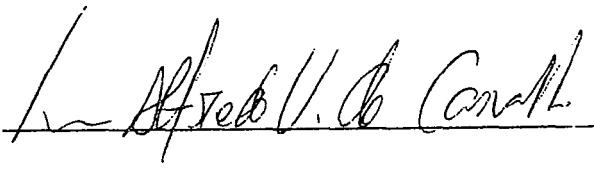
PRÉ-PROCESSAMENTO EM MINERAÇÃO DE DADOS: UM ESTUDO COMPARATIVO EM COMPLEMENTAÇÃO

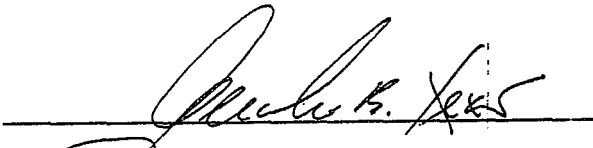
Jorge de Abreu Soares

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

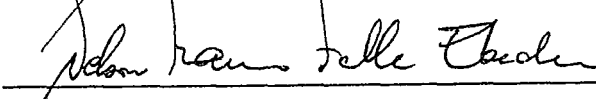
Aprovada por:



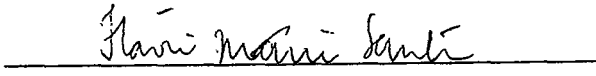
Prof. Luis Alfredo Vidal de Carvalho, D. Sc.



Prof. Geraldo Bonorino Xexéo, D. Sc.



Prof. Nelson Francisco Favilla Ebecken, D. Sc.



Profª. Flavia Maria Santoro, D. Sc.

Prof. Basílio de Bragança Pereira, Ph. D.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 2007

SOARES, JORGE DE ABREU

Pré-Processamento em Mineração de Dados: Um Estudo Comparativo em Complementação [Rio de Janeiro] 2007

XII, 232 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 2007)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Pré-Processamento de Dados 2. Imputação Composta 3. Comitês de Complementação de Dados

I. COPPE/UFRJ II. Título (série)

Àquelas por quem sempre dedicarei minha vida: Bárbara e Carolina.

AGRADECIMENTOS

IS 41:13 Porque eu, o Senhor teu Deus, te seguro pela tua mão direita, e te digo: Não temas; eu te ajudarei.

Por vezes imagino o quão difícil seria viver sem a presença de Deus... impossível imaginar, pois sem Ele não haveria vida! E, nesta tese de Doutorado, Ele mais uma vez manifestou-me Sua presença, fazendo despertar em mim um poder de superação que não imaginava que tivesse... Por esta razão, e por todas as demais razões existentes, agradeço (por toda a eternidade) a Deus e a Nosso Senhor Jesus Cristo por mais esta vitória (e todas as demais)!

No plano terreno, pude mais uma vez constatar, durante estes anos, que a autoria de qualquer trabalho – seja ele até uma tese de Doutorado! – nunca é, na verdade, apenas uma produção de seus autores. Levando-se em conta que a vida é um projeto, o seu sucesso só é alcançado com o trabalho – direto e indireto – de toda a equipe, onde cada um desempenha um papel importante para que o objetivo maior seja alcançado. E por esta razão registro aqui meu eterno agradecimento a esta "equipe" de grandes amigos, que me permitiram galgar este grande objetivo, materializado neste documento.

Apesar de parecer chavão, não posso deixar de concordar e registrar que a família é a base de tudo! É por ela que nos sacrificamos... e vem dela a força e o apoio indispensáveis para o sucesso em qualquer empreitada que venhamos a participar! Isso é especialmente verdadeiro quando em nossa vida aparecem os filhos (no meu caso filhas!). É principalemente porque é por eles que fazemos tudo! É basicamente por eles que suportamos qualquer dor! É simplesmente por eles que vivemos! E no meu caso não é diferente... Os melhores "resultados" obtidos em todos estes anos de Doutorado são assim denominados: Bárbara e Carolina. Estas duas "publicações em periódico internacional qualis $A^{+\infty}$" têm como co-autora a mais bela e maravilhosa pessoa que já conheci: minha esposa, Rejane. A você, amor de minha vida, agradeço primeiramente por existir! E aproveito a ocasião para te agradecer toda a inifita paciência que você teve, em intermináveis fins de semana "aprisionada" em casa com nossas filhas, nas inúmeras vezes em que você teve que assumir de forma solitária todos os afazeres

domésticos e não domésticos, e por todo o resto. Sem o seu apoio, este trabalho simplesmente não teria terminado. Aliás, sem você, nada teria sentido.

Ainda no seio familiar, agradeço à minha amada mãe, Dionéa, pela dedicação total desde os áureos anos de 1973. Seus ensinamentos, sua força, sua ajuda e incentivo são fudamentais para que eu continue caminhando e insistindo. À senhora, mãe, mais uma vez meu muito obrigado!

Da mesma forma fundamentais são as pessoas que a vida nos dá de presente meio que sem querer, e que, a partir do momento em que elas entram em nossas vidas, passam da mesma forma ao status de “imprescindível”. Desta forma, quero agradecer muito à minha sogra, Fátima, e à avó de minha esposa, Laura (agora também minha avó!) pelo incentivo, ajuda e amizade ao longo destes 17 anos que nos conhecemos, e que passamos a formar uma nova grande família. Estejam certas de que o apoio incondicional de vocês foi e é muito importante.

Existem ainda algumas pessoas que passam a pertencer à sua vida por opção. São os amigos que a vida nos dá de presente. E a eles dedico, um a um, um agradecimento especial. Espero não ter sido injusto esquecendo de algum deles. Mas se isto acontecer, peço de antemão as mais sinceras desculpas! Apesar de dedicar um dia inteiro a escrever apenas estes agradecimentos, posso cometer a imperdoável falha de omitir pessoas importantes.

Não poderia começar sem deixar de agradecer ao grande responsável pela existência desta tese: o Professor Luis Alfredo Vidal de Carvalho. Inicialmente orientador, revelou-se um amigo tão especial que é por vezes difícil acreditar que existam pessoas tão humanas e amigas. Detentor de um potencial acadêmico admirável, o Professor Luis Alfredo, nestes anos onde trabalhamos juntos, ofertou-me a oportunidade de reavaliar o real papel do ser humano neste planeta: o de ajudar o próximo, o de dar oportunidades iguais a todos, e, paradoxalmente, o de transmitir conhecimento da forma mais completa e simples possível o conhecimento. Ao grande amigo, Luis Alfredo, meu eterno agradecimento! E que a defesa desta tese seja o inicio de muitos trabalhos que venhamos a desenvolver juntos, pois a amizade já está estabelecida.

Da mesma forma agradeço à banca avaliadora deste trabalho, os professores Basílio Bragança, Flavia Santoro, Gerado Xexéo e Nelson Ebecken a enorme paciência

que tiveram comigo nesta etapa final. E agradeço também aos pertinentes comentários feitos com relação ao trabalho, que certamente me tranqüilizam e me dão a certeza absoluta de que todo o esforço empregado valeu a pena.

Posso me considerar um privilegiado, pois na segunda metade do período em que estive no doutorado, ganhei um co-orientador não oficial (que no final passou a ser oficial!) e um grande amigo. Alguém incansável, paciente, experiente, competente, dedicado, e extremamente organizado, que esteve sempre que solicitado à disposição, discutindo aspectos técnicos ou não. Pela indiscutível ajuda e amizade, agradeço a Ronaldo Goldschmidt por acreditar e investir neste trabalho.

Existem pessoas que se tornam amigas. E existem amigos tão amigos que queremos trazê-los para a nossa família. Por esta razão, faço questão de aqui registrar um agradecimento muitíssimo especial à minha hoje comadre e sempre amiga Claudia Ferlin, por todos esses anos de incondicional amizade. Este agradecimento não está limitado apenas ao escopo desta tese, mas a tudo o que você me ensina, ajuda, divide, colabora, trabalha... enfim, por tudo! Muito obrigado, você e Yale são amigos muito especiais!

Os amigos são realmente pessoas muito importantes em nossas vidas. E a amizade ratifica-se justamente nos momentos onde um mais precisa do outro. E, para a minha sorte, pude contar com um amigo de longa data. Por isto, muito agradeço a Eduardo Bezerra a amizade, o apoio incondicional, as revisões no texto, os compromissos assumidos em meu lugar... pelo conjunto da obra, muito obrigado!

A vida nos reserva gratas surpresas. E isto é especialmente verdade no magistério. Ontem um aluno destacado no curso de Ciência da Computação, hoje um excelente profissional, professor do curso que o formou com atuação marcante, aluno de Mestrado com um imenso potencial para uma bela carreira científica, agradeço ao amigo Rafael Castaneda pela importantíssima ajuda durante os meses finais. Sua participação neste trabalho foi simplesmente fundamental, e sem ela as idéias expostas neste trabalho não seriam materializadas em tempo recorde. Por esta razão agradeço imensamente sua colaboração e amizade.

À minha querida amiga Isabel Fernandes, apenas uma menção: estou certo de que Deus lhe retornará toda a força e boas energias que você me mandou durante todo este

tempo. E que Ele lhe dê como recompensa, em breve, também uma longa seção de agradecimentos a escrever! Muito obrigado por tudo.

Ao meu grande amigo e companheiro de jornada Roberto Ferreira dos Santos, registro aqui meu agradecimento pelo incentivo e amizade nos anos que tivemos a oportunidade de trabalhar lado a lado. Você também é um exemplo de pessoa, de pesquisador e de profissional no qual procuro me espelhar.

Tenho um obrigado muito especial a registrar às minhas amigas Flavia Santoro, Renata Araújo e Rosa Costa. Muito obrigado por não me deixarem esmoerecer! Vocês são, além de excelentes profissionais, professoras e pesquisadoras, pessoas muito especiais, e o incentivo que vocês me deram jamais será esquecido!

Outro especial agradecimento é dedicado à Professora Maria Luiza Campos. Sua conduta profissional e pessoal serviu-me de inspiração para o investimento na carreira acadêmica. Além disso, a qualidade de seus ensinamentos também me influenciou fortemente a seguir pela linha na qual atuo hoje acadêmica e profissionalmente. Assim, registro aqui meu muito obrigado.

Há quinze anos, ainda como calouro do curso de Bacharelado em Informática da UFRJ, conheci algumas pessoas que até hoje felizmente ainda fazem parte de minha vida, e que estão sempre presentes nos momentos bons (e nos que não são tão bons). Como este é um momento muito bom, agradeço a amizade e incentivo aos meus queridos amigos André Braga, Ednilson Carlos, Flavio Tavares, Gladstone Moisés, Gustavo Hajdu e sua esposa Patrícia Hajdu (compadre e comadre!), Marcelo Blois, Ricardo Choren e Rogério Pinheiro. Obrigado pela amizade de hoje e sempre!

Aos amigos do CEFET/RJ, que me acompanham nestes últimos três anos, sempre com uma palavra amiga de incentivo e conforto, meu muito obrigado: Elizabeth Freitas, João Quadros e Laércio Brito.

Aos amigos de UniverCidade que sempre me incentivaram a completar esta jornada, mesmo nos momentos de esmorecimento, registro meus sinceros agradecimentos: Ana Lagoa, André Sobral, Antônio Júnior, Carlos Augusto Ribeiro, Carmen Queiroz, Claudio Ribeiro, Fernando Padovani, Fernando Pina, Ismael Santos, Jorge Cássio Mello, Josir Gomes, Julio Nichioka, Leandro Chernicharo, Kelly Christine, Luiz Vivacqua, Marcelo Pereira, Marcio Bispo, Marco Tulio Laucas, Orlando

Bernardo, Oswaldo Peres, Ricardo Quintão, Ricardo Valença, Rodrigo Martins, Saulo Barbará e Silvia Nogueira.

Agradeço também ao Programa de Engenharia de Sistemas e Computação da COPPE, em especial ao seu corpo administrativo: Solange, Claudinha, Lúcia e Sônia. Agradeço também a CAPES pelo fomento ao este trabalho de pesquisa de tese.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D. Sc.)

PRÉ-PROCESSAMENTO EM MINERAÇÃO DE DADOS: UM ESTUDO COMPARATIVO EM COMPLEMENTAÇÃO

Jorge de Abreu Soares

Maio/2007

Orientador: Luis Alfredo Vidal de Carvalho

Programa: Engenharia de Sistemas e Computação

As aplicações atuais e a evolução tecnológica vêm promovendo a produção e o armazenamento de um grande volume de dados. Este cenário faz com que a existência de valores ausentes em registros das bases de dados inevitavelmente aumente. Estas lacunas prejudicam a análise dos dados, além de dificultar ou mesmo inviabilizar o processo de abstração de conhecimento a partir deles.

Desta forma, este trabalho tem por objetivo avaliar quais os efeitos da aplicação das tarefas de seleção de atributos e agrupamento de dados precedendo à complementação de dados ausentes em bases de dados. Também propusemos nesta tese a aplicação de comitês de complementação de dados para o processo de imputação. Esta abordagem busca modificar a clássica técnica de imputação múltipla, incorporando o conceito de meta-aprendizado normalmente encontrado em comitês de classificação.

Os resultados experimentais mostram significativa melhora da qualidade dos dados sugeridos quando são gerados pelas estratégias de imputação composta de pré-processamento, indicando que a imputação obtém melhores resultados quando restringe este processo aos registros mais relevantes do conjunto de dados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D. Sc.)

PREPROCESSING DATA IN DATA MINING: A COMPARATIVE STUDY IN INPUTATION

Jorge de Abreu Soares

May/2007

Advisor: Luis Alfredo Vidal de Carvalho

Department: Computer and Systems Engineering

Nowadays applications and technological evolution have caused the production and storage of huge volumes of data. This scenario facilitated the increased occurrence of missing values in data sets. Missing data is harmful for statistical analysis, complicating or even not allowing the process of extracting knowledge from these non preprocessed data.

Hence, this work aims at analyzing the effects of the application of selection and clustering tasks before the imputation of missing values in data sets. We have also proposed the application of imputation committees to the imputation process. This approach attempts to modify the multiple imputation technique integrating the meta learning concept, normally encountered in classification committees.

Experimental results show that we achieve relevant quality improvement of imputed data when generated by these composed preprocessing strategies, pointing out that the whole process gains when it works with the most relevant part of the dataset.

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

1.1 Considerações Iniciais

As aplicações do século XXI processam dados em uma quantidade da ordem de teraabytes e petabytes, já que a evolução tecnológica dos computadores permitiu que estes equipamentos apresentassem tempos de resposta em intervalos de tempo cada vez menores. Além disso, os dispositivos de armazenamento comercializados atualmente são oferecidos a preços cada vez mais atraentes, com um desempenho superior aos modelos anteriores, e aumentando significativamente a sua capacidade de armazenamento. É possível encontrarmos atualmente nas empresas especializadas em venda de hardware para aplicações domésticas unidades de armazenamento secundário na ordem de gigabytes. Empresas que demandam alta capacidade, confiabilidade e segurança de armazenamento encontram no mercado especializado soluções de armazenamento na ordem de terabytes, só imagináveis há alguns anos atrás com um grande investimento financeiro.

Todavia, corporações que não só geram grandes volumes de dados, mas também têm a necessidade de mantê-los armazenados, podem estar perdendo um grande diferencial competitivo se não atentarem para o fato de que estes dados, se devidamente interpretados, podem ser de extrema valia para as suas decisões estratégicas, já que eles podem esconder (e provavelmente escondem) padrões interessantes, que provavelmente refletem comportamentos dos consumidores, tendências de negócios, e outras informações extremamente importantes para o processo de tomada de decisões da empresa.

Motivado por este contexto, o processo de **Descoberta de Conhecimento em Bases de Dados** (*KDD – Knowledge Discovery in Databases*) (FAYYAD *et al*, 1996a) tornou-se uma área de pesquisa amplamente estudada pela comunidade científica de Computação. Também conhecido por **Mineração de Dados** (*Data Mining*), ela busca descobrir que tipos de relações intrínsecas podem existir em um conjunto de dados.

O princípio que norteia o processo de descoberta de conhecimento em bases de dados é a relação hierárquica que podemos observar entre o conceito de **dado**, **informação** e **conhecimento** (GOLDSCHMIDT, PASSOS, 2005). Várias definições podem ser encontradas sobre o que é um dado. A que melhor parece se adequar é aquela que diz que um dado é um *elemento conhecido de um problema*. Isto é especialmente verdadeiro, já que problemas não existem por si só, e tem consigo elementos de entrada e saída associados. Contudo, se estes dados apresentarem uma relação semântica, eles passam a ser considerados **informação**. Exemplificando, se um determinado conjunto se apresenta na forma {10, 40%, 4}, dizemos que este é um conjunto de dados. Todavia, a estrutura "40% de 10 é igual a 4" revela mais do que simplesmente dados: ela nos mostra uma relação explícita entre eles, com sentido.

Nesta mesma linha, podemos evoluir o conceito de informação para um outro mais amplo, que pode ser inferido de uma amostra maior dos dados. Quando, a partir deles, podemos descobrir relações mais amplas, que envolvem não só todo o conjunto de dados, mas também os elementos que os descrevem (os *metadados*), dizemos que obtemos não só uma informação, ma sim um **conhecimento** embutido nestes dados. Assim, se a partir de um conjunto de dados, podemos descobrir que as vendas de alguns produtos estão intimamente relacionadas em determinados dias da semana, podemos dizer que esta relação implícita aos dados foi minerada, descoberta, e que pode inclusive dar dicas estratégicas quanto à sua organização (os produtos em questão podem, às sextas-feiras, ser expostos lado a lado, facilitando a sua aquisição. A Figura 1.1 ilustra o conceito exposto.

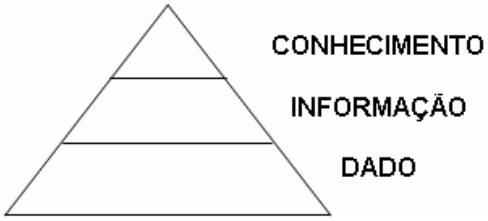


Figura 1.1 Relação hierárquica entre dado, informação e conhecimento.
Fonte: Adaptado de (*GOLDSCHMIDT, PASSOS, 2005*)

1.2 Motivação

O grande volume de dados gerados pelas aplicações, tratado na seção 1.1, gera inevitavelmente a ocorrência de alguns problemas de consistência dos dados. Estes

problemas podem advir tanto do processo de integração de diferentes fontes de dados, onde atributos semelhantes possuem nomes, tipos ou domínios diferentes, quanto do processo da carga de dados, onde alguns dos valores preenchidos podem representar valores fora da faixa esperada, ou mesmo não terem sido preenchidos. Outros motivos que podem fazer com que dados estejam ausentes são o óbito de pacientes, mal funcionamento de equipamentos de aferição de medidas, recusa de entrevistados de responder a certas questões, entre outros (BATISTA, MONARD, 2003a). Este tipo de problema pode comprometer toda a etapa de Mineração de Dados, e por isso requer especial atenção dos analistas de dados que desejam obter conhecimento a partir de uma relação.

Abordagens convencionais, tais como a remoção de registros ou de colunas com algum valor ausente em um de seus atributos, estão presentes na maioria dos pacotes estatísticos. Todavia, elas tornam a análise tendenciosa, já que esta remoção faz com que o conjunto de dados não represente de forma significativa a amostra original, já que os valores que estavam presentes e foram apagados por conta da redução horizontal ou vertical causam perda significativa de informação.

Assim, a imputação vem sendo amplamente pesquisada ao longo dos anos. Diversas soluções com origens tanto com métodos estatísticos quanto na Inteligência Computacional, com a utilização de técnicas de aprendizado de máquina apresentam-se como possíveis soluções ao desafio de complementar dados ausentes em conjuntos de dados (vide revisão bibliográfica apresentada nas seções 3.5 e 3.6). Todavia, apesar de bastante estudado, não encontramos na literatura nenhum trabalho que, a partir de um conjunto de valores de imputação sugeridos, gerasse um novo valor, este influenciado pelos anteriores.

Desta forma, avaliamos especificamente nesta tese o problema de dados ausentes em tabelas de bases de dados sob a ótica de diferentes estratégias de complementação de dados.

Dedicar uma especial atenção ao tratamento destes dados é de fundamental importância a todo o processo de Descoberta de Conhecimento em Bases de Dados. Diversas soluções são apresentadas na literatura, e serão tratadas no capítulo 3. Interessa-nos particularmente as soluções que envolvam a aplicação de algoritmos de

aprendizado de máquina na solução do problema de complementação de dados ausentes, e este é então nosso objeto de estudo nesta tese.

1.3 Objetivo da tese

O objetivo da tese é o de analisar os efeitos da aplicação de diversas estratégias de complementação de dados em bases de dados que possuem valores ausentes em suas tuplas. A complementação de dados, mais conhecida como imputação, é um problema de eminente importância na etapa de pré-processamento de dados, pois toda a etapa de mineração de dados pode ser severamente afetada caso os dados não tenham recebido um tratamento cuidadoso, no que diz respeito à complementação dos dados ausentes.

Assim, avaliamos o efeito da aplicação de técnicas de Inteligência Computacional na combinação das tarefas de seleção e agrupamento precedendo o processo de imputação de dados. Chamamos estas combinações de *estratégias*, e a sua instanciação de *planos de imputação*. A utilização de algoritmos de agrupamento precedendo a imputação de valores ausentes é uma técnica conhecida por imputação *hot-deck* (FORD, 1983, FULLER, KIM, 2001), e que busca, com a divisão do conjunto original em grupos, fazer com que o processo de imputação seja influenciado apenas pelos objetos do conjunto de dados que possuam alguma relação com aqueles onde algum valor é ausente. Esta relação, obtida de forma implícita, tenta fazer com que o processo se torne o menos tendencioso possível (FORD, 1983).

Todavia, queremos analisar também o impacto da aplicação da tarefa de seleção de atributos no processo de complementação de dados ausentes. Com isso, avaliamos se a redução do volume de dados a ser enviado para a tarefa de imputação pode melhorar a qualidade do processo como um todo. Analisamos também o impacto da aplicação da seleção antes ou depois do processo de complementação de dados ausentes. Assim, chegamos a uma configuração das seguintes estratégias para avaliação: 1) imputação; 2) seleção e imputação; 3) agrupamento e imputação; 4) seleção, agrupamento e imputação; e 5) agrupamento, seleção e imputação. Estas estratégias, quando materializadas com a aplicação de algoritmos que as implementam, são chamadas *planos de execução*. Com isso, cada estratégia materializa-se em um ou mais planos de execução.

Utilizamos para a seleção de dados a técnica de Análise de Componentes Principais (PCA – *Principal Component Analysis*) (SMITH, 2002, SHLENS, 2005); para o agrupamento, valemo-nos do clássico algoritmo de agrupamento dos *K*-centróides (*k-means*) (MCQUEEN, 1967); e para a imputação, utilizamos três diferentes técnicas: o algoritmo dos *k*-vizinhos mais próximos (*k*-NN) (MITCHELL, 1997, AHA, KIBLER, ALBERT, 1991, DASARATHY, 1990), as redes neuronais de aprendizado supervisionado *back propagation* (RUMELHART *et al*, 1986), além da média aritmética simples em duas versões: clássica e com a introdução de uma perturbação (valor aleatório que, também randomicamente, é adicionado ou subtraído ao valor da média). Para isso, construímos um sistema em Java chamado *Appraisal*, que gera, segundo uma ou mais estratégias de complementação de dados, sugestões de valores para tuplas com atributos ausentes.

Também propomos nesta tese a aplicação de comitês de complementação de dados para o processo de imputação de dados. Esta abordagem busca modificar a clássica técnica de imputação múltipla, sugerida por RUBIN *et al* (1988), incorporando o conceito de meta-aprendizado normalmente encontrado em comitês de classificação. A diferença de nossa proposta de comitê diferencia-se dos de classificação, pois estes elegem, segundo alguns critérios, um valor como sendo a melhor opção para a classificação de um dado objeto. Já que nosso objetivo é o de imputar valores desconhecidos em tuplas de tabelas, queremos analisar se é mais proveitoso gerarmos um novo valor para o atributo ausente, ao invés de eleger um valor sugerido. Desta forma, os $k \leq n$ resultados das estratégias sobre os dados servem de entrada para o comitê de complementação de dados, que produz um $(k+1)$-ésimo valor.

Para avaliar os resultados, utilizamos dois tipos de medidas. A primeira verifica o quão distante o resultado gerado é diferente do real, através da medição do erro relativo absoluto (RAD – *Relative Absolute Deviation*). Discutimos na tese que esta medida, apesar de ser a que melhor se adequa às nossas necessidades, por vezes não pode ser aplicada, e alternativas são consideradas. A outra medida de qualidade do resultado empregada é a reclassificação da tupla imputada. Nossas bases de teste contêm tuplas que originalmente possuem classificações. Todavia, todo o processo de imputação acontece sem levar este atributo-classe em conta. O quão freqüente o valor imputado levar a tupla a uma classe que não é a sua original nos dirá se o processo rendeu bons ou

maus resultados. A classificação é realizada com redes neuronais artificiais *back propagation*, utilizando dois tipos de saídas: a binária (onde cada neurônio da camada de saída representa um valor da classe), e a binária econômica (onde as classes têm associadas valores em binário, que as representam).

Resultados foram gerados em três bases do repositório de aprendizado de máquina da Universidade da Califórnia, Irvine (NEWMAN *et al*, 1998). Este repositório possui diversas bases de dados com as mais diferentes características, e serve como *benchmark* para diversos trabalhos na área de descoberta de conhecimento de bases de dados. Utilizamos então as bases *Iris Plants*, que possui 150 tuplas, quatro atributos numéricos e um classificador (que indica uma de três possíveis bases para classificação de plantas); a base *Pima Indians*, que possui 392 tuplas, oito atributos numéricos e uma classe. Esta base registra dados médicos de pacientes com diabetes, categorizando-os como diabéticos ou não. Por fim, também usamos a base *Wisconsin Breast Cancer*, com 682 tuplas, nove atributos numéricos e uma classe. Trata-se de uma base que registra características clínicas de pacientes com câncer de mama ou não. Idem para a descrição das bases.

Todos os atributos utilizados, à exceção da classe, são numéricos. Além disso, as bases de dados utilizadas em nossos experimentos estão originalmente completas. Os valores ausentes foram produzidos de forma artificial e controlada, através de um dos módulos do protótipo que implementamos para os experimentos desta tese. O mecanismo de ausência adotado foi o completamente aleatório (MCAR – *Missing Completely At Random*), com diferentes percentuais de tuplas com valores ausentes no atributo alvo dos testes. Estes índices variaram de 10% a 50%, com saltos de 10%. Os resultados gerados foram comparados, e as estratégias avaliadas em função da base, atributo, percentual de valores ausentes, valores sugeridos e erro nas reclassificações.

1.4 Organização do texto

O restante do documento está organizado da seguinte forma: o capítulo 2 aborda a fundamentação teórica deste trabalho, detalhando as etapas da etapa de Pré-Processamento de Dados. O capítulo 3 foca o problema alvo de estudo da tese, a Complementação de Dados Ausentes em Bases de Dados. Assim, explanamos os conceitos relacionados, e revisamos as principais soluções disponíveis na literatura que

propõem soluções para este problema. O capítulo 4 detalha a proposta desta tese: a imputação composta. Esta categoria de algoritmos de imputação estuda a aplicação de estratégias que combinem tarefas do processo de Descoberta de Conhecimento em Bases de Dados na complementação de dados ausentes numéricos. Diferentes combinações que envolvam as tarefas de agrupamento, seleção de variáveis e imputação são abordadas. Além disso, introduzimos a proposta de criação do comitê de complementação de dados, que, a partir de um conjunto de sugestões de valores de uma ou mais estratégias, gera um novo valor. O capítulo 5 analisa experimentalmente testes realizados em três bases de dados normalmente utilizadas como *benchmarks* de mineração de dados: *Iris Plants*, *Wisconsin Breast Cancer* e *Pima Indians Diabetes*, todas do repositório da Universidade da Califórnia, Irvine. O capítulo 6 tece considerações finais, indicando os trabalhos futuros relacionados a esta tese.

CAPÍTULO 2

A ETAPA DE PRÉ-PROCESSAMENTO DE DADOS NO CONTEXTO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

2.1 Introdução

Uma das mais importantes fases no processo de descoberta de conhecimento em bases de dados é a etapa de pré-processamento de dados do processo de Descoberta de Conhecimento em Bases de Dados. Nela, os dados são tratados para a tarefa de mineração de dados. Esta pressupõe que os dados de uma base de dados estão completos, sem erros e bem distribuídos. Todavia, os dados do mundo real são bastante imprecisos, e, como bem frisa MOTRO (1995), bancos de dados modelam o nosso conhecimento do mundo real. Este conhecimento é frequentemente permeado por incertezas. Assim, elas devem ser capazes de lidar com incertezas.

Esta preparação é de vital importância, tendo em vista que, se ela não for feita de forma adequada, todo o processo de descoberta de padrões ocultos nos dados pode não ser bem sucedido. Este processo demanda, na prática, 80% do total de esforços de tratamento de dados (ZHANG *et al*, 2003).

Muitos esforços foram e são comumente despendidos focando a tarefa de mineração de dados. Entretanto, a evolução na tarefa de pré-processamento de dados, extremamente importante para a obtenção de padrões de qualidade ocultos nos dados, não recebeu um investimento equivalente em pesquisa na última década. Com isso, apesar de termos à disposição ferramentas de mineração de dados bastante eficazes, tais como regras de associação, árvores de decisão e classificadores, de nada elas adiantarão se os dados que forem analisados contiverem dados ausentes, inconsistentes ou enviesados. Assim, podemos observar que a tarefa de pré-processamento guarda consigo uma importância tão grande quanto a tarefa de mineração de dados.

2.2 Contextualização: O Processo de Descoberta de Conhecimento em Bases de Dados

2.2.1 Introdução

A definição clássica do processo de Descoberta de Conhecimento em Bases de Dados, também conhecido por KDD – *Knowledge Discovery in Databases* (seu acrônimo em inglês), foi apresentada por FAYYAD, PIATETSKY-SHAPIRO e SMYTH (1996a):

"*KDD é um processo, de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados*".

Este processo tem dois objetivos: **verificação** e **descoberta** (FAYYAD *et al*, 1996a). No primeiro caso, o sistema limita-se a verificar hipóteses pré-estabelecidas. Já no segundo caso, o sistema é capaz de automaticamente descobrir novos padrões ocultos nos dados. A descoberta pode ser subdividida em **preditiva** – onde o sistema descobre padrões que indicam um comportamento ou acontecimento futuro de algumas entidades – e **descritiva**, que encontra padrões a serem apresentados de uma forma inteligível aos seres humanos.

O processo de Descoberta de Conhecimento em Bases de Dados é basicamente composto por três etapas: **Pré-Processamento**, **Mineração de Dados** e **Pós-Processamento**. Conforme mencionado no parágrafo anterior, o processo completo é por vezes somente chamado Mineração de Dados, já que esta é, por diversas vezes, considerada a etapa mais importante de todo o processo. E, de fato, é durante esta etapa que as relações entre os elementos da base de dados são descobertas.

HAND, MANNILA e SMYTH (2001) definem a etapa de mineração de dados da seguinte forma:

"*Mineração de Dados é a análise de (quase sempre grandes) conjuntos dados observados para descobrir relações escondidas e para consolidar os dados de uma forma tal que eles sejam inteligíveis e úteis aos seus donos*".

Já WITTEN e FRANK (2005) trabalham o conceito de mineração de dados com um alto viés de aprendizado de máquina:

"*Mineração de Dados é definida como o processo de descoberta de padrões nos dados. O processo pode ser automático ou (mais comumente) semi-automático. Os padrões descobertos devem ter significado de tal forma que eles tragam alguma vantagem, normalmente econômica. Os dados devem estar invariavelmente expressos em grandes quantidades*".

Nas duas definições acima, fica clara a diferença entre todo o processo de descoberta de conhecimento e a etapa de mineração de dados. Por esta razão, iremos adotá-la, assim como FAYYAD, PIATETSKY-SHAPIRO e SMYTH (1996a, 1996b) e GOLDSCHMIDT e PASSOS (2005). A figura 2.1 ilustra todo o processo.

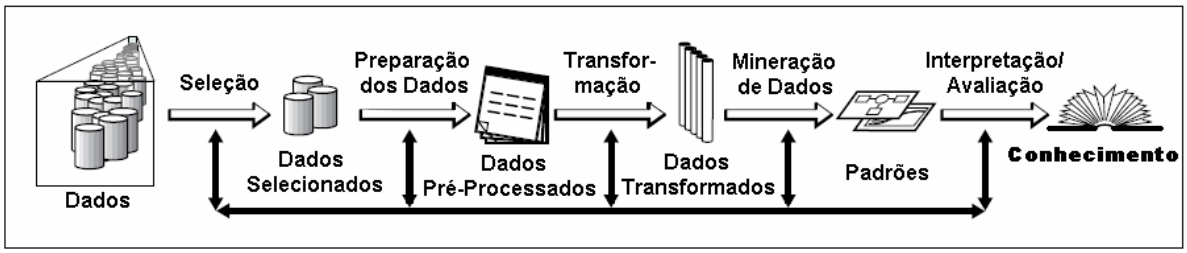


Figura 2.1 Visão geral do processo de KDD.
Fonte: Adaptado de (*FAYYAD* et al*, 1996b*)

GOLDSCHMIDT e PASSOS (2005) destacam a importância, neste contexto, da interferência do analista de conhecimento, que, com o seu conhecimento sobre a natureza dos dados, pode aumentar em muito os resultados de todo o processo. A figura 2.2 (GOLDSCHMIDT, PASSOS, 2005, GOLDSCHMIDT, 2003) representa esta idéia.

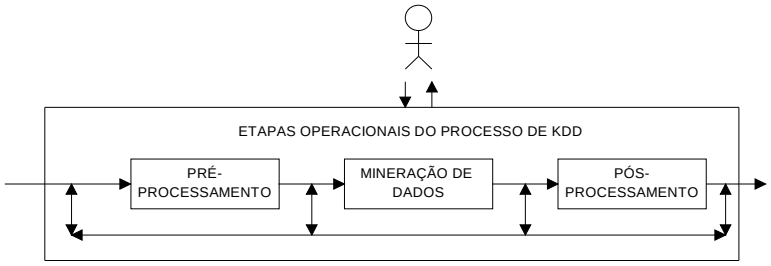


Figura 2.2 Etapas operacionais do processo de KDD.
Fonte: (*GOLDSCHMIDT, PASSOS, 2005*)

O processo de Descoberta de Conhecimento em Bases de Dados é multidisciplinar. Ele possui interfaces não só com a área de Banco de Dados, mas também com a Inteligência Computacional – especialmente com o Aprendizado de Máquina, com a Estatística, Reconhecimento de Padrões, Armazém de Dados (*Data Warehousing*), Visualização de Dados, entre outras. Por isso, focaremos nesta tese a

contribuição que técnicas que envolvam aprendizado de máquina podem contribuir com a melhoria do processo.

2.2.2 Tarefas de Mineração de Dados

A etapa de Mineração de Dados normalmente é a que atrai maior atenção, por ser ela que revela os padrões ocultos nos dados. Todavia, podemos tranquilamente afirmar que a etapa de pré-processamento é tão ou mais importante em todo o processo, pois dados com erros, incompletos, redundantes ou desnivelados podem comprometer todo o processo. Este é o nosso objetivo nesta tese: a de focar na etapa de pré-processamento, especificamente na tarefa de limpeza de dados. Por esta razão, dedicaremos especial atenção a esta etapa no próximo capítulo. No restante desta seção, trataremos das demais etapas envolvidas em todo o processo.

2.2.2.1 Associação

Regras de associação são implicações da forma $X \rightarrow Y$, onde X e Y, respectivamente denominados **antecedente** e **conseqüente**, são conjuntos de itens tais que $X \cap Y = \varnothing$ (esta última condição assegura que regras triviais não sejam geradas). O objetivo desta tarefa é então encontrar regras na forma acima que sejam freqüentes e válidas, e que atendam a critérios de **suporte** e **confiança** mínimos.

Os conceitos de suporte e confiança são extremamente importantes nesta tarefa, e, como já visto, servem como parâmetro para os algoritmos que extraem regras de associação de um conjunto de dados. O *suporte* está relacionado à freqüência que uma regra ocorre em uma tabela. Seja a regra $r = X \rightarrow Y$, nX e nY o número de vezes que os valores X e Y aparecem em seus respectivos atributos, e D o número de ocorrências (registros) da tabela. Seu suporte é definido como:

$$suporte\ (r) = \frac{nX \wedge nY}{D}$$

Já a *confiança* reflete a validade de uma regra. Esta medida procura expressar a qualidade de uma regra, indicando o quanto a ocorrência do seu antecedente pode assegurar a ocorrência do seu conseqüente. Ela é expressa por:

$$confiança\ (r) = \frac{nX \wedge nY}{nX}$$

Um interessante e didático exemplo de como obter associações a partir de um conjunto de dados é oferecido por GOLDSCHMIDT e PASSOS (2005). Neste exemplo (tabela 2.1), que simula operações de venda de produtos de um supermercado – cada operação de venda é chamada **transação** – o objetivo é descobrir produtos que sejam freqüentemente vendidos de forma conjunta.

Tabela 2.1: Relação das vendas de um Mini-Mercado em um período

Transação	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

As regras abaixo poderiam ser extraídas:

r1: Leite → Pão

r2: Café → Pão

r3: Pão ∧ Manteiga → Café

r4: Café ∧ Pão → Manteiga

e seus respectivos suporte e confiança são:

	r1	*r2*	*r3*	*r4*
suporte	20%	30%	30%	30%
confiança	100%	100%	75%	100%

Existem diversos algoritmos desenvolvidos especificamente para aplicação na tarefa de descoberta de associações. O mais famoso é o *Apriori*, de AGRAWAL, IMIELINSKI e SWAMI (1993). Porém, existem outras opções, tais como DHP–*Direct Hashing and Pruning* (HOLT, CHUNG, 1999), DIC–*Dynamic Itemset Counting* (BRIN *et al*, 1997), *Eclat* e *MaxEclat* (ZAKI *et al*, 1997) e *EstMerge* (SRIKANT,

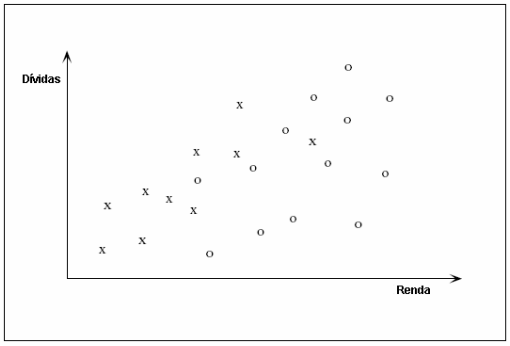
AGRAWAL, 1997). Existem versões destes algoritmos para funcionamento em ambientes paralelos e distribuídos.

2.2.2.2 Classificação

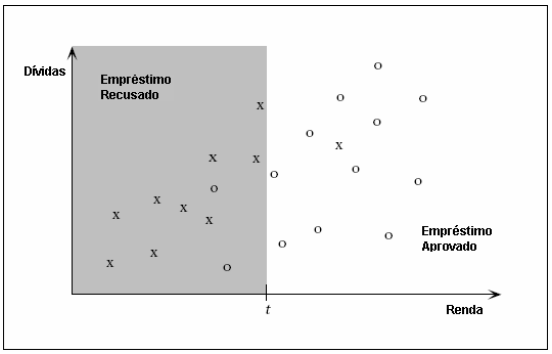
A tarefa de classificação essencialmente procura mapear registros de uma tabela a uma ou mais de *n* possíveis classes C_i. Se conseguirmos obter a função que realiza este mapeamento, quaisquer novas tuplas podem ser também mapeadas, sem a necessidade de conhecimento prévio da sua classe.

O exemplo da figura 2.3 (FAYYAD *et al*, 1996a) ilustra bem como a tarefa de classificação pode ser realizada. A figura 2.3(a) mostra um conjunto de dados de clientes de uma financeira, com dois atributos de entrada: a sua *renda* e o seu montante de *dívidas*. Além disso, pontos marcados com "X" representam clientes que não conseguiram pagar seus débitos, enquanto os marcados com "O" revelam aqueles com um bom histórico de pagamento. O objetivo deste exemplo é o de classificar estes clientes, em função de sua renda, montante de dívidas e histórico de pagamento de débitos passados, quais deles estão aptos a contrair um novo empréstimo (classes *Empréstimo Aceito* e *Empréstimo Recusado*).

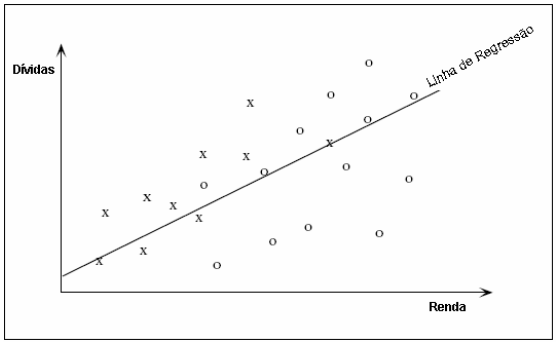
Se escolhêssemos uma função do tipo limiar (*threshold*), com *renda* = ***t***, todos os clientes que possuíssem uma renda menor do que *t* estariam inabilitados a receber o empréstimo (Figura 2.3a). Este classificador ignora o nível de endividamento do cliente, bem como a sua característica de honrar pagamentos, e talvez não fosse um bom classificador. Obteríamos uma evolução com a descoberta de uma função linear (Figura 2.3c), que buscasse relacionar as variáveis *renda* e *dívidas*. Esta função poderia gerar o modelo de classificação explicitado na figura 2.3d. Todavia, um modelo não linear (por exemplo, uma rede neuronal artificial), mostrado na figura 2.3e, que levasse em consideração todas as variáveis de entrada gerasse uma divisão irregular do espaço que se mostra mais precisa do que a linear. A figura 2.3f nos apresenta uma outra solução, baseada no conceito de vizinhos mais próximos, a ser abordada no próximo capítulo. Com este modelo, ao menos visualmente, obtemos a melhor classificação para este exemplo. Entretanto, todas estas possibilidades estão intimamente ligadas a um correto ajuste de parâmetros, e das características dos dados.



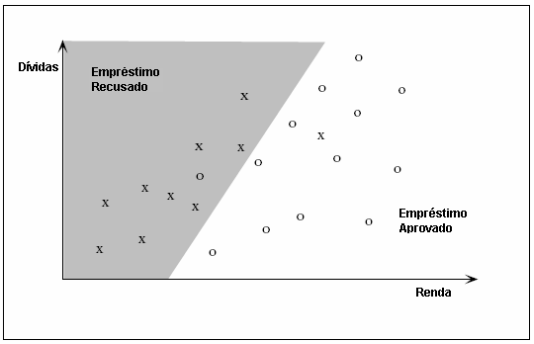
(a) Conjunto original de dado de clientes de uma financeira



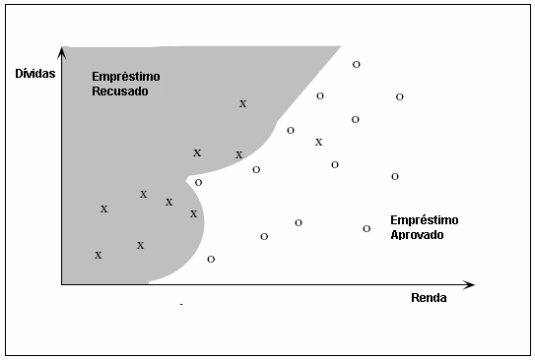
(b) Conjunto de dados classificados com threshold



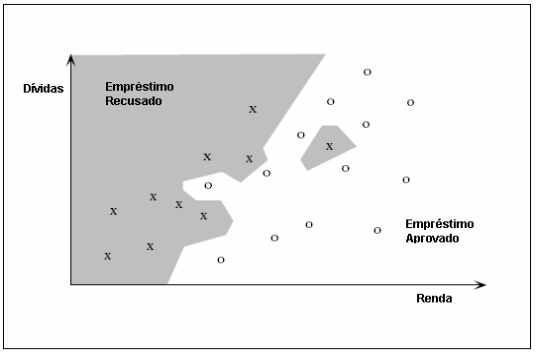
(c) Uma função linear classificadora dos dados



(d) Conjunto de dados classificados linearmente



(e) Classificação não linear



(f) Classificação com os vizinhos mais próximos

Figura 2.3 Um conjunto fictício de dados classificado de diversas formas.

Fonte: Adaptado de (*FAYYAD* et al*, 1996a*)

WITTEN e FRANK (2005) apresentam um outro interessante exemplo onde a classificação pode gerar um modelo de conhecimento útil para tomada de decisões. A tabela 2.2 apresenta um conjunto de dados oftalmológicos de pacientes que pretendem usar lentes de contato. O último atributo da tabela, *lentes de contato recomendadas*, indica se o paciente pode ou não usar lentes, e, em caso positivo, que tipo é o mais adequado ao seu caso. A partir deste conjunto de dados, podemos gerar um modelo de conhecimento que permita indicar ou não lentes de contato de forma automática.

A árvore de decisão mostrada na figura 2.4 foi gerada pelo algoritmo J4.8 (WITTEN, FRANK, 2005), uma variante do clássico algoritmo de classificação *C4.5* de QUINLAN (1993), especificado mais adiante nesta seção. Utilizamos neste exemplo o *framework* de mineração de dados *Weka* (WEKA, 2007, WITTEN, FRANK, 2005), uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Os algoritmos podem ser tanto aplicados diretamente ao conjunto de dados, quanto chamados a partir de um programa em Java.

Tabela 2.2 Dados sobre pacientes candidatos a usarem lentes de contato. O atributo classificador indica se o paciente pode usar lentes, e, em caso positivo, qual o tipo recomendado. Fonte: (*WITTEN, FRANK, 2005*)

idade	prescrição do problema	astigmatismo	taxa de produção de lágrimas	lentes de contato recomendadas
jovem	míope	não	reduzida	nenhuma
jovem	míope	não	normal	moldável
jovem	míope	sim	reduzida	nenhuma
jovem	míope	sim	normal	rígida
jovem	hipermétrope	não	reduzida	nenhuma
jovem	hipermétrope	não	normal	moldável
jovem	hipermétrope	sim	reduzida	nenhuma
jovem	hipermétrope	sim	normal	rígida
pré-presbiótico	míope	não	reduzida	nenhuma
pré-presbiótico	míope	não	normal	moldável
pré-presbiótico	míope	sim	reduzida	nenhuma
pré-presbiótico	míope	sim	normal	rígida
pré-presbiótico	hipermétrope	não	reduzida	nenhuma
pré-presbiótico	hipermétrope	não	normal	moldável
pré-presbiótico	hipermétrope	sim	reduzida	nenhuma
pré-presbiótico	hipermétrope	sim	normal	nenhuma
presbiótico	míope	não	reduzida	nenhuma
presbiótico	míope	não	normal	nenhuma
presbiótico	míope	sim	reduzida	nenhuma
presbiótico	míope	sim	normal	rígida
presbiótico	hipermétrope	não	reduzida	nenhuma
presbiótico	hipermétrope	não	normal	moldável
presbiótico	hipermétrope	sim	reduzida	nenhuma
presbiótico	hipermétrope	sim	normal	nenhuma

Considerando todos os conjuntos de dados possíveis, não existe um algoritmo de classificação que se sobressaia, segundo o teorema ***NFL****–No Free Lunch Theorem* (WOLPERT, 1996). Ou seja, a cada nova aplicação envolvendo a tarefa de classificação, devemos experimentar os algoritmos disponíveis para identificar os que obtêm melhor desempenho.

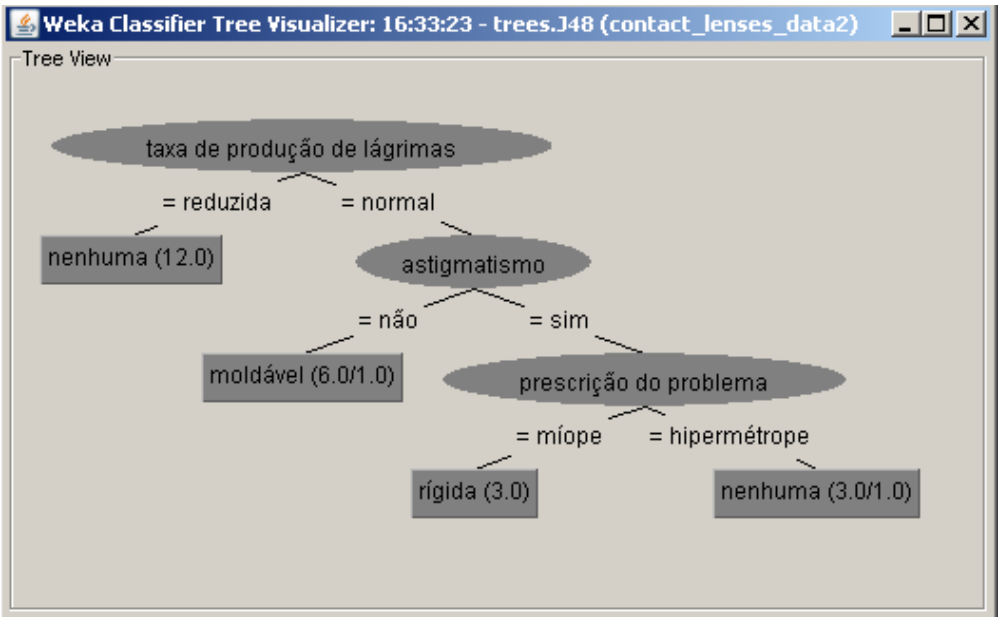


Figura 2.4 Exemplo de classificação com árvore de decisão obtida através dos dados sobre o uso de lentes de contato, a partir de algumas características de pacientes

O algoritmo de classificação C4.5 (QUINLAN, 1993) gera, a partir de um conjunto de dados com valores discretos, uma árvore de decisão baseado no conceito de **taxa de ganho de informação**. Além disso, ele consegue trabalhar com dados que apresentem valores ausentes em seus atributos. O algoritmo é especificado abaixo.

Entrada: Conjunto de treinamento T

Saída: Árvore de decisão em um classificador C

Algoritmo:

Procure por um atributo X que, submetido a um teste específico, gere as saídas O_1, ... , O_n.

Baseado nos valores O_i, o conjunto de testes é particionado em n partições disjuntas (T_1, T_2, ... , T_n), onde cada partição T_i contém todas as instâncias de T que satisfazem à saída O_i.

Se existirem tuplas com valores ausentes em X, utilizar o subconjunto com todos os valores conhecidos de X para calcular a taxa de ganho de informação.

Designar pesos nas atribuições de elementos às partições T_i:

a) se um elemento é associado diretamente, após a aplicação do teste, à saída O_i, isto indica que o peso da alocação dele em T_i é 1, e nas demais partições 0.

b) se um elementos não apresenta o valor de X, ele é alocado em todas as partições, com pesos w_j, $1 \leq j \leq n$.

c) Cálculo dos pesos:

$$cálculo\,dos\,pesos = \frac{\sum_{k=1}^{m}\left(w_k \mid E(k)\,satisfaz\,O_i\right)}{\sum_{l=1}^{m}\left(w_l \mid E(l)\,apresenta\,valor\,em\,X\right)}$$

onde *m* é o número de elementos.

GOLDSCHMIDT e PASSOS (2005) destacam exemplos de aplicação da tarefa de classificação: análise de crédito, análise de risco em seguros, diagnóstico de doenças e prescrição de tratamento, análise de defeitos em equipamentos, entre inúmeros outros.

2.2.2.3 Agrupamento

A tarefa de agrupamento reúne em um mesmo grupo objetos de uma coleção que mantenham algum grau de afinidade. Assim, a sua base é o conceito de similaridade, e o seu objetivo principal é o de maximizar a similaridade de objetos do mesmo grupo, e de minimizá-la entre os elementos de grupos distintos. A figura 2.5 explana esta idéia.

A diferença principal entre esta tarefa e a de classificação é que, nesta última, conhecemos previamente a que classe os elementos de uma coleção pertencem. No agrupamento, esta discriminação não é conhecida.

Esta tarefa não é exclusiva da etapa de mineração de dados. Conforme já mencionado anteriormente, o agrupamento é igualmente importante na etapa de pré-processamento de dados. Por esta razão, ela será detalhada no próximo capítulo, na seção 2.6.

2.2.2.4 Outras tarefas da etapa de Mineração de Dados

- **Descoberta de Associações Generalizadas**: A tarefa de descoberta de associações generalizadas não se restringe à busca por associações no nível mais primitivo de abstração. Leva em consideração a hierarquia conceitual eventualmente existente entre os itens de dados, de forma a identificar regras de associações que envolvam múltiplos níveis de abstração de conceitos.
- **Descoberta de Seqüências**: Extensão da tarefa de descoberta de associações, que considera o aspecto temporal entre as transações registradas na base de dados. Na descoberta de associações, os padrões a serem descobertos pertencem a cada transação. São denominados padrões intra-transação. No caso da

descoberta de seqüências, os padrões são denominados inter-transação, pois diversas transações devem ser analisadas em ordem cronológica de ocorrência.

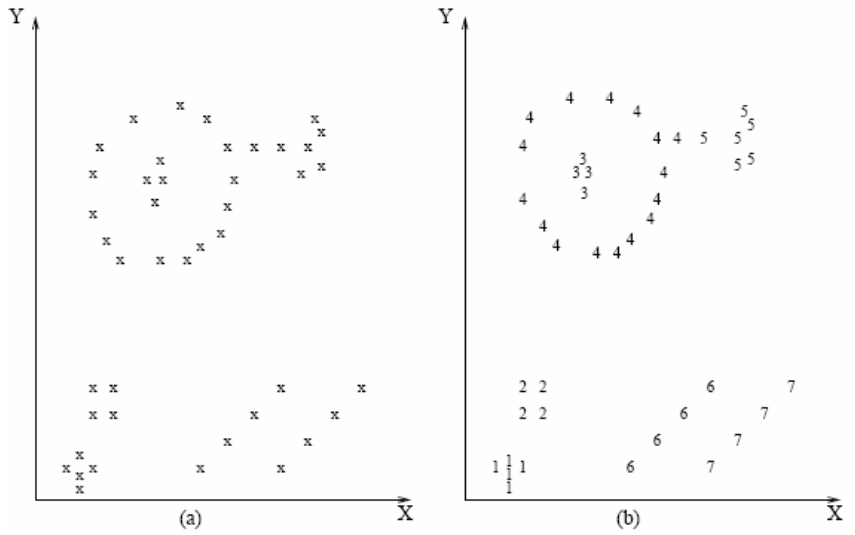


Figura 2.5 Exemplo de agrupamento de dados em sete grupos, onde o rótulo sobre o elemento indica a qual grupo ele pertence. Fonte: (JAIN et al, 1999)

- **Sumarização**: Consiste em identificar e apresentar, de forma concisa e compreensível, as principais características dos dados contidos em um conjunto de dados, gerando descrições para caracterização resumida dos dados e possivelmente comparação entre eles.
- **Previsão de Séries Temporais**: Uma série temporal é um conjunto de observações de um fenômeno ordenadas no tempo. A sua análise é o processo de identificação das características, padrões e propriedades importantes, utilizados para descrever em termos gerais o seu fenômeno gerador. Dentre os diversos objetivos da análise de séries temporais, o maior deles é a geração de modelos voltados à previsão de valores futuros.

Outras tarefas, tais como Descoberta de Associações Generalizadas, Descoberta de Seqüências, Descoberta de Seqüências Generalizadas e Previsão de Séries Temporais também integram a etapa de Mineração de Dados, e são bastante importantes na busca por padrões ocultos em bases de dados.

2.2.3 Pós-Processamento

Nesta etapa, o objetivo é transformar os padrões (obtidos na etapa de mineração de dados em conhecimento) inteligíveis, tanto ao analista de dados quanto ao especialista do domínio da aplicação. Estes dois podem, neste momento, definir novas estratégias de

investigação de dados, caso julguem necessário. Algumas das operações existentes nesta etapa são (GOLDSCHMIDT, PASSOS, 2005):

- **Simplificações do Modelo de Conhecimento**: O volume de padrões gerados pela etapa de mineração de dados pode gerar um conjunto de elementos de difícil trato. Isto acontece de forma mais freqüente com regras de associação. Um número grande de regras ou de variáveis em cada uma delas faz com que o conhecimento seja de difícil extração. Assim, o objetivo aqui é reduzir o volume de padrões gerados, de forma a facilitar o entendimento do modelo gerado.
- **Transformações do Modelo de Conhecimento**: Tarefa que converte um modelo de conhecimento em outro, buscando facilitar a sua compreensão. Um comum exemplo é a transformação de regras de associação em árvores de decisão.
- **Organização e Apresentação dos Resultados**: Facilitam o entendimento do modelo de conhecimento gerado, através da visualização em duas ou três dimensões. Aqui é freqüente o uso de tabelas, cubos, gráficos, planilhas, ou qualquer outra ferramenta que facilite o entendimento do modelo de conhecimento.

2.3 Natureza dos dados

Os atributos de uma tabela podem ser divididos em duas classes: **numéricos** e **categóricos**. Atributos **numéricos** são quantitativos, apresentando ou não limites inferior e superior, e possuindo a noção de ordem total, e subdividem-se em atributos **contínuos** e **discretos**. Já os atributos **categóricos** ou **nominais** indicam o conceito ao qual o objeto se enquadra, revelando em que categoria ele se encontra. Neste tipo de atributos não existe a noção de ordem entre valores, e podem ser representados por tipos alfanuméricos. Por fim, encontramos os atributos **discretos**, que diferem dos atributos categóricos por trazer consigo a noção de ordem. Como exemplo, considere uma tabela de dados de clientes de uma empresa. Nesta relação, podemos encontrar três atributos: *renda*, *sexo* e *escolaridade*, que são respectivamente dos tipos contínuo, categórico e discreto.

2.4 Etapa do Pré-Processamento de Dados

Algumas das mais importantes atividades da etapa de pré-processamento de dados são as seguintes (GOLDSCHMIDT, PASSOS, 2005, ZHANG *et al*, 2003):

- Agrupamento de dados
- Coleta e Integração
- Codificação
- Construção de Atributos
- Correção de Prevalência
- Discretização
- Enriquecimento
- Limpeza dos Dados
- Normalização de Dados
- Partição dos Dados
- Redução de Dados
- Seleção de Dados

2.4.1 Agrupamento de dados

Apesar de classicamente abordada no contexto da mineração de dados, o agrupamento é uma tarefa extremamente importante também na etapa de pré-processamento de dados, e que será abordada na seção 2.6 com maior destaque, dado o seu uso em nosso trabalho.

2.4.2 Coleta e Integração

Por diversas vezes, é importante e necessário que dados provenientes de diversas fontes sejam consolidados em uma base de dados. Esta etapa, chamada **coleta e integração**, é bastante comum construção de *data warehouses*[1], mas pode acontecer também em outros tipos de conjuntos de dados. Todavia, é inegável que o problema de integração de dados é mais intenso na construção de *data warehouses*. A carga contínua de seus arquivos aumenta consideravelmente a probabilidade de ocorrência de valores ausentes (RAHM, DO, 2000), problema alvo de estudo desta tese. Vários aspectos devem ser considerados no processo de integração de dados (HAN, KAMBER, 2005): a *integração de esquemas*, a *eliminação de redundâncias* e a *detecção e correção de dados com valores conflitantes*.

[1] Optamos por não traduzir o termo *Data Warehouse*, amplamente conhecido em Computação, para *Armazém de Dados*.

2.4.3 Codificação

Codificar significa transformar a natureza dos valores de um atributo. Na etapa de pré-processamento, isto pode acontecer de duas diferentes formas: uma transformação de dados numéricos em categóricos – codificação **numérico-categórica**, ou o inverso – codificação **categórico-numérica** (GOLDSCHMIDT, 2003).

Duas opções de codificação numérico-categórica são possíveis. O primeiro deles, o **mapeamento direto**, converte um valor numérico em um categórico (por exemplo, em um atributo que indique o sexo de funcionários, fazer o valor 0 corresponder ao valor "M", e 1 ao "F"). Já a segunda opção, conhecida como **mapeamento em intervalos**, ou **discretização**, divide o domínio do atributo em intervalos. Para ilustrar este conceito, considere um atributo que indique a renda de um cliente. Podemos, por alguma razão, admitir que rendas mensais de até R$ 1.000,00 assumem o valor *baixa*; de R$ 1.000,00 a R$ 4.000,00 a categoria *média*; e *alta*, para valores maiores de R$ 4.000,00.

A discretização envolve métodos que dividem o domínio de uma variável numérica em intervalos. Como exemplo considere o atributo renda com os seguintes valores, já organizados em ordem crescente: 1000, 1400, 1500, 1700, 2500, 3000, 3700, 4300, 4500, 5000. Assim, podemos discretizar os valores do domínio da seguinte forma:

- **Divisão em intervalos com comprimentos definidos pelo usuário**: Neste caso, o usuário define o número de intervalos e escolhe o tamanho de cada um deles. Por exemplo: três intervalos de comprimento 1600, 2800 e 1000.
- **Divisão em intervalos de igual comprimento**: O usuário define somente o número de intervalos. O comprimento destes intervalos é calculado a partir do maior e menor valor do domínio do atributo. Exemplo: quatro intervalos, de comprimento = (5000 – 1000)/4.
- **Divisão em intervalos de igual comprimento**: Variante do anterior. Utiliza algum critério específico para definir o número de intervalos. Ex: O número de intervalos (k) é 5 se o número de amostras for inferior ou igual a 25 e a raiz quadrada do número de amostras. O comprimento de cada intervalo é obtido por $h=R/k$. No exemplo, h=4000/5=800.

- **Divisão dos Valores em Grupos**: Consiste em agrupar os valores de um atributo em *clusters* (grupos) levando em consideração a similaridade existente entre tais valores. Ex: Considere k = 3 (3 grupos). Assim, uma possível divisão dos valores do exemplo em grupos seria:

  Grupo 1: 1000, 1400, 1500, 1700

  Grupo 2: 2500, 3000, 3700

  Grupo 3: 4300, 4500, 5000

A discretização é um problema complexo e vem sendo amplamente pesquisada. LIU, WHITE, THOMPSON *et al* (2002) revisam os métodos mais comuns que são utilizados nesta tarefa, e busca a padronização deste processo – oferecendo um vocabulário unificado para discutir vários métodos propostos por diversos autores, definindo todo o processo, e discutindo diferentes formas de avaliar os resultados deste processo. Além disso, este trabalho propõe um *framework* hierárquico para métodos de discretização, descrevendo métodos representativos de forma concisa. Ao descrever um método representativo, os autores o executam e geram resultados na base de dados *Íris Plants* (NEWMAN *et al*, 1998), também utilizada nesta tese.

A codificação categórico-numérica, por sua vez, transforma valores do domínio de um atributo na representação binária de números discretos entre 1 e N. Algumas implementações comuns desta opção de codificação:

a) **Representação Binária *1-de-N***: cada um dos k possíveis valores do domínio do atributo ($1 \leq k \leq N$) é associado a uma posição de uma cadeia de N bits. Assim, faz-se o bit k da cadeia assumir o valor um, e os demais bits são levados ao valor zero. Como exemplo, imagine um atributo categórico que possua em seu domínio os valores *excelente, bom*, *regular*, *ruim* e *péssimo*. A representação binária *1-de-N* deste atributo utiliza uma cadeia de cinco bits, representada conforme a tabela abaixo:

Excelente	10000
Bom	01000
Regular	00100
Ruim	00010
Péssimo	00001

b) **Representação Binária Econômica**: nesta opção, a representação dos N valores do domínio de um atributo são representados por $k = \lceil \log_2 N \rceil$ bits.

Nesta opção, associa-se um número seqüencial, a partir de um, para todos os valores do domínio. A representação é feita com a representação em binário deste número. Observe o exemplo anterior codificado segundo esta opção:

Excelente	000
Bom	001
Regular	010
Ruim	011
Péssimo	100

c) **Representação Binária por Temperatura**: os *N* valores do domínio do atributo voltam, nesta opção, a serem representados com N bits. A designação numérica adotada é importante na codificação, já que ela indicará o número de bits que assumem o valor um. Assim, utilizando o exemplo anterior, temos:

Excelente	00001
Bom	00011
Regular	00111
Ruim	01111
Péssimo	11111

Podemos aqui verificar o quão similares são os valores do domínio, utilizando a *Distância de Hamming*, que conta para cada posição *i* da cadeia de bits, se eles são iguais ou diferentes. Se forem diferentes, soma-se uma unidade a um contador. Por exemplo, a distância de Hamming entre os valores bom e ruim é igual a *dh(bom,ruim)* = 2, já que o segundo e o terceiro bit destes valores são diferentes.

Este conceito pode ser estendido para um mais genérico. Com todas as combinações de distâncias entre os elementos, dois a dois, podemos montar uma **matriz de similaridades**. Ainda tomando por base o nosso exemplo, vemos que a matriz de similaridades para o domínio em questão é:

DH	excelente	bom	regular	ruim	péssimo
excelente	0	1	2	3	4
bom	1	0	1	2	3
regular	2	1	0	1	2
ruim	3	2	1	0	1
péssimo	4	3	2	1	0

2.4.4 Construção de Atributos

A consolidação dos dados contidos em tabelas pode melhorar consideravelmente o processo de descoberta de conhecimento em bases de dados. Por isto, uma das tarefas da etapa de pré-processamento de dados é a **construção de atributos**, que cria novas colunas na tabela, refletindo alguma transformação dos dados existentes nas tabelas de um conjunto de dados.

Como exemplo, considere uma tabela com dados sobre pedidos, que contém, dentre outros atributos, as colunas *valor_unitário_item*, *total_itens*, *data_pedido* e *data_entrega*. Poderíamos construir dois novos atributos nesta tabela: um que indica o número de dias transcorridos entre o pedido e a entrega (*tempo_entrega = data_entrega - data_pedido*) e o valor total da nota (*valor_total_nota = valor_unitário_item * total_itens*). Estes dois atributos, na verdade, são produtos de consultas feitas na base de dados, e, por isso, poderiam ser calculados. Entretanto, este acesso à base de dados pode ser custoso se realizado muitas vezes. Além disso, a grande maioria dos algoritmos de mineração de dados não consegue trabalhar com atributos não consolidados.

2.4.5 Correção de Prevalência

Por vezes, os registros de um conjunto de dados apresentam a maioria de suas ocorrências voltadas a uma categoria específica. Como exemplo, considere uma base de dados de crédito pessoal de uma empresa financeira. Se, para a felicidade de seus proprietários, a grande maioria de seus clientes honrarem em dia, o percentual de objetos que indicam perfis de clientes que provavelmente gerarão problemas para honrar seus compromissos será bem pequeno (por exemplo, 1% dos clientes), e o algoritmo de classificação ficará neste caso enviesado.

Desta forma, a correção de prevalência busca corrigir este problema, equilibrando estatisticamente a ocorrência das classes na base. Os dois métodos mais comuns que implementam esta estratégia de correção são a **replicação** ou a **eliminação aleatória de registros**. Neste último caso, podemos adicionar um ou mais critérios ao processo de eliminação, que envolvam um ou mais atributos da tabela. A este processo chamamos **amostragem estratificada** (GOLDSCHMIDT, PASSOS, 2005).

Um terceiro método é a utilização de técnicas de Mineração de Dados adequadas para lidar com problemas de prevalência, valendo-se do conceito de **matriz de custo** (o

peso do erro associado aos registros cujas classes sejam menos numerosas é normalmente maior).

2.4.6 Enriquecimento de Dados

Por vezes, o analista de dados tem condições de, a partir de sua experiência e conhecimento do negócio, adicionar à tabela informações que aumentem a semântica dos dados. A este processo chamamos **enriquecimento de dados**. Estas novas informações podem também ser obtidas por meio de pesquisas ou consultas a bases externas.

Como exemplo, considere uma tabela com perfis de clientes. Esta tabela tem, dentre outros, os atributos *renda*, *despesas*, *tipo_residencia* e *bairro_residencia*. Podemos acrescentar a esta tabela a coluna *valor_medio_imovel*, e aumentar as chances de obtenção de um resultado mais satisfatório na mineração de dados. Note que esta tarefa diferencia-se da construção de dados, pela criação de atributos que não são produtos de consultas à tabela original.

2.4.7 Limpeza dos Dados

A análise e inferência de padrões ocultos de dados pressupõem obviamente que existam dados a serem processados. Assim, uma importante tarefa de pré-processamento é sem dúvida a limpeza dos dados, que pode ser dividida nas seguintes subtarefas:

- Complementação de Dados Ausentes
- Detecção de Ruídos
- Eliminação de Dados Inconsistentes

O objetivo deste trabalho é o de analisar **complementação de dados** e, por esta razão, merecerá uma atenção especial, sendo detalhada no Capítulo 3.

A **detecção de ruídos** é definida por HAN e KAMBER (2005) como a busca pelo ajuste de variáveis com erros aleatórios nos seus valores ou na sua variância (definiremos o conceito de variância na Seção 2.5.4.2, ao tratarmos do método de Análise de Componentes Principais, no contexto de Seleção de Variáveis).

Algumas técnicas de suavização de dados são as seguintes (HAN, KAMBER, 2005):

1. **Encaixotamento**: suavizam um conjunto de dados ordenados consultando a sua vizinhança, ou seja, valores ao seu redor. Os valores ordenados são distribuídos em um número de "compartimentos", ou bins. Já que este método consulta a vizinhança de valores, eles realizam suavização local.
2. **Agrupamento**: valores que destoam muito da tendência geral dos dados de um conjunto podem ser organizados em grupos. Intuitivamente, valores que não podem ser enquadrados em um cluster podem ser considerados ruidosos. Abordaremos sobre agrupamento na seção 2.6.
3. **Combinação da análise humana e computadorizada**: valores que destoam muito da tendência geral dos dados de um conjunto podem ser identificados através da combinação da análise humana e computadorizada.
4. **Regressão**: Dados podem ser suavizados se forem ajustados a uma função, como é feito na regressão. A regressão linear procura descobrir a melhor reta que representa duas variáveis, assim uma variável pode ser usada para predizer a outra. Regressão linear múltipla é uma extensão da regressão linear, onde mais de duas variáveis estão envolvidas e os dados devem ser ajustados a uma superfície plana multidimensional. Usando a regressão para encontrar uma equação matemática para se ajustar aos dados ajuda a suavizá-los. Por sua importância no contexto deste trabalho, abordaremos o tema com maior detalhamento na seção 2.7.

Vários métodos para a suavização de dados são também usados na redução de dados envolvendo discretização. Por exemplo, a técnica de encaixotamento descrita acima reduz o número de valores distintos por atributo. Esta é uma forma de redução de dados para métodos de mineração de dados que trabalham com dados discretos, tais como árvores de decisão, que repetidamente comparam valores de dados ordenados. Alguns métodos de classificação, tais como redes neuronais artificiais, possuem mecanismos internos de suavização de dados. A seção 2.7.4 trata do assunto.

A verificação da existência de **dados inconsistentes** é da mesma forma uma tarefa de limpeza de dados. Inconsistências podem facilmente acontecer no processo de integração de dados, onde tabelas com atributos com nomes diferentes devem se transformar em uma única. Dados inconsistentes também podem ser gerados por transações, se as restrições de integridade das tabelas de um banco de dados não tiverem

sido impostas da forma correta. Erros tais como falhas na integridade referencial (onde uma chave estrangeira de uma tupla assume um valor que não existe na tabela com que ela se relaciona), ou valores fora do domínio do atributo podem ser corrigidos ou manualmente (pela observação do analista de dados), ou por ferramentas, tanto as do Sistema Gerenciador de Banco de Dados, quanto as de Mineração de Dados.

O trabalho de MALETIC e MARCUS (2000) define a etapa de limpeza de dados de uma forma mais ampla, e a vê como um processo, composto das seguintes etapas:

- Definição e determinação dos tipos de erros
- Busca e identificação das ocorrências de erros
- Correção propriamente dita dos erros

Eles mencionam que cada uma destas três tarefas constitui um problema por si só. Uma ampla variedade de métodos e técnicas pode ser aplicada para cada uma delas. Assim, eles frisam que as soluções genéricas nesta área devem focar as duas primeiras subtarefas, já que a última é extremamente difícil de automatizar, por depender do domínio. Os autores propõem assim uma solução que envolva análise estatística, agrupamento, regras de associação e identificação de padrões dos dados, para a detecção de desvios de valores.

2.4.8 Normalização dos Dados

A tarefa de normalização de dados busca a diminuição do espectro de variação dos valores de uma variável, para evitar que esta alternância, se dada de forma brusca, possa comprometer o processo de mineração de dados. Os valores de um atributo são então mapeados para um dos intervalos [0.0, 1.0] ou [-1.0, 1.0].

O assunto por si só é amplo o bastante, e já merecedor de um estudo por si só. HAN e KAMBER (2005) enumeram algumas das normalizações mais comuns:

a) **Normalização Linear**: assume que a distribuição dos valores da coluna respeita uma função linear, preservando a relação existente entre os dados originais.

$$A' = \frac{(A - A_{\min})}{(A_{\max} - A_{\min})} * (novo\_\max_A - novo\_\min_A) + novo\_\min_A$$

b) **Normalização pela Soma**: divide o valor da variável pelo somatório dos valores da coluna.

$$A' = \frac{A}{\sum A}$$

c) **Normalização por Score-Z (normalização por média zero)**: desloca o eixo central para a média dos valores do atributo, e os normaliza em função do seu desvio-padrão:

$$A' = \frac{(A - \overline{A})}{\sigma}$$

d) **Normalização pelo Máximo**: considera que o maior valor existente para o atributo em questão é o fator de normalização.

$$A' = \frac{A}{A_{\max}}$$

e) **Normalização por Escala Decimal**: muda a escala de casas decimais dos valores do atributo. O número de casas decimais depende do valor máximo absoluto de *A*.

$$A' = \frac{A}{10^k}$$

Aqui, *k* é o menor inteiro tal que $\max|A'| < 1$.

2.4.9 Criação de Partições dos Dados

A criação de partição de dados consiste em se separar os dados em dois conjuntos disjuntos: um para **treinamento** (abstração do modelo de conhecimento) e outro para **testes** (avaliação do modelo gerado).

Exemplo: Considere uma base de dados sobre crédito.

Dados para Treinamento → Modelo de Conhecimento

Dados para Testes → Avaliação da Qualidade do Modelo

A partição de dados pode ser feita utilizando várias abordagens:

- ***Holdout**:* este método divide aleatoriamente os registros em uma percentagem fixa *p* para treinamento e (*1 – p*) para teste, considerando normalmente $p > \frac{1}{2}$.

  Esta abordagem é muito utilizada quando desejamos produzir um único modelo de conhecimento a ser aplicado posteriormente em algum sistema de apoio à decisão.

- ***Validação Cruzada com K Conjuntos (K-Fold Cross Validation)***: divide aleatoriamente o conjunto de dados com *N* elementos em *K* subconjuntos disjuntos (*folds*), com aproximadamente o mesmo número de elementos (*N/K*). Cada um dos *K* subconjuntos é utilizado como conjunto de teste e os (*K-1*) demais subconjuntos são reunidos em um conjunto de treinamento. O processo é repetido *K* vezes, sendo gerados e avaliados *K* modelos de conhecimento.

  Esta abordagem é muito utilizada para avaliar a tecnologia utilizada na formulação do algoritmo de Mineração de Dados, e quando a construção de um modelo de conhecimento para uso posterior não seja prioridade.

- ***Validação Cruzada com K Conjuntos Estratificada (Stratified K-Fold Cross Validation)***: aplicável em problemas de classificação, este método é similar à *Validação Cruzada com K Conjuntos*, sendo que ao gerar os subconjuntos mutuamente exclusivos, a proporção de exemplos em cada uma das classes é considerada durante a amostragem.

  Exemplo: se o conjunto de dados original possui duas classes com distribuição de 20% e 80%, cada subconjunto também deverá conter aproximadamente esta mesma proporção de classes.

- ***Leave-One-Out***: este método é um caso particular da *Validação Cruzada com K Conjuntos*, em que cada um dos *K* subconjuntos possui um único registro. É computacionalmente dispendioso e freqüentemente usado em pequenas amostras.

- ***Bootstrap***: O conjunto de treinamento é gerado a partir de *N* sorteios aleatórios e com reposição a partir do conjunto de dados original (contendo *N* registros).

  O conjunto de teste é composto pelos registros não sorteados do conjunto original para o conjunto de treinamento.

Este método consiste em gerar os conjuntos, abstrair e avaliar o modelo de conhecimento um número repetido de vezes, a fim de estimar uma média de desempenho do algoritmo de Mineração de Dados.

2.4.10 Redução de Atributos

A redução de atributos é a escolha de quais tuplas de um conjunto de dados deverão ser efetivamente consideradas na análise. Ela pode ocorrer em dois momentos:

1) Extração dos dados de diversas fontes e carga no conjunto de dados a ser analisado;
2) Escolha dentre os dados de um conjunto de dados, quais deverão ser efetivamente considerados na análise.

Na etapa de descoberta de conhecimento em bases de dados, a segunda opção é a mais freqüente.

2.4.11 Seleção de Atributos

Por ser uma das tarefas no estudo realizado por esta tese, dedicaremos a próxima seção (2.5) para abordar a seleção de atributos na etapa de pré-processamento de dados.

2.5 Seleção de Variáveis

2.5.1 Introdução

A seleção de dados é uma tarefa de pré-processamento muito importante para a etapa de mineração de dados, como por exemplo na geração de regras de associação, já que, com um menor número de atributos, temos condições de gerar de forma menos complexa regras mais simples. A consideração dos atributos mais relevantes reduz o tempo de produção das regras, além de aumentar o seu suporte e confiabilidade (FREITAS, 2002).

Sendo ***S*** um conjunto de dados com atributos $A_1, A_2, A_3, \ldots, A_n$, o problema da **redução de dados vertical** consiste em **identificar** qual das 2^n combinações desses **atributos** deve ser considerada no processo de descoberta de conhecimento. Ela tem como objetivo encontrar um conjunto mínimo de atributos de tal forma que a informação original correspondente à totalidade dos atributos seja ao máximo preservada. Quanto maior o valor de n, maior o desafio na escolha dos atributos.

A seleção pode ocorrer em dois momentos. O primeiro é o da carga da base de dados, quando seus dados provêm de diversas fontes diferentes. Neste momento, é importante que uma adequada seleção de atributos seja feita nas diversas tabelas, para evitar redundâncias ou inconsistências. O segundo momento onde esta seleção é importante é na etapa imediatamente anterior ao processo de mineração de dados. Alguns atributos de uma tabela podem pouco ou nada contribuir para o processo de inferência de dados, ou por não trazer consigo uma semântica na regra de negócio (representantes clássicos desta categoria são as chaves primárias), ou por possuírem uma forte restrição de unicidade (como as chaves candidatas, que, apesar de carregarem um significado na lógica do negócio, não agregam nenhum valor ao processo de descoberta de conhecimento), ou mesmo por campos de difícil tratamento, tais como os campos texto (campos do tipo "observação" são típicos representantes desta categoria).

Um conjunto de atributos bem selecionado pode conduzir a modelos de conhecimento mais concisos e com maior precisão. Além disso, a eliminação de um atributo é muito mais significativa em termos de redução do tamanho de um conjunto de dados do que a exclusão de um registro.

Se o método de seleção for rápido, o tempo de processamento necessário para utilizá-lo e, em seguida, aplicar o algoritmo de mineração de dados em um subconjunto dos atributos, pode ser inferior ao tempo de processamento para aplicar o algoritmo de mineração sobre todo o conjunto de atributos.

