



# ConBRepro

XIII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO



IA nas Engenharias

29 nov. a 01 de dezembro 2023

## Técnicas Visuais de Decisão Multicritério Aplicado a Estruturas de Aprendizado de Máquina Baseados em Árvore de Decisão

**Marcos Hendrio Pompeu da Silva**

PPGEPS - Pontifícia Universidade Católica do Paraná

**Gilberto Reynoso-Meza**

PPGEPS - Pontifícia Universidade Católica do Paraná

**Resumo:** Os modelos de aprendizado de máquina têm ganhado muito espaço nas grandes corporações, atuando como ferramenta de análise para selecionar as melhores decisões, gerando vantagem competitiva por meio da antecipação das ações. Por esse motivo desenvolver modelos eficientes e confiáveis é essencial para o sucesso de qualquer projeto. Nesse artigo iremos propor uma alternativa para avaliar modelos de classificação baseados em árvores de decisão através de decisão multicritério e análise de conceitos de design. Para isso utilizaremos três objetivos conflitantes, precisão, recall e quantidade de variáveis, que serão analisadas de forma conjunta. Utilizaremos algoritmos de aprendizado de máquina com características estruturais diferentes com o objetivo de ressaltar as variações de comportamento conforme introduzimos variáveis novas. Acreditamos que seja possível encontrar o melhor modelo com as melhores variáveis olhando não só para métricas já conhecidas de forma individualizada e sim para uma metodologia que consiga mesclar esses objetivos do ponto de vista multicritério, que irão contribuir para o sucesso de um bom modelo.

**Palavras-chave:** Tomador de decisão, aprendizado de máquina, seleção de características, análise multicritério.

## Visual Multicriteria Decision Techniques Applied to Machine Learning Structures Based on Decision Trees

**Abstract:** Machine learning models gained a lot of space in large corporations, and remained as an analysis tool to select the best decisions, gaining competitive advantage through anticipation. For this reason, developing efficient and followed models are essential for the success of any project. In this article, we will propose an alternative to fairly evaluate tree classification models through a multicriteria decision and analysis design concepts comparison. For this, we will use three conflicting objectives, precision, recall, and number of variables that will be joined. using machine learning algorithms with different characteristics to highlight behavior variations as input new variables. We believe that it is possible to find the best model with the best variables by looking not only at metrics already known individually but at a multi-criteria methodology that manages to merge the main objectives that will contribute to the success of a good model.

**Keywords:** Decision maker, machine learning, feature selection, multi-criteria analysis.

## 1. Introdução

Historicamente, ferramentas estatísticas avançadas têm sido empregadas no meio acadêmico por décadas, culminando em avanços significativos para a sociedade. O aprendizado de máquina, em particular, tem sido um pilar na geração de conhecimento validado por pesquisas rigorosas e experimentações. Estas ferramentas são essenciais para a automatização de processos, aprimoramento de produtos e eficiência operacional nas empresas (BARTON; TRIBUNAL, 2012, p. 81). Contudo, somente na última década, com o advento do crescimento exponencial de dados, avanços em ferramentas de processamento e armazenamento, e progressos na ciência de dados, é que as grandes corporações começaram a adotar e implementar tais práticas em escala ampla (LEE; SHIN, 2020, p. 158). Neste novo cenário, modelos preditivos permeiam diversas áreas de negócios, tornando a informação uma commodity vital para a sobrevivência e prosperidade das empresas (MCAFEE; BRYNJOLFSSON, 2012, p. 6). No entanto, um desafio persiste: a falta de transparência e interpretabilidade dos modelos. Esta lacuna muitas vezes impede a validação lógica por especialistas, retardando a adoção destas tecnologias (Vellido, 2018, p. 18071).

Diante deste contexto, o presente estudo propõe uma abordagem que busca equilibrar eficiência e complexidade em diferentes estruturas de aprendizado. Nosso foco é minimizar taxas de erro, ao mesmo tempo em que reduzimos a quantidade de variáveis pouco informativas. Através da otimização de características, decisão multicritério e análise de conceitos de design, pretendemos elucidar o comportamento das variáveis durante o treinamento de diversos modelos, simplificando a complexidade e melhorando a interpretabilidade. Este artigo está estruturado da seguinte forma: A metodologia (seção 2) detalha os procedimentos experimentais e ferramentas empregadas. O Referencial Teórico (seção 3) discute estudos anteriores que fundamentam a relevância do tema. A seção 4 apresenta os Resultados, discutindo os casos de uso e comparações. Concluímos com reflexões finais na seção 5.

## 2. Metodologia

Todo o experimento foi desenvolvido utilizando a linguagem python através métodos nativos ou pela importação de bibliotecas (pacotes) desenvolvidas pela comunidade científica. Para as etapas de treinamento dos modelos de aprendizado de máquina foram utilizadas diversas bibliotecas dentre as quais podemos destacar: pandas (McKinney, 2011, p. 1), numpy (WALT; COLBERT; VAROQUAUX; 2011, p. 23), matplotlib (HUNTER, 2007, p. 90), sklearn (PEDREGOSA; VAROQUAUX; GRAMFORT; MICHEL; THIRION; GRISEL; BLONDEL; PRETTENHOFER; WEISS; DUBOURG; VANDERPLAS; PASSOS; COURNAPEAU; BRUCHER; DUCHESNAY, 2011, p. 2826). E para as etapas de decisão multicritério o pymoo (BLANK; DEB, 2020, p. 89497).

