INTRODUÇÃO A MINERAÇÃO DE DADOS UTILIZANDO O WEKA

Marcelo DAMASCENO(1)
(1) Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte/Campus Macau, Rua das Margaridas, 300, COHAB, Macau-RN, 59500-000, e-mail:mailto:marcelo.damasceno@gmail.com

RESUMO

A utilização de técnicas de Mineração de Dados vem sendo aplicadas diariamente nas empresas. Estas técnicas são utilizadas na identificação de informações relevantes em grandes volumes de dados. Existem diversos *softwares* no mercado que utilizam técnicas de mineração para obter seus resultados. Alguns destes *softwares* são pagos, inviabilizando sua aplicação em pequenas empresas ou mesmo por instituições de ensino. O Weka é uma suite que contém diversas técnicas de mineração de dados, totalmente gratuito por ser um software livre. Este minicurso têm como objetivo apresentar diversas técnicas de mineração de dados utilizando o Weka. As técnicas serão apresentadas e discutidas na resolução de problemas reais, utilizando somente ferramentas presentes na suite Weka.

Palavras-chave: mineração de dados, weka, descoberta de conhecimento em banco de dados

1. INTRODUÇÃO

A todo momento dados vêm sendo armazenados, formando grandes volumes de dados. Os dados armazenados contêm informações ocultas de grande relevância para o negócio. Devido ao grande volume de dados, a extração destas informações não é uma tarefa trivial. Existe uma grande necessidade de teorias e ferramentas para o auxílio dos analistas para a extração e análise de informações úteis. O conjunto destas teorias e ferramentas é pertencente a um processo conhecido como Descoberta de Conhecimento em Banco de Dados (DCBD).

A DCBD está preocupada com o desenvolvimento de métodos e técnicas para que os dados façam sentido. A DCBD está concentrada no mapeamento dos dados brutos (informações brutas) em modelos mais compactos (relatórios, gráficos), genéricos (identificação de modelos que descrevem os dados) ou úteis (modelos preditivos que estimem valores futuros) que os dados originais.

No núcleo deste processo, DCBD, se encontra a aplicação de técnicas para a identificação e extração de informações relevantes ocultas nos dados. Estas técnicas são conhecidas como mineração de dados por identificarem padrões e conhecimentos relevantes para o negócio, ou seja, minerar os dados a procura do "ouro" (informações relevantes).

WEKA é uma suíte de mineração de dados muito popular no meio acadêmico, desenvolvido utilizando a linguagem Java. Criada nas dependências da Universidade de Waikato, Nova Zelândia. Atualmente é mantida por uma comunidade de entusiastas por ser um software livre disponível sobre a licensa GPL.

Este minicurso abordará de forma teórica e prática, questões relacionadas a técnicas de mineração de dados, resolvendo problemas clássicos utilizando a suite de mineração Weka. Iremos resolver problemas de agrupamento, predição, regressão numérica e classificação.

2. PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

O processo de Descoberta de Conhecimento em Banco de Dados (DCBD) é um processo não trivial de identificação de padrões novos, válidos e potencialmente úteis (FAYYAD et al., 1996). Estes padrões estão ocultos nos dados e devem ser novos para o sistema, de preferência para o usuário, válidos em relação aos dados armazenados e as políticas do negócio, úteis para sua devida utilização nas tarefas para o qual foi requisitado.

Podemos definir dados como um conjunto de fatos sobre determinado assunto e padrões como uma linguagem ou um modelo que descreve um sub-conjunto deste fatos. Ou seja, identificar um padrão é ajustar um modelo aos dados ou identificar uma estrutura no conjunto de dados, descrevendo-os de forma genérica.

A DCBD é um processo iterativo e interativo composto de diversas etapas que envolvem a preparação dos dados, procura por padrões, avaliação e refinamento. O processo é interativo pois todas as etapas contém tarefas e decisões a serem feitas pelos usuários. Iterativo, pois todas as etapas estão conectadas. Caso a etapa de avaliação não seja satisfatória, de acordo com o processo, podemos voltar para a etapa de preparação dos dados. A Figura 1 representa o processo.

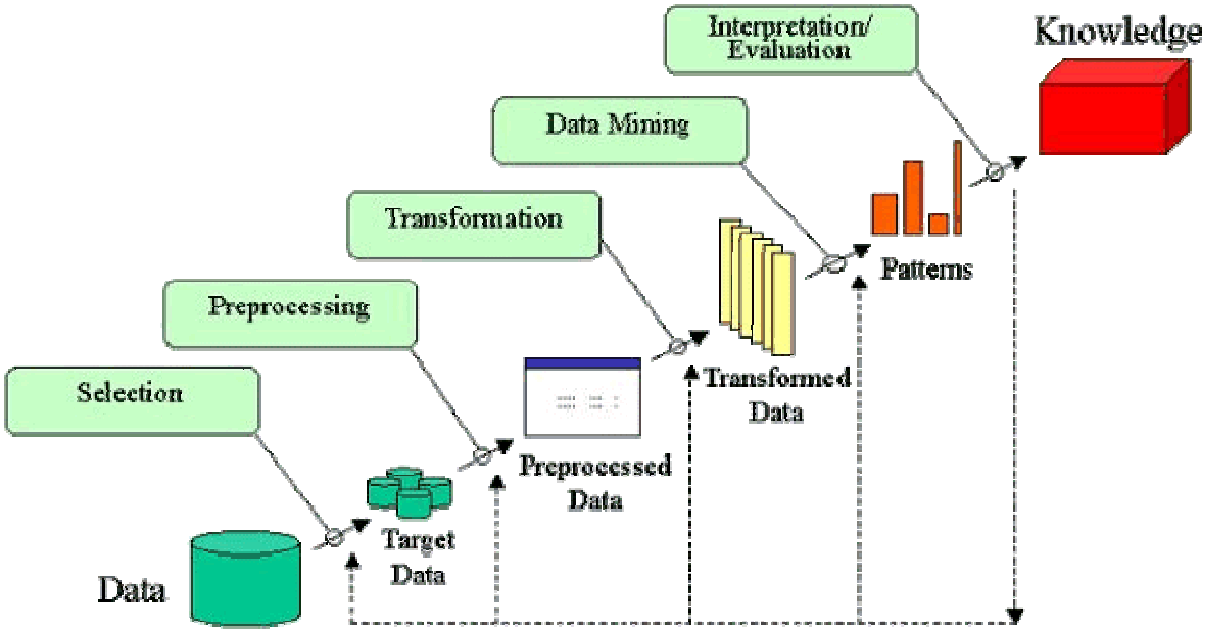

Figura 1 Etapas do DCBD (FAYYAD et al. 1996)

Primeiramente é necessário entender os desejos do usuário, isto é, catalogar as necessidades do ponto de vista do negócio e do usuário. Com os desejos catalogados, podemos definir o objetivo do DCBD. Somente poderemos definir o objetivo do processo de mineração quando os requisitos forem levantados, pois a escolha da técnica de mineração irá depender do objetivo a ser alcançado.