2.5.2 Abordagens para a seleção de dados

FREITAS (2002) define duas diferentes abordagens para a seleção de dados, chamadas originalmente *filter* e *wrapper*, e traduzidas respectivamente em GOLDSCHMIDT e PASSOS (2005) para **abordagem independente de modelo** e **abordagem dependente de modelo**. A diferença entre estes dois métodos reside no fato da experimentação da adequação dos atributos selecionados para a etapa de mineração de dados. No primeiro método (*filter*), os atributos são selecionados por algum critério, e utilizados na etapa de mineração de dados, sem levar em consideração o algoritmo de classificação que será aplicado aos atributos selecionados. Já no segundo método, um subconjunto de atributos é analisado por um algoritmo de classificação, que avalia o desempenho deste algoritmo com a seleção feita. Se um critério de desempenho mínimo não for atendido, um novo subconjunto de atributos é gerado, e nova avaliação

é feita. Este processo iterativo se encerra quando o critério é atendido, e o último subconjunto de atributos gerados é a saída do método. A figura 2.6 esquematiza graficamente as duas abordagens.

A abordagem *wrapper* oferece a clara vantagem de gerar um subconjunto de atributos que podem aumentar significativamente a precisão dos algoritmos de mineração de dados a serem executados sobre o conjunto de dados. Todavia, esta abordagem apresenta duas significativas desvantagens: a de ser muito mais lenta do que o método *filter*, já que a busca pelo melhor subconjunto de atributos é um processo que demanda muito tempo de processamento. Além disso, a melhor configuração de atributos para um dado algoritmo de classificação pode não ser tão boa para um outro classificador, o que faz com que a seleção de atributos feita pelo modelo *wrapper* seja dependente do algoritmo utilizado.

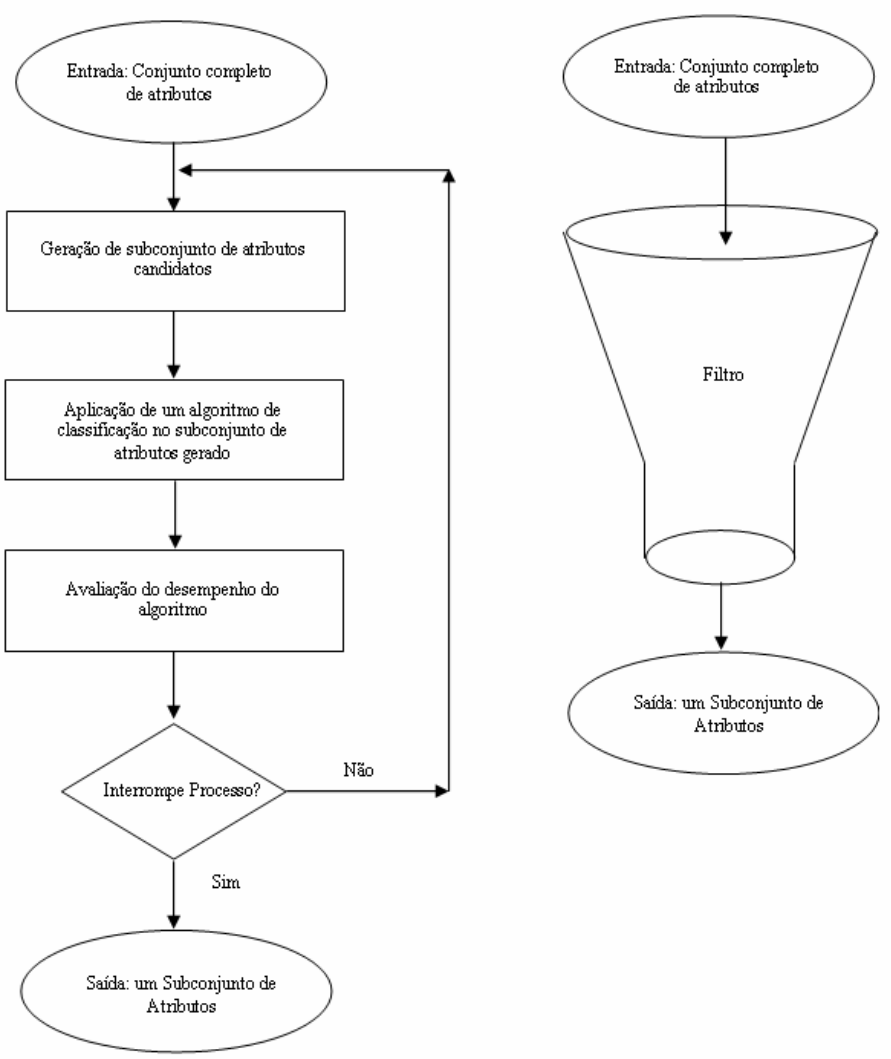


Figura 2.6 As abordagens filter *e* wrapper *para a seleção de atributos.*

Podemos também adotar algumas estratégias para a escolha de atributos:

1) **Seleção Seqüencial para Frente** (***Forward Selection***):

   Esta opção começa o processo com um subconjunto vazio de atributos candidatos. A seguir, cada atributo é adicionado ao subconjunto de candidatos, que é avaliado segundo medida de qualidade. Ao final de cada iteração, é incluído no subconjunto de atributos candidatos aquele atributo que tenha maximizado a medida de qualidade considerada.

2) **Seleção Seqüencial para Trás** (***Backward Selection***)

   Técnica oposta da anterior. Aqui, o conjunto de dados começa completo, e cada atributo é retirado a cada iteração, avaliando-o de alguma forma a sua qualidade. Ao final de cada passo, exclui-se do subconjunto de candidatos o atributo que tenha minimizado a medida de qualidade.

3) **Seleção Híbrida** (***Mixed Selection***)

   Neste caso, as técnicas anteriores são combinadas. A cada iteração, o algoritmo seleciona o melhor atributo (incluindo-o no subconjunto de atributos candidatos) e remove o pior atributo dentre os remanescentes do conjunto de atributos.

2.5.3 Métodos de Seleção de Atributos

Diversas técnicas podem ser encontradas na literatura como possíveis soluções para a prévia seleção de atributos – no contexto da etapa de pré-processamento – antes da efetiva mineração de dados.

HAN e KAMBER (2005) mencionam uma técnica de seleção de atributos chamada *discrete wavelet transform (DWT)*, baseada no processamento linear de sinais, que, quando aplicada a um vetor *D*, transforma-o em um novo vetor *D'*, de mesma dimensão, com coeficientes *wavelet*[2]. A redução da dimensão do vetor é obtida pela truncagem do conjunto *D'* modificado, considerando-se apenas os coeficientes maiores do que um valor especificado como parâmetro.

[2] Mantivemos o termo, para evitar erros de tradução.

Nossa opção para a seleção de atributos foi a de uso de uma técnica estatística bastante utilizada, chamada *Análise de Componentes Principais*, e que detalhamos na próxima seção (Seção 2.5.4).

2.5.4 Seleção de Atributos com Análise de Componentes Principais

2.5.4.1 Introdução

Neste trabalho, dedicamos uma especial atenção à seleção de atributos feita com o auxílio de uma técnica estatística bastante utilizada, a **Análise de Componentes Principais**, já que ela foi a escolhida para realizar a seleção de atributos do nosso sistema de imputação de dados ausentes. Sua escolha se deu pela estabilidade e praticidade no uso do método.

2.5.4.2 O que é a Análise de Componentes Principais

A Análise de Componentes Principais (PCA – *Principal Component Analysis*) (SHLEN, 2005, SMITH, 2002) é uma técnica estatística amplamente utilizada em várias áreas do conhecimento, tais como reconhecimento e compressão de imagens, e que reflete a aplicação direta dos conceitos de sistemas de base em Álgebra Linear. Seu objetivo é o de identificar, em um conjunto de dados com m atributos, um subconjunto de p atributos que sejam os reais representantes das características do referido conjunto de dados. Ao reduzir a dimensão do problema de n para p, $p < n$, buscamos encontrar mais facilmente a sua solução. Utilizaremos, ao longo desta seção, o acrônimo PCA para nos referirmos à Análise de Componentes Principais.

SHLEN (2005) exemplifica de forma muito bem sucedida a idéia da análise de componentes principais. Suponha que queiramos registrar, ao longo do tempo, as coordenadas de uma bola presa a uma mola. Considerando a base canônica, apenas uma dimensão é necessária para observar o movimento descrito (vide figura 2.7). Todavia, se decidíssemos mudar o sistema de coordenadas, colocando em três pontos distintos câmeras que registrassem o fenômeno físico, observaremos, para cada câmera, mudanças de posição da esfera presa à mola em duas dimensões, e não mais em uma.

Tomando por base a figura 2.8, podemos observar um conjunto de pontos visivelmente relacionados, representados na base canônica xy. Intuitivamente, é possível constatar que, à medida que o valor de x cresce, o de y também cresce. Isto pode indicar que, se nos fosse possível obter um novo sistema de bases que indicasse quais os sentidos das variações dos dados, poderíamos descobrir quais são as componentes das

coordenadas mais marcantes na caracterização destes dados. Isto poderia, inclusive, reduzir a dimensão do problema. Sendo assim, o objetivo da técnica PCA é descobrir quais seriam estas dimensões, que revelariam que novo sistema de coordenadas indicaria quais as dimensões acontecem a variação máxima dos dados.

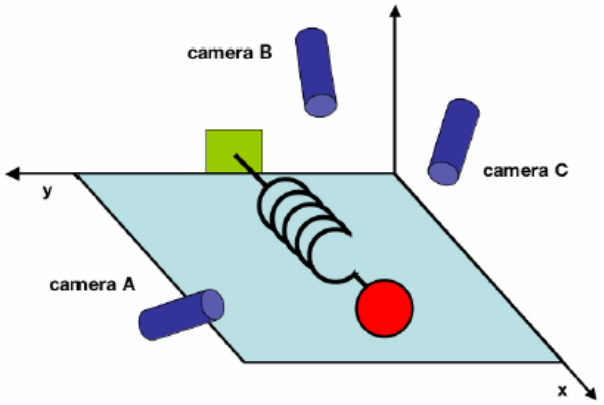
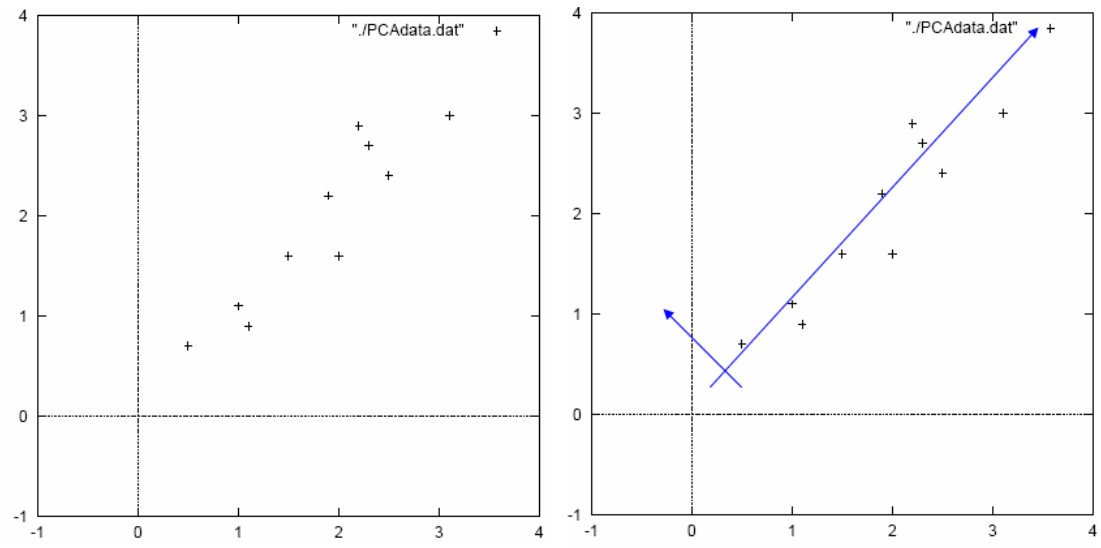


Figura 2.7 O movimento de um corpo preso a uma mola é registrado por três câmeras, produzindo diferentes coordenadas. Fonte (SHLENS, 2005)

Para isso, a técnica utiliza um conjunto de conceitos estatísticos, abordados a seguir. Considere que $X = (X_1, X_2, \ldots, X_n)$ e $Y = (Y_1, Y_2, \ldots, Y_n)$ são vetores do espaço de dimensionalidade n, representados na base canônica.



(a) (b)

Figura 2.8 O gráfico (a) mostra um conjunto de pontos em um gráfico que usa a base canônica de duas dimensões. Se obtivermos uma outra base, como na figura (b), poderemos observar que o novo sistema de coordenadas x'y' mostra uma variação bem mais significativa no novo eixo das abscissas. Modificado de SMITH (2002)

1) **Média Amostral**: o somatório das diversas componentes da amostra, dividida pelo número de observações:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

2) **Desvio Padrão**: medida que indica o quão esparsos (afastados, espalhados) da média os dados da amostra encontram-se:

$$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{(n-1)}}$$

3) **Variância Amostral**: medida unidimensional que indica como os dados da amostra variam, relativos à própria amostra. Trata-se de outra medida de dispersão dos dados, intimamente relacionada com o desvio padrão:

$$s^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{(n-1)}$$

SMITH (2002) comenta o fato de o denominador da variância e do desvio padrão ser *n-1*, e não *n*. A diferença tem origem na teoria estatística, e basicamente reporta à diferença entre amostra e população. A amostra é, na verdade, um subconjunto observado de valores da população, e que não reflete todas as suas características. Esta é a razão pela qual o denominador é dividido por *n-1* neste caso. A redução de uma dimensão simboliza que nem todos os dados da população foram considerados no cálculo da medida. Se tivéssemos todos os valores da população registrados, aí sim o denominador poderia ser substituído por *n*

4) **Covariância**: indica se existe alguma relação na variação dos dados de duas amostras de mesma dimensão:

$$cov(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Esta relação é extremamente importante para o algoritmo PCA, já que esta variação relacionada será a base para a identificação das componentes principais.

Se observarmos com atenção, veremos que a variância amostral é um caso particular de covariância:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

Quando as variáveis são independentes, a sua covariância é zero, já que o valor esperado do produto das variáveis é igual ao valor esperado de cada uma delas:

$$E(X \cdot Y) = E(X) \cdot E(Y)$$

5) **Matriz de Covariâncias**: Matriz simétrica, que indica todas as combinações de covariâncias entre duas diferentes medidas de uma amostra:

$$C^{n \times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$$

Por exemplo, em uma amostra com três tipos de medidas (*x*, *y* e *z*), a matriz de covariância seria:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

Conforme já citamos, a matriz é simétrica:

$$cov(x,y) = cov(y,x)$$

já que a forma com que *y* varia em função de *x* é a mesma que *x* varia em função de *y*. Além disso, por o cálculo da covariância envolver cálculo de quadrados, seu valor nunca é negativo ($cov(x,y) \geq 0$).

SHLENS (2005) cita que estas medidas estatísticas só são suficientes se considerarmos uma amostra com distribuição exponencial. Além disso, este trabalho cita conceitos importantes, utilizados como base do algoritmo de análise de componentes principais:

a) **Ruído e Rotação**: qualquer medida possui, com uma maior ou menor freqüência, uma taxa de ruído associada. A medida de erro está relacionada aos valores observados do fenômeno em questão. Assim, SHLENS (2005) apresenta uma medida comum chamada **taxa de variâncias** (SNR – *sign-to-noise ratio*), expressa por:

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$

onde σ^2_X é a variância amostral de *X*.

Valores altos de SNR (SNR >> 1) indicam dados precisos, com nenhum ou quase nenhum ruído. Já um baixo valor de SNR revela que os dados estão extremamente comprometidos.

Dependendo da posição do sistema de coordenadas utilizado, uma determinada medida pode ter mais ou menos ruído. Assim, um dos objetivos do algoritmo PCA é o de encontrar um sistema de coordenadas que reduza o ruído, fazendo com que o índice SNR seja máximo.

b) **Redundância**: A escolha das componentes de medida de observação de um determinado evento nem sempre é feita de forma adequada. No exemplo de SHLENS (2005), vemos que a escolha da posição das câmeras produz um conjunto de medidas observadas que, a princípio, são desnecessárias. Um movimento que acontece em apenas um eixo gera medidas em duas dimensões, devido à forma com que é observado. Assim, um outro claro objetivo da técnica PCA é o de reduzir, senão eliminar, as eventuais redundâncias nos registros das amostras, buscando reduzir a dimensionalidade do problema.

A covariância entre dois vetores x_i e x_j pode ser reescrita em forma matricial (considerando que a média de cada atributo já foi subtraída de seus valores originais):

$$\sigma^2_{x_i x_j} = \frac{1}{n-1} x_i {x_j}^T$$

Podemos reescrever a matriz de observações de valores como:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix}_{mxn}$$

e sua matriz de covariância relacionada como:

$$C_X \equiv \frac{1}{n-1} XX^T$$

A matriz C_X possui as seguintes propriedades:

- É simétrica, de ordem *m*
- Os elementos da diagonal representam a variância dos elementos dos atributos originais
- Os elementos que não se encontram na diagonal são as covariâncias entre os diferentes atributos.

Tomando por base os conceitos de ruído, rotação e redundância apresentados anteriormente, podemos dizer que altos valores da diagonal da matriz C_X refletem ampla variação dos valores do atributo, e mínimos valores fora da diagonal de C_X indicam redução da redundância.

O conceito que o algoritmo PCA utiliza durante todo o tempo é a linearidade. Desta maneira, buscamos uma transformação linear da forma:

$$\mathbf{PX} = \mathbf{Y},$$

onde **P** é a matriz que realiza a mudança de base. Queremos então, uma matriz **P** de tal forma que, ao calcularmos a matriz de covariância $\mathbf{C_Y}$, consigamos atender a dois requisitos:

1) a redundância (covariância) deve ser minimizada, ou seja, levada a assumir o valor zero;
2) a diagonal (variância) deve ser maximizada, indicando que queremos os valores a serem obtidos são os que melhor refletem as variações dos dados em um dado eixo (e que representam, consequentemente, as componentes mais importantes).

Desta forma, o objetivo do algoritmo é tornar diagonal a matriz $\mathbf{C_Y}$.

O algoritmo PCA assume que os vetores $\{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_m}\}$, as componentes principais do novo espaço vetorial, são ortonormais, ou seja, sua norma (tamanho) é igual à unidade ($\|p_i\| = 1$), ou seja, a matriz **P** é *ortonormal*. A tarefa agora é tentar

encontrar uma matriz **P**, onde **PX = Y**, tal que a matriz de covariância $\mathbf{C_Y} = \frac{1}{n-1} YY^T$ é diagonalizável.

O trabalho de SHLENS (2005), que serviu de base para esta seção, revela como este objetivo pode ser alcançado. Esta demonstração utiliza as manipulações algébricas descritas a seguir. Reescrevamos a matriz de covariância $\mathbf{C_Y}$ em função da matriz de transformação **P**:

$$\begin{aligned}
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^T \\
&= \frac{1}{n-1}(\mathbf{PX})(\mathbf{PX})^T \\
&= \frac{1}{n-1}\mathbf{PX}\mathbf{X}^T\mathbf{P}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{X}\mathbf{X}^T)\mathbf{P}^T \\
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{PAP}^T
\end{aligned}$$

Note que a nova matriz $\mathbf{A = XX^T}$ é simétrica.

Todavia, uma matriz simétrica é diagonalizável por uma matriz ortogonal de seus autovetores. Logo:

$$\mathbf{A = EDE^T}$$

onde **D** é a matriz diagonal e **E** é uma matriz de autovetores de **A**, organizada em colunas.

A matriz A possui $r \leq m$ autovetores ortonormais, onde r é a ordem da matriz A. Se $r < m$, dizemos que a matriz **A** é degenerada, ou todos os dados ocupam um subespaço de dimensão $r \leq m$. Mantendo a restrição de ortogonalidade, podemos remediar esta situação, selecionando $(m - r)$ vetores ortonormais adicionais para completar a matriz **E**. Estes vetores adicionais não alteram a solução final, porque as variâncias associadas a estes vetores é zero.

A seguir, selecionamos uma matriz **P** onde cada linha $\mathbf{p_i}$ é um autovetor da matriz $\mathbf{XX^T}$. Desta forma, temos que $\mathbf{P \equiv E^T}$. Substituindo esta equação em $\mathbf{A = EDE^T}$, temos $\mathbf{A = P^TDP}$. Todavia, sabe-se que a matriz inversa de uma matriz ortogonal é a sua transposta. Assim, neste caso, $\mathbf{P^{-1} = P^T}$. Então, podemos reavaliar $\mathbf{C_Y}$ da forma que segue:

$$
\begin{aligned}
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{PAP}^T \\
&= \frac{1}{n-1}\mathbf{P}(\mathbf{P}^T\mathbf{DP})\mathbf{P}^T \\
&= \frac{1}{n-1}(\mathbf{PP}^T)\mathbf{D}(\mathbf{PP}^T) \\
&= \frac{1}{n-1}(\mathbf{PP}^{-1})\mathbf{D}(\mathbf{PP}^{-1}) \\
\mathbf{C_Y} &= \frac{1}{n-1}\mathbf{D}
\end{aligned}
$$

que evidentemente diagonaliza a matriz de covariância $\mathbf{C_Y}$.

Então, em resumo:

1) Todo o desenvolvimento acima levou em consideração que conseguiremos encontrar uma matriz **P** ortonormal, de tal forma que **P** leva a matriz **X** em uma outra **Y** (**PX = Y**), cuja matriz de covariância $\mathbf{C_Y}$ é diagonalizável;
2) As componentes principais da matriz X são, na verdade, os autovetores da matriz $\mathbf{XX^T}$, ou as linhas de **P**;
3) O *i*-ésimo valor da matriz $\mathbf{C_Y}$ é a variância de **X** em $\mathbf{p_i}$;

Logo, o algoritmo de Análise de Componentes Principais é o que segue:

1) **Calcule a média $\overline{a_i}$ de cada um dos *m* atributos a_i do conjunto de dados, e subtraia de todos os valores esta média**.
2) **Calcule a matriz de covariância dos dados ajustados**.
3) **Calcule os autovetores e os autovalores da matriz de covariância**: o vetor de autovalores relacionados aos autovetores da matriz de covariância indicam a importância das componentes. Quanto maior o autovalor, mais importância a dimensão (variável) possui na distribuição dos dados.
4) **Forme o vetor de características**: Para reduzir a dimensão dos dados, escolha $p \leq m$ autovetores mais importantes, obtidos através do vetor de autovalores do passo anterior, para formar o **vetor de características C**. Este vetor é construído com os *p* primeiros autovetores, dispostos cada um deles nas colunas desta nova matriz.

$$C = (autovetor_1\ autovetor_2\ \ldots\ autovetor_p)$$

5) **Reconstrua os dados com as dimensões mais importantes**: Calculando a transposta do vetor de características **C**, e a multiplicamos pela matriz transposta dos dados ajustados (**X'**), com o autovetor mais importante na primeira posição. Com isso, temos:

$$X_{modificado} = C^T \times X'^T$$

2.6 Agrupamento de Dados

2.6.1 Introdução

O agrupamento é, sem nenhuma sombra de dúvida, uma das tarefas mais primitivas do ser humano. A busca de semelhanças e diferenças sempre esteve presente nas atividades mais diversas, como por exemplo, a separação de uma classe escolar para realizar diferentes atividades com meninos e meninas, a divisão de atletas em diferentes categorias (infantil, juvenil, júnior, sênior), a divisão dos dormitórios de uma casa entre os pais e os filhos (e quando os filhos têm sexos diferentes, uma nova divisão entre os quartos acontece a partir de certa idade!).

Quando nos referimos às áreas de conhecimento, novamente encontramos de forma marcante a agrupamento. Campos de conhecimento tais como a biologia, psiquiatria, psicologia, arqueologia, geologia, geografia, e marketing beneficiam-se desta consagrada técnica (JAIN *et al*, 1999). Estudos bastante interessantes de aplicação de agrupamento em neurociência computacional são encontrados utilizando técnicas de agrupamento.

E não poderia ser diferente na Computação. A tarefa de agrupamento encontra aplicações diretas (em áreas como o reconhecimento de padrões e a análise de imagens) ou indiretas. Neste último caso, podemos citar como exemplo o trabalho de CARVALHO, FISZMAN e FERREIRA (2001), que desenvolve modelos teóricos de autismo, valendo-se de técnicas de agrupamento. Todavia, o objetivo central de estudo desta técnica nesta tese está concentrado na aplicação do agrupamento no processo de descoberta de conhecimento em bases de dados, mais especificamente na etapa de pré-processamento de dados.

Por definição, **agrupar** significa reunir objetos que possuam as mesmas características. Tendo em vista que não conhecemos que características são essas, o agrupamento toma por base alguma medida de similaridade. BEZERRA (2006) define ainda o conceito de **modelo de agrupamento**, um conjunto de grupos gerados a partir

dos objetos de uma coleção. Além disso, BEZERRA (2006) complementa a definição mencionando o fato de que a agrupamento tem como objetivo maximizar uma função objetivo, implícita ou explícita, inerente aos dados.

Sendo assim, a finalidade da tarefa de agrupamento é maximizar a similaridade dos objetos de um grupo, e minimizá-la para objetos de grupos distintos. O sucesso de um algoritmo de agrupamento está na escolha de uma boa medida de similaridade.

2.6.2 Componentes da tarefa de agrupamento

Segundo JAIN, MURTY e FLYNN (1999), os componentes da tarefa de agrupamento são:

1) Representação dos objetos da coleção (contendo opcionalmente componentes de extração ou seleção)
2) Definição de uma medida de similaridade apropriada ao domínio dos dados
3) O algoritmo de agrupamento
4) Um mecanismo de abstração do conjunto de dados (opcional)
5) Avaliação dos resultados (opcional)

2.6.2.1 Medidas de similaridade

As medidas de similaridade são positivas, respeitam a desigualdade triangular – $d(a,c) \leq d(a,b) + d(b,c)$ – e se dividem em duas classes: similaridade envolvendo atributos **numéricos** e **categóricos**.

GOWDA e DIDAY (1992) classificam os atributos nos seguintes tipos:

1) Atributos Qualitativos
   a. Ordinais
   b. Nominais
2) Atributos Quantitativos
   a. Valores contínuos
   b. Valores discretos
   c. Intervalo de valores

Atributos **ordinais** trazem consigo o conceito de ordem na especificação do domínio do atributo. Como exemplo, considere uma pesquisa onde as respostas sejam todas obtidas em alguma escala de intensidade (1-fraco, 2-regular, 3-bom e 4-excelente). Este tipo de dado é comumente chamado de escala de temperatura, ou *likert data* (JÖNSSON, WOHLIN, 2006, JÖNSSON, WOHLIN, 2006). Já atributos **nominais** representam um conjunto de valores possíveis que o atributo pode assumir (domínio do

atributo), mas que não abrigam nenhuma noção de precedência. Por exemplo, considere o atributo *função* em uma tabela de alocação de projetos. Sabendo-se que o domínio deste atributo possui os valores *gerencial* ou *operacional*, podemos tranquilamente dizer que não observamos nenhuma conotação que envolva ordem neste conjunto.

Já atributos quantitativos são aqueles que podem ser medidos em uma escala. O que os diferencia é a natureza do possível conjunto de valores desta escala. Se sua unidade de medida sempre comportar um terceiro valor entre dois outros (por exemplo, os números reais), dizemos que os atributos são **contínuos**. Já se os possíveis valores não admitirem a propriedade acima, dizemos que eles são **discretos**. Nos dois casos, não existem limites inferior ou superior. Quando eles existirem, dizemos que estamos tratando de um **intervalo de valores**.

As métricas que envolvem similaridade de atributos categóricos normalmente estão ligadas às diferenças nas características dos valores do domínio deste tipo de atributo. Medidas comuns são a *Jaccard*, *overlap* (sobreposição), *co-seno* e *dice*. A tabela 2.3 mostra a forma pela qual estas medidas são calculadas. Considere V_1 e V_2 conjuntos de valores de características de dois objetos, e *Card(V)* a cardinalidade do conjunto *V*.

Tabela 2. 3 Opções de cálculo de similaridade de atributos categóricos

Jaccard	$Card(V_1 \cap V_2)/Card(V_1 \cup V_2)$
Sobreposição (Overlap)	$Card(V_1 \cap V_2)/\min(Card(V_1), Card(V_2))$
Co-seno	$Card(V_1 \cap V_2)/\sqrt{Card(V_1) \times Card(V_1)}$
Dice	$(2 \times Card(V_1 \cap V_2))/(Card(V_1) + Card(V_2))$

Atributos numéricos, por sua vez, também apresentam um conjunto de possibilidades de métricas aplicáveis a algoritmos de agrupamento. Todas as métricas de distâncias numéricas interpretam os objetos do conjunto original como vetores em um espaço *n* dimensional. A mais comumente usada é a **distância Euclidiana**:

$$d(a,b) = \sqrt[2]{\sum_{i=1}^{n}(a_i - b_i)^2} = \|a-b\|_2$$

um caso particular da **distância Minkowski**:

$$d_{Mi}(a,b) = (\sum_{i=1}^{n}|a_i - b_i|^p)^{\frac{1}{p}} = \|a-b\|_p$$

Outra métrica derivada da distância Minkowski é a **distância Manhattan**:

$$d(a,b) = \sum_{i=1}^{n} |a_i - b_i| = \|a - b\|_1$$

Estas distâncias, apesar de populares, apresentam o problema de se tornarem desproporcionais se algumas das componentes dos objetos possuírem valores muito maiores das demais. Uma possível solução para isto é a normalização dos valores dos atributos, onde cada um deles é dividido ou pelo maior valor do domínio, ou o maior valor encontrado naquele atributo em um dos objetos. Estas métricas também podem ser prejudicadas quando existe uma correlação linear forte entre os atributos dos objetos. Neste caso, podemos usar a **distância Malahanobis**:

$$d_{Ma}(a,b) = \sqrt{(a-b)^T Cov^{-1}(a-b)} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij}(a_i - b_i)(a_i - b_i)}$$

onde Cov^{-1} é a matriz de covariância entre os vetores *a* e *b*.

2.6.2.2 Classificação das Técnicas de Agrupamento

Na literatura, encontramos diferentes taxonomias para os algoritmos de agrupamento. Abordamos, nesta seção, a proposta por BEZERRA (2006), que divide os critérios de classificação de técnicas de agrupamento em três tipos:

1) A **estrutura do agrupamento**: a forma pela qual um algoritmo de agrupamento processa a coleção de objetos, para identificar os seus grupos associados.
2) A **natureza da pertinência dos objetos em grupos**: revela se os objetos de uma coleção podem pertencer a mais de um grupo simultaneamente; e
3) A **estratégia de agrupamento**: a forma pela qual o algoritmo de agrupamento interpreta os objetos da coleção para formar grupos;

2.6.2.2.1 Estrutura do Agrupamento

Quando se considera a estrutura do agrupamento resultante da aplicação de um algoritmo de agrupamento qualquer, existem os seguintes métodos de agrupamento: **partitivos** e **hierárquicos**. Vamos agora descrever esses métodos e, em cada um deles, dar exemplos de algoritmos representativos.

- *Métodos Partitivos*: um algoritmo partitivo gera *k* grupos mutuamente exclusivos de objetos, onde geralmente *k* é parâmetro do algoritmo. Um elemento marcante nesta classe de algoritmos é a idéia de centróide μ_i, um vetor de objetos de mesma dimensionalidade dos objetos da coleção, que atua como um representante daquele

grupo. Assim, cabe ao algoritmo partitivo otimizar uma função-objetivo, que envolve cálculos de distância entre os objetos e seus respectivos centros. Com o conceito de centro de um grupo, comumente estes métodos partitivos consideram que a coleção a ser agrupada provém de uma distribuição de probabilidades gaussiana subjacente.

Neste trabalho, utilizamos como técnica de agrupamento o algoritmo partitivo dos **k-centróides**, mais conhecido como *k-means* (MCQUEEN, 1967). Trata-se de um algoritmo clássico, amplamente utilizado, e que é abordado com mais detalhes na próxima seção.

- *Métodos Hierárquicos*: caracterizam-se por realizarem o agrupamento dos dados em níveis, produzindo ao final uma estrutura semelhante a um dendograma. A principal idéia desta classe de algoritmos é o cálculo de similaridade das distâncias entre todos os objetos da coleção inicial de dados, calculadas em pares. O par de objetos que possuir a maior medida de similaridade forma um grupo. A partir deste momento, este grupo é encarado como um novo objeto da nova coleção de dados, e o processo se repete: calculam-se as distâncias de todos os objetos da coleção aos pares. O par que apresentar a menor distância forma o novo grupo. Este processo se encerra quando apenas um grupo existir.

  Para exemplificar o exposto acima, considere o exemplo ilustrado na figura 2.9, obtido em JAIN, MURTY e FLYNN (1999), onde a coleção original de dados é **X = {A,B,C,D,E,F,G}**. Na primeira rodada de agrupamento, após calcular todas as distâncias entre os elementos, encontrou-se uma maior similaridade entre os objetos *B* e *C*. A partir deste momento, eles passam a não mais existir como elementos individuais para as demais etapas do algoritmo, e passam a ser encarados como um novo elemento (grupo). Na segunda rodada, o par *{D,E}* possui maior similaridade, e forma um novo grupo. Observe a quarta rodada: o par *{A,(B,C)}* possui similaridade maior do que todas as demais combinações dos elementos existentes do conjunto *{A,(B,C),(D,E),(F,G)}*.

  JAIN, MURTY e FLYNN (1999) especificam que a forma com que o algoritmo de agrupamento hierárquico calcula a similaridade entre os objetos a cada rodada pode ser implementada de três possíveis formas. A primeira, chamada *single-link*, é uma abordagem otimista. Ao calcular a similaridade entre dois padrões (objetos, sejam eles simples ou grupos formados em rodadas anteriores), considera-se o objeto do padrão com a menor distância entre os demais dos outros padrões. Já a

segunda opção, chamada *complete-link*, possui uma abordagem pessimista: a maior distância, ao invés da menor, é considerada para o cálculo da distância entre os padrões. A terceira opção apresenta uma proposta híbrida, onde uma média (centróide) do padrão é gerado para o cálculo das distâncias.

2.6.2.2.2 Natureza da Pertinência de Objetos

Algoritmos de agrupamento podem apresentar ou não interseção nas partições geradas. Note que o conceito aqui é diferente do apresentado na seção anterior. Algoritmos hierárquicos geram grupos que estão contidos um nos outros. Aqui, estamos nos referindo a conjuntos onde existe um grau (percentual) de pertinência de um elemento em um ou mais grupos. Algoritmos que trabalham gerando grupos com estas características são chamados **difusos** (*fuzzy clustering*). Já algoritmos que geram partições disjuntas são chamados **estritos** (*hard clustering*).

Formalmente, dizemos que quaisquer partições G_i e G_j resultantes do agrupamento de um conjunto de objetos X foram produzidas por um algoritmo de agrupamento difuso se $G_i \cap G_j \neq \varnothing$, e estrito se $G_i \cap G_j = \varnothing$.

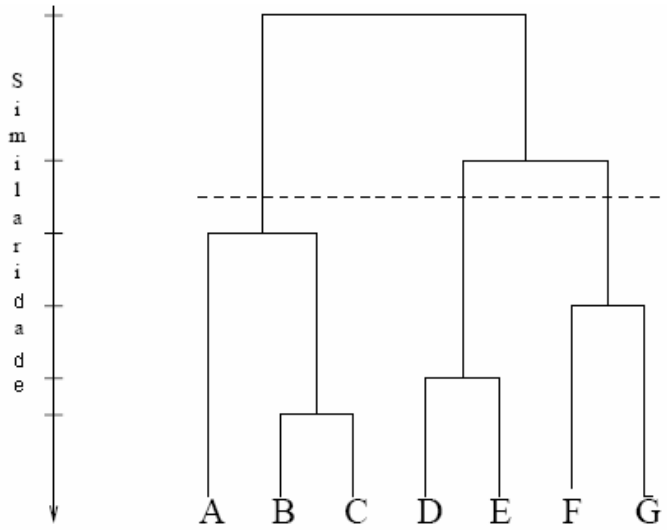


Figura 2.9 Exemplo de agrupamento hierárquico de uma coleção de objetos. Fonte: (*JAIN et al, 1999*)

2.6.2.2.3 Estratégia de Agrupamento

A estratégia de um algoritmo de agrupamento leva em consideração o seguinte pressuposto: os elementos de um grupo respeitam algum tipo de distribuição

probabilística, ou uma mistura de distribuições. Um exemplo desta classe de algoritmos é o **K-means**, que assume como premissa uma distribuição normal dos objetos da coleção de entrada. Outro exemplo é o **algoritmo EM** (*Expectation-Maximization*) (DEMPSTER, LAIRD, RUBIN, 1977), bastante usado também na tarefa de complementação de dados. Esta técnica tenta estimar os coeficientes da função densidade de distribuição probabilística, e pode também ser usada para agrupamento de dados. O algoritmo *K-means* é considerado um caso particular do algoritmo EM, para distribuição normal dos dados.

2.6.2.3 O algoritmo dos *K*-Centróides (*K-Means*)

O algoritmo partitivo dos ***K*-Centróides**, mais conhecido como *K-Means* (MCQUEEN, 1967), é um dos mais antigos e importantes algoritmos de agrupamento disponíveis na literatura. Seu uso é amplo até os dias atuais, mesmo depois de sua publicação há quarenta anos. A sua simplicidade e alto desempenho são os principais motivos para o seu amplo uso.

Sua versão original conta com objetos de coleções que possuam componentes numéricas.

Entrada: Número *K* de grupos e coleção *C* de objetos

Saída: *N* grupos com os objetos da coleção original *C* associados a cada um dos centróides

Algoritmo:

Gere *K* centróides.

Faça

Associe cada objeto da coleção a um centróide.

Calcule um novo centróide para cada grupo em função dos objetos alocados.

Até que os objetos não mudem de grupo, ou até que um número máximo de iterações tenha sido alcançado.

O *K-Means* é um algoritmo clássico de agrupamento, e, na maioria das vezes, adotado como primeira opção quando existe uma necessidade de agrupamento de dados. Esta foi também a nossa opção nesta tese, tendo em vista que o seu desempenho frente às demais opções é bastante interessante – sua complexidade é *O(nK)*, onde *n* é a cardinalidade do conjunto de dados originais, e *K* é a quantidade de grupos. Além disso,

ele é de simples implementação. E por ser um algoritmo bastante estudado ao longo destes anos, possui muitas variantes e resultados interessantes, em vários de seus passos.

Várias opções são encontradas na literatura sobre a geração dos *K* primeiros centróides. Em sua forma clássica, esta escolha é feita de forma aleatória. Porém, encontramos também implementações onde os *K* primeiros objetos do conjunto de dados originais são escolhidos como os primeiros centróides (TEKNOMO, 2007). Todavia, outras soluções de determinação dos centróides iniciais poderiam ser usadas, tais como a aleatória ou a técnica *farthest-first* (DASGUPTA, 2002). Esta consiste em encontrar *K* centróides, de forma que eles estão o mais distante uns dos outros. Nesta técnica, inicialmente escolhe-se um primeiro centróide aleatoriamente. O próximo centróide escolhido é o que é mais distante deste primeiro. O processo se repete até que os *K* centróides tenham sido escolhidos. Esta técnica é usada também para construir grupos hierárquicos, com bom desempenho em cada nível da hierarquia (DASGUPTA, 2002).

Dois conhecidos algoritmos derivados do *K-Means* são as técnicas de agrupamento *K-Medoids* (JAIN, DUBES, 1988) e *K-Modes* (HUANG, 1997). O primeiro diferencia-se do clássico algoritmo dos K-centróides por considerar que os centros dos grupos formados são elementos do próprio grupo, e não elementos calculados. Basicamente, a diferença reside no fato de que, quando um centróide é calculado e os objetos com menor distância para ele são associados, o que tiver a menor distância de todas é nomeada como centróide, e o algoritmo segue. Já o segundo algoritmo, o *K-modes* é a versão do *K-means* para dados categóricos.

A associação dos objetos aos centróides se dá pelo cálculo da distância de cada objeto a todos os centróides gerados, e vinculando o objeto ao centróide que produziu a menor distância de todas as geradas. A versão clássica do K-*Means* utiliza a distância Euclidiana como métrica.

2.7 Imputação

2.7.1 Objetivos

A tarefa de imputação é a responsável por recuperar valores ausentes nas tuplas de um conjunto de dados. Etapa vital no processo de complementação de dados, ela pode ser realizada valendo-se de diversas técnicas, dependendo primordialmente da natureza do atributo.

Nesta tese, temos como alvo de estudos dados numéricos. Por esta razão, utilizamos três técnicas de imputação: a média (com variantes), o algoritmo dos *k*-vizinhos mais próximos, e as redes neuronais artificiais que utilizam retropropagação de erros (*back propagation*). Por esta razão, detalhamos suas implementações nas próximas seções. Cabe registrar que estes não são as únicas técnicas de imputação que podem ser usadas. Várias outras, consagradas, poderiam ter sido adotadas, tais como a regressão linear ou não linear. Todavia, as duas primeiras técnicas (média e algoritmo *k*-NN) são as mais utilizadas na literatura relacionada com a complementação de dados, e, como nosso objetivo de estudo é o de analisar estratégias de imputação de dados, optamos por utilizá-las, para termos base de comparação com os estudos existentes na literatura (e expostos no Capítulo 3). As redes neuronais *back propagation* também foram escolhidas, pois desejávamos trabalhar também com uma técnica de Inteligência Computacional de uso difundido. Desta forma, o seu uso também foi adotado, e os seus resultados comparados com os demais gerados.

2.7.2 Imputação por média ou moda

A mais simples e também mais comum forma de imputação de dados consiste ou no cálculo da **média** (para atributos contínuos), ou da **moda** (para atributos categóricos) – onde o valor que mais aparece nas demais tuplas da relação é assumido para substituir um valor ausente.

A grande vantagem na imputação desta abordagem é o seu desempenho. Grandes bases de dados podem ter os seus valores ausentes recuperados pela substituição pela média ou moda em poucos segundos. Todavia, uma grande desvantagem surge: os dados tornam-se enviesados, já que, em primeiro lugar, refletem a tendência dos valores preenchidos, que podem representam uma ou mais subclasses do conjunto de dados. Além disso, substituindo todos os valores ausentes pela média ou pela moda fazem com que a diversidade dos dados diminua consideravelmente, reduzindo o desvio-padrão da amostra.

Algumas alternativas que podem ser adotadas para evitar que a média não distorça excessivamente os dados são a aplicação de uma perturbação aleatória ao valor da média. Esta perturbação pode adicionar ou remover um valor Δm da média, tentando reduzir os efeitos descritos anteriormente.

Outra possibilidade é a de uso da **média balanceada** (MAGNANI, MONTESI, 2004), uma média de tendência central que é menos sensível à variação de valores extremos. MAGNANI e MONTESI (2004) sugerem uma média com uma tolerância *t*, fornecida pelo usuário, e com um fator de corte de 100(1-*t*) %. Se, por exemplo, uma tolerância t = 0,03 for utilizada, valores menores e maiores do que 1,5% são descartados, e a seguir a média aritmética é calculada.

TEKNOMO (2007) aborda ainda algumas outras médias. Por exemplo, a **média geométrica** é utilizada quando todos os dados da amostra são positivos. Ela é calculada por:

$$\bar{x} = \sqrt[n]{x_1 * x_2 * \ldots * x_n} = \left(\prod_{i=1}^{n} x_i \right)^{1/n}$$

A média aritmética ponderada considera pesos w_i para cada componente x_i da amostra em questão:

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i * x_i}{\sum_{i=1}^{n} w_i}$$

A média harmônica H representa a capacidade média individual da ação de *n* elementos que estão agindo harmonicamente. Ou seja, H representa a capacidade de um elemento que é capaz de substituir cada um dos *n* agentes quando atuando em conjunto:

$$H = \frac{1}{\frac{1}{n}\left(\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n} \right)} = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right)^{-1}$$

TEKNOMO (2007) cita ainda as médias **Minkowski**, **Lehmer** e a **Generalizada Phillips**, que são casos genéricos combinados das médias acima expostas.

2.7.3 Imputação com o Algoritmo dos k-Vizinhos Mais Próximos

2.7.3.1 O que é o algoritmo

O algoritmo dos *k*-vizinhos mais próximos (*k-Nearest Neighbors*) (MITCHELL, 1997, AHA, KIBLER, ALBERT, 1991, DASARATHY, 1990) é uma técnica de aprendizado supervisionado que avalia, através de uma função de similaridade, quais são os *k* objetos mais próximos a um dado objeto (uma tupla de uma tabela, por exemplo) de um conjunto de dados.

2.7.3.2 O algoritmo dos *k*-vizinhos mais próximos como classificador

Freqüentemente o algoritmo *k*-NN é utilizado na tarefa de classificação. De um conjunto de *n* tuplas, imagine que não conhecemos o atributo classificador da tupla *i*. Utilizando-o, podemos saber quem são as *k* tuplas mais próximas da tupla *i*, e, deste subconjunto, verificar qual a classe que mais aparece nas tuplas onde o classificador é presente (moda). A similaridade normalmente é medida através do cálculo da distância Euclidiana entre duas tuplas t_i e t_j:

$$d(t_i, t_j) = \sqrt{\sum_{l=1}^{k} (t_{il} - t_{jl})^2} \ ,$$

onde *k* é o número de colunas da tabela.

O algoritmo *k*-NN para classificação de uma tupla t_i é o que se segue:

Entrada: Um conjunto de dados tabular com atributo classificador, uma tupla t_i sem classificação, e o número *k* de vizinhos da tupla t_i.

Saída: Tupla t_i com o atributo-classe C_i completo

Algoritmo:

Calcule a distância da tupla t_i para todas as demais tuplas t_j, $j \neq i$.

Ordene de forma crescente as tuplas pelas distâncias, e considere apenas as *k* primeiras.

Atribua à tupla t_i a classe mais freqüente das *k* tuplas selecionadas no passo anterior.

Por exemplo, considere um trecho do conjunto de dados *Íris Plants* (NEWMAN *et al*, 1998), utilizada como uma das bases de testes nesta tese:

id	pl	pw	sl	sw	classe
1	7,1	3,0	5,9	2,1	Iris-virginica
2	6,0	2,9	4,5	1,5	Iris-versicolor
3	6,5	3,0	5,8	2,2	Iris-virginica
4	5,5	4,2	1,4	0,2	Iris-setosa
5	6,3	3,3	6,0	2,5	Iris-virginica
6	5,5	3,5	1,3	0,2	Iris-setosa
7	6,6	3,0	4,4	1,4	Iris-versicolor
8	5,7	2,6	3,5	1,0	Iris-versicolor
9	5,0	3,2	1,2	0,2	Iris-setosa
10	6,8	2,8	4,8	1,4	Iris-versicolor
11	5,8	2,7	5,1	1,9	?
12	4,9	3,1	1,5	0,1	Iris-setosa
13	6,3	2,9	5,6	1,8	Iris-virginica
14	5,2	4,1	1,5	0,1	Iris-setosa
15	6,7	3,0	5,0	1,7	Iris-versicolor

Nesta tabela, o atributo *pl* representa o comprimento da pétala da planta (*petal length*), o atributo *pw* a largura da pétala (*petal width*), *sl* representa o comprimento do caule (*sepal length*) e *sw* a largura do caule (*sepal width*).

Como se pode observar, a 11ª tupla não contém o valor da classe (na verdade, a classe original é ***Iris-virginica***). O algoritmo *k*-NN poderia então ser aqui utilizado para a determinação da classe, a partir da determinação do número de tuplas vizinhas a serem selecionadas.

Calculando as distâncias entre a tupla destacada e todas as demais (em todos os casos, a chave primária da tabela não entra no cálculo das distâncias), temos:

$d_{11,1}$	$d_{11,2}$	$d_{11,3}$	$d_{11,4}$	$d_{11,5}$	$d_{11,6}$	$d_{11,7}$	$d_{11,8}$	$d_{11,9}$	$d_{11,10}$	$d_{11,12}$	$d_{11,13}$	$d_{11,14}$	$d_{11,15}$
1,56	0,77	1,07	4,34	1,33	4,24	1,21	1,84	4,35	1,16	4,14	0,74	4,30	0,97

Podemos ver que, se escolhermos o vizinho mais próximo (k = 1), assumiremos para a tupla 11 a classe da tupla 13 (***Iris-virginica***), que é a real classe da tupla 11. Porém, para k = 3, adotaríamos a classe ***Iris-versicolor***, pois existem duas ocorrências suas nas tuplas 2 e 15.

Este é um dos problemas que podem ocorrer com o uso deste algoritmo na tarefa de classificação. Se objetos de classes diferentes apresentarem valores de seus atributos muito próximos, a classificação pode falhar. Todavia, para classes onde haja uma boa distinção entre os campos das tuplas, o algoritmo tem um melhor desempenho. Observe o segundo exemplo:

id	pl	pw	sl	sw	Classe
1	7,1	3,0	5,9	2,1	Iris-virginica
2	6,0	2,9	4,5	1,5	Iris-versicolor
3	6,5	3,0	5,8	2,2	Iris-virginica
4	5,5	4,2	1,4	0,2	?
5	6,3	3,3	6,0	2,5	Iris-virginica
6	5,5	3,5	1,3	0,2	Iris-setosa
7	6,6	3,0	4,4	1,4	Iris-versicolor
8	5,7	2,6	3,5	1,0	Iris-versicolor
9	5,0	3,2	1,2	0,2	Iris-setosa
10	6,8	2,8	4,8	1,4	Iris-versicolor
11	5,8	2,7	5,1	1,9	Iris-virginica
12	4,9	3,1	1,5	0,1	Iris-setosa
13	6,3	2,9	5,6	1,8	Iris-virginica
14	5,2	4,1	1,5	0,1	Iris-setosa
15	6,7	3,0	5,0	1,7	Iris-versicolor

Neste caso, se a classe da tupla 4 não estivesse preenchida (***Iris-setosa***), as distâncias calculadas entre esta tupla e as demais seria:

$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,5}$	$d_{4,6}$	$d_{4,7}$	$d_{4,8}$	$d_{4,9}$	$d_{4,10}$	$d_{4,11}$	$d_{4,12}$	$d_{4,13}$	$d_{4,14}$	$d_{4,15}$
5,27	3,63	5,07	5,28	0,70	3,61	2,76	1,13	4,08	4,34	1,26	4,74	0,34	4,25

E, para k = 1, 2, 3 e 4, a classe escolhida seria ***Iris-setosa***, pois as tuplas 14, 6, 9 e 12 apresentam, respectivamente, as menores distâncias para a tupla 4.

2.7.3.3 A utilização do algoritmo dos *k*-vizinhos mais próximos no processo de imputação

Outra utilização para o algoritmo *k*-NN é aplicá-lo como um regressor de dados contínuos. Podemos ver esta opção como um caso genérico do anterior, pois a classificação feita foi, na verdade, uma imputação do atributo classificador da tupla. Quando tratamos de atributos com valores numéricos, o valor do atributo ausente é obtido calculando-se a média dos k vizinhos da tupla.

Abaixo segue o algoritmo *k*-NN para imputação de uma tupla t_i:

Entrada: Um conjunto de dados tabular com atributo classificador, uma tupla t_i com um atributo t_{ik} ausente, e o número k de vizinhos da tupla t_i.

Saída: Tupla t_i com o atributo t_{ik} prenchido.

Algoritmo:

Calcule a distância da tupla t_i para todas as demais tuplas t_j, $j \neq i$.

Ordene de forma crescente as tuplas pelas distâncias, e considere apenas as k primeiras.

Atribua à tupla t_i a média dos atributos t_{jk} das k tuplas selecionadas no passo anterior.

Valendo-se ainda da tabela utilizada como exemplo na seção anterior, considere que desejamos regredir o atributo *sl* (*sepallength*) da tupla 4:

id	pl	pw	sl	Sw
1	7,1	3,0	5,9	2,1
2	6,0	2,9	4,5	1,5
3	6,5	3,0	5,8	2,2
4	5,5	4,2	?	0,2
5	6,3	3,3	6,0	2,5
6	5,5	3,5	1,3	0,2
7	6,6	3,0	4,4	1,4
8	5,7	2,6	3,5	1,0
9	5,0	3,2	1,2	0,2
10	6,8	2,8	4,8	1,4
11	5,8	2,7	5,1	1,9
12	4,9	3,1	1,5	0,1
13	6,3	2,9	5,6	1,8
14	5,2	4,1	1,5	0,1
15	6,7	3,0	5,0	1,7

Novas distâncias devem ser calculadas, não considerando o atributo *sl*, que, neste caso, é ausente na tupla 4:

$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	$d_{4,5}$	$d_{4,6}$	$d_{4,7}$	$d_{4,8}$	$d_{4,9}$	$d_{4,10}$	$d_{4,11}$	$d_{4,12}$	$d_{4,13}$	$d_{4,14}$	$d_{4,15}$
2,75	1,90	2,53	2,59	0,70	2,02	1,80	1,11	2,25	2,28	1,25	2,21	0,33	2,26

Com $k = 1$, a tupla mais próxima à de identificador 4 é a tupla 14:

id	pl	pw	sl	sw
14	5,2	4,1	1,5	0,1

Assim, o atributo *sl* da tupla 4 assumiria o valor **1,5**, bastante próximo do real (no caso, **1,4**). Porém, com $k = 2$ temos as tuplas 14 e 6 como as mais próximas:

id	pl	pw	sl	sw
6	5,5	3,5	1,3	0,2
14	5,2	4,1	1,5	0,1

O valor a ser complementado para o atributo *sl* da tupla 4 é, neste caso, calculado com a média destes atributos das tuplas 14 e 6:

$$sl_4 = (1,3+1,5)/2 = 1,4$$

Conseguimos, aqui, obter exatamente o valor que foi perdido. Já para $k = 3$, as tuplas mais próximas são 14, 6 e 9:

id	pl	pw	sl	sw
6	5,5	3,5	1,3	0,2
9	5,0	3,2	1,2	0,2
14	5,2	4,1	1,5	0,1

e a média, neste caso, é:

$$sl_4 = (1,3+1,5+1,2)/3 \approx 1,3$$

O que conseguimos observar com este exemplo é que é um problema inerente ao algoritmo *k*-NN saber qual o melhor valor de *k*. Neste exemplo, $k = 2$ resolveria o nosso problema. Entretanto, em situações reais, com grandes bases de dados, como solucionar esta questão? Esta é uma das propostas desta tese.

2.7.3.4 Vantagens e desvantagens do algoritmo *k*-NN

Algumas vantagens da técnica dos *k*-vizinhos mais próximos são:

1) É robusto, inclusive para dados com ruídos;
2) Funciona bem, mesmo com um grande volume de dados;
3) Pode ser usado para regredir tanto atributos categóricos quanto numéricos;
4) Não cria modelos explícitos para cada atributo com dados ausentes;;

Já as desvantagens podem ser enumeradas:

1) O custo computacional é bastante elevado, já que devemos computar a distância de cada tupla para todas as outras;
2) A determinação de qual o melhor valor de *k* pode ser um problema bem complexo;
3) Como a essência do algoritmo reside na determinação da distância, utilizar uma outra forma de calculá-la afeta o desempenho do algoritmo?
4) É sempre adequado utilizar todos os atributos no cálculo da distância entre duas tuplas?

Não tratamos nesta seção os casos onde os registros das tabelas podem conter atributos discretos, por não ser objeto de estudo desta tese. Trabalhos como o de BATISTA (2003) apresentam soluções para determinar a participação deste tipo de atributos no cálculo da distância entre duas tuplas quaisquer da tabela.

JÖNSSON e WOHLIN (2004) mencionam que a vantagem do algoritmo dos K vizinhos mais próximos sobre a imputação por média reside no fato de que a substituição de valores ausentes utilizando o algoritmo *k*-NN é influenciada apenas pelo subconjunto de registros do conjunto de dados que são mais similares à tupla que necessita ter seu valor imputado, e não por todos os registros da tabela. Os autores citam que estudos mostram que a imputação com *k*-NN se comporta tão bem ou melhor do que outros métodos, tanto no contexto de projetos de Engenharia de Software (JÖNSSON, WOHLIN, 2004 apud SCHEFFER, 2002, SONG, SHEPPERD, CARTWRIGHT, 2005, STRIKE, EL EMAM, MADHAVJI, 2001) ou convencionais (JÖNSSON, WOHLIN, 2004 apud BATISTA, MONARD, 2001, CHEN, SHAO, 2000, TROYANSKAYA *et al*, 2001).

2.7.4 Imputação com Redes Neuronais Back Propagation

2.7.4.1 Modelagem computacional do processamento neuronal

Durante muitos anos, as pesquisas em Inteligência Artificial se basearam no postulado *cognitivista*, baseado na idéia de que as atividades do cérebro humano poderiam ser modeladas através de um conjunto de regras. Assim, o paradigma *simbolista* de representação do conhecimento foi bastante utilizado e referenciado na literatura, e vários trabalhos produziram sucesso com sua abordagem.

No paradigma simbolista, a mente é uma "máquina" de processar símbolos, além de ser centralizada e seqüencial. Assim, consegue-se modelar problemas de alto nível. E, para resolver estes problemas, utilizam-se dicas de solução, chamadas *heurísticas*. Isto torna o processo inteligente uma busca guiada e independente da estrutura.

Todavia, as tentativas de solução de problemas mais complexos, tais como o reconhecimento de padrões, a simulação dos sentidos humanos (visão, audição etc.), foram mal sucedidas. Isto porque esta classe de problemas não é de natureza tal que se possa resolver com um banco de conhecimento expresso em regras. Isto motivou o ressurgimento do chamado paradigma conexionista da Inteligência Artificial, que tenta reproduzir a inteligência humana modelando seus elementos neurofisiológicos: o neurônio (a nível elementar) e o cérebro.

Assim, no paradigma conexionista, o aspecto estrutural do ambiente de processamento de informações é muito importante. O cérebro passa a ser visto como uma rede de elementos computacionais, distribuídos no espaço e paralelos no tempo – o paralelismo compensa o baixo tempo de resposta de um neurônio (CARVALHO, 1989). A inteligência passa então a estar presente nas conexões, já que um neurônio isolado possui uma funcionalidade pobre.

Tomando por base os princípios de neurobiologia e de processamento paralelo/distribuído, os sistemas conexionistas passam então a modelar e resolver bem problemas de baixo nível, tais como reconhecimento de padrões, memória, entre outros. A inteligência passa a se expressar porque existe uma estrutura subserviente, e a mente emerge do comportamento coletivo de uma população de elementos. A Tabela 2.4 diferencia os dois paradigmas.

Tabela 2.4 Diferenças entre os paradigmas simbolista e conexionista

	Paradigma SIMBOLISTA	Paradigma CONEXIONISTA
CÉREBRO	Nada. É uma caixa preta.	É uma rede de elementos computacionais que interagem de forma paralela ou distribuída.
MENTE	Um processador de símbolos.	Emerge da computação coletiva de neurônios.
INTELIGÊNCIA	Está nas heurísticas.	Está nas conexões entre os neurônios.

O PROCESSO INTELIGENTE	É uma busca guiada por heurísticas, de forma seqüencial e centralizada.	É a evolução dinâmica dos elementos processadores, que recebem e enviam sinais excitatórios (cooperativos) ou inibitórios (competitivos).
QUANDO FUNCIONAM BEM	Atividades inteligentes ou cognitivas de alto nível (por exemplo, fala, raciocínio, planejamento). Consciente, voluntário.	Tarefas de mais baixo nível, como reconhecimento de padrões.

2.7.4.2 Aspectos Funcionais de um Neurônio

Do ponto de vista neurofisiológico extremamente simplificado (Figura 2.10), um neurônio é composto pelos seguintes elementos:

- SOMA: corpo celular
- NÚCLEO CELULAR: centro de processamento físico/químico da célula
- DENDRITO: ramificações que permitem a entrada de estímulos (sinais)
- AXÔNIO: ramificação única de saída, que reflete o processamento interno do neurônio, em função de seu estado interno e dos estímulos de entrada.

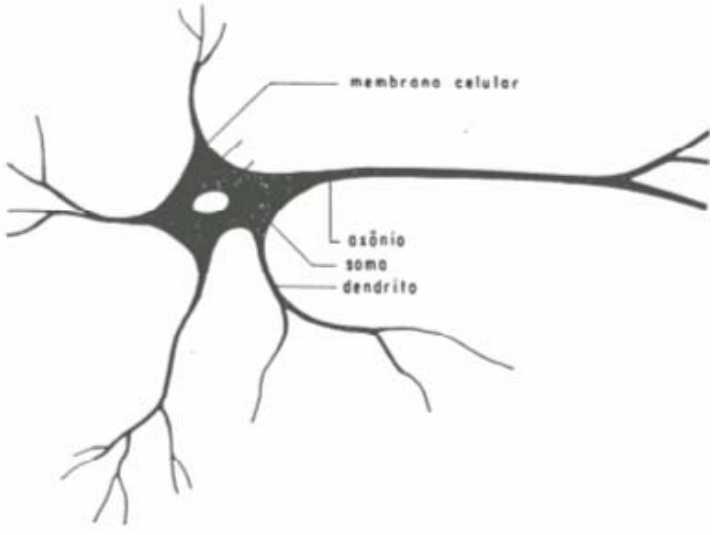


Figura 2.10 Esquema simplificado de um neurônio

O processamento em um neurônio se dá pela entrada de sinais provenientes de outros neurônios pelos dendritos. Estes sinais têm características diversas, e são alterados em intensidade (aumentados ou diminuídos) de acordo com as características da conexão. A

seguir, a resultante do processamento se acumula no soma do neurônio. Este processamento é realizado pela membrana celular, revelando uma integração no espaço e no tempo. Dependendo do sinal de entrada resultante e do processamento ocorrido, o axônio libera um sinal de saída.

Os neurônios se comunicam através de *sinapses*, que consistem na transmissão de um trem de impulsos nervosos de um neurônio para outro. Esta conexão se dá através da ligação do axônio de um neurônio com o dendrito de outro. Esta ligação é chamada *botão sináptico* (Figura 2.11). Na conexão, o axônio e o dendrito não se tocam, e o impulso elétrico é realizado pela liberação de neurotransmissores do axônio, que se encaixam a neuroreceptores existentes na membrana do dendrito do outro neurônio. A conexão entre todos os neurônios do cérebro nos dá a idéia de uma rede neuronal, a qual tentamos muito precariamente simular em modelos e implementações computacionais.

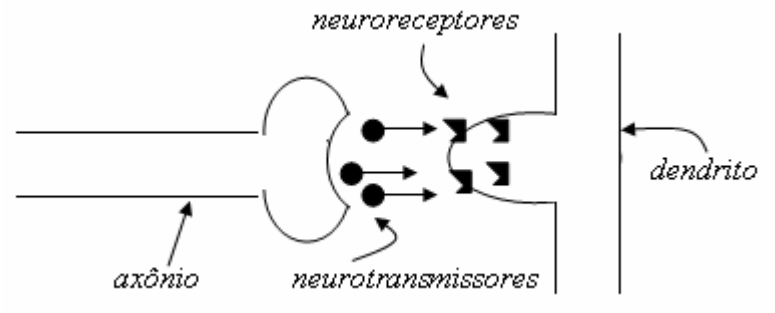


Figura 2.11 Botão sináptico

As conexões entre os neurônios podem ser de três tipos (Figura 2.12):

1. Axo-dendrítica
2. Axo-somática
3. Axo-axônica

O tipo mais comum de ligação entre os neurônios é a *axo-dendrítica*. Nesta ligação, estabelece-se um botão sináptico entre um axônio e um dendrito, conforme visto na figura 2.12. Porém, um axônio pode se comunicar diretamente com o corpo celular de um neurônio. Este tipo de ligação é chamada de *axo-somática*. E um tipo de conexão bem menos freqüente é a *axo-axônica*, conhecida também como **conexão de veto**. Este tipo de conexão faz com que o sinal passante no neurônio de um axônio seja interrompido com outro sinal de saída de outro neurônio.

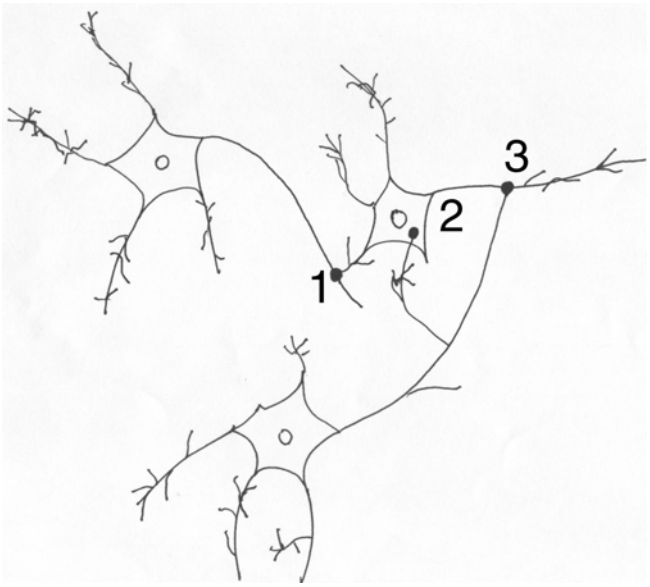


Figura 2.12 Tipos de ligação entre neurônios

A importância da ligação axo-dendrítica também varia de acordo com os seguintes fatores combinados:

- Calibre mais grosso ou mais fino
- Ligação mais próxima ou mais distante do soma

2.7.4.3 O Neurônio Linear

Uma primeira tentativa de modelagem neuronal foi obtida com o *neurônio linear* (ROSENBLATT, 1962, MINSKY, PAPERT, 1969, KOHONEN, 1974, KOHONEN, 1977, KOHONEN, 1984). Definimos os seus seguintes elementos (Figura 2.13):

- o_k é a saída do neurônio k, equivalente ao impulso nervoso de saída, transmitido pelo neurônio através de seu axônio. A função de saída do neurônio N_i pode ser expressa como:

  $$o_i(t) = f_i(a_i(t))$$

  sendo f_i a *regra de saída* do neurônio N_i.

  A resposta do neurônio linear $o_i(t)$ se confunde com o seu estado de ativação, ou seja, a regra de saída é a identidade:

  $$o_i(t) = a_i(t)$$

- w_{ki} é a *importância ou peso sináptico* de ligação entre os neurônios N_k e N_i. Normalmente é um escalar real. Se possui um valor positivo, indica que o neurônio N_k **excita** o neurônio N_i; se negativo, dizemos que N_k **inibe** N_i; e se $w_{ki} = 0$, sabemos que N_k não se comunica com N_i.

- u_i é o *impulso total de entrada*, e é definido como o somatório do produto entre o_i e os pesos sinápticos. Podemos explicitar esta equação ao longo do tempo como:

$$u_i(t) = g_i(o(t), W)$$

sendo W uma matriz que contém todos os pesos sinápticos da rede neuronal, g_i é a **regra de propagação** do neurônio N_i, que integra os impulsos recebidos pelo neurônio, e $o(t)$ é o vetor que contém todas as saídas dos neurônios da rede.

No caso do neurônio linear, temos:

$$u_i(t) = \sum_{j=1}^{n} w_{ij} o_j(t)$$

onde $\mathbf{w_{ij}}$ é a importância sináptica entre o neurônio j e o neurônio i, e $\mathbf{o_j}$ é a saída do neurônio j.

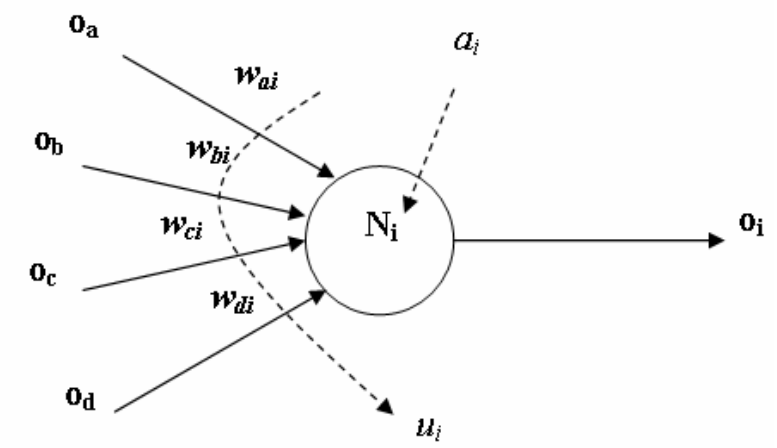


Figura 2.13 Elementos da modelagem de um neurônio

- a_i é o *estado de ativação* do neurônio N_i. Ele varia segundo a *regra de ativação* $h_i(t)$, que pode ser expressa como:

$$a_i(t+\Delta t) = h_i(a_i(t), u_i(t))$$

No caso do neurônio linear, o estado de ativação atual não depende dos anteriores, sendo linear em relação ao impulso total de entrada (esta é a razão pela qual o neurônio é dito *linear*). Assim, temos:

$$a_i(t) = \alpha\, u_i(t)$$

Combinando as três expressões anteriores, obtemos a equação que descreve o comportamento do neurônio linear:

$$o_i(t) = \sum_{j=1}^{n} \alpha\, w_{ij} o_j(t); \qquad \alpha \in \Re +$$

A simplicidade da representação linear é bastante interessante, mas oferece um espectro bastante limitado de aplicações. Esta modelagem não inclui ligações do tipo axo-axônicas.Isto motivou o desenvolvimento de um novo modelo: o *neurônio lógico* (McCULLOCH, PITTS, 1943), descrito a seguir.

2.7.4.4 O Neurônio de McCulloch-Pitts

Procurando conseguir alguma semelhança com o neurônio biológico, McCULLOCH e PITTS (1943) modelaram o estado de ativação de um neurônio baseado no seu *potencial limiar*. Com isso, o neurônio passa a ter apenas dois estados internos de ativação: **excitação** (representado pelo valor um), e **inibição** (zero). Por isso esse neurônio é conhecido como *lógico*, por assumir apenas dois valores.

Neste modelo, o impulso total de entrada e a regra de saída são expressos de forma idêntica ao neurônio linear. Já o estado de ativação interna do neurônio é explicitado como:

$$a_i(t) = \begin{cases} 1, \ se\, u_i(t) > \theta_i \\ 0, \ se\, u_i(t) \le \theta_i \end{cases}$$

onde θ_i é o dito potencial limiar do neurônio N_i. A figura 2.14 ilustra a função.

Sendo assim, a regra de saída do neurônio lógico pode ser escrita como:

$$o_i(t) = T\left(\sum_{j=1}^{n} w_j o_j(t) - \theta_i\right)$$

onde a *função limiar T(x)* é dada por:

$$T(x) = \begin{cases} 1, \ se\, x > 0 \\ 0, \ se\, x \le 0 \end{cases}$$

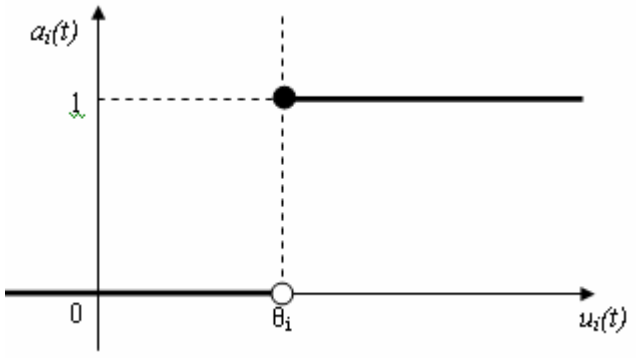


Figura 2.14 Regra de ativação do neurônio lógico

A introdução do parâmetro θ_i, como função idêntica ao potencial limiar do neurônio biológico, torna este modelo mais plausível para o estudo dos mecanismos de computação do sistema nervoso. Este parâmetro é conhecido como *threshold*, relacionado com os processos químicos internos do neurônio.

Todavia, este modelo possui algumas desvantagens:

- Só possui dois estados possíveis.
- Apresenta descontinuidade, que não é um fator biológico.

Uma solução para este problema foi apresentada por WILLIAMS (1986), com seu *neurônio semilinear*. A função de ativação passa a ser expressa como uma sigmóide (Figura 2.15), segundo a expressão:

$$a_i(t) = [1 + e^{-y(u_i(t)-\theta_i)}]^{-1};\ \ y \in \Re^+$$

Dessa forma, este neurônio possui infinitos estados de ativação, contidos no intervalo real existente entre zero e a unidade. Além disso, os problemas de otimização podem utilizar os neurônios semilineares, pois elas normalmente envolvem derivadas, neste caso particular as derivadas sempre positivas, o que é oferecido pela função sigmóide.

Apenas ilustramos o neurônio semilinear, mas existem diversas outras propostas de modelagem neuronal na literatura.

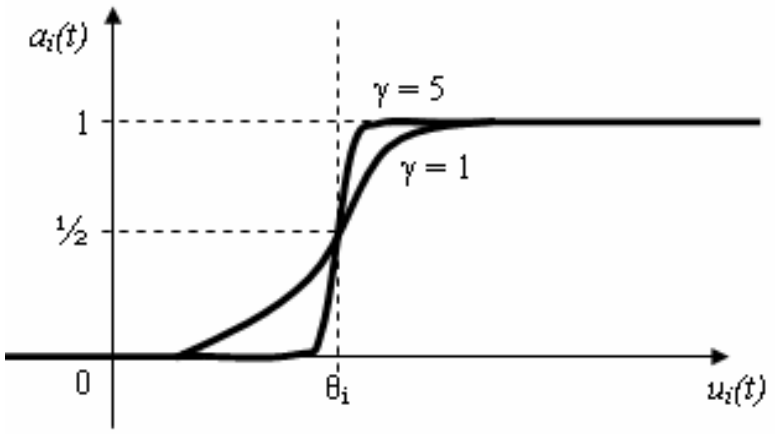


Figura 2.15 Regra de ativação do neurônio semilinear

2.7.4.5 Aprendizado por Retropropagação de Erros

Uma rede neuronal com aprendizado por retropropagação de erros, ou redes *back propagation* (RUMELHART *et al*, 1986), é uma arquitetura de rede neuronal artificial

feedforward de aprendizado supervisionado. Seu principio é o de, a partir de um conjunto de dados de entrada e de suas saídas respectivas, gerar uma rede que seja capaz de continuamente ser treinada, comparando as saídas produzidas pela rede com as que são desejadas, e ajustando os pesos das conexões sinápticas dos neurônios da rede em função da diferença (erro) entre os valores produzidos pela rede e os desejados.

A arquitetura da rede neuronal *back propagation* pode ser dividida em três camadas: a de entrada, a intermediária e a de saída. A figura 2.16, extraída de CARVALHO (2001), exemplifica bem esta idéia.

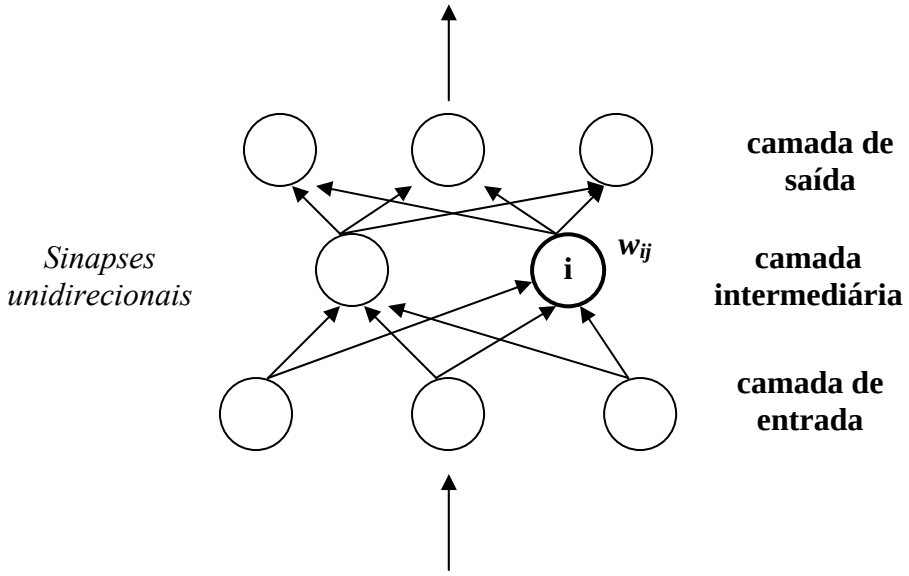


Figura 2.16 Arquitetura genérica de uma rede neuronal back propagation.
Fonte: (CARVALHO, 2001)

Os neurônios da camada de entrada são os responsáveis por receberem os sinais a serem processados pela rede. Tendo em vista que todos os neurônios da rede são interconectados, cada saída s_i de um neurônio N_i da camada de entrada é repassada a todos os neurônios da camada intermediária. Este esquema de interconexão se repete, até que todos os neurônios do último nível da camada intermediária enviem suas respostas a todos os neurônios da camada de saída.

O número de neurônios nas camadas de entrada e saída são determinados pela natureza e modelagem do problema. Já o número de níveis da camada intermediária, bem como o número de neurônios por nível desta camada, não possuem limites inferior e superior. Apesar de, a princípio, ser possível que uma rede neuronal não apresente camada intermediária de neurônios, os problemas que estas redes podem resolver são

somente os linearmente separáveis (HAYKIN, 1999), o que limita em muito o seu uso. Com camadas intermediárias, a rede neuronal pode resolver tanto problemas lineares quanto não lineares, aumentando enormemente o seu espectro de utilização (MEDSKER, L., LIEBOWITZ, 1994).

WALCZAK e CERPA (1999) comenta e referencia uma série de trabalhos que analisam a quantidade de níveis da camada intermediária, relacionando inclusive a quantidade de níveis com a capacidade de modelar problemas: um nível na camada intermediária criam um hiperplano, enquanto dois níveis na camada intermediária combinam hiperplanos para formar áreas convexas de decisão; três níveis combinam áreas convexas que contêm regiões côncavas. Como as heurísticas ligadas à determinação de número de níveis da camada intermediária estão muito atreladas a modelagens específicas de problemas, e levando-se em consideração que problemas que possuem um domínio com uma solução padrão não linear são solucionáveis com um único nível na camada intermediária (BANSAL *et al*, 1993), modelamos a rede neuronal utilizada neste trabalho com apenas um nível na camada intermediária.

A não existência de um consenso no número de níveis na camada intermediária ventila um leque de diversas possibilidades. Todavia, estas opções não são tão amplas quando a questão é determinar quantos neurônios devemos associar a cada nível da camada intermediária. O trabalho de WALCZAK e CERPA (1999) aponta uma série de referências que indicam várias opções para determinar o número de neurônios que devíamos utilizar em nossa camada intermediária. Assim, dadas as inúmeras possibilidades, e entendendo que estudos nesta área aplicados à tarefa de imputação de dados ausentes podem ser feitos em um momento posterior, utilizamos a heurística proposta por HECHT-NIELSEN (1988), FLETCHER e GOSS (1993), PATUWO, HU e HUNG (1993), que sugerem o uso de *(2n+1)* neurônios na camada intermediária, onde *n* é o número de neurônios na camada de entrada.

Para acertar os pesos das sinapses de seus neurônios, a rede neuronal *back propagation* utiliza neurônios semilineares, da seguinte forma:

$$u_j = \sum_i w_{ij} * o_j \qquad \text{(regra de propagação)}$$

$$o_j = f(u_j) \qquad \text{(regra de ativação sigmóide)}$$

Já que a rede utiliza um aprendizado supervisionado, onde as saídas desejada e real são comparadas, definimos então o erro do aprendizado do neurônio N_j como:

$$E = \frac{1}{2} * \sum_j (t_j - o_j)^2$$

Desejamos minimizar E em relação a w_{ij}:

$$\Delta w_{ij} \approx -\frac{\partial E}{\partial w_{ij}}$$

(CARVALHO, 2001) demonstra que, ao minimizar Δw_{ij}, obtemos a seguinte expressão de ajuste dos pesos:

$$\boxed{\Delta w_{ij} = \eta * \delta_{pj} * o_{pi}}$$

(η é a constante de proporcionalidade)

Todavia, existem diferenças entre os neurônios da camada de saída e os demais. Esta diferença se dá pelo fato de que neurônios que não de uma camada c_i, que não de saída, são influenciados pelos neurônios da camada c_{i+1} (esta é a razão de o algoritmo ser chamado de retro alimentação). Assim, temos dois possíveis casos:

1º Caso: O neurônio j é da camada de saída

O erro (fator de correção) neste caso é expresso por:

$$\boxed{\delta_j = (t_j - o_j) * f'(u_j)}$$

2º Caso: O neurônio j NÃO é da camada de saída

O erro (fator de correção) é obtido pela fórmula a seguir:

$$\boxed{\delta_j = \sum_k (\delta_k * w_{kj})}$$

Assim, o aprendizado consiste em duas fases (figura 2.17):

1) A rede é estimulada e as respostas desejadas t_i e as efetivas u_i são obtidas.