### 2.1 Cordas paralelas

Cordas paralelas é uma técnica muito utilizada que traz uma abordagem bidimensional com sucessivos eixos verticais paralelos e um único eixo horizontal por ponto. Os eixos verticais retratam os objetivos conflitantes que tem como finalidade facilitar a busca pelo ponto ideal. Já o eixo horizontal mostra como as variáveis de decisão estão atuando sobre os objetivos (REYNOSO-MEZA, 2014, p. 11).

Figura 1.a – Diagrama de dispersão

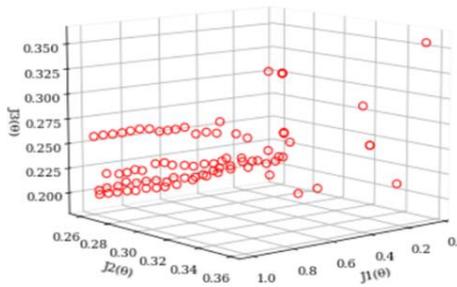
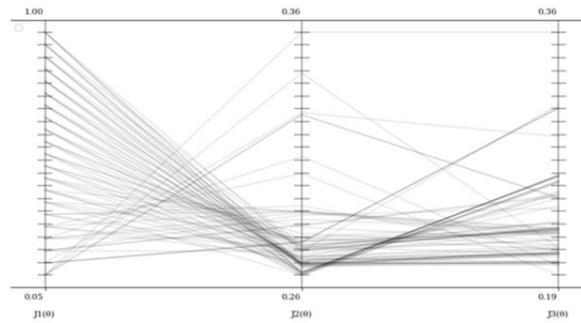


Figura 1.b – Cordas paralelas



Na figura 1.a temos o que chamamos de diagrama de dispersão com três objetivos e, portanto, três dimensões para avaliar os critérios e selecionar o ponto que mais faça sentido para a resolução do problema. Já na figura 1.b temos os mesmos três objetivos, porém evidenciado através de um plano bidimensional que permite visualizar o comportamento dos objetivos entre si. Além desta técnica existem outras como o diagrama de nível que também utiliza uma abordagem bidimensional só que neste caso teremos um gráfico para cada objetivo e ele será representado no eixo horizontal (REYNOSO-MEZA, 2014, p. 2).

## 2.2 Base de dados

Antes das etapas de treinamento e decisão multicritério é necessário selecionar uma base de dados que faça sentido para o estudo em questão. Para isso, selecionamos uma fonte de dados pública fornecida pelo Kaggle, uma das maiores comunidades de ciência de dados do mundo. Nessa comunidade, são disponibilizadas milhares de bases de dados amplamente utilizadas pela comunidade científica. A fonte de dados abordada no estudo apresenta uma variável alvo do tipo categórica com duas classes, ou seja, estamos trabalhando em problema de classificação binária (1 para o caso de positivo para diabetes e 0 para negativo). Além dessas características, a base será balanceada e os dados faltantes removidos (Kaggle,datasets, 2023).

## 2.3 Hiperparâmetros

Conforme abordado anteriormente iniciaremos pela etapa de treinamento dos modelos de aprendizado de máquina através da utilização da base de dados de diabetes. Para garantir a reprodutibilidade dos resultados iremos utilizar o mesmo random state (parâmetro disponível nos modelos baseados em árvores que visam garantir replicação dos resultados). Além deste, outros hiperparâmetros deverão ser pré-definidos dentre eles os de profundidade da árvore (max\_depth), quantidade mínima de instâncias para divisão dos nós (min\_samples\_split), quantidade mínima de instâncias para cada nó folha (min\_samples\_leaf) e para o critério de divisão dos nós, utilizaremos o índice de Gini (TANGIRALA, 2022, p. 613).

$$GINI(L) = 1 - \sum_{i=1}^j p_i^2 \quad (1)$$

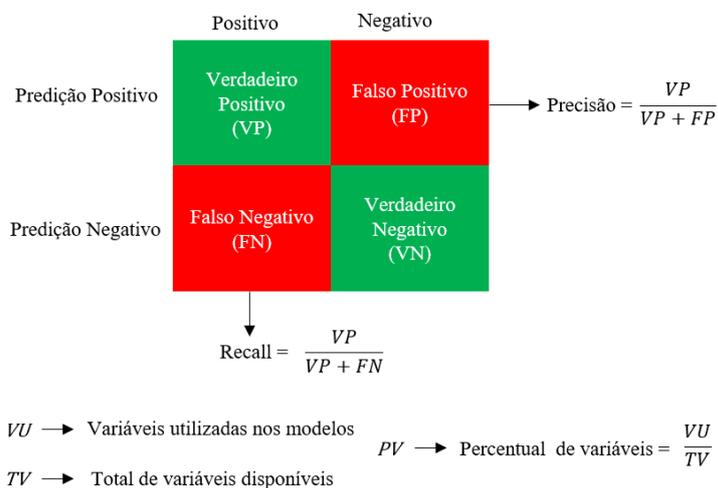
O índice de Gini é soma das probabilidades de as variáveis pertencerem a uma determinada classe ponderada pelo número de registros ou instâncias. O resultado será subtraído por menos um.

## 2.4 Métricas

Para avaliar a performance dos algoritmos de aprendizado de máquina foram escolhidos

três métricas. A precisão avalia as afirmações do que realmente são verdadeiras e que foram apontadas pelo modelo, sobre o total dos valores verdadeiros. Já o recall avalia a capacidade do modelo de identificar os valores verdadeiros. Quanto mais valores positivos o modelo identifica, maior será o valor do recall. Além dessas duas métricas já conhecidas no universo do aprendizado de máquina, proporemos uma terceira forma de avaliação, que será a quantidade de variáveis usadas pelas estruturas de aprendizado em relação à quantidade total de variáveis disponíveis.