Após o levantamento dos requisitos ter sido realizado, deve-se criar o conjunto de dados no qual o processo irá trabalhar. Este conjunto de dados deve conter todas as informações necessárias para que os algoritmos de mineração possam alcançar seu objetivo. Essa etapa é conhecida como Seleção.

A segunda etapa é composta por tarefas de pré-processamento. Técnicas de pré-processamento são responsáveis pela remoção de ruídos (erros e exemplos fora do padrão), pela definição de estratégias para lidar com valores faltosos e pela formatação dos dados de acordo com os requisitos da ferramenta de mineração.

A terceira etapa, conhecida como Transformação, tem por objetivo localizar características úteis para representar os dados. Responsável também pela seleção dos melhores exemplos e atributos presentes no conjunto de dados.

Logo após os dados terem sido limpos e pré-processados, iremos aplicar as técnicas de mineração de dados para alcançar os objetivos definidos na primeira etapa. Os objetivos identificados podem ser descritos como tarefas de classificação, regressão, agrupamento, predição, etc. É necessário escolher qual algoritmo de mineração deve ser utilizado após a determinação de qual tarefa de mineração será executada. As técnicas são escolhidas de acordo com as características dos dados e da técnica e com os requisitos apresentados pelos

usuários. Algumas técnicas de mineração contêm parâmetros que são utilizados em seu funcionamento, também faz parte desta etapa encontrar os melhores parâmetros, para que o método possa ser o mais preciso e ágil possível.

Somente após todas essas tarefas terem sido realizadas, é hora da execução propriamente dita do algoritmo de mineração. O algoritmo irá procurar por padrões utilizando as suas estratégias, utilizando os dados informados.

A quinta etapa é interpretar e avaliar os padrões identificados. Este passo inclui visualizar os padrões extraídos ou os modelos que resumem a estrutura e as informações presentes nos dados. Além da visualização, são utilizadas medidas tanto técnicas quanto subjetivas para avaliar os padrões extraídos. As medidas técnicas são informações referentes a precisão, erro médio, erro quadrático e taxas de falsos positivos e falsos negativos. Medidas subjetivas são referentes a informações como utilidade, entendimento ou complexidade dos padrões extraídos.

Ao final de todo o processo teremos conhecimento em forma de padrões. Sendo assim, poderemos utilizar os padrões extraídos para os quais eles foram desejados. Os padrões podem ser utilizados sozinhos ou embutidos em outros sistemas.

3. MINERAÇÃO DE DADOS

A noção de encontrar padrões úteis em grandes volumes de dados pode ser conhecido por diversos nomes, tais como mineração de dados, extração de conhecimento, descoberta de informações, arqueologia de dados e processamento de padrões de dados. O termo mineração de dados é o mais usado por profissionais da computação, estatísticos e analistas de dados.

A fase de mineração de dados é uma fase do processo de Descoberta de Conhecimento em Banco de Dados (DCBD). Esta etapa é responsável pela aplicação dos algoritmos que são capazes de identificar e extrair padrões relevantes presente nos dados (HAN, 2001; WITTEN, 2000).

O processo DCBD é interdisciplinar, tanto em sua aplicação, quanto das suas fundamentações teóricas. O processo pode ser aplicado a qualquer problema de identificação de padrões em dados e contém fundamentação de diversas áreas como a banco de dados, inteligência artificial, estatística, probabilidade e visualização de dados.

3.1 Aprendizado

Toda técnica de mineração passa por um processo chamado de treinamento. A fase de treinamento tem este nome por ser um processo de apresentação dos dados processados para o algoritmo de mineração, cujo objetivo é identificar, ou seja, “aprender” as características ou padrões úteis ao objetivo do processo de descoberta de conhecimento.

Utilizam-se dados processados para a realização do aprendizado. Após o aprendizado ter sido realizado, é aplicada uma avaliação, onde podemos verificar medidas estatísticas dos resultados alcançados. A avaliação do algoritmo treinado deve ser realizada utilizando dados não vistos pelo algoritmo, ou seja, inéditos. A utilização de dados inéditos fornecerá medidas realistas sobre o desempenho do algoritmo, pois os mesmos serão feitos a partir de dados não vistos na fase de treinamento. A fase de avaliação será realizada de forma correta caso a divisão do conjunto de dados seja realizada. O conjunto deve ser dividido em dados de treinamento e de teste. Ás vezes, é necessário dividir o conjunto de dados em 3 diferentes conjuntos: treinamento, validação e teste. O conjunto de validação é utilizado para ajustar valores dos parâmetros de alguns algoritmos e ao mesmo tempo uma boa generalização. Quando o conjunto de dados é dividido em dois, geralmente a divisão é de 70% do conjunto para o conjunto de treinamento e 30% para o conjunto de testes. Já, quando o conjunto será divido em 3 (três), usa-se a proporção 70% para treinamento, 20% para validação e 10% para testes.

3.2 Aprendizagem Supervisionada

Aprendizagem supervisionada é aquela que utiliza dados com a classe especificada, ou seja, a instância contém um atributo classe que especifica a qual classe ela pertence. Existem diversos métodos de mineração que trabalham com este tipo de aprendizado. Geralmente são técnicas preditivas, pois tentam predizer qual a classe de uma instância não vista, baseado nos exemplos utilizados em seu treinamento.

Através de classificadores, é possível determinar o valor de um atributo a partir de um subconjunto de atributos presente no conjunto de dados. Por exemplo, em uma locadora deseja-se saber o perfil de clientes que alugam mais de 3 filmes ou prever quais clientes aceitariam aderir a um programa de mensalidade para o aluguel de filmes. A forma de representação mais comum para estes exemplos são árvores de decisão ou regras, pois através destas, é possível visualizar regras com precedente e conseqüente. Por exemplo, cliente com renda superior a 3 salários, carro próprio, fã de ação e terror então adesão ao programa de mensalidade. Algoritmos como Id3, C45, J48, ADTree, UserClassifier, PredictionNode, Splitter, ClassifierTree, M5Primer, Prism, Part, OneR são classificadores presentes no Weka que produzem regras ou árvores de decisão. Classificadores também utilizam modelos matemáticos e podem não produzir regras ou árvores para a visualização de seus resultados. Exemplos destes classificadores são SMO e Redes Neurais.

Classificadores são utilizados quando se deseja predizer classes não numéricas. É dado o nome de regressão numérica para tarefas onde se deseja utilizar classes numéricas. Exemplos, aprendizado de uma função linear ou não linear, prever o volume de vendas de um determinado produto em um supermercado. Tarefas de regressão numérica são mais complexas, pois a classe é contínua, ou seja, o espaço do contradomínio da função pode ser gigantesco.

