

Business Intelligence e Mineração de Dados

Marco Rodrigues nº4652

Mestrado em Engenharia de Software

2016-2017

**Índice**

[1. Introdução 3](#_Toc488182669)

[1.1. Estrutura do Documento 3](#_Toc488182670)

[2. Data Mining 3](#_Toc488182671)

[3. Regras de Associação 4](#_Toc488182672)

[4. Trabalhos Relacionados 5](#_Toc488182673)

[5. Casos Práticos 6](#_Toc488182674)

[5.1. Dataset Groceries 6](#_Toc488182675)

[5.1.1. Instalação 6](#_Toc488182676)

[5.1.2. Criação das Regras de Associação 10](#_Toc488182677)

[5.1.3. Análise de Resultados 13](#_Toc488182678)

[5.2. Dataset Titanic 14](#_Toc488182681)

[5.2.1. Instalação 14](#_Toc488182682)

[5.2.2. Criação das Regras de Associação 15](#_Toc488182683)

[5.2.3. Análise de Resultados 16](#_Toc488182684)

[6. Conclusão 17](#_Toc488182685)

[7. Referências 18](#_Toc488182686)

# Introdução

O presente documento tem como objetivo explorar a técnica de data mining Classificação, fazendo um enquadramento teórico e seguindo depois exemplos práticos de forma a consolidar conhecimento e aplicar os conceitos teóricos.

# Estrutura do Documento

Este documento está estruturado em seis capítulos. O primeiro refere-se á introdução do documento. O segundo faz uma pequena introdução ao tema “data mining”. O terceiro capitulo faz um enquadramento ás Regras de Associação. O quarto capítulo contém os casos práticos realizados com Regras de Associação.

# Data Mining

O termo Data Mining, que traduzindo para português resulta em Mineração de Dados, consiste num conjunto de técnicas avançadas para utilizar sobre grandes quantidades de dados á procura de padrões consistentes, de forma que das mesmas seja possível extrair conhecimento [1].

Esse conjunto de técnicas inclui as seguintes:

* Árvores de Decisão (Classificação)
* Regras de Associação
* Redes Neuronais Artificiais
* Algoritmos Genéticos
* Avaliação de Técnicas e Algoritmos

# Árvores de Decisão

A Classificação é uma técnica de mineração de dados usada para classificar uma coleção de dados através da determinação de uma categoria para os mesmos. A utilização desta técnica permite, por exemplo, classificar entidades ou tipificar cenários, tarefas que se tornaram bastante comuns em soluções de suporte á decisão.

Para classificar são utilizadas Árvores de Decisão que correspondem a uma forma de representação de um conjunto de regras e seguem uma hierarquia de classes ou valores, tornando o esquema representado graficamente semelhante a uma árvore. Encaixam-se no tipo de aprendizagem supervisionada.

De forma resumida, as Árvores de Decisão classificam instâncias desde o nó raíz até aos terminais (folhas) e cada nó da árvore especifica um teste para os atributos da instância e cada ramo que descende desse nó corresponde a um dos valores possíveis para esse atributo.

Para construir uma árvore de decisão é necessário um conjunto de dados de treino e um conjunto de dados de teste. O conjunto de dados de treino é utilizado para identificar um modelo que classifica os dados considerando a variável de saída. Este conjunto deve ter casos perfeitamente definidos e corretamente classificados, dado que servem para que o algoritmo possa aprender como se deverá comportar perante situações idênticas. O conjunto de dados de teste serve para verificar o desempenho do algoritmo e avaliar a sua utilidade em tarefas de decisão. Representam a única forma de garantir que a estrutura resultante será bem sucedida em previsões de casos no futuro.

Do ponto de vista de quem vai utilizar o sistema o objetivo será encontrar uma árvore que melhor se adapte ao problema, ou seja, a que melhor classifique as instâncias do domínio em questão.

No caso de uma árvore de decisão não classificar todos os casos de forma correta, as exceções são adicionadas ao conjunto de treino de forma que o algoritmo possa “aprender” essas exceções e considerá-las no futuro.

Há dois tipos de árvores de decisão, as Árvores de Classificação e as Árvores de Regressão.

As Árvores de Classificação servem para qualificar os registos e associá-los com a classe determinada e garantir que essa mesma classificação esteja correta.

As Árvores de Regressão realizam estimativas do valor de uma determinada variável.

Há vários algoritmos de implementação de árvores de decisão que seguem uma metodologia top-down como o CART, CHAID, ID3 ou o C4.5.

# Trabalhos Relacionados

Em vários setores da indústria tem sido cada vez mais aposta a análise avançada de dados, e as Regras de Associação são técnicas utilizadas neste tipo de análises que têm contribuído bastante para aplicabilidades em casos reais.

