UNIVERSIDADE FEDERAL DO AMAZONAS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E TECNOLOGIA PARA RECURSOS AMAZÔNICOS

LINHA DE PESQUISA: ESTUDOS TEÓRICOS E COMPUTACIONAIS

MODELOS COMPUTACIONAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO E AGRUPAMENTO DE VARIEDADES DE TUCUMÃ (*Astrocaryum aculeatum* G. Mey.)

MAFRAN MARTINS FERREIRA JÚNIOR

ITACOATIARA
2015

UNIVERSIDADE FEDERAL DO AMAZONAS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E TECNOLOGIA PARA RECURSOS AMAZÔNICOS

MAFRAN MARTINS FERREIRA JÚNIOR

MODELOS COMPUTACIONAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO E AGRUPAMENTO DE VARIEDADES DE TUCUMÃ (*Astrocaryum aculeatum* G. Mey.)

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia para Recursos Amazônicos da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Ciência e Tecnologia para Recursos Amazônicos, área de concentração Desenvolvimento Científico e Tecnológico em Recursos Amazônicos, linha de pesquisa Estudos Teóricos e Computacionais.

Orientador: Prof. Dr. Jorge Yoshio Kanda

ITACOATIARA
2015

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Ferreira Junior, Mafran Martins
F383m MODELOS COMPUTACIONAIS BASEADOS EM APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO E AGRUPAMENTO DE VARIEDADES DE TUCUMÃ (Astrocaryum aculeatum G. Mey.) / Mafran Martins Ferreira Junior. 2015
141 f.: il. color; 31 cm.

Orientador: Jorge Yoshio Kanda
Dissertação (Mestrado em Ciência e Tecnologia para Recursos Amazônicos) - Universidade Federal do Amazonas.

1. Aprendizado de Máquina. 2. Reconhecimento de Padrões. 3. Mineração de Dados. 4. Recursos Amazônicos. 5. Tucumã (Astrocaryum aculeatum G. Mey.). I. Kanda, Jorge Yoshio II. Universidade Federal do Amazonas III. Título

MAFRAN MARTINS FERREIRA JUNIOR

Modelo Computacionais Baseados em Aprendizado de Máquina para Classificação e Agrupamento de Variedades de Tucumã (*Astrocaryum aculeatum* G. Mey.)

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia para Recursos Amazônicos da Universidade Federal do Amazonas, como parte do requisito para obtenção do título de Mestre em Ciência e Tecnologia para Recursos Amazônicos, área de concentração Desenvolvimento Científico e Tecnológico em Recursos Amazônicos.

Aprovado em 31 de julho de 2015.

BANCA EXAMINADORA

Dr. Jorge Yoshio Kanda, Presidente
Universidade Federal do Amazonas

Dr. Raimundo da Silva Barreto
Universidade Federal do Amazonas

Dr. Luís Antônio de Araújo Pinto
Universidade do Estado do Amazonas

À minha mãe Elciete da Silva Campos e aos meus avós maternos (*in memoriam*) Tamió Costa Campos e Creuza da Silva Campos pela educação e valores a mim ensinados.

AGRADECIMENTOS

Antes de tudo, agradeço a Deus por sua imensa bondade e por tudo que tem me proporcionado desde o momento de meu nascimento. Deus este que sempre me deu forças e nunca me permitiu fraquejar diante dos percalços da vida. Agradeço a Ele também pelos meus familiares e por ter me dado a oportunidade de viver em meio a pessoas maravilhosas que encontrei em meu caminho.

Agradeço aos meus avós maternos (*in memoriam*) Sr. Tamió Costa Campos e Profª. Creuza da Silva Campos por tudo que fizerem em vida por seus filhos e netos, nos proporcionando toda a estrutura e amor necessários à nossa formação pessoal e acadêmica. Agradeço à minha mãe Prof.ª Elciete da Silva Campos, uma mulher guerreira que possui um coração formado de amor e bondade para com o próximo.

Agradeço aos meus grandes amigos, sem os quais sem dúvida eu nada seria. Amigos estes que sempre estiveram comigo nos momentos bons e ruins. Também agradeço imensamente ao meu orientador Prof. Dr. Jorge Yoshio Kanda pela atenção e paciência dispensadas a mim, além de todo o conhecimento repassado com muito esmero.

Por fim, agradeço ao Instituto de Ciências Exatas e Tecnologia da Universidade Federal do Amazonas pela oportunidade de participar desse programa de mestrado, o qual agregou muito valor à minha formação.

Sejam fortes e corajosos, todos vocês que esperam no Senhor.

Salmos 31:24

RESUMO

O bioma amazônico possui uma gama de recursos naturais com alto valor econômico, os quais podem ser explorados de maneira sustentável para gerar emprego e renda. Dentre esses recursos destaca-se o tucumã, fruta nativa da região utilizada na culinária, no artesanato e comercializada pelos produtores locais. A palmeira do tucumã apresenta ampla variedade dentro de uma mesma espécie, cada uma diferenciando-se quanto à sua morfologia, população, procedência, entre outros fatores. Cientificamente, a classificação taxonômica do tucumã é referente às suas espécies, ainda não existe nenhuma forma manual ou automática de classificar variedades da espécie *Astrocaryum aculeatum* G. Mey., também conhecida como tucumã do Amazonas. A indicação da variedade a qual uma unidade do fruto pertence é realizada de forma empírica, podendo ser confusa perante o aparecimento de frutos com grande diferença em suas características. Nesse cenário, esta pesquisa objetivou gerar e avaliar modelos computacionais capazes de classificar e agrupar quatro variedades de tucumã, encontradas na região do município de Itacoatiara-AM. O estudo teve como objetivo secundário indicar qual das variedades possui melhor potencial econômico quanto às características do fruto já colhido. Para gerar os modelos foram utilizadas três técnicas de Aprendizado de Máquina: Árvores de Decisão e Redes Neurais Artificiais na tarefa de classificação, e na tarefa de agrupamento a técnica K-Médias, usando as medidas de distância *Euclidiana* e de *Manhattan*. Os resultados obtidos com base no conjunto de dados mostram que os modelos gerados com as técnicas de aprendizado de máquina apresentaram índices satisfatórios para a predição de classes de variedades de tucumã.

Palavras-chave: Aprendizado de Máquina, Reconhecimento de Padrões, Mineração de Dados, Recursos Amazônicos, Tucumã (*Astrocaryum aculeatum* G. Mey.).

ABSTRACT

The amazon biome has a range of natural resources with high economic value, which can be exploited in a sustainable way to generate jobs and income. Among these resources, we can spotlight the tucuman, native fruit from the Amazonian region used in cooking, crafts and sold by local producers. The tucuman palm tree presents many varieties within the same specie, each one differs from the other in its morphology, population, origin, among other factors. Scientifically, the tucuman taxonomic classification refers to its species. There isn't yet a manual or an automatic way of classifying varieties of *Astrocaryum aculeatum* G. Mey., also known as Amazonian tucuman. The indication of the variety to which a fruit unit belongs is performed empirically and may be confused when there are fruits with a large difference in their characteristics. In this scenario, this study aimed to generate and evaluate computer models able to classify and get into groups four varieties of tucuman found in the Itacoatiara-AM region. The secondary objective of this study was to indicate which of the varieties have the best economic potential regarding to the harvested fruit characteristics. To generate the models, three machine learning techniques were used: Decision Trees and Artificial Neural Networks in the classification task, and to the grouping task the K-Means technique was applied, using *Euclidean* and *Manhattan* distance measurements. The results obtained based on the data set show that the models generated with machine learning techniques presented satisfactory indexes for predicting of varieties' classes of tucuman.

Key-words: Machine Learning, Patterns Recognition, Data Mining, Amazonian Resources, Tucuman (*Astrocaryum aculeatum* G. Mey.).

SUMÁRIO

1 INTRODUÇÃO

A Região Amazônica possui alto potencial de desenvolvimento científico e tecnológico (MADEIRA, 2014). O Estado do Amazonas, por abrigar a Zona Franca de Manaus, representa um polo produtivo que atrai muitas empresas e aquece a economia da região, gerando empregos e contribuindo para o desenvolvimento do país. Sabendo-se disso, o desenvolvimento de pesquisas científicas é de vital importância, podendo culminar em relevantes contribuições tanto para a academia quanto para a indústria.

Os insumos advindos da fauna e flora da Região Amazônica são utilizados como base para pesquisas em diversas áreas do conhecimento. Nos últimos anos, a aplicação dos recursos de Informática contribuiu significativamente na melhoria da coleta e processamento de dados nesses estudos. Diante disso, observa-se a oportunidade do desenvolvimento de projetos que unam os conceitos, técnicas e ferramentas da Tecnologia da Informação (TI), aplicando-os diretamente na produção e descoberta de informações relevantes a respeito de insumos naturais da região.

Atualmente no âmbito da informática, diversas áreas oferecem ferramentas robustas para auxiliar na realização dessas pesquisas. A Inteligência Artificial (IA) é uma delas, atraindo cada vez mais o interesse de outros campos do conhecimento, devido ao fato de executar de maneira eficiente a tarefa de processamento de dados. Segundo Xue & Zhu (2009), a IA vem acompanhando os avanços tecnológicos da Internet, *Hardware, Software* e Multimídia, o que culminou em muitas experiências profissionais diversificadas no meio científico, oferecendo aos pesquisadores novos pensamentos e alguns novos métodos para analisar dados de forma rápida e precisa.

Dentre os conceitos de IA, um dos mais conhecidos é o Aprendizado de Máquina (AM) (do termo em inglês *Machine Learning* - ML). Hua *et al* (2009) definem

AM como uma disciplina que estuda a forma de usar computadores para simular atividades de aprendizagem humanas, abordando métodos de auto aperfeiçoamento para a obtenção de novos conhecimentos e novas habilidades. O objetivo do AM é organizar a estrutura do conhecimento obtido, podendo implicar na melhoria progressiva de seu próprio desempenho. O aprendizado da máquina é o núcleo da Inteligência Artificial, representa uma técnica fundamental que permite o computador desenvolver inteligência (XUE & ZHU, 2009). Em seu escopo mais amplo, a principal tarefa é desenvolver sistemas automáticos capazes de generalizar um conceito a partir de exemplos observados anteriormente, construindo uma aprendizagem funcional de interdependências entre os domínios de entrada e saída arbitrários (DENG & LI, 2013).

Carvalho (2010) aponta que os componentes do AM são representados pelas classes de algoritmos que conseguem melhorar seu desempenho por meio de ganho de algum tipo de experiência. A filosofia do AM é formalmente definida por Mitchell (1997) como: um algoritmo obtém aprendizado através da experiência **E** atuando sobre uma classe de problema **T** e medidas de performance **P**, se essa performance **P** em relação ao problema **T**, melhora com a experiência **E**.

Outro conceito de especial importância acerca do aprendizado de máquina é o Reconhecimento de Padrões. A utilização de máquinas capazes de identificar padrões é alvo de muitos estudos atuais, haja vista que essa tarefa se faz cada vez mais necessária no cotidiano da humanidade. Pesquisas ao redor do mundo resultaram em aplicações de AM capazes de reconhecer padrões em diversas áreas de pesquisas científicas, tais como: sistemas especialistas, raciocínio automatizado, compreensão de linguagem natural, visão computacional, robôs inteligentes e outros (HART *et al*, 2000; XUE & ZHU, 2009).

No contexto de AM, as principais abordagens existentes são o Aprendizado Supervisionado e o Aprendizado Não-Supervisionado. O primeiro é o mais comum, sendo o método mais utilizado em pesquisas científicas, pois a maioria dos problemas a serem solucionados é de natureza supervisionada (BRINK & RICHARDS, 2014). A segunda abordagem é utilizada para descobrir padrões em dados não-categorizados, representando um método utilizado para atividades de exploração de informações (FURNKRANZ *et al*, 2012). Existem outras abordagens menos usuais, como o Aprendizado Semi-Supervisionado e a Classificação de Multi-Classes, mas estes não fazem parte do foco desta pesquisa. Dessa forma, apenas as duas primeiras abordagens serão discutidas nesta dissertação.

Diversas técnicas de AM são encontradas na literatura para realizar a classificação e agrupamento de objetos com as mesmas características. Neste estudo, apresenta-se uma discussão a cerca de três técnicas de aprendizado de máquina, usadas como meio para alcançar os objetivos propostos.

1.1 Motivação

1.1.1 O porquê do uso de Técnicas de Aprendizado de Máquina

As aplicações de AM são bastante abrangentes, cada técnica pode ser empregada em algum tipo de domínio para tentar solucionar um problema. Para compreender como se dá uma aplicação prática considere, por exemplo, que existem dois cogumelos com suas aparências físicas extremante parecidas, sendo que um deles é venenoso e o outro é perfeitamente comestível. Para este cenário, a utilização das técnicas de AM representa um meio de classificar os cogumelos com base na descoberta

de padrões ocultos (ou não) que possam ser determinantes na predição correta dos mesmos, evitando que haja confusão na hora de separá-los.

A partir da compreensão do problema descrito acima, vislumbra-se uma gama de aplicações práticas para as técnicas de AM, por exemplo: no comércio seu uso pode ajudar a classificar e descobrir novos grupos distintos de clientes, caracterizando-os com base no seu padrão de compra. Em biologia, as técnicas podem ser aplicadas para classificar genes pela similaridade de suas funções, classificar e agrupar espécies e variedades plantas, ajudar a identificar toxinas, classificar problemas de saúde pública ou categorizar doenças, entre outras diversas situações nas quais se podem aplicar as técnicas de AM.

Segundo Weiss & Indurkhya (1995), um programa de computador pode tomar decisões baseadas na experiência contida em exemplos solucionados com sucesso. Daí vem a motivação para que tantos estudos utilizem AM, haja vista que suas técnicas representam um meio de criar modelos inteligentes capazes de aprender padrões e realizar a classificação automática de novos exemplos.

Nos últimos anos, aumentou significativamente a utilização das técnicas de AM em pesquisas acadêmicas. Isto se deve ao fato de que os modelos computacionais gerados apresentam resultados bastante satisfatórios em diversas áreas (XUE & ZHU, 2009).

Para realizar essas modelagens alguns *softwares* livres são encontrados, dentre os quais, o mais utilizado em ambiente acadêmico é o *Waikato Environment for Knowledge Analysis* – WEKA (WITTEN & FRANK, 2005). Essa ferramenta possui uma série de algoritmos de preparação de dados, de aprendizado de máquina e de validação de resultados.

O *software* em questão foi desenvolvido em linguagem de programação JAVA e possui código-fonte aberto, podendo ser encontrado na Web. A sua GUI (*Graphical User Interface*; Interface Gráfica do Usuário) possui alto nível de usabilidade e seus resultados apresentam dados estatísticos e analíticos sobre o domínio estudado. Embora a maioria de seus recursos seja acessada por meio da GUI, grande parte dos usuários desconhece que o WEKA fornece uma poderosa e flexível API (*Application Programming Interface*; Interface de Programação de Aplicações), que torna possível sua integração a qualquer tipo de sistema JAVA. Estas características permitem utilizar a WEKA API dentro de programas próprios, viabilizando a incorporação do código-fonte para criar modelos particulares de acordo com a necessidade de cada projeto.

Diante do exposto, o *software* WEKA foi a ferramenta selecionada para a execução desta pesquisa, pois permite que futuramente seu código possa ser usado para a confecção de novos modelos, baseando-se nas análises dos resultados gerados por este estudo.

1.1.2 O porquê do estudo com variedades de *Astrocaryum aculeatum* G. Mey.

Ao longo dos séculos, o extrativismo de recursos naturais sustentou mercados e contribuiu para o crescimento socioeconômico dos povos da Amazônia. Neste contexto, alguns recursos vegetais ganharam tanta importância e visibilidade, que não se pode imaginar a dissociação de suas imagens às comunidades da região norte (DIDONET, 2012). Dentre esses recursos estão muitas espécies de palmeiras frutíferas com alta relevância para o desenvolvimento da região, como por exemplo, a *Astrocaryum aculeatum* G. Mey (tucumã), espécie nativa usada na subsistência dos povos de áreas

rurais e de extrema importância para o mercado local e externo (CLEMENT *et al*, 2005).

No estado do Amazonas o tucumã é tão apreciado que já faz parte do cardápio diário da população, sendo consumido em diversas formas e comercializado nos mais diferentes estabelecimentos (desde feiras de produtores até restaurantes de alto padrão). Atualmente, todas as partes da planta são aproveitadas, mas a importância essencial do tucumã é pautada em seu fruto, com base no qual se desenvolveu um mercado promissor que vem crescendo a cada ano na região da Amazônia central (SCHROTH *et al*, 2004).

Em relação às cidades amazônicas onde há comércio de tucumã, Manaus se destaca como uma das mais promissoras, gerando emprego e renda para a população. De acordo com Didonet (2012), entre os anos 2011 e 2012, as três localidades com maior importância para o abastecimento do comércio manauara foram os municípios amazonenses de Itacoatiara e Autazes, seguidos pelo município paraense de Terra Santa. Os dados apontados pelo autor indicam que Itacoatiara foi a maior fornecedora de tucumãs, representando 15% de todo o abastecimento no período citado.

Devido a esses motivos, diversas áreas da ciência têm se dedicado a desenvolver pesquisas científicas com o objetivo de descobrir mais informações relevantes sobre esse fruto. Uma dessas áreas é a Biologia, que através do ramo da Taxonomia Vegetal preocupa-se com a identificação correta das espécies do gênero *Astrocaryum* (KAHN, 2008). A palmeira do tucumã possui características de plantas alógamas, ou seja, sua fecundação é cruzada, necessitando de um agente polinizador, como vento, insetos, morcegos, etc. (OLIVEIRA, 2001). Por esse fator, entre uma mesma espécie podem existir muitas variedades do fruto, apresentando diferenças morfológicas devido à influência do clima, do solo, entre outros (MENDONÇA, 1996).

Cientificamente, a classificação taxonômica das espécies de tucumã é baseada na observação e comparação das estruturas morfológicas da planta, como: tamanho e forma dos frutos; estrutura das folhas e flores; posicionamento dos cachos na palmeira; presença de pelos nas folhas; entre outras variáveis (FERREIRA & GENTIL, 2005; BACELAR-LIMA *et al*, 2006; KAHN, 2008). A avaliação de tais características confere à planta uma atribuição como determinada espécie.

Diante do exposto, compreende-se que de forma manual é difícil determinar o número específico de variedades de uma espécie de tucumã. Os estudos científicos existentes analisaram amostras pontualmente, além disso, o foco desses trabalhos foi na classificação de espécies, e não no estudo das variedades.

Com base nas pesquisas realizadas, constatou-se que ainda não existe uma metodologia manual ou automática para classificação de variedades de *Astrocaryum aculeatum*. O motivo dessa inexistência é que cada área geográfica pode ter muitos tipos diferentes de tucumã, além disso, os frutos de uma mesma variedade podem apresentar variações em algumas características morfológicas, o que torna inviável a criação de chaves de identificação taxonômica manuais. Neste cenário, o uso das técnicas de AM pode contribuir para a descoberta de informações relevantes, representando um meio de validar e automatizar a classificação das variedades da espécie *Astrocaryum aculeatum*, analisando as características extraídas do fruto para identificar padrões existentes. Sua utilização também pode indicar o número correto de variedades existentes por meio de agrupamento, auxiliando na possível descoberta de uma nova variedade do fruto, caso os algoritmos identifiquem muitas instâncias com valores extremamente diferentes dos padrões generalizados pelos modelos.

Por ser uma fruta nativa da região amazônica e apresentar alta concentração de suas variedades, o tucumã possui um grande potencial para comercialização e

fabricação de produtos derivados, principalmente no que diz respeito à sua polpa (MENDONÇA, 1996; KAHN & MOUSSA, 1999; CLEMENT *et al*, 2005). Segundo Didonet (2012), parte significativa dos frutos que chegam a Manaus é destinada ao beneficiamento (despolpamento). O autor aponta que entre 2011 e 2012, um total de 196,7 toneladas de tucumã foram despolpadas, representando 53% de todo o abastecimento naquele período.

Nos últimos anos, a procura pelo tucumã beneficiado aumentou significativamente. Uma das questões mais importantes quando se trata do comércio da polpa *in natura* é a quantidade que cada fruto possui (DIDONET, 2012). Com base nessa característica, os produtores de tucumã avaliam empiricamente qual variedade é mais viável ao despolpamento e qual é melhor para venda do fruto inteiro, pois muitas vezes a quantidade de polpa não é proporcional ao tamanho que o fruto apresenta.

Diante dos dados apresentados, a avaliação quantitativa da polpa por meio de outras análises também foi pertinente, auxiliando na indicação de qual variedade possui o maior número de unidades com elevada quantidade de polpa. A informação sobre a variedade do tucumã com o maior potencial produtivo é importante ser obtida, pois pode auxiliar na escolha da variedade de tucumã ideal para ser empregada em cada atividade econômica.

1.2 Objetivos da pesquisa

1.2.1 Objetivo Geral

Gerar e avaliar modelos computacionais capazes de classificar e agrupar quatro variedades de tucumã encontradas no município de Itacoatiara-AM.

1.2.2 Objetivos Específicos

- Analisar o fruto do tucumã a fim de extrair características relevantes para a formação dos conjuntos de dados;

- Comparar os modelos computacionais induzidos em cada técnica com diferentes níveis de parâmetros dos algoritmos;

- Avaliar a capacidade de predição dos modelos na classificação e agrupamento das variedades de tucumã;

- Apontar os melhores atributos preditivos de variedades de *Astrocaryum aculeatum*;

- Indicar os melhores modelos e parâmetros encontrados em cada técnica aplicada na pesquisa;

- Analisar e estimar o potencial de produtividade comercial de cada variedade de tucumã.

1.3 Organização da dissertação

Este trabalho foi estruturado em capítulos formados por subseções explicativas sobre cada parte do estudo. Além deste capítulo introdutório, mais seis outros capítulos estão organizados da seguinte forma:

Capítulo 2: REVISÃO DE LITERATURA

A revisão bibliográfica é apresentada nesse capítulo com intuito de estabelecer o nivelamento do conhecimento sobre os conceitos relacionados a esta pesquisa. Todo o arcabouço teórico necessário à compreensão deste estudo encontra-se descrito detalhadamente nessa parte do trabalho.

Capítulo 3: TRABALHOS RELACIONADOS

Nesse capítulo é realizada uma descrição geral do uso das técnicas de AM, elucidando os principais tipos de pesquisas acadêmicas que são realizadas dentro desse ramo da Inteligência Artificial. Também são apresentadas algumas contribuições de trabalhos com uso das mesmas técnicas aplicadas nesta pesquisa, dando embasamento para a compreensão acerca da relevância do uso de técnicas de AM na criação de modelos computacionais preditivos.

Capítulo 4: A ESPÉCIE Astrocaryum aculeatum G. Mey. (TUCUMÃ DO AMAZONAS)

Para o entendimento das características gerais da espécie estudada, esse capítulo descreve os aspectos mais relevantes do tucumã, como: principais espécies e suas características, regiões onde ocorrem, épocas de frutificação, cidades produtoras, entre outros. Além destes conceitos descritos, também são elencadas algumas importantes contribuições de pesquisas científicas em diferentes áreas utilizando o tucumã como objeto de estudo.

Capítulo 5: METODOLOGIA

Esse capítulo abarca todo o procedimento metodológico aplicado para o alcance dos objetivos deste trabalho, sendo apresentadas minuciosamente todas as tarefas manuais e computacionais realizadas no período de duração da pesquisa. Ao longo da explanação de cada atividade, todos os recursos envolvidos também foram abordados de forma ampla, contribuindo ainda mais na compreensão das ferramentas utilizadas.

Capítulo 6: RESULTADOS E DISCUSSÕES

Nessa seção estão presentes todos os resultados obtidos por meio da metodologia adotada para execução da pesquisa. As tabelas, gráficos, figuras e quadros

mostrados no capítulo são comentados criteriosamente, para elucidar a relação dos resultados com os objetivos proposto. A cada subseção uma discussão é feita para mostrar o que se pôde descobrir em relação ao uso dessas três técnicas de AM no domínio estudado.

Capítulo 7: CONCLUSÃO

Esse capítulo apresenta as impressões sobre o desenvolvimento da pesquisa, estimando a relevância que o estudo representou em relação aos objetivos alcançados. São também apresentadas algumas limitações que foram identificadas, assim como os trabalhos futuros e considerações finais a respeito deste estudo.

2 REVISÃO DE LITERATURA

2.1 Aprendizado de Máquina e o Reconhecimento de Padrões

A aprendizagem é a principal característica da inteligência humana, ela representa os meios básicos para a obtenção de conhecimento. De acordo com Hua *et al* (2009), o processo de aprendizagem humana integra a memória, o pensamento, a percepção, o sentimento, e outras atividades mentais relacionadas. Comparado à aprendizagem humana, o aprendizado de máquina é mais rápido, o acúmulo de conhecimento é facilitado e os resultados da aprendizagem são mais fáceis de demonstrar. Todavia, esse processo depende diretamente da ação humana, isto implica que todo o progresso do ser humano no campo de AM, vai aumentar a capacidade dos computadores em aprender, auxiliando no melhoramento do processamento das informações.

A aprendizagem é a atividade que processa a informação do lado de fora para dentro. Primeiro, obtém-se as informações do ambiente externo, em seguida, estas são processadas para gerar o conhecimento, que posteriormente é armazenado em um repositório, guardando muitos princípios gerais que norteiam uma parte da ação de execução. Devido o ambiente fornecer todos os tipos de informações para o sistema de aprendizagem, a qualidade dessas informações impacta diretamente na aprendizagem, determinando se ela será fácil e organizada ou difícil e desordenada (HUA *et al*, 2009).

O reconhecimento automático de padrões é a identificação e atribuição de classes de objetos por meio de máquinas. Os padrões apresentados para a identificação das classes podem ser de origem visual, oral ou eletromagnética (ABRAMSON *et al*,

1963). O estudo do reconhecimento automático de padrões demanda a compreensão dos conceitos de como ocorre o aprendizado de máquina.

2.1.1 Características do Método Indutivo

Hua *et al* (2009) apontam o método indutivo como uma das principais formas de aprendizagem. Este método baseia-se em exemplos concretos suficientes para generalizar conceitos e identificar grupos de características semelhantes. A aprendizagem indutiva é um método em que se aplica o conceito de consequência indutiva, identificando se o processo conta com a orientação de um "professor" ou não. Esse aprendizado indutivo pode ser categorizado em aprendizado por meio de exemplos (aprendizado com professor) e aprendizado por observação (aprendizado sem professor) (FANG, 2006).

Formalmente, um sistema de reconhecimento de padrões, no contexto de aprendizado de máquina, é responsável por associar classes (geralmente em forma de rótulos) a objetos. Classe é o nome dado a um conjunto de objetos com as mesmas características. Objeto é o nome dado a um conjunto de medidas chamadas de características ou atributos (KUNCHEVA, 2004).

No que concerne a aprendizagem indutiva, Tan *et al* (2006) descrevem a separação dos métodos de reconhecimento de padrões em dois grupos principais:

- Métodos **Supervisionados**, nos quais o algoritmo deve passar por uma etapa chamada de treinamento, de forma que o classificador escolhido aprenda um determinado padrão para o tipo de dados do cenário trabalhado, baseando-se em uma parte dos dados chamada conjunto de treinamento. Nesses métodos a ação a ser realizada é a Classificação;

- Métodos **Não Supervisionados**, nos quais o algoritmo não tem nenhuma informação prévia sobre as classes a que os objetos pertencem. Nesses métodos a ação a ser realizada é o Agrupamento.

Os métodos de reconhecimento de padrões estão relacionados ao sistema de extração de características utilizado para representar os objetos. Para Deng & Li (2013), quanto melhor é o método utilizado para se extrair as características e dessa forma representar os objetos, mais trivial pode ser a forma abordada para o reconhecimento de padrões. No entanto, representações pobres dos objetos podem exigir um reconhecimento de padrões mais robusto. Por isso, para a execução desta pesquisa foi adotado um conjunto considerável de variáveis detalhadas, o que permitiu posteriormente uma avaliação da importância desses atributos no domínio estudado.

O esquema geral de uma possível configuração para sistemas de reconhecimento de padrões foi apresentado por Carvalho (2010):

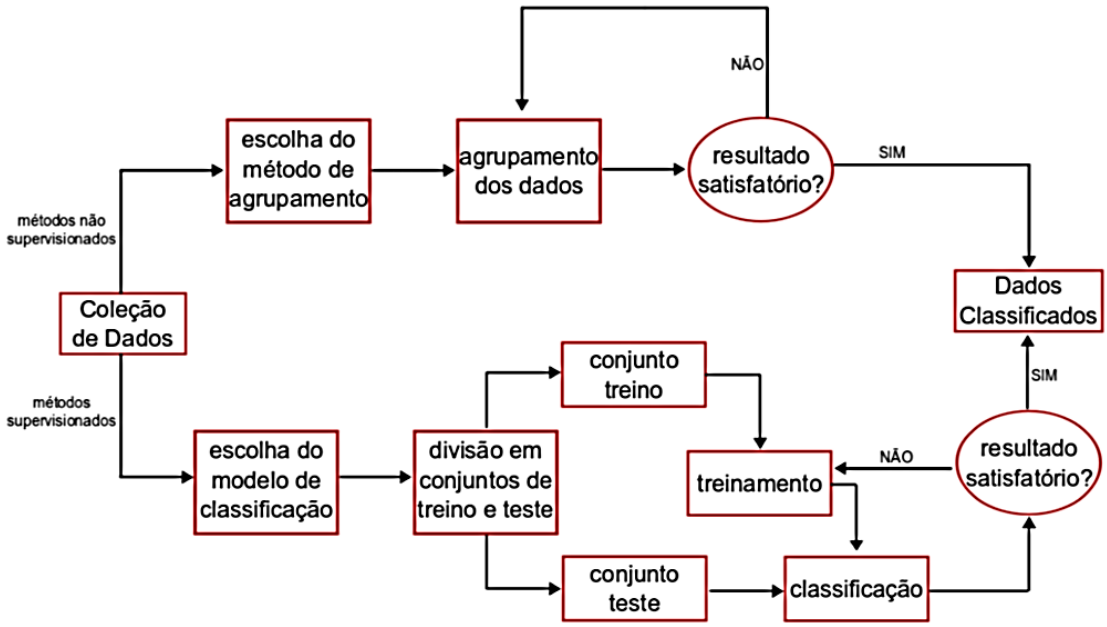

Figura 1 - Configuração de um sistema de reconhecimento de padrões (CARVALHO, 2010)

Na disciplina de AM, o interesse essencial gira em torno do funcionamento dos *agrupadores* e *classificadores*. Um *agrupado*r é uma ferramenta que permite o

particionamento dos dados em conjuntos cujos elementos compartilham características comuns. Já um *classificador* provê o mapeamento entre um espaço de características ou dados de entrada *X* para um conjunto discreto de rótulos *Y* (STROEH, 2009).

