MBA Business Analytics e Big Data

Análise Preditiva

Prof. Dr. João Rafael Dias

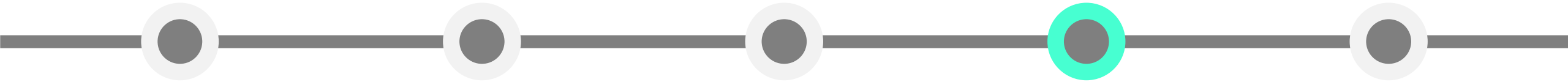
1º semestre - 2020

Agenda

Na aula de hoje...

Aprendizagem supervisionada
Regressão e classificação
Formas de treino e validação
Bias-variance trade-off
Avaliação e comparação de modelos
Prática no RStudio

Estrutura de uma árvore de decisão
Intuição
Particionamento dos nós na regressão e classificação
Poda da árvores vs *overfitting*



Introdução e motivações
Feature engineering
Tratamento de variáveis
Transformação de variáveis
Arquivos de trabalho
Prática no RStudio

Regressão linear múltipla
Coeficiente de determinação
Regressão logística
Odds e log odds
Comparação entre as regressões
Multicolinearidade
Seleção de variáveis *step-wise*
Prática no RStudio

Modelos de *ensemble*
Bootstrap
Random forest
Adaptive boosting
Prática no RStudio

Árvores de decisão

Prof. Dr. João Rafael Dias Pinto

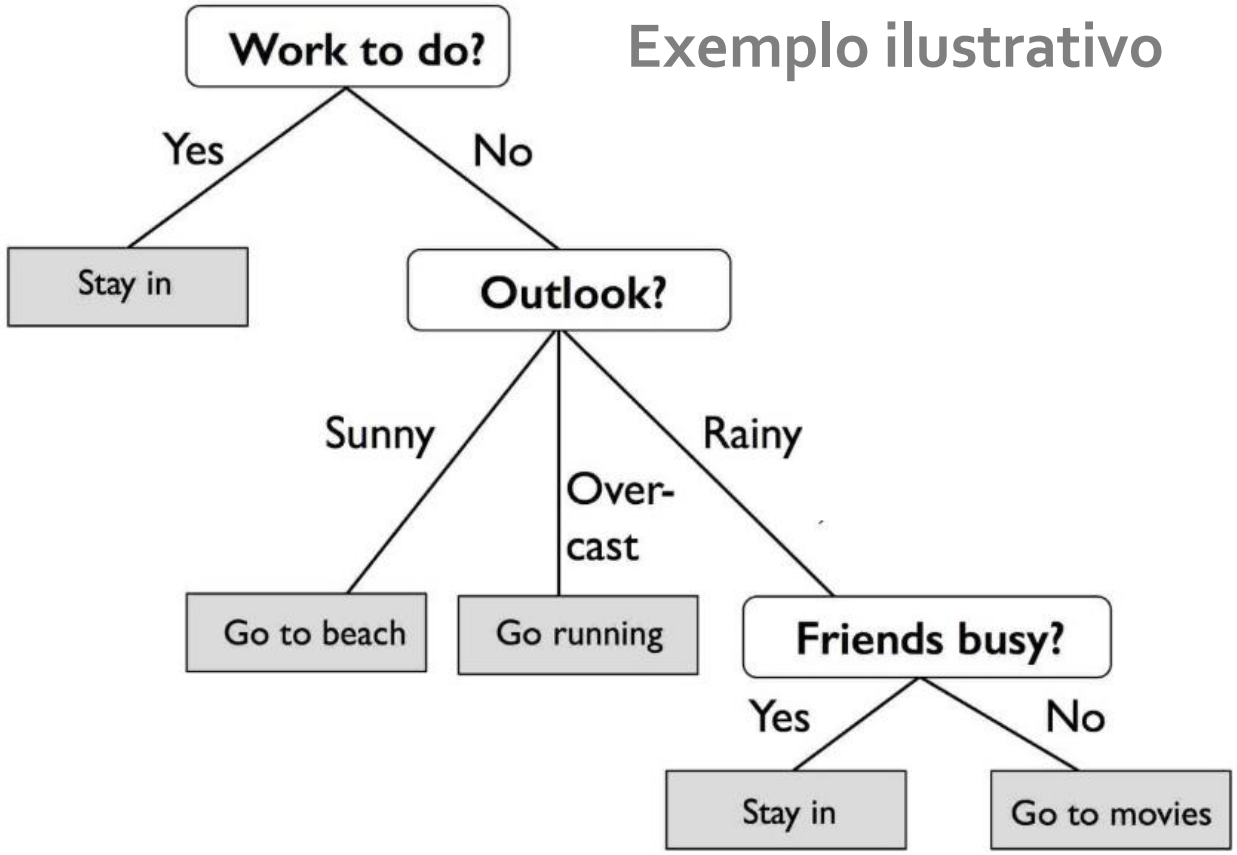
Árvores de decisão

Overview

- Trata-se de uma abordagem completamente diferente da regressão logística
- Comumente denominada de algoritmo de particionamento recursivo
- A ideia por trás é o de quebrar o conjunto de dados em diferentes subconjuntos mais homogêneos em termos da variável alvo. Cada subconjunto por sua vez é particionado sucessivamente até que algum critério pré-estabelecido seja atingido

O algoritmo prevê um *target* y aprendendo regras de decisão através das características **x** (variáveis indepententes)

