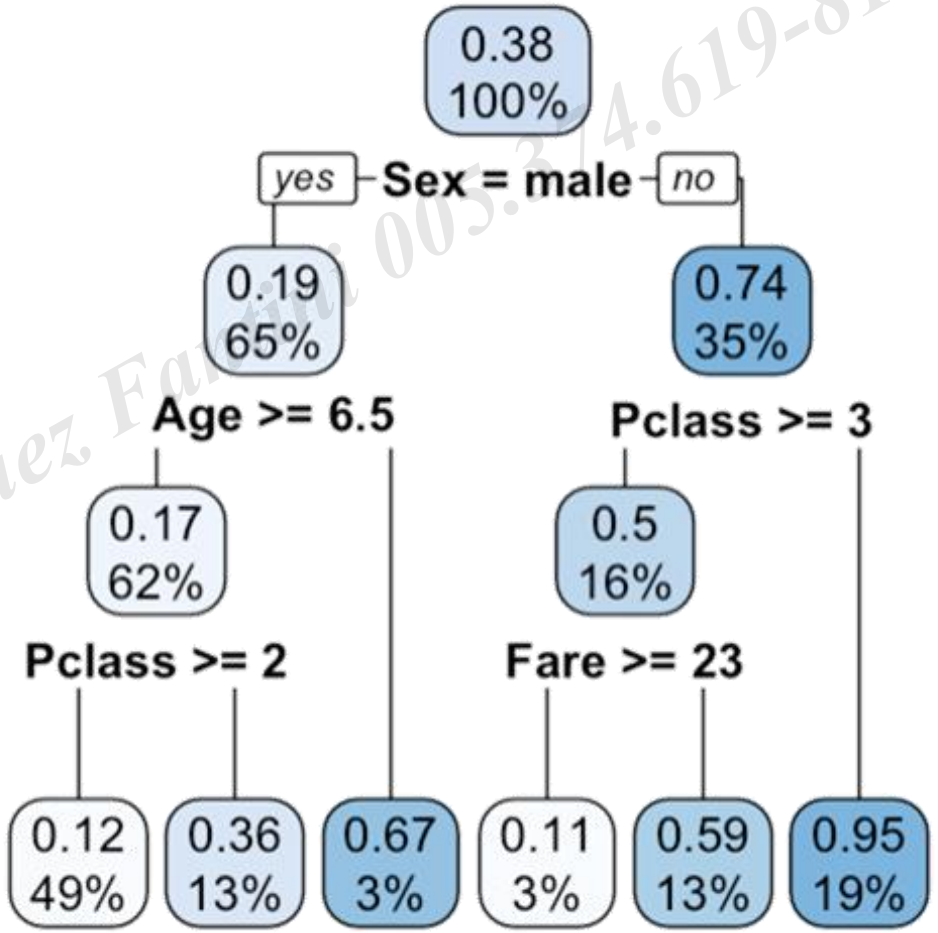
24 crianças (< 6,5 anos) das quais
16 sobreviveram (67%)
8 não sobreviveram

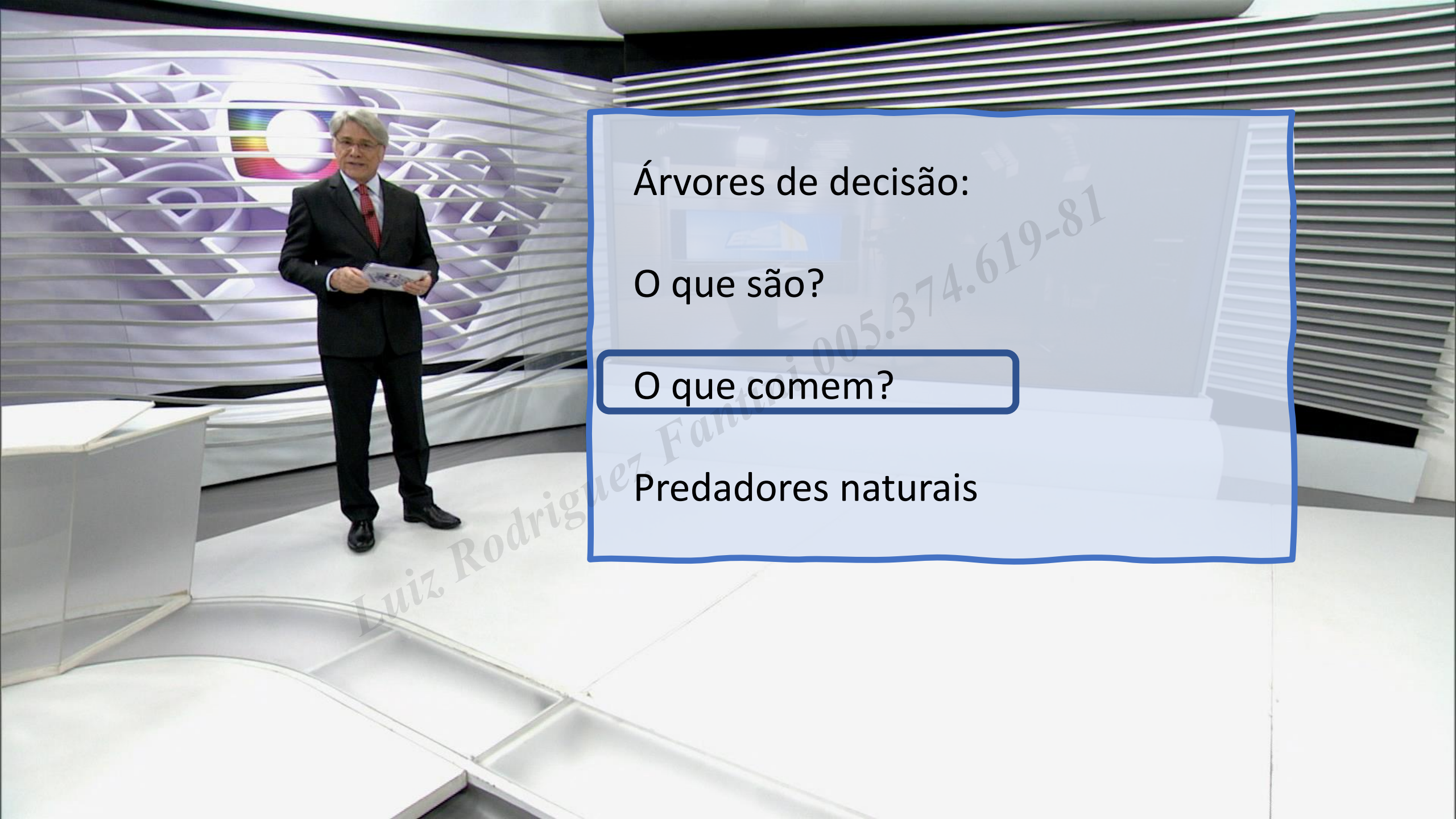
533 adultos (>=6,5 anos) dos quais
93 sobreviveram (17%)
553 não sobreviveram