O enfoque desta pesquisa concentra-se nas tarefas de classificação e agrupamento de objetos. Para o desenvolvimento das atividades relacionadas a esses processos foram escolhidas três técnicas de AM para induzir os modelos: os métodos Árvores de Decisão (AD) e Redes Neurais Artificiais (RNAs) para a classificação, e o método K-Médias para o agrupamento das variedades.

No que concerne o aprendizado supervisionado, Árvores de Decisão é, em teoria, a técnica de AM mais estudada em aplicações práticas. Algumas características principais baseiam essa preferência, tais como: possui suporte a diversos tipos de atributos (categóricos e numéricos), sua representação do conhecimento adquirido é facilmente compreendida, e o seu processo de aprendizado e treinamento é relativamente rápido, comparado a outros algoritmos (WITTEN & FRANK, 2005). Quanto a Redes Neurais Artificiais, essa técnica é amplamente utilizada para resolver problemas complexos, pois oferece um alto poder de processamento (WITTEN & FRANK, 2005). Apesar de seu custo computacional ser relativamente elevado, os algoritmos de RNAs têm sido empregados nas mais diversas pesquisas científicas. Em outro viés, no contexto do aprendizado não-supervisionado, a técnica K-Médias é uma das melhores para realizar a tarefa de agrupamento de objetos (HARTIGAN, 1975). Essa técnica é a mais utilizada, devido sua implementação ser simplificada e sua dinâmica basear-se em uma função que permite obter bons resultados em grupos isolados e compactos (JAIN & DUBES, 1988).

Diante do exposto, a escolha dessas três técnicas com características diferentes é importante para efeito de comparação, indicando qual a relevância de cada uma no domínio estudado.

2.2 Técnicas de AM utilizadas na pesquisa

2.2.1 Árvores de Decisão (AD)

Essa técnica é largamente utilizada em pesquisas científicas devido às suas aplicações práticas. Além de simplificada, a AD apresenta algumas vantagens em relação às demais, como por exemplo: a facilidade na interpretação, a organização e o baixo custo computacional (BREIMAN *et al*, 1984).

AD é um método robusto a ruídos que usa aproximação de funções discretas podendo aprender expressões disjuntivas. Os algoritmos de árvores de decisão realizam uma busca do tipo *top-down* no universo de dados para estimar todas as árvores possíveis. Essa técnica utiliza a entropia (medida da pureza do conjunto de instâncias) para realizar o cálculo da razão de ganho, penalizando os atributos com muitos valores possíveis (MITCHELL, 1997). Quando uma árvore apresenta perda na aprendizagem devido a sua alta complexidade, é necessário realizar a poda. A poda da árvore é o método mais utilizado para reduzir uma AD, garantindo que a mesma seja a mais generalista possível, ou seja, ela visa um ótimo global ao invés de ótimo local. Isto quer dizer que ocorre a busca pela melhor predição em cada nó, na esperança de que juntos eles possam realizar uma boa classificação final com base nos atributos (BREIMAN *et al*, 1984).

A interpretação das árvores de decisão é simples. A partir de um conjunto de dados de treinamento, essa técnica tem como objetivo criar um modelo que consiga identificar a qual classe um determinado objeto pertence. Para que essa tarefa de classificação seja eficiente, espera-se que o conjunto de dados usado atenda algumas condições (QUINLAN, 1993):

- Os valores dos atributos preditivos devem possuir um conjunto finito. Caso os atributos possuam muitos dados contínuos, alguns métodos podem ser adotados para transformar esses valores em discretos;
- O conjunto de dados de treinamento deve possuir uma quantidade finita de classes pré-definidas;
- Para a construção do modelo é preciso ter instâncias suficientes, caso contrário pode-se gerar um aprendizado tendencioso, uma vez que poucos exemplos não ajudam o modelo a generalizar todas as regras necessárias.

Considerando-se que todas as condições acima sejam atendidas, o método de árvores de decisão possui a seguinte dinâmica: de início, todos os objetos de treinamento são avaliados e o atributo que melhor realiza a separação de classes é escolhido para ser o nó raiz, gerando *n* ramificações a partir dos valores que cada nó raiz pode assumir. Cada ramo contém um conjunto de objetos atribuídos de acordo com o valor do atributo testado.

Uma ramificação de uma árvore pode conduzir a um nó ou a uma folha. Os pontos intermediários das árvores são chamados de nós e os pontos finais de cada ramo são chamados de folhas (CHEN *et al,* 2003), um exemplo destas árvores de decisão é apresentado na Figura 2.

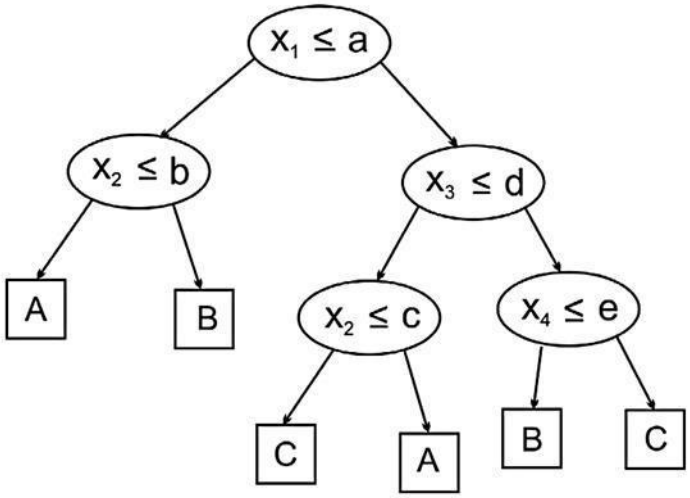

Figura 2 - Esquema de uma árvore de decisão com quatro variáveis e três classes (ZHOU *et al*, 2004)

As árvores de decisão são capazes de processar grandes volumes de dados sem precisar de alto nível de pré-processamento ou transformação dos mesmos. Além disso, são capazes de relacionar variáveis discretas e contínuas, validando os resultados por meio de métodos estatísticos (MONARD & BARANAUSKAS, 2005; SARMENTO, 2010).

Segundo Witten & Frank (2005), as árvores de decisão representam uma implementação da estratégia de "dividir para conquistar" aplicada ao problema de aprendizado de máquina. Uma árvore de decisão toma por entrada um objeto descrito por um conjunto de propriedades e retorna uma decisão do tipo Sim/Não. Outras saídas são possíveis, porém a configuração mais típica de uma árvore de decisão é de uma função booleana. Cada nó da árvore de decisão corresponde a um teste aplicado sobre uma propriedade do objeto de entrada, sendo que as arestas que ligam aos outros nós são rotuladas com os possíveis resultados do teste. As folhas das árvores contêm os valores booleanos a serem retornados quando as mesmas são alcançadas. Desta forma, cada nó provoca o particionamento do conjunto de entrada segundo o atributo testado (STROEH, 2009).

O processo de aprendizado materializa-se na técnica usada na construção da árvore de decisão. A pergunta reside em qual propriedade selecionar para estabelecer o próximo nó da árvore. Dado um nó *i*, seja $y \in 1,...,n$ um dos valores possíveis para o mesmo, e *f(i,j)* a probabilidade de se obter o valor *j* no nó *i*, então *f(i,j)* corresponde à proporção dos registros associados ao nó *i* para os quais $y = j$ (MITCHELL, 1997).

Toda árvore de decisão atribuirá classes aos objetos de acordo com sua proporção no conjunto de dados. No caso de uma amostra de objetos que pertencem a somente duas classes, por exemplo, *P* e *N*, um objeto qualquer pertencerá à classe *P* com probabilidade de *P/(P+N)* e à classe *N* com probabilidade de *N/(P+N)*. Neste sentido, quando uma árvore de decisão é usada para classificar um objeto, a mesma atribui a ele uma classe. Dessa forma, ela pode ser considerada como uma fonte de mensagem *P* ou *N* capaz de indicar a classe do objeto por meio da Função 1 (HAN & KAMBER, 2001):

$$I(p,n) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) \\ \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right) \qquad (1)$$

Se o atributo *A* com os valores *[A1, A2, ... , An]* é aplicado como raiz da árvore, ela dividirá todo o conjunto de dados em *C* partes, *C[C1, C2, ..., Cn]*, onde *Ci* contém aqueles objetos em *C* que possuem valores *Ai* de *A.* Considerando que *Ci* contém *pi* objetos da classe *P*, e *ni* da classe *N*, a informação necessária prevista para construção da sub-árvore para *Ci* é *I(pi, ni)*. Dessa forma, a Função 2 é necessária para a construção da árvore usando *A* como nó raiz (HAN & KAMBER, 2001):

$$E(A) = \sum_{i=1}^{v}\left[\frac{p_i + n_i}{p+n}\right] I(p_i, n_i) \qquad (2)$$

Esta função é obtida por meio de média ponderada em que o peso para o *i-ésimo* elemento é proporcional aos objetos em *C* que pertence a *Ci.* A informação ganha pela ramificação sobre *A* é representada pela Função 3 (HAN & KAMBER, 2001):

$$G(A) = I(p,n) - E(A) \quad (3)$$

Os algoritmos de classificação em AD realizam a verificação dos atributos com base nas funções descritas acima para determinar como nó raiz o melhor atributo preditivo, e logo após, executam a mesma função recursivamente para determinar as demais sub-árvores.

As primeiras versões dos algoritmos de AD eram limitadas a parâmetros discretos, mas os algoritmos sofreram modificações ao longo da evolução do *software* WEKA, permitindo que nas versões atuais sejam trabalhadas bases de dados com parâmetros contínuos. Este é o caso do algoritmo *J48*, que além dessas características, ainda realiza a poda automática para garantir melhores resultados. Por esses motivos, o *J48* foi selecionado para realizar as modelagens neste trabalho.

2.2.2 Redes Neurais Artificiais (RNAs)

Atualmente, as Redes Neurais artificiais (RNAs) têm se tornando um amplo campo de pesquisa na área de IA. As RNAs nos permitem projetar sistemas não-lineares que podem assumir um grande número de entradas gerando um relacionamento do tipo entrada-saída (HAYKIN, 1999). Entre suas vantagens destaca-se a capacidade de aprender exemplos e generalizar conceitos. Essas características estão relacionadas à capacidade de aprender através de um conjunto reduzido de exemplos e mesmo assim dar respostas coerentes na classificação de novas instâncias desconhecidas (BERTHOLD & DIAMOND, 1995).

O estudo das redes neurais artificiais foi inspirado em parte pela observação do sistema de aprendizagem biológico, o qual é constituído de teias muito complexas de neurônios interligados. A filosofia básica das RNAs é a construção de uma teia interligando várias unidades simples, onde cada unidade leva um número de entradas reais (possivelmente as saídas de outras unidades) e produz uma única saída real (que pode se tornar a entrada para muitas outras unidades) (MITCHELL, 1997).

O método de aprendizagem de rede neural fornece uma abordagem eficaz para certos tipos de problemas. Aprender a interpretar os dados coletados do mundo real através de sensores é uma tarefa difícil. Neste sentido, as redes neurais artificiais estão entre os métodos de aprendizagem mais eficazes para solucionar problemas complexos (HAYKIN, 1999).

Os algoritmos de RNAs têm se mostrado robustos quanto à tolerância a erros na classificação de dados e são aplicados com sucesso em sistemas de reconhecimento de fala e escrita, interpretação de cenários visuais e estratégias de controle da máquina de aprendizagem (MITCHELL, 1997).

A primeira estrutura de RNAs foi desenvolvida por McCulloch & Pitts (1943), esse modelo é mais conhecido como neurônio MCP, ou *Perceptron* Simples.

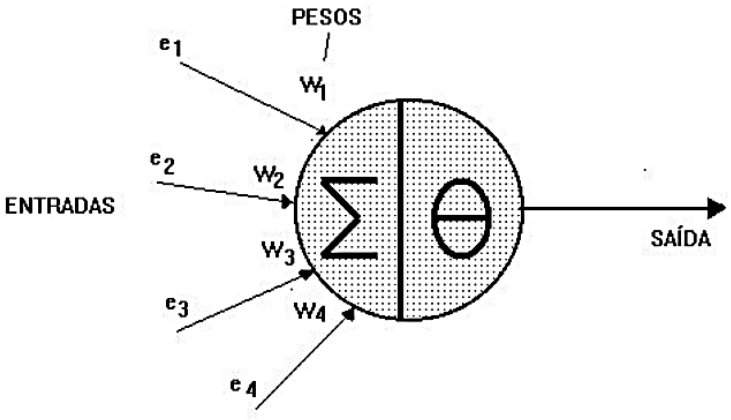

Figura 3 - Modelo de um neurônio MCP (MCCULLOCH & PITTS, 1943)

Sua estrutura é similar a um neurônio biológico e possui *n* terminais de entrada (x_1, x_2, ..., x_n) e um terminal de saída. Em comparação com um neurônio humano, os terminais de entrada correspondem aos dendritos e o terminal de saída corresponde ao axônio para emular as sinapses. Os terminais de entrada têm pesos (w_1, w_2, ..., w_n) associados a eles.

Para a ativação de um neurônio artificial MCP é preciso a aplicação de uma função linear cuja saída pode assumir 0 ou 1. Dependendo do valor ponderado das entradas, o nodo (neurônio) irá ativar sua saída seguindo a Função 4 (BRAGA, 2000):

$$\sum_{i=0}^{n} x_i \, w_i \geq \theta \tag{4}$$

Nesta função de ativação do MCP, *n* é o número de entradas do neurônio, w_i é o peso associado à entrada x_i, e θ é o limiar (*threshold*) do nodo. Existe uma simplificação no modo de disparo de cada camada realizada nesse modelo, onde todos os neurônios são avaliados ao mesmo tempo sendo disparados de maneira síncrona. O mesmo não ocorre no sistema biológico, já que não existe um mecanismo para realizar esse sincronismo.

Após a criação do primeiro modelo de neurônio artificial foram surgindo vários outros modelos que permitem a produção de saídas que não sejam obrigatoriamente 0 e 1. Estes modelos são baseados em diferentes funções de ativação. Para melhor exemplificação toma-se como ponto de partida a equação: $y=\alpha x$, onde *y* é a saída, *x* a entrada, e α é um número real que define a saída linear para os valores de entrada. A seguir, temos alguns exemplos de funções de ativação para redes neurais artificiais (BRAGA, 2000):

1. *Função degrau*: essa função tem como valores de saída 0 ou 1 e é definida como:

$$f(x) = \begin{cases} 1, se\ x \geq 0 \\ 0, se\ x\ < 0 \end{cases} \quad (5)$$

2. *Função rampa*: onde 0 e 1 são os limites da função e (-½ e ½) é o intervalo que define a saída linear:

$$f(x) = \begin{cases} 1, se\ x \geq \frac{1}{2} \\ x, se -\frac{1}{2} < x < \frac{1}{2} \\ 0, se\ x \leq -\frac{1}{2} \end{cases} \quad (6)$$

3. *Função sigmóide*: nessa função os valores pertencem a um intervalo contínuo, por exemplo, entre 0 e 1, onde α determina a inclinação da função:

$$f(x) = \frac{1}{1 + \exp(-\alpha x)} \quad (7)$$

Outro conceito importante além da função de ativação é a arquitetura das RNAs, cuja configuração é muito importante, uma vez que determina quais tipos de problemas podem ser tratados pela rede (BRAGA, 2000). Quanto às conexões entre os nodos nas camadas das RNAs podemos ter dois tipos:

1. *Feedforward*, ou acíclica: a saída de um neurônio em uma camada não pode ser utilizada como entrada em nenhuma camada anterior a ela;

2. *Feedback*, ou cíclica: a saída de algum neurônio de uma certa camada é utilizada como entrada para uma camada anterior a ela.

Diante do exposto, compreendemos que as redes MCP tratam apenas de problemas linearmente separáveis, pois possuem apenas uma camada (MINSKY & PAPERT, 1969). Neste cenário, para resolver problemas não-linearmente separáveis,

foram criadas as redes MLP (*MultiLayer Perceptron*). Este tipo de rede neural possui no mínimo duas camadas que permitem a aproximação de qualquer função contínua. As MLPs advêm do modelo de *perceptrons* proposto por Frank Rosenblatt em 1958 (HAYKIN, 1999). Cada neurônio de uma rede MLP representa um nodo de processamento.

Como já anteriormente discutido, um dos aspectos primordiais das RNAs é a função de ativação. No caso das MLPs a função mais empregada é a *sigmoidal logística*, representada na Figura 4:

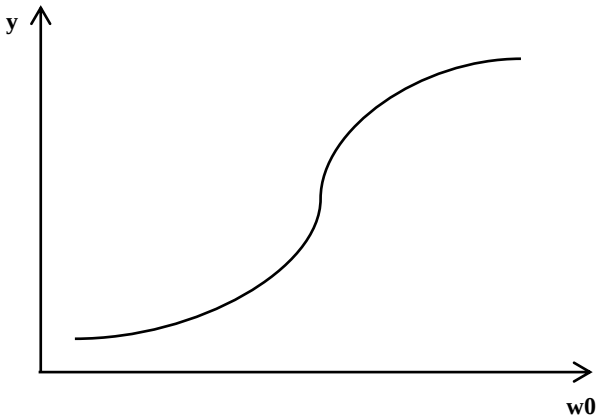

Figura 4 - Gráfico representativo da função *sigmoidal logística* (BRAGA, 2000)

Seguindo os parâmetros da função de ativação, uma rede multicamada realiza o processamento em cada nodo através da combinação dos processamentos realizados pelos nodos anteriores a este. Exemplificando este conceito, veremos a descrição dos nodos de uma rede com duas camadas intermediárias:

- *Primeira camada intermediária*: cada nodo traça retas no espaço de padrões de treinamento. Essas retas são criadas com base na função de ativação da camada e sua orientação é baseada no vetor de pesos;

- *Segunda camada intermediária*: cada nodo combina as retas traçadas pelos nodos da primeira camada intermediária, formando regiões convexas, onde o número de lados é definido pelo número de unidades conectadas a ele.

- *Camada de saída*: cada nodo forma regiões que são combinações das regiões convexas definidas pelos nodos da segunda camada conectados a ele.

As camadas intermediárias de uma rede MLP funcionam como extratores de características. As saídas da rede são definidas por meio de representações internas dos padrões de entrada gerados. O número de nodos contido em cada camada é definido empiricamente, esse número depende da distribuição dos padrões para treinamento e do método de validação da rede. A quantidade ideal de neurônios em uma MLP depende de vários fatores, entre os quais podemos citar (HAYKIN, 1999; MUKKAMALA *et al*, 2002):

- Complexidade da função a ser aprendida;

- Número de exemplos de treinamento;

- Quantidade de ruído presente nos exemplos;

- Distribuição estática dos dados de treinamento.

A alocação de neurônios intermediários deve subsidiar a solução do problema em um domínio específico. Porém, é preciso ter cuidado com a utilização de unidades em excesso, pois dessa forma o modelo pode apresentar sobreaprendizagem. Isto é conhecido como *overfitting*, significa que um modelo se especializou nos dados de treinamento, apresentando baixa taxa de acerto para dados desconhecidos (MITCHELL, 1997; MONARD & BARANAUSKAS, 2005). Por outro lado, se o número de neurônios nas camadas intermediárias for insuficiente, o tempo de execução até se encontrar uma solução ótima será muito elevado (JOO *et al*, 2003).

A forma mais eficaz de evitar o *overfitting* é estimar o erro de generalização durante o treinamento (MUKKAMALA *et al*, 2002). Para isso, a base de dados é dividida em dois conjuntos: o de treinamento e o de validação. O primeiro é utilizado na

atualização de pesos e o segundo é aplicado para estimar a capacidade de generalização da rede durante o processo de aprendizagem (HAYKIN, 1999).

Nesta pesquisa, o algoritmo de multicamadas utilizado foi o *MultiLayerPerceptron*, sua execução foi finalizada de acordo com as épocas de treinamento definidas nas propriedades do algoritmo, ou seja, a rede foi treinada por inteiro quantas vezes necessário, na seção 5 (metodologia do trabalho) são apresentados mais detalhes sobre essa configuração.

2.2.3 K-Médias

As técnicas de aprendizado de máquina não-supervisionado realizam a tarefa de agrupamento de objetos por meio de suas características. Esta dinâmica é chamada de *clusterização*. Representa uma forma de aprendizado auto-organizável, dispensando a presença de um "professor" que indique a associação das classes aos objetos (JAIN & DUBES, 1988).

A aprendizagem não supervisionada tem como objetivo extrair informações relevantes de dados não rotulados. Em seu escopo mais amplo, encontra-se a prática de definir medidas de similaridade entre dois ou mais *clusters*, assim como um critério global que pode ser, por exemplo, a soma do erro quadrático na tarefa de agrupamento (FACELLI, 2006).

As abordagens mais comuns de agrupamento são descritas em dois tipos: os *métodos hierárquicos* e os *métodos particionais*. Nos métodos hierárquicos o conjunto de dados é particionado várias vezes formando uma estrutura conhecida como *dendograma*, que representa a aglomeração dos nodos de acordo com a avaliação dos atributos preditivos. Esses métodos precisam de uma matriz que represente todas as

medidas das distâncias entre os agrupamentos formados, esta matriz é conhecida como *matriz de similaridade* (TAN *et al*, 2006).

Em outro viés, existem os métodos particionais, os quais realizam a divisão do conjunto de dados em *clusters* não interseccionados. Este tipo de particionamento garante que um objeto faça parte de apenas um dos grupos, impedindo também que ocorra a formação de sub-grupos, o que caracterizaria a dinâmica de um método hierárquico (TAN *et al*, 2006).

Comparando os dois tipos de métodos descritos, os métodos particionais apresentam a vantagem de trabalhar com conjunto de dados muito maiores, isto é devido ao seu baixo custo computacional. A sua principal desvantagem é a necessidade de informar o número de *clusters* a serem formados antes da execução do algoritmo, isso pode implicar em má interpretação dos resultados. Porém, se houver um bom conhecimento do domínio estudado, podem ser inferidas muitas informações relevantes com uso desse método.

A técnica K-Médias representa um método particional exclusivo, alocando um objeto em um único *cluster*. O algoritmo *SimpleKMeans* presente na biblioteca do WEKA é iterativo e muito empregado em diversos tipos de problemas de *clusterização*. Sua heurística realiza uma busca local baseada em aprendizado competitivo para minimizar a função de custo a partir de um conjunto inicial de centróides (HARTIGAN, 1975).

O objetivo dessa técnica é encontrar a melhor divisão de X dados em K grupos C_i, onde $i = 1, ..., K$, de forma que a distância dos entre os dados pertencentes a um grupo e seu respectivo centro seja minimizada (LLETÍ *et al*, 2004). Essa dinâmica consiste em usar os h primeiros casos de um conjunto de dados, para extrair valores que servirão de estimativas temporárias das médias dos K *clusters*, onde K é o número de

clusters especificado pelo usuário. Dessa maneira, o centro do *cluster* inicial é definido para cada caso baseado nos dados mais próximos. Posteriormente, esses pontos são comparados com os pontos mais distantes e com os outros *clusters* formados. A partir dessa tarefa inicial, um processo de atualização contínua interativa é executado, a fim de encontrar os centros dos *clusters* finais (HOLMES *et al*, 1994).

Seguindo esse princípio, o algoritmo aloca aleatoriamente os *X* pontos a *K* agrupamentos, calculando as médias dos vetores de cada grupo. Logo após, cada ponto é deslocado para o grupo ao qual seu vetor médio possui valor mais próximo. Com essa nova configuração dos pontos nos *K* grupos, outros vetores médios são calculados culminando na execução cíclica desse processo, até que todos os pontos se encontrem nos seus vetores médios mais próximos (HARTIGAN, 1975).

O critério de custo a ser minimizado é definido em função da distância dos elementos em relação aos centros dos agrupamentos. Geralmente, este critério é a soma residual dos quadrados das distâncias, ou seja, é a soma dos quadrados das distâncias dos elementos ao centróide do seu *cluster*. Para minimizar a soma do erro quadrático sobre todos os grupos é necessário atender a três parâmetros específicos: o número de grupos, a inicialização do grupo e a métrica da distância. O erro quadrático entre μ_k e os pontos no grupo C_k são definidos pela Função 8 (LOTZ *et al*, 2004):

$$J(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \qquad (8)$$

onde $X = x_i$ ($i = 1, ..., m$) é o conjunto de m pontos d-dimensionais; $C = c_k$, ($k = 1, ..., k$) é o conjunto de k *clusters*; e μ_k é a média de *clusters* C_k. Neste caso, como o objetivo é minimizar a soma do erro quadrado sobre todos os *clusters*, a Função 8 é redefinida, dando origem à Função 9:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (9)$$

Em resumo, cada agrupamento é representado pelo centro do grupo e cada padrão é atribuído ao agrupamento que está mais próximo. O procedimento geral pode ser descrito em poucos passos (FUNG, 2001):

1. Inicializar as médias das *k* partições;

2. Determinar para cada padrão a partição mais próxima;

3. Calcular a média de cada partição;

4. Se houver mudança na média das partições, voltar ao passo dois;

5. Resultado: a média das *k* partições.

O resultado do processamento desse método pode ser drasticamente afetado pela escolha das condições inicias. Porém, se houver uma base de dados bem estruturada, espera-se a convergência para um mínimo global. O bom desempenho do algoritmo depende muito da escolha adequada da medida de distância e do ponto inicial de partida do algoritmo (JAIN *et al*, 1999; KAINULAINEN, 2002).

2.2.3.1 Medidas de Distância

De acordo com Witten & Frank (2005), os métodos particionais de *clusterização* têm relação direta com diversas áreas que baseiam sua concepção, como por exemplo, a estatística, a matemática, a geometria, entre outras. Uma boa definição dos *clusters* depende primordialmente das medidas de distância aplicadas ao algoritmo, sejam elas de similaridade ou dissimilaridade. Na primeira, o objetivo é definir o grau de semelhança entre as instâncias e realizar o agrupamento de acordo com

a sua coesão; e na segunda realizar as mesmas tarefas, mas baseando-se nas diferenças dos atributos das instâncias.

Com relação às modelagens, as medidas de distância influenciam no custo computacional, na complexidade e na representação gráfica para a análise do modelo. Dependendo da medida de distância aplicada a um domínio específico, a identificação de *outliers* (objetos com valores muito discrepantes), o formato dos *clusters* e a formação de vizinhança entre os grupos de dados podem ser diferenciados (WITTEN & FRANK, 2005).

Para efeito da realização do trabalho, as medidas de distâncias que serão apresentadas nesta dissertação são a *Distância Euclidiana* e a *Distância de Manhattan*. A distância Euclidiana (DE) é definida por meio da raiz quadrada da soma dos resultados de cada subtração entre *x* e *y* ao quadrado em suas respectivas dimensões, executando os cálculos pela Função 10 (JAIN *et al*, 1999):

$$DE = \sqrt{(x1 - y1)^2 + (x2 - y2)^2 + \cdots + (xz - yz)^2} \qquad (10)$$

Já a distância de *Manhattan* (DM) possui uma definição mais simplificada, na qual é realizada apenas a soma das diferenças entre *x* e *y* em todas as dimensões, sendo o cálculo dessa medida de distância baseado na Função 11(JAIN *et al*, 1999):

$$DM = |x1 - y1| + |x2 - y2| + \cdots + |xh - yh| \qquad (11)$$

Para melhor compreensão acerca da dinâmica de cada distância, tomaremos como exemplo uma rota de GPS para um helicóptero e um carro partindo de um ponto a outro de uma cidade. A *Distância Euclidiana* seria a segmentação de uma reta que indicaria uma possível rota para o helicóptero, na qual não haveria preocupação com obstáculos por ser um veiculo aéreo (geometricamente seria a hipotenusa de um triângulo); e a *Distância de Manhattan* seria a segmentação de retas verticais e horizontais em um mapa, para indicar a rota do carro, já que este precisa seguir a

orientação das ruas entre os quarteirões (geometricamente representaria a soma dos catetos) (WITTEN & FRANK, 2005).

Diante do exposto, Para efeito da execução desta pesquisa foram selecionadas estas duas distância para realizar os treinamentos e testes dos modelos usando o algoritmo *SimpleKMeans*.