3.3 Aprendizagem Não Supervisionada

Aprendizagem não supervisionada é aquela que utiliza instâncias sem a determinação do atributo classe. Este tipo de aprendizado é utilizado geralmente para análise exploratória dos dados, utilizando técnicas de agrupamento ou regras de associação.

Técnicas de associação visam encontrar regras que procuram associações entre atributos presentes no conjunto de dados. Técnicas deste tipo são capazes de gerar regras do tipo: compra(pneu), compra(óleo) então compra(perfume para ambiente) com 70% de confiança. O Apriori é o algoritmo presente na suite Weka que é capaz de realizar tarefas de associação. Este algoritmo gera regras de associação utilizando no antecedente e no precedente das regras atributos presente no conjunto de dados.

Agrupamentos têm como objetivo relacionar instâncias com características em comuns. Lembrando que o agrupamento é uma técnica que utiliza o aprendizado não supervisionado, ou seja, não utiliza no processamento o atributo classe. A partir da definição de uma métrica de similaridade, os dados são agrupados, dando a possibilidade de encontrar relações interessantes entre as instâncias. Assim, o cliente do conhecimento gerado pode aplicar uma determinada ação em um subconjunto de instâncias presente nos dados. A suite Weka possui os algoritmos Cobweb e SimpleKMeans e EM para tarefas de agrupamento.

3.4 UCI

UCI é um repositório de dados que pode ser utilizado em tarefas de mineração de dados. Atualmente o repositório contém 194 conjuntos de dados, para atividades como classificação, regressão e agrupamento. O site organiza os conjuntos de dados em diferentes categorias, como tipos de atributo e dados, área de aplicação, número de atributos e instâncias e formato do arquivo.

Infelizmente o UCI não fornece os conjuntos de dados no formato padrão do WEKA, i.e, ARFF, mas isto pode ser contornado através do acesso ao site de conjunto de dados do WEKA (http://www.cs.waikato.ac.nz/~ml/weka/index_datasets.html). Tal site contém diversos conjuntos de dados, além dos presentes no UCI.

4. WEKA

A suite Weka (*Waikato Enviroment for Knowledge Analysis*) é formado por um conjunto de implementações de algoritmos de diversas técnicas de Mineração de Dados (UNIVERSITY OF WAIKATO, 2010).

O Weka está implementado na linguagem Java, que tem como principal característica a sua portabilidade, desta forma é possível utilizá-la em diferentes sistemas operacionais, além de aproveitar os principais benefícios da orientação a objetos. O WEKA é um software livre, ou seja, está sob domínio da licença GPL e está disponível em http://www.cs.waikato.ac.nz/ml/weka.

Alguns métodos implementados no WEKA:

- **Métodos de classificação**
  - Árvore de decisão induzida
  - Regras de aprendizagem
  - Naive Bayes
  - Tabelas de decisão
  - Regressão local de pesos
  - Aprendizado baseado em instância
  - Regressão lógica
  - Perceptron
  - Perceptron multicamada
  - Comitê de perceptrons
  - SVM
- Métodos para predição numérica
  - Regressão linear
  - Geradores de árvores modelo
  - Regressão local de pesos
  - Aprendizado baseado em instância
  - Tabela de decisão
  - Perceptron multicamadas
- Métodos de Agrupamento
  - EM
  - Cobweb
  - SimpleKMeans
  - DBScan
  - CLOPE
- Métodos de Associação
  - Apriori
  - FPGrowth
  - PredictiveApriori
  - Tertius

4.1 Arquivo ARFF

Para a aplicação de técnicas de mineração de dados necessitamos que os dados a serem utilizados estejam de forma organizada. Estes arquivos podem estar em alguma estrutura de dado, planilha ou banco de dados.

O WEKA possui um formato para a organização dos dados, seu nome é ARFF. Neste arquivo devem estar presentes uma série de informações, dentre elas: domínio do atributo, valores que os atributos podem representar e atributo classe. O arquivo ARFF é dividido em duas partes, a primeira contém uma lista de todos os atributos, onde se deve definir o tipo do atributo e/ou os valores que ele pode representar. Os valores devem estar entre chaves ({}) separados por vírgulas. A segunda é composta pelas instâncias presentes nos dados, os atributos de cada instância devem ser separados por vírgula, e aqueles que não contêm valor, o valor deve ser representados pelo caractere '?'.

As informações presentes no arquivo arff são especificadas utilizando marcações. Por exemplo, o nome do conjunto de dados é especificado através da marcação @relation, @atribute para os atributos e os dados em sim são definidos através da marcação @data.

4.2 Instalação, Configuração e Utilização

A suite Weka pode ser adquirida através do site http://www.cs.waikato.ac.nz/ml/weka/. No referido site, existe uma seção de download na qual é possível realizar diferentes versões (*stable*, *book* e *developer*) para diferentes plataformas (Windows, Linux e MacOS). Recomenda-se a utilização da versão *stable*, pois ela contém *bugs* resolvidos e vasta documentação.

Após a instalação da suite, não é necessário fazer nenhuma configuração adicional para a sua execução. Existem apenas configurações adicionais referentes a utilização de algoritmos específicos e acesso a banco de dados através de JDBC.

O Weka pode ser utilizado de três diferentes formas: interface gráfica, linha de comando e através de sua API. A interface gráfica fornece as diversas ferramentas para seus usuários através de janelas e seus elementos. A linha de comando é um meio utilizado para dar mais agilidade a processos repetitivos e acesso direto a funcionalidades que teriam mais passos a serem executados, caso fossem acessados via interface gráfica. A opção de acesso via API é utilizada por desenvolvedores de software por fornecer um meio prático para o uso das funcionalidades implementadas no Weka. Iremos exemplificar os diferentes problemas utilizando a interface gráfica, especificamente a ferramenta *Explorer*.

5. PROBLEMAS RESOLVIDOS

Esta seção irá focar na resolução de problemas utilizando a suite Weka. Será exemplificado problemas de classificação, regressão, agrupamento e associação.

5.1 Classificação

Um problema clássico de classificação é classificar a flor iris em 3 espécies utilizando o comprimento e a largura da sépala e da pétala. O conjunto de dados contém 150 instâncias de 3 espécies (setosa, virgínica e versicolor). Para resolver este problema iremos utilizar o algoritmo J48 que gera uma árvore de decisão. Assim, poderemos entender a relação entre os valores dos atributos e a espécie da flor.

Para a realização da tarefa, deve-se iniciar o Weka e abrir o arquivo ARFF contendo o conjunto de dados Iris. Na Figura 2, vemos uma representação da tela com o conjunto de dados já aberto. Na mesmo é possível ver todos os atributos, inclusive a classe. Além dos atributos, é possível visualizar alguns dados estatísticos como o valor máximo e mínimo, média e desvio padrão.