2) As sinapses são modificadas segundo as regras:

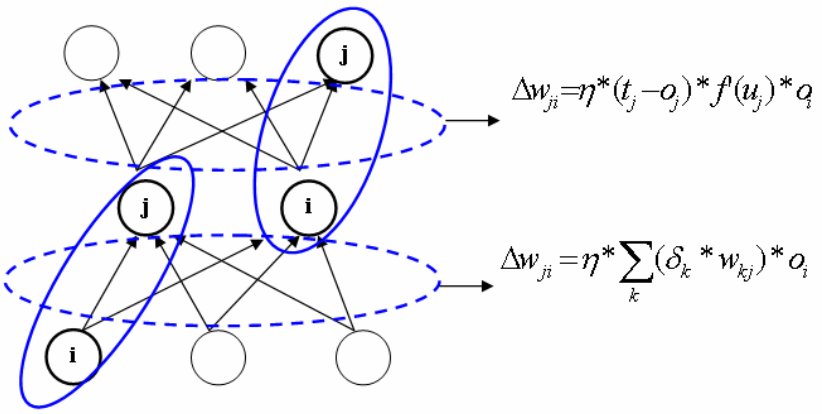


Figura 2.17 Ajuste de erros nas sinapses de uma rede neuronal back propagation

CAPÍTULO 3

O PROCESSO DE COMPLEMENTAÇÃO DE DADOS AUSENTES EM BASES DE DADOS

3.1 Introdução

Uma importante e desafiadora tarefa na etapa de pré-processamento de bases de dados – especificamente na subtarefa de limpeza de dados – é a complementação de valores que por alguma razão não constam registro nas tabelas de um conjunto de dados. A não existência de valores pode comprometer todo o processo de Descoberta de Conhecimento em Bases de Dados, já que alguns algoritmos podem gerar padrões pouco esclarecedores ou tendenciosos por conta desta ausência.

Apesar da freqüente ocorrência de valores desconhecidos em conjuntos de dados, muitos analistas de dados tratam os valores desconhecidos de forma bastante simplista (BATISTA, MONARD, 2003a). Alguns algoritmos sequer estão preparados para trabalhar com um conjunto de dados com lacunas. Como infelizmente não é possível seguir a utópica recomendação de ALLISON (2001), que diz que a única boa maneira de tratar dados ausentes é não tê-los, é extremamente importante que tenhamos à disposição formas de tratamento de valores ausentes em bases de dados.

Alguns métodos são muito fáceis de usar, mas eles têm vários contrapontos que limitam seu uso a problemas pouco interessantes (ALLISON, 2001, CHAPMAN, 1997, MAGNANI, MONTESI, 2004). A forma escolhida para tratar os dados incompletos pode ser determinante no seu sucesso. Esta escolha pode diferenciar um estudo tendencioso ou não (TWALA *et al*, 2005). Porém, não existe um método único que atue bem em todos os tipos de dados ausentes. Diferentes situações exigem diferentes soluções (CRÉMILLEUX, RAGEL, BOSSON, 1999, RAGEL, CRÉMILLEUX, 1999, TSENG, WANG, LEE, 2003, MAGNANI, 2004). O mais importante aspecto da substituição de valores ausentes é a não distorção das características originais da amostra (HRUSCHKA *et al*, 2003a).

Apesar de o problema de dados incompletos poder ser tratado adequadamente em vários conjuntos de dados reais, existem muito poucos trabalhos publicados ou estudos

empíricos avaliando a precisão das predições de métodos de complementação de dados usando algoritmos de aprendizado de máquina supervisionados, tais como árvores de decisão (TWALA *et al*, 2005 apud BREIMAN *et al*, 1984, LITTLE, RUBIN, 1987). Além disso, os métodos de imputação são independentes das técnicas de aprendizado de máquina utilizados, dando liberdade ao usuário para escolher o melhor método para o seu caso (BATISTA, MONARD, 2003a).

Inspirado nestas duas assertivas, esta tese analisa a aplicação de técnicas de Inteligência Computacional na tarefa de complementação de dados. Todavia, por questões operacionais (um elevado número de experimentos demandaria um grande tempo computacional, além de tornar mais complexa a análise dos resultados), nosso estudo limitou-se a algumas técnicas. Por esta razão, propomos um *framewok* de imputação de valores ausentes, onde outras técnicas podem, da mesma forma, ser aplicadas e avaliadas.

3.2 Possíveis causas da ausência de dados

As razões que justificam a existência de dados desconhecidos em um conjunto de dados são diversas. Algumas mais comuns incluem falta de tempo ou de comprometimento dos entrevistados, elevado custo de carga dos dados, falta de um treinamento adequado por parte dos coletores de dados, razões políticas, entre outros (CARTWRIGHT *et al*, 2003). Alguns valores podem ser ausentes por não fazer sentido, como por exemplo um atributo que indique o número de vezes que um paciente homem engravidou (ONISKO *et al*, 2002).

BROWN e KROS (2003) enumeram outras razões para que atributos de registros de tabelas não estejam preenchidos:

1) *Fatores operacionais*: erros na entrada de dados, estimativas erradas, remoção acidental de campos de tabelas, entre outras.

2) *Recusa na resposta em pesquisas*: alguns entrevistados podem deixar uma ou mais questões em branco por diversas razões: ou por se sentirem constrangidas em responder (perguntas como idade, renda ou orientação sexual), ou por não conhecerem o assunto (por exemplo, estudantes que são argüidos sobre que carreira desejam seguir);

3) *Respostas não aplicáveis*: por vezes, questões apresentadas em questionários não se aplicam aos entrevistados (por exemplo, perguntas envolvendo ganhos

de atividade rural para moradores de metrópoles, ou questões direcionadas a fumantes aplicadas a não fumantes)

Alguns valores são ausentes por razões que somente o coletor dos dados conhece, ou por privacidade dos dados (RUBIN, 1988). Os analistas de dados normalmente não estão familiarizados com os detalhes da carga de dados, como por exemplo, os que ocorrem em pesquisas. Assim, eles podem se beneficiar de um tratamento mais cuidadoso dos valores ausentes realizado previamente pelos coletores de dados, do que eles poderiam ter se eles próprios realizassem este pré-processamento (CHIU, SEDRANSK, 1986). Por este motivo, FORD (1983) aponta algumas das razões que justificam o porquê de a responsabilidade do ajuste de valores ausentes ser do coletor de dados:

1) Ele possui conhecimento sobre a natureza dos dados;
2) Os coletores de dados devem normalmente fazer estimativas sobre o conjunto de dados;
3) A maioria dos analistas de dados não querem a responsabilidade de ajustar os dados ausentes;
4) Um conjunto de dados sem valores ausentes permite que todas as análises futuras tenham um ponto inicial comum sem que cada analista de dados coloque o seu próprio valor inicial.

RAGHUNATHAN (2004), SCHAFER e GRAHAM (2002), e TWALA, CARTWRIGHT e SHEPPERD (2005) citam dois motivos para existirem dados ausentes. O primeiro caso é chamado de **dado ausente unitário** (*unit nonresponse*). Nesta situação, algumas das ocorrências completas dos dados podem não ter sido coletadas da amostra original, ou alguns entrevistados podem ter se recusado a responder a todos os questionamentos. Uma segunda opção seria a dos **itens de dados ausentes** (*item nonresponse*), onde alguns dos dados coletados estão preenchidos e outros não, ou os entrevistados podem não quererem ou não se sentirem à vontade para responder a alguns itens de uma pesquisa. SCHAFER e GRAHAM (2002), e TWALA, CARTWRIGHT e SHEPPERD (2005) acrescenta uma terceira situação, onde ocorrem **levas de dados ausentes** ("*wave*" *nonresponse*). Nesta situação, não encontramos respostas para um assunto de uma seção em uma ou mais levas, ou quando um

entrevistado deixa de responder uma seção para fazê-lo a posteriori, e não retorna. Os autores mencionam que o tipo mais comum é o de itens de dados ausentes.

3.3 Padrões de ausência de dados

SCHAFER e GRAHAM (2002) e TWALA, CARTWRIGHT e SHEPPERD (2005) mencionam que, para poder melhor escolher a técnica de preenchimento de valores ausentes, é necessário saber não só o mecanismo da ausência dos dados, mas também o seu **padrão de ausência**. Estes, por sua vez, são categorizados em *gerais* ou *aleatórios*, e *específicos*.

Os dados de padrão geral ou aleatório são aqueles onde podemos encontrar atributos ausentes em quaisquer registros do conjunto de dados. Já os padrões de ausência específicos são dos tipos *univariados* e *monotônicos*. Padrões univariados são aqueles onde a ausência dos dados está restrita a uma única variável da tabela. Já padrões monotônicos acontecem quando, a partir de um conjunto de atributos A_i, $1 \leq i \leq n$, não conseguimos encontrar dados nos atributos $A_j, A_{j+1}, A_{j+2}, \ldots, A_p$, $j < p$. A figura 3.1 mostra graficamente os conceitos apresentados.

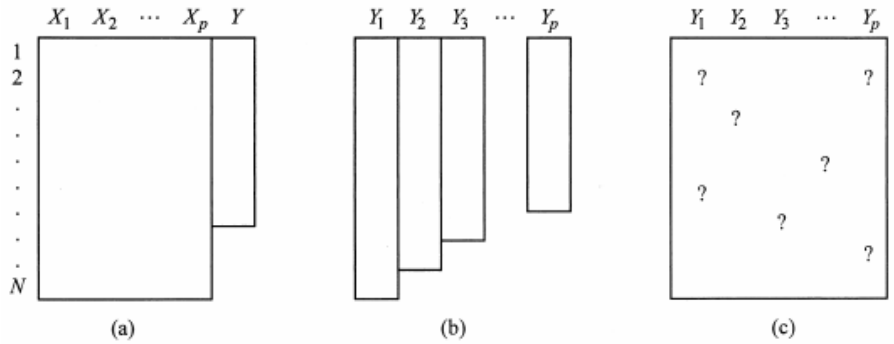


Figura 3.1 Padrões de ausência de respostas em um conjunto de dados retangulares: (a) padrões univariados, (b) padrões monotônicos; (c) padrões arbitrários. Fonte: SCHAFER e GRAHAM (2002)

3.4 Mecanismos de ausência de dados

Todos os trabalhos envolvendo complementação de dados ausentes levam inevitavelmente em conta o mecanismo que causou a ausência dos dados. Estes mecanismos podem ser de três tipos: completamente aleatório (MCAR – *Missing Completely At Random*), aleatório (MAR – *Missing At Random*), ou de não aleatório (NMAR – *Not Missing At Random*, ou IM – *Ignorable Missing*) (LITTLE, RUBIN, 1987).

Para exemplificar os conceitos abaixo explanados, consideramos uma tabela com algumas informações sobre vistorias de automóveis. Seu esquema e estado são explicitados na figura 3.2.

id	*placa*	*Km*	*total litros óleo motor*	*total litros óleo freio*	*condição dos pneus*	*parecer*
1	LBX5466	56715	3,8	0,8	Regular	Reprovado
2	KMX6901	34110	4,6	1,1	Bom	Aprovado
3	LOV6385	17954	3,9	1,5	Ruim	Reprovado
4	LUJ1186	8112	4,0	1,6	Bom	Aprovado
5	KWW5566	115462	3,5	1,9	Regular	Aprovado
6	CLV4223	60715	3,3	0,4	Regular	Reprovado
7	AMD1236	148003	4,2	1,3	Bom	Aprovado

Figura 3.2 Exemplo de uma tabela com dados totalmente preenchidos

Os dados ausentes de uma tabela de um conjunto de dados são ditos **completamente aleatórios** quando o motivo de sua ausência é desconhecido. Desta forma, considerando que alguns dados do atributo "*total_litros_óleo_freio*" se apresentassem como na figura 3.3:

id	*placa*	*Km*	*total litros óleo motor*	*total litros óleo freio*	*condição dos pneus*	*parecer*
1	LBX5466	56715	3,8	<*NULL*>	Regular	Reprovado
2	KMX6901	34110	4,6	1,1	Bom	Aprovado
3	LOV6385	17954	3,9	1,5	Ruim	Reprovado
4	LUJ1186	8112	4,0	1,6	Bom	Aprovado
5	KWW5566	115462	3,5	<*NULL*>	Regular	Aprovado
6	CLV4223	60715	3,3	0,4	Regular	Reprovado
7	AMD1236	148003	4,2	<*NULL*>	Bom	Aprovado

Figura 3.3 Tabela com dados ausentes completamente aleatórios

Classificamos este tipo de ausência como **completamente aleatória**, por não sabermos precisar as razões pelas quais algumas medidas do atributo não tiveram os seus valores registrados.

Sendo Z um conjunto de dados, subdividido em Z^{pres} (o subconjunto de dados com valores presentes) e Z^{aus} (o subconjunto com valores ausentes), e R um conjunto indicador de respostas ($R_i = 1$ indica que o i-ésimo elemento de Z está preenchido, e

R_i = 0 revela que o valor é ausente), regida por parâmetros φ. Expressamos a probabilidade de valores ausentes do tipo completamente aleatório como:

$$\mathbf{P(R \mid Z) = P(R \mid Z^{pres}, Z^{aus}) = P(R \mid \varphi)}$$

Entretanto, se alguns valores do atributo "*condição_dos_pneus*" se tornassem ausentes em função de valores menores que 1,0 litro do atributo "*total_litros_óleo_freio*", dizemos que estes são **valores ausentes aleatórios**. A figura 3.4 exemplifica o conceito exposto.

id	*placa*	*Km*	*total litros óleo motor*	*total litros óleo freio*	*condição dos pneus*	*parecer*
1	LBX5466	56715	3,8	0,8	Regular	Reprovado
2	KMX6901	34110	4,6	1,1	Bom	Aprovado
3	LOV6385	17954	3,9	1,5	<*NULL*>	Reprovado
4	LUJ1186	8112	4,0	1,6	Bom	Aprovado
5	KWW5566	115462	3,5	1,9	<*NULL*>	Aprovado
6	CLV4223	60715	3,3	0,4	Regular	Reprovado
7	AMD1236	148003	4,2	1,3	<*NULL*>	Aprovado

Figura 3.4 Exemplo de uma tabela com dados ausentes aleatórios

Dados são aleatórios somente (MAR) se:

$$\mathbf{P(R \mid Z) = P(R \mid Z^{pres}, \varphi)}$$

BATISTA e MONARD (2003a) alegam que, na maioria dos casos, os atributos não são independentes. Descobrindo-se a relação entre eles, avaliam-se os valores ausentes. Se isto acontecer, todos os dados ausentes podem então ser enquadrados como possuindo um mecanismo de ausência aleatório (MAR), e, a partir de sua conhecida relação, um modelo de regressão, e os valores ausentes são então obtidos em função dos demais atributos da relação. Como dificilmente conseguimos obter essa relação – especialmente em dados que são alvo de processamento de algoritmos de Mineração de Dados – o estudo de técnicas de complementação de dados ausentes considerando os mecanismos já descritos é plenamente justificável.

A ausência de valores de campos de uma tabela pode, entretanto, ser causada por valores que dependem do próprio atributo onde ocorreu a ausência. Assim, se, por exemplo, um dos aparelhos que realiza as medições de volume de óleo de freio não conseguir realizar medidas de valores menores que 1.0 litro, dizemos que a causa da

ausência é **não aleatória** (NMAR – *Not Missing At Random*), como pode ser observado na figura 3.5.

id	*placa*	*Km*	*total litros óleo motor*	*total litros óleo freio*	*condição dos pneus*	*parecer*
1	LBX5466	56715	3,8	0,8	Regular	Reprovado
2	KMX6901	34110	4,6	1,1	Bom	Aprovado
3	LOV6385	17954	3,9	1,5	<*NULL*>	Reprovado
4	LUJ1186	8112	4,0	1,6	Bom	Aprovado
5	KWW5566	115462	3,5	1,9	<*NULL*>	Aprovado
6	CLV4223	60715	3,3	0,4	Regular	Reprovado
7	AMD1236	148003	4,2	1,3	<*NULL*>	Aprovado

Figura 3.5 Tabela com dados ausentes não aleatórios

Alguns autores fazem considerações sobre os mecanismos de dados apresentados. TWALA, CARTWRIGHT e SHEPPERD (2005) afirmam que é impossível confrontar dados MAR e NMAR sem considerações adicionais. Já LITTLE e RUBIN (2002) dizem que dados MAR fazem mais sentido do que MCAR. De fato, o tratamento de dados MAR e NMAR aumenta consideravelmente a complexidade do problema, pois, nestes casos, a ausência acontece por uma conjunção de fatores a princípio desconhecida. Por exemplo, em nosso exemplo, se não soubéssemos que valores menores que 1,0 litro do atributo "*total_litros_óleo_freio"* causassem a omissão do atributo "*condição_dos_pneus*", poderíamos achar que esta ausência aconteceria quando o nível de óleo fosse 0,5 ou 0,9 ou qualquer outro valor. Assim, a menos que saibamos em que condições os dados se tornaram MAR ou NMAR, tratar os dados como MCAR apresenta-se sempre como uma boa primeira opção.

BROWN e KROSS (2003a, 2003b) e SCHAFER e GRAHAM (2002) especificam ainda um quarto mecanismo de ausência de dados: **valores fora da faixa esperada para um dado atributo**. Segundo os autores, situações como estas devem, na maioria dos casos, ser tratadas como valores ausentes. Apesar de não explicitamente mencionado, acreditamos que os autores levam em consideração situações onde a correção de desvios não pode ser tratada na tarefa de limpeza de dados. Se considerarmos que esta tarefa de pré-processamento (correção de desvios) sempre corrige dados extremos, então esta quarta categoria não se aplica. Como não encontramos na literatura nenhum outro trabalho que referenciasse este quarto

mecanismo de ausência de dados, consideramos que ele é, na verdade, a tarefa de detecção de desvios na limpeza dos dados.

Ainda no que diz respeito ao mecanismo de ausência dos dados, WAYMAN (2003) apud GRAHAM e DONALDSON (1993) classifica os mecanismos de dados em **acessíveis** e **inacessíveis**. Mecanismos *acessíveis* são aqueles onde a causa da ausência pode ser explicada, tais como com dados completamente aleatórios (MCAR) ou dados aleatórios (MAR). Por outro lado, mecanismos *inacessíveis* são aqueles onde o mecanismo é desconhecido. Dados que não são aleatórios (NMAR) ou dados aleatórios onde a causa da ausência é conhecida, mas não medida, são exemplos desta categoria.

3.5 Soluções para o tratamento de dados ausentes

Encontramos na literatura uma grande variedade de propostas de soluções para o tratamento de dados ausentes. Como este problema tem origem não só no processo de Descoberta de Conhecimento em Bases de Dados, mas também na análise estatística de amostras de dados, diversas técnicas foram desenvolvidas e quase sempre comparadas com outras, de forma a avaliar a qualidade da nova proposta.

Definitivamente não existe consenso entre os autores sobre a classificação das diversas técnicas de dados ausentes existentes. Os trabalhos publicados no tema propõem uma classificação própria, com um grau de generalização maior ou menor.

BATISTA e MONARD (2003a) propõem uma classificação para os métodos de tratamento de dados ausentes:

1) **Ignorar e descartar dados** (*Complete Case Analysis*): a única restrição é que o mecanismo de ausência dos dados deve ser completamente aleatório (MCAR).

2) **Descartar tuplas com atributos que têm muitos dados ausentes**: cabe análise. Atributos relevantes devem ser mantidos. Dados devem possuir um mecanismo de ausência completamente aleatório (MCAR).

3) **Estimativa de parâmetros**: rotinas de maximização de vizinhanças são usadas para estimar os parâmetros de um modelo definido para completar os dados.

4) **Imputação**: classe de rotinas que busca preencher valores ausentes com valores estimados. O objetivo é empregar as relações conhecidas entre os valores válidos do conjunto de dados para estimar estes valores.

Já TWALA, CARTWRIGHT e SHEPPERD (2005) listam três categorias de dados ausentes:

1) Análise de dados completos
2) Imputação
3) Procedimentos baseados em modelos

MYRTVEIT, STENSRUD e OLSSON (2001) categorizam as técnicas de tratamento de dados ausentes da seguinte forma:

1) Descarte de observações (registros) incompletos
2) Técnicas baseadas em imputação (simples ou múltipla)
3) Técnicas baseadas em designações de pesos
4) Técnicas baseadas em modelos

TSENG, WANG, LEE (2003) classificam os métodos de complementação de dados em:

1) Baseados em Imputação (*Imputation based*): para dados numéricos
2) Baseados em Mineração de Dados (*Data-mining based*): para dados categóricos

Segundo HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003a, 2003b), as formas de tratamento de valores ausentes são:

1) Ignorar as tuplas com valores ausentes
2) Preenchê-los manualmente (opção inviável em KDD)
3) Substituir o valor ausente por uma constante (pode levar a grandes distorções)
4) Usar média ou moda
5) Atribuir o valor mais provável.

MAGNANI (2004) propõe uma taxonomia mais refinada, buscando contemplar todas as possibilidades de técnicas de tratamento de valores ausentes:

1) **Métodos Convencionais**

   a) Remoção completa de casos (*Listwise Deletion* ou *Complete-Case Deletion*)

   b) Remoção em pares (*Pairwise Deletion*)

   c) Remoção de colunas com valores ausentes

2) **Imputação**:

   a) Imputação Global Baseada no Atributo Ausente

   b) Imputação Global Baseada nos Atributos não Ausentes

   c) Imputação Local

3) **Estimativa de Parâmetros**

4) **Gerenciamento Direto dos Dados Ausentes**

De todas as opções propostas, a classificação de MAGNANI (2004) é a que melhor caracteriza os atuais métodos de complementação de dados ausentes, por ser a que melhor estrutura as diferenças entre os métodos de complementação de dados ausentes. Todavia, esta organização restringe os métodos estatísticos, resumindo-os apenas ao método de estimativas de parâmetros. Apesar de resumida, a proposta apresentada por TWALA, CARTWRIGHT e SHEPPERD (2005) também se mostra adequada. Todavia, uma proposta híbrida poderia melhor categorizar os atuais métodos de complementação de dados existentes. Assim, a taxonomia que acreditamos englobar todos os possíveis casos é a explicitada a seguir:

1) **Métodos Convencionais**

   a) Remoção completa de casos (*Listwise Deletion* ou *Complete-Case Deletion*)

   b) Remoção em pares (*Pairwise Deletion*)

   c) Remoção de colunas com valores ausentes

2) **Imputação**:

   a) Imputação Global Baseada no Atributo Ausente

   b) Imputação Global Baseada nos Atributos não Ausentes

c) Imputação Local

3) Modelagem de dados

a) Métodos de Verossimilhança

b) Modelos Bayesianos

4) Gerenciamento Direto dos Dados Ausentes

5) Métodos Híbridos

a) Imputação Múltipla

b) Imputação Composta

3.5.1 Métodos Convencionais

3.5.1.1 Remoção Completa de Casos

A remoção completa de casos (*listwise deletion* ou *complete-case deletion*) é a opção de uso mais simples. Neste caso, todo registro de um conjunto de dados que possuir algum de seus atributos ausente é removido da amostra.

Apesar de fácil implementação e uso, esta técnica apresenta sérios problemas:

1) A remoção pode descartar grande parte dos dados, tornando-os tendenciosos;
2) A remoção também causa a poda de algumas regras, e com isso alterando o suporte e a confiança das regras que sobram;
3) O mecanismo de ausência dos dados deve ser completamente aleatório (MCAR).

3.5.1.2 Remoção em pares

A remoção em pares (*pairwise deletion*) é uma variante da remoção completa de casos. Utiliza um registro incompleto durante a análise somente quando a variável desejada não é ausente. A vantagem desta técnica é a de utilizar todos os dados disponíveis na base de dados. Como desvantagem, sua implementação é mais complexa do que a remoção completa de casos, e algumas vezes não pode ser usada.

3.5.1.3 Remoção de colunas ausentes

Qualquer atributo que apresente um valor ausente em qualquer dos registros da base de dados é totalmente removido. Sua aplicação é na maioria das vezes não recomendada, já que a perda de informação que existe com a remoção é significativa.

Além disso, a remoção de uma coluna altera toda a relação existente entre os demais atributos da tabela, pela perda dos atributos que estavam preenchidos, o que pode tornar os dados bastante tendenciosos.

Métodos convencionais tais como remoção completa de casos e imputação por média (WAYMAN, 2003 apud GRAHAM *et al*, 2002) são inaceitáveis, bem como regressão múltipla e remoção em pares (*pairwise deletion*).

3.5.2 Imputação

A imputação é o procedimento de substituição de valores ausentes. Este método permite que o tratamento de valores desconhecidos seja independente do algoritmo de máquina utilizado (BATISTA, MONARD, 2003b). Seu uso é bastante adequado para grandes bases de dados (CARTWRIGHT *et al*, 2003). A imputação pode ser feita de forma *determinística*, onde se utiliza como base os dados da tabela, ou *estocástica* (MAGNANI, 2004).

HU, SALVUCCI e COHEN (1998) descrevem os seguintes objetivos da imputação:

1) Permitir que usuários finais realizem estatísticas sobre os dados com ferramentas padrão, como se não existissem dados ausentes;

2) Obter inferências estatísticas válidas. Este objetivo pode ser alcançado com alguns métodos de imputação, e não com outros;

TWALA, CARTWRIGHT e SHEPPERD (2005) destacam que a maior fraqueza da imputação simples está no fato de que ela esconde a incerteza do dado, levando a intervalos de testes e confiança inválidos, já que os valores estimados são derivados dos dados existentes.

3.5.2.1 Imputação Global Baseada no Atributo com Valores Ausentes

A *imputação global baseada no atributo com valores ausentes* utiliza os valores existentes nas demais tuplas para preencher os que são desconhecidos. Eles podem ser de dois tipos:

1) *Determinísticos*: os mais comuns são a média ou a moda

2) *Estocásticos*: introdução de uma perturbação na média

As duas opções acima são ruins para a geração de regras de associação, pois as regras estão à procura de valores diversificados, não em resumos.

3.5.2.2 Imputação Global Baseada nos Demais Atributos

A *imputação global baseada nos demais atributos* produz novos valores a partir da relação que possa existir entre os atributos da amostra.

Problemas com a imputação global baseada nos outros atributos:

1) Que regressão usar?
    a. Pode haver valores ausentes também nos atributos de entrada do algoritmo de regressão
    b. A regressão é baseada no fato de que o modelo escolhido é o melhor para os dados, mas nem sempre é
2) Apenas um dado é imputado, o que faz um conjunto de dados reparado parecer não possuir incertezas, quando, na verdade, esta imprecisão é implícita.

3.5.2.3 Imputação Local (Procedimentos *hot-deck*)

Uma das técnicas mais utilizadas em imputação é a técnica *hot-deck* (FORD, 1983). A idéia consiste em se utilizar no processo de complementação de dados apenas um subconjunto completo dos dados, que atendem a algum critério de similaridade. A forma exata na qual o valor imputado é calculado não é importante no método.

Apesar de a técnica não ser embasada em uma teoria bem definida, o método procura reduzir o desvio (*bias*), classificando a amostra. Este objetivo é extremamente difícil de ser atingido.

Dentro do grupo, os objetos respeitam a um critério de similaridade, e os elementos tendem a ser homogêneos. Todavia, é importante que exista correlação entre os atributos classificadores – que serviram de base para a geração dos grupos – e os atributos ausentes, sob pena de obtenção de resultados equivocados. Além disso, a técnica assume que os registros possam ser agrupados, o que nem sempre é verdade. Porém, esta premissa pode ser adotada na etapa de Mineração de Dados.

A classificação pode não se basear exclusivamente em dados da amostra. Os coletores de dados podem observar dados sobre os respondentes, tais como raça, sexo,

ou faixa etária, ou mesmo condições geográficas, ou outros fatores para classificarem-na.

Algumas razões que impulsionam a utilização da técnica *hot-deck* (MAGNANI, 2004):

1) Consegue-se uma redução de desvio sem a imposição de um modelo rígido. Reduzindo o tamanho do grupo, os dados tendem a se tornar homogêneos;
2) Produção de um conjunto de dados limpo, sem valores ausentes;
3) Preservação da distribuição da população representada pela amostra.
4) Para alguns valores ausentes, nenhuma informação sobre imputação pode ser encontrada. Isto permite que outras técnicas de imputação possam ser usadas em conjunto;
5) Pode-se usar uma técnica diferente para cada grupo gerado.
6) Não precisa de um modelo robusto para prever valores ausentes
7) Não assume nenhuma distribuição em particular.

Um conjunto de dados com valores imputados não pode ser encarado como um conjunto de dados originalmente preenchido, já que a imputação oculta a incerteza inerente ao processo. FORD (1983) sugere que os dados imputados devem estar preferencialmente marcados, pois isto permite que eles sejam novamente gerados segundo a vontade do analista de dados que o usa. Devemos considerar também a marcação dos dados ausentes, pois alguns usuários não gostam de usar dados imputados, e outros pelo menos desejam saber se eles existem, e quantos são.

AUSTIN e ESCOBAR (2005) também se referem à adição de um sinal indicador (*flag*) ao lado de cada atributo de uma tupla que apresente valores ausentes. Todavia, esta abordagem enviesa os coeficientes de regressão linear múltipla (AUSTIN e ESCOBAR, 2005 apud JONES, 1996) e seu uso é desencorajado (AUSTIN e ESCOBAR, 2005 apud VACH, 1994).

A técnica *cold-deck* diferencia-se da técnica *hot-deck*, pois aquela utiliza dados de outra fonte, que não os dados correntes.

3.5.3 Modelagem de Dados

Esta classe de soluções para tratamento de valores ausentes engloba técnicas estatísticas e probabilísticas de obtenção de um modelo que consiga representar de forma genérica as características dos dados. As técnicas mais utilizadas desta categoria são os algoritmos de verossimilhança e os métodos bayesianos, que não são o alvo de estudo desta tese.

3.5.3.1 Métodos de Verossimilhança

Métodos de verossimilhança são técnicas que procuram estimar os parâmetros de uma função de distribuição estatística, para encontrar um modelo que represente o conjunto de dados, e, com isso ter condições de regredir qualquer valor ausente existente. Analisamos nesta seção o método mais aplicado na tarefa de complementação de dados, o algoritmo EM – *Expectation-Maximization*. Tratamos a seguir de uma solução encontrada na literatura, o *método de verossimilhança com informações completas*.

3.5.3.1.1 O algoritmo EM (Expectation-Maximization)

O algoritmo *EM* (*Expectation-Maximization*), proposto por DEMPSTER, LAIRD e RUBIN (1977) é um procedimento estatístico iterativo de verossimilhança, que estima os parâmetros de uma função de densidade (probabilística) de uma amostra. Seu objetivo é maximizar a função:

$$\theta_{n+1} = \arg\max_{\theta} Q(\theta)$$

Considere que o conjunto de dados D possa ser dividido em duas partes: D_C, o subconjunto onde todas as tuplas estão completas, e D_M, onde os registros apresentam algum valor ausente. Assim, podemos definir a função $Q(\theta)$ como:

$$Q(\theta) = E_Z\left[\log p(D_C, D_M \mid \theta) \mid D_C\right]$$

Para valores discretos, a equação pode ser reescrita como:

$$Q(\theta) = \sum_{Z} p(D_M \mid D_C, \theta_n) * \log p(D_C, D_M \mid \theta)$$

e para valores contínuos:

$$Q(\theta) = \int_{-\infty}^{+\infty} p(D_M \mid D_C, \theta_n) * \log p(D_C, D_M \mid \theta) dz$$

No passo E, os dados são lidos, uma tupla por vez. Assim que cada linha da tabela é carregada, seus atributos são contabilizados no cálculo de "estatísticas suficientes" (somas, somas de quadrados, soma de produtos cruzados). Se existem valores ausentes na tupla lida, eles contribuem diretamente para estas somas. Se uma tupla apresenta valor ausente em um atributo, a melhor previsão é utilizada no lugar deste valor não preenchido. No passo M, uma vez que todas as somas foram calculadas, pode-se facilmente obter a matriz de covariância. Estes dois passos continuam até que mudanças na matriz de covariâncias, entre um passo e o seguinte, tornem-se extremamente pequenas.

O que pudemos observar, ao analisar a literatura disponível no tema de complementação de dados, é que, apesar de o algoritmo EM ser bastante utilizado – tanto de forma isolada quanto combinado com outros algoritmos – não existe um consenso sobre a adequação de seu uso. SCHAFER e GRAHAM (2002) recomendam que sejam aplicados métodos de verossimilhança sempre que disponíveis, ou métodos de imputação múltipla paramétricos, tais como o algoritmo EM. Já MAGNANI (2004) aponta alguns possíveis problemas com a aplicação do algoritmo EM:

1) O processo consome muito tempo de processador, por ser iterativo;
2) Para usá-lo, devemos especificar a distribuição da amostra (o que quase nunca é possível no processo de Descoberta de Conhecimento em Bases de Dados);
3) Métodos estatísticos necessitam de modelos com fortes pressupostos, o que pode ser difícil no processo de Descoberta de Conhecimento em Bases de Dados.

3.5.3.1.2 O Método de Verossimilhança com Informações Completas

O método de verossimilhança com informações completas (FIML – *Full Information Maximum Likelihood*), desenvolvido por MYRTVEIT, STENSRUD e OLSSON (2001) é uma técnica que se ancora no princípio de maximização da *log*-vizinhança. Descrevemos o método a seguir.

Seja p o número de variáveis (colunas) e N o número de observações (tuplas de uma tabela). Assumindo que $y = [y_i]$, $1 \leq i \leq p$, é um vetor que corresponde a uma observação, e apresenta uma distribuição normal multivariada com média μ e matriz de covariância Σ.

Sabemos que alguns elementos y_i podem estar ausentes. Seja m_i e Ω_i respectivamente a média e a covariância das variáveis observadas para o elemento i. Estas medidas são obtidas pela remoção das colunas com valores ausentes do elemento i. Por exemplo, considere um conjunto de dados com p = 3 variáveis. Se o caso i=1 estiver completo ($y_1 = [y_{11} \quad y_{12} \quad y_{13}]^T$), sua média será $m_1 = \frac{y_{11} + y_{12} + y_{13}}{3}$, e sua matriz de covariância $\Omega_1 = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix}$. Porém, se o caso i=3 não apresentar o valor da 2ª coluna ($y_3 = [y_{31} \quad * \quad y_{33}]$), o cálculo de sua média e matriz de covariância não levarão em consideração o atributo ausente. Assim, teremos $m_3 = \frac{y_{31} + y_{33}}{2}$, e sua matriz de covariância $\Omega_3 = \begin{pmatrix} \theta_{11} & \theta_{13} \\ \theta_{31} & \theta_{33} \end{pmatrix}$.

Estes novos elementos calculados são usados no processo de estimativa. De maneira geral, a *log*-vizinhança do caso i é definida como:

$$\log l_i = K_i - \frac{1}{2}\log|\Omega_i| - \frac{1}{2}(y_i - m)'\Omega_i^{-1}(y_i - m_i)$$

A *log*-vizinhança para toda a amostra com dados preenchidos é:

$$\log L = \sum_{i=1}^{N} \log l_i$$

Dado um modelo que especifique o vetor μ e matriz de covariância Σ como funções dos seus parâmetros implica dizer que $\mu = \mu(\theta)$, e $\Sigma = \Sigma(\theta)$, onde θ é o vetor de parâmetros a ser estimado. O principal ponto a ser destacado é o fato de que o modelo é usado para prever o vetor μ e matriz de covariância Σ. Assim, formulamos as equações $\mu = \mu(\theta)$ e $\Sigma = \Sigma(\theta)$. O vetor de parâmetros θ é um elemento desconhecido com uma quantidade estocástica a ser estimada.

As estimativas de verossimilhança de θ são obtidas maximizando $\log L(\theta)$. O vetor de médias μ e a matriz de covariâncias Σ são agora funções de parâmetros θ desconhecidos no modelo teórico. Podemos imaginar nisto como um processo de

derivação onde queremos resolver a equação $\partial \log(\theta)\frac{\partial}{\theta} = 0$. O vetor de parâmetros estimados $\hat{\theta}$ é o vetor com a verossimilhança relacionada aos dados observados.

3.5.3.2 Imputação de valores ausentes com métodos Bayesianos

Trabalhos envolvendo métodos bayesianos também são encontrados na literatura, utilizadas na tarefa de complementação de dados ausentes em bases de dados. Embasado por consistente teoria estatística, soluções pertencentes a esta família de algoritmos representam uma boa parcela das soluções no processo de imputação de valores desconhecidos.

Uma rede bayesiana é um grafo acíclico direcionado onde os vértices representam as variáveis do problema, e os arcos indicam uma relação causal entre as variáveis (nós) conectadas. A intensidade de cada relacionamento (arestas) é dada por uma tabela de probabilidades condicionais.

Um exemplo de rede bayesiana é o problema de metástase de câncer (HRUSCHKA *et al*, 2002b apud PEARL, 1988, PARSONS, 1996), mostrado na figura 3.6. Os nós representam as variáveis (câncer, tumor cerebral, aumento total de cálcio no plasma, coma e dores de cabeça severas) e as ligações entre os nós mostram a influência causal entre as variáveis.

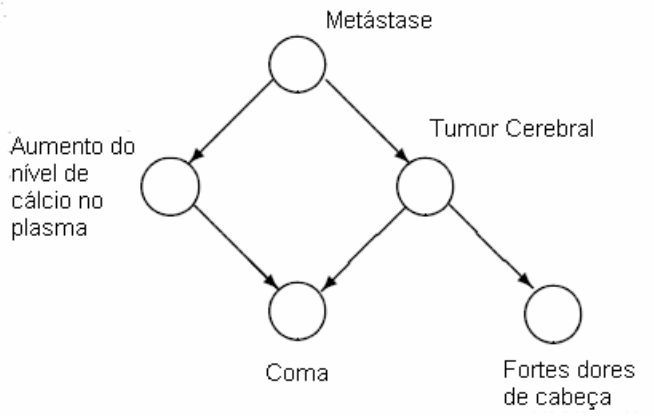


Figura 3.6 Conhecimento médico representado com uma rede bayesiana. Fonte: Adaptado de HRUSCHKA, HRUSCHKA JR. e EBECKEN (*2002b*); *e PARSONS* (*1996*).

Não existe influência causal entre as variáveis que não estão conectadas. A intensidade desta influência é dada pela probabilidade condicional $P(X_i \mid \prod_{X_i})$, onde X_i é a *i*-ésima variável, e $\prod_{X_i}$ o conjunto de nós-pai de X_i. Desta forma, a rede bayesiana

pode ser utilizada como uma ferramenta de representação do conhecimento gerando inferências.

3.5.4 Gerenciamento Direto de Dados Ausentes

Alguns métodos conseguem tratar valores ausentes em conjuntos de dados, sem a necessidade de imputação. Podemos citar como exemplo algoritmos de classificação baseados em árvores, que constróem árvores de decisão mesmo com a existência de dados ausentes.

3.5.5 Métodos Híbridos

Os métodos de imputação híbrida combinam dois ou mais métodos de imputação simples, com o objetivo de melhorar a qualidade do processo como um todo. Nesta categoria, podemos identificar duas principais técnicas: a imputação múltipla, proposta por RUBIN (1988), e a imputação composta, proposta no escopo desta tese. Detalhamos os dois casos nas subseções a seguir.

3.5.5.1 Imputação Múltipla

A imputação simples descrita anteriormente apresenta um sério problema: ele esconde a incerteza inerente ao dado imputado. Após o processo de complementação, as ferramentas de análise tratam os valores reparados como se fossem os reais, desconsiderando a sua incerteza (figura 3.7).

O processo de imputação múltipla proposto por RUBIN (1988) procura reduzir a incerteza inerente ao processo de imputação. O método produz *n* sugestões para cada atributo de um registro que contenha valores ausentes. Cada um destes *n* valores é então atribuído ao campo com valor desconhecido, criando um novo conjunto de dados, como se houvesse ocorrido uma imputação simples. Desta forma, uma única coluna de uma tupla de uma tabela gera *n* novos conjuntos de dados, que são analisados caso a caso usando métodos convencionais. Estas análises são combinadas em um segundo momento, gerando o resultado consolidado daquele conjunto de dados.

Sendo assim, as etapas do processo de imputação múltipla são as seguintes:

1) Para cada atributo que apresente valor ausente em um registro do conjunto de dados, gera-se um conjunto de *n* valores a serem imputados;

2) Realiza-se uma análise estatística em cada conjunto de dados, gerados a partir da utilização de uma das *n* sugestões de substituição geradas no item anterior;

3) Combinam-se os resultados das análises realizadas para produzir um conjunto de resultados.

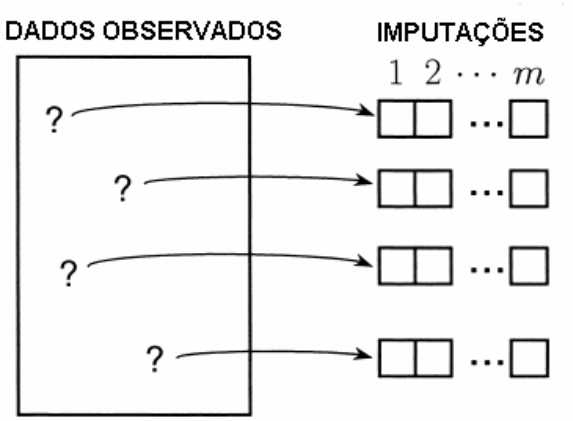


Figura 3.7 Esquema de Imputação Múltipla.
Fonte: Adaptado de SCHAFER e GRAHAM (*2002*).

ALLISON (2000) cita que o valor de *n* varia entre três e cinco, de forma heurística. Já CARTWRIGHT, SHEPPERD e SONG (2003) mencionam que o valor de *n* normalmente varia entre três e dez.

Podemos enumerar as seguintes vantagens do processo de imputação múltipla:

1) Utiliza os mesmos métodos de imputação simples;

2) Faz com que os coletores de dados reflitam sobre a incerteza dos dados imputados (que dados imputar!);

3) Mantém a consistência da base de dados entre diversas análises, já que o mesmo conjunto de dados com imputação múltipla é passado a todos os usuários.

Uma desvantagem do processo de imputação múltipla é a necessidade de o mecanismo de ausência dos dados ser aleatório (MAR). Esta é uma séria limitação no uso da técnica, pois nem sempre podemos garantir que isto aconteça (CARTWRIGHT *et al*, 2003).

YUAN (2000) cita um exemplo de sistema comercial de análise de dados em Inteligência Competitiva, o SAS, que implementa a imputação múltipla. A sua implementação do algoritmo EM é chamada PROC MI, que cria múltiplas sugestões

para dados multivariados *p*-dimensionais incompletos. Ele usa um método que analisa as variações das *n* imputações. Uma vez que os *m* conjuntos de dados reparados são analisados utilizando rotinas padrão, um outro procedimento, chamado PROC MIANALYZE, pode ser usado para gerar inferências estatísticas válidas sobre os parâmetros gerados, combinando os resultados dos m conjuntos de dados completos.

RUBIN (1988) destaca que pesquisas sobre como produzir uma moderna técnica de imputação múltipla de dados para a técnica *hot-deck* são necessárias.

WAYMAN (2003) cita que a imputação múltipla tem mostrado produzir estimativas de parâmetros não tendenciosos, que refletem a incerteza associada com as estimativas de dados ausentes.

Optamos por enquadrar a imputação múltipla na categoria de *métodos híbridos*, (categoria proposta nesta tese), e não um tipo de imputação, pois as múltiplas sugestões podem ser feitas não apenas por métodos de imputação simples, mas também por métodos baseados em modelos de dados, métodos convencionais, ou mesmo por outros métodos híbridos.

3.5.5.2 Imputação Composta

Também proposta no escopo desta tese, a imputação composta representa uma classe de técnicas de imputação que combinam uma ou mais tarefas usadas na etapa de Mineração de Dados para gerarem um novo valor a ser imputado. Por ser objeto de estudo desta tese, detalharemos melhor seus princípios no capítulo 4.

3.6 Métodos de tratamento de dados ausentes

Na seção 2.7, analisamos em detalhes no escopo da etapa de pré-processamento de dados do processo de Descoberta de Conhecimento em Bases de Dados os algoritmos de imputação por média ou moda, a utilização do algoritmo dos *k* vizinhos mais próximos e as redes neuronais *back propagation*. Todavia, estes não são os únicos métodos disponíveis para a complementação de dados desconhecidos em tabelas de um conjunto de dados. Nesta seção, abordaremos as diversas soluções existentes na literatura para tratar este problema.

Quando tratamos de atributos categóricos, além da média e da moda, outra opção é associar todos os possíveis valores ao item ausente, ou considerar o valor *nulo* com um

dos possíveis do domínio do atributo. Assim, este atributo pode ser tratado pelos métodos de mineração de dados (BROWN, KROS, 2003a, MAGNANI, 2004).

Uma parte dos algoritmos de tratamento de dados ausentes é classificada como *baseados em instâncias* (HRUSCHKA *et al*, 2003b). Algumas de suas características são:

- Não geram um modelo de predição (não existe uma fase de treinamento);
- Realizam uma busca em toda a base para estimar a melhor instância a ser usada;
- Apresentam um alto custo computacional;
- Podem produzir resultados mais apurados, já que o processo de aprendizado é específico para cada consulta.

BATISTA e MONARD (2003a) citam também os *modelos preditivos*, onde o atributo que possui valores ausentes é visto como um atributo de classe/regressão, e os demais atributos são entradas do processo de classificação/regressão. Os autores citam como vantagem da utilização destes métodos a freqüente correlação existente entre os atributos, que pode ser usada para criar um modelo de classificação/regressão para dados ausentes. Como desvantagens, vemos que os valores previstos pelo modelo são mais bem comportados do que os reais. Além disso, esta abordagem pressupõe que existam correlações entre os dados. Se não existirem, o modelo pode não ser preciso na previsão. HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003a) enumeram outros métodos de tratamento de valores ausentes, tais como as árvores de decisão e os métodos bayesianos.

Abordamos nesta seção os métodos mais comuns. A próxima seção trata dos trabalhos relacionados que estão disponíveis na literatura. Assim, as diversas outras opções de algoritmos de tratamento de valores ausentes poderão ser melhor analisadas.

3.7 Trabalhos relacionados

Dados relativos a respostas de pesquisas (*likert data*) são dados categóricos, ordinais, que representam o nível de concordância/discordância de um entrevistado com uma determinada pergunta. Por exemplo, questões do tipo “Qual o seu grau de satisfação com o serviço prestado?” podem apresentar respostas tais como “Muito Satisfeito/Satisfeito/Parcialmente Satisfeito/Insatisfeito/Totalmente Insatisfeito”.

Métodos de preenchimento de dados ausentes desta natureza foram estudados por JÖNSSON e WOHLIN (2004) e JÖNSSON e WOHLIN (2006).

O primeiro dos dois trabalhos (JÖNSSON, WOHLIN, 2004) propõe um processo de complementação de dados combinando uma versão alterada da remoção de casos completa com a aplicação do algoritmo dos *k* vizinhos mais próximos em dados relativos a respostas de pesquisas, em um contexto de Engenharia de Software. Apesar de a grande maioria dos trabalhos encontrados na literatura desaconselhar o uso da remoção completa de casos, os autores a utilizam como uma primeira etapa do processo de imputação, alegando que tuplas que possuam poucos valores preenchidos em seus atributos não têm muita contribuição a oferecer no processo de imputação. Todavia, esta remoção pode se dar de tal forma que os conjuntos de dados se tornem tão pequenos que inviabilizem um processo de imputação eficaz. Assim, o trabalho define dois parâmetros mínimos, chamados *limite de redução de casos* e *limite de redução do conjunto de dados*.

A seguir, os autores propõem que não só os casos completos sejam utilizados no algoritmo dos *k* vizinhos mais próximos. Em sua forma clássica, o algoritmo utiliza todas as variáveis no cálculo da distância entre dois objetos. Eles chamam esta abordagem de casos completos (CC – *complete cases*). Todavia, o trabalho sugere que não só estas instâncias do conjunto de dados sejam utilizadas, mas também todas as tuplas que contenham valores que o objeto sendo imputado apresente valores, além de possuírem valores no atributo que está sendo imputado. Esta estratégia é chamada de casos incompletos (IC – *incomplete cases*). Para exemplificar esta idéia, considere a tabela abaixo:

id	a_1	a_2	a_3	a_4
1	3	8	6	4
2	*	5	3	*
3	1	*	*	2
4	*	3	*	3
5	*	7	1	6

Se quisermos regredir o atributo a_3 do registro de identificador 4, a abordagem *CC* retornaria a tupla 1. Já a abordagem *IC* retornaria apenas a tupla 5, pois ela apresenta valores completos nos mesmos atributos da tupla 4 (a_2 e a_4), além do atributo a ser

imputado (a_3). A tupla 2 não foi selecionada pela abordagem *IC* pois não apresenta valor no atributo a_4, e a tupla 3 possui valor ausente no atributo a_2.

Os autores consideraram que os dados possuem mecanismo de ausência completamente aleatório (MCAR), e não consideram no processo de imputação do atributo a_i objetos que já possuam dados imputados em outros atributos a_j, $j \neq i$.

O trabalho chega às seguintes conclusões:

1) A imputação de dados de pesquisas com o algoritmo *k*-NN é viável. Os resultados mostram que a complementação de dados produz bons resultados qualitativos.

2) Por observação, os autores mencionam que nenhum experimento teve bons resultados com o valor de $k = 1$. Também de forma heurística, eles sugerem que os melhores resultados foram obtidos com o valor de *k* aproximadamente igual à raiz quadrada dos casos válidos, arredondada para o inteiro ímpar mais próximo.

3) A qualidade da imputação depende mais do número de casos completos do que do percentual de valores ausentes.

Em JÖNSSON e WOHLIN (2006), os autores estendem o estudo anterior, comparando o algoritmo dos *k* vizinhos mais próximos modificado com quatro outros métodos:

1) **Substituição Aleatória** (RDS – *Random Draw Substitution*): método de imputação que complementa um valor ausente a partir de um dos valores do domínio de respostas. Esta solução foi implementada neste trabalho, gerando com igual probabilidade um valor inteiro entre 1 e 5. Este método não considera que os dados respeitem uma distribuição probabilística nem nenhuma outra propriedade relevante.

2) **Imputação Aleatória**: os dados são aleatoriamente atribuídos não mais considerando todo o domínio de possíveis respostas, mas somente daquelas que ocorreram. Assim, se de cinco opções possíveis, apenas quatro delas foram marcadas, somente essas quatro são usadas como base para a imputação neste caso.

3) **Imputação por média**: a média de todas as respostas é utilizada para substituir valores ausentes na resposta do questionário. Conforme já discutido, isto reduz a variância dos dados, podendo torná-los tendenciosos.

4) **Imputação por moda**: o valor que mais aparece como resposta é escolhido para substituir os dados ausentes de uma pergunta.

Os resultados mostram que o método dos k vizinhos mais próximos teve um bom desempenho, mesmo quando muitos dados estavam ausentes. Porém, os resultados também foram animadores com o uso da média e da moda para os dados de respostas de pesquisas (*likert data*), alvo de estudo do trabalho. Além disso, o algoritmo dos *k* vizinhos mais próximos teve melhor desempenho com mais atributos.

BATISTA e MONARD (2001) destacam que os problemas com a qualidade dos dados que são enviados ao processo de Mineração de Dados são normalmente mais complexos do que aqueles encontrados em repositórios de dados. Por esta razão, a forma como os algoritmos de aprendizado de máquina que são utilizados para este fim tratam esta questão deve ser revista, já que soluções mais simplistas, tais como a substituição de um valor ausente pela sua média ou moda só é aplicável a conjuntos de dados com poucos registros e com dados completamente aleatórios. Estas pré-condições, se não respeitadas, levam os sistemas especialistas a descobrirem padrões inválidos a partir destes dados. Desta forma, um conjunto de dados com muitos valores ausentes ou que apresentam um mecanismo de ausência não aleatório deve ser tratado com outros métodos de imputação, e utilizam o algoritmo dos *k*-vizinhos mais próximos para tratar os valores desconhecidos. Os resultados são comparados com o algoritmo C5.0, uma variante do algoritmo C4.5 de QUINLAN (1993).

SONG, SHEPPERD e CARTWRIGHT (2005) mencionam que, para resolver o problema de dados ausentes em modelos de Engenharia de Software, vários métodos estão à disposição. Todavia, escolher um ou mais deles pode ser uma tarefa difícil, já que os modelos de predição assumem que a ausência se enquadra em algum mecanismo que nem sempre é possível identificar em problemas desta natureza, onde caracteristicamente encontramos poucos dados. Por esta razão, os autores experimentam a imputação com média na classe (CMI – *Class Mean Imputation*) e o algoritmo dos *k*-vizinhos mais próximos, assumindo em momentos distintos que os dados são completamente aleatórios (MCAR) e aleatórios (MAR), e chega à conclusão de que a

imputação por média na classe, para problemas de Engenharia de Software, é mais adequada por ser mais precisa. Todavia, os autores mencionam que as duas técnicas têm aplicações práticas para conjuntos de dados pequenos de Engenharia de Software com valores ausentes. Outro destaque é a da não importância estatística do mecanismo de ausência de dados, que, por esta razão, podem ser sempre assumidos como sendo aleatórios (MAR).

Encontramos na literatura algumas soluções para o tratamento de valores ausentes utilizando regras de associação. Como exemplo, o trabalho de CRÉMILLEUX, RAGEL e BOSSON (1999) e RAGEL e CRÉMILLEUX (1999), que apresentam o algoritmo MVC – *Missing Values Completion*. Esta técnica exige a intervenção do usuário no processo de complementação de dados, pois monta um conjunto de regras de associação e as submete para apreciação. O usuário decide qual regra ou quais regras utilizar, baseado em alguns parâmetros, onde os principais são a confiança e o suporte de cada uma das regras.

O algoritmo MVC utiliza um outro algoritmo proposto anteriormente, chamado RAR – *Robust Association Rules* (RAGEL, CRÉMILLEUX, 1998). Esta técnica descobre regras de associação em *conjuntos de dados de dados válidos*, ou seja, o maior subconjunto de dados que não contém valores ausentes em seus registros. Estas regras são obtidas sobre um subconjunto dos dados que não possui nenhum valor ausente, chamado *vdb* (*valid database*).

Com as regras geradas no passo anterior, o algoritmo MVC decide qual dado imputar, baseando-se nos conseqüentes das regras. Isto gera dois tipos de situações:

1) Todas as regras chegam à mesma conclusão: assume o valor que aparece no conseqüente;
2) Intervenção do usuário: baseado principalmente na confiança e o suporte de cada regra, mas também utilizando outros parâmetros, o usuário decide que regra usar, e assume o conseqüente dela como o valor a ser imputado.

Cabe destaque para algumas situações interessantes:

1) A regra de maior confiança indica um valor, e todas as demais concordam em outro valor
2) O sistema gera uma única regra

O método trabalha apenas com valores discretos, mas não trata a forma de discretização dos atributos numéricos.

TWALA, CARTWRIGHT e SHEPPERD (2005) comparam oito métodos de tratamento de dados ausentes: 1) Remoção Completa de Casos; 2) Imputação simples com os algoritmos DTSI (*Decision Tree Single Imputation*), *k* vizinhos mais próximos, Média ou Moda e de verossimilhança EM (*Expectation-Maximization*); 3) Imputação Múltipla com os algoritmos NORM (Dados contínuos multivariados), CAT e MIX (Dados contínuos e categóricos) e PNA (Dados agrupados)[1]; e 4) Métodos de Aprendizado de Máquina, com o algoritmo C4.5 (QUINLAN, 1993) e SVS – *Surrogate Variable Splitting*.

O algoritmo DTSI (*Decision Tree Single Imputation*), proposto no escopo deste trabalho, constrói uma árvore de decisão para prever o valor de um atributo x_i, fazendo com que ele passe a ser a base de formação da árvore, ou seja, o atributo x_i troca de papel com o atributo-classe na construção da árvore (este último passa a ser um dos atributos de entrada para a construção da árvore). Métodos baseados em árvores não fazem nenhuma suposição sobre a forma de distribuição dos dados, e não exigem uma especificação estruturada do modelo.

Métodos não paramétricos, tais como as árvores de decisão, são capazes de alcançar estimativas ótimas para qualquer combinação de dados, à medida que mais dados são observados. Esta característica não existe em métodos paramétricos.

Os testes apresentados pelos autores variaram os seguintes parâmetros: 1) percentual de valores ausentes no conjunto de treino e testes; 2) diferentes padrões de ausência de dados; 3) diferentes mecanismos de ausência de dados; e 4) diferentes percentuais de valores ausentes.

A base original utilizada nos testes composta por dados sobre projetos de software está completa, e os valores ausentes foram gerados artificialmente. Todos os atributos são categóricos. Os contínuos foram discretizados, e as instâncias são independentes. Os mecanismos de ausência implementados foram o completamente aleatório (MCAR), o aleatório (MAR) e o não aleatório (NMAR), com dados ausentes em um ou mais

[1] Os quatro programas acima pertencem ao pacote estatístico SPLUS (BECKER *et al*, 1988), e utilizam previamente o algoritmo EM.

atributos. O percentual de valores ausentes gerados foi de 15%, 30% e 50%. O mesmo percentual de valores ausentes existia no conjunto de treino e de testes. Para criar as ausências, os autores criaram um vetor com indicadores de ausência (valor zero) ou presença (valor um) dos atributos. Multiplicaram-se em seguida os atributos pelo vetor de indicadores.

O treinamento foi feito com 5-*fold cross validation* (quatro para treino, e uma para testes). No algoritmo dos *k* vizinhos mais próximos, o valor de *k* variou entre 1, 3, 5, 11, 15 e 21 vizinhos. A análise centrou-se não na taxa de erro, mas na sua variação em função das condições impetradas.

Os resultados apontam melhor desempenho da imputação múltipla com o algoritmo EM. O pior caso apresentou-se com a remoção completa de casos, por reduzir drasticamente o tamanho da amostra. Porém, os autores muito apropriadamente destacam que selecionar métodos de tratamento de dados ausentes é tarefa difícil, pois o mesmo procedimento pode ter um melhor desempenho em certas circunstâncias, e em outras não.

A baixa correlação dos atributos pode ter causado o fraco desempenho dos algoritmos de árvores. Outra razão que pode também explicar o problema com as árvores de decisão é a ausência de valores do pseudo-atributo classificador (o que possui valores ausentes) no conjunto de testes.

A qualidade da previsão de métodos diminui à medida que os valores ausentes crescem. Assim, os resultados do trabalho mostram, segundo os autores, que o mecanismo e o padrão de dados ausentes têm impacto no desempenho dos métodos, especialmente nos níveis mais baixos de ausência. Entretanto, à medida que a proporção de dados ausentes cresce, o fator determinante no desempenho dos métodos é como os valores ausentes estão distribuídos entre os atributos.

Os melhores resultados de predição foram obtidos com dados completamente aleatórios (MCAR), seguidos de dados aleatórios (MAR), e, por fim, dos não aleatórios (NMAR). Esta relação de precedência está de acordo com a teoria estatística de que dados completamente aleatórios são mais simples de tratar, e os dados não aleatórios os mais difíceis. Adicionalmente, os autores destacam que assumir que o mecanismo de ausência dos dados aleatório é mais razoável do que acreditar que eles são completamente aleatórios. Ainda, métodos que tratam valores aleatórios servem para

tratar valores completamente aleatórios, mas o contrário não ocorre. Isto deve explicar o bom desempenho do método de imputação múltipla com o algoritmo EM – um método que toma por base que o mecanismo dos dados é o de aleatoriedade (MAR), e o desempenho ruim da remoção completa de casos, um método que assume que o mecanismo dos dados é o completamente aleatório (MCAR). Além disso, o trabalho salienta que o impacto do manuseio de valores ausentes não depende apenas do fato de esta ausência acontecer no conjunto de treino ou de testes isoladamente, mas na combinação dos dois.

BATISTA e MONARD (2003a) analisam o desempenho do algoritmo dos *k* vizinhos mais próximos (*k*-NN) como um método de imputação, comparando-o com os algoritmos de imputação média ou moda, C4.5 (QUINLAN, 1993) e CN2 (BOLL, ST. CLAIR, 1995). O primeiro gera uma árvore de decisão sobre um atributo-classe. Já o segundo gera regras de associação, assumindo o valor da moda nos atributos ausentes antes de medir a entropia. Estes dois algoritmos lidam bem com dados ausentes no conjunto de treinamento e testes, exceto para o atributo-classe.

Desta forma, os autores analisam se o prévio tratamento dos dados em um conjunto que possui valores ausentes melhora o processo de classificação dos dados. Este tratamento é feito de duas diferentes formas: uma parte do conjunto de dados tem os valores ausentes preenchidos com o algoritmo *k*-NN, e o outro subconjunto com a média ou moda. Cada um destes conjuntos é então enviado aos algoritmos C4.5 e CN2 para classificação, e seu desempenho comparado com estes dois algoritmos classificando com valores ausentes no conjunto de testes.

A base, inicialmente completa, tem valores sujos artificialmente, com percentuais de ausência variando entre 10% e 60%, com saltos de 10%. Os autores também sujaram um, dois e três atributos nas bases, gerando diferentes conjuntos de dados, resultado da combinação de percentual e número de atributos de valores ausentes. Os dados ausentes são MCAR. O número de vizinhos utilizados no *k*-NN foram 1, 3, 5, 10, 20, 30, 50 e 100.

Assim, este conjunto foi dividido em 10 partes para treino e testes, para a validação com a técnica de *resampling* validação cruzada com 10 conjuntos (*10-fold cross validation*), e atributos com valores ausentes foram inseridos no conjunto de treinamento. Seis cópias deste conjunto foram divididas da seguinte forma, para

tratamento dos dados ausentes: duas cópias foram dadas ao algoritmo C4.5 e ao algoritmo CN2, sem nenhum tratamento prévio dos dados; outras duas cópias tiveram os valores ausentes tratados pelo algoritmo *k*-NN; e as últimas duas cópias tiveram seus valores ausentes preenchidos com média ou moda. Depois do tratamento dos dados ausentes, os conjuntos de testes foram oferecidos aos algoritmos C4.5 e CN2. As demais quatro partes dos conjuntos de dados foi utilizada para testar os classificadores, e a taxa de erro na classificação calculada. Este processo se repetiu por 10 vezes, e a taxa real de erro é obtida com a média das taxas de erro de cada um dos 10 passos.

Os melhores resultados foram alcançados com a aplicação do *k*-NN com 10 vizinhos. O algoritmo C4.5 gradualmente descarta atributos com valores ausentes, à medida que esta ausência aumenta. O C4.5 também descarta valores que foram imputados com média ou moda, já que eles diminuem o seu poder discriminatório (entropia). Os modelos de classificação tornaram-se mais simples à medida que mais atributos com valores ausentes foram inseridos e com o aumento percentual de dados incompletos. Assim, a imputação prévia de dados ausentes pode evitar que o modelo gerado de um conjunto de dados torne-se extremamente simplificado.

Os autores chamam a atenção para os resultados obtidos na base *Breast Cancer*. As características da base – de alta correlação entre os atributos – podem introduzir questões tais como se realmente os atributos devem ser tratados com algum método de imputação.

Em BATISTA e MONARD (2003b), os autores complementam o trabalho anterior, realizando um estudo sobre as características dos dados que podem levar a um desempenho ruim do método de imputação baseado no algoritmo dos *k* vizinhos mais próximos. A metodologia foi a mesma do artigo anterior, com as bases de dados *Wisconsin Breast Cancer*, *Sonar* e *TA*, todas do repositório da Universidade da Califórnia, Irvine (NEWMAN, 1998).

Neste experimento, os autores decidiram inserir valores desconhecidos nos atributos mais representativos de cada conjunto de dados para que se pudesse medir a efetividade dos métodos de tratamento de valores desconhecidos. Tal efetividade não pode ser medida se os atributos tratados forem não representativos, os quais provavelmente não seriam incorporados ao classificador pelo sistema de aprendizado.

Os resultados produzidos pelo conjunto de dados *Breast Cancer* são interessantes, pois o método 10-NNI foi capaz de predizer os valores desconhecidos com uma precisão superior a imputação pela média ou moda.

Quando o método 10-NNI é utilizado, o indutor C4.5 mantém os atributos com valores desconhecidos como os atributos mais próximos da raiz da árvore de decisão. Essa situação poderia ter sido uma vantagem se o conjunto de dados Breast não possuísse outros atributos com poder de predição similar aos atributos selecionados.

O conjunto de dados *Sonar* possui características similares ao conjunto de dados *Breast Cancer*, uma vez que seus atributos possuem forte correlação entre si. Outra característica interessante do conjunto de dados *Sonar* é que ele possui uma grande quantidade de atributos, 60 no total. Essa grande quantidade de atributos pode fornecer ao indutor diversas possibilidades durante a escolha dos atributos que irão compor o classificador.

Diferentemente do conjunto de dados *Breast Cancer*, o método 10-NNI foi capaz de superar o sistema C4.5 em duas situações: quando os valores desconhecidos foram inseridos no atributo 10 e nos atributos 10, 0 e 26. Para o indutor CN2, não é possível dizer que um dos métodos foi superior aos demais.

No conjunto de dados *TA*, a imputação não foi boa, pois a correlação entre os seus atributos é fraca. O modelo de predição também não foi capaz de incorporar as relações existentes entre os atributos. É possível observar que o método de imputação 10-NNI não foi capaz de superar os demais métodos de tratamento de valores desconhecidos.

Os autores chegam à conclusão de que quanto mais atributos com valores desconhecidos e quanto maior taxa de atributos ausentes, mais simples são os classificadores induzidos.

TSENG, WANG e LEE (2003) combinam imputação e agrupamento para tratar o problema de valores ausentes numéricos. Em um conjunto de dados, primeiramente realiza-se a complementação dos dados ausetes considerando todos os registros da tabela, a fim de obter estimativas menos precisas para os dados desconhecidos. Depois, um algoritmo de agrupamento partitivo cria grupos disjuntos. Para cada conjunto, a imputação é aplicada novamente, para gerar novamente os valores ausentes, considerando agora apenas a informação dos registros do grupo. A técnica foi batizada de *RegressionClustering* (RC), que funciona da seguinte forma:

1) Dividir o conjunto de registros D em dois subconjuntos: D_C (dados completos) e D_M (dados ausentes). Completar D_M com um algoritmo de imputação baseado em D_C, obtendo o conjunto D'.

2) Agrupar os elementos de D' em k grupos $C_1, C_2, \ldots, C_K$ ($\sum |C_i| = |D'|$)

3) Para cada grupo C_i, a imputação é aplicada a todos os registros R_j, tais que $R_j \in D_M \cap C_i$. A base usada para a imputação é o conjunto $\{R_C \mid R_C \in D_M \cap C_i\}$

Os autores destacam que a imputação global é uma solução que visa não perder os registros com valores ausentes, já que o algoritmo de agrupamento não lida com valores ausentes nos registros.

O algoritmo de agrupamento usado é chamado CAST em TSENG, WANG e LEE (2003) que, segundo os autores, oferece resultados mais eficientes e precisos do que o algoritmo de agrupamento dos *K*-centróides (seção 2.3.2.6).

O método RC foi comparado com resultados gerados pelo algoritmo EM, regressão (que os autores não informam claramente qual é), média, e com o número de grupos do algoritmo de agrupamento dos *K*-centróides variando entre 3 e 48, com saltos de 3 unidades.

Os valores gerados pelos algoritmos foram mensurados por uma medida de erro, chamada erro relativo absoluto (RAD – *Relative Absolute Deviation*), que é calculado da seguinte forma:

$$RAD = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| X_O^i - X_R^i \right|}{X_O^i}$$

onde X_O^i é o valor original do atributo X da tupla i, X_R^i é o valor imputado do atributo X nesta tupla i, e n é o total de tuplas com valores ausentes no atributo X.

Os experimentos foram executados sobre duas classes de bases de dados: uma baseada em grupos, e outra gerada aleatoriamente. Na primeira base, existiam 5000 registros, quatro grupos e dez atributos. O método proposto pelos autores sobressaiu-se sobre os demais. O pior desempenho foi obtido com a aplicação do algoritmo dos *K*-centróides. À medida que o percentual de valores ausentes aumentava, o algoritmo de agrupamento tornava-se pior e a regressão permanecia estável. O método proposto apresentou leve piora de desempenho em níveis mais altos de valores ausentes. Os

autores justificam a queda de desempenho de seu método alegando que a precisão da imputação é determinada pela distribuição dos dados, e não pelo percentual de valores ausentes. Na base gerada aleatoriamente, em geral todos os métodos apresentaram um desempenho inferior à base agrupada. Os métodos de agrupamento (o *K*-Means e o *RegressionClustering*) se saíram melhor do que os demais. Porém, os autores concluem que é difícil regredir dados em uma base gerada randomicamente.

Ao avaliar este trabalho, observamos os seguintes problemas:

1) O trabalho não especifica o mecanismo de ausência dos dados;
2) O percentual de valores ausentes vai até 20%, um índice baixo quando comparado a valores reais;
3) Os autores utilizam para seus testes uma base aleatoriamente gerada, que não existem na realidade;
4) Os autores não utilizam uma base *benchmark* reconhecida;
5) O trabalho não especifica a forma de inicialização dos centróides do algoritmo de agrupamento;
6) Como os autores complementam os valores ausentes de forma global antes de agruparem, esta regressão pode induzir à criação de grupos tendenciosos, principalmente se várias colunas tiverem os seus valores imputados.

HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003a) propõem a imputação de dados ausentes em um conjunto de dados com a utilização do vizinho mais próximo (algoritmo *k*-NN com $k = 1$), precedida do agrupamento de objetos baseado em algoritmos genéticos proposto pelos autores, chamado CGA – *Clustering Genetic Algorithm*. Neste trabalho, os experimentos foram realizados na base *Wisconsin Breast Cancer*, do repositório da Universidade da Califórnia, Irvine (NEWMAN *et al*, 1998). Apenas um atributo apresenta valores ausentes por vez, e todos são numéricos. A base utilizada possui um atributo classificador, que indica se o paciente está ou não com a doença. Este atributo foi desconsiderado na formação do grupo pelo algoritmo CGA. Para fins de comparação, os autores também realizam a imputação com a utilização da média aritmética simples. Outro fator de avaliação da qualidade do dado imputado realizada é a classificação das tuplas com valores preenchidos, e o cálculo da taxa média de acertos, comparada com a taxa média de classificação dos dados originais. Os

resultados mostram que o algoritmo do vizinho mais próximo (NN – *Nearest Neighbor*) foi melhor do que a média em todos os casos. A taxa média de classificação das tuplas imputadas também foi bastante satisfatória com o algoritmo NN, obtendo taxas de acerto variando entre 94.44% e 95.46%.

HRUSCHKA, HRUSCHKA JR e EBECKEN (2003b) comparam a substituição de valores ausentes com o método do vizinho mais próximo (caso particular do *k*-NN, com *k*=1), variando a forma como a distância é calculada. Eles utilizam a distância Euclidiana e a *Manhattan*. Os resultados foram comparados com a imputação com a média. Se várias tuplas apresentarem a mesma distância mínima, utiliza-se a primeira do grupo.

A medida de desempenho é feita com a média dos erros em cada método. As bases utilizadas foram a *Iris Plants*, *Wisconsin Breast Cancer*, *Pima Indians Diabetes* e *Wine Recognition* (NEWMAN *et al*, 1998). Todas estas bases possuem atributos numéricos e um atributo-classe, que foi descartado no processo de imputação. Este descarte é feito considerando-se que o objetivo do trabalho é analisar os efeitos do aprendizado não supervisionado no contexto do agrupamento de dados. As bases foram regredidas tanto em sua forma original quanto após serem normalizadas. O trabalho não leva em consideração o mecanismo de ausência dos dados (MCAR, MAR ou NMAR). A base *Wisconsin Breast Cancer* tem um atributo-classe com dois possíveis valores (benigno ou maligno). Além disso, os registros dessa base são linearmente inseparáveis.

Os resultados mostram que, na maioria dos casos, o método NN fornece melhores resultados do que a substituição por média. Além disso, os autores indicam que os melhores resultados são obtidos em bases normalizadas. Os melhores resultados foram alcançados com o algoritmo NN baseado na distância *Manhattan* (seção 2.6.2.1) e em bases normalizadas, isto porque a distancia Euclidiana (seção 2.7.3.2) é mais influenciada por valores fora da distribuição dos dados (*outliers*). No cálculo da distância entre as tuplas, o atributo a ser imputado não é considerado.

A medida de erro não é a única fonte de avaliação da qualidade do processo de imputação. Para verificar se os dados gerados pelo processo de imputação distorcem ao mínimo a relação existente entre as variáveis do conjunto de dados, o conjunto de dados reparado que apresentou o melhor resultado – base normalizada com distância *Manhattan* – foi dividido em grupos com o algoritmo de

agrupamento dos *K*-centróides, e comparado com os grupos gerados nos dados originais. Os resultados indicam que o método proposto (NN) gera bons resultados, ou seja, preserva as relações entre as variáveis no processo de agrupamento. Assim, o método de substituição com o vizinho mais próximo é adequado para a preparação de dados para o algoritmo dos K-centróides.

O trabalho de MAGNANI e MONTESI (2004) propõe um método qualitativo de imputação local que leva em consideração todos os registros de uma tabela, e não somente os que possuem todos os seus atributos preenchidos. A solução apresentada pelos autores busca aproveitar ao máximo toda a informação contida nos registros do conjunto de dados, desta forma evitando ao máximo tornar o conjunto de dados tendencioso.

A base da proposta deste trabalho é definir um novo algoritmo de agrupamento que não desperdice os atributos que estão preenchidos em tuplas que possuem valores ausentes. Com isso, o processo de agrupamento aproveita todos os dados existentes, mesmo os de tuplas com dados incompletos, o que minimiza a perda de informação. Este processo encontra-se no contexto da técnica de imputação local *hot-deck* (FORD, 1983). Algoritmos de agrupamento geralmente assumem que os registros já foram pré-processados, e todos os atributos estão completos.

O método considera que um conjunto de dados representa objetos. Estes, por sua vez, pertencem a conceitos. Conceitos são estritos (possuem uma função característica), e são organizados em classes hierárquicas ou nebulosas (funções membro). Um conceito possui uma única descrição baseada na relevância dos seus atributos. Alguns erros no conjunto de dados não mudam a estrutura do conceito de uma forma significativa.

O algoritmo de agrupamento utiliza uma função de similaridade descrita como:

$$f(x_{ik}, x_{jk}) = \frac{\sum_{k=1}^{m} w_k * sim(x_{ik}, x_{jk})}{\sum_{k=1}^{m} w_k},$$

onde:

$$sim(x_{ik}, x_{jk}) = \begin{cases} 0, se(x_{ik} \neq x_{jk}) \ (para\ atributos\ categóricos) \\ q_k, se(x_{ik} = ?) \vee (x_{jk} = ?) \\ 1, se(x_{ik} = x_{jk}) \\ 1 - \dfrac{\left| x_{ik} - x_{jk} \right|}{\left| \max_{\in[1,n]}(x_{pk}) - \min_{\in[1,n]}(x_{pk}) \right|}, caso\ contrário \end{cases}$$

O parâmetro *q* permite que o algoritmo seja mais ou menos restritivo com a importância dos valores ausentes dos registros.

O artigo utiliza para efetuar a imputação uma média chamada central[2], calculada para valores numéricos ausentes, em tuplas dentro de um grupo, que é menos sensível a valores completos do atributo muito distintos dos demais. Esta média é calculada segundo a fórmula 100*(1-t)%, onde *t* é a tolerância fornecida como parâmetro. Como exemplo, se temos uma tolerância $t = 0.03$, valores maiores e menores do que 1.5% são descartados para efeito do cálculo da média. Este parâmetro *t* também é utilizado para tratar valores ausentes.

CARTWRIGHT, SHEPPERD e SONG (2003) avaliam duas técnicas de imputação de dados no contexto de projetos de Engenharia de Software: a média amostral e o algoritmo do *k*-vizinhos mais próximos (*k*-NN), e os compara com a remoção completa de casos (*listwise deletion*).

O trabalho utiliza duas bases de projetos de softwares, uma proveniente de um banco de investimentos, com 17 tuplas, e outra de uma multinacional, com 24 tuplas. Ambas as bases possuem nos experimentos 15% e 18% de dados ausentes. Os autores alegam que estas condições são desafiadoras para técnicas de imputação. Os dados das bases são na maioria numéricos, mas existem também dados categóricos.

A escolha do modelo de imputação deve ser compatível com a análise a ser feita e preservar as relações entre os atributos das tuplas da tabela.

Quando utilizando o algoritmo *k*-NN, os autores regridem os dados ausentes com um valor de $k = 2$ e utilizando a distância Euclidiana que, segundo os autores, produz os melhores resultados. O algoritmo *k*-NN foi modificado para tratar também dados categóricos.

O trabalho busca verificar a qualidade do dado imputado com um modelo de regressão chamado *stepwise regression*. Este modelo é utilizado, pois a premissa é que o mecanismo de ausência dos dados utilizados é o aleatório (MAR). Com isso, considerando um atributo ausente X_i, podemos obter o seu valor a partir dos demais atributos independentes X_j, $i \neq j$, da seguinte forma:

$$X_i = \sum_{j=1}^{n} \beta_j * X_j, \; j \neq i$$

[2] Do inglês *mean_trimmed*

Neste procedimento, as variáveis independentes que são consideradas para integrar a equação são aquelas com maior correlação positiva ou negativa com a variável dependente (no caso, X_i), em ordem decrescente. Cada variável, antes de ser de fato integrada à equação, passa por um teste de validação de hipóteses.

O trabalho chega à conclusão de que o algoritmo *k*-NN apresentou uma melhor desempenho do que a utilização da imputação com a média, e ambas as opções são melhores do que deixar a base sem tratamento. Porém, os autores mencionam que não há um resultado conclusivo, e que mais testes precisam ser feitos. Mostramos nesta tese que todos os experimentos realizados mostram que o algoritmo dos *k* vizinhos mais próximos supera em qualidade os resultados gerados pela imputação com a média.

O trabalho de LAKSHMINARAYAN, HARP e SAMAD (1999) complementa dados de uma base com informações sobre processos industriais, tais como os de refinarias de óleo, fábricas químicas e plantas farmacêuticas. Os autores destacam que bases com dados relativos a estes processos normalmente possuem muitos dados ausentes, principalmente por conta de omissões na entrada manual dos dados. A ausência comumente ultrapassa a ordem de 50% dos valores, o que torna bastante difícil o processo de imputação. Os autores sugerem mecanismos que evitem tantos valores incompletos, como por exemplo o uso de códigos de barras, apesar de admitirem que sempre haverá a necessidade de preenchimento manual de dados.

Assim, a proposta deste trabalho é a de utilizar uma base contendo dados sobre a manutenção de dispositivos de controle. Os valores de 82 variáveis, algumas categóricas, outras numéricas, registram propriedades dos dispositivos, tais como o seu fabricante e modelo, ou o valor de vários parâmetros de erro de calibragem ou resultados aprovação ou não de testes de qualidade. Todavia, o foco do trabalho é o tratamento de dados categóricos. Para os autores, atributos numéricos não são relevantes na base de dados. De um total de 4.383 registros, nenhum estava completo, e somente 33 variáveis têm mais do que 50% dos valores completos.

