Pontifical Catholic University of Rio de Janeiro
Department of Industrial Engineering
Postgraduate Program in Production Engineering

Pontifícia Universidade Católica do Rio de Janeiro

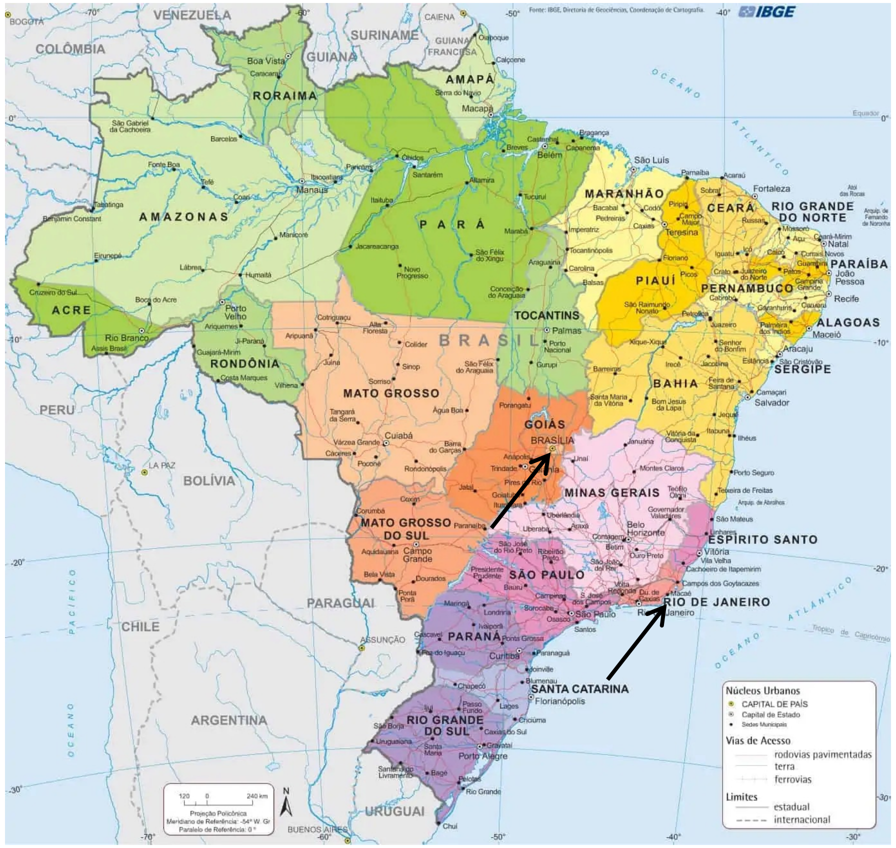
Introductory presentation
+
Decision trees: using optimization to enhance performance

Flávio Araújo Lim-Apo

Espoo
May 27, 2024

Brazil

4.394 km



4.319 km

Brasília

HEL-FRA: 1.500km
HEL-LIS: 3.360km

Education

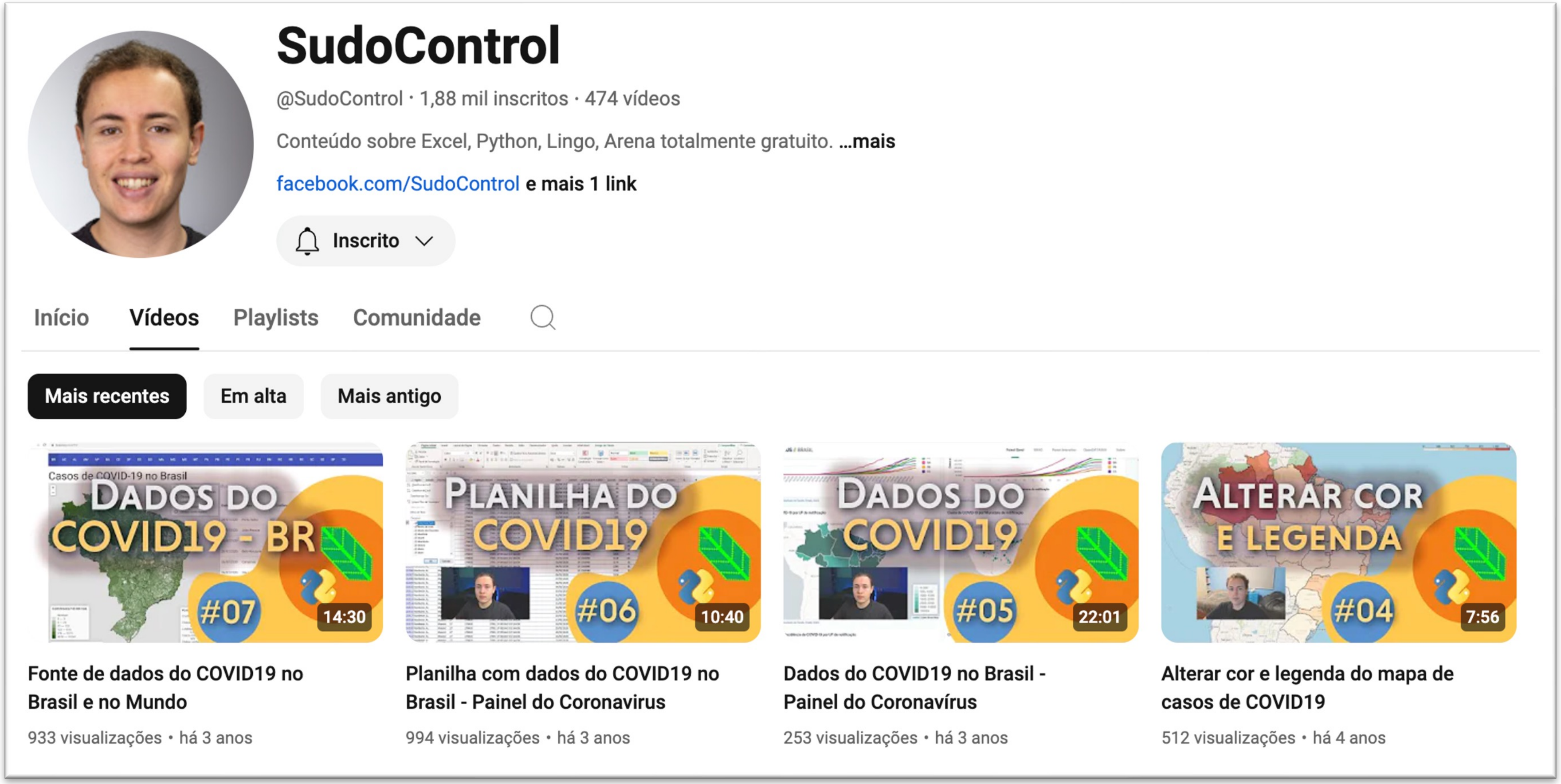
- BSc in Business Administration, 2013-2017
  - University of Brasilia (UnB)
  - “Data processing at major events: participant allocation”

