UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCIO NOVASKI

O TEOREMA DE PROBABILIDADE PELO ALGORITMO NAIVE BAYES PARA A TAREFA DE CLASSIFICAÇÃO NA SHELL ORION DATA MINING ENGINE

CRICIÚMA
2012

MARCIO NOVASKI

O TEOREMA DE PROBABILIDADE PELO ALGORITMO NAIVE BAYES PARA A TAREFA DE CLASSIFICAÇÃO NA SHELL ORION DATA MINING ENGINE

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel, no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientadora: Prof ª. MSc. Merisandra Côrtes de Mattos Garcia

CRICIÚMA

2012

MARCIO NOVASKI

O TEOREMA DE PROBABILIDADE PELO ALGORITMO NAIVE BAYES PARA A TAREFA DE CLASSIFICAÇÃO NA SHELL ORION DATA MINING ENGINE

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Inteligência Computacional.

Criciúma, 28 de novembro de 2012.

BANCA EXAMINADORA

Profª. Merisandra Côrtes de Mattos Garcia - Mestre - (UNESC) - Orientador

Prof. Leandro Antunes Berti - Doutor - (UNESC)

Prof. Kristian Madeira - Mestre - (UNESC)

A Deus e as minhas maiores riquezas, Graziela e Rafaella, aos meus pais, irmãos e meus queridos amigos por todo apoio e consideração.

RESUMO

As constantes inovações tecnológicas na atualidade proporcionam um aumento gradativo da quantidade de informação que é armazenada gerando grandes bases de dados, tornando necessário o uso de tecnologias que auxiliem na análise e entendimento dessas informações. O *data mining* destaca-se dentre essas tecnologias, possibilitando a obtenção do conhecimento por meio de algoritmos com finalidades específicas para cada problema proposto. Para isso é necessário o uso de ferramentas computacionais, as *Shells*, que na sua maioria são proprietárias. Por esse motivo, o Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da UNESC, mantém em desenvolvimento o projeto de uma ferramenta que implementa diversos métodos e tarefas do *data mining* denominada *Shell Orion Data Mining Engine*. O objetivo dessa pesquisa consiste em ampliar as funcionalidades da *Shell Orion*, implementando e demonstrando o funcionamento do algoritmo *Naive Bayes* para a tarefa de classificação. O algoritmo utiliza os conceitos estatístico e probabilístico da teoria de Thomas Bayes para determinar a classe a qual um determinado registro pertence. Para isso, baseia-se nas informações das probabilidades *a priori* e *a posteriori*, onde o resultado prevalece de acordo com a classe que apresentar a probabilidade máxima. Ao final da pesquisa foram realizados testes em uma base de dados e o desempenho do algoritmo foi avaliado usando algumas medidas de validação como sensibilidade, especificidade, acurácia, confiabilidade positiva e índice kappa. Os resultados apresentados mostraram que, para a base de dados escolhida durante os testes, o algoritmo apresentou uma taxa de acerto de 99,14% de acerto, o que comprovou o correto funcionamento do *Naive Bayes* na *Shell Orion Data Mining Engine*.

Palavras-chave: Inteligência Computacional. *Data Mining*. Classificação. Algoritmo *Naive Bayes*. Probabilidade.

ABSTRACT

Nowadays, the constant technological innovations provide a gradual increase in the amount of information that is stored, generating large databases and making necessary the use of technologies to help in the analysis and understanding of that information. Data mining stands out among these technologies, making possible the acquisition of knowledge through special purpose algorithms for each proposed problem. This fact requires the use of computational tools, the Shells, which are usually copyrighted. For this reason, the Research Group at Applicated Computation Intelligence Course of Computation Science at UNESC, keeps in development a project of a tool called Shell Orion Data Mining Engine which establishes many methods of data mining tasks. The purpose of this research is to increase the operations of Shell Orion, establishing and demonstrating the Naive Bayes algorithm's operation for the classification task. The algorithm makes use of the statistical and probabilistic concepts of the Thomas Bayes theory to determine the class to which a given record belongs. Is based on the information of probabilities a priori and a posteriori, where the outcome prevails according to the class that presents maximum likelihood. At the end of the research, several tests are done in a database and the performance of the algorithm has been evaluated using some validation measures such as sensitivity, specificity, accuracy, reliability positive and kappa statistic. The obtained results showed that, for the chosen database, the algorithm had an accuracy rate of 99.14%, which confirmed the correct activity of Naive Bayes on Shell Orion Data Mining Engine.

Keywords: Computation Intelligence, Data Mining. Classification, Naive Bayes Algorithm, Probability.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
CART	*Classification and Regression Trees*
HSQLDB	*HyperSQL Database*
ID3	*Induction of Decision Tree*
KDD	*Knowledge Discovery in Databases*
MAP	Máximo *A Posteriori*
RBF	*Radial Basis Function*
RCP	*Robust C-Prototypes*
RNA	Redes Neurais Artificiais
SGBD	Sistema Gerenciador de Banco de Dados
SPSS	*Statistical Package for Social Sciences*
TCC	Trabalho de Conclusão de Curso
UFMG	Universidade Federal de Minas Gerais
UFSC	Universidade Federal de Santa Catarina
UML	*Unified Modeling Language*
UNESC	Universidade do Extremo Sul Catarinense
UNIFEI	Universidade do Federal de Itajubá
URCP	*Unsupervised Robust C-Prototypes*
WEKA	*Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO

Atualmente a velocidade e o volume de informações geradas pelas organizações dos diversos segmentos aumentam gradativamente. Os avanços tecnológicos têm facilitado o processo de armazenamento destes dados, cujo conteúdo é essencial para o planejamento das ações e tomadas de decisões futuras. Consequentemente, a análise desta grande quantidade de informação tornou-se complexa para a capacidade humana.

Ferramentas estatísticas, modelos matemáticos, consultas estruturadas são comumente utilizadas no intuito de facilitar a obtenção de informações a partir destas grandes bases de dados. Porém, tais ferramentas possuem limitações que podem comprometer a precisão da informação gerada. Dentro deste contexto surge o conceito de *data mining* que utiliza técnicas de inteligência computacional como o reconhecimento de padrões, o aprendizado de máquina, métodos estatísticos e banco de dados para auxiliar na descoberta de conhecimento. O processo de *data mining* utiliza tarefas e métodos específicos de acordo com o tipo de problema e o objetivo da descoberta do conhecimento (GOLDSCHMIDT; PASSOS, 2005).

As diversas tarefas de *data mining* como classificação, clusterização, associação, sumarização e regressão, associadas a métodos, são implementadas em ferramentas conhecidas como *shells* que geralmente são comerciais e de custo elevado.

Uma das ferramentas existentes na área de *data mining* é a *Shell Orion Data Mining Engine*, implementada pelo Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação, na Universidade do Extremo Sul Catarinense. Consiste em um projeto de pesquisa estruturado em módulos distintos para cada tarefa de *data mining* onde já foram implementados algoritmos para associação, clusterização e classificação.

Dentre as tarefas utilizadas para a obtenção de conhecimento a partir de uma base de dados, a classificação é uma das mais importantes e populares. O método de classificação consiste em identificar características comuns entre os registros de uma base de dados e associá-las a um atributo específico designado como classe (GOLDSCHMIDT; PASSOS, 2005; HAN; KAMBER, 2006, tradução nossa).

Geralmente os métodos de classificação apresentam problemas de desempenho na identificação de padrões em uma grande base de dados, resultando em uma baixa performance

na sua capacidade de predição (OLSON; DELEN, 2008, tradução nossa).

Em alguns métodos tradicionais de classificação, por exemplo, a fase de treinamento do algoritmo envolve um longo tempo de aprendizado, tornando comum a necessidade de se repetir o treinamento do algoritmo. Este aspecto não é muito vantajoso no processo de *data mining* (HAN; KAMBER, 2006, tradução nossa).

Considerando-se isso, tem-se outros métodos de classificação, como por exemplo, os baseados no Teorema de Probabilidade de Thomas Bayes, tendo-se portanto os classificadores bayesianos. Segundo Coppin (2010) e Tan, Steinbach e Kum Coppin (2010) ar (2009) estes classificadores geralmente são usados para lidar com situações de incerteza, combinando o conhecimento prévio acerca de uma hipótese com novas evidências colhidas dos dados que se pretende analisar.

Dentre os classificadores bayesianos tem-se o *Naive Bayes* que consiste em predizer a classe mais provável a que uma determinada amostra desconhecida pertença. O modelo assume que o efeito do valor de um atributo em uma dada classe é independente dos valores dos outros atributos, o que simplifica os cálculos envolvidos visando um custo computacional menor durante a fase de treinamento do algoritmo (COPPIN, 2010; HAN; KAMBER, 2006, tradução nossa).

Com base neste contexto, este trabalho de pesquisa propõe desenvolver o algoritmo classificador *Naive Bayes* na tarefa de classificação da *Shell Orion Data Mining Engine*.

1.1 OBJETIVO GERAL

Disponibilizar na *Shell Orion Data Mining Engine* o algoritmo *Naive Bayes* para a tarefa de classificação.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos da pesquisa consistem em:

a) compreender o conceito de *data mining* e a tarefa de classificação;

b) entender o teorema de Bayes e o funcionamento do algoritmo *Naive Bayes*;

c) demonstrar o funcionamento matemático do algoritmo *Naive Bayes*;

d) desenvolver e aplicar o algoritmo *Naive Bayes* para classificação dos dados;

e) utilizar uma base de dados a fim de testar o funcionamento do algoritmo de classificação *Naive Bayes*;

f) implementar formas, a serem definidas, de análise de desempenho do algoritmo apresentado.

1.3 JUSTIFICATIVA

O processo de *data mining*, por meio de métodos específicos de generalização, tem como objetivo a identificação do conhecimento em grandes bases de dados, podendo facilitar a tomada de decisões pela predição da ocorrência de padrões e relações entre estes dados (GOLDSCHMIDT; PASSOS, 2005).

A dificuldade em perceber e interpretar os fatos identificados durante o processo torna a descoberta de conhecimento bastante complexa, fazendo-se indispensável o uso de ferramentas que auxiliem no trabalho. Dentre as ferramentas utilizadas destacam-se as *Shells* que, na sua maioria, são de uso comercial e restrito, o que por vezes dificulta a sua aquisição. Além disso, várias delas apresentam limitações relacionadas à conexão com diferentes Sistemas Gerenciadores de Bancos de Dados.

Por este motivo, o projeto acadêmico da *Shell Orion Data Mining Engine* tem como proposta o desenvolvimento de uma ferramenta gratuita que implemente diversas tarefas e métodos de *data mining* e permita a conexão com diferentes Sistemas Gerenciadores de Banco de Dados (SGBD), como por exemplo, PostGreSQL, Firebird, HSQLDB, MySQL e outros.

A *Shell Orion* já possui implementados para a tarefa de classificação, os algoritmos ID3, C4.5 e o CART que utilizam árvores de decisão e o algoritmo RBF que emprega redes neurais.

Este trabalho de pesquisa se justifica pela continuidade da *Shell Orion Data Mining Engine* proporcionando a ampliação das suas funcionalidades por meio da

implementação do algoritmo *Naive Bayes* no módulo de classificação.

A classificação é uma das mais realizadas tarefas cognitivas humanas no auxílio à compreensão do ambiente em que se vive. Esse contexto faz também da classificação uma das mais utilizadas tarefas do *data mining* (GOLDSCHMIDT; PASSOS, 2005).

Dentre os métodos inteligentes utilizados para a tarefa de classificação de dados estão as Redes Neurais Artificiais, os Algoritmos Genéticos e os Classificadores Bayesianos Estatísticos.

Os classificadores bayesianos normalmente são usados em situações de incerteza por aleatoriedade. Combinam o prévio conhecimento de uma hipótese com novas evidências dos dados que se pretende analisar (COPPIN, 2010; TAN;STEINBACH; KUMAR, 2009).

Domingos e Pazzani (1997) e Han e Kamber (2006) afirmam que o classificador bayesiano conhecido como *Naive Bayes* utiliza o Teorema de Bayes como método estatístico e tem como vantagem a facilidade de implementação por ser considerado um classificador simples, ter um menor tempo de aprendizado preditivo e apresentar excelente precisão em grandes bases de dados.

1.4 ESTRUTURA DO TRABALHO

Esta pesquisa é composta por seis capítulos, sendo que no capítulo 1 são descritos o tema proposto, os objetivos pretendidos e a justificativa para a realização dessa pesquisa.

No capítulo 2 são contextualizados os principais conceitos relacionados ao processo de descoberta do conhecimento (KDD), bem como os relacionados à etapa de *data mining*. Nessa parte do trabalho também são abordados os conceitos referentes à *Shell Orion*. A tarefa de classificação em *data mining* é o tema do capítulo 3, que descreve os seus métodos. Neste capítulo também são apresentados conceitos de estatística probabilística, probabilidade clássica e condicional.

O capítulo 4 contextualiza os classificadores bayesianos abordando o paradigma e inferência bayesiana bem como o teorema de Bayes e alguns trabalhos correlatos que usaram o algoritmo *Naive Bayes*. O algoritmo implementado e o seu funcionamento também são definidos neste capítulo.

As etapas do trabalho desenvolvido são descritas no capítulo 5, incluindo a

metodologia utilizada e os resultados obtidos pelo módulo do algoritmo *Naive Bayes*.

Por fim, no capítulo 6, tem-se a conclusão da pesquisa e sugestões de trabalhos futuros.

2 A DESCOBERTA DO CONHECIMENTO EM BASE DE DADOS (KDD)

Nos últimos tempos, o avanço tecnológico tem possibilitado um aumento rápido e exponencial da capacidade de armazenamento de informações nas organizações. Com o crescimento desse volume de dados, surgiu a necessidade de novas ferramentas e técnicas que permitissem uma análise das bases de dados em busca da descoberta de informações de forma inteligente, automática e eficaz (GOLDSCHMIDT; PASSOS, 2005). Ao encontro dessa necessidade surge a descoberta de conhecimento em base de dados, conhecida como *Knowledge Discovery in Databases* (KDD).

O campo de KDD, segundo Goldschmidt e Passos (2005), tem origem em diversas áreas, como: Estatística, Banco de Dados, Conhecimento de Padrões, Aprendizado de Máquina e Inteligência Computacional. Ainda de acordo com Goldschmidt e Passos (2005), as atividades relacionadas ao KDD são diversas e podem ser organizadas em três grupos:

a) **desenvolvimento tecnológico:** envolve o desenvolvimento, aperfeiçoamento e otimização de ferramentas e tecnologias que possam ser utilizadas no auxílio à busca de novos conhecimentos em bases de dados;
b) **execução do KDD:** consiste na efetiva busca do conhecimento em base de dados;
c) **aplicação dos resultados:** refere-se à aplicação de resultados obtidos com a utilização do processo de KDD dentro do contexto da aplicação. Esse processo busca identificar padrões, tendências, probabilidades ou associações, que possibilitam a descoberta do conhecimento implícito em uma base de dados.

O processo de KDD é interativo e iterativo. Interativo porque o usuário precisa interagir em decisões que possam surgir nas diversas etapas do processo. Iterativo porque, em alguns casos, pode haver laços de repetição entre as suas etapas sem que haja uma ordem ou sequência específica (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

2.1 ETAPAS DO PROCESSO KDD

As etapas do KDD (figura 1) caracterizam o início e o fim do processo. Consistem em três etapas básicas (HAN; KAMBER, 2006, tradução nossa):

a) **pré-processamento:** é a primeira etapa do processo onde os dados são preparados no intuito de otimizar os resultados da aplicação dos algoritmos de *data mining*. Este tratamento dos dados é efetuado por meio das seguintes

funções:

– **seleção dos dados:** identificar e selecionar os dados importantes que serão utilizados durante o processo,

– **limpeza dos dados:** a garantia de um resultado preciso no processo de KDD depende da integridade dos dados selecionados. Nesta etapa os dados são selecionados, filtrados e corrigidos,

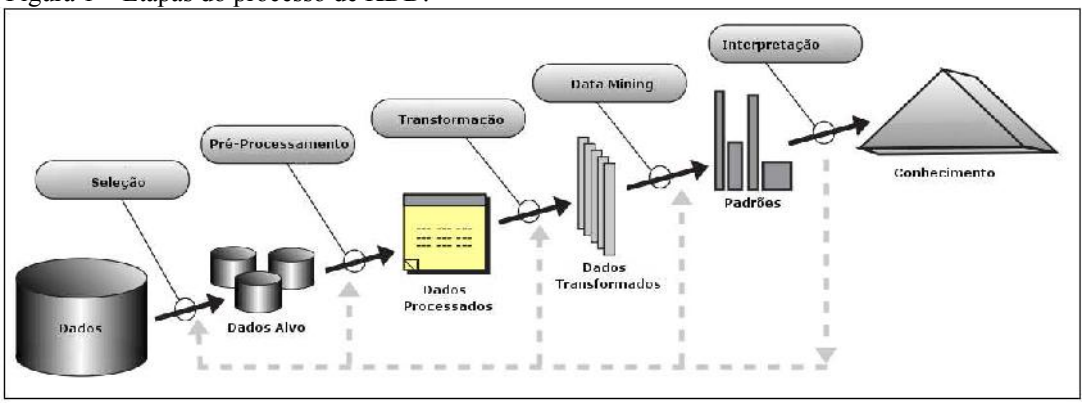
– **transformação dos dados:** é a padronização dos dados selecionados podendo envolver a conversão de dados heterogêneos e o preenchimento de valores nulos;

b) ***data mining:*** consiste na efetiva extração do conhecimento por meio da aplicação de algoritmos sobre os dados na base de conhecimento. É a principal etapa do processo de KDD (GOLDSCHMIDT; PASSOS, 2005);

c) **pós-processamento:** é a etapa de análise e avaliação do conhecimento obtido com a utilização do *data mining*. Segundo Goldschmidt e Passos (2005), esta avaliação e interpretação dos resultados deve ser realizada por dois especialistas: um no domínio da aplicação e outro em *data mining*.

A sequência das etapas realizadas no KDD é representada na figura 1.

Figura 1 – Etapas do processo de KDD.



Fonte: Adaptado de Fayyad, Piatetsky-Shapiro e Smyth (1996)

2.2 *DATA MINING*

Data mining consiste no uso de técnicas tradicionais de análise de dados combinado a tecnologias e algoritmos sofisticados para processar grandes volumes de dados. Permite ampliar as possibilidades de análise e exploração de novos tipos de dados, bem como analisar tipos antigos de novas maneiras (TAN;STEINBACH; KUMAR, 2009).

É a principal etapa do processo de KDD caracterizando a busca efetiva por conhecimento a partir de grandes bases de dados. Neste contexto, alguns autores fazem referência aos termos *data mining* e KDD como tendo mesmo conceito (GOLDSCHMIDT; PASSOS, 2005).

O *data mining* é amplamente aplicado em diversos campos de pesquisa, como por exemplo: segmentação de imagens de satélite, previsão de carga de sistemas elétricos, mineração de dados na web, marketing e vendas, entre outros (WITTEN; FRANK; HALL, 2011, tradução nossa).

Levando-se em consideração o vasto domínio de aplicação do *data mining,* existem diversas tarefas que podem ser selecionadas dependendo do objetivo a ser alcançado ou do problema que se deseja resolver.

A identificação do problema e dos objetivos do usuário são os fatores que definem a escolha de uma tarefa de *data mining* (GOLDSCHMIDT; PASSOS, 2005). As tarefas mais comuns, seus objetivos e exemplos de aplicação são relacionados a seguir:

a) **associação:** compreende em identificar padrões e relações entre dados de um determinado conjunto que apresentam ocorrências simultâneas, freqüentes e válidas que determinam uma tendência (GOLDSCHMIDT; PASSOS, 2005). Esta tarefa pode ser aplicada para identificar quais produtos em um determinado estabelecimento são vendidos em conjunto. A identificação desta tendência permite planejar a disposição dos itens nos pontos de venda do estabelecimento de forma a induzir a compra de outros produtos;
b) **classificação:** é uma das tarefas mais utilizadas no *data mining*. Consiste em encontrar regras para dados em uma base que permita associá-los a um modelo de classe. Este modelo pode ser usado posteriormente para classificar novos registros. Identificar grupos de clientes por preferência de compras, predefinir o grau de risco de inadimplência para potenciais clientes de um empréstimo, diagnosticar doenças e prescrever o seu tratamento são exemplos de aplicação da tarefa de classificação (GOLDSCHMIDT; PASSOS, 2005);
c) **regressão:** tem similaridade com a tarefa de classificação, distinguindo-se apenas por estar restrita a valores e atributos numéricos (KANTARDZIC, 2003, tradução nossa). Pode ser utilizada para estabelecer limites de créditos a clientes de um banco; predizer o lucro ou a perda em um eventual empréstimo (REZENDE, 2005);

d) **clusterização:** consiste em formar grupos de objetos ou elementos mais homogêneos utilizando métodos estatísticos específicos. Geralmente é aplicada quando não existem classes predefinidas como na tarefa de classificação (HAN; KAMBER, 2006, tradução nossa). A clusterização forma grupos de objetos com alta similaridade entre si e baixa similaridades com objetos dos demais grupos. Agrupar clientes por preferência de compra é um exemplo para a aplicação da tarefa de clusterização;

e) **previsão:** esta tarefa resume-se na previsão dos valores de um determinado atributo com base nos valores de outros atributos (TAN;STEINBACH; KUMAR, 2009). De acordo com Goldschmidt e Passos (2005) no contexto do *data mining* a tarefa de previsão tem sido usada em diversos problemas do mundo real, com o objetivo de reduzir os riscos causados por incerteza e auxiliar as tomadas de decisão, como por exemplo, prever valores que um cliente utilizará em caso de um empréstimo oferecido.

Cada tarefa do *data mining* possui uma particularidade, por este motivo a escolha da tarefa mais adequada requer conhecimento acerca do domínio de aplicação da base de dados a ser analisada (KANTARDZIC, 2003, tradução nossa).

Neste contexto, diferentes métodos podem ser aplicados levando-se em consideração as características do problema e os objetivos do usuário. Alguns destes métodos utilizados são:

a) **árvores de decisão:** são estruturas hierárquicas que remetem à ideia de uma árvore invertida desenvolvendo-se da raiz para as folhas. É uma abordagem do conceito “dividir para conquistar” utilizada para segmentar grandes coleções em pequenos subconjuntos (WITTEN; FRANK; HALL, 2011, tradução nossa);

b) **algoritmos genéticos:** são modelos computacionais que buscam obter a solução ótima ou aproximadamente ótima para problemas complexos por meio da evolução de populações ou soluções codificadas em cromossomos artificiais. É um processo adaptativo onde as soluções existentes são consideradas a cada momento e influenciam na busca por novas soluções conceito (GOLDSCHMIDT; PASSOS, 2005);

c) **lógica *fuzzy*:** parte do princípio que um elemento pode pertencer a um ou mais conjuntos, apresentando porém graus de pertinência diferentes. Tal raciocínio aproximado assemelha-se à capacidade humana de tratar as imprecisões (REZENDE, 2005);

d) **redes neurais artificiais:** são modelos matemáticos construídos com base no funcionamento do cérebro humano semelhantes às estruturas neurais biológicas, porém com menor número de conexões. A sua capacidade computacional é adquirida por meio de aprendizado e generalização (COPPIN, 2010; REZENDE, 2005). é adquirida

e) **inferência bayesiana:** modelo baseado na teoria da probabilidade de Bayes. Consiste na dedução da probabilidade *a posteriori* de que um dado pertença a uma classe específica a partir da probabilidade *a priori* desta mesma classe, ou seja, calcular a probabilidade de que um novo dado pertença a uma classe previamente determinada (NEAPOLITAN, 2004).

Os diversos métodos e tarefas de *data mining* podem ser implementados em ferramentas específicas conhecidas como *shells*. A tabela 1 descreve algumas das ferramentas de *data mining* disponíveis e as tarefas que disponibilizam.

Tabela 1 – Ferramentas e tarefas do *data mining.*

Ferramentas	Principais Tarefas	Alguns algoritmos implementados	Tipo
PolyAnalist *Megaputer Intelligence www.megaputer.com*	Classificação, Regressão, Associação, Clusterização, Sumarização, Detecção de Desvios	Naive Bayes	Comercial
WEKA University of Waikato www.cs.waikato.ac.nz	Classificação, Clusterização, Associação	IDd3, J48, DBScan, Apriori	Comercial
Intelligent Miner IBM www.ibm.com	Classificação, Sequência Clusterização, Regressão, Associação, Sumarização,	RBF, CART	Comercial
Oracle Data Mining Oracle www.oracle.com	Classificação, Regressão, Associação, Clusterização	Naive Bayes, K-Means, Apriori	Comercial
SAS Enterprise Miner SAS Inc. www.sas.com	Classificação, Regras de Associação, Regressão, Clustering, Sumarização	CART, C4.5	Comercial
Orion Data Mining Engine	Classificação, Associação, Clusterização	Apriori, CART, ID3, C4.5, RBF, K-Means, DBScan, SACA	Gratuita

Fonte: Adaptado de Goldschmidt e Passos (2005).

O custo de aquisição destas ferramentas pode se tornar inviável para as organizações devido à obrigatoriedade da licença de uso comercial, o que resulta em uma carência de *shells* gratuitas. Entre as ferramentas gratuitas disponíveis, encontra-se em

desenvolvimento pelo Grupo de Pesquisa em Inteligência Computacional Aplicada, do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense (UNESC), a *Shell* Orion *Data Mining Engine.*

2.2.1 *Shell Orion Data Mining Engine*

A *Shell Orion Data Mining Engine* é uma ferramenta em desenvolvimento que faz parte de um projeto iniciado no ano de 2005 pelos acadêmicos e professores do Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense.

Atualmente a *Shell Orion Data Mining Engine* possui implementadas as tarefas de associação, classificação e clusterização, desenvolvidas por acadêmicos em seus Trabalhos de Conclusão de Curso (TCC).

A tabela 2 apresenta a relação das tarefas e algoritmos já implementados na *Shell Orion* e ilustra a evolução do desenvolvimento da ferramenta.

Tabela 2 – Evolução da *Shell* Orion *Data Mining Engine.*

Ano	Tarefa	Método	Algoritmo	Atributos	Referência
2005	Associação	Regra de Associação	Apriori	Numérico	(CASAGRANDE, 2005)
2005	Classificação	Árvore de Decisão	ID3	Nominais	(PELEGRIN, 2005)
2007	Classificação	Árvore de Decisão	CART	Nominais e Numéricos	(RAIMUNDO, 2007)
2007	Clusterização	Particionamento	K-Means	Numéricos	(MARTINS, 2007)
2007	Clusterização	Redes Neurais	Kohonen	Numéricos	(BORTOLOTTO, 2007)
2008	Clusterização	Lógica Fuzzy	Gustafson-Kessel	Numéricos	(CASSETARI JUNIOR, 2008)
2009	Clusterização	Lógica Fuzzy	Gath-Geva	Numéricos	(PEREGO, 2009)
2009	Classificação	Árvore de Decisão	C4.5	Nominais e Numéricos	(MONDARDO, 2009)
2010	Classificação	Redes Neurais	RBF	Numéricos	(SCOTTI, 2010)
2010	Clusterização	Lógica Fuzzy	RCP	Numéricos	(CROTTI JUNIOR, 2010)
2010	Clusterização	Lógica Fuzzy	URCP	Numéricos	(CROTTI JUNIOR, 2010)
2010	Clusterização	Lógica Fuzzy	FCM	Numéricos	(CROTTI JUNIOR, 2010)
2011	Clusterização	Densidade	DBSCAN	Numéricos	(GAVA, 2011)
2012	Clusterização	Enxame – Colônia de formigas	SACA	Numéricos	(GHELLERE, 2012)
2012	Clusterização	Enxame - Colônia de formigas	Ant-Based Clustering	Numéricos	(GHELLERE, 2012)
2012	Clusterização	Enxame - Colônia de formigas	A²CA	Numéricos	(GHELLERE, 2012)

Fonte: Adaptado de Ghellere (2012).

