Research, Society and Development
ISSN: 2525-3409
ISSN: 2525-3409
rsd.articles@gmail.com
Universidade Federal de Itajubá
Brasil

Análise de Algoritmos de Indução de Árvores de Decisão

Rodrigues Okada, Hugo Kenji; Nascimento das Neves, André Ricardo; Shitsuka, Ricardo
Análise de Algoritmos de Indução de Árvores de Decisão
Research, Society and Development, vol. 8, núm. 11, 2019
Universidade Federal de Itajubá, Brasil
Disponível em: https://www.redalyc.org/articulo.oa?id=560662202029
DOI: https://doi.org/10.33448/rsd-v8i11.1473

Análise de Algoritmos de Indução de Árvores de Decisão

Analysis of Decision Tree Induction Algorithms

Análisis de algoritmos de inducción del árbol de decisión

Hugo Kenji Rodrigues Okada hugookadastm@gmail.com
Escola Superior Batista da Amazônia, Brasil
http://orcid.org/0000-0002-7364-7986
André Ricardo Nascimento das Neves
aricardo.neves@gmail.com
Escola Superior Batista da Amazônia, Brasil
http://orcid.org/0000-0002-2911-5376
Ricardo Shitsuka rshitsuka@yahoo.com
Universidade Federal de Itajubá, Brasil
http://orcid.org/0000-0003-2630-1541

Research, Society and Development, vol. 8, núm. 11, 2019

Universidade Federal de Itajubá, Brasil

Recepção: 03 Agosto 2019
Revised: 06 Agosto 2019
Aprovação: 10 Agosto 2019
Publicado: 24 Agosto 2019

DOI: https://doi.org/10.33448/rsd-v8i11.1473

Redalyc: https://www.redalyc.org/articulo.oa?id=560662202029

Resumo: Árvores de decisão são estruturas de dados ou métodos computacionais que possibilitam o aprendizado de máquinas supervisionadas não-paramétricas e são usados em tarefas de classificação e regressão. O objetivo do presente artigo é apresentar uma comparação entre os algoritmos de indução de árvores de decisão C4.5 e CART. Realiza-se um estudo quantitativo no qual os dois métodos são comparados a partir de análise dos seguintes aspectos: funcionamento e complexidade. Verificou-se que os experimentos realizados apresentaram percentuais de acerto praticamente iguais no tempo de execução para a indução da arvore, entretanto, para um parâmetro crucial que é o tempo de processamento que é importante para muitas aplicações, o algoritmo CART foi aproximadamente 46,24% mais lento do que o C4.5 para o mesmo tipo de processamento evidenciando-se, desta forma que este pode ser considerando como mais eficiente. Recomenda-se utilizar o algoritmo C4.5 em aplicações que nas quais haja a preocupação com o tempo de processamento.

Palavras-chave: Estrutura de dados, Inteligência artificial, Decisão computacional, C4.5, CART.

Abstract: Decision trees are data structures or computational methods that enable nonparametric supervised machine learning and are used in classification and regression tasks. The aim of this paper is to present a comparison between the decision tree induction algorithms C4.5 and CART. A quantitative study is performed in which the two methods are compared by analyzing the following aspects: operation and complexity. The experiments presented practically equal hit percentages in the execution time for tree induction, however, the CART algorithm was approximately 46.24% slower than C4.5 and was considered to be more effective.

Keywords: Data Structure, Artificial intelligence, Computational decision, C4.5, CART.

Resumen: Los árboles de decisión son estructuras de datos o métodos computacionales que permiten el aprendizaje automático supervisado no paramétrico y se utilizan en tareas de clasificación y regresión. El objetivo de este trabajo es presentar una comparación entre los algoritmos de inducción del árbol de decisión C4.5 y CART. Se realiza un estudio cuantitativo en el que se comparan los dos métodos mediante el análisis de los siguientes aspectos: operación y complejidad. Los experimentos presentaron porcentajes de aciertos prácticamente iguales en el tiempo de ejecución para la inducción

del árbol; sin embargo, el algoritmo CART fue aproximadamente un 46,24% más lento que C4.5 y se consideró más efectivo.

Palabras clave: Estructura de datos, Inteligencia artificial, Decisión computacional, C4.5, CART.

