CENTRO UNIVERSITÁRIO DA CIDADE DO RIO DE JANEIRO
ESCOLA DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
NÚCLEO DE PROJETOS E PESQUISAS EM APLICAÇÕES COMPUTACIONAIS

IMPUTAÇÃO COMPOSTA CATEGÓRICA

EM BASES DE DADOS

Diego Saraiva Monteiro

RIO DE JANEIRO, RJ
Julho de 2008

IMPUTAÇÃO COMPOSTA CATEGÓRICA

EM BASES DE DADOS

Projeto de Pesquisa apresentado como parte dos pré-requisitos para obtenção do título de Bacharel em Ciência da Computação do Centro Universitário da cidade do Rio de Janeiro - UniverCidade.

Professor Orientador: Jorge de Abreu Soares

Composição da Banca Examinadora:

Prof. Jorge de Abreu Soares, D. Sc.
Centro Universitário da Cidade do Rio de Janeiro

Prof. Ronaldo Ribeiro Goldschmidt, D. Sc.
Centro Universitário da Cidade do Rio de Janeiro

Prof.ª Cláudia Ferlin, M. Sc.
Centro Universitário da Cidade do Rio de Janeiro

Prof. Rafael Castaneda Ribeiro, B. Sc.
CEFET/RJ

RIO DE JANEIRO, RJ
Julho de 2008

Resumo do projeto apresentado ao Centro Universitário da Cidade do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação

IMPUTAÇÃO COMPOSTA CATEGÓRICA

EM BASES DE DADOS

Diego Saraiva Monteiro

Julho / 2008

Orientador: Prof. Jorge de Abreu Soares

Devido ao avanço da área de TI, e um aumento na capacidade de captação de dados, o tamanho das bases de dados vem aumentando de forma significativa nos últimos anos. Como conseqüência desse aumento, surge a necessidade de se analisar dados em grande volume, de forma que uma grande importância tem sido dada à complementação de valores ausentes em bases de dados, já que a consistência desses dados se torna essencial. O principal objetivo deste projeto é apresentar a pesquisa, formalização, teste e avaliação de mecanismos voltados à imputação composta de valores ausentes de natureza categórica em bases de dados.

Abstract of Dissertation presented to Centro Universitário da Cidade do Rio de Janeiro as a partial fulfillment of the requirements for the degree of Bachelor in Computer Science

CATEGORICAL COMPOSED IMPUTATION OF MISSING VALUES ON DATASETS

Diego Saraiva Monteiro

July / 2008

Advisor: Prof. Jorge de Abreu Soares

Due to the IT area advance and a raise on data gathering capacity the size of databases has increasead in a significant way in the last years. As consequence of this raise, the necessity of data analysis in a huge volume appears. With this, a great importance has been given the imputation of missing values in databases, since the consistency of these data becomes essential. The main goal of this project is to present a research, a formalization, test and the evaluation of mechanisms for composed imputation of categorical missing values in databases.

Dedico este projeto aos meus pais, a quem devo, além de todas as minhas vitórias, toda a fé depositada em meu futuro e a compreensão infinita em todos os momentos de minha vida.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus, que me permitiu estar concluindo mais esta etapa em minha vida.

Ao Prof. Jorge de Abreu Soares, D. Sc., orientando e ajudando sempre da melhor maneira possível, o que culminou no meu sucesso na conclusão deste projeto.

Ao Prof. Ronaldo Ribeiro Goldschmidt, D. Sc., orientando nas etapas iniciais e de não menos importância no meu sucesso na conclusão deste projeto.

Ao Prof. Rafael Castaneda Ribeiro, B. Sc., pelo suporte técnico na construção da ferramenta desenvolvida na linguagem *Java* neste projeto.

À Prof.ª Cláudia Ferlin, pela paciência e por estar sempre presente durante todo o curso de Ciência da Computação.

LISTA DE TABELAS

LISTA DE FIGURAS

ÍNDICE

CAPÍTULO 1

INTRODUÇÃO

1.1 Posicionamento e Justificativa

Nos dias atuais, vivemos em um cenário onde o processamento de dados em aplicações de grande porte e de grandes corporações gera ordens de terabytes e petabytes de dados. A evolução tecnológica dos computadores atuais permite que as bases de dados cresçam cada vez mais. Além disso, os preços dos dispositivos de armazenamento ficam casa vez mais atraentes, uma vez que estão sempre evoluindo rapidamente, aumento cada vez mais em capacidade de armazenamento. Hoje em dia vivemos uma época que não imaginávamos há alguns anos atrás, tanto em soluções domésticas como em soluções corporativas de hardware de armazenamento (SOARES, 2007).

Com o grande volume de dados gerado pelas corporações, surge também a necessidade de interpretá-los, já que esses dados podem conter (e provavelmente contêm) 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 (MITCHELL, 1977).

Para que essas bases de dados possam ser interpretadas corretamente, precisamos garantir que seus dados estejam completos. Hoje em dia, áreas como a Estatística, *Data Warehouse* e *Mineração de Dados* utilizam a Imputação para eliminar os valores ausentes e garantir que seus registros fiquem completos, o que garante a qualidade do resultado final no processo de descobrimento de conhecimento em bases de dados.

1.2 Motivação

Como tratada na seção 1.1, a manipulação de grande volume de dados gera a ocorrência de alguns problemas de consistência de dados. As causas desses problemas podem decorrer, por exemplo, de um processo de integração de dados mal planejado ou mal executado, um processo de integração onde as incompatibilidades dos sistemas gerem inconsistências nos dados, do mau funcionamento de equipamentos de aferição de medidas, recusa de entrevistados de responder a certas questões, entre outros (BATISTA, MONARD, 2003). Este tipo de problema pode comprometer todo o resultado final de descoberta de conhecimento dos dados analisados, e por isso requer uma grande atenção dos analistas responsáveis por essa tarefa.

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 à amostra original. Valores que estavam presentes e foram apagados por conta da redução horizontal ou vertical causam perda significativa de informação (SOARES, 2007).

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. SOARES (2007) avalia especificamente o problema de dados numéricos ausentes em tabelas de bases de dados sob a ótica de diferentes estratégias de complementação de dados.

Com isso, neste projeto analisamos o problema de complementar dados ausentes de natureza categórica através de diferentes estratégias de complementação, um problema

ainda não encontrado na literatura atual. Interessa-nos em especial as soluções que envolvam a aplicação de algoritmos de aprendizado de máquina na solução do problema de complementação de dados categóricos, e este é então nosso objeto de estudo neste projeto.

1.3 Objetivos do Trabalho

O objetivo deste trabalho é o de avaliar os resultados da aplicação de diversas estratégias de complementação de dados de natureza categórica em bases de dados que possuem valores ausentes em suas tuplas. A complementação de dados, mais conhecida como imputação (FORD, 1983, SOARES, 2007, BATISTA, MONARD, 2003), é um problema de eminente importância em uma das etapas de um processo de descobrimento, a etapa de pré-processamento de dados, pois todo o processo pode ser severamente afetado caso os dados não tenham recebido, um tratamento cuidadoso no que diz respeito à complementação dos dados ausentes (JAIN, 1999).

Para isso, utilizaremos uma variação do processo de imputação, proposto por SOARES (2007), precedido por tarefas usualmente aplicadas no processo de mineração de dados. Este processo é chamado de **imputação composta**.

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 categóricos. Queremos analisar o impacto da aplicação das seguintes configurações de estratégias:

1) Imputação;
2) Agrupamento e Imputação;
3) Seleção, Agrupamento e Imputação;
4) Agrupamento, Seleção e Imputação;

Sendo assim, geramos resultados 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.

1.4 Organização do Texto

Este documento possui mais quatro capítulos. O capítulo 2 descreve a fundamentação teórica deste trabalho, detalhando o processo de imputação e a solução proposta por SOARES (2007), que será implementada neste projeto. Com isso, explanamos os conceitos relacionados, e revisamos algumas soluções disponíveis na literatura com soluções para este problema.

No capítulo 3, é descrita a solução que foi proposta e implementada neste projeto para tratar o problema de imputação composta categórica em bases de dados. Neste capítulo descrevemos como conseguimos simular ambientes com valores ausentes.

O capítulo 4 detalha os testes realizados em três bases de dados normalmente utilizadas como benchmarks de mineração de dados: *Car Evaluation*, *Tic-Tac-Toe Endgame* e *Teaching Assistant Evaluation*, todas do repositório da Universidade da Califórnia, Irvine.

No capítulo 5 estão expostas as considerações finais, indicando possíveis trabalhos futuros para este projeto.

CAPÍTULO 2

IMPUTAÇÃO

2.1 Introdução

Uma das tarefas mais importantes do processo de extração de conhecimento de bases de dados, a tarefa de 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, 2003). Seu uso é bastante adequado para grandes bases de dados (CARTWRIGHT *et al*, 2003). A imputação pode ser feita de forma *determinística* ou *estocástica* (MAGNANI, 2004).

A ausência de dados pode ser conseqüência de vários fatores como o mau funcionamento do equipamento, a não entrada de dados, a ocorrência de inconsistência com outros dados registrados e assim o dado torna-se ausente, algumas vezes certos dados não são considerados importantes, enganos na entrada de dados e a indisponibilidade ou a inexistência dos mesmos (GOLDSCHMIDT, PASSOS, 2005).

MAGNANI (2004) propõe algumas soluções para o tratamento de valores ausentes através da tarefa de imputação, como por exemplo:

1. *Imputação Global Baseada no Atributo com Valores Ausentes*

   A *imputação global baseada no atributo com valores ausentes* utiliza valores existentes nas demais tuplas para preencher os que são desconhecidos. Eles podem ser de dois tipos, *Determinísticos,* os mais comuns são a média ou a moda e E*stocásticos*: utilizando a introdução de uma perturbação na média.

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.

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. Isto faz com que seja reduzido o desvio (*bias*), classificando a amostra. A técnica *cold-deck* diferencia-se da técnica *hot-deck*, pois aquela utiliza dados de outra fonte, que não os dados correntes.

2.2 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).

Os dados ausentes de uma tabela de um conjunto de dados são ditos **completamente aleatórios** (MCAR –*Missing Completely At Random*) quando o motivo de sua ausência é desconhecido, ou seja, não sabemos precisar as razões pelas quais algumas medidas do atributo não tiveram os seus valores registrados (SOARES, 2007).

Entretanto, se alguns valores de um atributo se tornarem ausentes em função de alguma condição de outro atributo, dizemos que estes são **valores ausentes aleatórios** (MAR – *Missing At Random*). Por exemplo, um dos aparelhos que realiza medições das

condições dos pneus de um carro não conseguir realizar medidas quando os valores medidos para o volume de óleo de freio for menor que 1.0 litro (SOARES, 2007).

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 medições de volume de óleo de freio de um carro 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*) (SOARES, 2007).

2.3 Áreas de Aplicação da Imputação

2.3.1 Data Mining

Uma área amplamente estudada na comunidade científica de Computação, a área de Descoberta de Conhecimento em Bases de Dados (*KDD – Knowledge Discovery in Databases*), também conhecida na literatura como Mineração de Dados (*Data Mining*), busca descobrir que tipos de relações intrínsecas podem existir em um conjunto de dados (SOARES, 2007).

Esse processo 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. A figura 2.1 mostra as três etapas de um processo de *KDD* (GOLDSCHMIDT, PASSOS, 2005).