O trabalho não descarta, para o tratamento de dados ausentes, a possibilidade de remoção de registros ou de colunas. Esta opção é usada quando estes registros constituem uma parcela insignificante do total dos dados, e, neste caso, os autores alegam que nenhuma distorção relevante é introduzida com esta eliminação. Para a imputação de dados, os autores utilizaram os algoritmos C4.5 (QUINLAN, 1993) e

AutoClass (CHEESEMAN, STUTZ, 1996), este último um método de agrupamento utilizado para descobrir estruturas intrínsecas nos dados, baseado na teoria de classificação Bayesiana. Além disso, os autores optaram por utilizar tanto a imputação simples quanto a imputação múltipla (RUBIN, 1988).

Os resultados dos experimentos realizados indicam que, para a imputação simples, o algoritmo C4.5 apresentou melhor desempenho do que o algoritmo *AutoClass*. Já para a imputação múltipla, os dois algoritmos foram comparativamente equivalentes e apresentaram bons resultados.

Um bom exemplo da importância da complementação de valores ausentes em bases de dados apresenta-se no tratamento de dados médicos (AUSTIN, ESCOBAR, 2005). Neste caso, a análise dos dados procura uma possível relação de causa e efeito de problemas de saúde com os dados dos pacientes. Todavia, os modelos de regressão só podem ser usados em dados completos, e registros com valores ausentes são então descartados, o que pode tornar tendencioso o conjunto de dados. Entretanto, não é difícil encontrarmos lacunas nos dados desta natureza, pois os profissionais responsáveis por este preenchimento normalmente não sabem qual será o seu uso futuro.

O trabalho de SCHEFFER (2002) mostra como a média e o desvio-padrão são afetados por diferentes métodos de imputação, considerando a existência de diferentes mecanismos de ausência de dados. Os autores sugerem que melhores opções do que as convencionais estão disponíveis em pacotes estatísticos, e com isso abrindo a possibilidade de obtenção de melhores resultados.

Os resultados obtidos sugerem que:

1) a utilização da média é a pior opção sempre;
2) a remoção de casos incompletos é ruim, por destruir a variância inerente ao conjunto de dados;
3) os métodos de imputação simples podem funcionar com dados com mecanismo de ausência aleatória (MAR), mas somente com até 10% de valores ausentes na base.

Os autores também tecem as seguintes recomendações:

1) não se deve usar remoção completa de casos se o seu mecanismo de ausência não for completamente aleatório (MCAR).

2) tentar evitar o acontecimento de dados ausentes;

3) se a regressão simples tiver de ser usada, use a imputação por regressão ou o algoritmo EM;

4) sempre que possível use imputação múltipla;

5) quando usar a imputação múltipla, use um modelo compatível ao modelo de análise onde for possível.

HRUSCHKA, HRUSCHKA JR. e EBECKEN (2002b) especificam um processo de imputação de dados ausentes com a montagem de uma rede bayesiana utilizando o algoritmo *K2* (COOPER, HERSKOVITS, 1992). Este algoritmo é um método bayesiano para a construção de uma rede probabilística a partir de um conjunto de dados, que assume que as variáveis são todas discretas (valores numéricos são discretizados), com todos os valores completos (se a amostra possui n variáveis, onde cada uma pode assumir m valores cada, então a amostra conterá m^n casos distintos para ser considerada completa). Além disso, os objetos do conjunto de dados devem ser independentes uns dos outros, e todas as variáveis devem ser previamente ordenadas (isto significa dizer que, se a variável x_i precede x_j, então não existirá no grafo bayesiano um arco direcionado do nó que representa x_j para o nó que representa x_i). Para cada atributo que apresente algum valor ausente nos registros do conjunto de dados, uma rede bayesiana é construída com os casos completos relativos àquele atributo. Os valores ausentes são obtidos utilizando-se os melhores valores que substituem os valores desconhecidos no conjunto original de dados. Os autores utilizaram bases *Adults* e *Mushrooms*, do repositório *UCI Machine Learning* da Universidade da Califórnia, Irvine (NEWMAN *et al*, 1998). Ambas as bases possuem um atributo-classe, que foi utilizado para validar os resultados das imputações realizadas, com o classificador *RoC* (*Bayesian Robust Classifier*) (RAMONI, SEBASTIANI, 1999), mensurando a taxa média de acerto da classificação dos registros da base original. Os resultados mostram interessantes resultados, tanto no processo de imputação quanto na classificação das tuplas com dados imputados.

O trabalho anterior foi estendido em HRUSCHKA, HRUSCHKA JR. e EBECKEN (2005), com a aplicação de um algoritmo de agrupamento também proposto

pelos autores, que utiliza como base a teoria de algoritmos genéticos, chamado CGA (*Clustering Genetic Algorithm*) antes do processo de imputação com redes bayesianas geradas pelo algoritmo K2. Os autores, neste trabalho, utilizaram as bases *Ruspini* e *Wisconsin Breast Cancer*, também do repositório *UCI* (NEWMAN *et al*, 1998), além de uma base com 200 objetos aleatoriamente gerados com dois atributos. Todas as bases possuíam 30% de valores ausentes em no máximo dois atributos. Os resultados mostram que a taxa média de classificação de tuplas imputadas com a combinação dos algoritmos CGA e K2 foi um pouco menor do que sem a aplicação do algoritmo de agrupamento. Os autores atribuem esta queda de desempenho ao baixo número de objetos com valores ausentes.

AUSTIN e ESCOBAR (2005) aplicam métodos Bayesianos de imputação a dados médicos, utilizando simulações de Monte Carlo, e compara os resultados a um método que usa uma estrutura multivariada dos dados, além da utilização da remoção completa de casos. Segundo os autores, nenhum método se destacou em todas as situações. Todavia, eles fazem duas recomendações: 1) deve-se evitar o uso de métodos de imputação multivariada, e 2) os modelos que utilizam a distribuição probabilística de Bernoulli tendem a tornar os dados menos tendenciosos.

MYRTVEIT, STENSRUD e OLSSON (2001) comparam o algoritmo de verossimilhança com informações completas (FIML) (seção 3.4.1.1.2) com os métodos de imputação por média, de uso do padrão mais similar (SRPI – *Similar Response Pattern Imputation*) e a remoção completa de casos (*listwise deletion*) em um ambiente de modelagem de custos de software. Os resultados dos testes mostram que a aplicação do método FIML sempre é melhor quando o mecanismo de ausência dos dados não for completamente aleatório (MCAR). Com uma posição que vai de encontro a todas as demais observadas, os autores indicam a utilização da remoção completa de casos combinada com um modelo de regressão quando os dados forem completamente aleatórios, e só sugerem imputação (tanto a simples quanto a múltipla) quando existe a premente necessidade de mais dados. Eles indicam o uso da técnica de verossimilhança FIML para construir um modelo para prever dados ausentes, que resultará em dados menos ou quase nada tendenciosos. Além disso, mencionam o evidente fato de que sempre é melhor tentar preencher os valores ausentes com os valores reais, tomando atitudes tais como convocar os coletores ou responsáveis pelos dados para buscar a

informação perdida. Por fim, o trabalho recomenda que técnicas de imputação podem ser usadas se o percentual de dados ausentes não ultrapassar a barreira dos 5%.

HUANG e LEE (2004) propõem uma alteração no cálculo da medida de similaridade do algoritmo dos *k* vizinhos mais próximos. A idéia dos autores é de utilizar a **análise relacional de Grey** (de GRA – *Grey Relational Analysis*) (HUANG, LEE, 2004 apud DENG, 1989), que busca avaliar a relação entre um objeto (chamado *referencial*) e outro (dito *comparado*), calculando o **coeficiente relacional de Grey** (GRC – *Grey Relational Coefficient*), e o **grau relacional de Grey** (GRG – *Grey Relational Grade*).

Considerando um conjunto de dados $X = \{x_0, x_1, \ldots, x_m\}$, onde cada elemento x_i de X possui n atributos, e cada atributo é denotado por $x_i(k)$, $1 \leq i \leq m$ e $1 \leq k \leq n$. Tomando a instância x_0 como base, o coeficiente GRC é calculado da seguinte forma:

$$GRC\left(x_o(p), x_i(p)\right) = \frac{\min_{\forall j} \min_{\forall k} \left|x_o(k) - x_j(k)\right| + \tau \max_{\forall j} \max_{\forall k} \left|x_o(k) - x_j(k)\right|}{\left|x_o(p) - x_i(p)\right| + \tau \max_{\forall j} \max_{\forall k} \left|x_o(k) - x_j(k)\right|}$$

onde $\tau \in [0,1]$. Os autores mencionam que normalmente o valor adotado para τ é 0,5. Podemos observar na fórmula acima que toda a idéia gira em torno do conceito da comparação do valor do atributo *k* com os valores extremos existentes na amostra. Quanto maior for o valor do GRC, maior a similaridade entre os elementos.

Os termos da equação são numéricos. Se necessitarmos calcular o valor de GRC de um atributo categórico *k*, então teremos:

$GRC(x_o(p),x_i(p)) = 1$, se $x_o(p)$ e $x_i(p)$ são iguais;

$GRC(x_o(p),x_i(p)) = 0$, se $x_o(p)$ e $x_i(p)$ são diferentes.

A consolidação dos resultados dos GRC's entre duas instâncias x_0 e x_i é medida pelo *grau relacional de Grey*, dado pela equação:

$$GRG(x_o, x_i) = \frac{1}{n} \sum_{k=1}^{n} GRC\left(x_o(k), x_i(k)\right),\ 1 \leq i \leq m.$$

Os valores assumidos pelo índice $GRG(x_0,x_i)$ estão limitados entre zero e um, onde o valor zero indica que não existe nenhuma semelhança entre as amostras x_0 e x_i, e um indicam que similaridade máxima.

Os dados são previamente normalizados, com um dos três métodos a seguir:

a) **Normalização máxima** (*Upper-bound effectiveness measurement*): o valor do atributo *j* do objeto x_p é recalculado em função do menor valor que ocorre no atributo.

$$x_p'(j) = \frac{x_p(j) - \min_{\forall i} x_i(j)}{\max_{\forall i} x_i(j) - \min_{\forall i} x_i(j)}$$

onde $x_i(j)$ é o valor do atributo *j* associado com a instância x_i, $x_p'(j)$ é o valor de saída do atributo *j* associado com a instância x_p obtida como resultado da etapa de pré-processamento, *m* é o número de instâncias, *n* é o número de atributos, $1 \leq i, p \leq m$ e $1 \leq j \leq n$.

Esta foi a opção de normalização adotada pelos autores.

b) **Normalização mínima** (*Lower-bound effectiveness measurement*) : o valor do atributo *j* do objeto x_p é recalculado em função do maior valor que ocorre no atributo.

$$x_p'(j) = \frac{\max_{\forall i} x_i(j) - x_p(j)}{\max_{\forall i} x_i(j) - \min_{\forall i} x_i(j)}$$

onde $x_i(j)$ é o valor do atributo *j* associado com a instância x_i, $x_p'(j)$ é o valor de saída do atributo *j* associado com a instância x_p obtida como resultado da etapa de pré-processamento, *m* é o número de instâncias, *n* é o número de atributos, $1 \leq i, p \leq m$ e $1 \leq j \leq n$.

c) **Normalização paramétrica** (*Moderate effectiveness measurement*)

$$x_p'(j) = \frac{\left| x_p(j) - x_{USER} \right|}{\max_{\forall i} x_i(j) - \min_{\forall i} x_i(j)}$$

onde $x_i(j)$ é o valor do atributo *j* associado com a instância x_i, x_{USER} é o valor especificado pelo usuário, $x_p'(j)$ é o valor de saída do atributo *j* associado com a instância x_p obtida como resultado da etapa de pré-processamento, *m* é o número de instâncias, *n* é o número de atributos, $1 \leq i, p \leq m$ e $1 \leq j \leq n$.

Os autores mencionam que este tipo de normalização é a menos indicada para o problema em questão.

O algoritmo proposto no trabalho, chamado algoritmo dos *k* vizinhos mais próximos baseado na análise de Grey (*Grey-Based Nearest Neighbor Algorithm*) é então especificado a seguir:

1) Calcule o coeficiente relacional de Grey (GRC) entre o objeto x_0 e x_i, $1 \leq i \leq m$;
2) Encontre os *k* vizinhos mais próximos, baseado no valor de GRC (x_0, x_i);
3) Derive *k* valores distintos, obtidos com os objetos do passo anterior;
4) Calcule o valor a ser imputado utilizando os *k* valores anteriores, calculando a média para valores numéricos, ou a moda, para valores categóricos.

O trabalho apresenta testes de comparação da aplicação do algoritmo acima descrito em várias bases de dados, e utiliza a raiz do erro médio quadrático como uma das medidas de desempenho, que é calculado da seguinte forma:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(e_i - \bar{e}_i\right)^2}$$

onde e_i é o valor original do atributo, $\bar{e}_i$ o valor estimado do atributo, e *m* o número total de imputações.

Os autores relatam resultados que superam a imputação com média e a imputação múltipla, utilizando o algoritmo EM.

Outra medida de avaliação dos resultados foi a reclassificação das tuplas imputadas frente às originais. Os resultados mostram que a relação entre os atributos se preserva.

CAPÍTULO 4

IMPUTAÇÃO COMPOSTA

4.1 Introdução

No capítulo 3, tivemos a oportunidade de analisar as diversas soluções propostas para o processo de imputação de valores ausentes em bases de dados. Todavia, à exceção da técnica *hot-deck*, a grande maioria das propostas busca comparar o desempenho da aplicação de um ou mais métodos de imputação simples. Vários trabalhos procuram se beneficiar da imputação múltipla proposta por RUBIN (1988), onde métodos de imputação simples são utilizados para gerar as várias imputações solicitadas pelo método. Ela combina a análise realizada nos dados, proveniente das imputações simples feitas pelos algoritmos utilizados como base no processo. Porém, não conseguimos identificar de forma clara nos trabalhos disponíveis como essa combinação de análises é realizada.

Após este estudo, não conseguimos identificar, em nenhum dos trabalhos, uma conclusão definitiva sobre os métodos de complementação de dados. A maioria das pesquisas aponta um melhor uso dos algoritmos de imputação de dados onde o mecanismo de ausência seja o aleatório (MAR), e concordam que dados ausentes onde o mecanismo de ausência seja o não aleatório (NMAR) são os de tratamento mais difícil. Todavia, não conseguimos extrair destes trabalhos uma orientação no uso de algoritmos de complementação de dados ausentes (quais os algoritmos mais indicados para uma dada configuração de um conjunto de dados que apresente valores ausentes), apesar de vários deles utilizarem o algoritmo dos *k* vizinhos mais próximos e o algoritmo EM.

Outro fator que consideramos relevante é a pouca ocorrência de análises de imputação com a técnica *hot-deck*, uma técnica composta que procura diminuir o conjunto de possíveis tuplas de entrada do processo de complementação de dados ausentes. O curioso é que, apesar de simples e já à disposição há mais de vinte anos, apenas o trabalho de TSENG, WANG e LEE (2003) utiliza a tarefa de agrupamento precedendo a imputação. Isto nos motivou a imaginar que poderíamos melhorar em muito a qualidade da tarefa de imputação se utilizássemos não só o agrupamento, mas também quaisquer outras tarefas do processo de descoberta de conhecimento de bases

de dados, já que é neste contexto em que reside esta tese. A aplicação de diferentes tarefas produz várias sugestões de imputação, que podem ser comparadas de alguma forma, e servir como base de um julgamento de qual ou quais delas se adaptam melhor em determinadas situações.

Estendendo a idéia anterior, imaginamos que poderíamos também utilizar estas diversas sugestões como entradas de um processo de imputação múltipla. Todavia, nossa idéia não foi a de combinar análises produzidas em dados resultantes da imputação simples de cada uma destas sugestões. Concebemos um modelo inspirado nos comitês de classificação, onde várias sugestões servem de base ou para um processo decisório, ou para uma combinação de resultados que pode produzir uma nova classificação para um registro de uma tabela de uma base de dados. Esta discussão nos levou a propor um novo modelo de comitês, chamados *comitês de complementação de dados ausentes*.

Tomando por base as idéias acima expostas, desenvolvemos nesta tese uma nova técnica de imputação, que nomeamos **imputação composta**. Nesta técnica, o processo de imputação de um atributo ausente é precedido da aplicação de outras tarefas, como por exemplo, o agrupamento de dados e seleção de colunas. Com isso, acreditamos melhorar a qualidade do dado imputado Nosso propósito é então formalizar este processo, e avaliá-lo segundo alguma métrica. Esta medida de qualidade pode ser realizada com a análise de um ou mais parâmetros, e dependente não só do mecanismo de ausência de dados, mas também de outras características do conjunto de dados, tais como a correlação entre os seus atributos.

Assim, na seção 4.2 formalizamos a técnica de imputação composta, destacando o fato de que sua descrição é abstrata. Na seção 4.3, materializamos nossa proposta com a implementação de um sistema de imputação composta, chamado *Appraisal*. Este sistema abriga todo o processo descrito: o de geração de diversos valores para imputação, fruto da aplicação de técnicas simples ou compostas de imputação; a geração de uma nova sugestão de imputação influenciada pelos valores anteriormente gerados (comitês de complementação de dados ausentes); e o processo de validação dos resultados, que é feito de duas formas: com a aferição do erro médio das imputações e com a reclassificação das tuplas com valor imputado. Esta última avaliação nos permite verificar se os valores imputados mantêm as relações intrínsecas às colunas da tabela.

4.2 Formalização da abordagem proposta

Nesta seção formalizamos o processo de imputação composta proposto nesta tese, buscando torná-lo abstrato, preciso e independente de uma implementação específica.

O processo de imputação composta baseia-se na definição dos seguintes elementos:

- T_i: uma tarefa do processo de Descoberta de Conhecimento em Bases de Dados (KDD).

  Exemplos: T_1 = seleção de atributos, T_2 = agrupamento, T_3 = criação de regras de associação, T_4 = imputação, entre outros.

- $\rightarrow$: Operador que define uma ordem de precedência de tarefas de KDD. A expressão $X \rightarrow Y$ significa que a tarefa X precede a tarefa Y.

  Exemplo: *agrupamento* $\rightarrow$ *imputação* significa que a tarefa de agrupamento precederá a de imputação.

- $E(\nu, B)$: estratégia utilizada no processo de imputação de um atributo ν de uma base de dados B. $E(\nu, B)$ é representada por $T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_m$, onde T_m é necessariamente uma tarefa de imputação

- A_i: algoritmo utilizado no processo de imputação.

  Exemplos: A_1 = média, A_2 = algoritmo dos *k* vizinhos mais próximos.

- $\Rightarrow$: Operador que define uma ordem de precedência de aplicação de algoritmos. A expressão $A_i \Rightarrow A_j$ significa que o algoritmo A_i é aplicado antes do algoritmo A_j.

- $P(\nu, B)$: plano de imputação utilizado no processo de imputação de um atributo ν de uma base de dados B. $P(\nu, B)$ é representada por $A_1 \Rightarrow A_2 \Rightarrow \ldots \Rightarrow A_p$, onde A_p é necessariamente um algoritmo de imputação.

Exemplo: $A_1 \Rightarrow A_2 \Rightarrow A_3$ representa a aplicação seqüenciada dos algoritmos A_1 = algoritmo dos *K* centróides, A_2 = análise de componentes principais e A_3 = algoritmo dos *k* vizinhos mais próximos.

- Ψ_i: instância da aplicação de um algoritmo A_i, segundo parâmetros $\Theta_i = \{\Theta_{i1}, \Theta_{i2}, \ldots, \Theta_{iq}\}$. $\Psi_i = f(A_i, \Theta_i)$
- $I(\nu, B)$: instância de um plano de imputação de um atributo ν de uma base de dados B, representada por uma seqüência ordenada de *q* instâncias de aplicações de algoritmos. $\Psi_1 \Rightarrow \Psi_2 \Rightarrow \ldots \Rightarrow \Psi_q$, onde Ψ_q é necessariamente uma instância de aplicação de algoritmo de imputação.
- $\varepsilon(I(\nu, B))$: uma medida do erro na execução de uma instância de um plano de imputação do atributo ν

O conjunto de valores para imputação assumidos por um plano de imputação será composto pelos valores da sua instância que apresentar o menor erro médio de todas as instâncias daquele plano ($\varepsilon(P(\nu)) = \varepsilon(I_k(\nu))$, onde $\varepsilon(I_k(\nu)) < \varepsilon(I_j(\nu)), \forall j \neq k$).

Desta maneira, podemos definir a imputação composta como a aplicação de uma ou mais estratégias no processo de complementação de dados ausentes em um atributo ν de uma base de dados B.

4.3 *Appraisal*: um sistema de imputação composta com comitês de complementação de dados ausentes

4.3.1 Introdução

Para que possamos experimentar quais os efeitos da aplicação da imputação composta, projetamos um sistema que implementa as idéias aqui apresentadas, desde a implementação dos planos de imputação até a medição da qualidade do dado imputado. Este sistema foi batizado com o nome *Appraisal*, inspirado no termo que, em português, significa “aquele que desempenha a função de avaliar a qualidade”.

O sistema é composto basicamente por quatro módulos, ilustrados na figura 4.1:

1) o módulo de execução dos planos de imputação, chamado *Crowner*;

2) o módulo de comitê de complementação de dados ausentes, chamado *Committee*, de geração de valores de imputação baseados nos demais valores sugeridos pelas estratégias e os outros atributos da tabela;

3) o módulo *Reviewer*, que verifica a qualidade das sugestões de imputação produzidas pelos módulos *Crowner* e *Committee*; e

4) o módulo *Eraser*, que simula valores ausentes em uma base de dados segundo um mecanismo e um percentual de ausência definidos pelo usuário.

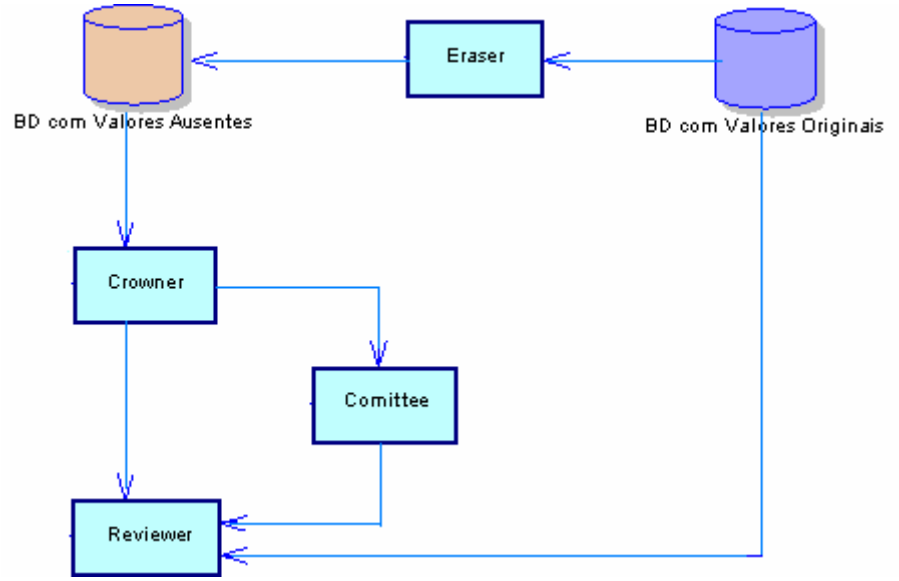


Figura 4.1 *Diagrama do sistema* Appraisal

O princípio norteador da construção do sistema é a flexibilidade na sua implementação e execução. Como será possível observar nas próximas seções deste capítulo, o *Appraisal* possui uma arquitetura voltada à construção de classes abstratas que não contêm implementações específicas de nenhum algoritmo. Com isso, tornamos o sistema escalável, já que as funcionalidades oferecidas podem ser executadas de diferentes formas, com a variação dos algoritmos que já estejam implementados ou que porventura venham a ser integrados no sistema. Na execução, o sistema *Appraisal* utiliza um artefato chamado *arquivo de propriedades*, que permite ao usuário configurar a base alvo das simulações e a coluna a ser regredida, as estratégias a serem executadas em um determinado processo de complementação de dados ausentes, os algoritmos que os planos de imputação irão utilizar, o tipo de erro utilizado nas medições (as opções hoje disponíveis são os erros absoluto e o de similaridade. Abordaremos estes erros no capítulo 5), os parâmetros dos algoritmos, e as entradas selecionadas para os comitês de complementação de dados ausentes.

Os métodos atualmente implementados no *Appraisal* estão preparados para trabalhar com dados numéricos. Todavia, a flexibilidade mencionada anteriormente permite que implementações que tratem de dados categóricos possam facilmente ser incorporadas no sistema. Nas próximas subseções detalharemos o funcionamento de cada um dos módulos integrantes do sistema *Appraisal*.

4.3.2 O Módulo Crowner

O módulo *Crowner* é o responsável por implementar os conceitos de estratégia, planos de imputação e instâncias de planos de imputação abordados nas seções 4.1 e 4.2. A partir da especificação do usuário de quais estratégias e planos de imputação devem ser executados, além da indicação do atributo da base de dados a ser imputado, o sistema inicia o processamento das instâncias de planos de imputação selecionadas para uma determinada simulação. Estas instâncias são montadas tomando como base os arquivos de propriedades de cada método, onde estão listados os valores exatos ou faixa de parâmetros a serem usados. O seu diagrama de atividades é apresentado na figura 4.2.

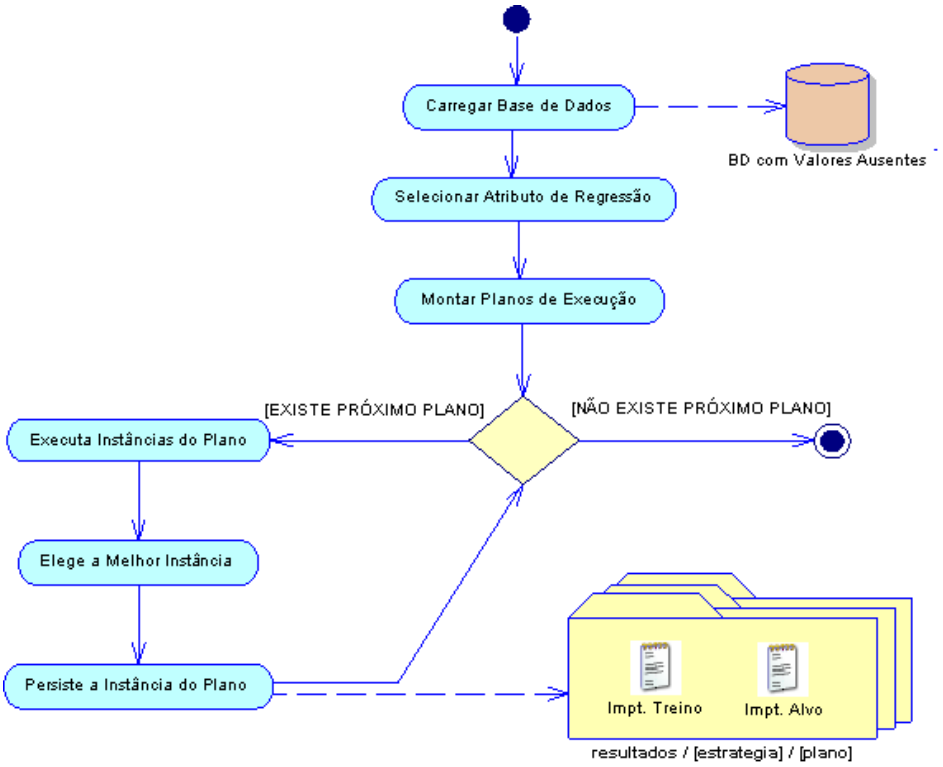


Figura 4.2 Diagrama de Atividades do módulo Crowner *do sistema* Appraisal

O diagrama de classes do módulo *Crowner*, apresentado na figura 4.3, revela bem a filosofia que norteou toda a construção do sistema *Appraisal*, baseado em uma arquitetura de baixo acoplamento.

A interface *Strategy* define o comportamento básico de uma estratégia, e é implementada por cinco classes distintas que representam cada uma das estratégias existentes no sistema. Além disso, qualquer implementação da classe *Strategy* necessariamente retorna uma instância da classe *StrategyResult*, que representa o resultado final do processamento de uma estratégia.

A implementação de uma estratégia agrupa planos, representados por objetos que implementam a interface *Plan*, que define o comportamento básico de um plano. Um plano é uma composição de diferentes tarefas (estágios) de seleção, agrupamento e imputação, que combinadas atendem ao objetivo de uma estratégia.

O comportamento básico de um estágio é definido pela classe abstrata *Stage*. Como exemplo, podemos citar a implementação do plano *SelectionClusteringPlan*, que é composto por uma tarefa (estágio) de **seleção** (associação com *SelectionStage*), seguido de uma tarefa (estágio) de **agrupamento** (associação com *ClusteringStage*), e finalizada por uma tarefa (estágio) de **imputação** (associação com *ImputationStage*).

Os estágios podem possuir múltiplas implementações, utilizando diferentes técnicas e algoritmos. Cada possível combinação dos diferentes estágios é uma materialização diferente de um plano de imputação. Acompanhando o exemplo acima, a classe *SelectionClusteringStartegy* é associada a três objetos *SelectionClusteringPlan*: um que combina os estágios *PcaStage* $\rightarrow$ *KmeansStage* $\rightarrow$ *KnnStage*, outro que combina os estágios *PcaStage* $\rightarrow$ *KmeansStage* $\rightarrow$ *BkPropStage*, e um que combina os estágios *PcaStage* $\rightarrow$ *KmeansStage* $\rightarrow$ *AvgStage*.

A *instância de um plano de imputação* é caracterizada pela execução de um plano de imputação, em um determinado momento, apresentando uma configuração de parâmetros. Conforme já abordado na seção 4.2, um plano é um conjunto de instâncias de planos de imputação, no sentido de que os parâmetros de execução das técnicas e algoritmos variam continuamente. Ainda tomando por base o exemplo do parágrafo anterior, um objeto da classe *SelectionClusteringPlan* que combine *PcaStage* $\rightarrow$ *KmeansStage* $\rightarrow$ *KnnStage* possui tantas instâncias quantas forem as combinações dos parâmetros de execução do estágio de execução das tarefas envolvidas, tais como a

escolha do número de colunas selecionadas e enviadas ao algoritmo de análise de componentes principais (PCA), ou o número de grupos a serem gerados pelo algoritmo de agrupamento dos *K* centróides (*K-Means*), e o número de vizinhos selecionados para a imputação com o algoritmo dos *k* vizinhos mais próximos (*k*-NN).

Para cada instância, há um resultado final definido pela classe *ImputationResult*, que representa o produto da execução. O resultado de um plano, implementado pela classe *PlanResult*, consiste na coleção dos *ImputationResult* de cada uma das instâncias. Finalmente, o objeto *StrategyResult*, é a coleção final dos objetos da classe *PlanResult* de todos os planos. Concluindo o exemplo, cada um dos três objetos *SelectionClusteringPlan* produz como resultado um objeto da classe *PlanResult*, que agrupam os objetos da classe *ImputationResult* de suas instâncias correspondentes. A estratégia produz um resultado final do tipo *StrategyResult*, que agrupa os três *PlanResult* gerados pelos planos.

Como o volume de dados produzido pelo sistema pode ser muito grande, foram implementados métodos para descarte dos piores objetos da *ImputationResult* em tempo real, bem como coleta dinâmica de lixo, a cada determinado número de operações processadas, além de mecanismos de persistência que podem periodicamente preservar o trabalho realizado, prevenindo possíveis ocorrências de falhas.

4.3.3 O Módulo Committee

4.3.2.1 Introdução

Os resultados provenientes das diversas sugestões de imputação simples geradas pelas estratégias do módulo *Crowner* nos instigaram a imaginar como poderíamos combinar estes resultados seguindo a filosofia da imputação múltipla. Como já dispúnhamos de uma rede neuronal *back propagation* implementada no sistema (seção 5.1.4.5), e já utilizada como uma das opções de complementação de valores ausentes, pensamos em construir uma arquitetura onde a fase de treinamento fosse influenciada não só pelos valores imputados, mas por todos valores dos atributos dos registros da tabela. Com isso, queremos avaliar se os valores imputados pelo comitê apresentam melhor precisão frente aos demais.

Assim, baseado no exposto, construímos o módulo *Comittee* do sistema *Appraisal*, cuja arquitetura será apresentada nesta seção. Entretanto, para que possamos melhor

entender os conceitos que embasam os comitês de aprendizado, passaremos antes pelos métodos que implementam os comitês de classificação.

4.3.2.2 Comitês de Classificação

Os comitês de classificação (ou comitês de aprendizado) constroem repetidamente diferentes classificadores utilizando um algoritmo de aprendizado básico – como por exemplo um gerador de árvores – e mudando continuamente a distribuição do conjunto de treinamento.

Os métodos mais aplicados em comitês de classificação são *Bagging*, *Boosting*, *SASC – Stochastic Attribute Selection Committee* e *Stochastic Attribute Selection Committees with Multiple Boosting*.

O método *Bagging* (BREIMAN, 1994) gera diferentes classificadores a partir de diferentes amostras geradas pela técnica *boostrap*. Esta técnica consiste em dividir um conjunto de dados com *N* elementos em um conjunto de treinamento, com *N* seleções uniformes com reposição, e um conjunto de teste, com os objetos não incluídos no conjunto de treinamento. O algoritmo muda estocasticamente os conjuntos de treinamento e usa pesos iguais para os classificadores no processo de votação. É adequado para utilização ambientes envolvendo processamento paralelo.

O método *Boosting* (SCHAPIRE *et al*, 1998) constrói classificadores seqüencialmente, alterando os pesos das amostras e privilegiando para a seleção aquelas classificadas erroneamente pelo classificador gerado anteriormente.

O método constrói classificadores em seqüência. Cada novo classificador é gerado a partir de um conjunto de dados onde as amostras classificadas erroneamente pelo classificador anterior têm maior peso e chance de seleção. Ele muda a distribuição do conjunto de treinamento em função do desempenho dos classificadores criados previamente e usa tal desempenho para definir um peso para o classificador no processo de votação do comitê. Sua formação é seqüencial, apresentando melhor acurácia do que o método *Bagging* (ZHENG, LOW, 1999).

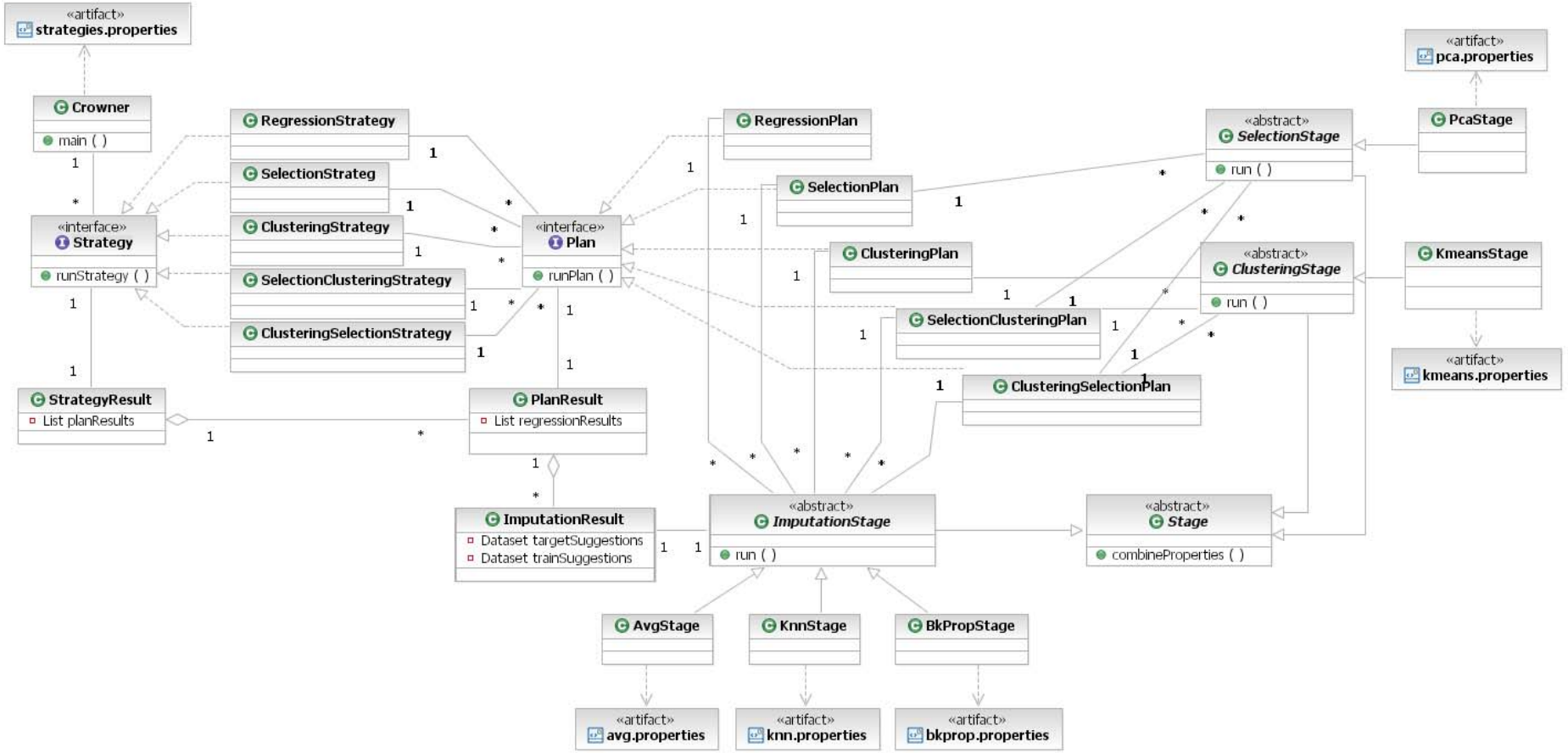


Figura 4.3 Diagrama de Classes do módulo Crowner *do sistema* Appraisal

O método SASC – *Stochastic Attribute Selection Committees* (ZHENG, LOW, 1999) constrói os membros de um comitê formado por classificadores baseados em árvores de decisão pela variação aleatória do conjunto de atributos disponíveis para criação dos nós de decisão. Em geral, a escolha de um nó em uma árvore de decisão se baseia no ganho da informação proporcionada pelos nós disponíveis. Na criação de cada nó, um subconjunto de atributos é selecionado aleatoriamente onde cada atributo disponível possui uma probabilidade *P* de ser selecionado. Após o processo de seleção, o algoritmo escolhe o atributo com maior de informação disponível no conjunto.

4.3.2.3 Comitês de Complementação de Dados Ausentes

As técnicas mais comuns de implementação de comitês de aprendizado apresentadas na seção anterior mostram que todo o processo está direcionado à tarefa de classificação. Não encontramos na literatura uma solução que utilizasse os conceitos apresentados à tarefa de imputação. Isso nos inspirou a propor um processo de imputação que unisse as idéias ligadas a comitês de classificação e ao processo de imputação múltipla: os comitês de complementação de dados ausentes.

Desta maneira, concebemos uma arquitetura de comitê de complementação de dados ausentes com aprendizado supervisionado que é influenciada tanto pelas informações originais das tuplas da tabela quanto pelas diversas sugestões de valores de imputação geradas pelos planos de imputação do módulo *Crowner*. Seu funcionamento divide-se em duas etapas:

1) Quando da execução de seus planos de imputação, o módulo *Crowner* gera valores de imputação não só para as tuplas que apresentem dados ausentes em um atributo. Todas as tuplas recebem uma sugestão de imputação para o atributo em tela. Estes valores servirão na etapa de treinamento do comitê.

2) De posse de todas as sugestões, o treinamento do comitê é feito considerando todas as colunas das tuplas que originalmente não apresentavam valores ausentes, com exceção da coluna alvo do processo de imputação, juntamente com todas as sugestões geradas pelos planos de imputação do módulo *Crowner*. Com isso, se uma tabela possui *x* tuplas completas, *n* atributos e *p* sugestões de imputação, o comitê receberá, a cada iteração da fase de treinamento, (*n*+*p*-1) entradas e uma saída. Este processo se repete *x* vezes

para cada ciclo de treinamento da rede. O número de ciclos da rede é parâmetro fornecido pelo usuário. Veja o esquema da figura 4.4.

3) Na fase de imputação, as *y* tuplas incompletas são enviadas uma a uma para o comitê de complementação de dados ausentes, que, a partir dos valores sugeridos pelos planos de imputação e os valores das demais colunas das tuplas gera uma sugestão de imputação do comitê, cuja qualidade é avaliada a posteriori pelo módulo *Reviewer*. A figura 4.5 ilustra a idéia.

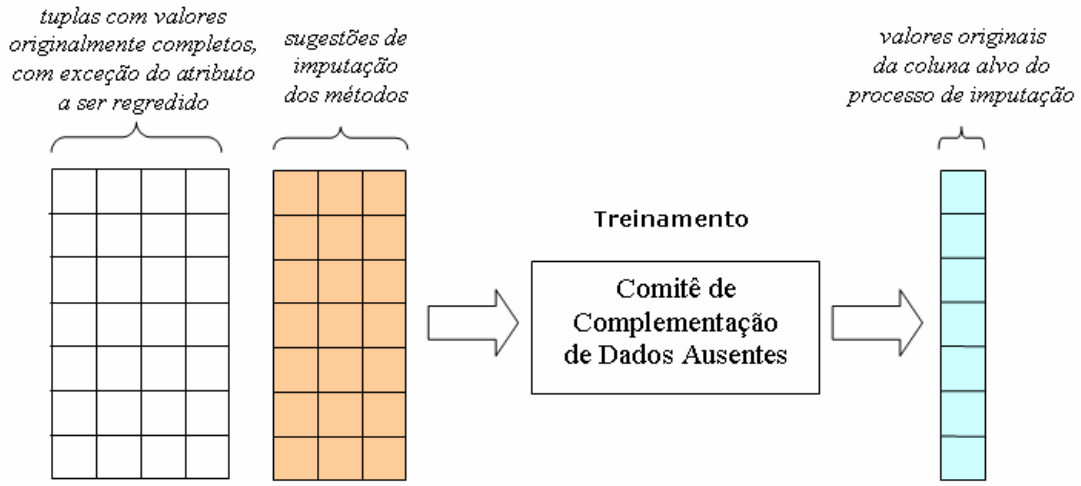


Figura 4.4 Fase de treinamento do comitê de complementação de dados ausentes do sistema Appraisal

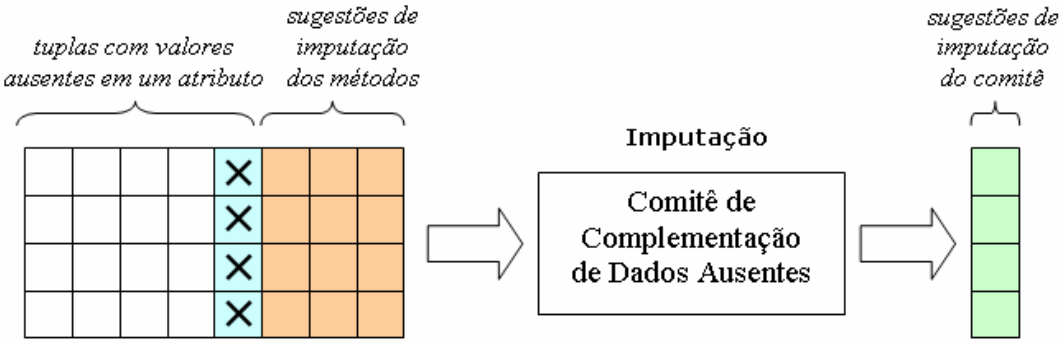


Figura 4.5 Geração de sugestões para valores ausentes do comitê de complementação de dados ausentes.

A figura 4.6 mostra o diagrama de classes do módulo *Committee*. Importante destacar que a sua implementação é feita com uma classe que implementa uma rede neuronal *back propagation*, mas qualquer outra classe que implemente um algoritmo regressor pode ser utilizada. Isto dá flexibilidade ao sistema, permitindo que outras opções sejam analisadas, com o simples desenvolvimento de classes que desempenhem a função de imputação desejada.

O diagrama de atividades do módulo *Committee* é apresentado na figura 4.7. Importante salientar que, na etapa de carga de imputações do conjunto de treino (sugestões de imputação das estratégias de complementação de dados ausentes), o módulo consulta o seu arquivo de propriedades para verificar quais as sugestões que devem ser levadas em conta no processo de imputação com o comitê de complementação de dados ausentes. Isto oferece uma grande flexibilidade ao usuário de descartar um ou mais resultados de imputação que ele porventura julgue inconveniente.

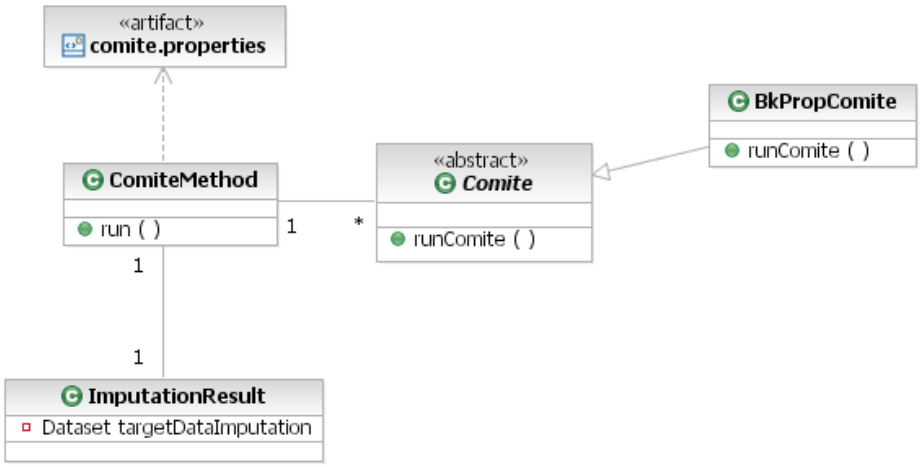


Figura 4.6 Diagrama de Classes do módulo Committee *do sistema* Appraisal

4.3.2.4 Formalização do processo de geração de valores com comitês de aprendizado

Assim como na seção 4.2, podemos formalizar o processo de geração de valores com comitês de aprendizado, definindo os seguintes elementos:

- $\Gamma(v, \mathrm{B})$: seleção e projeção de todos os valores completos do atributo v de tuplas da base B.
- $\Delta(v, \mathrm{B})$: coleção de sugestões de valores a serem imputados para registros completos em um atributo v de uma base B. $\Delta(v, \mathrm{B}) = u(\mathrm{I}_k(v))$
- $\Phi(v, \mathrm{B})$: conjunto de sugestões de valores para imputação em tuplas onde o valor do atributo v da base B é ausente.
- $\Omega(v, \mathrm{B})$: coleção de valores imputados para registros de uma base B em um atributo v. $\Omega(v, \mathrm{B}) = u(\mathrm{I}_k(v))$, onde u é uma função que, a partir de um conjunto de instâncias de planos de imputação, define as sugestões de valores

a serem imputados que apresentam menor erro médio ($\varepsilon(\mathrm{I}_k(v)) < \varepsilon(\mathrm{I}_j(v)), \forall j \neq k$).

Um comitê de complementação de dados ausentes c de valores ausentes de um atributo v de base B é a função que gera sugestões de valores para imputação a partir dos valores $\Gamma(v,\mathrm{B})$, do conjunto $\Delta(v,\mathrm{B})$ e do conjunto $\Phi(v,\mathrm{B})$, fruto das execuções das estratégias de imputação $\mathrm{E}(v,\mathrm{B})$:

$$\Phi(v,\mathrm{B}) = c(v,\mathrm{B},\Omega(v,\mathrm{B}),\Delta(v,\mathrm{B}),\Gamma(v,\mathrm{B}))$$

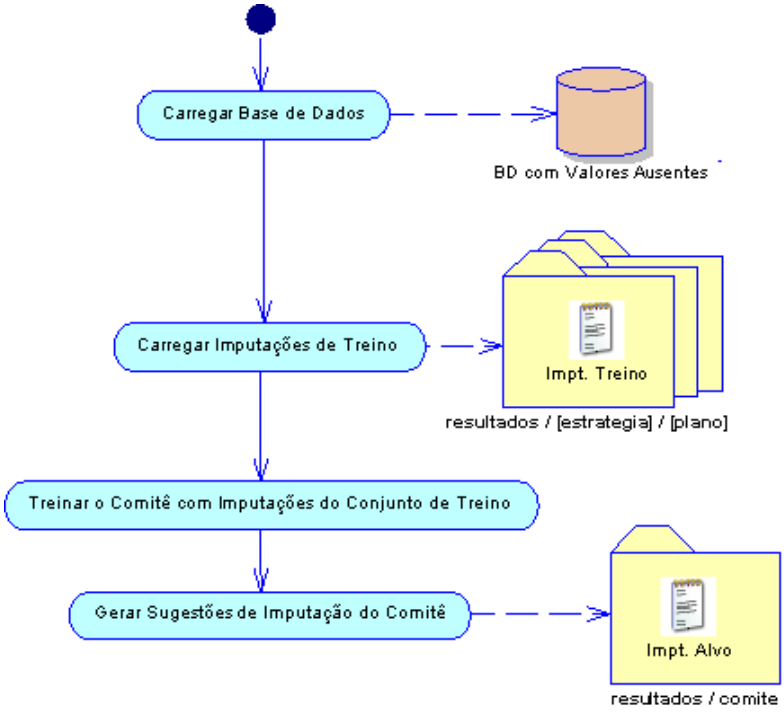


Figura 4.7 Diagrama de Atividades do módulo Committee *do sistema* Appraisal

4.3.4 O Módulo Reviewer

Com o objetivo de avaliar a qualidade das imputações propostas pelos planos de imputação, implementamos no sistema *Appraisal* o módulo *Reviewer*. É de responsabilidade deste módulo executar duas métricas de avaliação dos resultados gerados: a medida do erro entre o valor imputado em cada instância de plano de imputação e a reclassificação das tuplas com valores imputados.

A classe *ResultWriter* implementa as medidas de erro avaliadas pelo sistema, e solicitadas pelo usuário. Nosso interesse nesta tese foi o de avaliar os erros relativo absoluto e de similaridade, que serão tratados no capítulo 5, seção 5.2.2. Todavia, outros

erros podem ser implementados futuramente, para outros tipos de análises. A figura 4.8 apresenta o diagrama de classes do módulo *Reviewer*.

A classe *Classification* verifica se as relações existentes entre os atributos da tabela permanecem inalteradas com o processo de imputação. Para isso, ela submete os registros que receberam tratamento a um novo processo de classificação, e compara o resultado obtido com a real classe da tupla. Esta informação (a de conhecimento do valor do atributo-classe), apesar de disponível durante todo o processo, não é utilizada em nenhum outro momento senão este.

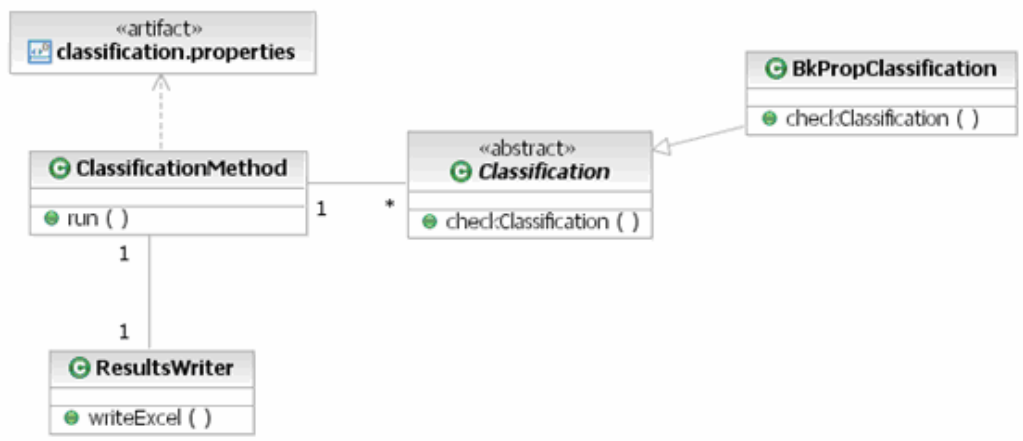


Figura 4.8 Diagrama de Classes do módulo Reviewer *do sistema* Appraisal

Utilizamos como algoritmo classificador uma rede neuronal *back propagation*, por ser uma rede amplamente conhecida por seu potencial de classificação. A camada de entrada recebe os atributos numéricos preditivos da tupla, e na camada de saída gera a classe da tupla. A representação da classe foi implementada utilizado as codificações binária e binária econômica, abordadas no capítulo 2, seção 2.4.3. Entretanto, todos os resultados apresentados no capítulo 5 referentes ao processo de classificação estão relacionados apenas à codificação binária econômica, que na maioria das vezes apresentou resultados com uma melhor precisão.

O diagrama de atividades do módulo *Reviewer* ilustra o seu funcionamento (figura 4.9). Como a classificação é feita de forma supervisionada, as tuplas da base de dados que apresentam valores completos no atributo de imputação são usadas como conjunto de testes do classificador: os atributos numéricos são entradas da rede, e a classe codificada como um número binário a saída.

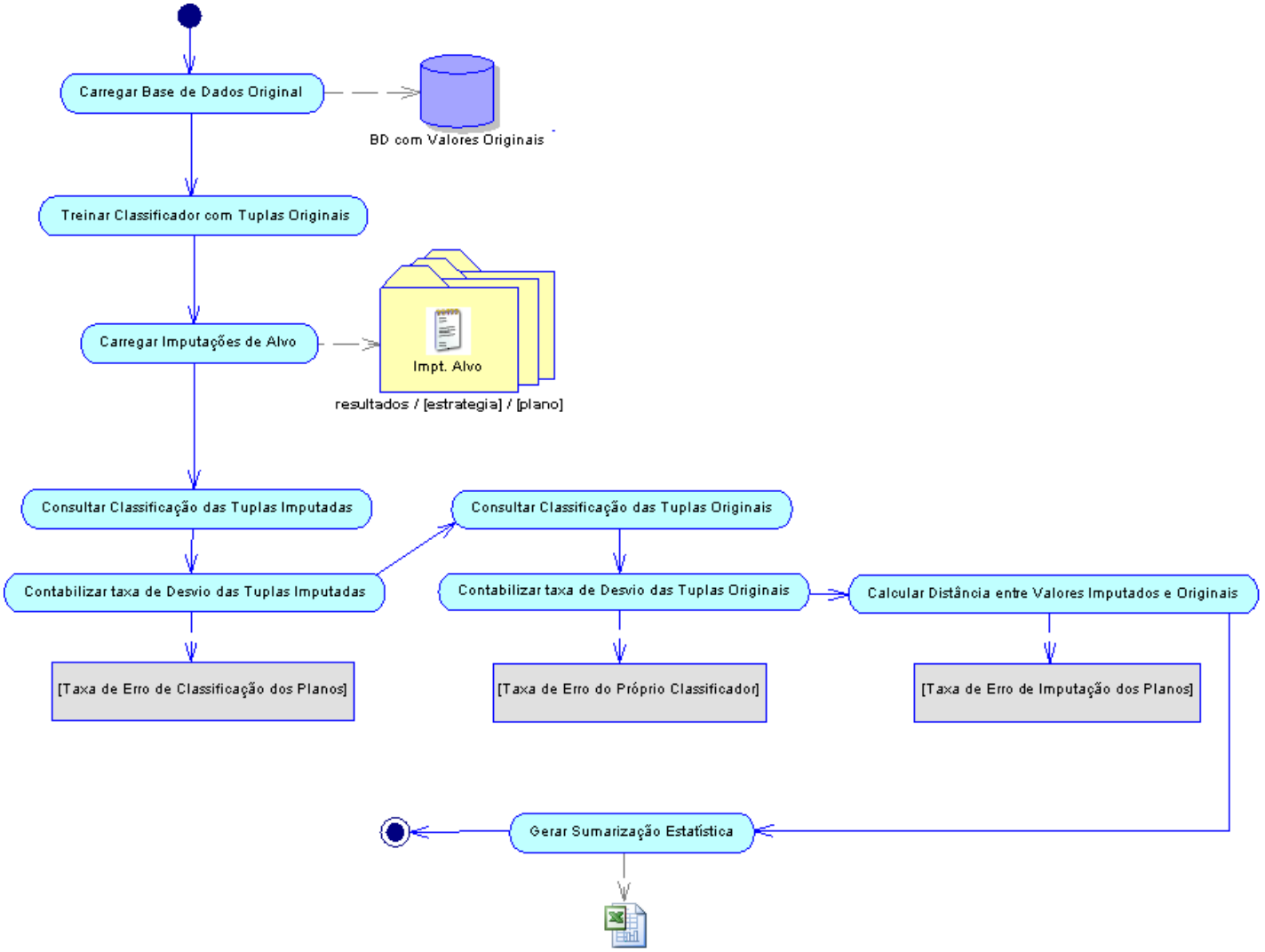


Figura 4.9 Diagrama de Atividades do módulo Reviewer *do sistema* Appraisal

Queremos também verificar qual a qualidade das categorizações feitas pelo classificador. Por esta razão, consultamos as versões originais das tuplas que apresentaram valores ausentes (antes de serem alteradas pelo módulo *Eraser*, que será tratado na próxima seção), e as submetemos ao classificador. O erro de classificação das tuplas originais é comparado então com o erro destas mesmas tuplas que sofreram imputação, para que possamos ter condições de relacionar as falhas de classificação das tuplas imputadas com a qualidade do classificador.

4.3.5 O módulo Eraser

O módulo *Eraser* do sistema *Appraisal* foi desenvolvido com o objetivo de simular valores ausentes em uma base de dados, segundo um mecanismo de ausência de dados definido. Como é nosso interesse ter total controle dos experimentos realizados, precisamos saber informações de como os dados são removidos, tais como o percentual de valores removidos, as regras que regeram a remoção nos casos indicados, entre outros.

Para concretizar a ausência de dados, o módulo *Eraser* atribui valor nulo à coluna especificada. Quando o mecanismo de ausência é o completamente aleatório (MCAR), podemos atribuir valores nulos a mais de um atributo, bastando, com isso, selecioná-lo no painel à esquerda, levando-os para o da direita (figura 4.10). O sistema com isso escolhe aleatoriamente um percentual de tuplas da base, índice esse especificado na parte inferior da janela, e tornam nulos os valores do atributo ou dos atributos selecionados.

Quando o mecanismo de ausência escolhido é o aleatório (MAR), a seleção do atributo que terá seus valores removidos é feita no painel à esquerda. À direita, o usuário especifica as condições que farão os valores do atributo anteriormente selecionado serem alterados para nulo. Assim, no exemplo da figura 4.11, as tuplas que possuem o atributo *Mitoses* com valores menores que cinco e o atributo *Bare_Nuclei* com valores iguais a dois são selecionadas, e 15% delas têm o valor do atributo *Normal_Nucleoli* removido.

No caso onde deseja-se remover atributos com o mecanismo de ausência não aleatória (NMAR), o usuário deve especificar condições independentes para cada um dos atributos que devem ter seus valores removidos. Assim, no exemplo da figura 4.12, o sistema seleciona dois subconjuntos da tabela: um com tuplas que possuam valores

iguais a sete no atributo *Uniformity_of_Cell_Shape*, e com valores menores ou iguais a 1,2 no atributo *Margnial_Adhesion*. Cada um destes subconjuntos terá 60% dos valores removidos respectivamente nos atributos *Uniformity_of_Cell_Shape* e *Margnial_Adhesion*.

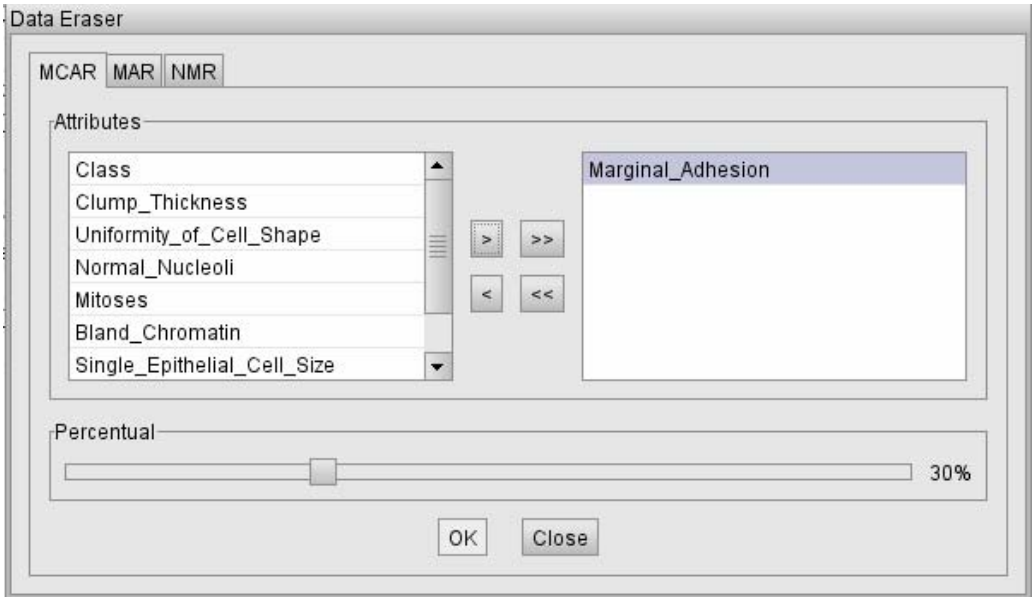


Figura 4.10 Exemplo de remoção de valores do atributo Marginal_Adhesion *com o mecanismo completamente aleatório do módulo* Eraser *do sistema* Appraisal

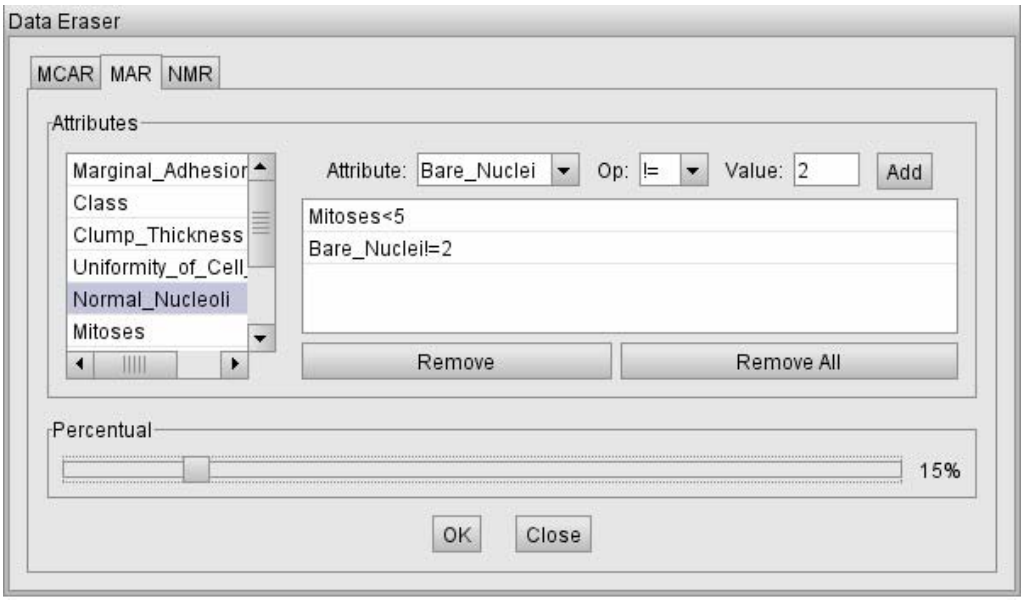


Figura 4.11 Exemplo de remoção de valores do atributo Normal_Nucleoli *com o mecanismo aleatório do módulo* Eraser *do sistema* Appraisal

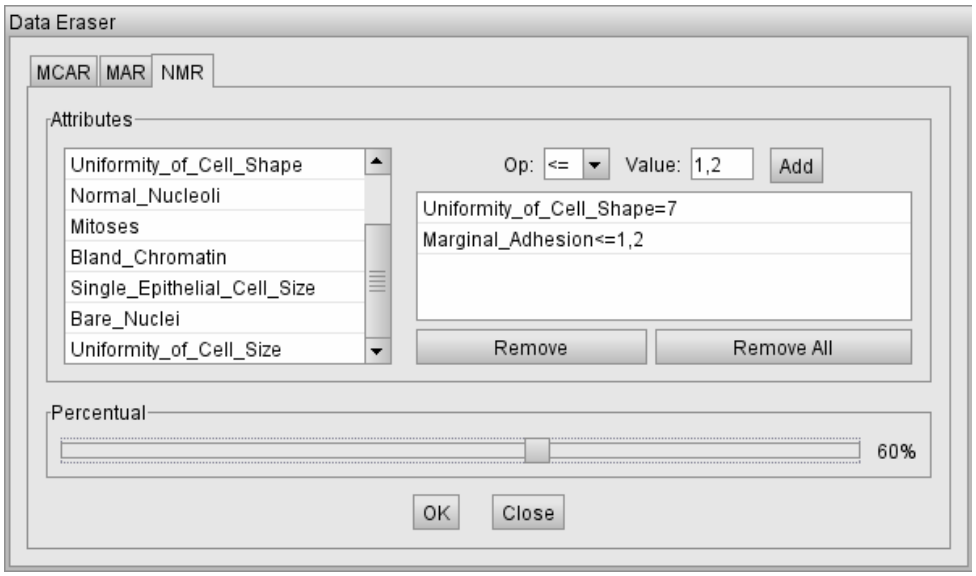


Figura 4.12 Exemplo de remoção de dois valores dos atributos Marginal_Adhesion *e do atributo* Uniformity_of_Cell_Shape *com o mecanismo não aleatório do módulo* Eraser *do sistema* Appraisal

CAPÍTULO 5

ANÁLISE DE RESULTADOS

5.1 Metodologia

5.1.1 Bases de dados utilizadas

Para avaliarmos o efeito real da aplicação das estratégias de complementação de dados, decidimos utilizar três bases de dados existentes no repositório da Universidade da Califórnia, Irvine (NEWMAN, 1998): *Iris Plants*, *Pima Indians Diabetes* e *Wisconsin Breast Cancer*. Três principais razões nos motivaram a escolher estas bases:

1) São conjuntos de dados frequentemente utilizados nos trabalhos relacionados à complementação de dados ausentes, por suas características estruturais (relação entre os atributos) e por representarem, de fato, dados reais;

2) Todos os atributos considerados em nosso estudo são numéricos. As três bases utilizadas em nossos experimentos possuem esta característica. Esta escolha deveu-se ao fato de que nos interessa primordialmente nesta tese avaliar qual o impacto da aplicação de tarefas específicas de pré-processamento antes da imputação no processo de complementação de dados. Os atributos categóricos, por sua natureza, trazem outras questões inerentes que pretendemos estudar em trabalhos futuros; e

3) Todas as bases possuem um atributo classificador dos registros. Esta informação, apesar de não ser considerada no processo de imputação, será importante no processo de validação, pois queremos saber se o processo de complementação de dados mantém a relação existente entre os atributos da tabela. Para isso, após a imputação, submetemos a tupla a um novo processo de classificação, e verificamos se a classe indicada pelo classificador é a classe real do registro.

Outra observação importante é que, em todas as bases, não levamos em consideração nos testes (e também nas descrições efetuadas a seguir) o atributo-chave da tabela, ou qualquer outro identificador único que a tabela apresente.

5.1.2 Descrição das bases

5.1.2.1 Iris Plants

Um dos mais utilizados conjuntos de dados em tarefas de Mineração de Dados é a base *Iris Plants*, construída por R. A. Fischer em meados dos anos 30 (WITTEN, FRANK, 2005), e que registra as medidas de comprimento e largura de pétalas e caule de três tipos de plantas: *Virginica*, *Versicolor* e *Setosa*. Existem 150 tuplas na base, e 50 delas (33,33%) estão localizadas em cada uma das três classes. Nenhum registro apresenta valores ausentes. Os valores mínimo, máximo, média e desvio-padrão de cada atributo seguem:

Os atributos da base são o comprimento da pétala (*petallength*), a largura da pétala (*petalwidth*), o comprimento do caule (*sepallength*) e a largura do caule (*sepalwidth*), todos expressos em centímetros. O atributo-classe registra a qual das três classes as medidas da tupla pertence.

	Valor mínimo	Valor máximo	Média	Desvio-Padrão
sepallength	4.3	7.9	5.84	0.83
sepalwidth	2.0	4.4	3.05	0.43
petallength	1.0	6.9	3.76	1.76
petalwidth	0.1	2.5	1.20	0.76

A matriz de correlação entre os atributos é a mostrada abaixo:

	sepallength	*sepalwidth*	*petalwidth*	*petallength*
sepallength	1.00	-0.11	0.82	0.87
sepalwidth	-0.11	1.00	-0.36	-0.42
petalwidth	0.82	-0.36	1.00	0.96
petallength	0.87	-0.42	0.96	1.00

Uma característica interessante desta base é que as tuplas da classe *Setosa* são independentes das demais classes, pois a única característica que as diferencia das outras é a medida do atributo *petalwidth*. Valores menores ou iguais a 0.6 cm implicam na classificação do registro na classe *Setosa*, independente dos demais atributos.

5.1.2.2 Pima Indians Diabetes

A base *Pima Indians Diabetes* registra dados de pacientes do sexo feminino de pelo menos 21 anos provenientes da tribo *Pima* do Arizona, EUA. Os registros contêm informações sobre as características das pacientes. São elas:

1) *pedigree_function*: função de características hereditárias da diabetes.
2) *glucose_concentration*: nível de glicose no sangue duas horas após a ingestão de glicose concentrada de um teste de tolerância à glicose.
3) *body_mass*: índice de massa corporal (peso em kg/(altura em m)^2)
4) *skin_fold_thickness*: espessura da pele do tríceps (em mm)
5) *blood_pressure*: pressão sanguínea diastólica (mm Hg).
6) *age*: idade, em anos.
7) *serum_insulin*: nível de insulina no sangue (mu U/ml).
8) *pregnancy_times*: número de vezes que a paciente engravidou.

Existe ainda um atributo classe, que indica se a paciente tem ou não diabetes.

Originalmente a base apresenta 768 registros, com 500 deles referentes a pacientes que não possuem diabetes, e 268 com resultado positivo para esta doença. Além disso, a descrição da base registra que não existe nenhum valor ausente nos atributos das tuplas.

Entretanto, ao analisar a base, pudemos constatar ocorrências de atributos com valor zero em colunas onde a princípio este valor não pertence ao domínio da maioria dos atributos. A situação encontrada foi a seguinte:

- *skin_fold_thickness*: 227 registros com valor zero;
- *body_mass*: 11 registros com valor zero;
- *serum_insulin*: 374 registros com valor zero;
- *blood_pressure*: 35 registros com valor zero;
- *glucose_concentration*: 5 registros com valor zero.

No total, encontramos 376 registros que possuem um destes atributos com valor zero. Acreditamos que esta é uma situação onde os atributos apresentavam valor ausente, que, em algum momento, foi substituído por zero, já que não é possível a

existência de pacientes com massa corporal ou pressão sanguínea diastólica nula. Por esta razão, decidimos remover da base todas as tuplas que apresentaram valor zero em qualquer dos atributos numéricos, com exceção do atributo *pregnancy_times*, já que é perfeitamente plausível uma paciente não ter passado por uma gestação. Com isso, a base passou a ter 392 registros, com 262 casos de diabetes comprovada, e 130 sem apresentar a doença. Os dados estatísticos sobre os atributos são os listados abaixo:

	Valor mínimo	Valor máximo	Média	Desvio-Padrão
age	21	81	30,86	10,18
blood_pressure	24	110	70,66	12,48
body_mass	18,2	67,1	33,08	7,01
glucose_concentration	56	198	122,62	30,82
pedigree_function	0,085	2,42	0,52	0,34
pregnancy_times	0	17	3,30	3,20
serum_insulin	14	846	156,05	118,69
skin_fold_thickness	7	63	29,14	10,50

E, sendo os atributos:

1: *pedigree_function*
2: *glucose_concentration*
3: *body_mass*
4: *skin_fold_thickness*
5: *blood_pressure*
6: *age*
7: *serum_insulin*
8: *pregnancy_times*

sua matriz de correlação é a representada abaixo:

Atributo	1	2	3	4	5	6	7	8
1	1.00	0.14	0.16	0.16	-0.02	0.09	0.14	0.01
2	0.14	1.00	0.21	0.20	0.21	0.34	0.58	0.20
3	0.16	0.21	1.00	0.66	0.30	0.07	0.23	-0.03
4	0.16	0.20	0.66	1.00	0.23	0.17	0.18	0.09
5	-0.02	0.21	0.30	0.23	1.00	0.30	0.10	0.21
6	0.09	0.34	0.07	0.17	0.30	1.00	0.22	0.68
7	0.14	0.58	0.23	0.18	0.10	0.22	1.00	0.08
8	0.01	0.20	-0.03	0.09	0.21	0.68	0.08	1.00

5.1.2.3 Wisconsin Breast Cancer

Dados sobre pacientes de câncer de mama do hospital da Universidade de Wisconsin, doados ao repositório UCI pelo Dr. William H. Wolberg, formam a base de

dados *Wisconsin Breast Cancer* (MANGASARIAN, WOLBERG, 1990), utilizada nesta tese como uma das bases de testes.

A base é composta por nove atributos, relacionados abaixo:

1) *Uniformity_of_Cell_Size*
2) *Clump_Thickness*
3) *Bland_Chromatin*
4) *Uniformity_of_Cell_Shape*
5) *Marginal_Adhesion*
6) *Mitoses*
7) *Bare_Nuclei*
8) *Normal_Nucleoli*
9) *Single_Epithelial_Cell_Size*

A base original possui 699 tuplas, onde 17 delas possuem valores ausentes. Para controle total de nossos experimentos, optamos por removê-las, permanecendo assim com 682 registros, onde 443 indicam pacientes sem a doença, e 239 com neoplasia mamária. Os dados relativos aos registros são os seguintes:

	Valor mínimo	Valor máximo	Média	Desvio-Padrão
Bare_Nuclei	1	10	3,54	3,64
Bland_Chromatin	1	10	3,44	2,44
Clump_Thickness	1	10	4,43	2,82
Marginal_Adhesion	1	10	2,83	2,86
Mitoses	1	10	1,60	1,73
Normal_Nucleoli	1	10	2,87	3,05
Single_Epithelial_Cell_Size	1	10	3,23	2,22
Uniformity_of_Cell_Shape	1	10	3,21	2,98
Uniformity_of_Cell_Size	1	10	3,15	3,06

Sendo

1: *Uniformity_of_Cell_Size*
2: *Clump_Thickness*
3: *Bland_Chromatin*
4: *Uniformity_of_Cell_Shape*
5: *Marginal_Adhesion*
6: *Mitoses*
7: *Bare_Nuclei*
8: *Normal_Nucleoli*
9: *Single_Epithelial_Cell_Size*

a matriz de correlação é a que se segue:

Atributo	1	2	3	4	5	6	7	8	9
1	1.00	0.64	0.76	0.91	0.71	0.46	0.69	0.72	0.75
2	0.64	1.00	0.55	0.65	0.49	0.35	0.59	0.53	0.52
3	0.76	0.55	1.00	0.74	0.67	0.35	0.68	0.67	0.62
4	0.91	0.65	0.74	1.00	0.69	0.44	0.71	0.72	0.72
5	0.71	0.49	0.67	0.69	1.00	0.42	0.67	0.60	0.59
6	0.46	0.35	0.35	0.44	0.42	1.00	0.34	0.43	0.48
7	0.69	0.59	0.68	0.71	0.67	0.34	1.00	0.58	0.59
8	0.72	0.53	0.67	0.72	0.60	0.43	0.58	1.00	0.63
9	0.75	0.52	0.62	0.72	0.59	0.48	0.59	0.63	1.00

5.1.3 Parâmetros relativos à ausência dos dados

Produzimos bases de dados com padrão de ausência *univariado* (apenas um atributo com valores ausentes por vez), copiando a base original e gerando um total de 10%, 20%, 30%, 40% ou 50% de dados aleatoriamente nulos, com o uso do módulo *Eraser* do sistema *Appraisal*. Com isso, formamos 20 bases distintas para o conjunto de dados *Iris Plants*, 40 bases para os dados *Pima Indians Diabetes*, e 45 bases para o conjunto de dados *Wisconsin Breast Cancer*. Cada base é identificada com o nome da base, concatenado à sigla do mecanismo ausência de dados, o nome do atributo e o percentual de valores ausentes. Por exemplo, a base *iris_mcar_petallength_10* refere-se à base que possui 10% de valores ausentes no atributo *petallength* da base *Iris Plants*.

A decisão de variar os percentuais de valores ausentes entre 10% e 50%, com saltos de 10% foi tomada por estes índices representarem percentuais que acontecem em situações reais. Além disso, interessa-nos observar o comportamento da execução das estratégias de complementação de dados em função do aumento do índice de ausência nas bases de dados. A faixa percentual de valores ausentes adotada cobre bem as características das bases reais, especialmente as taxas mais altas, já que não é incomum encontrarmos índices mais altos do que 50% de valores ausentes em bases de dados (LAKSHMINARAYAN *et al*, 1999).

Nosso interesse nesta tese foi o de avaliar os resultados do processo de imputação com dados que obedecem ao mecanismo de ausência completamente aleatório (MCAR). Com isso, queremos observar qual o comportamento da

complementação em dados onde não existe um motivo conhecido para a ocorrência da ausência. É nossa intenção avaliar em trabalhos futuros o desempenho do processo de imputação em dados com mecanismo de ausência aleatória (MAR) e não aleatória (NMAR).

5.1.4 Parâmetros dos algoritmos

5.1.4.1 Uso de média aritmética determinística ou estocástica

Ao classificar os métodos de tratamento de dados ausentes, MAGNANI (2004) cita, conforme descrito no capítulo 3, seção 3.5.2.1, que a imputação global baseada no próprio atributo pode ser feita de forma *determinística* ou *estocástica*. Como o método desta categoria implementado nesta tese foi a média aritmética, a aleatoriedade da segunda opção descrita pode ser obtida com a adição ou subtração de um valor aleatoriamente gerado pelo sistema. Além disso, a decisão sobre se o valor randômico será somado ou subtraído também é aleatória, e decidida a cada execução de uma instância de um plano. Com isso, tentamos suavizar o problema inerente à média determinística, de redução da variância dos dados, tornando-a menos tendenciosa.

Assim, implementamos as duas opções acima descritas, e, a cada execução de um plano onde a imputação de um atributo de uma tupla fosse feita utilizando a média (na imputação simples, seleção, agrupamento e imputação e agrupamento, seleção e imputação), duas instâncias eram geradas, uma calculando a média determinística, e a outra a média estocástica. Os resultados mostram o número de vezes onde cada uma delas obteve melhor desempenho:

	Determinística	Estocástica
Iris Plants	**75** (93,75%)	**5** (6,25%)
Pima Indians	**153** (95,63%)	**7** (4,38%)
Breast Cancer	**151** (83,89%)	**29** (16,11%)

Estes resultados parecem indicar que, independentemente do método de complementação e da base de dados utilizados, na grande maioria das vezes a média determinística vence. Isto demonstra que a perturbação no cálculo da média nas bases utilizadas nesta tese trouxe poucas vantagens. Todavia, esta tendência é observada apenas no escopo dos experimentos desta tese, e a princípio não pode ser generalizada para qualquer tipo de base de dados.

5.1.4.2 Parâmetros do algoritmo dos *k* vizinhos mais próximos

A escolha dos parâmetros relacionados ao algoritmo dos *k* vizinhos mais próximos foi bastante influenciada pelas soluções que encontramos disponíveis na literatura. Estas opções estão relacionadas a dois fatores:

5.1.4.2.1 Tipo de distância

Na seção 2.6.2, quando tratamos os algoritmos de agrupamento de dados, mostramos que o cálculo da distância entre dois objetos numéricos pode ser feito de diversas formas. Esta noção de similaridade é inerente também no algoritmo *k*-NN, já que ele informa, em grau decrescente de prioridade, quais os *k* objetos que mais se parecem com um selecionado. Apesar de a versão clássica do algoritmo k-NN utilizar a distância Euclidiana, observamos uma comparação feita em HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003b) da distância Euclidiana com a distância *Manhattan*, também mostrada no capítulo 2. Neste trabalho, os autores recomendam o uso da distância *Manhattan*, tanto pelos resultados obtidos (em 57.35% dos experimentos esta distância apresentou erros menores do que a distância Euclidiana). E como os autores utilizam as mesmas bases que nós usamos nesta tese, decidimos também realizar esta comparação, e analisar qual o desempenho do uso destas duas distâncias quando do uso de estratégias compostas, como as que utilizamos aqui.