Para o desenvolvimento da *Shell Orion* optou-se pela utilização da linguagem Java por tratar-se de uma ferramenta gratuita, multiplataforma e que permite a reutilização de códigos (PELEGRIN, 2005).

Uma das vantagens da utilização da plataforma Java é a sua Interface de Programação de Aplicações (*Application Programming Interface* - API) denominada *Java Database Connectivity* (JDBC). O uso desta API possibilita a conexão da *Shell Orion* com qualquer banco de dados desde que possua um *driver* disponível para ela. Isso aumenta a flexibilidade da ferramenta no que se refere a conexão com diversos SGBD (GAVA, 2011).

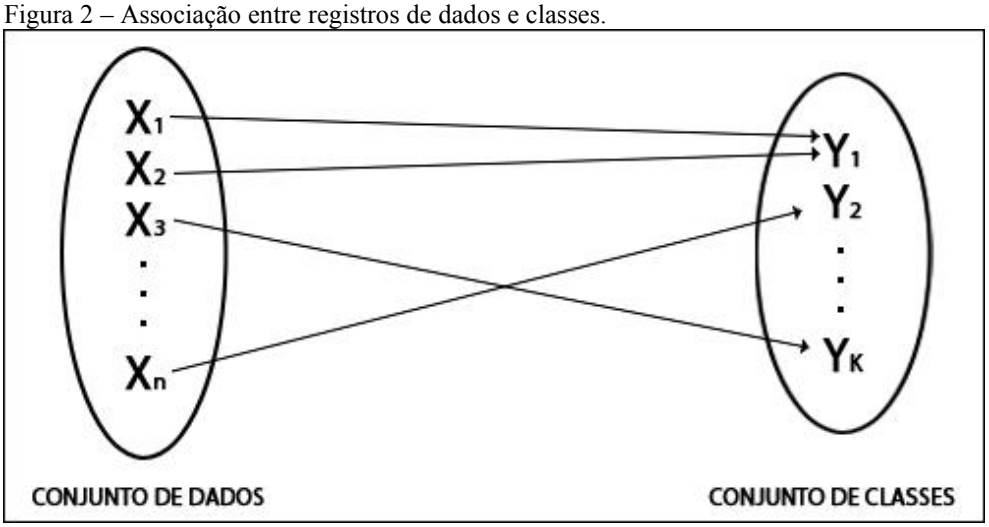
Desde que a *Shell Orion* foi iniciada em 2005, vários TCC fizeram parte do projeto implementando novos algoritmos na ferramenta. Dentre as tarefas implementadas tem-se a classificação. O foco dessa pesquisa é a implementação do algoritmo *Naive Bayes* na tarefa de classificação com o intuito de ampliar as suas funcionalidades.

3 A TAREFA DE CLASSIFICAÇÃO EM *DATA MINING*

A classificação é considerada uma das tarefas usadas com mais frequencia no processo de *data mining*. Consiste na identificação de propriedades comuns entre um conjunto de elementos em uma base de dados e a classificação destes de acordo com os critérios de um determinado modelo (GOLDSCHMIDT; PASSOS, 2005; WITTEN; FRANK; HALL, 2011 tradução nossa).

A tarefa de classificação, conforme mostra a figura 2, pode ser definida como a busca por uma função que possibilite a associação correta de um registro X_i em uma base de dados a um único registro Y_j que corresponde à classe (GOLDSCHMIDT; PASSOS, 2005).

Figura 2 – Associação entre registros de dados e classes.



Fonte: Goldsmidth e Passos (2005).

Durante o processo de classificação, os registros da base de dados a ser explorada são divididos em dois conjuntos. O conjunto dos dados de treinamento define o modelo de classificação. O conjunto dos dados de teste possui os registros que avaliam o modelo de classificação gerado (RUSSEL; NORVIG, 2004).

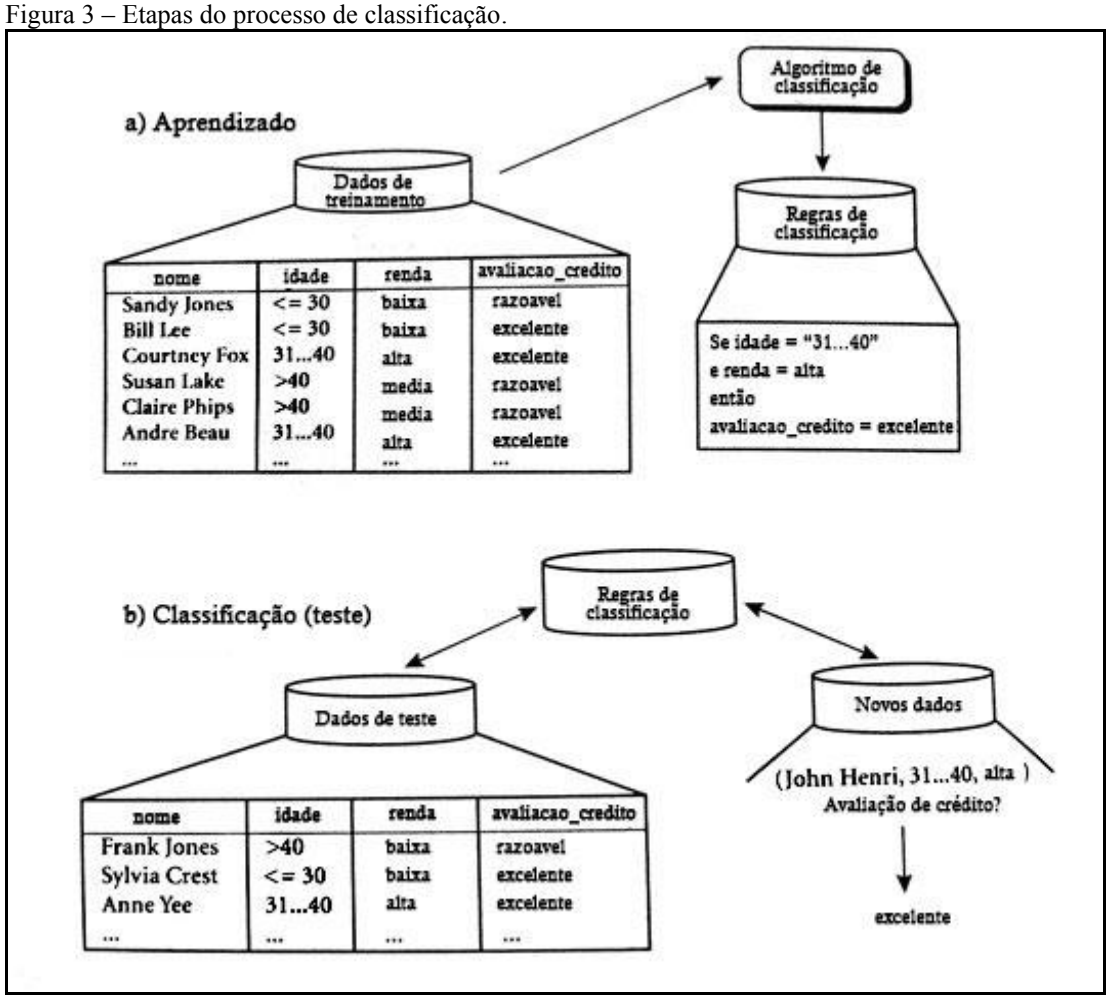
Um algoritmo classificador analisa os dados de treinamento e utiliza os dados de testes para avaliar a exatidão das regras de classificação. As regras então poderão ser aplicadas ao processo de classificação de novos dados caso a exatidão seja aceitável (HAN; KAMBER, 2006, tradução nossa).

Conforme Han e Kamber (2006) este processo acontece em duas etapas (figura 3):

a) **aprendizagem:** o conjunto de dados de treinamento é submetido a um algoritmo classificador e tem-se como resultado o classificador propriamente dito;

b) **teste:** a estimativa da exatidão do classificador é dada pelo conjunto de teste. Critérios como a precisão da predição, custo computacional e robustez são empregados para estimar a precisão do modelo.

Figura 3 – Etapas do processo de classificação.



Fonte: Adaptado de Han e Kamber (2006).

Segundo Tan, Steinbach e Kumar (2009) na tarefa de classificação as variáveis de entrada assumem valores contínuos[1] ou discretos[2], porém o objeto de saída deverá ser discreto.

Normalmente, na tarefa de classificação, é importante medir o desempenho de um classificador, pois essa análise fornece dados para avaliação da sua taxa de erro e generalização. A precisão e taxa de erros nos resultados aplicados em um conjunto de testes pode ser útil para comparar o desempenho de diferentes classificadores em um mesmo

[1] Valor contínuo é um número infinito de valores possíveis dentro do intervalo entre dois valores distintos (DEVORE, 2006).
[2] Valor discreto é um valor finito e numerável que pode ser atribuído a uma variável qualquer (DEVORE,2006).

domínio (TAN; STEINBACH; KUMAR, 2009).

Witten, Frank e Hall (2011) afirmam que a análise da taxa de erro é frequentemente utilizada para avaliar o desempenho de um classificador. Essa taxa é a proporção de erros cometidos ao longo do processo de classificação de um conjunto de eventos e mede o desempenho global de um classificador. Ao prever a classe de uma instância, se estiver correta é contado como um sucesso, caso contrário, um erro.

De acordo com Tan, Steinbach e Kumar (2009) para que essa análise da taxa de erro seja considerável, é necessário que os rótulos das classes dos registros de testes a serem classificados sejam conhecidos. Além disso, o ideal segundo Witten, Frank e Hall (2011), é que os testes sejam realizados em um conjunto de registros de teste que não desempenhou qualquer papel na formação do modelo de classificação para se obter uma estimativa confiável da taxa de erros.

Identificada a necessidade de avaliação do desempenho de classificadores, foram desenvolvidos alguns métodos para a realização das análises. Segundo Tan, Steinbach e Kumar (2009), os métodos usados com mais frequência são:

a) ***holdout*:** os dados com amostras rotuladas são alocados em dois conjuntos distintos: de treinamento e de testes. O modelo de classificação é formado a partir do conjunto de treinamento e o seu desempenho é medido a partir do conjunto de testes. Não existem fundamentos teóricos para fixar a proporção dos dados usados para treinamento e teste. Geralmente adota-se uma percentagem fixa p para treinamento e $(1-p)$ para testes, sendo $p=2/3$ e $(1-p)=1/3$ (REZENDE, 2005);

b) **amostragem aleatória:** consiste basicamente em repetir o método *holdout* diversas vezes de maneira aleatória. Segundo Rezende (2005) a amostragem aleatória, em alguns casos, apresenta melhores estimativas que o método *holdout*;

c) **validação cruzada ou *cross-validation*:** neste método cada registro é usado para treino um mesmo número de vezes e apenas uma única vez para teste. Em outras palavras, usa-se um dos subconjuntos de dados para treinamento outro para teste. Em seguida os papéis se invertem de modo que o subconjunto que antes era de teste agora passa a ser de treinamento e vice-versa. O erro total é o resultado da soma de erros das duas execuções;

d) ***bootstrap*:** consiste em repetir a classificação diversas vezes a partir dos experimentos replicados em um novo conjunto de treinamento obtido por

amostragem do conjunto original de registros com substituição, ou seja, um registro já escolhido volta para o conjunto de treinamento com a possibilidade de ser escolhido novamente. Essa característica difere o *bootstrap* dos demais métodos, pois estes não apresentam registros duplicados no conjunto de treinamento.

Os métodos de avaliação de desempenho auxiliam no entendimento da capacidade e limitações dos diversos classificadores no *data mining*. Segundo Rezende (2005) o aprendizado de máquina é uma poderosa ferramenta, porém não existe um único algoritmo com melhor desempenho para todos os problemas.

A escolha do método de avaliação de desempenho de classificadores depende do tipo de problema a ser analisado e do método de classificação escolhido. A aplicação da tarefa de classificação dentro do processo de *data mining*, requer o uso de métodos específicos. Dentre estes tem-se os estatísticos, as árvores de decisão, os algoritmos genéticos, as redes neurais, entre outros.

Esta pesquisa aborda o método baseado nos fundamentos da estatística probabilística e na inferência bayesiana como suporte para a construção de um classificador bayesiano.

3.1 O MÉTODO ESTATÍSTICO NA TAREFA DE CLASSIFICAÇÃO

Segundo Elian e Farhat (2006) e Finn (1996) a Estatística é uma ciência focada na organização, descrição, análise e interpretação dos dados. Desde a sua criação no século XVII, tem proporcionado o uso de ferramentas matemáticas e técnicas analíticas para lidar com grandes quantidades de dados. Atualmente, com volumes cada vez maiores de dados, a Estatística é mais do que nunca, um componente crítico do *data mining* que facilita a tomada de decisões eficazes.

Essa disponibilidade de ferramentas matemáticas e técnicas de análises na Estatística tem possibilitado o desenvolvimento de diversos algoritmos dentre os existentes no *data mining*, bem com tem auxiliado na avaliação dos resultados oriundos do teste (GOLDSCHMIDT; PASSOS, 2005).

Elian e Farhat (2006) afirmam que por auxiliarem na tomada de decisões na presença de incerteza por aleatoriedade, as técnicas estatísticas objetivam a análise de experimentos aleatórios. A incerteza está associada à probabilidade da ocorrência de um resultado de interesse.

3.2 A ESTATÍSTICA PROBABILÍSTICA

Devore (2006) afirma que probabilidade é o estudo da incerteza por aleatoriedade. Em um conjunto de diversos resultados, a teoria da probabilidade, por meio de métodos específicos, permite quantificar as chances ou crenças de ocorrência associadas a estes resultados.

O raciocínio probabilístico está presente em uma vasta diversidade de fenômenos que envolvem a incerteza por aleatoriedade e pode ser classificado em três dimensões (CAMPOS; RÊGO, 2011):

a) **grau de precisão:** representa a precisão esperada do raciocínio probabilístico ao representar fenômenos aleatórios;
b) **significado ou interpretação:** indica o que se pode aprender com a probabilidade e o que significa uma afirmação probabilística;
c) **estrutura matemática formal:** consiste na estrutura formal da função probabilística dada por regras, leis ou propriedades, escritas na forma computacional ou lógica.

Segundo Campos e Rêgo (2011), Lipschutz (1994), e Meyer (1969) denomina-se experimento aleatório ou não-determinístico todo experimento que, quando executado repetidamente, apresenta resultados diferentes devido ao acaso. Porém, quando a experiência é repetida uma grande quantidade de vezes é possível observar uma certa regularidade nos resultados. A percepção desta regularidade torna possível a construção de um modelo matemático probabilístico preciso, que auxiliará na identificação de padrões e características do processo experimental sem que haja necessidade de se refazer o experimento.

O conjunto de todos os resultados possíveis do processo experimental aleatório define o espaço amostral (DEVORE, 2006).

Um mesmo experimento ε, como lançar uma moeda em um determinando número de vezes, por exemplo, apresenta espaços amostrais diferentes dependendo do que se está tentando mensurar ou observar. Neste contexto, deve-se referir ao espaço como "um" espaço amostral e não como "o" espaço amostral (CAMPOS; RÊGO, 2011; MEYER, 1969).

De acordo com Devore (2006) qualquer grupo de resultados possíveis contido em um particular espaço amostral *S* associado a um experimento ε determina um evento.

Em termos, evento é qualquer subconjunto de um espaço amostral formado pelos resultados possíveis do experimento aleatório. Em um determinado experimento ε aleatório, se o resultado pertencer a um dado evento *A*, diz-se que *A* ocorreu (CAMPOS; RÊGO, 2011).

Tendo-se repetido *n* vezes um experimento ε, sendo *A* um evento associado a este experimento, considera-se n_A o número de vezes que o evento *A* ocorreu nas *n* repetições. A razão n_A/n é chamada de frequência relativa do evento *A* durante as repetições do experimento ε. É possível observar que a frequência tende a estabilizar-se aproximando-se de um limite. Esta estabilidade atribui-se a base da teoria da probabilidade (LIPSCHUTZ, 1994).

3.2.1 Probabilidade Clássica

De acordo com Lipschutz (1994), a probabilidade, definida como o estudo de eventos aleatórios ou não-determinísticos teve seu início, historicamente, com o estudo dos jogos de azar, como as cartas e a roleta. Sempre que um experimento consiste em *n* resultados possíveis e igualmente prováveis, como o lançamento de um dado, por exemplo, onde cada uma das faces tem a mesma chance de aparecer, a probabilidade de cada resultado ocorrer é 1/*n*. A probabilidade definida nestas condições é denominada probabilidade *a priori*.

Nesse contexto, a probabilidade *p*, de um evento *A*, foi definida da seguinte forma: se *A* ocorre de *s* maneiras diversas em um total de *n* maneiras igualmente prováveis, então tem-se:

$$P = P(A) = \frac{s}{n} \tag{1}$$

Onde *n* é o número de possíveis resultados (elementos do espaço amostral) de *A* e *s* é o número de resultados favoráveis de *A* (elementos de *A*) dentre o número de resultados possíveis. Este raciocínio resulta no conceito clássico de probabilidade.

3.2.2 Probabilidade Condicional

De acordo com Campos e Rêgo (2011), existem várias possíveis interpretações para a probabilidade e esta, por sua vez, é fundamentada em informação e conhecimento. A informação ou conhecimento da ocorrência de determinado evento pode exercer influência sobre a probabilidade dos demais eventos.

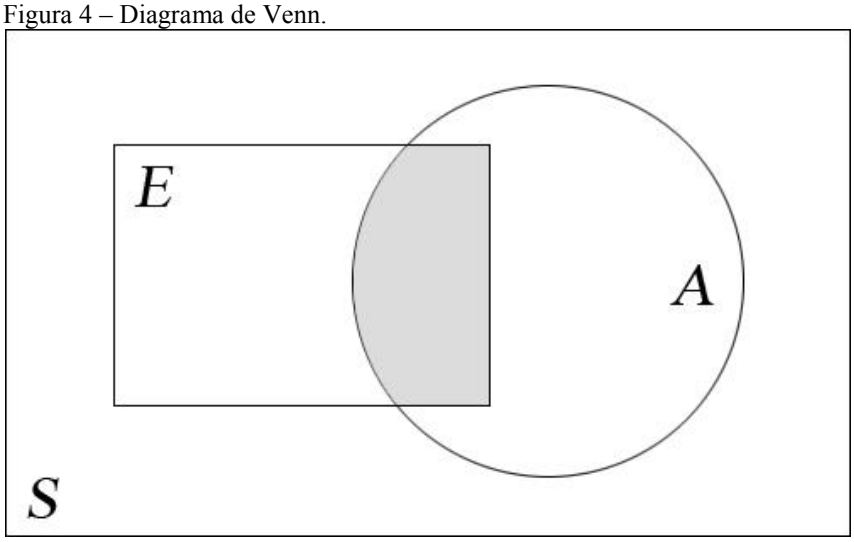
A probabilidade condicional trata a probabilidade da ocorrência de um evento *A*, tendo ocorrido um evento arbitrário *E* em um espaço amostral *S*, associados a um experimento ε, onde *P(E) > 0*. Tem-se, portanto, a probabilidade condicional de *A* dado *E* ou a probabilidade de *E* condicionada a *A*, representada por *P(A|E)*, definida como (LIPSCHUTZ, 1994):

$$P(A \mid E) = \frac{P(A \cap E)}{P(E)} \tag{2}$$

A cada cálculo de *P(A | E)*, necessariamente se está calculando *P(A)* em relação ao espaço amostral reduzido *A* ao invés de fazê-lo em relação ao espaço amostral original *S*. Em termos, calcular *P(A)* é equivalente a determinar a probabilidade de estar no subconjunto do evento *A* sabendo que também é necessário estar em *S, e*nquanto que calcular *P(A|E)* é equivalente a determinar a probabilidade de estar no subconjunto do evento *A* sabendo que também é necessário estar em *E*, ou seja, o espaço amostral é reduzido de *S* para *E* (LIPSCHUTZ, 1994).

A definição acima é ilustrada no Diagrama de Venn representado na figura 4.

Figura 4 – Diagrama de Venn.



Fonte: Adaptado de Lipschutz (1994).

O conceito de probabilidade condicional possibilita estabelecer a expressão para o cálculo da probabilidade da ocorrência simultânea de dois eventos *A* e *E*. Esta expressão é conhecida como Teorema da Multiplicação (DEVORE, 2006).

Segundo Lipschutz (1994), a multiplicação em cruz de ambos os lados da equação que define a probabilidade condicional resulta na seguinte fórmula:

$$P(A \cap E) = P(A \mid E) \cdot P(E) \text{ ou } P(A \cap E) = P(E \mid A) \cdot P(A) \quad (3)$$

Ou seja, a probabilidade condicional é definida pela probabilidade de um evento *A* tendo ocorrido o evento *E* e é medida pelo quociente da probabilidade da intersecção de *A* com *E* dividida pela probabilidade de *E* (equação 2). Multiplicando-se a equação (2) em cruz conclui-se que a probabilidade da intersecção do evento *A* com o evento *E* é igual a probabilidade de *A* tendo ocorrido o evento *E* multiplicada pela probabilidade de *E*. Como os eventos são independentes, ou seja, a ocorrência de um não afeta a ocorrência do outro, a probabilidade de *A* é condicionada a ocorrência do evento *E* (*P(A|E)*.

De acordo com Devore (2006), a regra mostrada na equação (3) apresenta mais utilidade quando estendida a experimentos que envolvem mais de duas etapas consecutivas, ou seja, quando um evento *E* tem *n* resultados possíveis $A_1, A_2, \ldots, A_n$. Neste contexto, o evento condicional *E* mostrado na equação (3) descreve o resultado da primeira etapa e *A* o resultado da segunda, conforme pode-se observar na equação (4):

$$P(A_1 \cap A_2 \cap A_3) = P(A_3 \mid A_1 \cap A_2) \cdot P(A_1 \cap A_2) = P(A_3 \mid A_1 \cap A_2) \cdot P(A_2 \mid A_1) \cdot P(A_1) \quad (4)$$

Onde A_1 ocorre primeiro, seguido por A_2 e consequentemente por A_3.

Segundo Devore (2006), em experimentos constituídos de diversas etapas, a aplicação simples do Teorema da Multiplicação fornece as possibilidades para a utilização do Teorema de Bayes no qual se baseiam os classificadores bayesianos.

4 CLASSIFICADORES BAYESIANOS

Alguns métodos tradicionais de classificação no *data mining* podem apresentar um tempo de aprendizado mais longo durante a fase de treinamento do algoritmo, tornando comum a necessidade de se repetir a etapa de aprendizagem do algoritmo (HAN; KAMBER, 2006, tradução nossa).

Considerando-se que esse aspecto não é positivo do ponto de vista do *data mining*, tem-se outros métodos de classificação, como por exemplo, os baseados no Teorema de Probabilidade de Thomas Bayes, tendo-se portanto os classificadores bayesianos. Segundo Coppin (2010) e Tan, Steinbach e Kumar (2009) estes classificadores geralmente são usados para lidar com situações de incerteza por aleatoriedade, combinando o conhecimento prévio acerca de uma hipótese com novas evidências colhidas dos dados que se pretende analisar, simplificando os cálculos envolvidos e obtendo um menor tempo no processo de aprendizagem.

Classificadores bayesianos são classificadores estatísticos que podem prever as probabilidades de associação a uma determinada classe, como por exemplo, estimar que um determinado registro pertença a uma classe em particular (HAN; KAMBER, 2006, tradução nossa).

De acordo com Coppin (2010), Han e Kamber (2006) tradução nossa e Tan, Steinbach e Kumar (2009) dentre estes classificadores, os mais comuns usados no *data mining* são:

a) **Classificador Ingênuo de Bayes ou *Naive Bayes*:** a principal característica é a suposição da independência condicional de classe, ou seja, o algoritmo assume que os atributos do evento são condicionalmente independentes;
b) **Redes Bayesianas ou Redes de Crenças Bayesianas:** a suposição da independência condicional não se aplica a todos os atributos. Permite especificar qual par de atributos é condicionalmente independente.

As redes bayesianas de crenças e o classificador ingênuo de Bayes são classificadores estatísticos que usam o paradigma bayesiano em lugar do paradigma da estatística clássica.

4.1 O PARADIGMA BAYESIANO

Segundo Paulino, Turkman e Murteira (2003) o conceito de probabilidade aplicado como grau de credibilidade, importante para o entendimento da filosofia bayesiana, vem de longa data. Jakob Bernoulli, por meio da sua obra *Ars Conjectandi* publicada em 1713, foi um dos primeiros autores a definir a probabilidade como grau de confiança em um dado evento, o qual não se sabe se é verdadeiro ou falso.

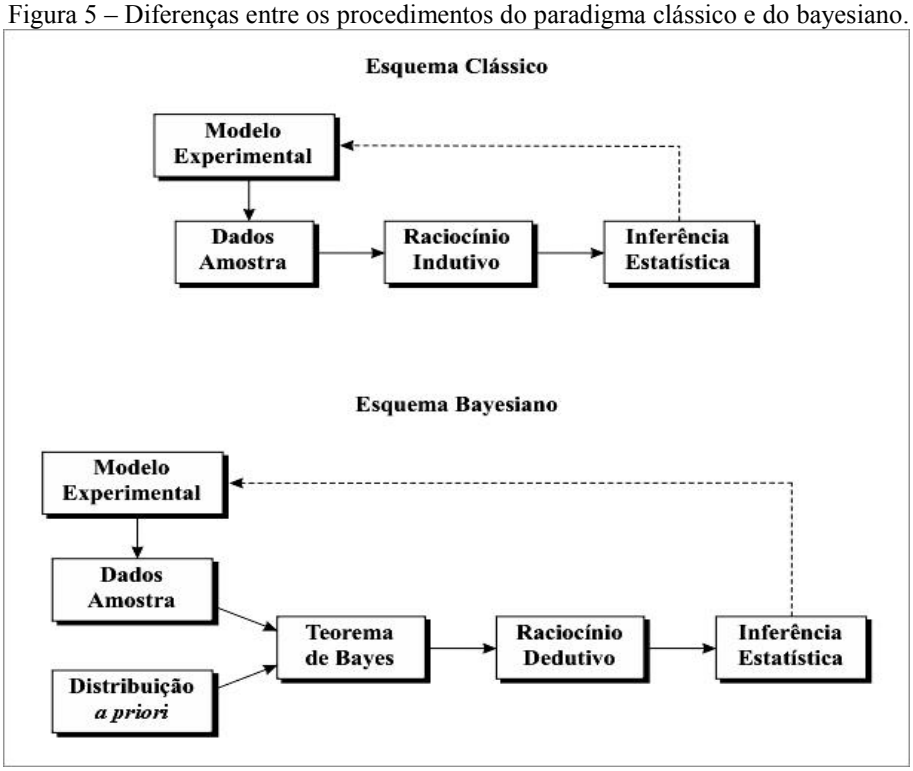
A mudança de paradigmas do clássico para o bayesiano foi considerada uma revolução científica (LINDLEY, 1990).

Richard Price, em 1763, foi responsável por desencadear a ascensão do paradigma bayesiano em problemas de inferência com a publicação da obra póstuma do reverendo Thomas Bayes intitulada *An Essay Towards Solving a Problem in the Doctrine of Chances*. O Teorema de Bayes tornou-se um dos poucos resultados da matemática que se propõe a determinar a aprendizagem conforme a experiência, ou seja, modificar a atitude inicial em relação às hipóteses, causas ou antecedentes, após receber a informação de que certo acontecimento se realizou (PAULINO; TURKMAN; MURTEIRA, 2003).