1. Introdução

A Inteligência Artificial faz o uso de várias técnicas e uma deles é a da Árvore de Decisão. Esta normalmente é utilizada em qualquer tipo de classificação de dados produzindo ótimos resultados além disso, ela é considerada como sendo uma técnica simples e relativamente fácil de ser empregada e que pode ser implementada por meio de diferentes algoritmos.

O problema a ser abordado neste artigo é analisar dois tipos de algoritmos de Árvores de Decisão, que são o CART (*Classification and Regression Tree*) e o C4.5 para se verificar a eficiência no funcioamento. Esta eficiência pode ser traduzida por meio de tempos menores de processamento que se tornam importantes em uma sociedade na qual as pessoas têm pressa e qualquer demora em algum processamento já se torna motivo para a impaciência do usuário.

O objetivo do presente artigo é apresentar uma comparação entre os algoritmos de indução de árvores de decisão C4.5 e CART. Este estudo comparativo é realizado por meio de um experimento executado com uma base de dados de classificação de tipos de câncer. Os dois métodos foram aplicados ao problema e foram comparados em termos de tempo de execução e eficácia.

Ao final foi escolhido o método que obteve os melhores resultados de predição e se encontrou a classe que se aproxima melhor do resultado com tempo menor de execução.

O artigo está organizado do seguinte modo: a seção 2 relata o que são árvores de decisão, os tópicos 3 e 4 apresentam o funcionamento e a complexidade dos algoritmos C4.5 e CART. Os experimentos são apresentados na seção 5 e, as considerações finais estão na 6.

2. O que são Árvores de Decisão

A computação procura trabalhar os dados transformando-os em informações que sejam úteis aos usuários. O trabalho com os dados exige que se tenha alguma estrutura. Estruturas de dados (ED) são coleções de valores, seus relacionamentos e operações nestes valores e estruturas. Entre os métodos importantes para trabalhar as estruturas na computação atual estão as árvores de decisão.

Para Cripaldi (2010), as árvores de decisão são formas de representar o conhecimento, construindo classificadores para predizer a que classe dados desconhecidos pertencem, sendo essas informações baseadas nos valores de um conjunto de dados, ou seja, métodos de classificação de dados. Alguns tipos de algoritmos de árvores de decisão são: SimpleChart, CART, RandomTree, REPTree, BFTree e o algoritmo C4.5.

O raciocínio dos algoritmos é selecionado e usado pelos programadores e analistas na elaboração dos seus códigos de programação conforme a necessidade e aplicação e eles podem possuir eficácias e eficiências diferentes conforme o tipo de problema.

Uma árvore de decisão é composta por nós (representações feitas em círculos ou quadrados) onde estes vértices estão conectados entre si por ramos (linhas), essa intercalação é feita até que encontre uma folha, sendo esta a representação das classes (Ragsdade, 2001). O primeiro nó é denominado de raiz e, é apresentado na árvore como sendo o atributo mais importante para conseguir classificar os dados, sendo os próximos nós menos relevantes e utilizados em ordem de importância. Cada percurso da árvore (com início na raiz e término em uma folha) corresponde a uma regra de classificação (Garcia, 2000).

A principal característica de uma árvore de decisão é como ocorre a tomada de decisões, este fato leva em consideração todos os atributos da base de dados, permitindo que usuários possam identificar quais atributos influenciam e são os mais importantes para determinado trabalho. Nascimento (2011) relata que a complexidade padrão de uma árvore de decisão é O(nlogn), sendo esta, gerada pela profundidade máxima de uma de suas folhas.

3. Algoritmo C4.5

O algoritmo C4.5 foi desenvolvido por J. Ross Quinlan no ano de 1993, serve para classificar problemas em aprendizagem de máquina e mineração de dados (Wu, 2009). Considerado o sucessor do ID3 (*Iterative Dichotomiser* 3) é provavelmente o mais popular entre os algoritmos de árvore de decisão.

Seu objetivo é construir uma árvore de decisão a partir de um conjunto de dados de treinamento. Os valores das entradas fornecidos para o C4.5 podem ser discretos ou contínuos. O C4.5 não restringe a divisão da árvore em binária, exceto para árvores de regressão (Ruggieri, 2002).