Os resultados dos testes mostram que nem sempre a distância *Manhattan* é a que gera melhores resultados. Depois da execução de todas as instâncias de um plano de imputação, a instância que apresenta o menor índice de erro relativo absoluto é eleita a melhor. Observe então o desempenho de cada uma destas distâncias por instância de plano vencedor e por base de dados:

	Euclidiana	*Manhattan*
Iris Plants	**59** (59,00%)	**41** (41,00%)
Pima Indians	**62** (31,00%)	**138** (69,00%)
Breast Cancer	**70** (31,11%)	**155** (68,89%)

Apesar de a distância Euclidiana ter ganho na base *Iris Plants*, o resultado mostrou-se bem equilibrado (59% contra 41%). Isto já não acontece nas demais bases, onde a distância *Manhattan* ganha em quase 70% dos casos. Assim, verificamos que, no escopo de nossos experimentos, a sugestão de uso da distância

Manhattan mencionada por HRUSCHKA, HRUSCHKA JR. e EBECKEN (2003b) é de fato a mais adequada na maioria dos casos, dados os resultados apresentados.

5.1.4.2.2 Determinação do valor de k adotado

Definitivamente não existe consenso entre os autores que pesquisam a complementação de dados ausentes utilizando o algoritmo *k*-NN. Alguns autores tentam propor heurísticas, como é o caso de JÖNSSON e WOHLIN (2004), que sugerem que o valor ideal de *k* deve ser a raiz quadrada do número *N* de casos completos, arredondado superiormente ($k = \lceil \sqrt{N} \rceil$). CARTWRIGHT, SHEPPERD e SONG (2003) sugerem *k* = 1 ou 2, mas alegam que *k* = 1 faz com que o algoritmo se torne muito sensível a valores fora da faixa. Então, chegam à conclusão que o valor ideal de *k* = 2. MYRTVEIT et al (2000) e HUISMAN (2000) sugerem *k* = 1. Já BATISTA e MONARD (2001) adotam como ideal o valor de *k* = 10. TROYANSKAYA *et al* (2001) dizem que o método é totalmente insensível à escolha de *k*. Porém, ressaltam que á medida que o *k* cresce, o valor da média aumenta, e se torna menos precisa, tendendo para o valor da média da coluna.

Considerando então que a grande maioria dos trabalhos observados na literatura apresenta uma metodologia para a escolha do valor do número de vizinhos mais próximos a ser adotado, decidimos em nossos testes variar o valor de *k* entre 1 e o número total de tuplas válidas. Isto teve o seguinte reflexo em nossos testes:

1) Base *Iris Plants*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-135	1-120	1-105	1-90	1-75

2) Base *Pima Indians*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-353	1-314	1-275	1-236	1-196

3) Base *Breast Cancer*

   Os valores que a princípio assumiríamos para esta base deveriam ser os seguintes:

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-614	1-546	1-478	1-410	1-341

Todavia, refletimos qual seria o impacto de uma variação tão ampla do número de vizinhos, que está também associada à variação do número de centróides e de colunas selecionadas (conforme veremos nas próximas subseções). Além disso, como já dito na seção anterior, nos planos onde a média é o algoritmo escolhido para a imputação, calculamos sua versão determinística e estocástica.

Todos estes testes na base *Iris Plants* demandaram 35,1 horas (ou quase um dia e meio). Na base *Pima Indians*, a variação durou 412,8 horas (ou um pouco mais de ininterruptos 17 dias). Como a base *Breast Cancer* possui um atributo a mais do que a *Pima Indians*, e cinco a mais do que a *Iris Plants*,optamos por adotar a heurística de JÖNSSON e WOHLIN (2004), e limitar o valor de *k* entre 1 e a raiz quadrada do número de casos válidos, com uma pequena variação: o valor de máximo de *k* foi calculado como a raiz quadrada do número de casos da base, e não apenas os válidos. Com isso, aumentamos um pouco a faixa de valores de *k*, e com isso passamos a ter na base *Breast Cancer* os seguintes valores:

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-27	1-27	1-27	1-27	1-27

Mesmo com a limitação acima imposta, os testes na base *Breast Cancer* duraram 1202,3 horas, ou 50,1 dias.

A seguir listamos, por base de dados, quais os menores e os maiores valores de *k* em instâncias de planos de imputação com menor taxa de erro:

	Menor *k*	Maior *k*
Iris Plants	1	37
Pima Indians	1	140
Breast Cancer	1	27

Baseado no resultado máximo das bases *Iris Plants* e *Pima Indians*, vemos que existem situações onde a heurística proposta por JÖNSSON e WOHLIN (2004), de que o melhor valor de *k* é a raiz quadrada dos casos válidos, não se aplica. Vejamos o porquê desta afirmação:

- Na primeira base (*Iris Plants*), se a heurística mencionada fosse válida, teríamos os seguintes valores limítrofes: para 10% de valores ausentes, o

número de casos válidos é igual a 105. Assim, o valor de sua raiz quadrada seguida da aplicação da função teto é 11. Com 50% de valores ausentes, temos 75 casos válidos, e k = 9. Como obtivemos um k com valor 37 pertencendo a uma das instâncias vencedoras de um plano de imputação, a heurística falhou neste caso.

- Efetuando a mesma análise para a base *Pima Indians*, com 10% de valores ausentes temos 353 casos e k = 19, e com 50% de valores ausentes 196 casos, com k = 14 casos. Como uma das melhores instâncias de planos de imputação obteve um valor de k igual a 140, verificamos que também aqui a heurística falha.

- Por fim, na base *Breast Cancer* havíamos nos baseado na heurística e limitado em 27 o número de vizinhos. Lembramos que este valor não representa exatamente o que a heurística de JÖNSSON e WOHLIN (2004) sugere. A tabela de resultados mostra que este valor foi obtido como uma das melhores instâncias de planos de imputação, o que sugere que, se tivéssemos permitido que o valor de k variasse por todas as possibilidades, assim como fizemos nas demais bases, este valor seria facilmente ultrapassado. Desta forma, concluímos de forma experimental que a heurística de fato não se aplica a nenhum dos casos.

5.1.4.3 Parâmetros do algoritmo de agrupamento dos *K* centróides

Como já abordado no capítulo 3, um procedimento *hot-deck* (FORD, 1983) consiste no uso de um algoritmo de agrupamento precedendo a imputação. Todavia, apesar de ser uma solução já consolidada na área, não pudemos observar, após pesquisa na literatura, nenhum estudo que analisasse qual o possível número (ou faixa) ideal de grupos a serem formados antes da aplicação do algoritmo de imputação.

Tendo em vista que, apesar de a técnica *hot-deck* de imputação de dados não ser uma proposta recente, não encontramos nenhum trabalho onde o tipo de variação do algoritmo de agrupamento fosse considerado. TSENG, WANG e LEE (2003) agrupam os dados antes de regredi-los, mas não exploram a forma como este agrupamento é feito. Assim, decidimos neste trabalho utilizar o algoritmo de agrupamento dos *K* centróides, por ser uma técnica bastante consolidada, e de baixo

custo computacional, frente às demais opções disponíveis de algoritmos de agrupamento. Entretanto, esta escolha implicou na necessidade de configuração de quatro parâmetros:

5.1.4.3.1 A forma de inicialização dos centróides

Várias opções são encontradas na literatura sobre a geração dos *K* primeiros centróides. Em sua forma clássica, esta escolha é feita de forma aleatória. Porém, encontramos também implementações onde os *k* primeiros objetos do conjunto de dados originais são escolhidos como os primeiros centróides (TEKNOMO, 2007). Todavia, outras soluções de inicialização dos centróides poderiam ser usadas, tais como a aleatória ou a técnica *farthest-first* (DASGUPTA, 2002). Esta consiste em encontrar *k* centróides, de forma que eles estão o mais distante uns dos outros. Nesta técnica, inicialmente escolhe-se um primeiro centróide aleatoriamente. O próximo centróide escolhido é o que é mais distante deste primeiro. O processo se repete até que os *k* centróides tenham sido escolhidos. Esta técnica é usada também para construir grupos hierárquicos, com bom desempenho em cada nível da hierarquia (DASGUPTA, 2002). Entretanto, nossa escolha na implementação do *Appraisal* foi a de utilizar sempre os *K* primeiros elementos da tabela como centróides iniciais, para garantir que, a cada rodada de uma instância dos planos de imputação, os grupos gerados fossem os mesmos.

5.1.4.3.2 A escolha da medida de distância a ser usada

A associação dos objetos aos centróides se dá pelo cálculo da distância de cada objeto a todos os centróides gerados, e vinculando o objeto ao centróide que produziu a menor distância de todas as geradas. A versão clássica do algoritmo dos *K* centróides utiliza a distância Euclidiana como métrica. Todavia, assim como na implementação do algoritmo dos *k* vizinhos mais próximos, consideramos também no *Appraisal* o uso da distância *Manhattan*, para fins de comparação de qualidade do resultado final gerado pelo sistema. Observe os resultados:

	Euclidiana	*Manhattan*
Iris Plants	**92** (51,11%)	**88** (48,89%)
Pima Indians	**160** (44,44%)	**200** (55,56%)
Breast Cancer	**208** (51,36%)	**197** (48,64%)

Podemos notar que, nas três bases, os resultados foram bastante equilibrados. Isto indica que a utilização da métrica de similaridade não é fator determinante na qualidade do dado imputado em relação ao processo de agrupamento.

5.1.4.3.3 A escolha do número de centróides

Pela não existência prévia de um estudo sobre o número de grupos a serem adotados na tarefa de agrupamento aplicada à complementação de dados ausentes, decidimos variar o número de grupos (número de centróides *K*) entre a unidade e o número de registros válidos de cada base de dados. Assim, a variação aconteceu segundo o especificado abaixo:

1) Base *Iris Plants*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *K*	1-135	1-120	1-105	1-90	1-75

2) Base *Pima Indians*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *K*	1-353	1-314	1-275	1-236	1-196

3) Base *Breast Cancer*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *K*	1-614	1-546	1-478	1-410	1-341

Registramos, para plano de imputação vencedor que envolvia agrupamento, o número de grupos gerados. Assim, conseguimos os melhores números de grupos (valores de *K* centróides), listados abaixo:

	Menor *K*	Maior *K*
Iris Plants	2	60
Pima Indians	2	44
Breast Cancer	1	50

Os valores acima revelam que o número ideal de centróides varia numa faixa ampla. Todavia, os valores acima indicam que o aumento do número de grupos atinge um valor limite, significando que a formação dos grupos só é válida com a existência de uma quantidade mínima de elementos dentro de cada um deles. Os

testes nos mostram que muitos grupos com poucos elementos não nos oferecem resultados satisfatórios em nenhum dos casos.

5.1.4.3.4 O número de iterações de cada rodada do algoritmo

O algoritmo dos *K* centróides completa uma iteração quando todos os objetos estão associados aos grupos formados, ou quando um número limite de iterações é alcançado. Decidimos estabelecer este limite em 1000 iterações, pois acreditamos ser esse um valor razoável para que a convergência do algoritmo aconteça. Todavia, a escolha deste limite não foi baseada em nenhuma teoria ou experimento prévio, pois nenhum dos trabalhos que tivemos acesso discute a faixa de possíveis valores deste parâmetro.

5.1.4.4 Parâmetros do algoritmo de Análise de Componentes Principais

Nosso objetivo é o de realizar, em algumas das estratégias, uma seleção de variáveis utilizado a análise de componentes principais. Todavia, não é nossa intenção modificar o conjunto original de dados, mas apenas saber quais os atributos da tabela da base de dados tem maior relevância.

Assim, para implementar a seleção de atributos, o sistema *Appraisal* cria um objeto que contém uma lista ordenada decrescente em memória principal, e que, quando solicitado, devolve as *p* colunas mais importantes da tabela (*p* é parâmetro do método), a partir do vetor de autovalores produzido pelo algoritmo de Análise de Componentes Principais (PCA). Estes atributos serão utilizados pelas estratégias que envolvam seleção (*Seleção* $\rightarrow$ *Imputação*, *Seleção* $\rightarrow$ *Agrupamento* $\rightarrow$ *Imputação* e *Agrupamento* $\rightarrow$ *Seleção* $\rightarrow$ *Imputação*). As listas de prioridade decrescente das bases utilizadas em nossos experimentos são apresentadas a seguir:

- Base *Iris Plants*
  1º) *petallength*
  2º) *petalwidth*
  3º) *sepalwidth*
  4º) *sepallength*
- Base *Pima Indians Diabetes*
  1º) *pedigree_function*
  2º) *serum_insulin*

3º) *age*

4º) *blood_pressure*

5º) *skin_fold_thickness*

6º) *body_mass*

7º) *glucose_concentration*

8º) *pregnancy_times*

- Base *Wisconsin Breast Cancer*

1º) *Single_Epithelial_Cell_Size*

2º) *Normal_Nucleoli*

3º) *Bare_Nuclei*

4º) *Mitoses*

5º) *Marginal_Adhesion*

6º) *Uniformity_of_Cell_Shape*

7º) *Bland_Chromatin*

8º) *Clump_Thickness*

9º) *Uniformity_of_Cell_Size*

Nesta tese, o método *filter* (seção 2.5.2) será o utilizado, já que nosso objetivo é o de avaliar a qualidade dos dados de gerados pelo nosso sistema de imputação.

Durante a execução do módulo *Crowner* do sistema *Appraisal*, os planos que previamente selecionam atributos acessam o arquivo *pca.properties*, e verificam quantas colunas devem ser utilizadas no processo de imputação. Com isso, realizamos testes com o número de atributos variando entre um e o total de atributos numéricos da tabela.

5.1.4.5 Implementação e parâmetros da rede neuronal back propagation

O *Appraisal* utiliza como um de seus regressores uma rede neuronal com arquitetura *back propagation*. Para entender o seu funcionamento, abordaremos nesta subseção a origem das redes neuronais, qual o seu paradigma, e a partir disto desenvolveremos as idéias ligadas à construção do aprendizado supervisionado com esta arquitetura de rede.

Para implementar a rede neuronal *back propagation*, que é usada tanto como um dos regressores quanto como classificador, utilizamos o pacote JOONE – *Java Object Oriented Neural Engine* (MARRONE, 2007). JOONE é um *framework*

desenvolvido em Java para construir e executar aplicações de Inteligência Artificial, baseadas em redes neuronais. Aplicações Joone podem ser construídas em máquinas locais, treinadas em um ambiente distribuído, e executadas em qualquer dispositivo. Ele é constituído por uma arquitetura modular, baseada em componentes acopláveis, que podem ser estendidos para construir novos algoritmos de aprendizado e arquiteturas de redes neuronais. Seu propósito é criar aplicações de Inteligência Computacional com código Java usando a sua API.

As aplicações JOONE são construídas utilizando os seus objetos como componentes. Estes são módulos anexáveis, reusáveis e persistentes. Desta forma pode-se construir aplicações combinando estes componentes com editores gráficos, e controlando a lógica com scripts.

JOONE pode ser usado para construir sistemas customizáveis, adotado de maneira embutida para aumentar as funcionalidades de uma aplicação existente, ou empregado para construir aplicações de dispositivos móveis. No seu núcleo, os componentes são elementos importantes para construir qualquer arquitetura de rede neuronal.

Como as camadas da rede são construídas como objetos, a camada intermediária pode ser composta por diversos níveis, não necessariamente conectados em série. A figura 5.1, extraída de (MARRONE, 2007), exemplifica esta idéia:

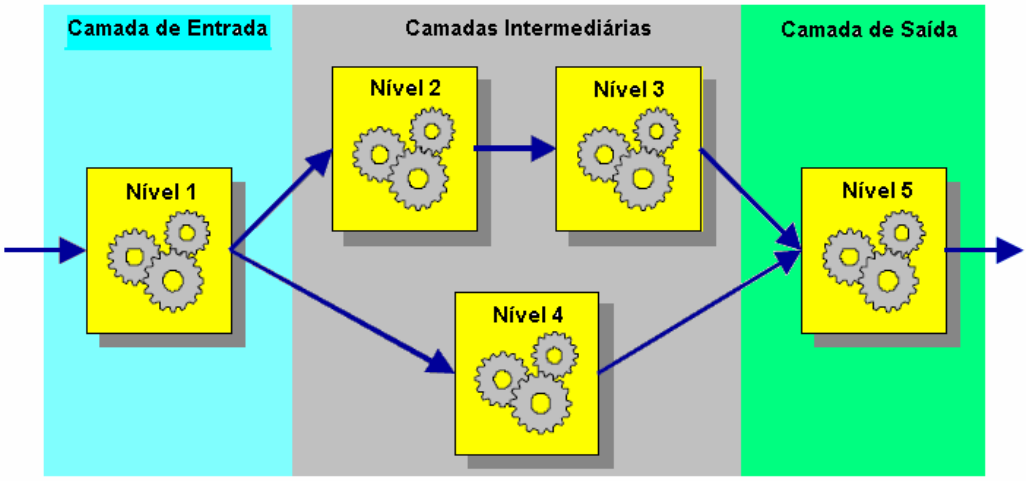


Figura 5.1 Esquema de interligação de camadas do JOONE.
Fonte: Adaptado de (*MARRONE, 2007*)

A documentação do sistema sugere que a camada de entrada da rede utilizada sempre seja linear, com pesos iguais à unidade. Com isso, ela funciona como uma

área de transferência para as camadas intermediárias, transferindo os dados sem modificações. Isto acontece, pois, se os dados de entrada fossem conectados a uma camada não linear, seria impossível enviar os dados de entrada a mais de uma camada, já que os componentes de entrada só podem ser anexados a uma única camada (Figura 5.2).

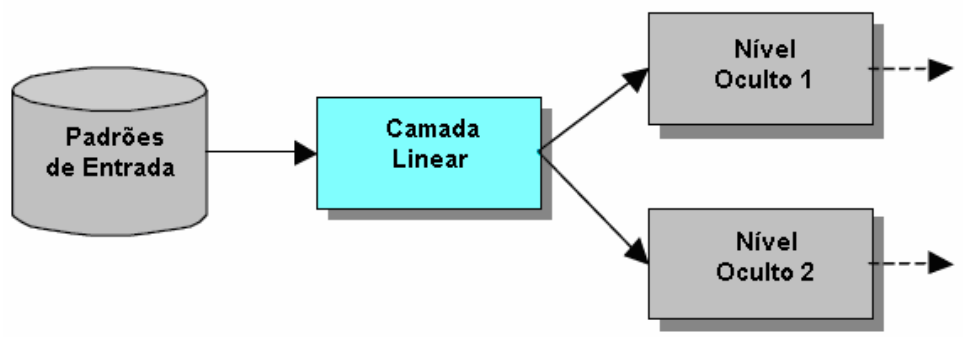


Figura 5.2 Esquema de interligação dos dados externos com a camada de entrada. Fonte: Adaptado de (*MARRONE, 2007*)

O sistema JOONE implementa diversos tipos de camadas. Com exceção da camada de entrada, a que utilizamos no sistema *Appraisal* para modelar a rede neuronal *back propagation* são do tipo sigmóide, para implementar o neurônio não linear. Camadas deste tipo só geram valores entre zero e um, dada a característica da função. Assim, os dados são normalizados por um filtro antes de seguirem para a camada de entrada, e têm os seus valores desnormalizados quando da produção dos resultados pela camada de saída. Isto permite que a rede não se torne susceptível a variações nos dados originais. A sua relativização faz com que as oscilações da rede se tornem estáveis e proporcionais, permitindo sua melhor utilização.

A componente principal que está sempre presente em cada rede neuronal JOONE é o objeto *Monitor*. Ele representa o ponto central no qual estão registrados todos os parâmetros que os demais componentes precisarão para funcionar corretamente, tais como a taxa de aprendizado, o *momentum*, o ciclo corrente, entre outros. Cada componente de uma rede neuronal (ambas camadas e sinapses) recebe um ponteiro com uma instância de um objeto *monitor*. Esta instância pode ser diferente para cada objeto, mas usualmente somente uma única instância é criada e usada. assim, cada componente pode acessar os mesmos parâmetros para a rede neuronal.

Dentre os parâmetros do monitor mais relevantes às nossas experiências, destacam-se a taxa de aprendizado (*learningRate*), o (*momentum*) e o número de ciclos de treinamento da rede (*cycles*). Os valores de *learningRate* e *momentum* devem estar na faixa [0,1]. Todavia, (MARRONE, 2007) cita que bons valores para estes parâmetros só são obtidos através de experimentos repetidos. Adotamos, seguindo a sugestão da documentação do JOONE, os valores *learningRate* = 0.5, *momentum* = 0.7 e *cycles* = 3000 para todas as experiências implementadas pelo *Appraisal* utilizando redes neuronais *back propagation*.

Para entender o papel destes parâmetros no funcionamento da rede, verifiquemos como eles são aplicados. O parâmetro *learningRate* indica o ritmo no qual a rede caminha na busca do mínimo da função de erro de *n* dimensões, que existe em função dos pesos das sinapses da rede. Treinar a rede significa modificar os pesos das sinapses, e encontrar a melhor configuração de valores que geram um erro mínimo para certos padrões de entrada. Valores altos deste parâmetro fazem com que a rede "passe" do valor ótimo, e caminhe para uma outra região.

Já o parâmetro *momentum* é um parâmetro que se refere à tentativa de fuga da função de uma região de mínimo local. Como o objetivo da minimização da rede é a de encontrar o mínimo global da função, se a rede encontrar um mínimo local poderá acreditar que aquele é o mínimo global. Então, ao se encontrar este mínimo, o valor do parâmetro *momentum* é adicionado a ele. Caso ele não seja o mínimo global, a rede terá condições de sair daquele vale e continuar a sua busca pelo menor valor da função.

O parâmetro *cycles* informa à rede quantas vezes o seu processo de treinamento deve acontecer. Assim como os demais parâmetros, não existe um valor que possamos considerar ótimo. O que fizemos nos experimentos foi variar o valor de *cycles* entre 1.000 e 10.000. Verificamos que a rede atingiu índices consideráveis de diminuição do erro com valores de *cycles* entre 3.000 e 5.000. Desta forma, dado o grande volume de experimentos feitos, decidimos utilizar o menor valor que atendia de forma satisfatória nossas necessidades de simulação (*cycles* = 3.000). Já para a execução do comitê de complementação de dados, obtemos melhores valores com o valor de *cycles* = 5.000.

Uma sinapse pode ser tanto de saída de uma camada e a entrada da camada seguinte da rede neuronal, já que ela representa um recurso compartilhado entre duas

camadas (não mais que duas, porque uma sinapse pode ser anexada a apenas uma camada de entrada e uma de saída). Assim, as sinapses são sincronizadas, para evitar que a camada que a usa como entrada tente lê-las antes de a camada de saída anterior escrever nela. Isto evita que duas camadas acessem simultaneamente uma sinapse compartilhada.

JOONE também implementa diversos tipos de sinapses, tais como a Direta, Completa, *Delayed* (com atraso), Kohonen, e Sanger. A que utilizamos em nosso trabalho foi a Completa, pela característica da rede neuronal *back propagation*. Como o próprio nome diz, este tipo de sinapse conecta todos os neurônios de uma camada com todos os da outra camada. Seus pesos mudam durante o processo de aprendizado.

5.1.5 Métricas

5.1.5.1 Medida do erro do processo de imputação

Os valores gerados pelos algoritmos de imputação de valores ausentes foram avaliados pela medida do erro relativo absoluto (RAD – *Relative Absolute Deviation*), que é calculado da seguinte forma:

$$RAD_X(i) = \frac{|a - a'|}{a}$$

onde *a* é o valor original do atributo *X* da tupla *i* e *a'* é o valor imputado do atributo *X* nesta tupla *i*. Esta foi a medida adotada no trabalho de TSENG, WANG e LEE (2003), abordada no capítulo 3. Escolhemos esta média por acreditarmos que ela melhor representa um dos fatores que desejamos medir: o quão distante o dado imputado está do valor original. Assim, conseguimos tornar relativo o erro absoluto, e nos tornamos menos susceptíveis à ordem de grandeza dos valores. Por exemplo, imagine que tenhamos dois atributos *x* e *y* de uma tupla, com valores originais 3 e 45. Considere que, após executar um algoritmo de imputação, cheguemos a valores x' e y' iguais a 4 e 40, respectivamente. Desta forma, os erros relativo absoluto destas duas medidas são dados por:

$$RAD_x(i) = \frac{|3-4|}{3} \approx 33{,}3\% \qquad RAD_y(i) = \frac{|45-40|}{45} \approx 11{,}1\%$$

Note que, se avaliássemos o resultado das imputações acima pelo erro absoluto (E_{ABS} = |*x*-*x'*|), teríamos os seguintes valores: E_{ABSx} = |3-4| = 1 e E_{ABSy} = |45-40| = 5, ou seja, o erro absoluto da imputação de *x* é cinco vezes maior do que o erro absoluto da complementação da variável *y*. Entretanto, notamos que, com o erro relativo absoluto, o erro da imputação de *y* é de magnitude bem inferior (1/3) do erro relativo à imputação de *x*. Assim, decidimos adotar esta medida, e, para todos os erros medidos para cada imputação, calculamos a sua média aritmética simples, definindo assim o erro médio relativo absoluto de um atributo *X* (RAD_X), gravando, desta forma, o erro médio relativo absoluto de um processo de imputação em uma tabela. Este erro médio pode ser expresso da seguinte forma:

$$\overline{RAD}_X = \frac{1}{m}\sum_{i=1}^{m} RAD_X(i)$$

sendo *m* o número de imputações realizadas na tabela.

Entretanto, a escolha do erro relativo absoluto nos fez confrontar com uma situação inicialmente não observada: como medir o erro quando o valor original na coluna for zero? Já que o denominador da medida de erro é o valor original, seu uso se torna inviabilizado quando esta situação ocorre.

Assim, decidimos adaptar a idéia existente de **similaridade** utilizada no trabalho de MAGNANI e MONTESI (2004), tratada no capítulo 3, na tarefa de agrupamento. Retomando a idéia, o algoritmo de similaridade definido pelos autores utiliza uma função de similaridade que mede, para cada atributo de um registro, o quão parecidos os registros são. Veja a equação abaixo:

$$sim(a_k, b_k) = 1 - \frac{|a_k - b_k|}{|\max_X(k) - \min_X(k)|}$$

Aqui, *a* e *b* são duas tuplas de uma tabela, *a*(*k*) indica o atributo *k* das tuplas *a* e *b*, $\max_X(k)$ e $\min_X(k)$ são respectivamente o maior e o menor valores apresentados na tabela para a coluna *k* para todas as tuplas *x* da tabela.

Duas tuplas são consideradas idênticas quando o valor da função $sim(a_k, b_k) = 1$. Quando isto acontece, o valor da diferença em módulo $|a_k - b_k|$ é igual a zero, o que nos faz deduzir que os atributos a_k e b_k possuem o mesmo valor. Se considerarmos apenas o

termo fração da função ($\frac{|a_k - b_k|}{|\max_X(k) - \min_X(k)|}$), podemos interpretar que este cálculo diz o quão diferente os atributos a_k e b_k são, relativos aos valores extremos encontrados na coluna. Assim, sem perda de generalidade, podemos considerar esta como uma possível medida de erro. E baseado neste raciocínio medimos o quão distante uma medida *x'*, imputada para substituir um valor ausente de um atributo *x* de uma tupla, com a medida de **erro de similaridade**, expressa abaixo:

$$E_{SIM} = \frac{|x - x'|}{|\max_x - \min_x|}$$

Utilizamos esta medida na base *Pima Indians Diabetes*, no atributo *pregnancy_times* (número de vezes que a paciente engravidou), onde valores zero ocorriam em vários registros.

5.1.5.2 Medida da preservação das características das tuplas imputadas

A medida da qualidade da imputação é sem sombra de dúvidas importante no processo de complementação de dados ausentes. Todavia, é da mesma forma relevante saber se o valor imputado preserva as características do registro, e mantém as relações existentes entre os atributos.

Desta forma, decidimos também avaliar se ao submetermos a tupla com valor imputado a um novo processo de classificação, a mesma continua sendo levada à sua classe. Cabe lembrar que durante todo o processo de complementação de dados, o atributo classe é desconsiderado.

Para atingir este objetivo, implementamos uma rede neuronal *back propagation* que classifica as tuplas das bases utilizadas nesta tese com a representação binária econômica (capítulo 2). Nesta opção, os atributos numéricos da tabela são as entradas da rede, e os valores do domínio são codificados em binário, e o número codificado é utilizado na camada de saída da rede.

Treinamos a rede classificadora com as tuplas que não apresentam valores ausentes. Isso significa que, quanto mais tuplas com valores ausentes a base possuir, menos tuplas serão enviadas para o treinamento do classificador. Os testes foram feitos com as tuplas que tiveram atributos ausentes imputados. Com isso, analisamos os casos onde as tuplas foram categorizadas de forma errônea. Para avaliar a qualidade da rede

classificadora, também submetemos ao processo de classificação as tuplas que tiveram valores imputados, porém na sua versão original. Como durante todo o processo a base original está à nossa disposição, identificamos quais tuplas apresentavam valores ausentes na base modificada, e as reclassificamos, comparando o resultado da rede com a classe original.

Assim, a medida do erro de classificação E_{CR} de um registro R de um atributo imputado X é calculada da seguinte forma:

$$E_{CR}(X) = \begin{cases} 1, se\, classe\, designada(R) \neq classe\, original(R) \\ 0, se\, classe\, designada(R) = classe\, original(R) \end{cases}$$

e o erro médio de classificação do atributo imputado X na base de dados B é dado por:

$$E_C(X) = \frac{1}{n} \sum E_{CR}(X) \; \forall R \in \mathrm{B}$$

Com este procedimento, pretendemos confrontar, para cada plano de imputação, a relação entre a taxa média de erro e a classificação do processo de imputação.

5.1.6 Condições ambientais dos experimentos

Realizamos os testes em um microcomputador com processador *Pentium Celeron D* 3 GHz, com 512 MB de memória principal, e 74,4 GB de disco rígido IDE, com o sistema operacional *Windows XP Professional* versão 2002 *Service Pack* 2 instalado. O *Appraisal* foi desenvolvido utilizando linguagem Java no ambiente Eclipse SDK versão 3.2.0 e JDK 1.5.0.11. Os dados estão armazenados em um Sistema Gerenciador de Banco de Dados MySQL versão 4.1.1.

5.2 Resultados dos experimentos

5.2.1 Matrizes de Correlação de Atributos

É de extrema importância que toda a análise realizada sobre o processo de imputação de valores ausentes leve em consideração inicialmente as matrizes de correlação dos conjuntos de dados que serviram de base para os testes.

Assim, reportando-nos à seção 5.1.2, explicitamos e avaliamos a seguir as matrizes de correlação das bases usadas em nossos experimentos. Tendo em vista que a matriz de correlação é simétrica, com todos os elementos de sua diagonal iguais a um,

destacaremos apenas a sua parte inferior, que explicita a informação de correlação entre atributos que desejamos saber.

5.2.1.1 Base Iris Plants

	sepallength	*sepalwidth*	*petalwidth*	*petallength*
sepallength	1.00	-0.11	0.82	0.87
sepalwidth	**-0.11**	1.00	-0.36	-0.42
petalwidth	**0.82**	**-0.36**	1.00	0.96
petallength	**0.87**	**-0.42**	**0.96**	1.00

A matriz nos mostra que os atributos *petallength*, *petalwidth* e *sepallength* possuem correlações significativas entre si, entre 0.82 e 0.96 (a mais alta, entre *petallength* e *petalwidth*). Por outro lado, os baixos valores de correlação entre atributo *sepalwidth* e os demais revelam que suas medidas não estão atreladas aos valores dos demais atributos.

5.2.1.2 Base Pima Indians Diabetes

Atributo	1	2	3	4	5	6	7	8
1	1.00	0.14	0.16	0.16	-0.02	0.09	0.14	0.01
2	**0.14**	1.00	0.21	0.20	0.21	0.34	0.58	0.20
3	**0.16**	**0.21**	1.00	0.66	0.30	0.07	0.23	-0.03
4	**0.16**	**0.20**	**0.66**	1.00	0.23	0.17	0.18	0.09
5	**-0.02**	**0.21**	**0.30**	**0.23**	1.00	0.30	0.10	0.21
6	**0.09**	**0.34**	**0.07**	**0.17**	**0.30**	1.00	0.22	0.68
7	**0.14**	**0.58**	**0.23**	**0.18**	**0.10**	**0.22**	1.00	0.08
8	**0.01**	**0.20**	**-0.03**	**0.09**	**0.21**	**0.68**	**0.08**	1.00

Atributos:

1: *pedigree_function*
2: *glucose_concentration*
3: *body_mass*
4: *skin_fold_thickness*
5: *blood_pressure*
6: *age*
7: *serum_insulin*
8: *pregnancy_times*

A matriz de correlação da base *Pima Indians Diabetes* nos apresenta um conjunto de dados onde suas colunas têm pouca correlação entre si, ou seja, o seu grau de independência é considerável. As maiores medidas de correlação não chegam a 70% (as maiores são as que existem entre os atributos *age* e *pregnancy_times*, de 0.68; *body_mass* e *skin_fold_thickness*, com 0.66; e *serum_insulin* e *glucose_concentration*,

com 0.58). Esta característica, unida com o fato de o mecanismo de ausência dos dados ser completamente aleatório (MCAR), pode tornar mais difícil o processo de complementação de dados nesta base. Avaliaremos esta conseqüência mais à frente.

5.2.1.3 Base Wisconsin Breast Cancer

Atributo	1	2	3	4	5	6	7	8	9
1	1.00	0.64	0.76	0.91	0.71	0.46	0.69	0.72	0.75
2	**0.64**	1.00	0.55	0.65	0.49	0.35	0.59	0.53	0.52
3	**0.76**	**0.55**	1.00	0.74	0.67	0.35	0.68	0.67	0.62
4	**0.91**	**0.65**	**0.74**	1.00	0.69	0.44	0.71	0.72	0.72
5	**0.71**	**0.49**	**0.67**	**0.69**	1.00	0.42	0.67	0.60	0.59
6	**0.46**	**0.35**	**0.35**	**0.44**	**0.42**	1.00	0.34	0.43	0.48
7	**0.69**	**0.59**	**0.68**	**0.71**	**0.67**	**0.34**	1.00	0.58	0.59
8	**0.72**	**0.53**	**0.67**	**0.72**	**0.60**	**0.43**	**0.58**	1.00	0.63
9	**0.75**	**0.52**	**0.62**	**0.72**	**0.59**	**0.48**	**0.59**	**0.63**	1.00

Atributos:

1: *Uniformity_of_Cell_Size*
2: *Clump_Thickness*
3: *Bland_Chromatin*
4: *Uniformity_of_Cell_Shape*
5: *Marginal_Adhesion*
6: *Mitoses*
7: *Bare_Nuclei*
8: *Normal_Nucleoli*
9: *Single_Epithelial_Cell_Size*

Ao contrário do que pudemos observar com a base *Pima Indians Diabetes*, o conjunto de dados *Wisconsin Breast Cancer* apresenta oito de seus nove atributos com índices razoáveis de correlação (valores maiores do que 0.5), a menos do atributo *Mitoses*, que possui correlações menores que 0.5 para todas as demais colunas. A maior correlação entre colunas acontece entre os atributos *Uniformity_of_Cell_Size* e *Uniformity_of_Cell_Shape* (0.91).

5.2.2 Estratégias de complementação de dados

5.2.2.1 Análise dos resultados por base de dados

Os gráficos desta seção apontam quantas vezes cada uma das estratégias apresentou o melhor desempenho (menor erro relativo absoluto) frente às demais. Esta apresentação é feita de duas formas: na primeira, contamos quantas vezes cada estratégia venceu frente às demais em todos os atributos de uma base de dados. Na segunda, analisamos quantas vezes cada uma das estratégias revelaram os melhores resultados de todos por base e por percentual de valores ausentes. Com isso,

queremos avaliar se o aumento do número de dados nulos nas colunas influencia de alguma forma o desempenho das estratégias.

A alta correlação entre três dos quatro atributos da base *Iris Plants* faz com que o agrupamento seja importante no processo de imputação. Por esta razão, o plano de imputação envolvendo somente o agrupamento foi o que apresentou melhores resultados em 60% dos casos. Se considerarmos os demais planos que obtiveram bons resultados, o agrupamento aparece em quase todos eles, resultando em um total de 95% de ocorrências bem sucedidas envolvendo de alguma forma esta tarefa.

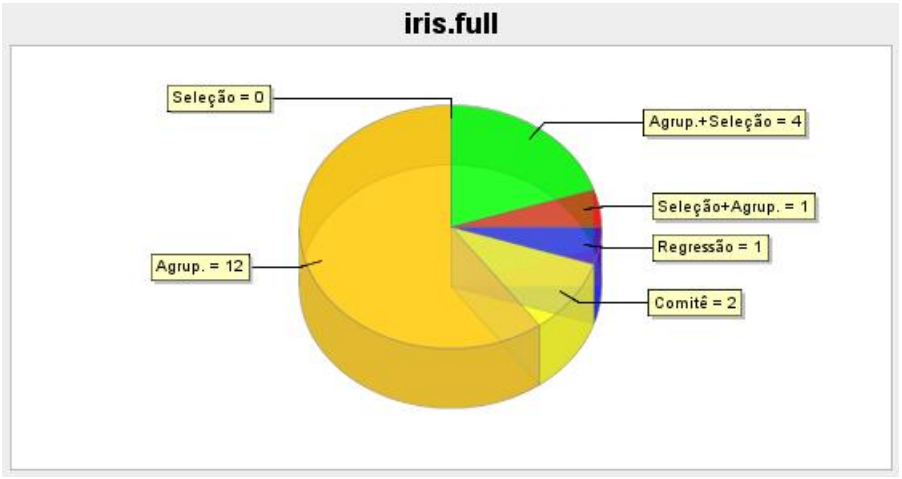
Com relação à base *Pima Indians* observamos uma situação um pouco diferente da anterior. Por conta da baixa correlação entre os atributos, a seleção de atributos, juntamente com o agrupamento, também se mostra importante no processo de complementação de dados ausentes, já que ela descarta atributos que não interferem no processo de imputação. Estratégias envolvendo seleção e agrupamento foram bem sucedidas em 52% dos casos. Quando consideramos todas as estratégias envolvendo agrupamento, este índice sobe para 85%.

Assim como no caso anterior, a base *Wisconsin Breast Cancer* indica que a conjunção da seleção e agrupamento predominaram os resultados comparativos das estratégias. Em 51%, a combinação destas duas tarefas revelou um erro médio menor do que as demais estratégias. Quando envolvemos também o agrupamento, este resultado chega a 84%.

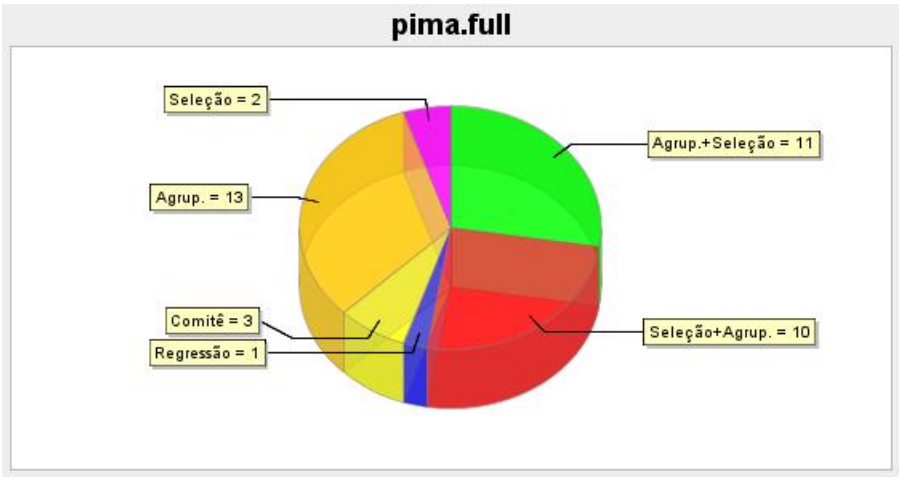
Esta análise nos leva a deduzir que o prévio agrupamento de dados precedendo a complementação de valores ausentes é de grande importância, já que a redução da amostra faz com que as tuplas similares à regredida melhorem a qualidade do processo de imputação.

5.2.2.2 Gráficos por base de dados

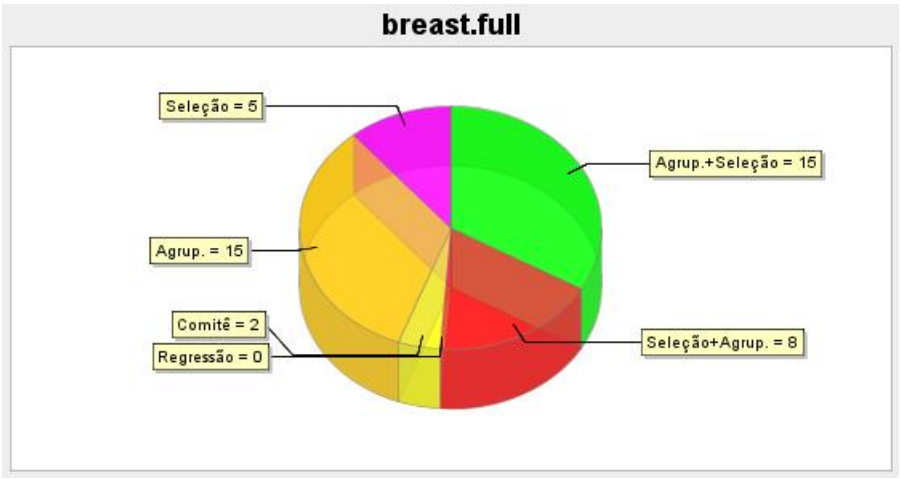
Iris Plants



Pima Indians Diabetes



Wisconsin Breast Cancer



5.2.2.3 Análise dos resultados por percentual de valores ausentes

À medida que aumenta o percentual de valores ausentes diminuímos a quantidade de informação disponível (e conseqüentemente o conhecimento oculto nos dados). Isto pode influenciar o treinamento da rede neuronal *back propagation*, o número de vizinhos do algoritmo dos *k* vizinhos mais próximos, e o tamanho dos grupos no processo de agrupamento. O que queremos avaliar com estes resultados é como esta diminuição de informação disponível afeta o desempenho das estratégias de complementação de dados. Analisaremos os resultados com resultados consolidados por base de dados.

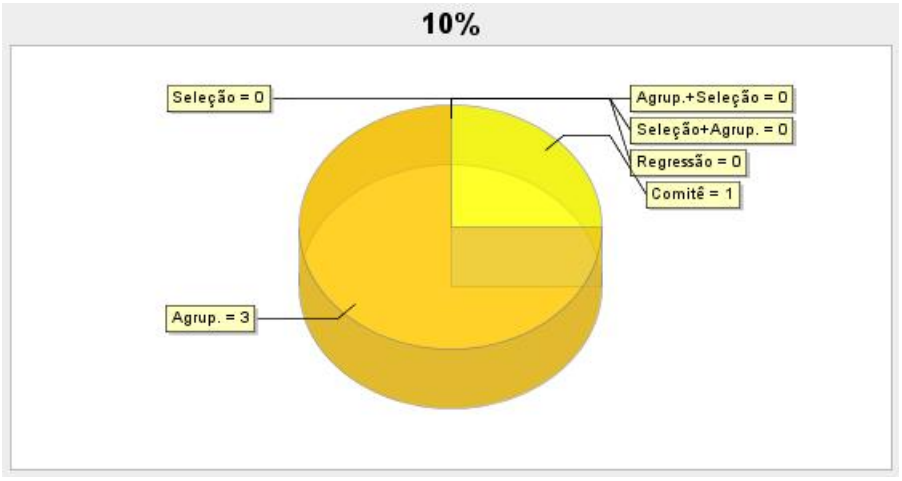
Na base *Iris Plants*, o agrupamento se mostrou a melhor estratégia em todos os casos, com exceção do percentual de 20% de valores ausentes. Porém, mesmo nesta exceção, uma estratégia composta envolvendo agrupamento foi a melhor sucedida. Acreditamos que este comportamento tenha ocorrido pelas características da base (atributos com valores variando dentro de um domínio de pequeno espectro, alta correlação entre a maioria dos atributos, poucas tuplas).

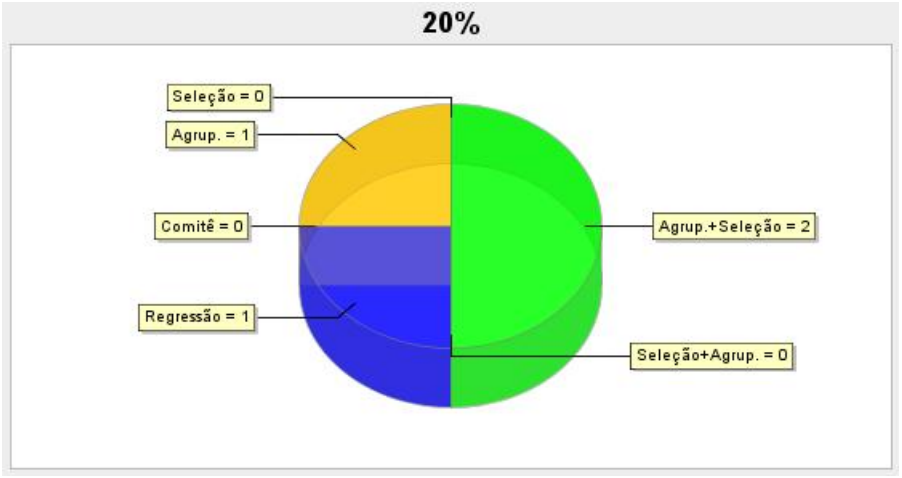
A seleção passa a desempenhar um papel mais importante nos resultados da base *Pima Indians Diabetes*. Os resultados revelam que estratégias compostas que se valem da seleção aparecem com os melhores índices de desempenho. Este comportamento nos leva a crer que a baixa correlação entre os seus atributos faça com que a seleção galgue um papel de maior destaque, pela eliminação de colunas que pouco ou nada contribuem no processo de imputação nesta base.

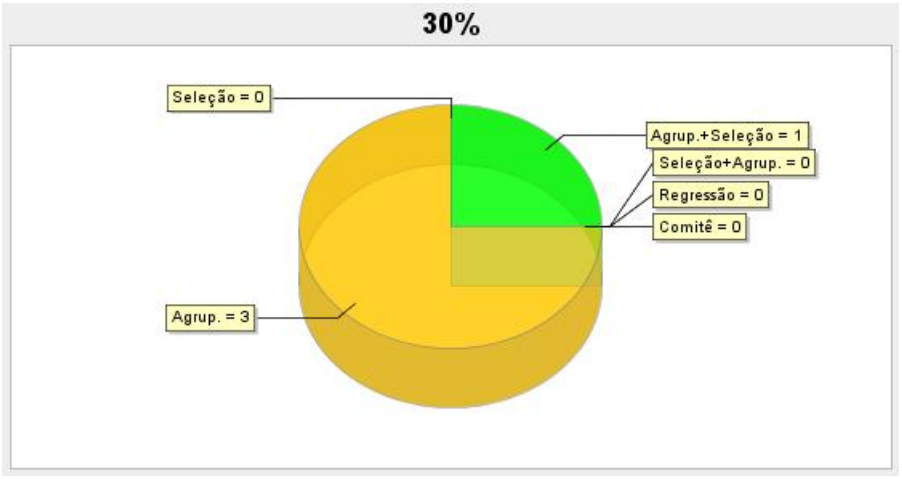
De forma geral, na base *Wisconsin Breast Cancer* a seleção influencia a qualidade do processo de imputação. Com um menor índice de valores ausentes, a estratégia de agrupamento seguida da seleção apresenta os melhores resultados. À medida que o número de tuplas com atributos ausentes aumenta, o agrupamento simples se destaca. Entretanto, mais uma vez destacamos o desempenho do agrupamento em todos os percentuais de valores ausentes nesta base, demostrando ratificar que esta é uma tarefa de extrema importância ao processo de complementação de dados ausentes.

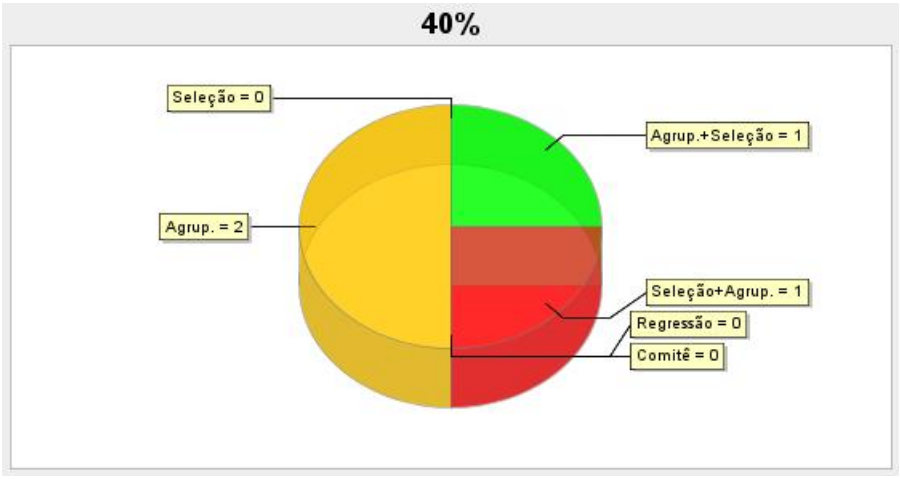
5.2.2.4 Gráficos por percentual de valores ausentes

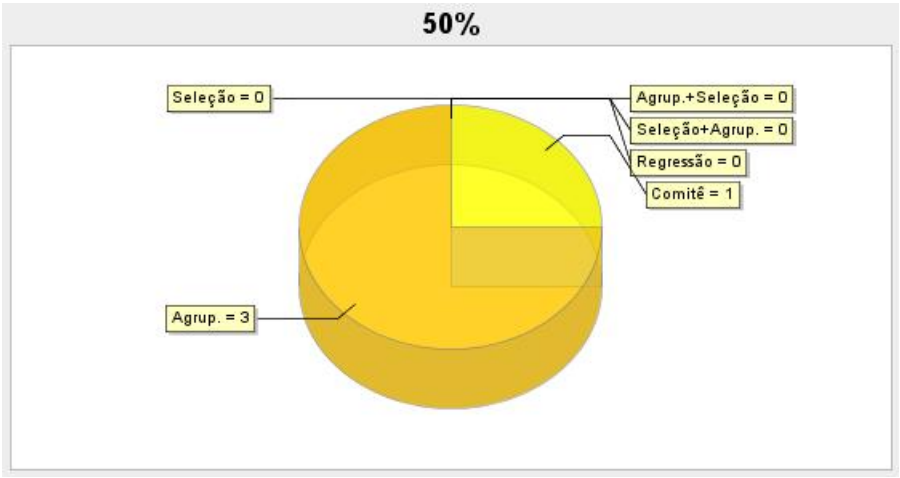
Iris Plants



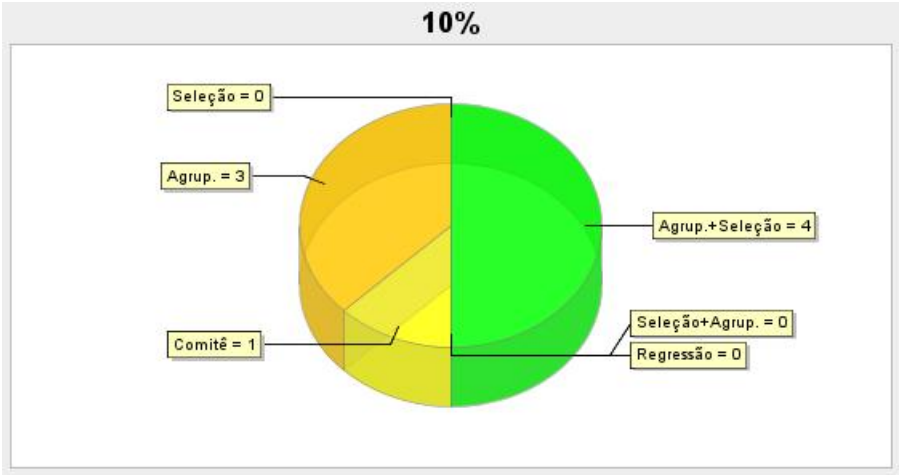


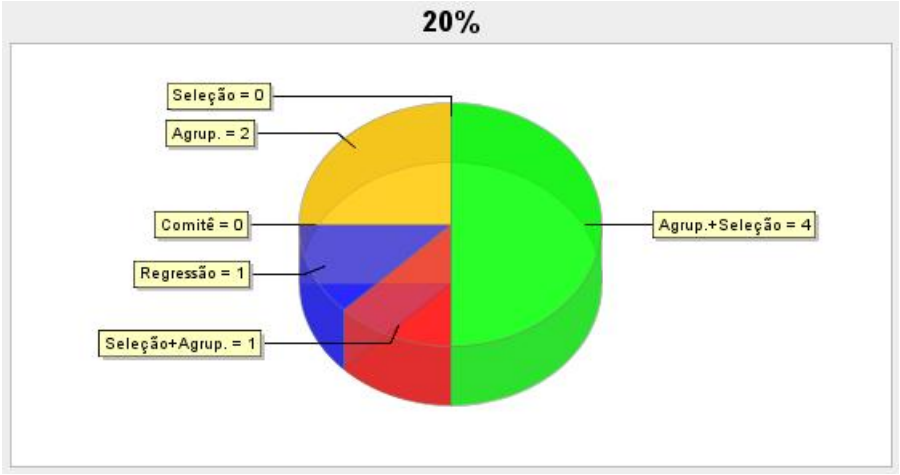


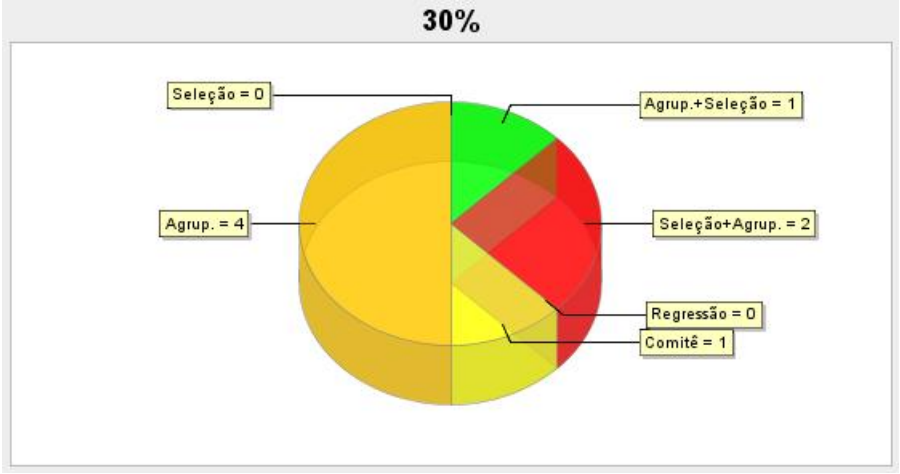


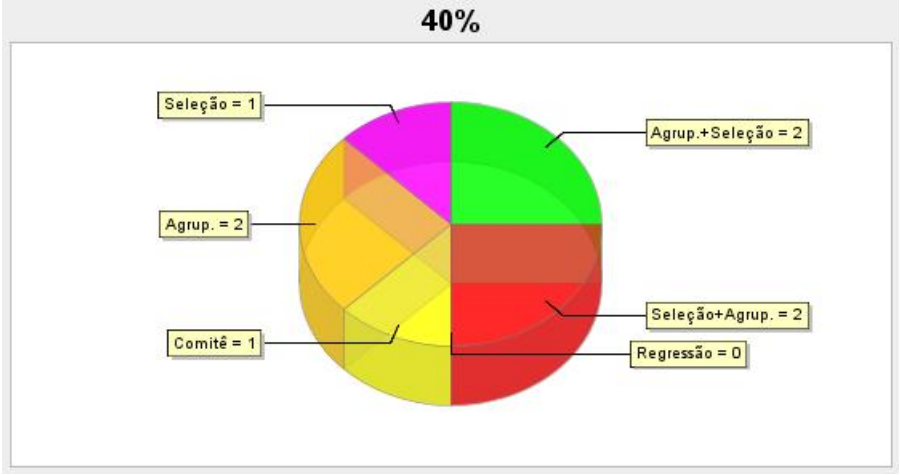


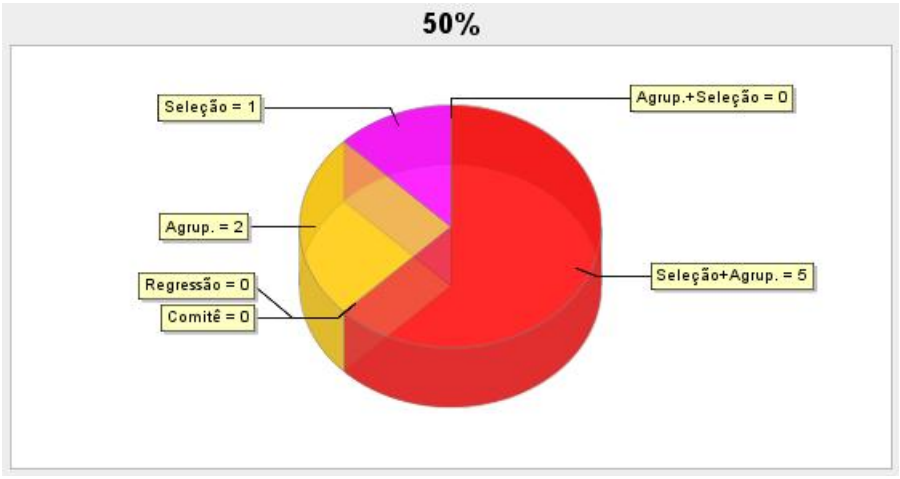
Pima Indians Diabetes



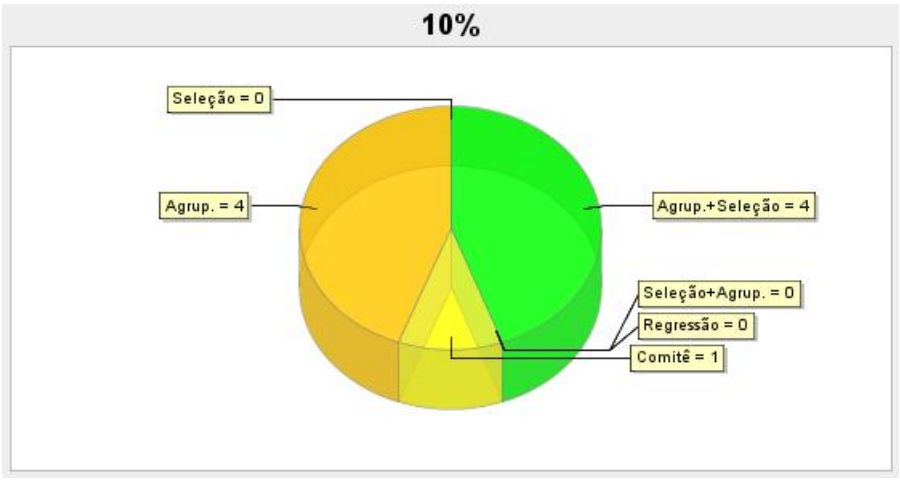

20%
Seleção = 0
Agrup. = 2
Comitê = 0
Regressão = 1
Seleção+Agrup. = 1
Agrup.+Seleção = 4

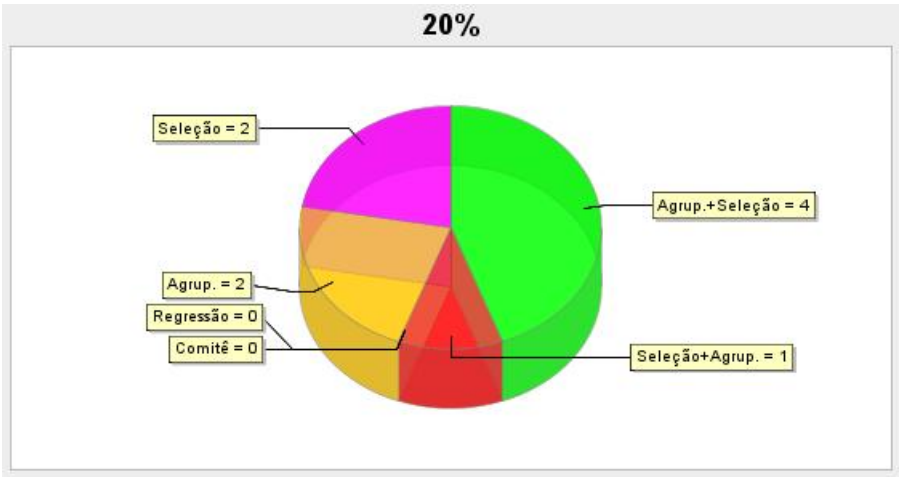

30%
Seleção = 0
Agrup. = 4
Agrup.+Seleção = 1
Seleção+Agrup. = 2
Regressão = 0
Comitê = 1

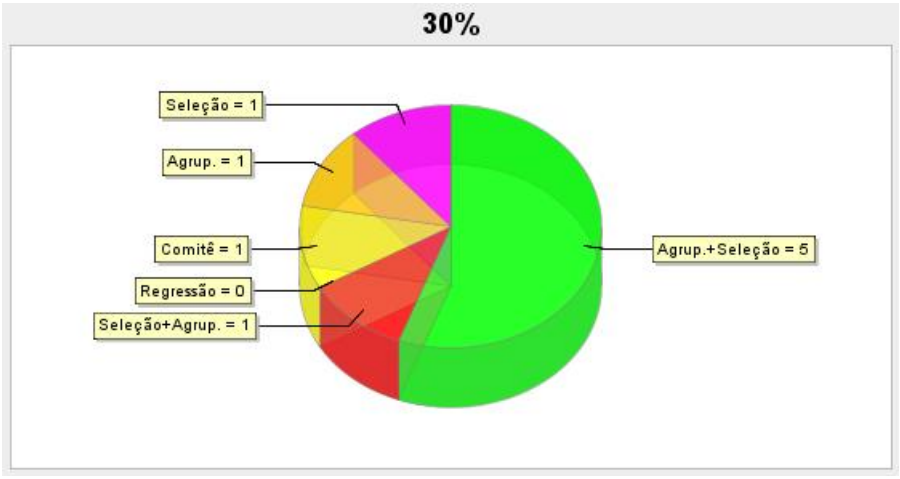

40%
Seleção = 1
Agrup. = 2
Comitê = 1
Agrup.+Seleção = 2
Seleção+Agrup. = 2
Regressão = 0

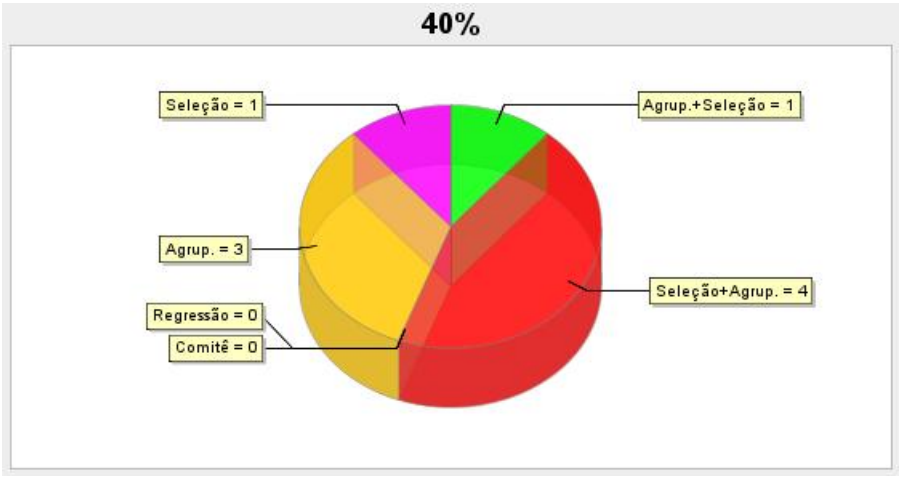


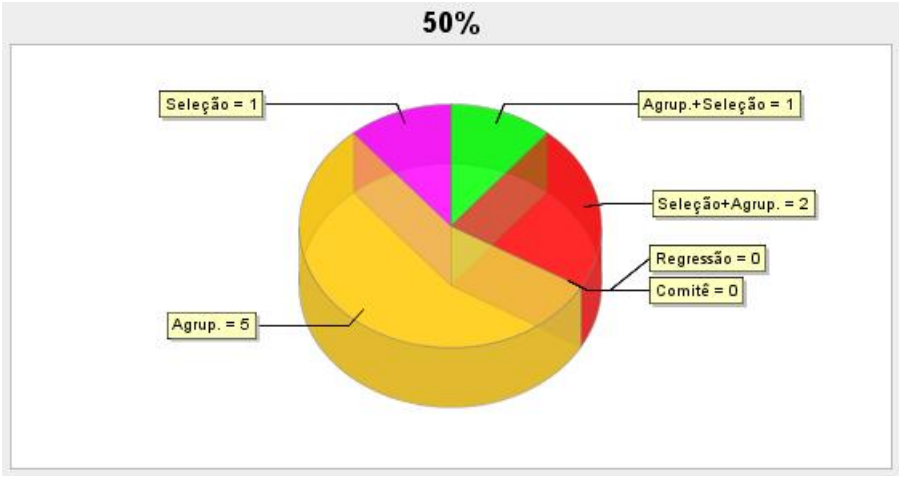
Wisconsin Breast Cancer






30%
Seleção = 1
Agrup. = 1
Comitê = 1
Regressão = 0
Seleção+Agrup. = 1
Agrup.+Seleção = 5


40%
Seleção = 1
Agrup. = 3
Regressão = 0
Comitê = 0
Agrup.+Seleção = 1
Seleção+Agrup. = 4


50%
Seleção = 1
Agrup. = 5
Agrup.+Seleção = 1
Seleção+Agrup. = 2
Regressão = 0
Comitê = 0

5.2.3 Execução dos planos de imputação

5.2.3.1 Análise dos resultados

Os resultados dos experimentos mostram que a imputação com a média é uma opção ruim, já que apresenta as maiores médias de erro relativo absoluto. Nas bases onde a correlação entre os atributos é significativa, a média quase sempre ficou em última colocação, galgando apenas um penúltimo lugar na base *Wisconsin Breast Cancer*. Quando a imputação por média passou a ser precedida pelo agrupamento com o algoritmo dos *K* centróides, o resultado tornou-se um pouco melhor, mas ainda bastante irregular em todas as bases de dados.

A seleção de atributos seguida do agrupamento e imputação com média piorou os resultados obtidos somente com a imputação com média precedida do agrupamento. Isto ocorreu de forma mais explícita nas bases *Iris Plants* e *Wisconsin Breast Cancer*, onde a correlação entre os seus atributos é alta. Na base *Pima Indians Diabetes*, a imputação precedida de seleção e agrupamento mostrou desempenho bastante irregular.

Quando experimentamos o agrupamento precedendo a seleção de atributos e a imputação por média observamos resultados mais regulares, apesar de ainda não tão satisfatórios, pelo fato de a imputação ter sido realizada utilizando a média. Porém, os resultados mostram que o agrupamento precedendo a seleção faz com que o processo de complementação de dados ausentes leve em conta os aspectos inerentes ao grupo, propiciando desta forma uma melhor seleção de atributos.

A imputação com o algoritmo dos *k* vizinhos mais próximos (*k*-NN) também se apresenta como uma opção instável entre as bases, apesar de seus resultados serem muito mais interessantes do que as obtidas com a média e com o uso de redes neuronais *back propagation*. A utilização dos *k* registros mais semelhantes no processo de imputação melhora a qualidade do dado imputado. Isto já foi observado nos diversos trabalhos apresentados no capítulo 3, e, em nossos experimentos, pudemos constatar este fato. Entretanto, os resultados apresentados a seguir mostram que a opção por estratégias compostas são mais interessantes do que a imputação com o algoritmo *k*-NN. Porém, seu uso não pode ser desconsiderado, principalmente quando o usuário não dispuser de tempo e recursos computacionais para complementar dados em uma base onde valores ausentes ocorram.

Os experimentos que envolveram a seleção de atributos precedendo a aplicação do algoritmo *k*-NN revelam resultados ruins para a base *Iris Plants*. Como esta base tem a característica de possuir poucos atributos, a seleção causa uma perda considerável de informação, que prejudica a determinação dos vizinhos que participarão do processo de imputação. Nas demais bases, a seleção melhorou os resultados obtidos com a imputação direta com *k*-NN. Acreditamos que isto deva ter ocorrido por se tratar de bases com mais atributos (pelo menos o dobro da *Iris Plants*) e tuplas (o dobro na *Pima Indians* e o quádruplo na *Breast Cancer*).

E mais uma vez a melhora do processo de imputação se torna melhor visualizada quando a correlação entre atributos da base é alta. Podemos verificar os bons resultados alcançados pelo plano de agrupamento precedendo a regresão com *k*-NN nas bases *Iris Plants* e *Wisconsin Breast Cancer*. A formação de grupos ajudou o algoritmo dos *k* vizinhos a melhor selecionar as tuplas mais similares. Na base *Pima Indians Diabetes*, os resultados não são tão bons. Todavia, isto não indica que o plano de agrupar antes de regredir com *k*-NN seja inadequado; apenas sinaliza para o fato de que outros planos se adequam melhor a bases com baixa correlação de atributos.

Os resultados mais animadores apareceram com a aplicação do plano de imputação envolvendo o agrupamento seguido de seleção de dados, e utilizando como algoritmo regressor o *k*-NN. Nas três bases de dados utilizadas, o desempenho foi bastante satisfatório e regular. A seleção feita em grupos previamente montados faz com que a seleção das características mais importantes auxilie positivamente a complementação de dados ausentes com o algoritmo *k*-NN, já que os vizinhos mais próximos são os melhores qualificados para participar do processo de imputação. Acreditamos que este é um dos resultados mais significativos obtidos nesta tese.

Também obtivemos bons resultados com a aplicação da seleção precedendo o agrupamento e a imputação com o algoritmo *k*-NN nas bases *Pima Indians Diabetes* e *Wisconsin Breast Cancer*. Como estas bases apresentam um bom número de atributos, a seleção teve um comportamento bastante satisfatório, principalmente na base *Pima Indians*, onde a correlação entre os atributos é fraca. Os resultados não foram tão bons na base *Iris Plants*, pelo motivo que já expusemos: quando selecionamos um subconjunto dos poucos atributos que a base possui, fazemos com

que a quantidade de informação disponível seja insuficiente para o processo de imputação.

Nossa idéia quando decidimos utilizar como um possível algoritmo de complementação de dados as redes neuronais *back propagation* foi a sua característica intrínseca de classificação e imputação. Acreditávamos obter bons resultados com o aprendizado supervisionado efetuado por esta arquitetura de rede neuronal, e esperávamos que seu desempenho fosse melhor do que realmente foi. Na base *Wisconsin Breast Cancer*, os resultados foram bem aquém do que poderíamos imaginar. Atribuímos este fato ao mecanismo de ausência de dados adotado, o completamente aleatório (MCAR). Como aproximadamente 65% dos registros desta base indicam pacientes sem a doença e 35% com portadores da malignidade, e dada a aleatoriedade da geração dos valores ausentes, os valores nulos podem ter se concentrado em uma das classes, fazendo com que a rede não fosse bem treinada para um dos padrões existentes. Na base *Pima Indians Diabetes*, a baixa correlação explica o fraco desempenho dos resultados. A rede neuronal não conseguiu, a partir dos dados fornecidos, capturar o padrão intrínseco existente neles. Os resultados na base *Iris Plants* se mostram muito irregulares, e a imputação com este plano nesta base não deve necessariamente ser descartada.

Quando analisamos os resultados da aplicação do plano de seleção e imputação com redes *back propagation*, os resultados da base *Wisconsin Breast Cancer* tornam-se melhores do que os com a imputação simples, porém ainda bastante irregulares. Já o agrupamento precedendo a imputação com redes *back propagation* mostram bons resultados apenas na base *Iris Plants*. Nas demais, o comportamento é bastante irregular. Isto indica que as mudanças promovidas no conjunto de treino da rede têm pouca influência na qualidade dos dados imputados, a menos que a base possua características bem comportadas, como é o caso da *Iris Plants*.

Quando utilizamos a imputação com o algoritmo *k*-NN, poucos vizinhos podem indicar a melhor opção de imputação. Ao utilizarmos uma rede neuronal *back propagation*, precisamos treiná-la com um número de tuplas mais significativo. Este fato pode explicar o porquê de os resultados da aplicação do plano de agrupamento seguido de seleção e imputação com redes *back propagation*, apesar de também se mostrarem bastante interessantes, serem inferiores aos da mesma estratégia com a imputação com o algoritmo dos *k* vizinhos mais próximos.

Os resultados envolvendo a seleção precedendo o agrupamento e imputação com *back propagation* mostram que, em qualquer uma das bases, o treinamento da rede se torna prejudicado pela falta de informação causada pela seleção. Os resultados são irregulares, principalmente na base *Pima Indians Diabetes*, onde a rede não consegue estabelecer uma relação entre as colunas, já que ela não é representativa.

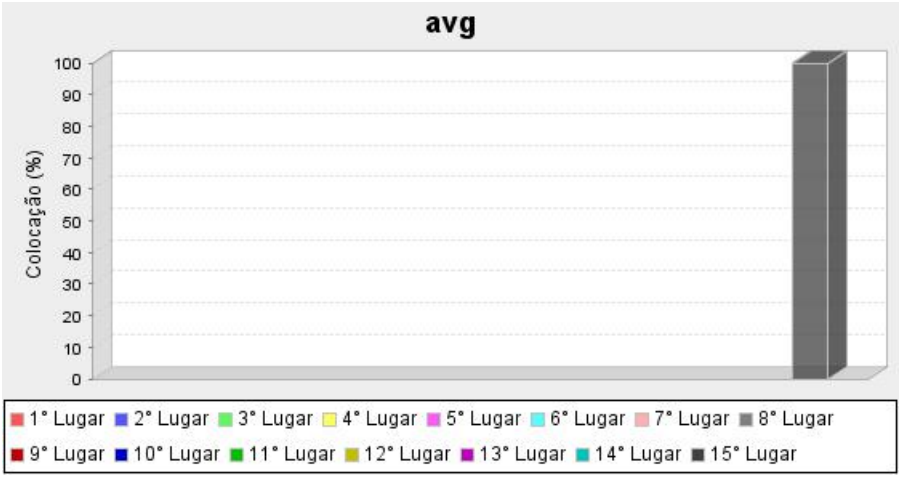
Por fim, os comitês de complementação de dados ausentes não apresentaram os resultados inicialmente imaginados, tendo comportamento irregular em todos os testes aos quais foi submetido. Apesar disso, este fato pode ser facilmente explicado pela influência que planos com desempenho ruim tem na geração da sugestão do comitê. Entretanto, seu uso não deve ser desconsiderado, pois em todas as bases os resultados gerados pelo comitê aparecem em algumas vezes nas primeiras colocações, o que indica que outros estudos sobre esta estrutura podem ser feitos, e melhores resultados podem ser alcançados.

De uma maneira geral, podemos dizer que todos os experimentos realizados nesta tese indicam que a seleção e o agrupamento de dados são tarefas importantes para a qualidade do processo de imputação quando combinadas. As opções envolvendo somente o agrupamento de dados também são boas, e, dependendo da disponibilidade de recursos e de tempo, além da característica das bases, também podem ser utilizadas.