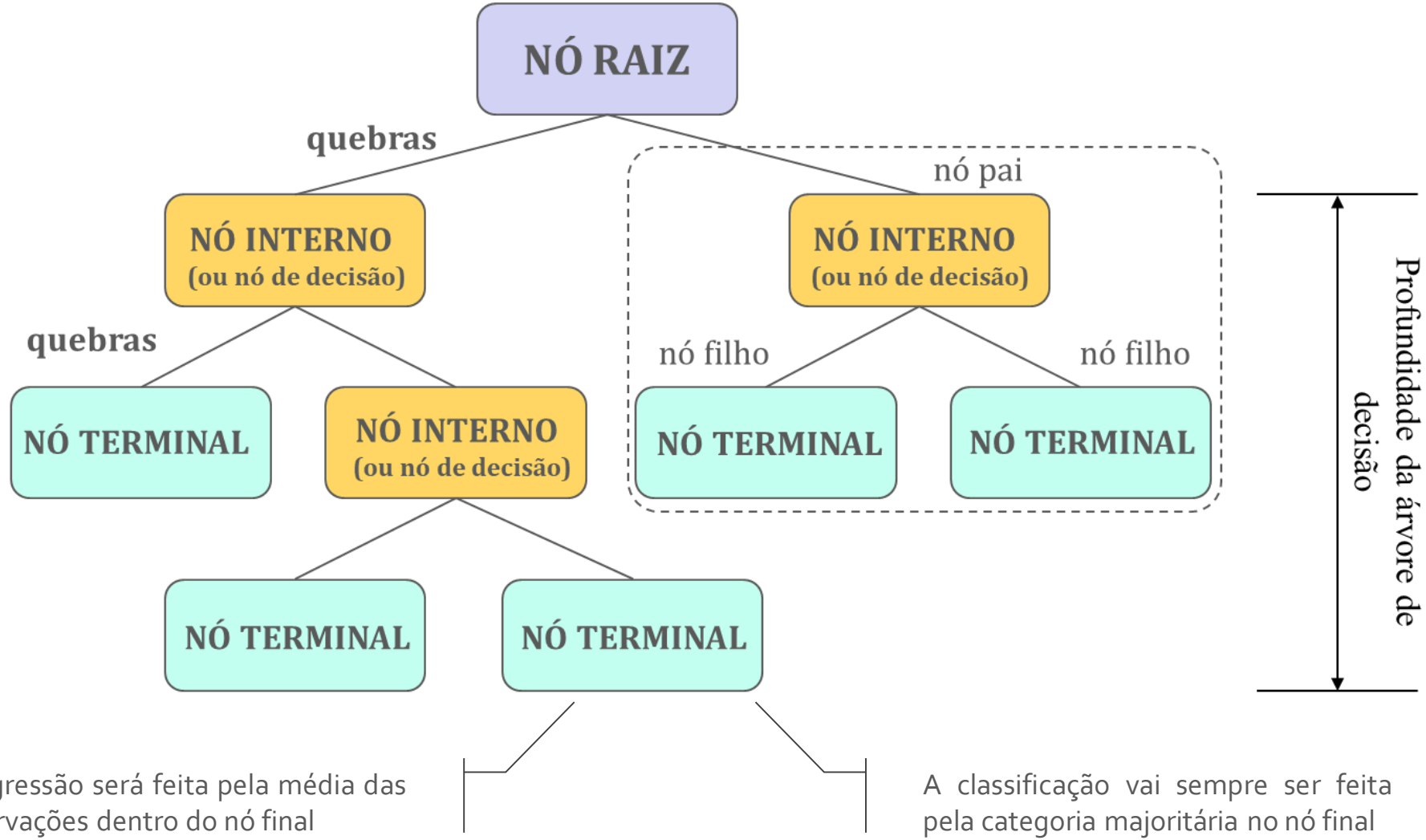
Exemplo ilustrativo



Árvores de decisão

Estrutura geral

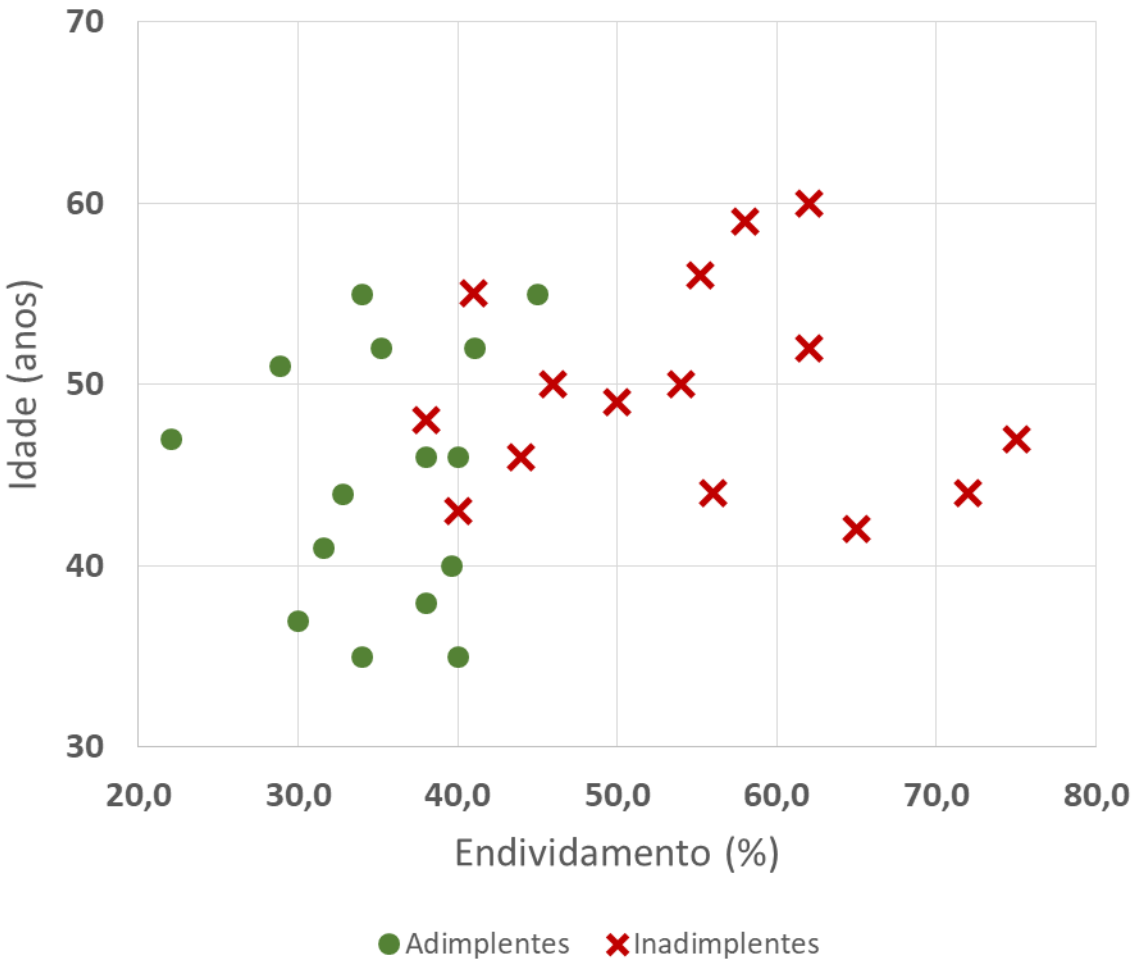
- A figura abaixo representa uma árvore de decisão em termos de nomenclatura