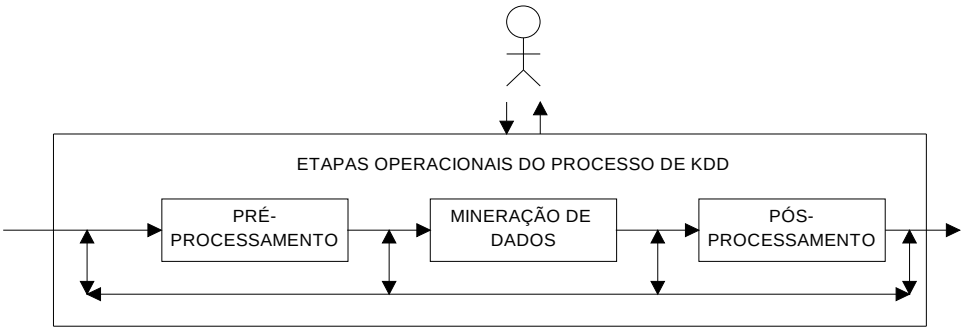

Figura 2.1 - Etapas operacionais do processo de KDD *(GOLDSCHMIDT, PASSOS, 2005)*

As etapas de um processo de *KDD* são descritas abaixo:

1. A etapa de Pré-Processamento - De uma forma geral, esta etapa consiste em preparar os dados para a etapa de Mineração de Dados. Entre as principais atividades de pré-processamento, podemos citar, de uma forma geral (GOLDSCHMIDT, PASSOS, 2005):
    a. Seleção de Dados
    b. Limpeza de Dados
    c. Codificação de Dados
    d. Enriquecimento dos Dados
2. A etapa de Mineração de Dados - Durante a etapa de Mineração de Dados é realizada a busca efetiva por conhecimentos úteis no contexto da aplicação de *KDD*. Envolve a aplicação de algoritmos sobre os dados em busca de conhecimento implícitos e úteis (GOLDSCHMIDT, PASSOS, 2005).
3. A etapa de Pós-Processamento - Essa fase envolve a visualização, a análise e a interpretação do modelo de conhecimento gerado na etapa de Mineração de Dados. Em geral, é nesta etapa que o especialista em *KDD* e o especialista no domínio da aplicação avaliam os resultados obtidos e definem novas alternativas de investigação dos dados (GOLDSCHMIDT, PASSOS, 2005).

A aplicação da imputação de valores ausentes durante a atividade de Limpeza de Dados da etapa de Pré-Processamento, mostra-se como muito importante, pois os problemas causados por ausência de dados podem comprometer o resultado final do processo de *KDD* (SOARES, 2007).

2.3.2 Data Warehouse

O processo de *KDD* pode iniciar-se com a construção de um *Data Warehouse* (*DW*). Este é um meio efetivo de organizar grandes volumes de dados para sistemas de suporte a decisão e aplicações de *KDD*. Pode-se definir um *DW* como um repositório integrado, orientado para análise, histórico, com dados apenas para leitura, designado para ser utilizado como base para suporte à decisão e sistemas *KDD* (INMON, 1993, POE, 1996). Um *DW* funciona como uma base de dados para dar suporte à decisão mantido separadamente das bases de dados operacionais da organização. Geralmente integra dados de diversas origens heterogêneas e por isso necessita de uma estrutura flexível que suporte consultas em tempo real e geração de relatórios analíticos. Um ponto crítico em um *DW* que é a integração de múltiplos dados, provenientes de bases de dados heterogêneas. A integração envolve padronizar atributos, formatos e convenções de nomes, além de remoção de inconsistências.

Com essa abordagem, verificamos que a imputação é uma importante tarefa que pode ser utilizada para a remoção de inconsistências em um *Data Warehouse*. Assim como no caso do *Data Mining*, a inconsistência dos dados pode comprometer o resultado final do processo de descoberta de conhecimento (INMON, 1993, POE, 1996).

2.3.3 Estatística

O que modernamente se conhece como Ciências Estatísticas, ou simplesmente Estatística, é um conjunto de técnicas e métodos de pesquisa que entre outros tópicos envolve o planejamento do experimento a ser realizado, a coleta qualificada dos dados, a inferência, o processamento, a análise e a disseminação das informações (LITTLE, RUBIN, 1987).

A imputação de valores ausentes é importante em todas as análises e é crítica em algumas, tal como a análise de séries cronológicas. A presença de valores ausentes distorcerá a análise estatística final, prejudicando o experimento. A imputação dos valores onde faltam dados é uma das áreas estatísticas que se desenvolve muito desde os anos 80 ALLISON (2005).

2.4 Trabalhos relacionados

Nas pesquisas realizadas, encontramos algumas soluções na literatura para o problema de imputação para dados de natureza categórica.

No primeiro trabalho que encontramos, HE (2006) sugere a utilização de métodos de agrupamento precedendo a imputação com moda que estendem os conceitos apresentados pelo algoritmo *K-Means,* chamados de *K-Modes* e *K-Median*. A diferença entre esses dois métodos é a forma como a distância entre os valores ausentes é calculada. No caso do algoritmo *K-Modes*, ele calcula a distância testando se os valores são iguais ou diferentes, já no *K-Median*, ele utiliza uma categorização *categórica-numérica* e utiliza a distância Euclidiana como métrica.

HE (2006) não se aprofunda muito em seus estudos, mas ressalta em seus resultados, a importância da definição da métrica de calculo do erro de imputação em se

tratando de dados de natureza categórica, já que seus resultados não se mostraram satisfatórios.

LEI (2006) propõe também em seus estudos melhorias para o cálculo da distância, também denominado por ele como cálculo de similaridade, utilizando o algoritmo *K-Modes* e aperfeiçoando-o. O método aperfeiçoado, denominado de *K-Representatives,* sugere que a similaridade entre os valores depende da freqüência relativa com que esses valores aparecem antes do agrupamento. Como resultado, ele ressalta a importância do teste de novas métricas no cálculo de similaridade.

SENTAS e ANGELIS (2006) estudam algumas técnicas para imputação categórica, e sugere uma nova técnica. As técnicas discutidas por ele são:

1. Remover a tupla com valores ausentes;
2. Uso da moda;
3. Utilização de imputação múltipla;
4. Utilização de um algoritmo *EM (Expectation-Maximization);*
5. Utilização do algoritmo proposto chamado *MLR (Multinominal logistics regression)*

Em seus resultados, SENTAS e ANGELIS (2006) observam que o método *MLR* atingiu melhor performance. Seu conceito básico está no fato da utilização de estudos probabilísticos categorizados para serem agregados na hora da imputação. A fórmula utilizada para esse cálculo é apresentada abaixo.

$$\log\left(\frac{prob(category_j)}{prob(category_q)}\right) = b_0^{(j)} + \sum_{i=1}^{k} b_1^{(j)} x_i \text{ com } \left(j = 1,2,\ldots,q-1\right)$$

Sendo k o número de registros e q o número de categorias.

LARSEN (2006) sugere a utilização de dois métodos para o processo de imputação categórica, a imputação fracionária (*IF*) e a imputação múltipla (*IM*), já conhecida. Ele

afirma que estes métodos criam múltiplas versões para os valores ausentes, o que reflete melhor a incerteza provocada por seus valores reais ausentes. O algoritmo *IM* gera um conjunto finito de imputações com uma distribuição com caráter de previsão posterior. O *IF* atribui pesos aos dados observados e seu foco central é o desenvolvimento de procedimentos para tabelas de contingência em dois sentidos parcialmente classificados e da final comparação com o *IM*.

Uma ferramenta muito conhecida de imputação é a ferramenta *PROC MI,* do sistema *SAS*, um software de Soluções em Inteligência Analítica de Negócios. ALLISON (2005) estuda o algoritmo chamado *Markov Chain Monte Carlo* (*MCMC*) que é o método padrão do *SAS* para imputação de dados. Este método é baseado na suposição da normalidade múltipla (SCHAFER, 1997) que implica que as imputações válidas podem ser geradas por equações de regressão linear. As razões para a popularidade de *MCMC* não são difíceis de descobrir. O algoritmo é extensamente disponível, computacionalmente eficiente, raramente provoca falhas, e, mais importante ainda, pode lidar com diferentes padrões de dados ausentes. Não obstante, pelo método supor a normalidade e as linearidades, ele pode não ser apropriado para imputar variáveis categóricas. Por esse motivo, ALLISON (2005) sugere algumas técnicas para melhorar a qualidade do processo de imputação, como, calcular as proporções estimadas, realizar a análise prévia dos casos de dados completos, realizar a imputação direta com arredondamentos ou utilizar métodos baseados em regressão logística. Esse estudo reforça a importância das métricas utilizadas para comparar valores categóricos. ALLISON (2005) também utiliza em seus testes, diferentes percentuais de valores ausentes nas bases de dados (10%, 20% e 30%), porém seus resultados não apontam ganhos ou perdas com essa variação percentual.

O trabalho de SOARES (2007) escolhido como base para nosso estudo, introduz o conceito de imputação composta, descrito na seção 2.5. Ele também utiliza diferentes

percentuais de ausência (10%, 20%, 30%, 40% e 50%) para verificar o impacto que isso pode acarretar. O mecanismo de ausência utilizado é o MCAR, ou seja, completamente aleatório. Esse mecanismo é utilizado em cerca de 90% dos trabalhos encontrados. Como este trabalho foi a base para nosso projeto, na próxima seção podemos ver com mais detalhes o que foi proposto por SOARES (2007).

2.5 Imputação Composta

SOARES (2007) propõe um processo de imputação precedida por tarefas usualmente aplicadas no processo de mineração de dados chamado **imputação composta**, e que serviu como base para o desenvolvimento deste trabalho. 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, acredita-se melhorar a qualidade do dado imputado 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.

A imputação composta busca tornar o processo de complementação de dados ausentes 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.

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

Exemplo: *agrupamento* → *imputação* significa que a tarefa de agrupamento precederá a de imputação.

- E(*v*, B): estratégia utilizada no processo de imputação de um atributo *v* de uma base de dados B. E(v, B) é representada por $T_1 \rightarrow T_2 \rightarrow ... \rightarrow T_m$, onde T_m é necessariamente uma tarefa de imputação.
- $A_{i:}$ algoritmo utilizado no processo de imputação.

Exemplos: A_1 = moda, 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(*v*, B): plano de imputação utilizado no processo de imputação de um atributo *v* de uma base de dados B. P(v, B) é representada por $A_1 \Rightarrow A_2 \Rightarrow ... \Rightarrow A_p$, 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 de agrupamento *K-Means*, A_2 = seleção com algoritmos genéticos e A_3 = algoritmo dos *k* vizinhos mais próximos.

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

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 ($\varepsilon(P(v)) = \varepsilon(I_k(v))$, onde $\varepsilon(I_k(v)) < \varepsilon(I_j(v))$, $\forall j \neq k$).

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.

Para materializar seus conceitos, SOARES (2007) desenvolveu em uma aplicação em *Java* chamada de *Appraisal*, que veremos com mais detalhes na próxima seção. Essas explanações são necessárias, visto que é a base deste projeto.

2.6 *Appraisal*

O sistema *Appraisal* (SOARES, 2007) implementa desde os planos para a execução da imputação composta assim como verifica a qualidade do dado que foi imputado. O sistema foi inspirado no termo que, em português, significa "aquele que desempenha a função de avaliar a qualidade".

