Capítulo 2 - Mineração de Dados e Regras de Associação

Introdução

A Mineração de Dados é uma disciplina tão vasta que qualquer publicação sobre o tema obriga o autor a selecionar alguns tópicos em detrimento de outros não menos importantes. A atividade de **Regras de Associação** foi o tópico escolhido para iniciarmos a apresentação das principais tarefas da Mineração de Dados por envolver ideias bem intuitivas. A analogia entre Regras de Associação e Cesta de Compras facilita o entendimento de como descobrir padrões de associação entre itens de um conjunto qualquer.

Mineração de Dados

Durante o processo de **Descoberta de Conhecimento em Bases de Dados**, **KDD**, é na etapa de **Mineração de Dados** que efetivamente são encontrados os padrões de associação implícitos nos dados. A análise automatizada dessa massa de dados visa detectar regularidades, ou quebra de regularidade, que constitui informação implícita, porém supostamente desconhecida, e útil para determinado fim.

A Mineração de Dados pode ser vista como a sistematização de teorias, técnicas e algoritmos desenvolvidos em outras disciplinas já consagradas, como a Estatística, a Inteligência Artificial, o Aprendizado de Máquina, a Base de Dados etc. (Figura 2.1). O propósito da Mineração de Dados é detectar automaticamente padrões de associação úteis e não óbvios em grandes quantidades de dados.

No dia a dia, uma quantidade incalculável de dados é gerada na forma de registros de vendas, textos brutos, imagens, sons, gráficos etc., tanto por sistemas computacionais como por seres humanos, constituindo uma espécie de informação não estruturada. Embora esta forma de registro de dados seja adequada para o ser humano, quando se trata de analisar grandes quantidades de dados de forma automatizada, é comum e conveniente que se introduza alguma **estrutura** que facilite o acesso e o processamento sistemático.

Figura 2.1– A Relação da Mineração de Dados com Algumas Disciplinas Correlatas.

Para que parte de toda essa informação não estruturada possa ser utilizada na Mineração de Dados, geralmente é feita uma seleção e um pré-processamento visando transformar dados brutos em coleções estruturadas de dados. Em termos práticos, para nossas considerações iniciais, a estrutura de representação de uma Base de Dados pode ser semelhante a uma tabela de dados, sendo cada linha dessa tabela uma **transação** ou um **exemplo**. Cada **transação** é composta por um ou mais **itens** ou, visto de outra forma, cada **exemplo** é caracterizado por seus **atributos**.

As Tabelas 2.1, 2.2 e 2.3 ilustram formas de dados estruturados convenientes para a Mineração de Dados.

Tabela 2.1 – Cestas de Compras.

TID	Itens
1	{Arroz, Feijão, Óleo}
2	{Queijo, Vinho}
3	{Arroz, Feijão, Batata, Óleo}
4	{Arroz, Água, Queijo, Vinho}
5	{Arroz, Feijão, Batata, Óleo}

Na Tabela 2.1 cada uma das Transações possui uma IDentificação (TID), e seus itens representam artigos vendidos em um supermercado. Se a tabela de itens for muito extensa, como costuma ser em casos reais, pode ser ainda mais

conveniente representar cada um de seus itens na forma de um atributo associado a um valor booleano, como mostra a Tabela 2.2.

Tabela 2.2 – Representação Booleana de Cestas de Compras.

TID	Arroz	Feijão	Batata	Óleo	Água	Queijo	Vinho
1	y	y	n	y	n	n	n
2	n	n	n	n	n	y	y
3	y	y	y	y	n	n	n
4	y	n	n	n	y	y	y
5	y	y	y	y	n	n	n

Um exemplo clássico de uma Base de Dados usada em artigos sobre Mineração de Dados é apresentada na Tabela 2.3 (QUINLAN, 1986, *apud* WITTEN & FRANK, 2005), composta por dados fictícios sobre as condições de tempo para que ocorra ou não a partida de um esporte não especificado. A tabela é composta por 14 exemplos (linhas), cada um com cinco atributos (colunas): Dia, Temperatura, Umidade, Vento e Partida. A tabela pode também ser interpretada de outra forma, como sendo composta por quatro atributos (Dia, Temperatura, Umidade e Vento) e uma classe (Partida), que representa o resultado da combinação dos quatro atributos.

Tabela 2.3 – Tabela do Tempo.

Dia	Temperatura	Umidade	Vento	Partida
Ensolarado	Elevada	Alta	Falso	Não
Ensolarado	Elevada	Alta	Verdadeiro	Não
Nublado	Elevada	Alta	Falso	Sim
Chuvoso	Amena	Alta	Falso	Sim
Chuvoso	Baixa	Normal	Falso	Sim
Chuvoso	Baixa	Normal	Verdadeiro	Não
Nublado	Baixa	Normal	Verdadeiro	Sim
Ensolarado	Amena	Alta	Falso	Não
Ensolarado	Baixa	Normal	Falso	Sim
Chuvoso	Amena	Normal	Falso	Sim
Ensolarado	Amena	Normal	Verdadeiro	Sim
Nublado	Amena	Alta	Verdadeiro	Sim
Nublado	Elevada	Normal	Falso	Sim
Chuvoso	Amena	Alta	Verdadeiro	Não

Uma análise mais atenta dos exemplos das Tabelas 2.1 e 2.3 mostra que alguns desses atributos sempre aparecem juntos e que, portanto, várias **Regras de Associação** podem ser extraídas dessas tabelas.

Regras de Associação

A representação do conhecimento através de regras, também conhecidas como regras *IF-THEN* ou Regras de Produção, é largamente utilizada porque, entre outras vantagens sobre formas alternativas de representação do conhecimento, regras são facilmente compreendidas pelo ser humano, fáceis de serem alteradas, validadas e verificadas, e de baixo custo para a criação de sistemas baseados em regras (PADHY, N. P., 2010).

Regras de Associação são uma forma específica de representação de conhecimento que descrevem padrões de associação implícitos entre um conjunto de atributos ou itens de uma Base de Dados, e que podem ajudar a predizer com alta probabilidade a presença, ou não, de outro conjunto de atributos ou itens.

Dito de forma equivalente, uma Regra de Associação revela que a presença de um conjunto **X** de itens numa transação implica outro conjunto **Y** de itens, i.e., **X** = {a, b, ...} $\Rightarrow$ **Y** = {p, ..., z}. Note que o fato de um conjunto de itens **X** (antecedente) estar sempre associado a outro **Y** (consequente) não significa obrigatoriamente que um seja a causa de outro, i.e., não há necessariamente relação de causalidade entre antecedente e consequente e sim mera ocorrência simultânea de itens com certa probabilidade.