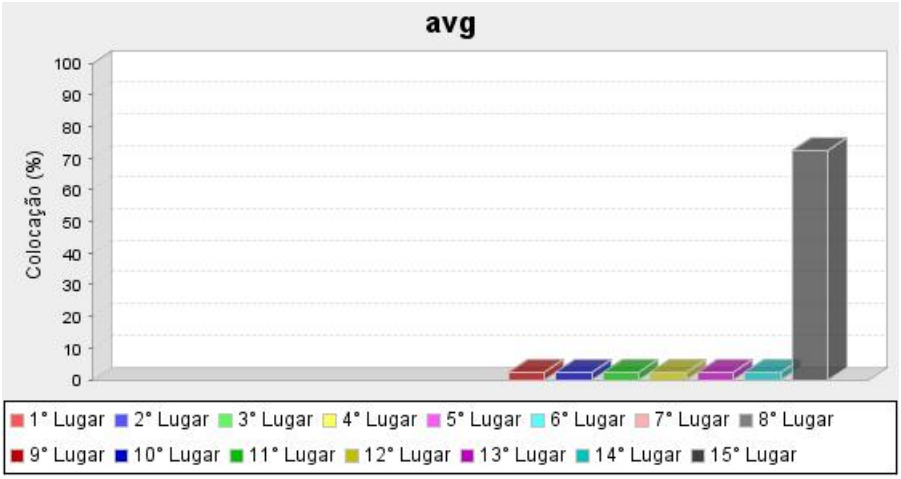
5.2.3.2 Gráficos

Plano 1: Imputação com Média

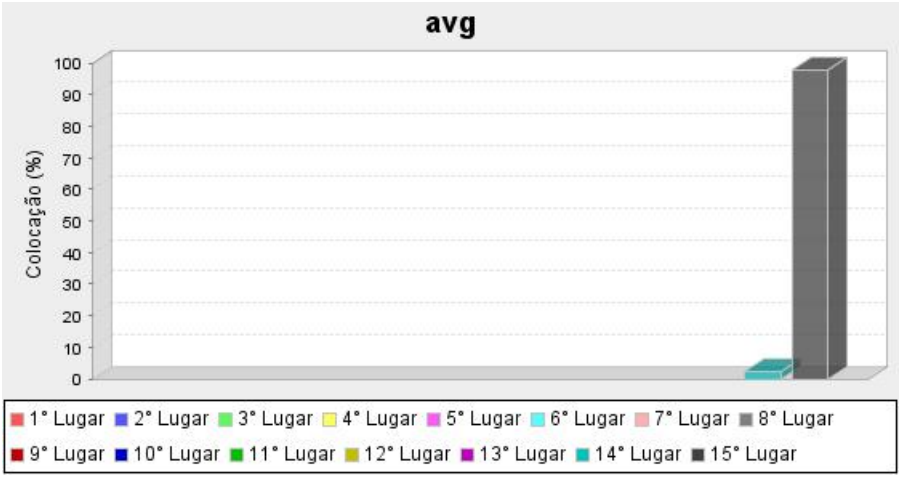
Iris Plants



Pima Indians Diabetes

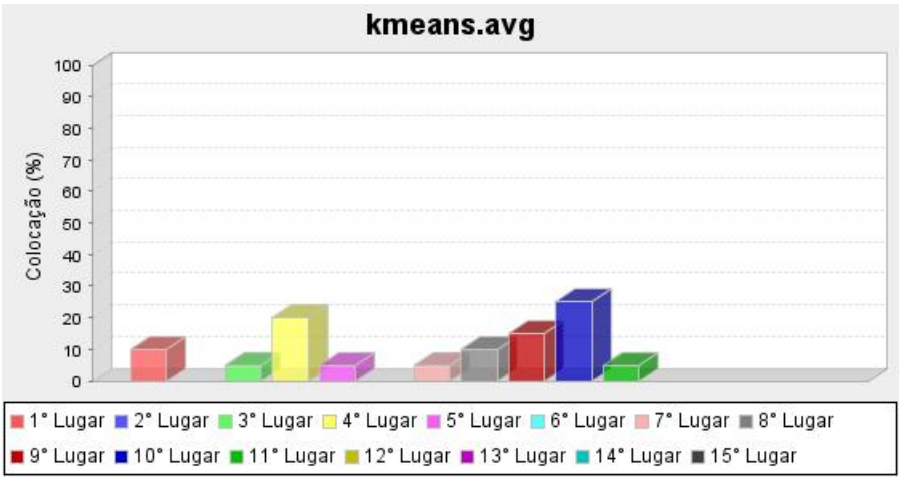


Wisconsin Breast Cancer

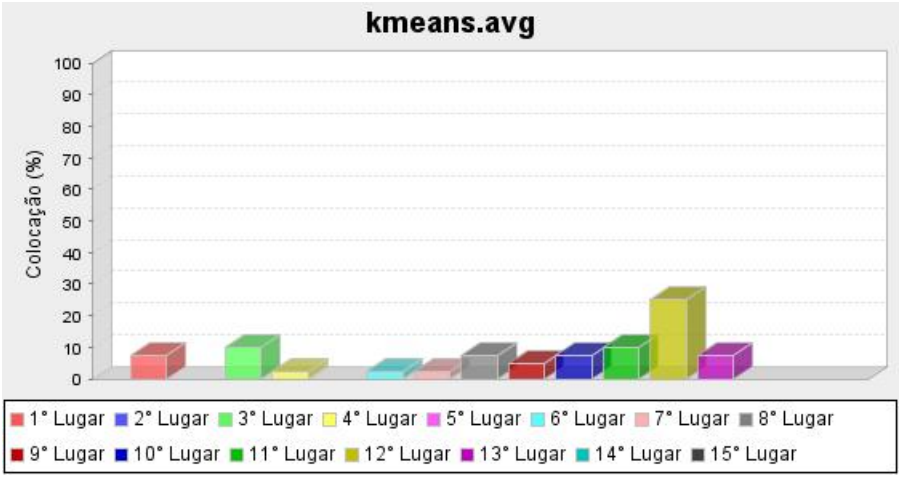


Plano 2: Agrupamento com K-Means e Imputação com Média

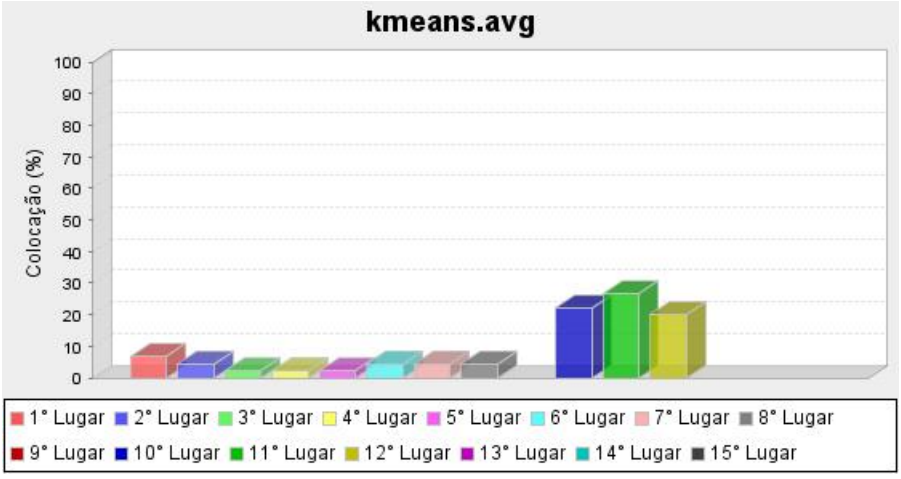
Iris Plants



Pima Indians Diabetes

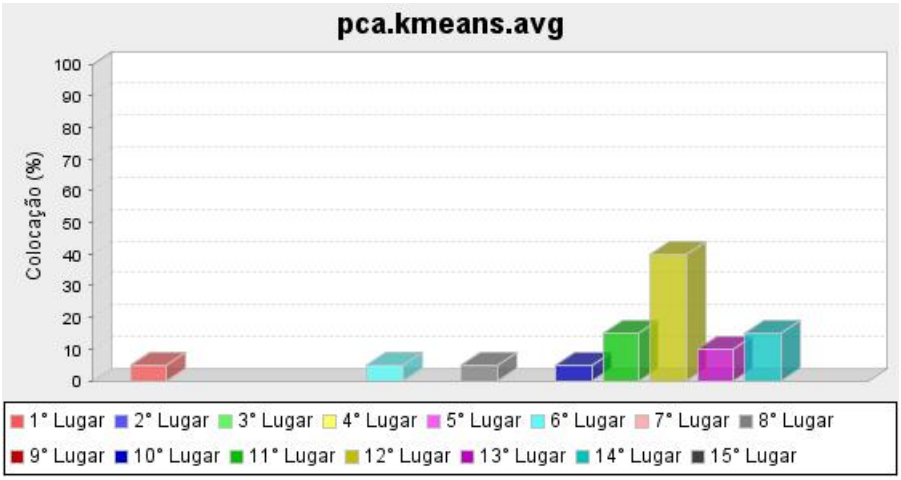


Wisconsin Breast Cancer

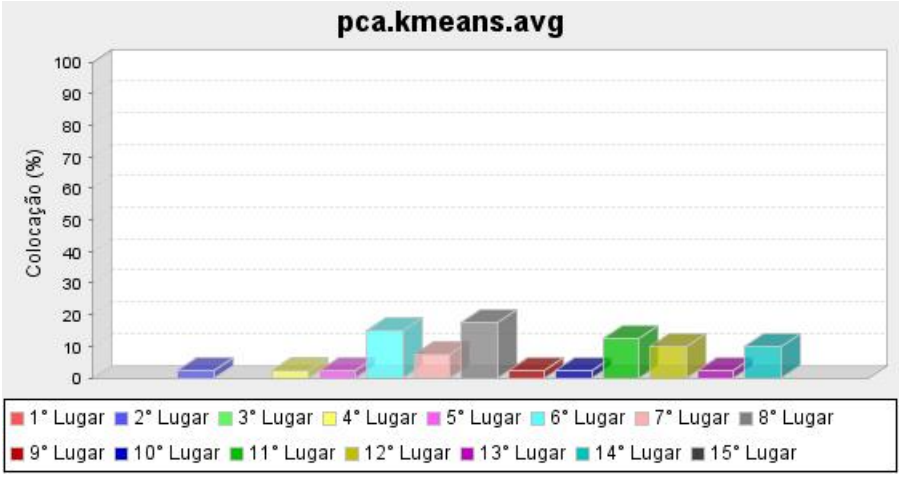


Plano 3: Seleção com PCA, Agrupamento com K-Means e Imputação com Média

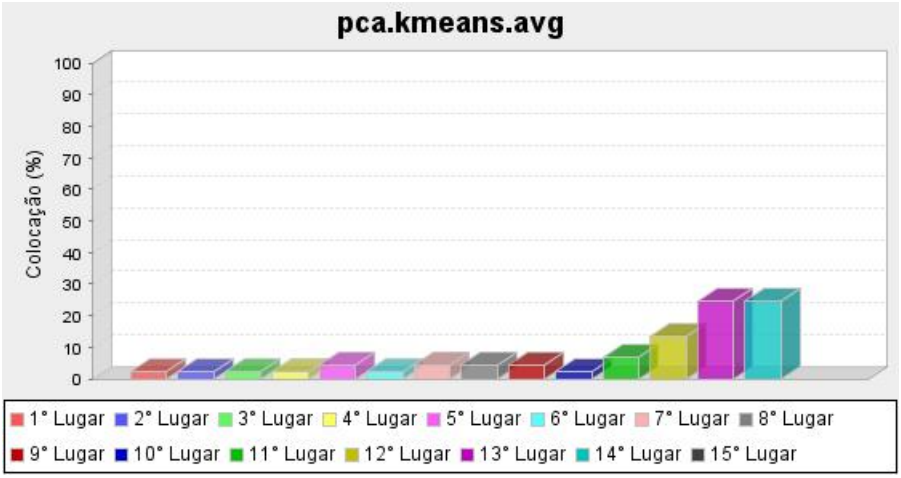
Iris Plants



Pima Indians Diabetes

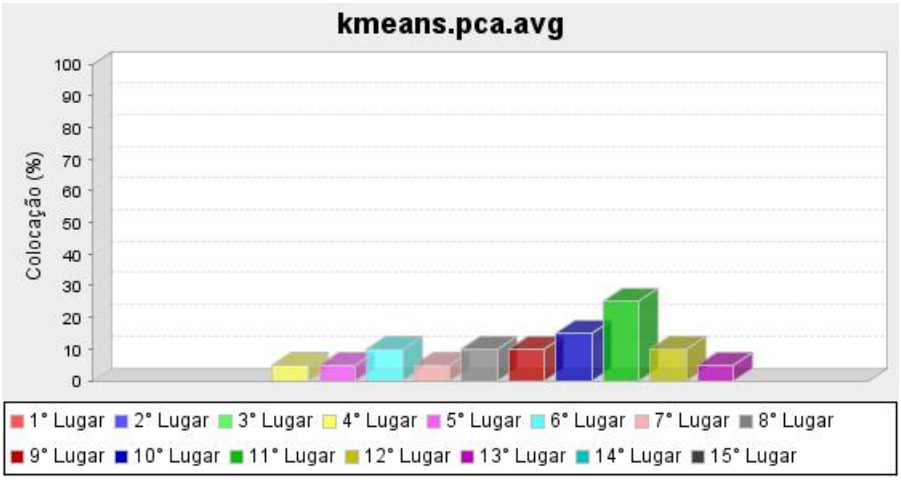


Wisconsin Breast Cancer

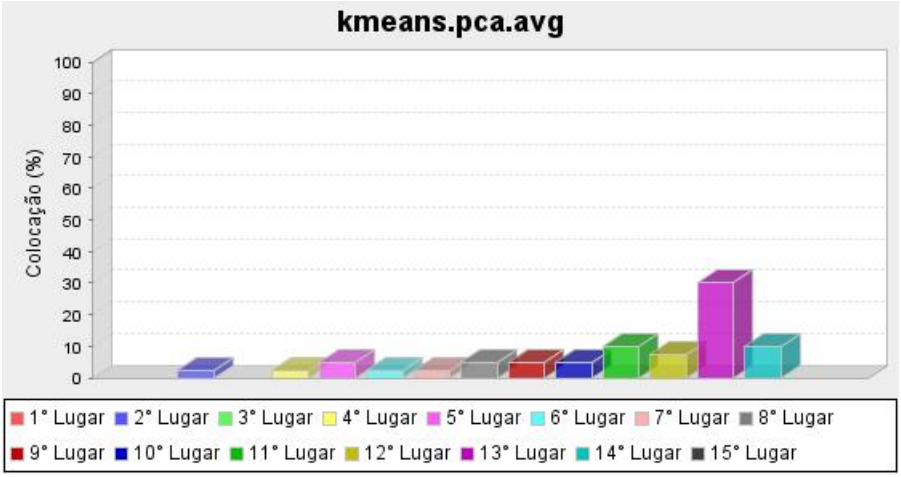


Plano 4: Agrupamento com K-Means, Seleção com PCA e Imputação com Média

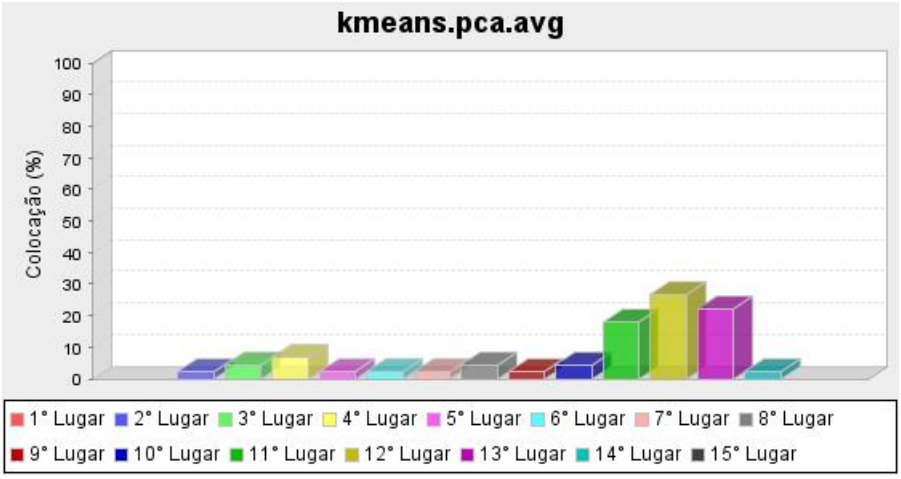
Iris Plants



Pima Indians Diabetes

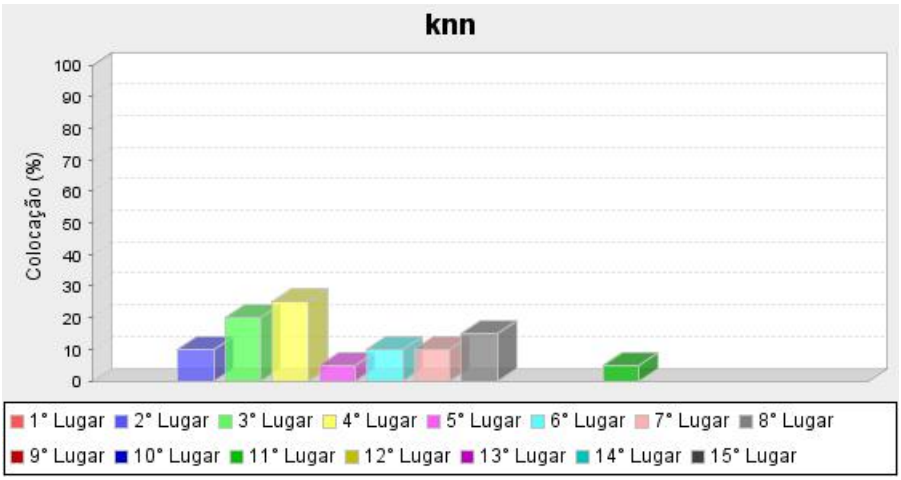


Wisconsin Breast Cancer

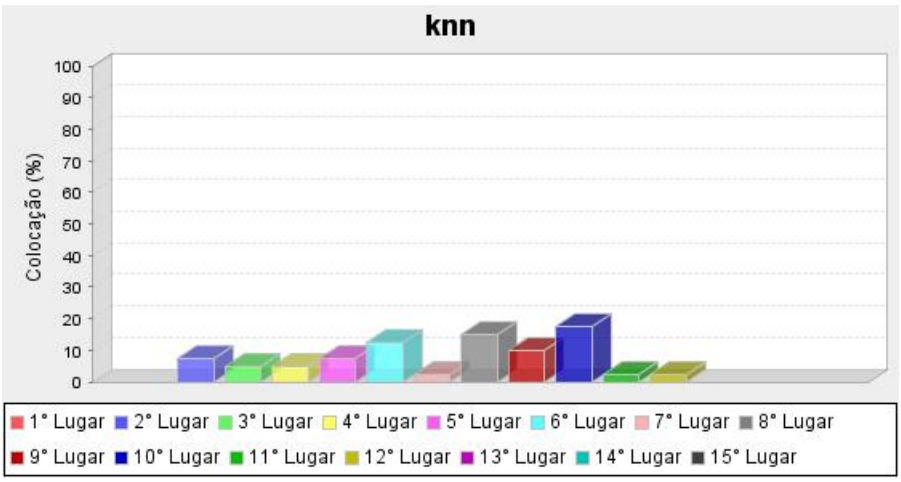


Plano 5: Imputação com *k*-NN

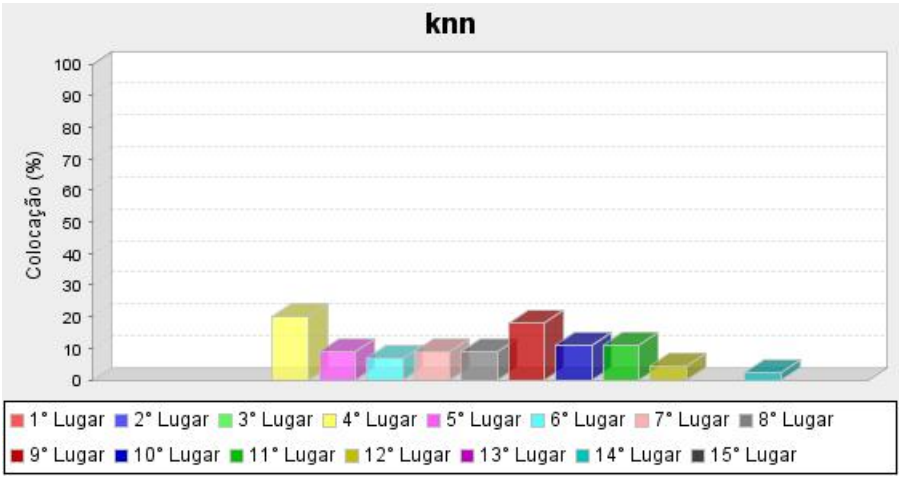
Iris Plants



Pima Indians Diabetes

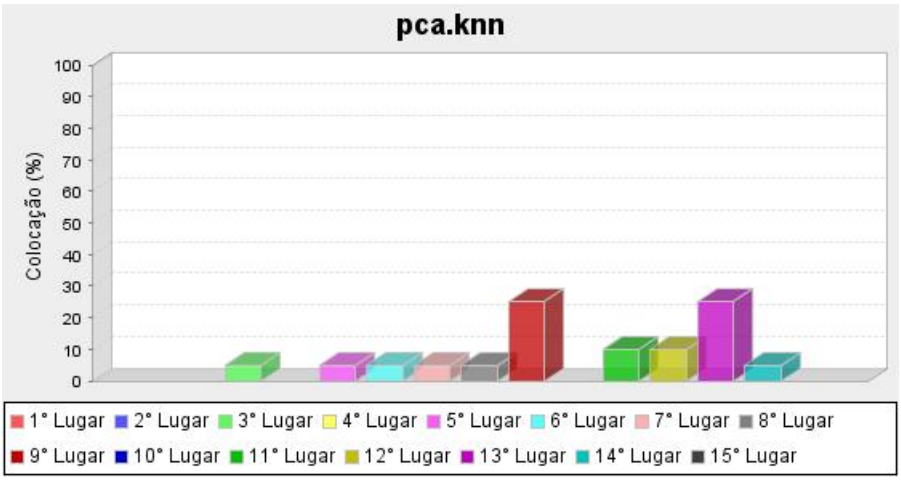


Wisconsin Breast Cancer

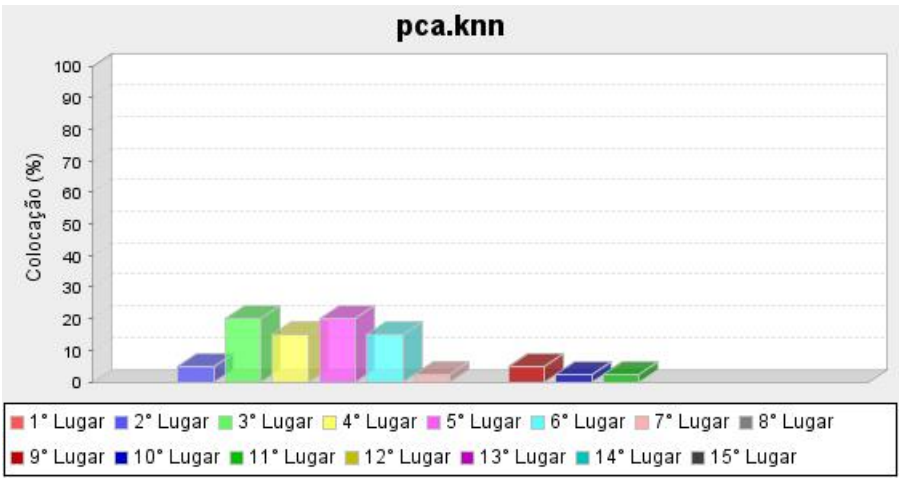


Plano 6: Seleção com PCA e Imputação com *k*-NN

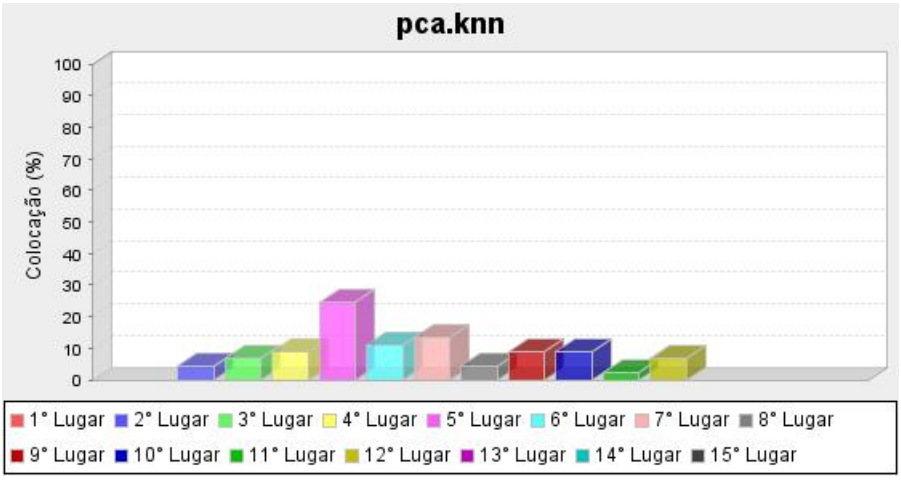
Iris Plants



Pima Indians Diabetes

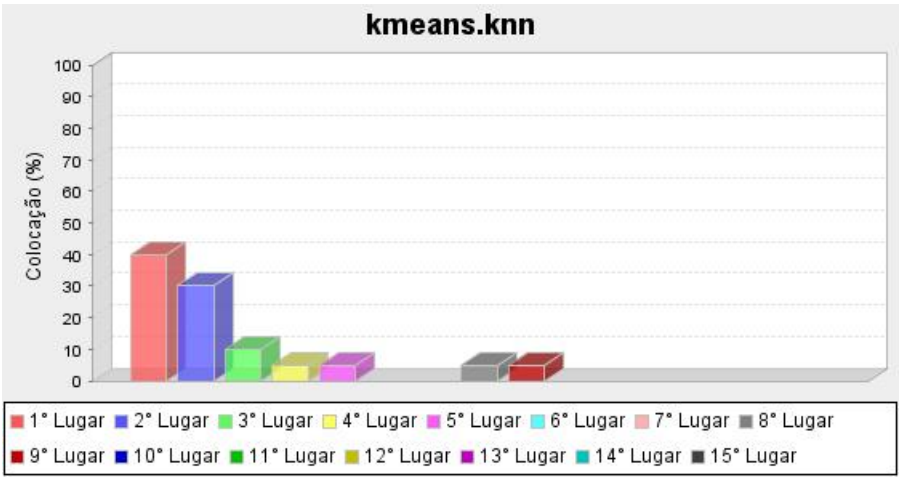


Wisconsin Breast Cancer

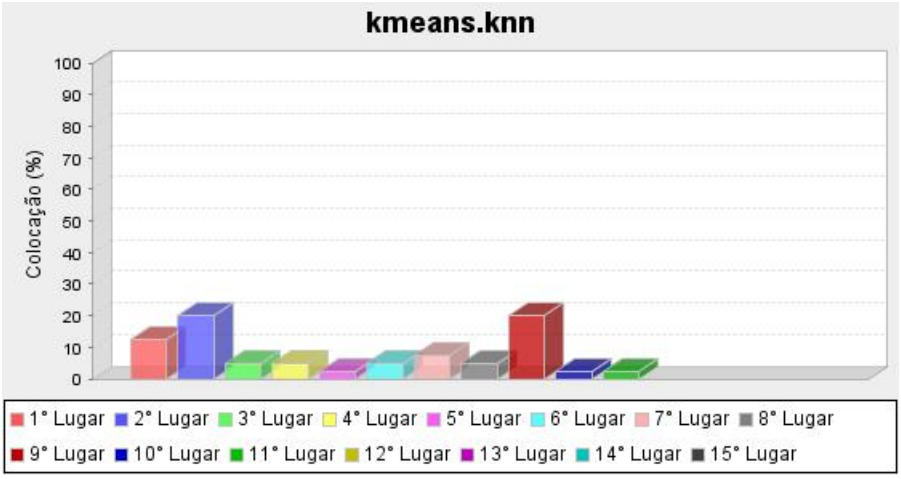


Plano 7: Agrupamento com *K*-Means e Imputação com *k*-NN

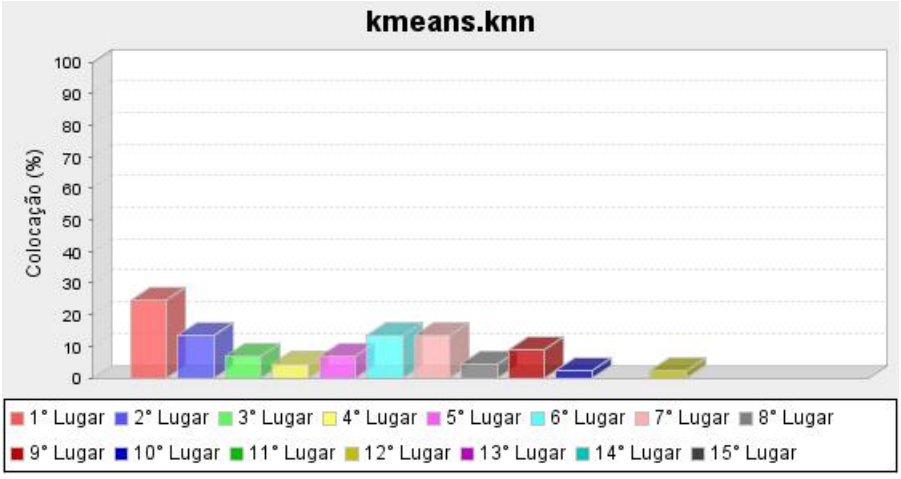
Iris Plants



Pima Indians Diabetes

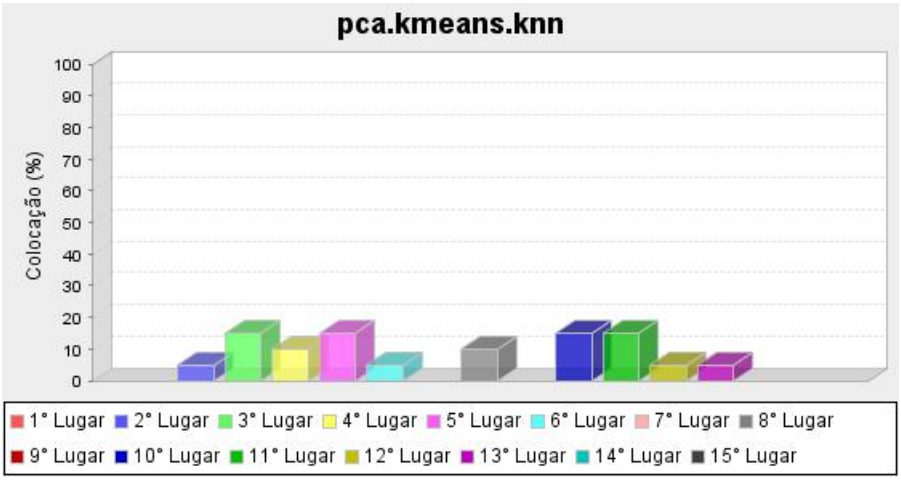


Wisconsin Breast Cancer

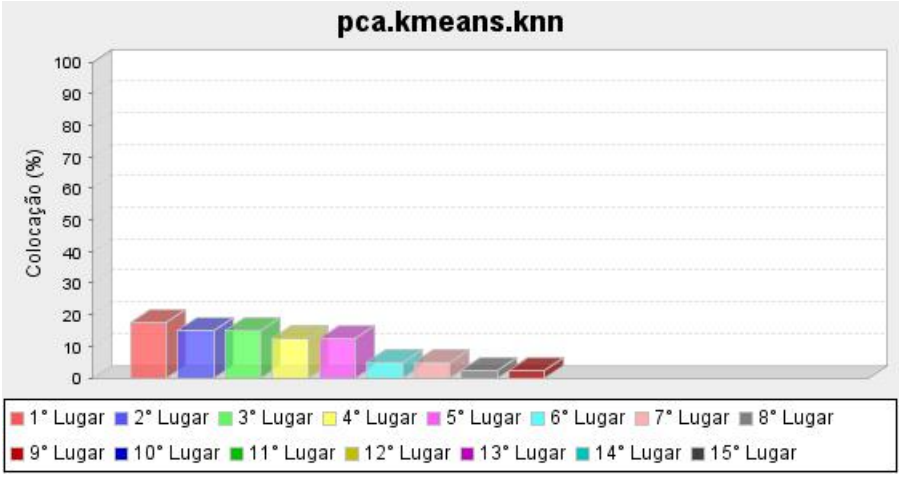


Plano 8: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *k*-NN

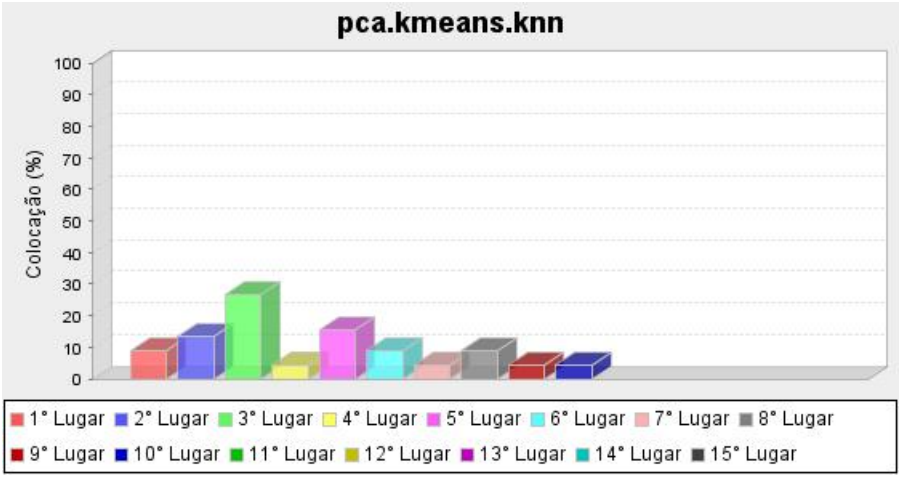
Iris Plants



Pima Indians Diabetes

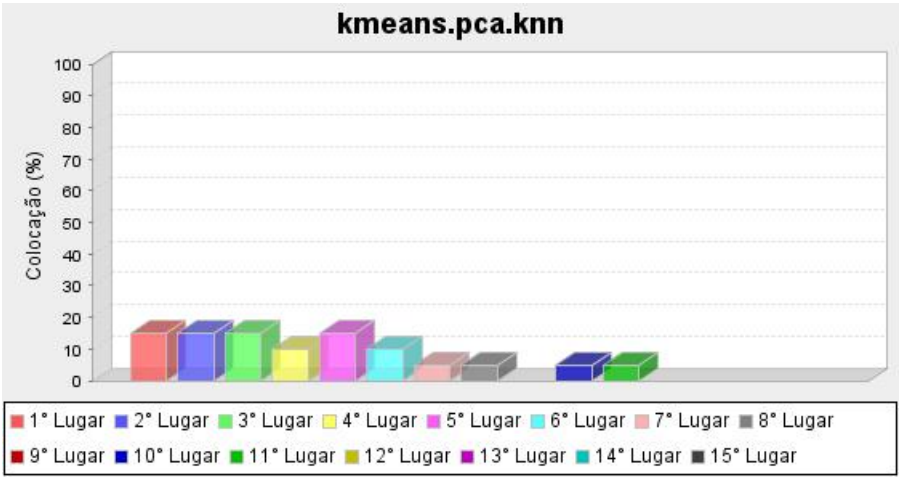


Wisconsin Breast Cancer

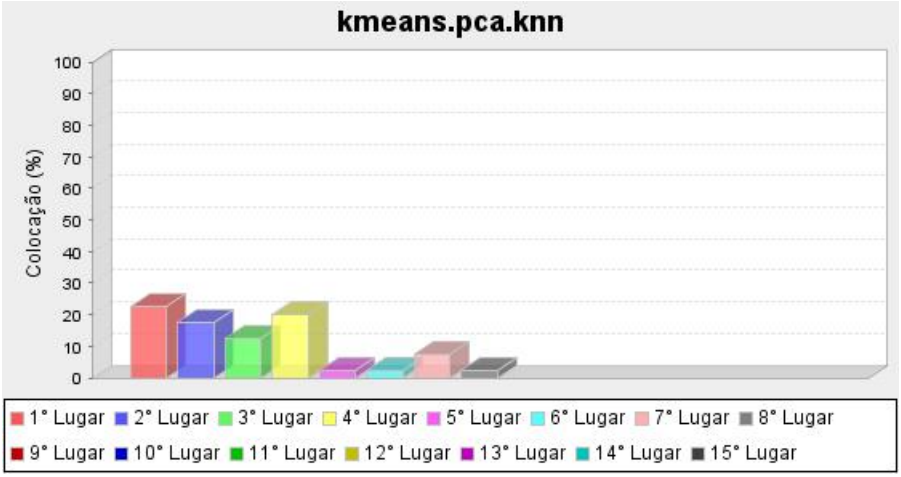


Plano 9: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *k*-NN

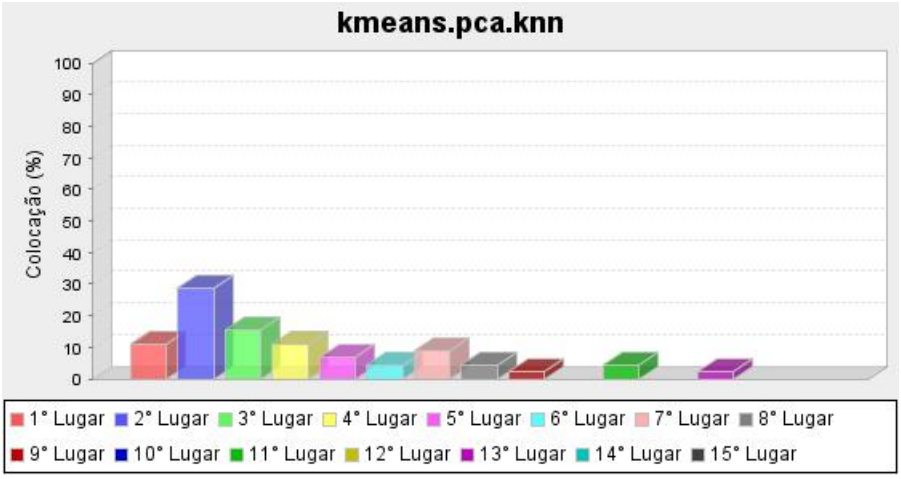
Iris Plants



Pima Indians Diabetes

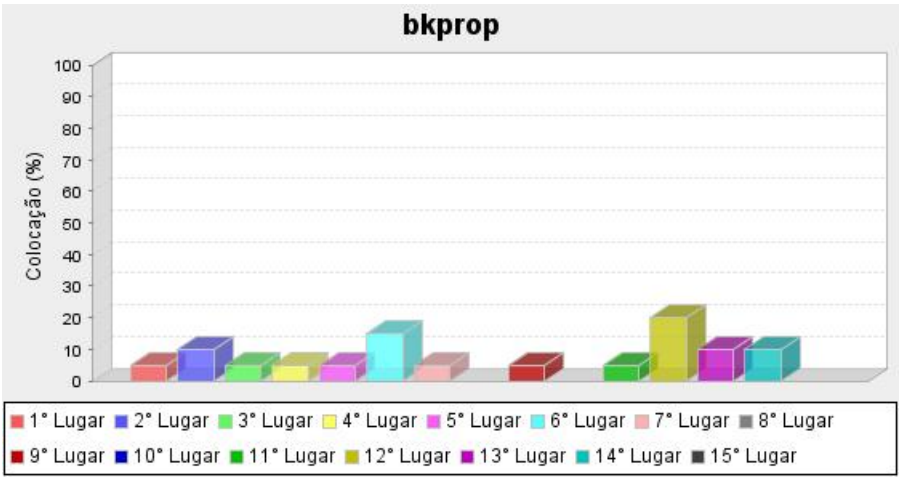


Wisconsin Breast Cancer

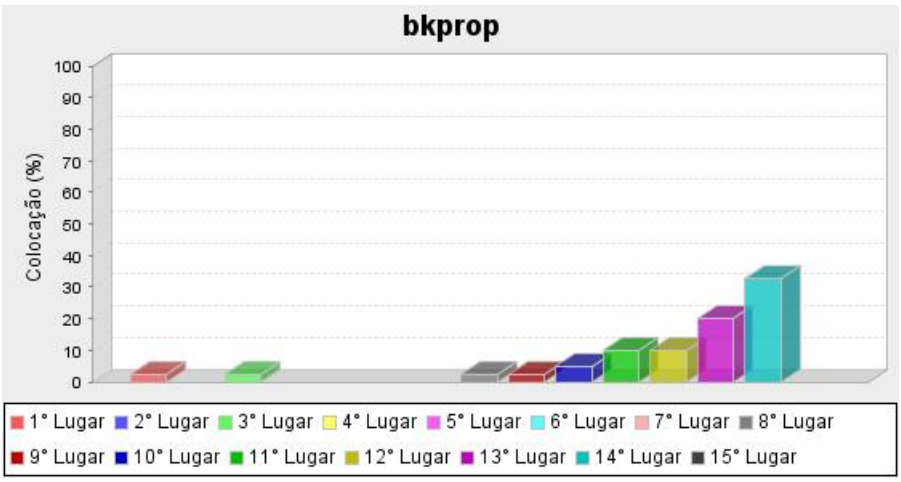


Plano 10: Imputação com *back propagation*

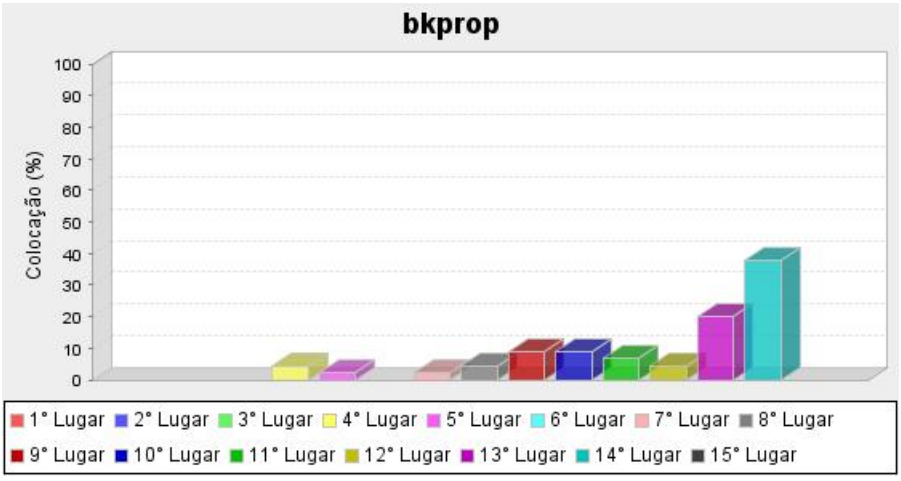
Iris Plants



Pima Indians Diabetes

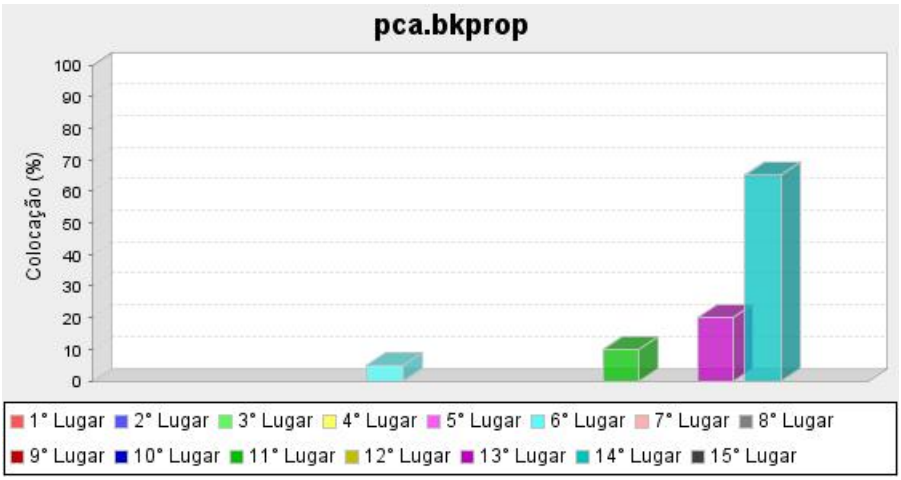


Wisconsin Breast Cancer

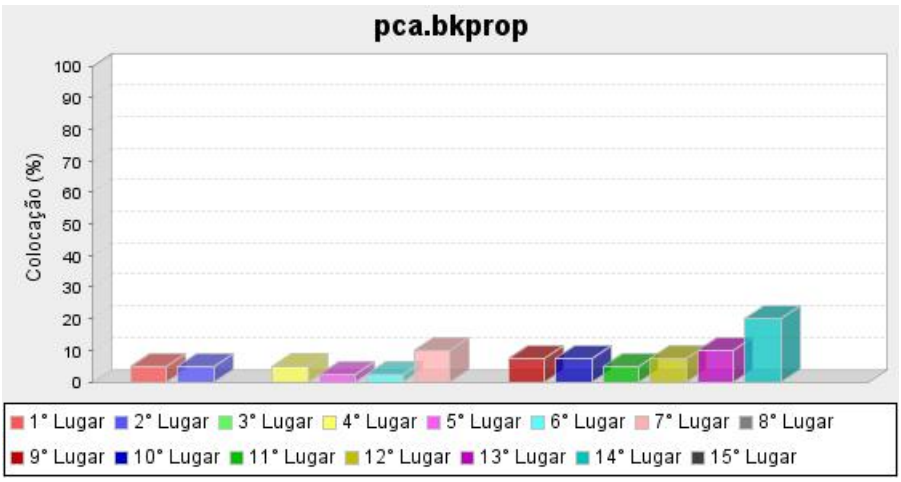


Plano 11: Seleção com PCA e Imputação com *back propagation*

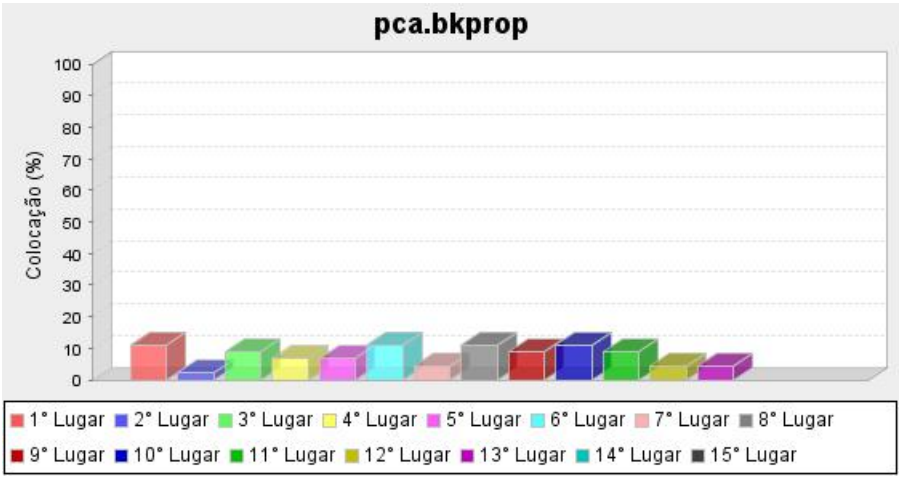
Iris Plants



Pima Indians Diabetes

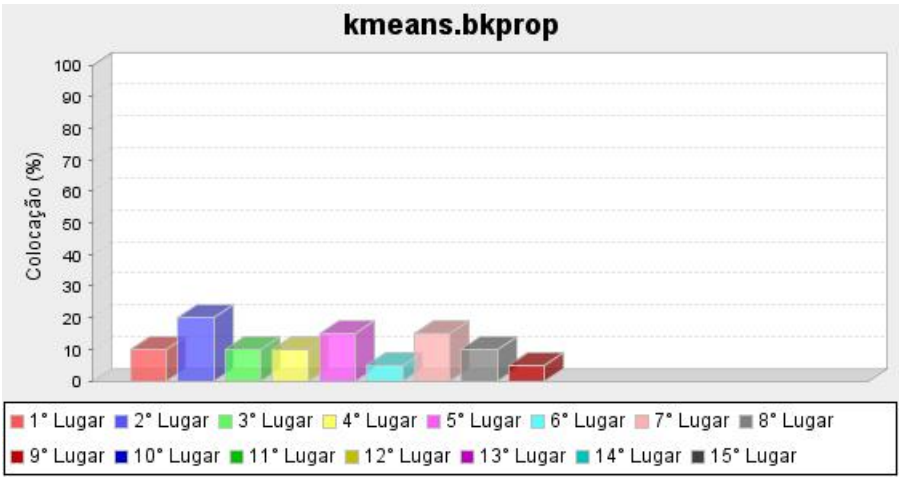


Wisconsin Breast Cancer

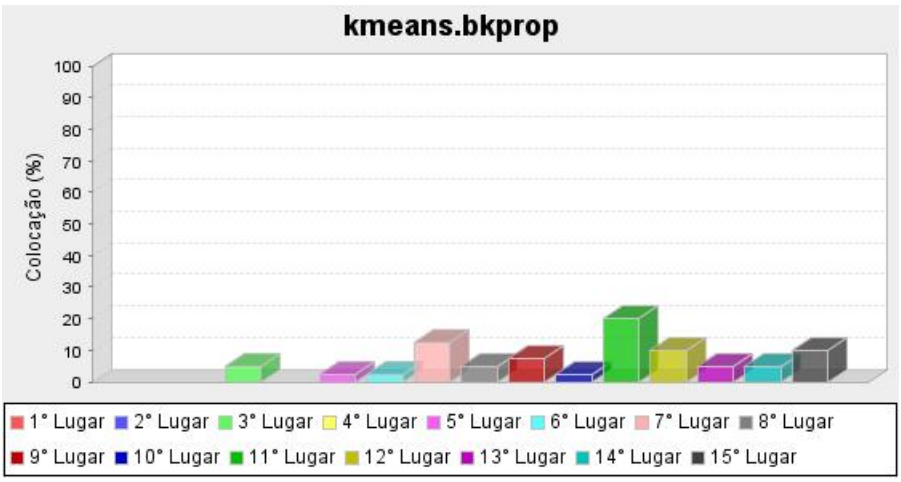


Plano 12: Agrupamento com *K*-Means e Imputação com *back propagation*

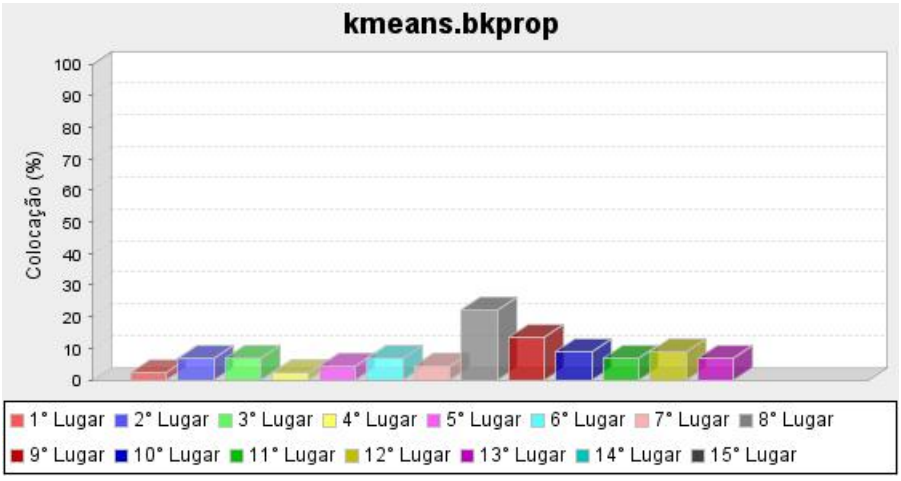
Iris Plants



Pima Indians Diabetes

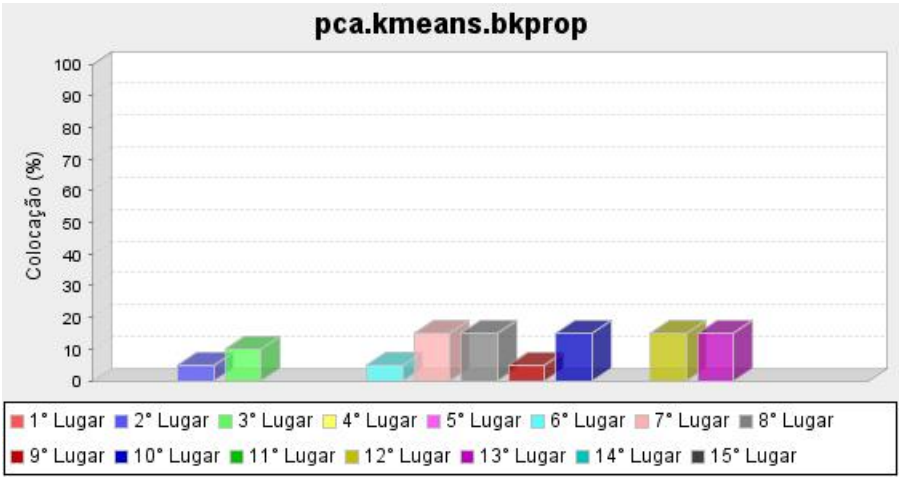


Wisconsin Breast Cancer

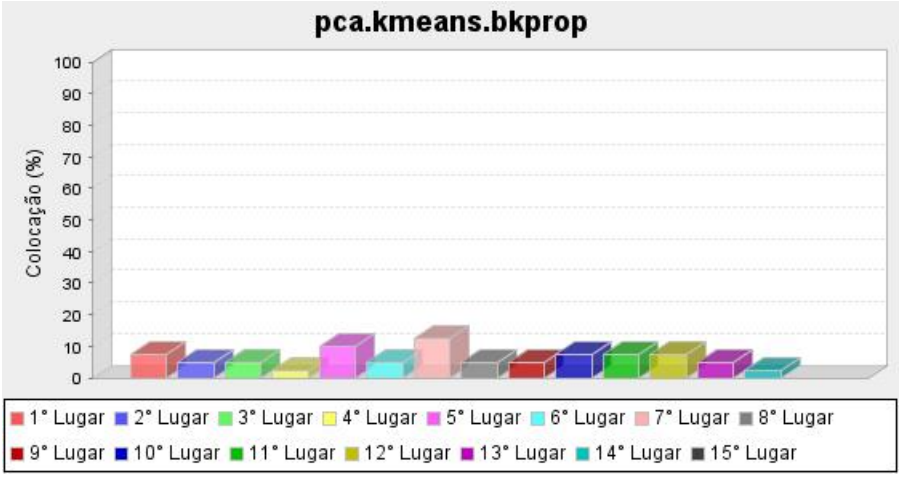


Plano 13: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *back propagation*

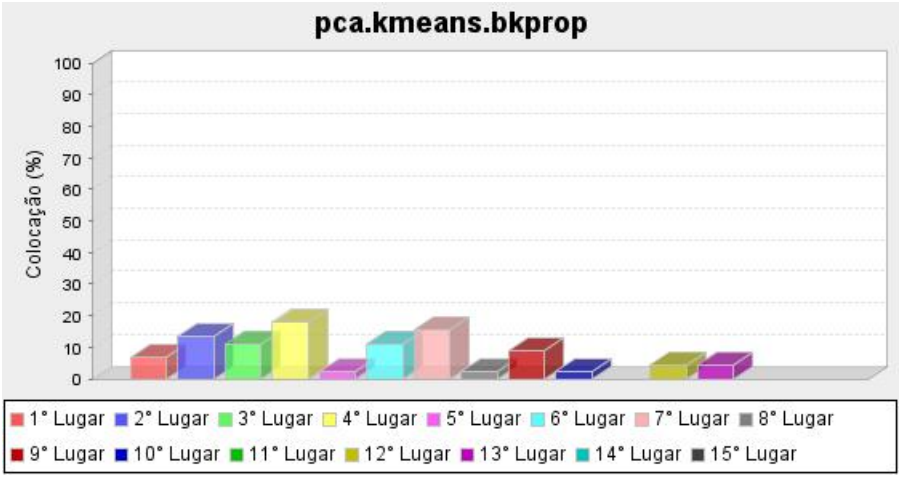
Iris Plants



Pima Indians Diabetes

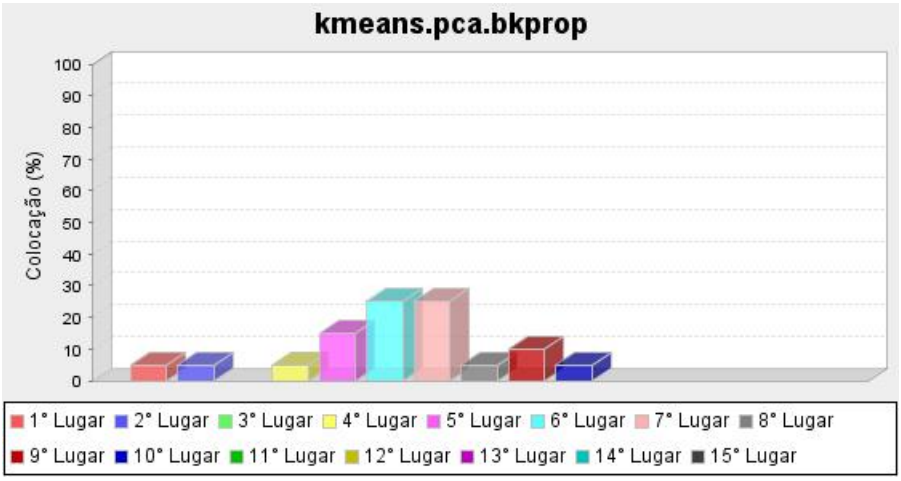


Wisconsin Breast Cancer

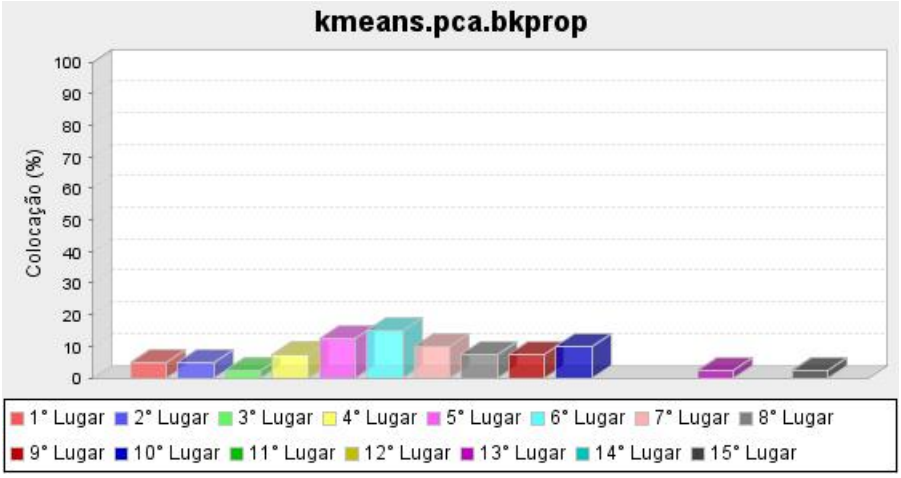


Plano 14: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *back propagation*

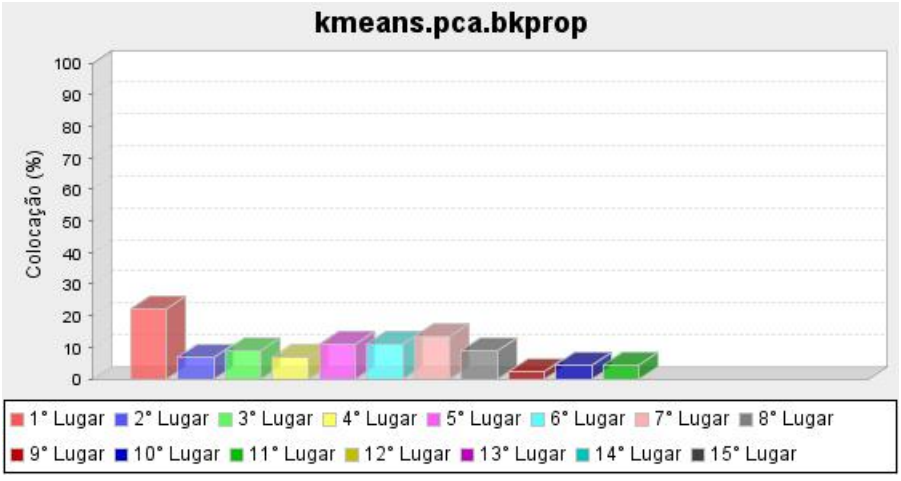
Iris Plants



Pima Indians Diabetes

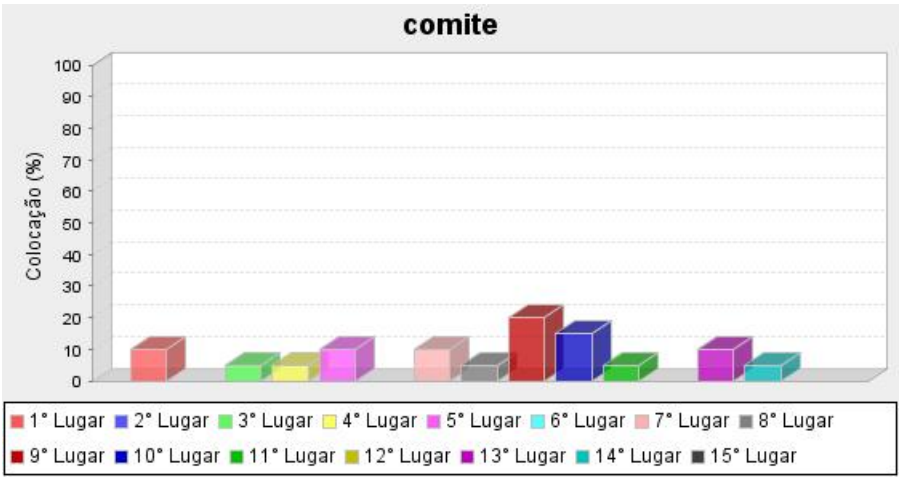


Wisconsin Breast Cancer

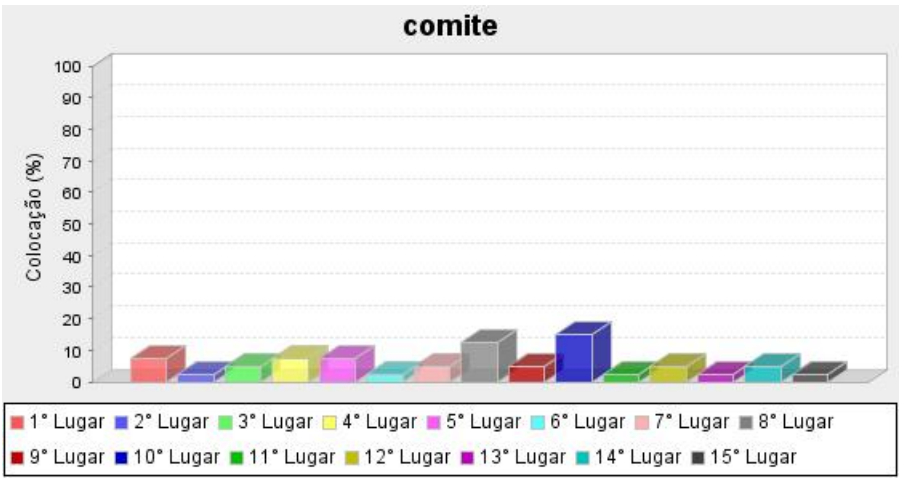


Plano 15: Comitês de Complementação de Dados

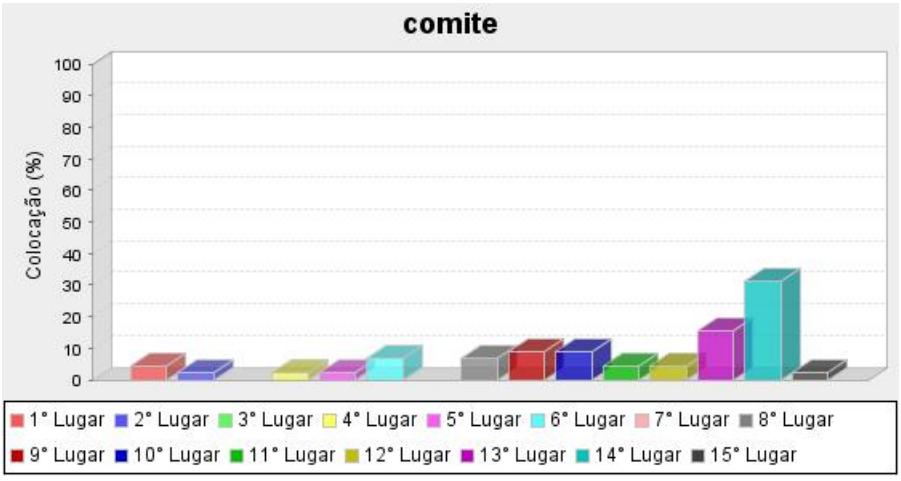
Iris Plants



Pima Indians Diabetes



Wisconsin Breast Cancer



5.2.4 Erros médios de classificação por percentuais de valores ausentes

5.2.4.1 Análise dos resultados

Avaliamos, para cada percentual de valores ausentes em uma base de dados, qual o erro de classificação de todas as tuplas de uma base dado um percentual de valores ausentes. Desejamos, com esta análise, verificar se o aumento do número de tuplas que apresentam valores desconhecidos em um de seus atributos impacta no processo de classificação da base de dados, já que, à medida que este índice cresce, o conjunto de treinamento enviado ao classificador diminui de proporção. Assim, nossa intenção é observar, com os dados dos experimentos, se esta redução no número de amostras utilizadas pelo classificador faz com que a qualidade do processo de classificação seja continuamente afetado, tanto na categorização de tuplas com atributos imputados quanto nas tuplas originais.

Contraditoriamente, o que pudemos observar é que, em todos os resultados de nossos experimentos, o percentual de valores ausentes não impactou a qualidade da classificação. Os índices médios de erros na classificação de tuplas por bases de dados e por taxa de valores ausentes revelaram-se aproximadamente constantes, tanto para tuplas com valores imputados quanto para as com valores originais. Em todos os planos de imputação, e em todos os percentuais de valores ausentes na base *Pima Indians Diabetes* os erros médios de classificação das tuplas imputadas e originais giraram aproximadamente entre 23% e 27,5%, índices altos de erro para os dois casos. Atribuímos este acontecimento à característica da base, que, como pudemos ver anteriormente, possui baixa correlação entre os atributos. Como a rede neuronal tenta capturar a relação intrínseca entre os dados, e como essa relação não é tão presente, os erros de classificação passam a acontecer em índices muito elevados. Chama-nos a atenção este erro médio alto em todas as situações de classificação das tuplas originais. Nas bases que possuem valores imputados, o fato novamente ocorre, independentemente do plano de imputação utilizado.

Os índices na base *Wisconsin Breast Cancer* revelam um comportamento semelhante, diferenciando-se no índice médio de erro de classificação, que variou entre 3,5% e 5,7%. Entretanto, verificou-se o mesmo fenômeno ocorrido na base anterior: os erros médios de classificação das bases com tuplas imputadas e originais

não variou por plano de imputação, e os valores comparativos destas duas situações foi bastante próximo.

Os dois casos apresentados mostram que a imputação não alterou significativamente o processo de classificação dos dados. Isto parece indicar que o processo de complementação de dados, de uma forma geral, não distorce as características intrínsecas dos dados, já que a diferença entre o erro médio de classificação das tuplas com valores imputados e originais sempre girou entre 1% e 3%, um índice de erro médio bastante baixo. Em algumas vezes, nas duas bases, o erro de classificação das tuplas imputadas apresentou menor erro médio do que com as tuplas originais.

Já a base *Iris Plants* revelou um comportamento um pouco diferente das outras duas na variação do percentual de valores ausentes. Na classificação das tuplas imputadas com a média, ficou evidente, com os testes realizados, que este plano de imputação realmente tornou os dados bastante tendenciosos. Isto aconteceu pelas características da base de dados: quatro atributos, três altamente correlacionados, e valores cujo domínio tem um espectro de variação bastante baixo. A taxa média de erro da imputação com média, comparada aos índices obtidos com a classificação das tuplas originais, mostra que este plano não se apresenta como uma boa opção de imputação.

Outro comportamento bastante interessante na base *Iris Plants* foi observado com a variação do percentual de valores ausentes e o plano de imputação adotado, com exceção da média. Em todos os casos, os índices de erro de classificação com 10% de valores ausentes na base de dados são maiores, na faixa de 10% e 15%, do que com os demais percentuais (de 20% a 50%), que girou entre 3% e 8%. Para tentar entender este acontecimento, analisamos a distribuição de valores nulos nos atributos da base *Iris Plants*, entre as três possíveis classes. Em cada base, temos um total de 15 tuplas com valores ausentes.

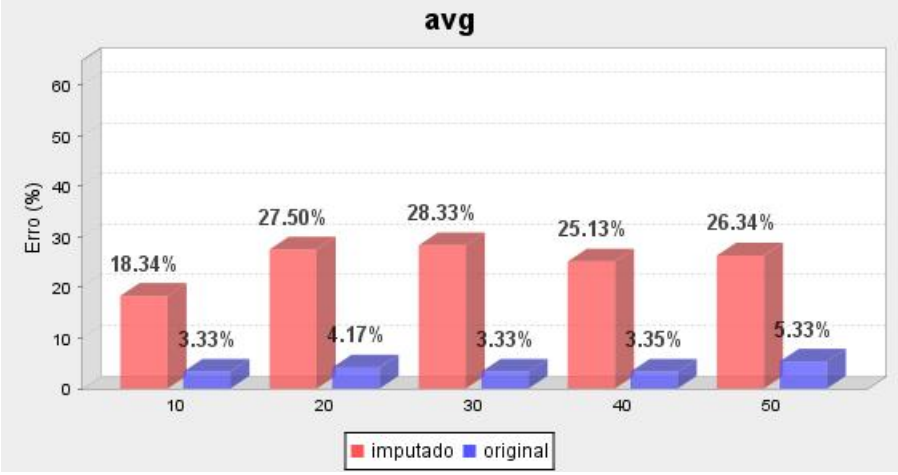
	Setosa	*Versicolor*	*Virgínica*
petallength	3 (**20%**)	8 (**53%**)	4 (**27%**)
petalwidth	4 (**27%**)	6 (**40%**)	5 (**34%**)
sepallength	7 (**47%**)	3 (**20%**)	5 (**34%**)
sepalwidth	2 (**14%**)	6 (**40%**)	7 (**47%**)

A situação ideal de distribuição de dados ausentes seria aquela onde cada classe apresentasse um percentual de 33,33% de tuplas nesta situação. Todavia, todo o nosso estudo é feito com dados que apresentam mecanismo de ausência de dados completamente aleatório (MCAR). Assim, a geração de valores nulos não poderia (e nem deveria) ser controlada. Por esta razão, três das quatro bases revelam um índice muito alto de valores ausentes em uma das três classes. Este acontecimento pode ter causado esta média de erro alta e constante de classificação, independente do plano de imputação utilizado (à exceção da média).