Ainda de acordo com Paulino, Turkman e Murteira (2003) no paradigma bayesiano a classificação *a posteriori* engloba, por meio do Teorema de *Bayes*, todas as informações acerca do parâmetro, ou seja, as informações iniciais mais a da experiência ou amostra.

A figura 5 ilustra as diferenças entre os procedimentos utilizando o paradigma clássico e o bayesiano.

Figura 5 – Diferenças entre os procedimentos do paradigma clássico e do bayesiano.



Fonte: Adaptado de Paulino, Turkman e Murteira (2003).

No esquema clássico a amostra de dados retirada do modelo experimental passa diretamente pelo raciocínio indutivo[3] antes de iniciar o processo de inferência, sem ter interpretação direta no que diz respeito à probabilidade. No esquema bayesiano a amostra é retirada do modelo experimental juntamente com a distribuição *a priori*, em seguida o teorema de Bayes é aplicado a fim de se obter a distribuição *a posteriori* e então, com o raciocínio dedutivo[4], chegar à inferência estatística (PAULINO; TURKMAN; MURTEIRA, 2003).

4.2 A INFERÊNCIA BAYESIANA

Segundo Neapolitan (2004), a inferência bayesiana é definida como o processo para a dedução da probabilidade *a posteriori* a partir da probabilidade *a priori*. Ou seja, é a dedução de valores invisíveis por meio da probabilidade onde a incerteza acerca de tais

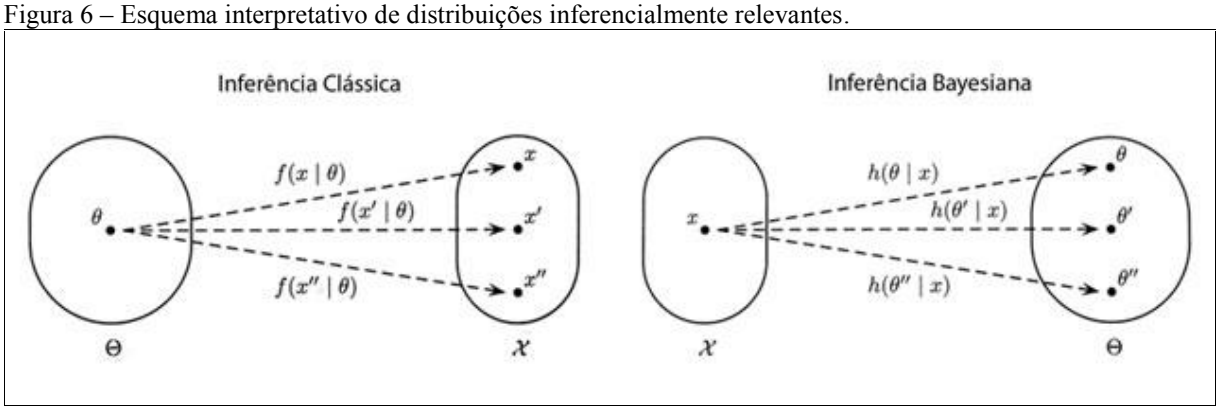
[3] Raciocínio indutivo é o processo pelo qual é possivel chegar-se a uma generalização, dadas diversas particularidades. Por exemplo, concluir que o paciente sofre de determinada doença observando os diversos sintomas que ele apresenta (RHODES, 1994).

[4] Raciocínio dedutivo é o processo pelo qual é possivel deduzir particularidades acerca de uma generalização. Por exemplo, concluir que o paciente possa apresentar determinados sintomas dado que sabe-se a doença da qual ele sofre (RHODES, 1994).

valores é modificada regularmente com base em observações de novos resultados utilizando o teorema de *Bayes*.

De acordo com Paulino, Turkman e Murteira (2003) um dos objetivos dos procedimentos bayesianos é realizar inferências sobre o parâmetro não observável θ. As inferências clássicas baseiam-se em probabilidades relacionadas a diferentes amostras x que podem ocorrer para algum valor desconhecido e fixo do parâmetro θ, ao passo que as inferências bayesianas baseiam-se nas credibilidades subjetivas, ou seja, probabilidades *a posteriori* relacionadas a diferentes valores do parâmetro θ e condicionadas ao valor de x. Neste contexto, o ponto x é fixo e considera-se a variação de θ como mostra a figura 6.

Figura 6 – Esquema interpretativo de distribuições inferencialmente relevantes.



Fonte: Adaptado de Paulino, Turkman e Murteira (2003).

A probabilidade bayesiana traça observações probabilísticas sobre os parâmetros, considerando variáveis aleatórias. O que interessa é o valor resultante de x e não o espaço amostral. Na probabilidade clássica o enfoque está nos dados e este fato relaciona-se diretamente com o espaço amostral que é percorrido nas sucessivas repetições do processo de amostragem (PAULINO; TURKMAN; MURTEIRA, 2003).

4.3 TEOREMA DE BAYES

O reverendo Thomas Bayes foi um teólogo e matemático inglês que viveu entre 1702 e 1761, ele desenvolveu este teorema que é bastante usado nos dias de hoje para resolver problemas nos quais não existe certeza (COPPIN, 2010).

De acordo com Devore (2006) e Meyer (1964) o Teorema de Bayes é também denominado Fórmula da Probabilidade das Causas ou dos Antecedentes. A regra fundamenta-se no cálculo da probabilidade *a posteriori* $P=(A_i \mid E)$ a partir das probabilidades *a priori*

$P=(A_i)$ e nas probabilidades condicionais $P=(E \mid A_i)$. Ou seja, no cálculo da probabilidade de que um evento venha a ocorrer, dado que parte da informação seja previamente conhecida (COPPIN, 2010).

Seja um experimento ε onde A e E são dois eventos de um espaço amostral S, considera-se que a probabilidade de observar simultaneamente A e E por meio do Teorema da Multiplicação é dada por $P(A \cap E) = P(A \mid E) \cdot P(E)$ ou $P(A \cap E) = P(E \mid A) \cdot P(A)$, pode-se afirmar que (LIPSCHUTZ, 1994):

$$P(A \mid E) \cdot P(E) = P(E \mid A) \cdot P(A) \tag{5}$$

Reordenando a expressão dada pela equação (5) tem-se o Teorema de Bayes:

$$P(E \mid A) = \frac{P(A \mid E).P(E)}{P(A)} \tag{6}$$

Como não se conhece necessariamente $P(A)$, a probabilidade pode ser reescrita na seguinte forma:

$$P(A) = P(A \cap E) + P(A \cap E') \tag{7}$$

Onde E' é o evento complementar de E.

Pelo Teorema da Multiplicação pode-se substituir $P(A \cap E)$ e $P(A \cap E')$ da expressão acima obtendo-se:

$$P(A) = P(A \mid E) \cdot P(E) + P(A \mid E') \cdot P(E') \tag{8}$$

Tendo-se estabelecido a expressão acima, substitui-se $P(A)$ no Teorema de Bayes obtendo-se a formulação alternativa:

$$P(E \mid A) = \frac{P(A \mid E) \cdot P(E)}{P(A \mid E) \cdot P(E) + P(A \mid E') \cdot P(E')} \tag{9}$$

Considerando-se então os eventos E_1, E_2, ..., E_n como mutuamente exclusivos e que formam a partição do espaço amostral S e A um evento qualquer, então:

$$P(E_i \mid A) = \frac{P(A \mid E_i) \cdot P(E_i)}{P(A \mid E_1) \cdot P(E_1) + P(A \mid E_2) \cdot P(E_2) + ... + P(A \mid E_n) \cdot P(E_n)} \quad (10)$$

Assim, define-se na equação (10) que desde que E_i constitua uma partição do espaço amostral S, um e somente um dos eventos E_i irá ocorrer. Esta condição define a probabilidade de um evento E_i ter ocorrido dado que o evento A ocorreu (DEVORE, 2006; LIPSCHUTZ, 1994; MEYER, 1969).

Meyer (1969) e Portugal (2004) afirmam que o Teorema de *Bayes* não é controverso do ponto de vista matemático. O seu resultado torna-se discutível apenas se houver uma escolha imprópria dos $P(E_i)$, ou seja, se as informações que deveriam ser previamente conhecidas para encontrar $P(E_i)$ apresentem algum grau de incerteza que possa comprometer os resultados, como por exemplo, ambiguidade, contradição ou imprecisão semântica.

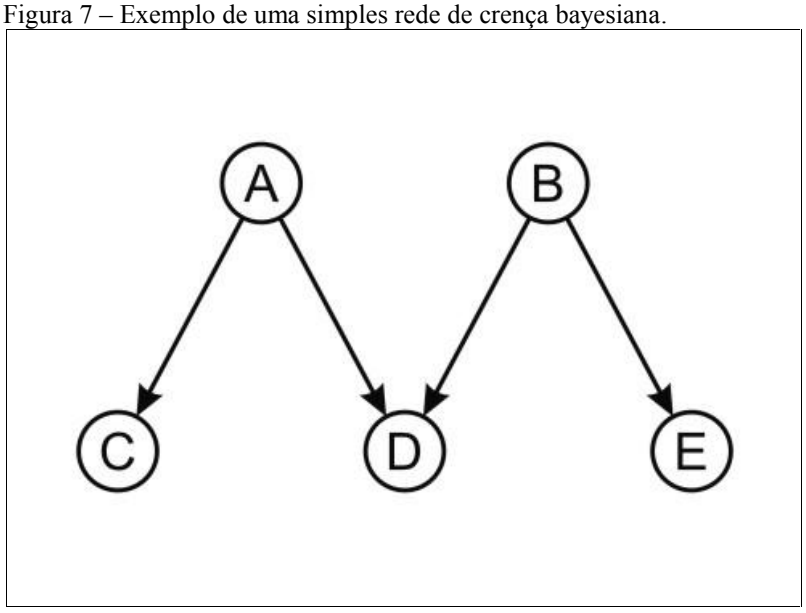
Esta característica de precisão nos resultados justifica o uso do Teorema de *Bayes* e do paradigma bayesiano por classificadores estatísticos, como o *Naive Bayes*, para estimar a probabilidade relevante em conjuntos de dados (DUDA; HART; STORK, 2000, tradução nossa).

4.4 O ALGORITMO NAIVE BAYES

O algoritmo *Naive Bayes* foi proposto por Richard Duda e Peter Hart no livro *Pattern Classification and Scene Analisys* em 1973 na Califórnia, Estados Unidos, originado do trabalho de reconhecimento de padrões e análise de cenas em aprendizado de máquina. O livro teve a sua segunda edição publicada em Nova Iorque por Richard Duda, Peter Hart e David Stork em 2000.

É um classificador estatístico fundamentado no paradigma bayesiano. Como as redes de crenças bayesianas, ele tem como base o teorema de Bayes, mas se difere por considerar a independência condicional entre os atributos. Nas redes bayesianas essa suposição de independência condicional não se aplica a todos os atributos (TAN; STEINBACH; KUMAR, 2009).

Figura 7 – Exemplo de uma simples rede de crença bayesiana.



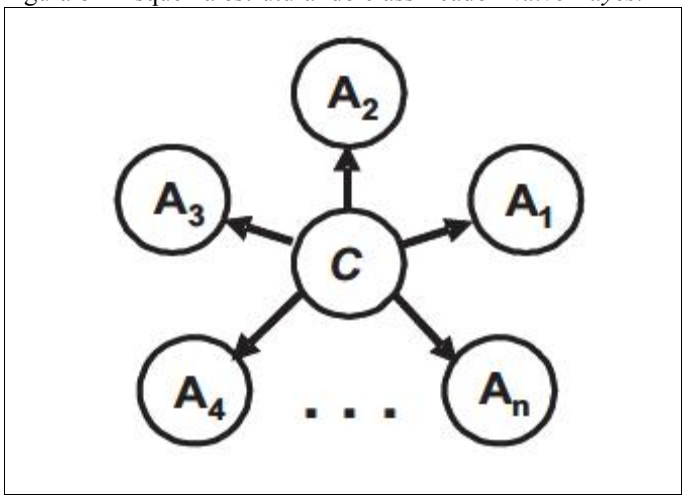
Fonte: Adaptado de Coppin (2010).

A rede bayesiana é um grafo acíclico onde cada nó representa as hipóteses e evidências e cada arco, quando conectado a dois nós, representa a dependência entre eles. A rede de crença bayesiana (figura 7) dispõe de cinco nós que representam duas evidências A e B e três hipóteses C, D e E. Os nós C e D são dependentes de A, e D e E são dependentes de B. A é independente de B, pois não existe um arco ligando um ao outro (COPPIN, 2010).

O classificador *Naive Bayes* pode ser considerado como uma rede bayesiana que apresenta uma estrutura em forma de estrela onde cada nó que representa os atributos são

independentes entre si (BORGELT; KRUSE, 2002). Esta estrutura é representada na figura 8 (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997):

Figura 8 – Esquema estrutural do classificador *Naive Bayes*.



Fonte: Adaptado de Friedman, Geiger e Goldszmidt (1997).

Cada registro de dados a ser classificado consiste em um conjunto de atributos que podem assumir diversos valores entre si, sendo classificados com um único rótulo de classe (TAN; STEINBACH; KUMAR, 2009; COPPIN, 2010).

Dois parâmetros de entrada são necessários para o funcionamento do algoritmo, sendo um conjunto de dados treinamento onde a classe para cada registro é conhecida e um conjunto de atributos a ser classificado que pode ser chamado de conjunto de testes.

Formalmente, para identificar a classificação de uma instância de dados calcula-se a probabilidade *a posteriori* de cada classificação possível. A classificação correta será aquela cuja probabilidade *a posteriori* for maior. Essa hipótese é conhecida como máxima *a posteriori* (MAP) ou hipótese MAP (COPPIN, 2010).

De acordo com Tan, Steinbach e Kumar (2009) o classificador *Naive Bayes* possui as seguintes características:

a) **robustez para lidar com valores faltantes:** caso apresente ausência de qualquer valor, a instância é ignorada durante o processo de construção do modelo;
b) **robustez para tratar atributos irrelevantes**: se X_i condicionado a Y for irrelevante, então $P(X_i|Y)$ é uniformemente distribuída e a probabilidade condicional de classe para X_i não causa impacto no cálculo da probabilidade *a posteriori*;

c) **deficiência para lidar com atributos correlacionados:** pode ter seu desempenho degradado por esses atributos uma vez que a suposição de independência condicional é violada para tais atributos.

O processo de classificação pelo algoritmo *Naive Bayes* consiste em dois passos (GOLDSCHMIDT; PASSOS, 2005):

a) realizar o cálculo da probabilidade $P(C=C_i|R)$, $i=1,2,...,n$;

b) indicar como saída do algoritmo a classe Cj tal que $P(C=Cj|R)$ seja máxima;

Estes passos são repetidos até que todos os registros sejam lidos.

A figura 9 exibe o pseudocódigo para o algoritmo *Naive Bayes*:

Figura 9 – Pseudocódigo do classificador *Naive Bayes*

```
1:   Algoritmo Naive Bayes
2:   Input: Exemplos para treinamentos, Conjunto de testes
3:   Output: Classe predita
4:   begin
5:       for cada classe Ci do
6:           calcular a probabilidade a priori P(Ci);
7:       end
8:       for cada atributo Ai de um conjunto de testes do
9:           calcular a probabilidade a posteriori P(Ai|Ci)
10:          multiplicar todos os valores calculados para a classe Ci
11:          multiplicar o valor obtido pela probabilidade a priori da classe Ci
12:          verificar classe predita pela maxima a posteriori
13:      end
14:  end
```

Fonte: Adaptado de Silva (2007)

Han e Kamber (2006) tradução nossa, descrevem a execução do algoritmo *Naive Bayes* da seguinte forma:

Dado um conjunto de treinamento S, sendo que cada amostra de S é representada por um vetor n-dimensional $(x_1,...,x_n)$ que corresponde aos valores de cada um dos n atributos $A_1,...,A_n$, considerando-se a existência de m classes $C_1,...,C_m$, que o número de amostras de treinamento por classe seja igual a $S_1,...,S_n$ e dada uma amostra desconhecida X, o classificador vai indicar que a amostra pertence a classe que tem maior probabilidade *a posteriori* condicionada a X.

Ou seja, o classificador atribui a amostra X à classe C_i se e somente se:

$$P(C_i \mid X) > P(C_j \mid X), 1 \le j \le m, j \ne i \qquad (11)$$

Para o cálculo da probabilidade *a posteriori*, aplica-se o teorema de Bayes reescrevendo-o como:

$$P(C_i \mid X) = \frac{P(X \mid C_i).P(C_i)}{P(X)},\ i = 1,...,m \tag{12}$$

Então:

$$P(X \mid C_i) \cdot P(C_i) > P(X \mid C_j) \cdot P(C_j), 1 \leq j \leq m, j \neq i \tag{13}$$

Do conjunto de treinamento, calcula-se os valores de $P(C_i)$:

$$P(C_i) = \frac{S_i}{S},\ i = 1,...,m \tag{14}$$

Segundo Coppin (2010) e Tan, Steinbach e Kumar (2009) como o objetivo é encontrar a maior probabilidade e $P(X)$ é uma constante independente de C_i, considera-se apenas a classe que maximizar o termo numerador e pode-se descartar a constante $P(X)$.

O classificador assume então que cada um dos atributos é condicionalmente independente:

$$P(C_i \mid X) = \prod_{k=1}^{n} P(x_k \mid C_i) \tag{15}$$

Se o atributo A_k é discreto, a probabilidade condicional $P(x_k|C_i)$ é dada pela fração de instâncias de treinamento C_i que recebem como atributo o valor de x_k. Por exemplo, se no conjunto de treinamento dois em cada cinco valores de x_k pertencer à classe C_i, então a probabilidade condicional para $P(x_k|C_i)$ é igual a 2/5 (TAN; STEINBACH; KUMAR, 2009). Desse modo:

$$P(x_k \mid C_i) = \frac{S_{ik}}{S_i} \tag{16}$$

Onde S_{ik} é o número de amostras do conjunto de treinamento com classe C_i que possuem valor x_k para A_k e S_i é o número total de amostras pertencentes à classe C_i.

Se o atributo A_k é contínuo, pode-se discretizar[5] cada atributo contínuo usando-os como um atributo discreto, dessa forma a probabilidade condicional $P(x_k|C_i)$ é dada pela fração de instâncias de treinamento C_i que estejam dentro do intervalo correspondente a x_k. Ou adota-se uma distribuição Gaussiana[6] (HAN; KAMBER, 2006, tradução nossa; TAN; STEINBACH; KUMAR, 2009):

$$P(x_k \mid C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi\sigma_{C_i}}} e^{\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}}} \quad (17)$$

Onde $g(x_k, \mu_{C_i}, \sigma_{C_i})$ é a função Gaussiana do atributo A_k eenquanto μ_{C_i} e σ_{C_i} são respectivamente a média[7] e o desvio padrão[8] dos valores de A_k das amostras de treinamento de C_i.

Para determinar a classe C_i da amostra X, substituem-se os valores calculados nas regras de classificação da equação 13.

4.4.1 O problema da frequência zero

O cálculo da probabilidade *a posteriori* a partir dos dados do conjunto de treinamento pode apresentar um problema relacionado à freqüência zero, ou seja, um determinado atributo x_k pode não ter valor algum pertencente à classe C_i . Por exemplo, se no conjunto de treinamento nenhum valor de x_k em cada 5 registros pertencer à classe C_i, então a probabilidade condicional para $P(x_k|C_i)$ é igual a 0/5=0. Nesse caso a instância pode ser classificada incorretamente (TAN; STEINBACH; KUMAR, 2009).

---

[5] Discretização é o processo de substituição de um valor contínuo de um determinado atributo pelo seu intervalo discreto correspondente (TAN; STEINBACH; KUMAR, 2009).

[6] A distribuição Gaussiana ou Normal é caracterizada por uma função de probabilidade, cujo gráfico descreve uma curva em forma de sino. Essa forma de distribuição evidência que existe maior probabilidade da variável aleatória assumir valores próximos do centro (BARBETTA; REIS; BORNIA, 2010).

[7] Média é um valor hipotético dado pelo resultado da divisão da soma dos valores obtidos de uma variável pelo número de valores contados (FIELD, 2009).

[8] Desvio Padrão determina a dispersão dos valores em relação à média, ou seja, é a medida de quão bem a média representa os valores da variável (FIELD, 2009).

Segundo Han e Kamber (2006) para evitar esse problema, aplica-se o teorema da aproximação de Laplace[9]:

$$\frac{n_c + 1}{n + k} \tag{18}$$

Onde n_c é o número de instâncias com o valor de x_k pertencente à classe C_i, n é o número total de instâncias de treinamento com classe C_i e k corresponde à quantidade máxima de valores distintos que x_k pode assumir, ou seja, se um atributo x_k pode assumir apenas os valores a, b e c, então a quantidade máxima de valores distintos que x_k pode assumir é k=3.

A constante 1, adicionada ao numerador da fração, anula a frequência zero e garante um valor válido para o cálculo da probabilidade. Usa-se então a equação (19) ao invés da equação (16):

$$P(x_k \mid C_i) = \frac{S_{ik} + 1}{S_i + k} \tag{19}$$

Dessa forma se, no conjunto de treinamento S, treinamento nenhum valor de x_k em cada 5 registros pertencer à classe C_i, sendo que hipoteticamente a quantidade máxima de valores assumida por x_k é 3, então a probabilidade condicional para $P(x_k|C_i)$ passa a ser:

$$P(x_k \mid C_i) = \frac{0 + 1}{5 + 3} = \frac{1}{8} = 0{,}125 \tag{20}$$

O uso do método de aproximação de Laplace possibilita que $P(x_k|C_i)$ seja > 0, garantindo que, caso as probabilidades *a posteriori* das demais classes sejam menores que o resultado obtido na iteração, $P(x_k|C_i)$ seja escolhida como a MAP.

De acordo com Coppin (2010), a suposição de independência condicional entre os valores dos atributos simplifica os cálculos envolvidos e faz do *Naive Bayes* um sistema de aprendizagem simples, mas efetivo.

[9] Pierre Laplace foi um matemático francês que viveu entre 1749 e 1827. Formulou a definição de que a probabilidade é a divisão do número dos casos prováveis pelo número de casos possíveis (KORB; NICHOLSON, 2004, tradução nossa; PAULINO; TURKMAN; MURTEIRA, 2003).

Compreendido o funcionamento do algoritmo *Naive Bayes*, serão apresentados no próximo capítulo alguns exemplos da sua aplicação na tarefa de classificação em *data mining*.

4.5 TRABALHOS CORRELATOS

A inferência bayesiana tem sido aplicada em diversos estudos e práticas como a tarefa de classificação no processo de *data mining* (GOLDSCHMIDT; PASSOS, 2005). Nesse contexto o classificador *Naive Bayes* vem despertando o interesse dos pesquisadores, pois embora sua implementação seja simples quando comparada a outros classificadores, geralmente apresenta bons índices de validação nos resultados preditivos e um tempo relativamente pequeno durante a fase de treinamento (DOMINGOS; PAZZANI, 1997, tradução nossa; HAN; KAMBER, 2006, tradução nossa). Alguns destes estudos e aplicações acerca do classificador *Naive Baye*s são abordados sucintamente a seguir.

4.5.1 Aprendizagem Bayesiana Simples em Dados Estruturais

Esta tese foi submetida por Michelangelo Ceci para obtenção do grau de Doutor em Filosofia da Ciência da Computação na *Università degli Studi di Bari*, na Itália em 2005.

Trata-se da implementação do algoritmo *Naive Bayes* em três abordagens no campo de *data mining*: a classificação probabilística no processo clássico de *data mining* incluindo o problema da classificação hierárquica; a classificação de dados persistentes em tabelas multi relacionais; a categorização de textos e reconhecimento de imagens em documentos do campo da Engenharia (CECI, 2005).

De acordo com o autor o algoritmo apresentou bons resultados em todos os experimentos, com destaque para a sua eficiência quando fortemente integrado a um banco de dados relacional. Nas tarefas de categorização e classificação de documentos apresentou resultados positivos no conceito de identificação de índices, títulos e número de páginas. Apenas no conceito de tabelas o algoritmo não apresentou bons resultados.

Segundo o autor, esta diferença de comportamento pode ser explicada pela complexidade inerente a algumas tarefas de aprendizagem em relação aos descritores utilizados.

4.5.2 Software Agregador de Notícias com Classificador Bayesiano

Esta pesquisa foi desenvolvida por Daniel Fagundes da Silva e submetida como Trabalho de Conclusão do Curso de Graduação em Ciências da Computação, Departamento

de Informática e Estatística, Centro Tecnológico da Universidade Federal de Santa Catarina – UFSC – no ano de 2007.

O objetivo do projeto consistiu no desenvolvimento de um mecanismo agregador a fim de facilitar o acompanhamento de *feeds* de notícias personalizando a ordem de apresentação dessas notícias para o usuário, de acordo com o monitoramento da utilização passada da aplicação por parte desse mesmo usuário (SILVA, 2007).

A solução proposta pelo autor foi a aplicação dos conceitos do Teorema de Bayes, por meio do algoritmo *Naive Bayes*, no intuito de predizer o interesse do usuário em determinada notícia. Analisando-se o conteúdo, atribuindo-se um valor correspondente ao possível grau de interesse do usuário naquela notícia (SILVA, 2007).

De acordo com o autor, a inferência bayesiana mostrou ser uma solução viável quando aplicada ao problema de classificação de notícias, pois tem baixo custo de implementação do algoritmo e poucos parâmetros são necessários para a obtenção de um valor de inferência satisfatório. Porém, o autor destaca a importância na escolha das palavras chaves para o processo de classificação, descartando as palavras desnecessárias, a fim de eliminar os ruídos que podem levar a um cálculo errôneo do grau de interesse na notícia.

4.5.3 Suporte à Decisão em um Sistema de Previsão de Doenças do Coração usando *Naive Bayes*

A Srta Subbalakshmi e os professores Ramesh e Chinna Rao do Instituto de Engenharia e Tecnologia de Kakinada, na Índia, em 2011, desenvolveram esta pesquisa no intuito de integrar uma ferramenta de apoio às decisões clínicas dos hospitais, baseadas nos registros armazenados em uma aplicação web contendo um questionário com informações clínicas dos pacientes. Com base nas respostas do usuário, o sistema pode extrair informações a partir dos dados armazenados em um banco de dados com históricos de doenças cardíacas.

O objetivo da pesquisa é melhorar a qualidade das decisões clínicas, reduzindo o esforço para se chegar ao diagnóstico, aumentando o índice de precisão destes e minimizando o elevado custo necessário para as análises clínicas que, Subbalakshmi, Ramesh e Rao, 2011, tradução nossa, é o principal entrave encontrado pelas organizações de saúde (hospitais, centros médicos).

O algoritmo *Naive Bayes* foi usado nessa pesquisa para criar modelos com capacidade de previsão, fornecendo novas maneiras de compreender e explorar os dados.

Os registros usados na pesquisa foram coletados de um banco de dados de doenças do coração da cidade de Cleveland e foram divididos em dois conjuntos, sendo um para treinamento e outro para testes devidamente normalizados. Dados faltantes e registros duplicados foram excluídos e o atributo *Diagnóstico* foi definido com os valores *0* e *1* para os resultados negativo e positivo respectivamente.

Os resultados da pesquisa apontaram que o sistema desenvolvido por meio do algoritmo *Naive Bayes* apresentou um bom desempenho com dados complexos, no que diz respeito à precisão, facilidade de interpretação do modelo implementado e acesso às informações detalhadas (SUBBALAKSHMI; RAMESH; RAO, 2011, tradução nossa).

5 O ALGORITMO NAIVE BAYES NA TAREFA DE CLASSIFICAÇÃO DA *SHELL ORION DATA MINING ENGINE*

A *Shell Orion*, uma ferramenta gratuita de auxílio no processo de descoberta do conhecimento em bases de dados, é mantida pelo Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense.

