EDI WILSON GARCIA

# PESQUISAR E AVALIAR TÉCNICAS DE MINERAÇÃO DE DADOS COM O USO DA FERRAMENTA ORACLE DATA MINING

ASSIS
2008

# PESQUISAR E AVALIAR TÉCNICAS DE MINERAÇÃO DE DADOS COM O USO DA FERRAMENTA ORACLE DATA MINING

EDI WILSON GARCIA

Trabalho de Conclusão de Curso apresentado ao instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Analisador (1): Luiz Ricardo Begosso

Analisador (2): Domingos de Carvalho Villela Junior

ASSIS
2008

EDI WILSON GARCIA

# PESQUISAR E AVALIAR TÉCNICAS DE MINERAÇÃO DE DADOS COM O USO DA FERRAMENTA ORACLE DATA MINING

Trabalho de Conclusão de Curso apresentado ao instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Área de concentração: Sistemas de Bancos de Dados; Data Mining; PL/SQL.

ASSIS
2008

# DEDICATÓRIA

*Aos meus familiares sem os quais não chegaria até aqui.*

*Aos meus amigos e professores que me ajudaram e apoiaram nos momentos mais necessários.*

*Sou eternamente grato a todos.*

# AGRADECIMENTOS

Aos professores que me auxiliaram a caminhar nessa etapa importante de minha vida e me ajudou a crescer.

Agradeço primeiramente ao Prof. Alex Sandro Romeo Poletto, meu orientador, que me auxiliou na escolha do tema e sempre que necessário se prontificou a ajudar.

Agradeço também a Prof. Mariza Atsuko Nitto, que expressou suas criticas e conselhos visando o desenvolvimento e a qualidade do trabalho.

Em especial a Nathália, por ser companheira em todos os momentos e me dar força nas horas mais difíceis.

Agradeço fraternalmente a todos.

# RESUMO

O principal objetivo deste trabalho é realizar um estudo das técnicas de mineração de dados existentes e aplicá-las em uma base de dados fictícia. Para tal, será utilizado o SGBD Oracle e a ferramenta de mineração Oracle Data Miner. O banco de dados que será usado como teste, será simulado pelo próprio sistema gerenciador de banco de dados, a fim de suprir as necessidades e ausência de uma base de dados real. Este trabalho está dividido em duas etapas. Na primeira etapa será realizado um processo para o conhecimento do problema, bem como suas análises e deduções a serem estudadas e; na segunda etapa será realizada a preparação dos dados e aplicação da mineração de dados.

**Palavras-chave:** Mineração de Dados, Banco de Dados, Oracle, KDD.

# ABSTRACT

The main objective of this work is through an application undertake a study of the mining techniques in existing data and applies them in a fictitious database. For that, it will be used the Oracle DBMS and the mining tool Oracle Data Miner. The database used as test will be simulated by the database-managed system in order to meet the needs and absence of a real basis. This work is divided into two stages. In the first stage, we will know the problem, as well as its analysis and deductions to be studied; and the second stage will be held to make the preparation of data and application of data mining.



**Keywords:** Data Mining, Data Base, Oracle, KDD.

# LISTA DE ILUSTRAÇOES

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AI | *Attribute Importance* |
| ABN | *Adaptive Bayes Network* |
| BI | *Business Intelligence* |
| CIA | *Central Intelligence Agency* |
| DBA | *Database Administrator* |
| DBMS | *Database Management System* |
| DM | *Data Mining* |
| DER | *Diagrama Entidade Relacionamento* |
| DW | *Data Warehouse* |
| ERP | *Enterprise Resource Planning* |
| GB | *GigaByte* |
| IBM | *International Business Machines* |
| KDD | *Knowledge-Discovery in Databases* |
| MB | *Megabyte* |
| MDL | *Minimum Description Length* |
| ODM | *Oracle Data Mining* |
| PC | *Computador Pessoal* |
| PL/SQL | *Procedural Language/Structured Query Language* |
| RSI | *Relational Software Inc.* |
| SDL | *Software Development Laboratories* |
| SGBD | *Sistema Gerenciador de Bando de Dados* |
| SGBDR | *Sistema Gerenciador de Bando de Dados* Relacionais |
| SQL | *Linguagem de Consulta Estruturada* |
| SVM | *Support Vector Machines* |
| TB | Terabyte |
| XML | *Linguagem Extensível de Marcação* |

SUMÁRIO

# 1. INTRODUÇÃO

Nos últimos anos houve um grande aumento na quantidade de informações armazenadas em sistemas de bancos de dados. Segundo Pila, a cada 20 meses dobra a quantidade de informação armazenada em banco de dados. Um grande número de empresas não explora o gigantesco volume de dados no sentido de extrair informações desconhecidas, que podem ser utilizadas no processo de tomada de decisões, ajudando no entendimento de processos, no conhecimento os costumes dos clientes, entre outras coisas. Para isso, é aplicada a técnica chamada *mineração de dados.*

O conceito de Data Mining (*mineração de dados*) é um processo que revela importantes informações, padrões, associações, mudanças, anomalias e estruturas dentro da grande quantidade de dados. Existe um termo global que se dá ao descobrimento de conhecimento útil em base de dados, nomeado *Knowledge Discovery in Databases* ou simplesmente KDD.

A extração dos dados utiliza um complexo processo e depende muito de um analista para preparar os dados, formular os problemas, analisar os resultados e aplicar técnicas estatísticas e de inteligência artificial.

## 1.1. OBJETIVOS

Este trabalho tem por objetivo apresentar uma ferramenta utilizada na mineração de dados, mais exatamente o Oracle Data Mining, e estabelecer um estudo de caso no sentido de simular uma real operação, obtendo-se assim, informações mais concretas sobre tal processo.

Essa simulação será aplicada em uma base de dados simulada pela Oracle. A intenção é aplicar mais de um teste no sentido de mostrar o desempenho com relação ao tempo entre as diversas formas de algoritmos, e a eficiência de cada um deles. A idéia é mostrar a importância desse processo nas tomadas de decisões, visando a descobrimento de valiosas informações sobre o tema abordado.

## 1.2. JUSTIFICATIVAS

Justifica-se o desenvolvimento desse tema, já que o assunto *Data Mining* chamou a atenção por sua ligação com as mais diversas áreas e ramos profissionais e estar diretamente vinculado a Banco de Dados, tornando-se uma ferramenta de grande importância para as tomadas de decisões empresariais.

## 1.3. MOTIVAÇÃO

A motivação está diretamente ligada à importância da técnica de mineração de dados, e por ser um assunto pouco explorado em relação aos demais. O recente interesse em desenvolver novas técnicas de pesquisas analíticas e a importância que ela tem na tomada de decisões empresariais também foram um importante fator para a escolha desse assunto.

## 1.4. PERSPECTIVA DE CONTRIBUIÇÃO

Com esse trabalho pretende-se contribuir com as empresas no sentido de estimular o uso de um recurso pouco explorado e que pode auxiliar em muito as decisões. Além disso, pretende-se preparar um material que poderá ser utilizado por estudantes de tecnologia da informação com as informações sobre a ferramenta que poderá auxiliar nas mais freqüentes dúvidas da ferramenta.

## 1.5. ESTRUTURA DO TRABALHO

Este trabalho esta dividido em oito capítulos sendo o primeiro esta introdução.
No segundo capítulo serão mostrados conceitos de *Data Warehouse.*
No terceiro capítulo serão apresentados os conceitos de KDD.
No quarto capítulo serão apresentadas noções, tarefas, técnicas e aplicações de *Data Mining.*

No quinto capítulo serão apresentadas as aplicações com a ferramenta *Oracle Data Mining.*

No sexto capítulo será apresentada a conclusão do trabalho.

No sétimo capítulo serão citadas as referências bibliográficas.

# 2. DATA WAREHOUSE

Hoje em dia as empresas utilizam sistemas para auxiliar na tomada de decisão, o que antes era realizado através de Bancos de Dados Operacionais, mas com o crescimento do negócio e a necessidade de informações surge a questão de um acesso mais elitizado aos dados, descartando as informações não relevantes para tomada de decisão. O termo *Data Warehouse (DW)* foi usado pela primeira vez em 1990 por Willian Inmon, e surgiu para organizar os dados corporativos para que pudessem ser acessados pelos tomadores de decisão com agilidade e confiabilidade.

”Um *Data Warehouse* é um banco de dados, com ferramentas, que armazena dados atuais e históricos de interesse potencial para os gerentes de toda a empresa.” (LAUDON, 2001, p.168).

O DW é um banco de dados físico, separado dos outros bancos, os dados são armazenados nos sistemas, sendo transferidos para o *DW* somente um resumo dos dados, o que realmente interessa aos tomadores de decisão. Esses dados são organizados como em um Banco de Dados Relacional, possibilitando acesso de consultas. As empresas geram informações com grande rapidez e quantidade chegando facilmente a casa dos TB.

## 2.1. BENEFÍCIOS

- Amplia o conhecimento do negócio;
- Aumenta a vantagem competitiva;
- Melhora o atendimento ao cliente;
- Facilita a tomada de decisão;
- Racionaliza os processos de negócio e
- Fornece uma visão consolidada dos dados da empresa.

## 2.2. CARACTERÍSTICAS

- Organização – os dados são organizados por assunto específico (cliente, fornecedor, produto).
- Consistência – os dados são consistentes, seguindo uma padronização, não importando de qual base de dados sejam provenientes.
- Variante de tempo – os dados são armazenados por um período longo, geralmente em torno de 5 a 10 anos, sendo utilizados para avaliar tendências e fazer previsões.
- Não-volatilidade – os dados armazenados não podem ser alterados, apenas consultados.
- Relacional – Normalmente utilizam estrutura relacional.
- Cliente/servidor – Utiliza a arquitetura cliente/servidor facilitando o acesso do tomador de decisão aos dados.

# 3. **KNOWLEDGE DISCOVERY DATABASE**

*Knowledge Discovery in Database (KDD)* é o nome dado ao processo de conhecimento que engloba o *Data Mining*. Com o aumento do volume de dados surgiu a necessidade de extrair conhecimentos, com isso foi criado um termo no final da década de 80, que atenderia a recentes problemas gerados.

Descoberta de Conhecimento em Bancos de Dados, conhecido originalmente como *KDD – Knowledge Discovery in Database*, tinha como objetivo explorar os dados e encontrar padrões existentes. No processo existem fases que induzem à novas hipóteses e descobertas nas bases de dados. Assim, o usuário pode decidir pela retomada dos processos de mineração, ou uma nova seleção de atributos. A principal característica do KDD é a extração de informações de uma base de dados. O processo é dividido em cinco etapas, ilustradas na Figura 1.

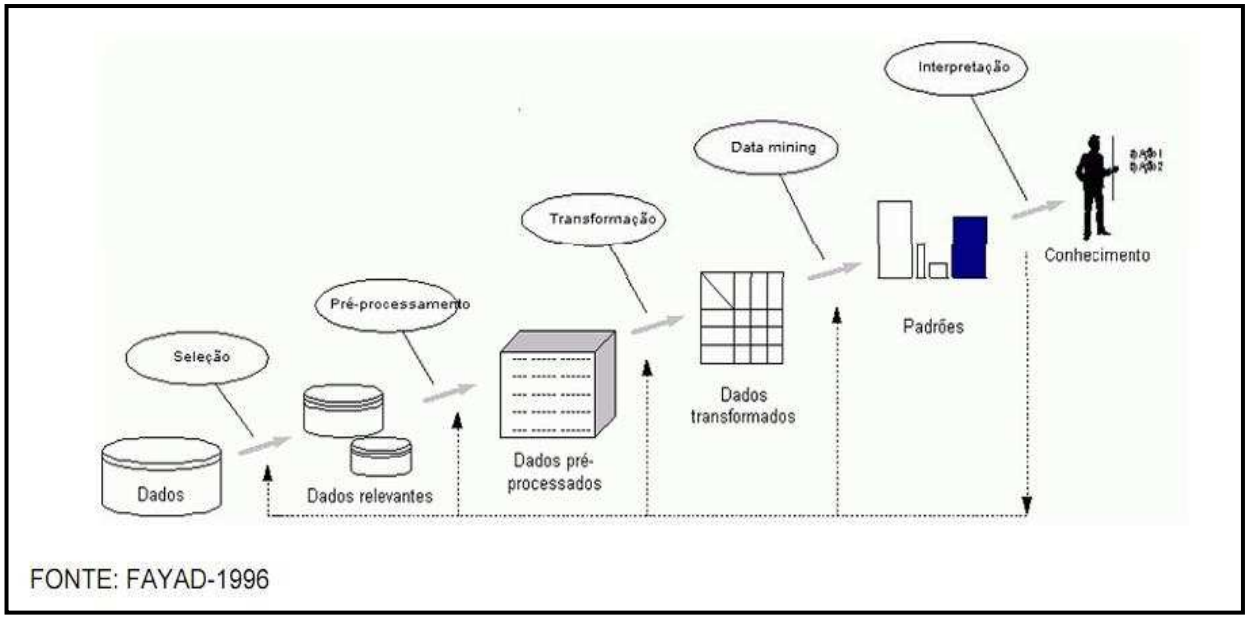



**Figura 1. Processo KDD**

Segundo Thomé, as cinco etapas são descritas da seguinte maneira:

a) Seleção – é a etapa que consiste na análise dos dados existentes e na seleção daqueles a serem utilizados na busca por padrões e na geração de conhecimento novo.

b) Pré-processamento – consiste no tratamento e na preparação dos dados para uso pelos algoritmos. Nesta etapa devemos identificar e retirar valores inválidos, inconsistentes ou redundantes.

c) Transformação – consiste em aplicar, quando necessário, alguma transformação linear ou mesmo não linear nos dados, de forma a encontrar aqueles mais relevantes para o problema em estudo. Nesta etapa geralmente são aplicadas técnicas de redução de dimensionalidade e de projeção dos dados.

d) Mineração – consiste na busca por padrões através da aplicação de algoritmos e técnicas computacionais específicas.

e) Interpretação – consiste na análise dos resultados da mineração e na geração de conhecimento pela interpretação e utilização dos resultados em benefício do negócio.

# 4. DATA MINING

*Data Mining (DM)* é um termo que denomina uma fase do processo KDD, que utiliza métodos de análises estatísticas e de inteligência artificial. Esse processo de mineração de dados descobre a partir de padrões, e associações as relações existentes entre os dados, com isso ajuda a traçar novas estratégias do negócio, identificar comportamento e necessidade dos consumidores, localizar áreas com potencial lucrativo, etc., tais questões seriam impossíveis de serem identificadas a olho “nú”, por razão do grande volume de dados armazenados. DM é considerada a principal fase do processo do descobrimento de conhecimento em bancos de dados (KDD).

A necessidade de extração de conhecimento do banco de dados ganhou um aliado com o surgimento do DW, porque a organização presente na sua estrutura faz com que os dados estejam de forma filtrada e consolidada, facilitando assim o processo do data mining. “Machine Learning” é um termo que também contribui ao crescimento do DM, representa uma combinação entre estatística e inteligência artificial que para descobrir relacionamentos utiliza conceitos como distribuição normal, variância, análise de regressão, desvio padrão e intervalos de confiança (Estatística). A inteligência artificial é baseada no pensamento humano, assim, a Machine Learning faz com que Softwares aprendam na medida em que estudam os dados, adicionando novas regras de decisão.

A literatura atual oferece várias definições para Data Mining. Existem distintas características, porém, a maioria tende para um único conceito: Data Mining é uma das fases do processo de KDD, onde dados são minerados através de algoritmos computacionais, objetivando produzir novas informações.

Seguem algumas definições:

“Data Mining é um passo do processo, consistindo de algoritmos de mineração que, sob algumas limitações aceitáveis de eficiência computacional, produz uma enumeração de padrões E sobre F" (FAYYAD, 1996).

“Data Mining é o uso de técnicas automáticas de exploração de grandes quantidades de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertos a olho nu pelo ser humano” (CARVALHO, 2002).

“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).

“Data Mining refere-se ao“ uso de uma variedade de técnicas para identificar informações úteis em bancos de dados e a extração dessas informações de tal maneira que elas possam ser usadas em áreas tais como teoria de decisão, estimação, predição e previsão.Os bancos de dados são geralmente volumosos, e na forma que se encontram nenhum uso direto pode ser feito deles; as informações escondidas nos dados é que são realmente úteis ”(GUIDE, 2000).

A etapa de *Data Mining* depende fundamentalmente do método utilizado para o tratamento dos dados. Esse é o passo que os padrões freqüentes e de interesse são descobertos a partir dos dados. *Data Mining* refere-se à descoberta de padrão como uma parte da descoberta do conhecimento. Os objetivos primários de *Data Mining* são: ***Predição***, que envolve o uso de algumas variáveis ou campos na base de dados para Predizer valores futuros ou desconhecidos de outras variáveis de interesse; ***Descrição***: busca obter padrões que descrevam os dados e descobrir um modelo a partir dos dados selecionados, essa etapa caracteriza-se pela existência do algoritmo que será capaz de extrair eficientemente o conhecimento implícito e útil de uma Base de dados. De modo geral, cada tarefa KDD extrai um tipo diferente de conhecimento do banco de dados, então cada tarefa requer um algoritmo diferente para a extração de conhecimento da base de dados. A Figura 2 mostra a relação entre tarefas e objetivos básicos do *Data Mining.*


**Figura 2. Hierarquia de Tarefas DM (Silva et al., 2007, p.12)**

## 4.1. TAREFAS EM DATA MINING

Nesta sessão serão abordadas as principais tarefas para a realização das atividades de Data Mining

### 4.1.1. Classificação

Consiste em examinar as características de um objeto ou situação e atribuir a ele uma classe pré-definida. Ou seja, esta tarefa objetiva a construção de modelos que permitam agrupamento de dados em classes. Essa tarefa é considerada preditiva, pois uma vez que as classes são definidas, ela pode prever automaticamente a classe de um novo dado. Existem várias técnicas para a classificação: Árvores de Decisão, Regressão Logística, Redes Neurais e Algoritmo Genético.

### 4.1.2. Associação

Estuda um padrão de relacionamento entre itens de dados. Por exemplo, uma análise das transações de compra em um supermercado pode encontrar itens que tendem a ocorrerem juntos em uma mesma compra (leites e pães), por exemplo. Os resultados desta análise podem ser úteis na elaboração de catálogos e layout de prateleiras de modo que produtos a serem adquiridos na mesma compra fiquem próximos um do outro. Essa tarefa é considerada descritiva, pois busca identificar padrões em dados históricos (Two Crows, 2006).

### 4.1.3 Regressão (Estimativa)

Objetiva definir um valor numérico de alguma variável desconhecida a partir dos valores de variáveis conhecidas. Exemplos de aplicação são: estimar a probabilidade de um paciente sobreviver dado o resultado de um conjunto de diagnósticos de exames; predizer quantos carros passam em determinado pedágio, tendo alguns exemplos contendo informações como: cidades mais próximas, preço do pedágio, dia da semana, rodovia em que está localizado o pedágio, entre outros. Essa tarefa é considerada preditiva. As técnicas utilizadas nessa tarefa são: Redes Neurais, Regressão Linear, Análise de Discriminante (Two Crows, 2006).

### 4.1.4. Sumarização (Clustering)

As informações podem ser particionadas em classes de elementos similares. Neste caso, nada é informado ao sistema a respeito das classes existentes. O próprio algoritmo descobre as classes a partir das alternativas encontradas na base de dados, agrupando assim um conjunto de objetos em classes de objetos semelhantes. Por exemplo, uma população inteira de dados sobre tratamentos de uma doença pode ser dividida em grupos baseados na semelhança de efeitos colaterais produzidos; acessos a Web realizados por um conjunto de usuários em

relação a um conjunto de documentos podem ser analisados para revelar clusters ou categorias de usuários. Essa tarefa é considerada descritiva. A técnica utilizada nessa tarefa é chamada de Análise de Cluster e utiliza uma área da Estatística denominada Análise Multivariada. (Two Crows, 2006).

