MATHEUS BATISTA FURLAN

ALGORITMOS E TÉCNICAS PARA MINERAÇÃO DE DADOS

Assis
2018

MATHEUS BATISTA FURLAN

ALGORITMOS E TÉCNICAS PARA MINERAÇÃO DE DADOS

Projeto de pesquisa apresentado ao curso de do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando (a): Matheus Batista Furlan
Orientador (a): Prof. Dr. Alex Sandro Romeo de Souza Poletto

Assis
2018

FICHA CATALOGRÁFICA

F985a FURLAN, Matheus Batista
Algoritmos e técnicas para mineração de dados/ Matheus Batista Furlan. – Assis, 2018.
51p.

Trabalho de conclusão do curso (Ciência da Computação). – Fundação Educacional do Município de Assis-FEMA

Orientador: Dr. Alex Sandro Romeo de Souza Poletto

1.Algoritmos. 2.Dados. 3.Weka.

CDD 005.131

RESUMO

A finalidade desse trabalho é produzir um estudo relacionado as técnicas e algoritmos utilizados para mineração de dados e emprega-los em uma base de dados não existente. A ferramenta escolhida para tal é o WEKA, software especializado em Machine Learning e mineração de dados e que possui uma grande coleção de algoritmos voltados para tarefa de Data Mining. Esse trabalho pode ser, basicamente, dividido em duas fases. A primeira é um estudo exploratório sobre os conceitos de Descobrimento de Conhecimento em Banco de Dados e sobre a Mineração de Dados e suas técnicas e tarefas. Na segunda fase irá ser feito aplicação de alguns algoritmos disponibilizados pela ferramenta WEKA em uma base de dados demonstrando, assim, o processo que é realizado.

Palavras-chave: Mineração; Dados; WEKA; Técnicas; Algoritmos.

ABSTRACT

The purpose of this work is to produce a study related to the techniques and algorithms used for data mining and employs them in a non-existent database. The tool chosen for this is WEKA, a software that specializes in Machine Learning and data mining and has a large collection of data mining algorithms. This work can be basically divided into two phases. The first is an exploratory study on the concepts of Knowledge Discovery in Database and on Data Mining and its techniques and tasks. In the second phase will be made application of some algorithms provided by the WEKA tool in a database demonstrating the process that is performed.

Keywords: Mining; Data; WEKA; Techniques; Algorithms.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

Diversas áreas da ciência, governo e empresarial têm produzido um elevado volume de dados, a tal ponto, que se tornou inviável a habitual habilidade de interpretação e exploração desses dados, tornando preciso a busca de novas ferramentas e formas de análise automática e inteligente de bancos de dados (FAYYAD et al., 1996).

Essa abundante e acessível quantidade de dados e a urgência de converter esses dados em conhecimento útil tornou a mineração de dados um assunto importante no setor da informação e da sociedade. As informações e conhecimentos conseguidos são empregados em várias áreas como na análise de mercado, detecção de fraude e retenção de clientes, controle de produção e exploração científica (HAN; KAMBER, 2006).

Existe uma estrutura maior conhecida como Knowledge Discovery in Databases (KDD), em português descoberta de conhecimento em banco de dados, que abrange um processo complexo, desde a preparação dos dados até a modelagem de conhecimento, sendo a mineração de dados a principal etapa (SRIVASTAVA, 2014).

A mineração de dados, conhecida por Data Mining, é um campo multidisciplinar que envolve as noções sobre análise estatística de dados, aprendizagem de máquina, reconhecimento de padrões e visualização de dados (CABENA et al., 1998).

O seu principal objetivo é a demanda de conhecimentos relevantes em grandes bancos de dados. É um empenho de auxílio entre máquinas e humanos. Os humanos são incumbidos de arquitetar bancos de dados, relatar problemas e estabelecer metas. Os computadores analisam os dados e buscam critérios que se juntam com os objetivos estabelecidos pelos humanos (WEIS; INDURKHYA, 1999).

Existem diversos algoritmos na mineração de dados que são responsáveis por definir e determinar a melhor tomada de decisão. A ferramenta WEKA tem a capacidade de realizar diversas tarefas para mineração de dados, como pré-processamento de dados, seleção de atributos, classificação, agrupamento e melhora a descoberta de conhecimento usando vários meta classificadores (SRIVASTAVA, 2014).

1.1. OBJETIVOS

O objetivo desse trabalho é realizar um estudo relacionado as principais técnicas de mineração de dados e demonstrações dos seus respectivos algoritmos, além da demonstração da ferramenta Open-Source WEKA, na qual, serão evidenciados, na prática, os conceitos e algoritmos apresentados.

1.2. JUSTIFICATIVA

Fundamenta-se a elaboração desse tema, visto que, o conceito de mineração de dados está sendo cada vez mais aplicado em várias áreas da ciência e em empresas ao redor do mundo, devido ao aumento constante do volume de dados e a necessidade da extração de conhecimento desses dados, sendo o Data Mining parte essencial desse processo, visando o reconhecimento de padrões, previsão, tomada de decisão entre outros.

1.3. MOTIVAÇÃO

Atualmente devido a evolução da Web e informatização dos serviços são criados milhares de dados que são de extrema importância para sobrevivência de empresas e no avanço de pesquisas científicas, porém para isso é necessário o conhecimento sobre o conceito de mineração de dados e de que maneira os algoritmos são aplicados e conseguem encontrar respostas que antes não seriam possíveis.

1.4. PERSPECTIVAS DE CONTRIBUIÇÃO

Produção de pequenas aplicações utilizando a ferramenta WEKA onde será demonstrado os algoritmos e técnicas utilizados no processo de mineração de dados além do material teórico relacionado ao processo de busca de conhecimento em banco de dados e os próprios algoritmos e técnicas que foram aplicados.

1.5. METODOLOGIA DE PESQUISA

A metodologia utilizada para esse projeto de pesquisa será do tipo exploratória, tendo em vista buscar o material bibliográfico científico sobre a preocupação apresentada, bem como o desenvolvimento de estudo de caso focalizando mais na parte relacionado aos algoritmos e técnicas de mineração de dados.

Para atender melhor aos objetivos estabelecidos, definiu-se que, primeiramente deve-se obter o conhecimento necessário para o desenvolvimento deste trabalho buscando informações relacionados ao Knowledge Discovery in Databases (KDD), Data Mining, e Algoritmos e Técnicas de Mineração de Dados

Em seguida serão estudadas as características de cada uma destas partes, estruturas, componentes e formas de aplicação, com o intuito de obter os conceitos envolvidos nestas para que seja possível definir os elementos para a aplicação do projeto com a mineração de dados, que será realizado com auxílio da ferramenta WEKA. Com isso será possível juntar informações relativas para o trabalho, sendo possível realizar um bom embasamento teórico para auxiliar o trabalho de conclusão de curso.

1.6. RECURSOS NECESSÁRIOS

Será necessário apenas a utilização de um computador com o sistema operacional Windows e o software Open-Source chamando Weka.

1.7. ESTRUTURA DO TRABALHO

O trabalho será composto por:

- **Capítulo 1 – Introdução:** Neste capítulo, será apresando os conceitos de descobrimento de conhecimento em banco de dados e mineração de dados.

- **Capítulo 2 – Knowledge Discovery in Databases (KDD):** Neste capítulo, será apresentado mais aprofundado o KDD com foco nas suas etapas.
- **Capítulo 3 – Mineração de Dados:** Neste capítulo, serão discutidos as técnicas e algoritmos utilizados no Data Mining.
- **Capítulo 4 – Ferramenta Weka**: Neste capítulo, será apresentado a ferramenta que será utilizado para as aplicações de mineração de dados.
- **Capítulo 5 – Desenvolvimento das aplicações:** Neste capítulo, serão apresentadas as aplicações criadas com a ferramenta Weka utilizando os algoritmos e técnicas já explicados, visando demostrar o processo de criação.
- **Capítulo 6 – Conclusão:** Neste capítulo, serão discutidos os algoritmos e técnicas de Data Mining, assim como a utilização da ferramenta Weka para desenvolver esse tipo de aplicação.
- **Referências**

2. KNOWLEDGE DISCOVERY DATABASE

Descoberta de Conhecimento em Banco de Dados, conhecido originalmente como Knowledge Discovery in Database ou KDD, é o processo que tem como principal objetivo extrair conhecimento de uma ampla base de dados. Fayyad et al. (1996) diz: “Extração de Conhecimento de Base de dados é o processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados”. Segundo Rezende (2003) para um melhor entendimento é interessante analisar essas características separadamente:

- Dados: coleção de elementos gravados em um repositório.
- Padrões: indica uma concepção de um subconjunto de dados em alguma linguagem descritiva de princípios.
- Processo: são as etapas pelas quais o processo de extração de conhecimento passa, como preparação de dados, busca por padrões e avaliação do conhecimento.
- Válidos: os padrões encontrados devem seguir um determinado grau de convicção, devem corresponder funções e princípios que certificam que os exemplos cobertos e os eventos referentes ao padrão sejam admissíveis.
- Novos: um padrão deve prover novas informações sobre os dados. É capaz medir isso a partir de comparações entre as modificações realizadas e os dados anteriores.
- Úteis: os padrões descobertos devem ser anexados para serem aproveitados.
- Compreensíveis: os padrões devem ser representados em uma linguagem que possa ser entendida pelos usuários possibilitando uma análise mais profunda dos dados.
- Conhecimento: o conhecimento é definido de acordo com o seu escopo de aplicação, utilidade, originalidade e compreensão.

Segundo Lacerda & Souza (2004), a essência do descobrimento é tanto iterativa quanto interativa. A iteratividade é por causa do processo ser executado em etapas contínuas de forma que seja viável retomar as etapas antecedentes estabelecendo uma conexão entre elas. No processo o usuário é encarregado de tomar diversas

decisões durante o ciclo, na seleção dos algoritmos, na modelagem das informações e as metas a serem seguidas assegurando, assim, a interatividade.

Para Goldschmidt e Passos (2005), o problema a ser submetido ao processo KDD é identificado por três fundamentos: conjunto de dados, o especialista do domínio da aplicação e pelos objetivos da aplicação. As informações obtidas com suporte da aplicação dos recursos do problema que compreende o modelo de conhecimento encontrados durante a aplicação de KDD e o histórico das ações que foram ocasionalmente efetuados.

O KDD é definido como um processo composto por várias etapas operacionais. A Figura 1. Ilustra uma configuração resumida etapas operacionais realizadas em processos de KDD. A primeira etapa é o pré-processamento onde são feitas as funções de compreensão, a organização e o tratamento dos dados com a finalidade de preparar os dados para a etapa seguinte, a Mineração de Dados. Na etapa de Mineração de Dados é executada, de fato, uma busca por informações e conhecimentos importantes no ambiente da aplicação do KDD. Por fim o pós-processamento engloba o tratamento das informações e conhecimento conseguidos na Mineração de Dados, com o objetivo de proporcionar a avaliação do benefício do conhecimento descoberto, sendo nem sempre necessário (BRACHMAN & ANAND, 1996).

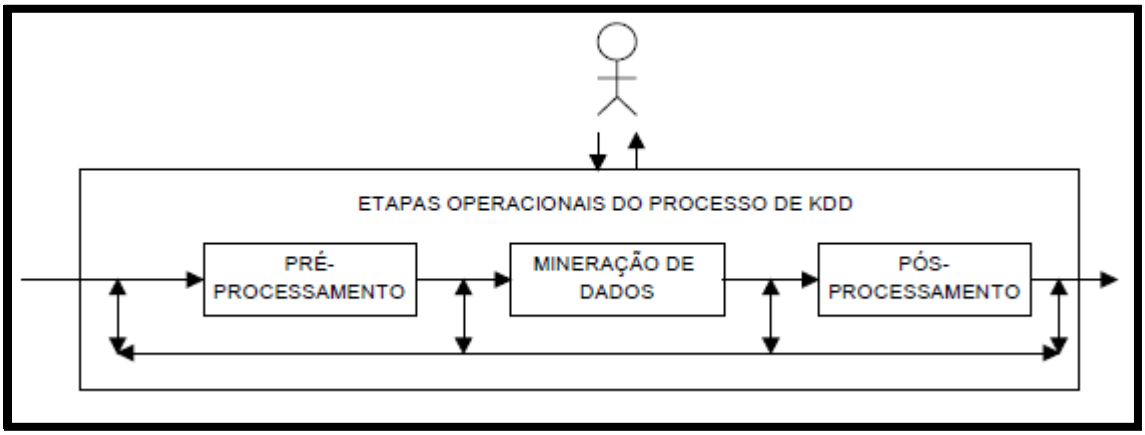


Figura 1: Etapas Operacionais do Processo KDD (In: BRACHMAN & ANAND, 1996)

2.1. ETAPAS OPERACIONAIS DO PROCESSO DE KDD

De acordo com o que já foi citado, o processo KDD se define por três etapas: O pré-processamento, Mineração de Dados e Pós-processamento. A fase de pré-processamento tem a finalidade de organizar os dados para algoritmos da fase seguinte, a de Mineração de Dados, e apresenta todas as funções associadas à captação, à organização e ao tratamento de dados.