A estrutura geral de uma Regra de Associação assume a seguinte forma:

If (Conjunto **X** de Itens) ***then*** (Conjunto **Y** de Itens), sendo $\mathbf{X} \cap \mathbf{Y} = \emptyset$.

Com base na Figura 2.3, várias Regras de Associação podem ser formuladas:

```
If (Temperatura=Baixa) then (Umidade=Normal)                                    (2.1)

If (Umidade=Normal) and (Vento=Falso) then (Partida=Sim)                        (2.2)

If (Dia=Ensolarado) and (Partida=Não) then (Umidade=Alta)                       (2.3)

If (Vento=Falso) and (Partida=Não) then (Temperatura=Elevada) and
(Umidade=Alta)                                                                  (2.4)
```

Estas são apenas algumas das muitas Regras de Associação que podem ser formuladas com base na Tabela 2.3. Para selecionar as Regras de Associação mais representativas, i.e., aquelas que se apliquem a um grande número de exemplos com alta probabilidade de acerto, precisaremos de métricas para avaliar o alcance ou a força de cada regra. Dois dos mais conhecidos indicadores são **Suporte** e **Confiança**.

Suporte – para cada regra do tipo **X ⇒ Y**, este parâmetro indica a quantos exemplos da tabela esta regra satisfaz (i.e., contém) tanto ao conjunto de itens de **X** quanto ao de **Y**, ou seja, indica sua **cobertura** com relação ao número total ***N*** de exemplos da tabela. Portanto,

$$\boldsymbol{Sup(X \rightarrow Y) = \frac{|X \cup Y|}{N}}$$

Por exemplo, com relação à primeira regra (2.1) há quatro exemplos na Tabela 2.3 em que {**X** ∪ **Y**} = {Temperatura=Baixa, Umidade=Normal}. Portanto,

$$Sup(Regra\ 2.1) = \frac{|X\ U\ Y|}{N} = \frac{|\{Temperatura = Baixa, Umidade = Normal\}|}{N} = \frac{4}{14} = 0{,}29$$

A Regra 2.2 também tem *Sup(Regra 2.2) = 4/14*, a terceira regra tem *Sup(Regra 2.3) = 3/14*, enquanto que a quarta regra tem *Sup(Regra 2.4) = 1/14*.

Confiança – a confiança de uma regra reflete o número de exemplos que contêm **Y** dentre todos aqueles que contêm **X** (veja bem, além de **X ⇒ Y,** podem existir regras do tipo **X ⇒ Z**, **X ⇒ W** etc.). Em outras palavras, o parâmetro Confiança determina quantos são os exemplos em que **X** implica **Y**, comparado com aqueles exemplos em que **X** pode ou não implicar **Y**. A este parâmetro costuma-se também dar o nome de **Acurácia**.

$$\boldsymbol{Conf(X \rightarrow Y) = \frac{|X \cup Y|}{|X|} = \frac{Sup(X=>Y)}{Sup(X)}}$$

Por exemplo, com relação à primeira Regra (2.1) há quatro exemplos na Tabela 2.3 em que {**X** ∪ **Y**} = {Temperatura=Baixa, Umidade=Normal} e, coincidentemente, quatro exemplos em que {**X**} = {Temperatura=Baixa). Portanto,

$$Conf(Regra\ 2.1) = \frac{|X\ U\ Y|}{|X|} = \frac{|\{Temperatura = Baixa, Umidade = Normal\}|}{|\{Temperatura = Baixa\}|} = \frac{4}{4} = 1{,}0$$

A Regra 2.2 também tem *Conf(Regra 2.2) = 4/4*, a terceira regra tem *Conf(regra 2.3) = 3/3*, enquanto que a quarta regra tem *Conf(Regra 2.4) = 1/2*.

Regras de Associação são particularmente úteis para analisar o comportamento de clientes e propor "vendas casadas". A informação de que clientes que compram o item A geralmente compram o item B pode aumentar significativamente as vendas de uma loja ou livraria, já que toda vez que um cliente manifestar a intenção de comprar o item A, a loja pode também lhe oferecer o item B.

Mas o fato de um simples conjunto de itens poder gerar muitas regras de associação faz com que o número de regras associadas a uma base de dados seja tão grande a ponto de a maioria dessas regras não ter qualquer interesse prático. Para contornar esta situação, antes de começar a gerar as regras de associação, é comum que sejam estabelecidos um valor de Suporte Mínimo (**SupMin**) e de Confiança Mínima (**ConfMin**). Regras com suporte muito baixo podem ser resultado de compras feitas ao acaso e, portanto, não fornecem informações de interesse. Por outro lado, regras com confiança muito baixa podem indicar que seu poder de predição é baixo e, portanto, não é muito aconselhável assumir que **X** implica **Y** com base nessas regras.

Agrawal (AGRAWAL *et al.*, 1993) ao introduzir o conceito de Regras de Associação propôs um algoritmo denominado **Apriori** no qual Regras de Associação são geradas em duas etapas:

- Dado um conjunto de transações **T**, primeiramente são criados conjuntos de itens frequentes, chamados de **Conjuntos Frequentes**, que devem satisfazer o limite de **SupMin**;
- a partir desses Conjuntos Frequentes são geradas **Regras de Associação** com confiança maior ou igual **ConfMin**.

Etapa 1: Geração de Conjuntos Frequentes com Suporte ≥ SupMin

As Tabelas 2.4 e 2.5 mostram versões simplificadas da Tabela 2.2, aqui adaptada para que cada item possa ser representado por apenas uma letra.

Tabela 2.4 – Versão Simplificada da Tabela 2.2.

TID	A	B	C	D	E	F	G
1	1	1	0	1	0	0	0
2	0	0	0	0	0	1	1
3	1	1	1	1	0	0	0
4	1	0	0	0	1	1	1
5	1	1	1	1	0	0	0

Tabela 2.5 – Versão Alternativa da Tabela 2.2.

TID	Itens
1	{A, B, D}
2	{F, G}
3	{A, B, C, D}
4	{A, E, F, G}
5	{A, B, C, D}

De acordo com o algoritmo Apriori, para se obter os possíveis Conjuntos Frequentes relacionados a um conjunto de transações, inicialmente devem ser criados Conjuntos Frequentes com 1 item apenas e que satisfaçam o critério de Suporte Mínimo. A seguir são criados recursivamente Conjuntos Frequentes com 2 itens, depois com 3 itens, e assim sucessivamente.