## 4.2. TÉCNICAS EM *DATA MINING*

Nesta sessão serão apresentadas as algumas técnicas de *Data Mining.*

### 4.2.1. Rede *Bayesiana (Adaptive Bayes Network)*

É um algoritmo da *Oracle* que dá apoio a árvore de decisões, o resultado é fornecido na forma de regras de fácil compreensão, com bom desempenho, tem possibilidade de informar parâmetros que definem o nível de precisão e o tempo de construção. É utilizado quando existe um grande volume de atributos, assim é feito o cálculo das probabilidades de que uma determinada amostra pertença a cada uma das classes possíveis, predizendo para a amostra, a classe mais provável. **(Silva et al.,2007, p.14)**

### 4.2.2 Árvore de Decisão (Decision tree)

Nesta técnica escolhe-se a variável que se quer avaliar e o software procura as mais correlacionadas e monta a árvores com várias ramificações. As Arvores de Decisão são meios de representar resultados de Data Mining na forma de árvore, e que lembram um organograma. Dado um grupo de dados com numerosas linhas e colunas, uma ferramenta de árvore de decisão pede ao usuário do software para definir em sua base de dados, qual será o atributo meta (objeto de saída) e então mostra o único e mais importante fator correlacionado com aquele objeto de saída como o primeiro ramo ou nó da árvore de decisão. Os outros atributos preditores subseqüentes são classificados como nós, do nó anterior, formando gradativamente

a árvore. Uma das melhores características da técnica é a facilidade de manipulação, aliada a comunicação visual da árvore que facilita a compreensão do usuário (Ferreira, 2005).

A Figura 3 mostra um exemplo de uma pequena árvore de decisão e seus componentes com o objetivo de avaliar o risco de crédito.

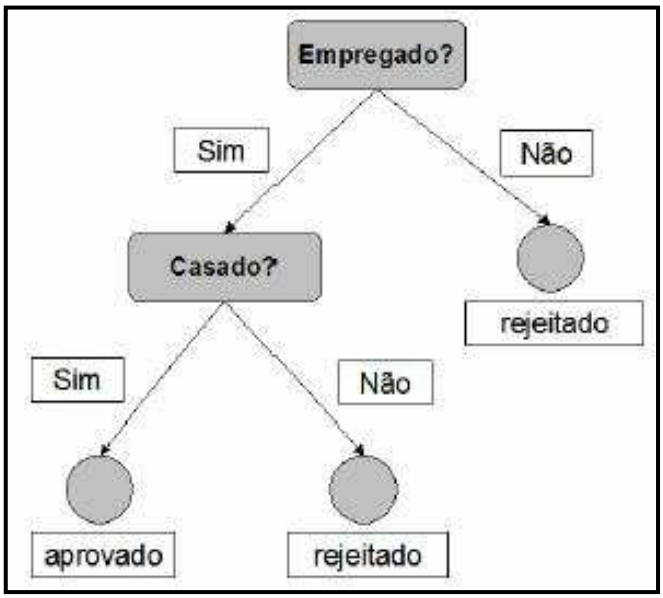



**Figura 3. Arvore de Decisão. (Silva et al.,2007, p.15)**

### 4.2.3. Associativas

Uma regra de associação é representada pela notação YX, ou seja, X implica em Y, onde X e Y são conjuntos distintos. O objetivo desta técnica é representar, com determinada certeza, uma relação existente entre o antecedente e o conseqüente de uma regra de associação. A associação é uma tarefa descritiva, pois visa identificar padrões em dados históricos.

Um exemplo típico de Regras de Associação é construído quando se utiliza uma cesta de compra. O objetivo é saber se determinado produto X implica na compra do produto Y. Esta implicação é avaliada através de dois fatores: suporte e confiança. O suporte de uma regra representa o percentual das transações em que a regra acontece em relação ao total de transações. A confiança não trabalha com todas as transações, apenas com as que possuem o antecedente da regra. Assim a confiança

é a razão entre o número de vezes em que o conseqüente da regra aparece, pela quantidade dessas transações (Souza, 2003).

A Figura 4 representa uma tabela de transações em que a regra de associação aparece. Por exemplo, para regra café -> leite, aparecem juntos em 60% das transações (1, 3, 4, 6, 8 e 10).

| Id Transação | Itens Comprados |
|---|---|
| 1 | Café, leite, manteiga e pão. |
| 2 | Milho, morango, pão. |
| 3 | Café, leite, farinha, cerveja. |
| 4 | Biscoito, café, carne, leite, presunto, vinho. |
| 5 | Adoçante, biscoito, peixe, queijo, vinho. |
| 6 | Adoçante, café, leite, pão. |
| 7 | Biscoito, milho, presunto, tomate. |
| 8 | Café, mel, leite, macarrão. |
| 9 | Frango, mel, tomate. |
| 10 | Biscoito, café, cerveja, leite, refrigerante. |

**Figura 4. Tabela Associação. (Silva et al.,2007, p.18).**

Já o fator confiança, ao invés de considerar todas as transações, trabalha apenas com as que possuem antecedente da regra. Por exemplo, para a regra café -> leite, a confiança é de 100%,ou seja, em todas as compras de café, há a compra de leite.

### 4.2.4. Redes Neurais

Redes Neurais é uma classe de modelagem de prognóstico que trabalha por ajuste repetido de parâmetro. Estruturalmente uma rede neural consiste em um número de elementos interconectados (chamados neurônios), organizados em camadas unidas por conexões. As Redes Neurais geralmente constroem superfícies complexas, utilizando equações algébricas e funções de ajuste. A função básica de cada neurônio é: avaliar os valores de entrada, calcular e comparar o total com um valor e determinar o valor de saída, a operação de cada neurônio é bastante simples, mas

procedimentos complexos podem ser criados pela conexão de um conjunto de neurônios.

Segundo Ferreira, (2005, p.36), “o neurônio artificial é dividido em 2 seções funcionais. A primeira seção combina todas as entradas que alimenta o neurônio. Essa etapa indica como as entradas serão computadas (regra de propagação). A segunda seção recebe esse valor e faz um cálculo determinando o grau de importância da soma ponderada utilizando uma função de transferência, ou função de ativação. Essa função determina com que grau a soma causará uma excitação ou inibição do neurônio. Os tipos mais comuns de funções de ativação são sigmóide e tangente hiperbólica, pois fornecem a característica de não linearidade para uma RNA”

A Figura 5 mostra um modelo de Rede Neural.

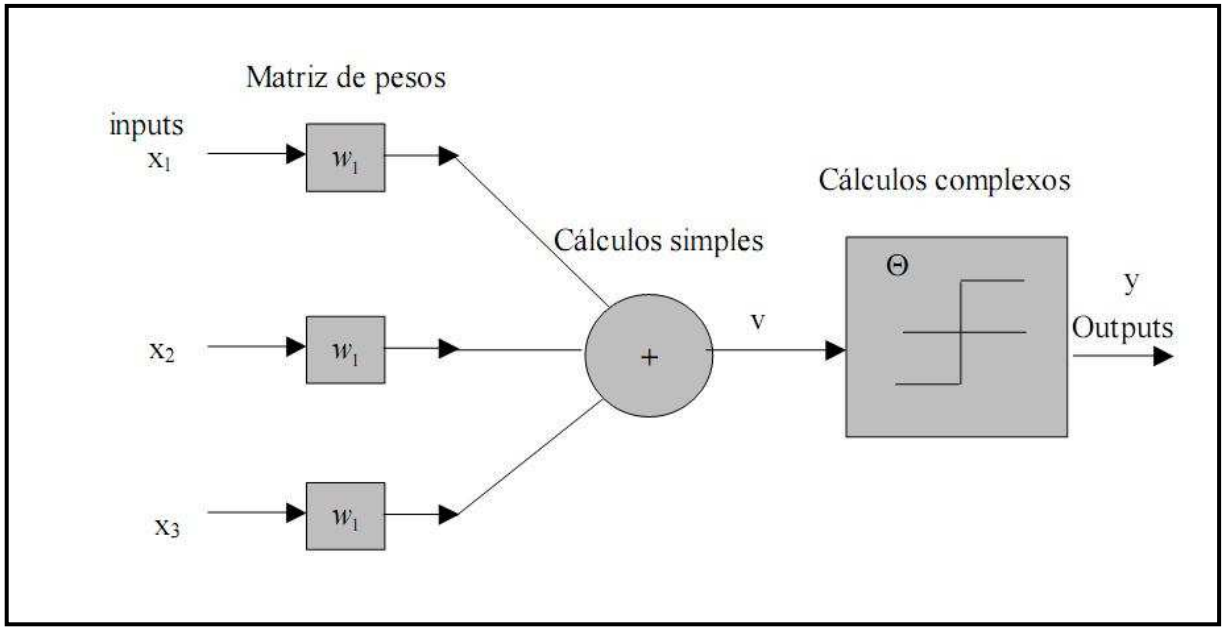



**Figura 5. Modelo Rede Neural**

## 4.3. APLICAÇÕES DE DATA MINING

Nesta sessão serão mostrados alguns exemplos de áreas em que o Data Mining vem sendo utilizado com bastante freqüência e sucesso.

### 4.3.1. Data Mining em Comércio

Utilizando de gigantescas bases de dados disponíveis na área comercial, grandes grupos supermercadistas estudam o comportamento de compra de seus clientes, é feito um cadastramento de clientes registrado através de um cartão específico, que no momento da compra, registra as características pessoais do cliente e as características dos produtos comprados. A análise dos dados pode motivar novos clientes ou ainda manter a clientela padrão, com promoções, eventos, vendas casadas, etc.

### 4.3.2. Data Mining em Finanças

Bancos, instituições financeiras e entidades de proteção ao crédito, utilizam técnicas de *Data Mining* em suas bases de dados para métodos para avaliação de crédito, visando antecipar o perfil do cliente em relação a sua inadimplência.

### 4.3.3. Data Mining em Seguros

Grandes companhias de seguro utilizam das técnicas de Data Mining para analisar as características dos clientes, predizendo quem cancelaria as suas apólices com certa margem de segurança. Os resultados obtidos oferecem valiosas informações, assim os administradores podem reduzir a quantidade de apólices canceladas e os custos das empresas.

### 4.3.4. Data Mining em Medicina

As técnicas de Data Mining através da análise do banco de dados, que contém um valioso histórico dos pacientes, conseguem, por exemplo, identificar correlações

entre variáveis imperceptíveis a olho nu, auxiliando no sucesso de órgãos transplantados e redução de efeitos de quimioterapia.

### 4.3.5. Data Mining no Governo

O Governo dos Estados Unidos varre banco de dados usando *Data Mining* com o objetivo de identificar transferência de fundos internacionais que se parece com lavagem de dinheiro, compradores de explosivo, armas e munições, na tentativa de diminuir o índice de crimes e atentados terroristas.

### 4.3.6. Data Mining em Telecomunicações

Técnicas em *Data Mining* podem ser utilizadas para destacar os hábitos dos usuários de telefones celulares, uma expectativa da contribuição para diminuir a clonagem e crimes contra a telefonia. É realizada uma ligação para o cliente quando um telefonema é realizado fora dos padrões do software, assim podendo confirmar se foi ou não uma fraude.

## 4.4. SOFTWARES PARA MINERAÇÃO DE DADOS

a) *Weka: software* open source (Java), desenvolvido pela Universidade de Waikato, contém uma série de algoritmos de DM;
b) *Intelligent Miner*: desenvolvido pela IBM, é uma ferramenta de DM interligado diretamente com o banco de dados DB2 da IBM;
c) *Oracle Data Miner*: desenvolvido pela *Oracle* que permite interligação direta com o banco de dados *Oracle*;
d) *Enterprise Miner*: desenvolvido para DM tradicionalmente utilizado na área de negócios, *marketing* e inteligência competitiva;
e) *Statistica Data Miner*: acrescenta as facilidades de mineração de dados ao tradicional pacote utilizado em aplicações de estatística.

# 5. PROPOSTA DE TRABALHO E ESTUDO DE CASO

A proposta deste trabalho será a apresentação da ferramenta *Oracle Data Mining,* mostrando no estudo de caso os principais recursos da ferramenta, algumas técnicas de *Data Mining* e seus respectivos resultados, utilizando uma base de dados fictícia disponível no site da *Oracle* e também gerar um código PL/SQL da atividade

## 5.1. ORACLE DATA MINING

A ferramenta de *Data Mining* utilizada neste trabalho será o *Oracle Data Mining*, disponível para *download* no site da *Oracle.*

A versão utilizada foi a 10.2, compatível com a versão 10g do sistema gerenciador de banco de dados, também utilizado para a realização dos testes. Para instalação da ferramenta será necessária a verificação de alguns requisitos de Hardware como: processador 550Mhz, espaço disponível em disco 2GB, 512 MB memória RAM, o dobro de RAM para memória virtual e adaptador de vídeo de 256 cores. Oracle Data Miner é uma interface gráfica que tem como objetivo facilitar e auxiliar os usuários através de preparação dos dados, mineração de dados e modelos dos processos e tarefas. A ferramenta é suportada em sistemas operacionais Windows 2000, Windows XP Professional Edition, e Linux. (ou superiores).

Antes de realizar a primeira conexão da ferramenta é necessário verificar se o *Schema DMSYS* está destravado, já que o mesmo vem travado por padrão, impossibilitando a execução do trabalho com *DM*, como mostra a Figura 6.

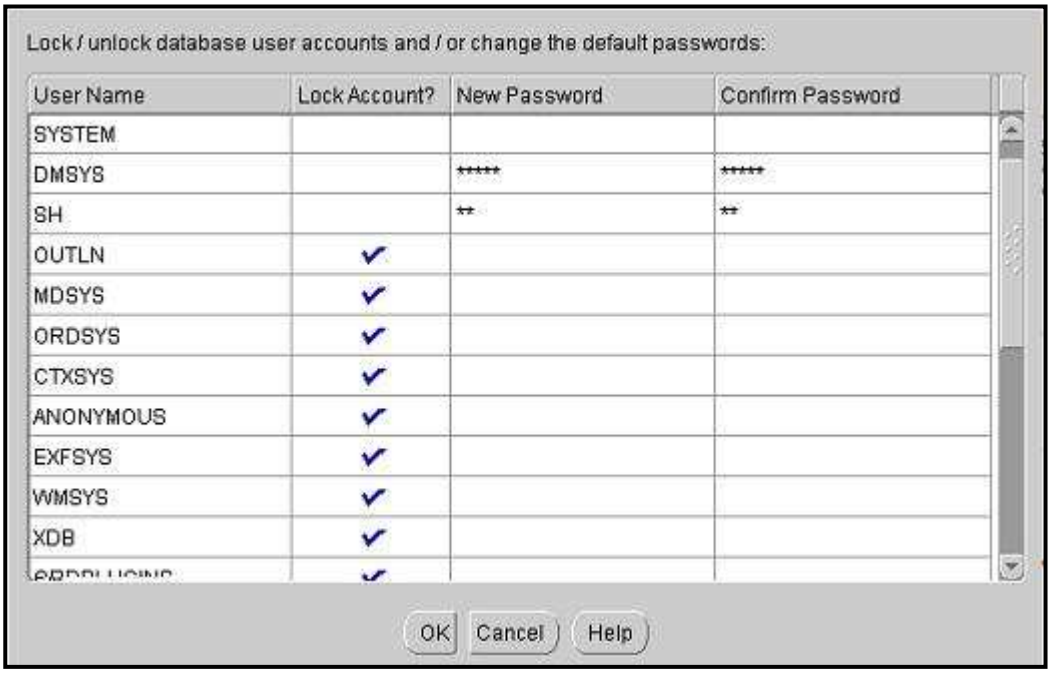



**Figura 6. Conta Detalhes**

Para efetuar os procedimentos necessários para realização da pesquisa foi utilizada uma base de dados fictícia disponível no site da Oracle. Essa base de dados contém informações dos hábitos de compras de consumidores e auxiliou na busca de problemas de negócios e entre outros, totalizando 2371 registros e 24 atributos. Para a importação de dados no ODM selecione *import* no menu data como mostra a Figura 7.


**Figura 7. Importar**

Após o processo de importação das tabelas poderemos visualizar a estrutura e os dados da tabela alternando entre as abas como descrito nas figuras 8 e 9.