As funcionalidades desta ferramenta estão em constante atualização por meio da inserção de novos métodos e algoritmos, estes por sua vez, desenvolvidos por alunos da universidade em seus Trabalhos de Conclusão de Curso.

Dentro deste contexto, essa pesquisa tem como finalidade incrementar as funcionalidades da *Shell Orion*, desenvolvendo o algoritmo *Naive bayes* para a tarefa de classificação.

Durante o processo de análise e avaliação do desempenho e qualidade do algoritmo são usadas bases de dados. Nessa pesquisa, optou-se pela utilização de uma base dados na área da saúde, contendo dados clínicos de pacientes com diagnóstico positivo para o câncer de mama.

5.1 BASE DE DADOS

Na avaliação do algoritmo *Naive Bayes*, a base de dados utilizada foi selecionada no UCI *Machine Learning Repository*[10] que disponibiliza informações obtidas a partir dos Hospitais da Universidade de *Wisconsin*, Madison nos Estados Unidos.

Dos 699 registros presentes na base de dados, 16 apresentavam a ausência ou indisponibilidade de valores para alguns de seus atributos e foram eliminados durante o processo de normalização da base, restando um total de 683 registros.

Cada registro possui 10 atributos numéricos que representam métricas para avaliação das características das células analisadas, e um atributo designado como classe que pode assumir os valores 2 para maligno ou 4 para benigno. O rótulo da classe deve ser nominal (TAN; STEINBACH; KUMAR, 2009), por esse motivo, durante a normalização da base o tipo do atributo classe foi convertido de numérico para nominal. A tabela 3 descreve as

[10] UCI Machine Learning Repository é um repositório de dados do Departamento de Informação e Ciências da Computação da Universidade da Califórnia, Irvine, disponibilizado gratuitamente em (http://archive.ics.uci.edu/ml/)

características de cada atributo e suas considerações técnicas extraídas do UCI *Machine Learning Repository.*

Tabela 3 – Base de dados *Breast Cancer Wisconsin (Original) Data Set*.

Atributo	Descrição	Valor
id	Atributo identificador	Número inteiro
clump_thickness	Atributo que faz referência às camadas dos agrupamentos de células	Número inteiro de 1 até 10
uniformity_ofcell_size	Identifica o nível de variação de tamanho das células	Número inteiro de 1 até 10
uniformity_ofcell_shape	Identifica o nível de variação da forma das células	Número inteiro de 1 até 10
marginal_adhesion	Identificador do nível de perda de aderência das células	Número inteiro de 1 até 10
single_epithelial_cell_size	Atributo que identifica o tamanho das células epiteliais	Número inteiro de 1 até 10
bare_nuclei	Atributo que identifica os núcleos não envolvidos pelo citoplasma	Número inteiro de 1 até 10
bland_chromatin	Atributo identificador do nível de textura uniforme no núcleo das células	Número inteiro de 1 até 10
normal_nucleoli	Atributo identificador do nucléolo	Número inteiro de 1 até 10
mitoses	Identificador do nível de multiplicação celular (mitose)	Número inteiro de 1 até 10
class	Atributo classe que assume o valor 2 para “benigno” e 4 para “maligno”	Nominal: 2 ou 4

Fonte: Adaptado de UCI *Machine Learning Repository.*

As amostras foram coletadas periodicamente pelo Dr. William H. Wolberg a partir dos seus casos clínicos, resultando em um agrupamento cronológico dos dados entre janeiro de 1989 e novembro de 1991.

5.2 METODOLOGIA

Foram empregadas durante o desenvolvimento as seguintes etapas metodológicas: levantamento bibliográfico; modelagem por meio do padrão UML; demonstração matemática do algoritmo *Naive Bayes*; implementação e realização de testes; análise de desempenho e validação dos resultados obtidos.

Durante o levantamento bibliográfico levou-se em consideração a fundamentação e o entendimento de todos os temas envolvidos na pesquisa, tais como o processo de descoberta de conhecimento em bases de dados, *data mining*, o algoritmo *Naive Bayes*, índices de avaliação de desempenho e qualidade para algoritmos de classificação de dados.

5.2.2 Modelagem do módulo do algoritmo *Naive Bayes*

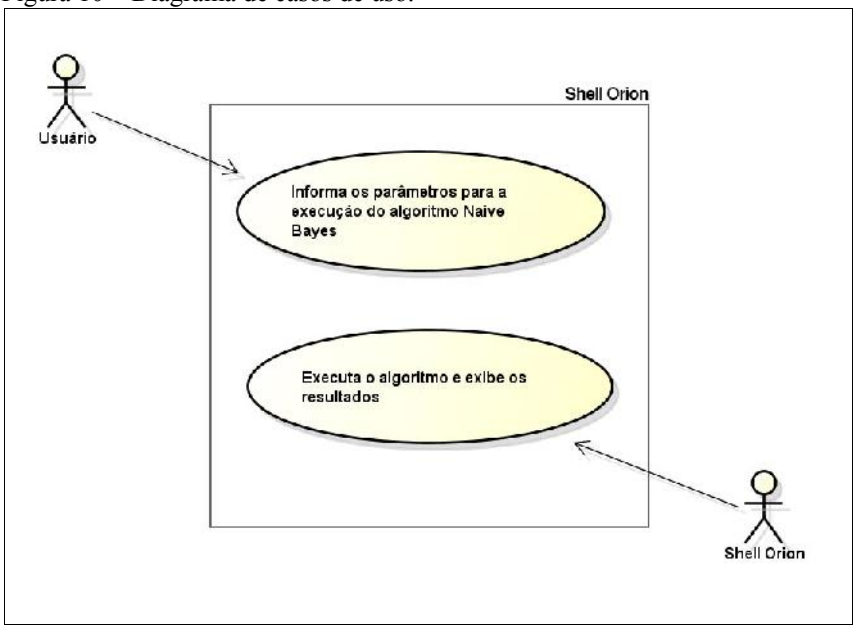
O desenvolvimento teve como primeiro passo a realização da modelagem dos processos executados pelo módulo do algoritmo *Naive Bayes* utilizando o padrão *Unified Modeling Language*[11] (UML). A modelagem UML tem como objetivo facilitar a compreensão dos processos executados pelo algoritmo e as etapas necessárias para a sua execução.

Foram desenvolvidos os diagramas de caso de uso, atividades e sequências com o auxilio da ferramenta *Astah Comunity*[12].

O diagrama de caso de uso é o tipo mais informal e genérico da UML e mostra uma visão geral do comportamento de um sistema. Normalmente é utilizado como ferramenta de auxílio na tarefa de levantamento e análise de requisitos, onde são definidas as necessidades do usuário e o entendimento geral do sistema como mostra a figura 10:

a) **informar os parâmetros de entrada do algoritmo:** o usuário deve informar os parâmetros necessários para a execução do algoritmo;
b) **execução do algoritmo**: os parâmetros de entrada informados pelo usuário são recebidos pela *Shell Orion* que inicia a execução do algoritmo *Naive Bayes* e retorna os resultados.

Figura 10 – Diagrama de casos de uso.

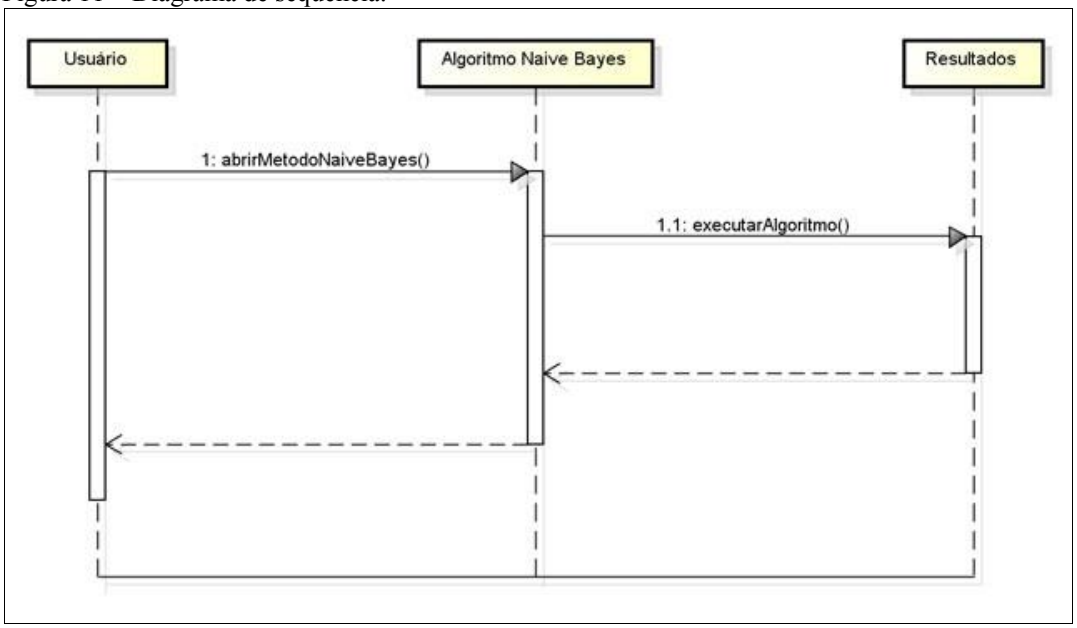


Fonte: Do autor.

[11] *Unified Modeling Language* é uma linguagem visual usada para modelar, especificar, visualizar, documentar e construir sistemas computacionais por meio do paradigma de Orientação a Objetos (FURLAN, 1998).
[12] Disponível para *download* gratuitamente em (http://astah.net/download).

O diagrama de sequência tem como objetivo ilustrar o modo com que as mensagens são trocadas entre os objetos envolvidos, organizando-as sequencialmente e obedecendo a ordem temporal do processo. Com base nos diagramas de casos de uso, identifica os agentes que geram o processo modelado (GUEDES, 2008).

Figura 11 – Diagrama de sequência.

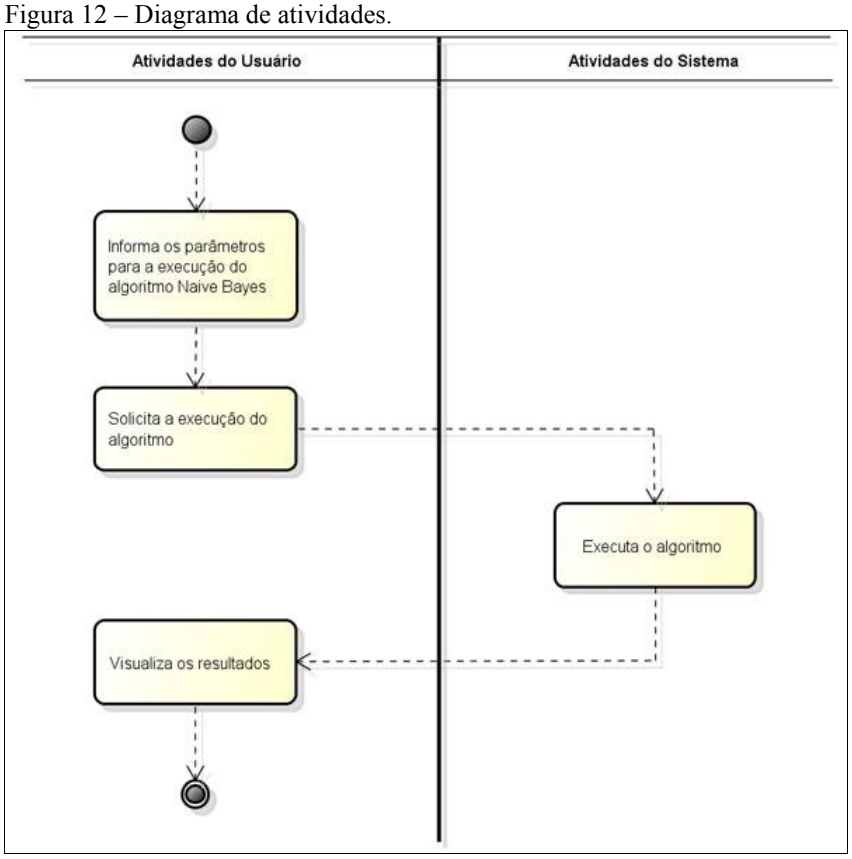


Fonte: Do autor.

A interação do usuário com o algoritmo *Naive Bayes* pode ser observada por meio do diagrama de sequência mostrado na figura 11. Os parâmetros de entrada são informados pelo usuário na tela inicial e após esta etapa a execução do algoritmo é solicitada e os resultados são apresentados ao usuário.

O diagrama de atividades foi modelado posteriormente. Segundo Guedes (2008), esse diagrama tem como objetivo representar os passos percorridos durante um processo para concluir uma determinada atividade.

Figura 12 – Diagrama de atividades.



Fonte: Do autor.

Por meio da figura 12 é possível observar o fluxo de atividades realizadas pelo usuário e pela *Shell Orion* durante o processo, necessárias para a execução do algoritmo *Naive Bayes*.

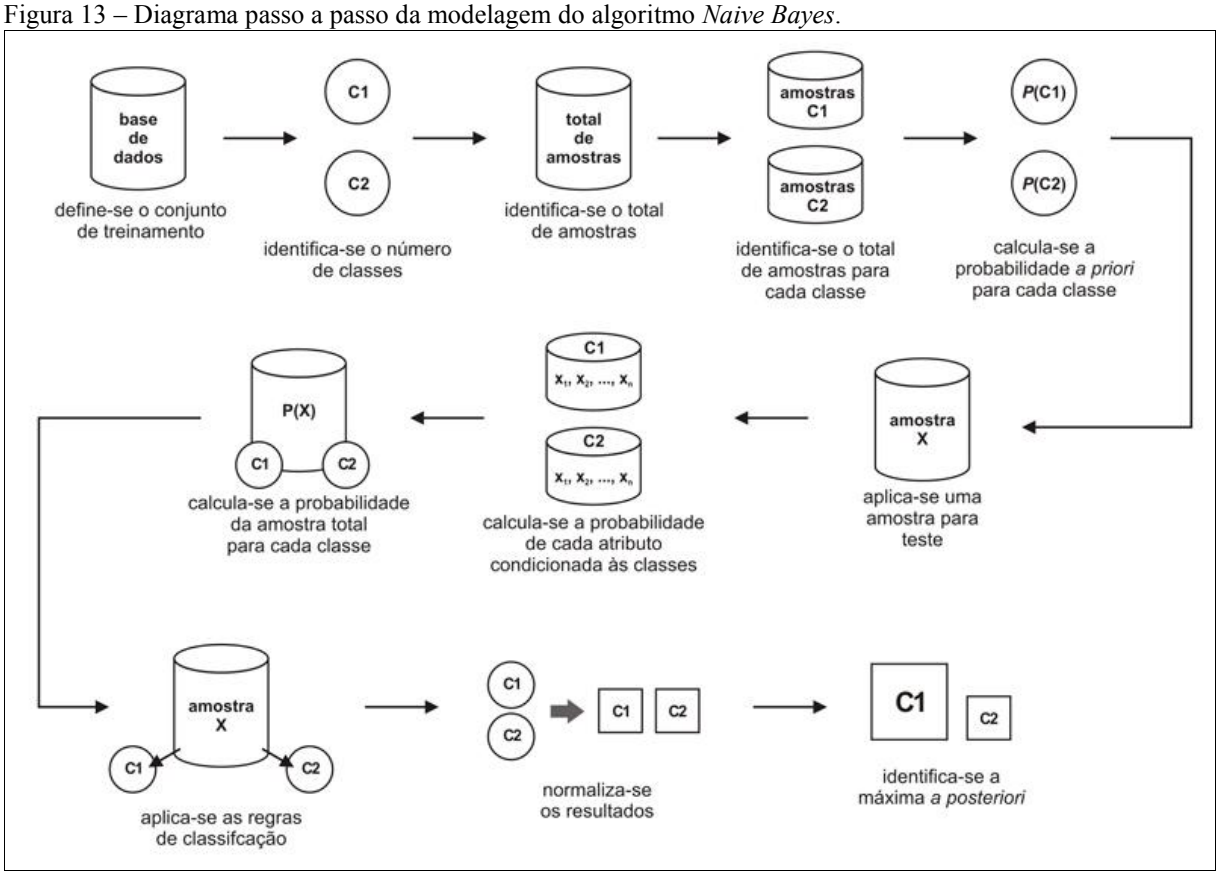
O usuário informa os parâmetros de entrada para o sistema e na sequência solicita a execução do algoritmo. Então a *Shell Orion* executa o algoritmo e retorna os resultados obtidos para o usuário.

Vale ressaltar que a modelagem por meio da UML contribuiu para um melhor entendimento do sistema desenvolvido e auxiliou na visualização e documentação do funcionamento do algoritmo *Naive Bayes* na *Shell Orion Data Mining Engine*.

5.2.3 Demonstração matemática do algoritmo *Naive Bayes*

Esta etapa da pesquisa demonstra o algoritmo *Naive Bayes* por meio da modelagem matemática, permitindo um melhor entendimento do seu funcionamento. A figura 13 ilustra os passos percorridos durante o processo da modelagem.

Figura 13 – Diagrama passo a passo da modelagem do algoritmo *Naive Bayes*.



Fonte: Do autor.

Os conceitos e formalismos do algoritmo *Naive Bayes* tiveram como base o capítulo 3 do livro *Pattern Classification and Scene Analisys* (DUDA; HART, 1973, tradução nossa) e o artigo intitulado *Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier* escrito por Pedro Domingos e Michael Pazzani em 1997, tradução nossa.

Na demonstração matemática do algoritmo *Naive Bayes*, foi usada como conjunto de treinamento *S*, uma base de dados contendo 14 instâncias (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}) com 5 atributos cada uma (*a, b, c, d, e*) como é demonstrado na tabela 4 sendo *e* o atributo classe.

Tabela 4 – Base de dados utilizada para a modelagem do algoritmo

Atributos	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}
a	1	1	2	3	3	3	2	1	1	3	1	2	2	3
b	1	1	1	2	3	3	3	2	3	2	2	2	1	2
c	1	1	1	1	2	2	2	1	2	2	2	1	2	1
d	1	2	1	1	1	2	2	1	1	1	2	2	1	2
e	1	1	2	2	2	1	2	1	2	2	2	2	2	1

Fonte: Do autor.

Para a aplicação do teste de classificação foi usada como conjunto de testes uma amostra desconhecida *X* com a mesma quantidade de atributos definidos na base de dados de treinamento *S* (tabela 5).

Tabela 5 – Amostra desconhecida *X*

Amostra	***a***	***b***	***c***	***d***	***e***
X	1	3	1	2	?

Fonte: Do autor.

Tendo definido o conjunto de treinamento *S* (tabela 4) e o conjunto de testes *X* (tabela 5), o primeiro passo consiste no treinamento de *S* e em seguida a classificação do conjunto de testes.

Para realizar o processo de classificação da amostra *X* a partir do conjunto de treinamento, é necessário maximizar a probabilidade de *X* dado as possíveis classes em $P(X|Ci)\cdot P(Ci)$, para i=1,2, considerando-se que *e* é o atributo classe, onde C_1 corresponde a e=1 e C_2 a e=2.

Os seguintes parâmetros são obtidos por meio do conjunto de treinamento:

a) **número total de amostras:** S=14
b) **número de amostras com classe C_1:** S_1=5, ou seja, das 14 amostras, 5 possuem a classe 1;
c) **número de amostras com classe C_2:** S_2=9, ou seja, das 14 amostras, 9 possuem a classe 2.

Com os parâmetros obtidos, calcula-se a probabilidade *a priori* de cada classe. Esse cálculo consiste na divisão do número de amostras S_i rotuladas como classe C_i pelo total de amostras *S* usando a equação (21):

$$P(C_i) = \frac{S_i}{S},\ i = 1,...,n \tag{21}$$

Tem-se portanto:

$$P(C_1) = P(e = 1) = \frac{5}{14} = 0{,}3571$$

$$P(C_2) = P(e = 2) = \frac{9}{14} = 0{,}6429$$

O próximo passo consiste em calcular $P(X|C_i)$. Tendo em vista que todos os atributos são discretos, é necessário antes calcular as probabilidades de cada atributo condicionadas às classes C_1 e C_2 usando a equação (22):

$$P(x_k \mid C_i) = \frac{S_{ik}}{S_i} \tag{22}$$

Onde S_{ik} representa o número de amostras de treinamento da classe C_i que possuem o valor x_k para cada atributo. O cálculo realizado verifica no conjunto de treinamento a ocorrência do valor de cada atributo x_k da amostra X dado a classe C_i. Esse valor é dividido pelo número de amostras S_i rotuladas como C_i.

Nesta etapa, para evitar-se o problema da frequencia zero, ou seja, caso não exista um valor válido de um atributo x_k pertencente à classe C_i, aplica-se o teorema da aproximação de Laplace equação (18):

$$\frac{n_c + 1}{n + k}$$

Onde n_c é o número de instâncias com o valor de x_k pertencente à classe C_i, n é o número total de instâncias de treinamento com classe C_i e k corresponde à quantidade máxima de valores distintos que x_k pode assumir, ou seja, se um atributo x_k pode assumir apenas os valores a, b e c, então a quantidade máxima de valores distintos que x_k pode assumir é k=3.

A constante 1, adicionada ao numerador da fração, anula a frequência zero e garante um valor válido para o cálculo da probabilidade. Usa-se então a equação (19) ao invés da equação (22) para os cálculos:

$$P(x_1 \mid C_1) = P(a = 1 \mid e = 1) = \frac{3}{5} = \frac{3+1}{5+3} = \frac{4}{8} = 0{,}5000$$

$$P(x_1 \mid C_2) = P(a = 1 \mid e = 2) = \frac{2}{9} = \frac{2+1}{9+3} = \frac{3}{12} = 0{,}2500$$

$$P(x_2 \mid C_1) = P(b = 3 \mid e = 1) = \frac{1}{5} = \frac{1+1}{5+3} = \frac{2}{8} = 0{,}2500$$

$$P(x_2 \mid C_2) = P(b = 3 \mid e = 2) = \frac{3}{9} = \frac{3+1}{9+3} = \frac{4}{12} = 0{,}3333$$

$$P(x_3 \mid C_1) = P(c = 1 \mid e = 1) = \frac{4}{5} = \frac{4+1}{5+2} = \frac{5}{7} = 0{,}7143$$

$$P(x_3 \mid C_2) = P(c = 1 \mid e = 2) = \frac{3}{9} = \frac{3+1}{9+2} = \frac{4}{11} = 0{,}3636$$

$$P(x_4 \mid C_1) = P(d = 2 \mid e = 1) = \frac{3}{5} = \frac{3+1}{5+2} = \frac{4}{7} = 0{,}5714$$

$$P(x_4 \mid C_2) = P(c = 2 \mid e = 2) = \frac{3}{9} = \frac{3+1}{9+2} = \frac{4}{11} = 0{,}3636$$

A aplicação do teorema da aproximação de Laplace nos cálculos realizados garante que o numerador da fração sempre será maior que zero, garantindo valores válidos nos resultados obtidos para a probabilidade.

Na etapa seguinte, obtém-se $P(X|C_i)$. Segundo Coppin (2010), Duda e Hart (1973), tradução nossa e Han e Kamber (2001), tradução nossa para reduzir o custo computacional durante o cálculo de $P(X|C_i)$, o *Naive Bayes* considera a independência condicional de classe, ou seja, o efeito do valor de um determinado atributo sobre uma classe é considerado independente dos valores dos outros atributos. Esta afirmação é dada pela equação (23):

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) \tag{23}$$

Então, os valores resultantes do cálculo da probabilidade na equação 22 para cada atributo dada uma classe C_i são multiplicados entre si para a classe C_1:

$$P(X \mid C_1) = P(X \mid e = 1) = P(x_1 \mid C_1) \times P(x_2 \mid C_1) \times P(x_3 \mid C_1) \times P(x_4 \mid C_1)$$

$$P(X \mid C_1) = P(X \mid e = 1) = 0{,}5000 \times 0{,}2500 \times 0{,}7143 \times 0{,}5714 = 0{,}0510$$

E para a classe C_2:

$$P(X \mid C_2) = P(X \mid e = 1) = P(x_1 \mid C_2) \times P(x_2 \mid C_2) \times P(x_3 \mid C_2) \times P(x_4 \mid C_2)$$

$$P(X \mid C_2) = P(X \mid e = 2) = 0{,}2500 \times 0{,}3333 \times 0{,}3636 \times 0{,}3636 = 0{,}0110$$

Como o teorema de Laplace foi aplicado nos cálculos da amostra X, deve-se aplicá-lo também ao cálculo das probabilidades a priori de cada classe C_i obtido anteriormente por meio da equação (21).

$$P(C_1) = P(e = 1) = \frac{5}{14} = \frac{5+1}{14+2} = \frac{6}{16} = 0{,}3750$$
$$P(C_2) = P(e = 2) = \frac{9}{14} = \frac{9+1}{14+2} = \frac{10}{16} = 0{,}6250$$

Após estes cálculos são aplicadas as regras de classificação considerando-se as probabilidades *a priori* de cada classe C_i multiplicadas pela probabilidade da amostra X dado uma classe C_i.

$$P(X \mid C_i) \times P(C_i) \tag{24}$$

Deste modo, tem-se:

$$P(X \mid C_1) \times P(C_1) = P(X \mid e = 1) \times P(e = 1) = 0{,}0510 \times 0{,}3750 = 0{,}0191$$
$$P(X \mid C_2) \times P(C_2) = P(X \mid e = 2) \times P(e = 2) = 0{,}0110 \times 0{,}675 = 0{,}0074$$

Os resultados obtidos são normalizados e convertidos em percentuais de probabilidade *a posteriori*:

$$P(e = 1) = 0{,}0191/(0{,}0191 + 0{,}0074) = 0{,}7208 \times 100 = 72{,}08\,\%$$
$$P(e = 2) = 0{,}0074/(0{,}0191 + 0{,}0074) = 0{,}2792 \times 100 = 27{,}92\,\%$$

De acordo com os resultados obtidos pelo algoritmo *Naive Bayes*, a amostra X é classificada como e=1 cuja máxima *a posteriori* é 72,08%.

Finalizada esta primeira etapa de cálculos para a classificação de um único registro, o próximo passo consiste em construir um novo conjunto de treinamento S pelo método *holdout* contendo 66% da quantidade de registros da base de dados e o restante é reservado para um novo conjunto de testes X. Após a definição dos novos conjuntos de dados,

repete-se o processo de classificação para cada registro do novo conjunto *X* e valida-se a precisão dos resultados por meio da matriz de confusão[13].

Essa pesquisa implementou o algoritmo *Naive Bayes* considerando que os valores dos atributos da base de dados são discretos na sua totalidade. O cálculo para a distribuição Gaussiana não foi implementado, ficando como sugestão para trabalhos futuros.

5.2.4 Índices Empregados na Validação

O conhecimento gerado pelo processo de *data mining* somente possuirá um grau de relevância satisfatório para o usuário se for considerado potencialmente útil, de fácil compreensão e válido (HAN; KAMBER, 2006, tradução nossa).

O grau de relevância das informações adquiridas no processo de *data mining* pode ser observado por meio das avaliações de desempenho do algoritmo classificador, geralmente representada pela taxa de erros resultante da classificação (WITTEN; FRANK; HALL, 2011, tradução nossa).

A análise de desempenho empregada no classificador desenvolvido nessa pesquisa realizou-se por meio de uma matriz de confusão, ilustrada na tabela 6, resultante do processo de classificação pelo método de testes *holdout*, aplicado no desenvolvimento.

O método *holdout* foi escolhido por permitir a avaliação de resultados obtidos por meio da comparação de vários processos de classificação usando percentuais diferentes, na mesma base de dados, para selecionar os conjuntos de treinamento e teste a cada execução. A vantagem da utilização desse método é que, como as amostras reservadas para o conjunto de testes não desempenharam nenhum papel na formação do classificador, os resultados são considerados mais confiáveis (WITTEN; FRANK; HALL, 2011, tradução nossa).