- MSc in Production Engineering (emph. in Transportation and Logistics), 2019-2021
  - Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
  - “Allocation of qualified employees on inspection missions of a regulatory agency”

- PhD student in Production Engineering (emph. in Operations Research), 2021-2025
  - Pontifical Catholic University of Rio de Janeiro (PUC-Rio)

YouTube

- While I was doing my bachelor, I created a YouTube channel and posted at least one video per day for a year
  - Operational Research, LINGO, What'sBest!, Solver...
  - Python, folium (package for maps), Excel...



SudoControl

@SudoControl · 1,88 mil inscritos · 474 vídeos

Conteúdo sobre Excel, Python, Lingo, Arena totalmente gratuito. ...mais

facebook.com/SudoControl e mais 1 link

Fonte de dados do COVID19 no Brasil e no Mundo
933 visualizações • há 3 anos

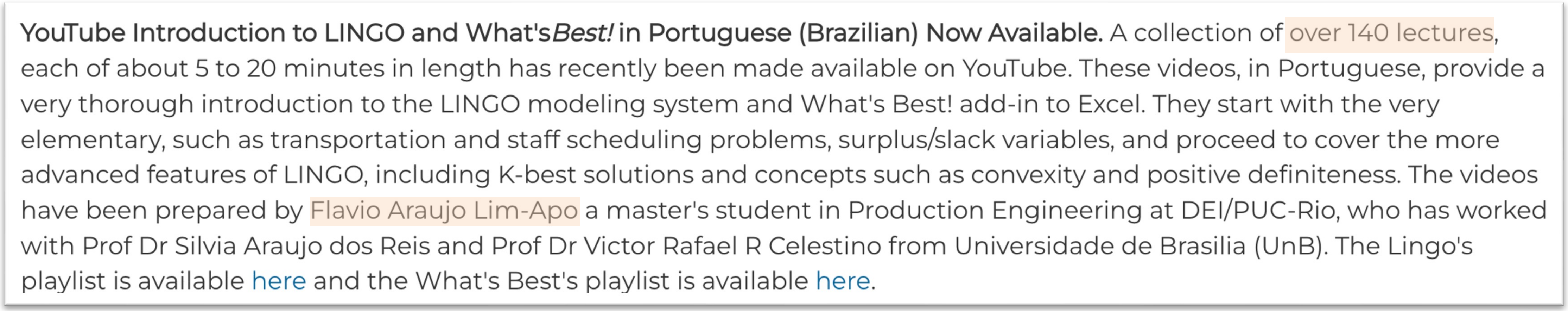
Planilha com dados do COVID19 no Brasil - Painel do Coronavirus
994 visualizações • há 3 anos

Dados do COVID19 no Brasil - Painel do Coronavírus
253 visualizações • há 3 anos

Alterar cor e legenda do mapa de casos de COVID19
512 visualizações • há 4 anos

Lindo

LINDO SYSTEMS INC.



YouTube Introduction to LINGO and What's*Best!* in Portuguese (Brazilian) Now Available. A collection of over 140 lectures, each of about 5 to 20 minutes in length has recently been made available on YouTube. These videos, in Portuguese, provide a very thorough introduction to the LINGO modeling system and What's Best! add-in to Excel. They start with the very elementary, such as transportation and staff scheduling problems, surplus/slack variables, and proceed to cover the more advanced features of LINGO, including K-best solutions and concepts such as convexity and positive definiteness. The videos have been prepared by Flavio Araujo Lim-Apo a master's student in Production Engineering at DEI/PUC-Rio, who has worked with Prof Dr Silvia Araujo dos Reis and Prof Dr Victor Rafael R Celestino from Universidade de Brasilia (UnB). The Lingo's playlist is available here and the What's Best's playlist is available here.