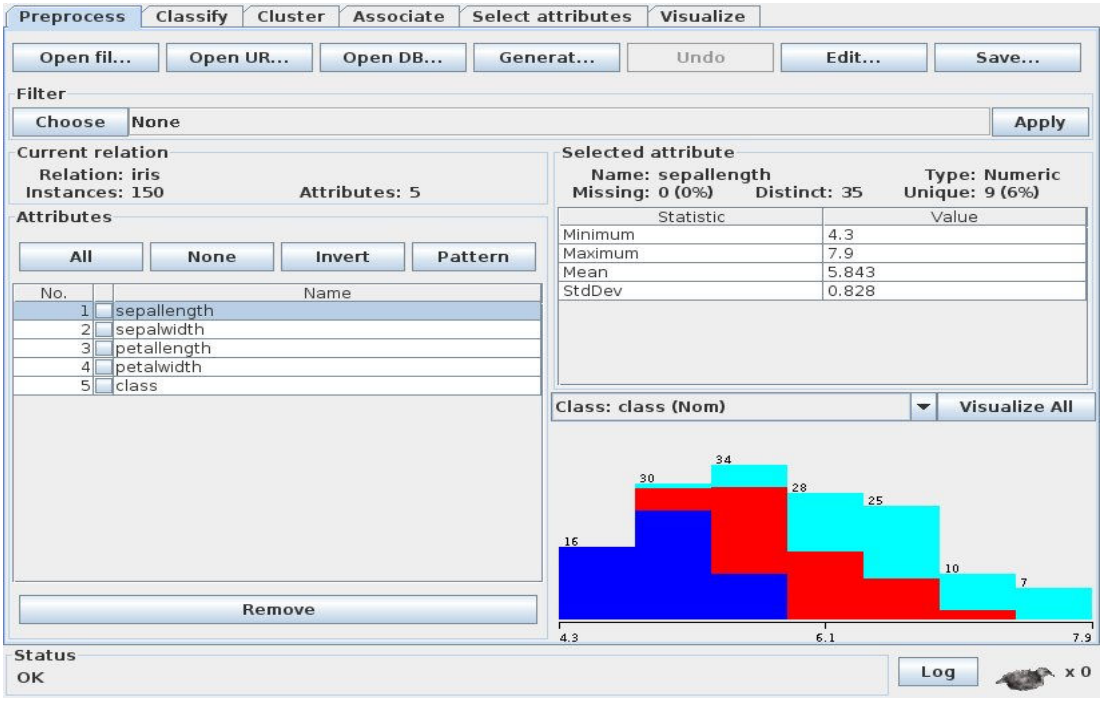

Figura 2Atributos do conjunto iris e algumas estatísticas

Após o carregamento do conjunto de dados, iremos executar o algoritmo J48. Iremos pular as etapas do processo de DCBD, pois o conjunto de dados já está pré-processado e transformado para o formato que o algoritmo requer. A Figura 3 exibe a aba de classificadores (*Classify*), com o algoritmo J48 selecionado, treinamento utilizando 70% do conjunto de dados e a árvore de decisão utilizando a estrutura de representação do Weka.

Podemos extrair 5 regras através da árvore representada na Figura 2. São:

- LarguraPétala <= 0.6 → Setosa
- LarguraPétala > 1.7 → Virginica
- LarguraPétala <= 1.7 e ComprimentoPétala <= 4.9 → Versicolor
- ComprimentoPétala > 4.9 e LarguraPétala <= 1.5 → Virginica
- ComprimentoPétala > 4.9 e LarguraPétala > 1.5 → Versicolor

Além das regras, podemos perceber que as medidas relativas a sépala não foram utilizadas para a determinação da espécie da flor. Sendo assim, as medidas da sépalas são irrelevantes para a tarefa de classificação.

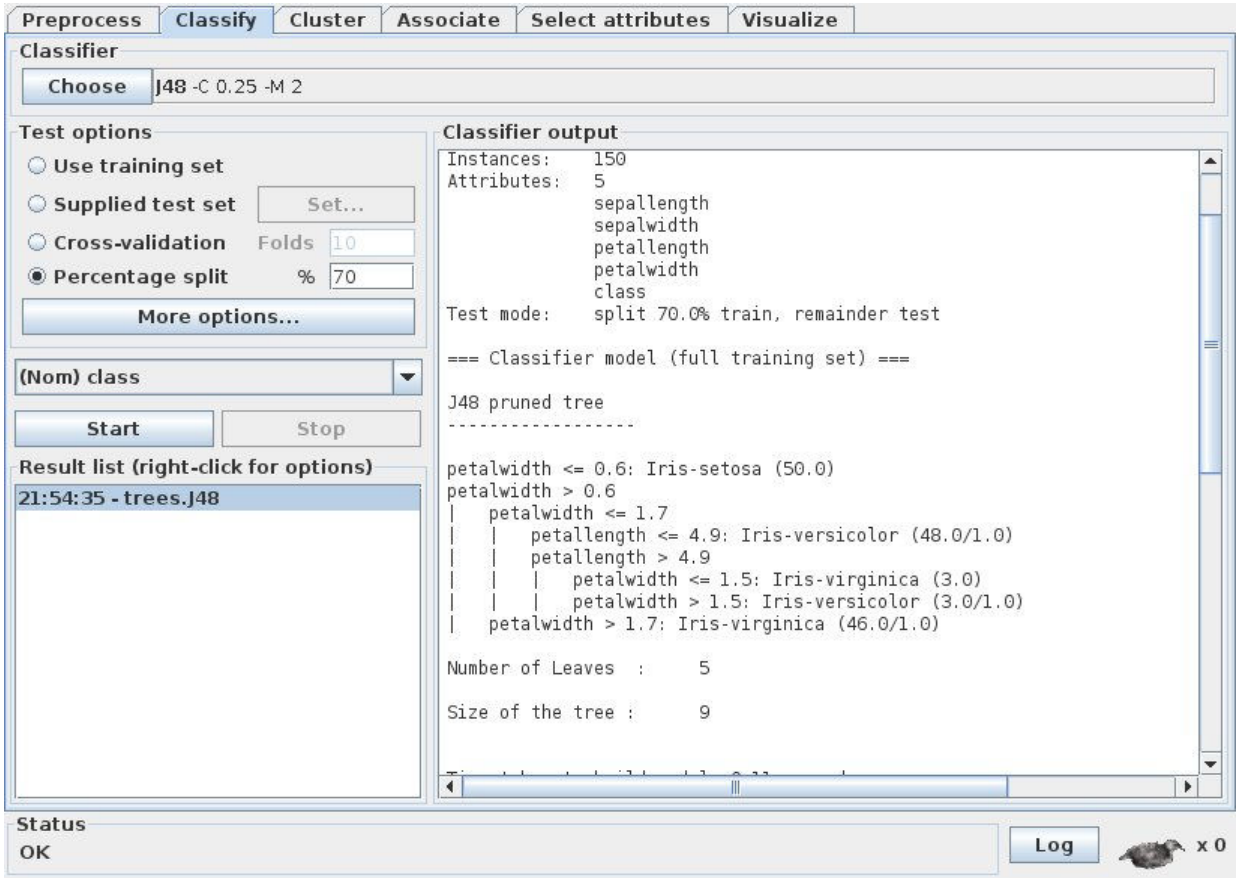

Figura 3Árvore de decisão utilizando o algoritmo J48

5.2 Regressão Numérica

Para exemplificar a utilização de algoritmos de regressão numérica, iremos utilizar um exemplo de um proprietário de uma casa que deseja saber o preço de sua propriedade. Para isso, ele pesquisou 5 variáveis de imóveis próximos ao seu: tamanho do imóvel e do terreno, quantidade de quartos, quartos com granito, banheiros e o preço do imóvel.