Para a construção da árvore, o algoritmo utiliza o paradigma de programação conhecido como divisão e conquista, combinado com o método guloso, pois, escolhe sempre as melhores partições e melhores atributos avaliados localmente, sem se preocupar se este passo, junto à sequência completa, irá ao final produzir a melhor solução (Barbosa, 2012). A seleção de quais atributos devem ser associados ao nó de decisão é feita através do uso do critério de ganho de informação (*info gain*). Este ganho mensura o quanto um atributo é capaz de separar o conjunto de instâncias em categorias distintas. Aquele que possuir o maior ganho é selecionado para ser incluído a árvore.

Durante o processo de indução da árvore de decisão, o algoritmo divide recursivamente um conjunto de instâncias até que cada subconjunto obtido desta divisão contenha casos de uma única classe. Apresentaremos a seguir, no Quadro 1, o pseudocódigo principal do C4.5:

Quadro 1 -
Pseudocódigo Algoritmo C4.5.

C4.5(D) Entrada: um valores de atributo do dataset D 01: Arvore = { } 02: Se D é "puro" OU outro critério de parada encontrado então 03: retorna 04: Para todos os atributos a ∈ D faça 05: Computa o critério de informação em a 06: abest = Melhor atributo de acordo com o cálculo do critério computado 07: Arvore = Cria um nó de decisão e testa o melhor na raiz 08: Dv = Induz subconjunto de D baseado em abest 09: Para todo Dv faça 10: Arvorev = C4.5 (Dv) 11: Anexa Arvorev ao ramo correspondente da arvore 12: retorna Arvore

Fonte: os autores.

A linha (01) cria o conjunto árvore inicialmente vazio. Nas linhas (02 e 03) é avaliado o critério de parada da construção da árvore, ou seja, caso *D* seja uma folha, significa que todas as instâncias de *D* pertencem à mesma classe então retorna. As linhas (04 e 05) do algoritmo mantém um laço invariante para todos os atributos $a \in D$ e calcula o critério de informação de a utilizando o ganho informação. Dado um conjunto de entrada *D* que pode ter c diferentes classes, a entropia de *D* é dada pela fórmula: $E(D)=\sum_{i=1}^{c} p_i \log(p_i)$, onde: pi é a proporção de dados em *D* que pertence à classe *i*. O ganho de informação para um atributo *"a"*, de um conjunto de dados *D*, nos dá a medida da diminuição da entropia esperada quando utilizamos o atributo *a*, para fazer a divisão do conjunto de dados.

Seja P(*a*) o conjunto de valores que o atributo *"a"* pode ter, seja *x* um elemento desse conjunto, e seja S*x* um subconjunto de *S* formado pelos dados em que $a = x$, a entropia que se obtém ao particionar *S* em função do atributo *"a"* é dada por: $E(a)=\sum_{x \in V(a)} \frac{|S_x|}{|S|} Entropia S_x$. O ganho de informação será então: Ganho(S, *a*) = Entropia(S) − E(*a*), onde: Entropia(S) é uma medida de não homogeneidade do conjunto S e E(*a*) é uma medida de não homogeneidade estimada para o conjunto.

Nas linhas (06 e 07) o atributo com o maior (melhor) valor computado é atribuído à variável *abest*. A partir desta escolha, adiciona o nó no conjunto Árvore, testando os diferentes valores possíveis de *abest* buscando a regra que melhor separe as instancias de *D*.

Na linha (8) o algoritmo induz a divisão do conjunto Árvore em um subconjunto menor *Dv* a partir da escolha do nó obtido na linha (6). Para todo o subconjunto *Dv* chama recursivamente o C4.5. e anexa o subconjunto o conjunto Árvorev. Este processo das linhas (9 e 11) se repete até que os subconjuntos se tornem "puros", isto e, todos os valores são identificados na mesma classe (folha), encerrando o crescimento da altura da árvore. Por fim, o algoritmo retorna o conjunto Árvorev.

3.1. Análise de complexidade do algoritmo C4.5

Segundo Ruggieri (2002) a fase da construção da árvore é de longe a fase mais dispendiosa do C4.5, portanto, concentramos apenas na construção da árvore para analisarmos o custo referente à criação de um único nó. Denotamos por *D* o conjunto de casos associados a um nó. Vamos introduzir mais algumas notações: *mc* é número de atributos contínuos e *md* é o número atributos discretos ainda não selecionado no nó anterior.