**Figura 2 – Matriz de confusão**



Como o foco é trabalhar na minimização dos objetivos (métricas) será necessário realizar alguns ajustes na formulação em relação a precisão e o recall. Após essa etapa será possível evidenciar os três objetivos com seus valores normalizados entre zero e um através de um plano tridimensional.

$EP$  → Erro de Precisão = 1 - Precisão

$ER$  → Erro de Recall = 1 - Recall

Portanto, o vector de objetivos a considerar é:

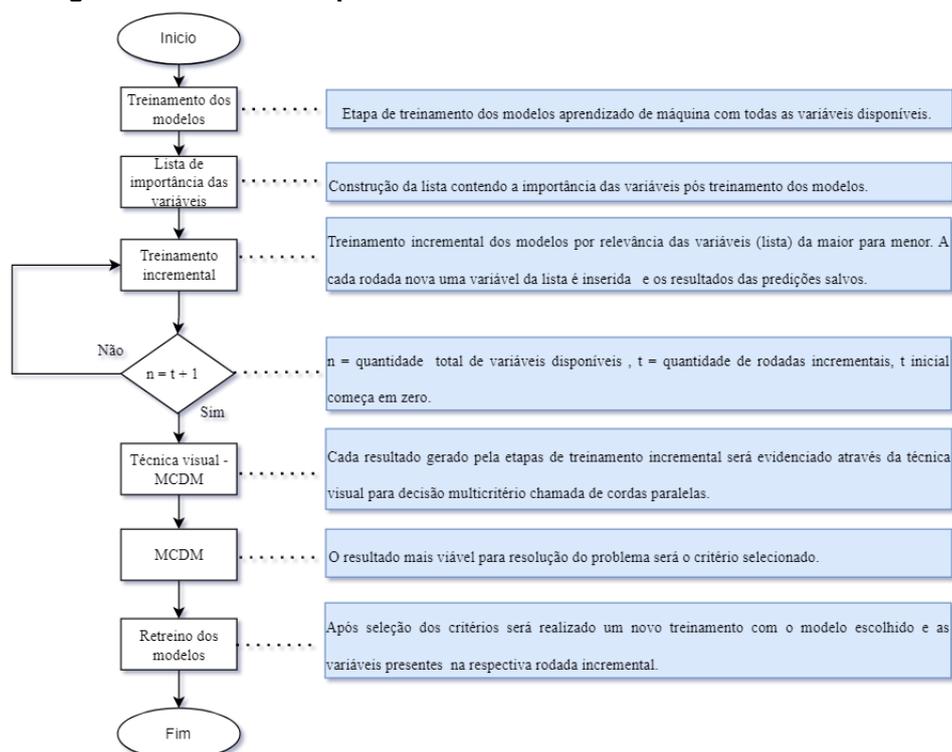
$$J(\theta) = [PV(\theta), EP(\theta), ER(\theta)] \quad (2)$$

Onde  $\theta$  é um vector de zeros e uns [0,1] indicando se a características vai ser utilizada da estrutura de aprendizagem ou não.

## 2.5 Fluxo de construção dos modelos de aprendizado de máquina

Inicialmente será realizada uma rodada de treinamento e teste com todas as variáveis disponíveis para todos os modelos. Após a conclusão iremos extrair a importância das variáveis em relação a variável alvo através da utilização de uma função disponível pela biblioteca sklearn chamada de `feature_importances_` utilizando o índice de Gini como critério. Cada modelo terá uma lista variáveis ordenadas por relevância da maior para menor. Como os modelos possuem estruturas de construção diferentes as sequencias serão distintas. Já com as listas definidas iremos realizar novas rodadas de treinamento, mas agora de forma incremental, ou seja, a primeira execução terá apenas a variável mais relevante, já segunda rodada terá as duas mais relevantes, a terceira as três mais relevantes e assim por diante até chegar na variável de menor relevância. Todos os treinamentos serão armazenados em uma estrutura tabular que posteriormente será usada nas técnicas de decisão multicritério.

**Figura 3: Fluxo das etapas de treinamentos e decisão multicritério**



### 3. Referencial Teórico

O crescimento do uso de ferramentas que utilizam dados como fontes de matéria prima para tomada de decisão tem crescido exponencialmente nas empresas que estão na vanguarda de seus seguimentos. Resolver problemas baseados em análises onde os dados são o centro de tudo nunca foi tão frequente no meio corporativo (KRATSCH; MANDERSCHIED; RÖGLINGER; SEYFRIED, 2020, p. 262). Por esse motivo, o interesse dos pesquisadores sobre o tema tem crescido nos últimos anos e com eles o surgimento de novas técnicas que mesclam áreas de diferentes seguimentos de atuação (COCKBURN; HENDERSON; STERN, 2019, p. 122).

#### 3.1 Aprendizado Supervisionado

Aprendizado de máquina refere-se a algoritmos de computador que se aprimoram automaticamente por meio da experiencia com dados (MITCHELL, 1997, p. 240). Aprendizado supervisionado usa como entrada  $I$  instancias (observações) com  $X$  variáveis (variáveis explicativas) para treinar uma estrutura de aprendizado usando informações confiáveis através da variável alvo  $T$  para cada uma das instancias.

Para o caso particular da classificação binária o modelo irá treinar usando um conjunto de dados  $[I \times X|T]$  a fim de ajustar seus parâmetros  $\beta$  na fase de otimização, usando algum critério de avaliação ou função de custo.

