

Árvore de Classificação Ótima Interpretável para o problema de classificação de níveis de estresse

Jader Duque Figueredo

Diretoria de Tecnologia da Computação (DTI), Universidade Federal de Itajubá

jader.figueredo@unifei.edu.br

Karen Cristina Soares Cavalheiro

PPGCI - Faculdade de Filosofia e Ciências (UNESP-Marília, SP)

karen.cavalheiro@unesp.br

Pedro Henrique Del Bianco Hokama

Instituto de Matemática e Computação (IMC), Universidade Federal de Itajubá

hokama@unifei.edu.br

RESUMO

Soluções baseadas em inteligência artificial têm sido amplamente utilizadas, mas nem sempre são apropriadas quando a interpretabilidade é essencial. Modelos recentes são capazes de gerar árvores de classificação ainda mais ajustadas que modelos heurísticos, usando otimização. Nós treinamos os modelos Árvore de Classificação e Regressão (CART) e Árvore de Classificação Ótima usando Otimização Inteira Mista (OCT-MIO) usando validação cruzada sobre um conjunto de dados de pessoas classificadas em 5 níveis de estresse. Os experimentos mostram que o modelo OCT-MIO alcança mais acertos que o CART para uma mesma altura de árvore, tornando-o uma melhor alternativa para problemas onde é necessário obter uma melhor acurácia, preservando a interpretabilidade.

PALAVRAS CHAVE. Árvore de Classificação Ótima Interpretável, modelo de aprendizado de máquina, classificação de níveis de estresse,

otimização inteira mista, aprendizado de máquina, saúde,

ABSTRACT

AI-based solutions have been widely adopted, but they are not always appropriate when interpretability is essential. Recent models can generate more accurate classification trees than heuristic-based ones by using optimization. We trained the Classification and Regression Tree (CART) and the Optimal Classification Tree using Mixed-Integer Optimization (OCT-MIO) models using cross-validation on dataset of people classified into 5 stress levels. The experiments show that the OCT-MIO model achieves more correct predictions than CART for the same tree depth, making it a better alternative for problems where higher accuracy is required while preserving interpretability.

KEYWORDS. interpretable optimal classification tree. machine learning model. stress levels classification.

mixed integer optimization. machine learning. health.

1. Introdução

Modelos de Inteligência Artificial têm sido amplamente utilizados em contextos pessoais e profissionais. Embora modelos mais acurados, como Redes Neurais, sejam eficazes, suas baixas interpretabilidades motivam o uso de alternativas como árvores de decisão [Linardatos et al., 2020]. A Árvore de Classificação Ótima usando Otimização Inteira Mista (*Optimal Classification Tree using Mixed Integer Optimization – OCT-MIO*) avalia todas as divisões possível em uma única etapa. Em contraste, a Árvore de Classificação e Regressão (*Classification and Regression Tree – CART*) realiza divisões recursivas *top-down*, em que cada decisão depende das anteriores, o que pode levar a soluções subótimas [Bertsimas e Dunn, 2017].

Além do CART e OCT-MIO, estudos recentes exploram as *Strong Optimal Classification Trees*, que utilizam decomposição de *Benders* para melhorar desempenho, acurácia e interpretabilidade [Aghaei et al., 2024]. van der Linden et al. [2023] melhoraram a escalabilidade das árvores ótimas propondo um *Framework* para Árvore de Decisão Ótima usando Programação Dinâmica, abordando a decomposição recursiva do problema, princípio de otimalidade, preservação da ordem, anti-monotonicidade e otimização generalizada.

Já não tão recentes, as Árvores de Classificação Ótimas usando Programação Linear Binária (BinOCT) simplificam o processo de otimização reduzindo a dependência aos dados de treinamento, reduzindo o tempo de treinamento através de limiares mais eficientes e com menor número de restrições [Verwer e Zhang, 2019]. As BinOCT, baseadas em Programação Linear Binária, simplificam a otimização ao reduzir a dependência dos dados de treinamento, com limiares mais eficientes e menos restrições, acelerando o treinamento [Verwer e Zhang, 2019].

Neste trabalho, implementamos os modelos CART e OCT-MIO e aplicamos eles ao conjunto de dados *Human Stress Detection in and Through Sleep*. Avaliamos Acurácia, Acurácia Balanceada, Macro Pontuação F1 e Tempo de Treinamento via Validação Cruzada (*K-Fold*). Também geramos e analisamos as representações visuais das árvores finais, relacionando-as com as métricas obtidas. Por fim, discutimos as capacidades de cada modelo e os contextos em que são mais apropriados.

Este trabalho visa contribuir com a sociedade e comunidade científica com o estudo da aplicação e análise de resultados comparativos entre dois modelos de classificação em um contexto de saúde, onde a interpretabilidade dos modelos é desejável. Não é nosso objetivo analisar qualitativamente os níveis de estresse durante e através do sono.

A Seção 2 apresenta os fundamentos teóricos, incluindo conceitos de árvores de classificação, estrutura dos dados e interpretabilidade. A Seção 3 detalha a modelagem dos modelos CART e OCT-MIO, com definição de variáveis, função objetivo e restrições (no caso do OCT-MIO), além das métricas e do método de validação utilizados e ainda ajustes do modelo OCT-MIO. A Seção 4 descreve os experimentos, incluindo configurações, pré-processamento e parametrizações. Na Seção 5, mostramos os resultados obtidos; a Seção 6 traz a análise e interpretação desses resultados. Por fim, as conclusões e aplicabilidade dos modelos são apresentadas na Seção 8.