O que é uma árvore de decisão?

E assim continuamos a “requebrar” a amostra até ”não valer a pena” fazer mais quebras.




Árvores de decisão:
O que são?
O que comem?
Predadores naturais
Luiz Rodriguez Fantini 005.374.619-81

Definições de impureza

- Gini
- Entropia de Shannon

Como a árvore encontra a melhor quebra?
Com uma métrica de ‘impureza'

Índice de Gini

$$I_g(p) = 1 - \sum_{i=1}^{J} p_i^2$$

- Impureza máxima com distribuição uniforme
- Impureza mínima na concentração total

Entropia

$$H = -\sum_{i=1}^{J} p_i \log_2(p_i)$$

Ganho de informação:

$$GI(T, a) = H(T) - H(T|a)$$

- Impureza máxima com distribuição uniforme
- Impureza mínima na concentração total

Algoritmo básico

1. Para cada variável, buscar a melhor regra binária
2. Escolher aplicar melhor segmentação dentre todas as variáveis
3. Recursivamente, para cada folha, repetir os passos 1 e 2 até que uma regra de parada seja atingida

Implementação web interativa:

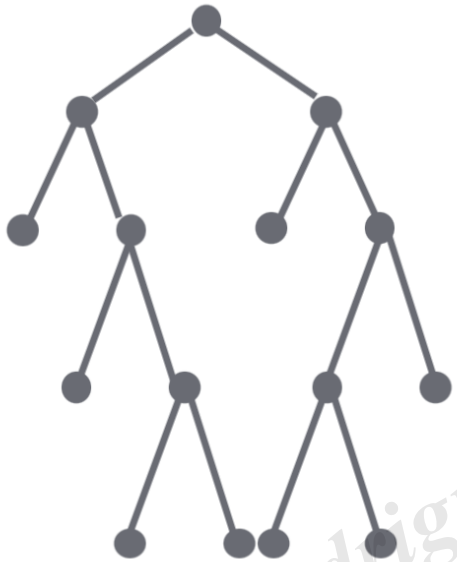
https://rawgit.com/longhowlam/titanicTree/master/tree.html

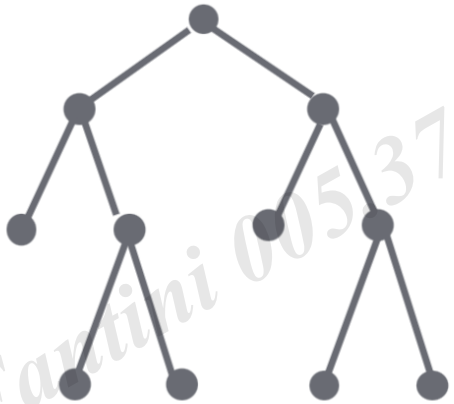
Hiperparâmetros

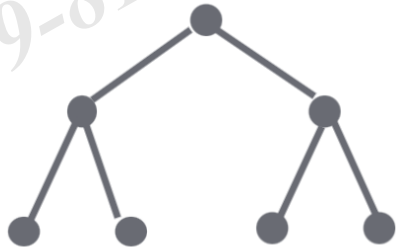
São parâmetros que controlam o algoritmo como:

1. Número mínimo de observações por folha
2. Profundidade máxima
3. CP – Custo de complexidade

Custo de complexidade







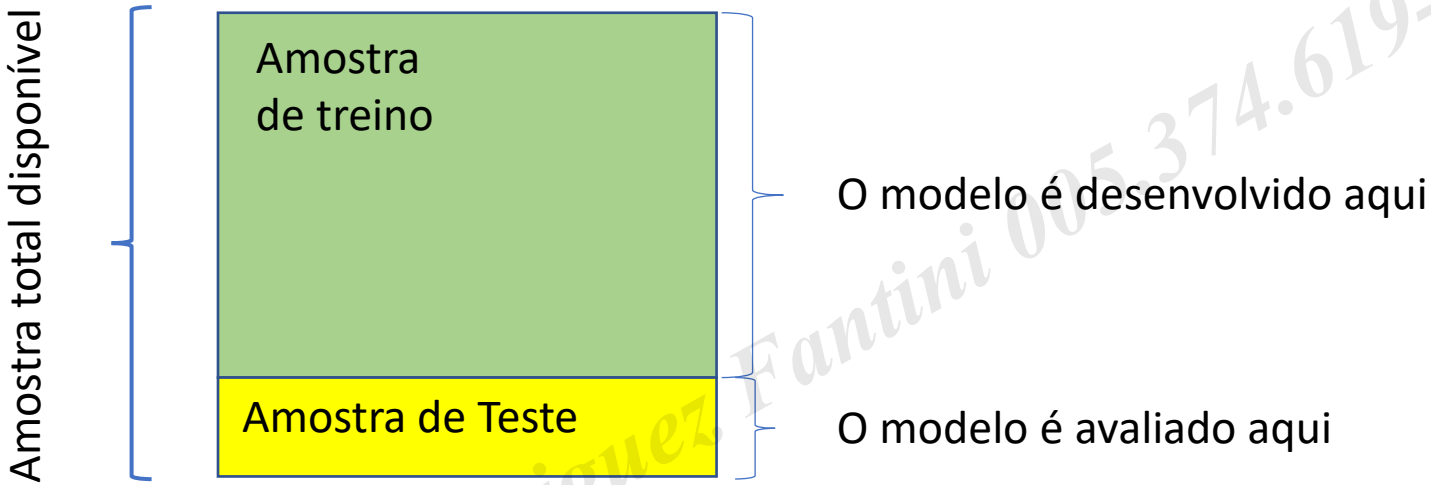
Custo de Complexidade

Baixo

Médio

Alto

Cross validation (validação cruzada)



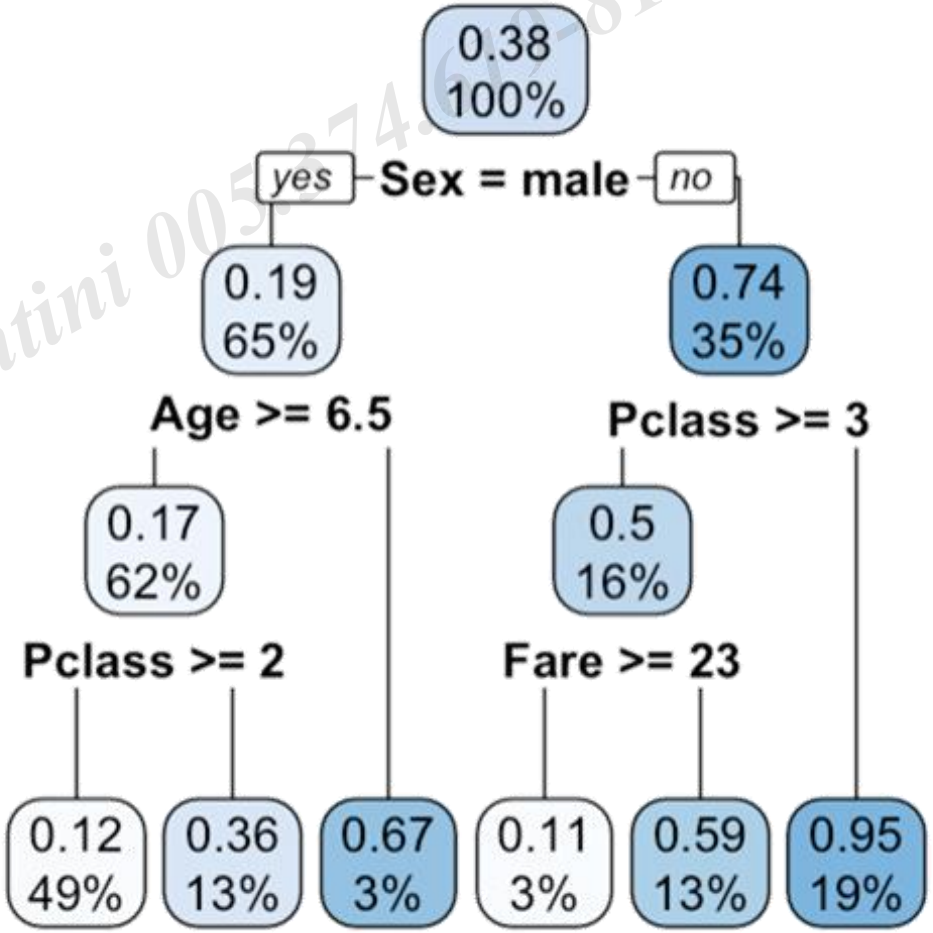
A estratégia mais simples é dividir a base em treino e teste.
Desenvolvemos o modelo na base de treino e avaliamos na base de teste.

OMML1 _script02-Algoritmo_avaliacao_overfitting

A árvore como um classificador

Requisitos:

Ter todas as variáveis.