Segundo artigo relacionado [2], em fábricas de químicos, alarmes podem disparar quando variáveis dos processos ultrapassam determinados limites, no entanto devido á interação de variáveis operacionais de processo é possível que um determinado número de alarmes seja disparado desnecessariamente e consequentemente provoque tempo perdido por parte dos operadores das fábricas. Motivado por esta necessidade foi publicado um artigo que através de mineração de dados é descoberta uma correlação entre as variáveis de processos de sequências de alarmes e desta forma consegue eliminar alarmes redundantes e otimizar o tempo de trabalho dos operadores.

Outro caso que demonstra a utilidade das Regras de Associação é o da aplicação das mesmas para análise de dados clínicos [3]. Neste caso foi feita uma análise a um dataset público e utilizada uma extensão do algoritmo FP-Growth, que permite encontrar padrões frequentes em bases de dados de grande dimensão. Com a utilização desta técnica foi possível retirar conclusões de valor do ponto de vista clínico, como por exemplo o facto de mais de 30% dos casos presentes no dataset que sofrem de hipertensão também apresentam problemas de alta pressão sistólica e de fígado.

Um terceiro caso é o da utilização das Regras de Associação para análise de dados de acidentes rodoviários [4], de forma a melhorar as políticas de segurança rodoviária por parte das autoridades. Foi utilizado um dataset de dados reais de uma das estradas mais utilizadas em Marrocos, de onde se pôde retirar regras como por exemplo se a nacionalidade do condutor era Marroquina e o acidente foi fatal, a causa do acidente foi “Adormecimento” com um grau de confiança de 88%, ou se a nacionalidade era Marroquina e a causa do acidente foi excesso de velocidade, então o acidente não foi fatal, provocando apenas feridos, com um grau de confiança também de 88%. Este tipo de regras pode ser utilizado para efetuar melhorias nas políticas de controlo dessa estrada por parte das autoridades, ou até mesmo para efetuar alterações na própria estrada.

# Casos Práticos

# Dataset BMW

Este exemplo prático será realizado com a ferramenta de data mining **WEKA** [5] e vai ser utilizado um dataset que diz respeito a um stand da BMW fictício, em que o mesmo está a começar uma campanha promocional para tentar recuperar antigos clientes com um contrato que garante dois anos extra de garantia.

Os quatro atributos que constam no dataset são: intervalo de vencimento [0=$0-$30k, 1=$31k-$40k, 2=$41k-$60k, 3=$61k-$75k, 4=$76k-$100k, 5=$101k-$150k, 6=$151k-$500k, 7=$501k+], ano/mês da primeira compra BMW, ano/mês da última compra BMW, resposta a oferta de extensão de garantia no passado.

O primeiro passo é carregar o ficheiro “bmw-training.arff” no Weka. Este ficheiro contém 3000 registos acerca de antigos clientes.

Para carregar é necessário abrir o sofware Weka, selecionar “Explorer”, “Open file” e por último escolher o ficheiro referido.

Após carregar os dados o Weka apresenta-se como na figura abaixo.

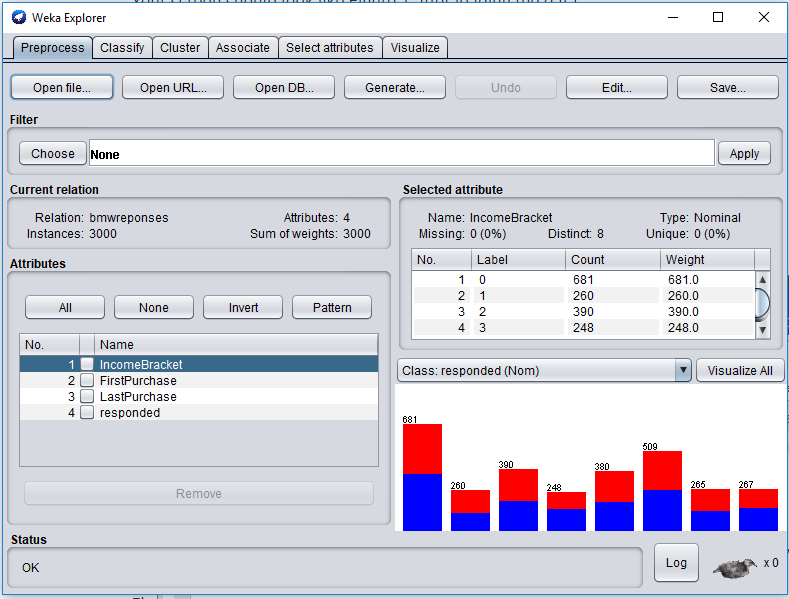


Figura - BMW-Training carregado no Weka

# Árvores de Decisão

Neste primeiro caso prático vai ser utilizado um algoritmo **Árvores de Decisão** **(Decision Trees)** para classificar. Para tal, é necessário ir á tab “Classify”, selecionar a opção “Choose” e na estrutura encontrar “trees” e “J48”. O “J48” é uma implementação em Java do algoritmo “C4.5”, e este é um algoritmo utilizado para gerar árvores de decisão.