2.1.1. PRÉ-PROCESSAMENTO

A fase do Pré-processamento engloba todas as funções relativas com a captação, a organização e o tratamento de dados, sendo essa etapa, responsável por preparar os dados para os algoritmos que serão utilizados na etapa de Mineração de Dados. As principais funções do pré-processamento são descritas da seguinte maneira:

- Seleção de Dados – essa função envolve, em resumo, a identificação dos dados existentes que devem, de fato, ser considerados durante o processo de KDD.
- Limpeza dos Dados – consiste no tratamento feito sobre os dados selecionados de maneira a garantir a qualidade dos fatos representados. As informações que estiverem inconsistentes, ausentes ou erradas devem ser arrumadas para que não prejudiquem a qualidade dos modelos de conhecimento que serão retirados ao final do processo de KDD.
- Codificação dos Dados – nessa etapa os dados devem ser codificados de forma Numérica – Categórica, transformando os valores reais em categoria ou intervalos; ou Categórica – Numérica, que representa numericamente valores de atributos categóricos, para que possam ser utilizados como entrada para os algoritmos de Mineração de Dados.
- Enriquecimento dos Dados – a atividade de enriquecimento se baseia em conseguir, de alguma forma, mais informações que possam tornar o processo de descobrimento mais “rico”. Por exemplo, realizar pesquisas para complementar os dados ou consultas em bases externas são exemplos de como conseguir esse enriquecimento.

2.1.2. MINERAÇÃO DE DADOS

A Mineração de Dados é considerada a principal etapa do processo de KDD, sendo vista por alguns autores como sinônimo do próprio processo de Descoberta de Conhecimento em Base de Dados. Nessa etapa são utilizados algoritmos e técnicas com objetivo de buscar conhecimento valioso em meio a tantos dados. Existem diversas técnicas e algoritmos de Mineração de Dados, que são escolhidas, levando em consideração, o tipo da tarefa em Mineração de Dados que é aplicada. No capítulo 3 será abordado, mas detalhadamente a etapa de Mineração de Dados, onde será descrito tanto as tarefas quanto os algoritmos e técnicas.

2.1.3. PÓS-PROCESSAMENTO

Essa etapa envolve o tratamento do conhecimento obtido na Mineração de Dados. Podem ser citados como exemplos de funções na etapa de Pós-processamento: Elaboração e organização do conhecimento obtido, simplificação do Modelo de Conhecimento, gráficos, diagramas ou relatórios demonstrativos. O objetivo é basicamente é facilitar a visualização do conhecimento adquirido para demonstração, sendo algumas fezes desnecessário esse tratamento (FAYYAD et al., 1996).

3. MINERAÇÃO DE DADOS

Data Mining ou Mineração de Dados são procedimentos para análise e exploração em amplos volumes de dados a partir de técnicas e algoritmos especializados que tem como objetivo de buscar padrões, previsões, associações, erros e assim por diante. A partir do conhecimento adquirido é possível realizar novas estratégias para o negócio, entender a necessidade e comportamento dos consumidores, prever o desempenho financeiro da organização, mitigação de riscos futuros entre outros, sendo que tais questões dificilmente seriam identificadas analisando os dados a olho "nu". A mineração de dados permite uma automatização no descobrimento de informações, porém ainda sim se faz necessário uma análise humana posterior sobre onde e de que maneira esse conhecimento pode ser aplicado.

Como visto no capítulo 2, o *Data Mining* faz parte de um processo maior conhecido como KDD, sendo ele a principal e mais importante etapa. Na literatura são encontrados diversos pareceres sobre a Mineração de Dados, sequem algumas definições:

"Mineração de Dados é um conjunto de técnicas que envolvem métodos matemáticos, algoritmos e heurísticas para descobrir padrões e regularidades em grandes conjuntos de dados" (POSSA et al, 1998).

"*Data Mining* é uma ferramenta utilizada para descobrir novas correlações, padrões e tendências entre as informações de uma empresa, através da análise de grandes quantidades de dados armazenados em *Data Warehouse* usando técnicas de reconhecimento de padrões, Estatísticas e Matemáticas" (NIMER & SPANDRI, 1998).

"Extração de conhecimento de base de dados (mineração de dados) é o processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados" (FAYYAD et al., 1996).

"Mineração de Dados é a exploração e análise de dados, por meio automático ou semiautomáticos, em grandes quantidades de dados, com o objetivo de descobrir regras ou padrões interessantes" (BERRY & LINOFF, 1997).

A fase de Mineração de Dados depende essencialmente da técnica e dos métodos que serão aplicados na base de dados, possibilitando o tratamento e descobrimento

das informações presentes. As tarefas em Mineração de Dados extraem tipos diferentes de conhecimento do banco de dados, sendo necessário a utilização de algoritmos diferentes para cada uma delas.

3.1. TAREFAS EM MINERAÇÃO DE DADOS

A tarefa é um tipo de Mineração de Dados com um propósito particular, da qual existem diversas, ou até dezenas, de implementações distintas por meio de vários algoritmos. Os algoritmos são divididos pelas tarefas levando em consideração o objetivo da implementação, ou seja, os algoritmos de uma mesma tarefa possuem a mesma finalidade.

Técnica	Descrição	Tarefas	Exemplos
Árvore de Decisão	Baseada em estágios de decisão (nós) e na separação de classes e subconjuntos, organiza os dados de forma hierárquica.	-Classificação -Predição	CART, CHAID, C5.0, ID-3.
Redes Neurais	Modelos inspirados na fisiologia do cérebro, nos quais o conhecimento é fruto do mapa de conexões neuronais e dos pesos dessas conexões.	-Classificação -Agrupamento -Predição	Perceptron, Rede MLP, Redes de Kohonen, Rede Hopfield, Rede BAM, Redes ART, Rede IAC, Rede LVQ, Rede Counterpropagation, Rede RBF, Rede PNN, Rede Time Delay, Neocognitron, Rede BSB.
Raciocínio Baseado em Casos	Baseado no método do vizinho mais próximo combina e compara atributos para estabelecer hierarquia de semelhança.	-Classificação -Agrupamento	BIRCH , CLARANS CLIQUE.
Algoritmos Genéticos	Métodos gerais de busca e otimização, inspirados na Teoria da Evolução, em que a cada nova geração, soluções melhores têm mais chance de ter "descendente".	-Classificação -Agrupamento	Algoritmo Genético Simples, Genitor, GA-Nuggets, GAPVMINER.
Conjuntos Fuzzy	Oferece uma grande vantagem para classificar dados com um alto nível de abstração.	-Classificação -Agrupamento	K-means. FCMdd
Regras de Indução	Processo para obter uma hipótese a partir de dados e fatos já existentes.	-Classificação -Predição	CART, CHAID
Regras de Associação	Estabelece uma correlação estatística entre atributos de dados e conjuntos de dados.	-Associação	Apriori, AprioriTid, AprioriHybrid, AIS, SETM.

Figura 2: Técnicas e Tarefas utilizadas na Mineração de Dados (In: GOLDSCHIMIDT, 2005)

3.1.1 DESCOBERTA DE ASSOCIAÇÕES

Nesta tarefa, os registros do conjunto de dados são chamados de transação, que é formada por um grupo de itens. A tarefa de descoberta de associação é caracterizada por buscar itens que constantemente aparecem juntos em transações do conjunto de dados. Exemplos de algoritmos implementados nessa tarefa são Apriori, GSP, DHP entre outros (ZAKI, 2000).

3.1.2. CLASSIFICAÇÃO

Nesta tarefa, existem dois tipos de atributo no conjunto de dados, o primeiro é do tipo atributo previsor e o segundo é chamado de atributo-alvo. Para os valores que diferem do atributo-alvo existe uma classe que faz referência a um grupo categórico relacionado a um conjunto predefinido. O objetivo da tarefa de classificação é descobrir uma função que relacione um conjunto de registros com determinado conjunto de classes. Descoberta essa função é possível aplicá-la para novos registros, dessa forma, conseguindo prever a qual classe esse registro pertence. Exemplos de técnicas que podem ser aplicadas na tarefa de classificação são Rede Neurais, Algoritmos Genéticos e Lógica Nebulosa (MICHIE et al., 1994).

3.1.3. REGRESSÃO

A tarefa de Regressão é parecida à tarefa de Classificação, onde são buscadas funções que relacionem os registros de uma base de dados em um intervalo de valores reais. A principal diferença é que o atributo-alvo adota valores numéricos. A tarefa de Regressão utiliza-se da Estatística, Rede Neurais entre outras técnicas que oferecem os recursos para sua implementação (MICHIE et al., 1994).

3.1.4. AGRUPAMENTO (CLUSTERIZAÇÃO)

Compreende um processo de partição dos elementos de um banco de dados em subconjuntos ou *clusters,* de uma maneira que os registros, que são semelhantes, fiquem agrupados diferenciando dos registros dos outros subconjuntos. Diferentemente da tarefa de classificação, não existem classes pré-definidas, os elementos são reunidos baseados na similaridade entre eles, sendo a Clusterização responsável por verificar as bases de dados (FAYYAD et al., 1996). Os algoritmos de k-Modes, k-Means, k-Prototypes entre outros são usados na implementação dessa tarefa.

3.2. TÉCNICAS EM MINERAÇÃO DE DADOS

As técnicas em Mineração de Dados estendem-se a qualquer teoria que possa fundamentar a implementação de método de *Data Mining.* Por exemplo, a teoria de Redes Neurais contribui com o desenvolvimento do algoritmo das Maquinas de Vetores de Suporte (SVM - *Support Vector Machines*), que são aplicados em tarefas de Classificação.

3.2.1. REDE NEURAIS

Segundo Haykin (2001), as redes neurais artificiais equivalem a padrões de processo paralelo distribuídos, composto por unidades simples de ajustes, que proporcionam apreender algum conhecimento ou relacionamento complexo experimental, habilitando-o para algumas aplicações como, por exemplo, previsão de novas características ou situações.

Nas redes neurais foram desenvolvidos fundamentos da estrutura do sistema nervoso, mais especificamente do cérebro humano, sendo a sua característica primaria a capacidade de aprender com base na mostra a exemplos. Uma rede neural é formada por sua arquitetura interna, uma rede interligada de neurónios, e a partir do

treinamento destas redes com base em exemplos, até que a própria arquitetura consiga aprender como solucionar o problema.

As redes neurais relacionam pesos sinápticos às conexões entre neurônios. Esses pesos mudam, por meio de algoritmos de aprendizagem, à medida que novas informações ou novas observações são incorporadas na rede. O modelo de rede neural espelha os pesos sinápticos que possibilitam, depois de ajustes regulares de uma função de ativação, a implantação de uma variável de saída ou de resposta a partir de dados de entrada (PERERA et al., 2011), como mostra a Figura 3.

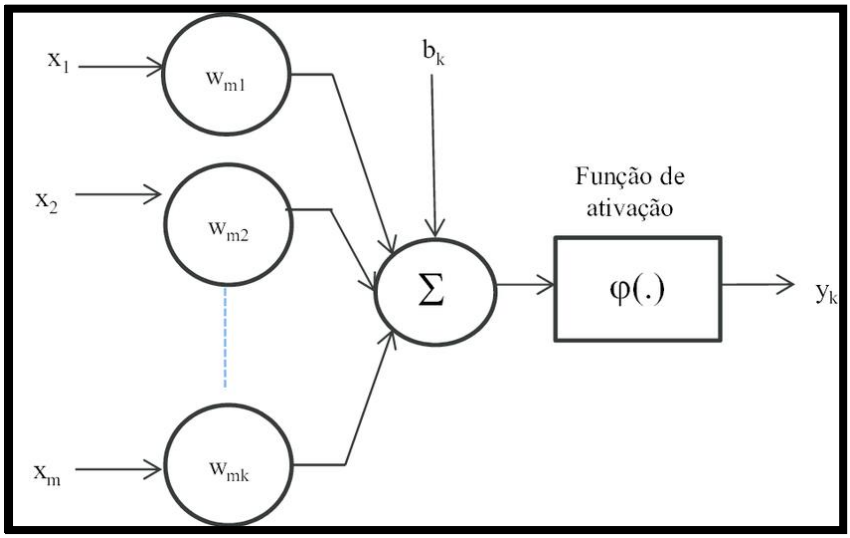


Figura 2: Modelo de rede neural (In: HAYKIN, 2001, p.36)

3.2.2. ÁRVORES DE DECISÃO

Segundo Han e Kamber (2000) árvores de decisão são padrões estatísticos que aplicam treinamento supervisionado para classificação e predição de dados. O modelo cria uma estrutura de decisão em forma de árvore, onde cada nó, pode ser classificado como atributo de teste, e cada nó-folha tem um rótulo de classe. A árvore sempre inicia por um único nodo, conhecido como nodo-raiz, e vai sendo dividida até levar a classe. O modo de particionamento de cada nodo é definido a partir da aplicação de um algoritmo. O C4.5, C5.0 e o CART são os exemplos mais populares de algoritmos de árvore de decisão.

A Figura 4 mostra o modelo padrão de uma árvore de decisão.

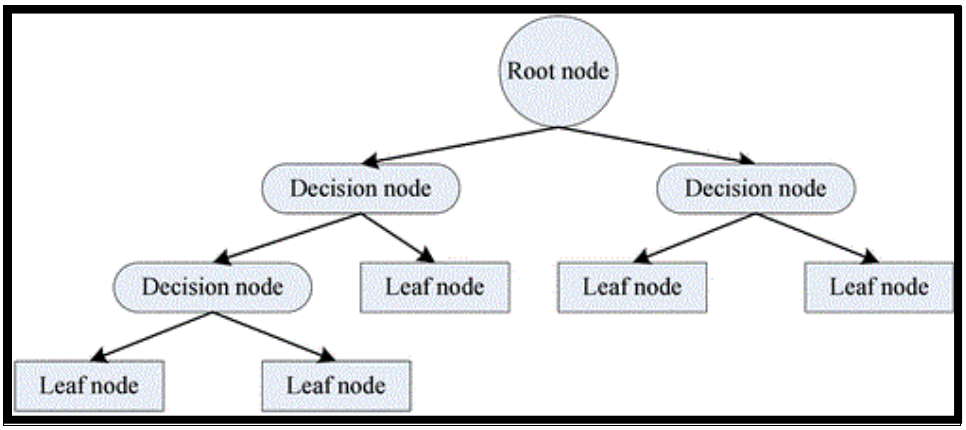


Figura 3: Modelo de árvore de decisão (In: SYACHRANI; JEONG; JUNG, 2012, p.636)

3.2.3. REGRAS DE ASSOCIAÇÃO

Uma apresentação tradicional das Regras de Associação é a anotação condicional, onde *SE X ENTÃO Y*, tradicionalmente mostrada no formato de implicação $X \rightarrow Y$, possuindo dois elementos: o que aparece anteriormente que é representado por *X* e o que é a consequência, representado pelo *Y* (AGRAWAL; IMIELINSKI; SWAMI, 1993).