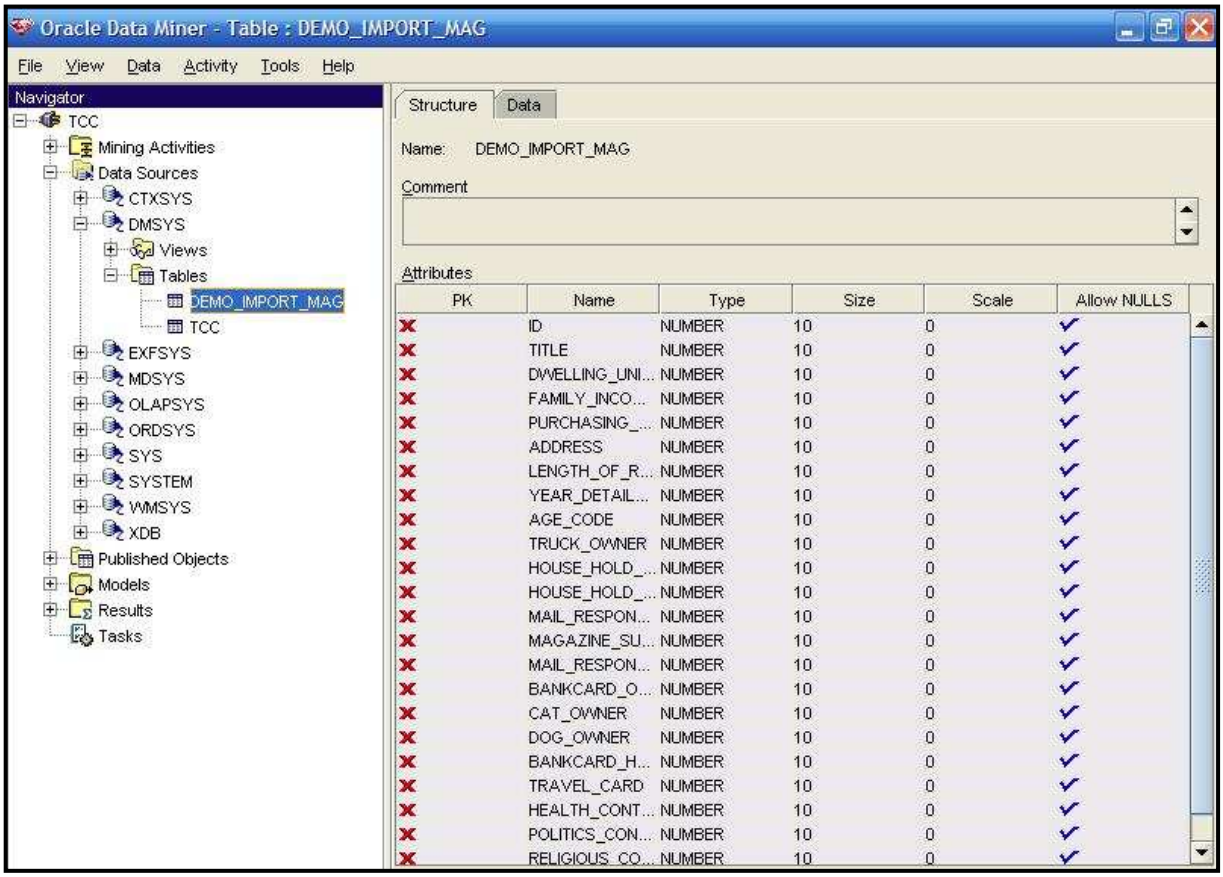



**Figura 8. ODM Estrutura**

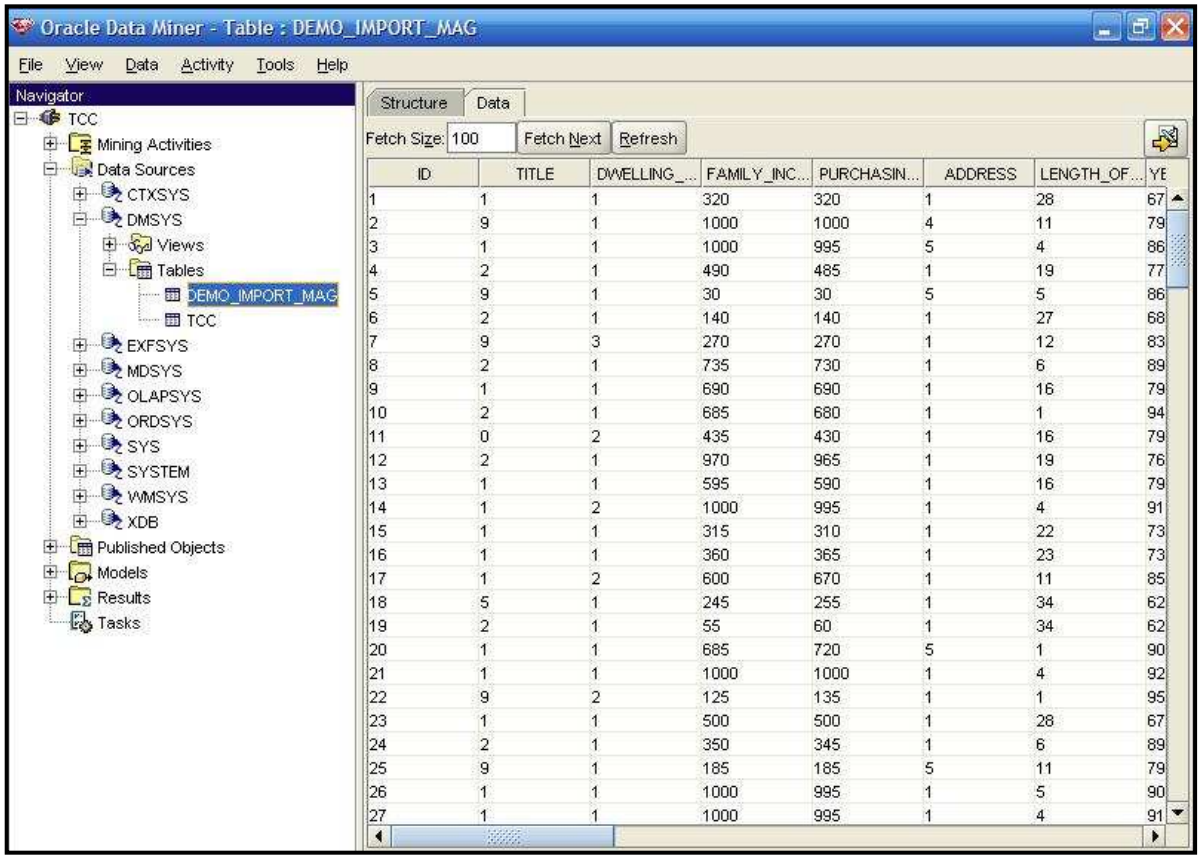



**Figura 9. ODM Dados**

Através do menu acessado, clicando com o botão direito em cima da tabela, obtém-se acesso as varias opções da ferramenta uma delas é um resumo estatístico sobre determinado atributo, contido em *Show Summary Single-Record.*

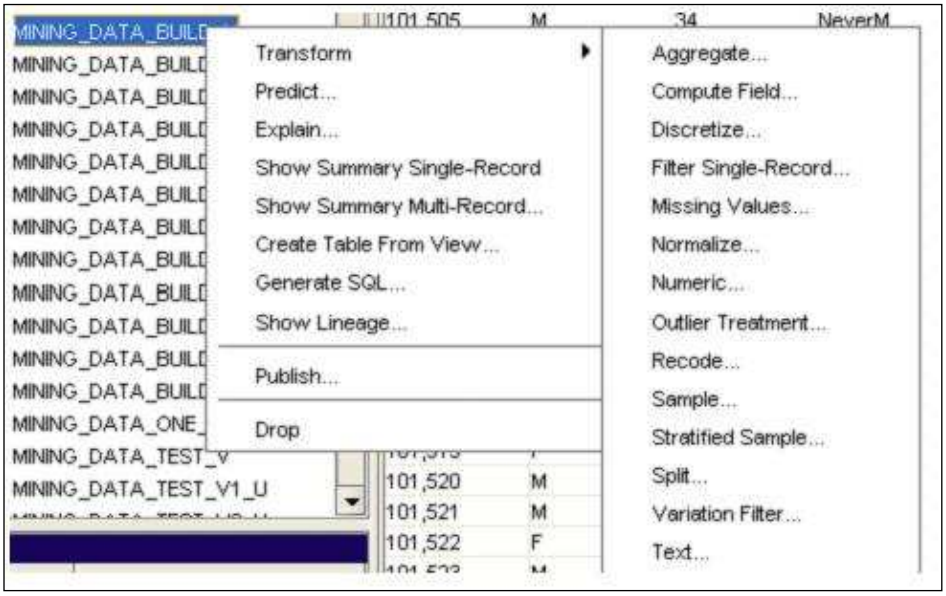
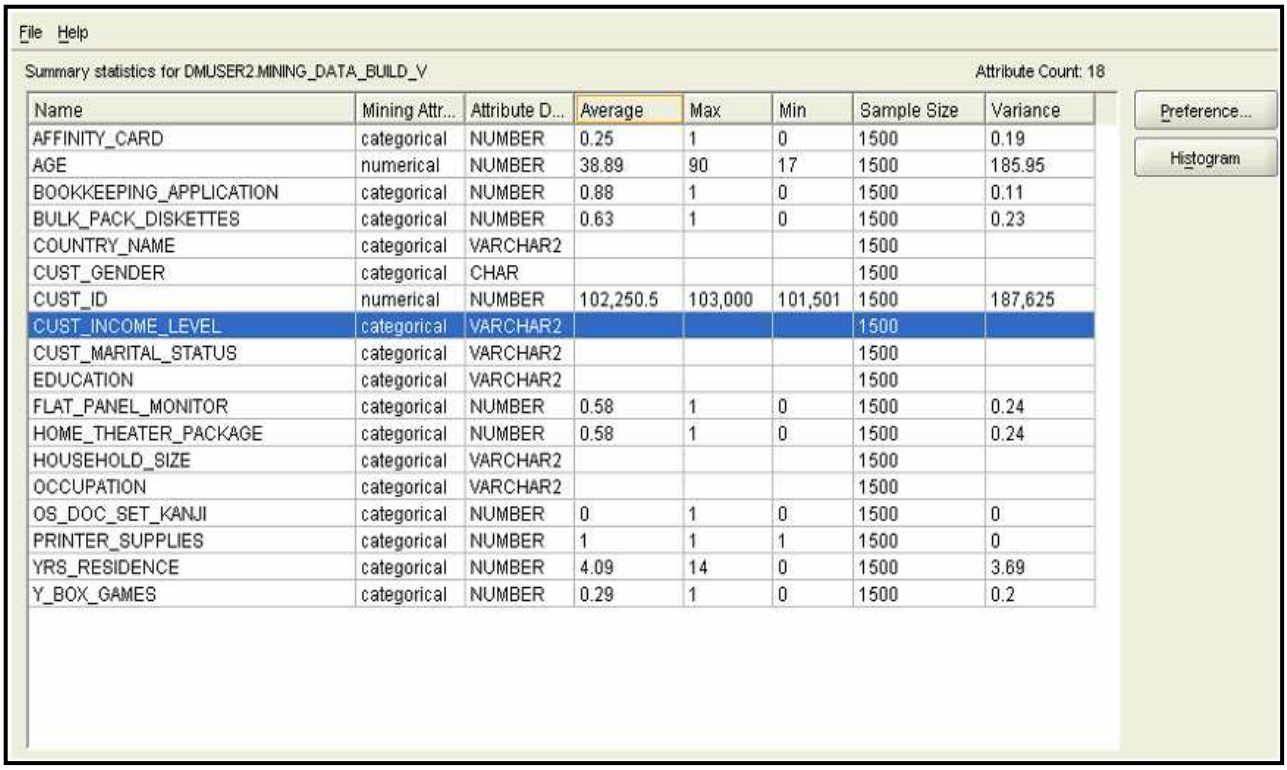



**Figura 10. Menu Resumo Estatístico**

A Figura 11 mostra o resumo estatístico do atributo selecionado, entre suas informações estão o tipo do atributo, o valor mínimo, o valor máximo, etc. Outra forma de visualizar essas estatísticas é através de gráficos e formas visuais que facilitam a compreensão dos resultados, essa forma é obtida clicando no botão *Histogram.*




| Name | Mining Attr... | Attribute D... | Average | Max | Min | Sample Size | Variance |
|---|---|---|---|---|---|---|---|
| AFFINITY_CARD | categorical | NUMBER | 0.25 | 1 | 0 | 1500 | 0.19 |
| AGE | numerical | NUMBER | 38.89 | 90 | 17 | 1500 | 185.95 |
| BOOKKEEPING_APPLICATION | categorical | NUMBER | 0.88 | 1 | 0 | 1500 | 0.11 |
| BULK_PACK_DISKETTES | categorical | NUMBER | 0.63 | 1 | 0 | 1500 | 0.23 |
| COUNTRY_NAME | categorical | VARCHAR2 | | | | 1500 | |
| CUST_GENDER | categorical | CHAR | | | | 1500 | |
| CUST_ID | numerical | NUMBER | 102,250.5 | 103,000 | 101,501 | 1500 | 187,625 |
| CUST_INCOME_LEVEL | categorical | VARCHAR2 | | | | 1500 | |
| CUST_MARITAL_STATUS | categorical | VARCHAR2 | | | | 1500 | |
| EDUCATION | categorical | VARCHAR2 | | | | 1500 | |
| FLAT_PANEL_MONITOR | categorical | NUMBER | 0.58 | 1 | 0 | 1500 | 0.24 |
| HOME_THEATER_PACKAGE | categorical | NUMBER | 0.58 | 1 | 0 | 1500 | 0.24 |
| HOUSEHOLD_SIZE | categorical | VARCHAR2 | | | | 1500 | |
| OCCUPATION | categorical | VARCHAR2 | | | | 1500 | |
| OS_DOC_SET_KANJI | categorical | NUMBER | 0 | 1 | 0 | 1500 | 0 |
| PRINTER_SUPPLIES | categorical | NUMBER | 1 | 1 | 1 | 1500 | 0 |
| YRS_RESIDENCE | categorical | NUMBER | 4.09 | 14 | 0 | 1500 | 3.69 |
| Y_BOX_GAMES | categorical | NUMBER | 0.29 | 1 | 0 | 1500 | 0.2 |

**Figura 11. Resumo Estatístico**

A Figura 12 mostra o resultado obtido.

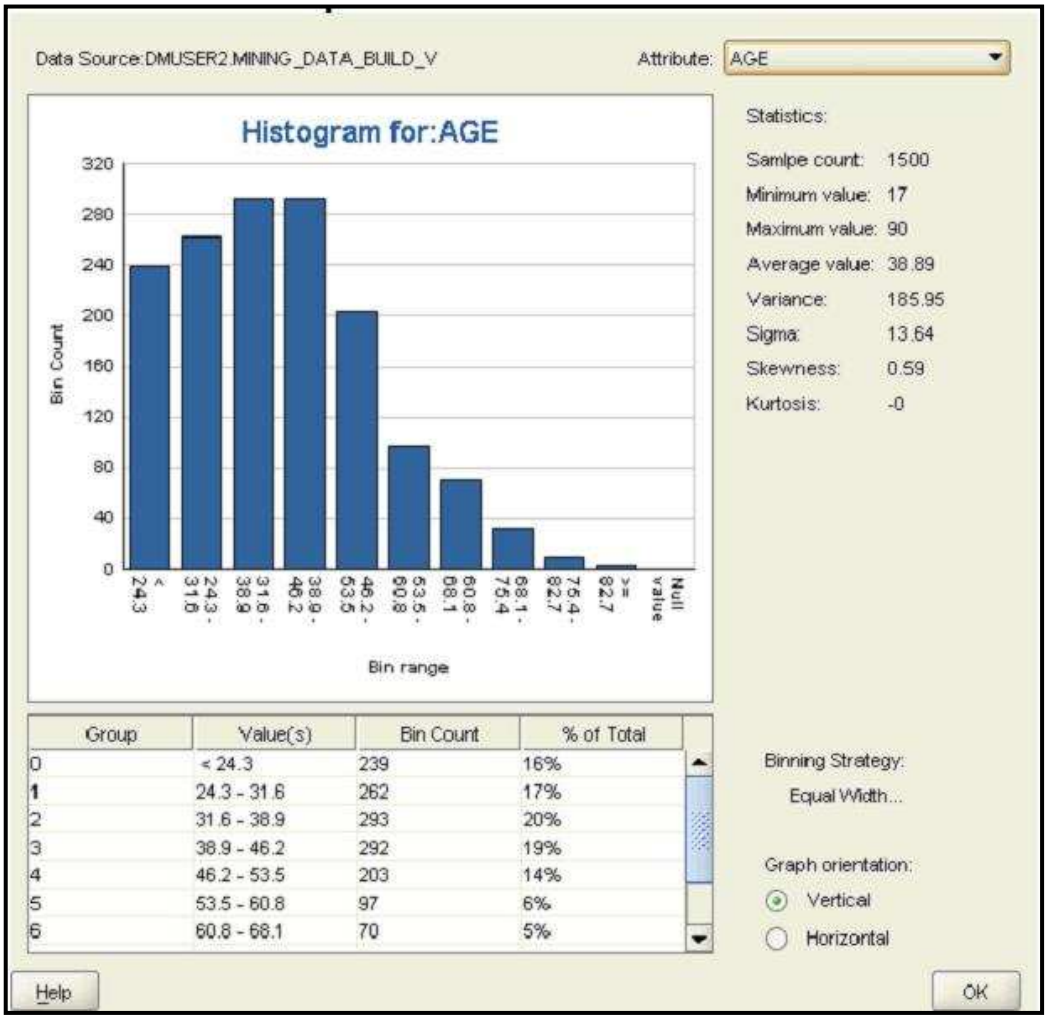



**Figura 12. Histograma**

A seguir serão descritas algumas características importantes a serem notadas, contidas no menu do ODM.

- Filter Single-Record – suponha que queremos evidenciar os consumidores com poderes de compra com valores entre 350 e 1000, com esse filtro poderemos criar uma *view* através de um editor de expressão como mostra a Figura 13.

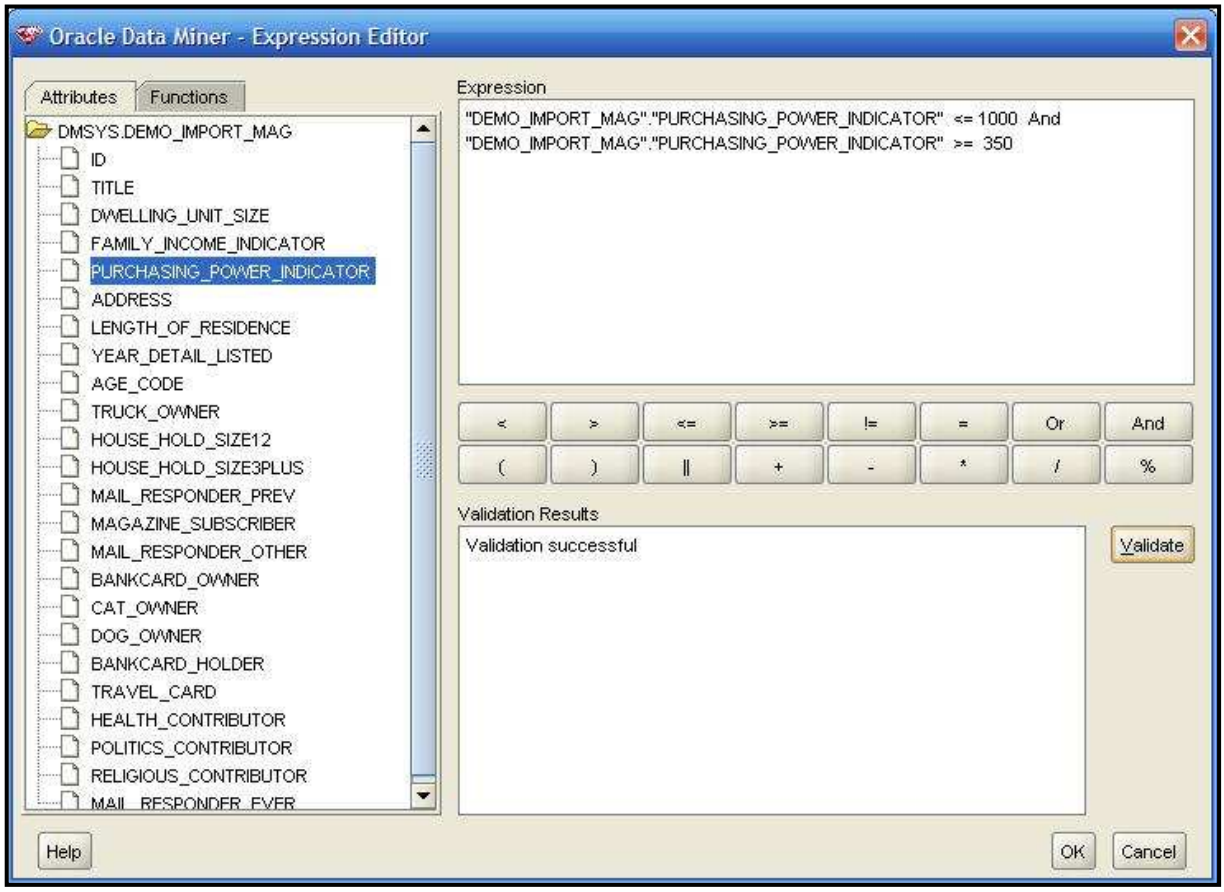



**Figura 13. Editor de Expressão**

- Recode – resumidamente nessa opção poderemos substituir valores de atributos por novos valores com auxilio de condições, por exemplo, com finalidade de construir um modelo operacional mais eficiente, podemos substituir os valores do atributo LENGTH_OF_RESIDENCE, que contém valores entre 1 e 34, e criar duas novas classes de casas LOW para valores menores ou iguais a 10 anos e HIGH para maiores que 10 anos. Para isso selecione o atributo e clique em *Define* como na Figura 14.