O sistema é composto basicamente por quatro módulos, apresentados na figura 2.2:

1. O módulo de execução dos planos de imputação: *Crowner*;
2. O módulo de comitê de complementação de dados ausentes: *Committee*;
3. O módulo *Reviewer*, que verifica a qualidade das sugestões de imputação produzidas pelo módulo *Crowner*;
4. O módulo *Eraser*, que simula valores ausentes em uma base de dados segundo um mecanismo e um percentual de ausência definidos pelo usuário.

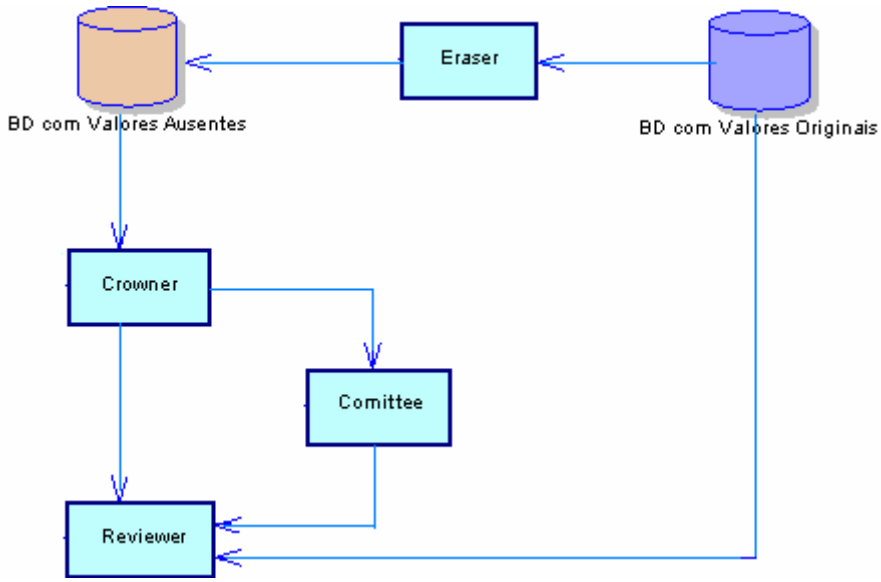

Figura 2.2 - Diagrama do sistema Appraisal *(SOARES, 2007)*

O módulo *Crowner* é o responsável por implementar os conceitos de estratégia, planos de imputação e instâncias de planos de imputação abordado na seção 2.5. 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.

No módulo *Committee* os resultados provenientes das diversas sugestões de imputação simples geradas pelas estratégias do módulo *Crowner* são combinados, seguindo a filosofia da imputação múltipla. Este módulo apresenta uma arquitetura onde a fase de treinamento é influenciada não só pelos valores imputados, mas por todos os valores dos atributos dos registros da tabela.

O módulo *Reviewer* é o responsável por 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. O módulo *Eraser* será mais bem descrito na seção 3.5.

Como o *Appraisal* foi desenvolvido para imputação composta apenas para dados de natureza numérica, os processos desenvolvidos inicialmente foram feitos para tratar apenas esse tipo de dados. Abaixo são listadas algumas tarefas para tratar o agrupamento, seleção e imputação de dados numéricos, já implementadas:

a) Agrupamento: agrupamento com *K* centróides (*K-Means);*

b) Seleção: análise de componentes principais (PCA);

c) Imputação: uso da média, do algoritmo dos *k* vizinhos mais próximos (*k*-NN) ou de Redes Neuronais *Back Propagation.*

Na execução, o sistema *Appraisal* utiliza *arquivos de propriedades*, que permite ao usuário configurar a base alvo das simulações, 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 e os parâmetros dos algoritmos.

CAPÍTULO 3

IMPUTAÇÃO CATEGÓRICA COM O *APPRAISAL*

3.1 Introdução

No capítulo 2 conhecemos algumas soluções propostas para o processo de imputação de valores ausentes em bases de dados. 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.

Conseguimos identificar na literatura disponível que apenas o trabalho de SOARES (2007) utiliza-se do processo de imputação composta, porém, trata apenas de dados de natureza numérica. Com isso, desenvolvemos neste projeto técnicas de imputação composta para dados de natureza categórica.

Nossa proposta é de integrar um componente ao *Appraisal,* para execução de tarefas de imputação composta com dados de natureza categórica. Isso é possível pela sua flexibilidade e escalabilidade.

Foram utilizados apenas dois módulos do *Appraisal* neste projeto, o *Crowner* e o *Eraser*, porém, este último, como falaremos na seção 3.5, não precisou ser desenvolvido para a versão categórica, pois já trabalha com o dado de qualquer natureza.

Na seção 2.6 vimos como é a arquitetura atual do *Appraisal*. Na figura 3.1 ilustramos como será a arquitetura final para este componente desenvolvido para tratar dados categóricos.

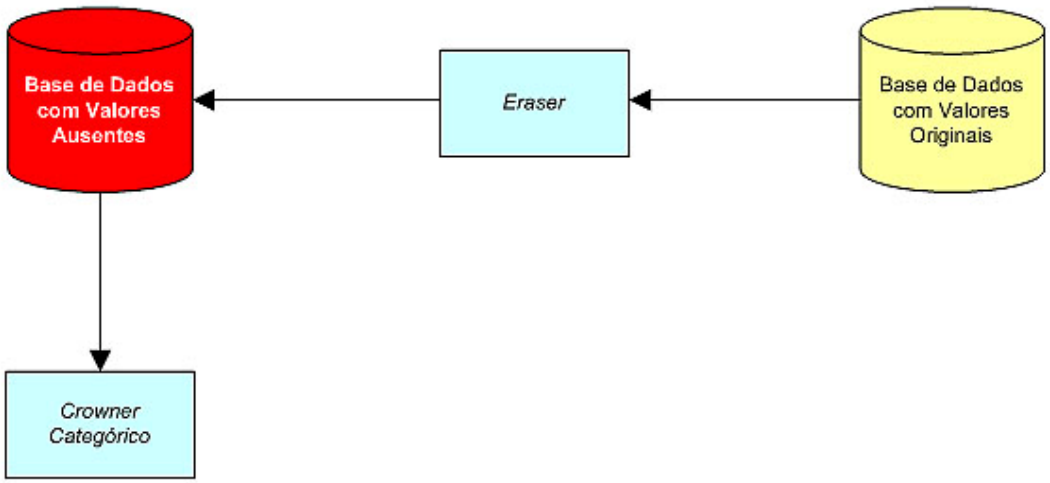

Figura 3.1 - Arquitetura proposta para a versão categórica do sistema Appraisal

Para entendermos melhor o funcionamento do módulo Crowner, ilustramos na figura 3.2, seu diagrama de atividades.

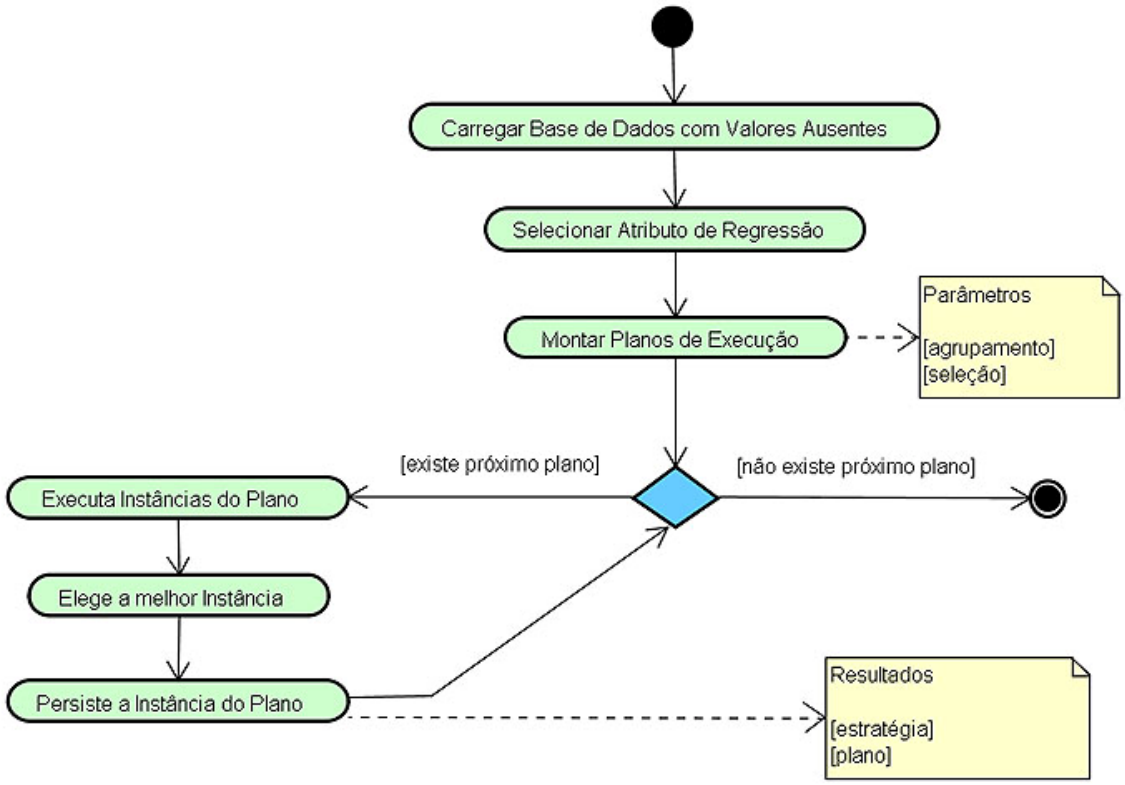

Figura 3.2 - Diagrama de Atividades do módulo Crowner *do sistema* Appraisal*, versão categórica*

Descrevemos abaixo o comportamento das principais classes implementadas no módulo *Crowner* versão categórica. A figura 3.3 ilustra esse diagrama de classes.

A interface *StrategyCategoric* 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 *StrategyCategoric* necessariamente retorna uma instância da classe *StrategyResultCategoric*, que representa o resultado final do processamento de uma estratégia.

A implementação de uma estratégia agrupa planos, representados por objetos que implementam a interface *PlanCategoric*, 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 *StageCategoric*. Como exemplo, podemos citar a implementação do plano *SelectionClusteringPlanCategoric*, que é composto por uma tarefa (estágio) de **seleção** (associação com *SelectionStageCategoric*), seguido de uma tarefa (estágio) de **agrupamento** (associação com *ClusteringStageCategoric*), e finalizada por uma tarefa (estágio) de **imputação** (associação com *RegressionStageCategoric*).

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 *SelectionClusteringStrategyCategoric* é associada a dois objetos *SelectionClusteringPlanCategoric*: um que combina os estágios *GeneticStage* → *CKmeansStage* → *CKnnStage* e outro que combina os estágios *GeneticStage* → *CKmeansStage* → *ModeStage*.

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. 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 *SelectionClusteringPlanCategoric* que combine *GeneticStage* → *CKmeansStage* → *CKnnStage* 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 seleção com algoritmos genéticos, 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 *RegressionResultCategoric*, que representa o produto da execução. O resultado de um plano, implementado pela classe *PlanResultCategoric*, consiste na coleção dos *RegressionResultCategoric* de cada uma das instâncias.

Finalmente, o objeto *StrategyResultCategoric*, é a coleção final dos objetos da classe *PlanResultCategoric* de todos os planos. Concluindo o exemplo, cada um dos dois objetos *SelectionClusteringPlanCategoric* produz como resultado um objeto da classe *PlanResultCategoric*, que agrupam os objetos da classe *RegressionResultCategoric* de suas instâncias correspondentes. A estratégia produz um resultado final do tipo *StrategyResultCategoric*, que agrupa os dois *PlanResultCategoric* gerados pelos planos.