2.3 Medidas de Desempenho

Para garantir uma correta indicação de qual algoritmo teve melhor desempenho é necessário que os resultados dos modelos possam ser avaliados e comparados. Nesse contexto, os métodos existentes mais relevantes para determinar a acurácia de um modelo são:

1. Teste e validação: a avaliação do desempenho de um modelo depende de sua validação e posteriormente de um teste. Validar um modelo quer dizer avaliá-lo em relação a sua capacidade de generalização de conceitos a partir de um conjunto de dados. No processo de validação também são realizados testes para averiguar as taxas de acerto obtidas após o treinamento, tendo como resultado a estimativa de quão preciso é este modelo na prática. Porém, essa validação é feita com base no conjunto de dados de treinamento, sendo necessário realizar um novo teste do modelo submetendo a ele novos dados desconhecidos (MITCHELL, 1997). Essa é a melhor forma de determinar o desempenho de um modelo depois de treinado e validado. Para realizar essas tarefas o WEKA oferece alguns recursos como: *Cross-validation* (validação cruzada), S*upplied test set* (conjunto de dados extras para teste), *Use training set* (utilização do mesmo conjunto de treinamento para execução dos testes), *Percentage Split* (separação de uma

porcentagem de instâncias do conjunto de treinamento para realizar os testes) (WITTEN & FRANK, 2005).

Não é necessário utilizar todos esses recursos. As duas principais maneiras de validar e testar modelos para determinar sua acurácia são: a validação cruzada e a utilização de um conjunto extra de dados para teste (MITCHELL, 1997).

2. Indicadores estatísticos: essas métricas auxiliam na análise dos resultados, como por exemplo, matriz de confusão, índice de correção e incorreção de instâncias mineradas, erro médio absoluto, erro médio relativo, precisão, *F-measure*, estatística *Kappa*, entre outros (WITTEN & FRANK, 2005).

Uma das funcionalidades principais do WEKA é a utilização das métricas citadas para quantificar o desempenho dos modelos. Essas medidas auxiliam na análise dos resultados, permitindo a compreensão do nível de aprendizado obtido. Nesta pesquisa, os indicadores utilizados para comparar os modelos são descritos a seguir:

- Matriz de Confusão: essa matriz oferece uma medida efetiva do modelo, nela são mostrados os números de classificações corretas e incorretas para cada classe de um conjunto de dados. Com base nos dados da matriz confusão é que são calculadas as demais métricas. Em um domínio com *n* classes a matriz de confusão construída será de *n x n*, e na diagonal dessa matriz estarão os valores de classificação corretas para cada classe (HOLMES *et al*, 1994).

- Taxa de acerto por classe: essa medida é conhecida como taxa de Verdadeiro Positivo (*true positive*). Seu cálculo baseia-se na relação entre a quantidade de objetos pertencentes a uma classe e a quantidade dos mesmos que são classificados corretamente como esta. Por outro lado, também existe a taxa de Falso Positivo (*false positive*), que indica a quantidade de objetos que não pertence àquela classe, mas são classificados como tal (WITTEN & FRANK, 2005). Entre as duas medidas, a mais

interessante sem dúvidas é a taxa de Verdadeiro Positivo, pois permite analisar separadamente cada classe para verificar o nível de acerto na predição dos objetos em cada uma delas.

- Erro médio absoluto: para saber a precisão de um cálculo é necessário conhecer o valor proporcional da quantidade de uma medida em relação ao seu valor real. Quanto mais próximos esses números estiverem um do outro, maior será a exatidão de uma medida (DILWORTH, 1992). Neste contexto, o erro médio absoluto é uma das métricas mais comuns de erro de previsão. Essa medida não leva em consideração se um erro foi sobrestimado ou subestimado, sendo uma métrica que fornece a média dos erros cometidos pelos modelos de previsão durante uma dada quantidade de períodos de treinamento. Para calcular o erro médio absoluto (EMA), subtrai-se o valor da previsão ao valor real em cada período de tempo, apresentando um resultado sempre positivo, em módulo, somando-se e dividindo-se pela quantidade de valores que foram usados para obter a soma. O erro médio absoluto pode ser entendido por meio da Função 12:

$$EMA = \frac{\sum_{t=1}^{n} |e_t|}{n} \tag{12}$$

onde *n* é número de períodos usados, e todo o numerador da função é chamado de soma corrente dos erros de previsão. O símbolo do módulo (| |) significa que o valor é absoluto, ignorando a direção do desvio. Das medidas estatísticas padrão, o erro médio absoluto é a métrica menos sensível a ruídos nos dados (WITTEN & FRANK, 2005).

- *Estatística Kappa*: Entre os indicadores de desempenho, a estatística *Kappa* possui grande relevância, pois é uma medida de confiabilidade para verificar a concordância entre as taxas de acerto alcançadas (COHEN, 1960). Para saber se uma dada classificação de um objeto é confiável, é necessário ter esse objeto classificado várias vezes por mais de um observador. A estatística *Kappa* é baseada no número de respostas concordantes, ou seja, no número de casos cujo resultado é o mesmo entre os

observadores (SIEGEL & CASTELLAN, 1998). Para chegar ao valor *Kappa* é realizado um cálculo para medir a concordância entre cada interobservador, estimando também o grau de concordância além do que se esperava pelo acaso. O valor do "acaso" pode ser, por exemplo, uma hipótese de que nenhuma das respostas é concordante, ou seja, *Kappa*=0. Os valores da estatística *Kappa* variam entre 0 e 1, onde o mais próximo de 0 significa o acerto por acaso e o mais próximo de 1 indica concordância exata da inferência dos valores pela técnica (FLEISS, 1981; CARLETTA, 1996).

2.3.1 Validação Cruzada (*cross-validation*)

A validação cruzada consiste em uma técnica que possibilita estimar a capacidade de generalização de um classificador (KOHAVI, 1995). Essa técnica divide o conjunto de treinamento em algumas partes mutuamente exclusivas. Uma dessas partes será o subconjunto a ser utilizado para validação ou teste. A cada execução do experimento esse conjunto vai mudando de acordo com o número de iterações definidas (HOLMES *et al*, 1994; PEÑA *et al*, 2005). O número padrão de iterações foi indicado por Witten & Frank (2005) após a realização de extensivos experimentos, os quais mostraram que 10 ciclos de validação cruzada são ideais para validar o modelo.

Esse método utiliza a base de dados em sua totalidade, gerando um resultado mais confiável, essa é a grande vantagem de sua utilização para validar modelos computacionais. O erro médio da validação cruzada é calculado realizando a média aritmética dos erros fornecidos por cada conjunto de testes (KOHAVI, 1995).

O *software* WEKA utiliza um método de validação cruzada chamado de *k-fold*. Este método realiza a divisão do conjunto total de dados em *k* subconjuntos mutuamente exclusivos do mesmo tamanho. A partir daí, um subconjunto é utilizado para teste e os

k-1 restantes são usados para estimar os parâmetros e realizar o cálculo da acurácia do modelo. Este processo é realizado *k* vezes de acordo com o número definido pelo usuário antes da execução do treinamento e teste dos modelos (WITTEN & FRANK, 2005).

2.3.2 Teste usando conjunto de dados extra (*Supplied test set*)

Como visto, a validação cruzada representa um método complexo e eficiente de medir a acurácia dos modelos preditivos em aprendizado de máquina. Apesar de a dinâmica desse método ser bastante confiável, ainda assim o teste de validação cruzada é feito usando os mesmos dados de treinamento. Durante os ciclos de validação, os subconjuntos de dados de teste são alternados *k* vezes, permitindo que em algum momento todos os dados sejam conhecidos pelo modelo (KOHAVI, 1995). Essa rotação é realizada com o objetivo de calcular a média de erro em todos os ciclos de validação, devido a isso os índices de desempenho desse método tendem a ser otimistas.

Neste sentido, para determinar a acurácia de forma ainda mais exata é necessário utilizar um conjunto de dados desconhecidos pelo modelo. Este conjunto irá ser submetido para que o classificador criado atribua classes aos objetos de acordo com os padrões generalizados no treinamento com dados diferentes (MITCHELL, 1997). Isto gera maior confiabilidade nos resultados, auxiliando na avaliação final dos modelos computacionais.

3 TRABALHOS RELACIONADOS

Nesta seção é apresentado um panorama geral dos trabalhos realizados na área de Aprendizado de Máquina. Esta discussão é importante, pois ajuda na compreensão de como as técnicas de AM são aplicadas em estudos científicos. Posteriormente, são expostos alguns trabalhos desenvolvidos com estrutura metodológica parecida com a utilizada no presente estudo, porém, aplicados a outros domínios, haja vista que não foram encontrados trabalhos com técnicas de AM utilizando dados de frutos amazônicos para realizar uma comparação.

3.1 Panorama do uso das técnicas de Aprendizado de Máquina

Na literatura são encontrados diversos trabalhos científicos com o uso de técnicas de AM para a realização de experimentos. Essas técnicas podem ser empregadas na solução de diferentes tipos de problemas práticos nos mais diversos domínios. O uso de ferramentas de AM oferece versatilidade no tratamento dos dados, o que explica por que muitos pesquisadores optam por utilizar técnicas de AM em suas pesquisas. Dependendo dos objetivos de um trabalho, existirão técnicas específicas que melhor se aplicam a cada situação. Neste contexto, os objetivos das pesquisas utilizando AM geralmente são pautados em avaliar técnicas para averiguar qual delas tem melhor desempenho em relação ao domínio estudado.

Os resultados dessas pesquisas se diferem muito uns dos outros. Isto é explicado pelo fato de que cada base dados é particular. A mesma técnica sempre apresentará resultados diferentes em cada trabalho, haja vista que a descoberta de

padrões é feita com base em um conjunto de dados único, particular a um domínio específico.

Existem diversas formas de se realizar trabalhos acadêmicos com algoritmos de AM. Entre os principais tipos de pesquisas científicas nesta área, dois merecem destaque. No primeiro, encontram-se os trabalhos de análise, construção ou melhoramento de algoritmos preditivos; e no segundo, os trabalhos que aplicam algoritmos existentes na descoberta de padrões e criação de modelos baseados em conjuntos de dados do mundo real. No contexto desse segundo tipo é que esta pesquisa foi baseada.

Diante dessa reflexão, neste capítulo são apresentados alguns trabalhos executados com bases de dados de diferentes domínios, mas que empregaram as mesmas técnicas utilizadas nesta pesquisa. O objetivo da apresentação dos mesmos é mostrar a relevância do uso de técnicas de AM nas mais diversas aplicações práticas. Não consta qualquer trabalho relativo à classificação e agrupamento de variedades de tucumã ou de outra fruta, uma vez que não foram encontrados registros deste tipo de trabalho durante o período de duração desta pesquisa.

3.2 Pesquisas científicas com uso de técnicas de Aprendizado de Máquina

Com relação à análise, construção e melhoramento de algoritmos de AM, dois trabalhos são descritos a seguir para exemplificar esse tipo de pesquisa. Vale ressaltar que estes dois primeiros trabalhos não estão diretamente relacionados com este estudo, porém é válido compreendê-los, pois deste tipo de pesquisa é que surgem os melhoramentos nos algoritmos usados em estudos como este apresentado na dissertação. Após a exposição dos mesmos são apresentadas pesquisas relacionadas com

este estudo, ou seja, trabalhos que tratam da aplicação de técnicas de AM em domínios específicos, discutindo os resultados em relação às metodologias adotadas em seus desenvolvimentos.

Muniz (2010) propôs resolver o problema de classificação binária por meio de um novo algoritmo, utilizando a combinação de árvores de decisão e algoritmo de programação inteira com o intuito de melhorar a execução de uma AD. O desempenho do algoritmo de combinação proposto foi comparado aos dos algoritmos-base separadamente, para averiguar se havia melhora após a hibridização. Para a realização das modelagens e testes o autor utilizou três bases de dados propostas por Quinlan (1987), Mangasarian & Wolberg (1990) e Kurgan *et al* (2001). Os treinamentos dos modelos foram feitos com as variáveis convertidas em diversos tipos para averiguar em quais cenários as três técnicas se sairiam melhor. Os resultados mostram que na maioria dos testes o algoritmo de programação inteira obteve melhores resultados sozinho, seguido do algoritmo híbrido proposto na pesquisa, ficando a árvore de decisão na última posição. Não foi possível indicar qual o melhor algoritmo para todas as bases de dados individualmente, pois mais testes precisam ser feitos. Apesar de o algoritmo de programação inteira ter se saído melhor separadamente, a solução de combinação proposta obteve melhores índices nos testes do que o algoritmo de árvores de decisão sozinho, ou seja, o objetivo de melhorar o desempenho da árvore foi atendido.

Em outra pesquisa, Matsubara (2008) investigou aspectos mais complexos sobre o aprendizado de máquina. Seu trabalho objetivou mostrar as relações existentes entre *ranking*, análise ROC e calibração em aprendizado de máquina. Durante a pesquisa, o autor investigou a viabilidade da criação de um algoritmo para ranking, e testou a análise ROC em diferentes aspectos para indicar os algoritmos e as melhores formas de utilizá-la. Como resultados da pesquisa, o autor encontrou uma forma comum

de representar resultados de *rankings* obtidos por meio de *Nayve Bayes* e árvores de decisão. Essa forma foi chamada de *ranking* lexográfico. Com base nessa descoberta foi criado do algoritmo *LexRank*, que apresenta a vantagem de obter a ordenação dos exemplos de classificação sem a necessidade de *scores*. Em relação à análise ROC, o resultado mais relevante é a descoberta de que o coeficiente angular de cada segmento do fecho convexo de uma curva ROC, é equivalente à razão de verossimilhança, a qual pode ser convertida na probabilidade a posteriori.

Trabalhos como esses têm grande relevância para área de inteligência artificial, pois contribuem para o melhoramento das técnicas e ferramentas que são utilizadas para a execução de pesquisas aplicadas. Na literatura, as pesquisa aplicadas são amplamente encontradas, haja vista que seus escopos tratam da investigação de informações relevantes sobre um determinado domínio real. Como já citado, este trabalho se encaixa nesse perfil. Para embasar seus objetivos norteadores, são apresentados alguns trabalhos executados com as mesmas técnicas escolhidas para uso neste estudo.

Em sua pesquisa Siviero & Hruschka Júnior (2011) aplicaram algoritmos de aprendizado máquina para classificação e agrupamento dos parâmetros mensurados numa seção de medidas no rio Atibaia/SP. A pesquisa objetivou prever a descarga sólida transportada no leito do rio. O desenvolvimento do estudo foi motivado pela importância do rio Atibaia na Bacia do Piracicaba/SP, sendo este o responsável pelo abastecimento de várias comunidades, além de ser o principal receptor das cargas difusas e pontuais da bacia. Os dados para modelagem computacional foram coletados entre o período de 03/1993 a 12/1994. No banco de dados original havia dados de área molhada (A) e perímetro molhado (P), porém os mesmos foram retirados por conter informação redundante. Deste modo foram utilizados somente os dados do raio hidráulico (Rh=A/P), não havendo a necessidade do tratamento de valores ausentes,

uma vez que todos os dados estavam completos. Sua base de dados foi formada por 40 medições acerca dos seguintes atributos: vazão, declividade da linha d′água, raio hidráulico, largura do espelho d′água, descarga sólida transportada no leito e em suspensão. Antes de gerar os modelos foi realizado o pré-processamento dos dados para discretização dos dados numéricos. Para os treinamentos e testes foram utilizados os algoritmos supervisionados de Árvore de Decisão C4.5, *Naive-Bayes* (NB), Regressão Logística (RL) e o algoritmo não-supervisionado *Expectation Maximization* (EM), todos presentes no *software* de mineração de dados WEKA. Para validar os modelos foi utilizado o método de validação cruzada (*cross-validation*). Os resultados dos algoritmos apresentaram os seguintes índices de classificação correta para cada classe: C4.5, 40%; NB, 47,5%; RL, 30%. O algoritmo EM (usado para realizar agrupamentos sem indicação prévia da quantidade de *clusters* esperada) identificou aleatoriamente 5 agrupamentos: 18%, 18%, 18%, 20% e 28%. Os autores apontam que a amostra de dados para o treinamento mostrou-se pequena, não sendo suficiente para gerar modelos com alta acurácia, o que pôde ser constatado nas taxas de classificações obtidas no aprendizado. Dos algoritmos de aprendizado supervisionado, *Naive-Bayes* foi o que apresentou melhor desempenho em comparação com o C4.5 (Árvore de Decisão) e Regressão Logística. Quanto à tarefa de agrupamento, o algoritmo EM realizou a alocação dos exemplos em 5 grupos, notou-se que nos grupos 0, 1 e 2, continham sete elementos representando 18%. Neste cenário, os autores constataram que é preciso realizar estudos posteriores a fim de verificar quais parâmetros esse algoritmo levou em consideração para o arranjo dos grupos. Os mesmos ainda supõem que as variáveis do banco de dados são não-lineares e os algoritmos utilizados possuem interação linear, sendo esta uma possível causa da não obtenção de êxito nas tarefas realizadas pelos algoritmos classificadores. Por fim, para ter um desempenho melhor nos algoritmos

utilizados, os autores sugerem a utilização de um banco de dados maior, haja vista que não se conseguiu fazer deduções mais contundentes com o número de instâncias coletadas, além de ser necessário utilizar outras técnicas e algoritmos para realizar comparações com os resultados já obtidos.

Em relação ao trabalho de Siviero & Hruschka Júnior (2011), podemos verificar que os resultados ainda não foram satisfatórios no domínio estudado. Um dos possíveis motivos desse insucesso nesses primeiros experimentos é a baixa quantidade de exemplos submetidos aos algoritmos. Como mencionado no Capítulo anterior, Quinlan (1993) indica que uma das condições para obter bons modelos preditivos é utilizar um número de instâncias suficiente para os modelos generalizarem os padrões. Outro fator que poderia ser levado em consideração é o ajuste dos parâmetros de cada algoritmo, os quais poderiam influenciar em uma possível melhora nas taxas de acerto.

Outro trabalho que possui uma dinâmica semelhante à adotada nesta pesquisa é o de Sarmento (2010). Em seu estudo, a autora testou quatro algoritmos de aprendizado de máquina para predizer a ocorrência de grupos de solo no Vale dos Vinhedos em Rio Grande do Sul. A carência de dados sobre esses solos motivou o desenvolvimento da pesquisa com técnicas de modelagem em aprendizado de máquina para estimar classes ou propriedades de solos. A metodologia do estudo é pautada na comparação de quatro algoritmos de aprendizado de máquina (três redes neurais: *Fuzzy ARTMap*, SOM e MLP; e uma árvore de decisão: *Gini*) quanto à predição de ordens de solo no Vale dos Vinhedos. O material usado na pesquisa foi composto pelo Modelo Numérico do Terreno (MNT) com resolução de 5 metros, uma base cartográfica digital, um mapa detalhado dos solos e um *software* de Sistema de Informação Geográfica (SIG) chamado *Idrisi*. A partir do MNT e da base cartográfica foram calculadas 07 variáveis topográficas e hidrológicas, cujos valores e identificação do grupo de solos foram lidos

em 1.288 pontos aleatoriamente distribuídos. Os dados destes pontos amostrais foram utilizados para formar a base de dados e treinar os algoritmos classificadores de grupos de solos. Os resultados foram avaliados através de matriz de erros, exatidão geral e estatística *Kappa*, tomando o mapa convencional como referência. De acordo com os resultados apresentados no estudo, a árvore de decisão obteve a melhor acurácia com 71% de acertos e estatística *Kappa* 0,58. Entre as três redes neurais, a rede MLP apresentou índices próximos aos da árvore de decisão, porém foi mais sensível à densidade de amostragem, obtendo estatística *Kappa* acima de 0,5. Contudo, houve a averiguação de que ambas as técnicas de AM mostraram-se promissoras para a predição da distribuição dos solos em RS. A autora também destaca que as árvores de decisão possibilitam o estudo de suas estruturas, sendo mais fáceis de compreender e visualizar as regras adotadas nos modelos. No entanto, as redes neurais artificiais não devem ser descartadas por não apresentarem sua estrutura interna, pois os bons resultados obtidos com esses algoritmos justificam sua aplicação nos mais diversos problemas reais. Como trabalhos futuros a pesquisadora aponta a necessidade de novos experimentos para testar a inclusão de variáveis preditoras adicionais, além de comparar os resultados computacionais com dados observados em campo para avaliar o grau de aproximação dos mapas estimados em relação aos mapas reais.

Analisando o trabalho de Sarmento (2010), constata-se que nesse estudo foi aplicado um número maior de exemplos nas modelagens com as duas técnicas usadas. Além disso, os parâmetros de cada algoritmo foram levados em consideração no momento das modelagens. Esses fatores podem ter contribuído positivamente na criação de modelos considerados pela autora como satisfatórios na predição de classes de solos no Vale dos Vinhedos em RS.

No contexto de agrupamento de objetos, Gil *et al* (2015) exploraram bases de dados astronômicos com parâmetros morfométricos de galáxias, a fim de descobrir padrões naturais de agrupamento como uma etapa anterior a classificação das galáxias. Segundo os autores, a morfologia fornece informações importantes sobre as propriedades físicas das galáxias, como a taxa de formação estelar e a cinemática. Neste sentido, um dos objetivos principais dos estudos extragaláticos é entender o que direciona a morfologia das galáxias e como elas evoluem com o tempo e o ambiente cósmico. A metodologia do estudo é fundamentada na análise exploratória de dados por meio de técnicas de agrupamento, objetivando analisar os resultados para detectar classes de galáxias mediante parâmetros morfométricos. O conjunto de dados da pesquisa foi formado por dados reais e sintéticos que continham medidas morfométricas de galáxias. Para a realização dos experimentos foram utilizados os algoritmos não supervisionados *Expectation Maximization* (EM) e *K-médias*, aos quais foram submetidos dados morfométricos reais do catálogo *Extraction de Formes Idealisées de Galaxies en Imagerie* (EFIGI), contendo galáxias de todos os tipos morfológicos. Após o agrupamento dos dados pelos algoritmos, foi utilizado o algoritmo *Silhouette* como método de validação para os resultados encontrados. Dados finais mostram que os algoritmos realizaram corretamente a identificação das galáxias por suas classes. Por meio do *Silhouette*, pôde-se deduzir que todos os objetos estavam localizados em seus respectivos grupos. Nesse estudo, o EM se mostrou mais adequado à aplicação, pois seu coeficiente de *Silhouette* é melhor do que o apresentado pelo *K-médias*. Apesar desse fato, ambos os algoritmos deram origem a resultados semelhantes, o que os torna aptos para aplicação na predição de galáxias. Com esses resultados, os autores concluíram que as galáxias espirais e elípticas apresentam algumas características morfométricas que as distinguem, os mesmos também apontam a necessidade de realizar outras análises de

agrupamento, submetendo aos algoritmos dados de catálogos de 14 mil objetos e grupos com cerca de 80 mil objetos, a fim de aprimorar a metodologia aplicada para a classificação de galáxias desses catálogos.

Mais trabalhos que auxiliaram na compreensão da aplicação das técnicas Árvores de Decisão, Redes Neurais Artificiais e K-Médias podem ser encontrados em Lapedes *et al* (1989), Toweel *et al* (1990), Craven & Shavlik (1994), Pedersen & Nielsen (1997), Bajic *et al* (2002), Matos (2007), Silva (2008), Pellucci *et al* (2011), entre outros.

Com base nas informações extraídas após as análises dos trabalhos citados, alguns procedimentos metodológicos foram adotados nesta pesquisa para tentar gerar modelos com maior acurácia, tais como: utilizar um conjunto de dados com um número de instâncias satisfatórias no treinamento dos modelos; analisar a importância de cada atributo preditivo na classificação dos objetos; testar diferentes níveis dos parâmetros de cada algoritmo; comparar medidas de distância; utilizar diferentes cenários de dados nas modelagens; aplicar e comparar métodos de validação de resultados; entre outros. Estes procedimentos adotados são descritos detalhadamente no Capítulo 5.

Até a finalização desta pesquisa, não foram encontrados na literatura trabalhos com técnicas de AM utilizando bases de dados de espécies ou variedades de tucumã. Diante disso, todos os dados produzidos acerca dos algoritmos utilizados neste estudo serão um ponto de partida para subsidiar novas comparações e aplicações de novas técnicas com dados de tucumãs.

4 A ESPÉCIE *Astrocaryum aculetatum* G. Mey. (TUCUMÃ DO AMAZONAS)

4.1 Aspectos gerais das espécies de tucumã

Para o entendimento do processo de execução das atividades desta pesquisa, neste capítulo é apresentada uma revisão a cerca das características principais do tucumã.

Dentre as inúmeras espécies de plantas frutíferas com potencial econômico, tecnológico e nutricional, o tucumã – espécie pertencente à família da *Arecceae* (Palmeiras) – vem despertando o interesse de estudos científicos em diversas áreas, como: alimentícia, farmacêutica, cosmética, aromatizantes e essências, etc. (CLEMENT *et al*, 2005). Culturalmente na região Amazônica, as populações do interior utilizam seus frutos e sementes na alimentação humana e animal, as folhas e estipes na construção de casas, assim como matéria prima para confecção de artesanato (MIRANDA, 2001).

As duas principais espécies de palmeira de tucumã encontradas na Amazônia Brasileira são *Astrocaryum aculeatum* G. Mey. (tucumã do Amazonas) e *Astrocaryum vulgare* Mart. (tucumã do Pará). Estas duas espécies se diferem quanto às características morfológicas e concentração geográfica (CLEMENT *et al*, 2005).

O tucumã do Amazonas é encontrado principalmente na Amazônia Central e Ocidental, nos Estados do Amazonas, Acre, Rondônia e Roraima, mas também em algumas partes do Pará, no Peru e na Colômbia (FERREIRA & GENTIL, 2005; KAHN, 2008). Esta espécie possui uma palmeira grande, podendo atingir até 25 metros de altura. Apresenta um único tronco grosso, envolto em espinhos compridos. Seus frutos são grandes e a polpa é pouco fibrosa e bastante nutritiva (CAVALCANTE, 2010).

O tucumã do Pará cresce geralmente em terra firme alta e de cobertura vegetal baixa. Embora encontrado no Amazonas, sua maior concentração está nos Estados do Pará e Amapá (CAVALCANTE, 2010). O tucumã do Pará apresenta grande capacidade de regeneração possuindo em média quatro estipes densamente espinhosos por touceira, podendo chegar a 15 metros de altura. Sua polpa é fibrosa e, embora em menor quantidade, também apresenta alto valor nutricional (FERREIRA & GENTIL, 2005). Devido a seus frutos serem muito pequenos e apresentar pouco teor de polpa, esta espécie dificilmente é comercializada, sendo apenas utilizada na alimentação familiar e de animais. Na indústria, o tucumã do Pará está sendo empregado em pesquisas para produção de Biodiesel (CLEMENT, 2005).

Figura 5 - Espécies de palmeiras de tucumã comuns na região Amazônica

Cavalcante (2010) destaca o potencial produtivo que o tucumã representa no mercado de alimentos, cosméticos, artesanato e óleos essenciais, sendo considerado um insumo promissor para a produção do biodiesel na Amazônia. Sua polpa é rica em caroteno (pró-vitamina A), proteínas, carboidratos, minerais e fibras, podendo ser consumida *in natura* ou na forma de suco, licor, sorvete, creme, entre outros.

O foco principal desta pesquisa foi nas variedades de *Astrocaryum aculeatum*, haja vista que esta espécie é amplamente encontrada e comercializada na região durante quase todos os meses do ano.

4.2 Períodos de frutificação do tucumã

A época de alta frutificação do *Astrocaryum aculeatum* geralmente ocorre sempre no primeiro semestre de cada ano, como mostrado na Figura 6. Porém, é bastante comum existir produção durante todo o ano em menor escala. Isto se deve ao fato de que a frutificação das palmeiras de tucumã depende diretamente das estações chuvosas (SCHROTH *et al*, 2004), portanto, algumas microrregiões sofrem variação e podem abastecer o mercado em outros meses diferentes aos de alta estação (KAHN & MOUSSA, 1999).

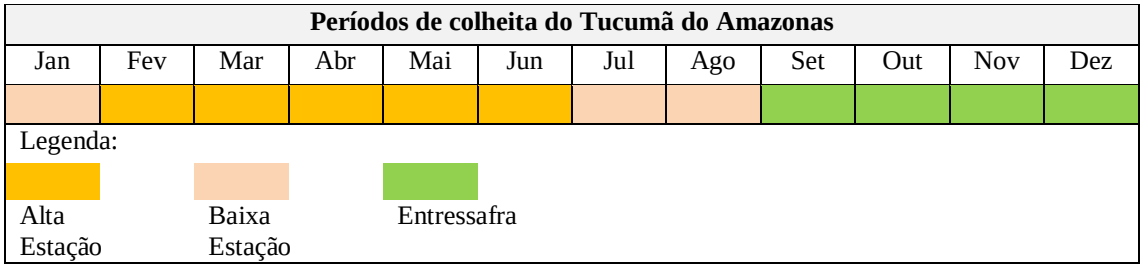

Figura 6 - Calendário anual de frutificação do *Astrocaryum aculeatum* (KHAN & MOUSSA, 1999)

Na região da cidade de Itacoatiara, a frutificação do tucumã começa no mês de dezembro e se entende em larga em escala até o mês de junho. A partir daí, há uma drástica redução na frutificação, influenciando na comercialização do fruto tanto local quanto regionalmente. Em seu trabalho, Didonet (2012) mostra a representatividade de Itacoatiara no fornecimento de tucumãs ao mercado manauara entre os meses de maio de 2011 a abril de 2012. Os dados de sua pesquisa são apresentados na Tabela 1.

