O Algoritmo de Classificação CART em uma Ferramenta de Data Mining

Lidiane Rosso Raimundo[1], Merisandra Côrtes de Mattos[1], Priscyla Waleska Targino de Azevedo Simões[1], Cristian Cechinel[2]

[1]Grupo de Pesquisa em Inteligência Computacional Aplicada - Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC - Brasil

[2]Curso de Engenharia da Computação - Universidade Federal do Pampa (UNIPAMPA/Bagé) – Pelotas, RS - Brasil

lidianerosso@pop.com.br,{mem,pri}@unesc.net

Abstract. *The computational advance regarding the processing and storage of data contributed to the formation of large repositories of data, this way requiring the development of technologies for analysis of information and acquisition of new knowledge. Among these technologies the data mining is as one of the alternatives, using algorithms to do so. This article presents the mathematical modeling and implementation of the CART algorithm for induction of decision trees in the task of classifying the process of data mining in a Shell called Orion.*

Keywords: *Data Mining, Classification, Decision Trees, CART Algorithm.*

Resumo. *O avanço computacional no que se refere ao processamento e armazenamento contribuiu para a formação de grandes repositórios de dados, tornando-se necessário o desenvolvimento de tecnologias destinadas à análise de informações e obtenção de novos conhecimentos. Dentre essas tecnologias o data mining constitui-se como uma das alternativas, utilizando para isso algoritmos. Este artigo apresenta a modelagem matemática e implementação do algoritmo CART para indução de árvores de decisão na tarefa de classificação do processo de data mining em uma Shell denominada de Orion.*

Palavras-chave*: Data Mining, Classificação, Árvores de Decisão, Algoritmo CART.*

1. Introdução

A habilidade de descobrir conhecimento útil e de agir sobre ele vem se tornando cada vez mais importante no mundo competidor em que se vive [Kantardzic 2003].

O método utilizado para a descoberta desses conhecimentos costuma ser a elaboração de modelos estatísticos, porém estes apresentam algumas limitações, como por exemplo, quando o número de informações é grande a análise se torna trabalhosa. Considerando-se aspectos como este, tem-se o data mining que reúne conhecimentos de

estatística, aprendizado de máquina e banco de dados. O processo de data mining consiste na utilização de métodos e algoritmos específicos para descobrir padrões nos dados, sendo responsável pela busca de conhecimento a partir de grandes quantidades de dados [Fayyad, Piatetsky-Shapiro, Smyth 1996].

As tarefas de data mining (associação, classificação, clusterização, regressão, entre outras) para serem realizadas necessitam que um determinado método (árvores de decisão, redes neurais artificiais, algoritmos genéticos, entre outros) seja adotado, de acordo com o conhecimento que se deseja extrair da base de dados. Esses métodos, bem como as tarefas mencionadas anteriormente, encontram-se disponíveis em diversas ferramentas de data mining.

As ferramentas de data mining disponíveis, em sua maioria são licenciadas, tendo-se poucas ferramentas gratuitas, como por exemplo, a Weka. Porém, esta não possui o algoritmo CART implementado, além de apresentar algumas limitações no que se refere a conexão com diferentes sistemas gerenciadores de banco de dados, trabalhando com um formato próprio denominado de .arff. Outro exemplo de ferramenta gratuita é a Shell Orion Data Mining Engine, cujo desenvolvimento em ambiente acadêmico encontra-se ainda em fase incial, tendo-se como objetivo a implementação de diferentes métodos e algoritmos de data mining, bem como a conexão com variados sistemas gerenciadores de banco de dados o que pode proporcionar uma facilidade na utilização dos dados no formato em que o usuário os tem disponíveis. Até o momento a Shell Orion apresenta implementados os métodos de classificação pelo algoritmo ID3 e CART, associação pelo Apriori, clusterização pelo K-means, Kohonen e Gustafson-Kessel e suporta conexões com os bancos de dados Firebird, MySQL, PostgreSQL e HSQL. No decorrer das pesquisas as funcionalidades da ferramenta serão ampliadas pela conexão com outros sistemas gerenciadores de banco de dados e a inserção de outras tarefas e métodos de data mining.