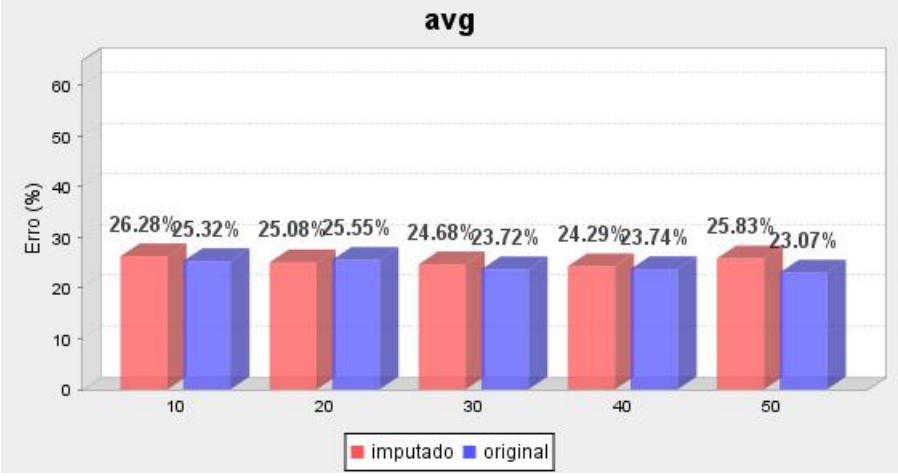
5.2.4.2 Gráficos

Plano 1: Imputação com Média

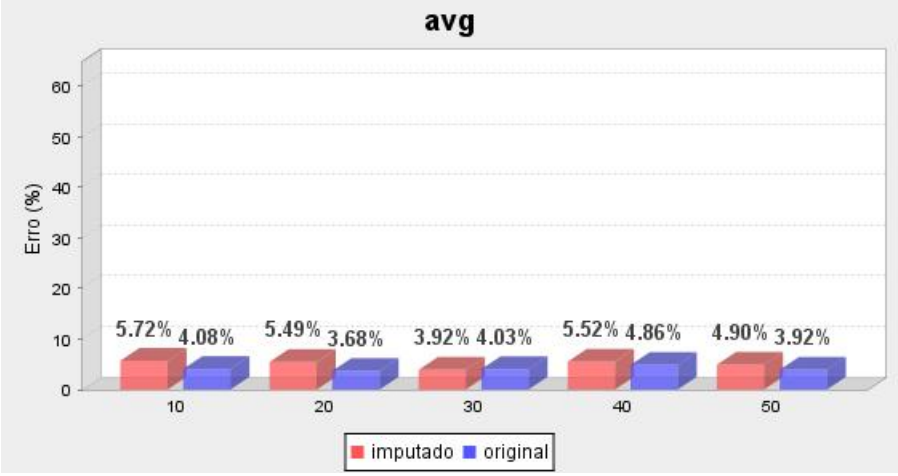
Iris Plants



Pima Indians Diabetes

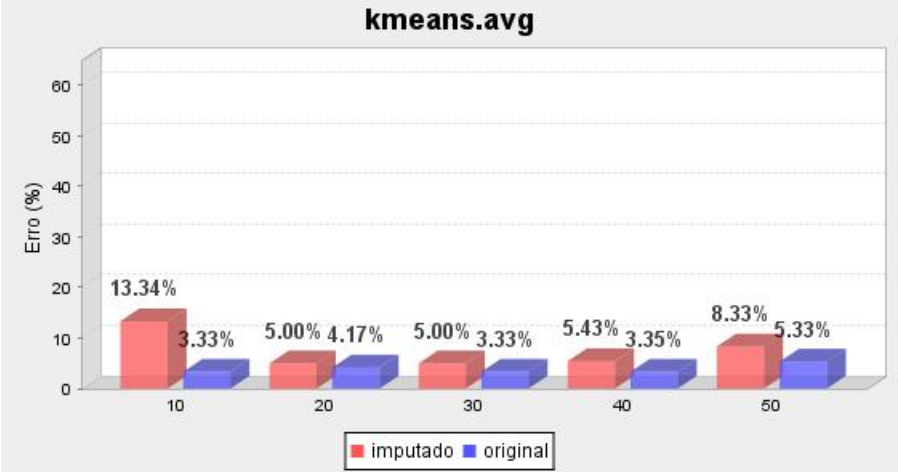


Wisconsin Breast Cancer

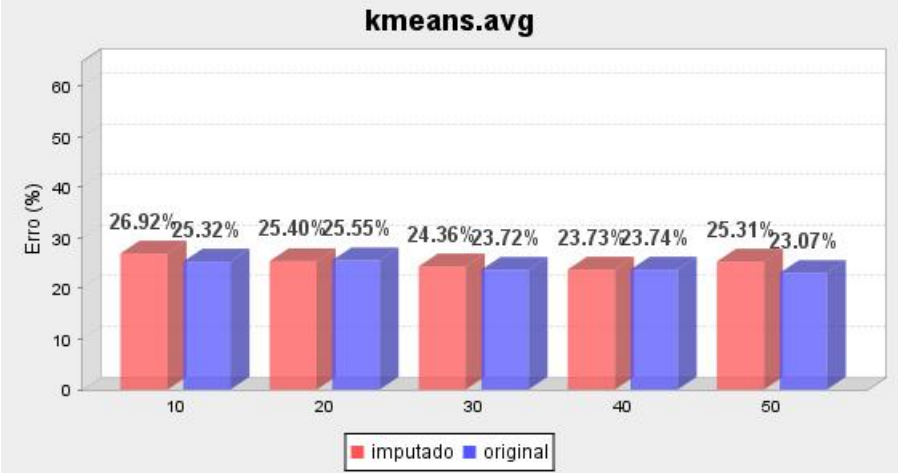


Plano 2: Agrupamento com K-Means e Imputação com Média

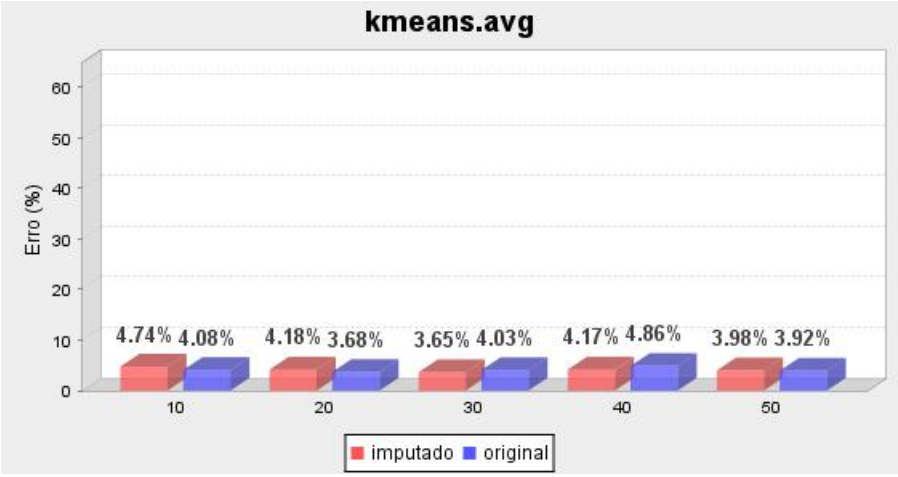
Iris Plants



Pima Indians Diabetes

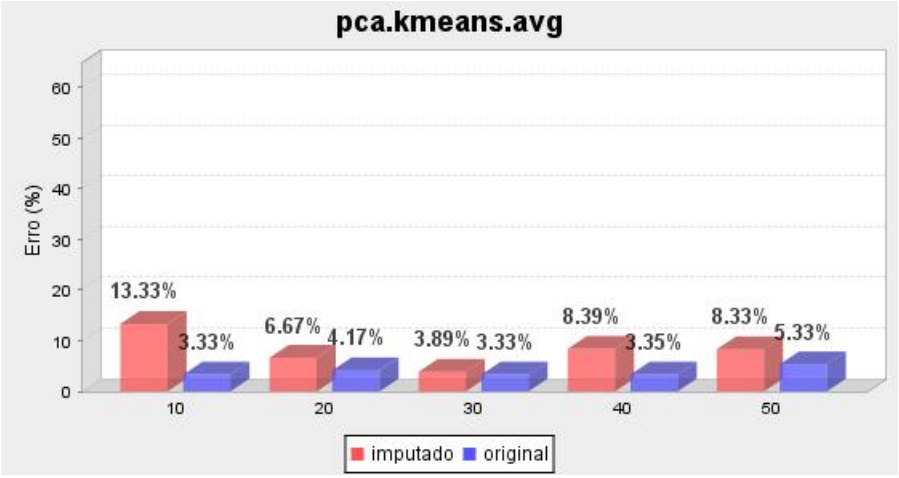


Wisconsin Breast Cancer

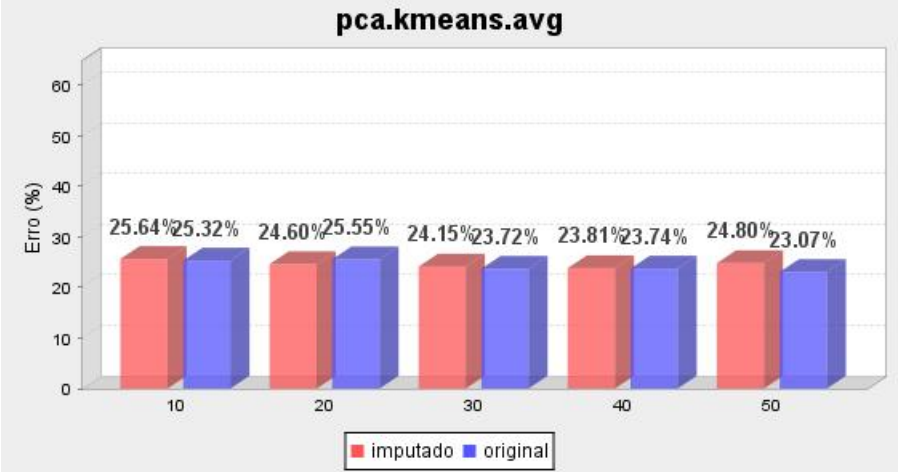


Plano 3: Seleção com PCA, Agrupamento com K-Means e Imputação com Média

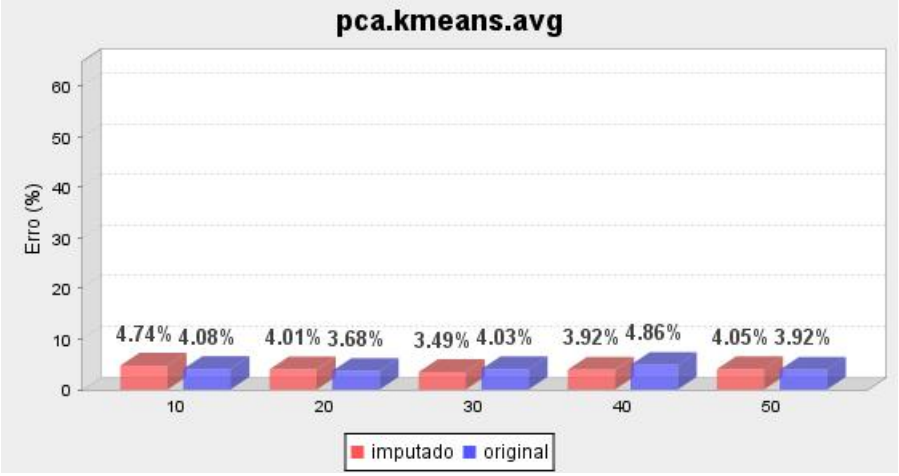
Iris Plants



Pima Indians Diabetes

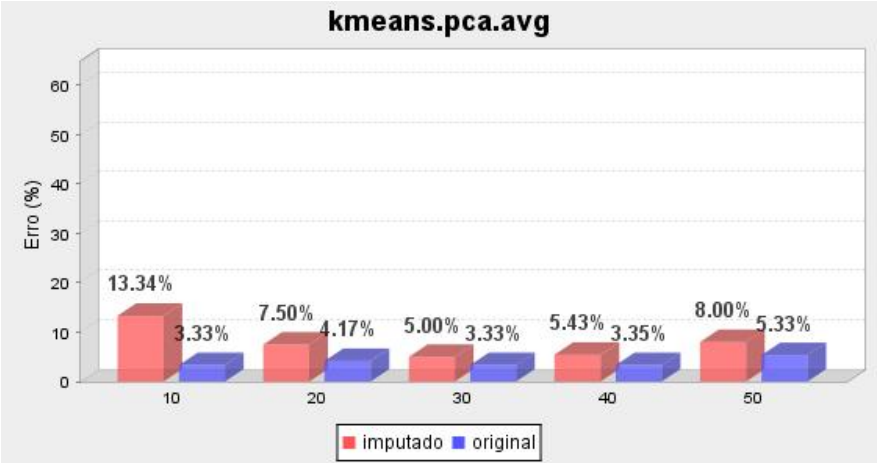


Wisconsin Breast Cancer

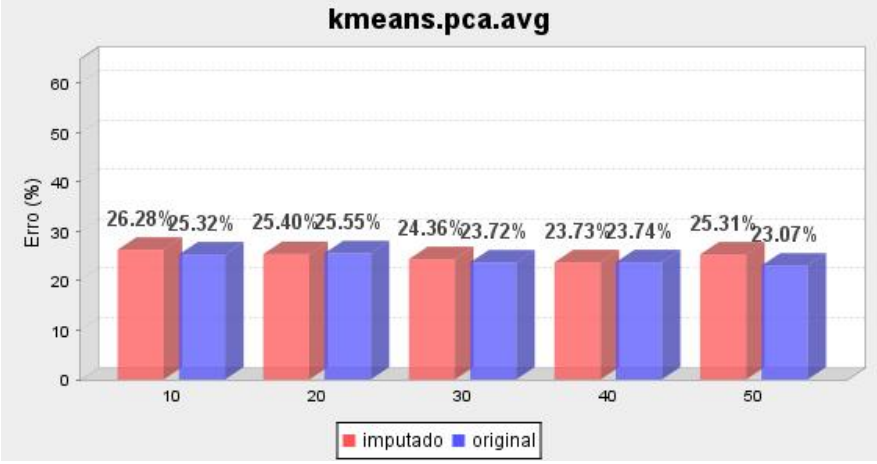


Plano 4: Agrupamento com K-Means, Seleção com PCA e Imputação com Média

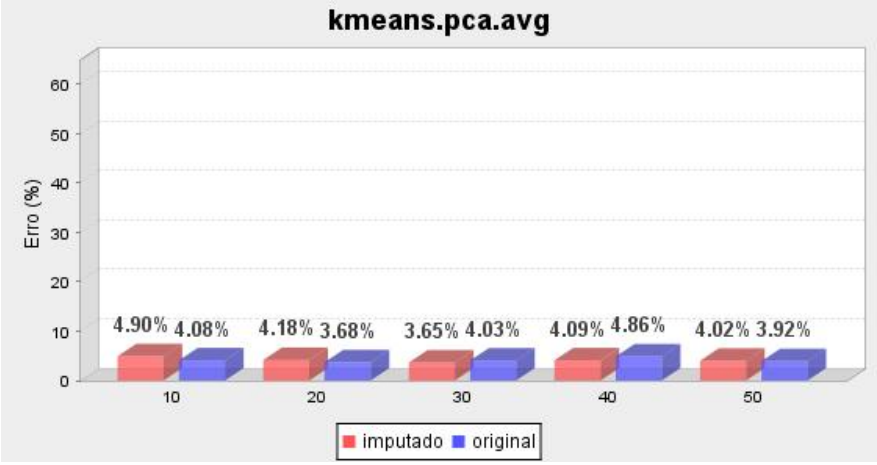
Iris Plants



Pima Indians Diabetes

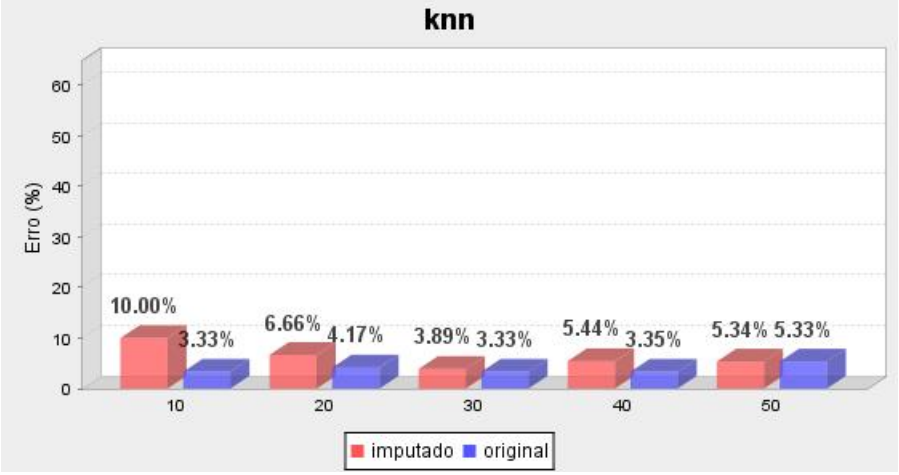


Wisconsin Breast Cancer

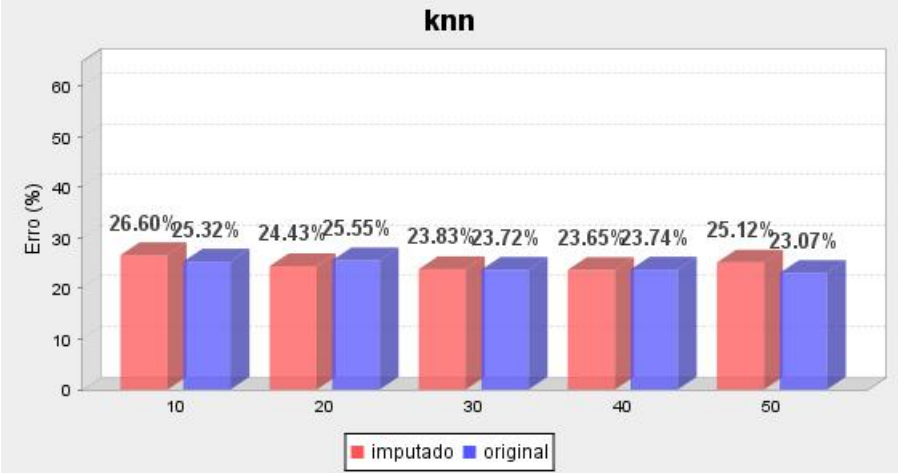


Plano 5: Imputação com *k*-NN

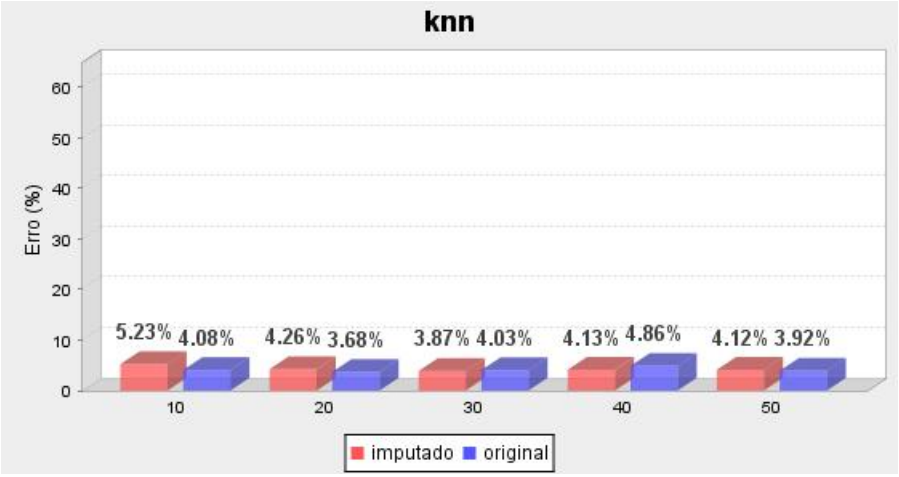
Iris Plants



Pima Indians Diabetes

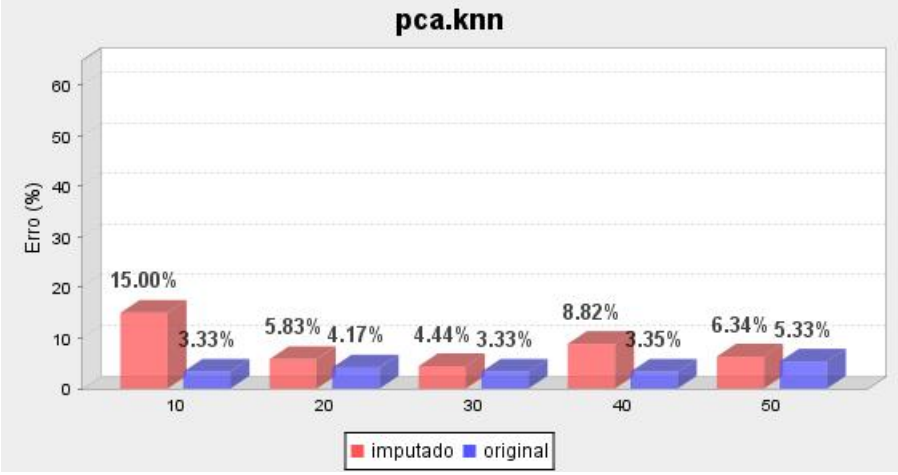


Wisconsin Breast Cancer

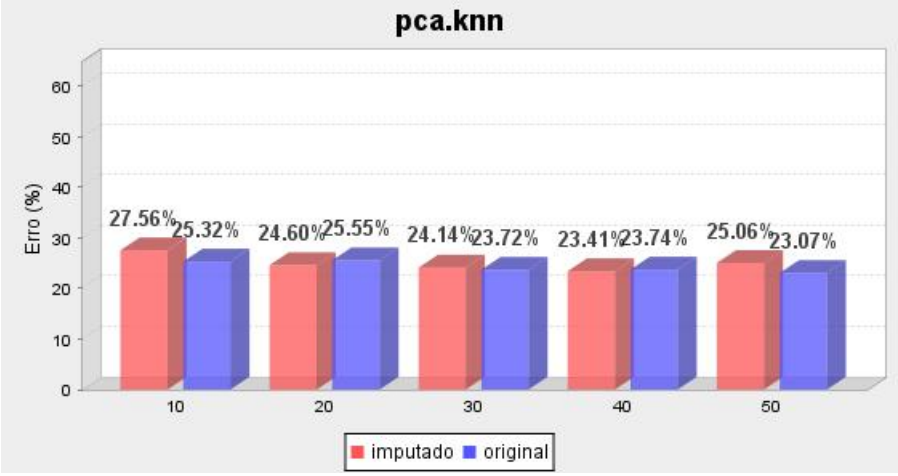


Plano 6: Seleção com PCA e Imputação com *k*-NN

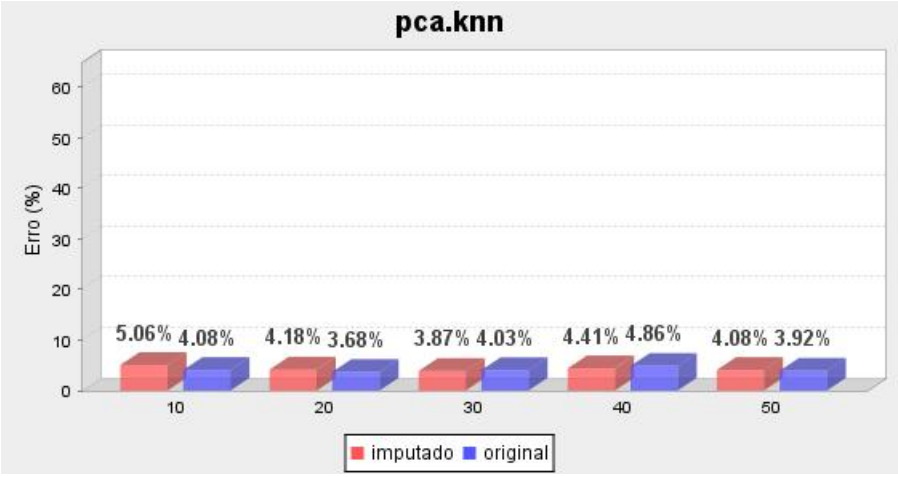
Iris Plants



Pima Indians Diabetes

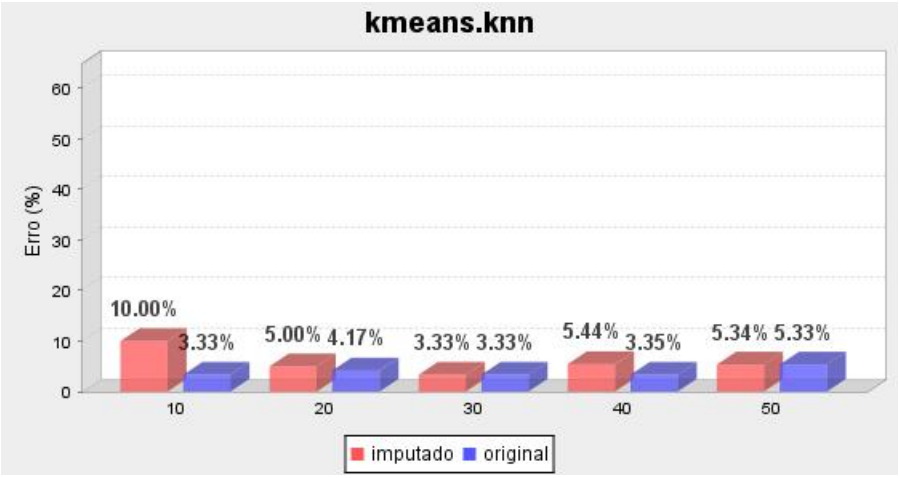


Wisconsin Breast Cancer

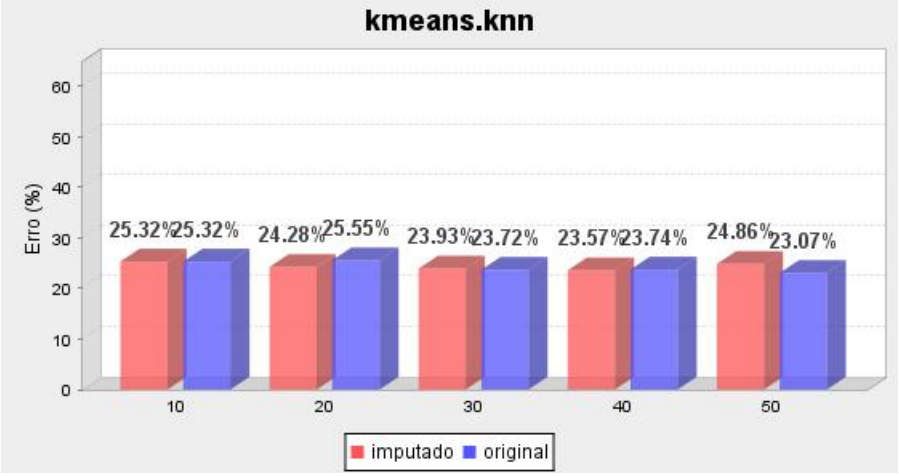


Plano 7: Agrupamento com *K*-Means e Imputação com *k*-NN

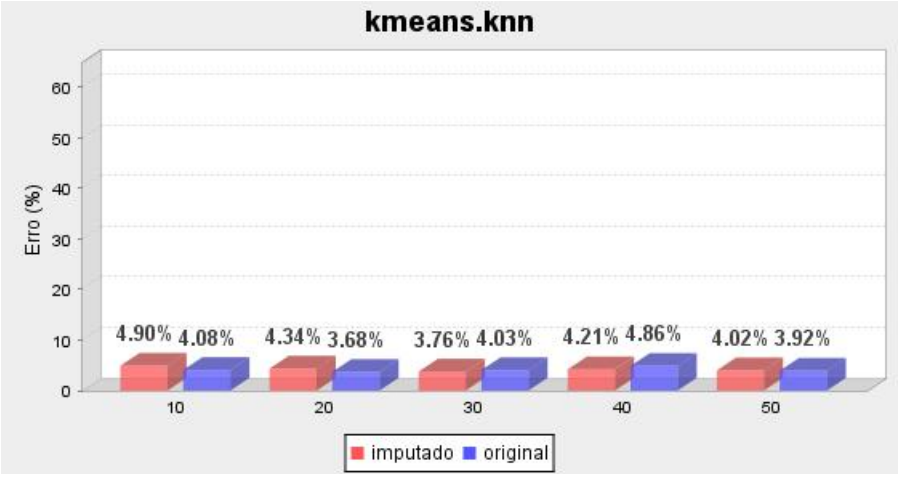
Iris Plants



Pima Indians Diabetes

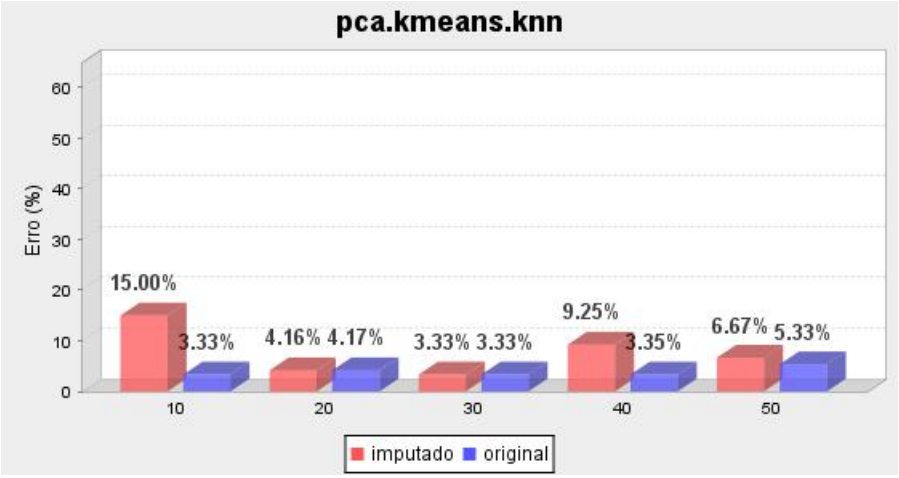


Wisconsin Breast Cancer

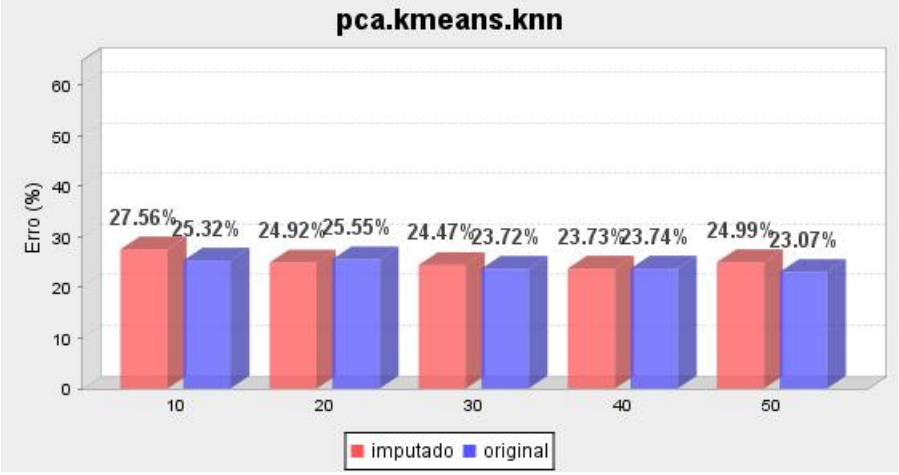


Plano 8: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *k*-NN

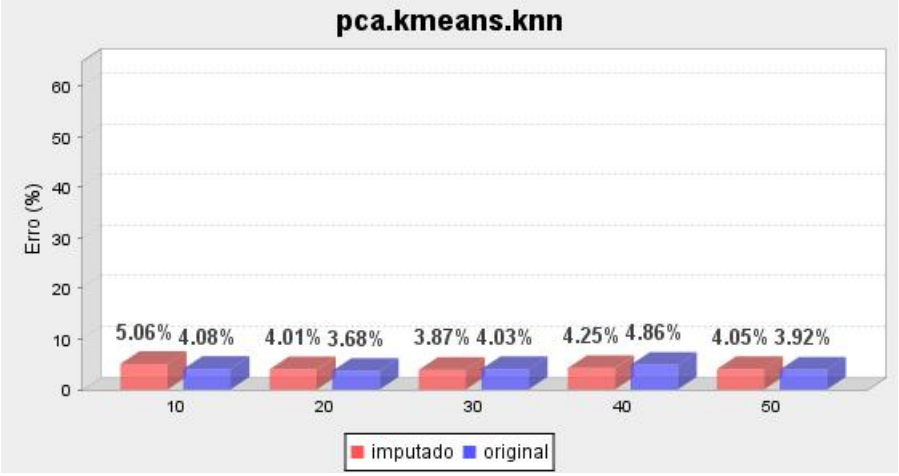
Iris Plants



Pima Indians Diabetes

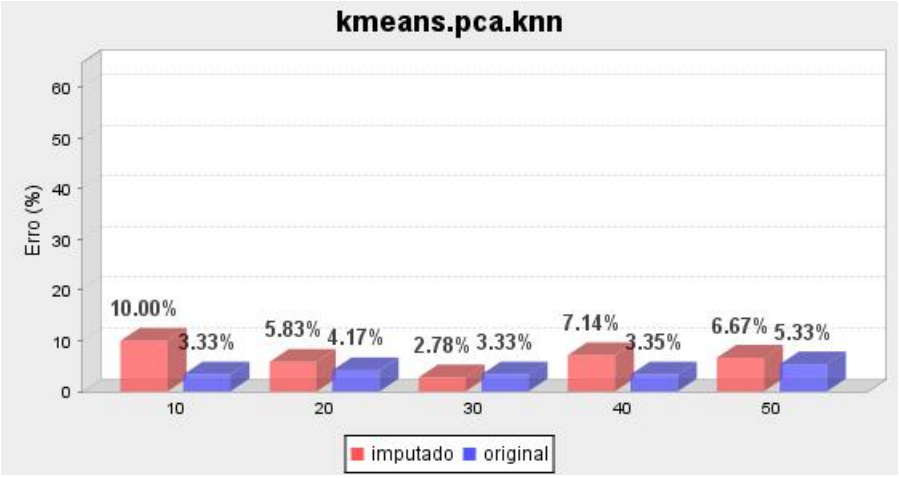


Wisconsin Breast Cancer

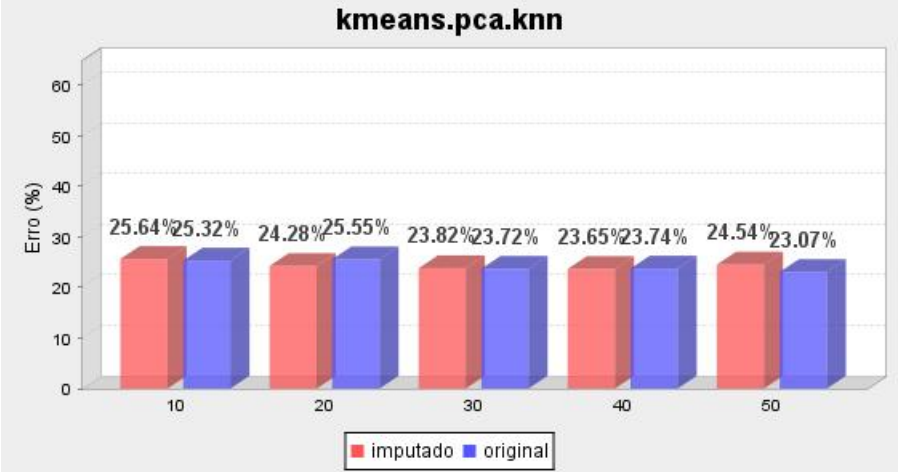


Plano 9: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *k*-NN

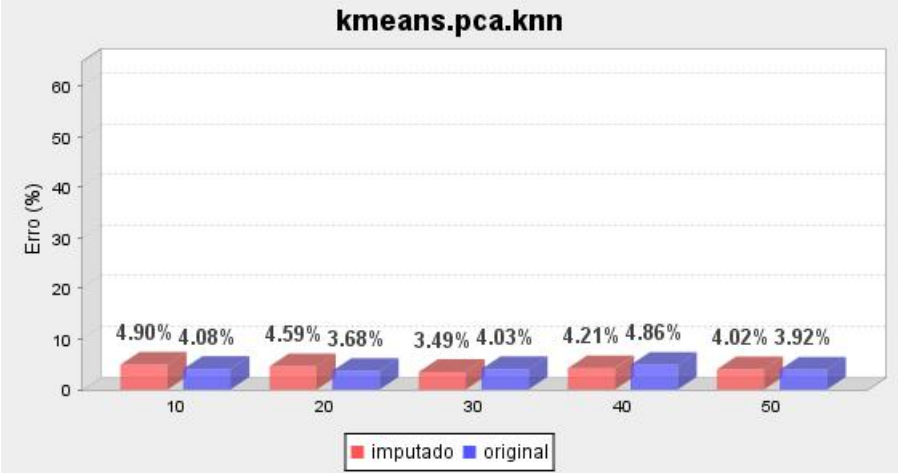
Iris Plants



Pima Indians Diabetes

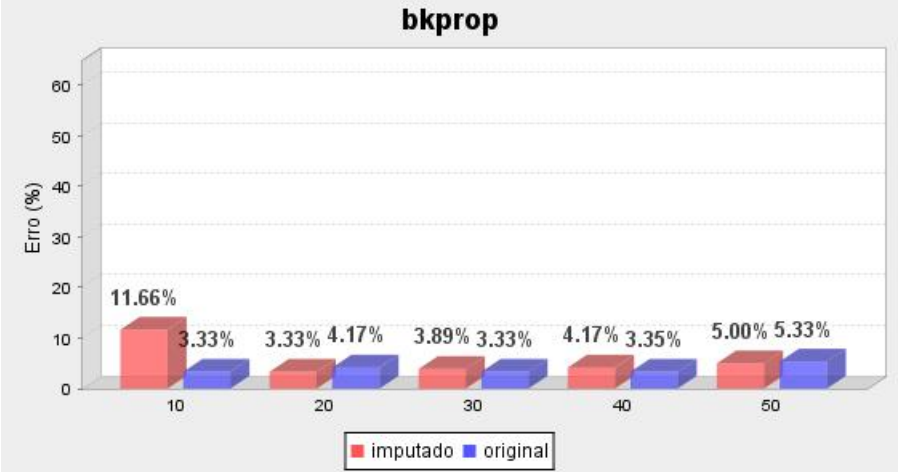


Wisconsin Breast Cancer

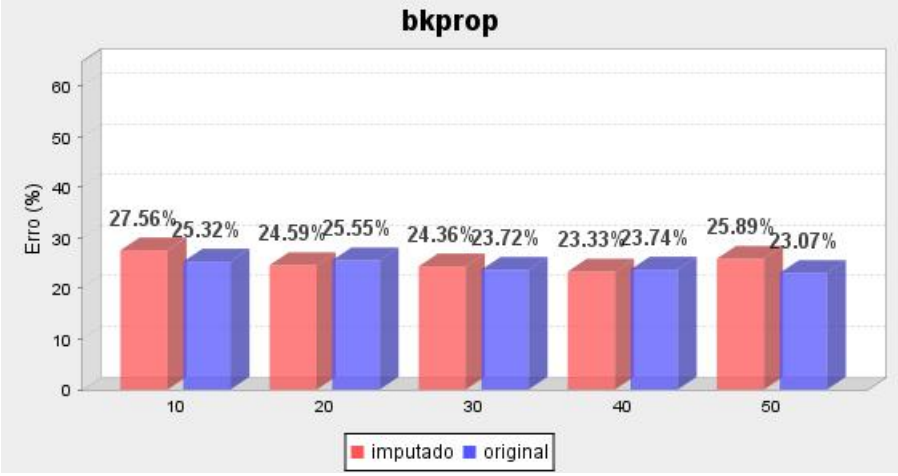


Plano 10: Imputação com *back propagation*

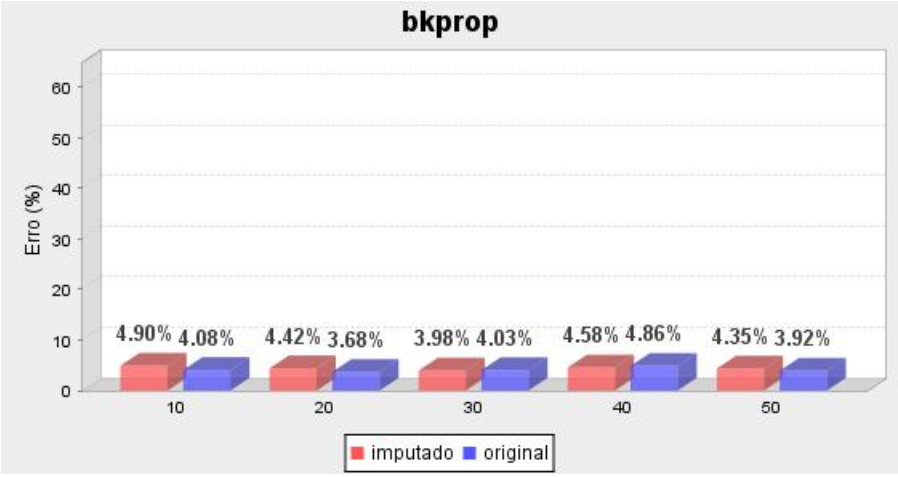
Iris Plants



Pima Indians Diabetes

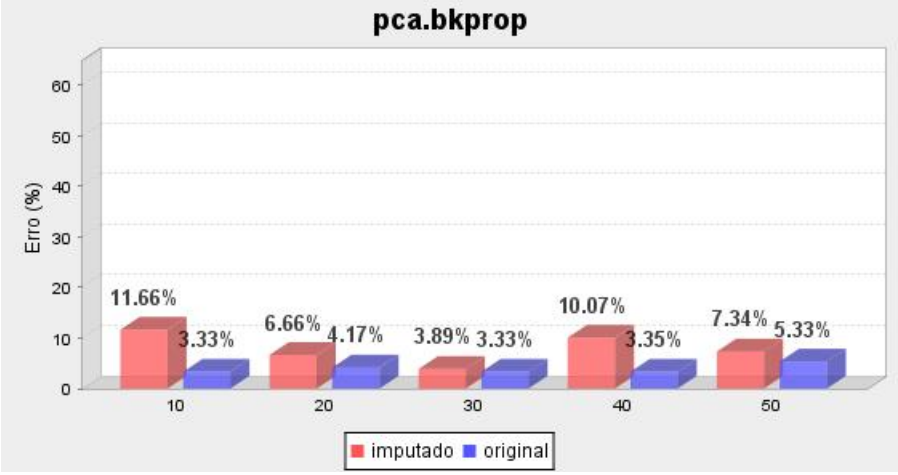


Wisconsin Breast Cancer

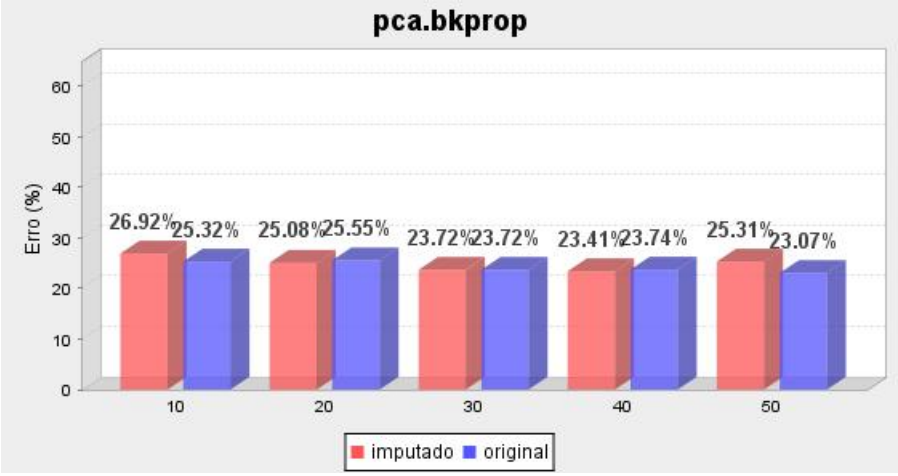


Plano 11: Seleção com PCA e Imputação com *back propagation*

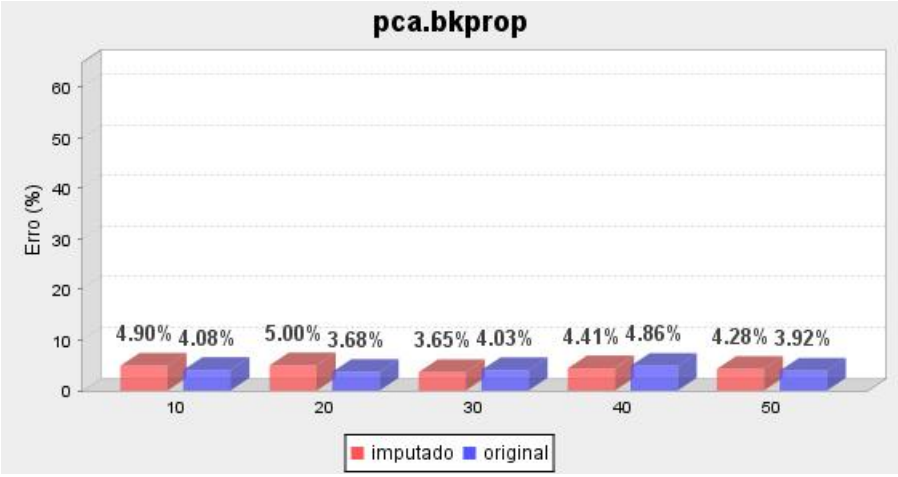
Iris Plants



Pima Indians Diabetes

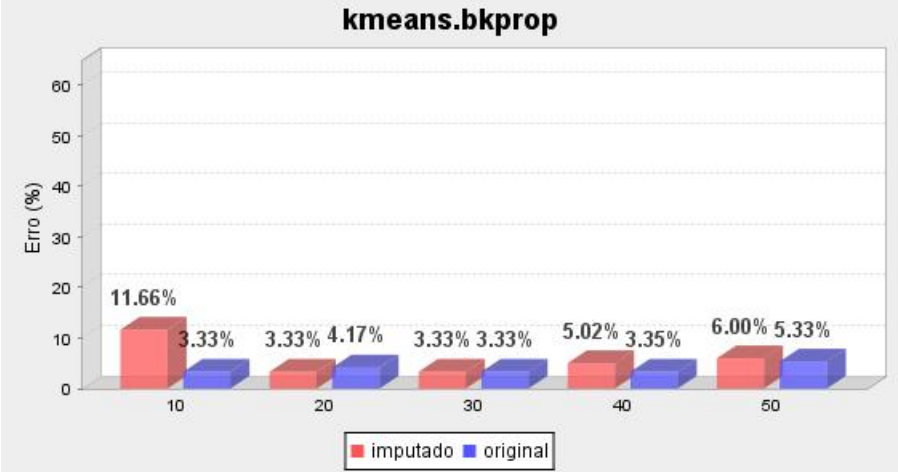


Wisconsin Breast Cancer

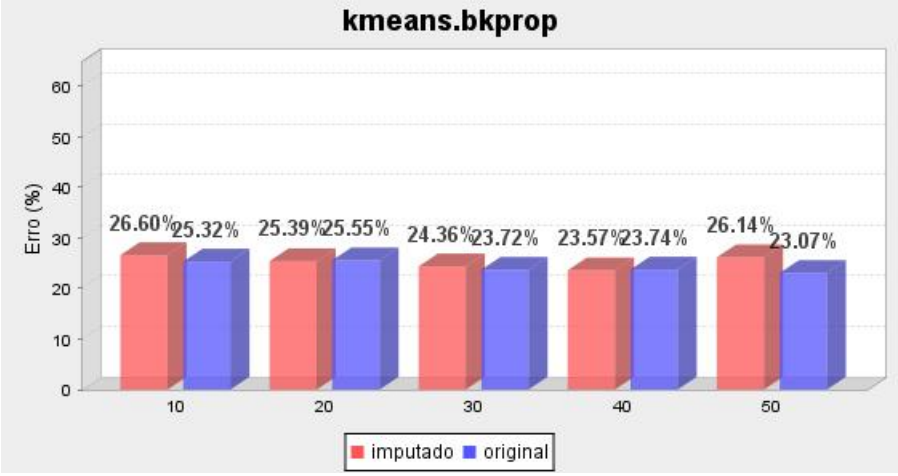


Plano 12: Agrupamento com *K*-Means e Imputação com *back propagation*

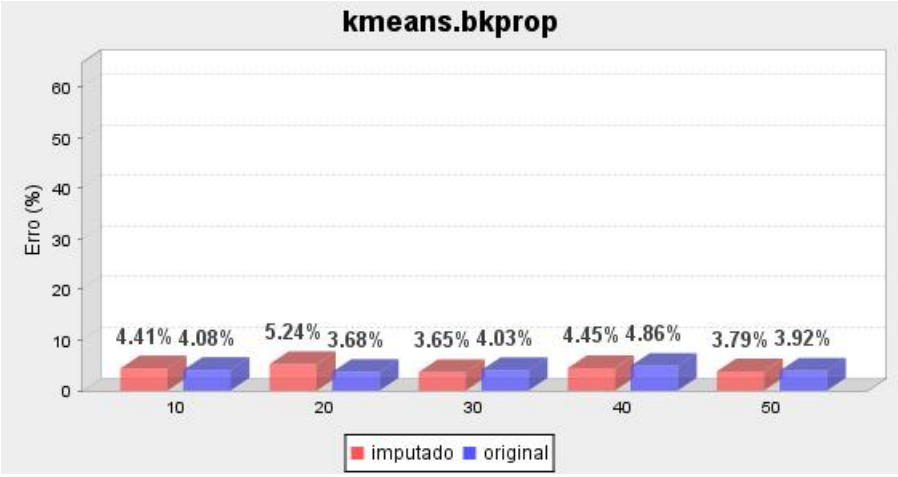
Iris Plants



Pima Indians Diabetes

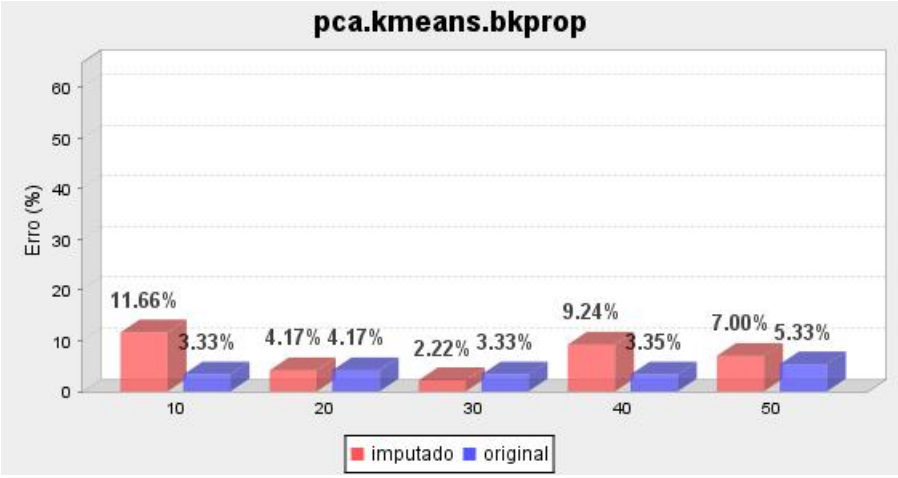


Wisconsin Breast Cancer

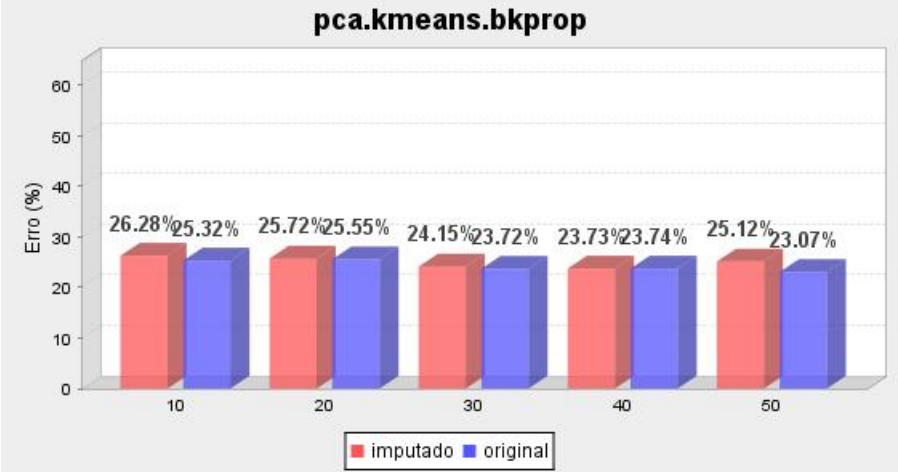


Plano 13: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *back propagation*

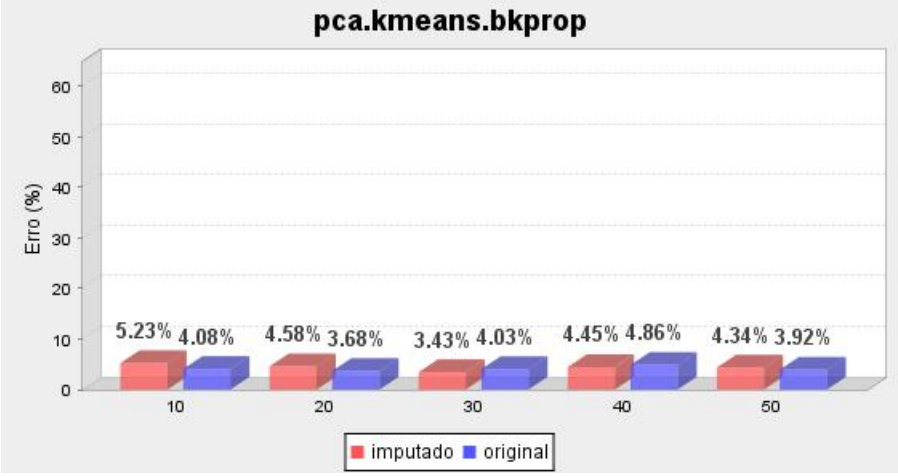
Iris Plants



Pima Indians Diabetes

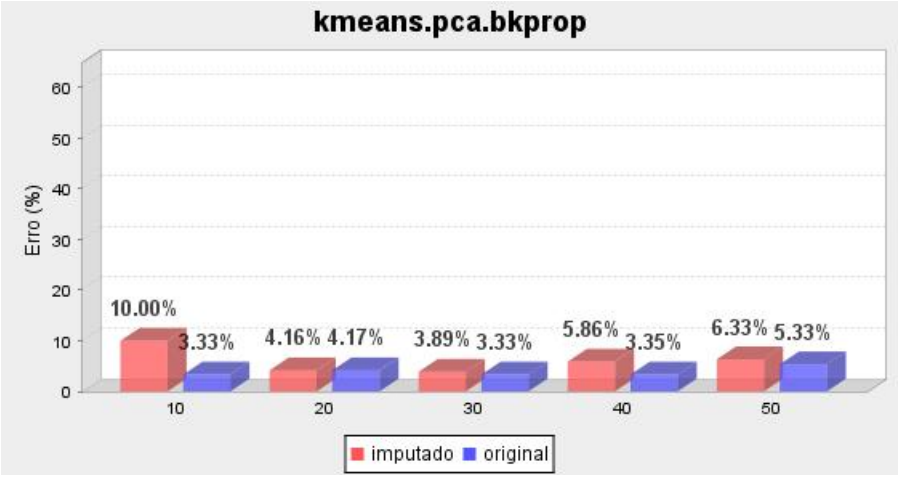


Wisconsin Breast Cancer

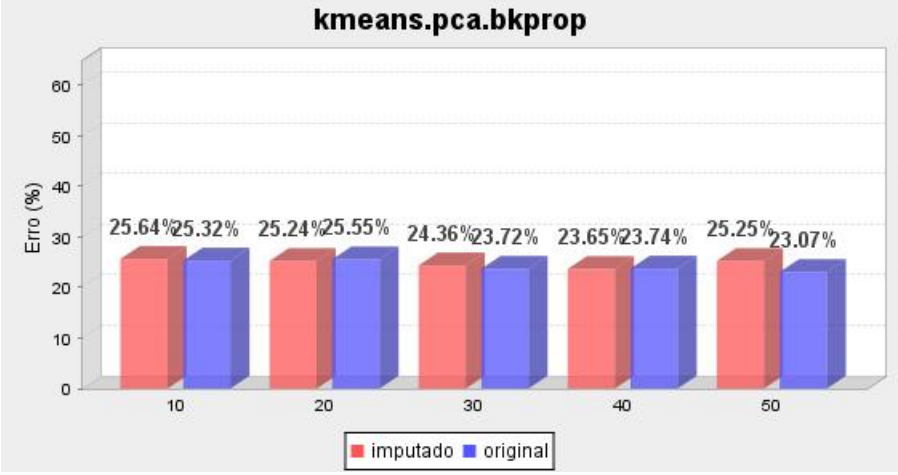


Plano 14: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *back propagation*

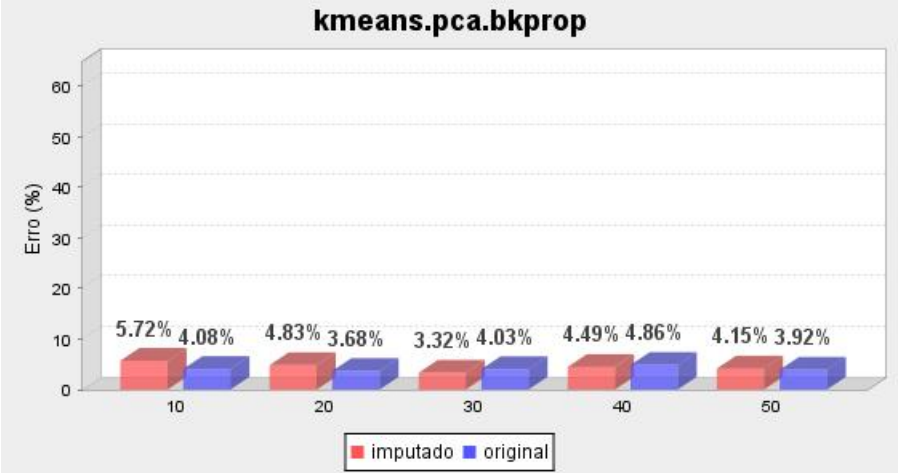
Iris Plants



Pima Indians Diabetes

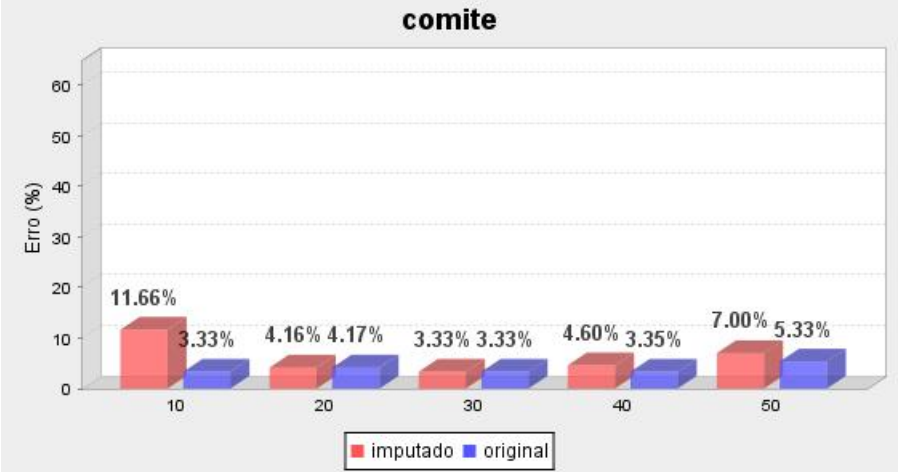


Wisconsin Breast Cancer

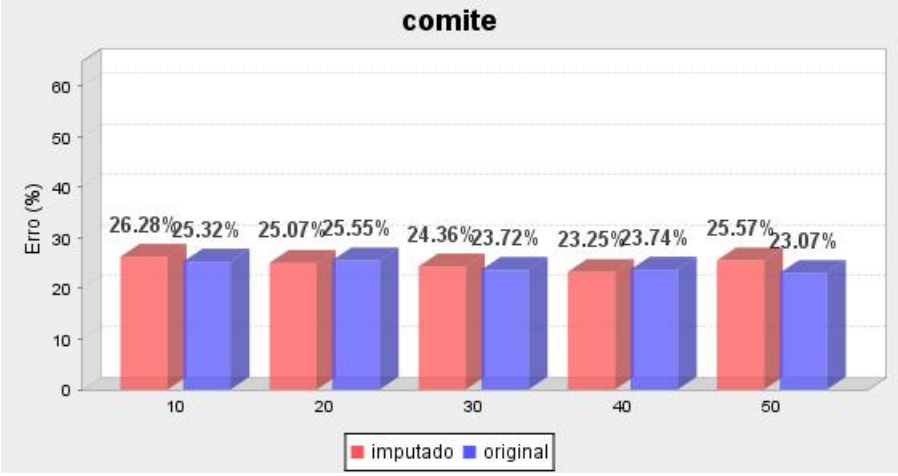


Plano 15: Comitês de Complementação de Dados

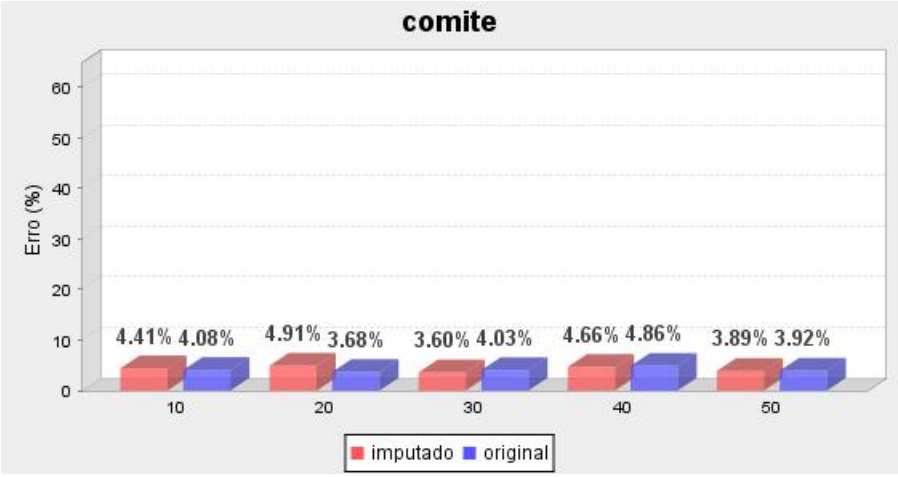
Iris Plants



Pima Indians Diabetes



Wisconsin Breast Cancer



5.2.5 Erros médios de classificação por atributos da base e por planos de imputação

5.2.5.1 Análise dos resultados

Aqui adotamos a mesma metodologia da subseção anterior. Analisamos, para cada base, atributo com valores imputados e plano de imputação utilizado, qual o erro médio de classificação ocorrido, tanto com tuplas com valores imputados quanto com as tuplas originais da base.

Os gráficos desta subseção avaliam o erro médio de classificação das tuplas por base e por atributo que sofreu o processo de imputação. Esse erro médio é comparado com o erro médio de classificação das tuplas originais. Com isso, queremos avaliar a qualidade do classificador (em quantas vezes o classificador respondeu de forma equivocada qual a classe correta da tupla com seus valores originais).

As observações feitas para a seção anterior para as bases *Pima Indians Diabetes* e *Wisconsin Breast Cancer* (erros médios de classificação por percentual de valores ausentes) também são válidas para a análise de erros médios de classificação por atributos da base e por planos de imputação. Ambos os conjuntos de dados apresentaram comportamento similar independentemente do plano de imputação utilizado: a base *Pima Indians Diabetes* revelando altas taxas médias de erro de classificação (provavelmente por conta da fraca correlação entre os seus atributos), com taxas variando entre 23,5% e 29%, mas com uma baixa diferença entre os erros médios de classificação entre as tuplas com valor imputado e as originais. Esta flutuação girou entre 1% e 3%, o que mostra que o problema não reside na imputação em si, mas na qualidade do classificador para esta base. A base *Wisconsin Breast Cancer* mostra erros médios de classificação girando entre 3% e 8,5%, e diferenças médias de erro entre tuplas originais e imputadas entre também entre 1% e 3%.

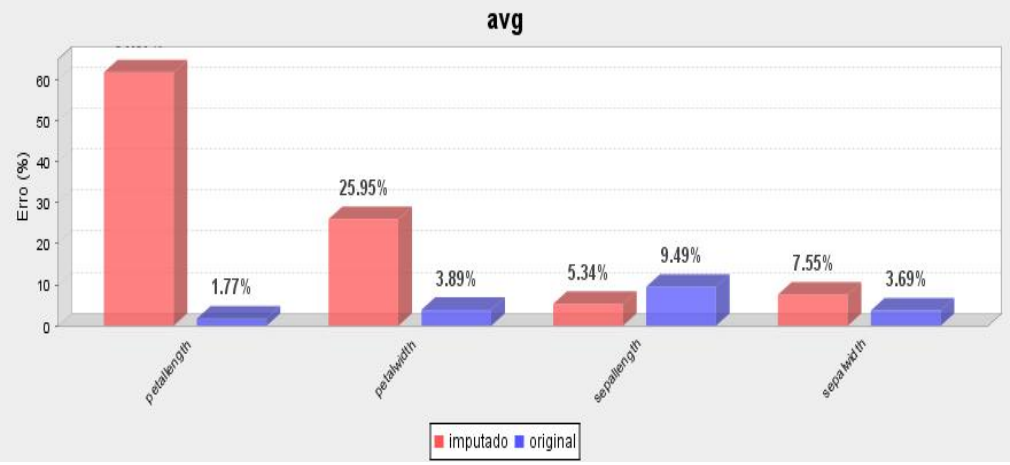
Na base *Iris Plants*, mais uma vez os erros médios de classificação com dados gerados pela imputação com média são bastante destoantes dos erros médios de classificação das tuplas originais nos atributos *petallength* e *petalwidth*. As tuplas regredidas no atributo *sepalwidth*, que apresenta baixa correlação com os demais atributos, não apresentaram uma alta taxa média de erros de classificação em nenhum dos planos de imputação. Em alguns planos, o atributo *petalwidth* foi mais sensível ao processo de complementação, e apresentou erros médios de classificação por volta

de 5% maiores do que as classificações das tuplas originais. Porém, o fato que nos chamou a atenção foi o erro médio de classificação das tuplas imputadas ser sempre menor do que com as tuplas originais no atributo *sepallength*. Não encontramos nenhuma razão plausível que justificasse tal acontecimento.