Árvores de decisão

Intuição

- Aqui vamos simular uma distribuição hipotética



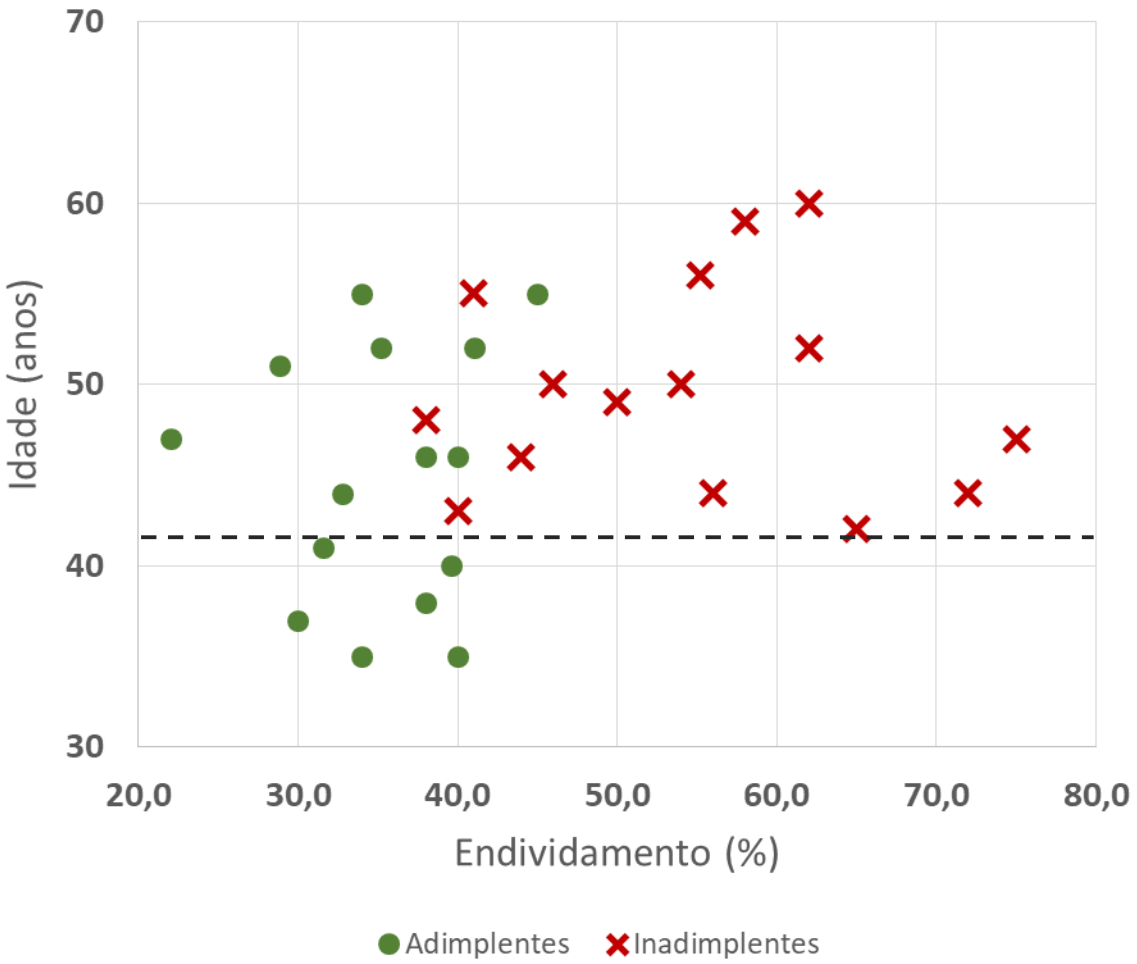
Nó pai

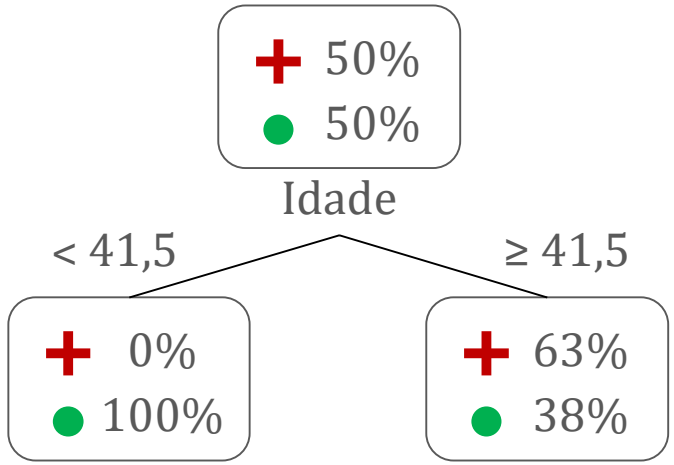
+ 50%
● 50%

Árvores de decisão

Intuição

- Aqui vamos simular uma distribuição hipotética



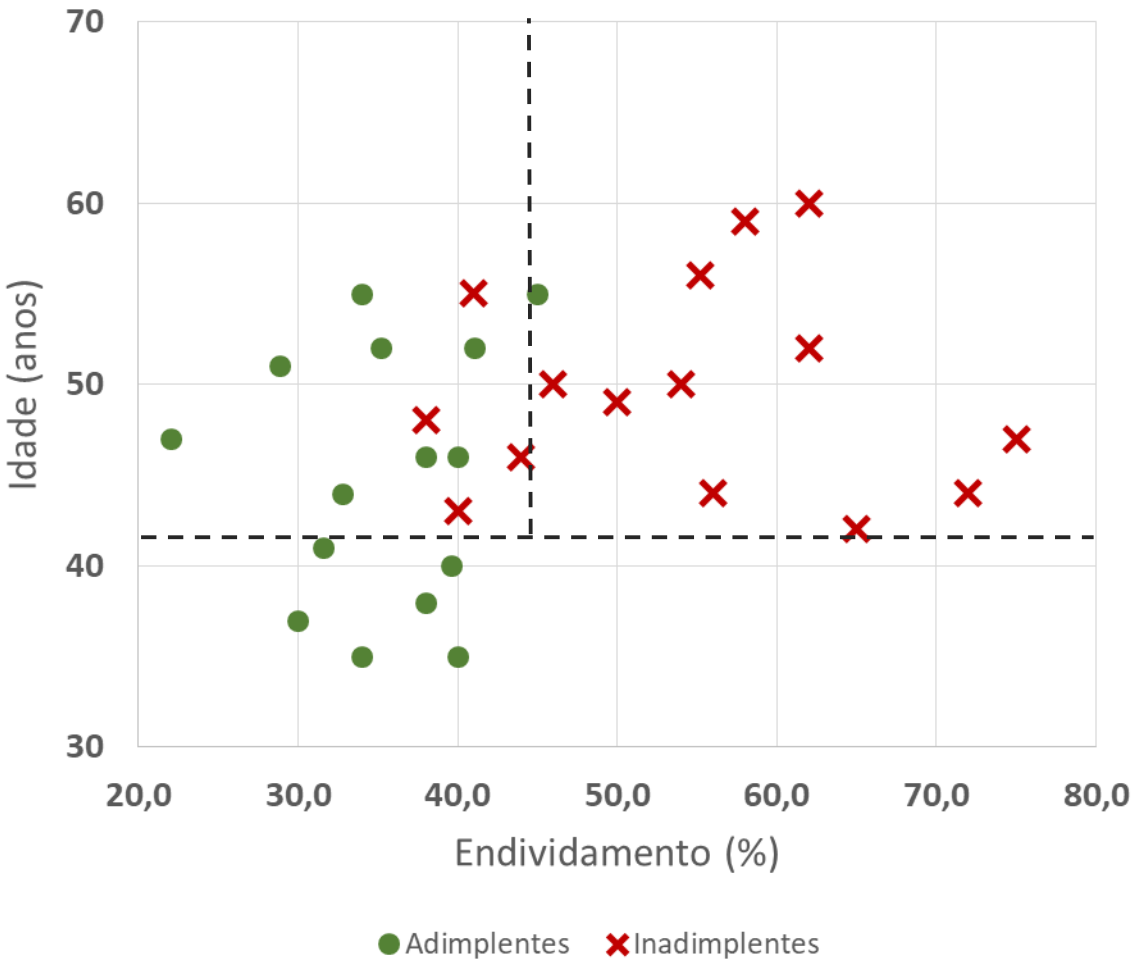


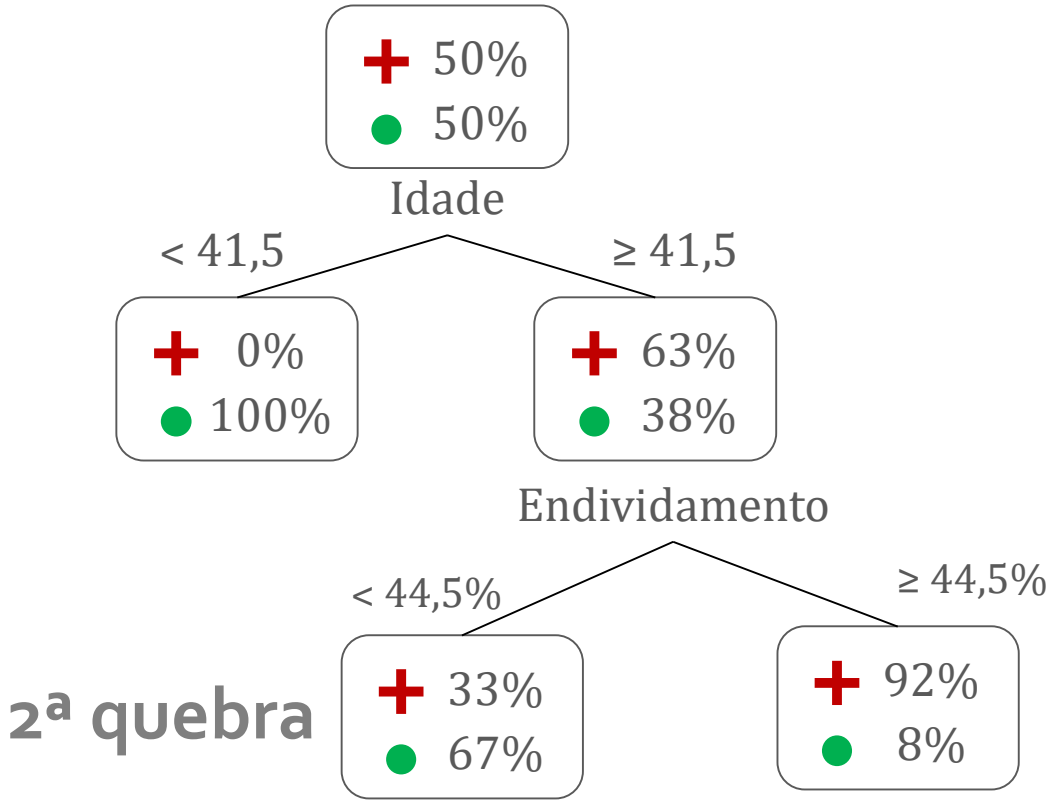
1ª quebra

Árvores de decisão

Intuição

- Aqui vamos simular uma distribuição hipotética



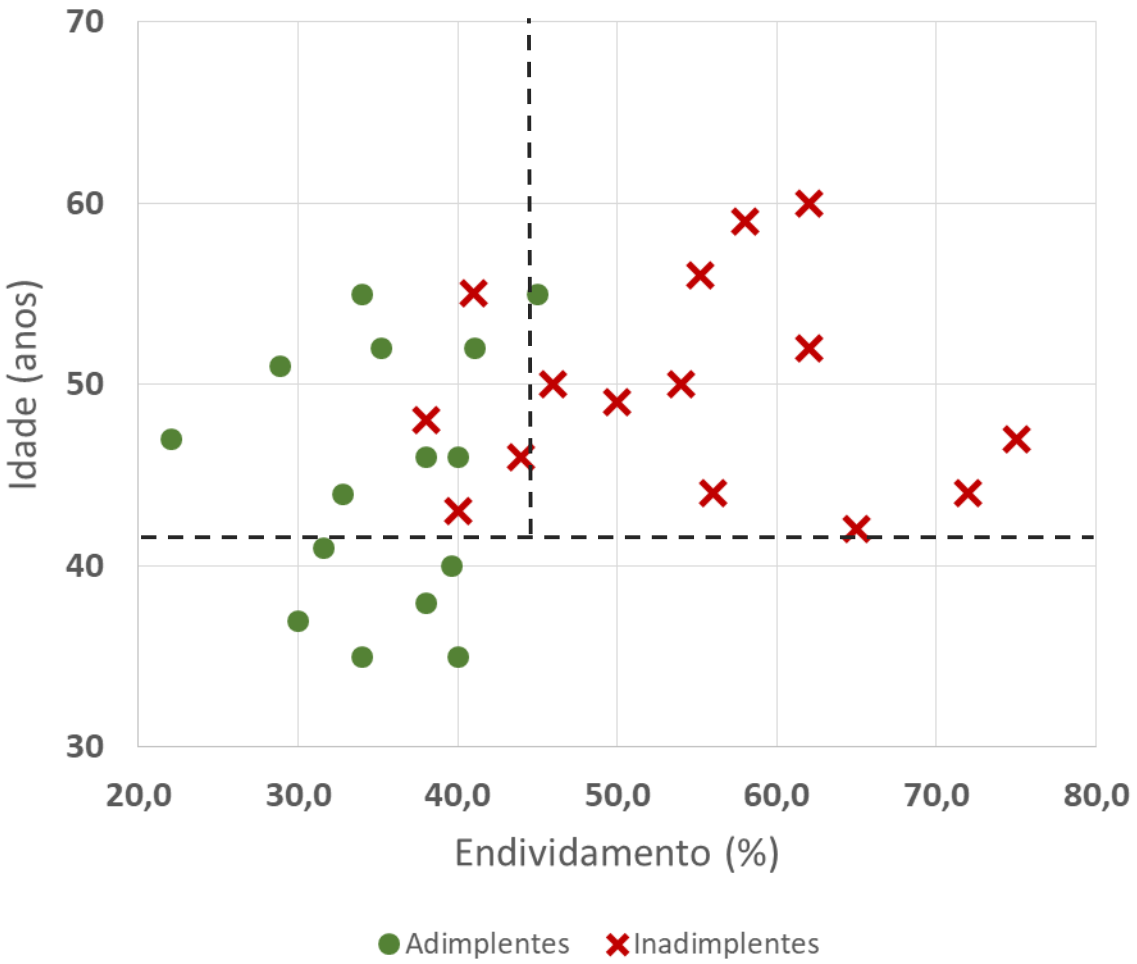


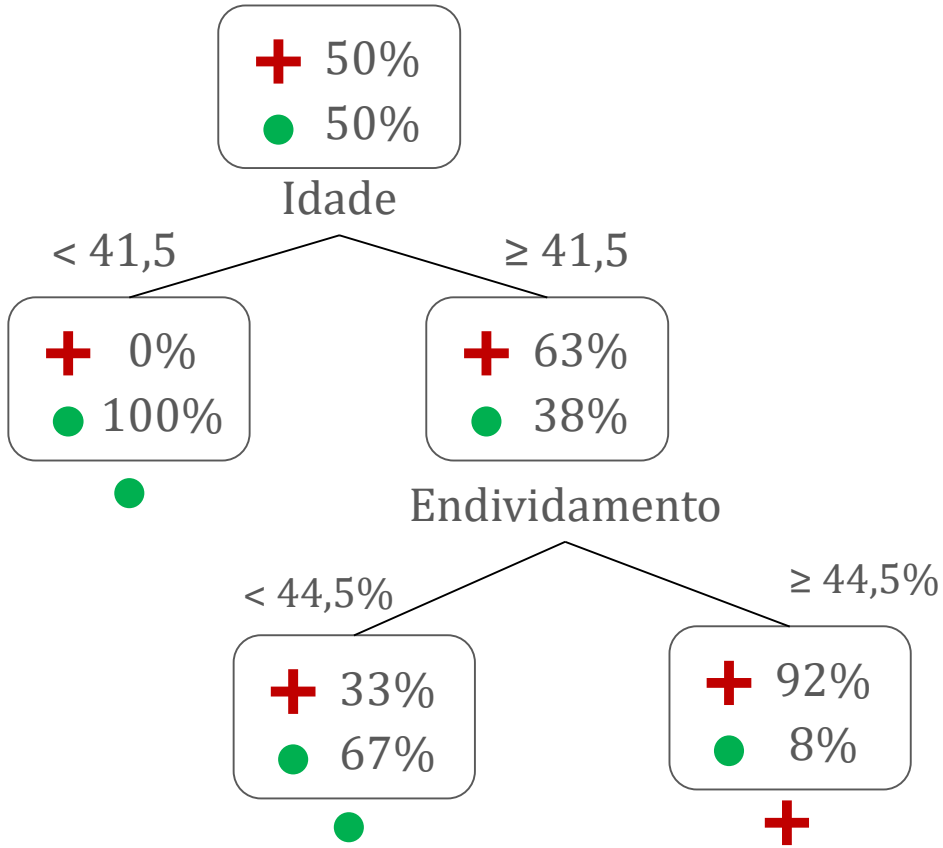
2ª quebra

Árvores de decisão

Intuição

- Aqui vamos simular uma distribuição hipotética



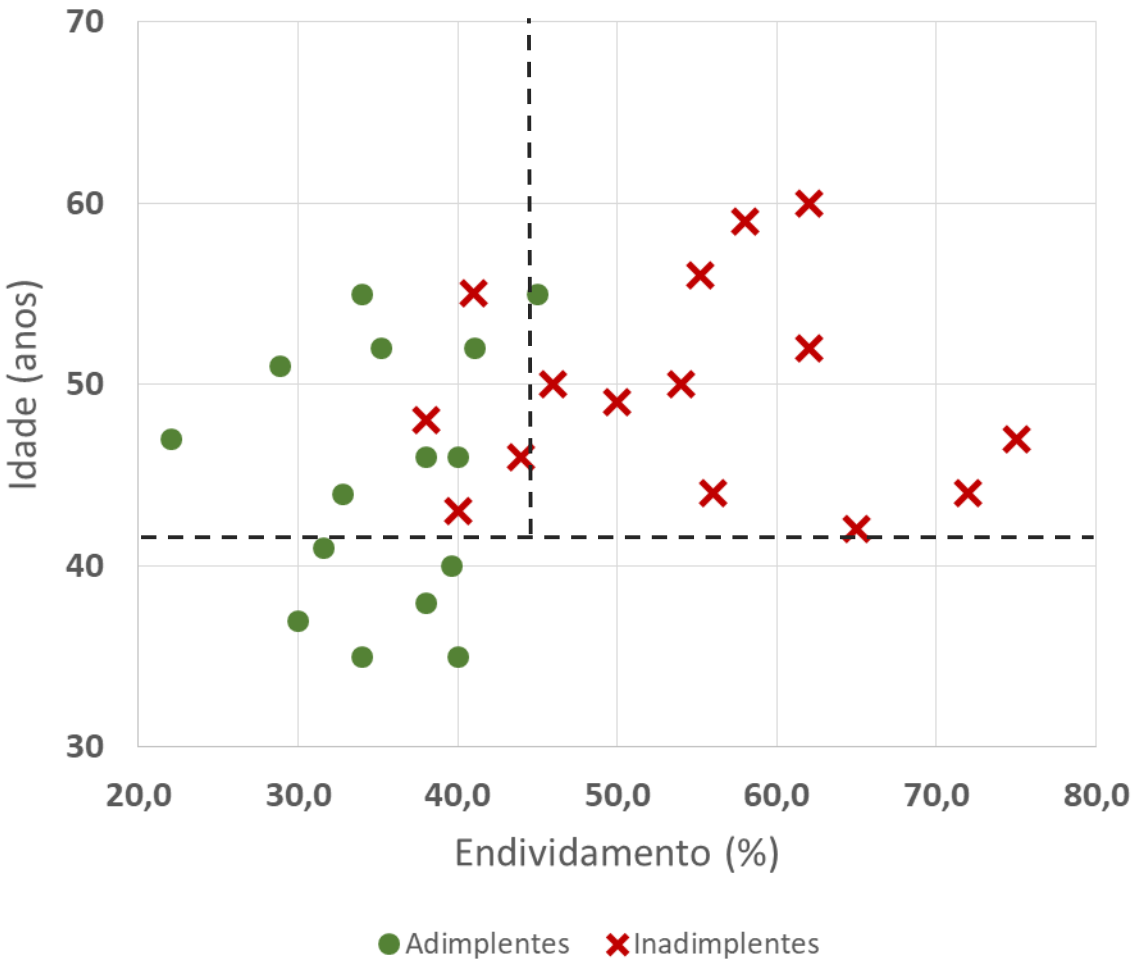