Nas opções “Test Options” deverá estar selecionada a opção “Use training set”, ficando o Weka com o seguinte estado antes de se criar o modelo.

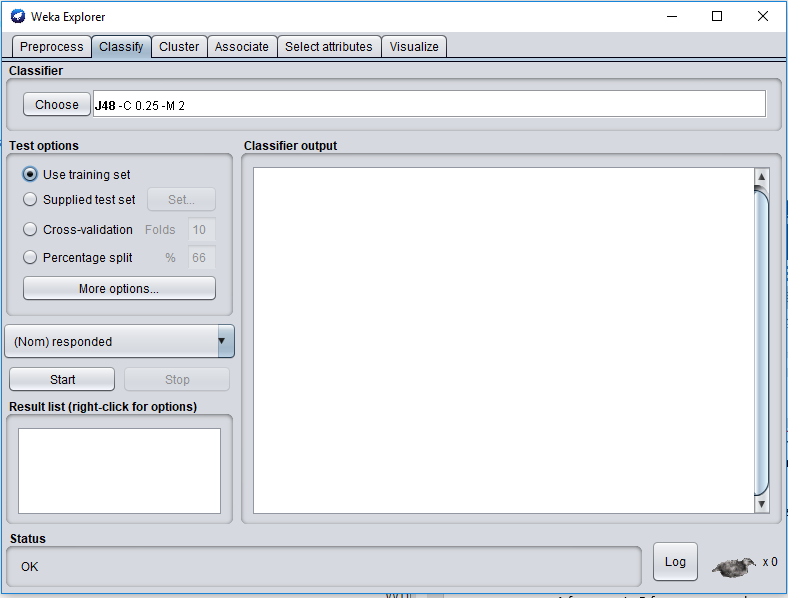


Figura - Classificar

O próximo passo é carregar “Start” para indicar ao Weka que comece a criar o modelo.

Neste caso prático o output gerado é o das imagens abaixo.

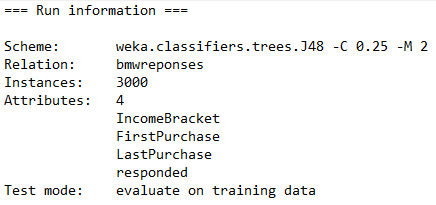


Figura 3 - Informação geral da execução

No primeiro excerto são apresentados alguns dados gerais acerca da execução, como o número de instâncias utilizadas, o número de atributos e quais foram, etc.

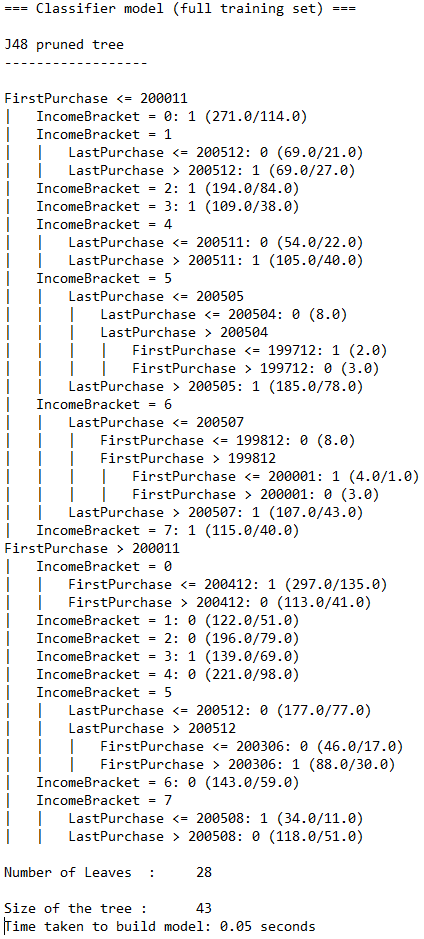


Figura - Árvore de Decisão

De seguida podemos ver o modelo classificador, isto é, a árvore de decisão gerada com o algoritmo J48 (que corresponde ao C4.5 em Java). Podemos também constatar que o modelo gerou 28 folhas (nós terminais na árvore) e no total tem 43 nós.