Como o volume de dados produzido pelo sistema pode ser muito grande, foram implementados métodos para descarte dos piores objetos da *RegressionResultCategoric* em tempo real, bem como coletas dinâmicas 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.

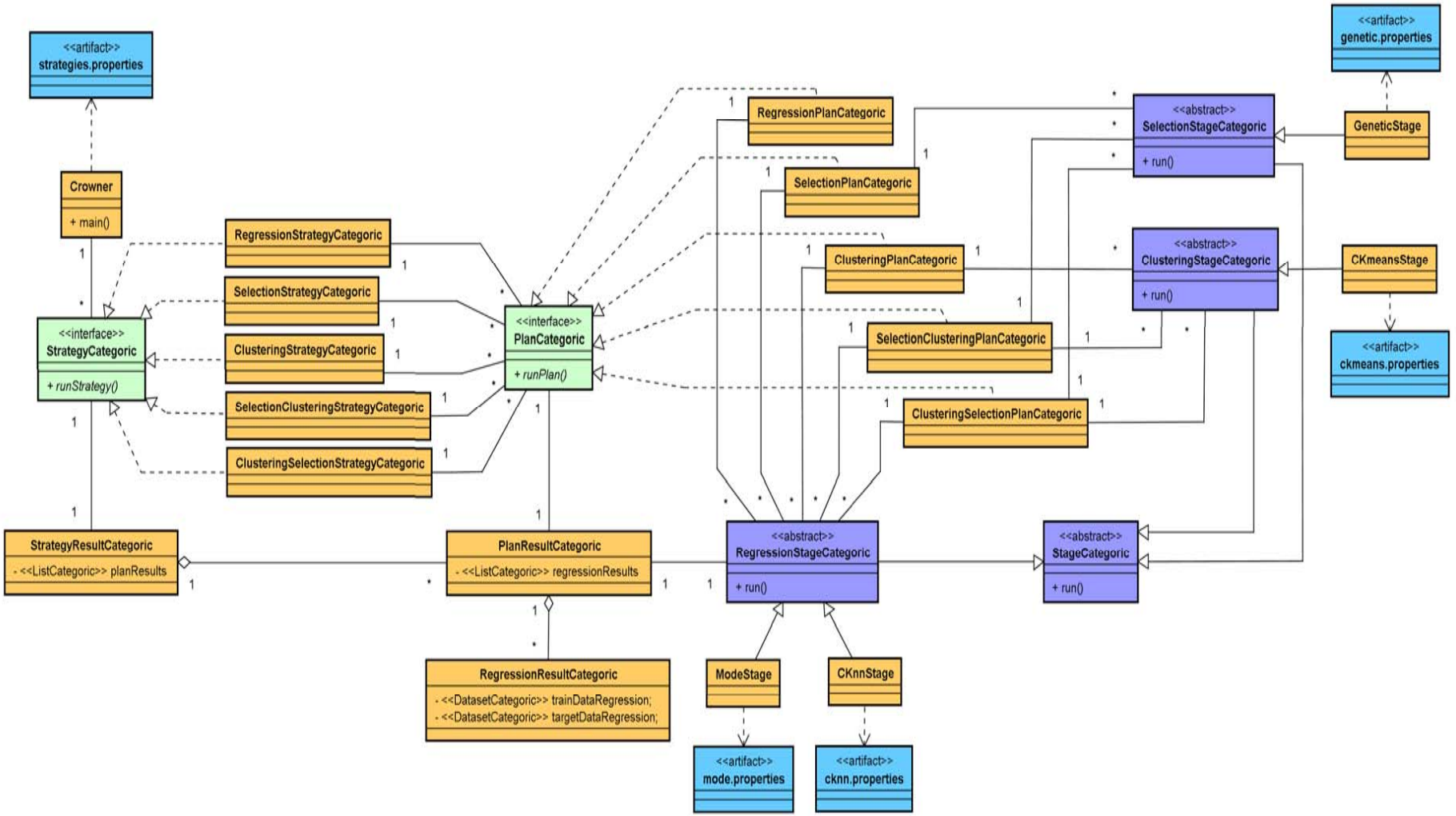

Figura 3.3 - Diagrama de classes do módulo Crowner *versão categórica do sistema* Appraisal

3.2 Seleção de Dados

A seleção de variáveis é uma das funções do processo de Descoberta de Conhecimento em Bases de Dados que reduz o subconjunto de atributos a serem utilizados pelo algoritmo de Mineração de Dados (GOLDSCHMIDT, PASSOS, 2005).

Considerando que os dados estejam organizados em uma estrutura tabular bidimensional a função de seleção de dados pode ter dois enfoques distintos redução de dados horizontal (tuplas) e redução de dados na vertical (colunas).

A redução de dados horizontal caracteriza-se pela diminuição do conjunto de dados (tuplas) da base, levando em conta a sua relevância para o objetivo do processo de descoberta de conhecimento.

Já a redução de dados vertical significa, em síntese, a eliminação de atributos (colunas) desnecessários na estrutura tabular bidimensional. Por exemplo, sendo S um subconjunto de dados com atributos $A_1, A_2, A_3, ..., A_n$. o problema da redução de dados vertical consiste em identificar qual dos 2^n elementos do conjunto das partes de $S\ (P(\{A_1, A_2, A_3,, ..., A_n\}))$ deve ser considerado no processo de *KDD*, com a formação mínima para que a informação original seja preservada.

Exemplo hipotético para $n = 3$, com os atributos (A_1, A_2, A_3):

$2^3=8$. Sendo assim, existem as seguintes combinações possíveis:

1. {}
2. $\{A_1\}$
3. $\{A_2\}$
4. $\{A_3\}$
5. $\{A_1, A_2\}$
6. $\{A_1, A_3\}$
7. $\{A_2, A_3\}$
8. $\{A_1, A_2, A_3\}$

O processo de seleção de dados é também um dos passos mais custosos computacionalmente e um dos mais importantes durante o tratamento dos dados, sendo normalmente necessários algoritmos de otimização, como, por exemplo, algoritmos genéticos.

Os algoritmos genéticos são uma classe de procedimentos de busca e otimização aleatórias capazes de realizar pesquisas adaptativas e robustas sobre um amplo espaço de pesquisa (ELMASRI *et al*, 2002). Estes algoritmos estão baseados nos processos genéticos dos organismos biológicos, codificando uma possível solução a um problema de "cromossomo" composto por cadeia de bits e caracteres. Estes cromossomos representam indivíduos que são levados ao longo de várias gerações, na forma similar aos problemas naturais, evoluindo de acordo com os princípios de seleção natural e sobrevivência dos mais aptos, descritos pela primeira vez por Charles Darwin em seu livro "Origem das Espécies". Emulando estes processos, os Algoritmos Genéticos são capazes de "evoluir" soluções de problemas do mundo real.

Utilizamos neste projeto o algoritmo de seleção de variáveis proposto por CONDE (2005), chamado **Seleção com Algoritmos Genéticos**.

O algoritmo baseia-se no modelo desenvolvido por YANG (1998) que sugere que cada indivíduo é representado por um vetor binário de dimensão *m*, sendo *m* o número total de características disponíveis. Sendo que o valor "*1"* significa que a característica deve ser considerada e "*0"* quando a característica deve ser descartada. Abaixo está um exemplo para esta representação, assumindo m igual a 3 (três atributos na tabela):

Seja um conjunto de atributos a serem analisados formados pelos atributos A_1, A_2, A_3, os indivíduos serão (2^m):

Individuo 1: 1 0 0
Individuo 2: 0 1 0
Individuo 3: 0 0 1
Individuo 4: 1 1 0
Individuo 5: 1 0 1
Individuo 6: 0 1 1
Individuo 7: 1 1 1
Individuo 8: 0 0 0

Deve-se levar em consideração que os atributos que fazem parte da chave primária ou chave candidata não devem ser considerados neste conjunto, assim como atributos de

alta granularidade e atributos que definem a classe. Atributos com alta granularidade em um determinado subconjunto faz com que a taxa de inconsistência seja baixa. Sendo assim, estes atributos devem ser removidos previamente do processo de seleção.

Neste método, a forma para saber se um subconjunto de características será escolhido ou não será através da qualidade de um determinado subconjunto. Esta qualidade seria imposta por uma taxa de inconsistência máxima, ou "*inconsistency rate*" (*IR*), predeterminada. Qualquer conjunto de atributos com taxa de inconsistência acima desta seria automaticamente descartado. Uma inconsistência ocorre quando duas ou mais instâncias (tuplas), tiverem o mesmo valor com exceção do atributo responsável pela definição da classe. E taxa de inconsistência é definida por LIU (1996) como sendo "a soma de todas as inconsistências encontradas, dividido pelo total de instâncias projetadas". A taxa de inconsistência pode ser calculada da seguinte fórmula:

$$IR = \frac{\sum_{i=1}^{g}\left(\sum_{j=1}^{c} c_j - \max(c_j)\right)}{N}$$

Sendo *IR* é a taxa de inconsistência, *g* o número de agrupamentos, *c* o número de classes, c_j a classe em cada agrupamento e *N* o número total de registros.

A partir da taxa de inconsistência define-se a função de aptidão (*fitness*), a ser utilizada pelo Algoritmo genético. A função de aptidão será o complemento desta taxa, expressa na fórmula abaixo:

$$f(x) = 1 - \frac{\sum_{i=1}^{g}\left(\sum_{j=1}^{c} \#c_j - \max(c_j)\right)}{N}$$

A partir do valor da função de aptidão dos indivíduos da primeira população é possível selecionar quais indivíduos são mais aptos para criação da geração seguinte, agora só falta definir o critério de parada para o algoritmo, como citamos abaixo:

- Quando for atingido o número de populações.
- Quando for definido o número de indivíduos possíveis.

3.3 Agrupamento 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. SILVA (2006) define ainda o conceito de modelo de agrupamento, um conjunto de grupos gerados a partir dos objetos de uma coleção. Além disso, SILVA (2006) complementa a definição mencionando o fato de que o agrupamento tem como objetivo, maximizar uma função objetiva implícita ou explícita, inerente aos dados. Na figura 3.4 exemplificamos um agrupamento.