Tabela 6 – Matriz de confusão

Classe	**Predita C_+**	**Predita C_-**
Verdadeira C_+	***Verdadeiros Positivos***	***Falsos Negativos***
Verdadeira C_-	***Falsos Positivos***	***Verdadeiros Negativos***

Fonte: Adaptado de Goldsmidth e Passos (2005).

[13] A matriz de confusão é uma matriz quadrada que fornece a medida verdadeira do desempenho do modelo de classificação, exibindo o número de classificações preditas *versus* o número de classificações corretas para cada classe de um conjunto de testes (REZENDE, 2005).

A matriz de confusão exibe a quantidade de registros de testes previstos como corretos (linhas) *versus* a quantidade de classificações preditas pelo modelo (colunas). A avaliação de desempenho do modelo é baseada na contagem destes registros (TAN; STEINBACH; KUMAR, 2009).

As células destacadas em cinza na tabela 6 formam a diagonal principal da matriz de confusão que representa o número de classificações corretas para cada classe. Todos os elementos fora dessa diagonal representam os erros na classificação.

Os dados registrados na matriz de confusão possibilitam identificar valores para as seguintes variáveis (GUILLET; HAMILTON, 2007):

a) **verdadeiros positivos (VP):** refere-se à quantidade de registros positivos classificados corretamente como tal;
b) **falsos positivos (FP):** corresponde aos registros negativos classificados incorretamente como positivos;
c) **verdadeiros negativos (VN):** registros negativos classificados corretamente como negativos;
d) **falsos negativos (FN):** diz respeito ao número de registros positivos incorretamente classificados como negativos;

As variáveis identificadas por meio da matriz de confusão permitem o cálculo de índices para análise da qualidade de um modelo classificador. Nesta pesquisa as seguintes métricas de avaliação de desempenho foram consideradas (GUILLET; HAMILTON, 2007; REZENDE, 2005):

a) **sensibilidade:** habilidade do modelo em identificar os registros que pertencem verdadeiramente à classe considerada. Equivale à proporção de verdadeiros positivos e é definida pela equação 25:

$$S = \frac{VP}{(VP + FN)} \tag{25}$$

b) **especificidade:** capacidade do modelo em identificar registros que não pertencem à classe considerada. É calculada considerando-se a quantidade de negativos verdadeiros encontrada pela equação 26:

$$E = \frac{VN}{(VN + FP)} \tag{26}$$

c) **acurácia:** grau de exatidão que o modelo apresenta para identificar as classes por meio da relação entre o valor estimado e o valor real, calculada por:

$$A = \frac{VP + VN}{(VP + VN + FP + FN)} \tag{27}$$

d) **erro:** refere-se à taxa do erro de classificação geral do modelo, dada por:

$$e = 1 - A \tag{28}$$

e) **confiabilidade positiva:** capacidade do classificador em identificar corretamente os verdadeiros positivos. É dada por meio da equação 29:

$$VPP = \frac{VP}{(VP + FP)} \tag{29}$$

f) **índice kappa:** é coeficiente de avaliação da concordância entre dois ou mais métodos de classificação. É baseado no número de respostas concordantes entre os classificadores e calculado pela equação 30:

$$k = \frac{\left(\frac{\sum_{i=1}^{c} x_{ii}}{n} \right) - \left(\frac{\sum_{i=1}^{c} \frac{x_{i+} x_{+i}}{n}}{n} \right)}{1 - \left(\frac{\sum_{i=1}^{c} z_{ii}}{n} \right)} \tag{30}$$

Onde:

a) n é o número total de registros;

b) c é o total de classes;

c) x_{ii} equivale aos elementos da diagonal principal da matriz de confusão;

d) x_{i+} é o valor total da linha i e x_{+i} o valor total da coluna i.

Os elementos para o cálculo do índice Kappa são mostrados na tabela 7.

Tabela 7 – Matriz de confusão – índice Kappa

Classe	Predita C_+	Predita C_-	total
Verdadeira C_+	x_{ii}	x_{ij}	x_{i+}
Verdadeira C_-	x_{ji}	x_{jj}	x_{j+}
total	x_{+i}	x_{+j}	n

Fonte: Do autor.

O índice kappa, classificado por Landis e Koch em 1977, de acordo com a tabela 8, pode variar entre zero e um, nenhuma e total concordância respectivamente (SCHWARTSMANN et al, 2006).

Tabela 8 – Classificação do coeficiente Kappa.

Índice Kappa	Interpretação
0	Pobre
0 – 0,2	Discreta
0,21 – 0,4	Fraca
0,41 – 0,6	Moderada
0,61 – 0,8	Substancial
0,81 – 1	Excelente

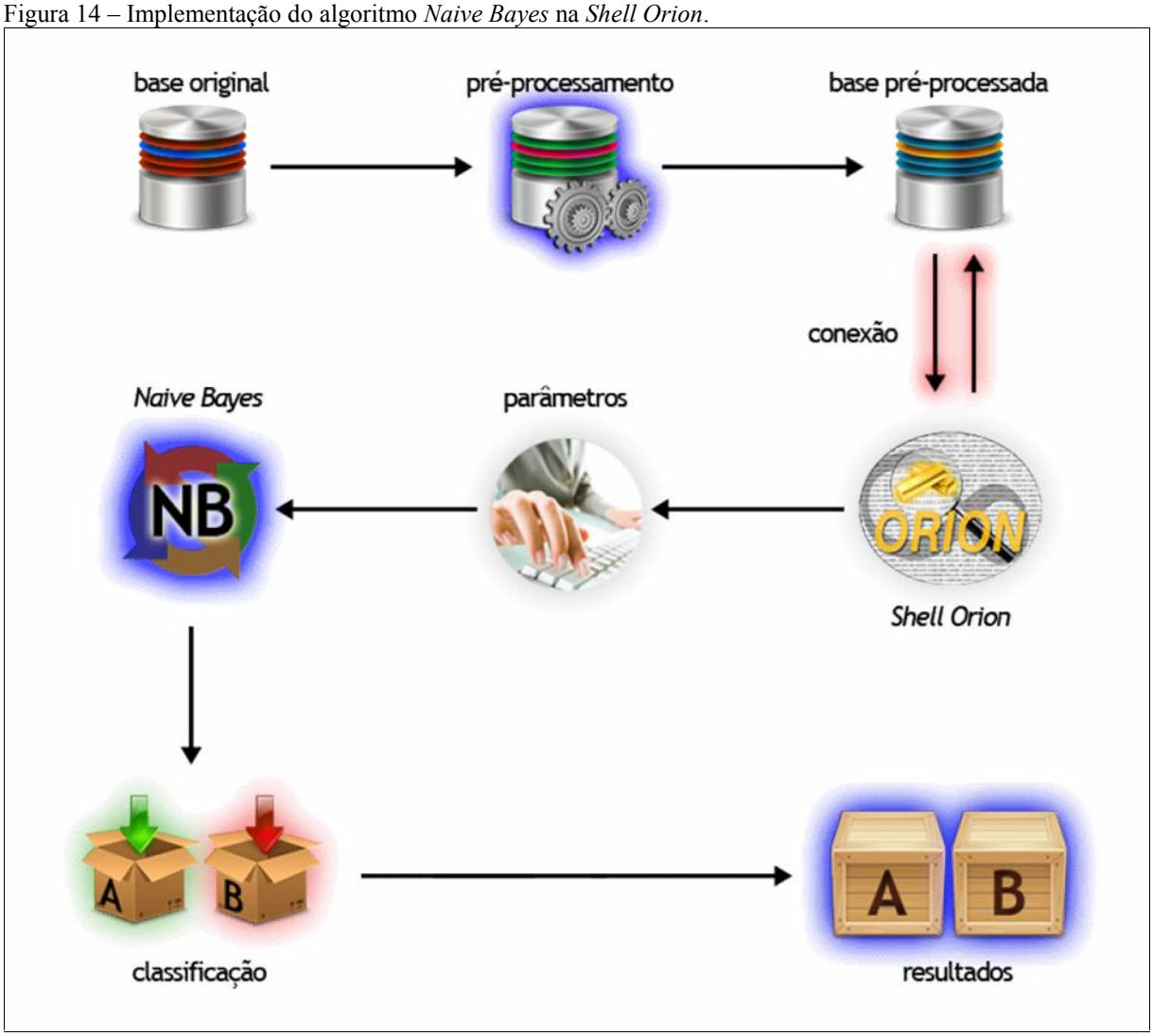
Fonte: Adaptado de Schwartsmann et al (2006).

5.2.5 Implementação e Realização de Testes

O algoritmo *Naive Bayes* foi desenvolvido no módulo de classificação da *Shell Orion Data Mining Engine*, por meio da linguagem de programação Java e do ambiente de programação integrado *NetBeans* 7.1.1.

A *Shell Orion* permite a conexão com diversos Sistemas Gerenciadores de Bancos de Dados (SGBD) desde que o SGBD escolhido disponha de um *driver* JDBC para estabelecer a conexão. Na implementação e realização dos testes do algoritmo *Naive Bayes*, o SGBD escolhido foi o *MySQL* por ser muito difundido, portável e gratuito. O processo de implementação e obtenção dos resultados é ilustrado na figura 14.

Figura 14 – Implementação do algoritmo *Naive Bayes* na *Shell Orion*.



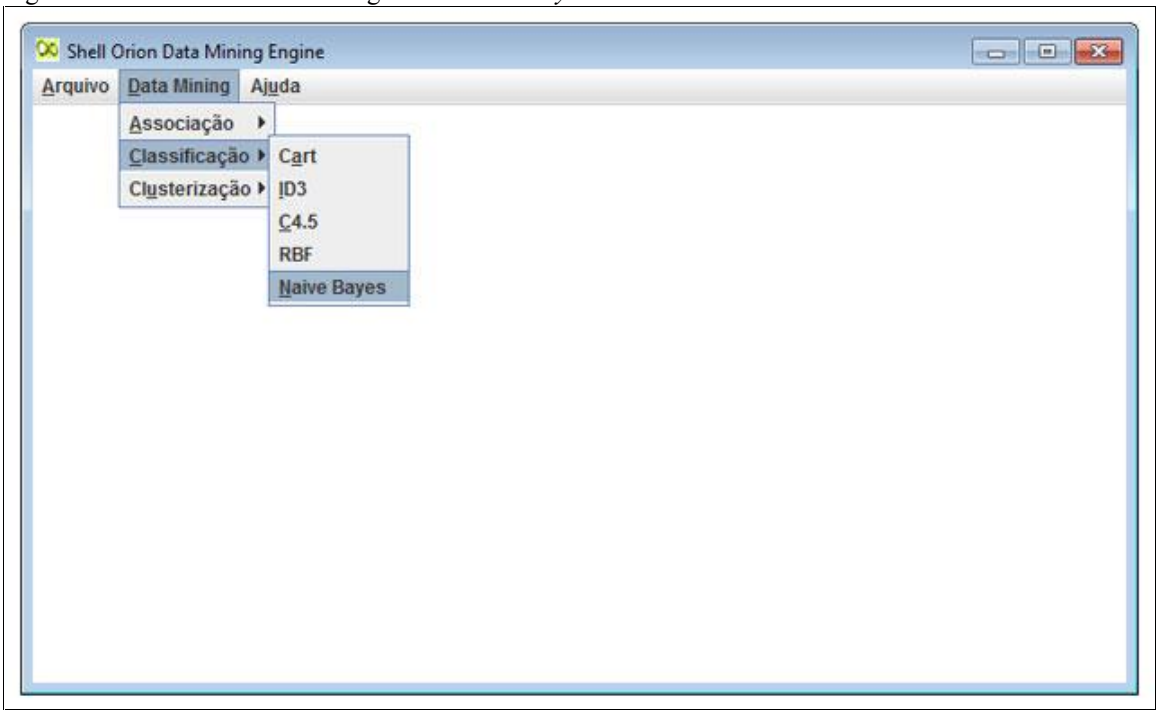
Fonte: Do autor.

Definido o SGBD, foram realizados testes com a base de dados selecionada para avaliar os resultados obtidos pelo algoritmo desenvolvido. A base de dados escolhida para a realização dos testes contém dados clínicos de pacientes com diagnóstico positivo para o câncer de mama.

Deve-se, portanto, realizar a inserção dos dados, devidamente pré-processados e normalizados, por meio de comandos SQL e após, estabelecer a conexão da *Shell Orion* com o SGBD no menu *Arquivo*, submenu *Conectar*.

Uma vez estabelecida a conexão com o SGBD, é possível acessar o algoritmo *Naive Bayes* por meio do menu *Data Mining*, submenu *Classificação*, selecionando-se o algoritmo *Naive Bayes* (figura 15).

Figura 15 – Acesso ao menu do algoritmo *Naive Bayes*.



Fonte: Do autor.

A tarefa de classificação por meio do algoritmo *Naive Bayes* exige que o usuário informe alguns parâmetros e atributos de entrada (figura 16). No quadrante esquerdo estão os parâmetros relacionados às opções de teste que o algoritmo deverá realizar e no quadrante direito os atributos de entrada a serem selecionados para que o processo de classificação seja realizado.

Os parâmetros que definem as opções de teste são:

a) **usar o conjunto de treinamento:** esta opção define que o conjunto de treinamento e o de teste sejam os mesmos. O algoritmo executa o processo de treinamento com todos os registros da base de dados e em seguida classifica os mesmos registros com base nos valores adquiridos durante o processo de treinamento;

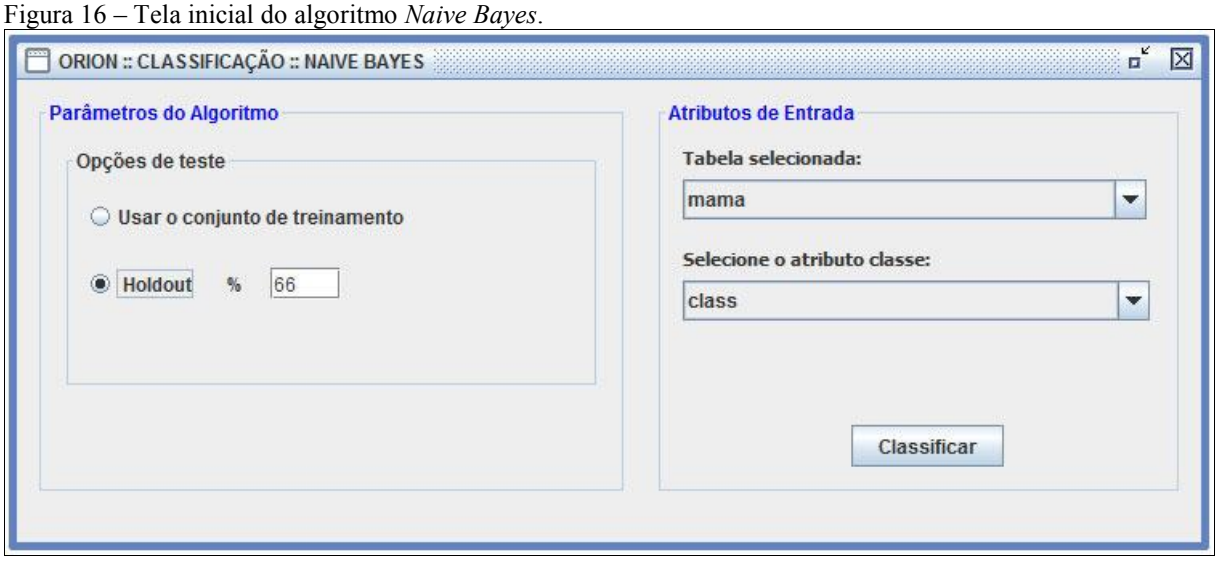
b) ***holdout*:** ao selecionar o método *holdout*, o usuário poderá definir o percentual de amostras na base de dados que deve ser reservado para o treinamento do algoritmo. O restante das amostras automaticamente é definido como teste.

Os atributos de entrada são iniciados com valores preestabelecidos, contudo o usuário tem a opção de alterar alguns deles:

a) **tabela selecionada:** atributo cujo valor é recuperado no momento da conexão com o banco de dados. Em geral o usuário não tem opção de alterá-lo;

b) **atributo classe:** por padrão o último atributo da tabela é sempre selecionado como classe. Porém, o usuário tem a opção de alterar o seu valor caso haja necessidade.

Figura 16 – Tela inicial do algoritmo *Naive Bayes*.



Fonte: Do autor.

Os valores informados para os parâmetros e atributos de entrada interferem diretamente no resultado apresentado pelo algoritmo, o que possibilita, em uma mesma base de dados, obter diferentes resultados de acordo com os valores informados pelo usuário.

A primeira opção nos parâmetros do algoritmo que define o mesmo conjunto de dados tanto para treinamento quanto para testes, pode apresentar resultados não muito confiáveis apesar de apresentarem altos índices de validação. Essa baixa credibilidade nos resultados se dá porque a partir do conjunto de treinamento se obtém o desempenho passado e o que se deseja avaliar é o desempenho futuro do algoritmo, e esse resultado é alcançado por meio de novos dados para teste (WITTEN; FRANK; HALL, 2011, tradução nossa).

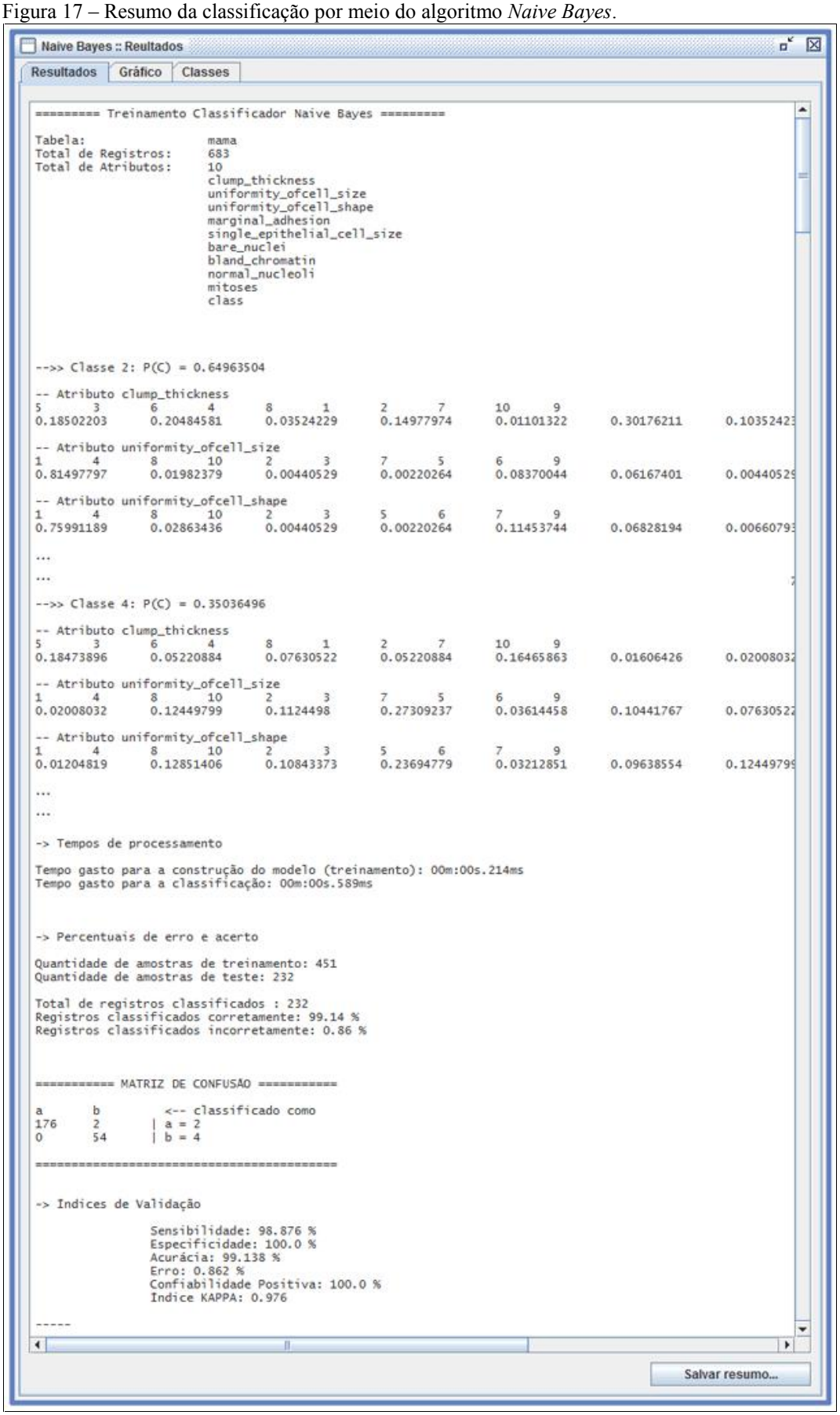
A opção *holdout* permite resultados mais confiáveis, uma vez que as amostras reservadas para o conjunto de testes não fazem parte do conjunto de treinamento (WITTEN; FRANK; HALL, 2011, tradução nossa). Porém, não é recomendado reservar um percentual menor que 50% para treinamento, pois quanto menor é o conjunto de treinamento, maior a variância do modelo, o que influencia diretamente nos resultados obtidos (TAN; STEINBACH; KUMAR, 2009).

É importante salientar que na *Shell Orion*, tem-se por *default*, a definição de 66% no parâmetro *holdout* a fim de manter a proporção de 2/3 de registros para treinamento e 1/3 para testes, mas caso o usuário deseje, ele pode alterar esta proporção.

Os resultados obtidos pelo algoritmo na *Shell Orion*, podem ser analisados por meio de resumo, árvore ou gráfico. A figura 17 mostra um relatório textual gerado pelo algoritmo contendo informações sobre os atributos de entrada, os tempos de treinamento e classificação, e os índices de validação do modelo.

A *Shell Orion* também disponibiliza a função de exportação dos resultados para um arquivo no formato de texto por meio do botão *Salvar resumo*.

Figura 17 – Resumo da classificação por meio do algoritmo *Naive Bayes*.

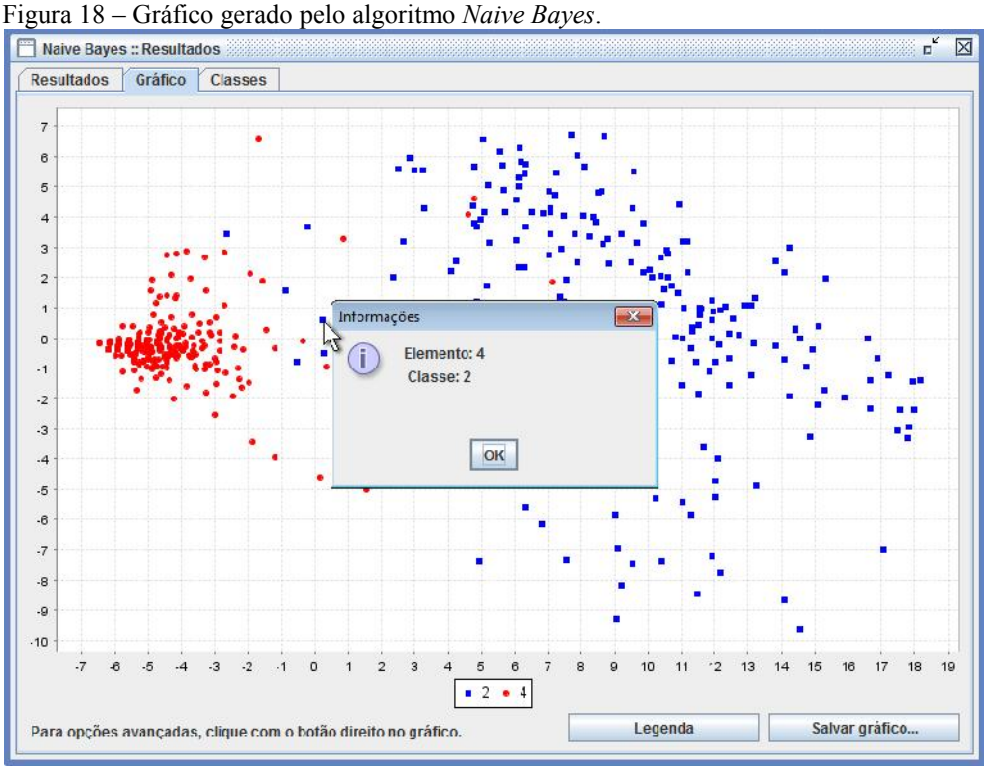


Fonte: Do autor.

O resumo exibe a identificação de duas classes com 178 e 54 elementos respectivamente que equivalem aos 44% dos registros reservados para teste pelo método *holdout*. Além disso, outros indicadores resultantes do processamento como tempo de execução, taxa de erro médio, parâmetros e atributos de entrada e os índices de validação do modelo, são apresentados.

De forma gráfica, a distribuição das amostras por classe pode ser visualizada na figura 18. A representação das classes identificadas se dá por meio do método *Principal Component Analysis*[14] (PCA). Esse método possibilita a projeção gráfica dos dados, transformando em uma matriz de duas dimensões qualquer base de dados de *n* dimensões por meio de sucessivas decomposições nos dados (MARTENS; NAES, 1993, tradução nossa).

Figura 18 – Gráfico gerado pelo algoritmo *Naive Bayes*.



Fonte: Do autor.

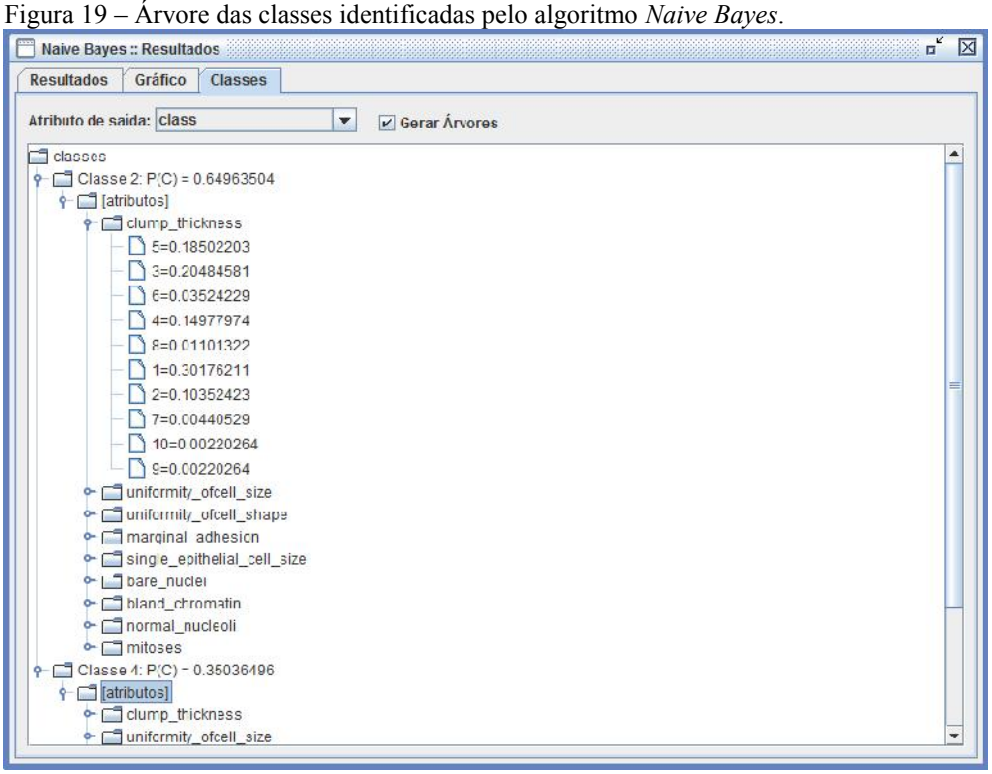
O gráfico mostrado na figura 18 exibe a distribuição dos dados originais da base do câncer de mama em suas respectivas classes diferenciadas pelas cores vermelha e azul. Ao ser clicado, o elemento exibe o seu *id* e o rótulo da classe a qual pertence.