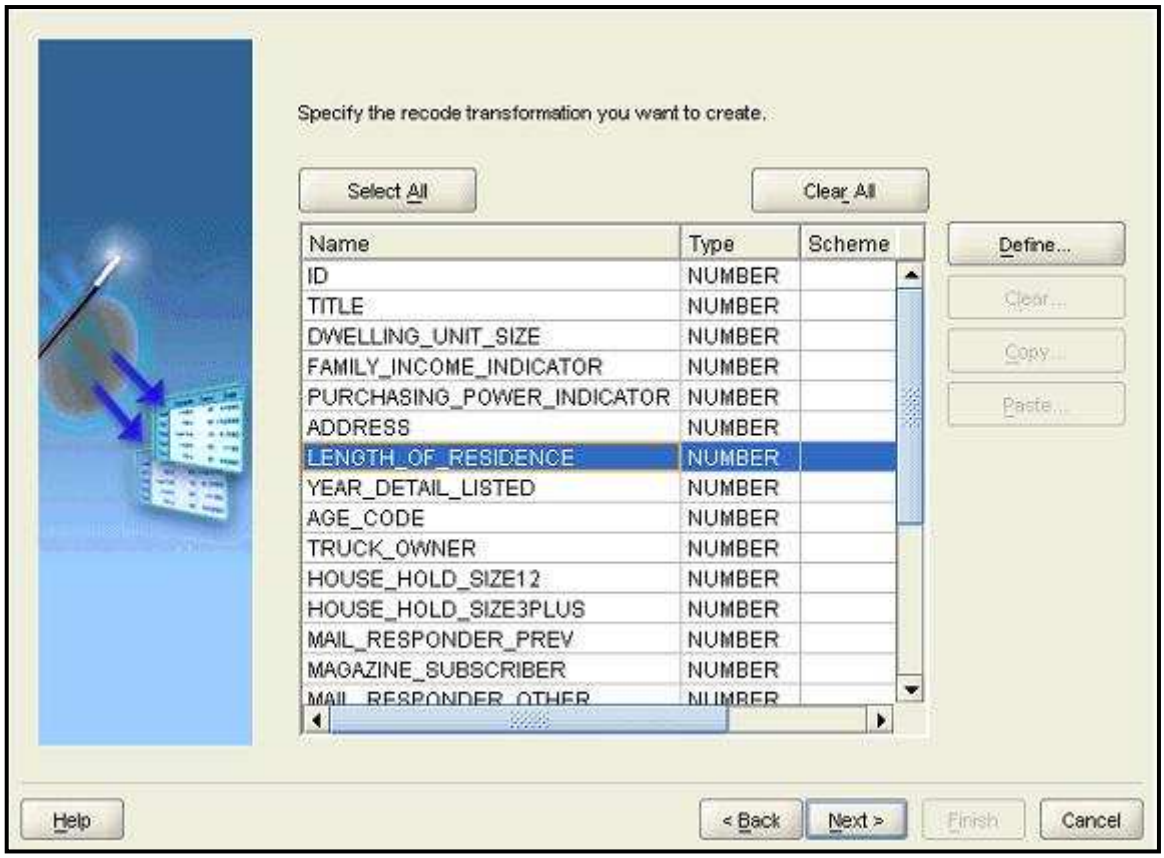



**Figura 14. Recode**

Na caixa de dialogo escolha a condição e digite o novo valor e clique em *Add,* repita a operação ate suprir as alterações necessárias para novos valores, podemos também definir valores para campos *NULL* e *OTHER* para outros valores. A figura 15 mostra a caixa de dialogo e suas opções.

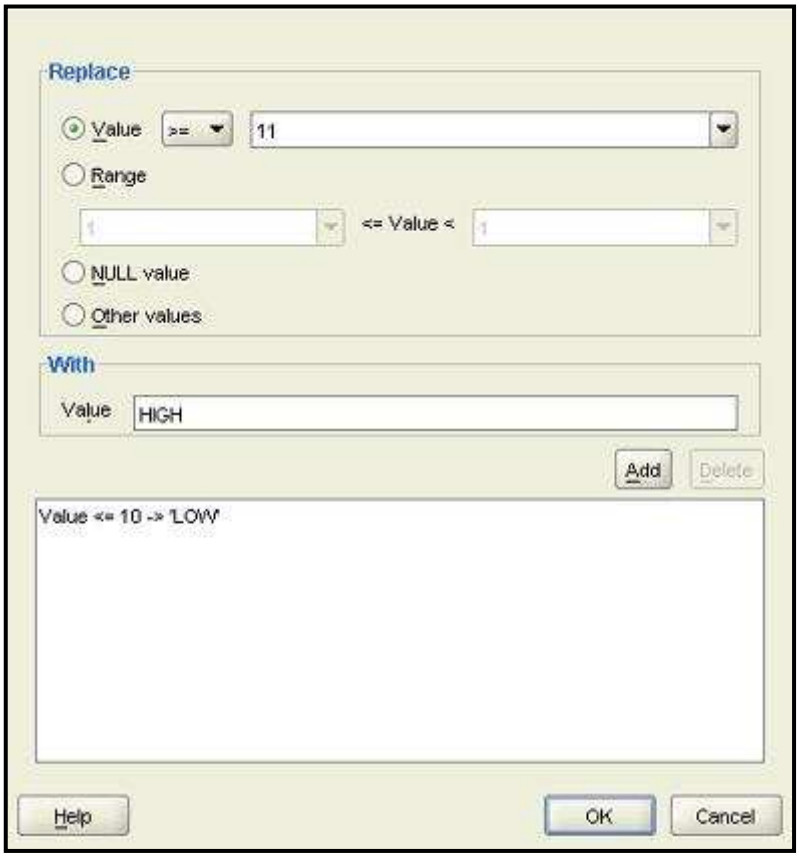



**Figura 15. Recode Dialogue**

Ao final do processo poderemos notar que houve alteração no tipo do atributo, o que antes era numérico, agora, está definido como VARCHAR2.

- Compute Field - Opção muito usada para gerar uma nova coluna através de atributos existentes, por exemplo para calcular campos de datas e gerar um valor numérico, como em SYSDATE - DATE_OF_BIRTH obteremos o valor da idade em dias. A construção da tarefa é feita através do Expression Editor como na Figura 14 e apenas uma opção é acrescentada que é o campo para nomear a nova coluna a ser criada.

### 5.1.1. ODM - Attribute Importance

Geralmente a bases de dados utilizadas para mineração de dados, contém muitos atributos que podem gerar ruídos atrapalhando o desempenho dos resultados. O ODM contém uma ferramenta chamada Attribute Importance (AI) que utiliza o algoritmo Minimum Description Length (MDL) para classificar os atributos por suas significâncias. AI pode ser usado para reduzir os problemas de classificação, obtendo do usuário conhecimento necessário para eliminar alguns atributos e assim ganhar velocidade e precisão no processo.

A construção de um modelo de AI é feito através de *Build* no menu *Activity* como mostra a Figura 16.

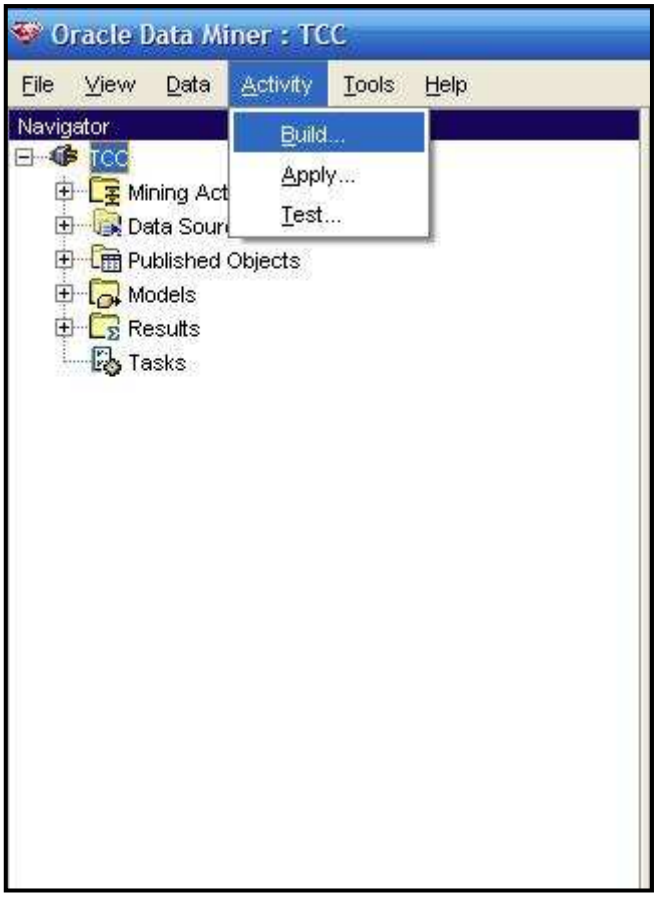



**Figura 16. Menu Build**

Selecione AI como na Figura 17, note que terá apenas uma opção no menu de escolha do algoritmo

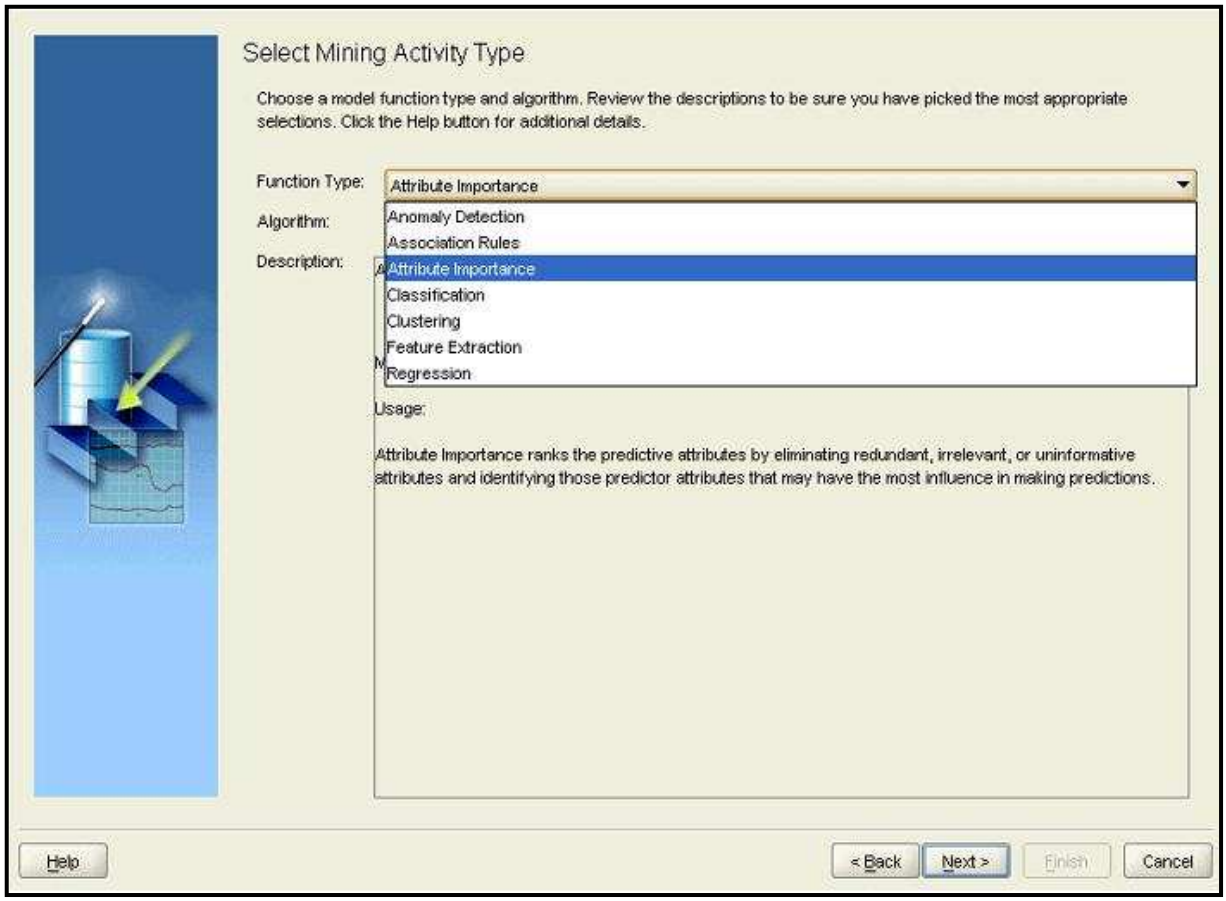



**Figura 17. AI Seleção**

Na próxima etapa será definida a tabela ou *view* a ser consultada e a identificação de um atributo único como mostra a Figura 18.

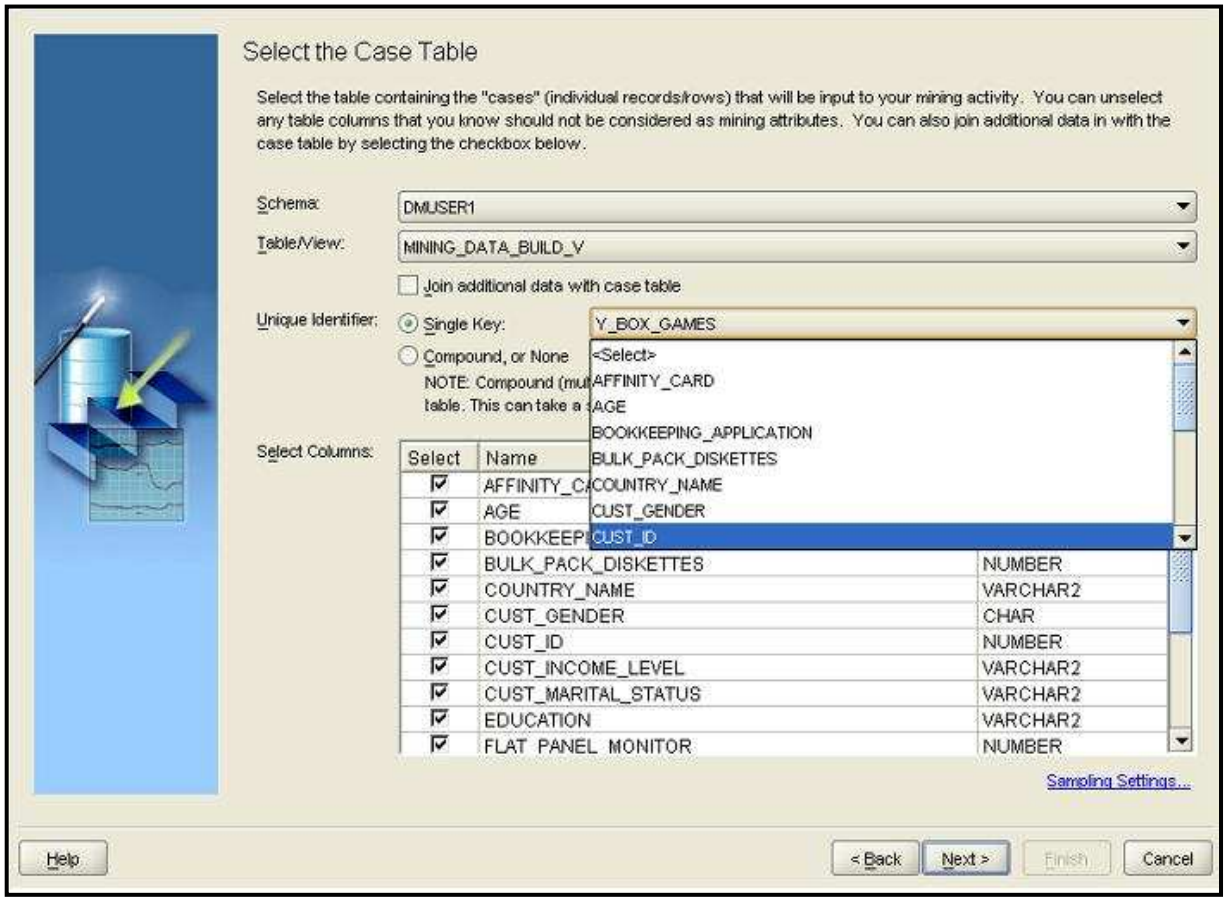



**Figura 18. AI Informações**

O importante é distinguir o atributo que contém a informação que possa ser considerada como uma informação de alto valor, no nosso exemplo esse atributo é o AFFINITY_CARD que será identificado na opção *target* como mostra a Figura 19.

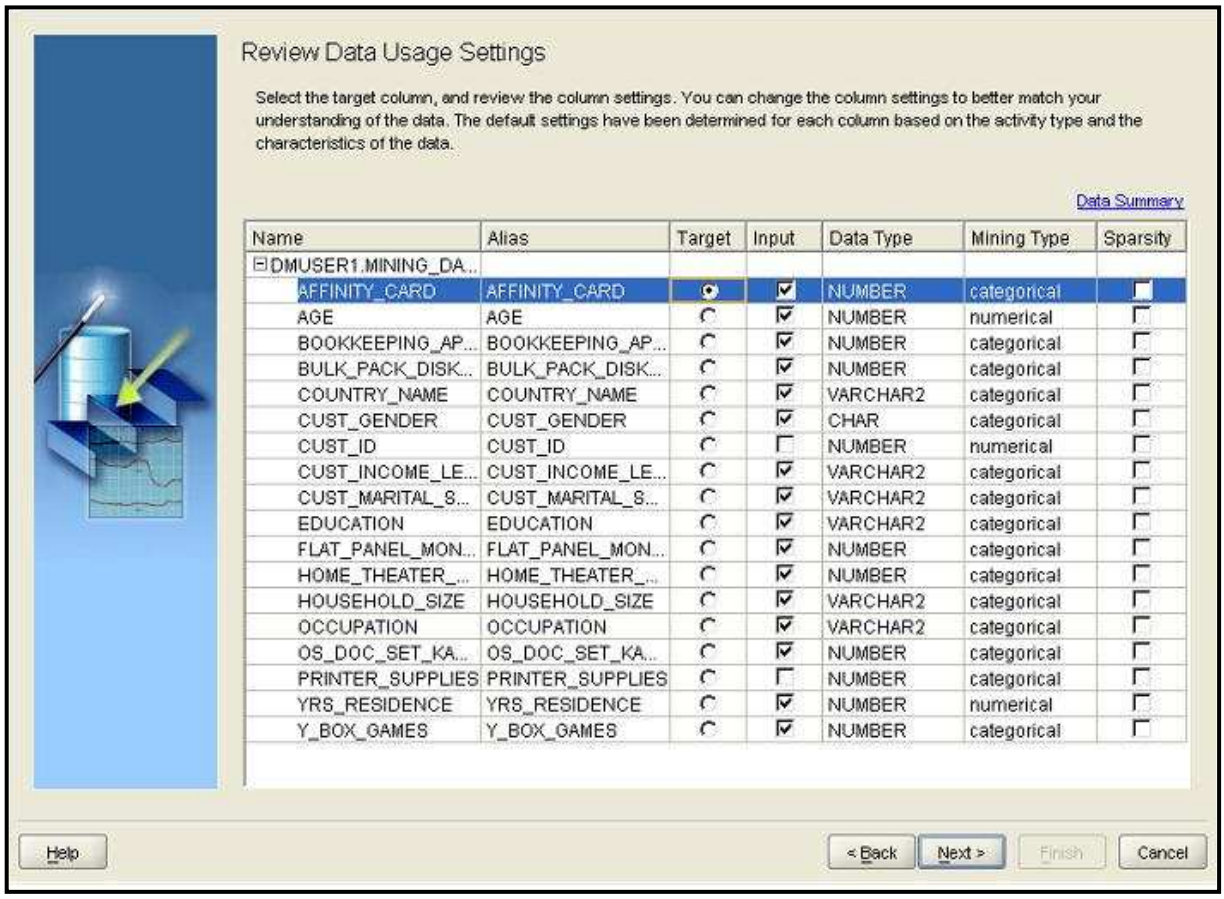



**Figura 19. AI Target**

Finalize os passos mantendo a checkbox *Run upon* Finish marcada, isso ira mostrar os passos do processo de construção do AI, a Figura 20 ilustra esses passos.