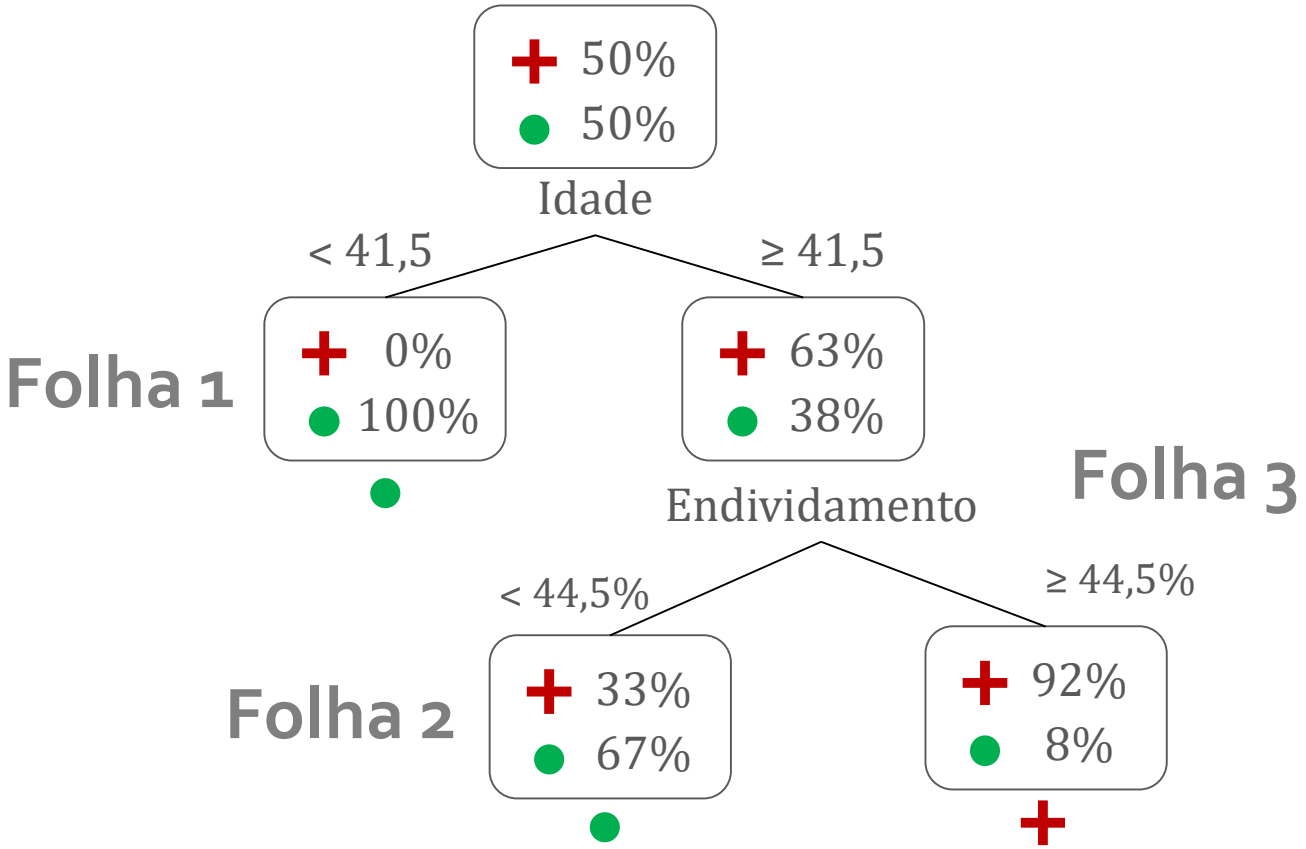
Lembrete: a classificação vai sempre ser feita pela categoria majoritária no nó final

Árvores de decisão

Intuição

- Aqui vamos simular uma distribuição hipotética





Lembrete: a classificação vai sempre ser feita pela categoria majoritária no nó final

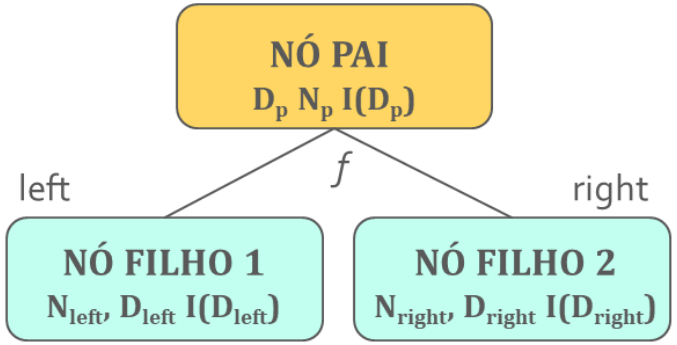
Árvores de decisão

Particionamento dos nós nas árvores [extra]

- Como mencionado anteriormente no processo da construção da árvore de decisão, o algoritmo vai realizando particionamentos sucessivos dos nós
- Existem diversos critérios entre as tarefas de regressão e classificação que permitem fazer a seleção da variável que melhor conduz à partição do nó