2. Fundamentação Teórica

Nesta seção apresentamos os conceitos fundamentais de árvores de classificação e interpretabilidade.

2.1. Árvore de Classificação

Árvore de Classificação (*Classification Tree*) é um tipo específico de Árvore de Decisão também formada por uma estrutura hierárquica de nós e arestas, cujo o objetivo é atribuir corretamente uma classe a um novo elemento a ser classificado. Um modelo de aprendizado de máquina baseado em árvores de classificação é um modelo que constrói sua estrutura hierárquica baseada em padrões encontrados em um conjunto de dados através de algoritmos.

Neste trabalho, os dados $(X, \mathbf{y})$ do problema estão estruturados em uma tabela, onde cada linha representa um elemento $x_i \in X$ para $i = 1, 2, \dots, n$, sendo n o número de elementos de X e cada coluna representa um atributo $p = 1, \dots, P$ de um elemento x_i , onde P é o número de atributos do conjunto. Cada interseção entre linhas e colunas contém um valor $x_{ip} \in [0, 1]$, para todo $i = 1, 2, \dots, n$ e $p = 1, 2, \dots, P$, há ainda um vetor coluna $\mathbf{y}$ onde $y_i \in \{1, 2, \dots, K\}$ é a classe do elemento x_i e K é o número de classes do problema, para todo $i = 1, 2, \dots, n$. Cada classe do problema é referenciada por $k \in \{1, 2, \dots, K\}$.

A estrutura de uma Árvore de Classificação é composta por um conjunto de nós $v \in V$ contendo um subconjunto V_b de nós galho (internos) e um subconjunto V_f de nós folha (terminais), tal que $V_b \cup V_f = V$ e $V_b \cap V_f = \emptyset$. v_1 é o nó raiz da árvore, v_2 é o nó filho esquerdo de v_1 , v_3 é o filho direito de v_1 e assim continua recursivamente. À partir de um nó v qualquer na árvore definimos v_L seu filho esquerdo, v_R seu filho direito e v' seu nó pai.

Todo nó galho $v \in V_b$ possui duas variáveis: $\mathbf{a}_v \in [0, 1]^P$ e b_v , onde $\mathbf{a}_v$ é um vetor binário de tamanho P em que cada posição refere-se a um atributo do conjunto de dados, portanto $\sum_{i=1}^P a_i = 1$ define que apenas um único atributo é utilizado para dividir os elementos x_i em v . Já $b \in [0, 1]$ é um valor real que define o limiar para x_{ip} no nó v .

Todo nó folha $v \in V_f$ possui uma variável $k \in \{1, 2, \dots, K\}$, onde K é o número de classes do problema, que determina a classe que rotula v e também rotula todos os elementos x_i que chegam a v .

Cada divisão na árvore ocorre através de uma condição $s \in S$, definida na forma $\mathbf{a}_v^T \mathbf{x}_i < b_v$ em cada nó $v \in V_b$. Assim, se o valor de um único atributo ($\mathbf{a}_v^T \mathbf{x}_i$) for menor que o limiar b em um determinado nó $v \in V_b$, então o elemento x_i seguirá para o nó filho esquerdo de v , caso contrário, x_i seguirá para o nó filho direito de v .

2.2. Interpretabilidade de Árvores de Classificação

Interpretabilidade e explicabilidade são termos frequentemente usados como sinônimos para descrever a capacidade de compreensão humana sobre modelos de aprendizado de máquina. No entanto, enquanto a interpretabilidade é mais associada à relação entre entradas e saídas do modelo, a explicabilidade diz respeito à compreensão de seu funcionamento interno [Linardatos et al., 2020].

A representação gráfica de uma Árvore de Classificação reflete fielmente sua estrutura lógica: os nós galhos indicam divisões aplicadas aos dados e os nós folhas representam as classificações finais. Isso permite inferir, visualmente, os critérios adotados pelo modelo em cada decisão. Por agrupar instâncias de forma não dispersa e utilizar regras simples e contrafactuais, as árvores são modelos intuitivos e de fácil interpretação [Molnar, 2020].

A interpretabilidade está diretamente ligada à altura da árvore. Como o número de nós cresce exponencialmente com a altura, árvores maiores tendem a ser mais complexas e menos interpretáveis. Ainda assim, árvores pequenas permanecem adequadas para contextos sensíveis que demandam transparência [Molnar, 2020].

3. Metodologia

Nesta seção apresentamos a modelagem dos modelos CART e OCT-MIO, nossos ajustes dos modelos, além de métricas, método de validação e conjunto de dados.

3.1. Árvore de Classificação e Regressão (CART)

A Árvore de Classificação e Regressão (*Classification and Regression Tree* - CART) é um modelo baseado em Árvore de Decisão que utiliza uma heurística para construir, de forma recursiva, uma árvore que classifica os dados de entrada $(X, \mathbf{y})$ [Breiman et al., 2017]. A cada nó, o algoritmo escolhe a divisão que mais reduz a impureza, gerando uma separação progressiva entre as classes.

A proporção de elementos da classe k em um nó v é dada por $p(k|v) = n_k/n_v$, onde n_k é o número de elementos da classe k e n_v o total de elementos no nó. A impureza do nó v é calculada por $i(v) = -\sum_{k=1}^K p(k|v) \log p(k|v)$, $v \in V$.

Essa impureza é maior quanto mais equilibrada for a distribuição de classes no nó e tende a zero conforme a distribuição se torna mais desigual.