#### 3.2 Árvores de Decisão

O algoritmo de arvore de decisão é um dos mais conhecidos e mais usados modelos de aprendizado de máquina para problemas de regressão e classificação (QUINLAN, 1987, p. 2). Um dos grandes fatores que tornam esse modelo popular é a forma como ele atua nas etapas de treinamento, através de uma representação clara dos critérios escolhidos gerando uma visualização amigável em todo processo de aprendizagem. Na prática é possível ver como as principais variáveis se comportam em relação a variável alvo, a figura

4.a evidencia um dado tabular com três variáveis sendo uma delas a variável alvo, na figura 4.b temos o espaço amostral das instancias em um plano bidimensional e na figura 4.c é evidenciado o fluxo de decisão do algoritmo para cada variável. Nas etapas da figura 4 é possível perceber como a estrutura básica dos modelos baseados em árvores são transparentes e interpretáveis, facilitando o entendimento do problema.

Figura 4.a

ID	Variável 1	Variável 2	Variável Alvo
1	3	10	Sim
2	1	30	Sim
3	2	40	Sim
4	40	25	Sim
5	6	25	Sim
6	13	31	Não
7	4	35	Sim
8	27	15	Sim
9	14	40	Não
10	25	35	Não
11	30	41	Não
12	45	36	Não
13	36	42	Não
14	36	30	Não
15	6	39	Sim
16	45	21	Não

Figura 4.b

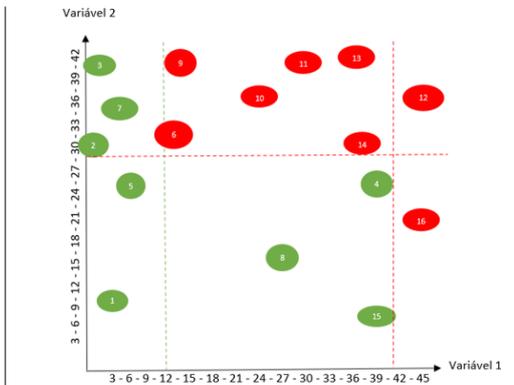
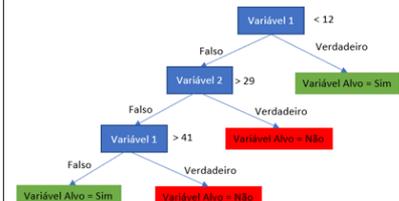


Figura 4.c

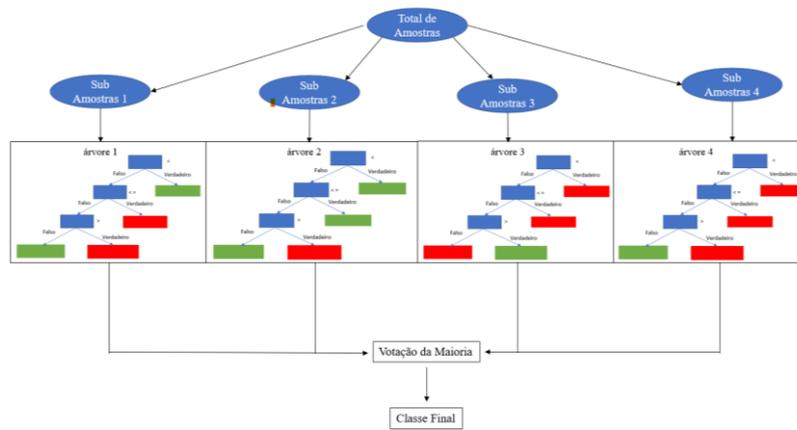


Normalmente uma árvore de decisão  $DT(\beta)$  é definido pelos seus parâmetros  $\beta = (n, v)$  ou seja, um conjunto de nós  $n$  e *thresholds*  $v$ . Dado um nó  $n$  na árvore de decisão  $DT(\beta)$  que corresponde a uma partição de instancias  $I_n \subseteq I$ , e a variável  $x \in X$ , definimos a partição do nó  $n$  em relação a variável  $x$  da seguinte forma:  $I_{n,L} = \{i \in I_n | i_x \leq v_x\}$  e  $I_{n,R} = \{i \in I_n | i_x > v_x\}$ . Onde,  $v_x$  é o *threshold* da variável  $x$  para o nó  $n$  e  $i_x$  é o valor para variável  $x$  para instancias  $i$ . A partição do nó  $n$  em relação a variável  $x$  divide o conjunto de instancias  $i_n$  em dois subconjuntos.  $I_{n,L}$  e  $I_{n,R}$  que correspondem aos ramos esquerdo e direito do nó da arvore de decisão DT. Cada subconjunto  $I_{n,L}$  e  $I_{n,R}$  pode ser usado para construir sub árvores descendentes do nó  $n$  na árvore de decisão através do particionamento recursivo. Baseados na definição anterior, podemos destacar outras estruturas aprendizagem que também utilizam árvores de decisão, porém com características mais complexas conforme descritivo abaixo.

### 3.3 Random forest

O modelo de floresta aleatória utiliza uma combinação de árvores, onde cada árvore treinada terá seu resultado avaliado com base nas combinações das amostras e variáveis escolhidas (BREIMAN, 2001, p. 7). Na figura 5 temos a exemplificação do treinamento de quatro árvores e cada uma delas com um conjunto de dados diferentes (sub amostras) que são selecionadas durante o processo de treinamento dos modelos, que escolherá quais variáveis serão mais relevantes. O fato dessa estrutura possuir diversas árvores torna essa técnica mais robusta, pois teremos mais opções de seleção de características tornado menos suscetível a super ajustes aos dados de treino (*overfitting*) permitindo uma maior capacidade de generalização. No entanto, é necessário compreender que devido a esse comportamento a técnica será mais custosa computacionalmente e a compreensão um pouco mais difícil.