Anos 2011 e 2012	***Jun 2011***	***Jul***	***Dez 2012***	***Jan***	***Fev***	***Mar***	***Abr***	*Total Anual em relação a todos os fornecedores*
Total mensal em relação a todos os fornecedores	19%	14%	24%	19%	23%	39%	8%	15%

Tabela 1 - Meses que Itacoatiara forneceu tucumã ao mercado de Manaus entre o período de maio de 2011 a abril de 2012 (DIDONET, 2012)

Em comparação com todos os outros fornecedores, Didonet (2012) aponta Itacoatiara como a localidade que mais forneceu tucumãs no período de sua pesquisa, representando uma quantia de 15% do total anual fornecido a Manaus. Com base nos dados, podemos observar que entre o período de alta estação o fornecimento foi praticamente contínuo, mostrando que o município de Itacoatiara possui um grande potencial dentro do mercado de tucumãs na região.

4.3 Contribuições de pesquisas científicas realizadas com tucumã

No Brasil, pesquisadores de distintas áreas do conhecimento desenvolveram estudos com o tucumã para investigar aspectos quanto à morfologia, caracterização química, propagação e reprodução, produção de biodiesel, nutrição, entre outros. A revisão dessas pesquisas comprova a importância do estudo desse fruto, mostrando seu potencial para o desenvolvimento científico e tecnológico da região Amazônica, agregando valor à produção de novos conhecimentos sobre recursos naturais.

Ferreira *et al* (2008) determinaram as características físico-químicas do fruto e do óleo extraído de tucumã. A pesquisa indicou que o tucumã possui importantes propriedades nutricionais, sendo fonte de calorias, pró-vitamina A, fibras e lipídios,

especialmente do ácido graxo oleico. Com essas informações observa-se a importância nutricional desse fruto na alimentação humana.

Em outra pesquisa, a viabilidade da propagação *in vitro* do gênero *Astrocaryum* foi investigada por Rodrigues *et al* (2013). Embriões zigóticos de sementes maduras e imaturas de tucumã do Amazonas foram inoculados em meio de cultivo semi-sólido de *Murashige e Skoog* (MS) suplementado com vitaminas. Os resultados apontam que os embriões sobreviventes apresentaram taxa crescente de brotação *in vitro*, oportunizando novas pesquisas nesta área.

Yuyama *et al* (2008) realizaram em seu estudo o processamento de frutos de tucumã por desidratação e pulverização para avaliar sua vida-de-prateleira em diferentes tipos de embalagens e temperaturas de armazenagem. Os frutos *in natura* e desidratados foram analisados quanto à umidade, pH, acidez, açúcares totais e redutores, proteínas, lipídios, cinzas, carboidratos, energia, β-caroteno e equivalente de retinol. A pesquisa mostrou que o tucumã desidratado e pulverizado, independente do tipo de embalagem e temperatura de armazenamento, pode ser estocado e consumido por até 150 dias, além de seu potencial nutricional como fonte de energia e β-caroteno.

Nos últimos anos pesquisas também demonstraram o potencial do tucumã para produção de biodiesel. Barbosa *et al* (2009) avaliaram a produção de biodiesel etílico a partir de diferentes lotes de óleos de tucumã do amazonas, com índices de acidez baixos e elevados por meio de transesterificação por catálise básica e ácida homogêneas. Os dados obtidos nos experimentos e análises permitiram identificar um excelente potencial de produção de biocombustível a partir do óleo das amêndoas de tucumã.

Em um âmbito relacionado a esta pesquisa, Didonet (2012) avaliou aspectos da comercialização dos frutos e da polpa de *Astrocaryum aculeatum* em feiras e mercados de Manaus. Para o desenvolvimento do trabalho o autor coletou informações sobre as

procedências dos frutos, a quantidade comercializada nos mercados, assim como a variação sazonal nos preços praticados nos mercados da cidade. Os resultados mostram que o comércio desse fruto vem crescendo a cada ano, principalmente quanto ao tucumã beneficiado, ou seja, a venda de sua polpa. A pesquisa aponta o tucumã como um dos recursos amazônicos mais promissores no que se refere à geração de emprego e renda no estado do Amazonas.

Diante dessa discussão e com base no cenário apresentado, este estudo aponta um caminho alternativo para utilização de recursos computacionais na região, contribuindo para a validação do processo de identificação de variedades da espécie *Astrocaryum aculeatum* por meio de características do fruto, além de contribuir para a descoberta de informações importantes para o beneficiamento e comercialização destas variedades.

5 METODOLOGIA

O presente estudo foi realizado no município de Itacoatiara, Estado do Amazonas, Região norte do Brasil. Esse município pertence à Mesorregião do Centro Amazonense e está localizado a leste de Manaus (capital do estado) acerca de 266 quilômetros de distância. Seu território ocupa uma área de 8 892,038 km², representando 0.1047% de todo o território brasileiro (IBGE, 2014).

Para o alcance dos objetivos descritos nesta dissertação, o foco do trabalho foi voltado para a execução de duas tarefas principais: treinar e validar os modelos usando diferentes parâmetros ajustados de acordo com a técnica aplicada; e realizar o teste de cada modelo usando uma base de dados extra, própria para esse fim. Posteriormente à finalização dessas tarefas, os índices de desempenho de cada modelo precisaram ser confrontados para finalmente indicar quais as melhores técnicas, algoritmos e configurações de parâmetros ideais para lidar com o domínio estudado. Abaixo é apresentada uma lista descritiva com as atividades desenvolvidas durante a pesquisa:

1. Pesquisa bibliográfica;
2. Coleta de dados das variedades de *Astrocaryum aculeatum*;
3. Tratamento dos dados e formação dos conjuntos de treinamento e teste;
4. Geração dos modelos computacionais com as técnicas de AM selecionadas;
5. Teste dos modelos computacionais;
6. Análise e comparação dos modelos gerados;
7. Avaliação do potencial de comercialização das variedades de *Astrocaryum aculeatum*;

As atividades listadas acima são detalhadas nas subseções a seguir, apresentado os procedimentos metodológicos executados em cada uma delas.

5.1 Seleção das variedades de tucumã para o estudo

As primeiras atividades desenvolvidas na execução da pesquisa foram inerentes ao nivelamento dos conhecimentos acerca dos conceitos relacionados ao aprendizado de máquina e ao tucumã. Uma pesquisa bibliográfica inicial apontou a existência de duas espécies principais do fruto presentes na região Amazônica – o tucumã do Pará e o tucumã do Amazonas (CAVALCANTE, 1996). Sabendo-se disso, visitas a produtores locais foram realizadas com o intuito de entrevistá-los a fim de conhecer melhor essas espécies e identificar quais as variedades mais comuns são cultivadas e comercializadas pelos produtores locais.

Após diversas pesquisas de campo foram obtidas duas informações importantes: (i) - Quatro variedades principais da espécie tucumã do Amazonas são amplamente comercializadas no município de Itacoatiara, estas são chamadas empiricamente de tucumã-arara, tucumã-vermelho, tucumã-mesclado e tucumã-ararinha. Vale ressaltar que as nomenclaturas das variedades citadas são atribuídas informalmente, existindo produtores que podem chamá-las por outros nomes, porém, os mais conhecidos nas feiras visitadas são estes quatro escolhidos para a pesquisa; (ii) - O tucumã do Pará não é comercializado devido ao seu tamanho menor e baixo aproveitamento do fruto, sendo apenas empregado na alimentação de famílias das zonas rurais (quando a palmeira apresenta frutos médios) e na alimentação animal (quando a palmeira apresenta frutos pequenos) principalmente na criação de suínos. Diante destas informações, o tucumã do Pará não foi objeto desta pesquisa, haja vista que o foco está nas variedades de tucumãs com potencial econômico. A Figura 7 mostra um exemplar de cada variedade de tucumã do Amazonas selecionada para o estudo.

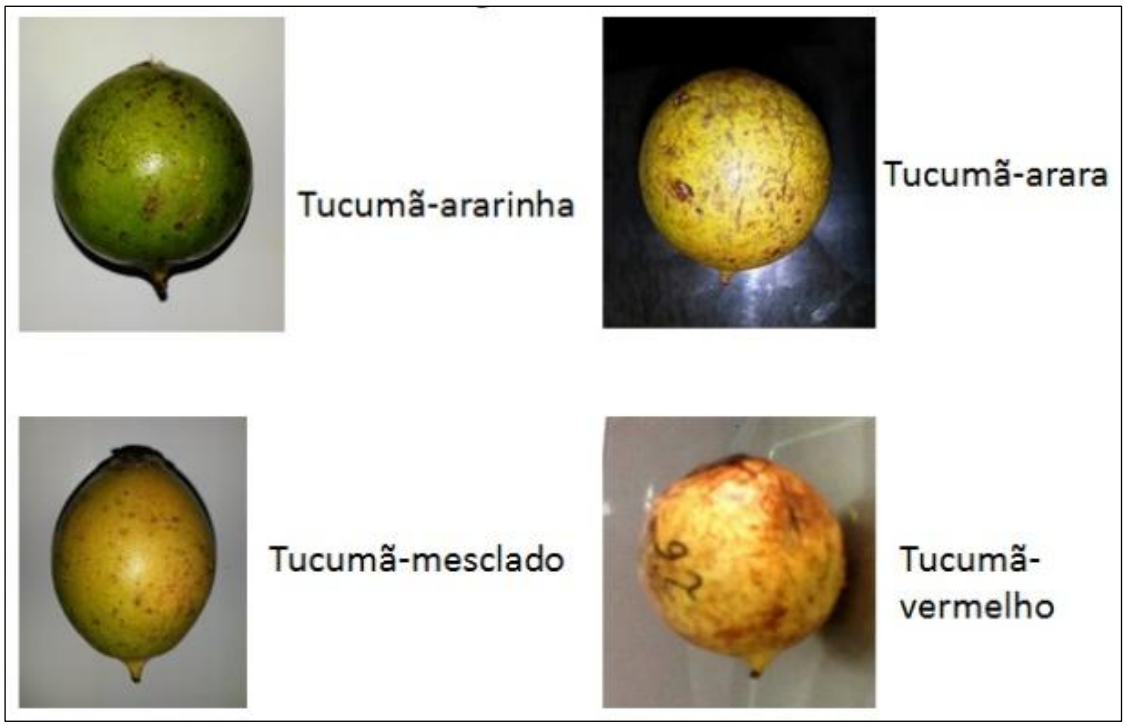

Figura 7 - Principais variedades de tucumã da espécie *Astrocaryum aculeatum* encontradas no município de Itacoatiara

5.2 Coleta de dados dos tucumãs

Para efeito desta pesquisa, os tucumãs utilizados no estudo procedem de sítios que se encontram próximos da rodovia AM-010, via que liga Itacoatiara a Manaus. As unidades do fruto foram compradas completamente maduras de produtores locais com cultivo em propriedades localizadas entre o quilômetro 01 e o quilômetro 20 da rodovia, sentido Itacoatiara-Manaus. A escolha dessa área geográfica foi baseada em dois motivos: (i) - é praticamente impossível coletar amostras de unidades de tucumã em toda a extensão territorial do município, devido ao difícil acesso a algumas localidades rurais; (ii) - a maioria dos produtores entrevistados nas feiras de Itacoatiara alegou ter suas propriedades localizadas entre o trecho da AM-010 citado.

Os frutos foram selecionados aleatoriamente para assegurar que unidades de diferentes árvores fossem utilizadas no estudo. Para certificar que as palmeiras eram da

espécie *Astrocaryum aculeatum*, visitas aos sítios onde os produtores coletavam os frutos foram feitas para identificar algumas árvores e conferir a compatibilidade com as características do tucumã do Amazonas.

A formação da base de dados envolveu a coleta de informações de 275 unidades de cada variedade, contabilizando um total de 1100 (mil e cem) instâncias, das quais 1000 (mil) foram coletadas no ano de 2014 e 100 (cem) no ano de 2015. As coletas foram feitas entre os meses de Janeiro a Julho de 2014 e Abril e Junho de 2015, época em que o tucumã apresenta um elevado nível de frutificação na região de Itacoatiara. Durante este período, as etapas de coleta de dados foram realizadas em meses diferentes para assegurar que houvesse uma análise dos frutos ao longo de toda a estação.

As aferições de peso foram realizadas em uma balança analítica da marca SHIMADZU, modelo BL320H com precisão de três casas decimais. Para as medições de tamanho foi necessário um paquímetro e a caracterização das colorações foi baseada em uma carta de cores com padrão RGB. No contexto desta pesquisa, as unidades de medidas utilizadas para registrar a biometria de circunferências e pesos foram respectivamente, milímetro e miligrama.

As etapas de coleta de dados durante a pesquisa contaram com um processo de medição individual dos atributos de cada um dos frutos das quatro variedades de tucumã. Estes atributos foram escolhidos de forma empírica, observando-se quais as características que melhor poderiam influenciar na separação de classes das variedades.

Todas as medições seguiram uma metodologia em duas fases:

Fase 1- cada unidade do fruto inteiro foi submetida à medição de um conjunto de atributos na seguinte ordem: Circunferência Horizontal (CH), Circunferência

Vertical (CV), Coloração do Epicarpo (casca do tucumã) (CE) e Presença de Rachaduras (PR).

Fase 2- o epicarpo, o mesocarpo (polpa) e o endocarpo (caroço) foram separados para realizar uma nova medição de atributos: Peso do Epicarpo (PE), Peso do Mesocarpo (PM), Peso do Endocarpo (PED), Peso do Fruto Inteiro (PFI) e Coloração do Mesocarpo (CM).

A separação das partes do tucumã foi feita manualmente, assemelhando-se ao método usado pela maioria dos produtores para beneficiar o fruto. Atualmente, poucos utilizam máquinas para despolpar os frutos, ainda sendo uma atividade essencialmente artesanal. Na Figura 8 é mostrado um exemplo do processo de pesagem das partes do tucumã separadas, desprezando o peso do vidro de relógio (equipamento laboratorial para pesagem de amostras) e da embalagem plástica protetora do mesocarpo.

Figura 8 - Separação e pesagem do endocarpo, mesocarpo e epicarpo dos frutos

Todos os dados obtidos sobre as variáveis selecionadas foram organizados em planilhas eletrônicas para tratamento posterior.

5.3 Tratamento dos dados

Antes da geração da base de dados para a modelagem computacional foi necessário executar algumas rotinas para identificar valores ausentes, reduzir discrepâncias de valores ruidosos e corrigir inconsistências. Os dados inconsistentes podem advir de erros de digitação, mensurações errôneas e presença de unidades de tucumã de outras espécies e/ou variedades, isto pode gerar dados anômalos que possivelmente interferirão no treinamento dos modelos computacionais.

Algumas técnicas são aplicáveis para valores ausentes, como por exemplo (HAN & KAMBER, 2001):

1 - Ignorar a tupla (instância completa formada pelos atributos de um objeto)

2 - Suprir valores ausentes:

a) manualmente;

b) através de uma constante global;

c) utilizando a média do atributo;

d) utilizando a média do atributo para todas as instâncias da mesma classe;

e) com o valor mais provável (regressão, inferência, etc.).

Algumas técnicas como 2b, 2c, 2d e 2e podem "viciar" os modelos ocasionando erros na classificação. A técnica 2e é uma estratégia interessante, pois em comparação com outros métodos utiliza um maior número de informações dos dados disponíveis.

Na base de dados formada para este estudo foram encontradas 14 instâncias com valores muito diferentes de suas classes e 06 valores ausentes em outras instâncias. Os exemplos com valores discrepantes podem ter vindo de unidades de outras variedades consideradas erroneamente e ainda da presença de frutos morfologicamente

mal formados. Para resolver o primeiro caso foram coletados atributos de novas unidades de tucumã para substituir as instâncias discrepantes, e no caso dos valores ausentes foi aplicada a técnica 2e realizando-se a inferência dos valores com base nas médias dos atributos.

5.4 Formação dos conjuntos de dados para modelagem no WEKA

O *software* WEKA possui funções para o pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização, podendo também ser usado para desenvolver novos algoritmos de aprendizagem de máquina, isto se deve ao fato dessa ferramenta ser um *software* com código aberto (WITTEN & FRANK, 2005). O sistema possui uma interface gráfica amigável e seus algoritmos fornecem relatórios com dados analíticos e estatísticos a respeito do domínio estudado.

Para que o *software* pudesse utilizar os dados coletados, foi preciso construir uma base de dados em formato compatível usando os exemplos organizados previamente nas planilhas eletrônicas. Um dos formatos de arquivos compatíveis com o WEKA é o *.ARFF*. Essa extensão representa um arquivo de texto puro, composto da seguinte maneira: Relação, compreendendo a primeira linha do arquivo, que deve ser iniciada com a expressão *@relation* seguida de uma palavra-chave que identifique a relação ou tarefa sendo estudada, por exemplo, *@tucuma*. Cada atributo é descrito nas linhas seguintes iniciadas com a expressão *@attribute* seguida do nome do atributo e seu tipo, que pode ser nominal, ou numérico (*real*), por exemplo, *@attribute peso-do-mesocarpo real*. Para atributos do tipo nominal as alternativas devem aparecer entre chaves separadas por vírgulas, por exemplo, *@attribute coloração-do-mesocarpo {cor 1, cor2, cor3,...}*.

Na tarefa de classificação o último atributo (atributo-alvo) descrito na lista deve identificar a classe das instâncias, no caso desta pesquisa o atributo-alvo é o tipo da variedade do tucumã. Depois de finalizado o cabeçalho de declaração é inserida a expressão *@data*, indicando que as demais linhas subsequentes irão conter as instância dos objetos coletados.

Cada linha de dados deve corresponder a uma instância e deve ter valores separados por vírgula correspondentes a mesma ordem dos atributos da seção *@attribute*. As frases precedidas do símbolo de porcentagem (%) são consideradas comentários e não são processadas. A Figura 9 mostra uma parte da base de dados usada neste estudo.

```
@relation tucuma

@attribute circunferencia-horizontal real
@attribute circunferencia-vertical real
@attribute coloracao-do-epicarpo {AMARELOESCURO, AMARELO, VERDECLARO, VERDEESCURO, AMARELOEVERDE}
@attribute presenca-de-rachaduras {SIM, NAO}
@attribute peso-do-epicarpo real
@attribute peso-do-mesocarpo real
@attribute peso-do-endocarpo real
@attribute peso-do-fruto-inteiro real
@attribute coloracao-do-mesocarpo {LARANJA, AMARELOCLARO, LARANJAESCURO, AMARELO, AMARELOESCURO}
@attribute tipo {TUCUMAVERMELHO, TUCUMAARARA, TUCUMAMESCLADO, TUCUMAARARINHA}


@data
42.4,   48.1,   AMARELOESCURO,  SIM,    7574,   13173,  32515,  53262,  LARANJA,        TUCUMAVERMELHO
40.9,   48.6,   AMARELO,        NAO,    8678,   11836,  29082,  49596,  LARANJA,        TUCUMAVERMELHO
41.0,   46.7,   AMARELOESCURO,  SIM,    7778,   15052,  28287,  51117,  LARANJA,        TUCUMAVERMELHO
40.6,   45.4,   AMARELOESCURO,  SIM,    8174,   11544,  27901,  47619,  LARANJA,        TUCUMAVERMELHO
40.3,   46.5,   AMARELOESCURO,  SIM,    9288,   12691,  27480,  49459,  LARANJA,        TUCUMAVERMELHO
40.2,   49.8,   AMARELOESCURO,  SIM,    7836,   14823,  27890,  50549,  LARANJA,        TUCUMAVERMELHO
41.7,   47.9,   AMARELO,        SIM,    8461,   12861,  29601,  50923,  LARANJA,        TUCUMAVERMELHO
44.6,   46.6,   AMARELOESCURO,  SIM,    8154,   18284,  33431,  59869,  LARANJA,        TUCUMAVERMELHO
43.0,   46.2,   AMARELO,        SIM,    9004,   13354,  36559,  58917,  LARANJA,        TUCUMAVERMELHO
43.5,   47.9,   AMARELOESCURO,  SIM,    7821,   16230,  31692,  55743,  LARANJA,        TUCUMAVERMELHO
41.7,   47.0,   AMARELOESCURO,  SIM,    7503,   14463,  30650,  52616,  LARANJA,        TUCUMAVERMELHO
45.3,   47.2,   AMARELOESCURO,  SIM,    7940,   14318,  34252,  56510,  LARANJA,        TUCUMAVERMELHO
42.3,   45.8,   AMARELOESCURO,  SIM,    10406,  12905,  30080,  53391,  LARANJA,        TUCUMAVERMELHO
40.8,   39.9,   AMARELO,        SIM,    7589,   13997,  23203,  44789,  LARANJA,        TUCUMAVERMELHO
46.2,   47.3,   AMARELO,        NAO,    9442,   18452,  39208,  67102,  LARANJA,        TUCUMAVERMELHO
46.4,   51.2,   AMARELO,        NAO,    9244,   24278,  34118,  67640,  LARANJA,        TUCUMAVERMELHO
42.3,   46.5,   AMARELOESCURO,  SIM,    9160,   15190,  37935,  62285,  LARANJA,        TUCUMAVERMELHO
45.8,   47.0,   AMARELOESCURO,  SIM,    8524,   17059,  32980,  58563,  LARANJA,        TUCUMAVERMELHO
44.7,   47.8,   AMARELO,        SIM,    8438,   12945,  31793,  53176,  LARANJA,        TUCUMAVERMELHO
39.4,   41.7,   AMARELO,        SIM,    7342,   15655,  20759,  43756,  LARANJA,        TUCUMAVERMELHO
42.7,   42.9,   AMARELO,        SIM,    7708,   16298,  24141,  48147,  LARANJA,        TUCUMAVERMELHO
41.2,   44.5,   AMARELOESCURO,  SIM,    6467,   12167,  31828,  50462,  LARANJA,        TUCUMAVERMELHO
40.5,   41.4,   AMARELO,        SIM,    7264,   13735,  22551,  43550,  LARANJA,        TUCUMAVERMELHO
41.2,   43.4,   AMARELO,        SIM,    8229,   16135,  23717,  48081,  LARANJA,        TUCUMAVERMELHO
41.6,   42.3,   AMARELOESCURO,  SIM,    7486,   14310,  29896,  51692,  LARANJA,        TUCUMAVERMELHO
43.7,   49.5,   AMARELOESCURO,  SIM,    9204,   16015,  30369,  55588,  LARANJA,        TUCUMAVERMELHO
43.8,   44.3,   AMARELOESCURO,  SIM,    7077,   14735,  32926,  54738,  LARANJA,        TUCUMAVERMELHO
40.9,   47.0,   AMARELOESCURO,  SIM,    7940,   14292,  29432,  51664,  LARANJA,        TUCUMAVERMELHO
42.0,   47.9,   AMARELOESCURO,  SIM,    7503,   14085,  31471,  53059,  LARANJA,        TUCUMAVERMELHO
42.6,   50.2,   AMARELOESCURO,  SIM,    8425,   13059,  33214,  54698,  LARANJA,        TUCUMAVERMELHO
44.6,   46.7,   AMARELO,        SIM,    7804,   16419,  33515,  57738,  LARANJA,        TUCUMAVERMELHO
45.9,   52.5,   AMARELOESCURO,  SIM,    11923,  18444,  37441,  67808,  LARANJA,        TUCUMAVERMELHO
41.2,   46.0,   AMARELOESCURO,  SIM,    12106,  10169,  32519,  54794,  LARANJA,        TUCUMAVERMELHO
```

Figura 9 - Exemplo de arquivo de dados no formato *.arff* com instâncias de tucumã

Para a criação dos arquivos *.ARFF* foi necessário dividir a base de dados em dois conjuntos distintos. O primeiro é o conjunto de dados de treinamento formado por 900 instâncias coletadas no período de 2014 (225 exemplos de cada variedade). O

segundo é o conjunto de dados de teste, sendo constituído pelas 100 instâncias restantes de 2014 somadas com as 100 instâncias coletadas no período de 2015 (50 exemplos de cada variedade). Essa divisão garantiu que os modelos gerados fossem testados usando dados de tucumãs de dois períodos diferentes de frutificação, permitindo uma melhor análise dos resultados.

Não foi encontrada na literatura uma normatização para definir a quantidade exata de instâncias para um determinado problema. Sabe-se que a qualidade dos atributos e dos dados terá grande influência no treinamento dos modelos (HUA *et al*, 2009). Diante disso, a decisão de formar a base de dados com 1100 instâncias teve como parâmetro outros estudos, e para garantir a qualidade procurou-se atender as condições descritas por Quinlan (1993), as quais foram mostradas anteriormente na subseção 2.2.1.

5.5 Carregamento da base de dados no WEKA

Na tela inicial da interface gráfica do WEKA (Figura 10) são exibidas quatro aplicações importantes, nas quais cada uma apresenta ferramentas com funções diferentes. As duas aplicações mais importantes são *Explorer*, para explorar os dados, e *Experimenter*, para realização de experimentos de comparação entre algoritmos diferentes de forma automatizada.

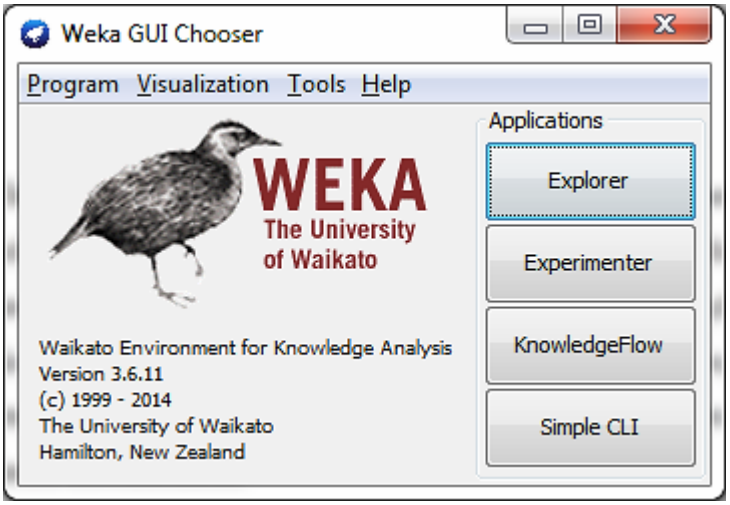

Figura 10 - Tela Inicial do *software* de mineração de dados WEKA

A aplicação *Explorer* permite que a base de dados seja carregada possibilitando aplicar diversos algoritmos de aprendizagem para gerar modelos computacionais. As guias estão distribuídas por tarefa:

- *Preprocess*: visualização e pré-processamento de dados (aplicação de filtros).

- *Classify*: Aplicação de algoritmos de classificação e regressão.

- *Cluster*: Aplicação de algoritmos de agrupamento.

- *Associate*: Aplicação de algoritmos de associação.

- *Select Attributs*: Seleção de atributos através de parâmetros específicos.

- *Visualize*: Visualização dos dados em pares de atributos.

O primeiro passo foi carregar a base de dados para o *software* na guia *Preprocess*. Nesse primeiro momento foram utilizados todos os nove atributos coletados mais o atributo-alvo de classificação das instâncias para a tarefa de treinamento.

Depois de carregada a base de dados, o *software* mostra todas as informações a respeito do mesmo, como: número de instâncias, atributos, classes, além de apresentar informações estatísticas sobre cada atributo (desvio médio padrão, valores mínimos e máximos, número de valores que aparecem apenas uma vez e número de valores diferentes).

Na parte gráfica da tela de pré-processamento de dados (Figura 11) é mostrada a partição dos valores por classe. Neste caso, as classes foram automaticamente representadas com cores distribuídas da seguinte forma: azul escuro para tucumã-vermelho; vermelho para tucumã-arara; verde para tucumã-mesclado; e azul claro para tucumã-ararinha.

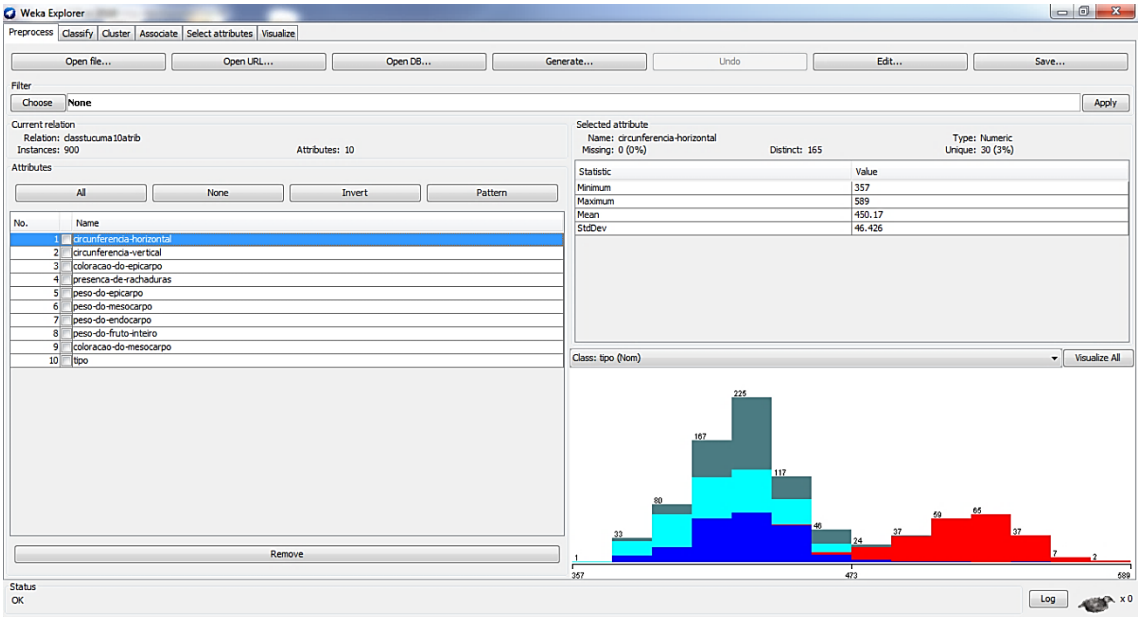

Figura 11 - Tela de visualização geral da guia *Preprocess*

A visualização da partição das classes por atributos auxilia na identificação preliminar de quais deles são mais relevantes quanto à eficácia para o processo de treinamento dos modelos, haja vista que o objetivo é encontrar um bom atributo classificador, ou seja, aquele que estabelece melhor uma fronteira entre os dados pertencentes a cada classe.