- Esses critérios na verdade conduzem à maximização de uma função durante a aprendizagem do algoritmo, que é denominada *information gain* (IG)

$$IG(D_p, f) = \boldsymbol{I(D_p)} - \left(\frac{N_{left}}{N_p} \boldsymbol{I(D_{left})} + \frac{N_{right}}{N_p} \boldsymbol{I(D_{right})} \right)$$

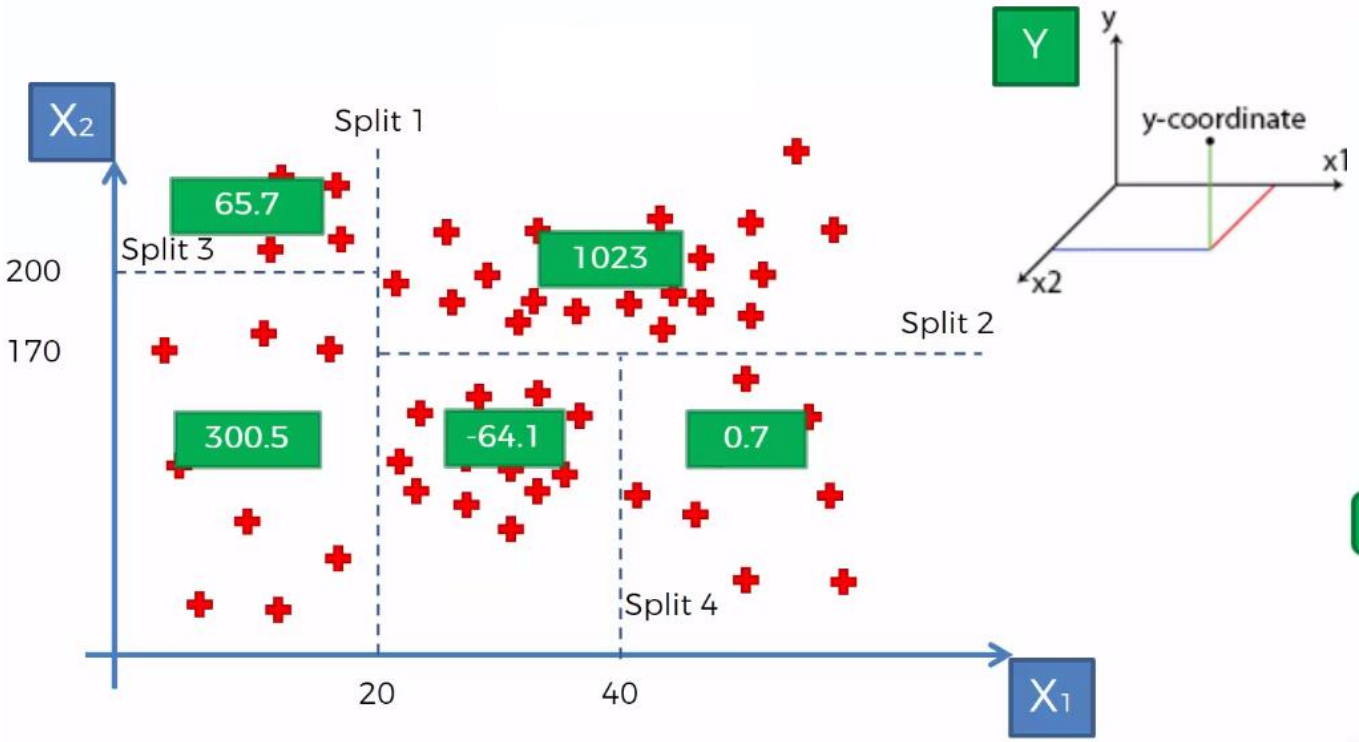
O ganho de informação é dado pela diferença de impureza do nó pai e dos nós filhos. Na expressão D_p, D_{left}, D_{right} são as observações que encontram-se nos nós pais e filhos. *f* é a *feature* usada na quebra, e *I* é uma medida de impureza do nó. N_p, N_{left} e N_{right} são os números de observações nos nós.