https://www.lindo.com/

Decision trees: using optimization to enhance performance

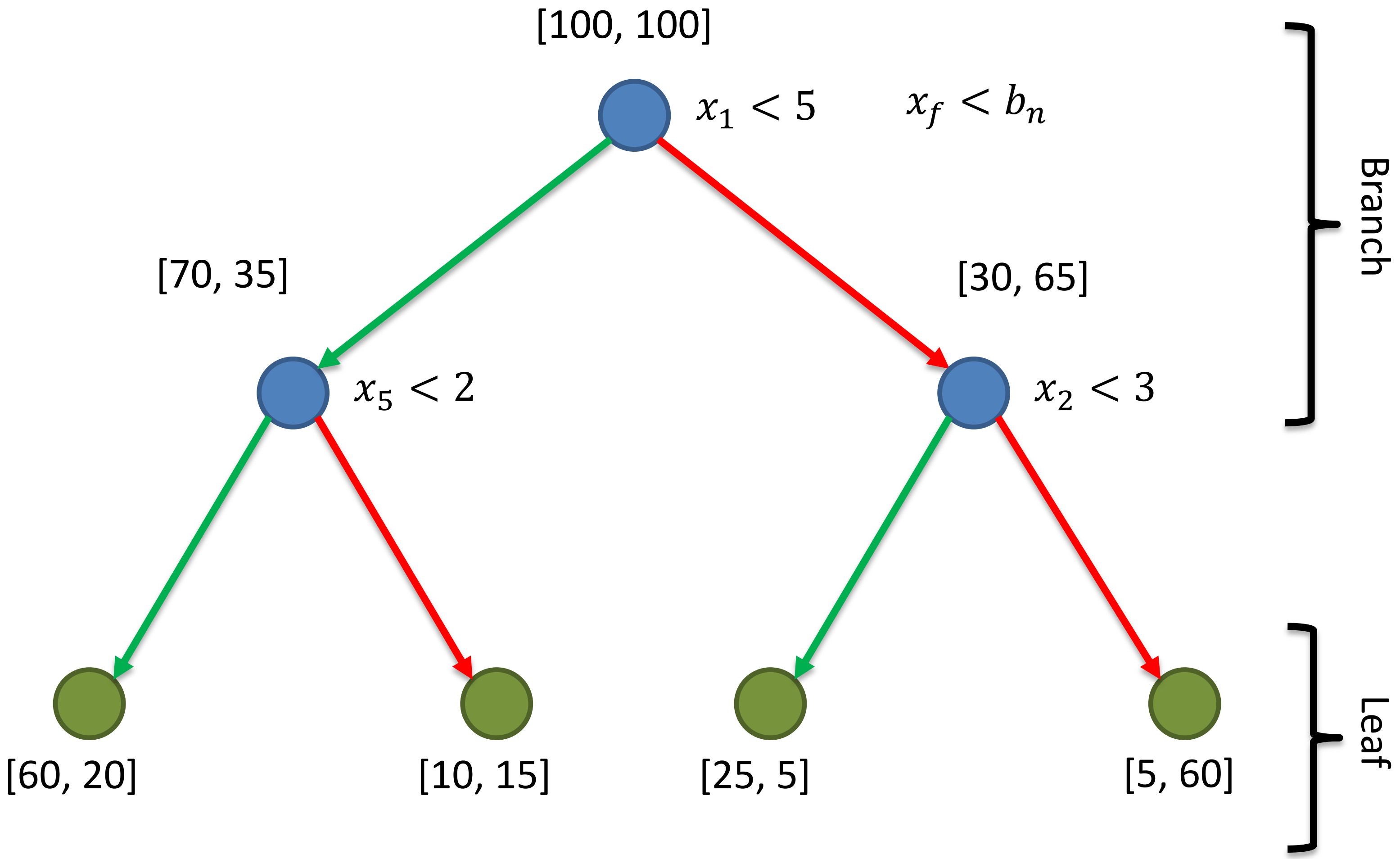
Introduction

- Predictive models identify patterns in historical data that allow prediction of future data
    - Predictive models can be interpreted or black box.

- Over the last 30 years decision trees have become among the most popular techniques for interpretable machine learning (Rudin, 2019). CART (Breiman et al., 1984), ID3 (Quinlan, 1986); C4.5 (Quinlan, 2014).