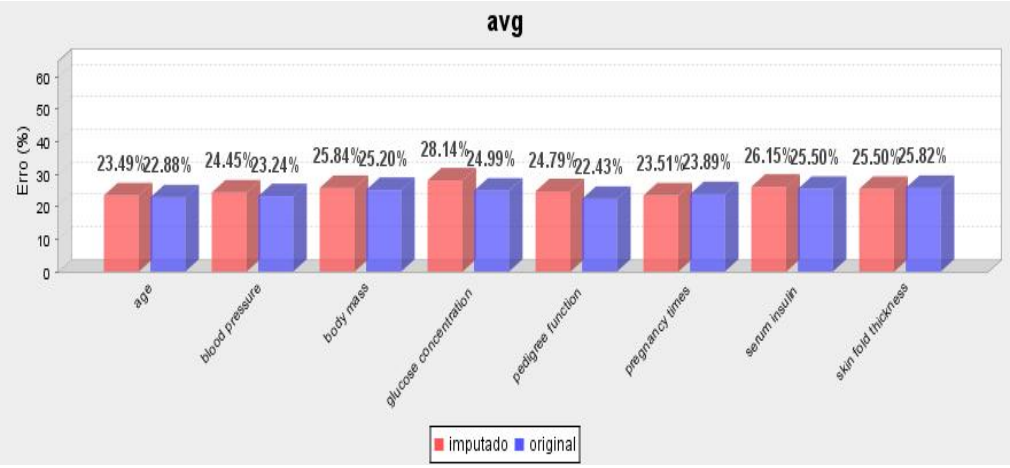
5.2.5.2 Gráficos

Plano 1: Imputação com Média

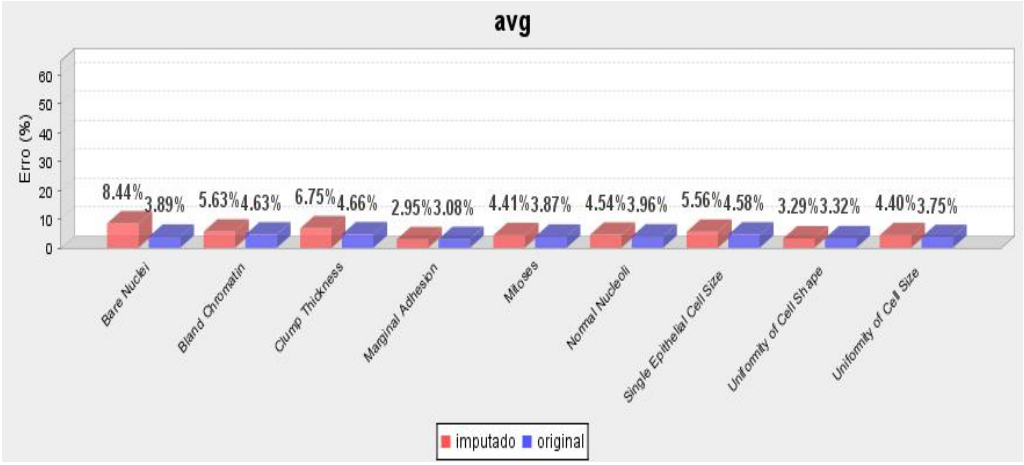
Iris Plants



Pima Indians Diabetes

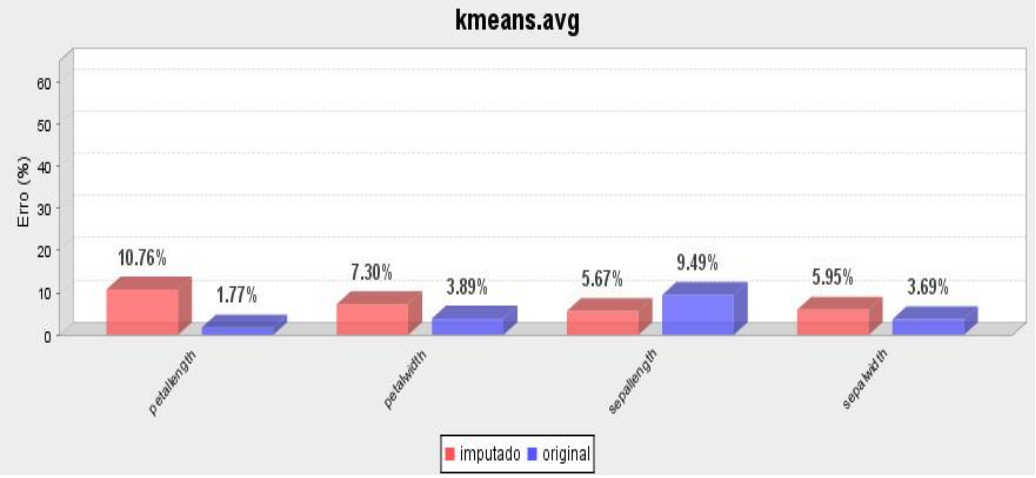


Wisconsin Breast Cancer

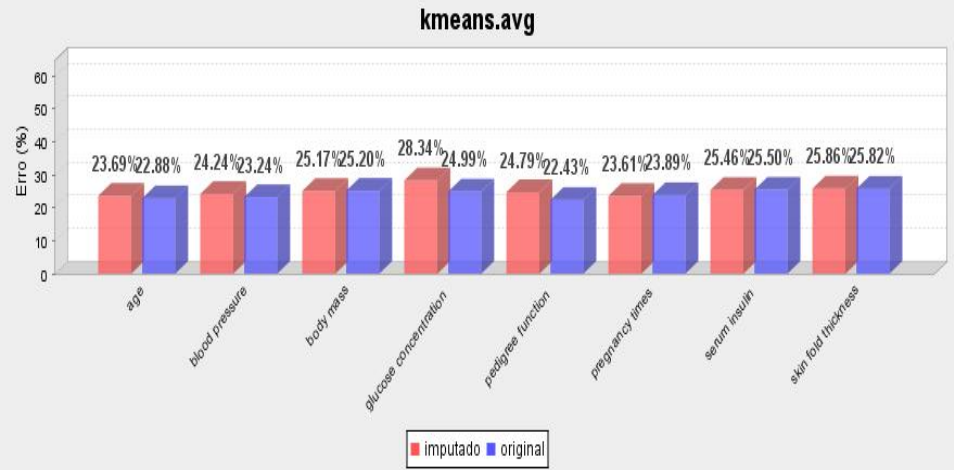


Plano 2: Agrupamento com K-Means e Imputação com Média

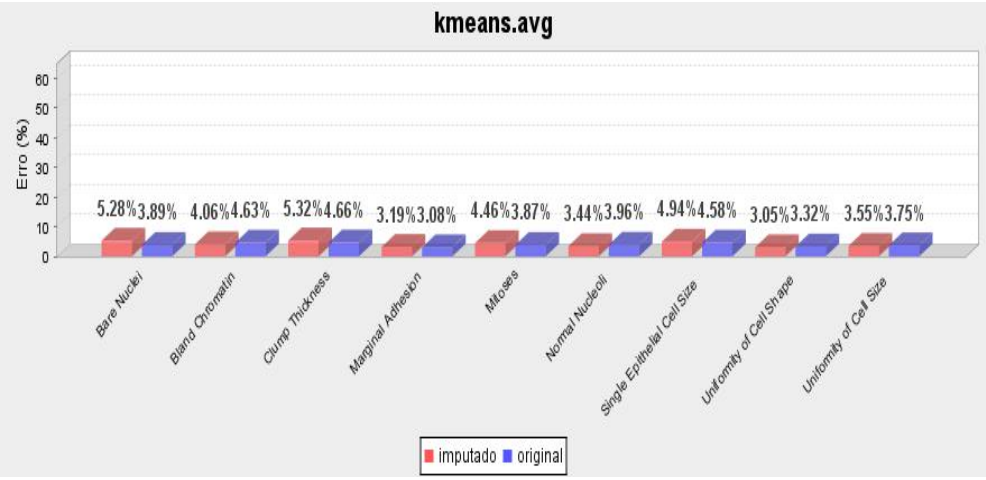
Iris Plants



Pima Indians Diabetes

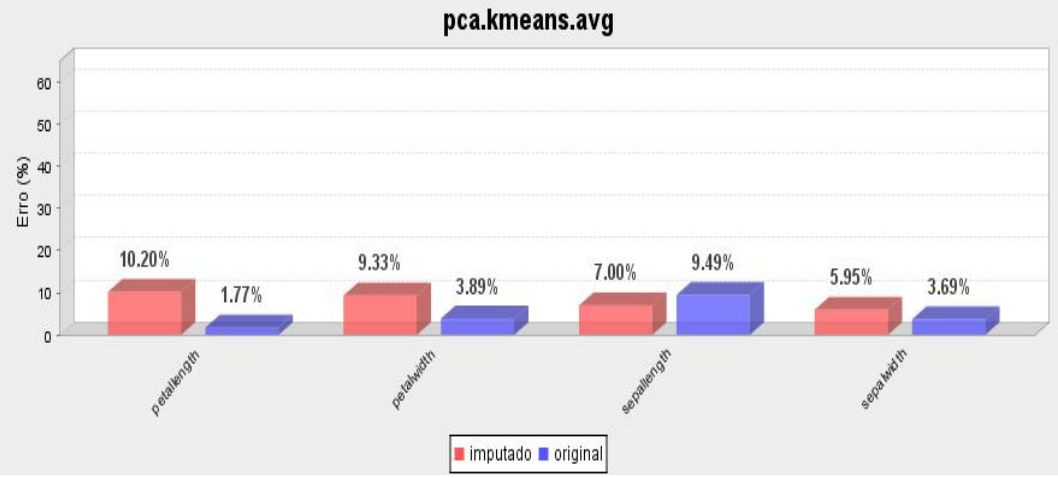


Wisconsin Breast Cancer

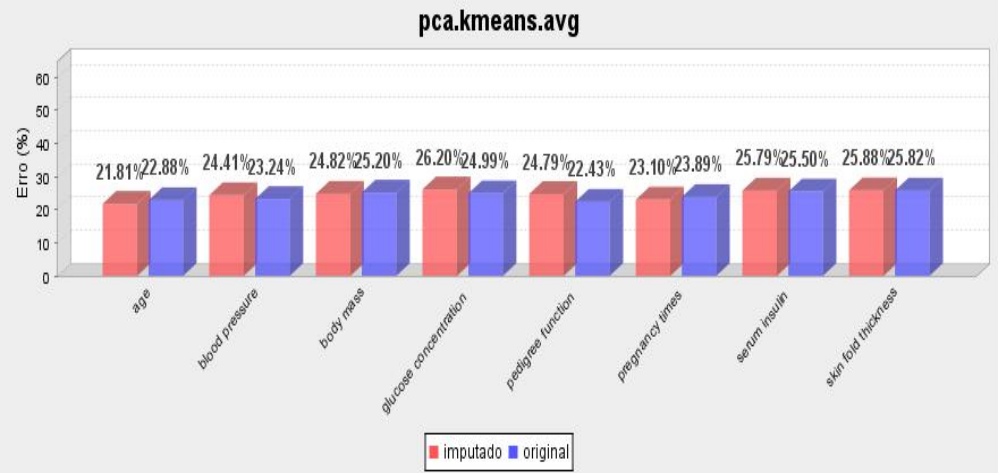


Plano 3: Seleção com PCA, Agrupamento com K-Means e Imputação com Média

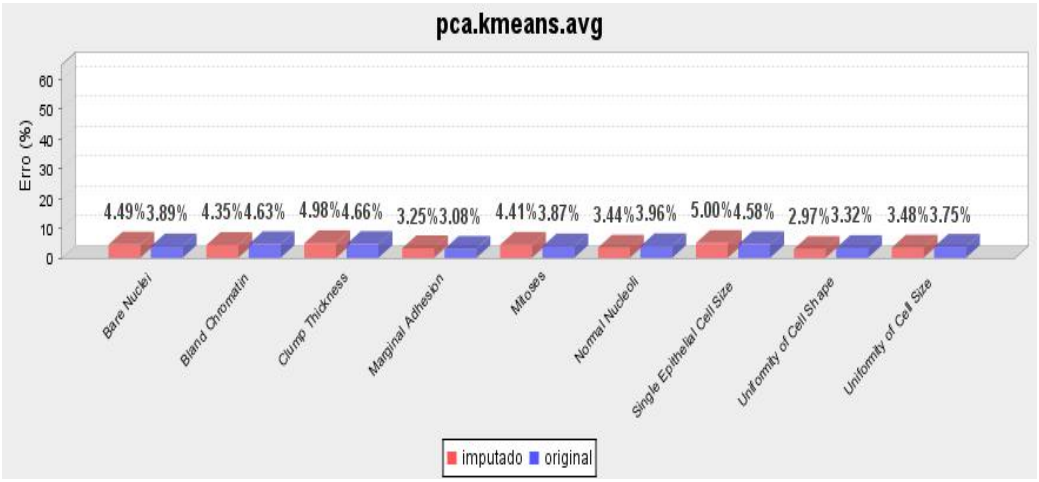
Iris Plants



Pima Indians Diabetes

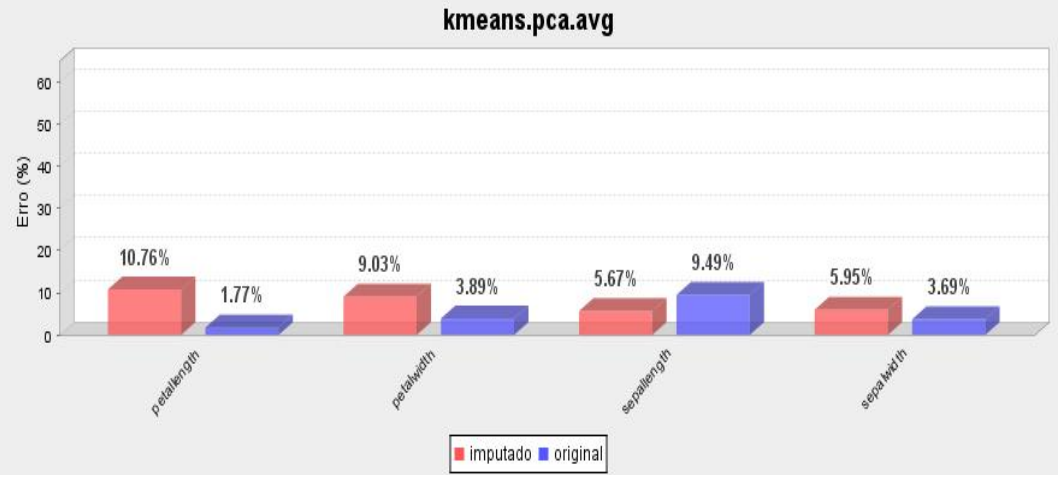


Wisconsin Breast Cancer

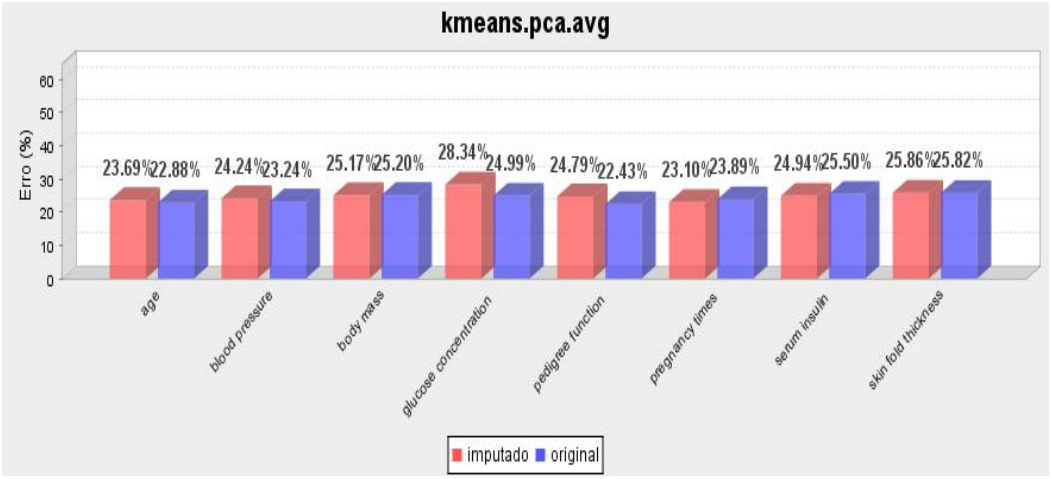


Plano 4: Agrupamento com K-Means, Seleção com PCA e Imputação com Média

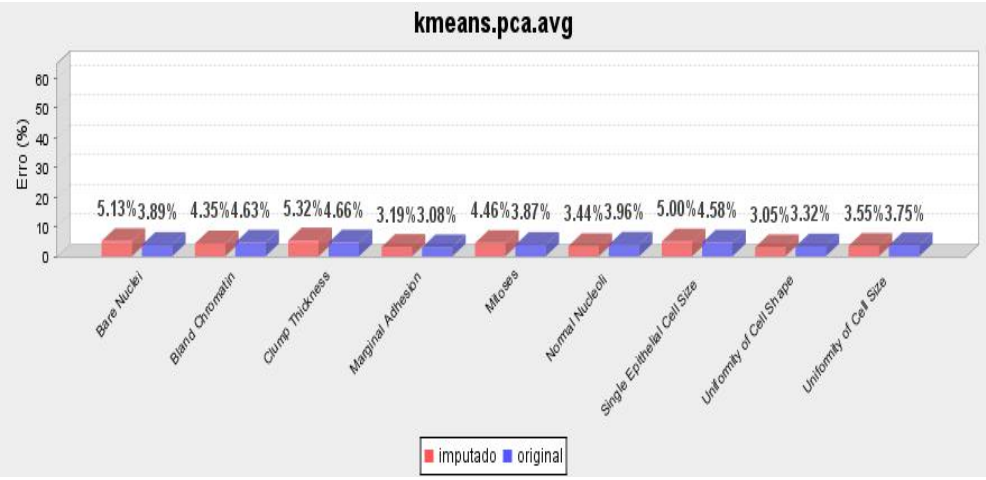
Iris Plants



Pima Indians Diabetes

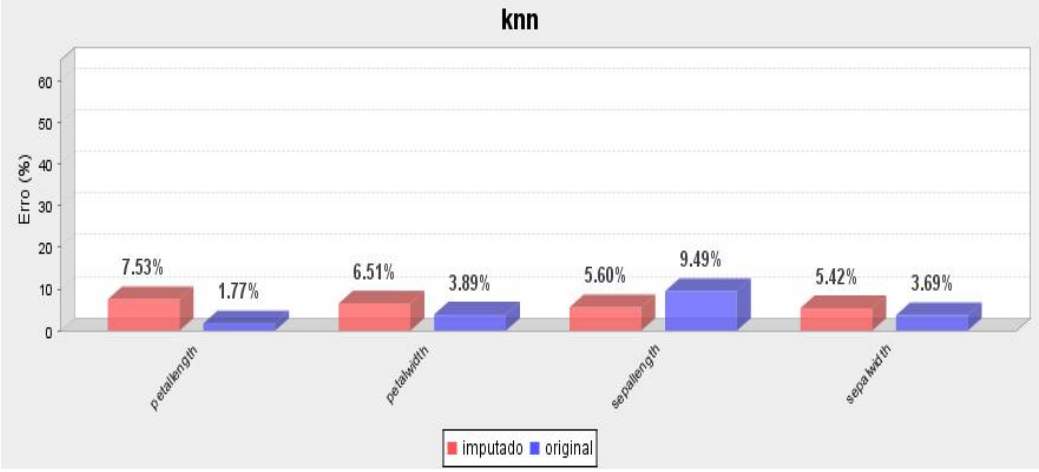


Wisconsin Breast Cancer

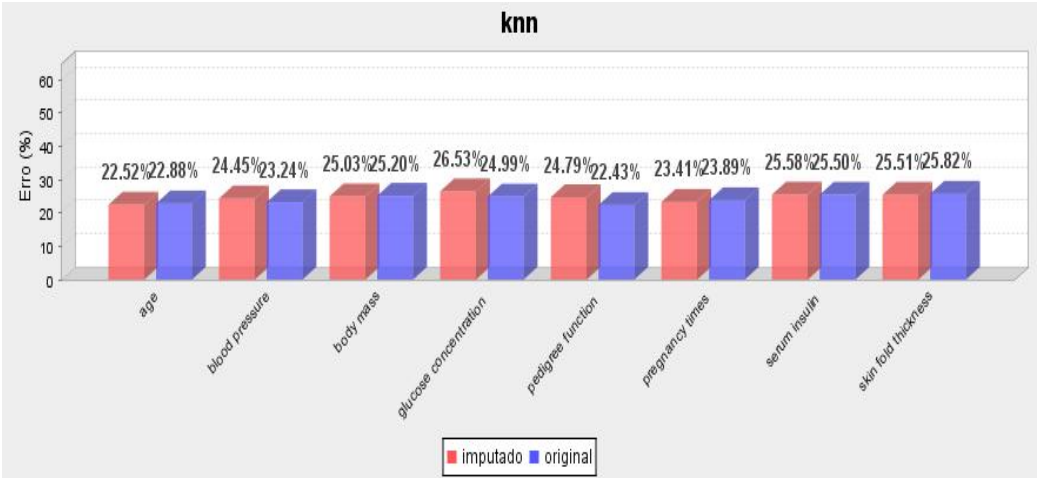


Plano 5: Imputação com *k*-NN

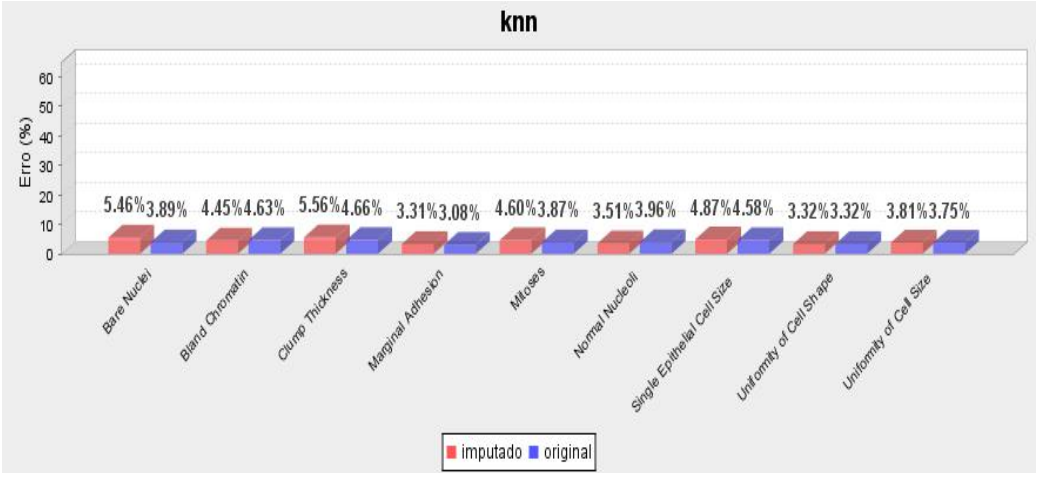
Iris Plants



Pima Indians Diabetes

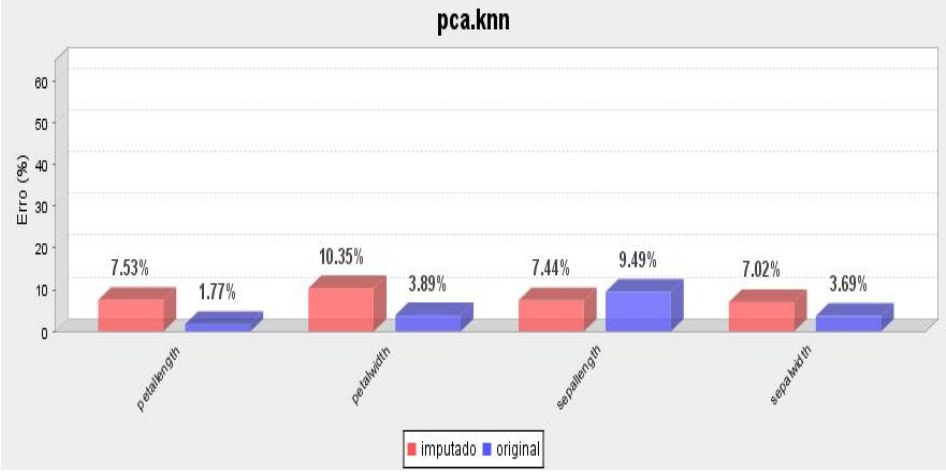


Wisconsin Breast Cancer

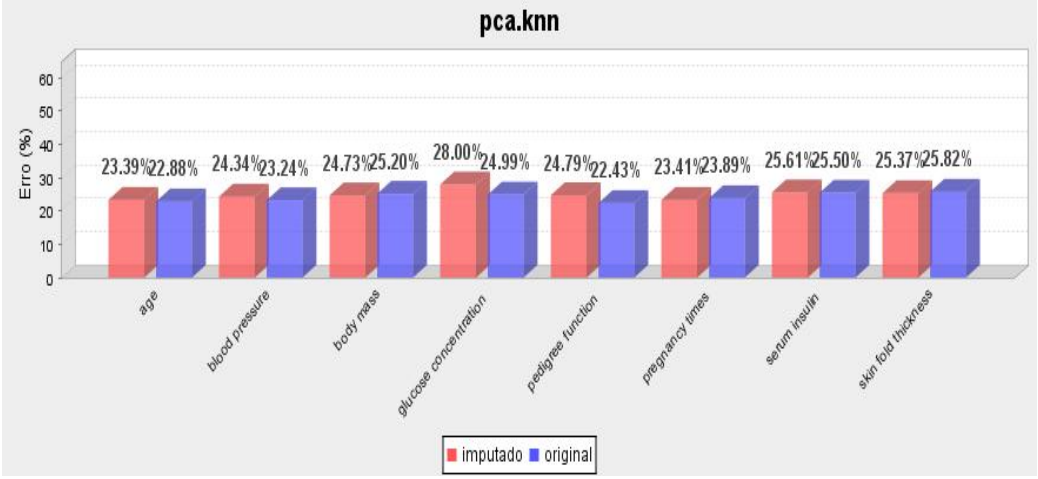


Plano 6: Seleção com PCA e Imputação com *k*-NN

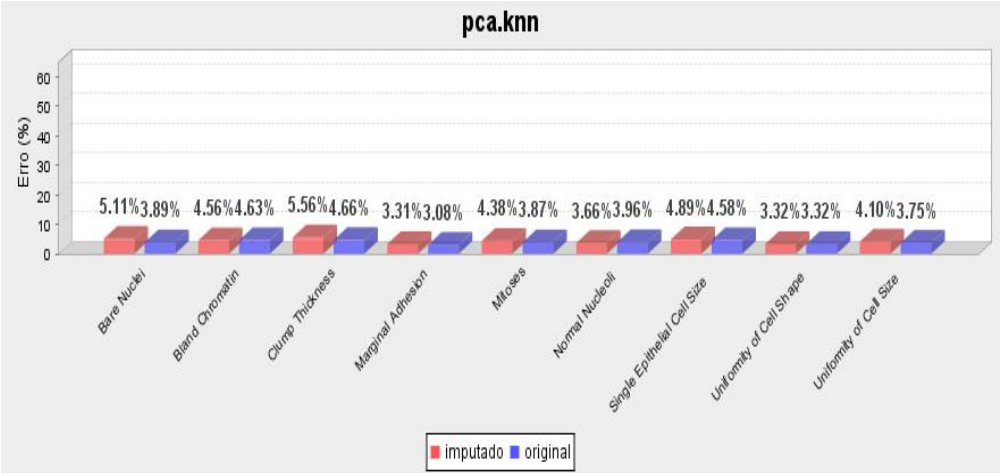
Iris Plants



Pima Indians Diabetes

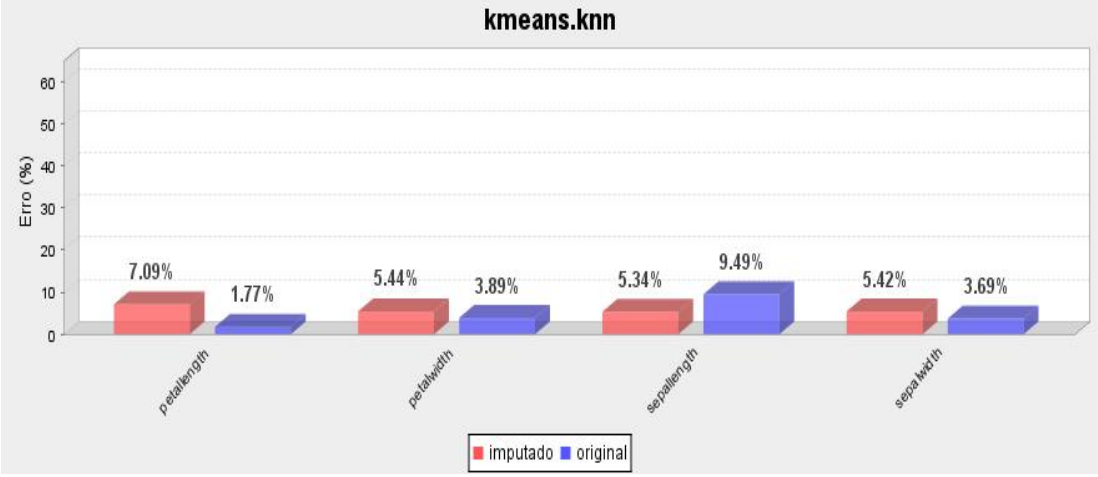


Wisconsin Breast Cancer

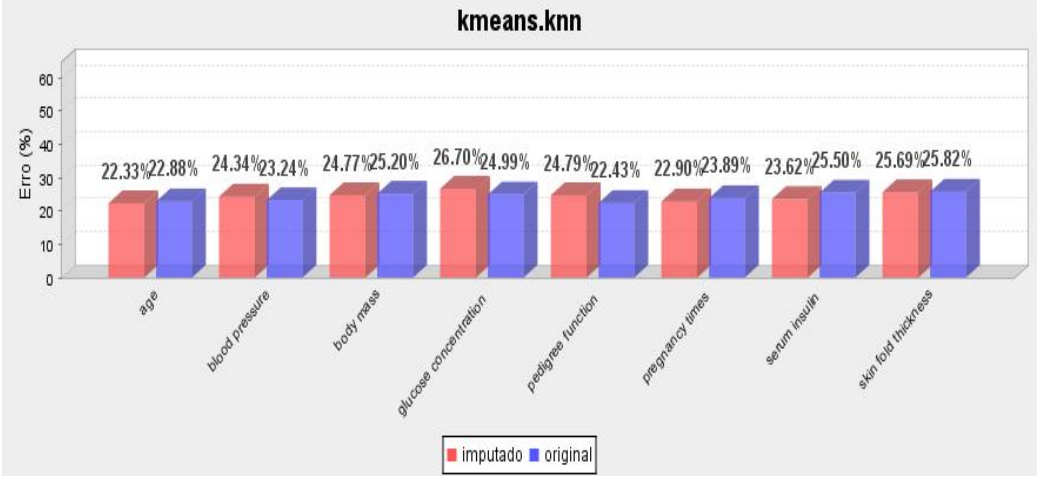


Plano 7: Agrupamento com *K*-Means e Imputação com *k*-NN

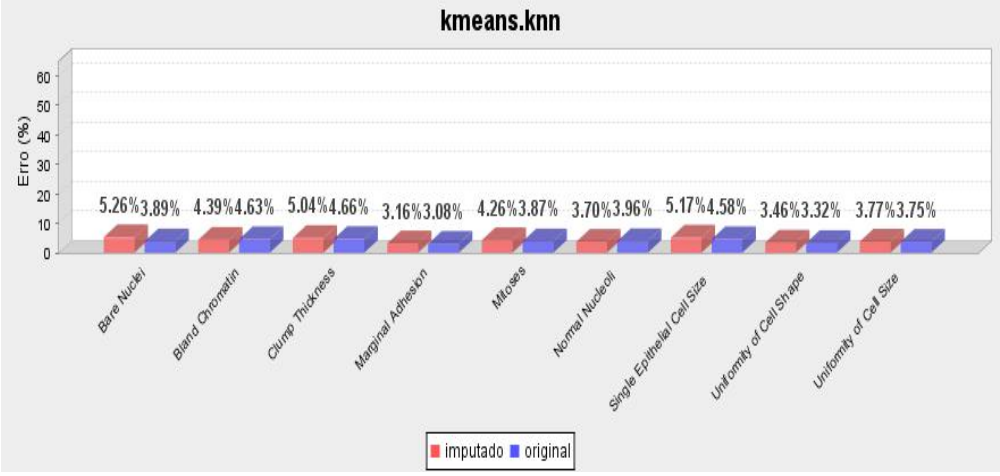
Iris Plants



Pima Indians Diabetes

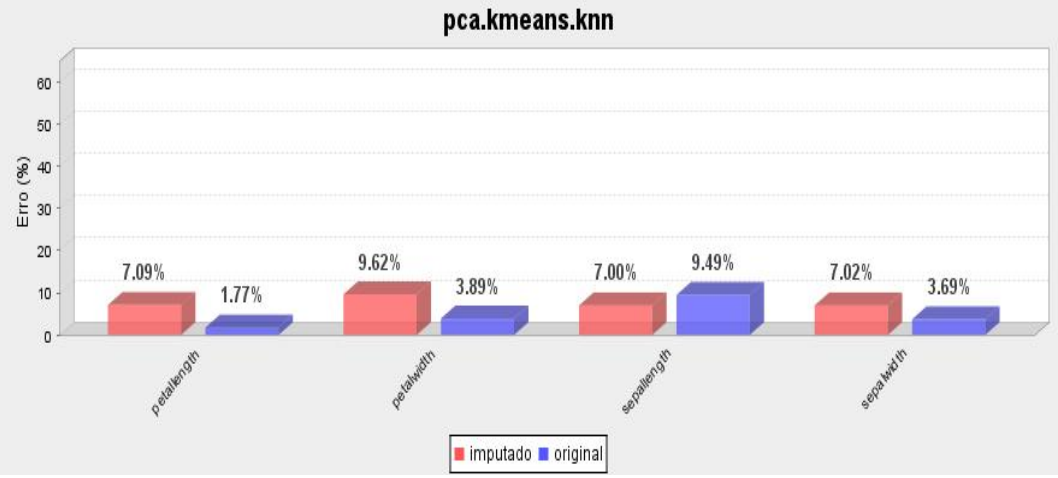


Wisconsin Breast Cancer

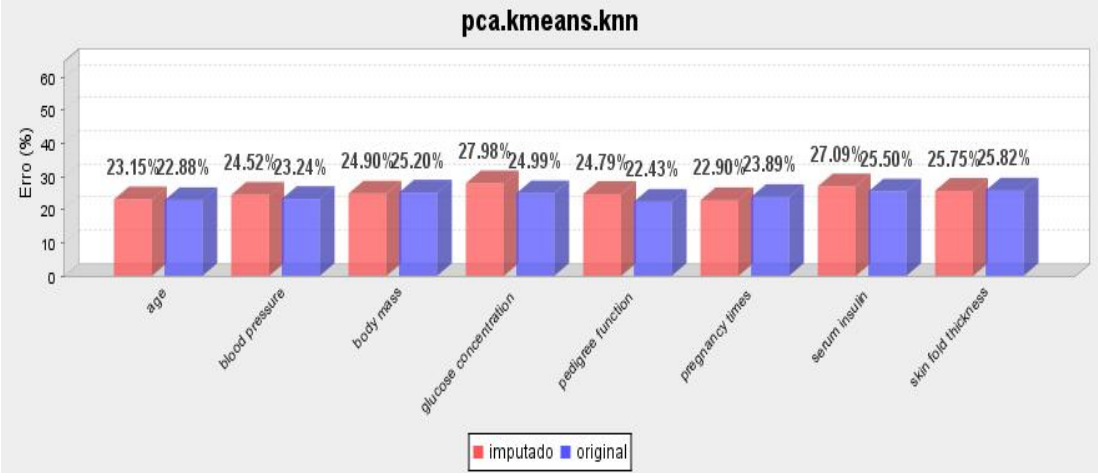


Plano 8: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *k*-NN

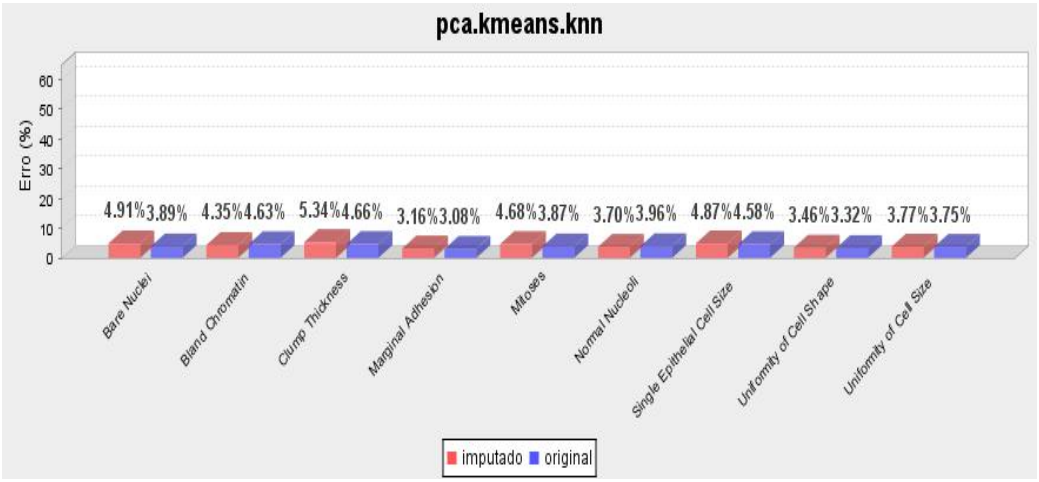
Iris Plants



Pima Indians Diabetes

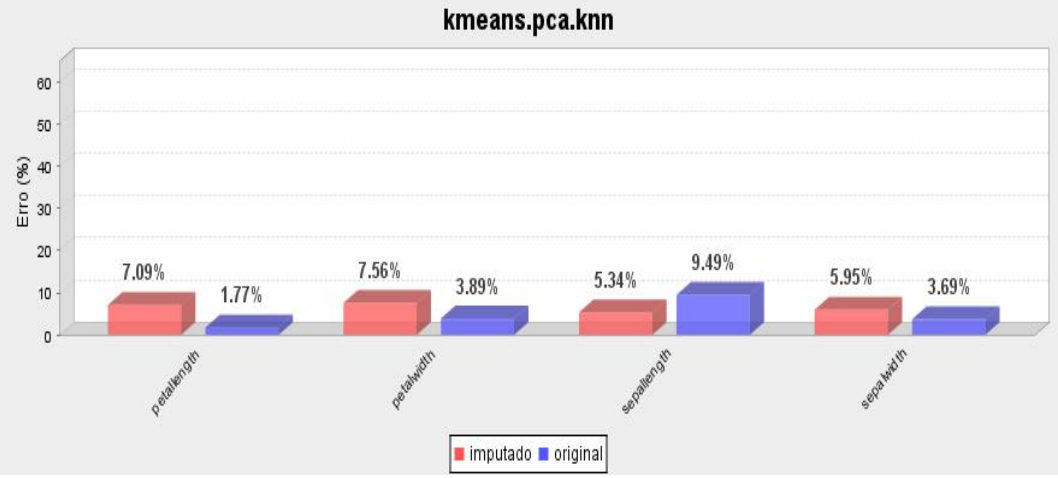


Wisconsin Breast Cancer

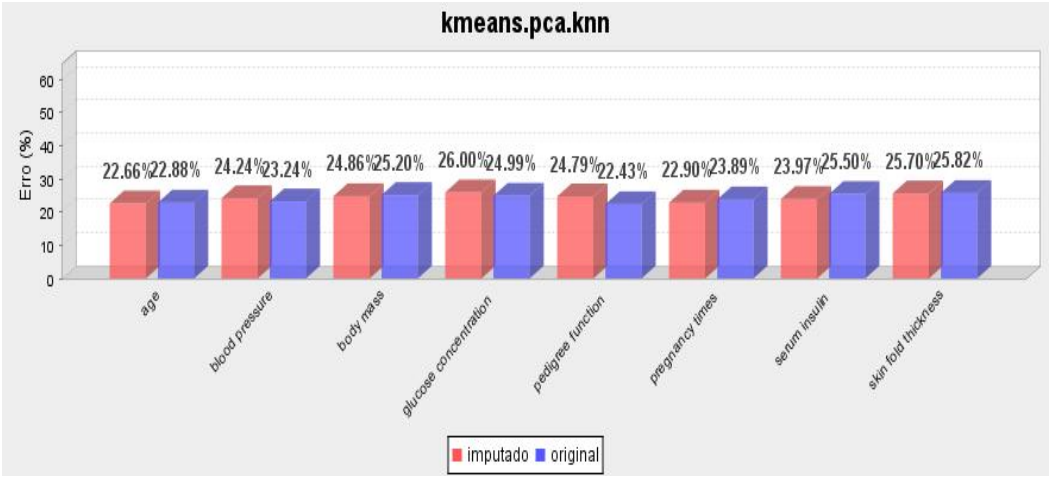


Plano 9: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *k*-NN

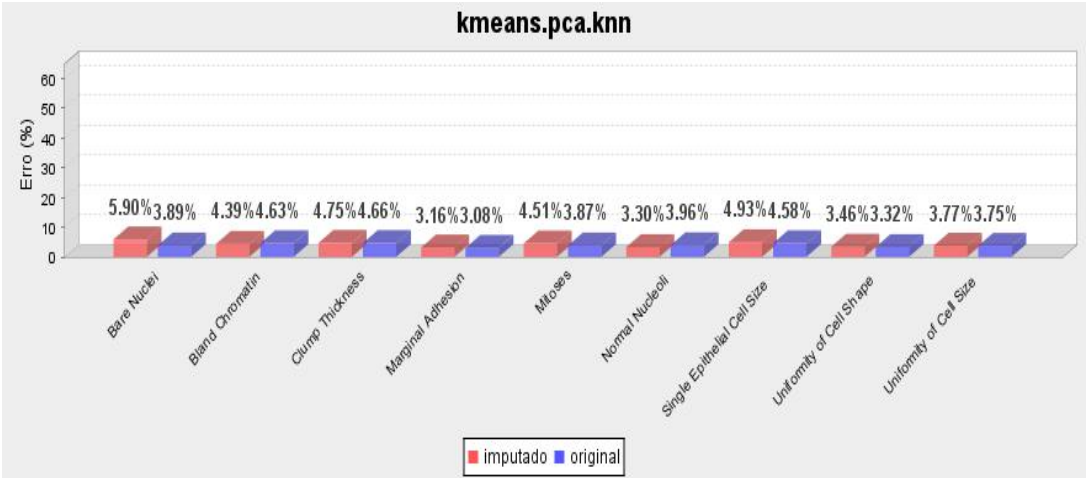
Iris Plants



Pima Indians Diabetes

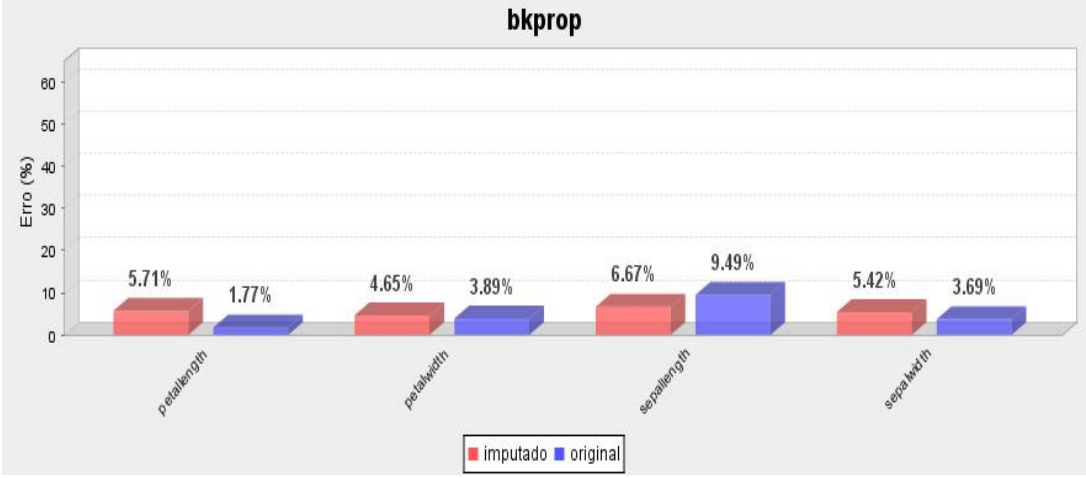


Wisconsin Breast Cancer

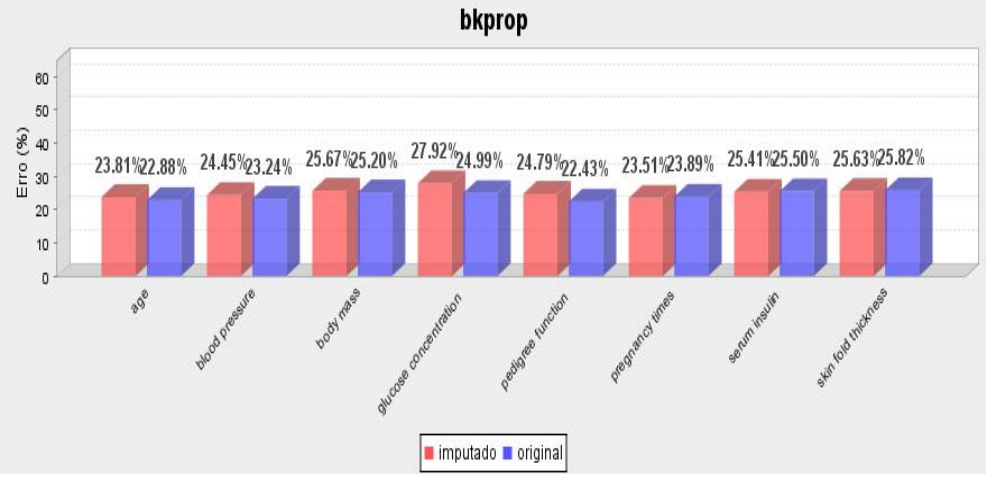


Plano 10: Imputação com *back propagation*

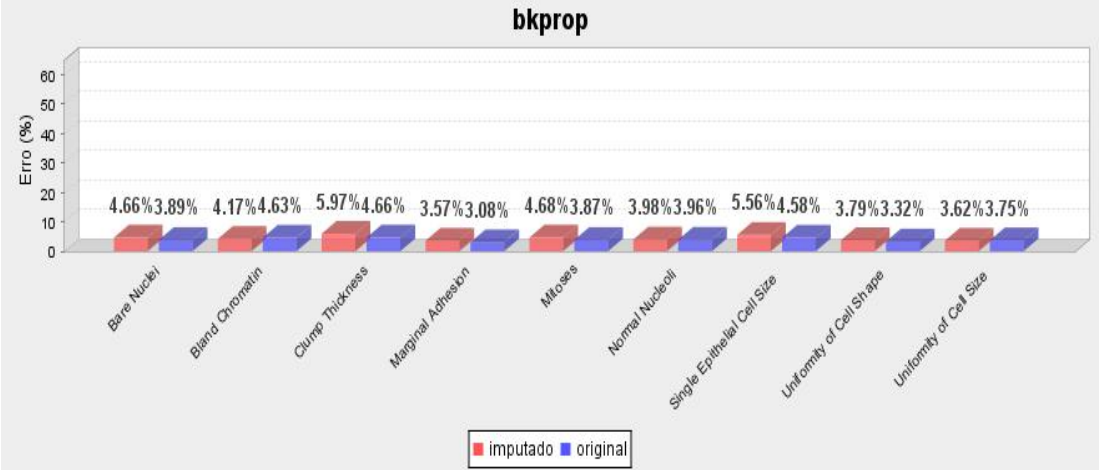
Iris Plants



Pima Indians Diabetes

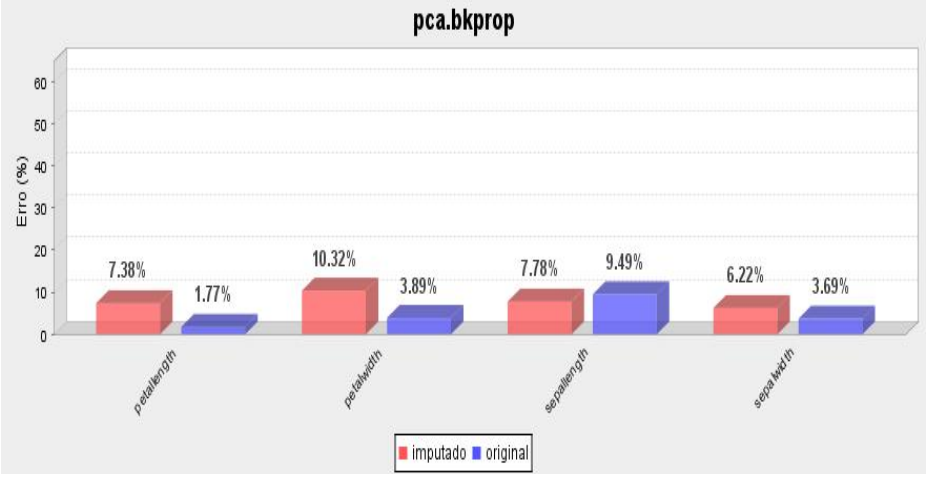


Wisconsin Breast Cancer

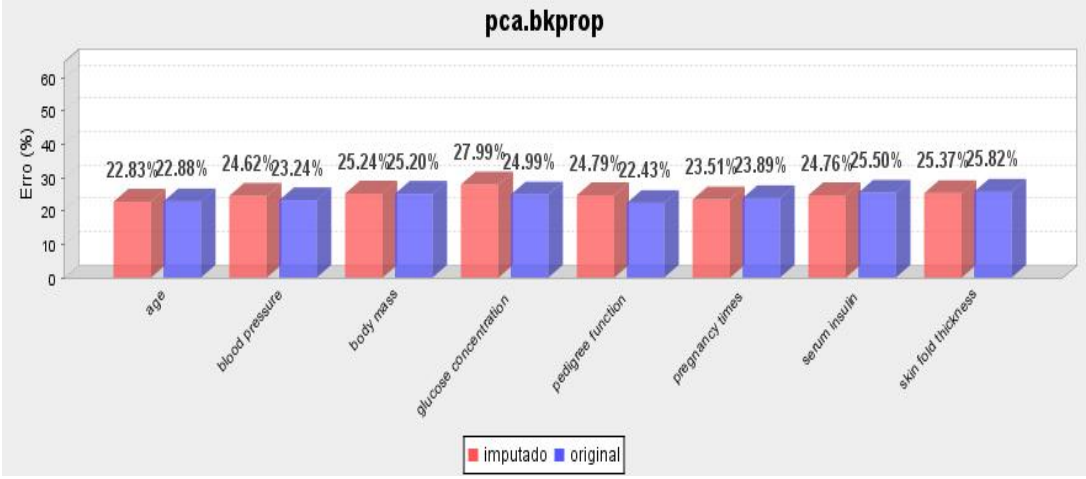


Plano 11: Seleção com PCA e Imputação com *back propagation*

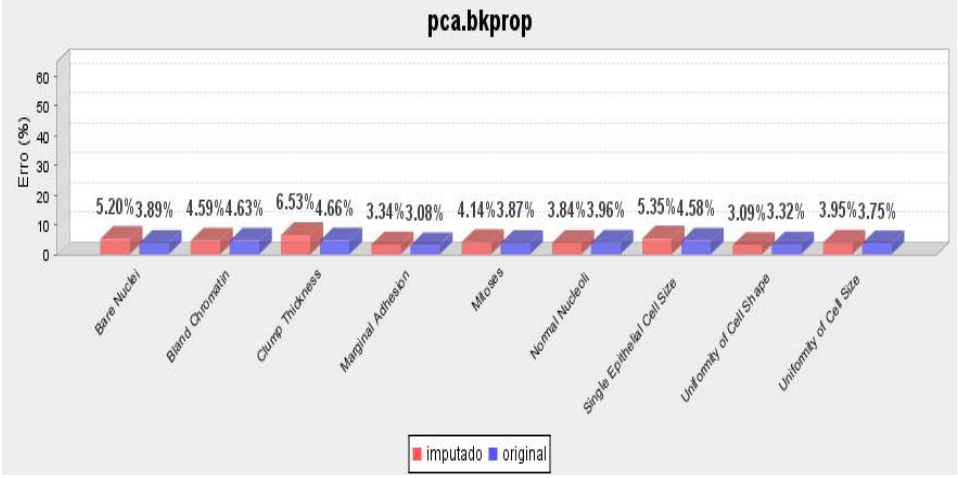
Iris Plants



Pima Indians Diabetes

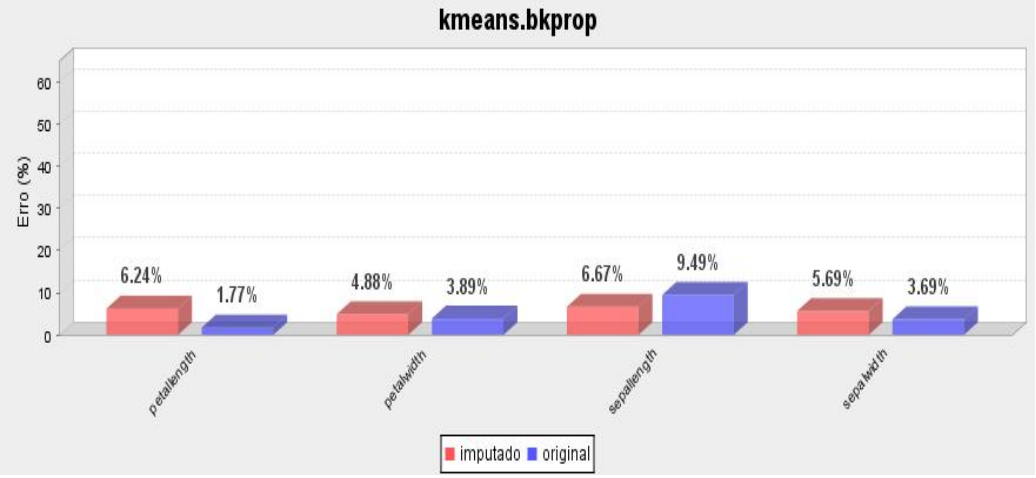


Wisconsin Breast Cancer

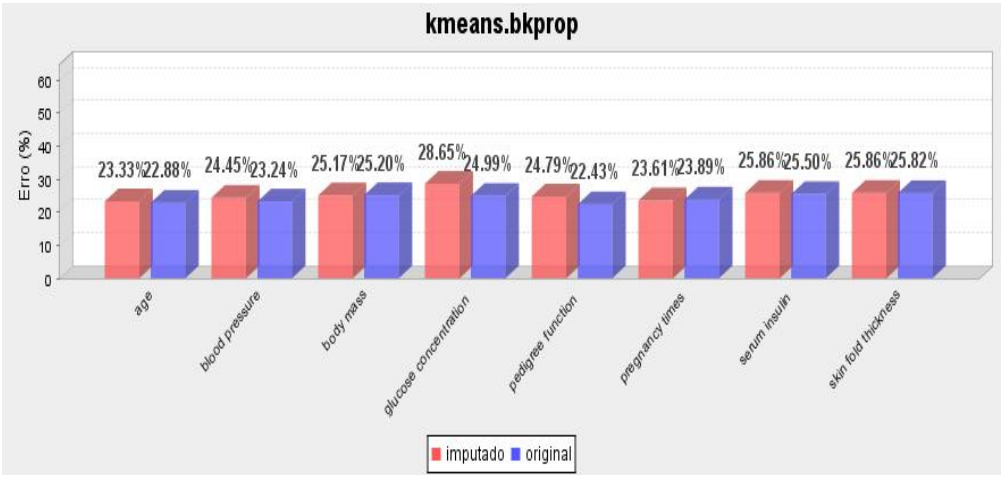


Plano 12: Agrupamento com *K*-Means e Imputação com *back propagation*

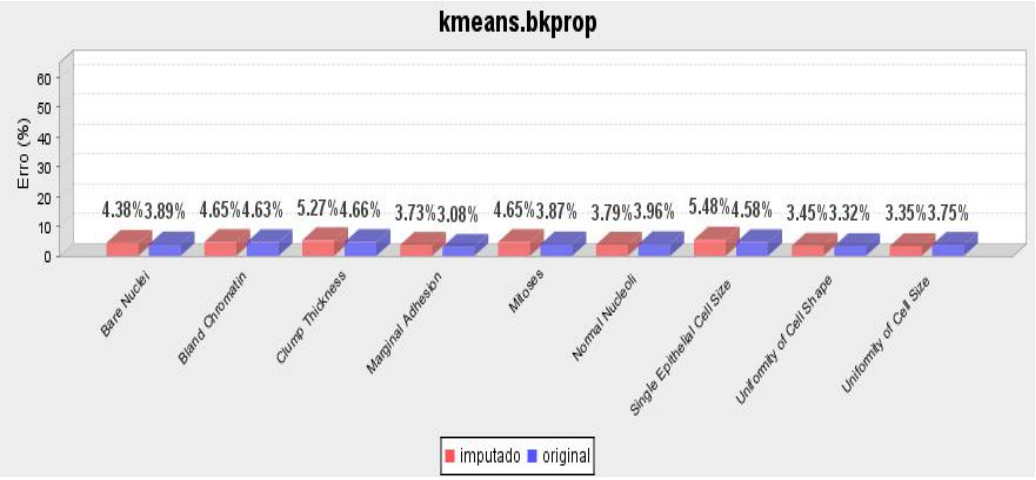
Iris Plants



Pima Indians Diabetes

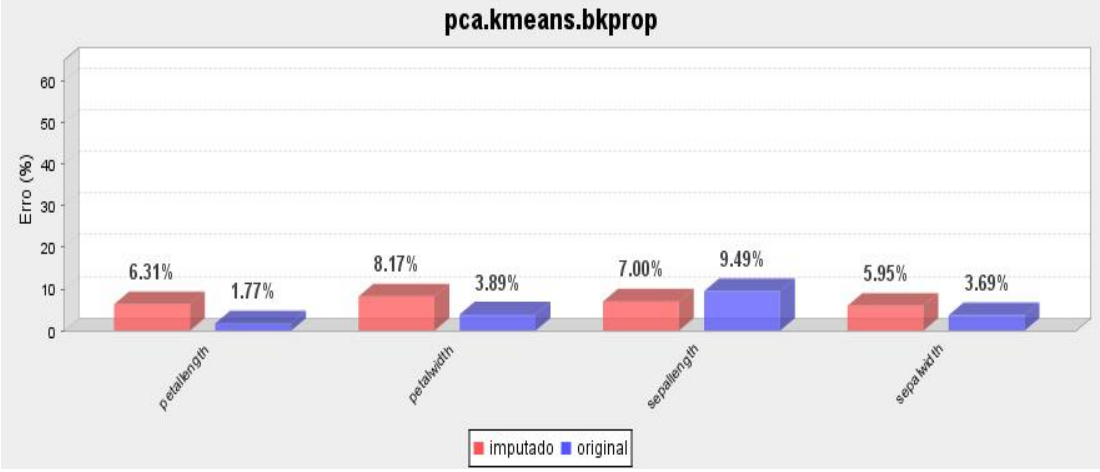


Wisconsin Breast Cancer

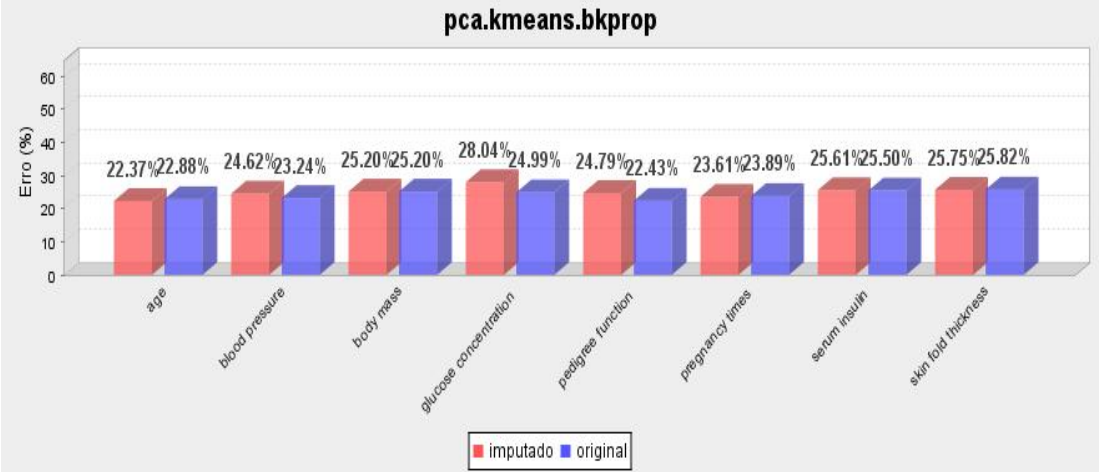


Plano 13: Seleção com PCA, Agrupamento com *K*-Means e Imputação com *back propagation*

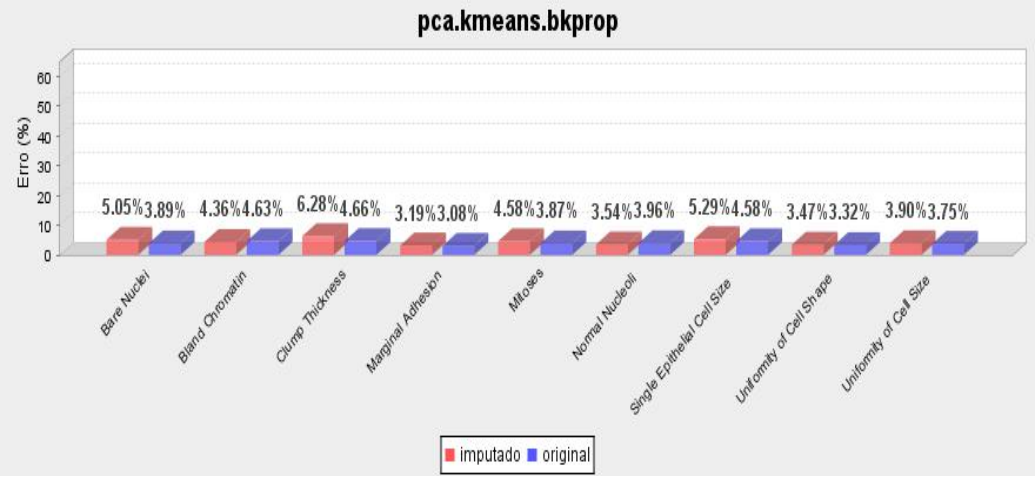
Iris Plants



Pima Indians Diabetes

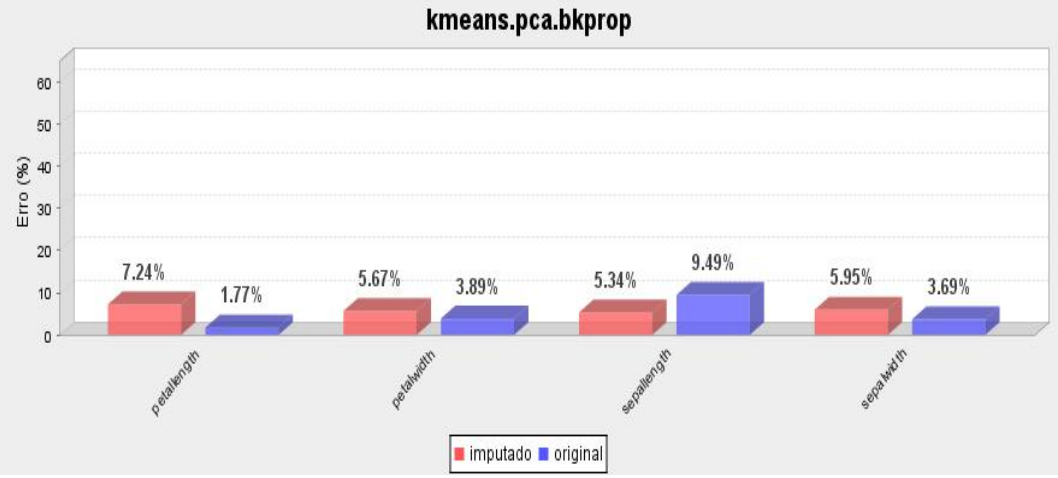


Wisconsin Breast Cancer

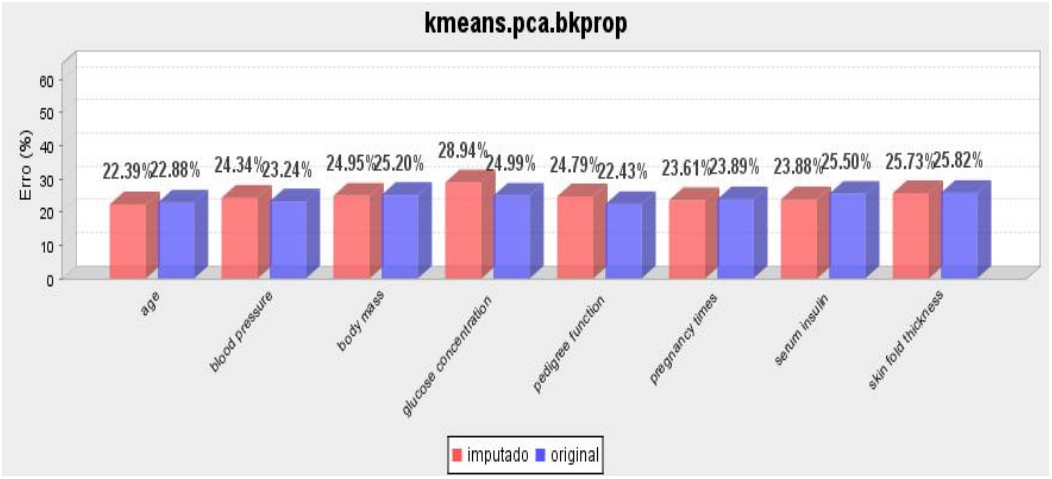


Plano 14: Agrupamento com *K*-Means, Seleção com PCA e Imputação com *back propagation*

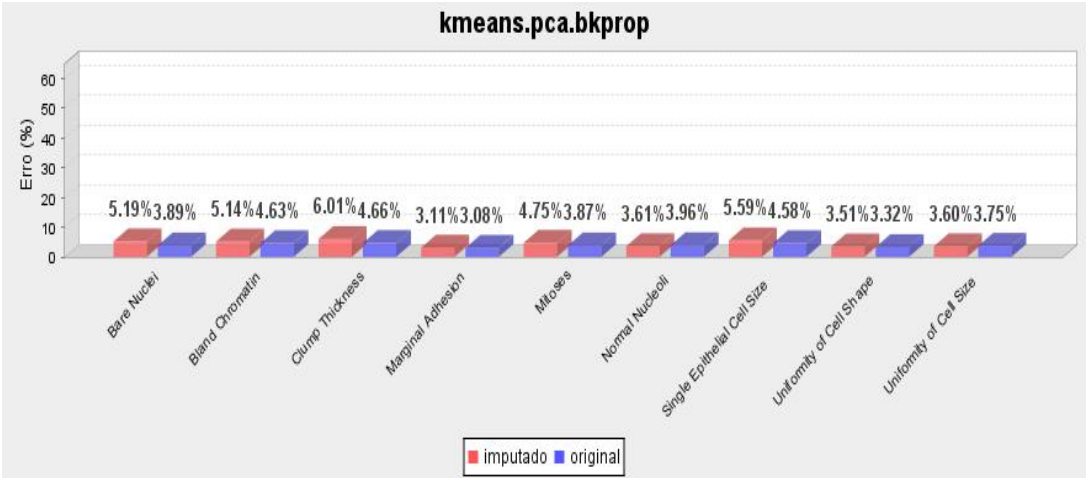
Iris Plants



Pima Indians Diabetes

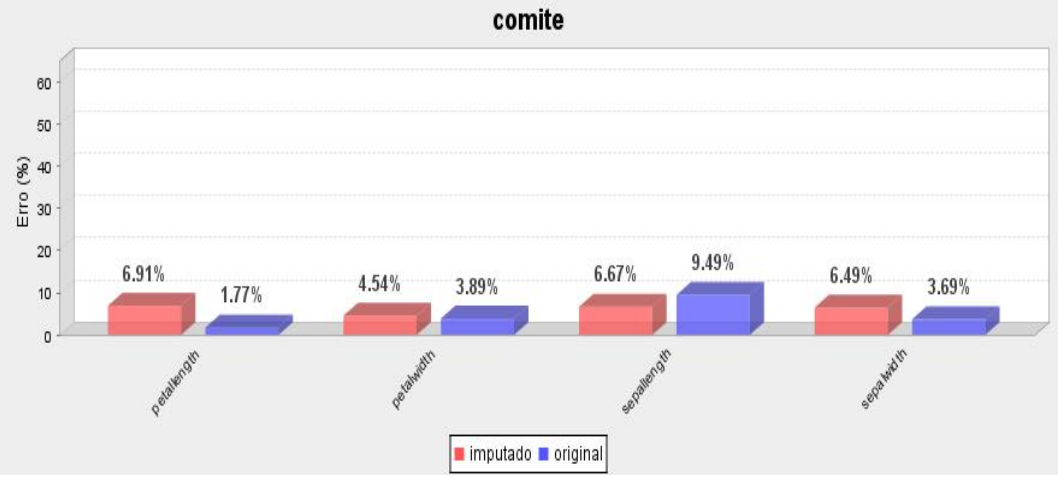


Wisconsin Breast Cancer

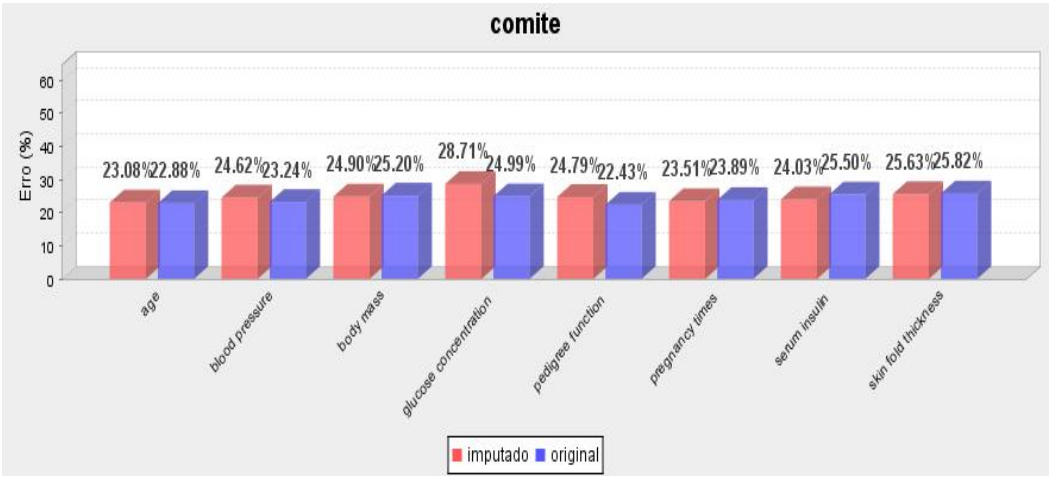


Plano 15: Comitês de Complementação de Dados

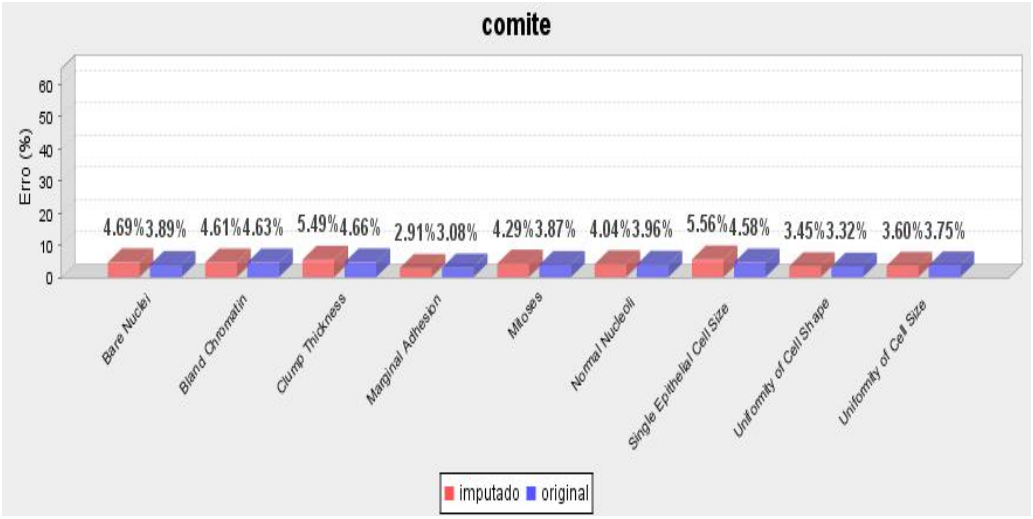
Iris Plants



Pima Indians Diabetes



Wisconsin Breast Cancer



CAPÍTULO 6

CONSIDERAÇÕES FINAIS

6.1 Resumo do Trabalho

Nesta tese, apresentamos o problema complementação de dados ausentes em conjuntos de dados, e destacamos a sua importância não só no processo de descoberta de conhecimento em bases de dados, mas também na produção de análises estatísticas que podem ser feitas sobre estes dados. Valores ausentes podem ser extremamente prejudiciais a ambos os processos, levando inclusive a conclusões e/ou padrões equivocados, situação que pode gerar problemas aqueles que venham a tomar decisões baseados nestas análises.

Após analisar as soluções existentes na literatura, pudemos observar que nenhum dos estudos realizados conseguiu apontar melhores práticas no tratamento de dados ausentes. A grande maioria dos estudos existentes no tema analisa a aplicação de um ou mais algoritmos de complementação de dados em conjuntos com características específicas, e concluem que um método apresenta um melhor desempenho do que outro. Todavia, estes trabalhos não aprofundam a análise no estudo das relações intrínsecas existentes no conjunto de dados, tanto em nível de registros quanto da correlação entre os atributos das tabelas.

No que diz respeito à relação existente entre os registros das tabelas, a técnica *hot-deck* mostra-se como uma primeira sinalização para o fato de que a divisão dos dados de uma tabela em grupos pode trazer grandes benefícios ao processo de imputação. Todavia, apesar de a proposta ter sido originalmente apresentada há mais de vinte anos, pouquíssimos trabalhos exploram esta técnica.

Já a análise do impacto da seleção dos atributos mais importantes no processo de complementação de dados não figura como objeto de pesquisa em nenhum dos trabalhos da área que tivemos acesso. Assim, propusemo-nos a analisar quais as conseqüências da seleção de atributos, baseada na sua correlação. Também nos interessou avaliar a conjunção da aplicação da seleção e agrupamento de dados em todo o processo de tratamento de dados ausentes.

Assim, propusemos um estudo baseado no conceito de estratégias de complementação de dados ausentes, onde cada estratégia refletia a aplicação seqüenciada das tarefas de seleção de atributos e agrupamento de dados, precedendo o processo de complementação de dados. Apesar de utilizarmos apenas estas duas tarefas, o processo foi formalizado para um conjunto qualquer de técnicas.

Durante o estudo das soluções disponíveis na literatura, não conseguimos identificar nenhuma taxonomia que abrigasse de forma satisfatória todas as possíveis técnicas de complementação de dados existentes. Assim, propusemos uma nova classificação de técnicas, que cria o conceito de *métodos híbridos* de complementação de dados. Esta categoria de métodos agrupa as soluções de complementação de dados que não utilizam apenas uma única técnica de imputação. Nesta classe enquadramos os métodos de imputação múltipla, já que sua definição solicita que um ou mais métodos de imputação simples sejam aplicados, e os métodos que se valem das estratégias que definimos no escopo desta tese. A estes métodos de imputação com a utilização de estratégias chamamos imputação composta e seu embasamento teórico foi abordado no capítulo 4.

Para materializar as idéias da imputação composta, construímos um sistema chamado *Appraisal*, que não só implementa a imputação com estratégias e planos de imputação, mas também avalia a qualidade do dado imputado, tanto do ponto de vista do erro do valor gerado (o quão distante ele está do valor original) quanto na preservação das características originais do conjunto de dados (a preservação da correlação existente entre os atributos da tabela).

Para isso, realizamos uma bateria de testes sobre dados com diferentes características, tanto estruturais (número de atributos e correlação entre as colunas) quanto em volume de dados. Assim, utilizamos três bases amplamente estudadas nos trabalhos relacionados ao tema: *Iris Plants*, *Pima Indians Diabetes* e *Wisconsin Breast Cancer*. Simulamos valores ausentes em seus atributos, um por vez, utilizando o mecanismo de ausência completamente aleatório (MCAR). O motivo desta escolha foi a de submeter os testes de imputação composta a condições de aleatoriedade extrema, onde pudéssemos avaliar o desempenho da aplicação de técnicas que não se beneficiassem de algum padrão específico de ausência nos dados.

Os resultados mostram que, em todas as situações, a estratégia composta de agrupamento de dados seguida da seleção dos atributos mais importantes é a que mostrou melhores resultados. A imputação com o algoritmo dos *k* vizinhos mais próximos mostrou o melhor desempenho em todas as bases, muito provavelmente por conta do princípio utilizado pelo algoritmo, de só utilizar no processo de complementação de dados as tuplas com maior grau de semelhança.

Baseados no processo de imputação múltipla, apresentamos também uma nova estrutura de imputação chamada **comitê de complementação de dados**. A essência desta proposta é a de sugerir um valor de imputação que fosse influenciado pelos resultados gerados pela execução prévia dos planos de imputação que materializam as estratégias da imputação composta. Para avaliar o seu desempenho, submetemos todos os *n* valores disponíveis ao comitê, com isso produzindo um (*n*+1)-*ésimo* valor. Os resultados obtidos mostram um comportamento irregular do comitê, muito provavelmente pela influência de planos de imputação que não tiveram bom desempenho em nenhum dos testes, tais como a média e a imputação direta com o uso de redes *back propagation*.

6.2 Contribuições da Tese

Apontamos como contribuição dos estudos realizados nesta tese os seguintes pontos:

1) Proposta de uma nova taxonomia para a classificação das técnicas de complementação de dados ausentes;

2) Definição de um novo paradigma de imputação de dados ausentes em bases de dados: a imputação composta, baseada no conceito de estratégias e planos de imputação de dados e com a utilização de técnicas do processo de Descoberta de Conhecimento em Bases de Dados (a seleção de atributos, tarefa da etapa de pré-processamento de dados, e o agrupamento, que pode ser usado tanto na etapa de pré-processamento quanto na de mineração de dados);

3) A utilização da seleção de atributos ausentes no processo de imputação;

4) A proposta de comitês de complementação de dados ausentes, que geram sugestões de imputação baseados em propostas já existentes, resultado da execução prévia de uma ou mais estratégias de complementação de dados;

5) A observação, através dos experimentos realizados nesta tese, de que as estratégias compostas produzem dados a serem imputados de melhor qualidade (com um menor índice de erros);

6) A verificação, também baseado no escopo dos testes feitos, de que o percentual de valores ausentes em um conjunto de dados não afeta a qualidade da classificação dos dados. Os resultados dos testes mostram ainda que, a menos da aplicação da imputação com a média em bases pequenas e de poucos atributos como é o caso da *Iris Plants*, a aplicação dos planos de imputação também não impactam de forma geral a qualidade da imputação.

7) A forma como o sistema de imputação composta *Appraisal* foi desenvolvido permite que outros algoritmos possam ser acoplados sem alteração da sua estrutura principal.

6.3 Trabalhos Futuros

Os estudos realizados nesta tese nos apresentaram uma série de questões ainda sem resposta, e que certamente merecem ser analisadas.

Um próximo passo natural ao estudo realizado nesta tese é avaliar todo o processo em atributos categóricos, para avaliar se os resultados aqui obtidos refletem-se também com a mudança da natureza dos atributos. A seguir, podemos consolidar a análise do processo de complementação de dados ausentes em bases mistas, com atributos numéricos e categóricos.

A experimentação de outras técnicas de seleção, agrupamento e imputação também podem e devem se seguir aos estudos realizados neste trabalho. A utilização de outras técnicas evolucionárias, por exemplo, podem revelar interessantes resultados.

No que diz respeito à seleção de atributos, é importante que estudemos a eficácia da complementação de dados relacionada a um ou mais algoritmos de complementação de dados, valendo-se, com isso, da técnica *wrapper* de seleção de dados de (FREITAS, 2002). Além disso, podemos mudar a forma como os atributos são selecionados em função da matriz de correlação dos dados da base. Podemos avaliar quais seriam os efeitos de retirarmos do processo um atributo que não tenha correlação com os demais do conjunto de dados, assim como acontece nas bases *Iris Plants* (com o atributo *sepalwidth*) e *Wisconsin Breast Cancer* (com o atributo *Mitoses*);

O efeito da imputação em cascata – a imputação de valores no atributo a_j depois de ter ocorrido imputação de outros atributos a_i, $i \neq j$ – é um interessante ponto que merece atenção. Uma possível analise que se segue a essa é a avaliação do impacto que a ordem de imputação causa na qualidade dos dados gerados (a_i antes de a_j, ou a_j antes de a_i).

Como pudemos observar no capítulo 5, a tarefa de agrupamento tem grande importância na qualidade dos dados imputados em tuplas que apresentam valores ausentes. Assim, estudos também podem ser desenvolvidos para avaliar novas formas de geração dos centróides no início do algoritmo de agrupamento dos *K* centróides ou nas métricas de similaridade, que podem utilizar outras medidas. Uma primeira medida a ser testada é a *Malahanobis*, que leva em consideração a covariância entre os atributos da tabela.

Acreditamos que os comitês de complementação de dados merecem especial atenção. Vários tipos de análises podem ser feitas com esta estrutura. Um primeiro conjunto de testes que realizaremos é a execução do comitê com subconjuntos das sugestões produzidas pelos planos de imputação, analisando como a qualidade de sua saída é alterada. Queremos avaliar se, utilizando estratégias menos dispendiosas, tais como a imputação direta com o algoritmo dos *k* vizinhos mais próximos ou uma seleção de atributos seguida de imputação com *k*-NN pode oferecer sugestões de imputação de melhor qualidade, a baixo custo computacional. A utilização de outros algoritmos de aprendizado de máquina no comitê que não as redes *back propagation* também serão objeto de análise posterior.

O conhecimento gerado pelas execuções das estratégias e planos de imputação pode servir de base a um sistema especialista que, a partir da análise de similaridade entre bases de dados e das simulações já realizadas, possa decidir qual o plano de complementação de dados ausentes mais indicado a utilizar em uma nova base de dados.

Os diversos testes realizados nesta tese duraram quase quatro meses de uso intenso do recurso computacional. Assim, um objeto quase que natural de estudo futuro é a implementação do sistema *Appraisal* em ambientes multiprocessados. As estratégias de complementação de dados podem ser implementadas com diversos níveis de granularidade em ambientes distribuídos e/ou paralelos.

A seqüência de operações definidas pelas estratégias e planos de imputação sugere uma forte interseção com sistemas baseados em fluxo (*workflow*). Assim, pretendemos avaliar o quão próximas estas duas áreas se encontram, e desenvolver o estudo promovido nesta tese apoiado nesta área de conhecimento.

Por fim, pretendemos obter resultados com bases de dados consideravelmente maiores das que usamos, já que nosso universo de dados de testes restringiu-se a bases pequenas, que normalmente cabem em memória principal. Outras questões não observadas podem ser levantadas com o uso de grandes bases de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

AGRAWAL, R., IMIELINSKI, T., SWAMI, A., 1993, "Mining Associations Between Sets of Items in Massive Databases". In: *Proceedings of the ACM SIGMOD 1993 International Conference on Management of Data*, pp. 207-216, Washington D.C., Mai.

AHA, D. W., KIBLER, D., ALBERT, M., 1991, "Instance-based Learning Algorithms", *Machine Learning*, v. 6, pp. 37-66.

ALLISON, P. D., 2000, "Multiple Imputation for Missing Data: A Cautionary Tale", *Sociological Methods & Research*, v. 28, pp. 301-309.

ALLISON, P. D., 2001, *Missing Data*. Sage Publications.

AUSTIN, P. C., ESCOBAR, M. D., 2005, "Bayesian modeling of missing data in clinical research", *Computational Statistics & Data Analysis*, v. 49, pp. 821-836.

BANSAL, A., KAUFFMAN, R. J., WEITZ, R. R., 1993, "Comparing the Modeling Performance of Regression and Neural Networks As Data Quality Varies: A Business Value Approach", *Journal of Management Information Systems*, vol. 10, n. 1, pp. 11-32.

BATISTA, G. E. A. P. A., 2003, *Pré-Processamento de Dados em Aprendizado de Máquina Supervisionado*. Tese de D. Sc., USP, São Paulo, SP, Brasil.

BATISTA, G. E. A. P. A., MONARD, M. C., 2001, "A Study of K-Nearest Neighbour as a Model-Based Method to Treat Missing Data". In: *Proceedings of the Argentine Symposium on Artificial Intelligence* (*ASAI'01*), pp. 1–9, Buenos Aires, Argentina.

BATISTA, G. E. A. P. A., MONARD, M. C., 2003a, "An Analysis of Four Missing Data Treatment Methods for Supervised Learning", *Applied Artificial Intelligence*, v. 17, n. 5 (May-Jun), pp. 519-533.

BATISTA, G. E. A. P. A., MONARD, M. C., 2003b, "Um Estudo Sobre a Efetividade do Método de Imputação Baseado no Algoritmo k-Vizinhos Mais Próximos". In: *IV Workshop on Advances & Trends in AI Problem Solving*, pp. 1–6, Chilán, Chile.

BECKER, R. A., CHAMBERS, J. M., WILKS, A. R., 1988, *The New S Language*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

BOLL, E. M., ST. CLAIR, D. C., 1995, “Analysis of rule sets generated by the CN2, ID3, and multiple convergence symbolic learning methods”. In: *Proceedings of the 1995 ACM 23rd annual conference on Computer science*, pp. 48 - 55, Tennessee, USA

BREIMAN, L., 1994, *Bagging Predictors*, Technical Report n. 421, Department of Statistics, University of California, Berkeley, California.

BREIMAN, L., FRIEDMAN, J., OLSHEN, R., STONE, C. J., 1984, *Classification and Regression Trees*, 1 ed., Chapman & Hall/CRC Publisher.

BRIN, S., MOTWANI, R., ULLMAN, J. D, TSUR, S., 1997, "Dynamic itemset counting and implication rules for market basket data". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 255-264, Tucson, Arizona, USA, Maio.

BROWN, M. L., KROS, J. F., 2003, “The impact of missing data on data mining”. In: Wang, J. (Author), *Data mining: opportunities and challenges*, 1 ed., chapter VII, pp. 174-198, IGI Publishing , Hershey, PA, USA.

BROWN, M. L., KROS, J. F., 2003a, “Data Mining and the impact of missing data”. In: Wang, J. (Author), *Industrial Management & Data Systems*, v. 103, n. 8, pp. 611-621 (11), Emerald Group Publishing Limited.

BROWN, M. L., KROS, J. F., 2003b, “The impact of missing data on data mining”. In: Wang, J. (Author), *Data mining: opportunities and challenges*, 1 ed., chapter VII, pp. 174-198, IGI Publishing , Hershey, PA, USA.

CARTWRIGHT, M. H., SHEPPERD, M. J., SONG, Q., 2003, “Dealing with missing software project data”. In: *Proceedings of the 9th International Symposium on Software Metrics*, pp. 154 – 165, Sep.

CARVALHO, L. A. V., 2001, *Datamining: A Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração*. 2 ed, São Paulo, SP, Editora Érica.

CARVALHO, L. A. V., FISZMAN, A., FERREIRA, N. C., 2001, “A Theoretical Model for Autism”, *Journal of Theoretical Medicine*, v. 25, pp. 123-140, USA.

CARVALHO, L.A.V., 1989, *Síntese de Redes Neuronais com Aplicações à Representação do Conhecimento e à Otimização*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

CHEESEMAN, P., STUTZ, J., 1996, “Bayesian classification (autoclass): Theory and results”. In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, CA.

CHEN, J., SHAO, J., 2000, “Nearest Neighbor Imputation for Survey Data”, *Journal of Official Statistics*, v. 16, n. 2, pp. 113-131.

CHIU, H. Y., SEDRANSK, J., 1986, “A Bayesian Procedure for Imputing Missing Values in Sample Surveys”; *Journal of the American Statistical Association*, v. 81, n. 395 (Set), pp. 667-676.

COOPER, G. F., HERSKOVITS, E., 1992, “A Bayesian Method for the Induction of Probabilistic Networks from Data”, *Machine Learning*, v. 9, n. 4, pp. 309-347.

CRÉMILLEUX, B., RAGEL, A., BOSSON, J. L., 1999, “An Interactive and Understandable Method to Treat Missing Values: Application to a Medical Data Set”. In: *Proceedings of the 5th International Conference on Information Systems Analysis and Synthesis (ISAS/SCI 99)*, pp. 137-144.

DASARATHY, B., 1990, *Nearest Neighbor (NN) norms: NN pattern classification techniques*, 1 ed., IEEE Computer Society Press, Los Alamitos.

DASGUPTA, S., 2002, “Performance guarantees for hierarchical clustering”. In: *Conference on Computational Learning Theory*, pp. 351–363.

DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B., 1977, “Maximum Likelihood from Incomplete Data via the EM Algorithm”, *Journal of the Royal Statistical Society*, Series B (Methodological), v. 39, n. 1, pp. 1-38.

DENG, J., 1989, “Introduction to Grey System”, *The Journal of Grey System*, v. 1, pp. 1-24.

FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996a, “From Data Mining to Knowledge Discovery in Databases”, *AI Magazine*, American Association for Artificial Intelligence, pp. 37-54.

FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996b, “The KDD Process for Extracting Useful Knowledge from Volumes of Data”, *Communications of the ACM*, v. 39, n. 11, pp. 27-34, Nov.

FLETCHER, D., GOSS, E., 1993, "Forecasting With Neural Networks: An Application Using Bankruptcy Data", *Information and Management*, v. 24, n.. 3, pp. 159-167.

FORD, B. L., 1983, "An Overview of Hot-Deck Procedures". In: Madow, W. G., Olkin, I. (auth.), Rubin, D. B. (ed.), *Incomplete Data in Sample Surveys*, 1 ed., vol. 2, Part IV, Chapter 14, pp. 185-207, Academic Press.

FREITAS, A. A., 2002, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. 1 ed. New York, Springer-Verlag Heidelberg.

FULLER, W. A., KIM, J. K., 2001, "Hot Deck Imputation for the Response Model", *Survey Methodology*, v. 31, n. 2, pp. 139-149.

GOLDSCHMIDT, R., 2003, *Assistência Inteligente à Orientação do Processo de Descoberta de Conhecimento em Bases de Dados*, Tese de D. Sc., PUC/RJ, Rio de Janeiro/RJ.

GOLDSCHMIDT, R., PASSOS, E., 2005, *Data Mining: Um Guia Prático*. 1 ed, Editora Campus.

GOWDA, K. C., DIDAY, E., 1992, "Symbolic Clustering Using a New Dissimilarity Measure", *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, pp. 368-378.

GRAHAM, J. W., CUMSILLE, P. E., ELEK-FISK, E., 2002, "Methods for handling missing data". In: Schinka, J. A., Velicer, W. F. (eds.). *Research Methods in Psychology*, v. 2 of Handbook of Psychology (I. B. Weiner, Editor-in-Chief), pp. 87-114, New York, John Wiley & Sons.

GRAHAM, J. W., DONALDSON, S. I., 1993, "Evaluating interventions with differential attrition: the importance of nonresponse mechanisms and use of follow-up data", *Journal of Applied Psychology*, v. 78, n. 1, pp. 119-128.

HAIR, J. F., ANDERSON, R. E., TATHAM, R. L., BLACK, W. C., 2005, *Análise Multivariada de Dados*. 5 ed. Porto Alegre/RS, Editora Bookman.

HAN, J., KAMBER, M., 2005, *Data Mining: Concepts and Techniques*. 2 ed. New York, Morgan Kaufmann Publishers.

HAND, D., MANNILA, H., SMYTH, P., 2001, *Principles of Data Mining*. 1 ed. Massachusetts, The MIT Press.

HAYKIN, S., 1999, *Redes Neurais: Princípios e prática*. 2 ed. Porto Alegre, RS, Editora Bookman.

HECHT-NIELSEN, R., 1988, "Applications of Counterpropagation Networks", *Neural Networks*, v. 1, pp. 131-139.

HOLT, J. D., CHUNG, S. M., 1999, "Efficient Mining of Association Rules in Text Databases". In: *Eight International Conference on Information and Knowledge Management (CIKM'99)*, Kansas City, USA, pp. 234-242, Nov.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2002b, "Missing values prediction with K2", *Intelligent Data Analysis Journal*, v. 6, n. 6, pp. 557-566.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2003a, "A Nearest-Neighbor Method as a Data Preparation Tool for a Clustering Genetic Algorithm". In: *Anais do 18º Simpósio Brasileiro de Banco de Dados (SBBD)*, pp. 319-327, Manaus, AM, Out.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2003b, "Evaluating a Nearest-Neighbor Method to Substitute Continuous Missing Values". In: *The 16th Australian Joint Conference on Artificial Intelligence - AI´03*, 2003, Perth. Lecture Notes in Artificial Intelligence (LNAI 2903), v. 2903, pp. 723-734. Heidelberg, Springer-Verlag.

HRUSCHKA, E. R., HRUSCHKA JR., E. R., EBECKEN, N. F. F., 2005, "Missing Values Imputation for a Clustering Genetic Algorithm". In: *First International Conference on Natural Computation (ICNC´05)*, Changsha. Lecture Notes in Computer Science 3612 (Advances in Natural Computation). Berlin, Springer-Verlag Berlin Heidelberg, v. 3612, pp. 245-254.

HUANG, C., LEE, H., 2004, "A Grey-Based Nearest Neighbor Approach for Missing Attribute Value Prediction", *Applied Intelligence*, v. 20, n. 3, pp. 239-252, May.

HUANG, Z., 1997, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining". In: *Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*. Fonte: http://citeseer.ist.psu.edu/article/huang97fast.html.

HUISMAN, M., 2000, "Imputation of Missing Item Responses: Some Simple Techniques", *Quality and Quantity*, v. 34, n. 4, pp. 331-351, Nov.

JAIN, A. K., DUBES, R. C., 1988, *Algorithms for Clustering Data*. 1 ed. New Jersey, Prentice Hall College Division.

JAIN, A. K., MURTY, M. N., FLYNN, P. J., 1999, "Data Clustering: A Review", *ACM Computing Surveys*, v. 31, n. 3 (Set), pp. 264-323.

JONES, M. P., 1996, "Indicator and stratification methods for missing explanatory variables in multiple linear regression", *Journal of the American Statistical Association*, v. 91, n. 433, pp. 222-230, Mar.

JÖNSSON, P., WOHLIN, C., 2004, "An Evaluation of k-Nearest Neighbour Imputation Using Likert Data". In: *Proceedings of the 10th IEEE International Symposium on Software Metrics* (*METRICS'04*), pp. 108-118, Chicago, USA, Sep.

JÖNSSON, P., WOHLIN, C., 2006, "Benchmarking k-nearest neighbour imputation with homogeneous Likert data", *Empirical Software Engineering*, v. 11, pp. 463-489.

KOHONEN, T., 1974, "An adaptive Associative Memory PrincipIe", *IEEE Transactions*, v. C-23, pp. 444-445.

KOHONEN, T., 1977, *Associative Memory: A System Theoretical Approach*, Springer, New York.

KOHONEN, T., 1984, *Self-organization and Associative Memory*, Springer-Verlag, Berlin.

LAKSHMINARAYAN, K., HARP, S. A., SAMAD, T., 1999, "Imputation of Missing Data in Industrial Databases", *Applied Intelligence*, v. 11, n. 3 (Nov), pp. 259-275.

LITTLE, R. J. A., RUBIN, D. B., 1987, *Statistical Analysis with Missing Data*. John Wiley and Sons, New York.

LIU, H., HUSSAIN, F., TAN, C. L., DASH, M., 2002, "Discretization: An Enabling Technique", *Data Mining and Knowledge Discovery*, v. 6, pp. 393, Kluwer Academic Publishers.

MAGNANI, M., 2004, "Techniques for Dealing with Missing Data in Knowledge Discovery Tasks". Obtido em http://magnanim.web.cs.unibo.it/index.html em 15/01/2007.

MAGNANI, M., MONTESI, D., 2004, "A new reparation method for incomplete data in the context of supervised learning". In: *Proceedings of the International Conference*

on Information Technology: Coding and Computing (ITCC´04), pp. 471-475, Las Vegas, Nevada, Abr.

MALETIC, J., MARCUS, A., 2000, "Data Cleansing: Beyond Integrity Analysis". In: *Proceedings of the Conference on Information Quality*, pp. 200–209, Boston, MA, Oct.

MANGASARIAN, O. L., WOLBERG, W. H., 1990, "Cancer diagnosis via linear programming", *SIAM News*, v. 23, n. 5, 1990, pp. 1-18, Sep.

MARRONE, P., 2007, *Java Object Oriented Neural Engine: The Complete Guide*. Obtido em http://ufpr.dl.sourceforge.net/sourceforge/joone/JooneCompleteGuide.pdf.

McCULLOCH, W. S., PITTS, W., 1943, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, v.5, pp.115-133.

MCQUEEN, J., 1967, "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

MEDSKER, L., LIEBOWITZ, J., 1994, *Design and Development of Expert Systems and Neural Networks*, Ed. Macmillan, New York

MINSKY, M. L., PAPERT, S., 1969, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, Massachusetts.

MITCHELL, T. M., 1997, *Machine Learning*. Ed. McGraw-Hill.

MOTRO, A., 1995, "Management of Uncertainty in Database Systems". In: Kim, W. (ed), *Modern Database Systems: The Object Model, Interoperability, and Beyond*, 1 ed, chapter 22, New York, ACM Press.

MYRTVEIT, I., STENSRUD, E., OLSSON, U. H., 2001, "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", *IEEE Transactions on Software Engineering*, v. 27, n. 11, Nov.

NEWMAN, D. J., HETTICH, S., BLAKE, C. L., MERZ, C. J., 1998, *UCI Repository of Machine Learning Databases*. Obtido em 12/10/2006 em http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

ONISKO, A., DRUZDZEL, M. J., WASYLUK, H., 2002, "An experimental comparison of methods for handling incomplete data in learning parameters of Bayesian

networks". In: *Proceedings of the Intelligent Information Systems 2002 Symposium*, pp. 351-360, Heidelberg.

PARSONS, S., 1996, "Current approaches to handling imperfect information in data and knowledge bases", *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 3 (Jun), pp. 353-372.

PATUWO, E., HU, M. Y., HUNG, M. S., 1993, "Two-Group Classification Using Neural Networks", *Decision Sciences*, v. 24, n. 4, pp. 825-845.

PEARL, J., 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1 ed., Morgan Kaufmann Publishers.

PYLE, D., 1999, *Data Preparation for Data Mining*. 1 ed, San Francisco/CA, Morgan Kaufmann Publishers.

QUINLAN, J. R., 1993, *C4.5: Programs for Machine Learning*. Ed. Morgan Kaufmann, San Francisco.

RAGEL, A., CRÉMILLEUX, B., 1998, "Treatment of Missing Values for Association Rules", In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 258-270, Melbourne, Australia, Apr.

RAGEL, A., CRÉMILLEUX, B., 1999, "MVC – A Preprocessing Method to Deal With Missing Values", *Knowledge-Based Systems*, v. 12, n. 5–6 (Oct), pp. 285-291.

RAGHUNATHAN, T. E., 2004, "What Do We Do with Missing Data? Some Options for Analysis of Incomplete Data", *Annual Review of Public Health*, v. 25, pp. 99-117, Apr.

RAHM, E., DO, H. H., 2000, "Data Cleaning: Problems and Current Approaches", *IEEE Bulletin of the Technical Comittee on Data Engineering*, v. 23, n. 4 (Dec).

RAMONI, M., SEBASTIANI, P., 1999, *An Introduction to Bayesian Robust Classifier*, KMI Technical Report KMI-TR-79, Knowledge. Media Institute, The Open University.

ROSENBLATT, F., 1962, *Principles of Neurodynamics*, Spartan Books, New York.

RUBIN, D. B., 1988, "An Overview of Multiple Imputation", In: *Proceedings of the Section on Survey Research Methods*, pp. 79-84, American Statistical Association.

RUMELHART, D.E., HINTON, G.E., WILLIAMS, R. J., 1986, "Learning Internal Representations by Error Back-propagation", In: Rumelhart D.E., McClelland J.L. (eds.), *Parallel Distributed Processing*, vol. 1, chapter 8, Editora MIT Press.

SCHAFER, J. L., GRAHAM, J. W., 2002, "Missing Data: Our View of the State of the Art", *Psycological Methods*, v. 7, n. 2, pp. 147-177.

SCHAPIRE, R. E., FREUND, Y., BARTLETT, P., LEE, W. S., 1998, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods", *The Annals of Statistics*, v. 26, n. 5, pp. 1651-1686.

SCHEFFER, J., 2002, "Dealing with Missing Data", *Research Letters in the Information and Mathematical Sciences*, v. 3, n. 1 (Apr), pp. 153-160.

SHLENS, J., 2005, "A Tutorial on Principal Component Analysis". Obtido em http://www.cs.cmu.edu/~elaw/papers/pca.pdf em 14/04/2007.

SILVA, E. B., 2006, *Agrupamento Semi-Supervisionado de Documentos XML*. Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

SMITH, L. I., 2002, "A Tutorial on Principal Component Analysis". Obtido em http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf em 14/04/2007.

SONG, Q., SHEPPERD, M., CARTWRIGHT, M., 2005, "A Short Note on Safest Default Missingness Mechanism Assumptions", *Empirical Software Engineering*, v. 10, n. 2, pp. 235-243, Apr.

SRIKANT, R., AGRAWAL, R., 1997, "Mining generalized association rules", *Future Generation Computer Systems*, v. 13, n. 2-3, pp. 161-180.

STRIKE, K., EL EMAM, K., MADHAVJI, N., 2001, "Software Cost Estimation with Incomplete Data", *IEEE Transactions on Software Engineering*, v. 27, pp. 890-908.

TEKNOMO, K., 2007, *K-Means Clustering Tutorials*. Obtido em http://people.revoledu.com/kardi/tutorial/kMean em 19/04/2007.

TEKNOMO, K., 2007, *Mean and Average*. Obtido em http://people.revoledu.com/kardi/tutorial/BasicMath/Average em 27/04/2007.

TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., BROWN, P., HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D., ALTMAN, R., 2001, "Missing value estimation methods for DNA microarrays", *Bioinformatics*, v. 17, n. 0, pp. 1-6.

TSENG, S., WANG, K., LEE, C., 2003, "A Preprocessing Method to Deal with Missing Values by Integrating Clustering and Regression Techniques", *Applied Artificial Intelligence*, v. 17, n. 5 (May-Jun), pp. 535-544.

TWALA, B., CARTWRIGHT, M., SHEPPERD, M., 2005, "Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases". In: *2005 International Symposium on Empirical Software Engineering*, pp. 105-114, Nov.

VACH, W., 1995, "Logistic Regression with Missing Values in the Covariates", *Technometrics*, v. 37, n. 4, pp. 460-461, Nov.

WALCZAK, S., CERPA, N., 1999, "Heuristic Principles for the Design of Artificial Neural Networks", *Information and Software Technology*, v. 41, n. 2, pp. 109-119.

WAYMAN, J. C., 2003, "Multiple Imputation For Missing Data: What Is It And How Can I Use It?". In: *Proceedings of the Annual Meeting of the American Educational Research Association*, Chicago, IL, Apr.

WEKA, 2007, *Weka 3: Data Mining Software in Java*. Disponível em http://www.cs.waikato.ac.nz/ml/weka/ em 30/04/2007.

WILLIAMS, R. J., 1986, "The Logic of Activation Functions", *Parallel Distributed Processing*, Eds. Rume1hart, D. E., McClelland, J. L., The MIT Press, Cambridge, Massachussetts.

WITTEN, I. H., FRANK, E., 2005, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. 2 ed. San Francisco/CA, Morgan Kaufmann Publishers.

WOLPERT, D. H., 1996, "The lack of a priori distinctions between learning algorithms". *Neural Computation*, v. 8, pp. 1341-1390.

YUAN, Y. C., 2000, "Multiple Imputation for Missing Data: Concepts and New Development". In: *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, Paper No. 267, Cary, NC: SAS Institute.

ZAKI, M. J., PARTHASARATHY, S., OGIHARA, M., LI, W., 1997, “New Algorithms for Fast Discovery of Association Rules” In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 283-286, 1997.

ZHANG, S., ZHANG, C., YANG, Q., 2003, “Data Preparation for Data Mining”, *Applied Artificial Intelligence*, v. 17, n. 5-6 (May-Jun), pp. 375-381.

ZHENG, Z., LOW, B. T., 1999, *Classifying Unseen Cases with Many Missing Values*. Technical Report (TR C99/02), School of Computing and Mathematics, Deakin University, Australia.

ZHENG, Z., WEBB, G. I., 1998, *Sthocastic Attribute Selection Committees with Multiple Boosting: Learning More Accurate and More Stable Classifier Committees*, Technical Report (TR C98/13), School of Computing and Mathematics, Deakin University, Australia.