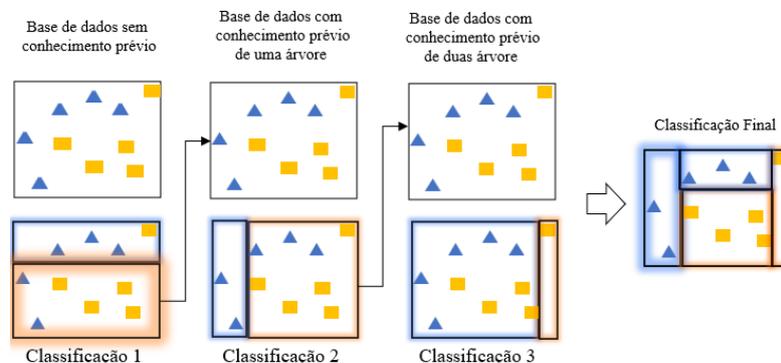
Figura 5 – Representação do algoritmo de florestas aleatórias na criação de arvores.



### 3.4 XGBoost

Nosso terceiro algoritmo é uma combinação entre um modelo baseado em árvore de decisão atuando em conjunto com técnica chamada de aumento de gradiente ou gradiente boosting (CHEN; GUESTRIN, 2016, p. 786). O aumento de gradiente tem a finalidade de combinar modelos com menor capacidade de predição (árvores de decisão) atuando sequencialmente e conforme novos modelos são treinados os próximos irão se adaptar com base no histórico dos resultados se diferenciando do modelo de floresta aleatória que atua na construção de arvores de forma paralela. Na Figura 6 é evidenciado tal comportamento onde cada etapa de treinamento (classificação) utilizará o conhecimento prévio da árvore anterior que permitirá uma maior robustez na classificação. Atualmente o XGBoost é dos métodos ensemble mais utilizados pela comunidade devido a sua boa capacidade predição para dados tabulares graças a sua robustez estrutural para minimizar as perdas.

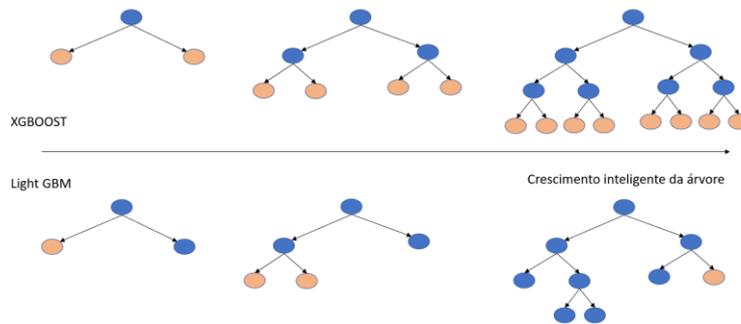
Figura 6 – Etapas de treinamento sequencial baseados na estrutura de aprendizagem XGBoost.



### 3.5 LGBM

Assim como o XGBoost o LGBM é um modelo que utiliza aumento de gradiente em sua estrutura, mas apresenta características diferentes (KE; MENG; FINLEY; WANG; CHEN; MA; YE; LIU, 2017, p.4). O modelo da Microsoft apresenta maior eficiência nas etapas de construção das árvores como é mostrado na figura 7 reduzindo significativamente o custo computacional, no entanto em alguns casos pode apresentar uma menor capacidade de predição ao compararmos com o XGBoost. A seguir temos o comparativo entre o LGBM que constrói árvore de forma mais eficiente e o XGBoost.

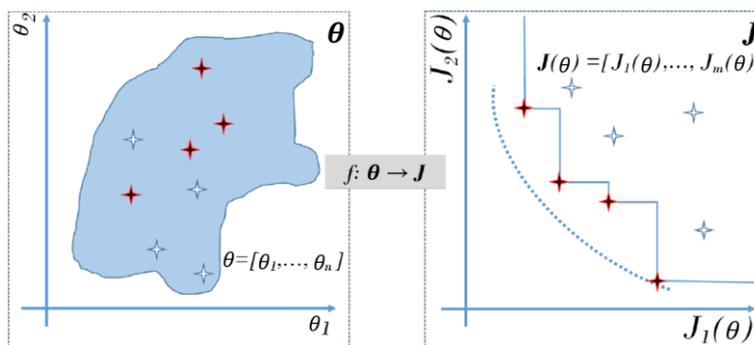
Figura 7 – Demonstração do modelo LGBM em relação ao XGBoost.



### 3.6 Comparação de conceitos de design

A frente de Pareto é uma forma de evidenciar os pontos não dominados, ou seja, as soluções em que pelo menos um dos objetivos apresenta melhor resultado quando comparado com as demais soluções. Esses conjuntos de soluções não dominadas possibilitam selecionar um ponto que mais faça sentido de acordo com os critérios estabelecidos (REYNOSO-MEZA, BLASCO, HERRERO, 2013, p. 2). Abaixo temos uma frente de Pareto com o intuito de minimizar os objetivos.