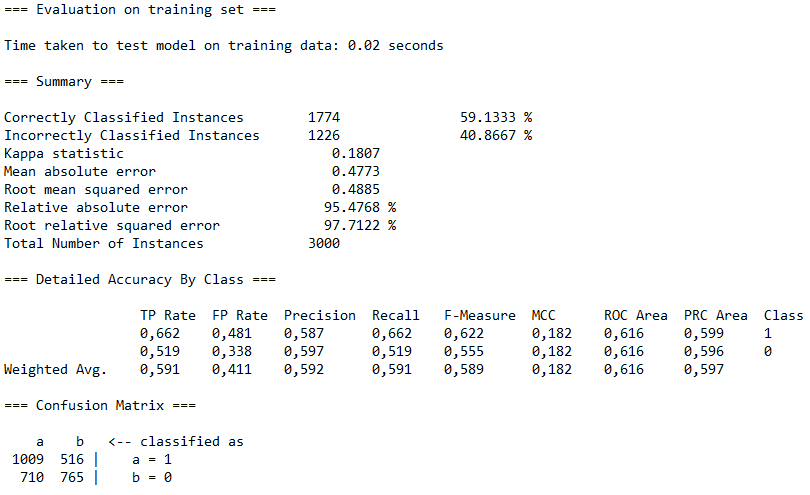


Figura - Sumário da árvore de decisão

Olhando para o sumário da árvore de decisão gerada, podemos perceber que:

* Correctly Classified Instances 1774 -> 59.13%. Este valor indica-nos que 1774 instâncias foram classificadas corretamente.
* Incorrectly Classified Instances 1226 -> 40.87%. Este valor indica-nos que 1226 instâncias foram classificadas de forma incorreta.
* Kappa Statistic 0.1807. Esta é uma estatística parecida com o coeficiente de correlação, isto é, mede a previsão de acordo com a sua classe verdadeira.
* Mean Absolute Error 0.4773. Esta estatística mede a magnitude dos erros sem considerar a sua direção, isto é, permite medir a precisão para variáveis contínuas.
* Root Mean Squared Error 0.4885. Esta estatística é uma forma diferente de calcular a anterior, o Mean Absolute Error, portanto também mede a magnitude dos erros, no entanto esta dá um peso maior a erros de maior dimensão.
* Relative Absolute Error 95.48%. Este valor é calculado com o Mean Absolute Error dividido pelo erro do classificador ZeroR.
* Root Relative Squared Error 97.71%. Este valor é calculado com o Root Mean Squared Error dividido pelo erro do classificador ZeroR.

Em baixo podemos também analisar a matriz da confusão que é uma tabela que permite identificar imediatamente o desempenho do modelo através do número de falsos positivos e falsos negativos resultantes do mesmo.

Neste caso prático tivemos 516 falsos positivos, isto é, uma instância de dados em que o modelo fez previsão de resultado positivo e o resultado real é negativo, e ainda 710 falsos negativos que são o oposto.

Para visualizar a árvore de decisão representada graficamente é necessário clicar com o botão direito no modelo gerado e selecionar “Visualiza Tree”, tal como na figura abaixo.

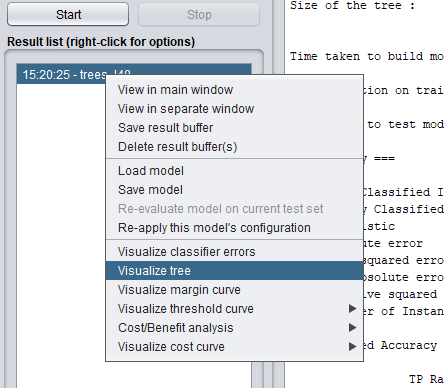


Figura - Visualizar Árvore de Decisão

E o resultado obtido é o seguinte.