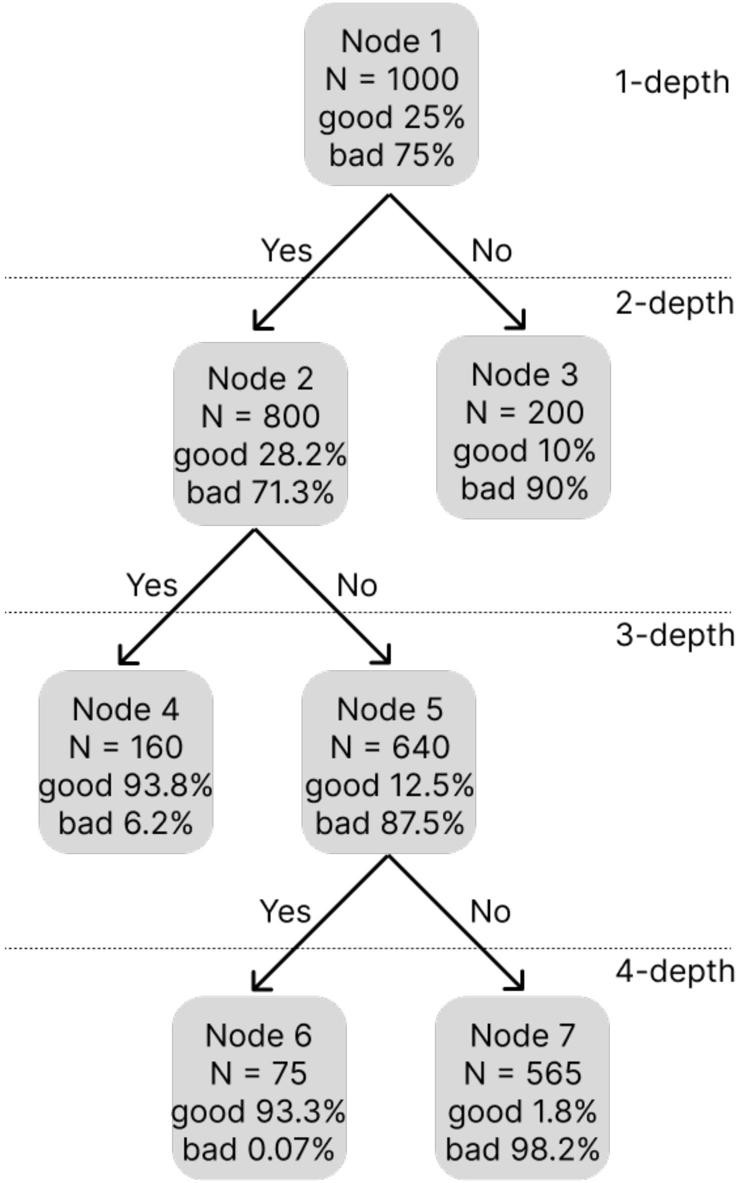
- Decision trees are an off-the-shelf procedure for data mining (Hastie et al., 2009)

Decision tree



Tradicional decision trees

- Traditional decision tree models create separation rules by level
  - To create the cuts, an impurity measure is used.
- For each level the rules are created, if one of the stopping criteria is not reached the procedure continues.
  - Depth;
  - Number of elements in the node;
  - Reach an impurity value
- The method is greedy, once the cutting variables are defined they are not changed



Tree Optimization

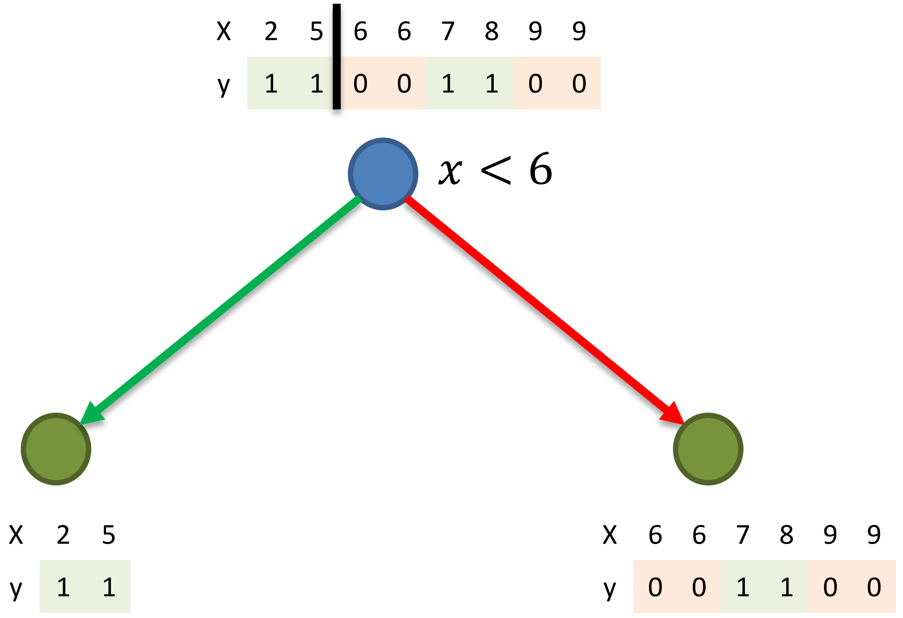
- The Optimal classification trees problem can have some categories:
  - **Traditional OCT** allows continuous or discrete input data (Bertsimas, 2017);
  - **BinOCT** formulation when the input data is binary (Verwer, 2019);
  - **OCT with hyperplanes**, split can use more than one variable (multivariate splits) (Bertsimas, 2017);
  - **FlowOCT** uses max-flow/min-cut duality to derive a Benders' decomposition (Aghaei, 2022).

- Explainability properties can also be used to understand the final model. There are studies on the sparsity and local interpretability of the tree model (Lundberg, 2018; Lundberg, 2020; Molnar, 2020; Ribeiro, 2016)

- Some approaches binarize the input data so that the decision tree defines the variable that should be used in each leaf branch without defining the split value (Aglin, 2020; Lin, 2020).