Para se saber qual o preço do imóvel do proprietário, iremos utilizar o algoritmo de regressão linear. A Figura 4 exibe os parâmetros para a execução do algoritmo e o resultado. Os parâmetros foram os mesmos utilizados na tarefa de classificação.

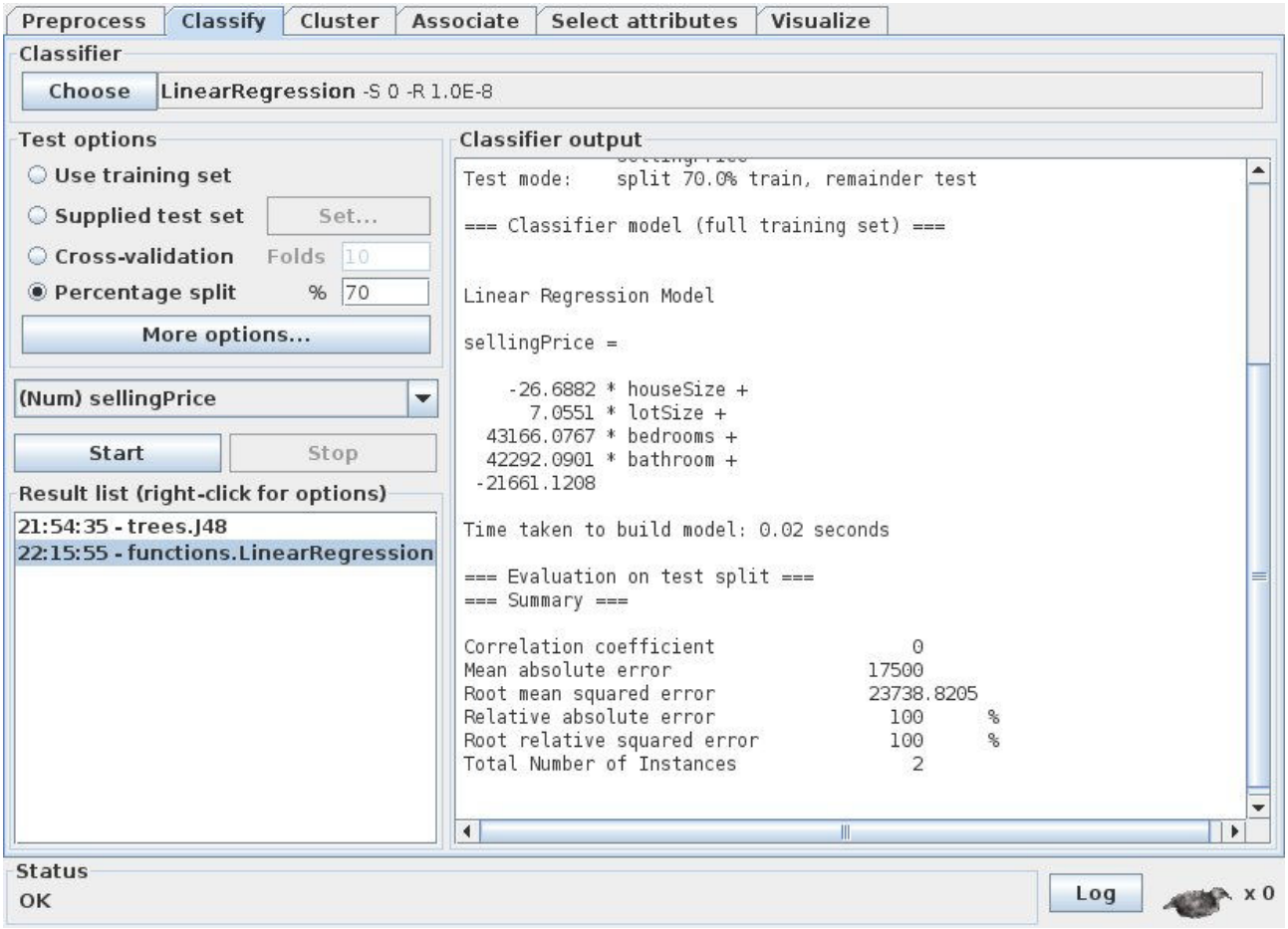

Figura 4Parâmetros da regressão linear e equação obtida

O algoritmo gerou tal equação:

Preço de Venda = -26.6882 * houseSize + 7.0551 * lotSize + 43166.0767 * bedrooms + 42292.0901 * bathroom - 21661.1208.

Assim, através da equação, o proprietário poderá propor um preço de acordo com o padrão observado nas redondezas de seu imóvel. Podemos interpretar a partir da equação que a presença de granito nos quartos não é importante. O tamanho da casa influi negativamente no preço do imóvel. Os quartos e os banheiros são mais importantes que o tamanho do lote, pois seus pesos são maiores que o peso referente ao lote.

5.3 Agrupamento

Como dito anteriormente, a técnica de agrupamento é destinada a análise exploratória dos dados. O agrupamento agrupa as instâncias de acordo com seus atributos utilizando medidas de similaridade. Iremos utilizar o conjunto de dados Iris, junto com o algoritmo CobWeb para exemplificar a aplicação de uma técnica de agrupamento utilizando o Weka.

Iremos verificar a acurácia do algoritmo em subdividir o conjunto de dados em um número correto de classes e a definição de cada classe a sua respectiva instância. Para isso iremos utilizar o conjunto Iris, sabendo o número de classes. Deve-se selecionar o algoritmo CobWeb, a partir da aba *Cluster,* com a opção *percentage split* (percentagem da divisão) com o valor 70 e a opção *Store to clusters evaluation* (Armazenar agrupamentos para visualização) marcada. A Figura 5 exibe o resultado da execução do algoritmo.