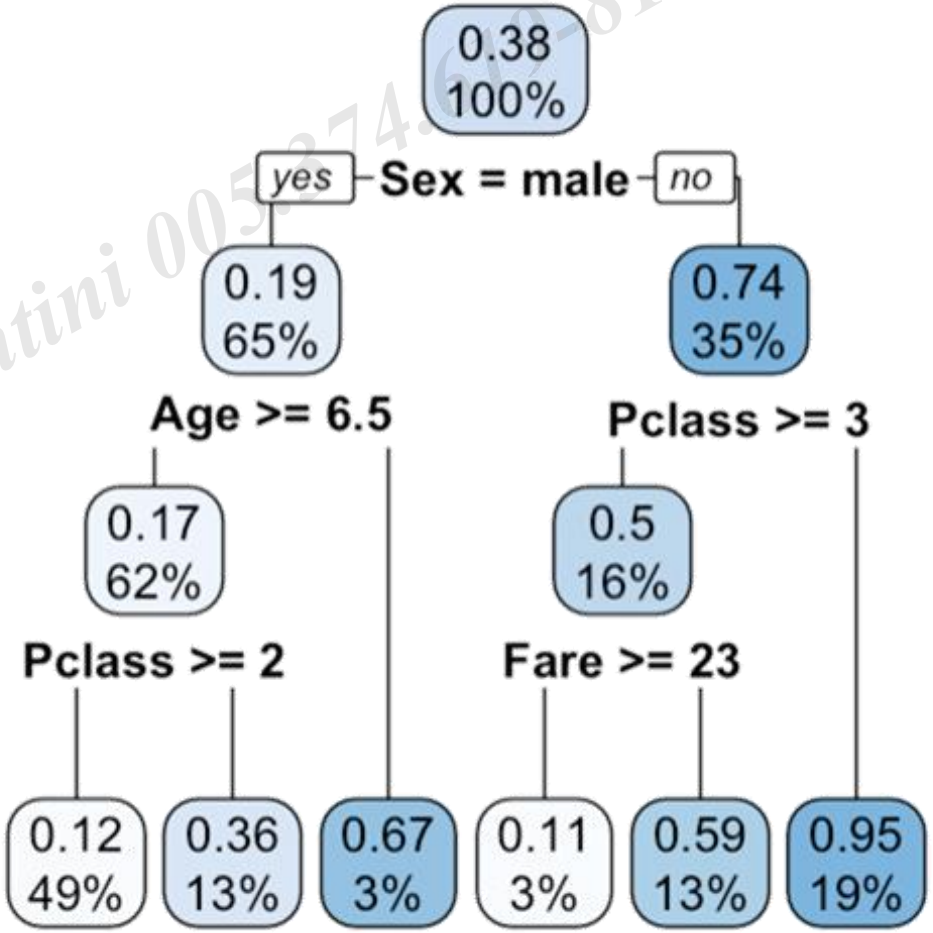
A árvore como um classificador

Probabilidade de evento da folha F:

$$P(S|F) = \frac{N_f^S}{N_f}$$

P(S|F) - probabilidade de sucesso da folha F

N_f - é o número de indivíduos na folha F
N_f^S - é o número de sobreviventes na folha F



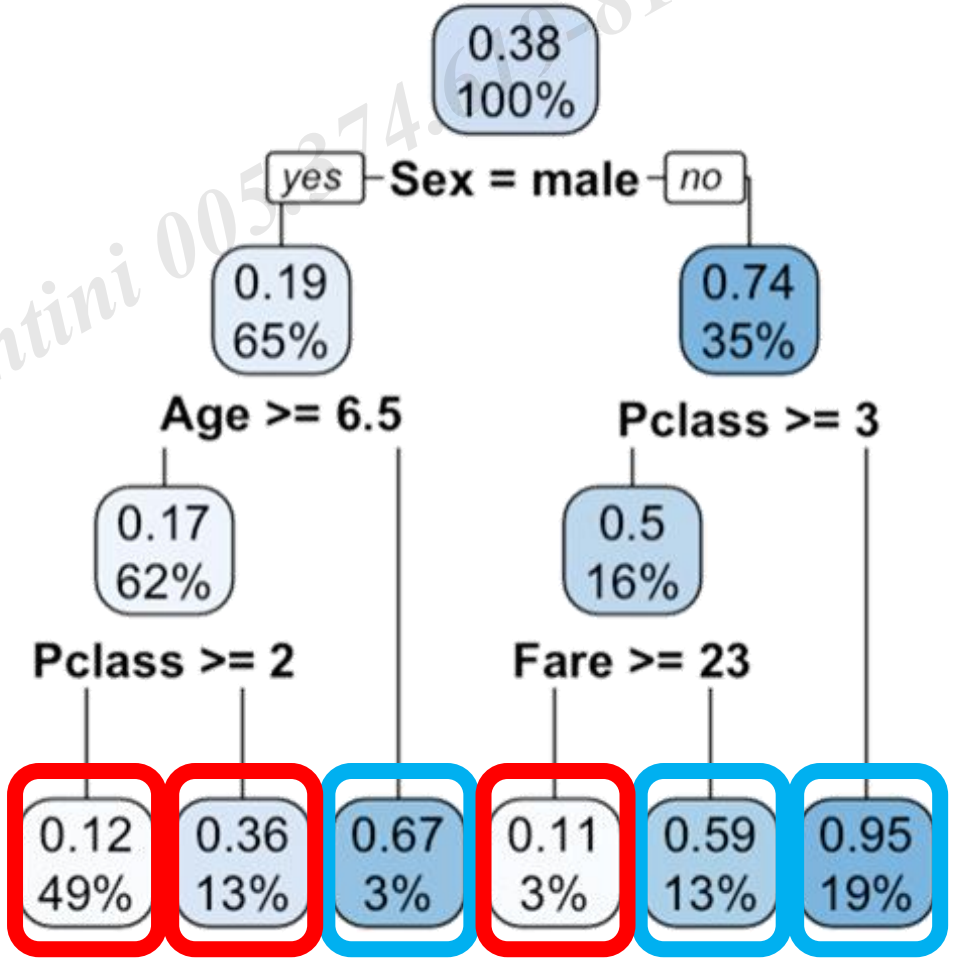
A árvore como um classificador

Classificação:

Classificação padrão:

Sobrevivente: $P(S|F) \geq 50\% \implies C(F) = \text{“}Y\text{”}$
Não sobreviventes: $P(S|F) < 50\% \implies C(F) = \text{“}N\text{”}$