A construção de um único nó requer, em média, as seguintes etapas: *md* * | *D* | para calcular a relação de ganho de informação para atributos discretos ou *mc* * | *D* | (log | *D* | + 1) para classificar casos em | *D* | (Linha 05) e computar os valores de ganho de informação. O C4.5 adota um algoritmo de ordenação *Quicksort* para realizar a permutação dos valores de ganho calculados. Cormen (2009) avalia o algoritmo *quicksort*, no caso médio, com complexidade de tempo nlogn. Estes passos dão origem à seguinte complexidade de tempo para o C4.5 de O(*m**n log n) sendo m os atributos e nlogn a ordenação do ganho de informação.

4. Algoritmo CART

A árvore de decisão CART (*Classification and Regression Trees*) tem como objetivo criar suas decisões através de uma divisão estritamente binária com alvo em seus atributos principais e seus percursores, ou seja, ao usar o primeiro nó (raiz da árvore) este só poderá estar ligado a dois nós, assim sucessivamente ao crescimento da árvore, até que encontre uma folha (Wu, 2009).

O CART utiliza o *Gini Index* para medir o índice de impureza do dataset a ser analisado, este cálculo é determinado por:

$$\text{Gini}(D)=1-\sum_{i=1}^{m} i^2,$$

Onde:

D representa o dataset.

Pi a probabilidade das instancias em D que pertencem à classe C_i.

O somatório é calculado para todas as classes. O *Gini Index* considera uma divisão binária para cada atributo, onde, cada atributo possui os valores distintos que ocorrem no *dataset* (Han, 2012).

A forma com que essa árvore cresce é relacionada à sua quantidade de decisões, gerando divisões binárias até que não existam mais possibilidade. Como resultado, o nó não pode mais ser dividido, este então é transformado em nó terminal (folha) (Bittencourt, 2003). Apresentamos a seguir, no Quadro 2, o pseudocódigo do algoritmo CART.

Quadro 2-
Pseudocódigo Algoritmo CART.

CART(D) Entrada: um valores de atributo do dataset D01: Se D ∈ a uma única classe então 02: retorna um nó folha. 03: Senão 04: Se D estiver vazio então 05: retorna um nó folha com a classe mais comum de D. 06: Senão 07: escolhe o melhor atributo F e cria dois nós. 08: Para cada possível valor vi de F: 09: Seja Dv subconjunto que tenha valor vi para F 10: Adicione as arestas a partir dos nós com o valor vi. 11: Se Dv estiver vazio então 12: Adicione um nó folha ligado a aresta com a classe v mais comum de Dv. 13: Senão 14: Arvore = CART(Dv) 15: Retorne a Arvore

Fonte: os autores.

Nas linhas (01 e 02) ocorre o critério de parada da construção da árvore, ou seja, caso *D* pertença à única classe, retorna uma folha. As linhas (04 e 05) verifica se *D* está vazio, retorna um nó folha que é mais comum para o *D*. Em (07) acontece a escolha do melhor atributo e cria dois nós, nesta linha acontece o critério de informação através do método *Gini index*.

As linhas (09 e 10) adiciona arestas aos nós de acordo com o subconjunto, gerado a partir do conjunto, menos o atributo já utilizado. Em (11 e 12) verifica se *Dv* está vazio, caso esteja adiciona um nó folha sendo a classe mais comum de *Dv*. A linha (14) retorna de forma recursiva o subconjunto (*Dv*) ao algoritmo CART, adicionando a árvore.

4.1. Análise de complexidade do algoritmo CART

A complexidade de tempo total do CART é O(nlog2n), onde n representa o número de nós folhas existentes na árvore. Este algoritmo sempre realiza uma partição binária, utilizando a fórmula do *Gini Index* para determinar a impureza de um atributo, dividindo em dois nós, o nó da direita e o nó da esquerda, julgando ser a melhor divisão.

Após o cálculo, o CART ordena estes atributos com o uso de um método de ordenação com tempo nlogn. Cormen (2015) diz que os algoritmos de ordenação *heapsort, mergesort* e *quicksort*, possuem no caso esperado complexidade de tempo nlogn.

Cada nó é gerado por (2p-1-1), onde p são os distintos valores a serem comparados, resultando na regra de divisão. Com isso temos que o custo total do CART é O(2#−1 #).