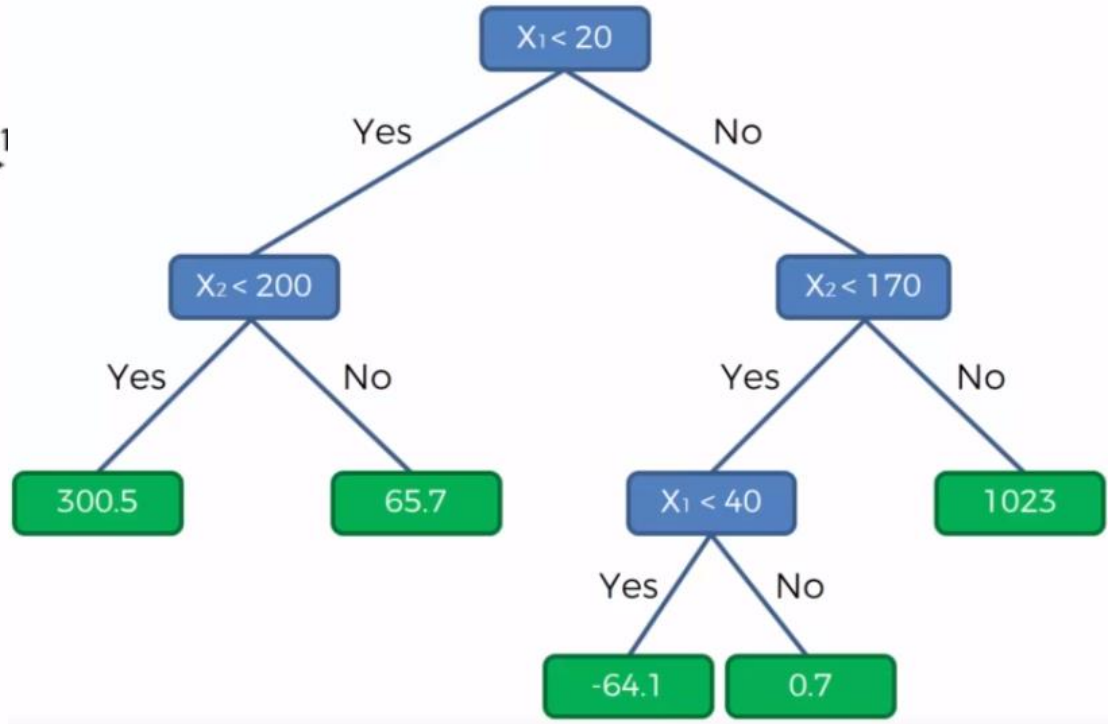


Árvores de decisão

Particionamento para regressão

- A árvore de decisão usada para regressão são usadas para *output* numérico contínuo
- O valor obtido nos nós finais é dado pela média das observações que estão contidas dentro daquele nó



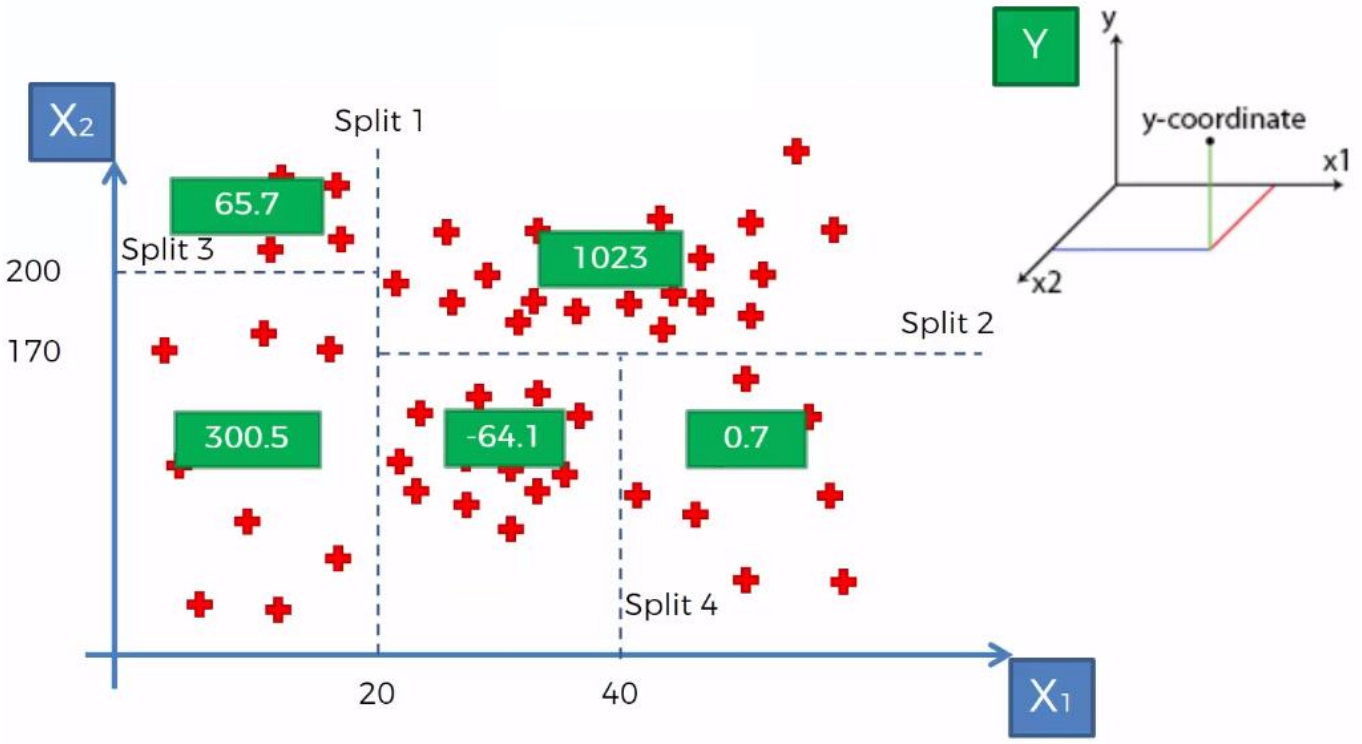


Para regressão a medida de impureza deve ser adequada para o *target* y contínuo. Ela é dada pelo erro médio quadrático (MSE) ponderados nós filhos (em algumas implementações pode se usar a variância)

Árvores de decisão

Particionamento para regressão

- A árvore de decisão usada para regressão são usadas para *output* numérico contínuo
- O valor obtido nos nós finais é dado pela média das observações que estão contidas dentro daquele nó



$$MSE = \frac{1}{N_t} \sum_{i \in D_t} (y_i - \bar{y}_t)^2$$

$$\bar{y}_t = \frac{1}{N_t} \sum_{i \in D_t} y_i$$

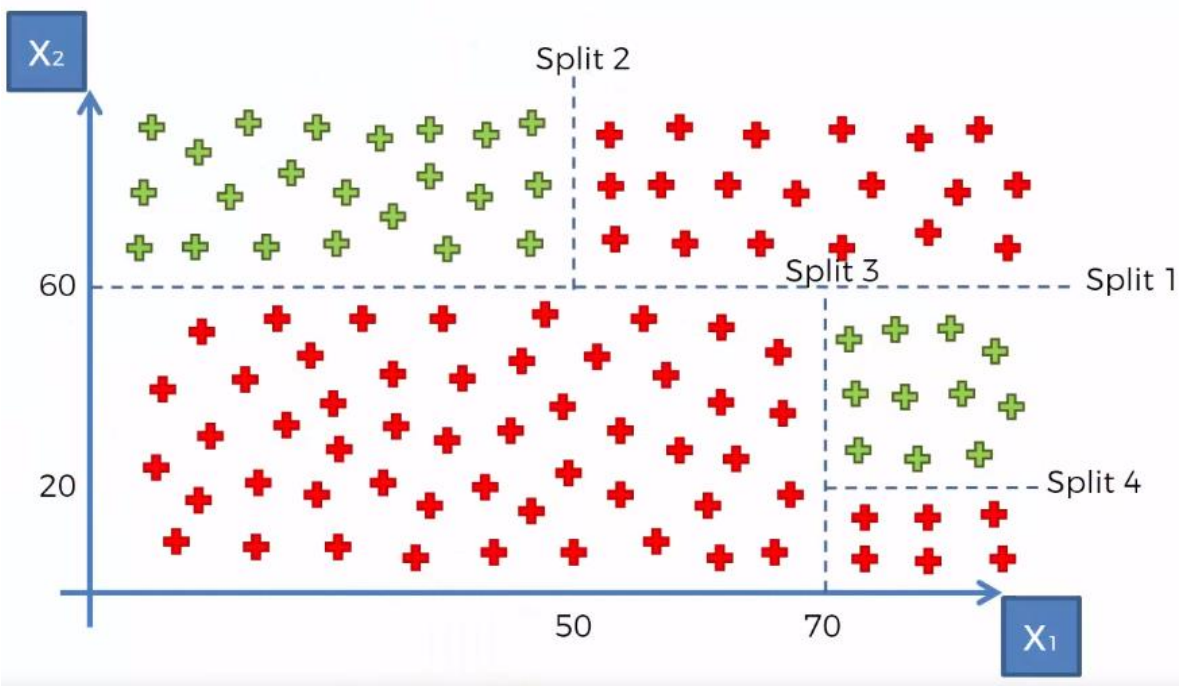
Para regressão a medida de impureza deve ser adequada para o *target* y contínuo. Ela é dada pelo erro médio quadrático (MSE) ponderados nós filhos (em algumas implementações pode se usar a variância)