Valor predito	Valor Verdadeiro	
	0	1
0	484	96
1	65	246



Avaliação do modelo

- Acurácia:

Acertos sobre tentativas

Valor predito	Valor Verdadeiro	
	0	1
0	484	96
1	65	246

No exemplo:

$$\frac{484 + 246}{891} = 82\%$$

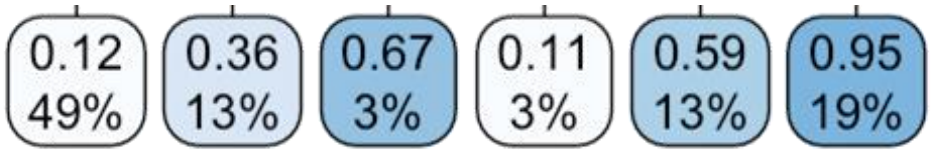
Árvore como diagnóstico

Sensitividade: $\frac{TP}{FN+TP} = \frac{246}{246+96} = 72\%$

Especificidade: $\frac{TN}{TN+FP} = \frac{484}{484+65} = 72\%$

Valor predito	Valor Verdadeiro	
	0	1
0	484	96
1	65	246

Valor predito	Valor Verdadeiro	
	0	1
0	TN	FN
1	FP	TP



Diagnóstico e pontos de corte

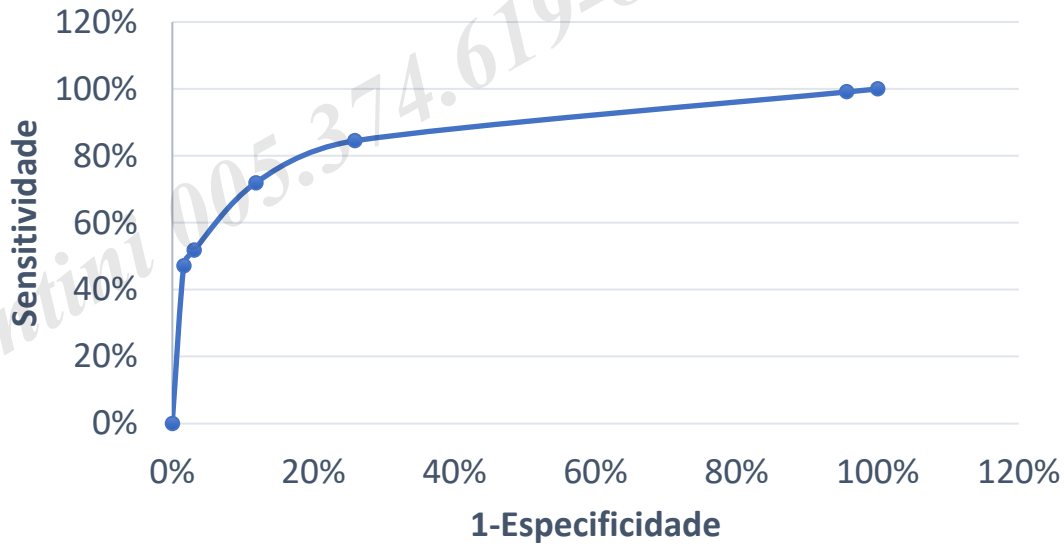
Corte	TP	FP	TN	FN
0% - 11,1%	342	549	0	0
11,1% - 11,5%	339	525	24	3
11,5% - 35,8%	289	142	407	53
35,8% - 58,9%	246	65	484	96
58,9% - 66,7%	177	17	532	165
66,7% - 94,7%	161	9	540	181
94,7% - 100%	0	0	549	342

Acurácia	Especificidade	1-Especificidade	Sensibilidade
38%	0%	100%	100%
41%	4%	96%	99%
78%	74%	26%	85%
82%	88%	12%	72%
80%	97%	3%	52%
79%	98%	2%	47%
62%	100%	0%	0%

Para cada ponto de corte, temos uma matriz de confusão.
No caso, temos 8 possíveis matrizes com a árvore treinada.