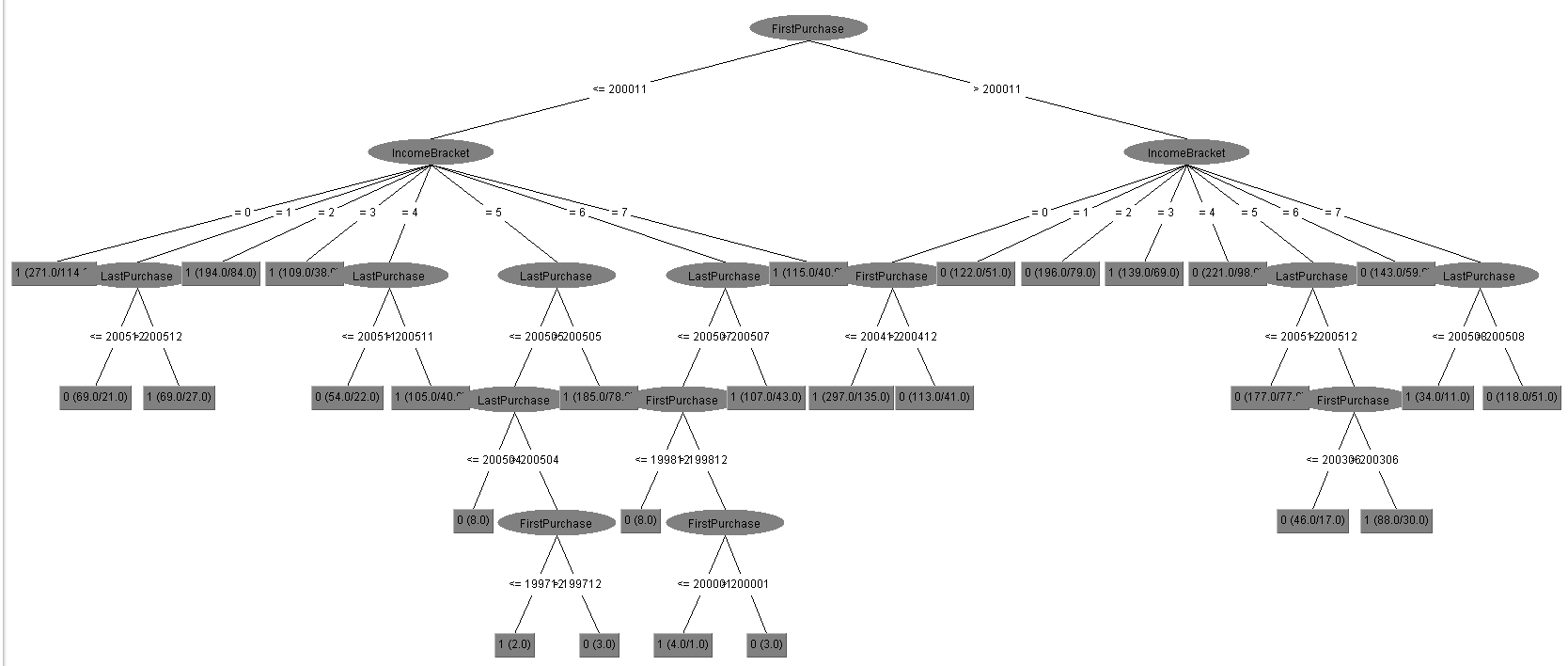


Figura - Árvore de Decisão representada graficamente

Se analisarmos mais detalhadamente a árvore gerada, podemos verificar que o primeiro nodo diz respeito ao atributo “Primeira Compra”.

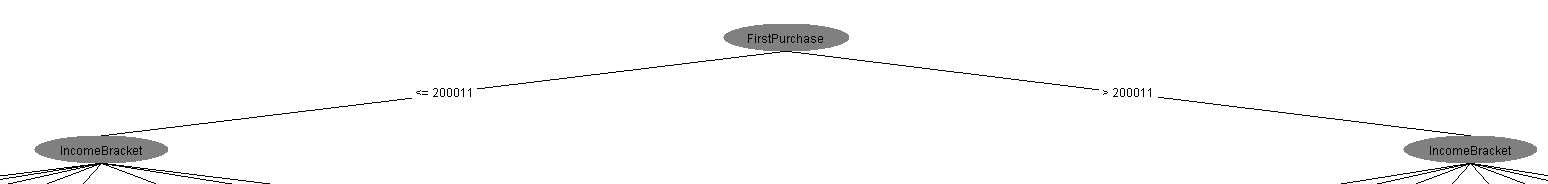


Figura - Primeiro nodo da árvore

A primeira regra que o algoritmo atribui aos dados é se a primeira compra é inferior ou igual ao ano 2000 e mês 11 seguem um caminho, senão seguem outro.