Usualmente as regras de associação são relacionadas a dois fatores chamados Suporte (Sup) e Confiança (Conf), que são responsáveis pela avaliação de, por exemplo, o quanto que determinado produto X implica na compra de um produto Y. O valor de Suporte representa a quantidade de vezes que uma associação ocorreu em relação ao total de registros, ou seja, é a probabilidade de uma transação corresponder a condição X. Enquanto o valor de Confiança mostra a porcentagem de acontecimentos do antecedente onde o produto consequente está associado, quer dizer, é a probabilidade de que uma operação satisfaça a condição Y, se ela satisfaz a condição X (LEVY, 1999).

3.2.4. RACIOCÍNIO BASEADO EM CASOS

Segundo Fernandes (2003), a técnica de Raciocínio Baseado em Casos encontra a solução para determinada situação por meio da recuperação e adaptação de soluções

parecidas que foram utilizadas no passado, dentro de um mesmo escopo do problema.

O processo acontece quando um novo evento é identificado ao sistema. Observado o novo obstáculo, são empregados conjuntos de métrica de similaridade para apontar quais situações semelhantes que ocorreram no passado, assim como se indicam as características-chave aplicadas nessa comparação (Fonseca, 2008).

Para Wangenheim et al. (2003), um sistema de Raciocínio Baseado em Casos pode ser dividido em 4 pilares básicos, que são:

- Representação do Conhecimento: Normalmente o conhecimento é demonstrado em forma de casos, que relatam casos concretos.
- Medida de Similaridade: É a maneira que será definida a similaridade entre o problema atual e os outros diversos casos, sendo utilizada diversas vezes, par a par, para todos os problemas e assim chegar em um valor de similaridade, sendo que os casos com maior similaridade são recomendados como virtual solução para o problema atual.
- Adaptação: situações antigas relatadas como casos dificilmente serão iguais ao problema presente, sendo assim, é necessários mecanismos que possam adaptar os casos, de maneira a satisfazer o problema presente.
- Aprendizado: Sempre que o sistema solucionar um caso com sucesso, esse deverá ser capaz de lembrar-se dessa questão no futuro como mais um novo problema.

A restrição em relação ao uso de sistemas de Raciocínio Baseado em Casos está no acesso as bases de dados completas, corretas e confiáveis que possuam em seus históricos, o relatório completo dos problemas e soluções passadas aplicadas e arquivadas. BIRCH, CLARANS e CLIQUE são os algoritmos mais famosos aplicados nessa técnica (DIAS, 2001).

3.2.5. ALGORITMOS GENÉTICOS

O Algoritmos Genéticos foi inventado nos anos 60 por John Holland, porém, foram seus alunos, na Universidade de Michigan, que os desenvolveram, por volta dos anos 1970.A técnica é fundamentada nas noções da seleção natural e da genética natural (FREITAS, 2007). Segundo Almeida (2006), os Algoritmos Genéticos constroem um sistema artificial, baseado no processo natural, através de operações que representam os mecanismos genético da natureza.

Para Mitchell (1996), essa ideia, de evolução natural, é utilizada na computação com o intuito de resolver diversos problemas computacionais, visto que, esse conceito pode ser aplicado em diversas áreas. Questões que envolvem demandas em espaços muito amplos de solução e procura de um conjunto de regras de classificação com suporte de uma base de dados são exemplos de problemas que podem ser resolvidos por essa técnica.

A Figura 5 mostra um fluxograma da estrutura de um algoritmo genético, onde é possível perceber que o algoritmo é formado de um ciclo, que define as gerações de indivíduos, em que são efetuados os elementos básicos dessa técnica: A função de avaliação, seleção de indivíduos a integrar a nova população, os operadores genéticos e a troca da antiga população pela nova que foi gerada, criando indivíduos mais aptos (COSTA et al., 2013).

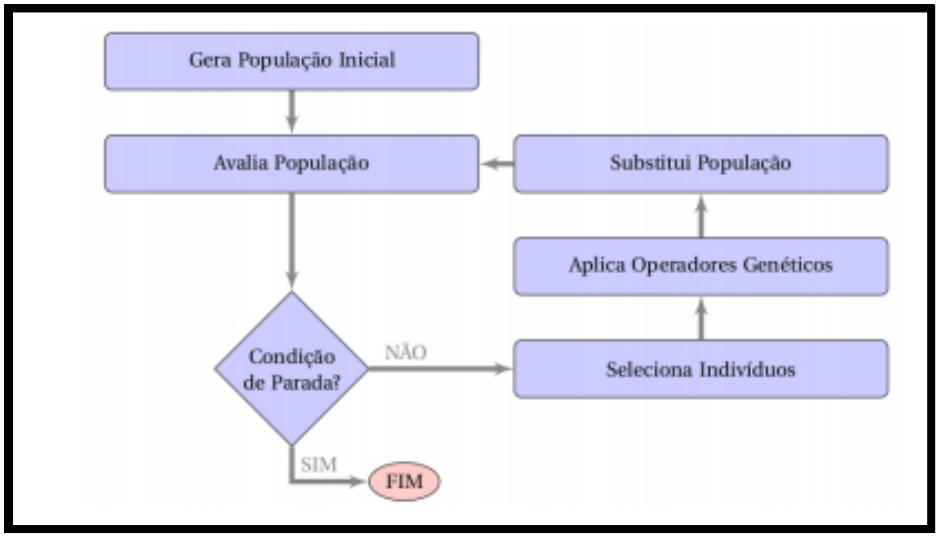


Figura 4: Fluxograma de um algoritmo genético (In: COSTA et al., 2013, p.15)

Técnicas apoiadas em modelos heurísticos como o algoritmo genético não asseguram que será encontrada uma solução ideal para o problema, porém é possível chegar em respostas próximas ou aceitáveis, lembrando que os algoritmos genéticos são aconselháveis para problemas muito complexos com grande quantidade de variáveis e restrições ou com amplos espaços de busca.

3.2.6. CONJUNTOS FUZZY

É uma técnica que possibilita fazer sistemas que trabalhem com informações imprecisas ou subjetivas. Ao contrário da Lógica Clássica, a Lógica Nebulosa, como também é conhecido os Conjuntos Fuzzy, proporciona maior flexibilidade na definição e na análise dos conceitos.

Na Teoria Clássica dos Conjuntos, um elemento pode pertencer ou não pertencer a um conjunto, sendo essa pertinência representado por 1(pertence) ou 0(não pertence) aos elementos de um conjunto universo. O problema dessa abordagem é que muitas vazes essa definição binária não é suficiente para descrever um elemento em várias situações do mundo real. Então a teoria de Conjuntos Fuzzy estende a Teoria Clássica de Conjuntos e proporciona que um elemento possa pertencer, com diferentes graus, a mais de um conjunto. (LOPES, 2010)

3.3. ALGORITMOS PARA MINERAÇÃO DE DADOS

3.3.1. MÁQUINA DE VETORES DE SUPORTE

A Máquina de Vetores de Suporte ou Support Vectors Machine (SVM) foi desenvolvida por Vapnik, com a intenção de resolver problemas de classificação de padrões. Para Haykin (1999) a SVM se trata de outro tipo de rede neurais alimentadas adiante, quer dizer, redes nas quais as saídas dos neurónios de uma camada alimentam os neurónios da camada posterior, não acontecendo a realimentação.

Segundo Chaves (2006), uma forma simples de narrar uma atividade de uma SVM é a partir de duas classes e um conjunto de pontos que são dessas classes, uma SVM determina o hiperplano que afasta os pontos de maneira a colocar a maior quantidade

de pontos de mesma classe de um lado, ao mesmo tempo que, maximiza a distância de cada classe a esse hiperplano. A margem de separação é a menor distância entre o hiperplano e os pontos dessa classe e o hiperplano criado é definido por um subconjunto dos pontos das duas classes, conhecido como vetores suporte.

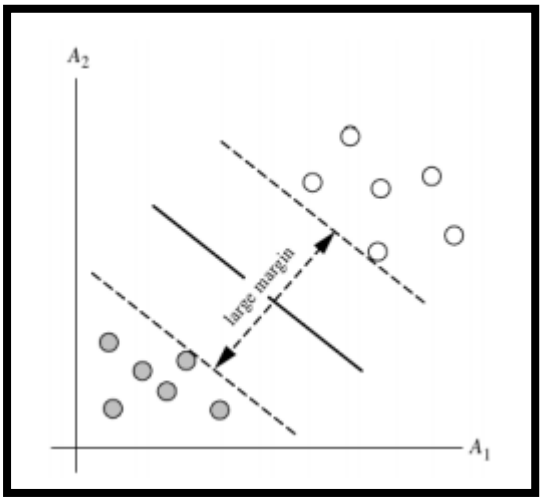


Figura 5: Hiperplano que separa as classes (In: Han and Kamber, 2000)

3.3.2. ALGORITMO C4.5.

O C4.5 é um algoritmo de árvore de decisão que foi publicado em 1987 por John Ross Quilan. Segundo Quilan (1993), o algoritmo tem a finalidade de estabelecer um modelo classificador na forma de árvore de decisão, a partir de dois estados ao longo do processo, que são: a folha que mostra um ponto final na classificação, na qual são atribuídos uma classe e o nó de decisão, onde apoiando-se no atributo em análise, poderá envolver uma ramificação sem seguida de uma folha ou uma subárvore para cada provável valor encontrado na base.

As principais contribuições do C4.5. são lidar com atributos categóricos e contínuos, tratar valores desconhecidos, utilizar a medida de razão de ganho para selecionar o atributo que melhor divide os exemplos, lida com questões em que atributos possuem custos distintos e apresenta um método de pós-poda das árvores geradas.

3.3.3. APRIORI

O algoritmo Apriori foi desenvolvido em 1993 por Agrawal e Srikant (1994) e é utilizado em tarefas de associação, em que são buscados itens semelhantes em determinado intervalo de tempo. Segundo Pasta (2011) esse algoritmo pode ser divido em dois passos, sendo:

- Primeiro passo é a pesquisa de itens frequentes, sendo que o usuário determina um limite mínimo para o apoio e o algoritmo busca todos os conjuntos de itens que se mostram com um apoio maior a esse limite.
- O segundo passo é a construção de regras baseadas nos conjuntos de itens encontrados no primeiro passo. A confiança de cada regra a calculada pelo algoritmo e conserva só aqueles em que a confiança é maior que o limite imposto pelo usuário.

A ideia primordial por trás do algoritmo Apriori é fazer com que qualquer subconjunto de um Conjunto de Itens seja contínuo. Ou seja, o conjunto de candidatos k itens é capaz de ser elaborado por uma combinação dos Conjuntos de Itens de tamanho k - 1, resultando, na eliminação dos itens que possuem algum subconjunto que não seja frequente (AGRAWAL; SRIKANT, 1994).

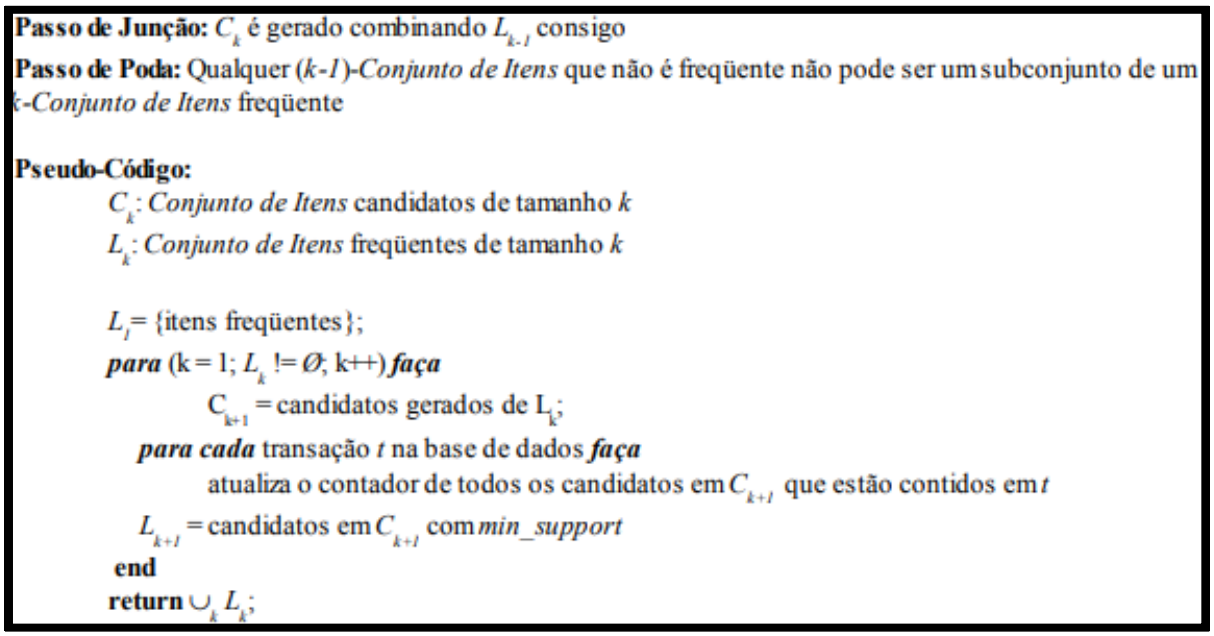


Figura 6: Pseudocódigo algoritmo Apriori (In: JUNIOR, 2008, p.19)

3.3.4. ALGORITMO K-MEANS

O algoritmo k-means é o mais conhecido quando falamos de tarefas de agrupamento, sendo conhecido também, na sua forma mais simples, como algoritmo de Lloyd, é trabalha da seguinte maneira:

Segundo Costa et al. (2013). Primeiramente são indicados o número k de grupos que está se buscando. Posteriormente k pontos são selecionados arbitrariamente para caracterizar os centroides dos grupos, desse modo, um conjunto normalmente de vetores, é particionado de maneira que cada elemento é atribuído ao grupo de centroide mais próximo, conforme com a distância euclidiana comum. Sempre que houver iteração do algoritmo os k centroides ou médias, são recalculados segundo os elementos presentes no grupo subsequente todos os elementos são realocados para o grupo no qual o novo centroide se encontra mais próximo, como mostra na Figura abaixo. Esse algoritmo é realizado várias vezes e é escolhido o melhor resultado.