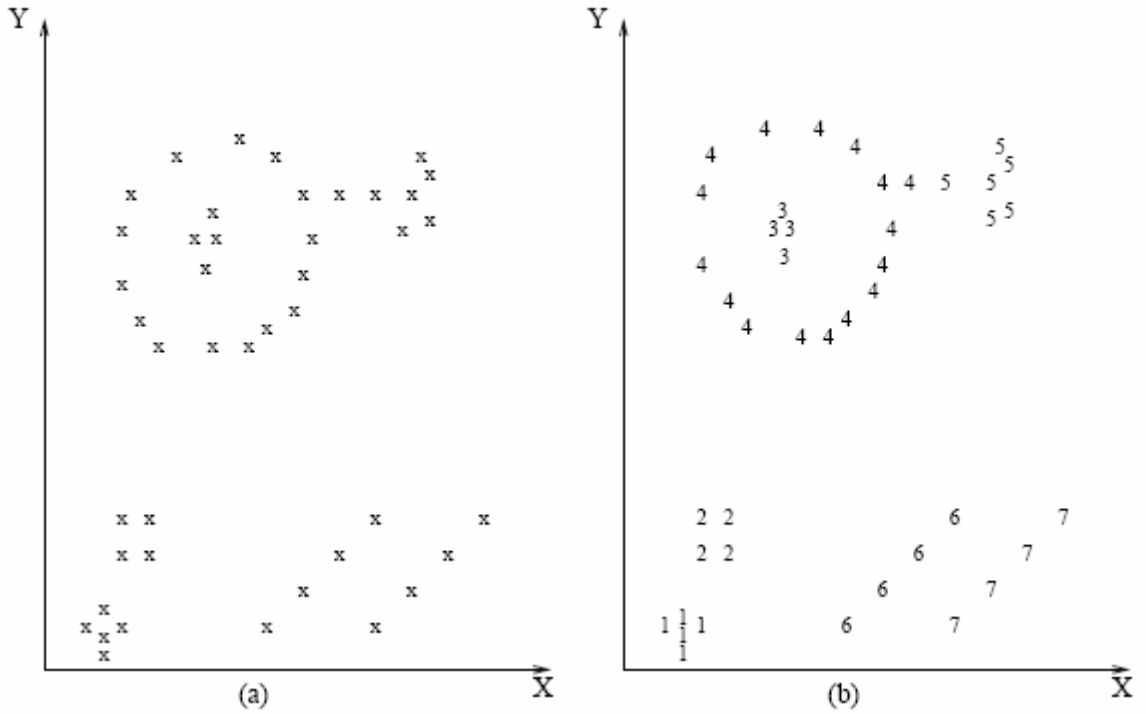

Figura 3.4 - Exemplo de agrupamento de dados em sete grupos, onde o rótulo sobre o elemento indica a qual grupo ele pertence. Fonte: (JAIN et al, 1999).

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.

Para essa tarefa escolhemos para usar como base neste projeto o algoritmo partitivo dos *K*-Centróides, mais conhecido como *K-Means* (MACQUEEN, 1967). 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. Este algoritmo em sua concepção trabalha apenas com dados numéricos e para realizarmos esta tarefa utilizando dados de natureza categórica, desenvolvemos o algoritmo chamado de *Categorical K-Means* (*CK-Means*).

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. (Cálculo da distância)

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.

A diferença apresentada para o método *K-Means* original está no cálculo da distância. 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, porém esse cálculo só pode ser feito com dados numéricos. Para calcular as distâncias entre elementos categóricos, introduzimos o **conceito de igualdade** como cálculo, onde para valores iguais atribui-se o valor "*0*" e para atributos diferentes, atribui-se "*1*".

$$d(x,y) = \begin{cases} 0, \textit{se } x = y \\ \\ 1, \textit{se } x \neq y \end{cases}$$, sendo x e y, elementos com valores categóricos.

3.4 Imputação Categórica

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

Como o objeto de estudo neste trabalho são atributos de natureza categórica, utilizamos duas técnicas de imputação muito conhecidas e muito utilizadas na literatura: a moda e uma variação do algoritmo dos *k*-vizinhos mais próximos (*k*-NN), chamado *Categorical k*-NN (*ck*-NN).

Atributos categóricos que possuam valores ausentes podem ser substituídos pela moda do conjunto de valores. A moda é o valor mais freqüente em um conjunto de valores. A moda pode não existir e, mesmo que exista, pode não ser única. Esta técnica pode ser utilizada em conjunto com outras técnicas para maximizar os resultados no complemento dos valores ausentes, e isso é o que veremos no final deste projeto. Exemplo: No grupo de valores $A = \{1, 1, 2, 2, 3, 3, 1, 1, 4, 4, 1, 1, 2, 2\}$, a moda deste conjunto é *1*, pois esse é o valor mais freqüente da amostra.

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.

No nosso caso, o algoritmo *ck*-NN utiliza a função de similaridade chamada **conceito de igualdade**, que também é utilizado no algoritmo *CK-Means*. A similaridade normalmente é medida através do cálculo da distância.

$$d(x,y) = \begin{cases} 0, \, se \; x = y \\ \\ 1, \, se \; x \neq y \end{cases}$$, sendo x e y, elementos com valores categóricos.

Abaixo segue o algoritmo *ck*-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} preenchido.

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 moda dos atributos t_{jk} das k tuplas selecionadas no passo anterior.

3.5 Simulação dos Dados Ausentes

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 3.5). O sistema escolhe com isso, 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 ser alterado para nulo. Assim, no exemplo da figura 3.6, as tuplas que possuem o atributo *safety* com valores iguais a "*low*" e o atributo *persons* com valores menores ou iguais a dois são selecionadas, e 20% delas têm o valor do atributo *maint* removido.

No caso onde se deseja 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. No exemplo da figura 3.7, o sistema seleciona dois subconjuntos da tabela: um com tuplas que possuam valores iguais a três no atributo *doors*, e com valores iguais a "*high*" no atributo *buying*. Cada um destes subconjuntos terá 40% dos valores removidos respectivamente nos atributos *doors* e *buying*.

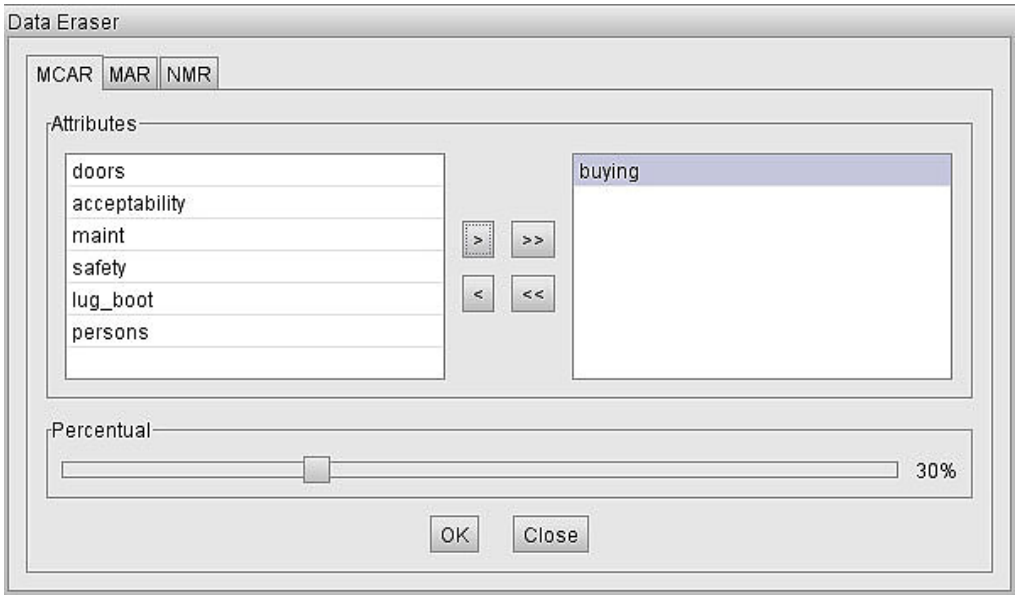

Figura 3.5 - Exemplo de remoção de valores do atributo buying *com o mecanismo completamente aleatório do módulo* Eraser *do sistema* Appraisal

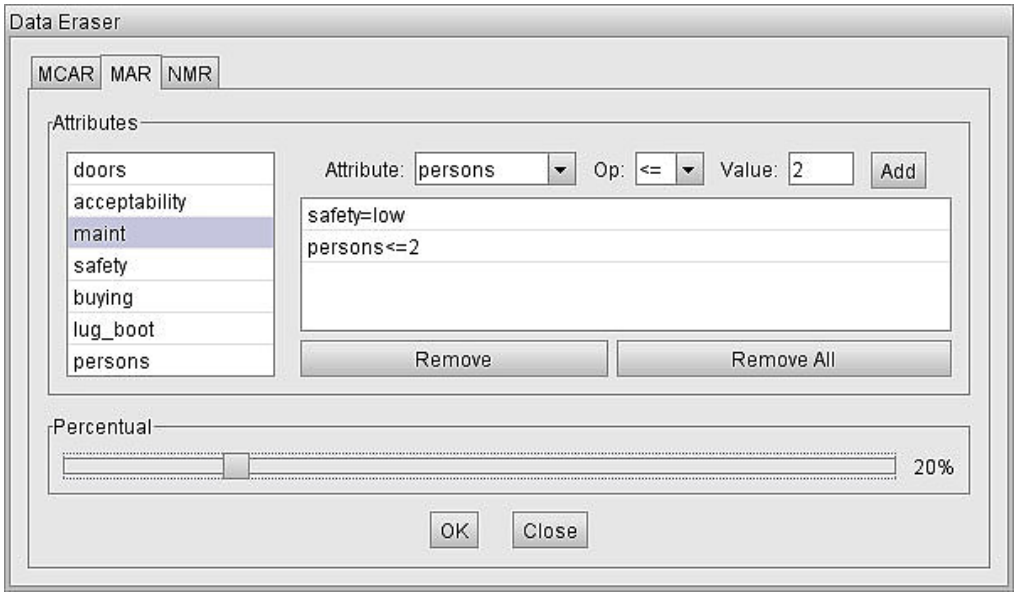

Figura 3.6 - Exemplo de remoção de valores do atributo maint *com o mecanismo aleatório do módulo* Eraser *do sistema* Appraisal

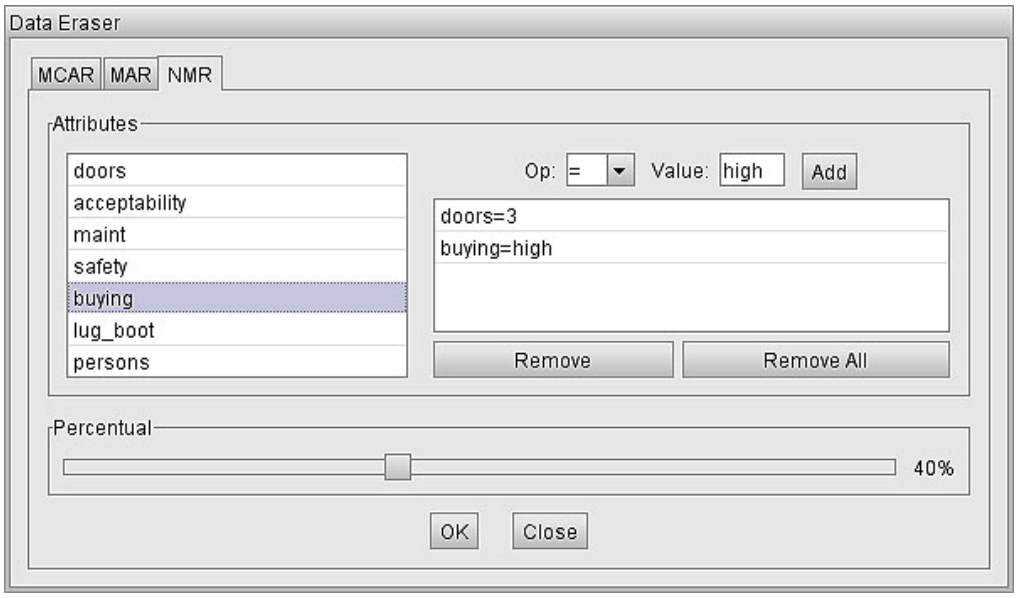

Figura 3.7 - Exemplo de remoção de valores dos atributos doors *e* buying *com o mecanismo não aleatório do módulo* Eraser *do sistema* Appraisal

CAPÍTULO 4

ANÁLISE DE RESULTADOS

4.1 Metodologia

4.1.1 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 *et al*, 1998): *Car Evaluation*, *Tic-Tac-Toe Endgame* e *Teaching Assistant Evaluation*. Três principais razões nos motivaram a escolher estas bases:

1) São conjuntos de dados freqüentemente 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 categóricos. As três bases utilizadas em nossos experimentos possuem esta característica. Esta escolha deveu-se ao fato de que nos interessa primordialmente neste trabalho avaliar tarefas de imputação no processo de complementação de dados categóricos;
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.