5. Metodologia, Resultados e Discussão

A pesquisa tem com finalidade a busca por novos saberes. Quando ela envolve números, porcentagens e/ou estatísticas ela é de natureza quantitativa como consideram Pereira, Shitsuka, Parreira & Shitsuka (2018). No presente estudo faz-se uma comparação de desempenho, quantitativo entre dois algoritmos de árvore de busca utilizados para o trabalho com dados.

Para fins de experimentos dos algoritmos C4.5 e CART executamos testes de indução de árvore utilizando um *dataset* do "Mapa Global de Câncer" referente aos diagnósticos de câncer usando expressão de assinaturas genéticas do tumor.

A base de treinamento contém 16.063 atributos e 144 instâncias com 14 diferentes classes de câncer {Breast, Prostate, Lung, Colorectal, Lymphoma, Bladder, Melanoma, Uterus Adeno, Leukemia, Renal, Pancreas, Ovary, Mesothelioma, CNS}. O *dataset* encontra-se disponível em http://eps.upo.es/bigs/datasets.html. Os experimentos foram executados no *software* de uso livre e gratuito WEKA v3.7 para Linux.

No WEKA, o algoritmo C4.5 é denominado de J48. O único parâmetro alterado para a execução do algoritmo foi *unpruned* definido em *true* com objetivo de não realizar a poda da árvore. O tempo de execução para a construção da árvore foi de 2,65 segundos, o número de folhas 18 e 35 nós atingindo um percentual de acerto de 96,5278%. Abaixo, no Quadro 3, segue o resultado da execução do algoritmo C4.5:

Quadro 3 -
Classifier model (full training set) === J48 unpruned tre.

D49958_at <= 156 | AA234791_at <= 196| | X93512_at <= 33 | | | M10050_at <= 26 | | | | RC_AA121266_at <= 105 | | | | | M61906_at <= 141 | | | | | | U66559_at <= -50 | | | | | | | AFFX-BioB-5_at <= -24: Uterus Adeno (2.0/1.0) | | | | | | AFFX-BioB-5_at > -24: Prostate (2.0/1.0)| | | | | U66559_at > -50 | | | | | | X58079_at <= 102 | | | | | | | | L07515_at <= 63 | | | | | | | | | M24461_at <= 76 | | | | | | | | | | RC_AA457376_at <= -10 | | | | | | | | | | | D50924_at <= 29: Pancreas (8.0) | | | | | | | | | | | D50924_at > 29: Bladder (8.0/1.0) | | | | | | | | | | RC_AA457376_at > -10 | | | | | | | | | | | D14134_at <= -2: Breast (8.0)| | | | | | | | | | D14134_at > -2: Lung (2.0/1.0) | | | | | | | | | M24461_at > 76: Lung (7.0) | | | | | | L07515_at > 63| | | | | | | | M10950_cds2_at <= -23: Melanoma (2.0/1.0) | | | | | | | | | M10950_cds2_at > -23: Mesothelioma (7.0) | | | | | | | X58079_at > 102 | | | | | | | | D37931_at <= 33: Melanoma (7.0) | | | | | | | | D37931_at > 33: Ovary (7.0) | | | | | M61906_at > 141: Uterus Adeno (7.0) | | | | RC_AA121266_at > 105: Prostate (7.0) | | | M10050_at > 26: Colorectal (7.0) | | X93512_at > 33: Renal (7.0) | AA234791_at > 196 | | D89377_at <= 84: Lymphoma (16.0) | | D89377_at > 84: Leukemia (24.0) D49958_at > 156: CNS (16.0) Number of Leaves : 18 Size of the tree : 35 Time taken to build model: 2.65 seconds

Fonte: os autores.

O CART é denominado de *SimpleCART* no *software* WEKA. O parâmetro *useprune* foi definido em *false* evitando a poda da árvore. O tempo de execução para a construção da árvore foi de 5,73 segundos, o número de folhas 17 e 33 nós alcançando um percentual de acerto de 95,1389%. Abaixo, Quadro 4, segue o resultado da execução do algoritmo CART:

Quadro 4 -
Classifier model (full training set) === CART Decision Tree.