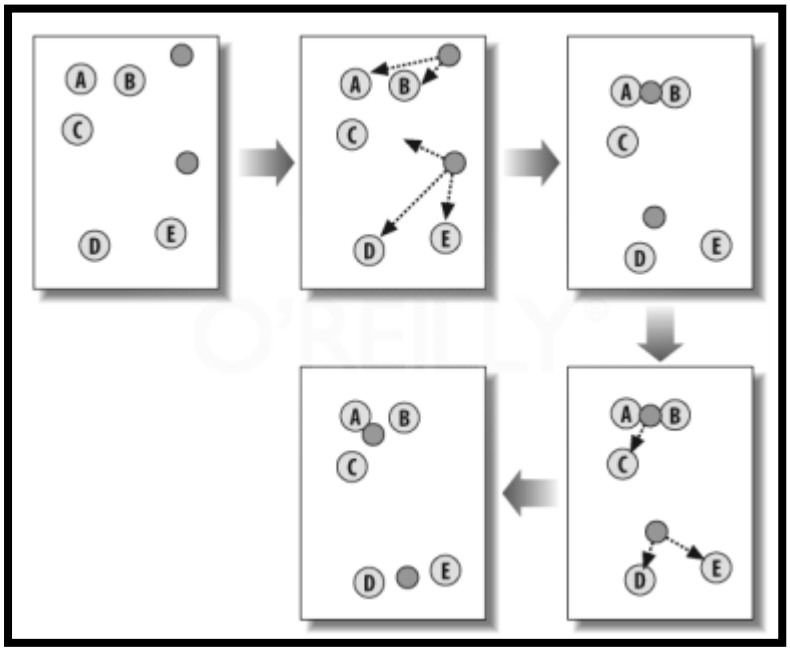


Figura 7: Algoritmo K-Means, passo a passo (In: Segaran, 2007)

4. WEKA

O Waikato Environment for Knowledge Analysis mais conhecido pela sigla WEKA foi desenvolvido por estudantes da Universidade de Waikato, na Nova Zelândia, no ano de 1999 e sua licença é GPL (General Public Licence), ou seja, é um programa de distribuição e divulgação livre. O Weka possui uma coleção de algoritmos que são implementados em problemas de mineração de dados, que envolvem ferramentas de pré-processamento, além disso, ele oferece suporte a todo o processo de mineração, incluindo desde a preparação dos dados de entrada, avaliação estatística de aprendizagem, visualização dos dados de entrada e resultados.

O Weka foi desenvolvido na linguagem JAVA, portanto é possível executa-lo em diversas plataformas, dentre ela o Windows, MAC Os X e Linux, sendo necessário apenas a instalação da Máquina Virtual Java. O software é formado por dois pacotes: um pacote autónomo para utilização direta dos algoritmos, utilizando um formato de dados particular, e um pacote de classes Java que executam esses algoritmos. Dessa maneira é possível desenvolver uma aplicação Java que utilize esses algoritmos e aplica-los em todos os bancos de dados pela conexão JDBC (Java DataBase Connectivity) (PASTA, 2011).

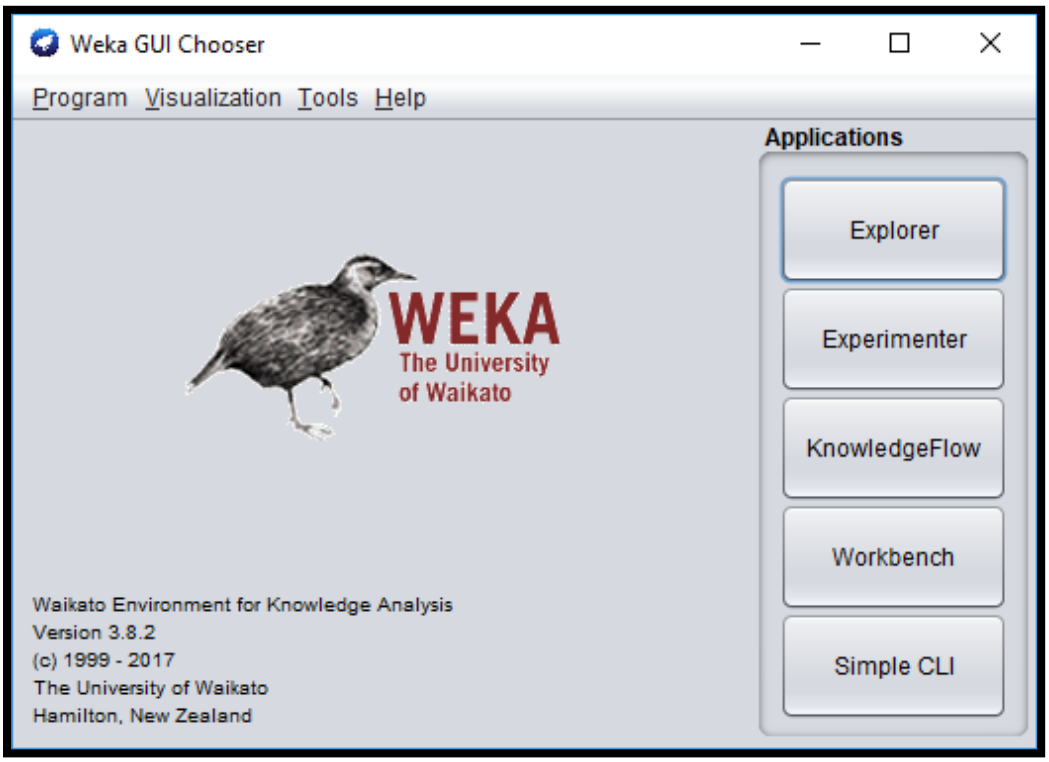


Figura 8: Tela inicial do software WEKA

O WEKA pode ser empregado de vários moldes, possuindo 5 interfaces implementadas, como mostra a Figura 8, e elas são:

1) Explorer: Essa interface possui as aplicações de pré-processamento, análise, (classificação, associação e clusterização) e visualização dos resultados.
2) Experimenter: Nessa interface é onde o usuário pode orientar testes estatísticos entre os esquemas de aprendizagem de ferramenta; usuário pode empregar vários algoritmos concomitante e confrontar os resultados, e então selecionar o melhor algoritmo para o seu grupo de dados.
3) Knowledge Flow: O ambiente possui praticamente as mesmas funções do Explorer, porém possui uma representação em forma gráfica, com a vantagem de permitir aprendizagem incremental.
4) Workbench: É um ambiente que combina todas as GUI em uma única interface.
5) Simple CLI: Essa interface prove um ambiente para a inserção dos comandos, apesar da aparência relativamente simples, é executável qualquer operação suportada pelo Weka.

As Figuras 9 e 10 abaixo mostram a diferença entre interface gráfica e a interface baseada em linhas de comando do software WEKA.

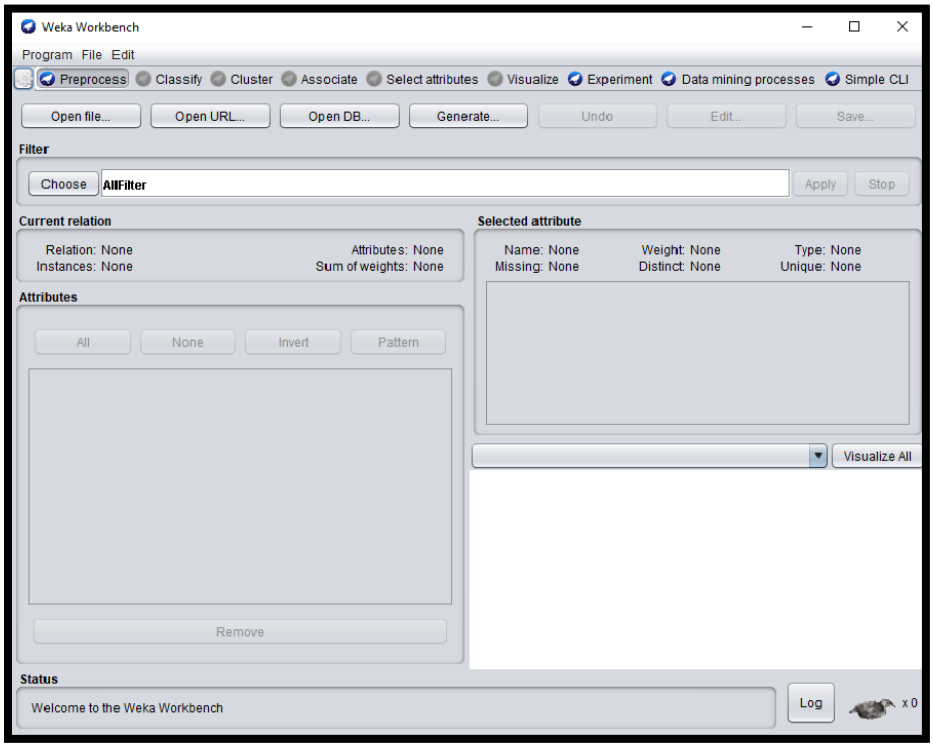


Figura 9: WEKA Workbench

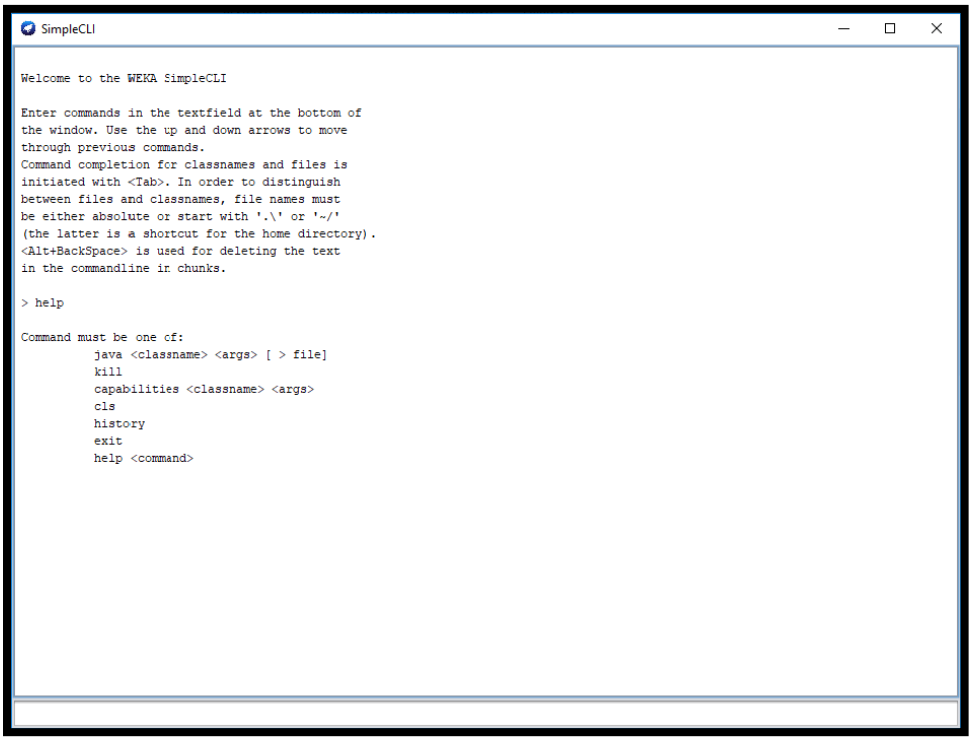


Figura 10: SimpleCLI

Segundo Hall et al. (2009), a interface disponibilizada pelo Weka possibilita que os algoritmos de aprendizagem e as variadas ferramentas para transformação possam ser aplicados as bases de dados sem que seja precisa a escrita de código. O Weka possui métodos para as situações padrões de Data Mining, que são as de regressão, classificação, regras de associação, agrupamento e seleção de atributos. O formato dos algoritmos da ferramenta Weka é chamado ARRF, que é um formato de entrada particular da ferramenta tendo a forma de uma tabela relacional simples, podendo ser lido de um arquivo e/ou criado com suporte de uma base de dados. Além do ARRF o software suporta outros formatos como CSV, C4.5. e binário.

O arquivo ARFF é separado em duas divisões, a primeira possui uma lisa de todos os atributos, onde são definidos o tipo do atributo e/ou os valores que ele pode representar. Esses valores precisam estar entre chaves destacados por virgula. A segunda é formada pelas instancias presentes nos dados, o atributo de cada instancia devem ser separadas por virgula, e os que não possuírem valor, o valor deve ser mostrado com o caractere “?”. Para definir as informações presentes existem marcações como por exemplo os atributos são marcados pelo @atribute, os dados

possuem a marcação @data e o conjunto de dados é especificado pela marcação @relation.