Além disto, estas pesquisas que estão sendo desenvolvidas enfatizam a modelagem matemática destes algoritmos, pois não basta implementá-los e utilizá-los é necessário conhecer como eles efetivamente funcionam e descobrem o conhecimento. Apesar de alguns destes, como por exemplo, o CART serem algoritmos bastante utilizados e conhecidos, há carência de bibliografias que demonstrem de forma clara e didática como ocorre o seu funcionamento.

Este artigo apresenta a modelagem matemática e implementação na Shell Orion Data Mining Engine do algoritmo CART para indução de árvores de decisão na tarefa de classificação.

A classificação é um processo que encontra propriedades comuns entre um conjunto de registros pertencentes a um banco de dados e as classifica em diferentes classes de acordo com um modelo [Han e Kamber 2001].

O processo de classificar pode ser entendido como a busca por uma função que possibilite a correta associação de cada registro de um conjunto de informações X, a um único registro de um conjunto Y, que corresponde a classe [Goldschmidt e Passos 2005].

A aplicação da tarefa de classificação é efetuada por meio de métodos, sendo que o de árvores de decisão consistiu-se na base do algoritmo de classificação CART. O

método de árvores de decisão é uma abordagem de aprendizado supervisionada, ou seja, compreende a abstração de um modelo de conhecimento a partir dos dados apresentados na forma de pares ordenados (entrada, saída desejada) [Goldschmidt e Passos 2005]. Neste método, a produção dos resultados apresenta simplicidade e legibilidade para a sua interpretação, fato este que, segundo Rodrigues (2005), tornou-se uma das principais vantagens da sua utilização.

No que se refere ao algoritmo CART uma das suas principais características é a capacidade de pesquisa e relações entre os dados, envolvendo as fases de construção e simplificação da árvore de decisão, escolhendo a melhor variável para divisão dos dados em dois nós, onde o procedimento de divisão é aplicado recursivamente aos dados em cada um dos nós-filhos e assim por diante [Hand et al 2001].

2. O Algoritmo CART para Indução de Árvores de Decisão

Os conceitos e o funcionamento do algoritmo CART apresentados são exclusivamente extraídos do livro Classification and Regression Trees, apresentado pelos desenvolvedores do algoritmo, os quatro estatísticos: Leo Breiman, Jerome Friedman, Richard Oslen e Charles Stone [Breiman et al 1984].

Na execução da tarefa de classificação por meio do algoritmo CART existem quatro elementos envolvidos no crescimento de uma árvore: conjunto de perguntas binárias, critérios de divisão, funcionamento do critério de Gini, associar uma classe a folha e a poda.

O conjunto Q de perguntas binárias é utilizado para efetuar a divisão de cada nó t. Nos casos em que a resposta for sim, segue-se para o nó esquerdo e nos que a resposta for não, para o nó direito.

O critério para a divisão baseia-se na construção de uma árvore de decisão pelo CART sendo que a primeira tarefa é descobrir qual dos atributos realiza a melhor divisão onde cada atributo é levado em consideração, sendo testado como possível divisor.

Na etapa de seleção do atributo, onde se procura diminuir a impureza[1] dos nós, pode-se definir o ganho de uma divisão como sendo a diferença entre a impureza do nó pai e a soma desta para os nós filhos. Se a divisão s separar o nó t em dois subconjuntos tL e tR com as proporções pL e pR, o ganho da impureza será dado por:

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \quad (1)$$

O algoritmo CART considera três critérios possíveis para selecionar a melhor divisão dos dados: Entropia, Critério de Twoing e Critério de Gini.

Nos testes efetuados pelos autores do CART, comparando o Gini com o Twoing, aplicando-se em vários conjuntos de dados, onde os resultados diferem, as divisões de Gini são as melhores, portanto, geralmente é preferível o uso do critério de Gini.

[1] Segundo Breiman et al (1984), um nó é puro quando todos os casos pertencem a uma única classe do nó. Quando o número de classes uniformemente distribuídas neste nó aumenta, maior é a impureza do nó.