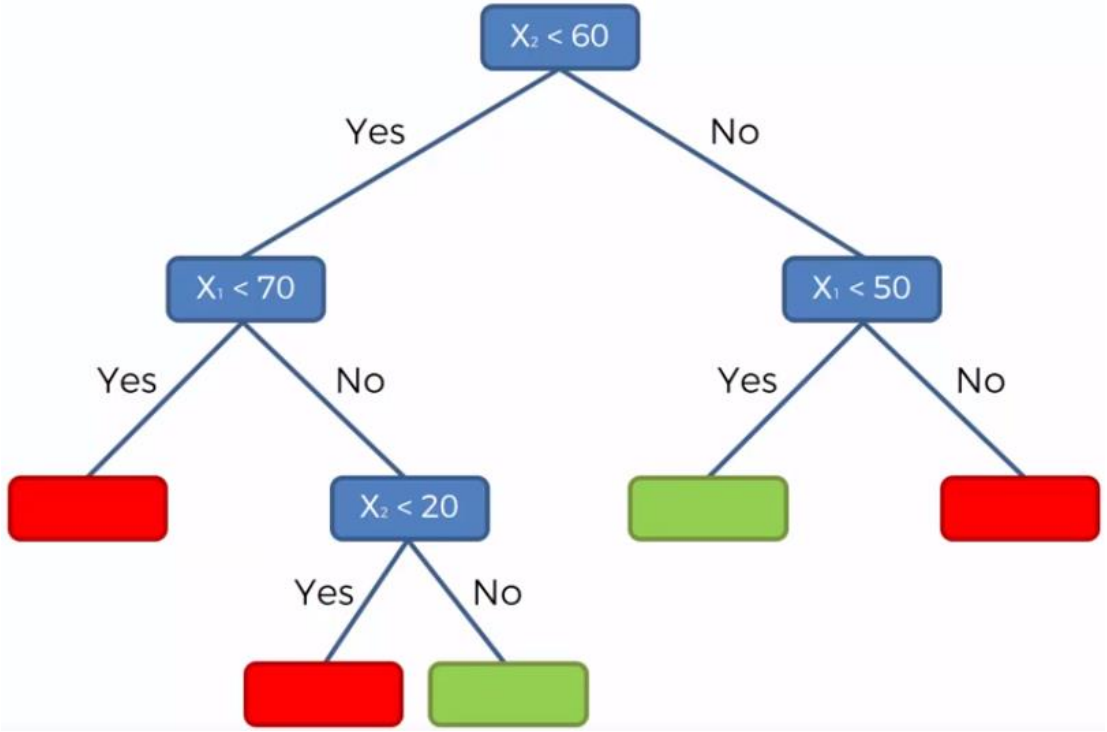
https://medium.com/pursuitnotes/decision-tree-regression-in-6-steps-with-python-1a1c5aa2ee16

Árvores de decisão

Particionamento para classificação

- A árvore de decisão usada para classificação são usadas para *output* categórico (*labels*)
- O *label* nos nós finais é dado pela categoria dominante das observações dentro daquele nó



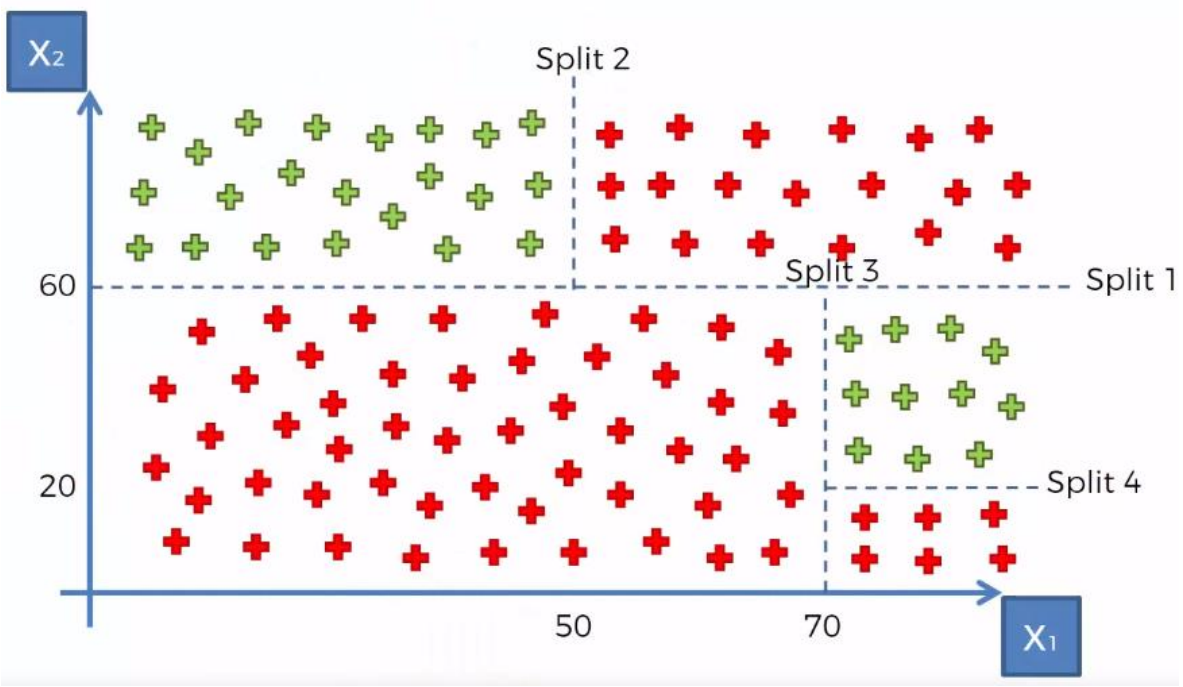


Para classificação a medida de impureza deve ser adequada para o *target* y contínuo. Ela é dada pelo índice de impureza de Gini ponderados nós filhos (em algumas implementações pode se usar a entropia)

Árvores de decisão

Particionamento para classificação

- A árvore de decisão usada para classificação são usadas para *output* categórico (*labels*)
- O *label* nos nós finais é dado pela categoria dominante das observações dentro daquele nó



$$IG(p) = 1 - \sum_{i=1}^{labels} p_i^2$$

$$p_i = \frac{n_i}{N_t}$$

Para classificação a medida de impureza deve ser adequada para o *target* y contínuo. Ela é dada pelo índice de impureza de Gini ponderados nós filhos (em algumas implementações pode se usar a entropia)

Árvores de decisão

Controlando o desenvolvimento das árvores

- Após a construção da árvore, um passo de poda pode ser realizado afim de reduzir a complexidade e o tamanho da árvore de classificação.
- Árvores muito grandes são suscetíveis ao fenômeno de *overfitting*; o corte de galhos da árvore pode contribuir para uma melhora da capacidade de generalização da árvore de classificação.
- Árvores com muitos nós terminais e poucos indivíduos em cada um também apresentam *overfitting*.