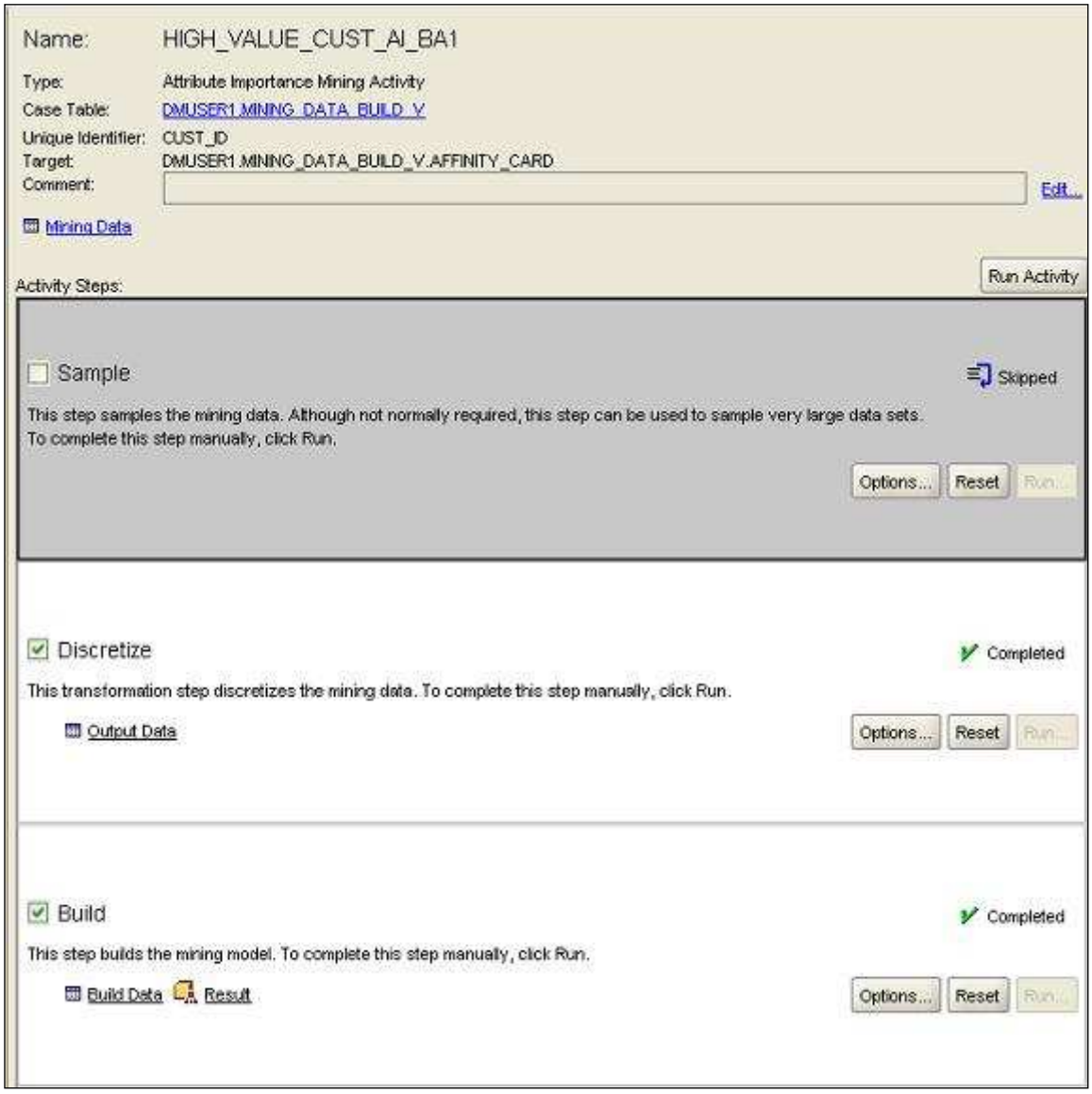
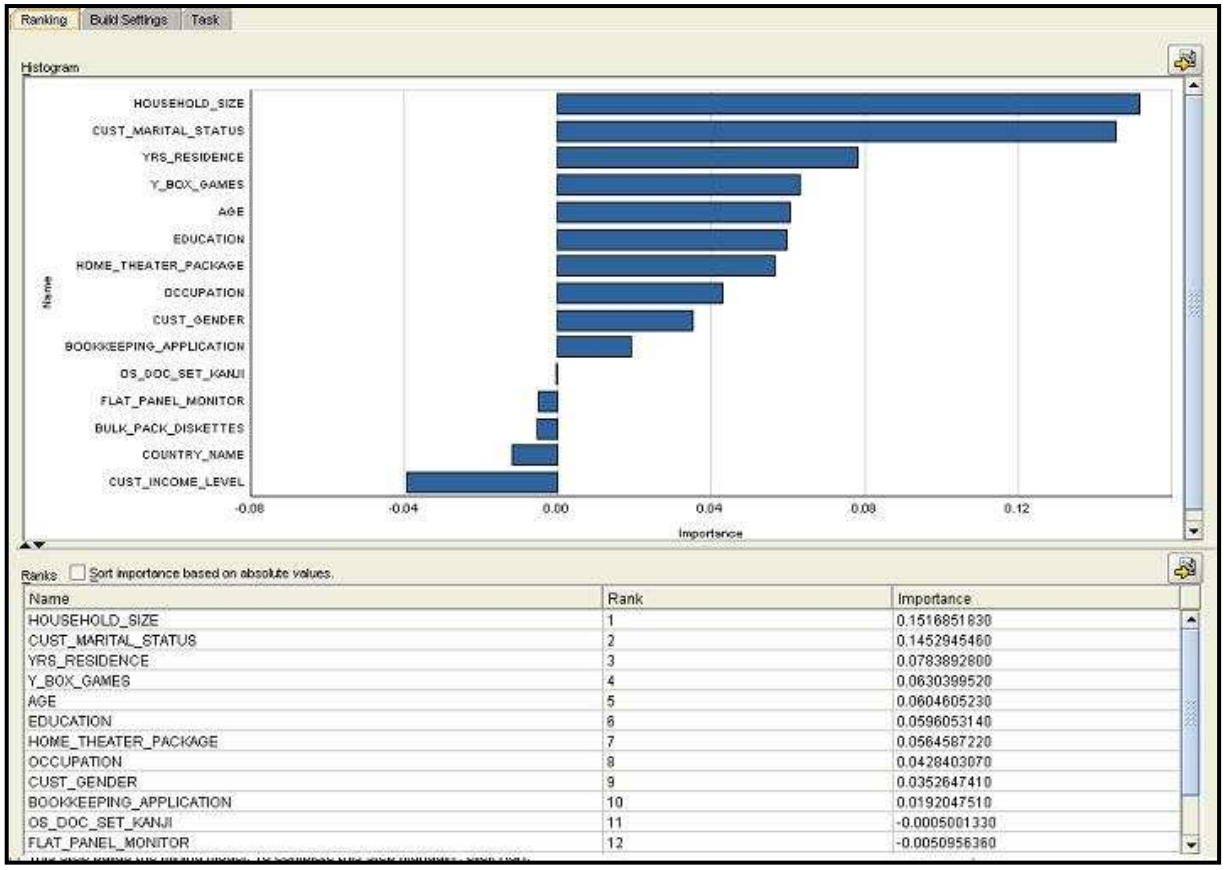



**Figura 20. AI Criação**

Quando todos os passos forem concluídos, clique na opção *Result* localizado no canto inferior da página para visualizar o gráfico de classificação dos atributos (Figura 21).




| Name | Rank | Importance |
|---|---|---|
| HOUSEHOLD_SIZE | 1 | 0.1516851830 |
| CUST_MARITAL_STATUS | 2 | 0.1452945460 |
| YRS_RESIDENCE | 3 | 0.0783892800 |
| Y_BOX_GAMES | 4 | 0.0630399520 |
| AGE | 5 | 0.0604605230 |
| EDUCATION | 6 | 0.0596053140 |
| HOME_THEATER_PACKAGE | 7 | 0.0564587220 |
| OCCUPATION | 8 | 0.0428403070 |
| CUST_GENDER | 9 | 0.0352647410 |
| BOOKKEEPING_APPLICATION | 10 | 0.0192047510 |
| OS_DOC_SET_KANJI | 11 | -0.0005001330 |
| FLAT_PANEL_MONITOR | 12 | -0.0050956360 |

**Figura 21. AI Resultado**

### 5.1.2*. ODM - Classification: Adaptive Bayes Network*

Adaptive Bayes Network uma das formas para solucionar os problemas de classificação para valores discretos como: 0 ou 1; Yes ou No; Low, Medium, or High; True ou False; Etc.ODM disponibiliza quatro opções para solucionar seu problema, convém ao usuário decidir qual a melhor escolha dentre os algoritmos disponíveis: *Adaptive Bayes Network, Decision Tree, Naive Bayes e Support Vector Machine.*

Para demonstrar a construção da atividade selecione a opção *Adptive Beyes Network,* iremos usar a tabela *MINING_DATA_BUILD_V* e o atributo *AFFINITY_CARD,* ao final da etapa podemos visualizar e ou modificar as opções avançadas clicando em *Advanced Settings,* a Figura 22 ilustra essas opções.

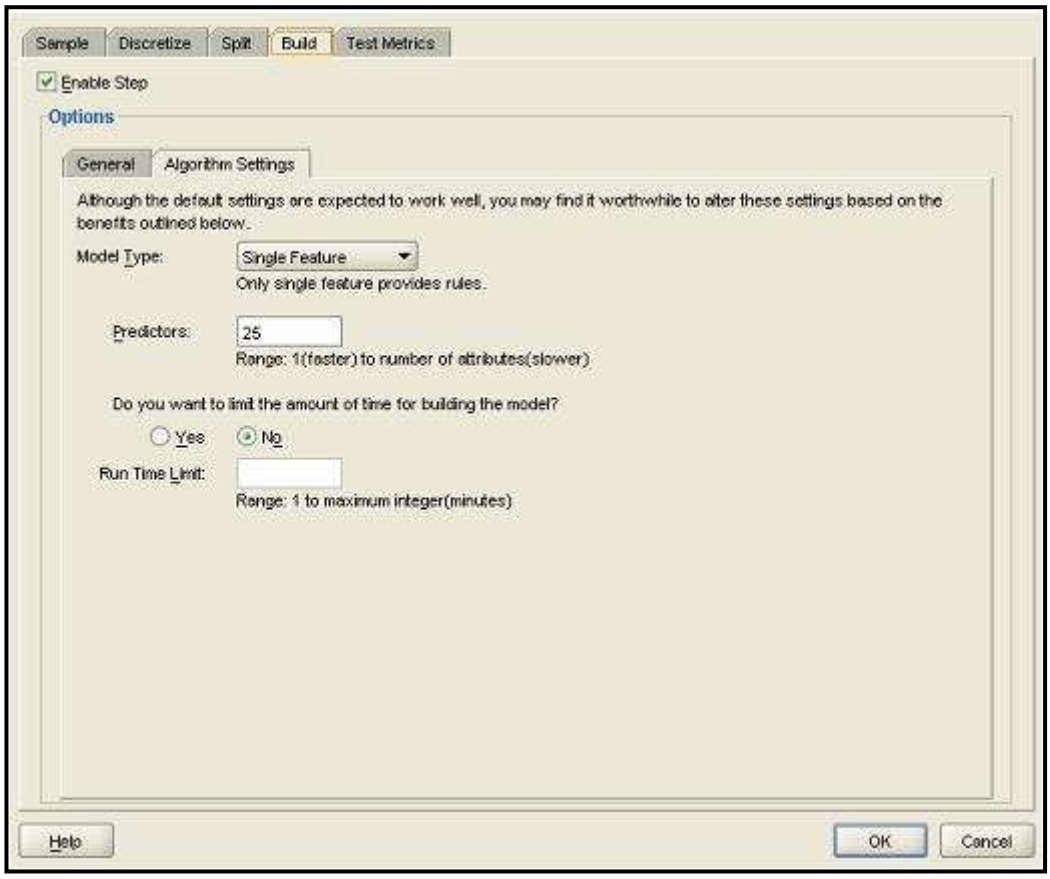



**Figura 22. ABN Advanced Settings**

Ao concluir as etapas podemos visualizar o resultado e verificar que foi montado uma pequena tabela com as informações importantes classificada pelo critério de suporte/confiança, como mostra a Figura 23.

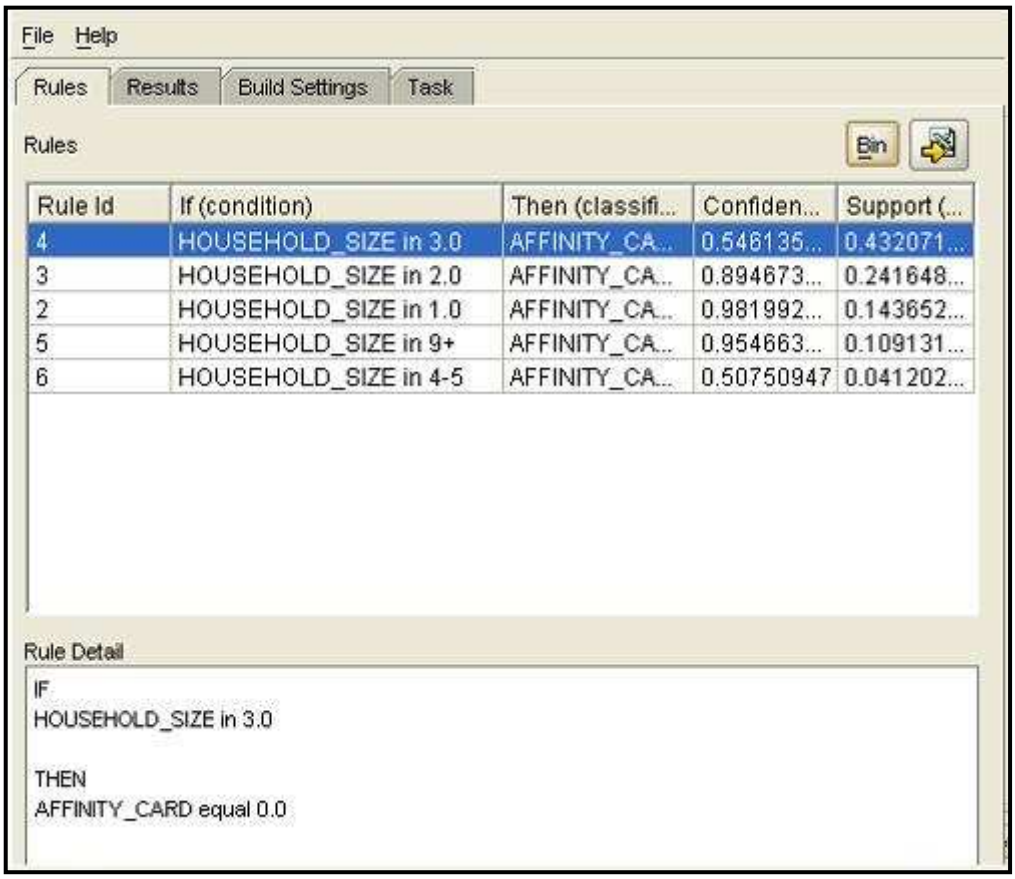



**Figura 23. ABN Result**

Podemos notar que ao final da etapa de processamento de construção da atividade, foi gerada uma linha com a opção *test metric (Figura 25),* essa opção oferece diferentes maneiras visuais de mostrar o resultado.

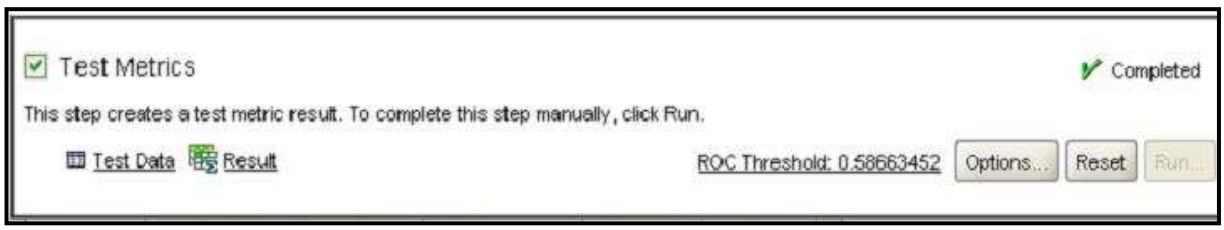



**Figura 24. Test Metric**

As próximas figuras ilustram as diversas características de resultados visuais classificatórios:

- **Predictive Confidence**

A Figura 25 é uma indicação visual da eficácia da atividade comparado com o palpite baseado na escolha do atributo *target*.

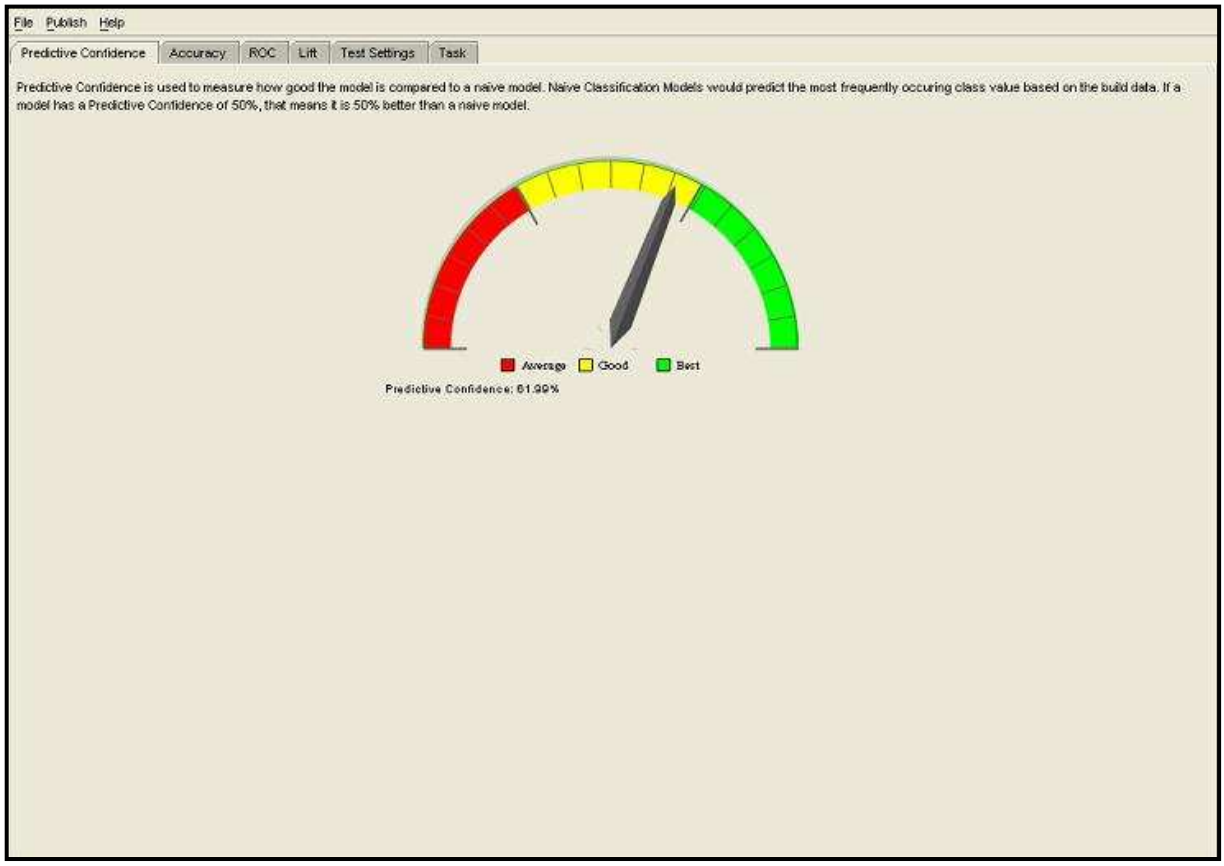



**Figura 25. Predictive Confidence**

- **Accuracy**

A Figura 26 é uma página que mostra um modelo de precisão aplicado no teste, onde as previsões são comparadas com os valores atuais.