Tree problem

Tradicional methods does not consider the depth of the tree



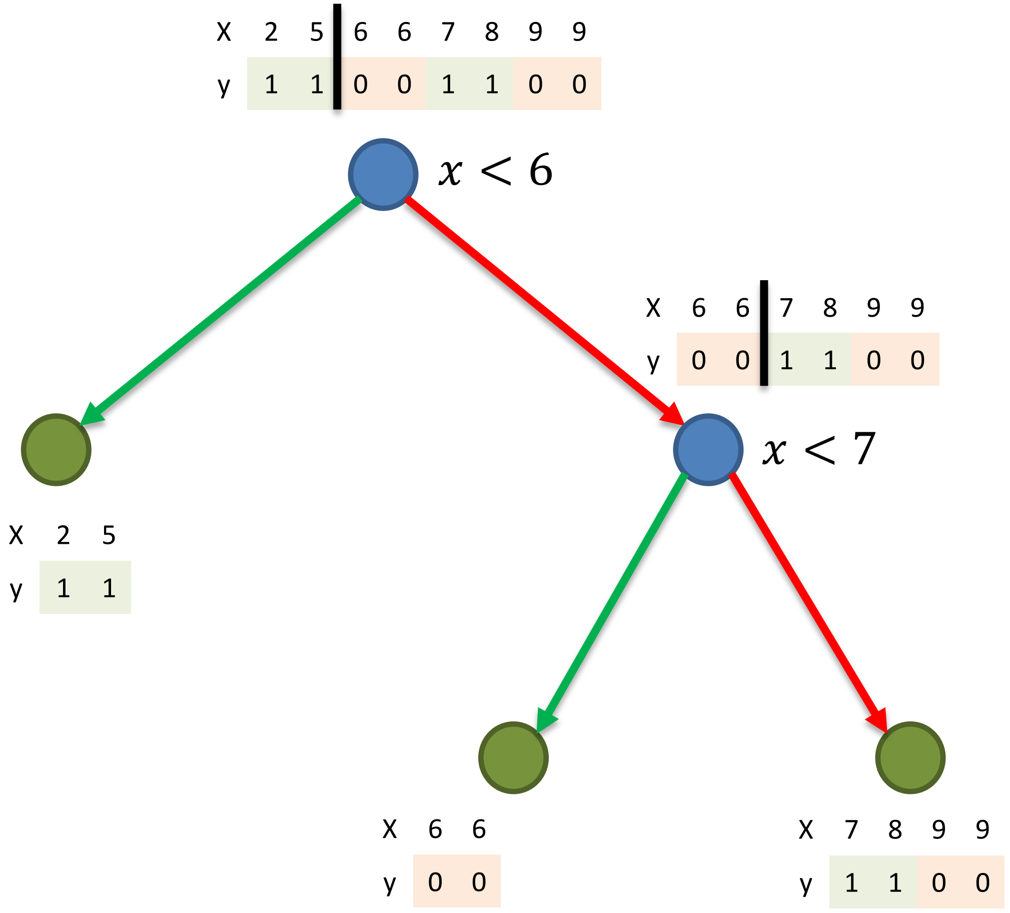
X	2	5	6	6	7	8	9	9
y	1	1	0	0	1	1	0	0

$x < 6$

X	2	5
y	1	1

X	6	6	7	8	9	9
y	0	0	1	1	0	0

$$Acc = \frac{6}{8} = 75\%$$



X	2	5	6	6	7	8	9	9
y	1	1	0	0	1	1	0	0

$x < 6$

X	2	5
y	1	1

X	6	6	7	8	9	9
y	0	0	1	1	0	0

$x < 7$

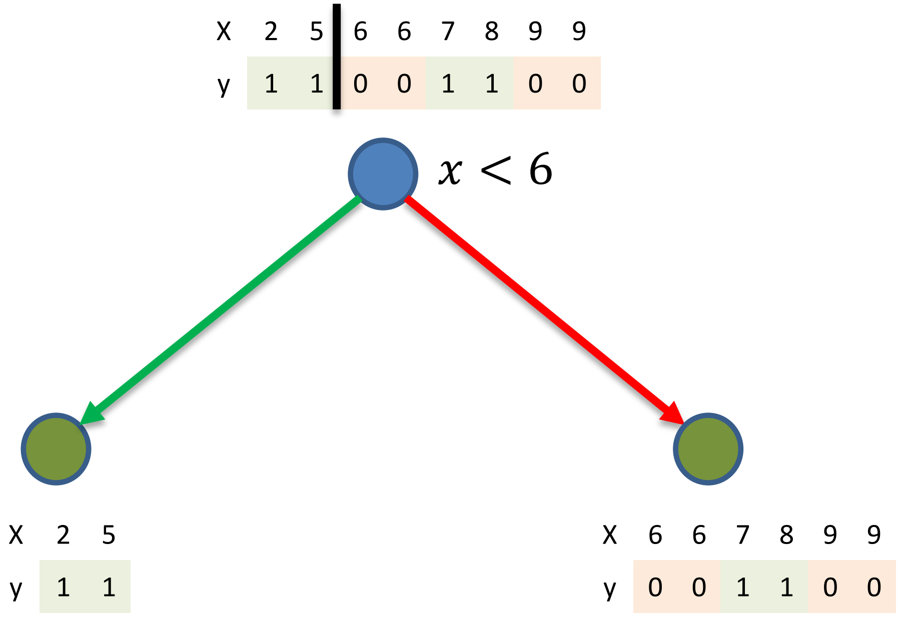
X	6	6
y	0	0

X	7	8	9	9
y	1	1	0	0

$$Acc = \frac{6}{8} = 75\%$$

Tree problem - proposed

Optimization considers the depth of the tree



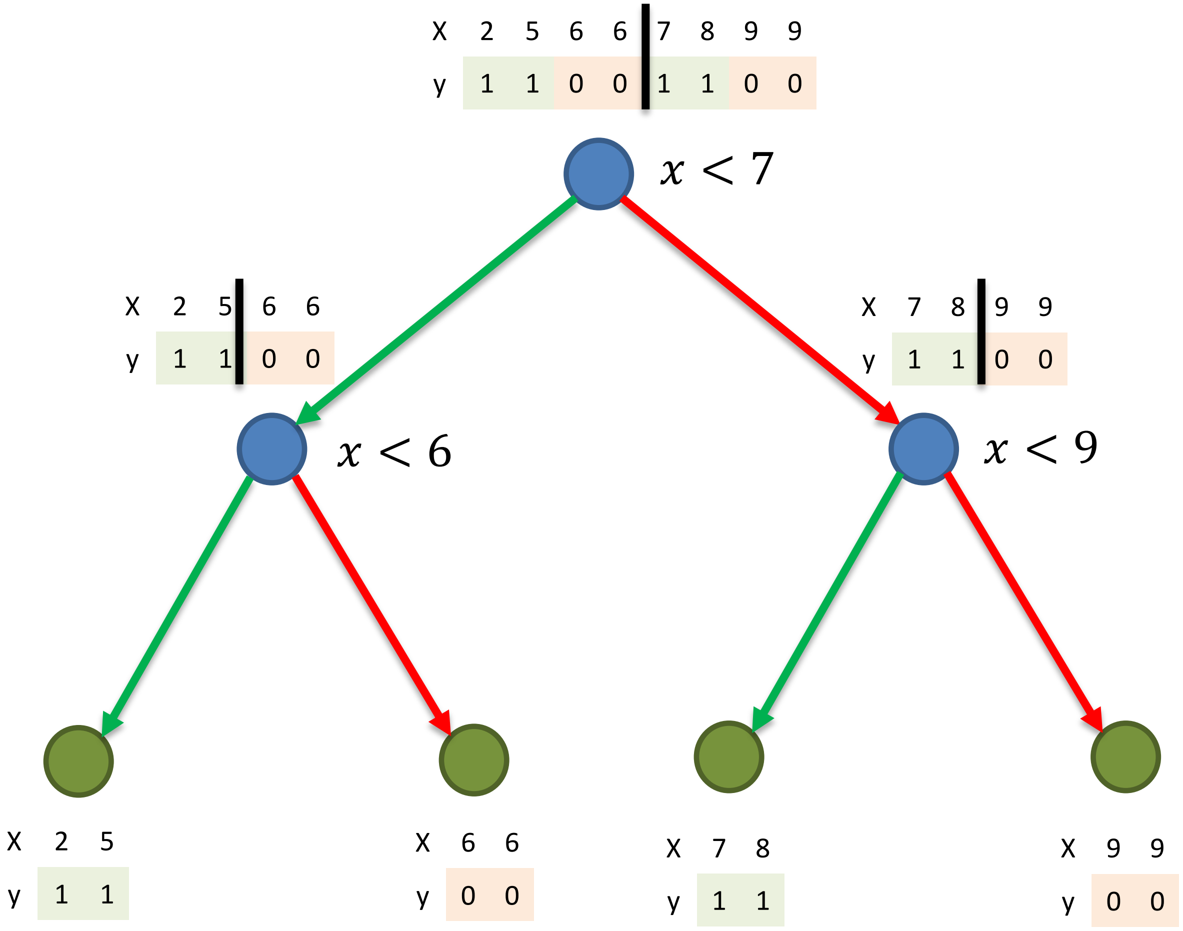
X	2	5	6	6	7	8	9	9
y	1	1	0	0	1	1	0	0

$x < 6$

X	2	5
y	1	1

X	6	6	7	8	9	9
y	0	0	1	1	0	0

$$Acc = \frac{6}{8} = 75\%$$



X	2	5	6	6	7	8	9	9
y	1	1	0	0	1	1	0	0

$x < 7$

X	2	5	6	6
y	1	1	0	0

$x < 6$

X	7	8	9	9
y	1	1	0	0

$x < 9$

X	2	5
y	1	1

X	6	6
y	0	0

X	7	8
y	1	1

X	9	9
y	0	0

$$Acc = \frac{8}{8} = 100\%$$

Objective

- How can we enhance decision tree training using optimization?