Uma estrutura em árvore, ilustrada na figura 19, é disponibilizada exibindo as informações de probabilidade *a priori* de cada classe identificada bem como as probabilidades

[14] Guia sobre as diferentes implementações do PCA está disponível em: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

dos valores de cada atributo condicionadas à classe analisada. Essa estrutura permite a análise individual dos valores distintos assumidos por cada atributo e das suas probabilidades condicionadas a cada classe, resultando em uma visualização mais detalhada.

Figura 19 – Árvore das classes identificadas pelo algoritmo *Naive Bayes*.



Fonte: Do autor.

Ao término da etapa de implementação, foram realizados testes a fim de validar os resultados obtidos pelo algoritmo e avaliar os tempos de processamento. A discussão sobre esses resultados é apresentada a seguir.

5.3 RESULTADOS OBTIDOS

Os testes realizados após a etapa de implementação do algoritmo *Naive Bayes* na *Shell Orion*, abrangem a análise dos resultados obtidos durante o processo de classificação por meio dos índices de validação, a análise do desempenho por meio da precisão dos resultados observados, o tempo de execução do algoritmo e a comparação do modelo implementado com o desenvolvido na ferramenta *Weka*.

Durante a realização dos testes com o algoritmo classificador *Naive Bayes*, foi utilizado um microcomputador com sistema operacional Windows 7 Professional, processador Intel Dual Core 2.3GHz e 4GB de memória RAM.

5.3.1 Classes identificadas pelo algoritmo *Naive Bayes*

Na avaliação da capacidade do algoritmo *Naive Bayes*, implementado na *Shell Orion*, em identificar corretamente a classe para cada registro, foi utilizada uma base de dados na área da saúde, contendo dados clínicos de pacientes com diagnóstico positivo para o câncer de mama (683 registros).

O algoritmo foi executado usando os seguintes parâmetros e atributos de entrada:

a) **método *holdout*:** conforme indicado pela literatura, optou-se por utilizar um percentual de 66% da base para treinamento e o restante para testes, o que corresponde respectivamente a 2/3 e 1/3 da base da dados. Essa métrica garante que os conjuntos de treinamento e testes sejam diferentes entre si;

b) **atributo de entrada:** atributo *class* selecionado como a classe a ser identificada.

O teste foi realizado com o objetivo de identificar quais registros pertencem as classes 2 (benigno) e 4 (maligno). Optou-se por repetir a execução dos testes por três vezes usando percentuais diferentes para os conjuntos de treinamento e teste. As tabelas 9,10 e 11 descrevem os resultados obtidos pelo classificador *Naive Bayes* nos três casos.

Tabela 9 – Classes identificadas na base de câncer de mama (*holdout* 50%).

Classe	Quantidade de elementos	Porcentagem de elementos	Classe	Porcentagem de acertos
2 (benigno)	260	76,24%	2 (253 ocorrências) 4 (7 ocorrências)	97,30%
4 (maligno)	81	23,76%	4 (81 ocorrências)	100%

Fonte: Do autor.

Tabela 10 – Classes identificadas na base de câncer de mama (*holdout* 66%).

Classe	Quantidade de elementos	Porcentagem de elementos	Classe	Porcentagem de acertos
2 (benigno)	178	76,72%	2 (176 ocorrências) 4 (2 ocorrências)	98,88%
4 (maligno)	54	23,28%	4 (54 ocorrências)	100%

Fonte: Do autor.

Tabela 11 – Classes identificadas na base de câncer de mama (*holdout* 88%).

Classe	Quantidade de elementos	Porcentagem de elementos	Classe	Porcentagem de acertos
2 (benigno)	68	82,93%	2 (66 ocorrências) 4 (2 ocorrências)	97,06%
4 (maligno)	14	17,07%	4 (14 ocorrências)	100%

Fonte: Do autor.

Na observação dos resultados apresentados na coluna *Classe* da tabela 9, é possível perceber que o algoritmo implementado classificou incorretamente apenas 7 em 260 registros para a classe 2, ou seja, dos 260 registros reais com classe 2, apenas 7 foram classificados como classe 4.

Os resultados nas tabelas 10 e 11, com 66% e 88% dos registros respectivamente, mostram um melhor desempenho do algoritmo que identificou apenas dois registros em classes incorretas nas duas amostragens, ou seja, dos 178 registros reais com classe 2 na tabela 10, apenas 2 foram classificados como classe 4 e dos 68 registros reais com classe 2 na tabela 11, apenas 2 foram classificados como classe 4, sendo que, nas três amostragens, nenhum registro real com classe 4 foi classificado incorretamente.

A análise dos resultados apresentados mostra um desempenho satisfatório do algoritmo que, mesmo na pior taxa de acerto (tabela 11), apresentou valores próximos de 100%.

Por outro lado, o melhor desempenho, observado (tabela 10), confirma o que a literatura defende acerca da proporção de registros que deve ser reservada para treinamento e testes utilizando-se o método *holdout*. Confirmou-se também a influência direta dos parâmetros de entrada sobre os resultados alcançados pelo algoritmo *Naive Bayes*.

Após a conclusão do correto funcionamento do algoritmo classificador, realizou-se uma análise com o objetivo de avaliar o desempenho do classificador.

A fim de obter-se as métricas para a avaliação de desempenho do algoritmo, foi construída uma matriz de confusão a partir dos resultados gerados pelo teste que apresentou um maior percentual de acerto (tabela 10), conforme mostra a tabela 12.

Tabela 12 – Matriz de confusão para a classificação na base do câncer de mama

Classe	Predita – Classe 2	Predita – Classe 4	Total
Verdadeira – Classe 2	**176**	**2**	*178*
Verdadeira – Classe 4	**0**	**54**	*54*
Total	*176*	*56*	*232*

Fonte: Do autor.

Os valores observados na matriz de confusão (tabela 12) permitem a extração das seguintes variáveis:

a) **verdadeiros positivos:**

$$VP = 176$$

b) **falsos positivos:**

$$FP = 0$$

c) **verdadeiros negativos:**

$$VN = 54$$

d) **falsos negativos:**

$$FN = 2$$

Os valores, para cada variável dos itens a, b, c e d, obtidos a partir da matriz de confusão são demonstrados na tabela 13.

Tabela 13 – Relação de falsos negativos e verdadeiros negativos e falsos positivos e verdadeiros positivos

Classe	FN	VN	VP	FP
Classe 2	2	0	176	0
Classe 4	0	54	0	0
Total	2	54	176	0

Fonte: Do autor.

Com os valores das variáveis recuperadas da matriz de confusão, calcularam-se os índices de validação do modelo:

a) **sensibilidade:**

$$S = \frac{VP}{(VP + FN)}$$

$$S = \frac{176}{(176 + 2)}$$

$$S = 0{,}9888$$

b) **especificidade:**

$$E = \frac{VN}{(VN + FP)}$$

$$E = \frac{54}{(54 + 0)}$$

$$E = 1{,}0000$$

c) **acurácia:**

$$A = \frac{(VP + VN)}{(VP + VN + FP + FN)}$$

$$A = \frac{(176 + 54)}{(176 + 54 + 0 + 2)} = \frac{230}{232}$$

$$A = 0{,}9914$$

d) **erro:**

$$e = 1 - A$$

$$e = 1 - 0{,}9914$$

$$e = 0{,}0086$$

e) **confiabilidade positiva:**

$$VPP = \frac{VP}{(VP + FP)}$$

$$VPP = \frac{176}{(176 + 0)}$$

$$VPP = 1{,}0000$$

f) **índice kappa:**

$$k = \frac{\left(\frac{x_{11}+x_{22}}{232}\right) - \left(\frac{\frac{x_{31}\cdot x_{13}}{232} + \frac{x_{32}\cdot x2_{23}}{232}}{232}\right)}{1 - \left(\frac{\frac{x_{31}\cdot x_{13}}{232} + \frac{x_{32}\cdot x_{23}}{232}}{232}\right)}$$

$$k = \frac{\left(\frac{176+54}{232}\right) - \left(\frac{\frac{176\cdot 178}{232} + \frac{56\cdot 54}{232}}{232}\right)}{1 - \left(\frac{\frac{176\cdot 178}{232} + \frac{56\cdot 54}{232}}{232}\right)}$$

$$k = \frac{0{,}9914 - 0{,}6382}{1 - 0{,}6382}$$

$$k = 0{,}9762$$

Os índices de validação que tiveram seus cálculos demonstrados anteriormente, na tabela 14 são reunidos a fim de proporcionar um resumo deles.

Tabela 14 – Resumo dos índices de validação

Índice	Valor
sensibilidade	98,88%
especificidade	100%
acurácia	99,14%
erro	0,86%
confiabilidade	100%
kappa	0,976

Fonte: Do autor.

A análise dos valores obtidos nos índices de validação mostra que o desempenho do módulo desenvolvido pode ser considerado satisfatório por apresentar índices próximos de 100% e taxa de erro próxima de zero.

Para o coeficiente de concordância Kappa o classificador *Naive Bayes* apresentou índices próximo de um, considerado um resultado significativo, visto que de acordo com Colgaton (1991) o índice Kappa é uma métrica bastante interessante, pois constitui-se em um coeficiente satisfatório para avaliar a precisão de um modelo de classificação, considerando a matriz de confusão como um todo no seu cálculo, inclusive os elementos de fora da diagonal principal.

Com a etapa de análise de desempenho finalizada e verificado o correto funcionamento do modelo de classificação implementado, realizaram-se alguns testes, a fim de verificar o tempo de processamento do algoritmo quanto submetido a diferentes parâmetros.

5.3.2 Tempos de processamento do algoritmo *Naive Bayes*

O processo de avaliação do tempo de processamento do modelo foi feito em duas etapas. Primeiramente considerou-se a comparação dos tempos de processamento durante o processo de treinamento e de classificação do modelo desenvolvido. Posteriormente foram comparados os tempos de processamento do algoritmo *Naive Bayes* com os da ferramenta Weka[15] na sua versão 3.6.7, durante o processo de classificação.

A *Waikato Environment for Knowledge Analysis* (Weka) é uma ferramenta distribuída sob os termos da *General Public License* (GNU)[16]. Ela foi desenvolvida na Universidade de *Waikato*, na Nova Zelândia, sendo frequentemente utilizada em pesquisas na área de *data mining*. Implementa diversos algoritmos de *data mining* em Java, permitindo a sua execução em diversas plataformas (WITTEN; FRANK; HALL, 2011, tradução nossa).

Em cada etapa da avaliação foi considerada a quantidade de registros em diferentes tamanhos, para isso a base de dados original foi replicada gradativamente. Essa replicação deu-se por meio do crescimento geométrico[17] resultando em diferentes cargas que foram testadas posteriormente nos ensaios.

O padrão de crescimento geométrico é definido pela equação 30.

---

[15] Disponibilizada gratuitamente em (http://www.cs.waikato.ac.nz/~ml/weka/).
[16] General Public License. Informações em (http://www.gnu.org).
[17] Uma progressão geométrica (seqüência geométrica) é uma seqüência de termos onde existe um termo inicial a, e cada termo subseqüente é obtido pelo produto do anterior por um valor constante *r* chamado de razão (GERSTING, 1995).

$$C_k = N \times 2^{k-1} \tag{30}$$

Onde:

a) C_k: representa o tamanho da carga a ser testada, determinado pela quantidade de registros;

b) N: é o número de registros da base de dados original;

c) k: define o número de iterações.

A base de dados utilizada possuía originalmente 683 registros. O tamanho das cargas de dados a partir da quantidade inicial foi definido conforme mostra a tabela 15.

Tabela 15 – Definição dos tamanhos das cargas de dados

Iteração	Quantidade de registros	Quantidade de atributos
1	683	3
2	1366	3
3	2732	3
4	5464	3
5	10928	3
6	21856	3
7	43712	3
8	87424	3

Fonte: Do autor.

A análise dos dados observados durante as duas etapas dos testes de processamento foi realizada com o auxílio do pacote estatístico *Statistical Package for Social Sciences*[18] (SPSS).

Durante os testes utilizaram-se apenas três atributos, pois não foi avaliada a precisão obtida pelo processo de classificação, somente os tempos de processamento foram considerados.

5.3.2.1 Tempos de processamento Treinamento *versus* Classificação

A primeira etapa de testes foi realizada coletando os tempos de processamento durante os processos de treinamento e classificação do modelo desenvolvido na *Shell Orion*.

Os tempos observados são mostrados na tabela 16.

[18] O SPSS para Windows oferece diversas possibilidades de cálculo estatístico e análise exploratória de dados. O download de uma versão de testes pode ser obtida em: http://www-01.ibm.com/software/analytics/spss/.

Tabela 16 – Tempos de processamento nos processos de treinamento e classificação.

Quantidade de registros	Treinamento	Classificação
683	00m: 00s. 054ms	00m: 00s. 047ms
1366	00m: 00s. 076ms	00m: 00s. 233ms
2732	00m: 00s. 103ms	00m: 00s. 816ms
5464	00m: 00s. 152ms	00m: 03s. 236ms
10928	00m: 00s. 269ms	00m: 12s. 477ms
21856	00m: 00s. 516ms	00m: 54s. 051ms
43712	00m: 00s. 959ms	04m: 11s. 126ms
87424	00m: 02s. 018ms	22m: 40s. 877ms

Fonte: Do autor.

Observou-se que o tempo de processamento usado para o treinamento é significativamente menor que o tempo de classificação, isso resulta em um baixo custo computacional mesmo quando a carga de registros atinge um percentual de 12800% do valor inicial. A carga elevada de registros impacta de forma mais significativa apenas no desempenho do algoritmo durante a classificação, no que se refere ao tempo de processamento e custo computacional.

Para as análises de processamento inicialmente utilizou-se, por meio do pacote estatístico SPSS, o teste de *Shapiro-Wilk*. Esse teste avalia se a distribuição dos valores de uma determinada amostra é significativamente diferente de uma distribuição normal. Se o valor é significativo, indica um desvio da normalidade. O critério para avaliação do resultado do teste é o *p-value* que tem um nível de significância preestabelecido em 0,05, ou seja, a distribuição gaussiana é confirmada se $p>0,05$, caso contrário o resultado indica uma distribuição significativamente diferente de uma distribuição gaussiana (FIELD, 2009).

Após a realização do teste de *Shapiro-Wilk* (figura 20), confirmou-se que os tempos dentro dos grupos treinamento e classificação não distribuíam-se de forma gaussiana, pois obtiveram os índices $p = 0,007$ e $p = 0,000025$ para treinamento e classificação respectivamente.

Figura 20 – Teste de *Shapiro-Wilk* nos tempos de processamento.

Testes de Normalidade

Naive Bayes		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estatística	gl*	Significância	Estatística	gl*	Significância
tempo	TREINAMENTO	,268	8	,093	,742	8	,007000
	CLASSIFICAÇÃO	,380	8	,001	,532	8	,000025

* Grau de Liberdade

↑
$p < 0,05$

Fonte: Do autor.

A alternativa utilizada para obter a normalização dos dados foi a transformação logarítmica dos tempos, ou seja, calculou-se o log natural para cada tempo em milissegundos (ms). Por exemplo, para o tempo de 54 ms, calculou-se ln(54) = 3,988984047. Essa transformação repetiu-se o demais tempos, originando uma nova variável com base neperiana que após a aplicação do teste de *Shapiro-Wilk*, obteve distribuição gaussiana em ambos os grupos $p = 0,724$ e $p = 0,968$ (figura 21).

Figura 21 – Teste de *Shapiro-Wilk* após a transformação logarítmica dos tempos.

Testes de Normalidade

Naive Bayes		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estatística	gl*	Significância	Estatística	gl*	Significância
LogTempo	TREINAMENTO	,156	8	,200	,951	8	,724
	CLASSIFICAÇÃO	,106	8	,200	,981	8	,968

* Grau de Liberdade

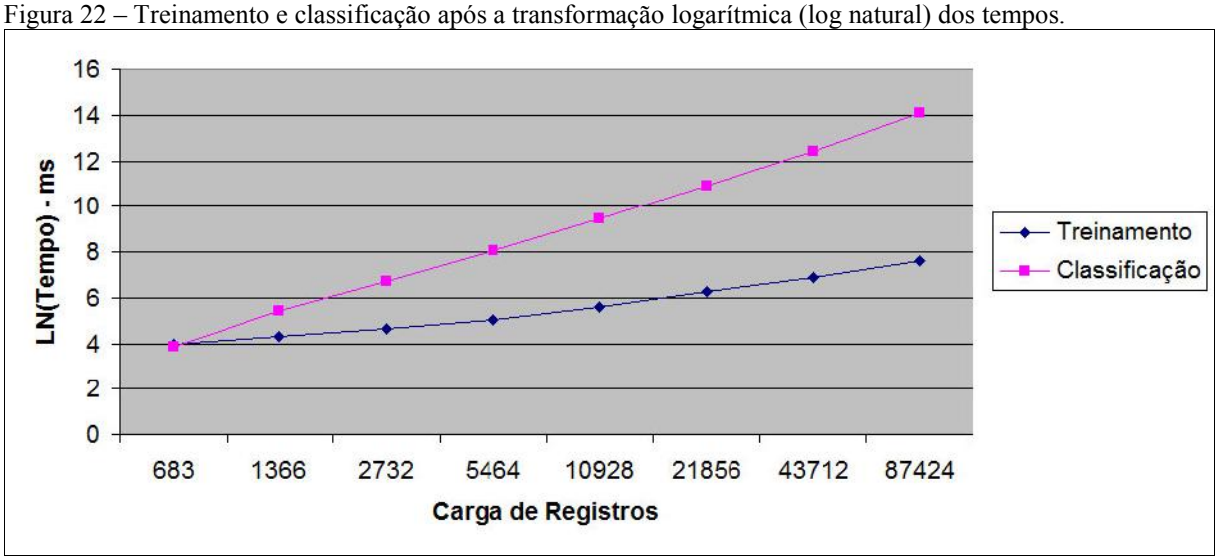
↑
$p > 0,05$

Fonte: Do autor.

As análises que se seguem foram realizadas com o logaritmo dos tempos, medidos em milissegundos (ms), sendo descritos os valores de p primeiramente relativo ao grupo treinamento e em seguida em relação ao grupo de classificação.

A diferença nos tempos de processamento após a transformação logarítmica dos tempos entre os processos de treinamento e classificação é mostrada de forma gráfica na figura 22.

Figura 22 – Treinamento e classificação após a transformação logarítmica (log natural) dos tempos.



Fonte: Do autor.

A fim de verificar a homogeneidade das variâncias observadas em cada grupo foi realizado o teste *F* de *Levene*. Esse teste verifica a hipótese de que a variância observada nos grupos é a mesma, ou seja, que a diferença entre as variâncias é zero (FIELD, 2009).

A realização do teste *F* de *Levene* rejeitou a hipótese nula (p = 0,017), revelando a existência de heterogeneidade entre as variâncias o que levou a utilização do teste *t* para variâncias heterogêneas.

O teste *t* de *Student* foi usado a fim de comparar as médias dos tempos nos processos de treinamento e classificação (figura 23). Esse teste é usado para comparar dois conjuntos de dados com valores quantitativos considerando seus valores médios. Baseia-se em uma distribuição normal de valores (BARBETTA; REIS; BORNIA, 2010; FIELD, 2009).

Figura 23 – Teste t de Student nos tempos de treinamento e classificação.

Teste *t* de *Student*

Naive Bayes		N	Média	Desvio Padrão	Erro Padrão da Média
LogTempo	TREINAMENTO	8	5,5369	1,28161	,45312
	CLASSIFICAÇÃO	8	8,8718	3,52393	1,24590

Fonte: Do autor.

A realização do teste *t* de *Student* para amostras independentes revelou que a média do logaritmo do tempo do processo de treinamento, 5,54 ± 1,28 ms, foi significativamente mais rápida quando comparada com a média do logaritmo do tempo do

processo de classificação, que resultou em 8,87 ± 3,52 ms (p = 0,034), conforme observa-se na tabela 17.

Tabela 17 – Resultados do teste t de *Student* - treinamento e classificação.

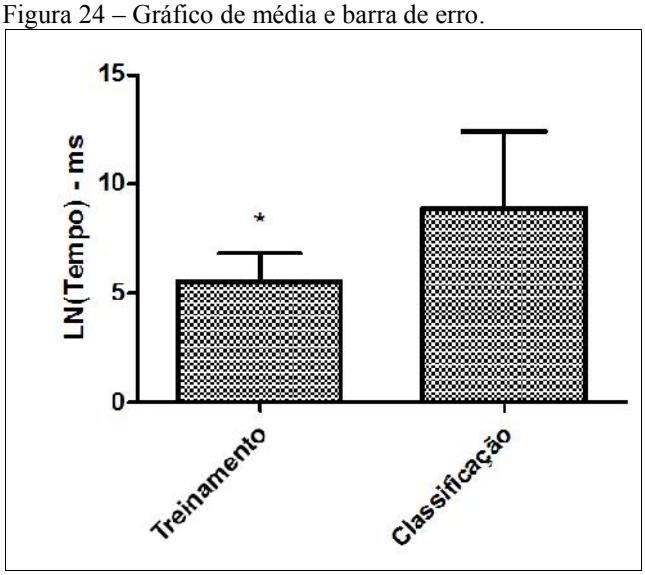
	n	**ln(tempo)**	**Valor-*p***
Treinamento	8	5,54 ± 1,28 ms	0,034
Classificação	8	8,87 ± 3,52 ms	

Fonte: Do autor.

O valor-p (0,034) indica que a diferença entre o tempo gasto durante o treinamento e o gasto pela classificação foi significativa, rejeitando a hipótese nula (H_0), portanto, infere-se que a variação não ocorre ao acaso, mas em função da tarefa executada.

A figura 24 mostra de forma mais ilustrativa a diferença significativa entre as médias dos tempos de processamento obtidas pelo teste *t* de *Student*.

Figura 24 – Gráfico de média e barra de erro.



Fonte: Do autor.

O gráfico de média e barra de erro (figura 24) revela as médias dos tempos de execução representadas pelos retângulos preenchidos e o desvio padrão para o treinamento e classificação representados pelas hastes em forma de *T*. O asterisco sobre a haste do retângulo que representa o treinamento indica que a média do tempo foi significativamente menor comparada com a média do tempo gasto para a classificação.

5.3.2.2 Tempos de processamento *Shell Orion versus* Weka

Durante a segunda etapa de testes realizou-se a coleta dos tempos de processamento dos processos de treinamento e classificação do algoritmo *Naive Bayes* na *Shell Orion* e dos tempos de processamento usados pela ferramenta Weka para executar as mesmas tarefas nas mesmas condições (tabela 18).

Tabela 18 – Tempos de processamento da *Shell Orion* e da ferramenta Weka.

Quantidade de registros	***Shell Orion***	**Weka**
683	00m: 00s. 047ms	00m:00s.000ms
1366	00m: 00s. 233ms	00m:00s.001ms
2732	00m: 00s. 816ms	00m:00s.005ms
5464	00m: 03s. 236ms	00m:00s.010ms
10928	00m: 12s. 477ms	00m:00s.019ms
21856	00m: 54s. 051ms	00m:00s.028ms
43712	04m: 11s. 126ms	00m:00s.038ms
87424	22m: 40s. 877ms	00m:00s.159ms

Fonte: Do autor.

Os tempos mostrados na tabela 18 registram uma diferença significativamente maior entre o tempo consumido pelos processos de treinamento e classificação do algoritmo *Naive Bayes* na *Shell Orion* e o tempo levado pelos mesmos processos na ferramenta Weka.

Os valores de tempo na Weka apresentam uma variação discreta nas oito iterações com diferentes cargas de registros. A *Shell Orion* mantém essa variação discreta apenas nas cargas iniciais. As variações só começam a ficar significativas quando a carga de registros atinge um percentual de 800% do valor inicial. Essa característica reflete o impacto causado no desempenho do algoritmo pelo aumento geométrico da carga de registros.

Após finalizados os testes, utilizou-se o teste de *Shapiro-Wilk* para verificar se os tempos dentro dos grupos *Shell Orion* e Weka distribuíam-se de forma gaussiana (figura 25), o que não se confirmou para ambos os grupos por apresentar valores de significância menor que 0,05 (p = 0,000026 e p = 0,001).

Figura 25 – Teste de *Shapiro-Wilk* nos tempos de processamento da *Shell Orion* e da Weka.

Testes de Normalidade

Ferramenta		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estatística	gl*	Significância	Estatística	gl*	Significância
tempo	SHELL ORION	,379	8	,001	,533	8	,000026
	WEKA	,334	8	,009	,647	8	,001000

* Grau de Liberdade

↑
$p < 0,05$

Fonte: Do autor.

Utilizou-se mais uma vez, para obter a normalização dos dados, a transformação logarítmica dos tempos, originando uma nova variável com base neperiana que, após a aplicação do teste de *Shapiro-Wilk* (figura 26), obteve distribuição gaussiana em ambos os grupos ($p = 0,906$ e $p = 0,980$).

Figura 26 – Teste de *Shapiro-Wilk* após a transformação logarítmica dos tempos.

Testes de Normalidade

Ferramenta		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Estatística	gl*	Significância	Estatística	gl*	Significância
LogTempo	SHELL ORION	,121	8	,200	,971	8	,906
	WEKA	,137	7	,200	,985	7	,980

* Grau de Liberdade

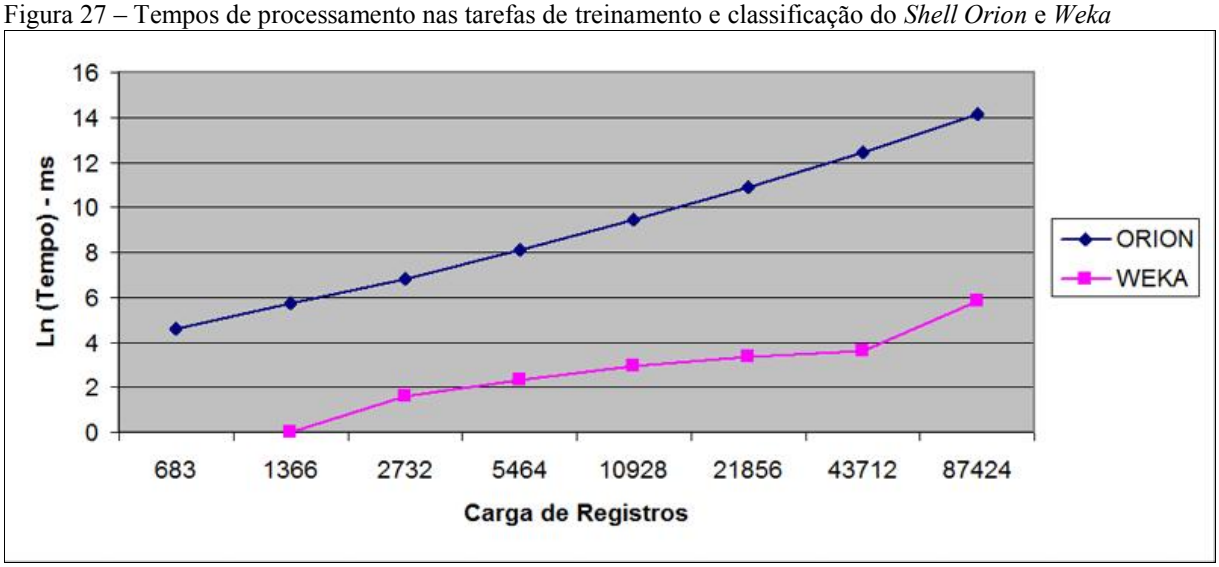
↑
$p > 0,05$

Fonte: Do autor.