Os possíveis Conjuntos Frequentes com 1 item apenas, e seus respectivos valores de Suporte, estão representados na Tabela 2.6.

Tabela 2.6 – Possíveis Conjuntos Frequentes com 1 Item.

Itens	Suporte
{A}	4/5
{B}	3/5
{C}	2/5
{D}	3/5
{E}	1/5
{F}	2/5
{G}	2/5

Suponhamos que o **SupMin** tenha sido definido como 2/5, ou seja, 40%. De acordo com este critério, o conjunto {E} não satisfaz **SupMin** e deve ser eliminado. Portanto os Conjuntos Frequentes com 1 Item que satisfazem o critério de **SupMin** maior ou igual a 2/5 estão representados na Tabela 2.7.

Ao adotar o procedimento de poda dos candidatos a Conjunto Frequente que não satisfazem o critério de SupMim, o número total de Conjuntos Frequentes gerados pode cair significativamente. Em princípio, dada uma Base de Dados com *k* itens, o número de possíveis Conjuntos Frequentes é $|CF| = 2^k-1$ (excluindo o conjunto vazio). Como em nossa Tabela 2.4 há 7 itens, $|CF| = 2^7-1 = 127$.

Tabela 2.7 – Conjuntos Frequentes com 1 Item e SupMin ≥ 2/5.

Itens	Suporte
{A}	4/5
{B}	3/5
{C}	2/5
{D}	3/5
{F}	2/5
{G}	2/5

A seguir devem ser formados novos Conjuntos Frequentes com 2 Itens, partindo-se dos Conjuntos Frequentes com 1 Item. Note que o Suporte de um Conjunto Frequente com 2 Itens pode ter no máximo o menor valor de Suporte de cada um de seus subconjuntos, i. e., dos respectivos Conjuntos Frequentes com 1 Item. De acordo com esta mesma propriedade, conhecida como **Princípio Apriori** ou antimonotônico, qualquer subconjunto de um Conjunto Frequente também será um Conjunto Frequente. Por exemplo, se {A, B} for um Conjunto Frequente, então {A} e {B} também são Conjuntos Frequentes e têm Suporte ≥ SupMin.

A Tabela 2.8 mostra os possíveis Conjuntos Frequentes com 2 Itens e os respectivos valores de Suporte.

Os conjuntos de 2 itens foram obtidos por combinação dos conjuntos de 1 item, enquanto que os valores de Suporte foram obtidos inspecionando-se as Tabelas 2.4 e 2.5.

Note que os detalhes de como os conjuntos de itens são efetivamente gerados dependem da forma como o algoritmo Apriori foi implementado, e diferem um pouco da exposição simplificada que se adotou aqui por razões didáticas.

Para calcular o Suporte dos Conjuntos Frequentes foi necessário ler a Base de Dados, que em nosso caso é pequena e está representada pela Tabela 2.4. Em uma implementação computacional é altamente desejável que toda a Base de Dados possa ser lida na memória principal do computador. Porém, se a Base de Dados for muito grande ela provavelmente terá de ser lida no disco rígido. Para minimizar o número de vezes que a Base de Dados é consultada, muitos candidatos a Conjuntos Frequentes podem ser inicialmente criados e depois eliminados, obtendo-se assim significativos ganhos de tempo.

Tabela 2.8 – Possíveis Conjuntos Frequentes com 2 Itens.

Itens	Suporte
{A, B}	3/5
{A, C}	2/5
{A, D}	3/5
{A, F}	1/5
{A, G}	1/5
{B, C}	2/5
{B, D}	3/5
{B, F}	0
{B, G}	0
{C, D}	2/5
{C, F}	0
{C, G}	0
{D, F}	0
{D, G}	0
{F, G}	2/5

Para nós, neste momento, o importante é compreender como os Conjuntos Frequentes com ***k*** Itens podem ser gerados de forma relativamente simples pela combinação de Conjuntos Frequentes com ***k-1*** Itens.

Aplicando-se novamente o critério de **SupMin ≥ 2/5**, restam apenas os Conjuntos Frequentes com 2 Itens apresentados na Tabela 2.9.

Tabela 2.9 – Conjuntos Frequentes com 2 Itens e SupMin ≥ 2/5.

Itens	Suporte
{A, B}	3/5
{A, C}	2/5
{A, D}	3/5
{B, C}	2/5
{B, D}	3/5
{C, D}	2/5
{F, G}	2/5

O próximo passo agora consiste em criar novos Conjuntos Frequentes com 3 Itens, partindo-se dos Conjuntos Frequentes com 2 Itens, cujo resultado é mostrado na Tabela 2.10.

Tabela 2.10 – Possíveis Conjuntos Frequentes com 3 Itens.

Itens	Suporte
{A, B, C}	2/5
{A, B, D}	3/5
{A, C, D}	2/5
{A, F, G}	1/5
{B, C, D}	2/5
{B, F, G}	0
{C, D, F}	0
{C, F, G}	0

Neste caso, novamente, alguns Conjuntos Frequentes não satisfazem o critério do **SupMin ≥ 2/5**, havendo a necessidade de poda para que estes conjuntos não participem da etapa seguinte.

Tabela 2.11 – Conjuntos Frequentes com 3 Itens e SupMin ≥ 2/5.

Itens	Suporte
{A, B, C}	2/5
{A, B, D}	3/5
{A, C, D}	2/5
{B, C, D}	2/5

Vamos agora gerar novos Conjuntos Frequentes com 4 Itens, partindo-se dos Conjuntos Frequentes com 3 Itens (Tabela 2.11), cujo resultado é mostrado na Tabela 2.12.

Tabela 2.12 – Possíveis Conjuntos Frequentes com 4 Itens.

Itens	Suporte
{A, B, C, D}	2/5

Se houvesse ao menos dois Conjuntos Frequentes com 4 Itens poderíamos ainda tentar gerar Conjuntos Frequentes com 5 Itens. Mas como há apenas um Conjunto Frequente com 4 Itens, esta primeira etapa do algoritmo Apriori termina aqui.