Mediante essa definição, optou-se pela utilização do critério de Gini neste estudo e na implementação do algoritmo CART na Shell Orion Data Mining Engine.

No critério de Gini é medida a heterogeneidade dos dados e pode ser entendido como a probabilidade condicional de erro dado um conjunto de treinamento selecionado de forma aleatória que é dividido em um nó t, onde cada classe j tem uma probabilidade p(j/t).

Segundo o critério de Gini, a impureza de um nó é dada por:

$$i(s,t) = 1 - \sum_{j} p^2(j/t) \qquad (2)$$

Onde: p (j / t) – probabilidade a priori da classe j se formar no nó t.

Em um problema com três classes, por exemplo, sendo J = 3, as probabilidades são escritas por p(1/ t), p(2 / t) e p(3 / t). Assim, se um nó t possuísse um total de 150 casos, onde para uma variável explicativa numérica esses casos fossem distribuídos igualmente nas três classes, ou seja, 50 para cada uma (50/50/50), as probabilidades das classes, seriam:

p (1 / t) = 50/150 =0.3333 p (2 / t) = 50/150 = 0.3333 p (3 / t) = 50/150 = 0.3333

Dada uma distribuição da probabilidade para cada classe, pode-se calcular a impureza deste nó. Neste caso o resultado do critério de Gini, por meio da equação (2), consiste em:

i(s,t) = 1 - ((1/3)2 +(1/3)2 +(1/3)2) = 1 - (0,111 + 0,111 + 0,111) = 1 - 0.333 = 0.667

Neste exemplo o índice de Gini é i(s,t) = 0.667. Supondo-se que o cálculo apresentado pertence ao nó pai da árvore, e que este foi subdivido em dois nós filhos, sendo que o nó esquerdo contém 50 casos (50/0/0) e o nó direito 100 casos (0/50/50), o cálculo de Gini para os nós filhos é dado por:

i(tL) = 1 – ((1/1)2 +(0)2 +(0)2) i(tR) = 1 – ((0) 2 +(1/2) 2 +(1/2) 2)

i(tL) = 1 – 1 i(tR) = 1 – (0,25 + 0,25)

i(tL) = 0 i(tR) = 0,5

Tendo-se o resultado de Gini para os nós pai e filhos, é possível encontrar o ganho da divisão por meio da equação (1):

Δi(s,t) = 0.667 – ((50/150) * 0 + (100/150) *0.5) = 0.667 – 0.3333 = 0.3333

Esse procedimento é realizado para todas as variáveis envolvidas no problema, sendo que a escolhida para a divisão do nó será a que possuir o maior índice de Gini. Depois de encontrado o melhor ponto de divisão, e com isso efetuada a divisão do nó, caso não exista mais ganho em dividi-, faz-se a associação de uma classe à folha.

O critério utilizado para associar uma determinada classe à folha é a atribuição da classe mais provável, cuja classe a ser atribuída à folha será a mais provável dentro dos exemplos que se encontram nessa folha. Seu cálculo é dado por:

$$\max_j(p_j) = \max_j \frac{N_j}{N} \tag{3}$$

Onde: j – 1....J, sendo J o número de classes

N – número total de exemplos na folha

Nj – número de exemplos da classe j na folha

Supõe-se que um nó t tornou-se folha contendo 53 casos com a seguinte distribuição entre três classes: classe 1 = 50, classe 2 = 1 e classe 3 = 2. Aplicando-se a equação (3) para encontrar a classe que será atribuída a essa folha, tem-se:

Classe 1 = 50/53 = 0,9434 Classe 2 = 1/53 = 0,0189 Classe 3 = 2/53 = 0,0377

Observando-se os resultados demonstrados, verifica-se que a classe 1 foi a que apresentou a maior probabilidade, portanto, deve-se atribuí-la ao nó t. Após o término da construção da árvore, é adotada uma técnica chamada poda.