**Figura 8 – Visualização das soluções e escolha dos pontos não dominados para construção da frente de Pareto (Reynoso-Meza; Carrillo-Ahumada; Morales, 2017, p. 367)**



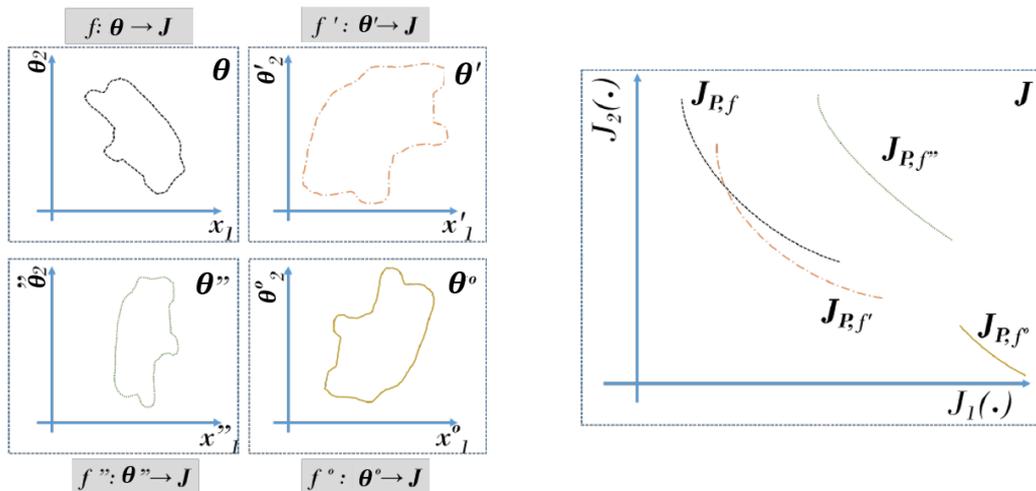
Um vetor multi-objetivo pode ser definido como:

$$\mathbf{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \dots, J_m(\boldsymbol{\theta})] \quad (3)$$

Onde  $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_n]$  é definido como o vetor de decisão com  $\dim(\boldsymbol{\theta}) = n$ ;  $\mathbf{J}(\boldsymbol{\theta})$  como o vetor objetivo. Percebeu-se que não existe uma solução única nos MOPs, pois geralmente não existe uma única solução que é melhor em todos os objetivos. Assim, define-se um conjunto de soluções, o conjunto de Pareto  $\boldsymbol{\theta}_P$ . Cada solução no conjunto de Pareto define um vetor objetivo na frente de Pareto  $\mathbf{J}_P$ . É importante notar que dependemos apenas da frente de Pareto e as aproximações dos conjuntos,  $\mathbf{J}_P^*$ ,  $\boldsymbol{\theta}_P^*$ . Todas as soluções na frente de Pareto são um conjunto de soluções de Pareto ótimas e não dominadas.

Em (MATTSON; MESSAC, 2005, p. 86), algum refinamento é incorporado à noção de frente de Pareto para diferentes conceitos de design. Uma frente de Pareto é definida para um conceito de design específico (ou simplesmente um conceito); essa é uma ideia sobre como resolver um determinado MOP. Em um contexto de aprendizado de máquina, tais conceitos de design estão relacionados a diferentes estruturas de aprendizado. Na Figura 6 esta ideia é ilustrada. Quatro conceitos de projeto  $\boldsymbol{\theta}$ ,  $\boldsymbol{\theta}'$ ,  $\boldsymbol{\theta}''$ ,  $\boldsymbol{\theta}^\circ$  são definidos para um MOP bi-objetivo específico, e uma frente de Pareto foi aproximada para cada um deles. Com a comparação visual entre eles, é possível perceber as três relações clássicas entre as frentes de Pareto.

Figura 9 – Exemplificação dos espaços objetivos das quatro estruturas de aprendizagem (Reynoso-Meza; Carrillo-Ahumada; Morales, 2017, p. 368)



- Dominância Parcial: se compararmos  $J_{P,\theta}^*$  e  $J_{P,\theta'}^*$ , o primeiro domina numa determinada região, mas o posterior na outra.
- Dominância Total:  $J_{P,\theta''}^*$  é totalmente dominado por  $J_{P,\theta}^*$  e  $J_{P,\theta'}^*$ .
- Sem relação de dominância:  $J_{P,\theta^o}^*$  está em uma região de troca onde não existe outra aproximação.