- The research aims to combine CART with a mixed-integer linear program to improve a decision tree.

1. Using mathematical modeling to optimize the decision tree;
2. Comparing the proposed enhancer results with traditional CART in literature datasets.

Methodology

- The method uses CART as an initial solution. It is necessary to inform the maximum desired depth of the tree

1. **Quick viable solution.** CART finds an initial solution that will be partially used in optimization.

2. **Decision Tree Enhance (DTE).** Fixe the values of the matrix (a) obtained with CART (the variables that are used in each branch node of the decision tree). And optimize the split value of each node (b_n)

Methodology

- The optimization model receives matrix A as input, which contains the variables that are used in each branch node.

- These values are fixed, reducing the complexity of the problem, and the model is optimized

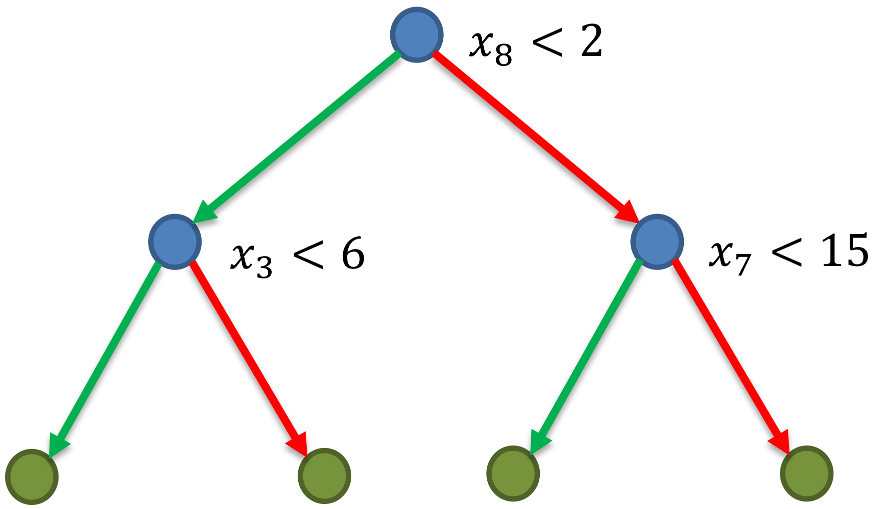
- The solution obtained by DTE is a local optimum