A técnica da poda por minimização do custo de complexidade, cujo objetivo é a remoção ou corte dos ramos e sub-árvores que não são relevantes para a resolução do problema, consiste na execução de três passos: criar uma árvore inicial; podá-la para obter um conjunto de árvores menores; utilizar uma estimação de erro de modo a selecionar a melhor de todas as árvores criadas.

A poda implica na adoção de um critério de comparação entre árvores de igual dimensão, utilizando o número de nós terminais (folha). A maneira correta de efetuar essa comparação é por meio do cálculo do erro estimado por resubstituição.

O erro estimado por resubstituição de um nó t é dado por:

$$r(t) = 1 - \max_j p(j/t) \tag{4}$$

Sendo que o erro total desse nó será, portanto:

$$R(t) = r(t)p(t) \tag{5}$$

Considerando $\tilde{\mathrm{T}}$ como sendo o número de nós terminais de uma árvore T, o erro estimado por substituição total pode ser calculado como sendo:

$$R(\mathrm{T}) = \sum_{t \in \tilde{\mathrm{T}}} R(t) \tag{6}$$

Assim, consegue-se definir o custo-complexidade de uma árvore T por meio da equação:

$$R_\alpha(\mathrm{T}) = R(\mathrm{T}) + \alpha\left|\tilde{\mathrm{T}}\right| \tag{7}$$

Onde: R(T) – erro estimado por resubstituição da árvore T

$\left|\tilde{\mathrm{T}}\right|$ – número de nós terminais da árvore

α – constante ≥ 0 chamada de parâmetro de complexidade

No processo de geração da seqüência de árvores, considera-se tL e tR como sendo os nós esquerdo e direito resultantes da divisão do nó t. Devendo-se verificar para qualquer árvore:

$$R(t) \geq R(t_L) + R(t_R) \tag{8}$$

Para encontrar a árvore T1 se deve percorrer todos os nós de T_{max} e, caso $R(t) = R(t_L) + R(t_R)$, elimina-se os nós tL e tR. Este procedimento destina-se a eliminar todos os nós inúteis de forma a criar uma primeira árvore a partir da qual inicia-se a verdadeira seqüência de redução, explicada a seguir.

Seja {t} o nó que resulta da eliminação do ramo Tt, para todos os ramos da árvore dado que, $R_\alpha(T_t) = R(T_t) + \alpha\left|\tilde{T}_t\right|$ e que $R_\alpha(\{t\}) = R(\{t\}) + \alpha$, deve-se procurar o valor de α para o qual tem-se $R_\alpha(T_t) = R_\alpha(\{t\})$. Deste modo, obtém-se a equação $R(T_t) + \alpha\left|\tilde{T}_t\right| = R(\{t\}) + \alpha$, que resolvendo-a por ordem de α, resulta em:

$$\alpha = \frac{R(t) - R(T_t)}{\left|\tilde{T}_t\right| - 1} \tag{9}$$

Contudo, deve-se então escolher o ramo que contém um valor de α mais baixo. Depois de aplicada a poda deve-se escolher qual delas será o classificador final.

A árvore final é encontrada por meio de uma técnica denominada Cross-Validation ou Validação Cruzada, onde após calculada a sua seqüência de reduções, divide-se esse conjunto aleatoriamente em V (normalmente 10) subconjuntos. Dessa maneira serão construídas V árvores diferentes, utilizando para tal (V-1)/V dos casos, sendo que o restante 1/V é utilizado para avaliar o erro.

Seja $T^{(v)}(\alpha)$, onde v=1,...,V a árvore que possui o menor custo-complexidade para cada α. Utilizando o subconjunto do conjunto de treinamento inicial que não foi empregado no seu desenvolvimento, define-se $N_{ij}^{(v)}$ como o número de casos da classe j que foram classificados incorretamente pela árvore $T^{(v)}(\alpha)$. Desse modo, o número total de casos da classe j mal classificados por todas as árvores de complexidade α é dado por:

$$N_{i,j} = \sum_{i,j} N_{ij}^{(v)} \tag{10}$$

Conseqüentemente, a probabilidade do erro por validação cruzada será:

$$R^{(CV)}(T(\alpha)) = \frac{1}{N}\sum_{i\#j} N_{i,j} \tag{11}$$

Resultando-se assim, no cálculo do erro para uma árvore de complexidade α. Nota-se que as árvores construídas com base no conjunto de treinamento inicial são idênticas a Tk para valores de α tal que $\alpha_k \leq \alpha < \alpha_{k+1}$. Por fim,

$$\alpha_k^{'} = \sqrt{\alpha_k \alpha_{k+1}} \tag{12}$$

Onde $\alpha_k^{'}$ é o centro geométrico entre α_k e α_{k+1}. O erro por validação cruzada da k-enésima árvore da seqüência de reduções efetuada sobre a árvore produzida e baseada no conjunto de treinamento total será dada por:

$$R^{(CV)}(\mathrm{T}_k) = R^{CV}\left(\mathrm{T}(\alpha_k^{'})\right) \tag{13}$$

Portanto, será escolhida a árvore que permita minimizar o valor do erro calculado podendo-se utilizar esse valor como estimativa de erro para o classificador final.

3. O Algoritmo CART na Tarefa de Classificação da Shell Orion Data Mining Engine

Inicialmente realizou-se a apresentação dos cálculos referentes ao índice de Gini, atribuição da classe mais provável para uma folha, poda por minimização do custo-complexidade e valores de α para selecionar a árvore final. Os cálculos a serem apresentados foram aplicados em alguns atributos da base de dados[2] referente as informações de três tipos de plantas da família das Iridáceas: setosa, versicolor e virgínica. Os dados estão distribuídos em 150 registros e possui quatro atributos numéricos: comp_setala (comprimento da sépala), larg_setala (largura da sépala), comp_petala (comprimento da pétala), larg_petala (largura da pétala) e um atributo categórico: especies (espécies).

Considerando-se que o objeto de saída escolhido seja especies, deve-se identificar quais as classes são geradas por esse atributo, ou seja, os valores que o mesmo pode assumir: íris_setosa (50 ocorrências), íris_versicolor (50 ocorrências), íris_virginica (50 ocorrências).

Aplicando-se a equação 1 nesta base de dados, inicialmente encontrou-se o Gini e o ponto de divisão dos atributos. De modo a simplificar o entendimento do cálculo do critério de Gini são apresentados na Figura 1 os passos que devem ser seguidos para encontrar o melhor ponto de divisão para cada atributo.

Após realizadas as iterações para todos os atributos da base, a fim de encontrar aquele que irá efetuar a divisão do nó raiz, chega-se aos resultados apresentados na Tabela 1.

[2] Esta base de dados está disponível nas ferramentas WEKA 3.5.5 e CART 5.0

1. cria-se uma variável (Ex: *melhorGini)*;
2. uma variável recebe o primeiro valor do atributo (Ex: *divisaoCorrente* = 4.9);
3. uma variável que corresponderá a quantidade de dados que estão à direita, recebe o valor zero (Ex: *direita* = 0);
4. uma variável que corresponderá a quantidade de dados que estão à esquerda, recebe o total de dados existentes para o atributo (Ex: *esquerda* = 6);
Enquanto existirem dados, executam-se os passos:
5. caso o primeiro dado da lista do atributo for maior que *divisaoCorrente*, encontrar o valor de Gini;
6. calcula-se o índice de Gini para o primeiro dado da lista do atributo;
7. se for o primeiro Gini encontrado, *melhorGini* recebe este valor; se não for, caso o valor de Gini encontrado for maior que o armazenado em *melhorGini*, então *melhorGini* passa a conter o valor de Gini encontrado;
8. *divisaoCorrente* recebe o próximo dado do atributo (Ex: *divisaoCorrente* = 5.4);
9. *direita* recebe o que possui + 1 e *esquerda* o -1;
10. caso executado o passo 6, encontrar o ponto de divisão: (valor do passo 5 + *divisaoCorrente*) /2.