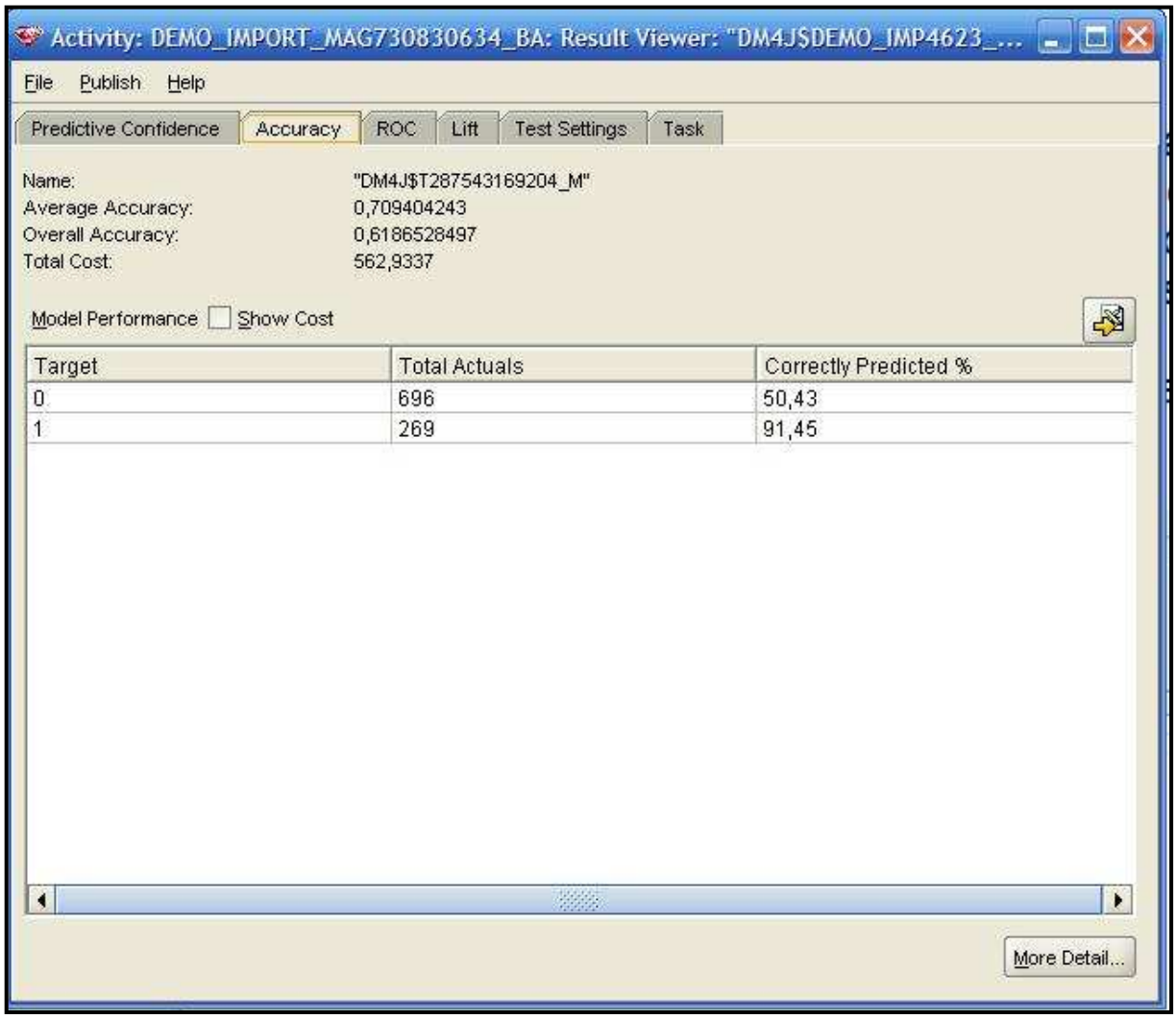



**Figura 26. Acuracy**

- **ROC**

A Figura 27 mostra as possíveis mudanças nos parâmetros através de um gráfico onde se observa os efeitos das alterações dos parâmetros.

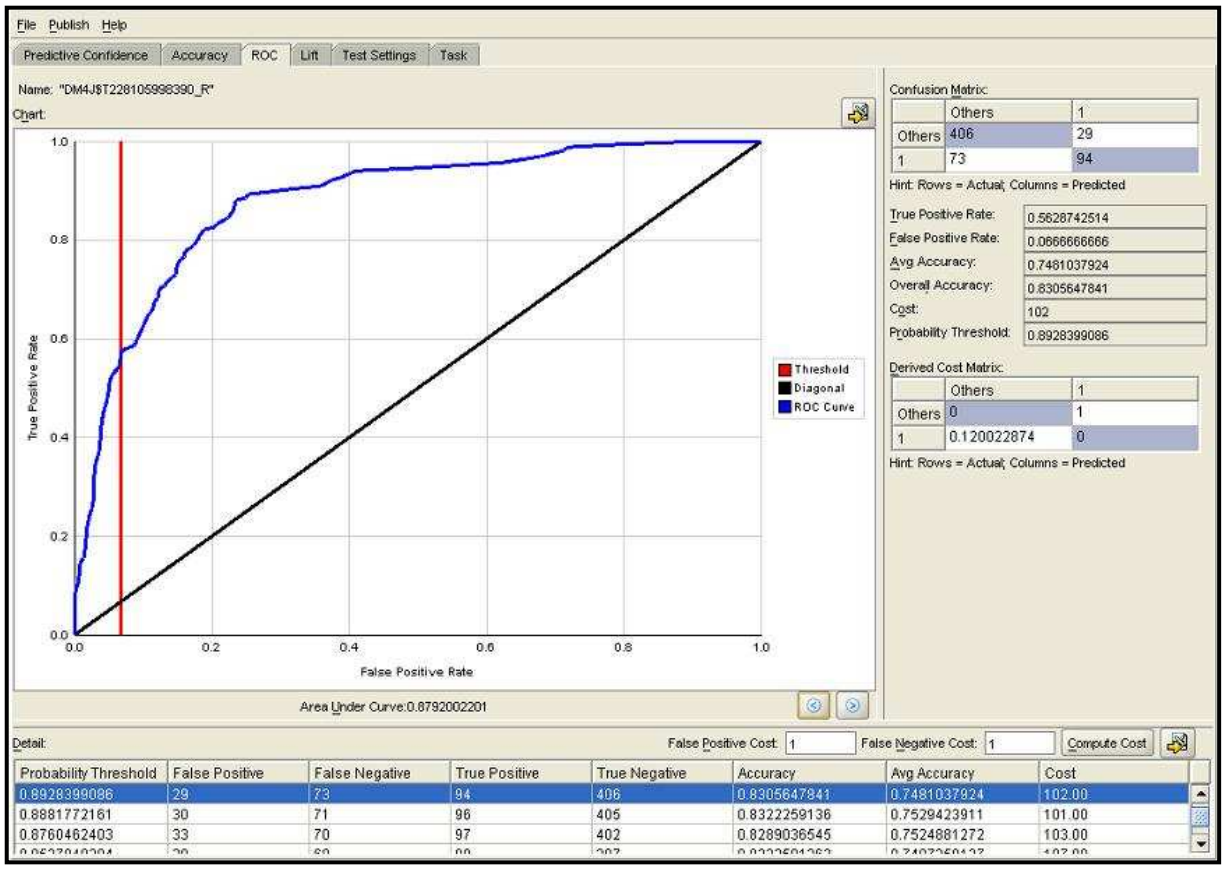



**Figura 27. ROC**

- **Lift**

A Figura 28 mostra os resultados obtidos através de dois gráficos com diferentes interpretações, esses resultados são obtidos através da junção entre a previsão e os valores atuais do *target.*

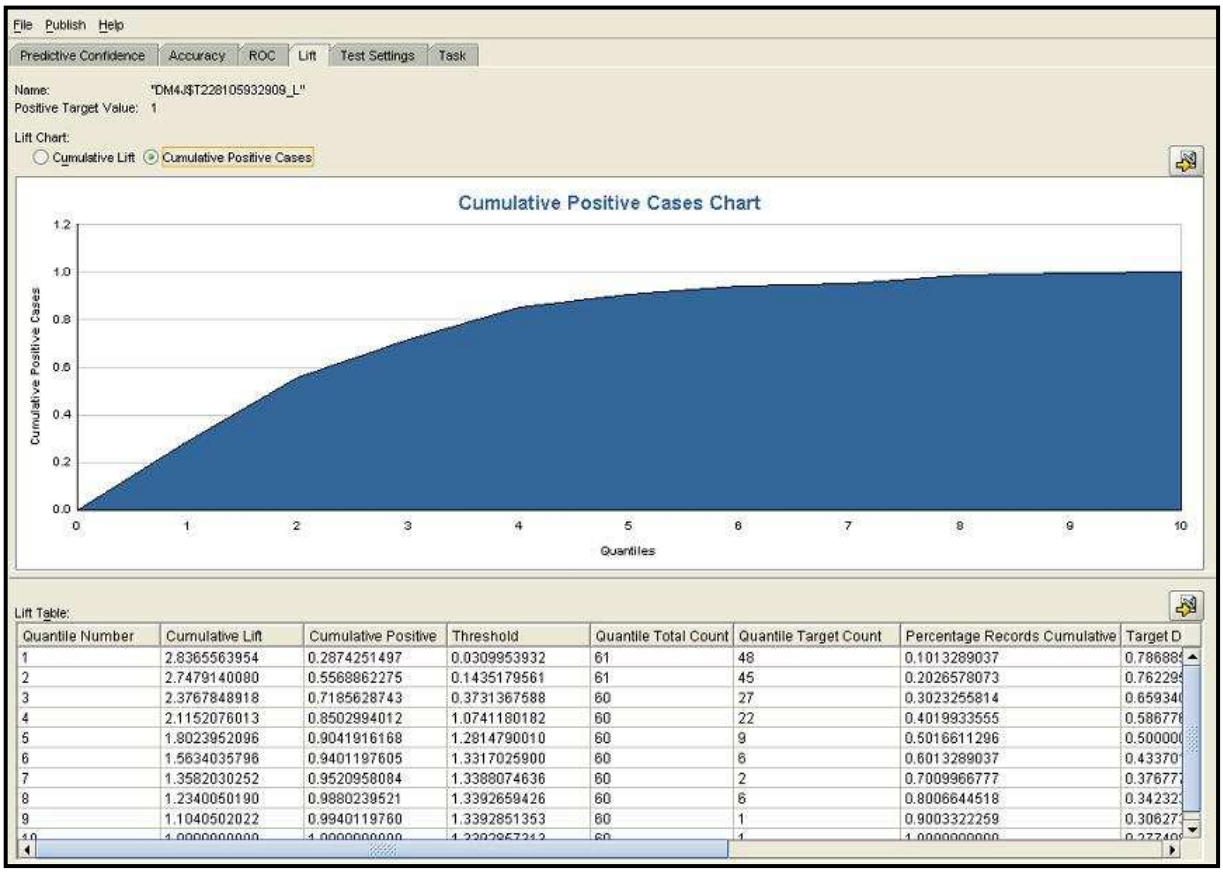



**Figura 28. Lift**

### 5.1.3. *Regression: Support Vector Machines*

O algoritmo SVM é usado para predizer a continuidade de valores, casualmente chamada regressão. O processo de construção da atividade é semelhante ao método de classificação, uma particularidade da regressão é o item *Result* em *Residual Plot* que indica em forma residual a diferença entre o valor atual e o valor predito, ilustrado na Figura 29.

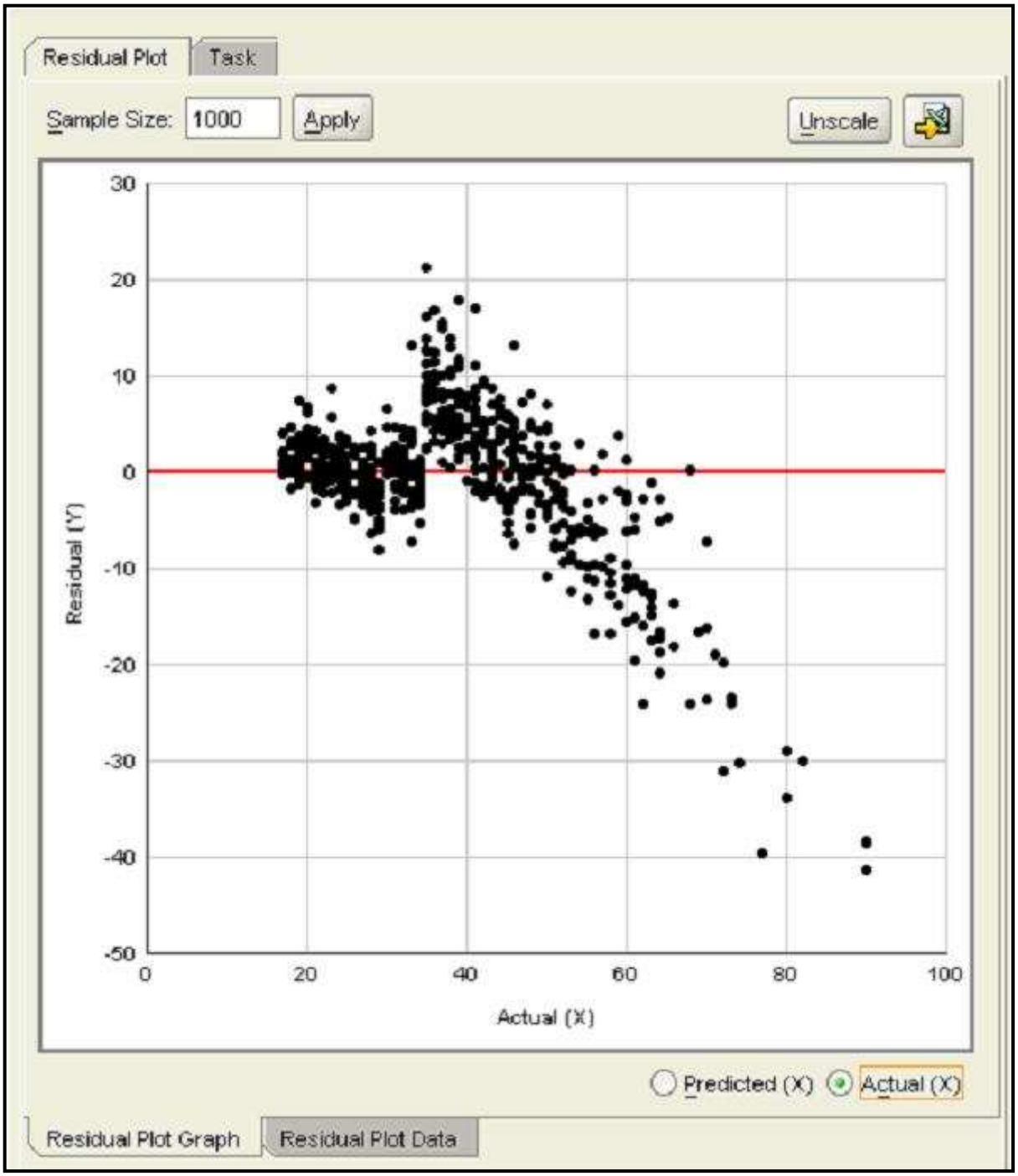



**Figura 29. Gráfico Residual**

Segue em anexo um código PL/SQL gerado pelo *ODM,* que descreve uma atividade associativa e utiliza o algoritmo Apriori para executar suas tarefas de mineração.

# 6. CONCLUSÃO

Este trabalho teve como meta mostrar as técnicas de mineração e seus algoritmos para auxiliar na tomada de decisões, para isso foram exigidas longas horas de pesquisas, onde existiu certa dificuldade para encontrar, na literatura, informações detalhadas sobre o assunto. A participação do XXIII Simpósio Brasileiro de Banco de Dados foi de extrema importância, com as informações extraídas do Mini-curso de *DM* e participação do IV Workshop em Algoritmos e Aplicações de Mineração de Dados. Apesar das dificuldades o resultado final foi satisfatório.

Durante a pesquisa foi notada a importância de uma tomada de decisão correta em uma empresa. Para utilizar as técnicas de mineração de dados é necessário um grande conhecimento sobre os fluxos de negócios da empresa e o usuário precisa ter domínio sobre ferramenta, assim como suas técnicas e regras. É de suma importância a presença de um profissional com conhecimento para trabalhar junto ao usuário auxiliando em suas duvidas técnicas e dando suporte.

Apesar de exigir um conhecimento técnico para utilizar o *ODM*, a ferramenta se mostrou muito eficaz e eficiente em suas tarefas, sua interface gráfica é agradável, seus resultados são de fáceis entendimentos assim como a maneira de configuração para o uso. Uma ferramenta altamente recomendada para as empresas realizarem técnicas de *DM* em seus dados.

# 7. REFERÊNCIAS BIBLIOGRÁFICAS

AMO, Sandra. Curso de Data Mining Ministrado na UFU Disponível em: <http://www.deamo.prof.ufu.br/CursoDM.html/>. Acesso em: 2 junho 2008.

AGRAWAL, Rakesh. Material produzido em 1994: “Fast Algorithms for Mining Association Rules”. Disponível em: <http://arbor.ee.ntu.edu.tw/~chyun/dmpaper/agrafa94.pdf/>. Acesso em: 11 maio 2008.

CAMARA, Fabio. **Informática Corporativa, Conceitos, Termos e Siglas.** Florianópolis: Visual Books, 2001.

CARVALHO, Luís Alfredo Vidal de. **Datamining: A mineração de dados no marketing, medicina, economia, engenharia e administração**. 2ª ed. São Paulo: Érica, 2001.

CARVALHO, A. P. de L. F. **Redes Neurais artificiais**. Disponível em: <http:// www.icmc.sc.usp.br >. Acesso em: 02 agosto 2008.

FAYYAD, U., SHAPIRO, G.P., **Data Mining and Knowledge Discovery in Databases: An overview**. Communications ACM, Special Issue on Data Mining. Novembro 1996.

FERREIRA, Jorge Brantes. Mineração **de Dados na Retenção de Clientes em Telefonia Celular**. PUC. Rio de Janeiro 2005.

GUIDE. **Guide to Data Mining. Introdução ao Data Mining**. Disponível em: <http://www.data-mining-guide.net/>. Acesso em: 07 agosto 2008.

JORGE, Alípio. “Extração de Conhecimentos de Dados II”, Disponível em: <http://www.liacc.up.pt/~amjorge/Aulas/madsad/ecd2/>. Acesso em: 15 julho 2008.

**KDD e Data Mining**. Disponível em <http://www.kdnuggets.com />. Acesso em: 11 setembro 2008.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Gerenciamento de Sistemas de Informação.** 3 ed. Rio de Janeiro, 2001.

LIEBSTEIN, Lourdes. **Artigo sobre Data Mining**. Disponível em: <http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_lourdes.pdf>. Acesso em: 26 julho 2008.

MIRANDA, Dhalila, REIS, Diego Belon. Projeto de iniciação cientifica sobre Data Mining. Disponível em: <http://www.inf.aedb.br/datamining/index.html/>. Acesso em: 25 maio 2008.

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

NOGUEIRA, Rilson. **Artigo sobre algoritmos genético na mineração de dados**. Disponível em: <http://www.frb.br/ciente/Textos%20CienteFico%202003.2/INFO/Banco%20de%20Dados/Os%20Algoritmos%20Gen%E9ticos%20na%20Minera%E7%E3o%20de%20Dados.pdf >. Acesso em: 5 maio 2008.

OGLIARI, Paulo José. Curso de Data Mining Disponível em: <http://www.inf.ufsc.br/~ogliari/cursodedatamining.html/>. Acesso em: 20 abril 2008.

ORACLE. **Oracle Data Mining**. Disponível em: <http://www.oracle.com/technology/products/bi/odm/index.html>. Acesso em: 01 maio 2008.

PILA, Adriano Donizete, **Datawarehouse.** Disponível em: <http://www.igce.unesp.br/igce.br/igce/grad/computacao/cintiab/datamine/datawarehouse.html>. Acesso em: 20 maio 2008.

SILVA, Fabio Alves; GOES, Leandro. **Estudo e Avaliação de Técnicas de Mineração de Dados no SGBD Comercial Oracle**. Centro Universitário Padre Anchieta. Jundiaí, São Paulo, 2007.

SOUZA, Michel de. **Data Mining**. Disponível em: <http://imasters.uol.com.br/artigo/1482/bi/data_mining/>. Acesso em: 9 setembro 2008.