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.

4.1.2 Descrição das Bases

4.1.2.1 *Car Evaluation*

Esta base categórica avalia carros quanto a sua aceitabilidade classificando-as em *inaceitável, aceitável, boa e muito boa.* Existem 1728 tuplas na base, nenhum valor ausente e, a distribuição das classes é vista na tabela 4.1.

Tabela 4.1 – Distribuição das classes na base de dados Car Evaluation

Classe	**Nº de tuplas**	**%**
inaceitável	1210	70,023
aceitável	384	22,222
bom	69	3,993
muito bom	65	3,762

Os atributos da base são o preço (*buying*), preço de manutenção (*maint*), número de portas (*doors*), capacidade de pessoas (*persons*), tamanho da mala (*lug_boot*) e segurança estimada (*safety*). Na tabela 4.2 podem ser vistos os possíveis valores para cada campo.

Tabela 4.2 - Valores dos atributos da base de dados Car Evaluation

Atributo	**Valores possíveis**
buying	muito-alto, alto, médio e baixo
maint	muito-alto, alto, médio e baixo
doors	2, 3, 4 e 5+
persons	2, 3 e +
lug_boot	pequena, média, grande
safety	baixa, média, alta

4.1.2.2 *Tic-Tac-Toe Endgame*

A base *Tic-Tac-Toe Endgame* representa todas as configurações possíveis para o fim de um "jogo-da-velha". A classe apresenta dois valores, positivo ou negativo, representando a vitória do "x". Os atributos representam os nove quadrados do jogo, como na figura 4.1.

1	2	3
4	5	6
7	8	9

Figura 4.1 - Exemplo de tabuleiro de um jogo da velha

1) *top-left-square*
2) *top-middle-square*
3) *top-right-square*
4) *middle-left-square*
5) *middle-middle-square*
6) *middle-right-square*
7) *bottom-left-square*
8) *bottom-middle-square*
9) *bottom-right-square*

Os possíveis valores que podem ser assumidos pelos atributos são "x", "*o"* ou "b", que representa o espaço vazio. Esta base possui 958 tuplas, das quais 65,3% delas representam a classe "positivo", ou seja, uma vitória para "x*"*.

4.1.2.3 *Teaching Assistant Evaluation.*

Estes dados consistem em avaliações de desempenho do ensino em três semestres regulares e dois semestres de verão no Departamento de Estatísticas da Universidade de *Wisconsin-Madison*. Os registros foram divididos em três categorias aproximadamente iguais em distribuição ("baixo", "médio" e "alto") para dar forma à variável da classe *attribute*. Originalmente, a base apresenta 151 registros e não apresenta valores ausentes.

A base é composta, além da classe, de mais cinco atributos, relacionados abaixo:

1) *native_english_speaker:* indica se o professor é nascido ou não em um país de língua inglesa.
2) *instructor:* instrutor do curso, composto de 25 categorias.
3) *course:* curso, composto de 26 categorias.
4) *summer:* indica se o curso foi ministrado em semestre regular ou semestre de verão.
5) *size:* tamanho da classe.

4.1.3 Parâmetros relativos à ausência de 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 30 bases distintas para o conjunto de dados *Car Evaluation*, 45 bases para os dados *Tic-Tac-Toe Endgame*, e 25 bases para o conjunto de dados *Teaching Assistant Evaluation*. Cada base é identificada com o nome da base, concatenado ao nome do atributo e o percentual de valores ausentes. Por exemplo, a base *car_buying_10* refere-se à base que possui 10% de valores ausentes no atributo *buying* da base *Car Evaluation*.

Utilizaremos a solução utilizada por SOARES (2007) em que é utilizada uma variação de percentuais de valores ausentes entre 10% e 50%, com saltos de 10%. Esta medida foi tomada pelo fato de simular situações reais de ausência de dados. A faixa percentual de valores ausentes adotada cobre bem as características das bases reais, especialmente nas 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 monografia 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.

4.1.4 Parâmetros do algoritmo ck-*NN*

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: tipo de distância e a determinação do valor de *k* adotado.

4.1.4.1 Tipo de distância

O cálculo da distância entre atributos categóricos baseia-se nos experimentos encontrados na literatura. Neste trabalho utilizamos apenas um tipo de cálculo de distância, que é amplamente debatida em trabalhos relacionados e baseia-se no *conceito de igualdade*, atribuindo-se "*0"* para a distância entre dois atributos com o mesmo valor e "*1*" entre atributos com valores diferentes.

$$d(x,y) = \begin{cases} 0, \, se \; x = y \\ \\ 1, \, se \; x \neq y \end{cases}$$

Sendo x o valor original do atributo X de uma tupla e y é o valor a ser comparado para ter sua distância calculada.

Embora os valores dos atributos categóricos não possam ser análogos aos valores numéricos diretamente, ainda existem alguns graus da similaridade entre valores dos atributos na maioria das séries de dados categóricos.

Embora a computação da distância seja simples e fácil em algoritmos convencionais *k*-NN categóricos, esta não pode representar o significado dos valores do atributo exatamente. De fato, é quase impossível computar os valores numéricos representativos entre valores categóricos do atributo diretamente pela mão ou pelo conceito de seres humanos.

4.1.4.2 Determinação do valor de *k* adotado

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, arredondando para cima ($k = \sqrt{N}$). 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* (2001) 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*.

Com base nos trabalhos encontrados e pensando no tempo computacional a ser gasto nesse processo em se tratando de atributos categóricos, escolhemos variar *k* de 1 até a raiz quadrada de N arredondada para cima ($k = \sqrt{N}$), sendo N o número total de tuplas da base testada.

Nas tabelas 4.3, 4.4 e 4.5 podemos visualizar os valores utilizados na variação do *k* por percentual de valores ausentes.

Tabela 4.3 – Variação do k *por percentual de valores ausente da base* Car Evaluation

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-42	1-42	1-42	1-42	1-42

Tabela 4.4 – Variação do k *por percentual de valores ausente da base* Tic-Tac-Toe Endgame

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-31	1-31	1-31	1-31	1-31

Tabela 4.5 - Variação do k *por percentual de valores ausente da base* Teaching Assistant Evaluation

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-136	1-121	1-106	1-91	1-76

Na base *Teaching Assistant Evaluation,* devido ao pequeno número de registros, a variação do *k* foi de acordo com o número de registros válidos como na tabela 4.5, porém na grande maioria das vezes o vencedor foi um *k* de valor baixo.

Todos os testes realizados na Base *Car Evaluation* consumiram 151,5 horas, equivalente há um pouco mais de seis dias. Na base *Tic-Tac-Toe Endgame,* a execução durou 32,5 horas, o que equivale aproximadamente há um dia e meio. Já a base *Teaching Assistant Evaluation,* levou 12,8 horas, ou aproximadamente metade de um dia.

4.1.5 Parâmetros do algoritmo CK-Means

Como já abordado antes, 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.

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: a forma de inicialização dos centróides, a escolha da medida de distancia a ser usada, a escolha do número de centróides e o número de interações de cada rodada do algoritmo.

4.1.5.1 A forma de inicialização dos centróides

Na busca por soluções de inicialização dos *K* primeiros centróides, dentro do universo de registros categóricos, foram encontradas basicamente duas soluções para a inicialização. Encontramos inicializações na forma aleatória e encontramos também implementações onde os *k* primeiros objetos do conjunto de dados originais são escolhidos como os primeiros centróides (TEKNOMO, 2007). Para não dependermos da sorte, e como este trabalho trata de atributos categóricos, 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.

4.1.5.2 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. Assim como na implementação do algoritmo dos *k* vizinhos mais próximos, utilizamos também o uso da distância baseada no *conceito de igualdade*, atribuindo-se "*0"* caso a distância entre o centróide o atributo for iguais, caso contrário, o valor "*1"* é assumido.

$$d(x,y) = \begin{cases} 0, \, se \; x = y \\ \\ 1, \, se \; x \neq y \end{cases}$$

Sendo x o valor original do atributo X de uma tupla e y é o valor do centróide a ser comparado para ter sua distância calculada.

4.1.5.3 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, decidiu-se então, variar o número de grupos (número de centróides *K*) da mesma maneira em que foi variado para o algoritmo dos *k* vizinhos mais próximos, onde o valor ideal de *k* deve ser a raiz quadrada do número *N* de tuplas da base testada, arredondando para cima ($k = \sqrt{N}$).

Nas tabelas 4.6, 4.7 e 4.8 podemos visualizar os valores utilizados na variação do *k* por percentual de valores ausentes.

Tabela 4.6 - Variação do k *por percentual de valores ausente da base* Car Evaluation

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-42	1-42	1-42	1-42	1-42

Tabela 4.7 - Variação do k *por percentual de valores ausente da base* Tic-Tac-Toe Endgame

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-31	1-31	1-31	1-31	1-31

Tabela 4.8 - Variação do k *por percentual de valores ausente da base Teaching Assistant Evaluation*

Percentual de valores ausentes	10%	20%	30%	40%	50%
Valor de *k*	1-136	1-121	1-106	1-91	1-76

Na base *Teaching Assistant Evaluation,* devido ao pequeno número de registros, a variação do k foi de acordo com o número de registros válidos como na tabela 4.8, porém na grande maioria das vezes o vencedor foi um k de valor baixo.

4.1.5.4 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.

4.1.6 Parâmetros do algoritmo de Seleção com Algoritmos Genéticos

Nosso objetivo é o de realizar, em algumas das estratégias, uma seleção de variáveis utilizando Algoritmos Genéticos. 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 pela Seleção com Algoritmos Genéticos. Estes atributos serão utilizados pelas estratégias que envolvam seleção (*Seleção → Agrupamento → Imputação e Agrupamento → Seleção → Imputação*). As listas de prioridade decrescente das bases utilizadas em nossos experimentos são apresentadas a seguir:

- Base *Car Evaluation*
    - *1º) buying*
    - *2º) maint*
    - *3º) safety*

4º) *persons*

5º) *doors*

6º) *lug_boot*

- Base *Tic-Tac-Toe Endgame*

1º) *middle-middle-square*

2º) *top-left-square*

3º) *top-right-square*

4º) *bottom-left-square*

5º) *bottom-right-square*

6º) *top-middle-square*

7º) *middle-left-square*

8º) *middle-right-square*

9º) *bottom-middle-square*

- Base *Teaching Assistant Evaluation*

1º) *native_english_speaker*

2º) *summer*

3º) *size*

4º) *course*

5º) *instructor*

4.1.7 Medida do erro do processo de imputação

Os valores gerados pelos algoritmos de imputação de valores ausentes foram avaliados pela medida do *erro de igualdade*, que é calculado da seguinte forma:

$$ERR_X(j) = \sum_{j=1}^{m} \delta(x_j, y_j)$$

onde

$$\delta(x_j, y_j) = \begin{cases} 0, se\ x_j = y_j \\ 1, se\ x_j \neq y_j \end{cases}$$

Sendo x_j o valor original do atributo X da tupla j e y_j é o valor imputado para o atributo X nesta tupla j. Escolhemos esta medida por acreditarmos que ela melhor representa um dos fatores que desejamos medir: se o dado categórico inserido é o mesmo que o original.