Definimos ainda a **proporção absoluta** como $p(v) = n_v/n$ e a **proporção relativa** como $p_v = n_v/n_{v'}$, com v' sendo o nó pai de v . A **impureza absoluta** de um nó é $I(v) = p(v)i(v)$, e a **impureza relativa** é $p_v i(v)$.

A heurística do CART avalia a redução da impureza em uma divisão s no nó v com base nas impurezas dos filhos v_L e v_R , sendo $\Delta i(s, v) = i(v) - p_{v_L} i(v_L) - p_{v_R} i(v_R)$, $s \in S$, $v \in V$ a variação da impureza relativa e $\Delta I(s, v) = I(v) - I(v_L) - I(v_R)$, $s \in S$, $v \in V$ a variação da impureza absoluta. A melhor divisão (s^*) é aquela que maximiza $\Delta i(s, v)$, dada por $\Delta i(s^*, v) = \max_{s \in S} \Delta i(s, v)$, $s \in S$, $v \in V$.

O algoritmo continua dividindo cada nó enquanto a impureza for maior que zero. Para evitar sobreajuste, o modelo pode interromper a divisão se a **variação mínima da impureza absoluta** $\Delta I(s^*, v)$ for menor que um limiar fornecido como parâmetro. Neste trabalho, também utilizamos um segundo parâmetro: a **altura máxima** da árvore.

Cada nó terminal $v \in V_f$ é rotulado com a classe majoritária (k^*), isto é, $p(k^*|v) = \max_k p(k|v)$.

A impureza total da árvore é dada por $I(T) = \sum_{v \in V_f} I(v) = \sum_{v \in V_f} i(v)p(v)$, $v \in V_f$.

3.2. Árvore de Classificação Ótima usando Otimização Inteira Mista (OCT-MIO)

A Árvore de Classificação Ótima utilizando Otimização Inteira Mista (*Optimal Classification Tree using Mixed Integer Optimization* - OCT-MIO) é um modelo de aprendizado de máquina proposto por Bertsimas e Dunn [2017], baseado em árvores de decisão, que recorre à formulação MIO para ajustar um modelo de classificação a um conjunto de dados tabular $(X, \mathbf{y})$ e gerar uma Árvore de Classificação sobre estes dados.

Esse modelo possui alguns parâmetros principais: o número mínimo de elementos por folha $n_{min} \in \mathbb{Z}_+$; o parâmetro $\alpha \in \mathbb{R}_+$, que regula o equilíbrio entre a acurácia e a complexidade da árvore na função objetivo; o tempo máximo de treinamento $t_{max} \in \mathbb{Z}^+$, em segundos; e a altura máxima permitida da árvore, denotada por $D \in \mathbb{Z}^+$. A partir de D , define-se o número máximo de nós da árvore como $N_{max} = 2^D - 1$, e a árvore real resultante possui $N \leq N_{max}$ nós, indexados por $v \in 1, \dots, N_{max}$. O número de nós internos é representado por $N_{V_b} \in \mathbb{Z}_+$ e o número de folhas por $N_{V_f} \in \mathbb{Z}^+$.

Na estrutura da árvore, nós galho são nós $v \in V_b = \{1, \dots, \lfloor N/2 \rfloor\}$, enquanto nós folha são nós $v \in V_f = \{\lfloor N/2 \rfloor + 1, \dots, N\}$. Para um dado nó v , define-se $p(v)$ como o seu nó pai, $A(v)$ como o conjunto de seus nós ancestrais (isto é, todos os nós do caminho da raiz até v), $A_L(v)$ como o subconjunto de ancestrais nos quais a condição $\mathbf{a}^\top \mathbf{x} < b$ é satisfeita, e $A_R(v)$ como o subconjunto complementar em que essa condição é falsa.

Na formulação proposta, algumas variáveis binárias são essenciais para a modelagem. A variável $d_v \in \{0, 1\}^N$, indica com $d_v = 1$ se o nó v realiza uma divisão (ou seja, se é um nó interno) e $d_v = 0$, caso contrário; $z_{iv} \in \{0, 1\}$ representa com $z_{iv} = 1$ se o elemento $\mathbf{x}_i$ chega ao nó v e $x_{iv} = 0$, caso contrário; e $l_v \in \{0, 1\}$ indica com $l_v = 1$ se o nó folha v contém algum elemento e $l_v = 0$, caso contrário. Além disso, são introduzidas duas constantes: M , um número suficientemente grande para permitir o uso de restrições relaxadas (neste trabalho, define-se $M = 2$), e ϵ , um valor pequeno positivo utilizado para garantir o uso adequado de desigualdades estritas. A constante ϵ é calculada como $\epsilon = \min x_{(i+1)p} - x_{ip} \mid x_{(i+1)p} \neq x_{ip}$, $i = \{1, \dots, n-1\}$, $p = \{1, \dots, P\}$.

Para contabilizar os acertos e erros de classificação, define-se $n_{kv} \in \mathbb{Z}_+$ como o número de elementos da classe k que chegam ao nó folha v , e $n_v \in \mathbb{Z}_+$ como o total de elementos que chegam a v . A classe majoritária que rotula o nó folha v é dada por $c_v = \arg \max_{k=1, \dots, K} n_{kv}$. Com isso, o número de classificações incorretas no nó v é dado por $R(v) = n_v - \max_{k=1, \dots, K} \{n_{kv}\}$.