SOARES, Jair, QUINTELLA, Rogério. Artigo sobre Descoberta de conhecimento em bases de dados públicas Disponível em: <http://www.sei.ba.gov.br/publicacoes/publicacoes_sei/bahia_analise/analise_dados/pdf/tecno_informa/jair_sampaio.pdf >. Acesso em: 21 junho 2008.

TURBAN, Efrain; MCLEAN, Ephraim; WETHERBE, James. **Tecnologia da Informação para gestão.** 3 ed. Porto Alegre: Bookman, 2004.

TWO CROWS. **Introduction to Data Mining and Knowledge Discovery.** 3 ed. Disponível em: <http://www.twocrows.com>. Acesso em: 02 outubro 2008.

ZANUSSO, Maria Bernadete. Trabalho Sobre Data Mining Disponível em <http://www.dct.ufms.br/~mzanusso/Data_Mining.htm/>. Acesso em: 16 maio 2008.

# ANEXO A

```
CREATE PACKAGE "DATAMININGACTIVITY1" AUTHID DEFINER AS

PROCEDURE "MINING_DATA_BUI48892982_BA"(case_table IN VARCHAR2 DEFAULT
'"DMSYS"."MINING_DATA_BUILD_V"',
                        additional_table_1 IN VARCHAR2 DEFAULT NULL,
                        model_name IN VARCHAR2 DEFAULT 'MINING_DATA_B2957_AB',
                        confusion_matrix_name IN VARCHAR2 DEFAULT '"DM4J$T885103350452_M"',
                        lift_result_name IN VARCHAR2 DEFAULT '"DM4J$T885103352846_L"',
                        roc_result_name IN VARCHAR2 DEFAULT '"DM4J$T88510337573_R"',
                        test_metric_name IN VARCHAR2 DEFAULT '"DM4J$MINING_D8096_TM"',
                        feature_table IN VARCHAR2 DEFAULT NULL,
                        mapping_table IN VARCHAR2 DEFAULT NULL,
                        drop_output IN BOOLEAN DEFAULT FALSE);

END;


/
CREATE PACKAGE BODY "DATAMININGACTIVITY1" AS

c_long_sql_statement_length CONSTANT INTEGER := 32767;

SUBTYPE SQL_STATEMENT_TYPE      IS VARCHAR2(32767);
SUBTYPE LONG_SQL_STATEMENT_TYPE IS DBMS_SQL.VARCHAR2A;

TYPE TABLE_ARRAY is TABLE OF VARCHAR2(62);
TYPE LSTMT_REC_TYPE IS RECORD (
    lstmt dbms_sql.VARCHAR2A,
    lb    BINARY_INTEGER DEFAULT 1,
    ub    BINARY_INTEGER DEFAULT 0);
TYPE LSTMT_REC_TYPE_ARRAY is TABLE OF LSTMT_REC_TYPE;
TYPE QUERY_ARRAY is TABLE OF SQL_STATEMENT_TYPE;
TYPE TARGET_VALUES_LIST      IS TABLE OF  VARCHAR2(32);
TYPE VALUE_COUNT_LIST        IS TABLE OF  NUMBER;

PROCEDURE dump_varchar2a(vc2a dbms_sql.VARCHAR2A) IS
  v_str varchar2(32767);
BEGIN
  DBMS_OUTPUT.PUT_LINE('dump_varchar2a:');
  FOR i IN 1..vc2a.COUNT LOOP
    v_str := vc2a(i);
    DBMS_OUTPUT.PUT_LINE(v_str);
  END LOOP;
END;

PROCEDURE ls_append(
  r_lstmt IN OUT NOCOPY LSTMT_REC_TYPE,
  p_txt   VARCHAR2)
IS
BEGIN
  r_lstmt.ub := r_lstmt.ub + 1;
  r_lstmt.lstmt(r_lstmt.ub) := p_txt;
END ls_append;

PROCEDURE ls_append(
  r_lstmt IN OUT NOCOPY LSTMT_REC_TYPE,
  p_txt   LSTMT_REC_TYPE) IS
BEGIN
  FOR i IN p_txt.lb..p_txt.ub LOOP
    r_lstmt.ub := r_lstmt.ub + 1;
    r_lstmt.lstmt(r_lstmt.ub) := p_txt.lstmt(i);
  END LOOP;
END ls_append;

FUNCTION query_valid(
  p_query VARCHAR2) RETURN BOOLEAN
IS
```

```
  v_is_valid  BOOLEAN;
BEGIN
  BEGIN
    EXECUTE IMMEDIATE p_query;
    v_is_valid := TRUE;
  EXCEPTION WHEN OTHERS THEN
    v_is_valid := FALSE;
  END;
  RETURN v_is_valid;
END query_valid;

FUNCTION table_exist(
  p_table_name  VARCHAR2) RETURN BOOLEAN IS
BEGIN
  RETURN query_valid('SELECT * FROM ' || dbms_assert.simple_sql_name(p_table_name));
END table_exist;

FUNCTION model_exist(
  p_model_name  VARCHAR2) RETURN BOOLEAN
IS
  v_model_cnt NUMBER;
  v_model_exists BOOLEAN := FALSE;
BEGIN
  SELECT COUNT(*) INTO v_model_cnt FROM DM_USER_MODELS WHERE NAME = UPPER(p_model_name);
  IF v_model_cnt > 0 THEN
    v_model_exists := TRUE;
  END IF;
  --DBMS_OUTPUT.PUT_LINE('model exist: '||v_model_exists);
  RETURN v_model_exists;
EXCEPTION WHEN OTHERS THEN
  RETURN FALSE;
END model_exist;

PROCEDURE drop_table(
  p_table_name  VARCHAR2)
IS
  v_stmt  SQL_STATEMENT_TYPE;
BEGIN
  v_stmt := 'DROP TABLE '||dbms_assert.simple_sql_name(p_table_name)||' PURGE';
  EXECUTE  IMMEDIATE v_stmt;
EXCEPTION WHEN OTHERS THEN
  NULL;
  --DBMS_OUTPUT.PUT_LINE('Failed drop_table: '||p_table_name);
END drop_table;

PROCEDURE drop_view(
  p_view_name  VARCHAR2)
IS
  v_stmt  SQL_STATEMENT_TYPE;
BEGIN
  v_stmt := 'DROP VIEW '||dbms_assert.simple_sql_name(p_view_name);
  EXECUTE  IMMEDIATE v_stmt;
EXCEPTION WHEN OTHERS THEN
  NULL;
  --DBMS_OUTPUT.PUT_LINE('Failed drop_view: '||p_view_name);
END drop_view;

PROCEDURE drop_model(
  p_model_name  VARCHAR2)
IS
  v_diagnostics_table VARCHAR2(30);
BEGIN
  DBMS_DATA_MINING.DROP_MODEL(p_model_name);
  SELECT SETTING_VALUE INTO v_diagnostics_table
  FROM TABLE(DBMS_DATA_MINING.GET_MODEL_SETTINGS(p_model_name))
  WHERE SETTING_NAME = 'GLMS_DIAGNOSTICS_TABLE_NAME';
  IF (v_diagnostics_table IS NOT NULL) THEN
    drop_table(v_diagnostics_table);
  END IF;
EXCEPTION WHEN OTHERS THEN
  NULL;
  --DBMS_OUTPUT.PUT_LINE('Failed drop_model: '||p_model_name);
END drop_model;
```

```
FUNCTION create_new_temp_table_name(prefix IN VARCHAR2, len IN NUMBER)
RETURN VARCHAR2 IS
  v_table_name      VARCHAR2(30);
  v_seed            NUMBER;
BEGIN
  dbms_random.seed(SYS_GUID());
  v_table_name := 'DM$T' || SUBSTR(prefix, 0, 4) || dbms_random.string(NULL, len-8);
  --DBMS_OUTPUT.PUT_LINE('create_new_temp_table_name: '||v_table_name);
  RETURN v_table_name;
END create_new_temp_table_name;

FUNCTION create_new_temp_table_name(prefix IN VARCHAR2)
RETURN VARCHAR2 IS
BEGIN
  RETURN create_new_temp_table_name(prefix, 30);
END create_new_temp_table_name;

FUNCTION ADD_TEMP_TABLE(tempTables IN OUT NOCOPY TABLE_ARRAY, temp_table IN VARCHAR2) RETURN
VARCHAR2 IS
BEGIN
  tempTables.EXTEND;
  tempTables(tempTables.COUNT) := temp_table;
  return temp_table;
END;

PROCEDURE DROP_TEMP_TABLES(tempTables IN OUT NOCOPY TABLE_ARRAY) IS
  v_temp  VARCHAR2(30);
BEGIN
  FOR i IN 1..tempTables.COUNT LOOP
    v_temp := tempTables(i);
    drop_table(v_temp);
    drop_view(v_temp);
    tempTables.DELETE(i);
  END LOOP;
END;

PROCEDURE CHECK_RESULTS(drop_output IN BOOLEAN,
                        result_name IN VARCHAR2) IS
BEGIN
  -- drop all results if drop = true, otherwise make sure all results don't exist already
(raise exception)
  IF result_name IS NOT NULL THEN
    IF drop_output THEN
      drop_table(result_name);
      drop_view(result_name);
    ELSIF (table_exist(result_name)) THEN
      RAISE_APPLICATION_ERROR(-20000, 'Result table exists: '||result_name);
    END IF;
  END IF;
END;

PROCEDURE CHECK_MODEL(drop_output IN BOOLEAN,
                      model_name IN VARCHAR2) IS
BEGIN
  -- drop all results if drop = true, otherwise make sure all results don't exist already
(raise exception)
  IF model_name IS NOT NULL THEN
    IF drop_output THEN
      drop_model(model_name);
    ELSIF (model_exist(model_name)) THEN
      RAISE_APPLICATION_ERROR(-20001, 'Model exists: '||model_name);
    END IF;
  END IF;
END;

PROCEDURE create_table_from_query(query IN OUT NOCOPY LSTMT_REC_TYPE)
IS
  v_cursor      NUMBER;
  v_feedback    INTEGER;
BEGIN
  v_cursor := DBMS_SQL.OPEN_CURSOR;

  DBMS_SQL.PARSE(
    c             => v_cursor,
```

```
    statement     => query.lstmt,
    lb            => query.lb,
    ub            => query.ub,
    lfflg         => FALSE,
    language_flag => dbms_sql.native);
  v_feedback := DBMS_SQL.EXECUTE(v_cursor);
  DBMS_SQL.CLOSE_CURSOR(v_cursor);

EXCEPTION WHEN OTHERS THEN
  IF DBMS_SQL.IS_OPEN(v_cursor) THEN
    DBMS_SQL.CLOSE_CURSOR(v_cursor);
  END IF;
  RAISE;
END;

FUNCTION get_row_count(tableName IN VARCHAR2)
RETURN INTEGER IS
   v_stmt  VARCHAR(100);
   qcount INTEGER := 0;
BEGIN
  v_stmt := 'SELECT COUNT(*) FROM '|| tableName;
  EXECUTE  IMMEDIATE v_stmt INTO qcount;
  RETURN qcount;
END get_row_count;

PROCEDURE SET_EQUAL_DISTRIBUTION (
   counts IN OUT VALUE_COUNT_LIST )
IS
  v_minvalue          NUMBER := 0;
BEGIN
  FOR i IN counts.FIRST..counts.LAST
  LOOP
    IF ( i = counts.FIRST )
      THEN
         v_minvalue := counts(i);
      ELSIF ( counts(i) > 0 AND v_minvalue > counts(i) )
      THEN
        v_minvalue := counts(i);
    END IF;
  END LOOP;

  FOR i IN counts.FIRST..counts.LAST
  LOOP
    counts(i) := v_minvalue;
  END LOOP;
END SET_EQUAL_DISTRIBUTION;

PROCEDURE GET_STRATIFIED_DISTRIBUTION (
   table_name         VARCHAR2,
   attribute_name     VARCHAR2,
   percentage         NUMBER,
   attr_values    IN OUT NOCOPY TARGET_VALUES_LIST,
   counts         IN OUT NOCOPY VALUE_COUNT_LIST,
   counts_sampled IN OUT NOCOPY VALUE_COUNT_LIST )
IS
  v_tmp_stmt           VARCHAR2(4000);
BEGIN
  v_tmp_stmt :=
      'SELECT /*+ noparallel(t)*/ ' || attribute_name ||
      ', count(*), ROUND ( ( count(*) * ' || percentage || ') / 100.0 ) FROM '|| table_name ||
      ' WHERE ' || attribute_name ||' IS NOT NULL GROUP BY ' || attribute_name;
  EXECUTE IMMEDIATE v_tmp_stmt
  BULK COLLECT INTO attr_values, counts, counts_sampled;
END GET_STRATIFIED_DISTRIBUTION;

FUNCTION GENERATE_STRATIFIED_SQL (
                       v_2d_temp_view    VARCHAR2,
                       src_table_name    VARCHAR2,
                       attr_names        TARGET_VALUES_LIST,
                       attribute_name    VARCHAR2,
                       percentage        NUMBER,
                       op                VARCHAR2,
                       equal_distribution IN BOOLEAN DEFAULT FALSE) RETURN LSTMT_REC_TYPE
IS
```

```
  v_tmp_lstmt           LSTMT_REC_TYPE;
  attr_values_res       TARGET_VALUES_LIST;
  counts_res            VALUE_COUNT_LIST;
  counts_sampled_res    VALUE_COUNT_LIST;
  tmp_str               VARCHAR2(4000);
  sample_count          PLS_INTEGER;

BEGIN
  GET_STRATIFIED_DISTRIBUTION(src_table_name, attribute_name, percentage, attr_values_res,
counts_res, counts_sampled_res);
  IF ( equal_distribution = TRUE )
  THEN
    SET_EQUAL_DISTRIBUTION(counts_sampled_res);
  END IF;

  v_tmp_lstmt.ub := 0; -- initialize
  ls_append(v_tmp_lstmt, 'CREATE TABLE ');
  ls_append(v_tmp_lstmt, v_2d_temp_view);
  ls_append(v_tmp_lstmt, ' AS ');
  ls_append(v_tmp_lstmt, '( SELECT ');

  FOR i IN attr_names.FIRST..attr_names.LAST
  LOOP
    IF ( i != attr_names.FIRST )
    THEN
       ls_append(v_tmp_lstmt,',');
    END IF;

    ls_append(v_tmp_lstmt, attr_names(i));
  END LOOP;

  ls_append(v_tmp_lstmt, ' FROM (SELECT /*+ no_merge */ t.*, row_number() over(partition by
'||attribute_name||' order by ora_hash(ROWNUM)) RNUM FROM ' || src_table_name || ' t) WHERE
RNUM = 1 OR ');

  FOR i IN attr_values_res.FIRST..attr_values_res.LAST
  LOOP
    IF ( i != attr_values_res.FIRST )
    THEN
       tmp_str := ' OR ';
    END IF;
    IF ( counts_res(i) <= 2 ) THEN
      sample_count := counts_res(i);
    ELSE
      sample_count := counts_sampled_res(i);
    END IF;
    tmp_str := tmp_str ||
     '( ' || attribute_name || ' = ''' || REPLACE(attr_values_res(i), '''', '''''') || '''' ||
      ' AND ORA_HASH(RNUM,(' || counts_res(i) || ' -1),12345) ' || op || sample_count || ') ';
    ls_append(v_tmp_lstmt, tmp_str );
  END LOOP;
  ls_append(v_tmp_lstmt, ') ');
  return v_tmp_lstmt;
END GENERATE_STRATIFIED_SQL;


PROCEDURE "MINING_DATA_BUI48892982_BA"(case_table IN VARCHAR2 DEFAULT
'"DMSYS"."MINING_DATA_BUILD_V"',
                       additional_table_1 IN VARCHAR2 DEFAULT NULL,
                       model_name IN VARCHAR2 DEFAULT 'MINING_DATA_B2957_AB',
                       confusion_matrix_name IN VARCHAR2 DEFAULT '"DM4J$T885103350452_M"',
                       lift_result_name IN VARCHAR2 DEFAULT '"DM4J$T885103352846_L"',
                       roc_result_name IN VARCHAR2 DEFAULT '"DM4J$T88510337573_R"',
                       test_metric_name IN VARCHAR2 DEFAULT '"DM4J$MINING_D8096_TM"',
                       feature_table IN VARCHAR2 DEFAULT NULL,
                       mapping_table IN VARCHAR2 DEFAULT NULL,
                       drop_output IN BOOLEAN DEFAULT FALSE)
IS
  additional_data  TABLE_ARRAY := TABLE_ARRAY(
    additional_table_1
  );
  v_tempTables              TABLE_ARRAY := TABLE_ARRAY();
  v_2d_view                 VARCHAR2(30);
  v_2d_view_build           VARCHAR2(30);
```

```
  v_2d_view_test            VARCHAR2(30);
  v_2d_temp_view            VARCHAR2(30);
  v_txn_views               TABLE_ARRAY := TABLE_ARRAY();
  v_txn_views_build         TABLE_ARRAY := TABLE_ARRAY();
  v_txn_views_test          TABLE_ARRAY := TABLE_ARRAY();
  v_txn_temp_views          TABLE_ARRAY := TABLE_ARRAY();
  v_case_data               SQL_STATEMENT_TYPE := case_table;
  v_case_id                 VARCHAR2(30) := 'DMR$CASE_ID';
  v_tmp_lstmt               LSTMT_REC_TYPE;
  v_target_value            VARCHAR2(4000) := '1';
  v_num_quantiles           NUMBER := 10;
  v_build_data              VARCHAR2(30);
  v_test_data               VARCHAR2(30);
  v_prior                   VARCHAR2(30);
  v_build_setting           VARCHAR2(30);
  v_apply_result            VARCHAR2(30);
  v_build_cm                VARCHAR2(30);
  v_test_cm                 VARCHAR2(30);
  v_diagnostics_table       VARCHAR2(30);
  v_accuracy                NUMBER;
  v_area_under_curve        NUMBER;
  v_avg_accuracy            NUMBER;
  v_predictive_confidence   NUMBER;
  v_confusion_matrix        VARCHAR2(30);
  v_gen_caseId              BOOLEAN := FALSE;
  v_txt_build               VARCHAR2(30);
  v_txt_test                VARCHAR2(30);
  v_content_index           VARCHAR2(30);
  v_content_index_pref      VARCHAR2(30);
  v_category_temp_table     VARCHAR2(30);
  v_term_final_table        VARCHAR2(30);
  v_term_final_table_index  VARCHAR2(30);
  v_mapping_table_index     VARCHAR2(30);
  v_term_final_table_test   VARCHAR2(30);
  pragma autonomous_transaction;
BEGIN
  execute immediate 'Alter session set NLS_NUMERIC_CHARACTERS=".,"';

  CHECK_MODEL(drop_output, model_name);
  CHECK_RESULTS(drop_output, feature_table);
  CHECK_RESULTS(drop_output, mapping_table);
  CHECK_RESULTS(drop_output, test_metric_name);
  CHECK_RESULTS(drop_output, confusion_matrix_name);
  CHECK_RESULTS(drop_output, lift_result_name);
  CHECK_RESULTS(drop_output, roc_result_name);

  IF (v_gen_caseId) THEN
    v_case_data := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
    EXECUTE IMMEDIATE 'CREATE TABLE '||v_case_data||' as SELECT rownum as DMR$CASE_ID, t.*
FROM ('||case_table||') t ';
    EXECUTE IMMEDIATE 'ALTER TABLE '||v_case_data||' add constraint
'||create_new_temp_table_name('PK')||' primary key (DMR$CASE_ID)';
  END IF;

  ----- Start: Input Data Preparation -----
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  ls_append(v_tmp_lstmt, 'CREATE VIEW ');
  ls_append(v_tmp_lstmt, v_2d_temp_view);
  ls_append(v_tmp_lstmt, ' AS ');
  ls_append(v_tmp_lstmt, ' ( ');
  ls_append(v_tmp_lstmt, 'SELECT "MINING_DATA_BUILD_V"."CUST_ID" as "DMR$CASE_ID",  TO_CHAR(
"MINING_DATA_BUILD_V"."AFFINITY_CARD") AS "AFFINITY_CARD",
 "MINING_DATA_BUILD_V"."AGE" AS "AGE",
 TO_CHAR( "MINING_DATA_BUILD_V"."BOOKKEEPING_APPLICATION") AS "BOOKKEEPING_APPLICATION",
 TO_CHAR( "MINING_DATA_BUILD_V"."BULK_PACK_DISKETTES") AS "BULK_PACK_DISKETTES",
 "MINING_DATA_BUILD_V"."COUNTRY_NAME" AS "COUNTRY_NAME",
 "MINING_DATA_BUILD_V"."CUST_GENDER" AS "CUST_GENDER",
 "MINING_DATA_BUILD_V"."CUST_INCOME_LEVEL" AS "CUST_INCOME_LEVEL",