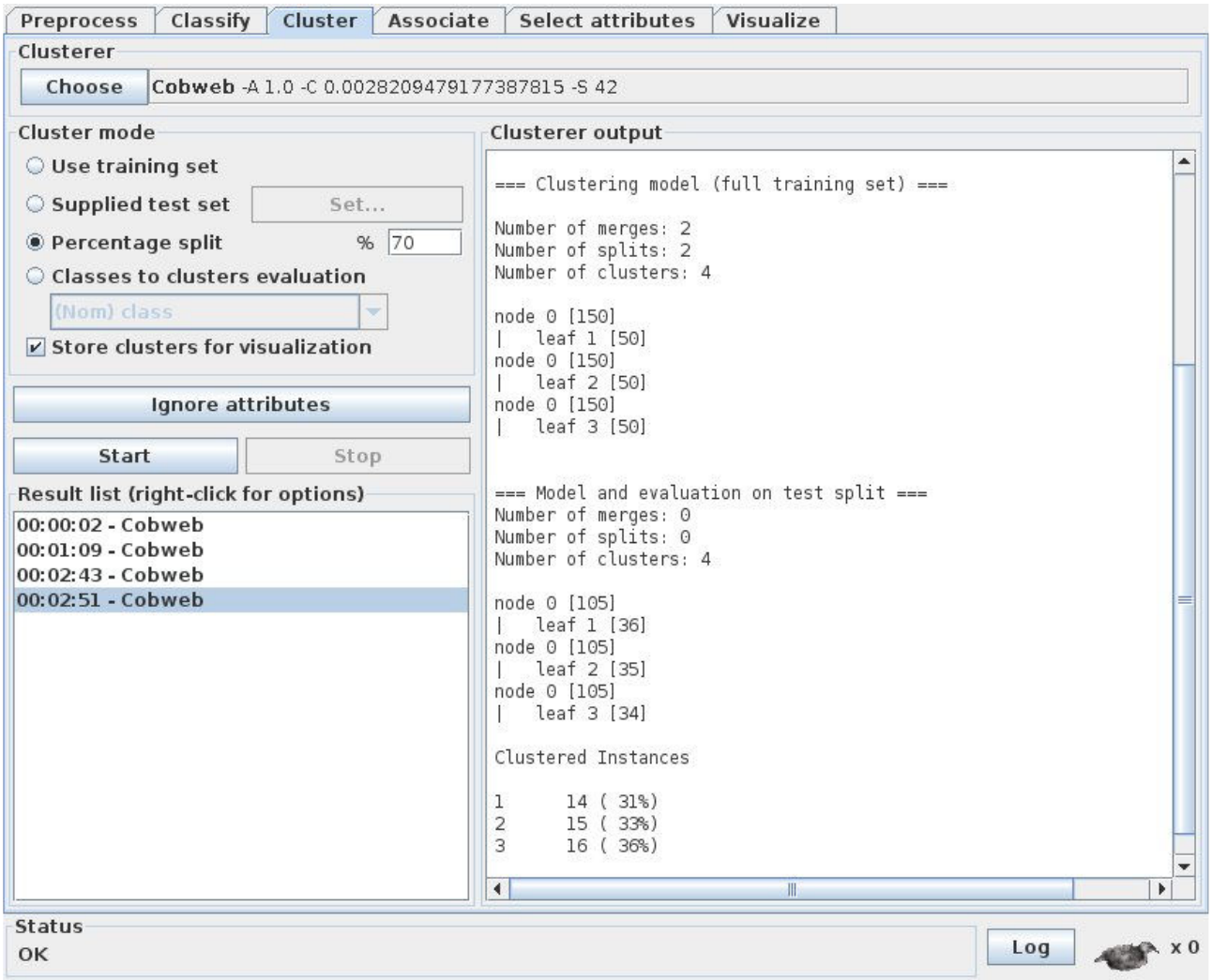

Figura 5: Resultado da execução do algoritmo CobWeb

Através da Figura 5 podemos visualizar que o algoritmo dividiu o conjunto de dados em 4 grupos, sendo que um grupo contém nenhuma instância. Cada grupo ficou exatamento com 50 instâncias, número correto de instâncias; podemos comparar este resultado com o conjunto de dados.

Pode-se visualizar o agrupamento com mais detalhes através de gráficos. Os gráficos permitem a observação dos atributos, classes e grupos das instâncias utilizadas. A Figura 6 representa um gráfico, cuja abscissa representa o número da instância, a ordenada, a classe que a instância representa e as cores representam o grupo que determinada instância pertence. Pode-se perceber ainda, que todas as instâncias foram corretamente agrupadas, pois as pertencentes a cada classe foram agrupadas em um mesmo grupo.

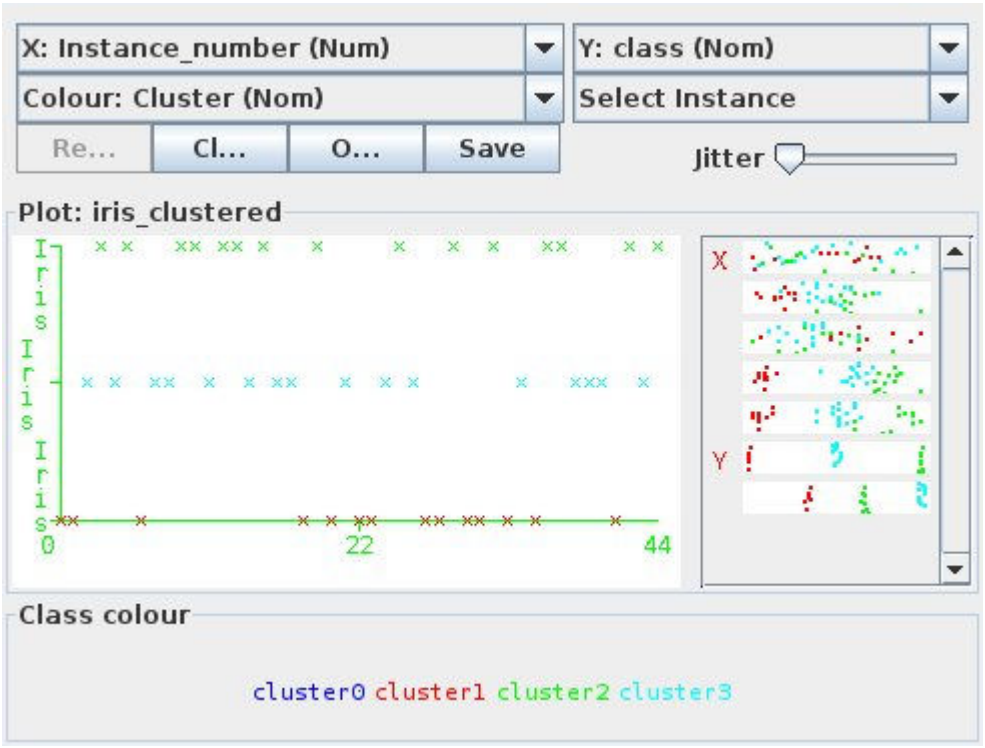

Figura 6: Gráfico do agrupamento do conjunto Iris

Pode-se também visualizar os grupos através de uma árvore. A árvore desta execução pode ser observada através da Figura 7. Também é possível visualizar detalhes sobre as instâncias presentes em cada grupo. A Figura 8 representa todas as instâncias pertencentes ao grupo 1. É possível saber detalhes sobre uma instância em específico, para isso é necessário apenas selecionar tal opção. A Figura 8 representa os detalhes de uma instância. É possível visualizar todos os valores desta instância.