Etapa 2: Geração de Regra de Associação a partir dos Conjuntos Frequentes

Uma vez obtidos os Conjuntos Frequentes com Suporte ≥ SupMin, é possível extrair de cada Conjunto Frequente com *k* itens 2^k-2 Regras de Associação (excluindo o conjunto vazio na posição de antecedente (∅⇒CF) ou de consequente(CF⇒∅)). Na Etapa 1 foram gerados os seguintes Conjuntos Frequentes:

Conjuntos Frequentes com 1 Item (total de 6 CFs)

{A}, {B}, {C}, {D}, {F}, {G}

Conjuntos Frequentes com 2 Itens (total de 7 CFs)

{A, B}, {A, C}, {A, D}, {B, C}, {B, D}, {C, D}, {F, G}

Conjuntos Frequentes com 3 Itens (total de 4 CFs)

{A, B, C}, {A, B, D}, {A, C, D}, {B, C, D}

Conjunto Frequente com 4 Itens (total de 1 CF)

{A, B, C, D}

Para extrair as Regras de Associação de um Conjunto Frequente é necessário primeiramente gerar todos os subconjuntos não-vazios desse Conjunto Frequente **CF**, e para cada subconjunto **S** de CF produzir uma Regra de Associação do tipo **S ⇒ (CF - S)** que satisfaça o critério de Confiança ≥ ConfMin.

Por exemplo, dado o CF = {A, B, C}, seus subconjuntos não-vazios possíveis são S = {{A}, {B}, {C}, {A, B}, {A, C}, {B, C}. Portanto, é possível extrair seis Regras de Associação do CF = {A, B, C} que envolvam os três itens:

{A} ⇒ {B, C},

{B} ⇒ {A, C},

{C} ⇒ {A, B},

{A, B} ⇒ {C},

{A, C} ⇒ {B},

{B, C} ⇒ {A}.

Como o Suporte de todos os subconjuntos já terá sido calculado na Etapa 1, não será necessário percorrer novamente a Base de Dados para calcular a Confiança de cada Regra de Associação. Basta reutilizar estes valores calculados, pois

$$Conf(S \rightarrow CF - S) = \frac{|S \cup CF - S|}{|S|} = \frac{|CF|}{|S|} = \frac{Sup(CF)}{Sup(S)}$$

Voltando ao exemplo inicial da Tabela 2.5, como o Suporte de CF = {A, B, C} é 2/5 (veja Tabela 2.11), e com os Suportes de seus subconjuntos

Sup({A}) = 4/5 (veja Tabela 2.6),

Sup({B}) = 3/5 (veja Tabela 2.6),

Sup({C}) = 2/5 (veja Tabela 2.6),

Sup({A, B}) = 3/5 (veja Tabela 2.9),

Sup({A, C}) = 2/5 (veja Tabela 2.9), e

Sup({B, C}) = 2/5 (veja Tabela 2.9),

a Confiança de cada uma das seis regras possíveis será:

***Conf*(A ⇒ B, C) =** (2/5) / (4/5) = **0,50**

***Conf*(B ⇒ A, C) =** (2/5) / (3/5) = **0,66**

***Conf*(C ⇒ A, B) =** (2/5) / (2/5) = **1,00**

***Conf*(A, B ⇒ C) =** (2/5) / (3/5) = **0,66**

***Conf*(A, C ⇒ B) =** (2/5) / (2/5) = **1,00**

***Conf*(B, C ⇒ A) =** (2/5) / (2/5) = **1,00**

Suponha que para o problema em questão tenha sido adotado **SupMin = 40%** e **ConfMin = 90%**, então apenas três das regras acimas seriam aproveitadas:

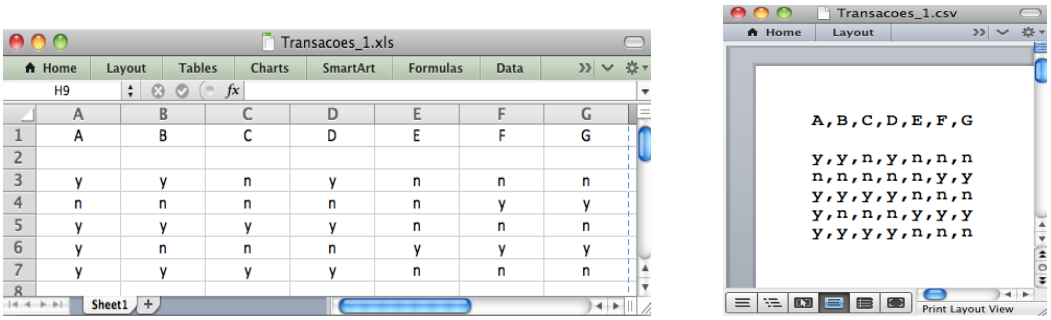
***Conf*(C ⇒ A, B) = 1,00**	**{Batata} ⇒ {Arroz, Feijão}**
***Conf*(A, C ⇒ B) = 1,00**	**{Arroz, Batata} ⇒ {Feijão}**
***Conf*(B, C ⇒ A) = 1,00**	**{Feijão, Batata} ⇒ {Arroz}**

Aplicando-se o procedimento explicado acima para todos os 18 CFs obtidos na Etapa 1, seriam geradas aproximadamente 30 Regras de Associação com SupMin = 40% e ConfMin = 90% (na realidade, chegamos ao número 30 através de simulação no Weka, como será mostrado na Atividade Prática com o Weka).

Como Gerar Regras de Associação Usando a Ferramenta Weka

Nesta seção será apresentado um pequeno tutorial sobre a geração de Regras de Associação usando o algoritmo "Apriori" implementado na ferramenta de Aprendizado de Máquina para tarefas de Mineração de Dados Weka (Weka, 2013). A versão utilizada é a 3.6.7. Para fazer uma simulação no Weka, a Base de Dados terá de ser escrita ou no formato CSV *(Comma-Separated Value)* (".csv") ou no formato "ARFF" *(Attribute-Relation File Format)*, um formato bastante simples e intuitivo dessa ferramenta. Com o arquivo ".arff" carregado, podemos ajustar os parâmetros Suporte e Confiança e rodar o algoritmo Apriori.

Passo 1 - Vamos supor que nossa Base de Dados tenha sido retirada de uma planilha eletrônica (".xls") e salva no formato ".csv", como mostra a Figura 2.2.