Com a utilização dessa medida, temos como indicar qual estratégia de imputação foi melhor para certo atributo em uma base específica. Dentre todas as estratégias utilizadas, a que apresentar a menor taxa de erro será escolhida como a melhor.

4.1.8 Condições ambientais dos experimentos

Os experimentos foram realizados em um microcomputador com processador *AMD Athlon* XP 2000, com 512 MB de memória principal, e 80 GB de disco rígido IDE, com o sistema operacional Windows XP Professional versão 2002 Service Pack 2 instalado. O componente de imputação composta categórica incorporado ao *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 SGDB *MySQL* versão 4.1.1.

4.2 Resultado dos Experimentos

4.2.1 Variação do k *no Algoritmo* ck-*NN*

Na tabela 4.9 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 que foram encontrados após a execução.

Tabela 4.9 - Valores do maior e do menor k *"vencedor" em cada base de dados no algoritmo* ck-*NN*

	Menor *k*	Maior *k*
Car Evaluation	1	42
Tic-Tac-Toe Endgame	8	28
Teaching Assistant Evaluation	1	89

4.2.1.1 Gráficos dos valores de *k* encontrados para o algoritmo *ck*-NN

Car Evaluation

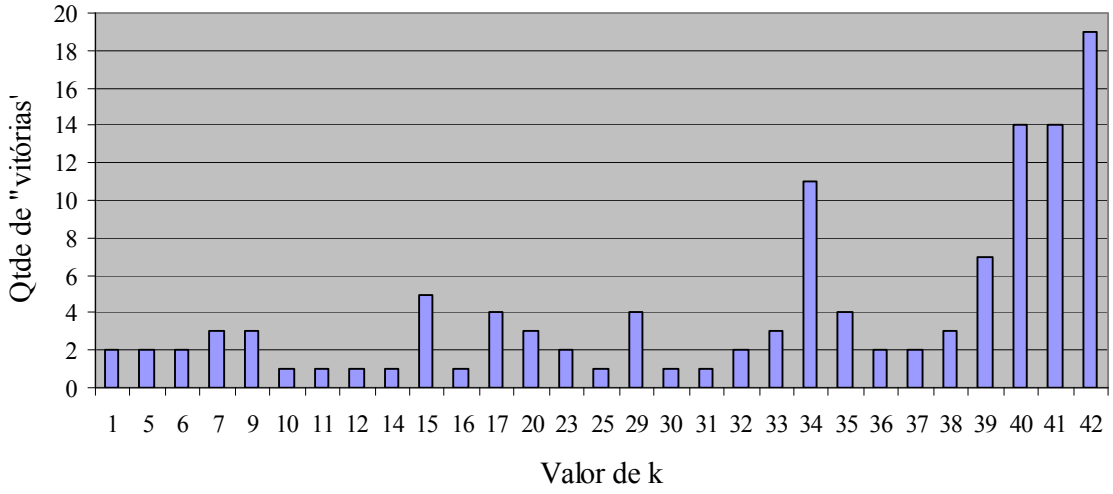

Figura 4.2 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Car Evaluation *após a execução do algoritmo* ck-*NN*

Tic-Tac-Toe Endgame

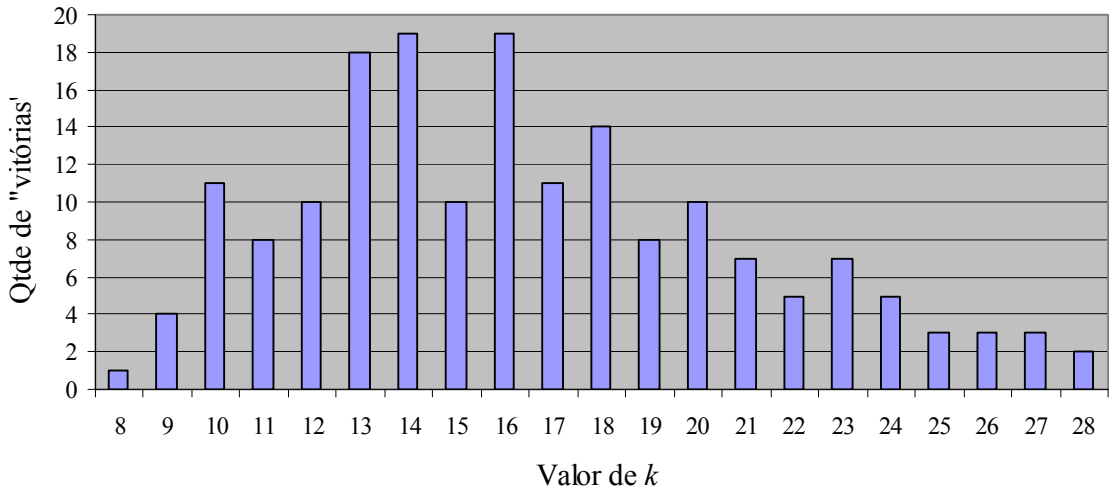

Figura 4.3 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Tic-Tac-Toe Endgame *após a execução do algoritmo* ck-*NN*

Teaching Assistant Evaluation

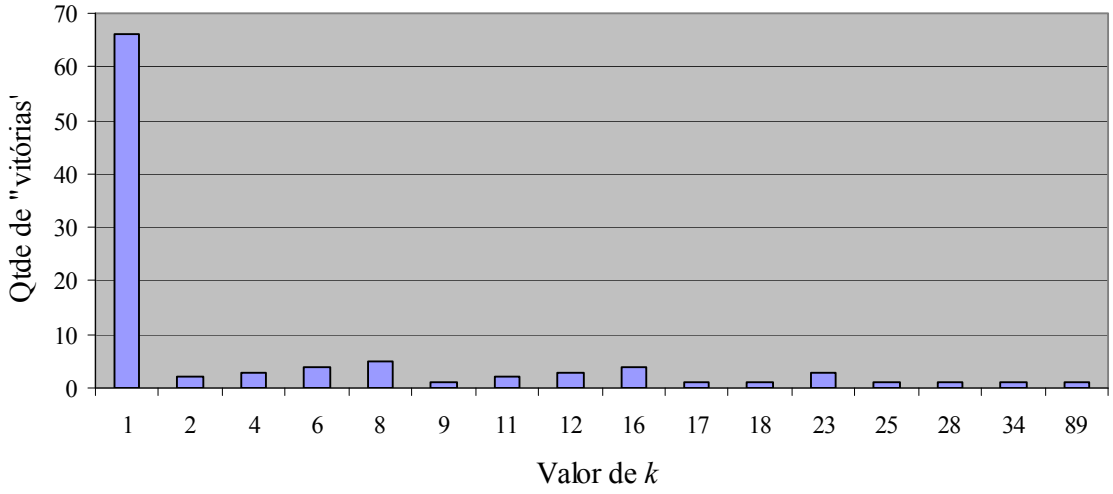

Figura 4.4 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Teaching Assistant Evaluation *após a execução do algoritmo* ck-*NN*

4.2.2 Variação do k *no Algoritmo* CK-Means

Na tabela 4.10 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 que foram encontrados após a execução:

Tabela 4.10 - Valores do maior e do menor k *"vencedor" em cada base de dados no algoritmo* CK-Means

	Menor *k*	Maior *k*
Car Evaluation	2	38
Tic-Tac-Toe Endgame	2	31
Teaching Assistant Evaluation	2	24

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.

4.2.2.1 Gráfico dos valores de *k* encontrados para o algoritmo *CK-Means*

Car Evaluation

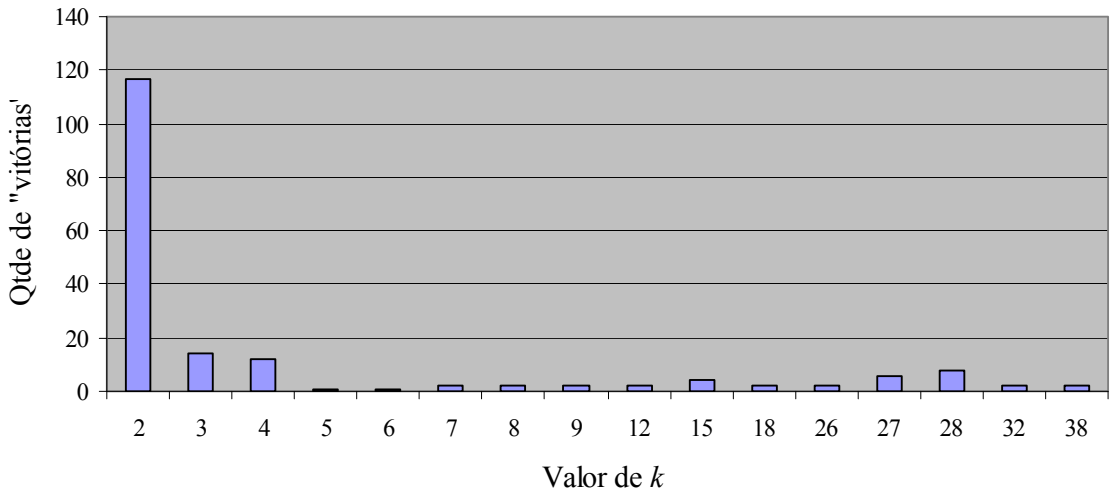

Figura 4.5 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Car Evaluation *após a execução do algoritmo* CK-Means

Tic-Tac-Toe Endgame

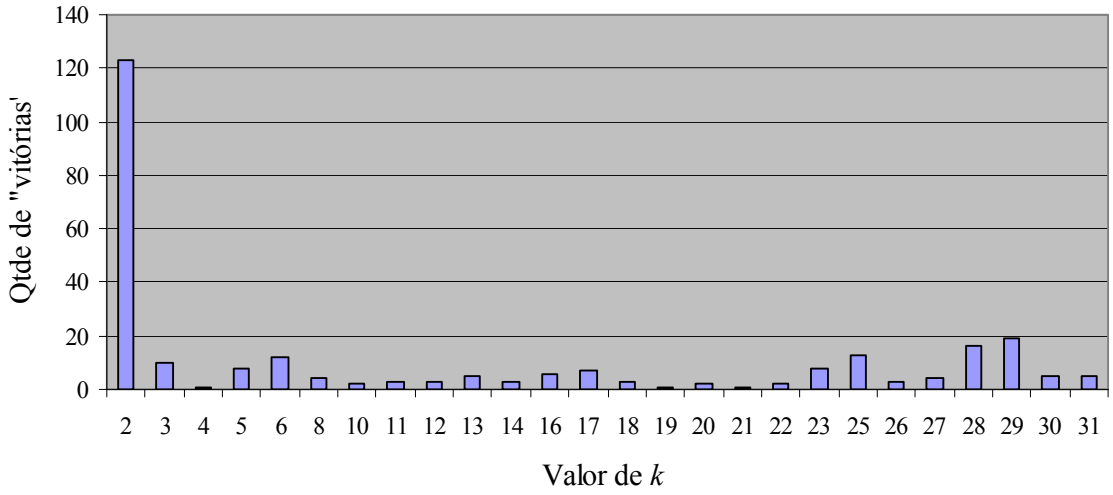

Figura 4.6 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Tic-Tac-Toe Endgame *após a execução do algoritmo* CK-Means