As análises realizaram-se considerando o logaritmo dos tempos, medidos em milissegundos (ms), sendo descritos os valores de *p* primeiramente relativo ao grupo *Shell Orion* e em seguida em relação ao grupo de Weka.

A projeção gráfica que ilustra a diferença nos tempos de processamento entre o modelo desenvolvido e a ferramenta Weka é mostrada na figura 27.

Figura 27 – Tempos de processamento nas tarefas de treinamento e classificação do *Shell Orion* e *Weka*



Fonte: Do autor.

Do ponto de vista estatístico pode-se observar que o gráfico que representa a *Shell Orion* (figura 27) exibe uma certa linearidade, sendo que o último intervalo mostra um ângulo de inclinação menor com relação ao gráfico que representa a Weka. Essa diferença sugere que, futuramente aplicando-se mais testes e aumentando a carga de registros, os gráficos tendem a se cruzar em algum momento.

Para verificar a homogeneidade das variâncias observadas em cada grupo foi realizado o teste *F* de *Levene*, que aceitou a hipótese nula (p = 0,054), revelando a existência de homogeneidade entre as variâncias o que levou a utilização do teste t para variâncias homogêneas.

O teste *t* de *Student* foi usado a fim de comparar as médias dos tempos nos processos de treinamento e classificação na *Shell Orion* e Weka (figura 28).

Figura 28 – Teste t de Student na analise dos tempos da *Shell Orio*n e Weka.

Teste *t* de *Student*

Ferramenta		N	Média	Desvio Padrão	Erro Padrão da Média
LogTempo	SHELL ORION	8	9,0278	3,32343	1,17501
	WEKA	7	2,6993	1,61180	,60920

Fonte: Do autor.

A aplicação do teste *t* de *Student* para amostras independentes revelou que a média do logaritmo do tempo da ferramenta Weka, 2,70 ± 1,61 ms, foi significativamente

mais rápida em comparação com a média do logaritmo do tempo da tarefa de classificação do algoritmo *Naive Bayes*, resultando em 9,03 ± 3,32 ms (p = 0,001), conforme apresentados na tabela 19.

Tabela 19 – Resultados do teste t de *Student* para a *Shell Orion* e a Weka

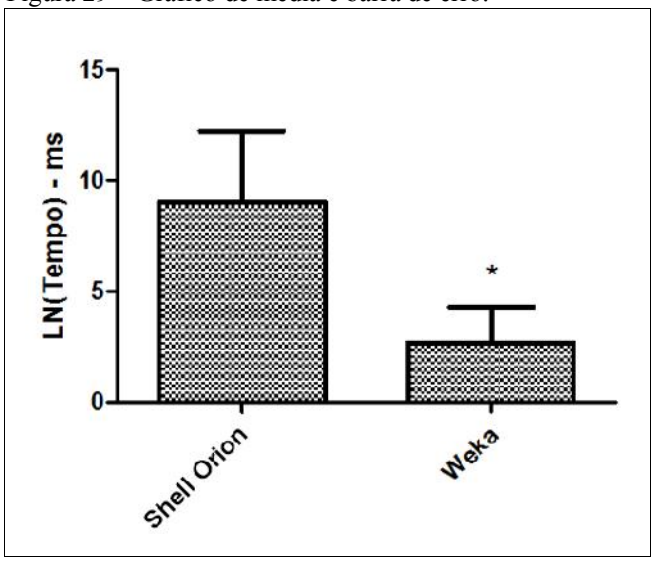
	n	**ln(tempo)**	**Valor-*p***
Shell Orion	8	9,03 ± 3,32 ms	0,001
Weka	7	2,70 ± 1,61 ms	

Fonte: Do autor.

O valor-p (0,001) indica que a diferença entre o tempo gasto durante o treinamento e classificação pela *Shell Orion* e o tempo gasto pela Weka foi significativa, rejeitando a hipótese nula (H_0), portanto, infere-se que a variação não ocorre ao acaso.

A diferença entre as médias dos tempos de processamento obtidas pelo teste t de *Student* é ilustrada na figura 29.

Figura 29 – Gráfico de média e barra de erro.



Fonte: Do autor.

O gráfico de média e barra de erro (figura 29) revela as médias dos tempos de execução representadas pelos retângulos preenchidos e o desvio padrão para o treinamento e classificação na *Shell Orion* e Weka representados pelas hastes em forma de *T*. O asterisco sobre a haste do retângulo que representa a Weka indica que a média do tempo foi significativamente menor comparada com a média do tempo gasto pela *Shell Orion*.

Os testes de processamento aplicados tiveram como objetivo avaliar o tempo consumido pelo modelo durante o processo de treinamento e classificação em quantidades diferentes de registros.

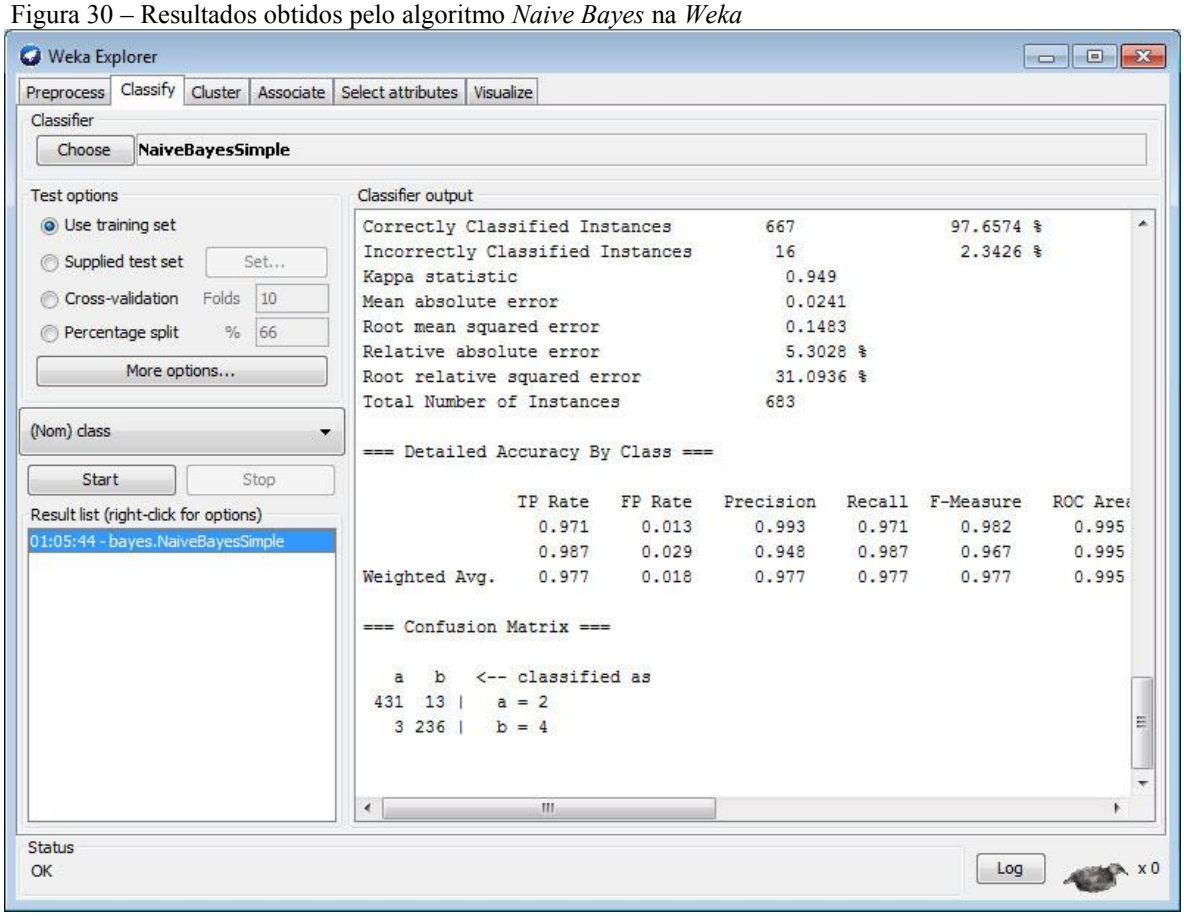
5.3.3 Comparação dos índices de validação com a ferramenta Weka 3.6.7

Em uma terceira etapa de testes analisou-se a qualidade dos resultados gerados, para isso a ferramenta Weka também foi usada com a finalidade de comparar os resultados obtidos no processo de classificação com os da *Shell Orion*. Apenas os índices de validação foram considerados nessa avaliação, portando os tempos de processamento foram ignorados.

Os testes iniciaram considerando a mesma base de dados, com 683 registros, usada anteriormente. Foram aplicados os dois métodos de testes disponíveis no algoritmo *Naive Bayes* implementado na *Shell Orion*.

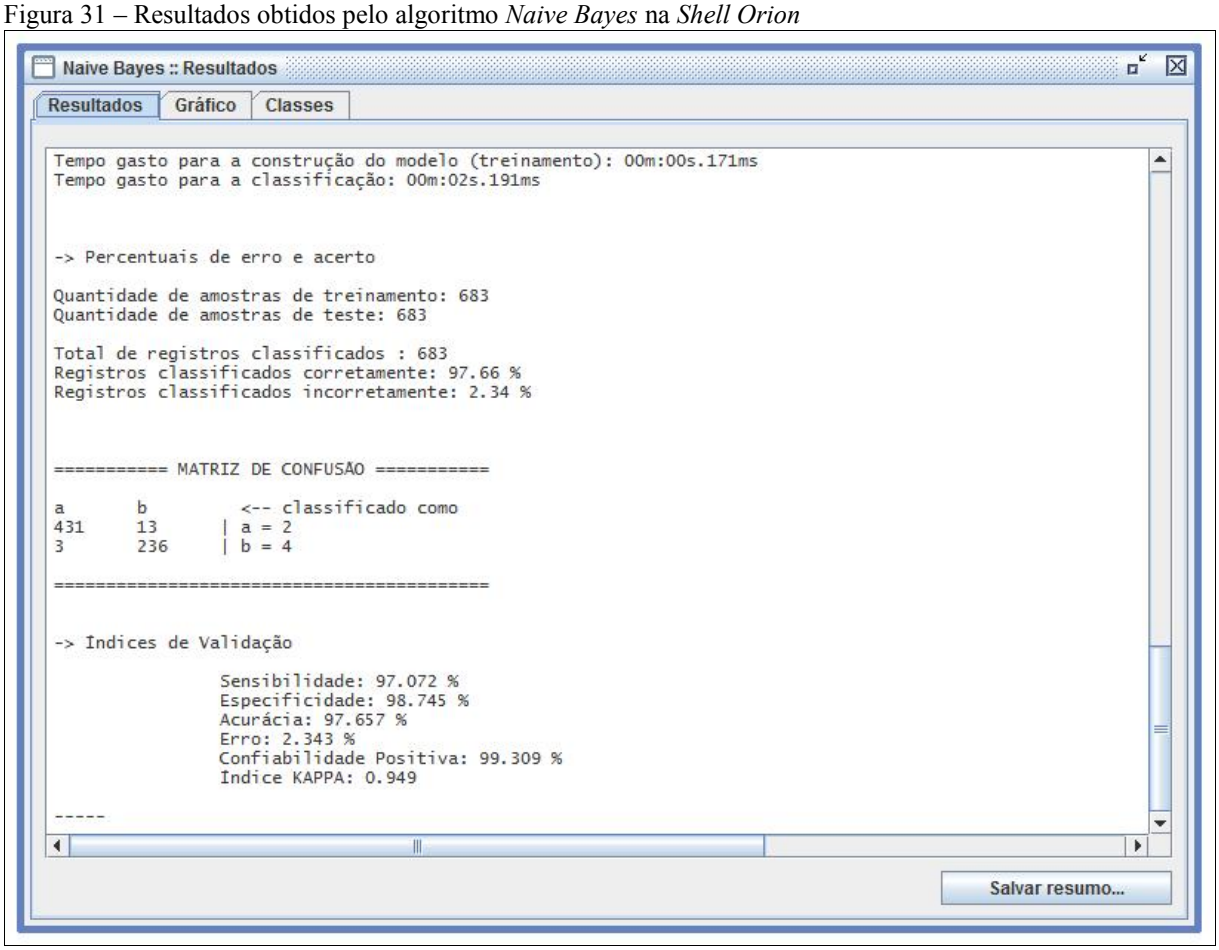
Na primeira análise optou-se por usar como teste o mesmo conjunto de treinamento construído com todos os 683 registros da base de dados. Nas figuras 30 e 31, tem-se respectivamente os resultados apresentados pela ferramenta Weka e pela *Shell Orion*.

Figura 30 – Resultados obtidos pelo algoritmo *Naive Bayes* na *Weka*



Fonte: Do autor.

Figura 31 – Resultados obtidos pelo algoritmo *Naive Bayes* na *Shell Orion*

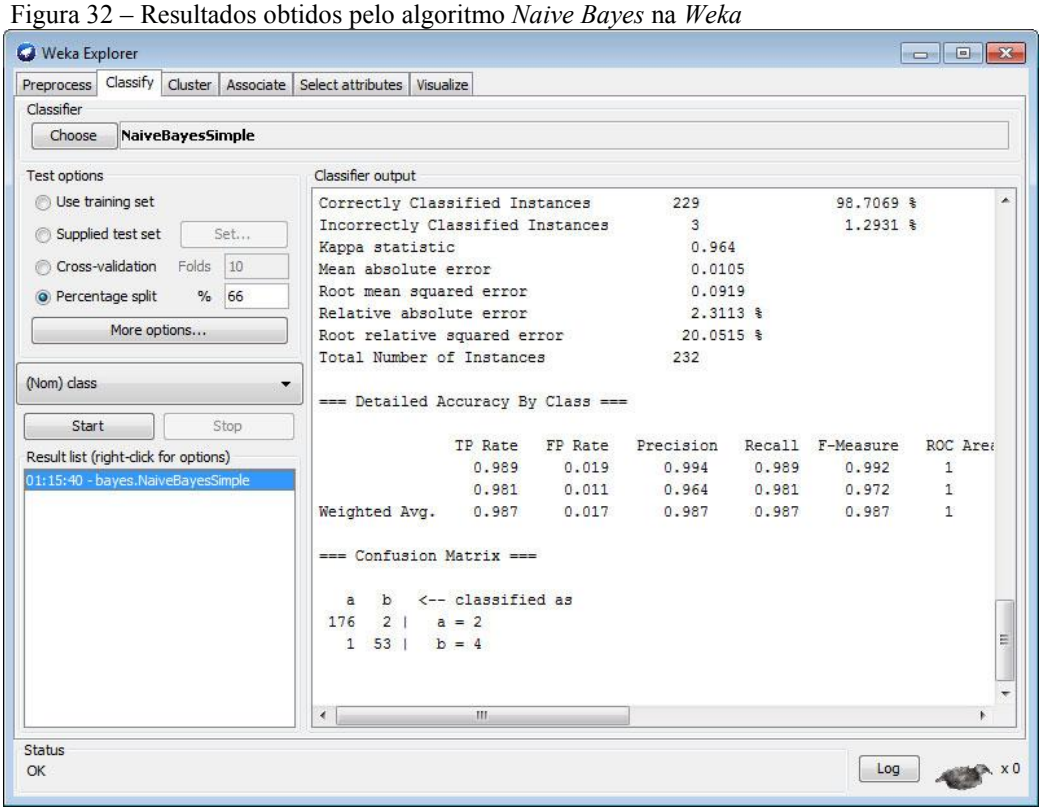


Fonte: Do autor.

Os resultados mostram que o algoritmo *Naive Bayes*, implementado na *Shell Orion*, obteve índices com valores idênticos aos apresentados pela ferramenta Weka na análise desta base de dados referente ao câncer de mama. O percentual de registros classificados correta e incorretamente foram os mesmos (97.66 %), bem como o índice Kappa que apresentou um valor de concordância (0,949) classificado como excelente.

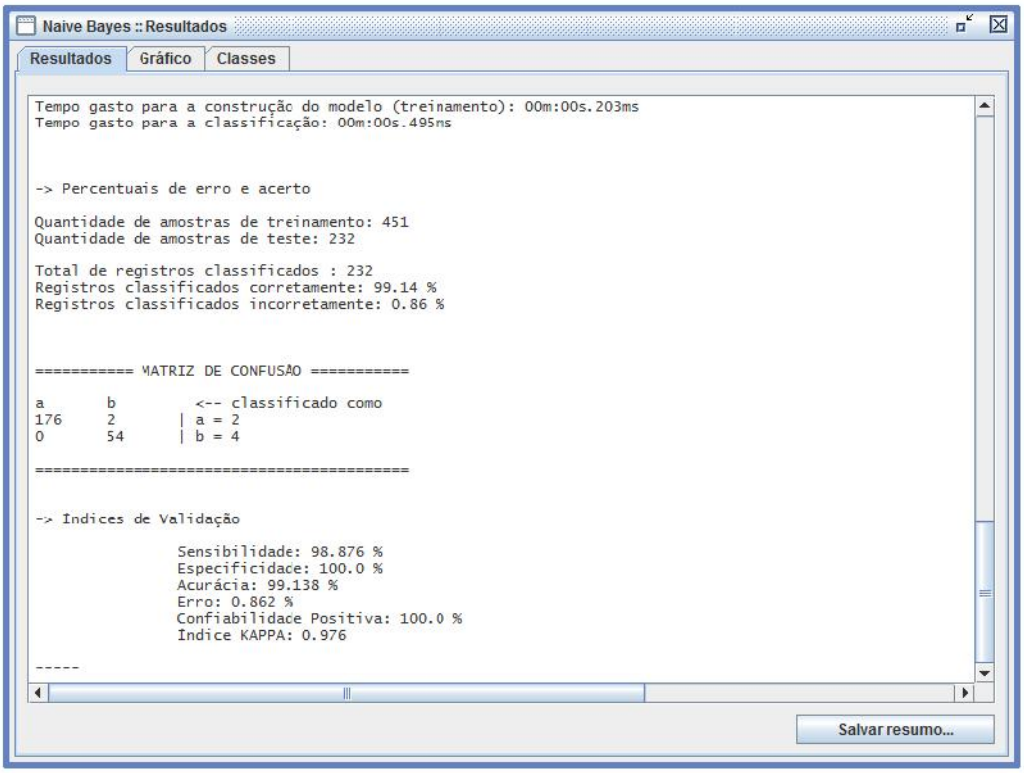
Finalizado o teste usando a base completa para treinamento e validação, realizou-se um novo teste por meio do método *holdout*, reservando 66% da base de dados para treinamento e o restante para testes. O método *holdout* disponível na Weka é conhecido como *Percentage Split*. Nas figuras 32 e 33, tem-se respectivamente os resultados apresentados pela ferramenta Weka e pela *Shell Orion* por meio do método *holdout*.

Figura 32 – Resultados obtidos pelo algoritmo *Naive Bayes* na *Weka*



Fonte: Do autor.

Figura 33 – Resultados obtidos pelo algoritmo *Naive Bayes* na *Shell Orion*



Fonte: Do autor.

Os resultados apresentados mostram que o algoritmo *Naive Bayes* implementado na *Shell Orion*, obteve alguns índices ligeiramente melhores no que diz respeito a percentuais de acerto (*Shell Orion* – 99,14%; Weka – 98,71%) e índice Kappa (*Shell Orion* – 0,976; Weka – 0,964). Conclui-se que ambas as ferramentas têm comportamento semelhante com relação à qualidade dos resultados do processo de classificação.

Analisando os testes de tempo de processamento, a Weka apresentou tempos melhores quando comparados aos tempos do modelo implementado na *Shell Orion*. Contudo, considerando-se os testes de validação, o algoritmo *Naive Bayes* na *Shell Orion* apresentou resultados satisfatórios confirmando o correto funcionamento do módulo desenvolvido para classificação.

6 CONCLUSÃO

O progresso constante nas tecnologias de armazenamento digital tem ocasionado um aumento gradativo na geração de informação e na criação de repositórios de dados cada vez maiores. O processo de *data mining* tem desempenhado um papel importante no auxílio à tarefa de extrair novos conhecimentos e auxiliar nas tomadas de decisão.

Essa pesquisa fundamentou-se nos principais conceitos relacionados à tarefa de classificação, por meio do algoritmo *Naive Bayes* que baseia-se na estatística e probabilidade bayesiana possibilitando, por meio da classificação e análise das probabilidades, a identificação da classe a que determinada amostra de dados pertence.

Dentre as diversas dificuldades encontradas durante a realização da pesquisa destacam-se a escolha dos métodos adequados para a validação dos resultados gerados pelo algoritmo, a escolha de uma base de dados adequada para a realização da tarefa de classificação pelo algoritmo *Naive Bayes*, o entendimento de alguns formalismos matemáticos que compõem o desenvolvimento e o próprio fluxo de funcionamento do algoritmo e a carência de bibliografia explicativa no que diz respeito à demonstração dos cálculos que elucidem o funcionamento completo do algoritmo a ser implementado. Contudo, estas dificuldades foram superadas com o auxílio da modelagem matemática do algoritmo.

Mesmo considerando as dificuldades encontradas ao longo da pesquisa, os objetivos foram alcançados, indicando o correto funcionamento do algoritmo proposto, visto que os resultados avaliados apresentaram bons índices de validação relacionados à precisão e confiabilidade dos valores obtidos.

Avaliou-se também o tempo de processamento consumido durante o processo de treinamento e classificação pelo algoritmo. Observa-se, pelos resultados obtidos, que o tempo de processamento gasto para o treinamento do algoritmo é significativamente menor do que o tempo gasto para a classificação. Essa informação confirma os argumentos encontrados nas bibliografias de que o algoritmo apresenta um menor tempo de aprendizado preditivo em relação a outros classificadores.

A comparação com a ferramenta Weka mostrou que os resultados obtidos pela pesquisa com relação aos tempos de processamento não foram muito bons, porém os índices de validação empregados na avaliação dos resultados da classificação foram ligeiramente melhores que os da ferramenta Weka. A *Shell Orion* obteve 230 classificações corretas enquanto a Weka obteve 229. O índice de especificidade da *Orion* foi de 100% enquanto o índice obtido pela ferramenta Weka foi de 98,1%. A acurácia na *Shell Orion* atingiu o índice

de 99,14% com uma taxa de erro de apenas 0,86% enquanto que na Weka a acurácia foi de 98,7% com uma taxa de erro de 1,3%. O índice Kappa, que é o coeficiente de avaliação da concordância entre os resultados, na *Shell Orio*n alcançou o valor de 0,967 enquanto a Weka apresentou um índice de 0,964. Os resultados obtidos pela *Shell Orion* confirmaram o correto funcionamento do algoritmo.

Com base no conhecimento adquirido pela pesquisa, são descritas a seguir algumas sugestões de trabalhos futuros no intuito de dar continuidade ao desenvolvimento das funcionalidades da *Shell Orion Data Mining Engine*:

a) estudar e implementar no algoritmo outros métodos de testes tais como o *Cross-Validation* e o *Bootstrap*;
b) desenvolver outras variações do algoritmo *Naive Bayes* como por exemplo *Flexible Naive Bayes* a fim de comparar os resultados entre eles, vantagens e desvantagens de cada um;
c) estudar e implementar métodos para classificação de dados com atributos contínuos por meio da distribuição Gaussiana, visto que o modelo implementado classifica dados com valores categóricos e numéricos discretos.
d) desenvolver outros métodos associados ao *Naive Bayes* visando reduzir o tempo de classificação em grandes conjuntos de dados;
e) utilizar os resultados obtidos pelo algoritmo em outras tarefas da *data mining*.

REFERÊNCIAS

BARBETTA, Pedro Alberto; REIS, Marcelo Menezes; BORNIA, Antonio Cezar. **Estatística:** para cursos de engenharia e informática. 3. ed. São Paulo: Atlas, 2010.

BORGELT, C.; KRUSE, R. **Graphical Models** - Methods for Data Analysis and Mining. J. Wiley & Sons, Chichester, United Kingdom, 2002. Disponível em: < http://books.google.je/books?id=UjXoSyRHYRQC&lpg=PP1&hl=pt-BR&pg=PP1#v=onepage&q&f=false>. Acesso em: 11 jun 2011.

BORTOLOTTO, Leandro Sehnem. **O Método de Redes Neurais pelo Algoritmo Kohonen para Clusterização na Shell Orion Data Mining Engine**. 2007. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

CAMPOS, Marcilia A.; RÊGO, Leandro C. **Métodos Probabilísticos e Estatísticos para Engenharias e Ciências Exatas**. 2011. Disponível em: < cin.ufpe.br/~lheva/estatistica/LivroProbabilidade.pdf> Acesso em: 23 ago 2012.

CASAGRANDE, Diego Paz. **O Módulo da Técnica de Associação pelo Algoritmo Apriori no desenvolvimento da Shell de Data Mining Orion**. 2005. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

CASSETTARI JUNIOR, José Márcio. **O Método de Lógica Fuzzy pelo Algoritmo Gustafson-Kessel na Tarefa de Clusterização da Shell Orion Data Mining Engine.** 2008. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2008.

CECI, Michelangelo. **Naive Bayesian Learning from Structural Data.** A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate Division of the University of Bari, Italy, 2005. Disponível em: <www.di.uniba.it/~ceci/micFiles/phdThesisCeci.pdf>. Acesso em: 5 jun 2011.

COPPIN, Ben. **Inteligência artificial**. Rio de Janeiro: LTC, 2010.

COLGATON, R. G. **A review of assessing the accuracy of classifications of remotely sensed data**. Remote Sensing of Environment, v. 49, n. 12, p. 1671-1678, 1991.

CROTTI JUNIOR, Ademar. **O Método de Lógica Fuzzy pelos algoritmos Robust CPrototypes e Unsupervised Robust C-Prototypes para a Tarefa de Clusterização na Shell Orion Data Mining Engine**. 2010. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2010.

DEVORE, Jay L. . **Probabilidade e estatística**: para engenharia e ciências. São Paulo: Thomson, 2006.

DOMINGOS, P.; PAZZANI, M. **On the optimality of the simple Bayesian classifier under zero-one loss**. Machine Learning. 1997.

DUDA, Richard O.; HART, Peter E. **Pattern Classification and Scene Analysis**. JohnWiley & Sons, 1973. Disponível em: < http://gigabook.us/g/download/0471056692/Pattern-Classification-Scene-Analysis/> Acesso em: 11 jun 2011.

DUDA, Richard O.; HART, Peter E.; STORK, David G. **Pattern Classification**. 2nd Edition. New York: JohnWiley & Sons, 2000. Disponível em: < http://gigabook.us/g/download/0471056693/Pattern-Classification-%282nd-Edition%29/> Acesso em: 11 jun 2011.

ELIAN, Silvia Nagib; FARHAT, Cecília Aparecida Vaiano. **Estatística básica**. São Paulo: LCTE, c2006.

GAVA, Éverton Marangoni. **O Método de Densidade pelo Algoritmo Density-Based Spatial Clustering of Applications with Noise (DBSCAN) na Tarefa de Clusterização da Shell Orion Data Mining Engine**. 2011. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2011.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gegory; SMYTH, Padhraic. **From Data mining to Knowledge Discovery in Databases**. AI Maganize, v. 17, n. 3, p. 37-54, 1996. Disponível em: <http://www.daedalus.es/fileadmin/daedalus/doc/MineriaDeDatos/fayyad96.pdf>. Acesso em: 6 de abr 2011.

FERNANDES, Edite Manuela da G.P. **Estatística Aplicada**. Braga: Universidade do Minho, 1999.

FIELD, Andy. **Descobrindo a estatística usando o SPSS**. 2. ed. Porto Alegre: Artmed, 2009.