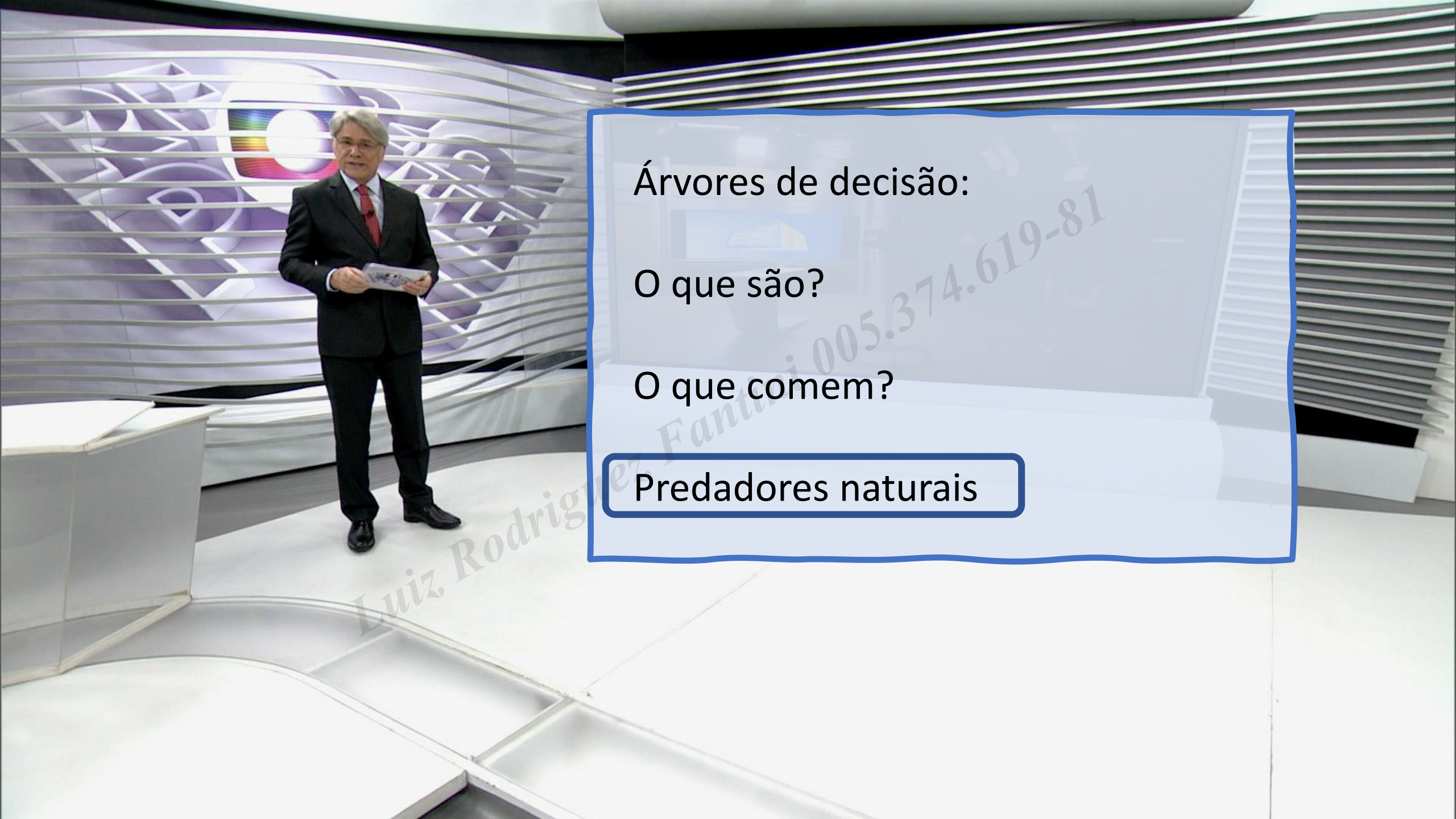
Curva ROC

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.

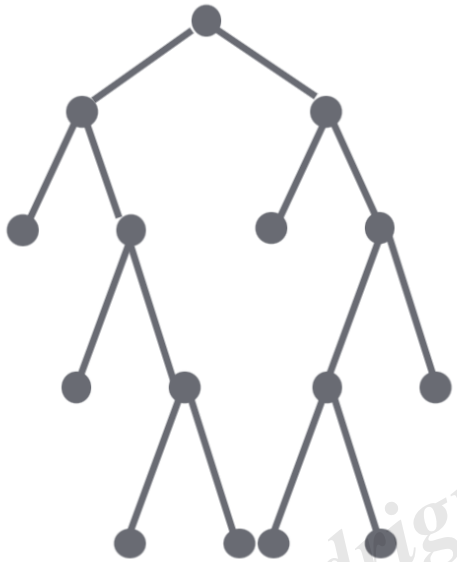
OMML1 _script02-Algoritmo_avaliacao_overfitting

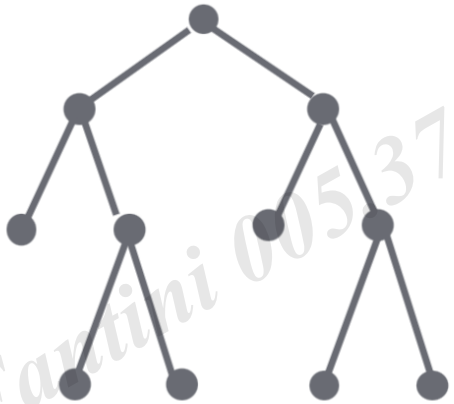

Árvores de decisão:
O que são?
O que comem?
Predadores naturais

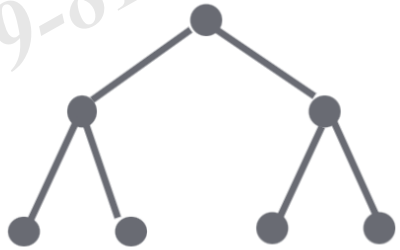
O que é

Como evitar

Poda da árvore *(Prunning)*







Acurácia

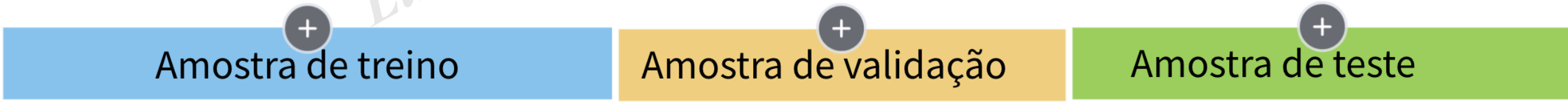
Base de treino: 95%	Base de treino: 70%	Base de treino: 65%
Base de validação: 40%	Base de validação: 60%	Base de validação: 64%