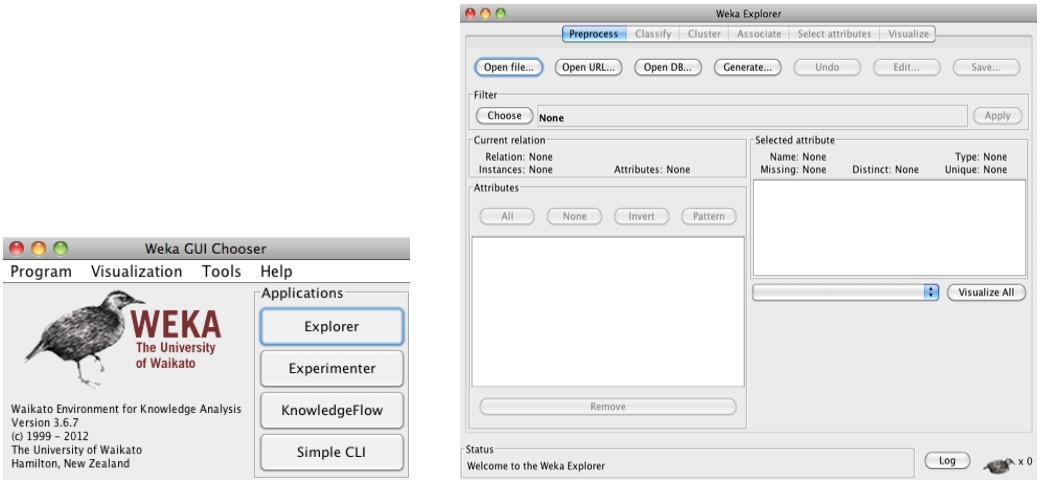
(a) **(b)**

Figura 2.2 – A Base de Dados Transacoes_1 na Forma (a) Planilha ".xls" e (b) ".csv".

Como o Weka tem um conversor interno do formato ".csv" para ".arff", vamos primeiramente usar este recurso. Depois vamos mostrar como transformar manualmente o arquivo ".csv" para ".arff".

Obs.: Certifique-se que em seu arquivo ".csv" o separador de células seja efetivamente a vírgula "," e não ";". Se o arquivo ".csv" gerado pela sua planilha utilizar ";", faça a substituição para ", ". Caso contrário, ocorrerá um erro de leitura no Weka e o arquivo será interpretado de forma completamente diferente do esperado.

Passo 2 – Dispare o Weka ("GUI Chooser") e tome a opção "Explorer", que corresponde à versão com recursos gráficos e ícones (em vez de linha de comando). Veja Figura 2.3.



(a) **(b)**

Figura 2.3 – Telas Iniciais do Weka (a) GUI Chooser e (b) Explorer.

Passo 3 – Com a aba superior "Preprocess" escolhida, dê um clique em "Open file...". Uma janela denominada "Open" deve se abrir. Ajuste a opção de "File Format:" para ".csv", e escolha o arquivo "Transacoes_1.csv", conforme mostra a Figura 2.4.

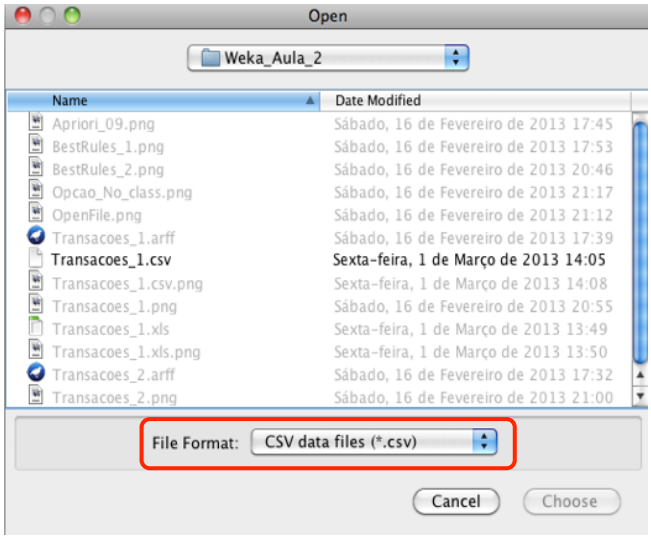


Figura 2.4 – Janela "Open" com a Opção "File Format:" em ".csv".

Passo 4 – A tela do Weka Explorer deve apresentar os sete atributos do arquivo "Transacoes_1", como mostra a Figura 2.5.

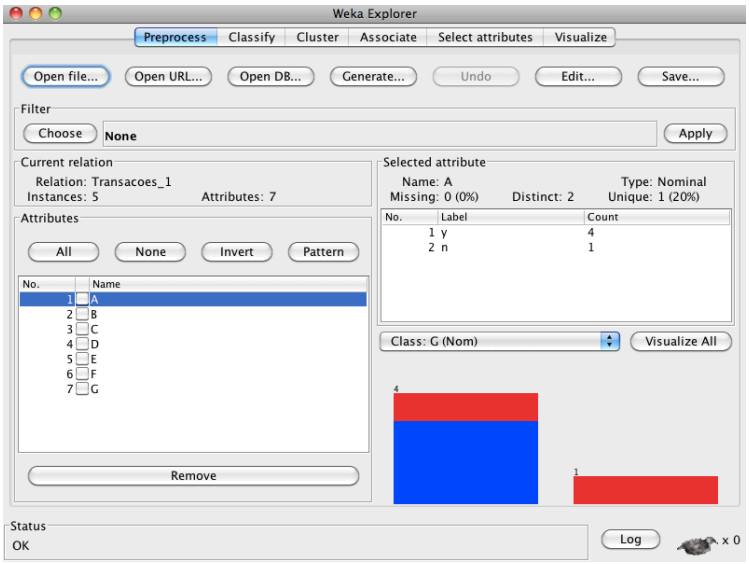


Figura 2.5 – Os Sete Atributos do Arquivo "Transacoes_1" São Mostrados.

Passo 5 – Como nossa "Base de Dados" é muito pequena, a conversão manual do arquivo ".csv" para ".arff" pode ser feita muito rapidamente.

Digite no arquivo "Transacoes_1.csv" as palavras-chave "@relation", "@attribute" e "@data", de acordo com a Figura 2.6, salve e feche o arquivo ".csv". Mude a terminação do arquivo de ".csv" para ".arff". Há ainda outras alternativas: Crie um arquivo "Transacoes_1.txt" com o conteúdo mostrado abaixo na Figura 2.6 (certifique-se de que se trata efetivamente de arquivo tipo ".txt" e não, por exemplo, "Transacoes_1.txt.doc" ou "Transacoes_1.txt.rtf"). Feche o arquivo e mude a terminação para ".arff", ou seja, para "Transacoes_1.arff".