X59798_at < -206.0: Leukemia(24.0/1.0) X59798_at >= -206.0 | D49958_at < 167.5 | | U10485_at < 134.5 | | | M10050_at < 66.0 | | | | X93512_at < 30.5 | | | | | AA234665_at < 422.5 | | | | | | M61906_at < 165.5 | | | | | | | U63824_at < 65.0 | | | | | | | | AA456343_at < 148.5 | | | | | | | | | M24461_at < 74.5 | | | | | | | | | | RC_AA478809_at < 4.0: Pancreas(7.0/0.0) | | | | | | | | | | RC_AA478809_at >= 4.0 | | | | | | | | | | | D31120_at < -2.5 | | | | | | | | | | | | RC_AA446964_at < 42.5 | | | | | | | | | | | | | AFFX-CreX-5_st < 11.0: Breast(7.0/0.0) | | | | | | | | | | | | | AFFX-CreX-5_st >= 11.0 | | | | | | | | | | | | | | AFFX-BioB-5_at < -9.0: Colorectal (1.0/2.0) | | | | | | | | | | | | | | AFFX-BioB-5_at >= -9.0: Prostate(1.0/1.0) | | | | | | | | | | | | | RC_AA446964_at >= 42.5: Bladder(8.0/1.0) | | | | | | | | | | | D31120_at >= -2.5: Melanoma(7.0/0.0) | | | | | | | | | M24461_at >= 74.5: Lung(7.0/1.0) | | | | | | | | AA456343_at >= 148.5: Ovary(6.0/0.0) | | | | | | | U63824_at >= 65.0 | | | | | | | AFFX-BioDn-5_at < -128.5: Mesothelioma(8.0/0.0) | | | | | | | | AFFX-BioDn-5_at >= -128.5: Breast(1.0/1.0) | | | | | | M61906_at >= 165.5: Uterus Adeno(7.0/0.0) | | | | | AA234665_at >= 422.5: Prostate(7.0/0.0) | | | | X93512_at >= 30.5: Renal(7.0/0.0) | | | M10050_at >= 66.0: Colorectal(7.0/0.0) | | U10485_at >= 134.5: Lymphoma(16.0/0.0) | D49958_at >= 167.5: CNS (16.0/0.0) Number of Leaf Nodes: 17 Size of the Tree: 33 Time taken to build model: 5.73 seconds

Fonte: os autores.

A literatura apresenta alguns estudos comparativos como é o caso de Nascimento Jr.(2017) que verifica o grau de cobertura ou Giasson et al. (2013) que compara 5 algoritmos em problemas de mapeamento de solos e Carvalho (2005) que ao testar vários algoritmos também obteve sucesso com o emprego do C4.5.

O presente trabalho testa uma base de saúde e nela verifica, ao longo dos experimentos que eles apresentaram taxas de acerto semelhantes no tempo de execução para a indução da arvore. Já no aspecto da velocidade de processamento, observou-se que o algoritmo CART foi cerca de 46,24% mais lento do que o algoritmo C4.5. e, deste modo, extrapolando-se e levando em conta os outros dados de literatura, considera-se que se pode trabalhar com outras bases de dados para indução da árvore considerando-se e confirmando-se a melhor eficiência do algoritmo C4.5.

6. Considerações finais

O presente artigo contribui de modo semelhante a um pequeno tijolo na construção do saber sobre os algoritmos e o trabalho com dados e informações para os interessados em computação e performance ou desempenho no processamento de dados.

As árvores geradas pelo CART são melhores de serem compreendidas, uma vez que são árvores estritamente binárias enquanto que o C4.5 pode gerar árvores de maior ordem.

Os dois algoritmos possuem complexidade de tempo assintoticamente iguais sendo esta O(nlogn). Entretanto, possuem custos diferenciados de acordo com as suas técnicas de decisão e divisão das instâncias.

Os experimentos apresentaram percentuais de acerto praticamente iguais no tempo de execução para a indução da arvore, entretanto, o

algoritmo CART foi aproximadamente 46,24% mais lento do que o C4.5. Portanto pode-se concluir que independente da base de dados utilizada para indução da árvore, o algoritmo C4.5 possui uma melhor eficiência.

Torna-se interessante que se realizem estudos futuros que mostrem o desempenho de outros algoritmos bem como o trabalho com outros conjuntos de dados para se alcançar uma maior representatividade e saber sobre o tema.