A função objetivo apresentada na Expressão 1 visa minimizar a soma dos erros de classificação, normalizada pela linha de base da acurácia ($\frac{1}{R}$), combinada com a penalização da complexidade da árvore proporcional a α . As restrições subsequentes modelam a consistência estrutural e semântica da árvore, garantindo que cada elemento chegue exatamente a uma folha, que as divisões sejam corretamente aplicadas e que a atribuição de classes e os cálculos de perda estejam corretos.

$$\begin{aligned}
 \min \quad & \frac{1}{\hat{R}} \sum_{v \in V_f} R(v) + \alpha \sum_{v \in V_b} d_v, \\
 \text{s. a.} \quad & \sum_{p=1}^P a_{pv} = d_v, \quad a_{pv} \in \{0, 1\}, \quad p = 1, \dots, P, \quad \forall v \in V_b, \\
 & 0 \leq b_v \leq d_v, \quad \forall v \in V_b, \\
 & d_v \leq d_{p(v)}, \quad \forall v \in V_b \setminus \{1\}, \\
 & z_{iv} \leq l_v, \quad v \in V_b, \\
 & \sum_{i=1}^n z_{iv} \geq n_{\min} l_v, \quad v \in V_f, \\
 & \sum_{v \in V_f} z_{iv} = 1, \quad i = 1, \dots, n, \\
 & \mathbf{a}_m^T \mathbf{x}_i < b_m + M_1(1 - z_{iv}), \quad \forall v \in V_f, \quad i = 1, \dots, n, \quad \forall m \in A_L(v), \\
 & \mathbf{a}_m^T \mathbf{x}_i \geq b_m - M_2(1 - z_{iv}), \quad \forall v \in V_f, \quad i = 1, \dots, n, \quad \forall m \in A_R(v), \\
 & d_v \leq l_{g(v_L)}, \quad \forall v \in V_b, \quad g(v_L) \in V_f, \\
 & d_v \leq l_{g(v_R)}, \quad \forall v \in V_b, \quad g(v_R) \in V_f, \\
 & n_{kv} = \sum_{i \in X^k} z_{iv}, \quad k = 1, \dots, K, \quad v \in V_f, \quad X^k \subseteq X = \{i | y_i = k\}, \\
 & n_v = \sum_{i=1}^n z_{iv}, \quad \forall v \in V_f, \\
 & c_v = \arg \max_{k=1, \dots, K} \{n_{kv}\}, \\
 & \sum_{k=1}^K c_{kv} = l_v, \quad \forall v \in V_f, \\
 & R(v) \geq n_v - n_{kv} - M(1 - c_{kv}), \quad k = 1, \dots, K, \quad \forall v \in V_f, \\
 & R(v) \leq n_v - n_{kv} + M c_{kv}, \quad k = 1, \dots, K, \quad \forall v \in V_f, \\
 & R(v) \geq 0, \quad \forall v \in V_f.
 \end{aligned} \tag{1}$$

3.3. Aprimoramentos

Neste trabalho nós implementamos ambos os modelos. O modelo OCT-MIO foi modelado conforme em [Bertsimas e Dunn, 2017], contudo, nós observamos que, apesar de todas as

restrições impostas à função objetivo, o modelo eventualmente gerava árvores com nós $v \in V_b$ com apenas uma ramificação, o que é indesejável em uma Árvore de Classificação. Observamos que factualmente não havia, na modelagem original, uma restrição que impusesse a um nó $v \in V_b$ necessariamente tenha 2 ramificações. Observamos também que, quando um nó tinha apenas uma ramificação, o valor do limiar b estava entre 0 e 1, porém b estava além da amplitude de valores do atributo p dos elementos $x \in X_v$, fazendo com que a divisão ocorresse ora apenas para direita, ora apenas para esquerda.

Para resolver este problema, nós modelamos e implementamos as restrições adicionais

$$d_v \leq l_{g(v_L)}, \quad \forall v \in V_b, \quad g(v_L) \in V_f,$$

$$d_v \leq l_{g(v_R)}, \quad \forall v \in V_b, \quad g(v_R) \in V_f,$$

onde $g(v_L)$ e $g(v_R)$ são os nós folha mais à direita das subárvores esquerda e direita de v , respectivamente. Assim, garantimos que todo nó galho na árvore gerada tenha necessariamente 2 ramificações.

3.4. Métricas

As métricas de classificação são utilizadas para avaliar o desempenho dos modelos. A Acurácia corresponde à razão entre o número de classificações corretas e o total de previsões realizadas. Em problemas multiclases, esse valor é obtido somando-se as classificações corretas de todas as classes.

A Acurácia Balanceada é a média das revocações (*Recalls*) por classe, enquanto a Macro Pontuação F1 corresponde à média harmônica entre Precisão e Revocação, calculada individualmente por classe. Também foi considerado o Tempo de Treinamento, medido em segundos.

Neste trabalho, as expressões "métricas de classificação" referem-se especificamente à Acurácia, Acurácia Balanceada e Macro Pontuação F1.

3.5. Validação Cruzada

A validação cruzada (*Cross-Validation*) permite estimar o desempenho preditivo dos modelos. Utilizamos o método *K-fold Cross-Validation* com $k = 5$, dividindo o conjunto de dados em cinco partes.

Em cada *fold*, um subconjunto é reservado para teste e os demais para treinamento. Ao final dos cinco ciclos, calculamos a média (AVG) e o desvio padrão (DP) das métricas de classificação e do tempo de treinamento.

3.6. Conjunto de Dados

Human Stress Detection in and thought Sleep ([Laavanya, 2022]) é um conjunto de dados, utilizados por [Rachakonda et al., 2021] e [Rachakonda et al., 2018], composto de 630 registros de pessoas, classificadas em níveis de stress de 1 a 5. Cada registro possui 8 atributos obtidos durante e através do sono destas pessoas. O gráfico da Figura 1 mostra a dispersão dos elementos do conjunto da perspectiva de 2 atributos.