A Figura 11 mostra um exemplo de um arquivo ARRF.

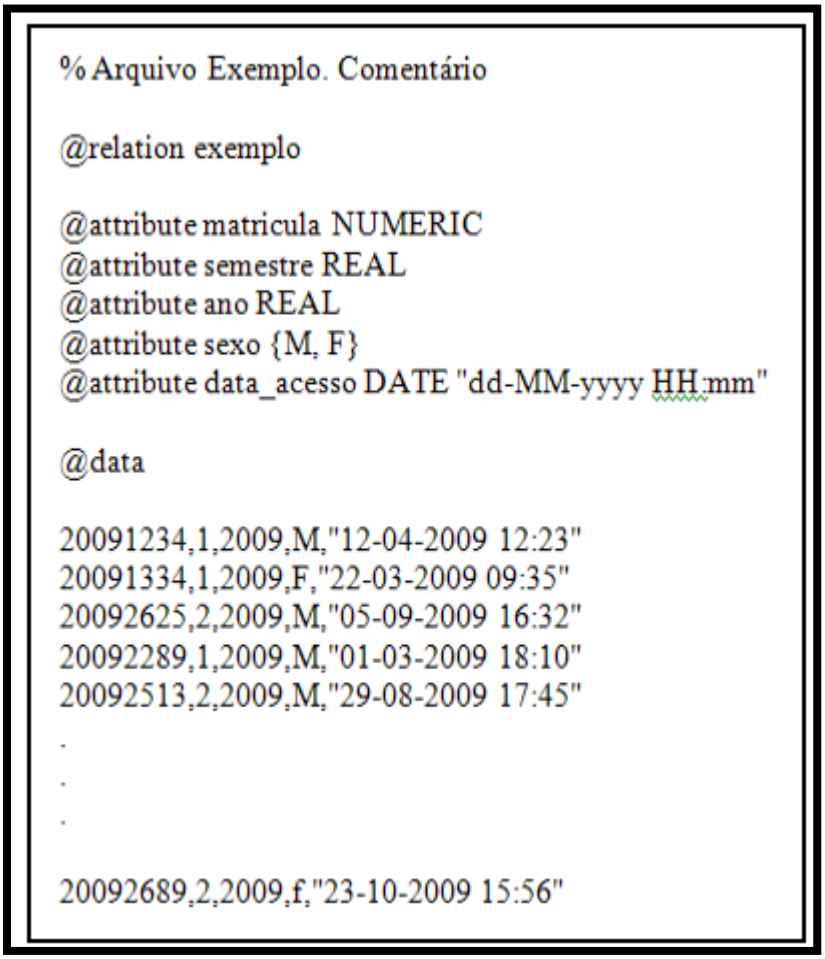


Figura 11: Exemplo de arquivo formato ARFF (In: PASTA,2011, p.94)

5. ESTUDO DE CASO

O objetivo desse capítulo é a demonstração do software Waikato Environment for Knowledge Analysis, mais conhecido como WEKA, apresentando no estudo de caso os recursos que a ferramenta possui e a aplicação de alguns algoritmos e técnicas que foram descritos no trabalho, manipulando uma base de dados fictícia.

5.1. WEKA EXPLORER

A ferramenta de WEKA possui diversas interfaces, sendo que, a utilizada nesse trabalho foi o Explorer, onde é possível testar e explorar diversos tipos de algoritmos para a mineração de dados. A Figura 12. demonstra como é a interface inicial de pré-processamento do Weka Explorer.

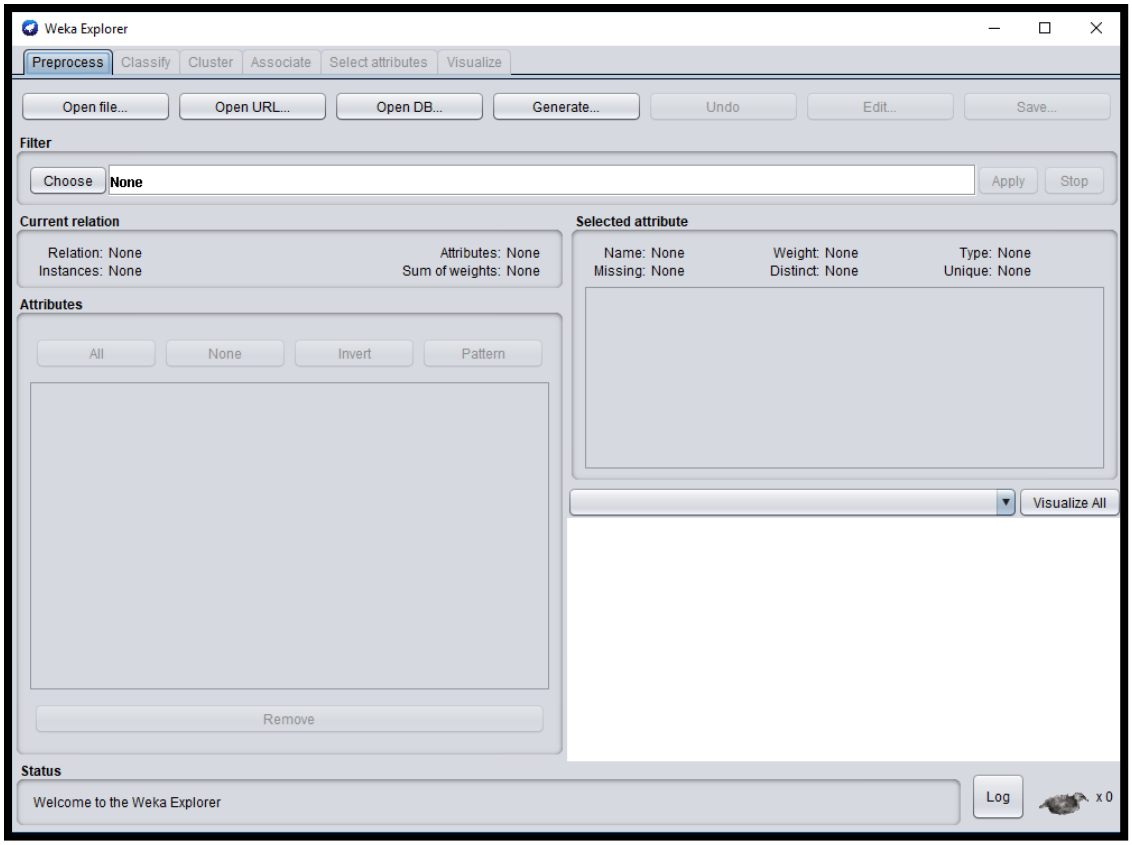


Figura 12: Preprocess **-** Weka Explorer

Na guia Preprocess é possível selecionar a base de dados, em que será submetida os algoritmos, lembrando que o próprio WEKA fornece bases de dados fictícias sendo possível também, baixar esses dados de outros lugares. Para a realização da pesquisa foram utilizadas as bases de dados disponibilizadas pelo WEKA. Para fazer a sua seleção, basta acessar a opção Open file no menu e selecionar a base desejada, como mostra a Figura 13.

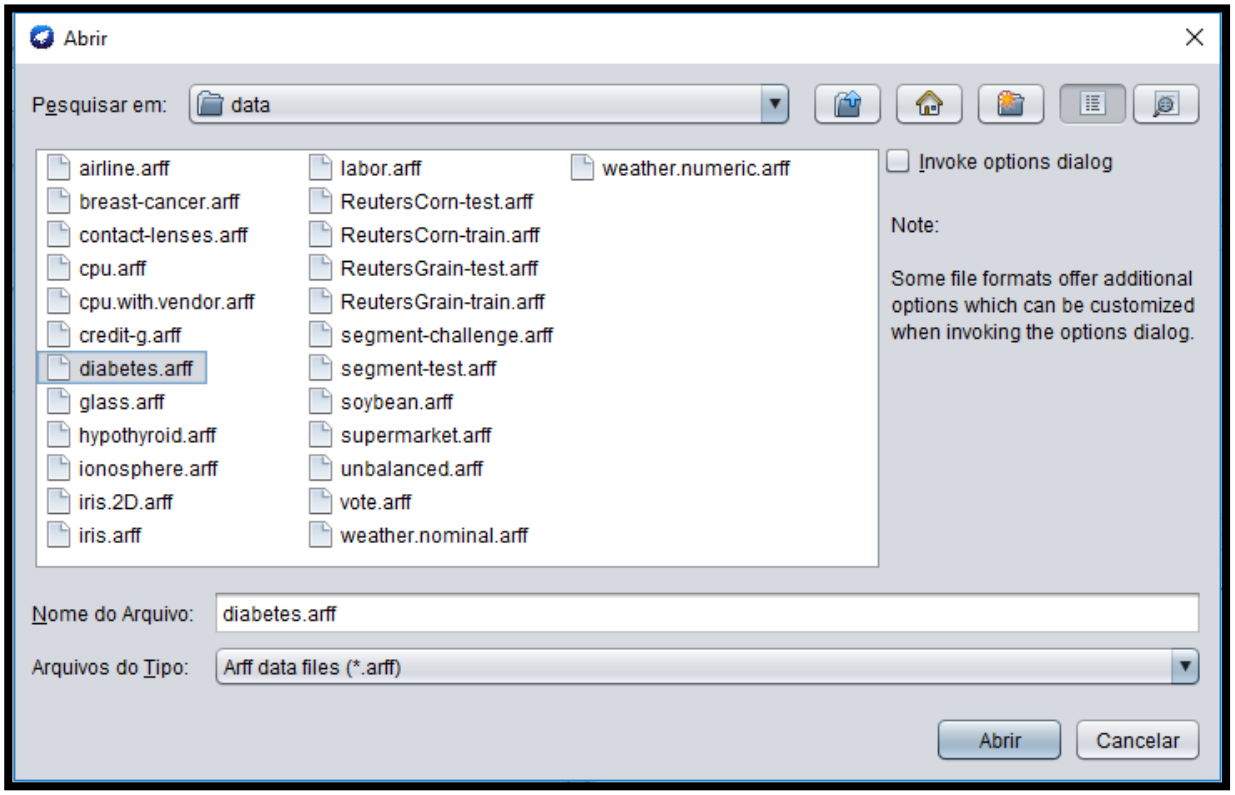


Figura 13*:* Bases de Dados no WEKA

Depois de selecionado o conjunto de dados, na interface será mostrado diversas informações relacionadas a base de dados, como é possível observar na Figura 14. Além disso, na região superior da interface é possível realizar a importação da base de dados de outras formas através do Open URL, onde os dados são importados pela internet; Open DB, para conectar a dados por uma conexão Jdbc e Generate para criar dados aleatórios tendo como base um algoritmo específico. É possível também, a partir do comando Save, salvar os dados em um formato diferente e em um outro local.