Referências

Barbosa, J.M., Carneiro,T.G.S. & Tavares, A.L. (2012). *Métodos de Classificação por Árvores de Decisão*. Disciplina de Projeto e Análise de Algoritmos do PPGCC - Programa de Pós-Graduação em Ciência da Computação do Departamento de Computação (DECOM) da Universidade Federal de Ouro Preto (UFOP). Disponível em: <http://www.decom.ufop.br/menotti/paa111/files/PCC104-111-ars-11.1-JulianaMoreiraBarbosa.pdf>;. Acesso em: 03 Ago. 2019.

Bittencourt, H. R. & Clarke, R. T. (2003). *Use of classification and regression trees (CART) to classify remotely-sensed digital images*. In: Anais do International Geoscience and Remote Sensing Symposium. pp. 3751-3753. Disponível em:<www.researchgate.net/profile/Robin_Clarke4/publication/4064611_Use_of_classification_and_regression_trees_(CART)_to_classify_remotely-sensed_digital_images/links/0f317534ff5f1e2b46000000.pdf> Acesso em: 02 ago. 2019.

Carvalho, D.R. (2005). *Árvore de decisão / algoritmo genético para tartar o problema de pequenos disjuntos em classificação de dados*. Tese (Doutorado) no Programa de Pós-Graduação em computação de alto desempenho / sistemas computacionais do Programa de Engenharia Civil da Universidade Federal do Rio de Janeiro. Disponível em: http://www.ipardes.gov.br/biblioteca/docs/tese_deborah_carvalho.pdf. Acesso: 6 ago. 2019.

Cormen, T. H. (2009). *Introduction to algorithms*. MIT press, USA.

Garcia, S. C. (2003). *O uso de árvores de decisão na descoberta de conhecimento na área da saúde*. Tese (Doutorado) na Universidade Federal doRio Grande do Sul. Disponível em: <http://hdl.handle.net/10183/4703>;. Acesso em: 03 ago. 2019.

Giasson, E, Hartemink, A.E, Tornquist, C.G., Teske, R, & Bagatini, T. (2013). Avaliação de cinco algoritmos de árvores de decisão e três tipos de modelos digitais de elevação para mapeamento digital de solos a nível semidetalhado na Bacia do Lageado Grande, RS, Brasil. *Ciência Rural*, 43(11): 1967-1973. https://dx.doi.org/10.1590/S0103-84782013001100008

Han, J. & Kamber, M. (2002). Data Mining: Concepts and Techniques. 3.ed. Morgan Kaufmann/Elsevier, Waltham, MA, USA.

Nascimento, P. T. S. & Façanha, S. L. O. (2008). Árvore de decisão incompleta: reduzindo a complexidade para acelerar a decisão. In: Anais do Encontro da Associação Nacional de Pós-Graduação e Pesquisa em Administração, 32(1). Disponível em: <http://www.anpad.org.br/admin/pdf/ESO-A1183.pdf>;. Acesso em: 3 ago. 2019.

Nascimento Jr., L.A.F. (2017). Aplicando método do gradiente ótimo na otimização do cálculo do grau de cobertura das regras em árvores de decisão Fuzzy. *Revista Brasileira de Computação Aplicada* (ISSN 2176-6649), Passo Fundo, 9(3):31-43, out. 2017.

Pereira, A.S., Shitsuka, D.M., Parreira, F.J. & Shitsuka, R. (2018). Metodologia da pesquisa cientifica. Santa Maria/RS, Ed. UAB/NTE/UFSM. Disponível em: https://repositorio.ufsm.br/bitstream/handle/1/15824/Lic_Computacao_Metodologia-Pesquisa-Cientifica.pdf?sequence=1. Acesso em: 3 ago. 2019.

Ragsdale, C. T. (2010). *Spreadsheet modeling and decision analysis*. 6.ed. Cengage Learning, USA.

Ruggieri, S. (2002). Efficient C4.5. Knowledge and Data Engineering, *IEEE Transactions*, 14(2):438-444. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=991727&tag=1. Acess on: Aug., 3rd, 2019.

Wu, X. & Kumar, V. (2009). *The top ten algorithms in data mining*. Chapman & Hall/CRC, Boca Ratton, USA.

Porcentagem de contribuição de cada autor no manuscrito

Hugo Kenji Rodrigues Okada – 40 %
André Ricardo Nascimento das Neves – 40 %
Ricardo Shitsuka – 20 %