4. Experimentos

Os experimentos visam comparar o comportamento e o desempenho entre os dois modelos de árvore de classificação (OCT-MIO e CART).

Todos os experimentos foram realizados em um computador com Processador: 12th Gen Intel® Core™ i5-1235U × 12; Memória: RAM 32 GB, DDR4, 3200 Mhz; Processador Gráfico: Mesa Intel® Graphics (ADL GT2); Armazenamento: 1 TB HD + 0,5 TB SSD; Sistema Operacional: Ubuntu 22.04.4 LTS - 64-bit; Compilador C++: gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0; Resolvedor (*Solver*): IBM ILOG CPLEX Optimization Studio_22.1.0.

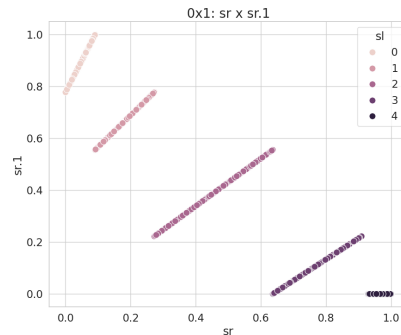


Figura 1: Conjunto Stress - Gráfico de Dispersão

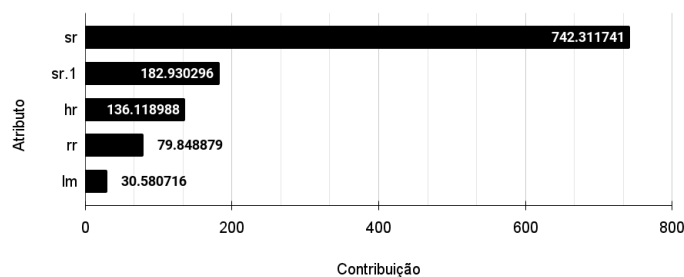


Figura 2: Contribuição de cada atributo do conjunto “Stress” segundo o algoritmo *SelectKBests*

Antes do treinamento, os dados foram pré-processados. Na etapa de limpeza, nós verificamos que os dados não continham elementos faltantes, outliers ou duplicidades, pois o conjunto já havia sido tratado nestes aspectos. Portanto, todos os dados foram transformados em dados numéricos, inclusive as classes. Em seguida os atributos foram normalizados para valores entre 0 e 1 usando a normalização Min-Max, onde $Min = 0$ e $Max = 1$, (MMN0), conforme esperado na modelagem do OCT-MIO. Por fim, foi feita a redução de dimensionalidade utilizando o algoritmo *SelectKBests* com $K = 5$, onde os 5 atributos preservados representam uma parcela significativa dos atributos que mais contribuem para a representação do problema, conforme a Figura 2.

Para ambos os modelos foram definidos 3 valores consecutivos para o parâmetro altura máxima, sendo eles altura mínima, altura mínima + 1 e altura mínima + 2, a fim de obtermos as 3 menores árvores de alturas diferente para o mesmo problema. Ambos os modelos não definem explicitamente um parâmetro “altura mínima”, contudo os valores utilizados para altura máxima são propositalmente definidos à partir de uma altura mínima suficiente para que número de nós folhas seja igual ou maior que o número de classes do conjunto de dados. Como o conjunto de dados Stress possui 5 classes, a altura mínima para que a Árvore de Classificação possa classificar 5 classes é 3, conforme a expressão $\lceil \log_2 K \rceil < K \leq 2^{\lceil \log_2 K \rceil}$, onde $\lceil \log_2 K \rceil$ é a altura mínima necessária para que o número de folhas ($2^{\lceil \log_2 K \rceil}$) seja maior ou igual ao número de classes do problema (K). Sendo assim, os experimentos foram realizados para 3 alturas, sendo o parâmetro altura máxima igual a 3, 4 e 5.

Quanto aos parâmetros específicos, o modelo OCT-MIO foi treinado com as seguintes definições de parâmetros: $\alpha = 0.0$, $n_{min} = 1$ e $t_{max} = 36000$ segundos (10 horas) e 300 segundos (5 minutos). Já o modelo CART foi treinado com o parâmetro mínima variação da impureza = 0.0.

Para treinamento dos modelos foi utilizado a técnica *K-fold Cross-Validation*, com $K = 5$,

para os 3 valores de altura máxima, onde, em cada *fold*, nós medimos as métricas: Acurácia, Acurácia Balanceada, Macro Pontuação F1 e Tempo de Treinamento. Ao final de 5 folds, nós calculamos tanto a média aritmética (AVG) quanto o desvio padrão (DP) das métricas obtidas.

Na tabela de resultados, valores marcados entre parênteses “()” são valores onde o modelo OCT-MIO obteve todas as soluções ótimas, já os valores do modelo OCT-MIO sem parênteses são os valores onde ao menos 1 das soluções não provou otimalidade. Os valores marcados em **negrito** significam que estes foram os melhores resultados de média na comparação OCT-MIO vs CART (para uma mesma altura máxima). Por fim, o asterisco (*) significa que ao menos um *fold* não encontrou solução factível dentro do tempo limite e, portanto, a média marcada pelo asterisco é composta por menos de 5 soluções.