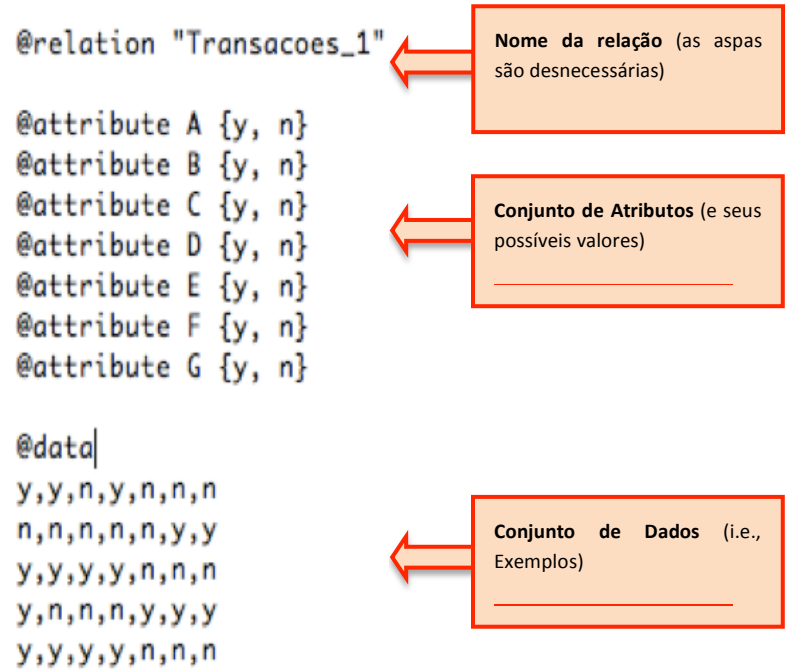


Figura 2.6 – Arquivo ARFF (Transacoes_1.arff), com Itens Ausentes Representados por "n".

Passo 6 – Com o arquivo "Transacoes_1.arff" pronto, disparar o Weka, selecionar a aba "Preprocess", depois clicar na opção "Open file..." e escolher o arquivo "Transacoes1_.arff", conforme mostra a Figura 2.7.

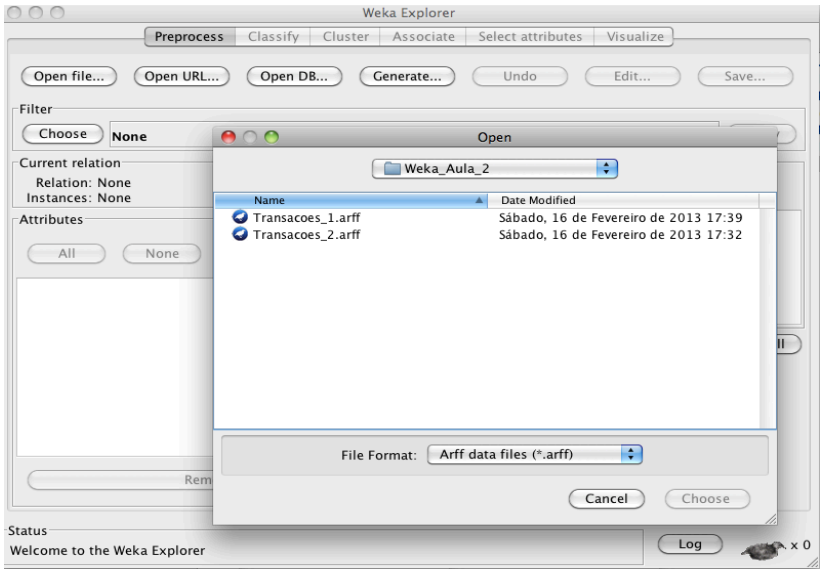


Figura 2.7 – Aba "Preprocess" + "Open file..." para Escolha do Arquivo ARFF.

Passo 7 – Depois de abrir o arquivo "Transacoes_1.arff", ainda com a aba "Preprocess" selecionada, escolha "No class" (ao lado de "Visualize all"), conforme ilustra a Figura 2.8. (Como vamos gerar Regras de Associação, qualquer um dos atributos pode funcionar como "classe". Este conceito vai ser melhor explicado quando formos estudar Regras de Classificação.).

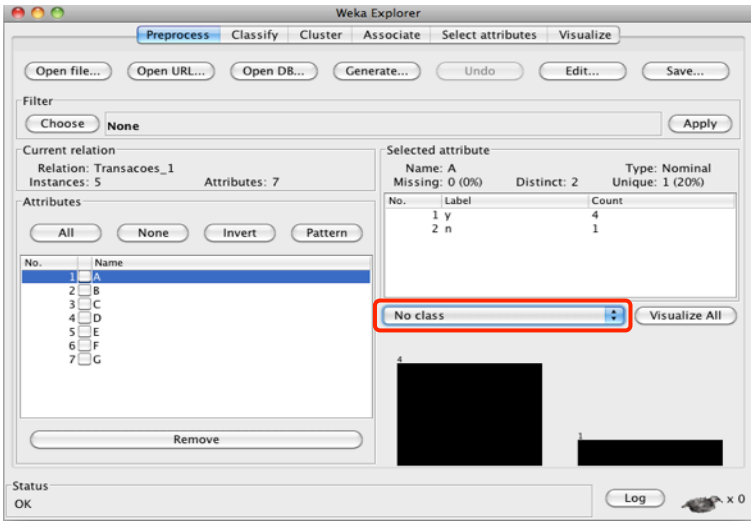


Figura 2.8 – Seleção da Opção "No class" para Regras de Associação.

Passo 8 – Na aba superior do Weka, escolher “Associate” e ao lado de “Choose” clicar duas vezes sobre o algoritmo “Apriori”, conforme mostra a Figura 2.9.

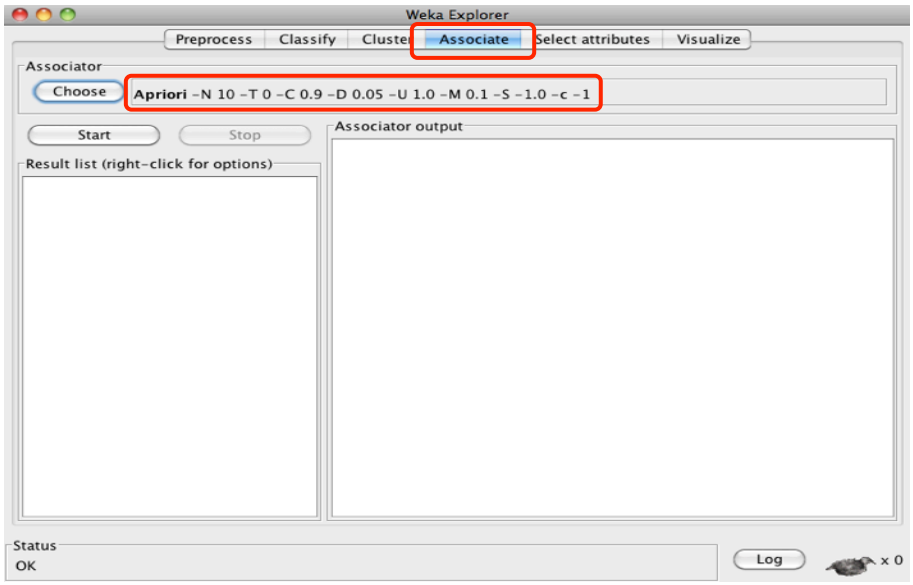


Figura 2.9 – Ajuste dos Parâmetros de Entrada do Algoritmo Apriori.