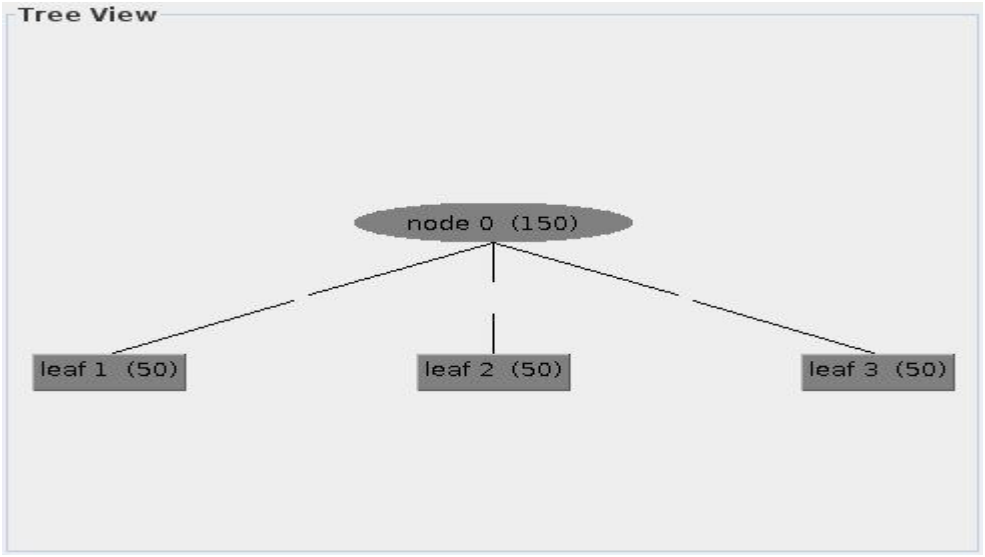

Figura 7: Árvore de agrupamento do conjunto Iris

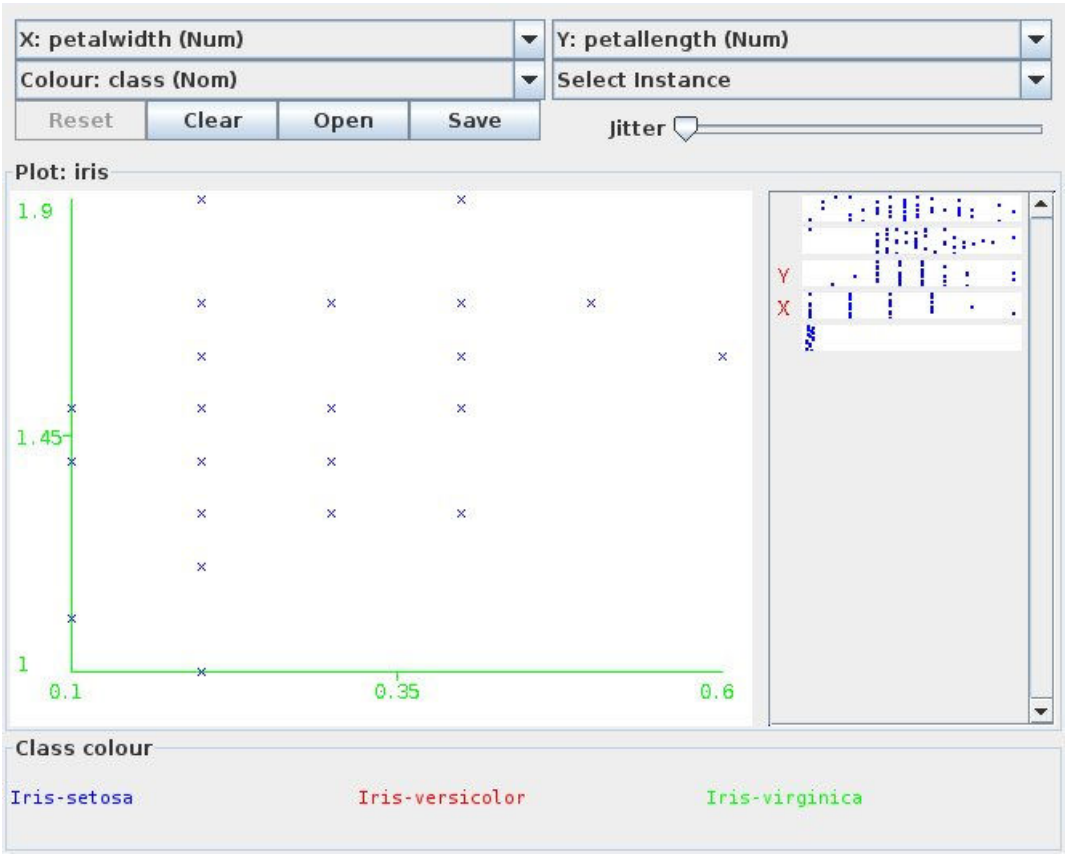

Figura 8: Gráfico com as instâncias pertencentes ao grupo 1

Figura 9: Valores dos atributos da instância 42

5.4 Associação

A técnica de associação é uma técnica exploratória (AGRAWAL, 1994). A associação gera regras que descrevem os padrões mais relevantes presentes nos dados. As regras são compostas por precedentes e conseqüentes, ou seja, a regra contém no precedente um subconjunto de atributos e seus valores e no conseqüente um subconjunto de atributos que decorrem do precedente. Por exemplo, quem compra ovos, manteiga, farinha de trigo e chocolate também compra fermento. As regras servem para diversas utilidades, dentre elas a descoberta de padrões de comportamento, aplicações em marketing e desenvolvimento de novos produtos.

Iremos utilizar o algoritmo Apriori implementado na suite para exemplificar a aplicação da associação no Weka. O conjunto de dados que iremos utilizar será o supermercado.arff. Este conjunto de dados descreve um conjunto de compras de clientes em um supermercado. Cada instância é uma compra e caso o cliente compre certo atributo, o valor para ele será 't'. O conjunto é composto por 217 atributos e 4627 instâncias.

Para utilizar o algoritmo Apriori, necessitamos primeiramente carregar o arquivo supermercado.arff. Após o arquivo ter sido aberto, iremos à aba *Association.* A Figura 10 representa a aba *Association* com o algoritmo Apriori selecionado. Além da seleção, é possível visualizar também algumas regras.