5. Resultados

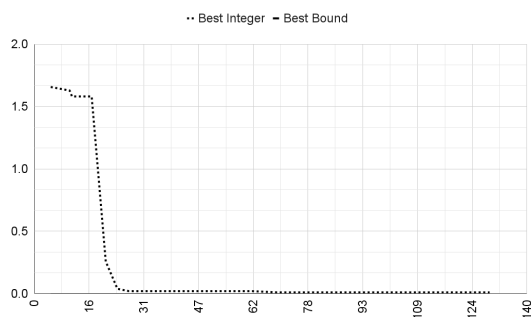
Observe que modelo OCT-MIO, quando treinado com tempo máximo de execução = 300 segundos (5 minutos), nem sempre encontra soluções ótimas ou até mesmo factíveis, já quando treinado com tempo máximo de execução = 36000 segundos (10 horas) o modelo provou otimalidade de todas as soluções.

Tabela 1: Métricas: OCT-MIO 5min e OCT-MIO 10h vs CART

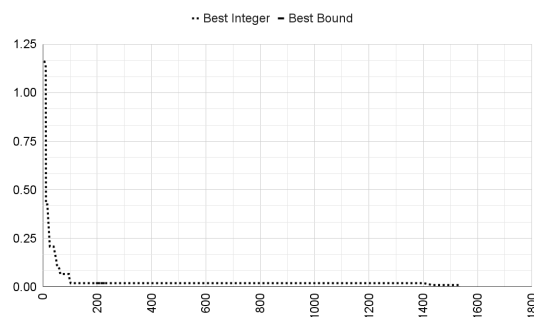
Acurácia						
Altura	OCT-MIO 5 min	DP	OCT-MIO 10 h	DP	CART	DP
3	0.994	±0.009	(0.994)	±0.009	0.987	±0.012
4	0.992	±0.010	(0.992)	±0.010	0.987	±0.012
5	0.992*	±0.009	(0.992)	±0.008	0.987	±0.012
Acurácia Balanceada						
Altura	OCT-MIO 5 min	DP	OCT-MIO 10 h	DP	CART	DP
3	0.994	±0.008	(0.994)	±0.008	0.988	±0.011
4	0.992	±0.010	(0.992)	±0.010	0.988	±0.011
5	0.992*	±0.009	(0.992)	±0.008	0.988	±0.011
Macro Pontuação F1						
Altura	OCT-MIO 5 min	DP	OCT-MIO 10 h	DP	CART	DP
3	0.994	±0.008	(0.994)	±0.008	0.987	±0.012
4	0.992	±0.010	(0.992)	±0.010	0.987	±0.012
5	0.992*	±0.009	(0.992)	±0.008	0.987	±0.012
Tempo de Treinamento (segundos)						
Altura	OCT-MIO 5 min	DP	OCT-MIO 10 h	DP	CART	DP
3	230.917	±102.114	(680.148)	±736.185	0.183	±0.008
4	230.342	±101.440	(1515.665)	±2805.574	0.191	±0.004
5	294.733*	±12.607	(4631.813)	±4014.700	0.196	±0.004

Os gráficos 3, 4 e 5 mostram a convergência do modelo OCT-MIO até a solução ótima para as árvores de altura máxima igual a 3, 4 e 5, respectivamente. Os *folds* que compõem os gráficos foram selecionados arbitrariamente para que os pares fossem compostos por gráficos do menor (a) e maior (b) tempo de execução.

Por fim, apresentamos as árvores geradas pelos 2 modelos treinados com 100% do conjunto de dados, onde a Figura 6 é referente à mesma árvore gerada pelo CART para “altura mínima”,

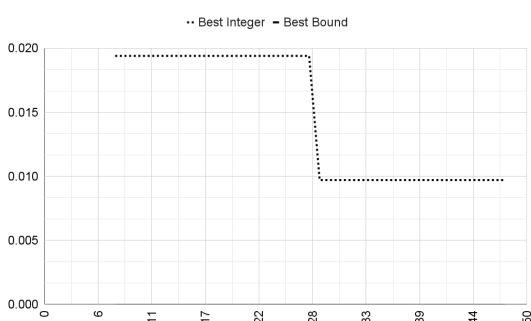


(a) Fold 1

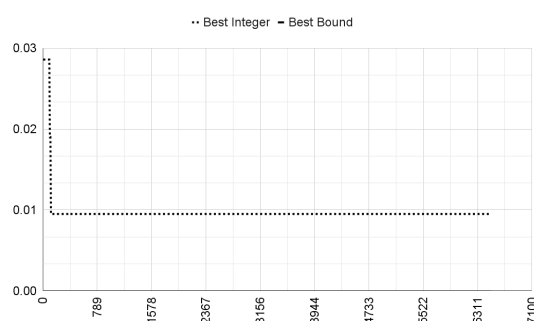


(b) Fold 3

Figura 3: Convergência do modelo OCT-MIO para a árvore de altura máxima = 3



(a) Fold 1



(b) Fold 5

Figura 4: Convergência do modelo OCT-MIO para a árvore de altura máxima = 4

“altura mínima”+1 e “altura mínima”+2. Já as Figuras 7, 8 e 9 são referentes às árvores geradas pelo OCT-MIO para “altura mínima”, “altura mínima”+1 e “altura mínima”+2, respectivamente.

6. Discussões

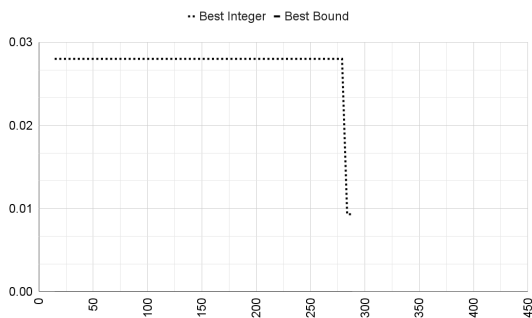
Os resultados da Tabela 1 mostram que o OCT-MIO obteve melhores resultados em todas as métricas de classificação, tanto quando treinado com tempo máximo de execução de 10 horas, quanto com 5 minutos, em comparação ao modelo CART. Isso mostra a robustez do modelo OCT-MIO.