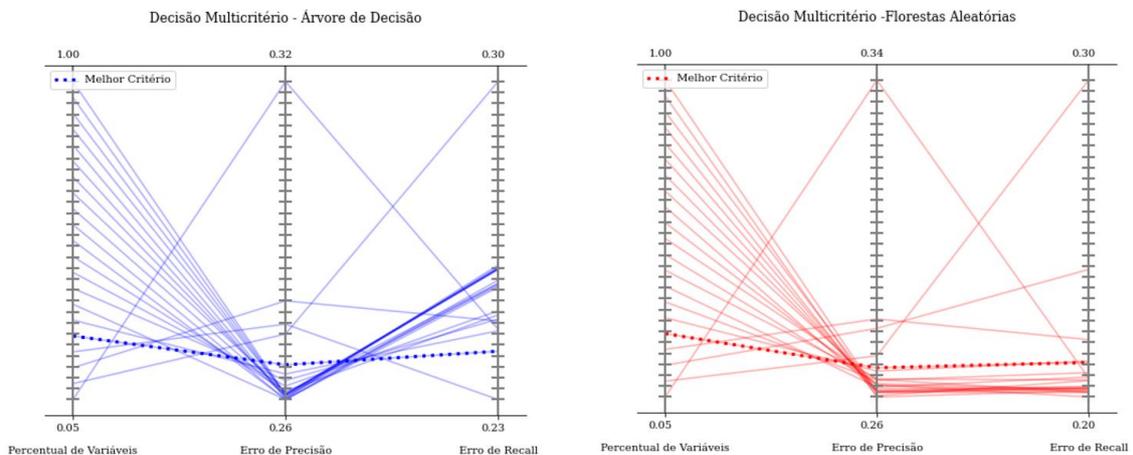
#### 4. Resultados

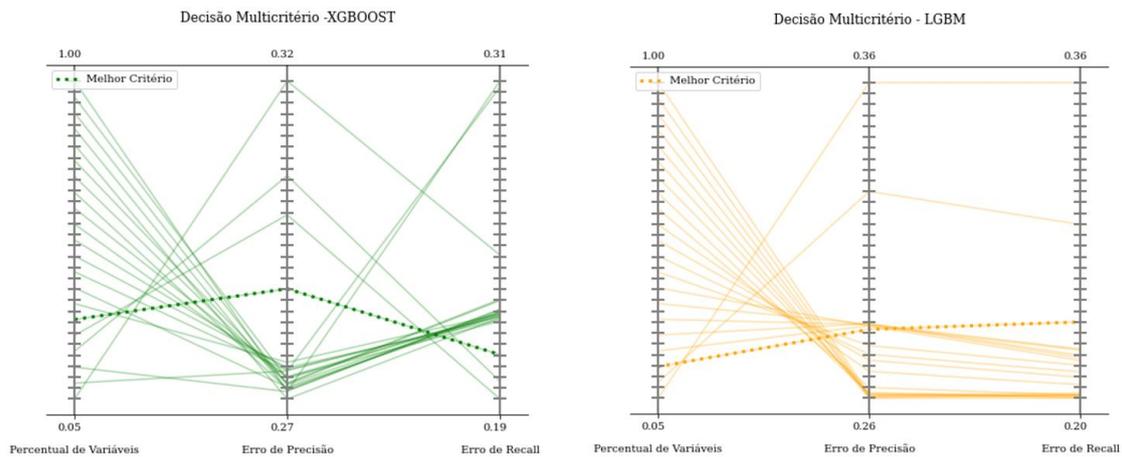
Para esta etapa usaremos a biblioteca pymoo como principal ferramenta, já que possui diversos métodos pré-definidos que visam facilitar a construção dos códigos relacionados a decisão multicritério.

##### 4.1 Capacidade de predição dos modelos de forma individualizada

Para facilitar a compreensão será mostrado os resultados das estruturas de aprendizado de forma individualizada. E para essa simulação utilizaremos a base de dados anteriormente descrita.

Figura 10 – Cordas paralelas evidenciando os resultados das estruturas de aprendizagem, base de dados de diabetes



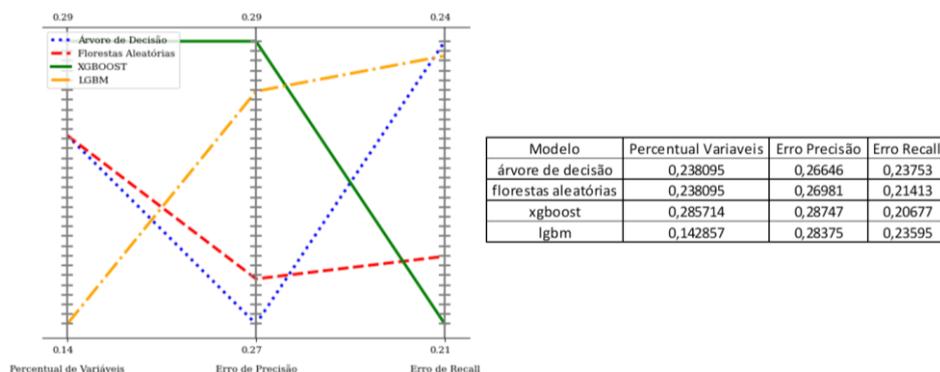


Nas imagens acima o número de cordas é proporcional ao número de variáveis presentes na base de dados. Onde cada simulação retorna o percentual de variáveis utilizadas pelos modelos, erro de precisão e o erro de recall. Fica claro que cada modelo possui valores de predição diferentes permitindo escolher aquele que mais faça sentido de acordo com os critérios de interesse. Para essas simulações cada objetivo recebeu o mesmo peso, ou seja, será priorizado o equilíbrio entre os três eixos verticais selecionando o ponto que consiga minimizar os valores de forma mais eficiente. Para o cálculo de seleção de um critério de muitos pontos utilizamos a função de escalarização (ASF) amplamente utilizada para MCDM (WIERZBICKI, 1980, p. 476) que é evidenciada na corda tracejada e com maior intensidade de cor dentre as demais. Para essa simulação foram utilizadas quatro frentes de Pareto, uma para cada modelo. Em cordas paralelas as trocas de linhas representam troca de *trade-off* e quando não há o cruzamento podemos afirmar que estão correlacionadas. No modelo de árvore de decisão percebemos a troca de *trade-off* entre os objetivos de erro de precisão e erro de recall. O que não podemos dizer do modelo de florestas aleatórias, que apresentam uma correlação direta entre os mesmos objetivos. Outra característica importante se dá pelo fato de o XGBoost possuir grande semelhança com o modelo de árvore de decisão e que por consequência se diferencia que LGBM que apresenta uma maior proximidade com o modelo de florestas aleatórias para os mesmos objetivos anteriormente citados. Agora, ao analisar o percentual de variáveis todas as estruturas de aprendizagem tendem a seguir o mesmo comportamento, ou seja, todas possuem troca de *trade-off*.