Figura 1. Passos para o cálculo do critério de Gini

Conforme se pode observar, o atributo a ser utilizado para a divisão do nó será o *comp_petala*, pois foi o que atingiu o maior índice de Gini e seu ponto de divisão será 2.45. O atributo *larg_petala* atingiu o mesmo valor, porém nestes casos, o primeiro atributo encontrado é o selecionado. Na Figura 2 tem-se a primeira divisão, onde nos casos em que *comp_petala* $\leq$ 2.45 segue para a esquerda do nó e se *comp_petala* $\geq$ 2.45 direciona-se para a direita.

Tabela 1. Índice de Gini encontrado para os atributos

Atributo	Indice de Gini	Ponto de Divisão
comp_sepala	0.2277603	5.45
larg_sepala	0.1203704	3.35
comp_petala	0.3333333	2.45
larg_petala	0.3333333	0.8

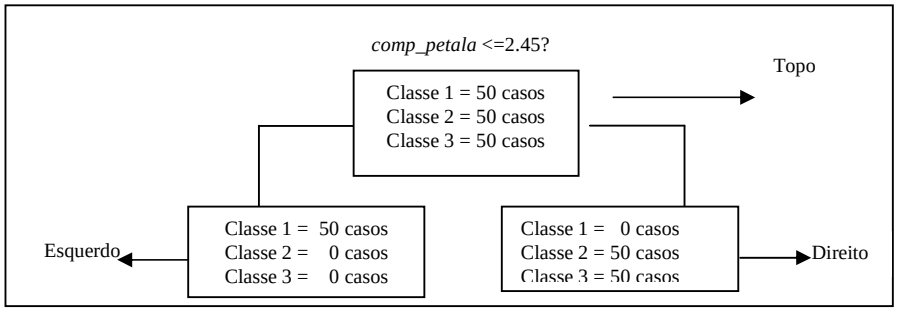

Figura 2. Divisão do primeiro nó da árvore

Aplicando-se todos os procedimentos mencionados anteriormente com o objetivo de efetuar a divisão do nó esquerdo, obtém-se para todos os atributos o índice de Gini igual a zero. Portanto, esse será um nó terminal, ou seja, uma folha da árvore e quando se encontra uma folha atribui-se a classe mais provável (maior probabilidade) a ela, que neste caso é a classe 1 conforme demonstrado a seguir:

Classe 1 = 50/50 = 1 Classe 2 = 0/50 = 0 Classe 3 = 0/50 =0

O desenvolvimento da árvore dá-se aplicando todos esses procedimentos. Assim, quando todos os casos dos nós da árvore apresentarem a mesma classe encerra-se o

desenvolvimento da mesma. Após desenvolvida a árvore, deve-se podá-la de modo a gerar uma seqüência de sub-árvores. A seguir, tem-se os dados necessários para encontrar os valores de α para cada árvore. O cálculo é efetuado da seguinte maneira: percorre-se a árvore partindo da raiz, ao encontrar o último nó terminal, aplica-se a equação 9 (Tabela 2).

Tabela 2. Parâmetro de complexidade para cada subárvore gerada

Árvore	Nós terminais	Parâmetro de Complexidade
1	5	0.0 (sempre inicia com 0)
2	4	[(2-1)/150]/(2-1) = 0.0066666667
3	3	[(5-3)/150]/(2-1) = 0.0133333333
4	2	[(50-6)/150]/(2-1) = 0.2933333333
5	1	[(100-50)/150}/(2-1) = 0.3333333333

Posteriormente, aplica-se a validação cruzada onde o total de erros de cada validação é armazenado em uma lista que será percorrida encontrando-se a posição que possui o menor valor. A seguir, procura-se por esta posição na lista de α (Tabela 2). Encontrando-se este valor o mesmo será utilizado na fórmula (12), que resulta no melhor parâmetro de complexidade em relação ao erro obtido. Então, a partir do momento em que esse valor for menor que o α encontrado para cada sub-árvore ocorre a simplificação da árvore (poda).

As regras e a árvore de decisão geradas pelo CART na Shell Orion podem ser observados na Figura 3 e 4.