Os valores das métricas de classificação mostram uma ligeira, porém constante, superioridade de desempenho do OCT-MIO, reforçando a hipótese de que o modelo é capaz de encontrar soluções ótimas (na etapa de treinamento), gerando árvores mais ajustadas aos dados e com melhor desempenho (na etapa de testes) em comparação ao modelo CART.

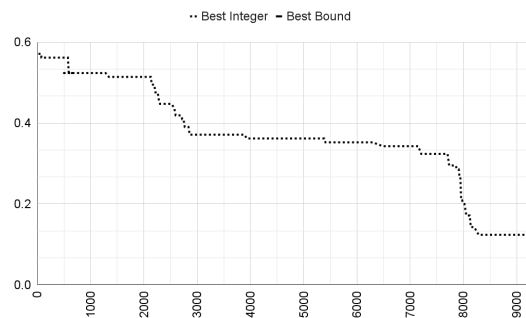
Os valores de desvio padrão (DP) das métricas de classificação ainda sustentam que os valores de média são, em geral, descolados dos respectivos valores do modelo CART.

Os resultados de tempo de treinamento do modelo OCT-MIO (Tabela 1), apesar de maiores que os do CART, mostraram que o modelo é capaz de obter resultados de métricas de classificação superiores, mesmo não provando a otimalidade de todas as soluções.

Observe ainda nas Figuras 3, 4 e 5 que, mesmo em um tempo inferior ou, às vezes, muito menor que o tempo total de treinamento, o OCT-MIO é capaz de obter soluções cujo valor objetivo é igual ou próximo ao das soluções ótimas, tornando considerável sua utilização como um modelo



(a) Fold 4



(b) Fold 5

Figura 5: Convergência do modelo OCT-MIO para a árvore de altura máxima = 5

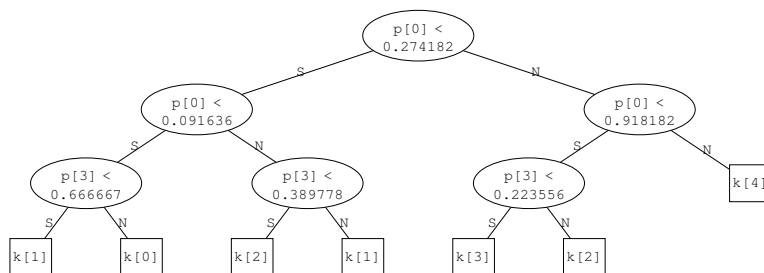


Figura 6: Árvore de classificação para o conjunto *Stress* com altura máxima = altura mínima, altura mínima+1 e altura mínima+2 pelo modelo CART treinada com 100% do conjunto de dados

heurístico.

Por fim, as árvores geradas pelos dois modelos com 100% do conjunto de dados refletem visualmente as diferenças de abordagem dos dois modelos.

Na Figura 6 a árvore gerada pelo modelo CART utiliza menos nós até a altura máxima. Isso acontece porque as árvores geradas pelo CART, quando treinado com 100% do conjunto de dados, conseguiram separar corretamente os elementos de todas as classes. Isso é bom, mas não garante que o modelo também conseguirá separar corretamente novos elementos das mesmas classes, como é possível observar analisando as métricas de classificação do modelo durante as etapas de testes da Tabela 1, onde o modelo CART obteve métricas de classificação inferiores ao modelo OCT-MIO.

O modelo OCT-MIO também conseguiu classificar corretamente todos os elementos de cada classe, quando treinado com 100% do conjunto de dados, porém, como mostram as Figuras 7, 8 e 9 o modelo gera soluções diferentes e com mais divisões que o CART para cada altura máxima, resultado da obtenção de soluções ótimas globais e não estagnação em soluções ótimas locais. O modelo OCT-MIO, quando treinado com o parâmetro $\alpha = 0.0$, não faz quaisquer penalizações sobre número de divisões até a altura máxima.

Soluções ótimas globais podem sugerir um ajuste excessivo (*overfitting*) aos dados, contudo, as métricas de classificação da Tabela 1 mostram que o modelo OCT-MIO é capaz de gerar árvores mais generalizadas que o modelo CART, contradizendo a suposição de ajuste excessivo.

Outra observação importante é que as árvores da Figura 8 e 9, geradas pelo modelo OCT-

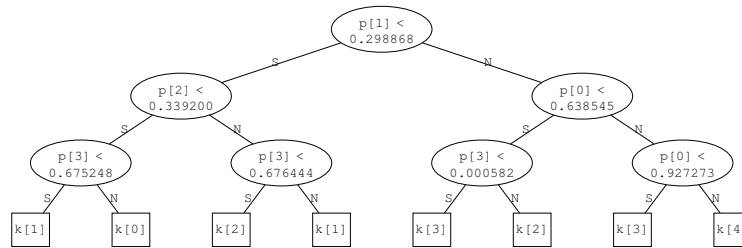


Figura 7: Árvore de classificação para o conjunto *Stress* com altura máxima = altura mínima pelo modelo OCT-MIO treinada com 100% do conjunto de dados