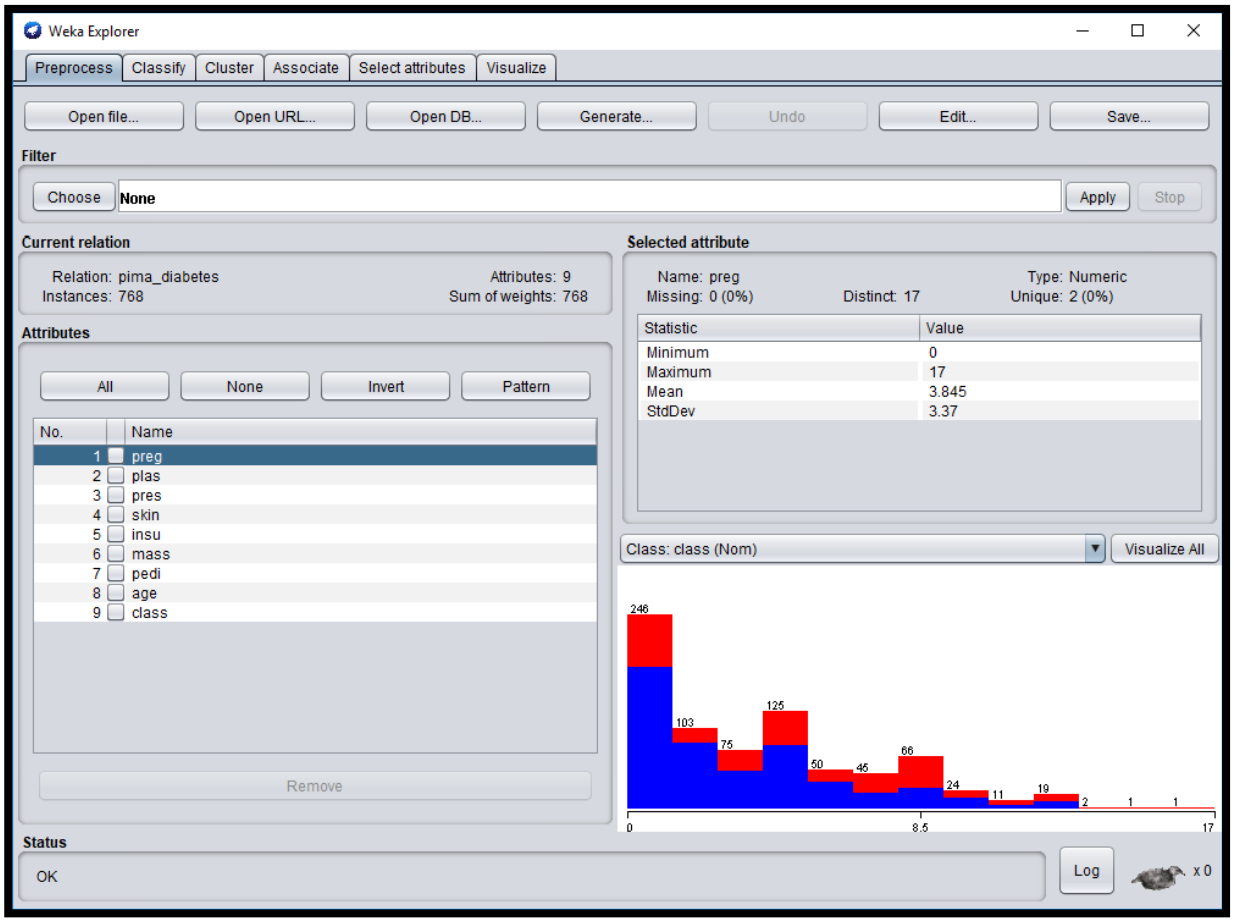
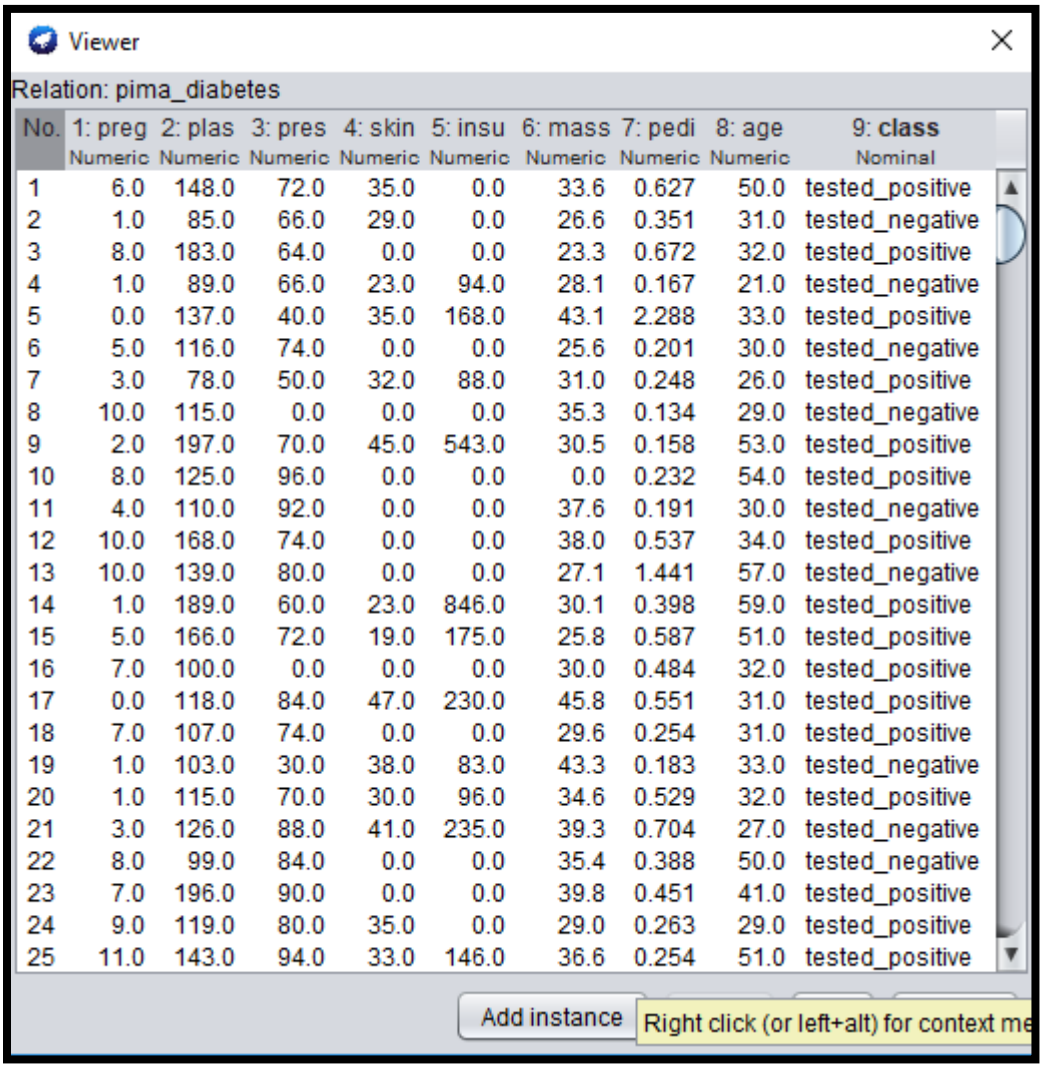


Figura 14: Informações sobre a base de dados

Ainda na tela de Preprocess, existem diversas funcionalidades que serão descritas a seguir:

Edit: esse comando permite que os dados sejam carregados em formato de grade sendo possível a visualização e alteração dos mesmos. Além disso, clicando sobre as instancias é realizável outras funcionalidades como excluir, copiar ou localizar instancias. Outras funcionalidades são possíveis clicando sobre o nome da coluna como substituição de valores, alteração de dados, exclusão e até ordenação. A Figura 15 mostra a tela do Edit.



No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	148.0	72.0	35.0	0.0	33.6	0.627	50.0	tested_positive
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	tested_negative
3	8.0	183.0	64.0	0.0	0.0	23.3	0.672	32.0	tested_positive
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	tested_negative
5	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33.0	tested_positive
6	5.0	116.0	74.0	0.0	0.0	25.6	0.201	30.0	tested_negative
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	tested_positive
8	10.0	115.0	0.0	0.0	0.0	35.3	0.134	29.0	tested_negative
9	2.0	197.0	70.0	45.0	543.0	30.5	0.158	53.0	tested_positive
10	8.0	125.0	96.0	0.0	0.0	0.0	0.232	54.0	tested_positive
11	4.0	110.0	92.0	0.0	0.0	37.6	0.191	30.0	tested_negative
12	10.0	168.0	74.0	0.0	0.0	38.0	0.537	34.0	tested_positive
13	10.0	139.0	80.0	0.0	0.0	27.1	1.441	57.0	tested_negative
14	1.0	189.0	60.0	23.0	846.0	30.1	0.398	59.0	tested_positive
15	5.0	166.0	72.0	19.0	175.0	25.8	0.587	51.0	tested_positive
16	7.0	100.0	0.0	0.0	0.0	30.0	0.484	32.0	tested_positive
17	0.0	118.0	84.0	47.0	230.0	45.8	0.551	31.0	tested_positive
18	7.0	107.0	74.0	0.0	0.0	29.6	0.254	31.0	tested_positive
19	1.0	103.0	30.0	38.0	83.0	43.3	0.183	33.0	tested_negative
20	1.0	115.0	70.0	30.0	96.0	34.6	0.529	32.0	tested_positive
21	3.0	126.0	88.0	41.0	235.0	39.3	0.704	27.0	tested_negative
22	8.0	99.0	84.0	0.0	0.0	35.4	0.388	50.0	tested_negative
23	7.0	196.0	90.0	0.0	0.0	39.8	0.451	41.0	tested_positive
24	9.0	119.0	80.0	35.0	0.0	29.0	0.263	29.0	tested_positive
25	11.0	143.0	94.0	33.0	146.0	36.6	0.254	51.0	tested_positive

Figura 15: Base de dados carregada para edição

Filter: nessa opção é possível aplicar uma diversidade de funções que permitem mudar os dados de várias maneiras.

Current relation: mostra o nome da relação, número de atributos, número de linhas e a soma das pessoas das instancias.

Attributes: mostra uma relação de todos os atributos da instancia. É possível selecionar esses atributos na caixa de verificação e clicar no botão Remove, para exclui-los. Também existem outros botões como All, seleciona todos; None, nenhum; Invert, inverte a seleção; Pattern, informar uma expressão regular em Pearl. Clicando sobre o atributo é possível ver as suas informações na área de Selected Attribute.

5.2. CLASSIFICAÇÃO E REGRESSÃO NO WEKA

Para executar os algoritmos classificadores é necessário entrar na guia Classify, como mostra a Figura 16. Nessa interface é possível observar o botão Choose que permite realizar a escolha de qual algoritmo, disponível, será utilizado para mineração, conforme mostra a Figura 17. É importante lembrar que alguns algoritmos não vão estar disponíveis, pois diferentes algoritmos de classificação são utilizados para diferentes tipos de dados.

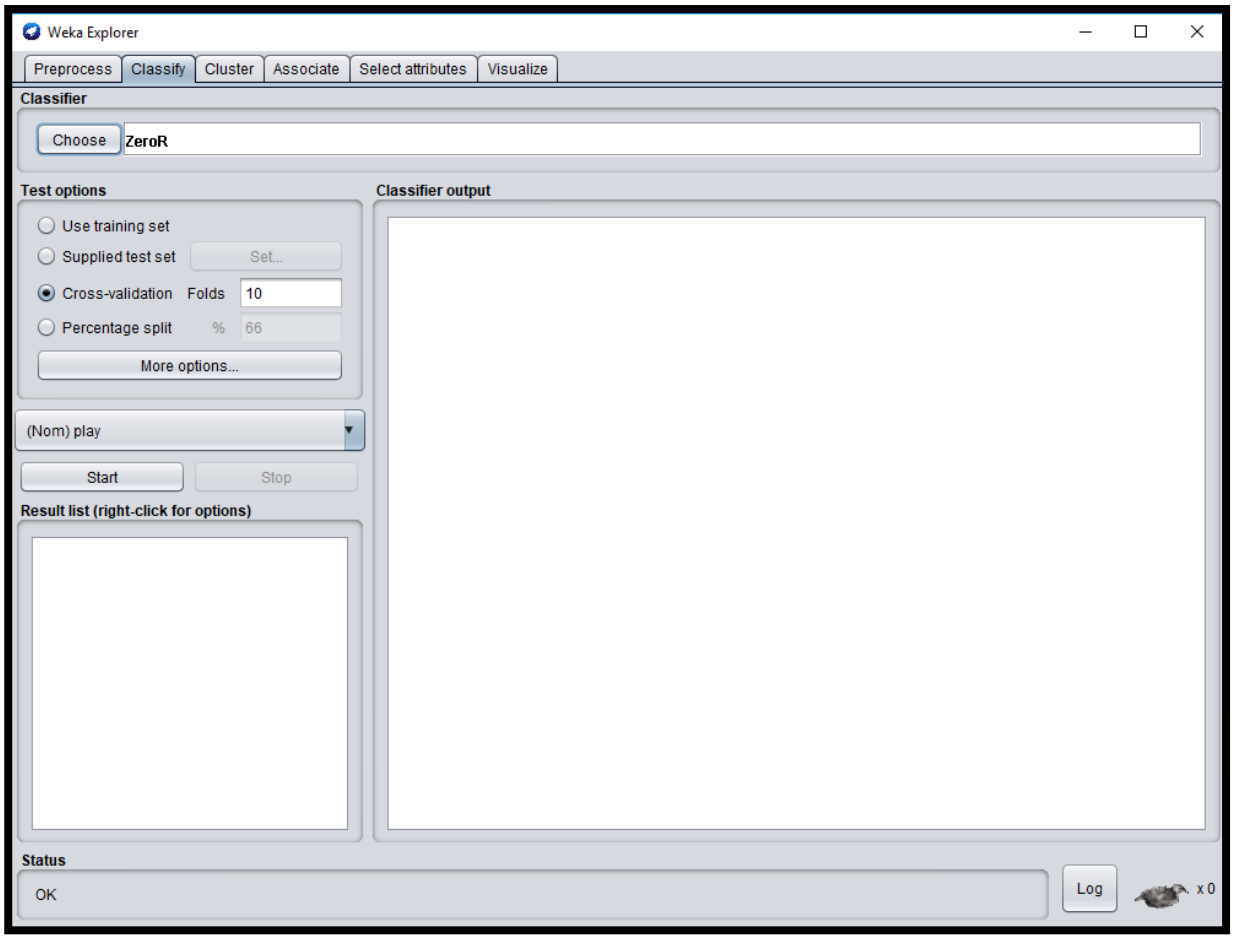


Figura 16: Interface Classify

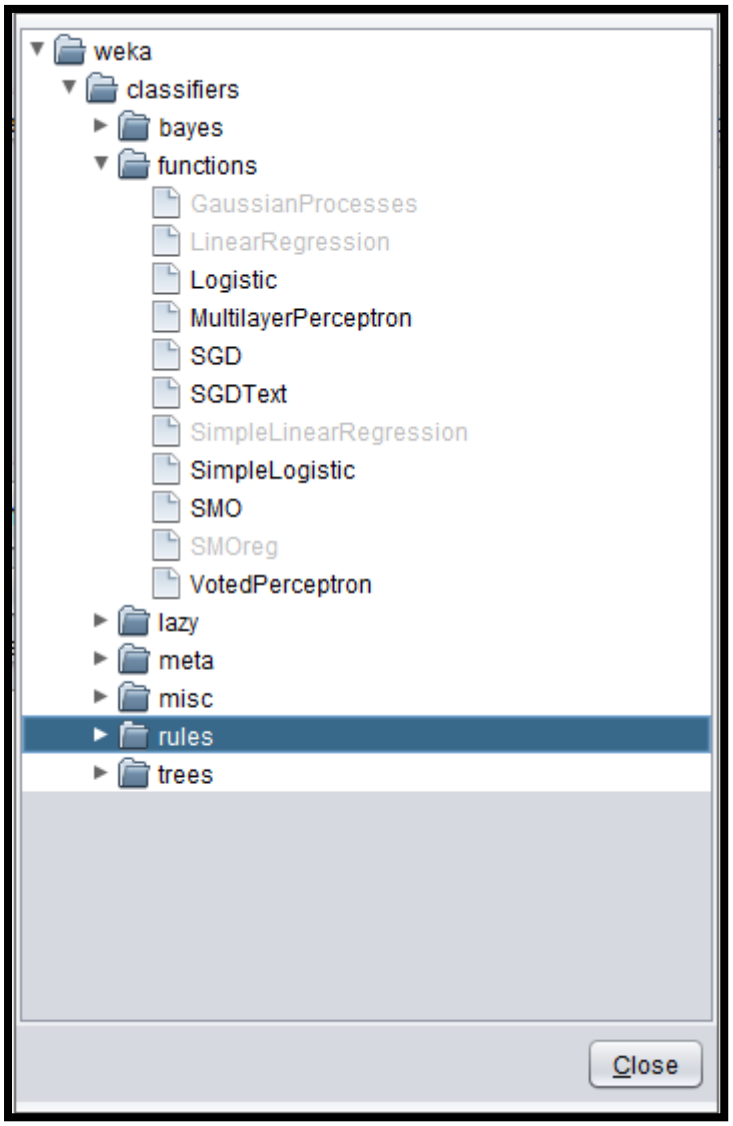


Figura 17: Classificadores

Como visto na Figura 16, a interface do Classify oferece 4 opções de teste. Nesse trabalho, nessa etapa, será usada a opção Use training set, onde o modelo será avaliado sobre o mesmo conjunto de dados em que foi treinado.