Arvores muito simples, com pouca profundidade possuem erros elevados porém consistentes nas amostras de treino e teste

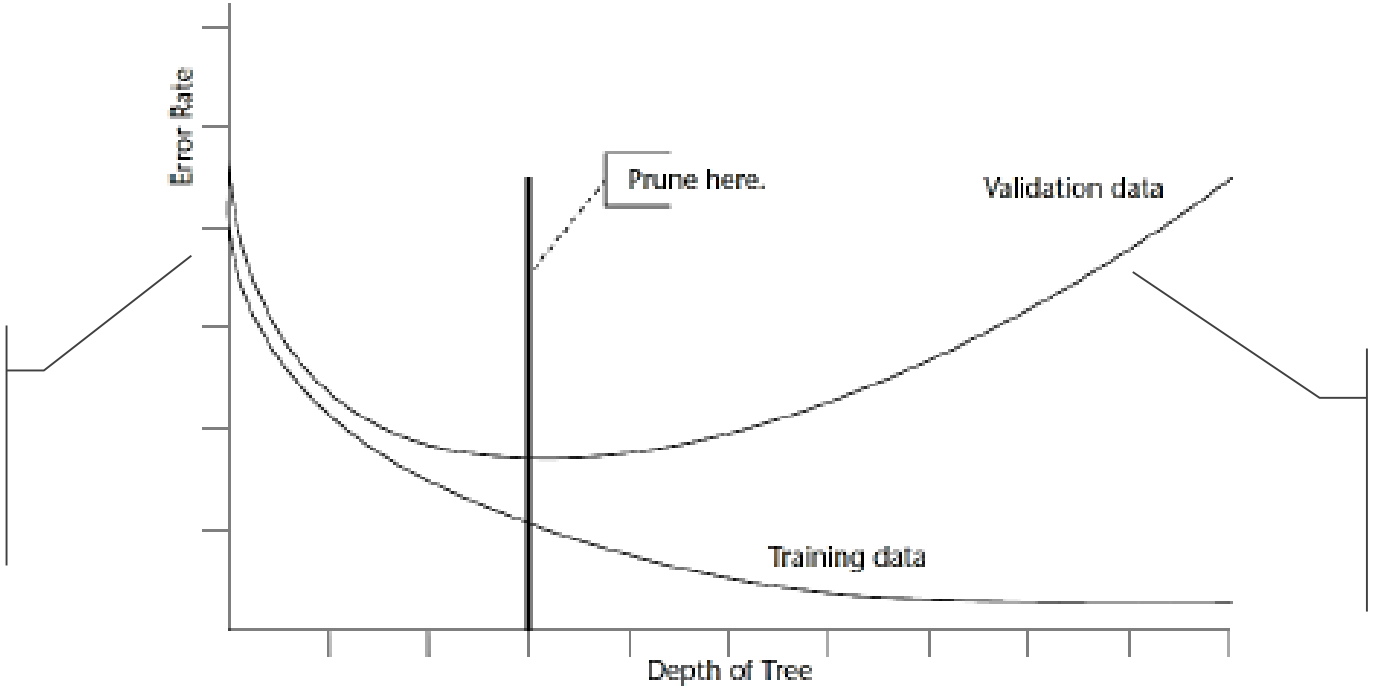


Figure 6.7 Pruning chooses the tree whose miscalculation rate is minimized on the validation set.

Arvores muito complexas, com bastante profundidade possuem erros elevados na amostra de teste e possui um ajuste quase perfeito na amostra de treino

Linoff & Berry, Data Mining Techniques, Wiley

Árvores de decisão

Controlando o desenvolvimento das árvores

Pré-poda

- **O algoritmo interrompe o processo de construção da árvore quando:**
- a redução da medida de impureza não é mais significativa
- o número de observações em um nó a ser particionado atinge um mínimo pré-definido
- a partição, independente da *feature* escolhida gera nós filhos com um número de observações menor que um valor pré-determinado
- atinge uma profundidade máxima
- a partição gera nós filhos com distribuições que não são significantemente diferentes em relação ao nó pai

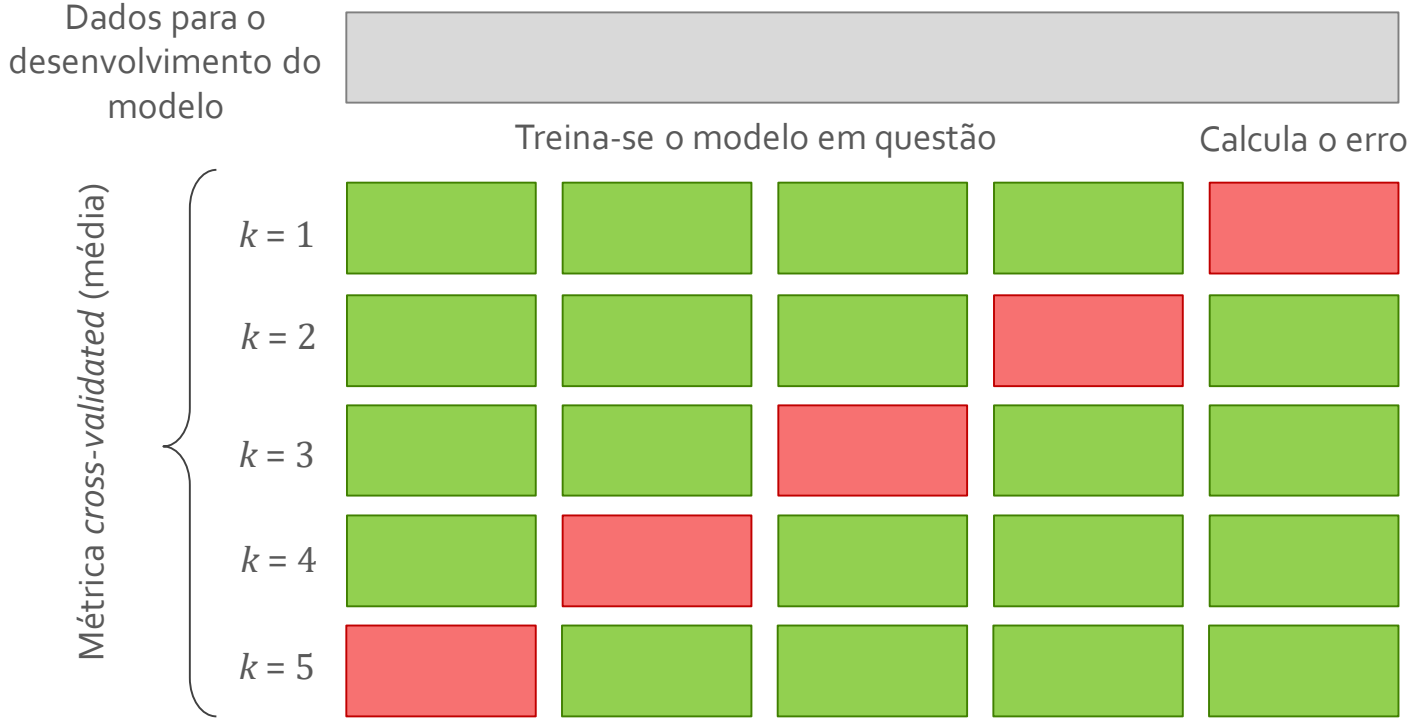
Pós-poda

- **Aqui o algoritmo constrói a árvore com todo o particionamento possível**
- os ramos inferiores da árvore vão sendo substituídos por nós terminais
- caso não haja o comprometimento de performance na amostra de teste, podam-se esses ramos
- o algoritmo seleciona sub-árvores candidatas à poda por um critério que leva em consideração o aumento do erro ao podar o ramo em questão e a redução do nó
- evita-se árvores de grande complexidade
- método preferível ao pré-poda por permitir o desenvolvimento da árvore na sua forma mais completa

Árvores de decisão

Poda da árvore usando *cross validation* [extra]

- Uma das formas de poda da árvore é usando a técnica de otimização com base no valor cp (*complexity parameter*) que leva em consideração o número de folhas terminais
- Essa técnica é baseada no *minimal cost complexity pruning* que leva em consideração o número de quebras e o erro de predição.
- O objetivo é construir uma árvore com o melhor número de quebras que conduza a menor ocorrência de *overfitting* e à menor taxa de erro



$$\sum_{T\ terminal\ nodes} erro_i + \lambda\ splits$$

O método subdivide o dado usado na aprendizagem do modelo em *k* sub-amostras de tamanho igual (*folds*)
Das *k* sub-amostras, *k*-1 são usadas para treinar o modelo e um é usada como validação. O processo é repetido *k* vezes, sendo que a cada vez uma porção do dado é usada para validação e o restante para treinamento. No final faz-se a média do erro

Árvores de decisão

Prós e contras do algoritmo

Prós

É uma técnica que não assume linearidade e nem normalidade

Pode lidar com variáveis quantitativas e qualitativas (em muitas implementações)

Visualização e interpretabilidade simples

Facilidade do entendimento das regras construídas (facilidade de interpretação pelos tomadores de decisão)

Consegue lidar com *missing data* com mais facilidade

Contras

Os resultados são bastante dependentes do conjunto de dados de treino (i.e. alta variância)

Podem apresentar grande instabilidade em bases de dados muito pequenas

Podem apresentar *overfitting*

Podemos correr o risco de não ter sido encontrada a melhor árvore

Prática no RStudio

Prof. Dr. João Rafael Dias Pinto

...foco de hoje

- **Treinando os algoritmos de árvore de decisão sobre as bases de estudo**

  Criando as amostras de treino e teste. Ajuste dos algoritmos, aprimoramento dos resultados, poda e visualização da árvore. Avaliações dos *outputs* dos modelos