FINN, Anderson. **The Statistical Analysys of Data**. Springer-Verlag, 1996. Disponível em: <www.csc.lsu.edu/~xiaoz/stat.pdf> Acesso em: 15 maio 2011.

FRIEDMAN, Nir; GEIGER, Dan; GOLDSZMIDT, Moises. Bayesian Network Classifiers. Kluwer Academic Publishers, Boston, 1997. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.4123&rep=rep1&type=pdf >Acesso em: 12 jun 2011.

FURLAN, José Davi. **Modelagem de objetos através da UML**. São Paulo: Makron Books: 1998.

GHELLERE, Samuel Lodetti. **Algoritmo Standard Ant Clustering Algorithm na Tarefa de Clusterização da Shell Orion Data Mining Engine**. 2012. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2012.

GOLDSCHMIDT, Ronaldo; PASSOS, **Emmanuel Lopes. Data Mining**: um guia prático. Rio de Janeiro: Elsevier, 2005.

HAN, Jiawei; KAMBER, Micheline. **Data mining**: concepts and techniques. San Francisco: Morgan Kaufmann, 2006.

KANTARDIZIC, Mehmed. **Data mining**: Concepts, Models, Methods, and Algorithms. John Wiley & Sons, 2003.

KORB, Kevin B.; NICHOLSON, Ann E. Bayesian artificial intelligence. Florida: Chapman & Hall, 2004.

LINDLEY, Dennis V. **The 1988 Wald Memorial Lectures**: The Present Position in Bayesian Statistics. Statistical Science, 1990. Disponível em:< projecteuclid.org/euclid.ss/1177012253> Acesso em: 12 jun 2010.

LIPSCHUTZ, Seymour. **Probabilidade**. 4. ed. rev. São Paulo: Makron Books, 1994.

MARTINS, Denis Piazza. **O Algoritmo de Particionamento K-means na Tarefa de Clusterização da Shell Orion Data Mining Engine**. 2007. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

MARTENS, Harald; NAES, Tormod. **Multivariate calibration.** Chichester: **Wiley-Interscience,** 1993.

MEYER, Paul L. **Probabilidade aplicações à estatística**. Rio de Janeiro: Ao Livro Técnico, 1969.

MONDARDO, Ricardo Lineburger. **O algoritmo C4.5 na tarefa de classificação da Shell Orion Data Mining Engine**. 2009. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2009.

NEAPOLITAN, Richard E. **Learning bayesian networks**. New Jersey: Pearson Prentice Hall, c2004.

OLSON, David L.; DELEN, Dursun. **Advanced Data Mining Techniques**. Lincoln: Springuer, 2008.

PAULINO, Carlos Daniel; TURKMAN, Maria Antónia Amaral; MURTEIRA, Bento. **Estatística Bayesiana**. Lisboa: Fundação Calouste Gulbenkian, 2003.

PELEGRIN, Diana Colombo**. A Tarefa de Classificação e o Algoritmo ID3 para Indução de Árvores de Decisão na Shell de Data Mining Orion**. 2005. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2005.

PEREGO, Daniel. **O Método de Lógica Fuzzy pelo algoritmo GATH-GEVA na tarefa de clusterização da Shell Orion Data Mining Engine**. 2009. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2009.

PORTUGAL, Agnaldo C. **Probabilidade e Raciocínio Científico**. Episteme, Porto Alegre, n. 18, p. 19-40, jan./jun. 2004. Disponível em: <http://www.ilea.ufrgs.br/episteme/portal/pdf/numero18/episteme18_artigo_portugal1.pdf> Acesso em: 21 out 2012.

RAIMUNDO, Lidiane Rosso. **O Algoritmo CART na Tarefa de Classificação da Shell Orion Data Mining Engine**. 2007. Trabalho de Conclusão de Curso - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina, 2007.

REZENDE, Solange Oliveira. **Sistemas inteligentes**: fundamentos e aplicações. Barueri, SP: Manole, 2005.

RHODES, Raphael H. **Hipnotismo sem mistério**. 8..ed Rio de Janeiro: Ed. Record, 1994.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2004.

SCOTTI, Ana Paula. **O Método de Redes Neurais com Função de Ativação de Base Radial para a Tarefa de Classificação na Shell Orion Data Mining Engine**. 2010. Trabalho de Conclusão de Curso – Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, Santa Catarina. 2010.

SILVA, Antonio A. S. **AÎURI**: Um Portal Para Mineração de Textos Integrado a Grids Computacionais. Dissertação de Mestrado - Curso de Ciências em Engenharia Civil, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2007. Disponível em: <http://www.coc.ufrj.br/index.php/component/docman/cat_view/1-mestrado/88-2004?Itemid=> Acesso em: 29 out 2012.

SILVA, Daniel Fagundes. **Software Agregador de Notícias com Classificador Bayesiano.** 2007. Trabalho de Conclusão de Curso – Departamento de Informática e de Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2007.

SUBBALAKSHMI, G.; RAMESH, K.; RAO, M. Chinna. **Decision Support in Heart Disease Prediction System using Naive Bayes**. Indian Journal of Computer Science and Engineering, 2011. Disponível em < http://www.periodicos.capes.gov.br/> Acesso em: 30 ago 2012.

SCHWARTSMANN, C. R.; BOSCHIN, L. C.; MOSCHEN, G. M.; GONÇALVES, E. Z.; RAMOS, A. S. N.; GUSMÃO, P. D. F.; JACOBUS, L. S. **Classificação das fraturas trocantéricas**: avaliação da reprodutibilidade da classificação AO. Revista Brasileira de Ortopedia. v. 41, n. 7, p. 264-7, 2006. Disponível em: <http://www.4shared.com/get/64712151/e204d086/Classificao_das_fraturas_troca.html> Acesso em: 21 ago 2012.

TAN, Pan-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Datamining**: mineração de dados. Rio de Janeiro: Ciência Moderna, 2009.

WITTEN, Ian H.; FRANK, Eibe; HALL, Mark A. Data mining practical machine learning tools and techniques. 3. ed. Burlington: Morgan Kaufmann, 2011.

APÊNDICE A – ARTIGO

O Teorema de Probabilidade pelo Algoritmo *Naive Bayes* para a Tarefa de Classificação na *Shell Orion Data Mining Engine*

Marcio Novaski[1], Merisandra Côrtes de Mattos Garcia[2]

[1]Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma– SC

[2]Professora do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma– SC

mnovaski@terra.com.br, mem@unec.net

Resumo. *Os constantes avanços tecnológicos têm permitido que as mais diversas organizações gerem repositórios de dados cada vez maiores tornando necessário o desenvolvimento de tecnologias que auxiliem nas análises e obtenção de conhecimento a partir destes dados. O data mining destaca-se dentre essas tecnologias utilizando algoritmos específicos para extração de possíveis padrões existentes nesses conjuntos de dados. Este artigo demonstra a modelagem matemática e o desenvolvimento do algoritmo Naive Bayes, que utiliza o Teorema de Bayes e os conceitos de estatística e probabilidade para a tarefa de classificação. Esta tarefa consiste da identificação de propriedades comuns entre os elementos de uma base de dados e a associação desses elementos a uma classe predefinida, sendo que o algoritmo implementado atribui ao registro o rótulo de classe que apresentar a probabilidade máxima a posteriori, considerando a independência condicional entre os atributos, o que simplifica os cálculos necessários tornando mais rápido o processo de treinamento do algoritmo.*

1. Introdução

Atualmente um dos maiores problemas relacionados aos grandes bancos de dados é a complexidade das análises e tratamento na tentativa de obtenção de informações significativas a partir desses dados. Nesse contexto é necessário o desenvolvimento de técnicas específicas associadas a tecnologias computacionais para facilitar a extração de conhecimento e auxiliar a tomada de decisões, sendo que essa procura por relações entre os dados ficou conhecida como *Knowledge Discovery in Databases* (KDD), tendo o *data mining* como uma das principais etapas desse processo.

As tarefas utilizadas pelo processo de *data mining* são implementadas em ferramentas conhecidas como *shells*, sendo que existe em desenvolvimento a *Shell Orion Data Mining Engine*, que consiste em um projeto acadêmico implementado pelo Grupo de Pesquisa em Inteligência Computacional Aplicada do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense. Dentre as diversas tarefas existentes no processo de data mining, a Shell Orion atualmente disponibiliza as tarefas de associação, classificação e clusterização.

A classificação é vista como uma das mais populares e mais utilizadas tarefas do data mining [Goldschmidt e Passos 2005]. Consiste em associar um registro de uma determinada base de dados a uma classe predefinida. Os classificadores estatísticos baseados no teorema

de Bayes, também conhecidos como classificadores bayesianos, utilizam a probabilidade como principal recurso para identificação da classe. Usados para lidar com situações de incerteza por aleatoriedade, destacam-se por apresentar um menor tempo no processo de aprendizagem.

Neste artigo apresenta-se o desenvolvimento do algoritmo Naive Bayes, baseado do teorema de Bayes, no modulo de classificação da Shell Orion.

2. *Data Mining*

O data mining pode ser definido como o uso de técnicas tradicionais de análise de dados combinado a tecnologias e algoritmos sofisticados para processar grandes volumes de dados [Tan, Steinbach e Kumar 2009].

Considerando-se o vasto domínio de aplicação do data mining, existem diversas tarefas que podem ser selecionadas dependendo do objetivo a ser alcançado ou do problema que se deseja resolver, dentre elas estão [Goldschmidt e Passos 2005]:

a) **associação:** consiste em identificar padrões e relações entre dados de um determinado conjunto;
b) **classificação:** busca encontrar regras para dados em uma base que permita associá-los a um modelo de classe;
c) **clusterização:** separa os registros de uma base de dados em subconjuntos de objetos similares.

2. A tarefa de classificação em *data mining*

A classificação consiste na identificação de propriedades comuns entre um conjunto de elementos em uma base de dados e a classificação destes de acordo com os critérios de um determinado modelo [Witten, Frank e Hall 2011].

Durante o processo de classificação, os registros da base de dados são divididos em dois conjuntos, sendo um conjunto de treinamento que define o modelo de classificação e um conjunto de teste que avalia o modelo de classificação gerado [Russel e Norvig 2004].

O processo de classificação é dividido em duas etapas [Han e Kamber 2006]:

a) **aprendizagem:** o conjunto de dados de treinamento é submetido a um algoritmo classificador e tem-se como resultado o classificador propriamente dito;
b) **teste:** etapa que define a estimativa da exatidão do classificador dada pelo conjunto de teste.

É importante medir o desempenho de um classificador, pois essa análise fornece dados para avaliação da sua taxa de erro, precisão e generalização [Tan, Steinbach e Kumar 2009].

A aplicação da tarefa de classificação dentro do processo de data mining, requer o uso de métodos específicos. Dentre estes tem-se os estatísticos amplamente utilizados na construção de classificadores bayesianos.

3.1 Classificadores Bayesianos

Classificadores bayesianos são classificadores estatísticos que podem prever as probabilidades de associação a uma determinada classe, como por exemplo, estimar que um determinado registro pertença a uma classe em particular [Han e Kamber 2006].

São usados para lidar com situações de incerteza por aleatoriedade e geralmente apresentam um menor tempo de aprendizado durante o processo de treinamento.

Os classificadores bayesianos são baseados no teorema de Bayes e fundamentam-se basicamente nos conceitos da probabilidade. Dentre os classificadores bayesianos mais comuns estão as Redes de Crenças Bayesianas e o *Naive Bayes*.

3.2 O Algoritmo *Naive Bayes*

O algoritmo *Naive Bayes* foi proposto por Richard Duda e Peter Hart no livro *Pattern Classification and Scene Analisys* em 1973 na Califórnia, Estados Unidos, originado do trabalho de reconhecimento de padrões e análise de cenas em aprendizado de máquina.

É um classificador estatístico que utiliza os conceitos da probabilidade para identificar a qual classe predefinida pertence um determinado registro. A sua principal característica é a suposição da independência condicional, ou seja, ele assume que o efeito do valor de um atributo a uma determinada classe é independente dos valores dos demais atributos [Coppin 2010].

Dois parâmetros de entrada são necessários para o funcionamento do algoritmo, sendo um conjunto de treinamento e um conjunto de testes. Para identificar a classificação de um registro, calcula-se a probabilidade a posteriori para cada classe possível. A classificação correta será aquela cuja probabilidade a posteriori for maior, conhecida como máxima a posteriori (MAP) [Coppin 2010].

4 O Algoritmo *Naive Bayes* na Tarefa de Classificação da *Shell Orion Data Mining Engine*

A modelagem do algoritmo *Naive Bayes* iniciou-se com a construção dos diagramas de caso de uso, atividades e seqüência utilizando os padrões UML. Posteriormente foi desenvolvida a demonstração matemática do funcionamento do algoritmo com a finalidade de facilitar o entendimento e a sua implementação.

O algoritmo necessita de dois conjuntos de dados sendo um para o treinamento e outro para testes. Definidos os dois conjuntos de dados, o primeiro passo consiste em:

a) **identificar o número de classes**;
b) **identificar o total de amostras no conjunto de treinamento**;
c) **identificar o total de amostras para cada classe**;

Com os parâmetros obtidos, calcula-se primeiramente a probabilidade *a priori* de cada classe conforme a fórmula:

$$P(C_i) = \frac{S_i}{S}, i = 1,...,n$$

Visto que o cálculo da probabilidade é dado por uma fração, se o numerador assumir o valor zero o resultado pode ser comprometido. Para evitar-se esse problema utiliza-se o teorema da aproximação de Laplace adicionando a constante 1 ao numerador e a variável *k* ao denominador:

$$\frac{n_c + 1}{n + k}$$

A fórmula para o cálculo da probabilidade *a priori* de cada classe é alterada para:

$$P(C_i) = \frac{S_i + 1}{S + k}, i = 1,...,n$$

O próximo passo consiste em calcular a probabilidade de cada atributo da amostra de teste condicionada a cada uma das classes aplicando o teorema de Laplace à fórmula:

$$P(x_k \mid C_i) = \frac{S_{ik} + 1}{S_i + k}$$

Estima-se então as probabilidades da amostra total $P(X|C_i)$ para cada uma das classes considerando a independência condicional:

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

Posteriormente as regras de classificação são aplicadas considerando-se a probabilidade *a priori* de cada classe:

$$P(X \mid C_i) \times P(C_i)$$

O resultado obtido pela a aplicação das regras de classificação define a máxima *a posteriori* que indicará a qual classe pertence o registro analisado.

4.1 Implementação

O algoritmo *Naive Bayes* foi desenvolvido no módulo de classificação da *Shell Orion Data Mining Engine*, por meio da linguagem de programação Java e do ambiente de programação integrado *NetBeans* 7.1.1.

Para a execução do algoritmo é necessário que seja informado pelo usuário o método de teste a ser realizado. Entre as opções disponíveis na *Shell Orion* tem-se o método *holdout* e *usar o conjunto de treinamento*.

A opção *holdout* permite resultados mais confiáveis, uma vez que as amostras reservadas para o conjunto de testes não fazem parte do conjunto de treinamento [Witten, Frank e Hall 2011].

A sugestão de proporção para dados reservados é de 2/3 para treinamento e 1/3 para testes, mas a alteração do percentual de dados reservado para treinamento também é permitida.

Como forma de auxiliar as análises de desempenho dos resultados obtidos, foram empregados os seguintes índices de validação:

a) **sensibilidade:** é a capacidade do classificador para encontrar os registros que realmente pertencem a classe considerada;
b) **especificidade:** consiste na capacidade do classificador para encontrar os registros que não pertencem a classe considerada;
c) **acurácia:** é o grau de exatidão, relação entre os valores estimados e os valores reais;
d) **erro:** refere-se à taxa de erro geral, ou seja, a proporção de registros classificados incorretamente;
e) **confiabilidade positiva:** é a capacidade do classificador para identificar corretamente os verdadeiros positivos;
f) **índice kappa:** é o coeficiente de avaliação da concordância entre os resultados.

4.2 Resultados Obtidos

A base de dados utilizada na avaliação do algoritmo *Naive Bayes* na *Shell Orion* refere-se a dados clínicos de pacientes com diagnóstico positivo para o câncer de mama dos hospitais da Universidade de *Wisconsin*, Madison nos Estados Unidos. É composta por 683 registros com 10 atributos e 2 classes que identificam o resultado do diagnóstico como benigno ou maligno.

Os testes realizados abrangem a análise da precisão dos resultados obtidos e a comparação com os resultados da ferramenta Weka 3.6.7 (http://www.cs.waikato.ac.nz/ml/weka/), a comparação entre os tempos de processamento durante o treinamento e a classificação na *Shell Orion*, e os tempos de processamento entre a *Shell Orion* e a ferramenta Weka.

4.2.1 Classes identificadas pelo algoritmo *Naive Bayes*

O teste para a identificação das classes e validação do desempenho foi realizado utilizando o método *holdout*. Optou-se por repetir o teste por três vezes usando percentuais diferentes para cada teste.

A melhor taxa de acertos confirmou que a proporção de 2/3 dos registros reservados para treinamento apresenta resultados mais precisos.

A tabela 1 mostra a matriz de confusão que exibe o número de classificações preditas versus o número de classificações corretas. Por meio da contagem destes registros tem-se a avaliação de desempenho do modelo [Tan, Steinbach e Kumar 2009].

Tabela 1. Matriz de confusão – Classificação da base do câncer de mama na Shell Orion

Classe	Predita – Classe 2	Predita – Classe 4	Total
Verdadeira – Classe 2	**176**	**2**	*178*
Verdadeira – Classe 4	**0**	**54**	*54*
Total	*176*	*56*	*232*

Dos valores obtidos na matriz de confusão pode-se extrair as seguintes variáveis mostradas na tabela 2:

Tabela 2. Relação de verdadeiros e falsos positivos e verdadeiros e falsos negativos.

Variável	Valor
Verdadeiros Positivos	**176**
Falsos Positivos	**0**
Verdadeiros Negativos	**54**
Falsos Negativos	**2**

Com os valores das variáveis da tabela 2, calcularam-se os índices de validação do modelo. A tabela 3 demonstra os valores obtidos pela *Shell Orion*.

Tabela 3. Índices de validação – Shell Orion.

Variável	*Shell Orion*
Registros classificados corretamente	**230**
Registros classificados incorretamente	**2**
Sensibilidade	**98,88%**
Especificidade	**100%**
Acurácia	**99,14%**
Erro	**0,86%**
Confiabilidade positiva	**100%**
Índice Kappa	**0,976**

A análise dos valores obtidos nos índices de validação mostra que o desempenho do módulo desenvolvido pode ser considerado satisfatório por apresentar índices próximos de 100% e taxa de erro próxima de zero.

4.2.2 Tempos de processamento do algoritmo *Naive Bayes*

Com o objetivo de se obter conjuntos de diversos tamanhos, a base de dados original foi replicada gradativamente por meio do crescimento geométrico, tornando possível a visualização do comportamento de diferentes tamanhos de cargas (centenas a milhares) dentro de um pequeno intervalo de iterações:

$$C_k = N \times 2^{k-1}$$

Na tabela 4 são verificados os tempos de processamento para as etapas de treinamento e classificação obtidos pelo algoritmo em relação ao tamanho da base de dados.

Tabela 4. Tempos de processamento nos processos de treinamento e classificação.

Quantidade de registros	Treinamento	Classificação
683	00m: 00s. 054ms	00m: 00s. 047ms
1366	00m: 00s. 076ms	00m: 00s. 233ms
2732	00m: 00s. 103ms	00m: 00s. 816ms
5464	00m: 00s. 152ms	00m: 03s. 236ms
10928	00m: 00s. 269ms	00m: 12s. 477ms
21856	00m: 00s. 516ms	00m: 54s. 051ms
43712	00m: 00s. 959ms	04m: 11s. 126ms
87424	00m: 02s. 018ms	22m: 40s. 877ms

Para as análises dos valores obtidos utilizou-se o pacote estatístico *Statistical Package for Social Sciences* (SPSS).

Inicialmente aplicou-se o teste de normalidade *Shapiro-Wilk* com a finalidade de verificar se a distribuição dos valores da amostra era uma distribuição normal. O teste resultou em um *p-value* menor que o nível de significância preestabelecido (0,05) constatando que a distribuição não era normal.

Optou-se por normalizar os dados por meio da transformação logarítmica dos tempos. Após o cálculo do *log* natural dos tempos, aplicou-se novamente o teste *Shapiro-Wilk* e obteve-se uma distribuição gaussiana em ambos os grupos.

A fim de analisar a homogeneidade entre as variâncias observadas, foi aplicado o teste *F* de *Lavene*, que verifica se a variância observada nos grupos é a mesma [Field 2009], constatando-se a heterogeneidade entre as variâncias e levando à utilização do teste *t* de *Student*. Este teste compara dois conjuntos de dados com valores quantitativos considerando seus valores médios baseando-se e uma ditribuição normal de valores [Barbetta, Reis e Bornia 2010].

A realização do teste *t* de *Student* para amostras independentes revelou que a média do logaritmo do tempo do processo de treinamento foi significativamente mais rápida do que a média do logaritmo do tempo do processo de classificação, rejeitando a hipótese nula (H0), portanto, infere-se que a variação não ocorre ao acaso, mas em função da tarefa executada.

4.2.3 Tempos de processamento *Shell Orion versus* Weka

A fim de comparar o desempenho em relação ao tempo de processamento entre as duas ferramentas, realizou-se a coleta dos tempos gastos pela *Shell Orion* e pela Weka para executar as mesmas tarefas nas mesmas condições.

A tabela 5 mostra os tempos coletados:

Tabela 5.Tempos de processamento da Shell Orion e da ferramenta Weka.

Quantidade de registros	Treinamento	Classificação
683	00m: 00s. 054ms	00m: 00s. 047ms
1366	00m: 00s. 076ms	00m: 00s. 233ms
2732	00m: 00s. 103ms	00m: 00s. 816ms
5464	00m: 00s. 152ms	00m: 03s. 236ms
10928	00m: 00s. 269ms	00m: 12s. 477ms
21856	00m: 00s. 516ms	00m: 54s. 051ms
43712	00m: 00s. 959ms	04m: 11s. 126ms
87424	00m: 02s. 018ms	22m: 40s. 877ms

Após a coleta dos tempos de processamento, aplicou-se o teste de normalidade *Shapiro-Wilk* com a finalidade de verificar se a distribuição dos valores da amostra era uma distribuição normal. O teste resultou em um *p-value* menor que o nível de significância preestabelecido (0,05) constatando que a distribuição não era normal.

Novamente optou-se por normalizar os dados por meio da transformação logarítmica dos tempos. Após o cálculo do *log* natural dos tempos, aplicou-se uma vez mais o teste *Shapiro-Wilk* e obteve-se uma distribuição gaussiana em ambos os grupos.

A fim de analisar a homogeneidade entre as variâncias observadas, foi aplicado o teste *F* de *Lavene*, constatando-se a heterogeneidade entre as variâncias e levando à utilização do teste *t* de *Student* que comparou as médias dos tempos na Shell Orion e na Weka.

A realização do teste *t* de *Student* para amostras independentes revelou que a média do logaritmo dos tempos na Weka foi significativamente mais rápida do que a média do logaritmo dos tempos na *Shell Orion*, rejeitando a hipótese nula (H0), portanto, infere-se que a variação não ocorre ao acaso.

4.2.4 Comparação dos índices de validação com a ferramenta Weka 3.6.7

Nessa etapa analisou-se a qualidade dos resultados gerados pela Shell Orion e comparados com os resultados obtidos pela ferramenta Weka. Apenas os índices de validação foram considerados nessa avaliação, portando os tempos de processamento foram ignorados.

Em uma primeira análise optou-se por usar como teste o mesmo conjunto de treinamento construído com todos os 683 registros da base de dados original. Os resultados são apresentados na tabela 6.

Tabela 6. Índices de validação – Comparação entre Shell Orion e Weka

Variável	*Shell Orion*	Weka
Registros classificados corretamente	**431**	**431**
Registros classificados incorretamente	**667**	**667**
Sensibilidade	**97,07%**	**97,07%**
Especificidade	**98,75%**	**98,75%**
Acurácia	**97,66%**	**97,66%**
Erro	**2,34%**	**2,34%**
Confiabilidade positiva	**99,31%**	**99,31%**
Índice Kappa	**0,949**	**0,949**

Os resultados mostram que o algoritmo Naive Bayes, implementado na Shell Orion, obteve índices com valores idênticos aos apresentados pela ferramenta Weka na análise desta base de dados referente ao câncer de mama. O percentual de registros classificados correta e incorretamente foram os mesmos (97.66 %), bem como o índice Kappa que apresentou um valor de concordância (0,949) classificado como excelente.

Na análise seguinte optou-se por utilizar o método *holdout* reservando 66% da base de dados para treinamento e o restante para teste. Os valores obtidos para os índices de validação nas duas ferramentas são mostrados na tabela 7.

Tabela 7. Índices de validação – Comparação entre *Shell Orion* e Weka.

Variável	*Shell Orion*	Weka
Registros classificados corretamente	**230**	**229**
Registros classificados incorretamente	**2**	**3**
Sensibilidade	**98,88%**	**98,88%**
Especificidade	**100%**	**98,1%**
Acurácia	**99,14%**	**98,7%**
Erro	**0,86%**	**1,3%**
Confiabilidade positiva	**100%**	**99,4%**
Índice Kappa	**0,976**	**0,964**

A análise dos valores obtidos nos índices de validação e a comparação com a Weka mostrou que a *Shell Orion* obteve resultados melhores em quase todos os índices confirmando o seu correto funcionamento, especialmente no que diz respeito a percentuais de acerto (*Shell Orion* – 99,14%; Weka – 98,71%) e índice Kappa (*Shell Orion* – 0,976; Weka – 0,964).

Concluiu-se que ambas as ferramentas têm comportamento semelhante com relação à qualidade dos resultados do processo de classificação.

5. Considerações Finais

Este artigo apresentou o algoritmo *Naive Bayes* que utiliza os conceitos de estatística e probabilidade baseados no teorema de Bayes, para a tarefa de classificação da *Shell Orion Data Mining Engine*, contribuindo com a ampliação das funcionalidades da ferramenta.

Pode-se confirmar, diante dos resultados obtidos, a aplicabilidade do algoritmo para a tarefa de classificação visto que os resultados avaliados apresentaram bons índices de validação relacionados à precisão e confiabilidade dos valores obtidos.

Concluiu-se que o algoritmo foi implementado com sucesso, pois apresentou resultados satisfatórios obtidos no processo de classificação e ainda melhores quando comparados com uma ferramenta de *data mining* conceituada, confirmando o seu correto funcionamento.

Referências

BARBETTA, Pedro Alberto; REIS, Marcelo Menezes; BORNIA, Antonio Cezar. **Estatística:** para cursos de engenharia e informática. 3. ed. São Paulo: Atlas, 2010.

COPPIN, Ben. **Inteligência artificial**. Rio de Janeiro: LTC, 2010.

DUDA, Richard O.; HART, Peter E.; STORK, David G. **Pattern Classification**. 2nd Edition. New York: JohnWiley & Sons, 2000. Disponível em: < http://gigabook.us/g/download/0471056693/Pattern-Classification-%282nd-Edition%29/> Acesso em: 11 jun 2011.

FIELD, Andy. **Descobrindo a estatística usando o SPSS**. 2. ed. Porto Alegre: Artmed, 2009.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel Lopes. **Data mining:** uma guia prático. Rio de Janeiro: Elsevier, 2005.

HAN, Jiawei; KAMBER, Micheline. **Data mining**: concepts and techniques. San Francisco: Morgan Kaufmann, 2006.

TAN, Pan-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Datamining**: mineração de dados. Rio de Janeiro: Ciência Moderna, 2009.

WITTEN, Ian H.; FRANK, Eibe; HALL, Mark A. Data mining practical machine learning tools and techniques. 3. ed. Burlington: Morgan Kaufmann, 2011.