A indicação inicial dos melhores atributos por meio da análise dos gráficos nem sempre é clara, uma vez que quando o conjunto de dados possui muitos atributos, cada técnica irá se comportar de maneira diferente, porém sempre podem existir aqueles que visualmente mostram uma boa separação das instâncias. Abaixo são mostrados os gráficos gerados no WEKA para cada atributo da base de dados de tucumãs (Figuras 12, 13 e 14).

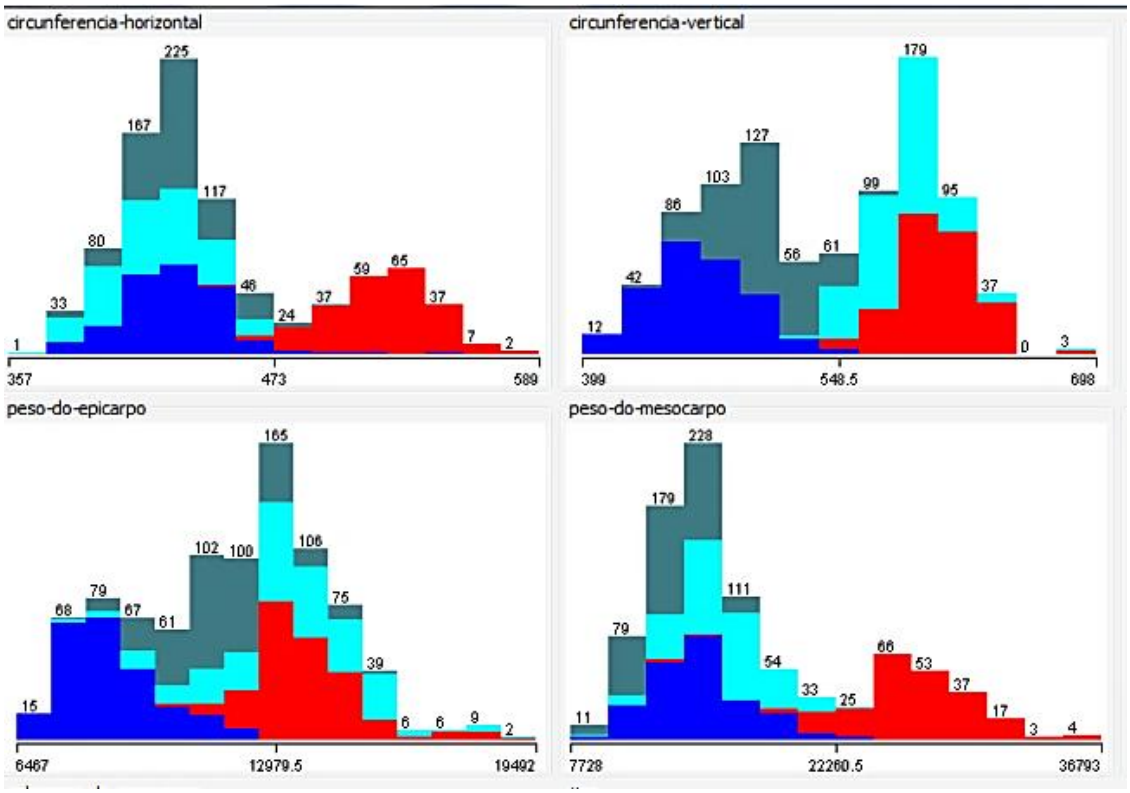

Figura 12 - Gráficos de separação de classes por meio dos atributos *circunferência-horizontal*, *circunferência-vertical*, *peso-do-epicarpo* e *peso-do-mesocarpo*

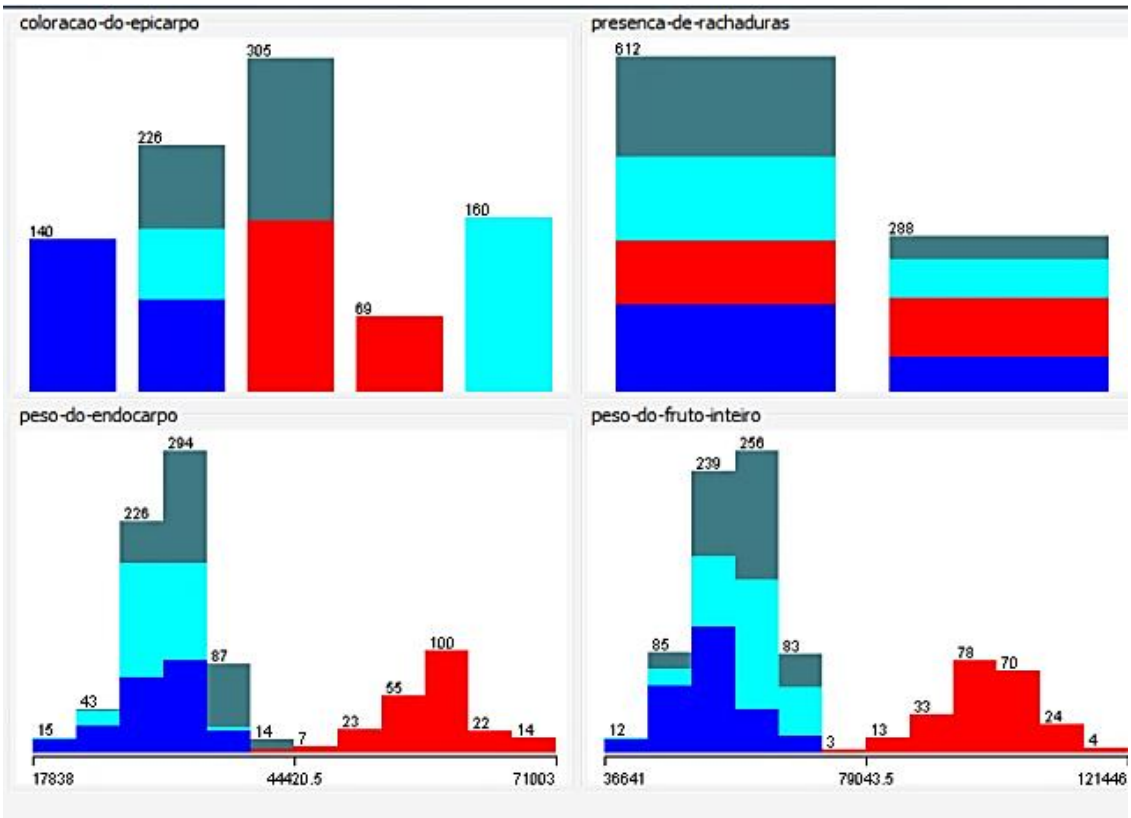

Figura 13 - Gráficos de separação de classes por meio dos atributos *coloração-do-epicarpo*, *presença-de-rachaduras*, *peso-do-endocarpo* e *peso-do-fruto-inteiro*

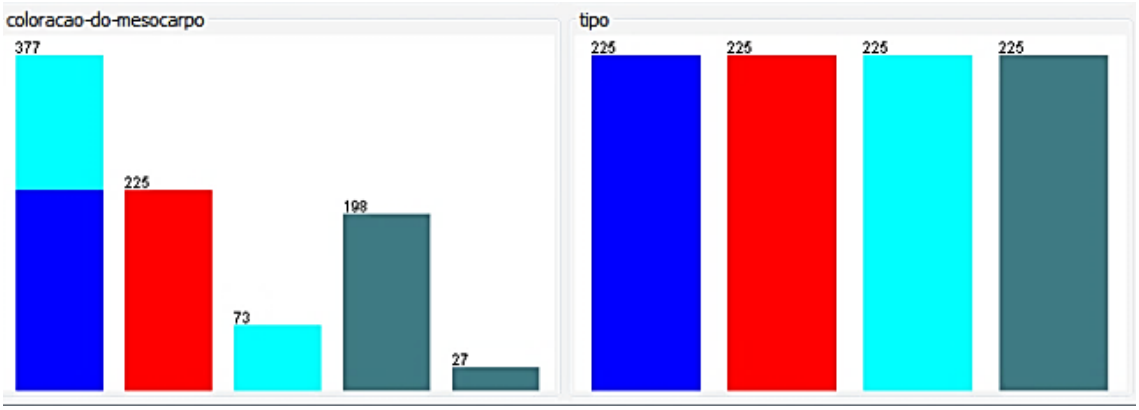

Figura 14 - Gráficos de separação de classes por meio do atributo *coloração-do-mesocarpo* e do atributo-alvo *tipo*

Como já previsto, observou-se nos gráficos que entre todos os atributos, o atributo-alvo (tipo) realiza perfeitamente a separação das 900 unidades, já que foram inseridas 225 instâncias de cada uma das quatro variedades de tucumã. A partir dessa informação, o objetivo é descobrir qual dos outros nove atributos realiza melhor a separação das classes. Visualmente, com base na observação informal dos gráficos, constata-se que entre os atributos nominais *coloração-do-mesocarpo* é o melhor na tarefa de separação das instâncias, e que dos atributos numéricos *circunferência-vertical* é o que tem o melhor desempenho nessa tarefa. Estas informações inferidas empiricamente são importantes, mas precisam ser validadas por meio da seleção de atributos feita por algoritmos específicos. Essa tarefa é mostrada em subseção posterior.

Nesta pesquisa, a base de dados foi formada por quantidades iguais de cada variedade do tucumã. Isso significa que temos um problema equilibrado, fornecendo melhores parâmetros para treinamento dos modelos computacionais no contexto desta pesquisa. Caso existisse, por exemplo, 50 unidades para a primeira variedade, 100 para a segunda, 200 para terceira e 650 para a quarta teríamos um problema desequilibrado. Sendo assim, a configuração da base de dados utilizada segue os padrões para um bom treinamento dos modelos com o uso das técnicas selecionadas.

5.6 Treinamento, validação e teste dos modelos computacionais

5.6.1 Modelos treinados com as técnicas Árvores de Decisão e Redes Neurais Artificiais

Em todas as modelagens é necessário escolher uma forma de validar e testar os modelos para indicar os níveis de acurácia, no *software* WEKA são encontradas diversas maneiras de calcular a porcentagem de acerto, essas opções são:

- *Use training set*: Neste caso é utilizado o mesmo conjunto de dados de treinamento para executar o teste. Essa opção retorna uma porcentagem muito otimista sobre o classificador, por isso não é conveniente utilizá-la.

- *Supplied test set*: Essa opção utiliza um conjunto extra de dados para fazer o teste, sendo necessário organizá-los em uma estrutura idêntica ao do conjunto de treinamento, porém em um arquivo de dados separado.

- *Cross-validation*: Calcula a porcentagem de acertos esperada fazendo uma validação cruzada de *k-partes*. Por padrão do *software* WEKA, *k* é igual a 10.

- *Porcentage split*: Nessa opção o conjunto total de dados será dividido em duas partes: os primeiros 66% serão para construir os modelos e os 33% restantes serão usados para fazer o teste.

Nesta pesquisa todos os classificadores foram treinados com os 900 exemplos, validados por meio de *cross-validation* e testados com o conjunto de 200 dados desconhecidos pelos modelos. Para a validação cruzada foi adotada a configuração padrão de k=10.

O primeiro algoritmo de classificação usado foi o *ZeroR*. Esse algoritmo classifica todos os dados de acordo com a classe majoritária. Isso quer dizer que, por

exemplo, se em uma base de dados 90% são da classe A e 10% da classe B, ele classificará todas as instâncias como pertencentes à classe A. É conveniente usar primeiro esse classificador, pois a porcentagem de acertos indicada por ele tem que ser superada pelos outros classificadores utilizados na pesquisa, ou seja, ele estabelece uma taxa mínima de acertos para usar como parâmetro.

Depois de obter essa taxa mínima deu-se início aos treinamentos dos modelos com os algoritmos das técnicas de árvore de decisão e redes neurais artificiais. Para a primeira técnica foi escolhido o algoritmo *J48* e para a segunda o algoritmo *MultilayerPerceptron*.

5.6.1.1 Modelagem com o algoritmo *J48*

Todos os algoritmos têm parâmetros que podem ser ajustados para tentar melhorar o desempenho na classificação de objetos. Essas mudanças têm a ver com a complexidade do algoritmo classificador, no caso do *J48* e de outros algoritmos, é o *confidenceFactor* (fator de confiança) que influencia diretamente na complexidade do modelo construído. Sabe-se ainda que esta complexidade tem a ver com o *overfitting*, quanto menor é o fator de confiança, mais simples o modelo tende a ser e vice versa (WITTEN & FRANK, 2005). Este parâmetro varia entre 0 e 1, sendo definido por padrão no WEKA em 0.25.

Para efeito desta pesquisa foram utilizados três diferentes fatores de confiança pra criar os modelos, além do padrão de CF=0.25, foram adotadas taxas de CF=0.001 (modelos mais simples) e CF=1.0 (modelos mais complexos). Todos os modelos utilizaram validação cruzada de k=10.

O meta-classificador *CostSensitiveClassifier* em combinação com o *J48* foi aplicado para tentar criar um modelo de árvore de decisão com melhor desempenho, usando a melhor taxa de CF encontrada. O *CostSensitiveClassifier* permite introduzir uma matriz similar a matriz de custo, de forma que possamos forçar o classificador base (no caso o *J48*) a melhorar a predição de uma classe.

5.6.1.2 Modelagem com o algoritmo *MultilayerPerceptron*

Diferentes modelos foram criados utilizando a técnica de redes neurais artificiais por meio do *MultilayerPerceptron*. Uma dos parâmetros deste algoritmo é o *validationSetSize*, que corresponde a uma técnica de parada no treinamento do algoritmo. Quando é definido um número *x* nesta variável, o treinamento do modelo irá realizar uma pausa a cada *x* ciclos para fazer a estimativa de erro da rede sobre o conjunto de teste. Se for identificado que um erro no conjunto de validação tende constantemente a piorar, o treinamento é interrompido. Por padrão esse parâmetro é definido no WEKA em 0, ou seja, não será realizada nenhuma avaliação periódica durante o treinamento, em vez disso, a rede treinará com base no número especificado no parâmetro *trainingTime* (épocas de treinamento), que por padrão é 500.

No início desta pesquisa investigou-se o desempenho de dois modelos com RNAs utilizando *validationSetSize*=0 e *validationSetSize*=10. Por comparação, o valor 10 foi escolhido com base no número padrão de ciclos de validação cruzada indicado por Witten & Frank (2005). No caso do modelo treinado com *validationSetSize*=10, além de passar pelos 10 ciclos de validação cruzada, o mesmo também foi submetido a uma pausa a cada 10 épocas de treinamento para estimar o nível de erro. Porém, os valores deste parâmetro podem variar de 0 a 99, o que significa que somente estes dois

testes não são suficientes para determinar qual a configuração ideal de *validationSetSize*. Diante disso, os resultados com esse parâmetro são apresentados apenas em caráter experimental para determinar se este pode ou não influenciar nas taxas de acerto dos modelos.

No âmbito das RNAs outro parâmetro que tem grande influência no processo de treinamento é o *learningRate* (taxa de aprendizado). Quando a taxa de aprendizado é muito baixa, o treinamento da rede pode tornar-se muito lento, porém, uma taxa muito alta pode provocar oscilações no treinamento impedindo a convergência do processo de aprendizado. O valor da taxa de aprendizado pode variar de 0.1 a 1.0. Diversos testes experimentais no WEKA apontaram para o um número padrão de LR definido em 0.3 (WITTEN & FRANK, 2005). Nesta pesquisa, três níveis de taxa de aprendizado foram analisados para investigar a influência desse parâmetro nas RNAs durante a classificação de variedades de tucumã.

Devido o modelo com *validationSetSize*=0 (padrão) também ter sido construído com LR=0.3, não foi necessário fazer uma nova modelagem para esse padrão, sendo apenas preciso criar mais dois modelos comparativos. Nesses outros dois modelos, as taxas de aprendizado foram elevadas (respectivamente) ao mínimo (LR=0.1) e ao máximo (LR=1.0), observando-se qual a influência de cada nível para o desempenho das RNAs criadas.

5.6.1.3 Avaliação e seleção automática de atributos

O objetivo da avaliação e seleção de atributos é identificar quais são os mais relevantes, realizando a eliminação de atributos redundantes. Por relevantes, entendem-se os atributos que possuem alta correlação com as classes e não com os outros

atributos. A seleção de atributos pode diminuir o tempo computacional e em muitos casos aumentar a acurácia dos modelos classificadores (FREITAS, 1998).

A dinâmica da avaliação de atributos consiste em medir como este interage com o algoritmo de aprendizado. Essa medição pode ser feita por meio de duas abordagens principais: *Filter* e *Wrapper* (KOHAVI & JOHN, 1998).

Nesta pesquisa esses dois métodos de avaliação foram aplicados para estimar a relevância dos atributos preditivos. Além das duas abordagens citadas, um método *Ranker* também foi empregado nessa fase de avaliação. O ranqueamento realiza um cálculo do mérito de cada atributo em relação à sua capacidade de separar as classes, resultando em uma lista ordenada dos atributos por mérito obtido (WITTEN & FRANK, 2005).

O método *Filter* implementa um processo separado antes da aplicação efetiva do algoritmo de aprendizagem (FREITAS, 1998). Esse processo introduz um filtro para identificar os atributos irrelevantes, o qual considera características gerais do conjunto de dados para selecionar alguns atributos e excluir os demais. Dessa forma, o método de filtro é independente do algoritmo de aprendizado, sua meta é selecionar um subconjunto de atributos que permita um bom desempenho no modelo (JONH *et al*, 1994).

No método *Wrapper* o processo também ocorre externamente ao algoritmo-base, porém, utilizando o próprio algoritmo como uma espécie de caixa preta para analisar o subconjunto de atributos selecionado a cada iteração. De forma mais específica, o método *Wrapper* gera subconjuntos de atributos candidatos extraídos do conjunto de treinamento, e os avalia com base na precisão obtida pelo algoritmo-base. Esse processo é cíclico e ocorre até que o critério de parada seja satisfeito, apresentando os atributos avaliados como melhores para a classificação (KOHAVI & JOHN, 1998).

Até esta fase da pesquisa foram utilizados todos os nove atributos para gerar os modelos computacionais classificadores. Porém, para o aprendizado de máquina é necessário a eliminação de atributos possivelmente redundantes ou irrelevantes. Se há um número excessivo de atributos, isto pode fazer com que o modelo seja complexo demais e acabe produzindo *overfitting* (WITTEN & FRANK, 2005). No WEKA a avaliação e seleção de atributos podem ser feitas escolhendo um método de busca e um método de avaliação.

Para averiguar quais são os atributos mais relevantes foram utilizados os seguintes métodos:

1. Avaliação de atributos individuais (método *Ranker*):
    a. Método de busca: *Ranker*
    b. Método de avaliação: *InfoGainAttributeEval*

Com esses métodos, os atributos são avaliados individualmente medindo-se o ganho de informação no que diz respeito à classificação dos objetos do conjunto de dados. A avaliação realizada indica o mérito que cada atributo possui em relação à separação das classes, gerando uma lista ordenada após a análise de cada atributo.

2. Avaliação de conjuntos de atributos (método *Filter*):
    a. Método de busca: *GreedyStepwise*
    b. Método de avaliação: *CfsSubsetEval*

Usando esses dois métodos, uma pesquisa é realizada avançando e retrocedendo no universo dos atributos. Essa pesquisa inicia com um atributo de um ponto arbitrário e vai adicionando outros atributos para averiguar a capacidade de separação das classes pelo subconjunto formado. Quando se percebe que a adição de um eventual atributo diminui essa capacidade, o algoritmo para a execução. Esse processo é realizado algumas vezes para avaliar todos os possíveis subconjuntos de atributos. A

avaliação dessa dinâmica é baseada na capacidade preditiva de cada atributo em relação ao grau de redundância entre seus pares no subconjunto avaliado. No final do processo, uma lista é mostrada com a classificação dos atributos, indicando a ordem em que os mesmo foram selecionados.

3. Avaliação de conjuntos de atributos com base em algoritmos de classificação (método *Wrapper*):
    a. Método de busca: *GreedyStepwise*
    b. Método de avaliação: *ClassifierSubsetEval*

Nessa avaliação, a dinâmica de busca de subconjuntos é a mesma do método 2, porém a avaliação dos subconjuntos de atributos é baseada no treinamento de algum algoritmo classificador. Com esse método de avaliação, a estimativa do mérito dos conjuntos de atributos é realizada por meio de testes durante a criação dos modelos.

No método *Wrapper* foi preciso escolher um algoritmo como base, para isto foram usados o *J48* e o *MultilayerPerceptron* aplicando a eles os melhores índices de CF e LR encontrados pelos experimentos. Para todos os métodos de seleção de atributos foi selecionada validação cruzada de k=05. Optou-se em reduzir a quantidade de ciclos de validação cruzada para deixar a avaliação dos atributos mais precisa, apesar disso dobrar o tempo de treinamento dos modelos (WITTEN & FRANK, 2005).

Nesses primeiros testes, os dados originais não foram modificados, ou seja, as instâncias continuam tendo todos os atributos. A avaliação realizada simplesmente permitiu identificar quais deles são mais relevantes para a classificação de variedades de tucumã. Diante dessa situação, é preciso investigar se uma seleção automática de atributos melhora ou piora a taxa de acerto de um modelo classificador. Para testar essa hipótese foi utilizado um meta-classificador chamado *AttributeSelectedClassifier*. O objetivo desse meta-classificador é passar um filtro de seleção de atributos e depois

realizar o treinamento e teste do modelo, usando exclusivamente os atributos que foram selecionados.

Para a seleção automática de atributos nesta pesquisa foi escolhida a abordagem do método *Filter*, uma vez que nessa etapa, o objetivo foi excluir os atributos menos importantes, deixando apenas os atributos mais relevantes a serem aplicados na criação dos modelos pelos algoritmos e parâmetros escolhidos (FREITAS, 1998).

Por tratar-se de um meta-classificador foi necessário especificar os algoritmos-base, um método de busca e um método de avaliação, a saber:

a. Algoritmos-base: *J48* e *MultilayerPerceptron* com os melhores índices de CF e LR escolhidos na pesquisa.
b. Método de busca: *BestFirst*
c. Método de avaliação: *CfsSubsetEval*

A busca nesse método pode começar de um conjunto vazio e pesquisar avançando entre os atributos adicionando um a um, ou começar com o conjunto completo de atributos e pesquisar retrocedendo, ou ainda começar em qualquer ponto do universo de atributos e pesquisar em ambas as direções (considerando todas as possíveis adições e exclusões de um único atributo em um determinado ponto) (WITTEN & FRANK, 2005).

Para avaliar os atributos, esse método observa o valor de um subconjunto de atributos, levando em consideração a capacidade preditiva individual de cada um, em relação ao grau de redundância entre eles (FREITAS, 1998).

As validações e testes desses modelos foram realizadas com validação cruzada de K=10 e uso do conjunto de dados extra. Depois de treinados, os modelos foram

analisados para verificar se houve ou não melhora no desempenho preditivo após a seleção automática de atributos.

Posteriormente ao término de todas as tarefas de classificação, comparações foram realizadas entre os modelos para apontar qual deles obteve melhor taxa de desempenho de acordo com as variáveis adotadas na pesquisa.

5.6.2 Modelos treinados com a técnica K-Médias

Para gerar os modelos de agrupamento com a técnica K-Médias foi utilizado o algoritmo *SimpleKMeans* configurado com as distâncias *Euclidiana* e de *Manhattan*. Neste método de classificação não-supervisionada é necessário informar previamente o número de *clusters* esperados. Por padrão do WEKA esse número é 2, mas para esta pesquisa o parâmetro precisou ser alterado para 4, haja vista que este é o número de variedades escolhidas inicialmente para o estudo. Com o resultado dos agrupamentos é possível obter novas informações e validar algumas inferências feitas no inicio do estudo, como por exemplo, a importância dos atributos preditivos. Essas tarefas de investigação são realizadas com base no conhecimento do domínio estudado, realizando análises aprofundadas nos modelos criados.

Na tarefa de agrupamento não há possibilidade de realizar validação cruzada, por isso, para testar os modelos foi utilizado o conjunto de dados de teste com 200 instâncias desconhecidas.

Ao final de todas as modelagens, os resultados obtidos foram comparados para indicar qual medida de distância melhor se aplica aos atributos da pesquisa, além de subsidiar a inferência de informações importantes sobre as variedades da espécie em questão.

5.7 Estimativa do teor de polpa de cada variedade de tucumã

Para complementar os objetivos do estudo, algumas atividades foram realizadas para investigar qual das quatro variedades apresenta maior teor de polpa. Primeiramente, alguns pacotes adicionais foram instalados no WEKA para permitir a criação de gráficos em 3D, possibilitando a melhor visualização dos objetos no universo de dados. Com base nesses gráficos foram realizadas algumas análises a fim de comparar as variedades.

Outra forma de realizar essa tarefa foi calcular a média aritmética dos atributos *peso-do-epicarpo*, *peso-do-mesocarpo* e *peso-do-endocarpo* de cada uma das quatro variedades seguindo a fórmula descrita na Função 13:

$$\textit{Valor médio do atributo} = \frac{(\textit{valor 1} + \textit{valor 2} + \ldots \textit{valor n})}{n} \qquad (13)$$

Depois de obtidas as médias dos atributos, a estimativa de quanto cada parte do tucumã ocupa em relação ao todo foi realizada com base na Função 14:

$$\textit{Valor percentual} = \frac{A_d}{T_d} * 100 \qquad (14)$$

onde A representa a média de um dos três atributos de peso (*peso-do-epicarpo*, *peso-do-mesocarpo* e *peso-do-endocarpo*), T corresponde à soma das médias dos atributos de peso de cada variedade, e d corresponde uma variedade do fruto. Por meio desses cálculos serão obtidos os valores médios do teor de polpa em relação ao peso total dos frutos de cada variedade de tucumã.

Apesar de serem fórmulas matemáticas simples, estes cálculos aliados às análises dos gráficos gerados pelo WEKA ajudarão na indicação preliminar de qual das variedades possui maior potencial econômico em relação ao comércio da polpa.

6 RESULTADOS E DISCUSSÕES

6.1 Resultados da modelagem computacional na tarefa de classificação

O modelo gerado com validação cruzada pelo algoritmo *ZeroR* e testado com a base de dados desconhecidos, apresentou taxa de acerto global de 25%, número já esperado, visto que as quatro variedades possuem 50 instâncias cada uma. Apesar de o algoritmo *ZeroR* classificar todos os objetos de acordo com a classe majoritária, em problemas balanceados a primeira classe encontrada será a base para a predição de todas as instâncias do conjunto de dados. Na porcentagem de acertos por classe (TP *rate* - *True Positive Rate*) observou-se que na a primeira classe (tucumã-vermelho) o modelo acerta 100% (TP *rate* =1) e para as demais, falha completamente na classificação (TP *rate*=0). Na Tabela 2 podemos ver a matriz de confusão onde é exibida a classificação de todas as instâncias como tucumã-vermelho.

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
50	0	0	0	B	=	Tucumã-arara
50	0	0	0	C	=	Tucumã-mesclado
50	0	0	0	D	=	Tucumã-ararinha

Tabela 2 - Matriz de confusão do teste do modelo gerado pelo algoritmo *ZeroR*

Já sabemos que a porcentagem de acerto global a ser superada por todos os modelos é de 25%. A partir dessa primeira análise, são apresentados os resultados gerados com os algoritmos *J48* e *MultilayerPerceptron* combinados com as variações dos parâmetros aplicados durante a pesquisa.

6.1.1 Resultados com o algoritmo *J48*

A complexidade de uma árvore de decisão depende diretamente do fator de confiança escolhido no algoritmo. Por padrão, esse parâmetro é definido no WEKA em 0.25, mas é preciso entender que dependendo do problema deve-se alterá-lo para averiguar qual o melhor índice para gerar os classificadores.

Os primeiros modelos criados com esse algoritmo foram induzidos em três níveis diferentes de CF e apresentaram os seguintes resultados:

1. Modelo gerado com o algoritmo *J48* configurado com CF=0.25 (índice padrão).

A árvore de decisão criada possui 12 folhas e 17 nós, implementando as seguintes regras:

```
J48 pruned tree
------------------
peso-do-fruto-inteiro <= 71942
|   circunferencia-vertical <= 53.8
|   |   coloracao-do-mesocarpo = LARANJA: TUCUMAVERMELHO
|   |   coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAVERMELHO
|   |   coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO
|   |   coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA
|   |   coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARINHA
|   circunferencia-vertical > 53.8
|   |   coloracao-do-epicarpo = AMARELOESCURO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = AMARELO
|   |   |   circunferencia-vertical <= 54.4: TUCUMAVERMELHO
|   |   |   circunferencia-vertical > 54.4: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = VERDECLARO: TUCUMAARARINHA
|   |   coloracao-do-epicarpo = VERDEESCURO: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = AMARELOEVERDE: TUCUMAMESCLADO
peso-do-fruto-inteiro > 71942: TUCUMAARARA
```

Após os dez ciclos de validação cruzada e de ter sido testada com o conjunto de dados extra, a árvore de decisão configurada com o fator de confiança padrão apresentou as taxas de acerto exibidas no Quadro 1.