5.2.1. MÁQUINA DE VETOR DE SUPORTE COM SMO

Para exemplificação dos algoritmos de classificação, nesse estudo de caso, será utilizado o classificador de Máquina de Vetor de Suporte. Na interface de preprocess, carregue a base de dados weather.nominal.arff, após isso, basta entrar na guia Classify, clicar no botão Choose e selecionar o SMO que está na pasta functions,

marque, no Test options, o Use training set, e, por fim, basta clicar em Start. O resultado será mostrado no Classifier output, como é mostrado na Figura 18.

```
=== Run information ===

Scheme:       weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classi
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: yes, no

BinarySMO

Machine linear: showing attribute weights, not support vectors.

         0.543  * (normalized) outlook=sunny
 +      -1.0266 * (normalized) outlook=overcast
 +       0.4837 * (normalized) outlook=rainy
 +       0.2834 * (normalized) temperature=hot
 +      -0.2614 * (normalized) temperature=mild
 +      -0.0219 * (normalized) temperature=cool
 +      -1.0219 * (normalized) humidity=normal
 +      -0.7872 * (normalized) windy=FALSE
 +       0.1354

Number of kernel evaluations: 84 (85.859% cached)



Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds

=== Summary ===

Correctly Classified Instances          12               85.7143 %
Incorrectly Classified Instances         2               14.2857 %
Kappa statistic                          0.6585
Mean absolute error                      0.1429
Root mean squared error                  0.378
Relative absolute error                 30.7692 %
Root relative squared error             78.8263 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,400    0,818      1,000    0,900      0,701    0,800     0,818     yes
                 0,600    0,000    1,000      0,600    0,750      0,701    0,800     0,743     no
Weighted Avg.    0,857    0,257    0,883      0,857    0,846      0,701    0,800     0,791

=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 2 3 | b = no
```

Figura 18: Saída do Classificador

A saída gera diversas informações seguindo uma estrutura básica que varia de acordo com o algoritmo usado, os dados, e as opções de treino e testes definidos. A saída é composta por:

- Run information: mostra um resumo da execução, com o nome da relação, instancias e atributos, em seguida, é relatado o modelo de teste utilizado, que no exemplo foi o evaluate on training data.
- Classifier Model: a partir daqui inicia o modelo. No exemplo de classificador foi utilizado o SMO então a estrutura mostra o linear kernel usado, o peso de cada atributo e a forma que o dado foi classificado, no exemplo, yes ou no. No final mostra o tempo gasto para executar o modelo, que no exemplo foi 0,02 segundos.
- Summary: mostra um resumo do desempenho do modelo. Foram classificados corretamente 12 instancias e 2 incorretamente. Além disso, esta divisão traz outros vários dados estatísticos sobre o desempenho do modelo.
- Detailed Accuracy By Class: exibe o desempenho por classe, neste cenário, teremos uma linha para cada classe mais uma linha de média. No exemplo, com 2 classes, temos uma linha para yes, uma para no e uma com a média.
- Confusion Matrix: por fim mostra a matriz de confusão, que é uma tabela de contingência criada entre os dados de teste e a classificação feita pelo algoritmo. É possível ver, por exemplo, que 9 instancias eram da classe yes e de fato foram classificadas como yes, enquanto 2 instancias eram no e foram classificados como yes.

5.3. AGRUPAMENTO COM KMEANS

Para demostrar o agrupamento, carregue o conjunto de dados Iris, como foi explicado na Seção 5.1, e em seguida, clique na guia Cluster. Em Cluster clique em choose e selecione SimpleKMeans. Nesse algoritmo é preciso informar a quantidade de clusters por meio do parâmetro K. Para isso, clique em SimpleKMeans, e altere numClusters de 2 para 3, segundo a Figura 19.

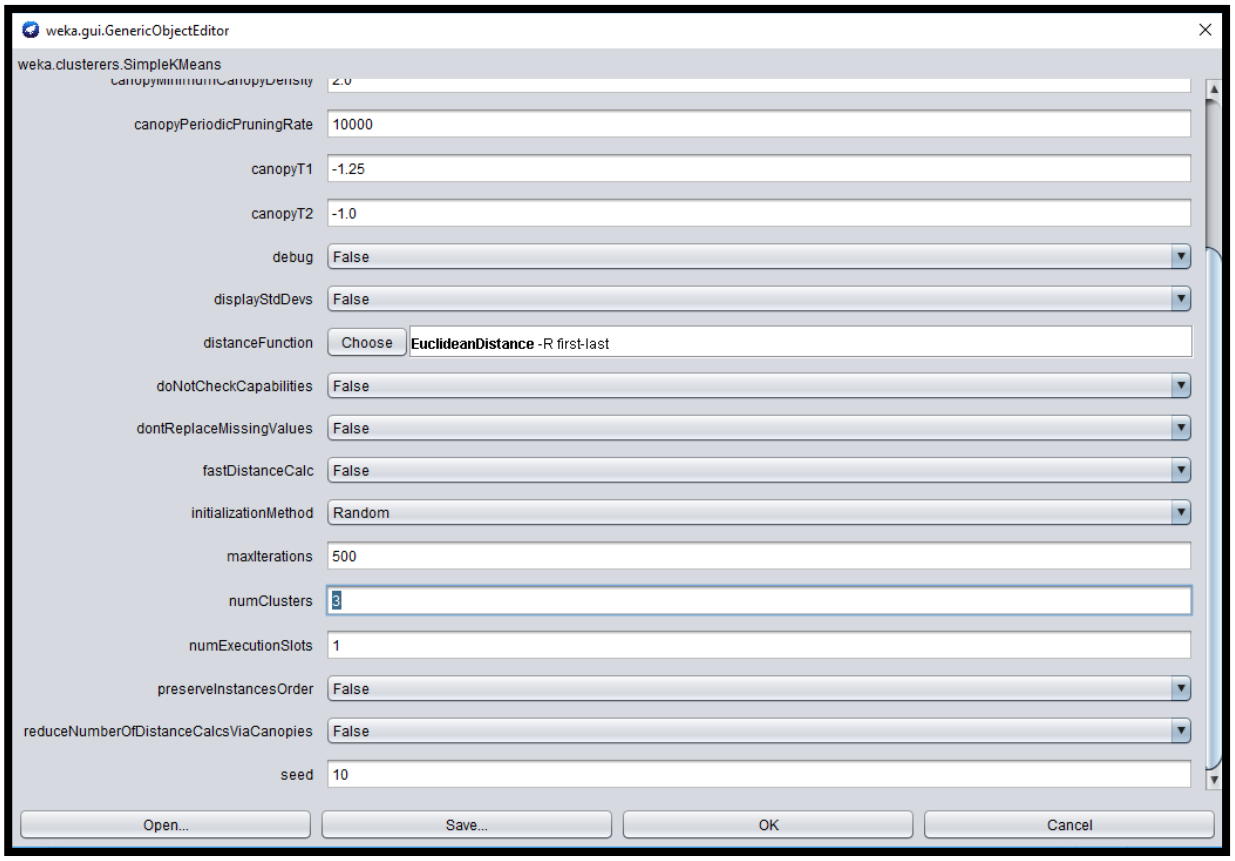


Figura 19: Configurações de K Means

Nas opções do Cluster mode, selecione Classes to cluster evaluation, e escolha (Nom) class, pois isso fará com que o agrupamento seja medido em relação a classe. Por fim clique em Start e observe a saída no Clusterer output.

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 100
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
Ignored:
              class
Test mode:    Classes to clusters evaluation on training data

=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 6.998114004826762

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4
Cluster 1: 6.2,2.9,4.3,1.3
Cluster 2: 6.9,3.1,5.1,2.3

Missing values globally replaced with mean/mode

Final cluster centroids:
                           Cluster#
Attribute      Full Data          0          1          2
                 (150.0)     (61.0)     (50.0)     (39.0)
============================================================
sepallength       5.8433     5.8885      5.006     6.8462
sepalwidth         3.054     2.7377      3.418     3.0821
petallength       3.7587     4.3967      1.464     5.7026
petalwidth        1.1987      1.418      0.244     2.0795



Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       61 ( 41%)
1       50 ( 33%)
2       39 ( 26%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 47  0  3 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :       17.0     11.3333 %
```

Figura 20: Saída do agrupamento

A base escolhida de dados Iris, possui 4 atributos: sepallength, sepalwidth, petallength, petalwidth e um atributo class com três classes, Iris-setosa, Iris-versicolor e Iris-virginica. Na saída, a parte de Clustered Instances, é possível observar três clusters criados e a quantidade de instâncias que foram inseridas em cada cluster, dessa maneira, depois da matriz de confusão, o Weka coloca cada cluster a uma espécie de Iris, fundamentado na maior frequência desta em cada grupo. É possível ver no final que foram 17 instâncias foram agrupadas incorretamente, o que representa mais ou menos 11%.

É possível também observar essa saída de forma gráfica, para isso, clique com o botão direto sobre o SimpleKMeans criado no Result List e no menu selecionar Visualize cluster assignments e assim será exibido o gráfico como na Figura 21.

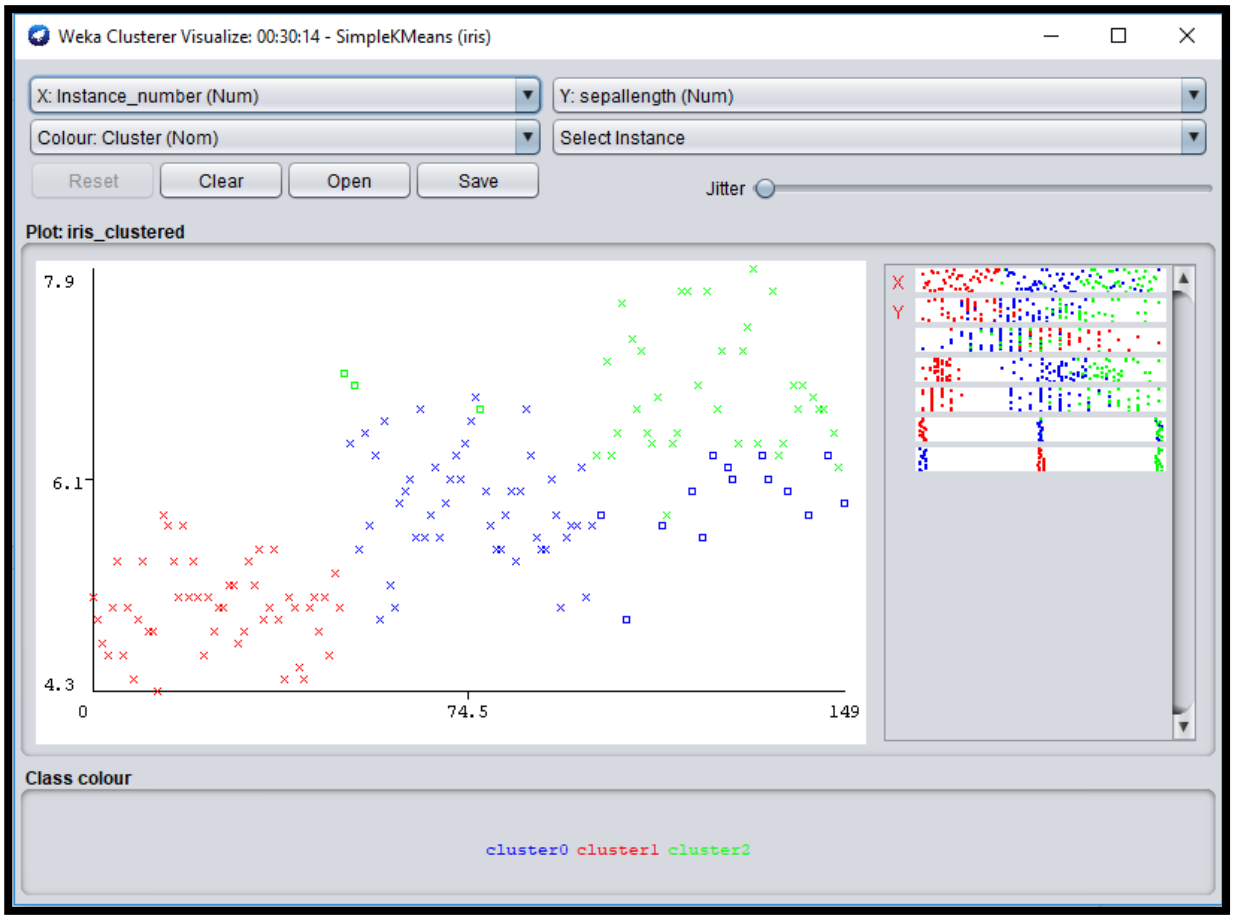


Figura 21: Visualização do Cluster

5.4. ASSOCIADOR COM APRIORI

Para demonstração será utilizado a base de dados weather.nominal para encontrar uma relação entre os atributos.

Para começar, carregue o weather.nominal.arff e depois vá para guia Associate e selecione o algoritmo Apriori. Caso deseje realizar algumas mudanças nos parâmetros, basta clicar no nome do algoritmo e abrirá a Janela Generic Object Editor, nesse exemplo serão mantidos os valores padrões.

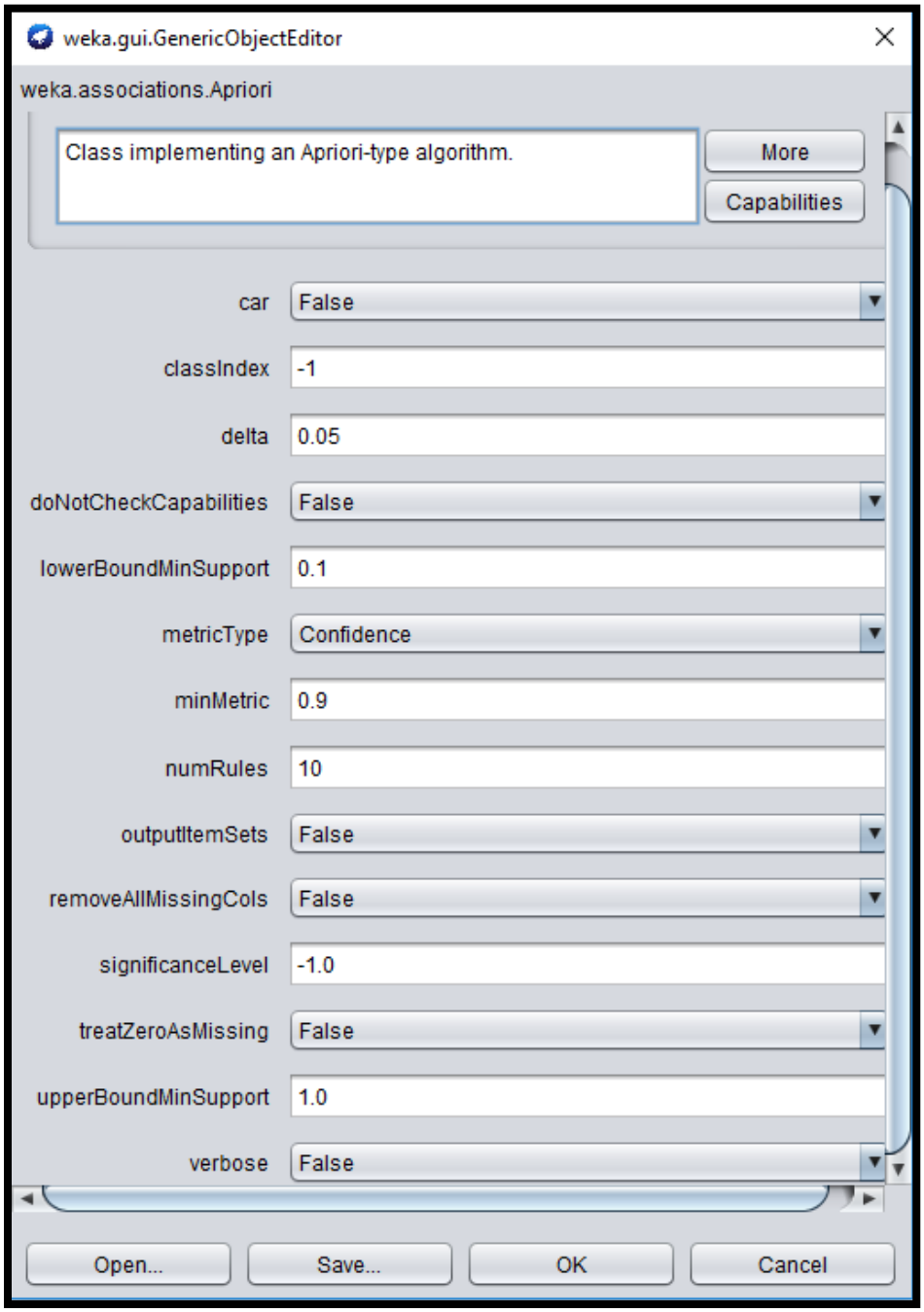


Figura 22: Generic Object Editor

No final apenas clique em Start e o Weka irá produzir a seguinte saída:

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. outlook=sunny temperature=hot 2 ==> humidity=high 2    <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2    <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```

Figura 23: Saída Associadores

No começo são mostrados dados sobre a relação e o associador utilizado. O mais interessante é a sessão final chamada Best rules found, que é o conjunto de regras mineradas. Basicamente as regras antecedentes geram uma consequência, por exemplo, na regra 1 se o clima estiver apenas nublado a consequência de jogar é sim. Da mesma forma o contrário também é mostrado, ou seja, na regra 10, por exemplo,

o antecedente é que a temperatura está calor e jogar é não, com isso, é possível dizer que o clima vai estar ensolarado devido as associações feitas pelo algoritmo.

6. CONCLUSÃO

Esse trabalho teve como objetivo realizar uma pesquisa sobre mineração de dados abordando primeiramente, o processo maior, no qual ela está inserida, conhecido como Descoberta de Conhecimento em Banco de Dados e analisar técnicas e algoritmos além da demonstração de uma ferramenta conhecida para a mineração de dados.

Atualmente uma enorme quantidade de dados variados tem sido produzida em alta velocidade principalmente via internet a partir de mídias sócias, e-mails, vídeos entre outros. Com esse número elevado de dados se torne impossível uma análise simplesmente humana, se fazendo necessário o uso da tecnologia.

A mineração de dados é o processo responsável por analisar um grande conjunto de dados à procura de relações, padrões, sequencias, anomalias e a partir disso gerar informações uteis em diversas áreas como empresarial, cientifica entre outras. O campo da mineração de dados é muito amplo pois as informações buscadas em um conjunto de dados podem ser variadas, dependendo apenas do interesse que o usuário tem sobre os dados. Existem diversas técnicas e algoritmos que tem funções diferentes, por exemplo, o usuário pode realizar uma classificação dos dados de um determinando conjunto ou se ele proferir é possível fazer uma associação entre os dados para encontras padrões, sendo assim, a mineração de dados abrange todas essas possibilidades.

Existem diversas ferramentas que possibilitam o uso dos algoritmos de mineração de dados sendo possível aplica-los em diversos tipos de conjuntos de dados gerando diversos informações. A ferramenta escolhida para esse trabalho foi o Weka que é um software open-source desenvolvido em Java que possui uma vasta biblioteca com vários algoritmos e até mesmo conjuntos de dados que podem ser utilizados para testes.

O uso da ferramenta WEKA se mostrou simples apesar do se fazer necessário um conhecimento técnico prévio para sua utilização. Sua interface é simples e se mostrou eficaz no comprimento das tarefas além de possuir diversas ferramentas tanto de pré-processamento quanto para visualização dos resultados, onde é possível observar as

informações geradas escritas ou em forma gráfica, sendo totalmente indicada a sua utilização.

7. REFERÊNCIAS BIBLIOGRÁFICAS

AGRAWAL, R., & SRIKANT, R. **Fast algorithms for mining association rules**. In *Proc. 20th int. conf. very large data bases, VLDB,* Vol. 1215, pp. 487-499, Setembro, 1994.

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. **Mining Association Rules beween Sets of Items in Large Databases**. Proceedings of the ACM SIGMOD Conference. Washington, USA, Maio, 1993.

ALMEIDA, Felipe Schaedler de. **Otimização de estruturas de materiais compósitos laminados utilizando algoritmos genéticos**. Dissertação de Mestrado, PPGEC/UFRGS, Porto Alegre, 2006.

BRACHMAN, R. J.; ANAND, T. **The Process of Knowledge Discovery in Databases**. The KDD Process for Extracting Useful Knowledge from Volumes of Data, 1996, p. 37-57.

BERRY J. A. Michael; LINOFF Gordon; **Data Mining Techiques for Marketing, Sales, and Customer Support.** Jonh Wiley & Sons, Inc., 1997.

CABENA, P., HADJINIAN, P., STADLER, R., VERHEES, J., e ZANASI, A. **Discovering data mining: from concept to implementation**. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

CHAVES, Adriana da Costa Ferreira. **Extração de Regras Fuzzy para Máquinas de Vetores Suporte (SVM) para Classificação em Múltiplas Classes.** Rio de Janeiro, 2006.

COSTA, Evandro; BAKER, Ryan S.; AMORIN, Lucas; MAGALHÃES, Jonathas; MARINHO, Tarsis. **Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações**. *Jornada de Atualização em Informática na Educação* 1.1 (2013): 1-29.

DIAS, Maria M. **Um Modelo de Formalização do Processo de Desenvolvimento de Sistemas de Descoberta de Conhecimento em Banco de Dados**. 2001. 212 f. Tese (Doutorado em Engenharia da Produção) – Universidade Federal de Santa Catarina – UFSC, Florianópolis, 2001.

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.

FERNANDES, A. M. R. **Inteligência Artificial: noções gerais.** Ed. Visual Books, Florianópolis, 2003.

FREITAS, Alex A. **A Review of Evolutionary Algorithms for Data Mining**. Canterbury, 2007.

FONSECA, Oswaldo L. H. **Análise de crédito utilizando inteligência artificial - validação com dados do cartão BNDES.** 2008. 143 f. Tese (Doutorado) - Universidade do Estado do Rio de Janeiro, Instituto Politécnico, Nova Friburgo, 2008.

GOLDSCHMIDT, R. R.; PASSOS, E. **Data Mining: Um Guia Prático**. Rio de Janeiro: Elsevier, 2005.

HALL, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). **The weka data mining software: an update**. SIGKDD Explor. Newsl., 11(1):10–18.

HAN, J. and Kamber, M. **Data mining: concepts and techniques**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

HAN, J; KAMBER, M. **Data Mining: Concepts and Techniques**. Elsevier, 2006.

HAYKIN, S.: **Redes Neurais: Princípios e Prática. 2.ed**. Porto Alegre, Bookman, 2001.

JUNIOR, E. P. F. (2008). **Estudo comparativo entre algoritmos de regras de associação de forma normal e incremental de dados** (Doctoral dissertation, Pontifícia Universidade Católica do Paraná).

LACERDA, M.P.; SOUZA, R.C.F. **Aplicação da Mineração de Dados em Sistema de Avaliação de professor e aluno.** Monografia (conclusão de curso). Universidade Federal do Pará. Belém, 2004.

LOPES, P. D. A. **Agrupamento de Dados Semissupervisionado na Geração de Regras Fuzzy**. Monografia (Mestrado). Universidade Federal de São Carlos. São Carlos, 2010.

LEVY, E. - **The Lowndown on Data Mining**. Teradatareview, Summer, 1999.

MECHIE, D.; SPIEGELHALTER, D.; TAYLOR, C. **Machine Learning, Neural and Statistical Classifications.** Ellis Horwood, 1994.

MITCHELL, Melanie. **An introduction to genetic algorithms**. Cambridge. MIT Press. 1997, 207p.

NIMER, F; SPANDRI, L.C. **Data Mining.** Revista Developers. v.7, fevereiro, 1998. p.32.

PASTA, Arquelau. **Aplicação da técnica de Data Mining na base de dados do ambiente de gestão educacional: um estudo de caso de uma instituição de ensino superior de Blumenau - SC**. Dissertação de Mestrado, UNIVALI, São José, 2011.

PERERA, L.C.J.; KIMURA, Herbert; HORTA, R.A.M.; LIMA, F.G.; **Uma Análise em Data Mining: Árvores de Decisão, Redes Neurais e Support Vector Machines**. In: Encontro da ANPAD. Rio de Janeiro. Anais do XXXV Encontro da ANPAD, v.1, setembro, 2011

POSSA, B. A. V.; CARVALHO, M. L. B. De; REZENDE, R.S.F.; MEIRA JR., W. **Data Mining: Técnicas para Exploração de Dados**. Universidade Federal de Minas Gerais, 1998.

QUINLAN, J. R. C4.5: **Programs for Machine Learning**. Morgan Kaufmann Publishers, 1993.

REZENDE, S.O. (Coord.), **Sistemas Inteligentes: Fundamentos e Aplicações**. Barueri, SP, Brasil, Rezende, S.O., 2003.Editora Manole.

SRIVASTAVA S. **Weka: A Tool for Data preprocessing, Classification, Ensemble, Clustering and Association Rule Mining**. International Journal of Computer Applications (0975 – 8887), v.88, n.10, fevereiro, 2014.

SEGARAN, T. (2007). **Programming collective intelligence**. O'Reilly, first edition

SYACHRANI, Syadaruddin; JEONG, Hyung Seok "David"; CHUNG, Colin S. **Decision tree–based deterioration model for buried wastewater pipelines**. Journal of Performance of Constructed Facilities, v. 27, n. 5, p. 633-645, 2012.

WANGENHEIM, Christiane Gresse von. WANGENHEIM, Aldo von. **Raciocínio Baseado em Casos.** Barueri, São Paulo: Manole, 2003. 293p.

WEIS, Sholom M., INDURKHYA Nitim. **Predict Data Mining**. Morgan Kaufmann Publishers, Inc, 1999.

ZAKI, M. J. **Parallel and Distributed Data Mining: An Introduction**. Large-Scale Parallel Data Mining. Spinger-Verlag Berlin Heidelberg, 2000.