Amostra de treino | Amostra de validação

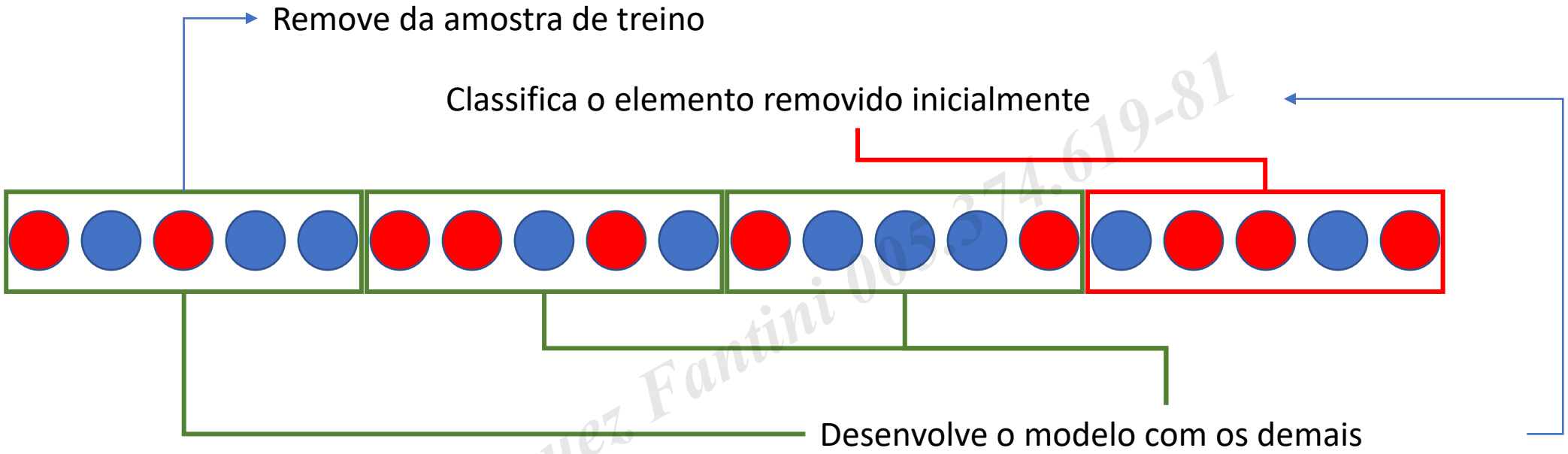
Estratégias de cross validation

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



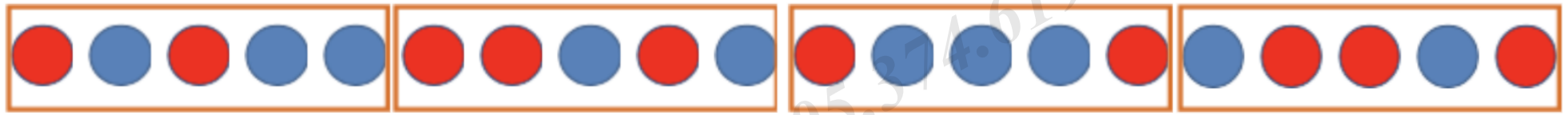
K-fold



- 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

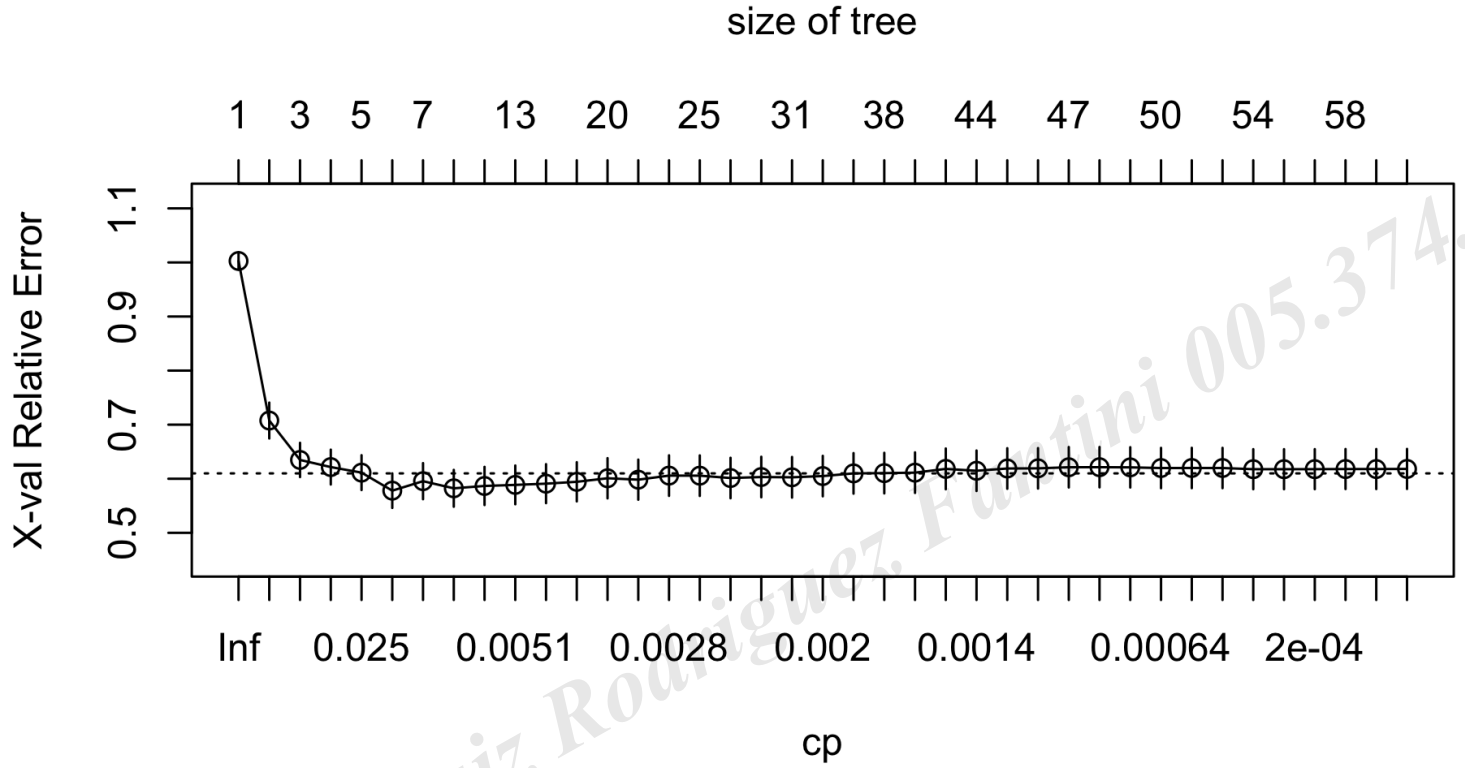
K-fold

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



	Acurácia 1	Acurácia 2	Acurácia 3	Acurácia 4	**Acurácia Média**
Modelo 1	62%	58%	61%	59%	**60%**
Modelo 2	50%	51%	49%	47%	**49%**
Modelo 3	72%	68%	71%	75%	**72%**