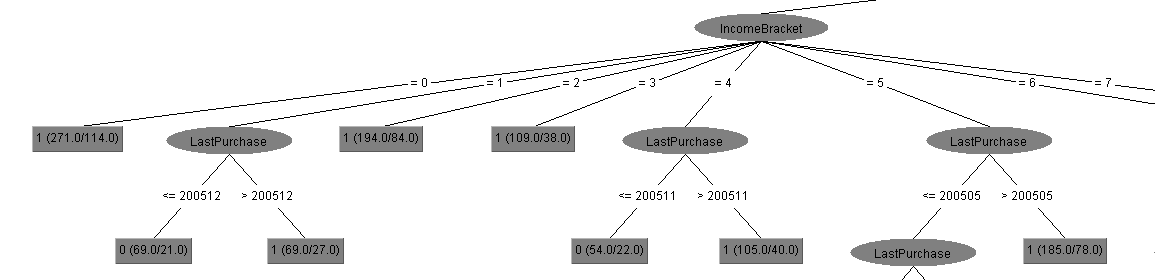


Figura - Mais nodos da árvore

A regra seguinte é relativa ao Income Bracket. Para cada valor possível (entre 0 e 7) segue um nodo diferente. Podemos, no entanto, perceber que se o Income Bracket pertencer ao intervalo 0, o modelo está a prever resultado positivo, isto é que o cliente compre o BMW e aceite o contrato de garantia estendido por dois anos extra. Se o Income Bracket estiver no intervalo 1, o modelo analisa ainda o atributo “Last Purchase” e se este for inferior ou igual ao ano 2005 e ao mês 12, a previsão é negativa, se for superior a previsão é positiva.

Este é o tipo de lógica implementada nas árvores de decisão de classificação, formam uma hierarquia através de conjuntos de regras para permitir obter previsões acerca das classificações para novos conjuntos de dados.

De seguida é relevante utilizar um conjunto de dados de teste para avaliar o modelo, ou seja, a sua precisão com os dados de teste não deverá oscilar muito comparando-a com os dados de treino.

Para carregar os dados de teste é necessário na tab “Classify” selecionar a opção “Supplied set test” que é para indicar ao Weka que a próxima execução será para teste e clicar em “Set”, “Open file” e selecionar o “BMW-test.arff”. De seguida fazer “Start”.

O resultado da execução com os dados de teste é o seguinte.

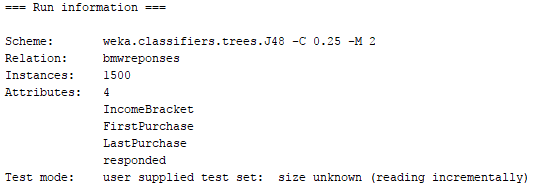


Figura - Execução de dados de teste

Nesta imagem podemos verificar que os dados de teste tinham apenas 1500 instâncias, ao contrário dos casos de treino que tinham 3000.

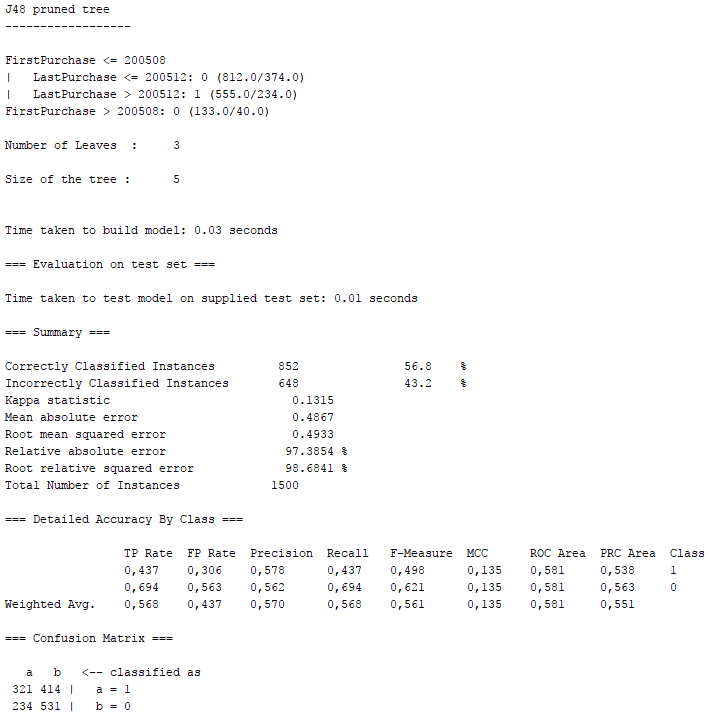


Figura 11 - Execução de dados de teste

Nesta imagem podemos constatar que a percentagem de instâncias classificadas corretamente é 55.7% que anda perto dos 59.1% gerados pelos casos de treino. Esta proximidade indica-nos que o modelo não deverá quebrar quando forem aplicados novos casos no futuro, isto é, o modelo vai ser capaz de classificar os mesmos.

No entanto, um modelo com uma percentagem de instâncias corretas na casa dos 60% não é um modelo que dê grandes garantias, não tem muita precisão, logo não é um bom modelo.

# Vizinho Mais Próximo

De seguida vão ser utilizados os mesmos datasets “BMW-training.arff” e “BMW-test.arff” mas usando o algoritmo de classificação **Vizinho Mais Próximo (Nearest Neighbor)** em vez das Árvores de Decisão.