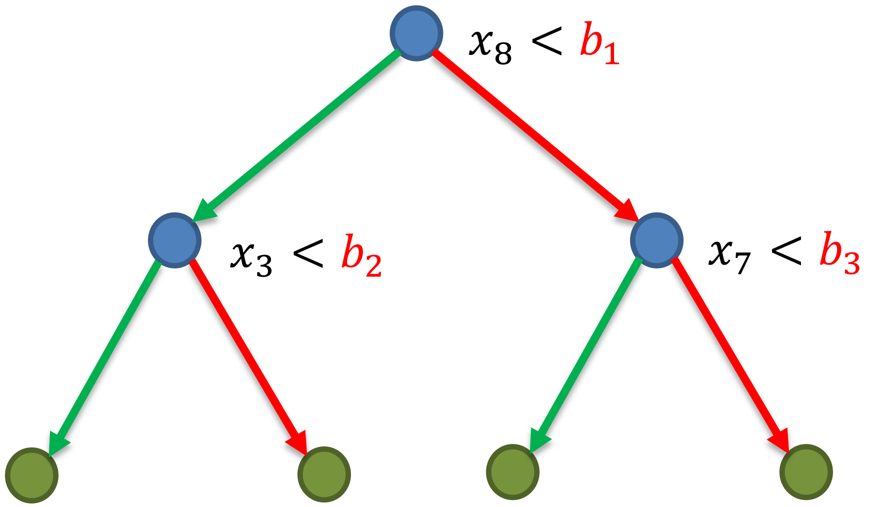
- With this, it is possible to obtain the optimal solution for that Matrix A, which yet does not guarantee that this is the global optimum.

- **Fix the features of each branch node (A) and find the best-split value (b).**

Methodology

- Train a tree using CART. Features used in each node are fixed, and the split value of each branch node can be optimized.





Methodology - datasets

- 53 datasets from the UCI Machine Learning Repository already used in the literature were used to verify the performance of the proposed techniques

- The implementation was developed in Julia using JuMP and Gurobi 10.0.2. Tests were performed on a MacBook Air M2 with 8 CPU cores and 8 GB RAM.

- Training (75%) and validation (25%).

- We removed all elements with NA and hot-encoded all categorical variables for data preparation. Continuous values do not need to be normalized.

Methodology – model

X[1]	X[2]	y
5	2	1
10	9	1
10	5	0
22	9	0
7	1	1

Xss[1]	Xss[2]	y
1	2	1
3	4	1
3	3	0
4	4	0
2	1	1

$$Max \sum_{co \in CO} Qt_{co} \tag{1}$$

$$Qt_{co} \leq Q_{co,k} + M \times Qb_{co,k} \quad \forall\ co \in Co, k \in K \tag{2}$$

$$\sum_{k \in K} Qb_{co,k} = |K| - 1 \quad \forall\ co \in Co \tag{3}$$

$$Q_{co,k} = \sum_{i \in I} yb_{k,i} \times Z_{i,co} \quad \forall\ co \in Co, k \in K \tag{4}$$

$$\sum_{co \in Co} Z_{i,co} = 1 \quad \forall\ i \in I \tag{5}$$

$$Xss_{tb,i} + \theta \leq S_{tb} + \sum_{rn \in RL_{tb}} Z_{i,rn} \quad \forall\ tb \in TB, i \in I \tag{6}$$

$$Xss_{tb,i} + \theta \geq S_{tb} + \sum_{ln \in LL_{tb}} Z_{i,ln} \quad \forall\ tb \in TB, i \in I \tag{7}$$

$$Z_{i,co} \in \mathbb{Z}^{+} \quad \forall\ i \in I, co \in Co \tag{8}$$

Methodology – model

$$Max \sum_{co \in CO} Qt_{co} \tag{1}$$

$$Qt_{co} \leq Q_{co,k} + M \times Qb_{co,k} \quad \forall\ co \in Co, k \in K \tag{2}$$

$$\sum_{k \in K} Qb_{co,k} = |K| - 1 \quad \forall\ co \in Co \tag{3}$$

$$Q_{co,k} = \sum_{i \in I} yb_{k,i} \times Z_{i,co} \quad \forall\ co \in Co, k \in K \tag{4}$$

$$\sum_{co \in Co} Z_{i,co} = 1 \quad \forall\ i \in I \tag{5}$$

$$Xss_{tb,i} + \theta \leq S_{tb} + \sum_{rn \in RL_{tb}} Z_{i,rn} \quad \forall\ tb \in TB, i \in I \tag{6}$$

$$Xss_{tb,i} + \theta \geq S_{tb} + \sum_{ln \in LL_{tb}} Z_{i,ln} \quad \forall\ tb \in TB, i \in I \tag{7}$$

$$Z_{i,co} \in \mathbb{Z}^{+} \quad \forall\ i \in I, co \in Co \tag{8}$$

(1) - Objective function maximizes the sum of the winning classes of each leaf node.