 "MINING_DATA_BUILD_V"."CUST_MARITAL_STATUS" AS "CUST_MARITAL_STATUS",
 "MINING_DATA_BUILD_V"."EDUCATION" AS "EDUCATION",
 TO_CHAR( "MINING_DATA_BUILD_V"."FLAT_PANEL_MONITOR") AS "FLAT_PANEL_MONITOR",
 TO_CHAR( "MINING_DATA_BUILD_V"."HOME_THEATER_PACKAGE") AS "HOME_THEATER_PACKAGE",
 "MINING_DATA_BUILD_V"."HOUSEHOLD_SIZE" AS "HOUSEHOLD_SIZE",
 "MINING_DATA_BUILD_V"."OCCUPATION" AS "OCCUPATION",
```

```
 TO_CHAR( "MINING_DATA_BUILD_V"."OS_DOC_SET_KANJI") AS "OS_DOC_SET_KANJI",
 TO_CHAR( "MINING_DATA_BUILD_V"."Y_BOX_GAMES") AS "Y_BOX_GAMES",
 "MINING_DATA_BUILD_V"."YRS_RESIDENCE" AS "YRS_RESIDENCE" FROM (' || v_case_data || ')
"MINING_DATA_BUILD_V" ');
  ls_append(v_tmp_lstmt, ' ) ');
  create_table_from_query(v_tmp_lstmt);
  v_2d_view := v_2d_temp_view;

  ----- End: Input Data Preparation -----



  ----- Start: Discretize Transformation -----
  v_tmp_lstmt.ub := 0; -- initialize
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  ls_append(v_tmp_lstmt, 'CREATE VIEW ');
  ls_append(v_tmp_lstmt, v_2d_temp_view);
  ls_append(v_tmp_lstmt, ' AS ');
  ls_append(v_tmp_lstmt, ' ( ');
  ls_append(v_tmp_lstmt, 'SELECT  "AFFINITY_CARD",( CASE WHEN "AGE" < 27 THEN 1
WHEN "AGE" <= 34 THEN 2
WHEN "AGE" <= 42 THEN 3
WHEN "AGE" <= 51 THEN 4
WHEN "AGE" > 51 THEN 5
 end)  "AGE", "BOOKKEEPING_APPLICATION", "BULK_PACK_DISKETTES", DECODE ("COUNTRY_NAME"
,''United States of America'',''United States of America''
,''Argentina'',''Argentina''
,''Italy'',''Italy''
,''Brazil'',''Brazil''
,''Canada'',''Canada''
,NULL,NULL,''other'') "COUNTRY_NAME", "CUST_GENDER", DECODE ("CUST_INCOME_LEVEL"
,''J: 190,000 - 249,999'',''J: 190,000 - 249,999''
,''L: 300,000 and above'',''L: 300,000 and above''
,''I: 170,000 - 189,999'',''I: 170,000 - 189,999''
,''K: 250,000 - 299,999'',''K: 250,000 - 299,999''
,''F: 110,000 - 129,999'',''F: 110,000 - 129,999''
,NULL,NULL,''other'') "CUST_INCOME_LEVEL", DECODE ("CUST_MARITAL_STATUS"
,''Married'',''Married''
,''NeverM'',''NeverM''
,''Divorc.'',''Divorc.''
,''Separ.'',''Separ.''
,''Widowed'',''Widowed''
,NULL,NULL,''other'') "CUST_MARITAL_STATUS", "DMR$CASE_ID", DECODE ("EDUCATION"
,''HS-grad'',''HS-grad''
,''< Bach.'',''< Bach.''
,''Bach.'',''Bach.''
,''Masters'',''Masters''
,''Assoc-V'',''Assoc-V''
,NULL,NULL,''other'') "EDUCATION", "FLAT_PANEL_MONITOR", "HOME_THEATER_PACKAGE", DECODE
("HOUSEHOLD_SIZE"
,''3'',''3''
,''2'',''2''
,''1'',''1''
,''9+'',''9+''
,''4-5'',''4-5''
,NULL,NULL,''other'') "HOUSEHOLD_SIZE", DECODE ("OCCUPATION"
,''Crafts'',''Crafts''
,''Exec.'',''Exec.''
,''Cleric.'',''Cleric.''
,''Sales'',''Sales''
,''Prof.'',''Prof.''
,NULL,NULL,''other'') "OCCUPATION", "OS_DOC_SET_KANJI", "Y_BOX_GAMES",( CASE WHEN
"YRS_RESIDENCE" < 3 THEN 1
WHEN "YRS_RESIDENCE" <= 4 THEN 2
WHEN "YRS_RESIDENCE" <= 5 THEN 3
WHEN "YRS_RESIDENCE" > 5 THEN 4
 end)  "YRS_RESIDENCE" FROM ');
  ls_append(v_tmp_lstmt, v_2d_view);
  ls_append(v_tmp_lstmt, ' ) ');
  create_table_from_query(v_tmp_lstmt);
  v_2d_view := v_2d_temp_view;

  ----- End: Discretize Transformation -----
```

```
  ----- Start: Stratified Split Transformation -----
  v_tmp_lstmt.ub := 0; -- initialize
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
ls_append(v_tmp_lstmt, GENERATE_STRATIFIED_SQL(v_2d_temp_view, v_2d_view,
TARGET_VALUES_LIST('"AFFINITY_CARD"',
'"AGE"',
'"BOOKKEEPING_APPLICATION"',
'"BULK_PACK_DISKETTES"',
'"COUNTRY_NAME"',
'"CUST_GENDER"',
'"CUST_INCOME_LEVEL"',
'"CUST_MARITAL_STATUS"',
'"DMR$CASE_ID"',
'"EDUCATION"',
'"FLAT_PANEL_MONITOR"',
'"HOME_THEATER_PACKAGE"',
'"HOUSEHOLD_SIZE"',
'"OCCUPATION"',
'"OS_DOC_SET_KANJI"',
'"Y_BOX_GAMES"',
'"YRS_RESIDENCE"'), '"AFFINITY_CARD"', 60, ' < ' ));
  create_table_from_query(v_tmp_lstmt);
  v_2d_view_build := v_2d_temp_view;

  v_tmp_lstmt.ub := 0; -- initialize
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
ls_append(v_tmp_lstmt, GENERATE_STRATIFIED_SQL(v_2d_temp_view, v_2d_view,
TARGET_VALUES_LIST('"AFFINITY_CARD"',
'"AGE"',
'"BOOKKEEPING_APPLICATION"',
'"BULK_PACK_DISKETTES"',
'"COUNTRY_NAME"',
'"CUST_GENDER"',
'"CUST_INCOME_LEVEL"',
'"CUST_MARITAL_STATUS"',
'"DMR$CASE_ID"',
'"EDUCATION"',
'"FLAT_PANEL_MONITOR"',
'"HOME_THEATER_PACKAGE"',
'"HOUSEHOLD_SIZE"',
'"OCCUPATION"',
'"OS_DOC_SET_KANJI"',
'"Y_BOX_GAMES"',
'"YRS_RESIDENCE"'), '"AFFINITY_CARD"', 60, ' >= ' ));
  create_table_from_query(v_tmp_lstmt);
  v_2d_view_test := v_2d_temp_view;

  ----- End: Stratified Split Transformation -----



  ----- Start: Mining Data Preparation -----
  v_tmp_lstmt.ub := 0; -- initialize
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  ls_append(v_tmp_lstmt, 'CREATE VIEW ');
  ls_append(v_tmp_lstmt, v_2d_temp_view);
  ls_append(v_tmp_lstmt, ' AS ');
  ls_append(v_tmp_lstmt, ' ( ');
  ls_append(v_tmp_lstmt,
  'SELECT caseTable."AFFINITY_CARD"
, caseTable."AGE"
, caseTable."BOOKKEEPING_APPLICATION"
, caseTable."BULK_PACK_DISKETTES"
, caseTable."COUNTRY_NAME"
, caseTable."CUST_GENDER"
, caseTable."CUST_INCOME_LEVEL"
, caseTable."CUST_MARITAL_STATUS"
, caseTable."DMR$CASE_ID"
, caseTable."EDUCATION"
, caseTable."FLAT_PANEL_MONITOR"
, caseTable."HOME_THEATER_PACKAGE"
, caseTable."HOUSEHOLD_SIZE"
, caseTable."OCCUPATION"
, caseTable."OS_DOC_SET_KANJI"
```

```
, caseTable."Y_BOX_GAMES"
, caseTable."YRS_RESIDENCE"
 FROM ('); ls_append(v_tmp_lstmt, v_2d_view_build); ls_append(v_tmp_lstmt, ') caseTable

'
  );
  ls_append(v_tmp_lstmt, ' ) ');
  create_table_from_query(v_tmp_lstmt);
  v_2d_view_build := v_2d_temp_view;

  v_tmp_lstmt.ub := 0; -- initialize
  v_2d_temp_view := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  ls_append(v_tmp_lstmt, 'CREATE VIEW ');
  ls_append(v_tmp_lstmt, v_2d_temp_view);
  ls_append(v_tmp_lstmt, ' AS ');
  ls_append(v_tmp_lstmt, ' ( ');
  ls_append(v_tmp_lstmt,
  'SELECT caseTable."AFFINITY_CARD"
, caseTable."AGE"
, caseTable."BOOKKEEPING_APPLICATION"
, caseTable."BULK_PACK_DISKETTES"
, caseTable."COUNTRY_NAME"
, caseTable."CUST_GENDER"
, caseTable."CUST_INCOME_LEVEL"
, caseTable."CUST_MARITAL_STATUS"
, caseTable."DMR$CASE_ID"
, caseTable."EDUCATION"
, caseTable."FLAT_PANEL_MONITOR"
, caseTable."HOME_THEATER_PACKAGE"
, caseTable."HOUSEHOLD_SIZE"
, caseTable."OCCUPATION"
, caseTable."OS_DOC_SET_KANJI"
, caseTable."Y_BOX_GAMES"
, caseTable."YRS_RESIDENCE"
 FROM ('); ls_append(v_tmp_lstmt, v_2d_view_test); ls_append(v_tmp_lstmt, ') caseTable

'
  );
  ls_append(v_tmp_lstmt, ' ) ');
  create_table_from_query(v_tmp_lstmt);
  v_2d_view_test := v_2d_temp_view;

  v_build_data := v_2d_view_build;
  v_test_data := v_2d_view_test;

  ----- End: Mining Data Preparation -----



  v_prior := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  EXECUTE IMMEDIATE 'CREATE TABLE ' || v_prior || ' (TARGET_VALUE VARCHAR2(4000),
PRIOR_PROBABILITY NUMBER)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_prior || ' VALUES (''0'', 0.7466670000000001)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_prior || ' VALUES (''1'', 0.2533329999999999)';
  COMMIT;



  v_build_setting := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  EXECUTE IMMEDIATE 'CREATE TABLE ' || v_build_setting || ' (setting_name VARCHAR2(30),
setting_value VARCHAR2(128))';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''ABNS_MAX_BUILD_MINUTES'',
''0'')';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''ABNS_MAX_NB_PREDICTORS'',
''10'')';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''ABNS_MAX_PREDICTORS'',
''25'')';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''ABNS_MODEL_TYPE'',
''ABNS_SINGLE_FEATURE'')';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''ALGO_NAME'',
''ALGO_ADAPTIVE_BAYES_NETWORK'')';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_build_setting || ' VALUES (''CLAS_PRIORS_TABLE_NAME'',
:priorTable)' USING v_prior;
```

```
  COMMIT;


  -- BUILD MODEL
  DBMS_DATA_MINING.CREATE_MODEL(
    model_name          => model_name,
    mining_function     => dbms_data_mining.classification,
    data_table_name     => v_build_data,
    case_id_column_name => v_case_id,
    target_column_name  => 'AFFINITY_CARD',
    settings_table_name => v_build_setting);

  v_test_cm := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
  EXECUTE IMMEDIATE 'CREATE TABLE ' || v_test_cm || ' (actual_target_value VARCHAR2(4000),
predicted_target_value VARCHAR2(4000), cost NUMBER)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_test_cm || ' VALUES (''0'', ''0'', 0.0)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_test_cm || ' VALUES (''0'', ''1'',
1.3392857142857142)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_test_cm || ' VALUES (''1'', ''0'',
3.9473684210526314)';
  EXECUTE IMMEDIATE 'INSERT INTO ' || v_test_cm || ' VALUES (''1'', ''1'', 0.0)';
  COMMIT;


  -- TEST MODEL
  IF (test_metric_name IS NOT NULL) THEN
    -- CREATE APPLY RESULT FOR TEST
    v_apply_result := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));

    DBMS_DATA_MINING.APPLY(
      model_name          => model_name,
      data_table_name     => v_test_data,
      case_id_column_name => v_case_id,
      result_table_name   => v_apply_result);

    EXECUTE IMMEDIATE 'CREATE TABLE ' || test_metric_name || ' (METRIC_NAME VARCHAR2(30),
METRIC_VARCHAR_VALUE VARCHAR2(31), METRIC_NUM_VALUE NUMBER)';
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''MODEL_NAME'', :model)' USING model_name;
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''TEST_DATA_NAME'', :test_data)' USING v_test_data;
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''MINING_FUNCTION'', ''CLASSIFICATION'')';
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''TARGET_ATTRIBUTE'', :target)' USING 'AFFINITY_CARD';
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''POSITIVE_TARGET_VALUE'', :target_value)' USING v_target_value;
    COMMIT;

    IF confusion_matrix_name IS NULL THEN
      v_confusion_matrix := ADD_TEMP_TABLE(v_tempTables, create_new_temp_table_name('DM$T'));
    ELSE
      v_confusion_matrix := confusion_matrix_name;
    END IF;

    DBMS_DATA_MINING.COMPUTE_CONFUSION_MATRIX (
      accuracy                     => v_accuracy,
      apply_result_table_name      => v_apply_result,
      target_table_name            => v_test_data,
      case_id_column_name          => v_case_id,
      target_column_name           => 'AFFINITY_CARD',
      confusion_matrix_table_name  => v_confusion_matrix,
      score_column_name            => 'PREDICTION',
      score_criterion_column_name  => 'PROBABILITY',
      cost_matrix_table_name       => v_test_cm,
      apply_result_schema_name     => null,
      target_schema_name           => null,
      cost_matrix_schema_name      => null

      );
    -- DBMS_OUTPUT.PUT_LINE('**** MODEL ACCURACY ****: ' || ROUND(accuracy, 4));

    IF (confusion_matrix_name IS NOT NULL) THEN
```

```
      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_NUM_VALUE) VALUES (''ACCURACY'', :accuracy)' USING v_accuracy;
      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''CONFUSION_MATRIX_TABLE'', :confusion_matrix_name)' USING
confusion_matrix_name;
      COMMIT;

      -- Average Accuracy
      EXECUTE IMMEDIATE '
        WITH
        a as
          (SELECT a.actual_target_value, sum(a.value) recall_total
             FROM ' || confusion_matrix_name || ' a
             group by a.actual_target_value)
          ,
        b as
          (SELECT count(distinct b.actual_target_value) num_recalls
             FROM ' || confusion_matrix_name || ' b)
          ,
        c as
          (SELECT c.actual_target_value, value
             FROM ' || confusion_matrix_name || ' c
             where actual_target_value = predicted_target_value)
          ,
        d as
          (SELECT sum(c.value/a.recall_total) tot_accuracy
             FROM a, c
             where a.actual_target_value = c.actual_target_value)
        SELECT d.tot_accuracy/b.num_recalls * 100 avg_accuracy
        FROM b, d' INTO v_avg_accuracy;
      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_NUM_VALUE) VALUES (''AVG_ACCURACY'', :avg_accuracy)' USING v_avg_accuracy;
      COMMIT;
    END IF;

    -- Predictive Confidence
    EXECUTE IMMEDIATE '
      WITH
      a as
        (SELECT a.actual_target_value, sum(a.value) recall_total
           FROM ' || v_confusion_matrix || ' a
           group by a.actual_target_value)
        ,
      b as
        (SELECT count(distinct b.actual_target_value) num_classes
           FROM ' || v_confusion_matrix || ' b)
        ,
      c as
        (SELECT c.actual_target_value, value
           FROM ' || v_confusion_matrix || ' c
           where actual_target_value = predicted_target_value)
        ,
      d as
        (SELECT sum(c.value/a.recall_total) tot_accuracy
           FROM a, c
           where a.actual_target_value = c.actual_target_value)
      SELECT (1 - (1 - d.tot_accuracy/b.num_classes) / GREATEST(0.0001, ((b.num_classes-
1)/b.num_classes))) * 100
      FROM b, d' INTO v_predictive_confidence;
    EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME, METRIC_NUM_VALUE)
VALUES (''PREDICTIVE_CONFIDENCE'', :predictive_confidence)' USING v_predictive_confidence;
    COMMIT;

    IF lift_result_name IS NOT NULL AND v_target_value IS NOT NULL THEN
      DBMS_DATA_MINING.COMPUTE_LIFT (
        apply_result_table_name   => v_apply_result,
        target_table_name         => v_test_data,
        case_id_column_name       => v_case_id,
        target_column_name        => 'AFFINITY_CARD',
        lift_table_name           => lift_result_name,
        positive_target_value     => v_target_value,
        num_quantiles             => v_num_quantiles,
        cost_matrix_table_name    => v_test_cm,
        apply_result_schema_name  => null,
```

```
        target_schema_name        => null,
        cost_matrix_schema_name   => null

        );
      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''LIFT_TABLE'', :lift_result_name)' USING lift_result_name;
      COMMIT;
    END IF;

    IF roc_result_name IS NOT NULL AND v_target_value IS NOT NULL THEN
      DBMS_DATA_MINING.COMPUTE_ROC (
        roc_area_under_curve        => v_area_under_curve,
        apply_result_table_name     => v_apply_result,
        target_table_name           => v_test_data,
        case_id_column_name         => v_case_id,
        target_column_name          => 'AFFINITY_CARD',
        roc_table_name              => roc_result_name,
        positive_target_value       => v_target_value,
        score_column_name           => 'PREDICTION',
        score_criterion_column_name => 'PROBABILITY');
      -- DBMS_OUTPUT.PUT_LINE('**** AREA UNDER ROC CURVE ****: ' || area_under_curve );

      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_VARCHAR_VALUE) VALUES (''ROC_TABLE'', :roc_result_name)' USING roc_result_name;
      EXECUTE IMMEDIATE 'INSERT INTO ' || test_metric_name || ' (METRIC_NAME,
METRIC_NUM_VALUE) VALUES (''AREA_UNDER_CURVE'', :v_area_under_curve)' USING
v_area_under_curve;
      COMMIT;
    END IF;
  END IF;

  DROP_TEMP_TABLES(v_tempTables);

EXCEPTION WHEN OTHERS THEN
  DROP_TEMP_TABLES(v_tempTables);
  RAISE;
END;

END;


/
```