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	98,4444 %	Taxa de acerto global	95,5%
Erro médio absoluto	0,0098	Erro médio absoluto	0,0241
Estatística *Kappa*	0,9793	Estatística *Kappa*	0,94

Quadro 1 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *J48* - CF=0.25

Esse modelo apresentou uma boa taxa de acertos com os dados de treinamento, mas quando submetido ao teste errou 4,5% das indicações de classes. A matriz de confusão gerada mostra onde ocorreram os erros na classificação:

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	8	42	0	C	=	Tucumã-mesclado
0	0	1	49	D	=	Tucumã-ararinha

Tabela 3 - Matriz de confusão do teste do modelo gerado pelo algoritmo *J48* - CF=0.25

Observando os erros na matriz da Tabela 3 podemos ver que a classe mais prejudicada foi tucumã-mesclado. Para tentar melhorar o desempenho na classificação dos tucumãs, dois novos fatores de confiança foram testados.

2. Modelo gerado com o algoritmo *J48* configurado com CF=0.001.

Uma das formas de averiguar se houve *overffiting* no primeiro modelo foi reduzir o fator de confiança ao mínimo possível para criar uma árvore de decisão mais simples. Neste caso, a árvore construída possui 10 folhas e 13 nós, implementando as seguintes regras:

J48 pruned tree
---
circunferencia-vertical <= 53.8
| *coloracao-do-mesocarpo = LARANJA: TUCUMAVERMELHO*
| *coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAVERMELHO*
| *coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO*

| *coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA*
| *coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARINHA*
circunferencia-vertical > 53.8
| *coloracao-do-mesocarpo = LARANJA: TUCUMAMESCLADO*
| *coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAARARA*
| *coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO*
| *coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA*
| *coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARA*

As taxas obtidas na validação e teste desse modelo com menos regras foram:

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	98,6667%	Taxa de acerto global	99,5%
Erro médio absoluto	0,0091	Erro médio absoluto	0,0047
Estatística *Kappa*	0,9822	Estatística *Kappa*	0,9933

Quadro 2 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *J48* - CF=0.001

A matriz de confusão do modelo mais simples (Tabela 4) apresenta uma configuração bem melhor em relação à matriz do modelo anterior, errando apenas a predição de uma instância. Neste caso, uma unidade de tucumã-ararinha foi classificada como tucumã-arara.

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	1	0	49	D	=	Tucumã-ararinha

Tabela 4 - Matriz de confusão do teste do modelo gerado pelo algoritmo *J48* - CF=0.001

Com a diminuição do fator de confiança a árvore se simplifica e as taxas de acerto aumentam. Dessa forma, fica provado que estava havendo *overffiting* no modelo com CF=0.25, pois, um modelo com menos regras obteve um desempenho maior na classificação. Por esse motivo, não foi interessante prosseguir com as modelagens usando esse fator de confiança padrão.

Apesar de o modelo mais simples apresentar bons índices, ainda foi preciso investigar se um modelo com maior complexidade se sairia melhor neste domínio. Para isso, o nível de CF foi elevado ao máximo para criar um novo modelo.

3. Modelo gerado com o algoritmo *J48* configurado com CF=1.0

Com o fator de confiança elevado, as árvores de decisão são criadas com uma complexidade maior. Dependendo do problema de cada estudo, o nível máximo do fator de confiança pode melhorar ou piorar a predição de classes. Este terceiro modelo foi criado com este fator para comparar com os anteriores. A nova árvore de decisão possui 20 folhas e 27 nós, implementando as seguintes regras:

```
J48 pruned tree
------------------
circunferencia-vertical <= 53.8
|   coloracao-do-mesocarpo = LARANJA
|   |   coloracao-do-epicarpo = AMARELOESCURO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = AMARELO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = VERDECLARO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = VERDEESCURO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = AMARELOEVERDE: TUCUMAMESCLADO
|   coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAVERMELHO
|   coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO
|   coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA
|   coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARINHA
circunferencia-vertical > 53.8
|   peso-do-endocarpo <= 35010
|   |   coloracao-do-epicarpo = AMARELOESCURO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = AMARELO
|   |   |   circunferencia-vertical <= 54.4: TUCUMAVERMELHO
|   |   |   circunferencia-vertical > 54.4: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = VERDECLARO: TUCUMAARARINHA
|   |   coloracao-do-epicarpo = VERDEESCURO: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = AMARELOEVERDE: TUCUMAMESCLADO
|   peso-do-endocarpo > 35010
|   |   coloracao-do-mesocarpo = LARANJA: TUCUMAMESCLADO
|   |   coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAARARA
|   |   coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO
|   |   coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA
|   |   coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARA
```

As taxas de predição da árvore com maior fator de confiança foram dispostas no Quadro 3.

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	98,7778%	Taxa de acerto global	99,5%
Erro médio absoluto	0,0059	Erro médio absoluto	0,0192
Estatística *Kappa*	0,9837	Estatística *Kappa*	0,9933

Quadro 3 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *J48* - CF=1.0

Comparando-se os índices obtidos nos dois últimos modelos (CF=0.001 e CF=1.0), podemos observar que não houve muita diferença entre os resultados após os modelos testados. A matriz de confusão do modelo com CF=1.0 (Tabela 5) mostra que novamente uma unidade de tucumã-ararinha foi classificada erroneamente, dessa vez como tucumã-mesclado.

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	1	49	D	=	Tucumã-ararinha

Tabela 5 - Matriz de confusão do teste do modelo gerado pelo algoritmo *J48* - CF=1.0

Uma comparação dos resultados dos testes dos modelos com CF=0.001 e CF=1.0 é mostrada no Quadro 4 para averiguar as diferenças entre eles.

J48 - CF=0.001		*J48* - CF=1.0	
Taxa de acerto global	99,5%	Taxa de acerto global	99,5%
Erro médio absoluto	0,0047	Erro médio absoluto	0,0192
Estatística *Kappa*	0,9933	Estatística *Kappa*	0,9933

Quadro 4 - Comparação entre os índices de acerto após os testes dos modelos gerados pelo algoritmo *J48* com CF=0.001 e CF=1.0

Observando os dados no Quadro 4, a árvore de decisão mais simples apresenta erro médio absoluto menor, contudo, a taxa de acerto global e a estatística *Kappa* foram idênticas nos dois modelos, sugerindo até esse momento que, por pouca diferença, o melhor fator de confiança seria 0.001. Porém, levando em consideração a fase de treinamento e validação, é possível constatar que o modelo com maior fator de confiança errou menos indicações de instâncias do que o modelo mais simples. Uma breve análise nas matrizes de confusão das validações cruzadas permite comparar esses dados. As Tabelas 6 e 7 mostram, respectivamente, as matrizes de confusão dos modelos com CF=0.001 e CF=1.0 após os 10 ciclos de *cross-validation*.

A	B	C	D	← Classificado como		Classe
221	0	3	1	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
4	0	221	0	C	=	Tucumã-mesclado
0	0	4	221	D	=	Tucumã-ararinha

Tabela 6 - Matriz de confusão da validação do modelo gerado pelo algoritmo *J48* - CF=0.001 - *cross-validation*=10

A	B	C	D	← Classificado como		Classe
221	0	3	1	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
4	0	221	0	C	=	Tucumã-mesclado
0	0	3	222	D	=	Tucumã-ararinha

Tabela 7 - Matriz de confusão da validação do modelo gerado pelo algoritmo *J48* - CF=1.0 - *cross-validation*=10

Depois da comparação das matrizes de confusão das validações observa-se que o modelo com CF=0.001 erra a predição de 12 instâncias, enquanto que o modelo com

CF=1.0 erra uma a menos. O Quadro 5 mostra a comparação dos índices de acerto obtidos nos dois modelos após a validação cruzada.

J48 - CF=0.001		*J48* - CF=1.0	
Validação do modelo (*k*=10)		Validação do modelo (*k*=10)	
Taxa de acerto global	98,6667%	Taxa de acerto global	98,7778%
Erro médio absoluto	0,0091	Erro médio absoluto	0,0059
Estatística *Kappa*	0,9822	Estatística *Kappa*	0,9837

Quadro 5 - Comparação entre os índices de acerto após a validação dos modelos gerados pelo algoritmo *J48* com CF=0.001 e CF=1.0

Nota-se que na fase de treinamento e validação o modelo com maior fator de confiança obteve melhores resultados nas três métricas analisadas. Apesar de o modelo mais simples ter apresentado menor taxa de erro médio absoluto após o teste, no processo de validação ele obteve índices inferiores aos do modelo mais complexo. Mitchell (1997) aponta que o teste com um conjunto de dados extra é a melhor forma de determinar a acurácia de modelos computacionais. Entretanto, também é importante realizar a interpretação do problema no domínio estudado, a fim de fazer melhores indicações de parâmetros de modelagem. Diante dessa informação, uma justificativa baseada no contexto do tucumã foi dada para embasar a escolha de qual é o melhor fator de confiança neste estudo.

Confrontando os dados expostos nos Quadros 4 e 5 poderíamos inferir que os fatores de confiança 0.001 e 1.0 são ambos aplicáveis para classificação de variedades de tucumã, com destaque para a árvore mais simples. Porém, existem outros fatores importantes a serem observados. Devemos levar em consideração que a árvore construída com CF=0.001 é composta por apenas dois atributos (*circunferência-vertical* e *coloração-do-mesocarpo*), sendo um deles um atributo nominal de cor. Os atributos de cor podem sofrer interferência humana em sua determinação por meio da carta de

cores, haja vista que não foi implementado nenhum *software* para caracterizar as colorações nesta pesquisa. Além disso, por ser uma planta alógama, a cada estação a morfologia do tucumã pode variar devido a fatores naturais. Neste sentido, o modelo com fator de confiança elevado (CF=1.0) pode garantir uma classificação mais segura, uma vez que possui mais regras e estas são implementadas com mais atributos, o que garante uma melhor avaliação de cada fruto antes de indicar a sua classe.

Com base nessas análises, o fator de confiança escolhido para prosseguir com as modelagens foi o de 1.0, pois o modelo gerado com ele apresentou melhores taxas na validação e os índices pós-teste também foram satisfatórios. Assim decidido, a matriz de confusão da validação do modelo com CF=1.0 (Tabela 7) foi analisada para encontrar onde se produziu o maior erro de classificação. Neste caso, observa-se que na classe C (tucumã-mesclado) quatro instâncias são classificadas como A (tucumã-vermelho). Para tentar melhorar as taxas de acerto foi aplicado o meta-classificador *CostSensitiveClassifier*. Na matriz de custo deste meta-classificador foi aumentado o índice de 1.0 para 2.0 na mesma posição em que está localizado o numero 4 na matriz de confusão da validação do modelo. Esse recurso é para tentar forçar o classificador a melhorar a predição nessa classe. Após a nova modelagem obteve-se uma configuração melhor da matriz de confusão com o uso do meta-classificador. O número de instâncias com classificação errada caiu de 11 para 7, como é mostrado na Tabela 8.

A	B	C	D	← Classificado como		Classe
222	0	3	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
2	0	223	0	C	=	Tucumã-mesclado
0	0	2	223	D	=	Tucumã-ararinha

Tabela 8 - Matriz de confusão da validação do modelo gerado pelo meta-classificador *CostSensitiveClassifier* usando o algoritmo *J48* - CF=1.0

Essa melhora ocorreu na fase de treinamento e validação, sendo necessário testar esse novo modelo para verificar se o meta-classificar também influencia na classificação de dados desconhecidos. A Tabela 9 mostra o resultado após o teste do modelo com o conjunto de dados extra.

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	1	49	D	=	Tucumã-ararinha

Tabela 9 - Matriz de confusão do teste do modelo gerado pelo meta-classificador *CostSensitiveClassifier* usando o algoritmo *J48* - CF=1.0

Após ser submetido à classificação dos novos dados, o modelo combinatório apresentou configuração idêntica ao modelo sem o meta-classificador, inclusive errando a predição da uma instância na mesma classe. Seu índice de acerto global e a estatística *Kappa* foram os mesmos do modelo anterior, com destaque apenas para uma redução no erro médio absoluto, que passou de 0,0192 para 0,0028. Uma comparação das taxas de acerto por classe (Quadro 6) mostra que na tarefa de separação das classes, os dois modelos obtiveram desempenho idêntico após os testes.

Antes do uso do meta-classificador		Após o uso do meta-classificador	
TP rate	Classe	TP rate	Classe
1	Tucumã-vermelho	1	Tucumã-vermelho
1	Tucumã-arara	1	Tucumã-arara
1	Tucumã-mesclado	1	Tucumã-mesclado
0,980	Tucumã-ararinha	0,980	Tucumã-ararinha

Quadro 6 - Comparação das taxas de acerto por classes nos testes dos modelos gerados pelo algoritmo J48 - CF=1.0 antes e depois do uso do meta-classificador *CostSensitiveClassifier*

Quanto ao uso do meta-classificador, observa-se na Tabela 8 que as três classes com predições erradas melhoraram após a validação. Porém, a matriz de confusão do teste desse modelo (Tabela 9) mostra que não houve melhora na predição de classes após o teste com dados desconhecidos, apresentando nesta etapa as mesmas taxas de TP *rate* que o modelo sem o uso do meta-classificador.

Perante os testes com novos dados de outras bases, a alteração do nível de custo de uma classe pode ser prejudicial às outras (WITTEN & FRANK, 2005). No caso desta pesquisa isso pode ser considerado verdadeiro, uma vez que a cada ano poderão surgir modificações na morfologia dos tucumãs. Neste sentido, manter alterado o custo de uma classe pode ser perigoso, pois não sabemos quais dados de tucumãs virão a ser submetidos aos modelos em pesquisas posteriores.

Para afirmar se o uso de uma meta-classificador de matriz de custo pode influenciar positivamente ou não na classificação de variedades do fruto, mais testes com dados de tucumãs de outras estações precisam ser feitos. Por esses motivos, o uso do meta-classificador *CostSensitiveClassifier* foi considerado inadequado para prosseguir com as modelagens, sendo apenas considerado como um recurso válido para tratar classes pontualmente, haja vista que na validação de seu modelo houve alguma melhora na predição de classes.

Diante dos dados apresentados, podemos concluir que a árvore construída com CF=0.25 apresenta *overffinting* errando a predição de 9 instâncias. Com CF=0.001 a árvore simplifica, havendo um aumento na taxa de acerto. Já com CF=1.0 a complexidade da árvore cresce, mas não ocorre *overfitting*, visto que os índices mantêm-se na média em relação ao modelo mais simples.

Por esses motivos, o melhor fator de confiança indicado para lidar com o conjunto de atributos do estudo é o de 1.0, sendo este escolhido para realizar as demais modelagens e comparações.

No que concerne à avaliação de atributos, podemos observar que as características *coloração-do-mesocarpo* e *circunferência-vertical* estão entre os principais construtores de regras das árvores criadas, mostrando que as inferências iniciais sobre esses dois atributos são válidas.

6.1.2 Resultados com o algoritmo *MultilayerPerceptron*

Com este algoritmo, inicialmente, foram treinados dois modelos com duas configurações de *validationSetSize* diferentes:

1. Modelo gerado com o algoritmo *MultilayerPerceptron* configurado com *validationSetSize*=0 (índice padrão).

A rede neural artificial gerada possui 13 nodos e apresentou as seguintes taxas na validação e teste:

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	100%	Taxa de acerto global	100%
Erro médio absoluto	0,0026	Erro médio absoluto	0,0023
Estatística *Kappa*	1	Estatística *Kappa*	1

Quadro 7 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - *validationSetSize*=0

Observando os resultados percebe-se que na fase de teste o erro médio absoluto diminui em relação ao da validação, o que significa que o modelo gerado é satisfatório para classificar instâncias desconhecidas. Para as quatro classes, as taxas de verdadeiro positivo foram iguais a 1, acertando a predição de todas as unidades tanto na validação

quanto no teste, os resultados das classificações são mostrados nas matrizes de confusão das Tabelas 10 e 11.

A	B	C	D	← Classificado como		Classe
225	0	0	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
0	0	225	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 10 - Matriz de confusão da validação do modelo gerado pelo algoritmo *MultilayerPerceptron* - *validationSetSize*=0

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 11 - Matriz de confusão do teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - *validationSetSize*=0

2. Modelo gerado com o algoritmo *MultilayerPerceptron* configurado com *validationSetSize=10*.

A rede neural artificial gerada também possui 13 nodos e apresentou as seguintes taxas na validação e teste:

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	99,8889%	Taxa de acerto global	100%
Erro médio absoluto	0,0029	Erro médio absoluto	0,0024
Estatística *Kappa*	0,9985	Estatística *Kappa*	1

Quadro 8 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - *validationSetSize*=10

Com a alteração do parâmetro *validationSetSize* de 0 para 10, constata-se que o modelo erra a predição de uma instância de tucumã-mesclado. Porém, após o teste com os dados desconhecidos o modelo acerta todas as 200 instâncias, apresentando taxa de acerto global e estatística *Kappa* idênticas as do modelo anterior, tendo apenas uma diferença no erro médio absoluto. As matrizes de confusão de validação e teste desse modelo são apresentadas nas Tabelas 12 e 13.

A	B	C	D	← Classificado como		Classe
225	0	0	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
1	0	224	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 12 - Matriz de confusão da validação do modelo gerado pelo algoritmo *MultilayerPerceptron - validationSetSize*=10

Na validação do modelo, para as classes tucumã-vermelho, tucumã-arara e tucumã-ararinha as taxas de TP *rate* foram iguais a 1, e para a classe tucumã-mesclado a taxa foi de 0,996.

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 13 - Matriz de confusão do teste do modelo gerado pelo algoritmo *MultilayerPerceptron - validationSetSize*=10

Nos primeiros dois modelos computacionais gerados com RNAs, as taxas foram excelentes quanto à predição de classes de tucumã, mesmo usando diferentes números de *validationSetSize*. Porém, como já mencionado, esse parâmetro varia de 0 a

99. Diante disso, o *validationSetSize* foi analisado apenas em caráter comparativo com o número de ciclos de validação cruzada indicada por Witten & Frank (2005). Portanto, o que se pode afirmar com base nesses experimentos, é que o aumento desse parâmetro para 10 influenciou negativamente o treinamento da RNA. Por isso, para as próximas modelagens não foi escolhido nenhum número de *validationSetSize*, deixando a rede treinar baseada no parâmetro de épocas de treinamento.

Os próximos modelos criados foram analisados com base no parâmetro de taxa de aprendizado. O objetivo da criação desses modelos é comparar o desempenho dos mesmos, usando três taxas de LR diferentes. O primeiro modelo apresentado nesta subseção já possui LR=0.3 sem alterar *validationSetSize*, por isso, apenas foi necessário criar mais dois modelos para testar os níveis de LR=0.1 e LR=1.0.

Após os treinamentos e testes, os novos modelos apresentaram os seguintes resultados:

3. Modelo gerado com o algoritmo *MultilayerPerceptron* configurado com LR=0.1.

A rede neural artificial gerada novamente possui 13 nodos e apresentou as seguintes taxas na validação e teste:

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	100%	Taxa de acerto global	100%
Erro médio absoluto	0,0045	Erro médio absoluto	0,004
Estatística *Kappa*	1	Estatística *Kappa*	1

Quadro 9 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.1

Esse modelo gerado com a menor taxa de aprendizado realizou a classificação correta de todas as instâncias, tanto na validação quanto no teste com conjunto de dados extra. As Tabelas 14 e 15 mostram os resultados das classificações nas duas fases.

A	B	C	D	← Classificado como		Classe
225	0	0	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
0	0	225	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 14 - Matriz de confusão da validação do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.1

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 15 - Matriz de confusão do teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.1

Para finalizar as modelagens com as diferentes taxas de aprendizado, um último modelo foi treinado com nível máximo para averiguar a influência desse fator nas RNAs.

4. Modelo gerado com o algoritmo *MultilayerPerceptron* configurado com LR=0.1.

A rede neural artificial gerada possui 13 nodos e apresentou as seguintes taxas na validação e teste:

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	99,8889%	Taxa de acerto global	100%
Erro médio absoluto	0,0017	Erro médio absoluto	0,013
Estatística *Kappa*	0,9985	Estatística *Kappa*	1

Quadro 10 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=1.0

Esse modelo realizou a classificação incorreta de uma instância de tucumã-mesclado como tucumã-vermelho após a validação, mas no teste com conjunto de dados extra o modelo acerta todas as 200 instâncias. As Tabelas 16 e 17 mostram as classificações realizadas com a taxa de aprendizado máxima nas duas fases.

A	B	C	D	← Classificado como		Classe
225	0	0	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
1	0	224	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 16 - Matriz de confusão da validação do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=1.0

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 17 - Matriz de confusão do teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=1.0

Depois da criação dos três modelos com níveis diferentes de LR, podemos observar que a taxa de aprendizado tem influência direta no treinamento dos modelos computacionais. Para comparar os desempenhos obtidos, os índices de cada um dos três modelos foram dispostos nos Quadros 11 e 12.

MultilayerPerceptron - Validação dos modelos (*k*=10)			
Nível de LR	LR=0.3	LR=0.1	LR=1.0
Taxa de acerto global	100%	100%	99,8889%
Erro médio absoluto	0,0026	0,0045	0,0017
Estatística *Kappa*	1	1	0,9985

Quadro 11 - Comparação entre os índices de validação dos modelos gerados pelo algoritmo *MultilayerPerceptron* com três níveis de LR

MultilayerPerceptron - Testes dos modelos (*supplied test set*)			
Nível de LR	LR=0.3	LR=0.1	LR=1.0
Taxa de acerto global	100%	100%	100%
Erro médio absoluto	0,0023	0,004	0,0013
Estatística *Kappa*	1	1	1

Quadro 12 - Comparação entre os índices de testes dos modelos gerados pelo algoritmo *MultilayerPerceptron* com três níveis de LR

Com base em uma análise dos resultados apresentados, podemos constatar que as três taxas de LR apresentaram bons índices tanto nas validações quanto nos testes. Porém, o modelo com nível máximo de aprendizado errou a classificação de uma instância na validação, apesar de no teste este ter apresentado melhor erro médio absoluto entre os três. Diante desse cenário, optou-se em prosseguir com as modelagens utilizando a taxa padrão de LR=0.3 indicada por Witten & Frank (2005), haja vista que a média dos índices desse modelo é melhor se consideradas as fases de validação e teste.

6.1.3 Resultados obtidos com a avaliação de atributos

No início deste estudo pôde-se constatar por meio de análises dos gráficos gerados por atributos, que *circunferência-vertical* e *coloração-do-mesocarpo* eram os melhores atributos separadores de classes. Se analisarmos as condições de decisão das árvores construídas pelo algoritmo *J48*, também conseguimos perceber que esses atributos, principalmente o primeiro, são decisivos na predição da classe correta.

Para fazer a prova dessa inferência utilizou-se a guia *Select attributes* para aplicar 3 métodos de avaliação de atributos, a fim de averiguar quais são os mais relevantes para a classificação de variedades de tucumã. A seguir são apresentados os resultados obtidos por meio das três abordagens escolhidas.

Método 01 - Avaliação individual de atributos (*Ranker*) (método de busca: *Ranker;* método de avaliação: *InfoGainAttributeEval)*

O método *Ranker* ordena os atributos e mostra duas informações: o *Average Merit* e o *Average Rank*, ambos com desvio padrão. O primeiro se trata da medida de correlações nos cinco ciclos de validação cruzada executados. O segundo se refere à ordem média em que um atributo ficou em cada um dos ciclos de validação. A Tabela 18 apresenta a lista ordenada dos atributos avaliados por esse método.

Average merit	*Average rank*	(n°) Atributo
1,593 +- 0,005	1 +- 0	9 *coloração-do-mesocarpo*
1,266 +- 0,007	2 +- 0	3 *coloração-do-epicarpo*
1,119 +- 0,025	3 +- 0	2 *circunferência-vertical*
0,965 +- 0,008	4 +- 0	7 *peso-do-endocarpo*
0,908 +- 0,014	5 +- 0	8 *peso-do-fruto-inteiro*

0,822 +- 0,012	6 +- 0	6 *peso-do-mesocarpo*
0,750 +- 0,016	7 +- 0	1 *circunferência-horizontal*
0,733 +- 0,014	8 +- 0	5 *peso-do-epicarpo*
0,038 +- 0,008	9 +- 0	4 *presença-de-rachaduras*

Tabela 18 - Seleção de atributos com o método de busca *Ranker* e o método de avaliação *InfoGainAttributeEval*

Com o resultado do primeiro método de avaliação podemos ver que os três primeiros atributos possuem índice de mérito maior que 1 e desvio padrão próximo de 0, ou seja, nos cinco ciclos de validação cruzada esses atributos ficaram na mesma ordem de seleção. Além disso, podemos constatar que o pior atributo de separação é *presença-de-rachaduras* com mérito de 0,038.

Método 02 - Avaliação de conjuntos de atributos (*Filter*) (método de busca: *GreedyStepwise;* método de avaliação: *CfsSubsetEval*)

Para realizar mais uma prova foi aplicado um método de filtro para averiguar novamente os atributos mais relevantes. O *CfsSubsetEval* seleciona subconjuntos de atributos e os avalia de acordo com sua relevância em relação a separação correta das classes. Os resultados de sua execução são apresentados na Tabela 19.

Número de ciclos de validação cruzada (%)	Atributo
0(0 %)	1 *circunferência-horizontal*
5(100 %)	2 *circunferência-vertical*
5(100 %)	3 *coloração-do-epicarpo*
0(0 %)	4 *presença-de-rachaduras*
5(100 %)	5 *peso-do-epicarpo*

1(20 %)	6 *peso-do-mesocarpo*
2(40 %)	7 *peso-do-endocarpo*
3(60 %)	8 *peso-do-fruto-inteiro*
5(100 %)	9 *coloração-do-mesocarpo*

Tabela 19 - Seleção de subconjuntos de atributos com o método de busca *GreedyStepwise* e o método de avaliação *CfsSubsetEval*

Os dados mostram que os três atributos apontados como melhores pelo primeiro método também estão entre os quatro melhores apontados neste segundo método. Os atributos *circunferência-vertical, coloração-do-epicarpo, peso-do-epicarpo* e *coloração-do-mesocarpo* são selecionados em todos os cinco ciclos de validação, seguidos dos demais atributos.

Método 03 - Avaliação de Conjuntos de atributos com base em algoritmos de classificação (*Wrapper*) (método de busca: *GreedyStepwise;* método de avaliação: *ClassifierSubsetEval*)

Até esta etapa ficou comprovado o grau de importância de cada atributo quanto à classificação de variedades de tucumã. Porém, a seleção de atributos pode tanto melhorar quanto piorar o desempenho de um algoritmo na geração de um modelo. Neste sentido, um último método de avaliação de atributos foi aplicado para averiguar qual a importância de cada atributo em relação à dinâmica de treinamento dos algoritmos. Para essa avaliação foram usados os algoritmos *J48* com CF=1.0 e *MultilayerPerceptron* com LR=0.3. Depois de submetidos à avaliação com validação cruzada de k=5, os métodos utilizados nessa abordagem apresentaram os dados exibidos na Tabela 20.

Algoritmo *J48* com CF=1.0		Algoritmo *MultilayerPerceptron* com LR=0.3	
Número de ciclos de validação cruzada (%)	Atributos	Número de ciclos de validação cruzada (%)	Atributos
1(20 %)	1 *circunferência-horizontal*	3(60 %)	1 *circunferência-horizontal*
5(100 %)	2 *circunferência-vertical*	5(100 %)	2 *circunferência-vertical*
1(20 %)	3 *coloração-do-epicarpo*	4(80 %)	3 *coloração-do-epicarpo*
0(0 %)	4 *presença-de-rachaduras*	0(0 %)	4 *presença-de-rachaduras*
0(0 %)	5 *peso-do-epicarpo*	0(0 %)	5 *peso-do-epicarpo*
0(0 %)	6 *peso-do-mesocarpo*	0(0 %)	6 *peso-do-mesocarpo*
3(60 %)	7 *peso-do-endocarpo*	0(0 %)	7 *peso-do-endocarpo*
0(0 %)	8 *peso-do-fruto-inteiro*	0(0 %)	8 *peso-do-fruto-inteiro*
5(100 %)	9 *coloração-do-mesocarpo*	5(100 %)	9 *coloração-do-mesocarpo*

Tabela 20 - Seleção de subconjuntos de atributos com o método de busca *GreedyStepwise* e método de avaliação *ClassifierSubsetEval*

Nesse último teste de avaliação de atributos fica confirmada a hipótese de que os atributos *circunferência-vertical* e *coloração-do-mesocarpo* são os dois mais importantes para uma separação mais clara das classes de tucumãs. Nota-se que tanto o algoritmo *J48* quanto o *MultilayerPerceptron*, ambos selecionaram os dois atributos em todos os cinco ciclos de validação cruzada, validando novamente a inferência feita no início da pesquisa.

6.1.4 Resultados obtidos com a seleção automática de atributos

Para averiguar se uma seleção automática de atributos melhora ou não as taxas de acerto dos modelos, foi aplicado um meta-classificador chamado *AttributeSelectedClassifier*. A avaliação dos atributos no processo de seleção automática contou com o uso de um método *Filter*.

Antes de treinar os modelos computacionais esse método realizou uma avaliação de quais atributos são mais relevantes para a separação das classes. Somente após essa avaliação é que o meta-classificador treinou os modelos com base nos algoritmos e parâmetros configurados. Na avaliação da abordagem *Filter* foram utilizados o método de busca *BestFirst* e o método de avaliação *CfsSubsetEval*. O Quadro 13 mostra quais foram os atributos indicados como melhores após o pré-processamento pelos métodos utilizados.