Post-prunning com crossvalidation



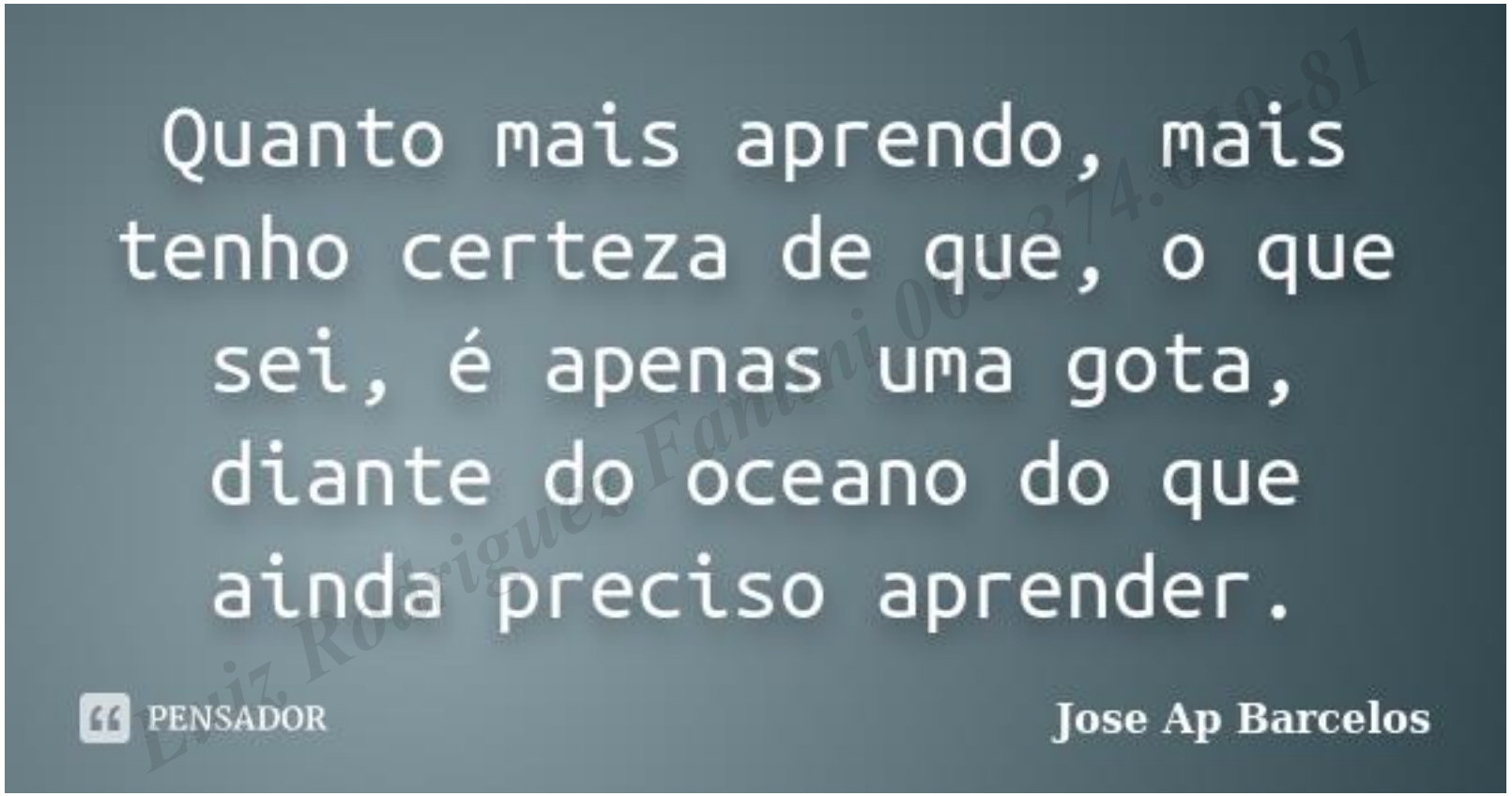
printcp(arvore)

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

OMML1 _script02-Algoritmo_avaliacao_overfitting

Conclusão

- Robustas, interpretáveis, flexíveis
- Sem suposições probabilísticas
- Necessário *cross-validation*



Por hoje é só ;)

linkedin.com/in/joao-serrajordia

Algoritmos famosos

- CART
- CHAID
- ID3
- C4.5
- C5.0

Stack overflow interessante sobre isso:

https://stackoverflow.com/questions/9979461/different-decision-tree-algorithms-with-comparison-of-complexity-or-performance