(2) - indicates each leaf node's total number of right predictions. The value of Qt is limited by the class with the largest number of Q elements. The number of right predictions for each node is equal to the number of the class that had the most elements at that node.

(3) - defines the classes that were not winners; there can be only one winning class in each node. When Qb = 1, it indicates that that class was not the majority class of that node.

(4) - indicates the total number of elements of each class that we have in each leaf node

(5) - each element must be allocated to exactly one leaf node.

(6) e (7) defines the tree's structure and which rules must be followed for each element to reach a leaf node.

Results

- ➢ 53 datasets, 1-5 depth = 265
- ➢ When there is an improvement, DTE is better than CART by 1.9% in-sample data and 0.8% out-of-sample.
- ➢ Improvement in the accuracy of in-sample and out-of-sample data

Fig 1. Number of datasets that found the optimal, overall improvement in train and test

		1 second			5 seconds			10 seconds			30 seconds			600 seconds		
Data	Datasets	Opt.	Train	Test	Opt.	Train	Test	Opt.	Train	Test	Opt.	Train	Test	Opt.	Train	Test
Categorical	45	22	0,0%	0,0%	29	0,0%	0,0%	32	0,0%	0,0%	37	0,0%	0,0%	40	0,0%	0,0%
Continuous	155	61	0,4%	0,3%	71	0,6%	0,5%	79	0,6%	0,5%	89	0,7%	0,8%	109	0,9%	0,7%
Mixed	65	26	0,5%	-0,2%	30	0,7%	-0,3%	33	0,8%	-0,4%	35	0,8%	-0,2%	44	1,0%	-0,4%
Total	265	41,1%	0,3%	0,1%	49,1%	0,5%	0,2%	54,3%	0,5%	0,2%	60,8%	0,7%	0,3%	72,8%	0,8%	0,3%

Fig 2. Number of datasets that had improvement, improvement in train and test

		1 second			5 seconds			10 seconds			30 seconds			600 seconds		
Data	Datasets	Imp.	Train	Test	Imp.	Train	Test	Imp.	Train	Test	Imp.	Train	Test	Imp.	Train	Test
Categorical	45	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%
Continuous	155	32	1,7%	1,8%	48	1,8%	1,8%	54	1,7%	1,6%	64	1,8%	1,7%	75	1,8%	1,5%
Mixed	65	13	2,5%	-0,9%	23	2,0%	-0,9%	23	2,3%	-0,7%	26	2,3%	-0,7%	33	2,0%	-0,8%
Total	265	17,0%	2,0%	1,0%	26,8%	1,9%	0,9%	29,1%	1,9%	0,8%	34,0%	1,9%	1,0%	40,8%	1,9%	0,8%

Results

- A time limit of 5 seconds and datasets with continuous features (31).
  - There was an improvement in trees with a depth of 1-3
- Even for the decision tree with 1 level depth, accuracy is improved.
  - DTE optimizes accuracy and CART reduces impurity.

Depth	Imp.	Train	Test
1	4	2,5%	3,1%
2	19	2,5%	3,5%
3	14	1,4%	0,6%
4	8	0,7%	0,1%
5	3	0,8%	-1,1%
Mean	48	1,8%	1,8%

Conclusion

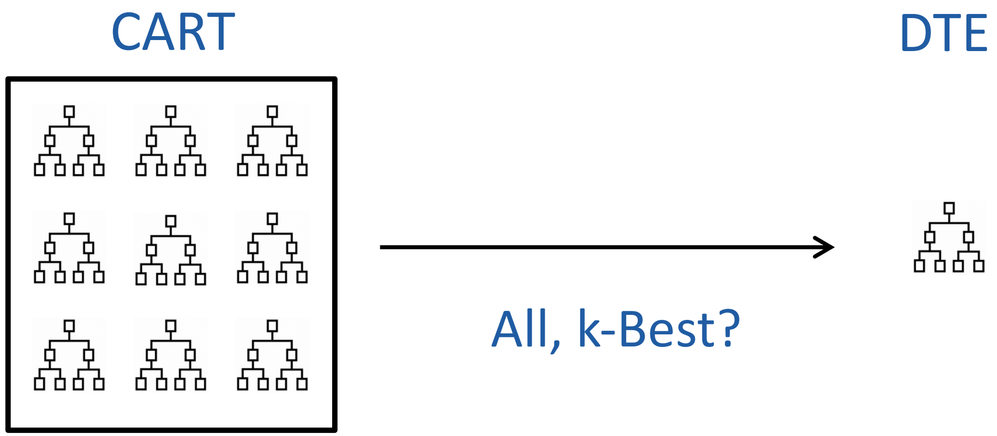
- DTE use CART and optimization to improve the performance of decision trees.

- DTE can provide better results than CART with a small increase in model training time.

- DTE can adapt other objective function and does not use discretization.
  - Obj. Func.: Constraining recall or precision, balancing sensitivity and specificity or other problem-specific metrics.

Next steps

- Test and evaluate the tree generation flow
  - Multi-start



Sample:

- Features
- Elements

- DTE can be used as a decision tree generator for other techniques that use some type of staking, such as Random Forest.

Pontifical Catholic University of Rio de Janeiro
Department of Industrial Engineering
Postgraduate Program in Production Engineering

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Introductory presentation
+
Decision trees: using optimization to enhance performance

Flávio Araújo Lim-Apo

Espoo
May 27, 2024