$$\text{Minimize}_x \text{ ASF}(x, z, w) = \max_{i=1}^M \left( \frac{f_i(X) - z_i}{w_i} \right) \quad (4)$$

## 4.2 Comparação das estruturas de aprendizado em relação aos critérios de escolha

Figura 11 – Critério de escolha para cada estrutura de aprendizado



Ao compararmos os pontos de escolha de cada modelo podemos chegar nas seguintes conclusões. O modelo de árvore de decisão apresenta o menor erro de precisão, porém possui o pior erro de recall dentre todos e utiliza uma quantidade relevante de variáveis aumentando a complexidade do modelo. Já o algoritmo de floresta aleatória possui valores intermediários quando comparado com as demais estruturas de aprendizado. O XGBoost possui o maior erro de precisão e utiliza mais variáveis, no entanto tem o menor erro de recall. O LGBM fornece o menor percentual de variáveis, todavia não apresenta um resultado satisfatório nas demais métricas, que vai ao encontro de suas propriedades estruturais. Dito isso, o modelo selecionado deverá ser aquele que mais faça sentido para o problema priorizando os objetivos mais importantes. Caso quantidade de variáveis fosse sua prioridade o modelo selecionado seria o LGBM, pois possui uma complexidade menor. Agora, se a busca fosse por uma estrutura de aprendizado mais equilibrada o escolheríamos o modelo de floresta aleatória.

## **5. Conclusões**

Essa nova era, onde modelos de aprendizado de máquina estão presentes em diversas áreas de negócio, traz consigo a necessidade vital de informação para as empresas. Por isso mesclar ferramentas que melhorem a eficiência e que contribuam para redução da complexidade são essenciais. E a combinação de técnicas multicritério e aprendizado de máquina surgem como uma alternativa. Nessa pesquisa foram aplicados testes em quatro modelos baseados em árvore de decisão, onde cada estrutura teve um comportamento distinto em relação aos dados de entrada e conseqüentemente critérios de escolha diferentes. Além de fornecer uma alternativa visual para facilitar o entendimento de como os modelos de aprendizado de máquina estão se comportando durante as etapas de treino e teste. Os modelos foram separados com o intuito de facilitar o entendimento. No entanto, para tornar essa metodologia mais viável devemos selecionar o ponto de maior eficiência dentre todas as estruturas de aprendizado disponíveis, analisando as opções de forma simultânea e retornando o ponto ideal em relação aos três objetivos, assunto que será abordado futuramente. Outro ponto a ser destacado, é que utilizamos apenas uma única base de dados que torna análise menos robusta e para aumentar confiabilidade novos experimentos serão inseridos, ou seja, teremos diversos treinamentos com diferentes bases de dados que iremos acrescentar com intuito de mitigar essa limitação.

## **Referencias**

### **Livro**

Tom Mitchell. Machine Learning, 1997.

### **Capítulo de Livro**

John D. Hunter. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 2007. p. 90-95.

Iain M. Cockburn, Rebecca Henderson, Scott Stern. The Impact of Artificial Intelligence on Innovation An Exploratory Analysis. Nber. 2019. p 115-146

### **Artigo de periódico**

In Lee, Yong Jae Shin. Machine Learning for enterprises: applications, algorithm selection, and challenges. Elsevier. p. 157-170, 2020.

Alfredo Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. Springer. p. 18069–18083, 2018.

Stéfan van der Walt, S. Chris Colbert, Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. IEEE. p. 22-30, 2011.

Gilberto Reynoso-Meza, Xavier Blasco, Javier Sanchis, Juan M. Herrero. Comparison of Design Concepts in Multi-Criteria Decision-Making Using Level Diagrams. Polytechnic University of Valencia. p. 124-141, 2013.

Julian Blank, Kalyanmoy Deb. Pymoo: Multi-Objective Optimization in Python. IEEE.  
Wolfgang Kratsch, Jonas Manderscheid, Maximilian Röglinger, Johannes Seyfried (2020). Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction. Springer. p. 261-276, 2020.

J.R.Quinlan. Simplifying decision trees. International Journal of Man Machine Studies. Elsevier. p. 221-234, 1987.

Leo Breiman. Random Forests. Springer. p. 5-32, 2001.  
Tianqi Chen, Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. ACM digital library. p. 785-794, 2016.

Christopher A. Mattson, Achille Messac. Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. Springer. p. 85-115, 2005.

Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization, in Multiple criteria decision making theory and application. Springer. p. 468-486, 1980.

### **Trabalho em evento**

Dominic Barton, David Tribunal. Making Advanced Analytics Work For You. Harvard Business Review. 2012. p. 79-84.

Andrew McAfee, Erik Brynjolfsson (2012). Big Data: The Management Revolution. Harvard Business Review. 2012. p. 1-9.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011.p. 2825-2830.

Suryakanthi Tangirala. Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm. International Journal of Advanced Computer Science and Applications. 2022. p. 612-619.

Guolin Ke<sup>1</sup>, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 2017. p.1-9.

Gilberto Reynoso-Meza, Jesús Carrillo-Ahumada, Leonardi Martín Torralba Morales. Pareto Front Analysis of Control Structures in Multimodel Systems: The Work Scenario Concept Introduction. SBAI. 2017. p. 366-371.

### **Internet**

Wes McKinney. Pandas: a Foundational Python Library for Data Analysis and Statistics. 2011. p.1-9.

<https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset>>Acesso em: 20 fev. 2023.