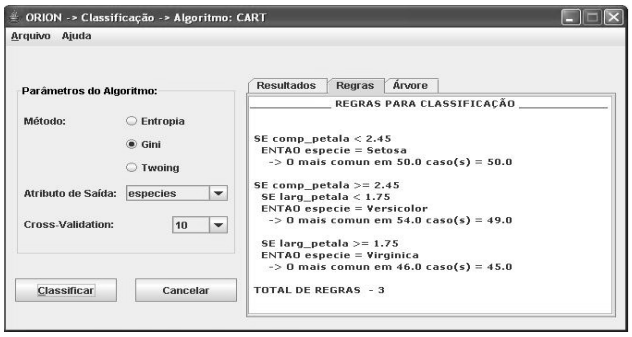

Figura 3. Regras geradas pelo algoritmo CART

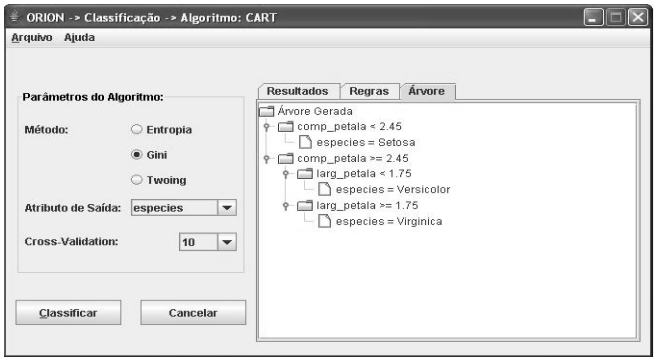

Figura 4. Árvore de decisão construída pelo algoritmo CART

4. Conclusão

O processo de data mining é fundamental não somente para a descoberta de novos conhecimentos, mas também para confirmação dos já existentes podendo proporcionar benefícios significativos às instituições no que se refere a tomada de decisão e vantagem estratégica.

A sua aplicação se dá por meio de tarefas e métodos, sendo que nesta pesquisa os estudos concentraram-se acerca do algoritmo CART para indução de árvores de decisão, possibilitando a classificação correta dos dados e não exigindo para isso transformações a fim de adaptá-los ao algoritmo. Além disso, torna as árvores menores em relação a outros métodos de indução já que implementa a simplificação, o que vem a auxiliar no entendimento das relações descobertas.

O algoritmo CART na Shell Orion Data Mining Engine apresentou desempenho satisfatório se comparado com outras ferramentas, pois classificou corretamente os dados, teve um percentual de erro de 4% e, apesar da diversidade de cálculos presentes na sua execução teve um tempo de processamento aceitável. O desempenho e o tempo de processamento foram considerados satisfatórios baseando-se nos resultados obtidos por meio das ferramentas WEKA 3.5.5 e CART 5.0, em simulações efetuadas utilizando-se a mesma base de dados.

Referências

Breiman, L. et al. (1984) “Classification and Regression Trees”, Chapman e Hall/CRC, New York.

Fayyad, U., Piatetsky-Shapiro, G. e Smyth, P. (1996) “From Data mining to Knowledge Discovery in Databases”, In: The American Association for Artificial Intelligence, AI Magazine, http://kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf.

Goldschmidt, R. e Passos, E. (2005) “Data Mining: um guia prático”, Elsevier, Rio de Janeiro.

Gonçalves, E. C. (2006) “Extração de Árvores de Decisão com a Ferramenta de Data Mining Weka**”,** Instituto Brasileiro de Geografia e Estatística, http://www.devmedia.com.br/articles/viewcomp.asp?comp=3388.

Han, J. e Kamber, M. (2001) “Data mining**:** concepts and techniques”, Morgan Kaufmann Publishers, San Francisco.

Hand, D., Mannila, H. e Smyth, P. (2001) “Principles of Data Mining**”,** The MIT Press, Massachusetts.

Oliveira, R. (2006) “Uso de Data Mining para obter perfis de clientes com maior lucratividade”, Monografia (Especialização em Gerenciamento em Banco de Dados), Universidade do Extremo Sul Catarinense, Criciúma.

Russell, S. J. e Norvig, P. (2004) “Inteligência artificial”, Elsevier, Rio de Janeiro.