O primeiro passo é carregar o ficheiro “bmw-training.arff” no Weka, através de “Explorer”, “Open file” e por último escolher o ficheiro referido.

De seguida, na tab “Classify” ir a “Choose” e escolher “lazy”->”IBk”.

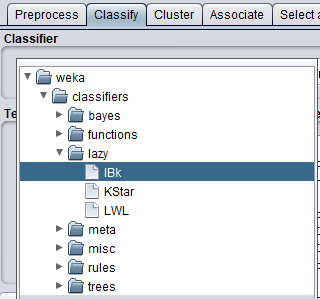


Figura - Selecionar o algoritmo IBk

O IBk é uma implementação do algoritmo Vizinho Mais Próximo e IB significam “Instance-based” e “k” representa um número de vizinhos para considerar.

O passo seguinte é selecionar “Use training set” nas “Test options” e clicar “Start”. O modelo gerado foi o seguinte.

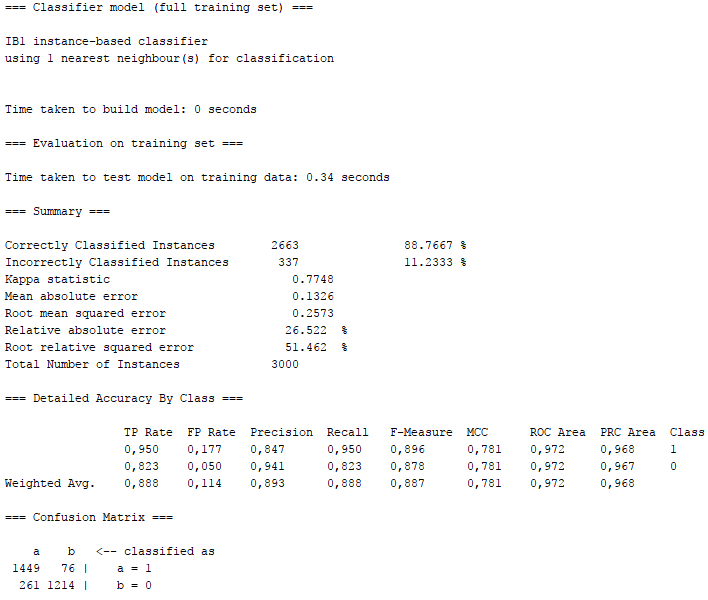


Figura - Modelo gerado com algoritmo Vizinho Mais Próximo

Se analisarmos os resultados obtidos com o algoritmo de classificação Vizinho Mais Próximo podemos facilmente perceber que foi criado um modelo com 88.7% de instâncias classificadas corretamente, o que seria um valor bastante aceitável. Também é possível perceber que o modelo gerou 76 falsos positivos e 261 falsos negativos.

Uma propriedade importante do algoritmo Vizinho Mais Próximo é o número de vizinhos a considerar para gerar o modelo. Este valor é possível ser alterado para gerar novos modelos, para tal é necessário clicar com o botão direito do rato por cima de IBk e a propriedade que corresponde ao número de vizinhso é o “KNN”.