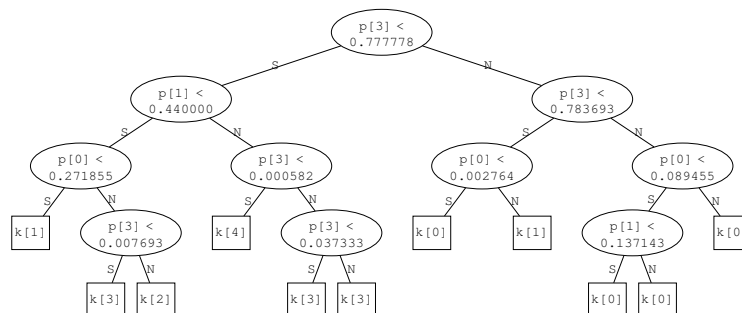


Figura 8: Árvore de classificação para o conjunto *Stress* com altura máxima = altura mínima+1 pelo modelo OCT-MIO treinada com 100% do conjunto de dados

MIO, fizeram divisões desnecessárias, isto é, separam elementos de uma única classe (Isso possivelmente foi causado também pela definição do parâmetro $\alpha = 0.0$). Contudo, uma etapa de poda ou mesmo a definição de $\alpha > 0.0$ poderiam ser aplicados para eliminar essas divisões, tornando a árvore mais simples e mais interpretável.

7. Conclusões

Modelos de aprendizado de máquina baseados em árvores são uma alternativa viável quando a interpretabilidade é essencial.

Os resultados indicam que o modelo OCT-MIO gera árvores com desempenho superior ao CART para uma mesma altura máxima, explorando mais divisões, mas mantendo a interpretabilidade. Essa característica o torna adequado para cenários críticos, como saúde ou finanças, onde decisões exigem explicabilidade e erros podem ser custosos.

Já o CART mostrou-se mais apropriado para cenários menos críticos, nos quais simplicidade e menor tempo de treinamento são prioridades, ainda mantendo boa interpretabilidade por gerar árvores mais enxutas.

Mesmo sem sempre garantir a otimalidade, o OCT-MIO demonstrou capacidade de encontrar soluções muito próximas do ótimo em tempos razoáveis, sendo, portanto, uma alternativa competitiva ao CART em diversos contextos.

Como trabalhos futuros iremos repetir os experimentos com mais conjuntos de dados, aumentar para 10 o número de *folds* da validação cruzada e incluir mais valores para os parâmetros dos modelos.

Referências

Aghaei, S., Gómez, A., e Vayanos, P. (2024). Strong optimal classification trees. *Operations Research*.

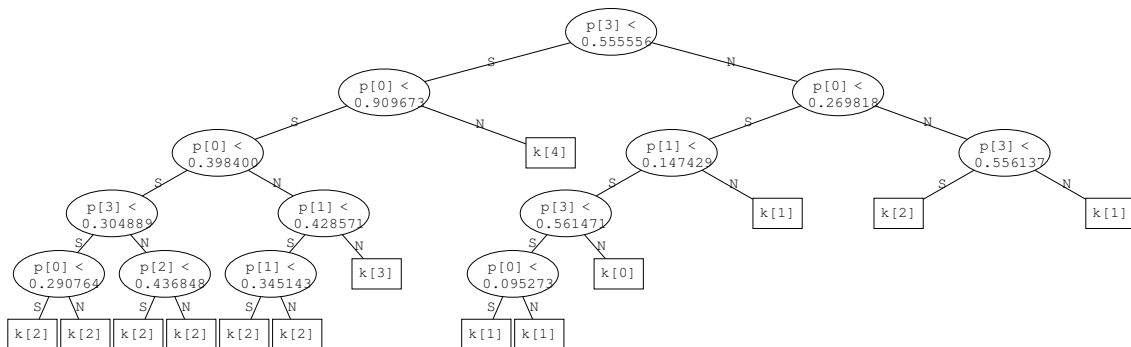


Figura 9: Árvore de classificação para o conjunto *Stress* com altura máxima = altura mínima+2 pelo modelo OCT-MIO treinada com 100% do conjunto de dados

- Bertsimas, D. e Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.
- Breiman, L., Friedman, J. H., Olshen, R. A., e Stone, C. J. (2017). *Classification and regression trees*. Routledge.
- Laavanya, R. (2022). Human stress detection in and through sleep. <https://www.kaggle.com/datasets/laavanya/human-stress-detection-in-and-through-sleep>. Kaggle Dataset.
- Linardatos, P., Papastefanopoulos, V., e Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.
- Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.
- Rachakonda, L., Bapatla, A. K., Mohanty, S. P., e Kougianos, E. (2021). Sayopillow: Blockchain-integrated privacy-assured iomt framework for stress management considering sleeping habits. *IEEE Transactions on Consumer Electronics (TCE)*, 67(1):20–29.
- Rachakonda, L., Mohanty, S. P., Kougianos, E., Karunakaran, K., e Ganapathiraju, M. (2018). Smart-pillow: An iot based device for stress detection considering sleeping habits. In *Proceedings of the 4th IEEE International Symposium on Smart Electronic Systems (iSES)*, p. 161–166.
- van der Linden, J., de Weerdt, M., e Demirović, E. (2023). Necessary and sufficient conditions for optimal decision trees using dynamic programming. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., e Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, p. 9173–9212. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5fce9627e15c84db572a66e029b1fc-Paper-Conference.pdf.
- Verwer, S. e Zhang, Y. (2019). Learning optimal classification trees using a binary linear program formulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1625–1632. URL <https://ojs.aaai.org/index.php/AAAI/article/view/3978>.