Passo 9 – Na janela que se abre, ajustar o SupMin (lowerBoundMinSupport) para 0.4, a ConfMin (minMetric) para 0.9 e o número de regras mostradas (numRules) para 1000, conforme mostra a Figura 2.10. Clicar em “OK”.

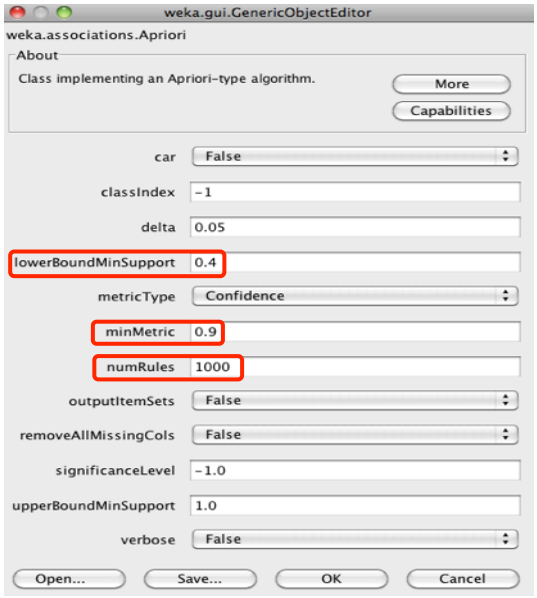


Figura 2.10 – Ajuste dos Parâmetros SupMin e ConfMin.

Passo 10 – Ao clicar em "Start" centenas de Regras de Associação serão geradas, a maioria delas sem qualquer interesse, conforme ilustra a Figura 2.11. Um dos riscos da geração de Regras de Associação é que muitas delas podem não ter qualquer significado prático. Para contornar este tipo de problema, é possível introduzir pequenas mudanças na forma como os atributos são declarados e reduzir significativamente o número de regras geradas.

```
Best rules found:

 1. B=y 3 ==> A=y 3    conf:(1)
 2. D=y 3 ==> A=y 3    conf:(1)
 3. F=n 3 ==> A=y 3    conf:(1)
 4. G=n 3 ==> A=y 3    conf:(1)
 5. D=y 3 ==> B=y 3    conf:(1)
 6. B=y 3 ==> D=y 3    conf:(1)
 7. B=y 3 ==> E=n 3    conf:(1)
 8. F=n 3 ==> B=y 3    conf:(1)
 9. B=y 3 ==> F=n 3    conf:(1)
10. G=n 3 ==> B=y 3    conf:(1)
11. B=y 3 ==> G=n 3    conf:(1)
12. D=y 3 ==> E=n 3    conf:(1)
13. F=n 3 ==> D=y 3    conf:(1)
14. D=y 3 ==> F=n 3    conf:(1)
15. G=n 3 ==> D=y 3    conf:(1)
16. D=y 3 ==> G=n 3    conf:(1)
17. F=n 3 ==> E=n 3    conf:(1)
18. G=n 3 ==> E=n 3    conf:(1)
19. G=n 3 ==> F=n 3    conf:(1)
20. F=n 3 ==> G=n 3    conf:(1)
```

Figura 2.11 – Algumas Regras de Associação Geradas com o Arquivo "Transacoes_1.arff".

Passo 11 – Uma forma de diminuir o número de regras é substituir os valores ausentes de atributo "n" por "?". Crie um arquivo "Transacoes_2.arff" conforme mostra a Figura 2.12.

```
@relation "Transacoes_2"

@attribute A {y, n}
@attribute B {y, n}
@attribute C {y, n}
@attribute D {y, n}
@attribute E {y, n}
@attribute F {y, n}
@attribute G {y, n}

@data
y,y,?,y,?,?,?
?,?,?,?,?,y,y
y,y,y,y,?,?,?
y,?,?,?,y,y,y
y,y,y,y,?,?,?
```

Figura 2.12 – Arquivo "Transacoes_2.arff" com Itens Ausentes Representados por "?".

Isso vai evitar que o Weka crie regras sem qualquer significado prático envolvendo itens ausentes, como por exemplo, {F=n} ⇒ {G=n} (Regra 20 na Figura 2.11). Embora a regra {F=y} ⇒ {G=y} (i.e., "quem compra queijo também costuma comprar vinho") possa ser de interesse, a regra de que "quem não compra queijo também não compra vinho") dificilmente trará alguma informação prática. Numa Base de Dados muito grande, regras desse tipo podem aparecer em quantidades proibitivamente grandes.

Com o arquivo "Transacoes_2.arff" foram geradas 30 Regras de Associação (Figura 2.13), sendo que as regras ilustrativas do texto de teoria do Capítulo 2 envolvendo o CF = {A, B, C} aparecem na Figura 2.13 como as regras 15, 16 e 17.

```
Minimum support: 0.4 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 7

Size of set of large itemsets L(3): 4

Size of set of large itemsets L(4): 1

Best rules found:

 1. B=y 3 ==> A=y 3    conf:(1)
 2. D=y 3 ==> A=y 3    conf:(1)
 3. D=y 3 ==> B=y 3    conf:(1)
 4. B=y 3 ==> D=y 3    conf:(1)
 5. B=y D=y 3 ==> A=y 3    conf:(1)
 6. A=y D=y 3 ==> B=y 3    conf:(1)
 7. A=y B=y 3 ==> D=y 3    conf:(1)
 8. D=y 3 ==> A=y B=y 3    conf:(1)
 9. B=y 3 ==> A=y D=y 3    conf:(1)
10. C=y 2 ==> A=y 2    conf:(1)
11. C=y 2 ==> B=y 2    conf:(1)
12. C=y 2 ==> D=y 2    conf:(1)
13. G=y 2 ==> F=y 2    conf:(1)
14. F=y 2 ==> G=y 2    conf:(1)
15. B=y C=y 2 ==> A=y 2    conf:(1)
16. A=y C=y 2 ==> B=y 2    conf:(1)
17. C=y 2 ==> A=y B=y 2    conf:(1)
18. C=y D=y 2 ==> A=y 2    conf:(1)
19. A=y C=y 2 ==> D=y 2    conf:(1)
20. C=y 2 ==> A=y D=y 2    conf:(1)
21. C=y D=y 2 ==> B=y 2    conf:(1)
22. B=y C=y 2 ==> D=y 2    conf:(1)
23. C=y 2 ==> B=y D=y 2    conf:(1)
24. B=y C=y D=y 2 ==> A=y 2    conf:(1)
25. A=y C=y D=y 2 ==> B=y 2    conf:(1)
26. A=y B=y C=y 2 ==> D=y 2    conf:(1)
27. C=y D=y 2 ==> A=y B=y 2    conf:(1)
28. B=y C=y 2 ==> A=y D=y 2    conf:(1)
29. A=y C=y 2 ==> B=y D=y 2    conf:(1)
30. C=y 2 ==> A=y B=y D=y 2    conf:(1)
```