Método de avaliação de atributos *Filter*	
N° do atributo	Atributos Selecionados
2	*circunferência-vertical*
3	*coloração-do-epicarpo*
5	*peso-do-epicarpo*
6	*peso-do-mesocarpo*
7	*peso-do-endocarpo*
9	*coloração-do-mesocarpo*

Quadro 13 - Atributos selecionados pelo método de avaliação de atributos *Filter*

Nota-se que com a abordagem de filtro foram selecionados seis atributos. A lista apresentada mostra os atributos em ordem de posição no conjunto de dados, ou seja, não é levada em consideração nessa etapa a ordenação por mérito de cada um. A

seguir são apresentados os resultados em cada algoritmo de classificação, mostrando como eles usaram os atributos selecionados por esse método para construir seus modelos preditivos.

1. Modelo gerado com seleção automática de atributos pelo algoritmo *J48* - CF=1.0.

Esse modelo foi criado com base na seleção de atributos feita pelo método *Filter* usando o meta-classificador *AttributeSelectedClassifier*. A árvore de decisão construída possui 10 folhas e 13 nós, implementando as regras a seguir:

```
J48 pruned tree
------------------
coloracao-do-mesocarpo = LARANJA
|   circunferencia-vertical <= 53.6: TUCUMAVERMELHO
|   circunferencia-vertical > 53.6
|   |   coloracao-do-epicarpo = AMARELOESCURO: TUCUMAVERMELHO
|   |   coloracao-do-epicarpo = AMARELO: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = VERDECLARO: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = VERDEESCURO: TUCUMAMESCLADO
|   |   coloracao-do-epicarpo = AMARELOEVERDE: TUCUMAMESCLADO
coloracao-do-mesocarpo = AMARELOCLARO: TUCUMAARARA
coloracao-do-mesocarpo = LARANJAESCURO: TUCUMAMESCLADO
coloracao-do-mesocarpo = AMARELO: TUCUMAARARINHA
coloracao-do-mesocarpo = AMARELOESCURO: TUCUMAARARINHA
```

Analisando a árvore construída nesse modelo podemos perceber que para a formação das regras, apenas um subconjunto de três atributos foi escolhidos pelo *J48* (*circunferência-vertical*, *coloração-do-epicarpo* e *coloração-do-mesocarpo*). Depois dos dez ciclos de validação cruzada e da realização do teste com o conjunto de dados extra, a árvore de decisão gerada com os três atributos apresentou os índices exibidos no Quadro 14.

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	99,7778 %	Taxa de acerto global	99%
Erro médio absoluto	0,0016	Erro médio absoluto	0,0054
Estatística *Kappa*	0,997	Estatística *Kappa*	0,9867

Quadro 14 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *J48* - CF=1.0 com seleção automática de atributos

Constata-se que esse modelo apresentou erros na classificação de objetos nas fases de validação e teste. As matrizes de confusão das Tabelas 21 e 22 mostram onde ocorreram os erros de classificação em cada etapa.

A	B	C	D	← Classificado como		Classe
224	0	1	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
1	0	224	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 21 - Matriz de confusão da validação do modelo gerado pelo algoritmo *J48* - CF=1.0 com seleção automática de atributos

A	B	C	D	← Classificado como		Classe
48	0	2	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 22 - Matriz de confusão do teste do modelo gerado pelo algoritmo *J48* - CF=1.0 com seleção automática de atributos

Observando os erros nas matrizes de confusão, percebe-se que quando o número de atributos é reduzido, o modelo erra a predição de duas instâncias na validação e duas no teste (mesmo com a árvore de decisão configurada com o maior

fator de confiança). Como já vimos anteriormente, após a seleção de atributos com o método *Filter*, a construção da árvore foi feita com apenas três dos atributos avaliados, sendo que dois deles são atributos nominais de cor. Essa redução do número de atributos, assim como ocorre na árvore com CF=0.001, pode ser prejudicial no contexto da classificação de variedades de tucumã devido a sua palmeira apresentar fecundação cruzada.

Diante desses resultados apresentados, considera-se como verdadeira a hipótese de que a árvore de decisão construída com mais atributos em suas regras é melhor para a classificação no domínio do tucumã.

2. Modelo gerado com seleção automática de atributos pelo algoritmo *MultilayerPerceptron* com taxa de aprendizado padrão (LR=0.3).

Nesse modelo gerado com a técnica de redes neurais artificial, o meta-classificador realizou o treinamento usando um subconjunto de quatro atributos entre os seis considerados relevantes pelo método *Filter*. Os atributos usados para criar o modelo foram: *circunferência-vertical, coloração-do-epicarpo, peso-do-epicarpo, coloração-do-mesocarpo*. Nesse novo cenário, a rede criada possui 12 nodos e apresentou os resultados de validação e teste dispostos nas no Quadro 15.

Validação do modelo (*k*=10)		Teste do modelo (*supplied test set*)	
Taxa de acerto global	100%	Taxa de acerto global	100%
Erro médio absoluto	0,0027	Erro médio absoluto	0,0023
Estatística *Kappa*	1	Estatística *Kappa*	1

Quadro 15 - Comparação entre os índices de validação e teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.3 com seleção automática de atributos

As taxas de predição apresentadas pela RNA com seleção de atributos são satisfatórias. Observa-se que tanto na fase de validação quanto de teste o modelo não

errou nenhuma predição de classe, ainda havendo uma breve redução do erro médio absoluto. As Tabelas 23 e 24 mostram que esse modelo foi isento de erros.

A	B	C	D	← Classificado como		Classe
225	0	0	0	A	=	Tucumã-vermelho
0	225	0	0	B	=	Tucumã-arara
0	0	225	0	C	=	Tucumã-mesclado
0	0	0	225	D	=	Tucumã-ararinha

Tabela 23 - Matriz de confusão da validação do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.3 com seleção automática de atributos

A	B	C	D	← Classificado como		Classe
50	0	0	0	A	=	Tucumã-vermelho
0	50	0	0	B	=	Tucumã-arara
0	0	50	0	C	=	Tucumã-mesclado
0	0	0	50	D	=	Tucumã-ararinha

Tabela 24 - Matriz de confusão do teste do modelo gerado pelo algoritmo *MultilayerPerceptron* - LR=0.3 com seleção automática de atributos

Na classificação das variedades de tucumã, podemos perceber que após a seleção de atributos, mesmo reduzindo o numero de atributos de 9 para 4, a RNA continua com ótimos índices de predição.

Esse nível de acerto apresentado por esse modelo é relativamente esperado, pois como visto na subseção 2.2.2, as RNAs possuem a capacidade de lidar com problemas complexos, e em alguns casos, conseguem generalizar os modelos com base em poucas informações disponíveis.

Após o término de todas as modelagens foi necessário analisar os resultados para indicar quais os melhores modelos preditivos de variedades de *Astrocaryum aculeatum*.

Nesta pesquisa, o tempo de construção dos modelos não foi usado como parâmetro de comparação entre eles, pois os algoritmos utilizados são de técnicas bem diferentes. Desde a fase de levantamento bibliográfico, já se tinha o conhecimento de que a técnica de RNAs é a que apresenta o maior custo computacional. Também não é viável compará-los quanto ao nível de compreensão, pois as RNAs são como caixas pretas, e não revelam muitos detalhes após a finalização do treinamento. Ao contrário das RNAs as árvores de decisão são mais fáceis de interpretar e compreender sua dinâmica na classificação de objetos. Por esses motivos, na comparação dos modelos foram usados os três índices de acurácia escolhidos para esta pesquisa.

Diante do exposto, os desempenhos dos modelos iniciais foram comparados com os índices dos modelos gerados após o teste com seleção automática de atributos, apresentando o cenário exposto no Quadro 16.

Algoritmo *J48* com CF=1.0			
Modelo gerado sem seleção de atributos		Modelo gerado com seleção de atributos	
Taxa de acerto global	99,5%	Taxa de acerto global	99%
Erro médio absoluto	0,0192	Erro médio absoluto	0,0054
Estatística *Kappa*	0,9933	Estatística *Kappa*	0,9867
Algoritmo *MultilayerPerceptron* com LR=0.3			
Modelo gerado sem seleção de atributos		Modelo gerado com seleção de atributos	
Taxa de Acertos	100%	Taxa de Acertos	100%
Erro médio absoluto	0,0023	Erro médio absoluto	0,0023
Índice *Kappa*	1	Índice *Kappa*	1

Quadro 16 - Comparação entre os modelos gerados pelos algoritmos *J48* com CF=1.0 e *MultilayerPerceptron* - LR=0.3, antes e depois da seleção automática de atributos

Diante destes dados, podemos concluir que não houve mudanças drásticas no desempenho dos modelos gerados com a seleção automática de atributos em relação aos

modelos anteriores a ela. Com base nessa comparação nota-se que no caso das árvores de decisão a seleção automática de atributos piora o desempenho do modelo, sendo melhor utilizar o conjunto de atributos inteiro e deixar o modelo escolher por si próprio quais atributos utilizar para construir as regras. Já nas RNAs percebe-se que o desempenho foi o mesmo, tanto com os nove atributos quanto após a redução para quatro no processo de seleção automática. Todos os modelos apresentados obtiverem taxas de erro médio absoluto próximas de 0, provando que os resultados após o treinamento, validação e teste desses modelos foram satisfatórios para a classificação de variedades de tucumã.

Ao final dos experimentos de classificação, fica provado que os melhores atributos preditivos são *coloração-do-mesocarpo* e *circunferência-vertical*, haja vista que esses dois atributos foram selecionados para a construção de todos os modelos gerados na tarefa de classificação deste estudo.

6.2 Resultados da modelagem computacional na tarefa de agrupamento

Os modelos criados para agrupamento das variedades de tucumã foram treinados usando o conjunto de dados sem a influência do atributo-alvo. As medidas de distância Euclidiana e de *Manhattan* foram aplicadas aos modelos para averiguar como os *clusters* são formados baseando-se na dinâmica de cálculo dessas medidas.

Para gerar os agrupamentos, o parâmetro de números de *clutsers* foi alterado de 2 (padrão WEKA) para 4 (número de variedades selecionadas para o estudo). A efeito de comparação, os dados foram trabalhados em dois cenários. No primeiro, as variáreis nominais foram convertidas para binárias e todo o conjunto de dados foi normalizado para tentar igualar os pesos dos valores nos atributos. Ao final desse tratamento, a base

de dados ficou com um total de 17 atributos. Já no segundo cenário, os dados não foram pré-processados, deixando que cada algoritmo aliado a uma medida de distância usasse suas próprias funções internas para lidar com os dados originais.

Os testes de todos os modelos foram feitos com o conjunto de dados extra, realizando-se a comparação entre os resultados das modelagens feitas nesta etapa da pesquisa.

6.2.1 Resultados com o algoritmo *SimpleKMeans*

Para compreender a análise realizada sobre os modelos, primeiramente é preciso conhecer a notação usada pelo *SimpleKMeans* para representar os agrupamentos. Cada *cluster* foi numerado e rotulado com uma classe, permitindo que se fizesse a interpretação dos resultados obtidos. O Quadro 17 mostra como o algoritmo realizou a rotulação de cada *cluster*.

Número do *Cluster*	Classe com a qual ele está rotulado
Cluster 0	Tucumã-mesclado
Cluster 1	Tucumã-ararinha
Cluster 2	Tucumã-arara
Cluster 3	Tucumã-vermelho

Quadro 17 - Notação dos *clusters* atribuída pelo algoritmo *SimpleKMeans*

Essa atribuição de número e classe aos *clusters* foi a mesma em todos os modelos gerados na pesquisa.

No primeiro cenário, onde os dados foram pré-processados, dois modelos foram criados com cada uma das medidas de distância, obtendo os seguintes resultados:

1. Modelo gerado com o *SimpleKMeans* usando a medida de distância Euclidiana com os dados normalizados (17 atributos).

Na geração desse modelo foram realizadas 6 iterações para descobrir os melhores centróides. A formação dos *clusters* no treinamento é mostrada no Quadro 18.

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	140	16%
1	108	12%
2	374	42%
3	278	31%

Quadro 18 - *Clusters* formados no treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados normalizados

Observa-se nesses resultados que o *Cluster* 2 (tucumã-arara) é o mais desbalanceado em relação aos demais. O esperado era que aproximadamente cerca de 25% de instâncias fossem alocadas corretamente em cada *cluster*. Para entender onde houve os maiores erros no agrupamento das variedades, a matriz de erros do treinamento desse modelo é apresentada na Tabela 25.

0	1	2	3	← Alocação no *Cluster*
140	62	0	23	=Tucumã-mesclado
0	9	149	67	=Tucumã-ararinha
0	0	225	0	=Tucumã-arara
0	37	0	188	=Tucumã-vermelho

Tabela 25 - Matriz de erros do treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados normalizados

Analisando a alocação das instâncias, percebe-se que esse modelo fez 338 indicações incorretas entre os 900 exemplos, apresentando uma taxa de erro de

37,5556%. A variedade tucumã-ararinha é que mais possui unidades com indicação errada em outros *clusters*, sendo que a maioria de suas instâncias (216) foi classificada como tucumã-arara e tucumã-vermelho.

Para testar o desempenho desse modelo, o conjunto de dados extra foi submetido, gerando a seguinte configuração:

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	31	16%
1	23	12%
2	88	44%
3	58	29%

Quadro 19 - *Clusters* formados no teste do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados normalizados

O *software* WEKA não fornece índice de incorreção ou matriz de erros após os teste dos modelos, apresentando apenas a formação final dos *clusters*. Diante disso, a interpretação dos modelos pode ser feita analisando os gráficos e informações das instâncias agrupadas em cada teste.

Após uma análise realizada nos *clusters* formados na fase de teste (Quadro 19), nota-se que no *Cluster* 2 (tucumã-arara) foram alocadas mais unidades do que nos demais *clusters*. Todas as instâncias agrupadas erroneamente no *Cluster* 2 são de tucumã-ararinha. E no *Cluster* 3 (tucumã-vermelho), as unidades indicadas erroneamente são de tucumã-ararinha e de tucumã-mesclado.

Alguns desses erros podem ser justificados pelas semelhanças morfológicas entre os dois pares das variedades, onde tucumã-arara e tucumã-ararinha têm características parecidas, e tucumã-vermelho tem muita semelhança com tucumã-

mesclado, haja vista que os valores de peso e coloração são muito parecidos entre os pares citados.

Mesmo sabendo-se disso, os resultados desses modelos não foram satisfatórios com relação à separação ideal das variedades, uma vez que a classe tucumã-ararinha foi muito prejudicada no agrupamento. Porém, mais testes precisaram ser feitos para embasar melhor essas inferências.

2. Modelo gerado com o *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados normalizados (17 atributos).

Para encontrar os centróides ideais para esse modelo foram necessárias 7 iterações com a distância de *Manhattan*. A configuração dos *clusters* após o treinamento é apresentada no Quadro 20.

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	271	30%
1	204	23%
2	225	25%
3	200	22%

Quadro 20 - *Clusters* formados no treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados normalizados

Com essa configuração dos dados, observa-se que nesse modelo, o *Cluster* 0 (tucumã-mesclado) apresenta maior concentração de instâncias do que os demais. O algoritmo erra a indicação de 201 instâncias, obtendo taxa de incorreção de 22,3333%. Neste caso, a classe mais prejudicada foi tucumã-vermelho, a qual não apresentou nenhuma correção na indicação de suas instâncias na fase de treinamento. A Tabela 26 mostra as alocações nos *clusters* com essa medida de distância no cenário de dados atual.

0	1	2	3	← Alocação no *Cluster*
160	65	0	0	=Tucumã-mesclado
0	25	0	200	=Tucumã-ararinha
0	0	225	0	=Tucumã-arara
111	114	0	0	=Tucumã-vermelho

Tabela 26 - Matriz de erros do treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados normalizados

Como visto, todas as instâncias de tucumã-vermelho são agrupadas em outros *clusters* diferentes. Assim como no modelo anterior, a classe tucumã-ararinha também foi prejudicada, sendo a segunda pior com 200 instâncias indicadas erroneamente. Após o teste com os exemplos extras esse modelo apresentou o resultado a seguir:

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	75	38%
1	31	16%
2	50	25%
3	44	22%

Quadro 21 - *Clusters* formados no teste do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados normalizados

Analisando as instâncias agrupadas no teste desse modelo, identificou-se que o *Cluster* 0 (tucumã-mesclado) foi formado pelas instâncias errôneas de tucumã-vermelho e as unidades corretas de tucumã-mesclado. Também foi constatado que o *Cluster* 1 (tucumã-ararinha) foi o mais heterogêneo, possuindo as instâncias corretas de tucumã-ararinha e as incorretas de tucumã-vermelho e tucumã-mesclado. Já o *Cluster* 3 (tucumã-vermelho) foi todo formado por instâncias incorretas de tucumã-ararinha.

Diante dos dados, nota-se que os resultados dos testes nos dois modelos seguiram as tendências de erro apresentadas nos treinamentos. As indicações das variedades nos *clusters* foram bem diferentes com o uso das duas medidas de distância. O Quadro 22 mostra uma comparação da formação dos *clusters* nos dois modelos após os testes.

SimpleKMeans - Distância Euclidiana			*SimpleKMeans* - Distância de *Manhattan*		
N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias	N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	31	16%	0	75	38%
1	23	12%	1	31	16%
2	88	44%	2	50	25%
3	58	29%	3	44	22%

Quadro 22 - Comparação dos *clusters* formados nos testes dos modelos gerados pelo algoritmo *SimpleKMeans*, usando as medidas de distância Euclidiana e de *Manhattan* com os dados normalizados

No quadro comparativo, vemos que em todos os *clusters* existem diferenças no agrupamento das variedades. Percebe-se ainda que cada medida de distância beneficia classes distintas em cada modelo. A única variedade que apresentou indicação correta para todas as suas instâncias no treinamento e no teste, foi tucumã-arara.

Vale ressaltar que nessas modelagens foram utilizados os 17 atributos gerados com o pré-processamento dos dados. Neste sentido, pode ser que a conversão do formato desses atributos tenha influenciado negativamente no desempenho desses modelos. Para averiguar essa hipótese foram gerados dois modelos com as mesmas medidas de distância usando os dados originais com 9 atributos.

3. Modelo gerado com o *SimpleKMeans* usando a medida de distância Euclidiana com os dados originais (9 atributos).

Para a geração desse modelo o algoritmo realizou 7 iterações, produzindo os *clusters* dispostos no Quadro 23.

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	187	21%
1	194	22%
2	341	38%
3	178	20%

Quadro 23 - *Clusters* formados no treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados originais

Observando os números, vemos que assim como no modelo gerado com os dados tratados, o *Cluster* 2 (tucumã-arara) novamente é o mais desbalanceado. Porém, nesse modelo os demais *clusters* são aparentemente equilibrados. Na etapa de treinamento, este modelo errou a indicação de 217 instâncias, com taxa de incorreção de 24,1111%. Se comparada à taxa do primeiro modelo gerado com distância Euclidiana, houve uma boa melhora, errando 121 instâncias a menos. A Tabela 27 mostra como foram alocadas as instâncias em cada *cluster* na fase de treinamento.

0	1	2	3	← Alocação no *Cluster*
185	40	0	0	=Tucumã-mesclado
2	101	116	6	=Tucumã-ararinha
0	0	225	0	=Tucumã-arara
0	53	0	172	=Tucumã-vermelho

Tabela 27 - Matriz de erros do treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados originais

Observando os números na Tabela 27, constata-se que realmente os *clusters* são mais equilibrados. As quantidades das indicações erradas em cada variedade são menores se comparados com a matriz de erro do primeiro modelo (Tabela 25). Os maiores erros de agrupamento nesse modelo foram da variedade tucumã-ararinha, onde 116 unidades são indicadas como tucumã-arara. Para averiguar se essa melhora também ocorre com os dados desconhecidos, o conjunto de teste foi submetido ao modelo, gerando os seguintes *clusters*:

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	44	22%
1	34	17%
2	81	41%
3	41	21%

Quadro 24 - *Clusters* formados no teste do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância Euclidiana com os dados originais

Analisando as instâncias dispostas nos *cluster*, nota-se que na fase de teste este modelo também alocou melhor as unidades em cada variedade. Neste caso, o *Cluster* 1 (tucumã-ararinha) é o mais heterogêneo, pois recebeu algumas instâncias erradas de tucumã-vermelho e de tucumã-mesclado, porém, ainda possui uma quantidade razoável de elementos corretos de tucumã-ararinha. Já o *Cluster* 2 (tucumã-arara) possui todas as unidades corretas de tucumã-arara mais as unidades restantes de tucumã-ararinha que foram erroneamente indicadas a ele. No geral, esse modelo apresentou melhor desempenho entre dos dois gerados com distância Euclidiana.

Para fazer as últimas análises, mais um modelo foi criado com os dados originais usando a medida de distância de *Manhattan*.

4. Modelo gerado com o *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados originais (9 atributos).

Na construção desse modelo foram necessárias 9 iterações para definir os melhores centróides. Após o treinamento do modelo, os seguintes *clusters* foram criados:

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	242	27%
1	179	20%
2	234	26%
3	245	27%

Quadro 25 - *Clusters* formados no treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados originais

Com esse modelo percebemos que os *clusters* são formados mais balanceados do que o anterior gerado com essa distância. Quanto à taxa de incorreção, esse modelo apresentou 24,3333%, errando a indicação de 219 instâncias. Embora visualmente pareça mais equilibrado, é preciso averiguar como foi feita a alocação de cada unidade nos *clusters*. A Tabela 28 mostra como esse modelo realizou essa divisão.

0	1	2	3	← Alocação no *Cluster*
190	34	0	1	=Tucumã-mesclado
52	93	10	70	=Tucumã-ararinha
0	1	224	0	=Tucumã-arara
0	51	0	174	=Tucumã-vermelho

Tabela 28 - Matriz de erros do treinamento do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados originais

Observadas as indicações corretas de cada variedade na Tabela 28, vemos que esse modelo apresentou melhores formações de *clusters* do que o anterior (modelo com distância de *Manhattan* usando os dados tratados). Apesar de esse modelo errar a indicação de 18 instâncias a mais que o outro, ainda assim ele é considerado melhor, pois no primeiro, nenhuma unidade de tucumã-vermelho foi indicada corretamente ao *cluster* ideal, e 200 unidades de tucumã-ararinha foram agrupadas erroneamente. Com esse novo modelo ocorre o contrário, um número considerável de instâncias é alocado corretamente nos *clusters* formados.

Para testar esse modelo foram submetidos a ele os 200 exemplos desconhecidos, gerando a seguinte configuração dos *clusters*:

N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	54	27%
1	32	16%
2	51	26%
3	63	32%

Quadro 26 - *Clusters* formados no teste do modelo gerado pelo algoritmo *SimpleKMeans* usando a medida de distância de *Manhattan* com os dados originais

Após o teste, os *clusters* criados são um pouco mais desbalanceados do que na fase de treinamento. Observou-se que no *Cluster* 3 (tucumã-vermelho), além das unidades corretas de tucumã-vermelho estão presentes quase metade das unidades de tucumã-ararinha indicadas erroneamente. O *Cluster* 1 (tucumã-ararinha) é o que apresenta menor número de instâncias, contendo unidades de três variedades diferentes. Novamente, constata-se que tucumã-ararinha é a classe com o maior número de indicações erradas, tendo unidades alocadas erroneamente em três *clusters* diferentes.

O Quadro 27 apresenta a comparação dos *clusters* gerados pelos dois modelos, com as duas medidas de distância usando os dados originais.

SimpleKMeans - Distância Euclidiana			*SimpleKMeans* - Distância de *Manhattan*		
N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias	N° do *cluster*	Quantidade de instâncias no *cluster*	Percentagem em relação ao total de instâncias
0	44	22%	0	54	27%
1	34	17%	1	32	16%
2	81	41%	2	51	26%
3	41	21%	3	63	32%

Quadro 27 - Comparação dos *clusters* formados nos testes dos modelos gerados pelo algoritmo *SimpleKMeans*, usando as medidas de distância Euclidiana e de *Manhattan* com os dados originais

Em comparação ao cenário com os dados pré-processados (Quadro 22), os dois modelos criados com os dados originais realizaram melhor a tarefa de agrupamento. Isso quer dizer que o processo de conversão dos atributos nominais para binários e a posterior normalização dos dados, não são recomendados para o agrupamento das variedades com algoritmo *SimpleKMeans*, mostrando-se prejudicial com as duas distâncias usadas.

Entre os dois modelos gerados com os dados originais, aquele que pode ser considerado como melhor, foi o que usou a medida de distância Euclidiana, pois as distribuições das variedades foram mais homogêneas em cada *cluster*. Os maiores erros nesse modelo foram da variedade tucumã-ararinha, porém, a grande maioria das indicações errôneas dessa variedade foi no *cluster* de tucumã-arara. Isso é mais aceitável do que a dispersão dessas instâncias de tucumã-ararinha em outros *clusters*

(como ocorreu com a distância de *Manhattan*), haja vista que morfologicamente o tucumã-ararinha se assemelha mais ao tucumã-arara do que a qualquer outra variedade.

No entanto, mesmo embasando essa escolha no contexto do tucumã, mais modelagens precisam ser feitas com mudança de outros parâmetros para tentar melhorar o desempenho dos modelos, pois como discutido, em todos os cenários a variedade tucumã-ararinha foi a que mais apresentou erros no agrupamento de suas instâncias.

6.3. Análise das variedades de tucumã em relação ao teor de polpa

A procura pela polpa do tucumã aumentou nos últimos anos em relação ao comércio do fruto (DIDONET, 2012). Nos períodos de colheita, os produtores escolhem as variedades que consideram mais rentáveis para o beneficiamento, destinando as demais para a venda do fruto inteiro. Neste contexto, a avaliação do teor de polpa das variedades estudadas foi pertinente no que concerne a venda e a utilização da polpa *in natura*.

Para realizar estas estimativas, duas dinâmicas foram aplicadas neste estudo. A primeira consistiu na análise dos gráficos gerados pelo WEKA, para realizar inferências sobre as variedades com relação ao seu mesocarpo. A segunda foi realizar a prova dessas hipóteses inferidas inicialmente, por meio de cálculos matemáticos da média aritmética dos atributos de peso de cada variedade.

Os gráficos em três dimensões foram gerados por meio de um pacote especial chamado *scatterPlot3D* instalado na versão de desenvolvedor do *software* WEKA (versão 3.7.12). No primeiro gráfico gerado, as informações dos atributos *peso-do-fruto-inteiro, peso-do-mesocarpo* e o atributo-alvo *tipo* foram cruzadas no WEKA, sendo seu resultado mostrado na Figura 16.

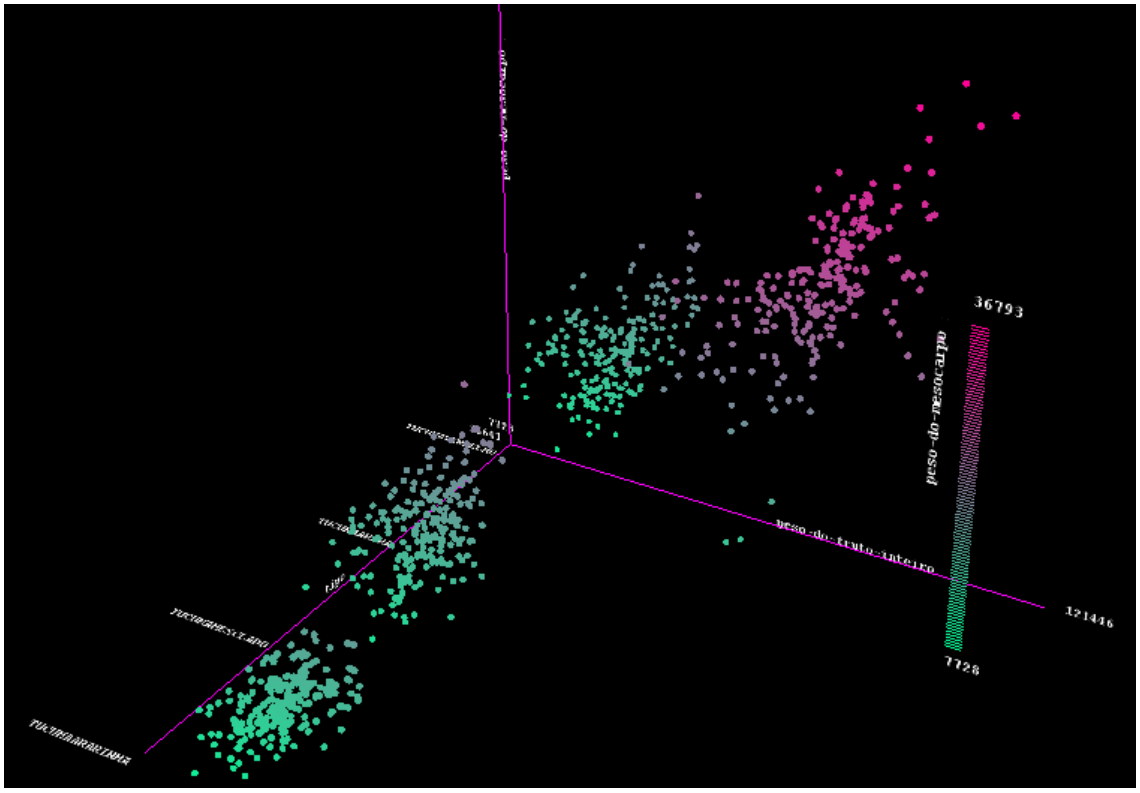

Figura 16 - Gráfico em 3D da separação das instâncias por interseção dos atributos *peso-do-fruto-inteiro*, *peso-do-mesocarpo* e o atributo-alvo *tipo*

No eixo *x* estão os valores dos pesos dos frutos, no eixo *y* estão os pesos das polpas e no eixo *z* encontram-se as variedades de tucumã. A coloração adotada representa uma escala que vai do mais verde ao mais roxo. Quanto mais próximo do roxo, que dizer que mais polpa a unidade possui. Nesta primeira análise o peso dos frutos não está diretamente relacionado com o peso da polpa, sendo apenas usado para ajudar na separação dos grupos no gráfico.