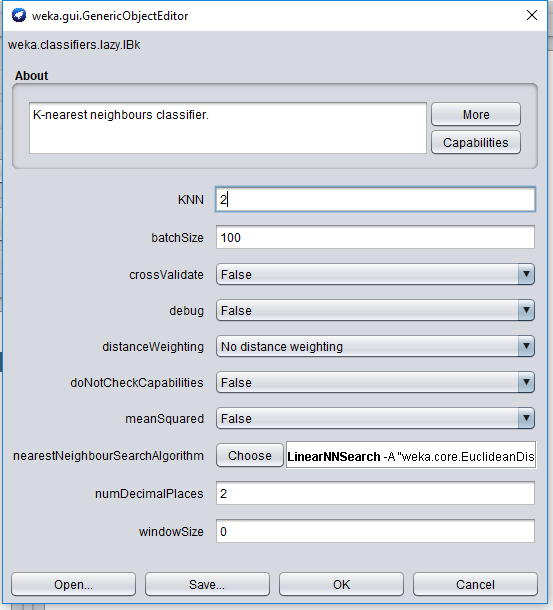


Figura - Propriedades do algoritmo Vizinho Mais Próximo

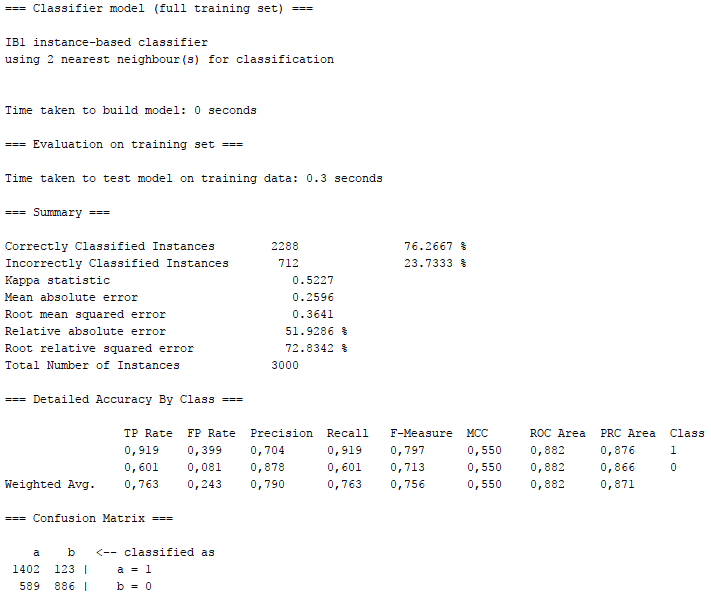


Figura 15 - Considerados 2 Vizinhos Mais Próximos

Após gerar novo modelo com 2 Vizinhos Mais Próximos, a precisão do modelo diminui para 76.27%. O número de falsos positivos aumenta para 589 e o número de falsos negativos aumenta para 123.

# Dataset Pima Indians Diabetes

O Pima Indians Diabetes é um dataset que representa



# Conclusão

Neste trabalho foram realizados dois casos práticos utilizando as Regras de Associação como técnica de mineração de dados a dois datasets diferentes.

No primeiro caso foi utilizado o dataset “Grocerie” que representava um conjunto de dados de compras numa mercearia. Após a aplicação de técnicas para a descoberta de Regras de Associação, os resultados obtidos foram os seguintes: quem compra “herbs” aumenta 3.95 vezes a probabilidade de comprar “root vegetables”, quem compra “berries” (frutos vermelhos) aumenta 3.79 vezes a probabilidade de comprar “whipped/sour cream” (natas), quem compra “other vegetables”, “tropical fruit” e “whole milk” aumenta 3.76 vezes a probabilidade de comprar “root vegetables”, quem compra “beef” e “other vegetables” aumenta 3.68 vezes a probabilidade de comprar “root vegetables” e quem compra “tropical fruit” e “other vegetables” aumenta 3.48 vezes a probabilidade de comprar “pip fruit”.

No segundo caso foi utilizado o dataset relativo á tragédia do Titanica e após a aplicação de técnicas para a descoberta de Regras de Associação, foi possível concluir que: quem viajou em segunda classe e era criança (1%), sobreviveu com 100% de grau de confiança. E se era segunda classe e criança, a probabilidade de ter sobrevivido aumentava 3.09 vezes, quem viajou em primeira classe e era do sexo feminino (6.4%), sobreviveu com grau de confiança de 97%, sendo que viajar em primeira classe e ser do sexo feminino aumentava em 3.01 vezes a probabilidade de sobreviver, 19% (suporte) dos casos totais são do sexo feminino e da faixa etária adulta com um grau de confiança de 90% (confidence) um lift que nos diz que 95% dos casos de sexo feminino são adultos, 40% (suporte) dos casos totais pertenciam á classe “Crew” (tripulação) e são adultos, o grau de confiança é de 100%, e o lift diz-nos que se for da classe “Crew” é altamente provável (105%) de ser da faixa etária adulta.

# Referências

[1] – Bibliografia sobre Data Mining. Disponibilizada pelo Prof. Jorge Ribeiro na página do Moodle da disciplina de Business Intelligence e Mineração de Dados do Mestrado em Engenharia de Software.

[2] – Artigo sobre a utilização de Regras de Associação em fábrica de químicos. J. Wang, Li, Huang, & Su, “Association rules mining based analysis of consequential alarm

sequences in chemical processes”, Mar.2016

[3] – Artigo sobre a utilização de Regras de Associação para análise de dados clínicos. B. Wang et al., “Comprehensive Association Rules Mining of Health

Examination Data with an Extended FP-Growth Method”, Jan.2017

[4] – Artigo sobre a utilização de Regras de Associação para análise de dados sobre acidentes rodoviários numa estrada em Marrocos. Addi, Tarik, & Fatima, “An Approach Based On Association Rules Mining To Improve Road Safety In Morocco”, Apr.2016

[5] – Sítio da ferramenta de Data Mining R. Disponível em: <https://www.r-project.org/>

[6] – Sítio da ferramenta de Data Mining Rstudio. Disponível em: <https://www.rstudio.com/>

[7 ] – Sítio dedicado a Data Mining e que disponibiliza o dataset Titanic. Disponível em: <http://www.rdatamining.com/data>