Figura 2.13 – As 30 Regras de Associação Geradas com o Arquivo "Transacoes_2.arff".

Há outras formas de melhorar a qualidade dos resultados e controlar o número de regras geradas, por exemplo, através do parâmetro ***Lift***, cujo significado fica como lição de casa.

Considerações Finais

Um conjunto de Regras de Associação constitui uma forma de conhecimento extraído de uma Base de Dados, sendo esta representação do conhecimento geralmente um tipo de aprendizado muito útil para aplicações práticas, como o aumento de vendas de uma rede de supermercados, o projeto de catálogos de novos produtos ou o lançamento de campanhas promocionais baseadas em vendas casadas. Geralmente quando fazemos busca na Web, ao digitarmos uma palavra de busca é comum que outras palavras sejam sugeridas. Isso ocorre porque a ferramenta de busca está usando Regras de Associação e tem em sua Base de Dados registros de que pessoas que buscam a Palavra_1 geralmente buscam também a Palavra_2, a Palavra_3, e assim por diante.

Nesta primeira abordagem da extração de conhecimento a partir de uma Base de Dados foi suficiente apenas um procedimento algoritmo, sem necessidade de inferências. Nos próximos capítulos vamos mostrar que na prática é comum nos depararmos com situações para as quais não se conhece um algoritmo que produza o conhecimento necessário para uma tomada de decisão. Para estes casos, será necessário pensar num mecanismo de inferência que nos permita chegar à conclusão mais plausível para determinada situação. Esse é o caso de sistemas conhecidos como Sistemas Especialistas, que auxiliam por exemplo um médico a fazer diagnóstico a partir dos sintomas do paciente. Como nem sempre os sintomas declarados pelo paciente são compatíveis com determinada doença, ou então porque o paciente omite determinados sintomas importantes para o diagnóstico correto, o sistema precisa fazer inferências comparando sua Base [permanente] de Conhecimento com os sintomas declarados.

Regras de Associação frequentemente usam atributos nominais (por exemplo, temperatura elevada, amena, baixa) e mais raramente atributos numéricos (por exemplo 40°C, 23°C, 4°C), porque algoritmos para extração de Regras de Associação com atributos numéricos não costumam apresentar bom desempenho em grandes Bases de Dados. Além disso, ao não levar em conta por exemplo o preço de um artigo ou a quantidade de itens vendidos em cada transação, as Regras de Associação geralmente se transformam numa forma simplista de representação do conhecimento extraído da Base de Dados.

No exemplo da Cesta de Artigos mostramos como gerar Regras de Associação que indiquem venda casada dos artigos mais comum. Mas, frequentemente, os especialistas em vendas não estão muito interessados nestes itens porque a associação entre eles já é conhecida. Na realidade, estes especialistas buscam pares de itens dos quais um deles é um produto barato e o outro tem alta taxa de lucro. Nestes casos, lançar uma superpromoção do produto barato faz com que as vendas do produto com alta taxa de lucro aumentem.

Em nossa Cesta de Artigos está implícito o padrão de associação entre Queijo e Vinho. Talvez aí, numa campanha de inverno, cadeias de supermercados possam fazer promoções de queijos com o único propósito de vender mais vinhos. Mas como as vendas de ambos eram relativamente baixas, esta regra não satisfez os critérios estabelecidos de ***SupMin*** e ***ConfMin***. E, no entanto, é possivelmente este tipo de informação a mais procurada. O que fazer para conseguir minerar as pérolas de informação?

Lista de Exercícios

1. Explique com suas próprias palavras a importância do **Suporte Mínimo (SupMin)** e **Confiança Mínima (ConfMin)** para a geração de **Regras de Associação**.

2. Explique com suas próprias palavras o que é **Conjunto Frequente** no contexto das **Regras de Associação**.

3. Crie uma pequena **Cesta de Compras** (± 5 Exemplos) com itens relacionados ao seu ambiente de trabalho, ou à área de seu TCC, ou a qualquer outra área de seu interesse, e gere as **Regras de Associação** no Weka. Anexe o respectivo arquivo ".arff", e um pequeno relatório sobre a simulação.

Referência Bibliográfica

AGRAWAL, R.; IMIELINSKI, T. & SWAMI, A. **Mining Association Rules Between Sets of Items in Large Databases**. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC. New York: ACM, 1993.

PADHY, N. P. **Artificial Intelligence and Intelligent Systems.** New Delhi: Oxford University Press, 2010.

QUINLAN, J. R. **Induction of Decision Trees**. Machine Learning, Vol. 1, No. 1, pp. 81-106. Boston: Kluwer Academic Publishers, 1986.

ROCHA, M.; CORTEZ, P. & NEVES, J. M. **Análise Inteligente de Dados: Algoritmos e Implementação em Java.** Lisboa: FCA – Editora de Informática, 2008.

TAN, P. N.; STEINBACH, M. & KUMAR, V. **Introdução ao Data Mining Mineração de Dados.** Rio de Janeiro: Editora Ciência Moderna Ltda., 2009.

WITTEN, I. H. & FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques.** Second Edition. Amsterdam: Morgan Kaufmann Publishers, 2005.

Weka. The Waikato University. In http://www.cs.waikato.ac.nz/ml/weka. Acessado em 03.03.13.

WITTEN, I. H. & FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques.** Second Edition. Amsterdam: Morgan Kaufmann Publishers, 2005.