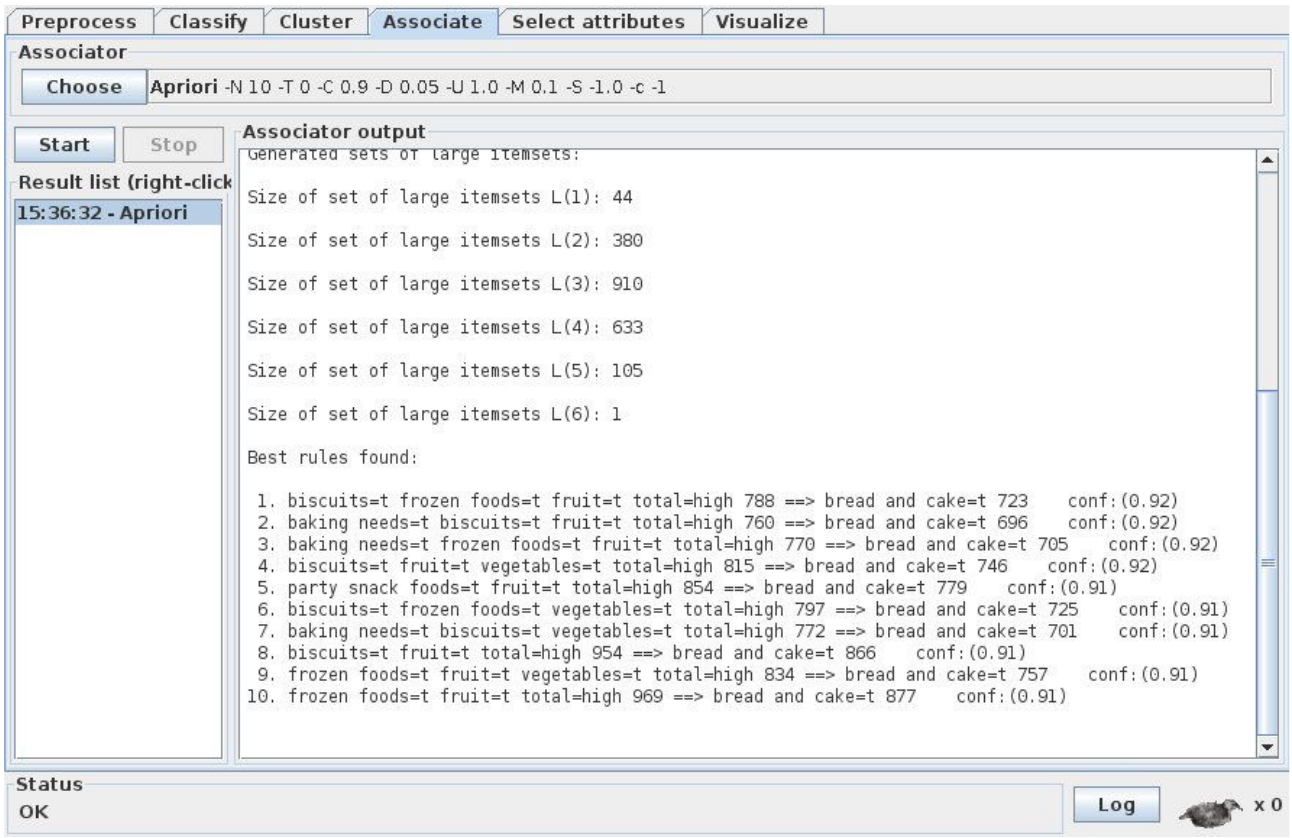

Figura 10: Aba associação com algumas regras geradas

A suite Weka retorna as 10 melhores regras, número que pode ser alterado nas configurações do algoritmo. O número após o precedente significa o número de instâncias que a regra cobre; já o número após o conseqüente significa o número de instâncias preditas corretamente. O valor após a palavra conf é o valor da confidência da regra. A confiança significa o percentual de ocorrência da regra. A confiança é calculada a partir da Equação: $\text{conf}(A \rightarrow B) = \frac{quantidadedeinstânciascontendoAquantoB}{quantidadedeinstânciascontendoA}$.

A Figura 9 apresenta 10 regras:

- Se o cliente comprou biscoitos, comida congelada, frutas e o total foi alto então ele comprou pão e bolo. 92% de confiança;
- Se o cliente comprou utensílios de cozimento, frutas e o total foi alto então ele comprou pão e bolo. 92% de confiança;
- Se o cliente comprou utensílios de cozimento, comida congelada, frutas e o total foi alto então ele comprou pão e bolo. 92% de confiança;
- Se o cliente comprou biscoito, frutas, vegetais e o total foi alto então ele comprou pão e bolo. 92% de confiança;
- Se o cliente comprou salgadinhos, frutas e o total foi alto então ele comprou pão e bolo. 91% de confiança;
- Se o cliente comprou biscoitos, comida congelada, vegetais e o total foi alto então ele comprou pão e bolo. 91% de confiança;

- Se o cliente comprou utensílios de cozimento, biscoitos, vegetais e o total foi alto então ele comprou pão e bolo. 91% de confiança;
- Se o cliente comprou biscoitos, frutas e o total foi alto então ele comprou pão e bolo. 91% de confiança;
- Se o cliente comprou comida congelada, frutas, vegetais e o total foi alto então ele comprou pão e bolo. 91% de confiança;
- Se o cliente comprou comida congelada, frutas e o total foi alto então ele comprou pão e bolo. 91% de confiança;

6. CONCLUSÕES

Este documento é apenas uma introdução a mineração de dados e ao seu uso utilizando a suite Weka. A suite Weka tem muito mais potencialidades do que as apresentadas neste minicurso. Deve-se adquirir experiência e maior conhecimento da ferramenta com seu uso e na resolução de problemas reais.

A mineração de dados e conseqüentemente a DCBD, possuem uma vasta aplicação, tanto comercial quanto acadêmica. Este minicurso fornece apenas orientações iniciais nesta área e apresenta algumas técnicas utilizando a suite Weka.

O avanço da tecnologia e conseqüentemente do volume de dados propicia o aumento da área de atuação de um analista de dados. O analista tem em suas mãos um grande mercado consumidor, que pode fornecer aos seus clientes soluções fáceis, ágeis e inteligente. A leitura e o aprendizado contínuo são peças chaves para um profissional de mineração de dados de sucesso.

REFERÊNCIAS

AGRAWAL, R.; SRIKANT, R. **Fast algorithms for mining association rules in large databases**. Proceedings of the International Conference on Very Large Databases, Santiago, Chile, 1994.

FAYYAD, U. M.; PIATESKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery: An Overview**. In: Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. Morgan Kaufmann, 2001.

UNIVERSITY OF WAIKATO. **Weka 3 – Machine Learning Software in Java**. Disponível no site da University of Waikato (2010). URL: http://www.cs.waikato.ac.nz/ml/weka

WITTEN, I.; FRANK, E. **Data Mining – Practical Machine Learning Tools**. Morgan Kaufmann, 2000.