Analisando a imagem, constata-se que a variedade com mais unidades coloridas com as tonalidades de roxo são do tipo tucumã-arara. Nas próximas posições estão praticamente empatados o tucumã-mesclado, o tucumã-vermelho e o tucumã-ararinha. Para obter outra visão das instâncias, um segundo gráfico foi gerado alterando o eixo *x* para o atributo representante dos pesos das polpas (*peso-do-mesocarpo*). A Figura 17 mostra como ficou essa nova configuração.

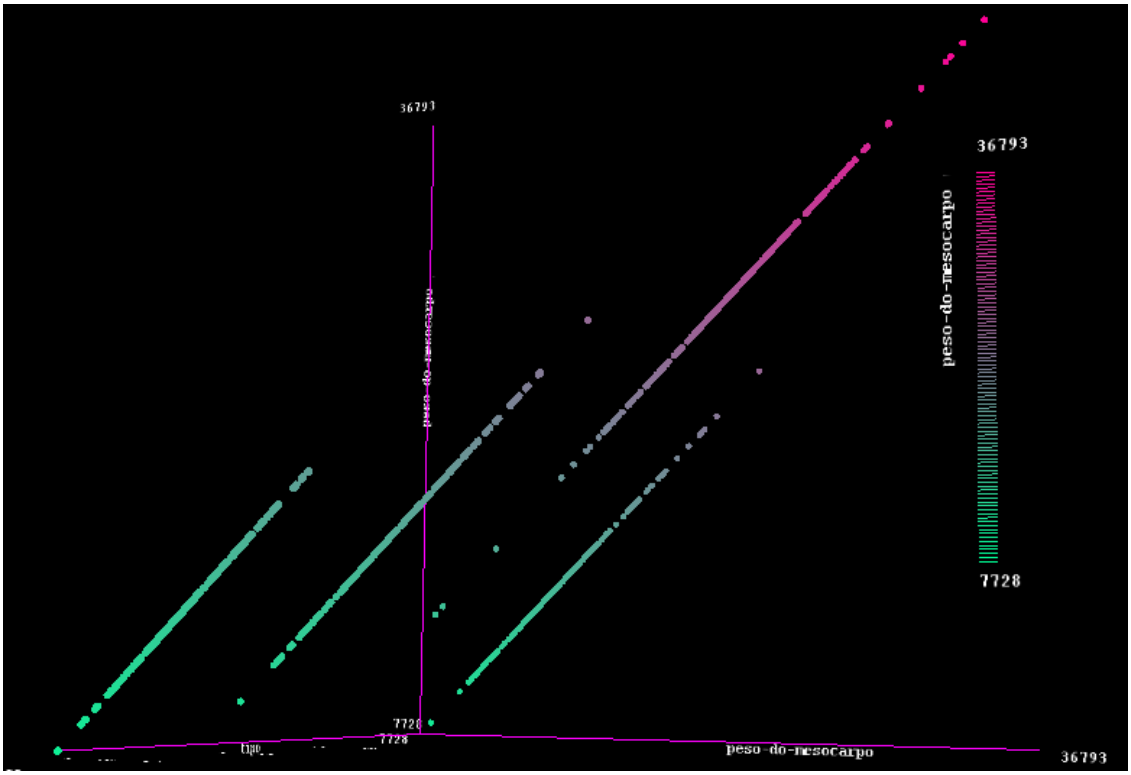

Figura 17 - Gráfico em 3D da separação das instâncias por interseção dos atributos *peso-do-fruto-inteiro*, *peso-do-mesocarpo* e o atributo-alvo *tipo*

Nessa nova imagem podemos ver quatro fileiras de instâncias de cada tipo de tucumã, organizadas da esquerda para a direita na seguinte ordem: tucumã-ararinha, tucumã-mesclado, tucumã-arara e tucumã-vermelho. Novamente, pela análise da coloração, podemos ver que tucumã-arara possui o maior número de unidades com elevado teor de polpa.

Essa inferência foi feita com base na análise visual dos gráficos separadores gerados no WEKA, porém, os atributos referentes ao peso do epicarpo, do mesocarpo e do fruto inteiro dos tucumãs, não foram levados em consideração para realizar essa estimativa inicial. Diante disso, foi aplicado o cálculo matemático da média aritmética de cada atributo para indicar qual das variedades possui maior valor médio do teor de polpa em relação ao peso do fruto inteiro.

Na Tabela 29 são apresentados os resultados dos cálculos das médias dos atributos de peso de cada variedade baseados nos 1100 exemplos do conjunto de dados.

Variedade	Peso Médio do Epicarpo (mg)	Peso Médio do Mesocarpo (mg)	Peso Médio do Endocarpo (mg)	Peso Médio do fruto inteiro (mg)
Tucumã-vermelho	8.866	14.587	30.732	54.185
Tucumã-arara	13.642	26.410	57.948	98.000
Tucumã-mesclado	13.320	15.877	31.817	61.014
Tucumã-ararinha	11.502	12.810	33.808	58.120

Tabela 29 - Médias dos atributos de peso de cada uma das quatro variedades de tucumã

Depois de obtidos os valores das médias dos atributos de peso, foi necessário calcular quanto cada parte do tucumã representa do seu valor total. A Tabela 30 apresenta o resultado final dos cálculos de estimativa das médias do teor de polpa, do peso da casca e do peso do caroço das quatro variedades de tucumã.

Variedade	Epicarpo	Mesocarpo	Endocarpo	Total
Tucumã-vermelho	16,36%	26,92%	56,72%	100%
Tucumã-arara	13,92%	26,95%	59,13%	100%
Tucumã-mesclado	21,83%	26,02%	52,15%	100%
Tucumã-ararinha	19,79%	22,04%	58,17%	100%

Tabela 30 - Médias dos atributos de peso do epicarpo, mesocarpo e endocarpo em relação à média do peso do fruto inteiro de cada uma das quatro variedades de tucumã

Com base na análise dos resultados, fica provada a hipótese de que o tucumã-arara é a variedade com maior valor médio de teor de polpa. Porém, os índices do tucumã-vermelho e do tucumã-mesclado ficaram bem próximos aos do líder tucumã-arara, restando na última posição o tucumã-ararinha com a menor média alcançada.

Como já citado, um produtor geralmente realiza a escolha empírica de quais variedades irá beneficiar e de quais irá vender o fruto inteiro por quantidade. Neste

sentido, com base nos resultados encontrados, algumas indicações podem ser feitas: o tucumã-arara e o tucumã-vermelho seriam os mais lucrativos para a venda da polpa *in natura*, e o tucumã-mesclado e o tucumã-ararinha seriam ideais para a venda por unidade do fruto inteiro, gerando mais lucro no comércio das variedades de tucumã.

Na venda da polpa os produtores obtêm um lucro mais elevado do que na venda do fruto inteiro. O processo de despolpamento artesanal é demorado e demanda habilidades manuais. Para aumentar a produção de polpa, alguns comerciantes adotam como estratégia associar-se ou contratar funcionários especificamente para essa atividade. Dessa forma, por ser uma tarefa difícil e apresentar baixo rendimento no processo manual, o preço do kg da polpa chega a ser 50% mais caro do que o kg do fruto inteiro (DIDONET, 2012).

O mercado do tucumã vem ganhando força a cada ano, com destaque para o comércio da polpa. Em um dos trabalhos pioneiros sobre a importância econômica do tucumã, Kahn & Moussa (1999) apontaram pouca ou nenhuma importância da comercialização da polpa no mercado à época de sua pesquisa. No entanto, aproximadamente uma década depois Didonet (2012) apresentou resultados que mostram que 53% dos frutos que entraram nas feiras e mercados de Manaus entre maio de 2011 e abril de 2012 foram destinados ao beneficiamento. Essa quantidade de tucumãs beneficiados gerou aos feirantes uma renda bruta de cerca de R$ 900.000,00, demonstrando a sua importância atual no mercado do tucumã (DIDONET, 2012).

Diante do exposto, constata-se a importância da descoberta das melhores variedades para cada modalidade de comércio de tucumã apresentadas neste estudo.

7 CONCLUSÃO

Com os resultados obtidos constata-se que a base de dados formada com 1100 instâncias foi suficiente para gerar modelos satisfatórios por meio dos algoritmos de AM. Dentre as técnicas utilizadas para classificação, redes neurais artificiais apresentou melhores índices de acurácia na predição de classes de variedades de tucumã, tanto na validação quanto no teste de seus modelos. A melhor configuração de taxa de aprendizado encontrada para o algoritmo *MultilayerPerceptron* foi o padrão indicado por Witten & Frank (2005) em 0.3. Com relação à técnica de árvores de decisão, o algoritmo *J48* apresentou índices com pouquíssima diferença em relação às RNAs, tendo como melhor configuração de fator de confiança o nível mais elevado definido em 1.0. Os dois algoritmos de classificação supervisionada testados com essas configurações, apresentaram taxas de 100% de acerto e erro médio absoluto próximo de 0, o que significa que a utilização desses modelos é viável para a classificação de variedades da espécie *Astrocaryum aculeatum*.

Após a tarefa de avaliação e seleção automática de atributos ficou comprovado que entre os atributos preditivos escolhidos, *circunferência-vertical* e *coloração-do-mesocarpo* são os dois melhores na separação de classes de tucumã.

Quanto à tarefa de agrupamento, o algoritmo *SimpleKMeans* formou melhores *clusters* quando utilizada a medida de distância Euclidiana. Também foi descoberto que a modificação dos dados por meio de pré-processamento prejudicou o treinamento dos modelos com as duas medidas de distância, sendo melhor utilizar os dados originais coletados nesta pesquisa. Diante disso, a técnica K-Médias mostrou-se eficiente na geração de *clusters* com as variedades estudadas, mas ainda é preciso buscar novos parâmetros para tentar melhorar a formação dos agrupamentos.

Quanto à indicação de qual é a melhor variedade do tucumã, pode-se afirmar - baseado nos resultados dos experimentos - que o tucumã-arara é o que possui maior potencial econômico em relação à comercialização da polpa *in natura*, pois, matematicamente, a média do peso das polpas de suas unidades em relação à média do peso dos frutos inteiros foi a melhor encontrada entre as variedades adotadas na pesquisa.

Diante do exposto, a utilização de técnicas de aprendizado de máquina é pertinente no que concerne à classificação de variedades de *Astrocaryum aculeatum*. As metodologias de classificação automatizadas criadas com cada algoritmo obtiveram sucesso principalmente na tarefa de classificação supervisionada. Os modelos computacionais gerados apresentaram índices satisfatórios na predição das instâncias após os testes, representando um caminho promissor para o uso de recursos computacionais na classificação taxonômica de variedades de tucumã da espécie estudada.

7.1 Limitações do estudo

Infelizmente, os algoritmos *MultilayerPerceptron* e *SimpleKMeans* não oferecem muitos detalhes sobre o treinamento dos modelos criados, impedindo que sejam analisados cada atributo com relação à sua relevância para o desempenho dos modelos. Ao contrário desses dois algoritmos, o *J48* mostra todas as regras de construção das árvores de decisão, permitindo uma análise mais simplificada do modelo. Neste contexto, não foi possível comparar os modelos das três técnicas quanto á sua estrutura de formação, influência de cada atributo, relação entre os atributos, entre outros fatores relacionados ao treinamento dos modelos.

7.2 Trabalhos futuros

A partir deste trabalho algumas possibilidades de novas pesquisas podem ser desenvolvidas. Outros testes podem ser feitos para validar ainda mais os resultados obtidos com os modelos gerados neste trabalho. Novos modelos podem ser criados aplicando outros meta-classificadores e diferentes parâmetros em cada algoritmo. Pode-se também diminuir o número de atributos para ver como os novos modelos se comportam. Os dois principais atributos são um nominal e um numérico, neste contexto, a exclusão das características de cor pode influenciar nos índices de acertos dos modelos. Diante disso, novas modelagens poderão ser feitas apenas com atributos numéricos para comparar os desempenhos dos modelos nestes dois cenários.

Com relação ao agrupamento das instâncias de tucumã, outros tratamentos devem ser realizados na base de dados para tentar melhorar o desempenho dos modelos gerados com as duas medidas de distância trabalhadas. Outro fator a ser investigado é o custo computacional gerado pelos modelos com as três técnicas nos cenários de dados adotados.

Todos os modelos criados na pesquisa foram testados com 200 instâncias de dados de tucumãs coletados em duas estações de frutificação diferentes. Neste cenário é necessária a coleta de mais unidades em outras estações de anos subsequentes aos do estudo. Com essa metodologia, os modelos poderão ser testados novamente com cada conjunto de dados separados por ano de coleta. A comparação dos modelos com esse novo procedimento pode indicar o nível da influência das possíveis modificações morfológicas das variedades de tucumã sobre os modelos computacionais. Por fim, após esses novos testes é preciso aplicar métodos estatísticos avançados para comparar os dados, a fim de afirmar com mais certeza quais são os melhores modelos

computacionais e as melhores variedades de tucumã de acordo com os critérios adotados na pesquisa.

7.3 Considerações finais

Neste estudo foram apresentados conceitos e definições de alguns termos amplamente utilizados em Aprendizado de Máquina, além de uma descrição sobre três das principais técnicas de AM utilizadas em pesquisas científicas.

A compreensão das diferentes estruturas de cada técnica permite a decisão de como aplicá-las e qual delas utilizar em um determinado contexto. Também é necessário compreender os pontos fortes e as limitações de cada uma delas para poder usá-las com êxito baseando-se no conhecimento do domínio estudado. Além da compreensão dos algoritmos de AM, é igualmente importante poder avaliar o desempenho dos modelos gerados. No contexto da classificação e agrupamento de variedades de tucumã, as técnicas de AM se mostraram robusta e foram eficientes na geração dos modelos preditivos.

REFERÊNCIAS

ABRAMSON, N.; BRAVERMAN, D.; SEBESTYEN, G. “Pattern recognition and machine learning”. Information Theory, IEEE Transactions on, vol. 9, no. 4, p. 257-261, 1963.

BACELAR-LIMA, C. G.; MENDONÇA, M. S.; BARBOSA, T. C. “Morfologia floral de uma população de tucumã, *Astrocaryum aculeatum* G. Mey. (*Arecaceae*) na Amazônia Central”. Acta Amazônica, vol. 36(4), p. 407-412, 2006.

BAJIC, V. B.; CHONG, A.; SEAH, S. H.; BRUSIC, V. “An Intelligent System for Vertebrate Promoter Recognition”. IEEE Intelligent Systems 4, p. 64-70, 2002.

BARBOSA, B. S.; KOOLEN, H. H. F.; BARRETO, J. D. S.; FIGLIUOLO, R.; NUNOMURA, S. M. “Aproveitamento do Óleo das Amêndoas de Tucumã do Amazonas na Produção de Biodiesel”. Acta Amazônica, vol. 39(2), p. 371-376, 2009.

BERTHOLD, M. R.; DIAMOND , J. “Boosting the performance of RBF networks with dynamic decay adjustment”. Advances in Neural Information Processing, vol. 7, p. 512-528, 1995.

BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. Redes Neurais Artificiais Teoria e Aplicações. Livros Técnicos e Científicos Editora, Rio de Janeiro, 2000.

BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. Classification and Regression Trees. Chapman & Hall, 1984.

BRINK, H.; RICHARDS, J. Real World Machine Learning. [S.l.]: Manning Publications C.O, 2014.

CAVALCANTE, P. B. “Frutas Comestíveis da Amazônia”, 7. Ed. rev. atual. Belém: Museu Paraense Emílio Goeldi, 2010.

CARLETTA, J. C. “Assessing agreement on classification tasks: the *Kappa* statistic”. Computational Linguistics, vol. 22(2), p. 249-254, 1996.

CARVALHO, T. J. “Aplicação das técnicas de visão computacional e aprendizado de máquina para detecção de exsudatos duros em imagens de fundo de olho”. Dissertação de Mestrado - Universidade Estadual de Campinas, Instituto de Computação - Campinas, [SP.:s.n.], 2010.

CAVALCANTE, P. B. Frutas comestíveis da Amazônia; coleção Adolfo Ducke, 6ª edição, Belém Pará, p. 219-220, 1996.

CHEN, Y.; HSU, C.; CHOU, S. “Constructing a multi-valued and multi-labeled decision tree”. Expert Systems with Applications, vol. 25(2), p. 199-209, 2003.

CLEMENT, C. R.; LLERAS, P. E.; VAN LEEUWEN, J. “O potencial das palmeiras tropicais no Brasil: acertos e fracassos das últimas décadas”. Revista Brasileira de Agrociência, vol. 9, p. 67-71, 2005.

COHEN, J. “A Coefficient of Agreement for Nominal Scales”. Journal of Educational and Psychological Measurement, p. 37-46, 1960.

CRAVEN, M. W.; SHAVLIK, J. W. “Machine Learning approaches to gene recognition”. IEEE Expert 9, p. 2-10, 1994.

DENG, L.; LI, X. "Machine Learning Paradigms for Speech Recognition: An Overview". Audio, Speech, and Language Processing, IEEE Transactions on, vol. 21, no. 5, p.1060-1089, 2013.

DIDONET, A. A. “O mercado de um produto florestal não madeireiro e o resíduo sólido gerado pela sua comercialização: o caso do tucumã (*Astrocaryum aculeatum* G. Mey.) nas feiras de Manaus”. Dissertação de Mestrado - Instituto Nacional de Pesquisas da Amazônia, Manaus, 2012.

DILWORTH, J. B. Operations management: design, planning, and control for manufacturing and services. Singapura: McGraw-Hill, 1992.

FACELLI, K. “Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento”. Tese de Doutorado - Instituto de Ciências Matemática e de Computação. Universidade de São Paulo, São Carlos, 2006.

FANG, W. “Analysis of the Methods of Machine Learning”. Fu Jian Computer, vol. 11, p. 35-36, 2006.

FERREIRA, E. S.; LUCIEN, V. G.; AMARAL, A. S.; SILVEIRA, C. S. “Caracterização físico-química do fruto e do óleo extraído de tucumã (*Astrocaryum vulgare* Mart)”. Alimentação Nutricional, Araraquara, vol. 19, no. 4, p. 427-433, 2008.

FERREIRA, S. A. N.; GENTIL, D. F. O. “Morfologia da plântula em desenvolvimento de *Astrocaryum aculeatum* Meyer (*Arecaceae*)”. Acta Amazônica, vol. 35, no. 3, p.337-342, 2005.

FISHER, R. A. “The Use of Multiple Measurements in Taxonomic Problems”. In Annals of Eugenics 7, p. 179-188, 1936.

FLEISS, J. L. Statistical methods for rates and proportions. New York: John Wiley, p. 212-236, 1981.

FREITAS, A. A.; LAVINGTON, S. H. Mining Very Large Databases with parallel Processing, Kluwer, 1998.

FUNG, G. “A comprehensive Overview of a Basic *Cluster*ing Algorithms”, 2001. Disponível em <http://www.cs.wisc.edu/~gfung/*cluster*ing.pdf>. Acessado em fevereiro de 2014.

FURNKRANZ, J.; GAMBERGER, D.; LAVRAC, N. Foundations of Rule Learning. [S.l.]: Springer-Verlag Berlin, 2012.

GIL, V. O.; FERRARI, F.; EMMENDORFER, L. “Investigação da aplicação de algoritmos de agrupamento para o problema astrofísico de classificação de galáxias”. Revista Brasileira de Computação Aplicada, Passo Fundo, vol. 7, no. 2, p. 52-61, 2015.

HAN, J.; KAMBER, M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.

HART, P. E.; DUDA, R. O.; STORK, D. G. Pattern Classification. Wiley-Interscience; 2 edition, 2000.

HARTIGAN, J. A. *Cluster*ing algorithms. Wiley New York, 1975.

HAYKIN, S. Neural Networks: A Comprehensive Foundation. Prentice Hall, 1999.

HOLMES, G.; DONKIN, A.; WITTEN, H. "WEKA: a machine learning workbench". Intelligent Information Systems. Proceedings of the 1994 Second Australian and New Zealand Conference on, vol., no., p. 357-361, 1994.

HUA, W.; CUIQIN, M.; LIJUAN, Z. "A Brief Review of Machine Learning and Its Application". Information Engineering and Computer Science. ICIECS. International Conference on, vol., no., p. 1-4, 2009.

IBGE. Informações da Ferramenta IBGE Cidades@ e da Diretoria de Pesquisas, Coordenação de População e Indicadores Sociais, 2014.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. "Data *cluster*ing: a review". ACM Computing Surveys (CSUR), p. 264–323, 1999.

JAIN, A.; DUBES, R. Algorithms for *Cluster*ing Data. Prentice-Hall, Englewood Cliffs, NJ, 1988.

JOHN, G. H.; KOHAVI, R.; PFLEGER, K. "Irrelevant Features and the Subset Selection Problem". 11th International Conference in Machine Learning, p. 121-129, 1994.

JOO, D.; HONG, T.; HAN I. "The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors". Expert Systems with Applications, vol. 25, p. 69–75, 2003.

KAHN, F. “The genus *Astrocaryum* (*Arecaceae*)”. Revista Peruana de Biologia, vol. 15(1), p. 31-48, 2008.

KAHN, F.; MOUSSA, F. “Economic importance of *Astrocaryum aculeatum* (*Palmae*) in Central Brazilian Amazonia”. Acta Botânica Venezuela, vol. 22(1), p. 237–245, 1999.

KAINULAINEN, J. “*Cluster*ing Algorithms: Basics and Visualization”, 2002. Disponível em <http://www.cs.baylon.edy/~hamerly/papers/thesis.pdf>. Acessado em fevereiro de 2014.

KOHAVI, R. “A study of cross-validation and bootstrap for accuracy estimation and model selections”. In: International Joint Conferences on Artificial Intelligence (IJCAI), 14th, Montréal, Québec. Proceedings. Morgan Kaufmann, 1995.

KOHAVI, R.; JOHN, G. H. “The Wrapper Approach”. In: H. Liu & H. Motoda (Eds.) Feature Extraction, Construction and Selection: a data mining perspective, p. 33-49. Kluwer, 1998.

KUNCHEVA, L. I. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, 2004.

KURGAN, L. A.; CIOS, K. J.; TADEUSIEWICZ, R.; OGIELA M., GOODENDAY, L. “Knowledge discovery approach to automated cardiac SPECT diagnosis”, Artificial Intelligence in Medicine, vol. 23, no. 2, p. 149–169, 2001.

LAPEDES, A.; BARNES, C.; BURKS, C.; FARBER, R.; SIROTKIN, K. “Application of Neural Networks and other Machine Learning algorithms to DNA sequence analysis”. In: Bell G and Marr T (eds) Computers and DNA, SFI in the sciences of complexity, vol. 7, p. 157-182, 1989.

LLETÍ, R. et al. “Selecting variables for k-means *cluster* analysis by using a genetic algorithm that optimizes the silhouettes”. Analytica Chimica Acta, vol. 515, p. 87–100, 2004.

LOTZ, J. M.; PRIMACK, J.; MADAU, P. “A new nonparametric aproach to galaxy morphological classifica-tion”. The Astronomical Journal, vol. 128, p. 163–182, 2004.

MADEIRA, W. V. “Plano Amazônia Sustentável e Desenvolvimento Desigual”. Ambiente & Sociedade, São Paulo, vol. 17, no. 3, p. 19-34, 2014.

MANGASARIAN, O. L.; WOLBERG, W. H. “Cancer diagnosis via linear programming”. SIAM News, vol. 23(5), p. 1-18, 1990.

MATOS, R. A. “Comparação de Metodologias de Análise de Agrupamentos na Presença de Variáveis Categóricas e Contínuas”. Dissertação de Mestrado - Universidade Federal Minas Gerais, Instituto de Ciências Exatas - Belo Horizonte, 2007.

MATSUBARA, E. T. “Relações entre ranking, análise ROC e calibração em aprendizado de máquina”. Tese de Doutorado - Universidade de São Paulo, São Carlos, 2008.

MCCULLOCH, W. S.; PITTS, W. “A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biophysics, Elmsford, vol. 5, p. 115-133, 1943.

MENDONÇA, M. S. “Aspectos morfológicos das sementes de algumas espécies de palmeiras (*Arecaceae=Palmae*) da Amazônia”. Tese de Doutorado - Universidade do Estado do Amazonas, 1996.

MINSKY, S.; PAPERT, M. Perceptrons: An introduction to computacional geometry. Massachusetts: MIT Press, 1969.

MIRANDA, I. P. A. Frutos de palmeiras da Amazônia. Manaus: MCT/INPA, 2001.

MITCHELL, T. M. Machine Learning. McGraw-Hill, New York, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre Aprendizado de Máquina. in Sistemas Inteligentes: Fundamentos e Aplicações. Ed. Manole Ltda, Baueri, Cap. 4, p. 89 - 114, 2005.

MUKKAMALA, S.; JANOSKI, G.; SUNG, A. “Intrusion detection using neural networks and support vector machines”. IEEE International Joint Conference on Neural Networks, p. 1702-1707, 2002.

MUNIZ, M. H. “Uma Abordagem para o Problema de Classificação utilizando Programação Inteira”. Dissertação de Mestrado - Universidade Federal Minas Gerais - Belo Horizonte, 2007.

OLIVEIRA, M. S. P. “Caracterização morfológica de frutos em acessos de tucumãzeiro (*Astrocaruym vulgare* Mart.)”. In: Simpósio de Recursos Genéticos para a América Latina e Caribe, Proceedings, p. 351-353, 2001.

PEDERSEN, A. G.; NIELSEN, H. “Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis”. In: Proc. Int. Conf. Intell. Syst. Mol. Biol. (ISMB'97), p. 226–233, 1997.

PELLUCCI, P. R. S.; DE PAULA, R. R.; OLIVEIRA, W. B. S.; LADEIRA, A. P. “Utilização de Técnicas de Aprendizado de Máquina no Reconhecimento de Entidades Nomeadas no Português”. E-xacta, Belo Horizonte, vol. 4, no. 1, p. 73-81, 2011.

PEÑA, J. M.; BJÖRKEGREN J.; TEGNÉR, J. “Learning dynamic Bayesian network models via cross-validation”. Pattern Recognition Letters, vol. 26, p. 2295-2308, 2005.

QUINLAN, J. “Simplifying decision trees”. International Journal of Man-Machine Studies, no. 27, p. 221-234, 1987.

QUINLAN, J. C. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, 1993.

RODRIGUES, P. H. V.; FERREIRA, F. F.; AMBROSANO, G. M. B.; GATO, A. M. G. “Propagação in vitro de tucumã do Amazonas”. Ciência Rural, Santa Maria, vol. 43, no. 1, p. 55-59, 2013.

SARMENTO, E. C. “Comparação entre quatro algoritmos de aprendizagem de máquina no mapeamento digital de solos no Vale dos Vinhedos, RS, Brasil”. Dissertação de

Mestrado - Universidade Federal do Rio Grande do Sul, Faculdade de Agronomia - Porto Alegre, 2010.

SCHROTH, G.; DA MOTA, M. S. S.; LOPES, R.; DE FREITAS, A. F. “Extractive use, management and in situ domestication of a weedy palm, *Astrocaryum aculeatum*, in the central Amazon”. Forest Ecology Management, vol. 202, p. 161–179, 2004.

SIEGEL, S.; CASTELLAN N. Nonparametric Statistics for the Behavioral Sciences. 2.ed. New York: McGraw-Hill, p. 284-285, 1988.

SILVA, M. S. “Uma Abordagem Evolucionária Para o Aprendizado Semi-Supervisionado em Máquinas de Vetores de Suporte”. Dissertação de Mestrado - Universidade Federal Minas Gerais - Belo Horizonte, 2008.

SIVIERO, M. R. L.; HRUSCHKA JÚNIOR, E. R. “Algoritmos de Aprendizado de Máquina Aplicados à Parâmetros Mensurados no Rio Atibaia/SP”. XIX Simpósio Brasileiro de Recursos Hídricos, 2011.

STROEH, K. “Uma abordagem para a correlação de eventos de segurança baseada em técnicas de aprendizado de máquina”. Dissertação de Mestrado - Universidade Estadual de Campinas, Instituto de Computação - Campinas, [SP.:s.n.], 2009.

TAN, P. N.; STEINBACH, M.; KUMAR, V. Introduction to Data Mining. Pearson Education, Inc., Boston, 2006.

TOWELL, G. G.; SHAVLIK, J. W.; NOORDEWIER, M. O. “Refinement of approximate domain theories by knowledge-based neural networks”. In Proceedings of the Eighth National Conference on Artificial Intelligence, p. 861-866, 1990.

WEISS, S. M.; INDURKHYA, N. “Rule-based machine learning methods for functional prediction”. J. Artif. Int. Res., AI Access Foundation, USA, vol. 3, no. 1, p. 383–403, 1995.

WITTEN, I. H.; FRANK, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, 2nd edition, San Francisco, California, 2005.

XUE, M.; ZHU, C. "A Study and Application on Machine Learning of Artificial Intellligence". Artificial Intelligence, JCAI '09. International Joint Conference on , vol., no., p. 272-274, 2009.

YUYAMA, L. K. O.; MAEDA, R. N.; PANTOJA L.; AGUIAR J. P. L.; MARINHO H. A. "Processamento e avaliação da vida-de-prateleira do tucumã (*Astrocaryum aculeatum* Meyer) desidratado e pulverizado". Ciência e Tecnologia de Alimentos, vol. 28, no. 2, p. 408-412, 2008.

ZHOU, B.; ZHANG, X.; WANG, R. "Automated soil resources mappaing based on decision tree and Bayesian predictive modeling". Journal of Zhejiang University Science, vol. 5, no. 7, p. 782-795, 2004.