Teaching Assistant Evaluation

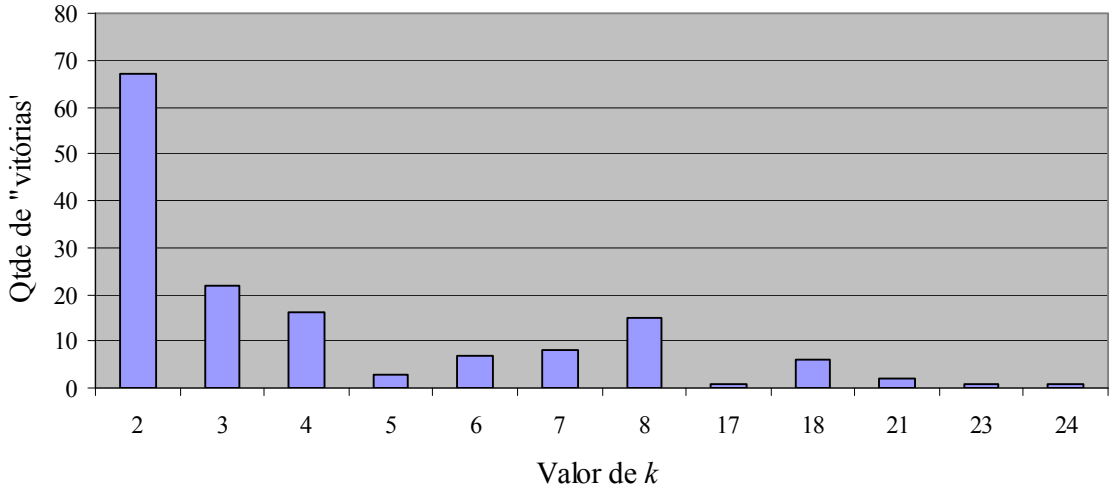

Figura 4.7 - Resultado da quantidade de vezes os valores de k *foram "vencedores" em todas os planos de imputação na base* Teaching Assistant Evaluation *após a execução do algoritmo* CK-Means

4.2.3 Estratégias de complementação de dados

4.2.3.1 Análise dos resultados por bases 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 revelou 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.

Analisando a base *Car Evaluation*, percebemos que a estratégia de agrupamento e regressão demonstrou os melhores resultados em 31% dos casos. Analisando a seleção de uma forma geral, percebemos que não houve nenhum ganho substancial na descoberta dos valores ausentes. A regressão aplicada isoladamente apresentou resultados satisfatórios na descoberta da informação em 25% dos casos. A estratégia que apresentou menos "vitórias" perante as outras foi a de Seleção, Agrupamento e Regressão com 17% dos casos. Já a estratégia de Agrupamento, Seleção e Regressão saíram-se melhor em 27% dos casos.

Com relação à base *Tic-Tac-Toe Endgame*, observamos uma situação um pouco diferente da anterior. A estratégia de regressão aplicada isoladamente apresentou sucesso em 81% dos casos. Se considerarmos todas as estratégias envolvendo agrupamento, este índice desce para 19%. Quando consideramos as estratégias envolvendo seleção, 17% dos casos obtiveram sucesso.

Assim como na primeira base, a base *Teaching Assistant Evaluation* indica que os resultados foram equilibrados. Porém, a estratégia que mais obteve sucesso, 28% dos casos, foi a conjunção de Agrupamento, Seleção e Regressão. A conjunção da seleção e agrupamento predominou nos resultados comparativos das estratégias. Em 50% dos casos, a combinação destas duas tarefas revelou um erro médio menor do que as demais estratégias. Quando envolvemos apenas o agrupamento, este resultado chega a 75%.

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.

4.2.3.2 Gráficos por bases de dados

Car Evaluation

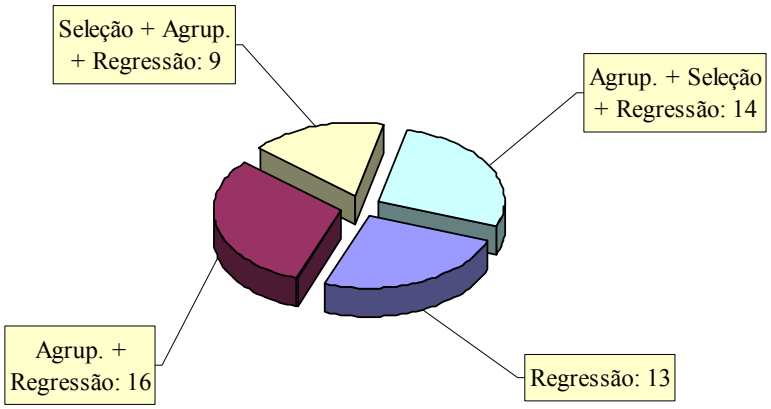

Figura 4.8 – Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation

Tic-Tac-Toe Endgame

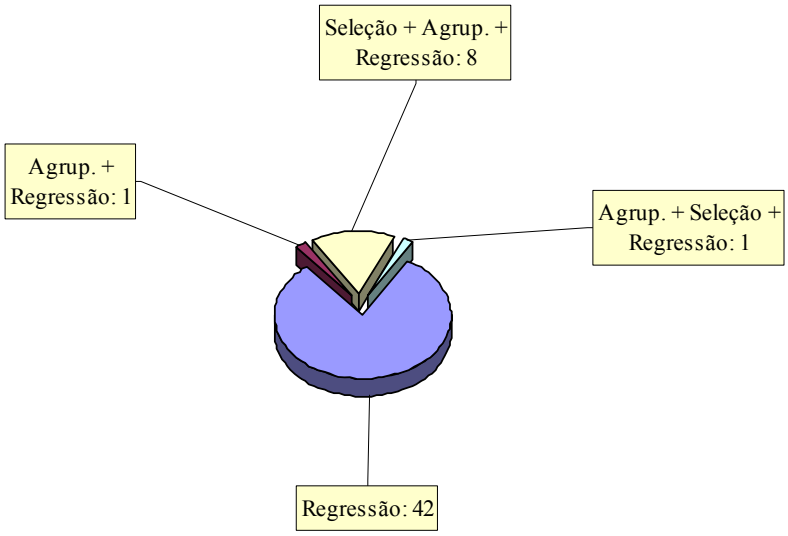

Figura 4.9 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame

Teaching Assistant Evaluation

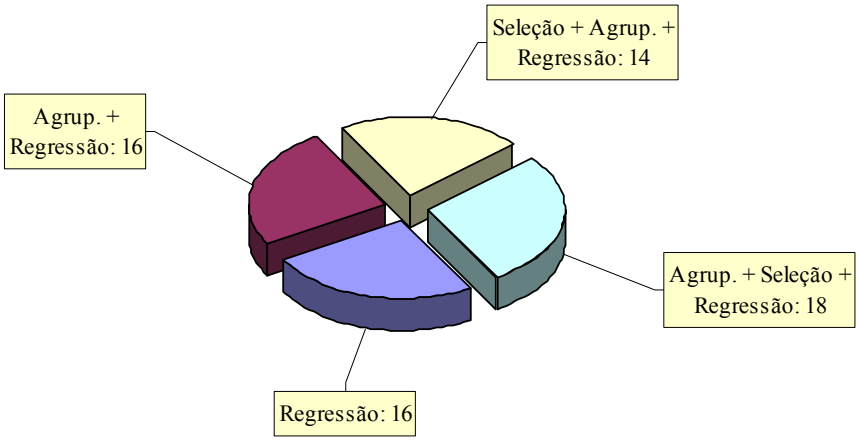

Figura 4.10 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation

4.2.3.3 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 que os dados ocultam). Isto pode influenciar 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 *Car Evaluation*, o agrupamento se mostrou a melhor estratégia em quase todos os casos, com exceção dos percentuais de 10% e 40% de valores ausentes, que apresentaram a regressão simples como o melhor resultado. Na maioria dos casos, o que predomina é o equilíbrio entre as estratégias, com exceção dos percentuais de 20% e 50%

onde a regressão simples e a estratégia composta de seleção, agrupamento e regressão apresentaram resultados pouco satisfatórios.

A regressão simples desempenha um papel mais importante nos resultados da base *Tic-Tac-Toe Endgame*. Os resultados revelam que em todos os casos, a estratégia simples aparece com o melhor índice de desempenho. Este comportamento nos leva a crer que a alta correlação entre os seus atributos faça com que a regressão galgue um papel de maior destaque, pelo fato de eliminar colunas ou agrupar os dados pouco ou nada contribuem no processo de imputação nesta base.

De forma geral, na base *Teaching Assistant Evaluation* as estratégias se equilibram em relação às taxas de erros medidos. Com um menor índice de valores ausentes, a estratégia de regressão simples apresenta os melhores resultados. À medida que aumenta o número de tuplas com atributos ausentes, a regressão simples não se torna mais tão satisfatória. Entretanto, mais uma vez destacamos o desempenho do agrupamento em quase todos os percentuais de valores ausentes nesta base, demonstrando ratificar que esta é uma tarefa de extrema importância ao processo de complementação de dados ausentes, mesmo com a complexidade exigida em bases categóricas em comparação a bases numéricas.

4.2.3.4 Gráficos por percentual de valores ausentes

Car Evaluation

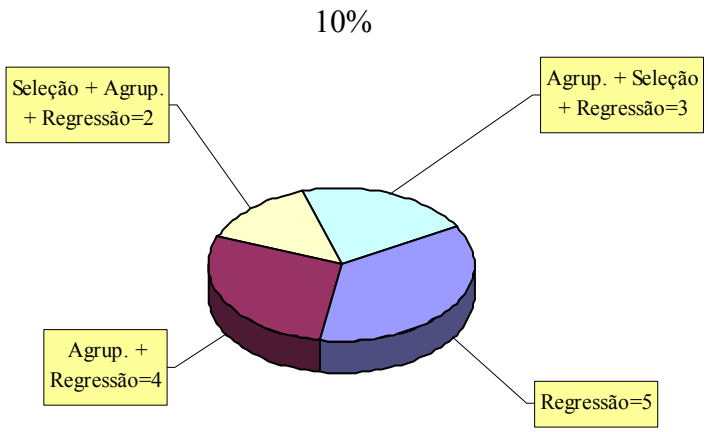

Figura 4.11 - Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation *na ausência percentual de 10% dos valores*

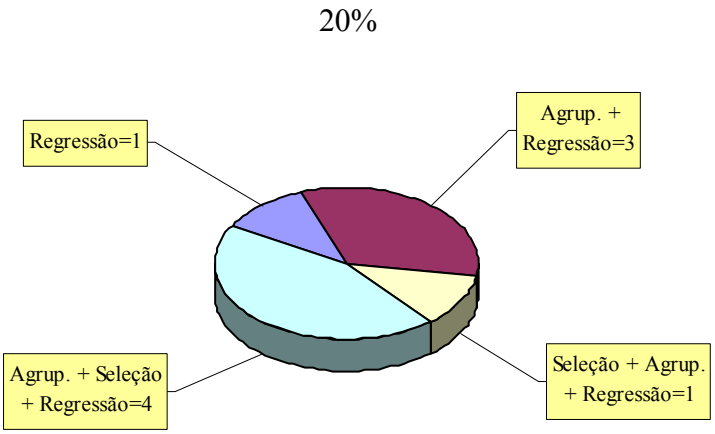

Figura 4.12 - Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation *na ausência percentual de 20% dos valores*

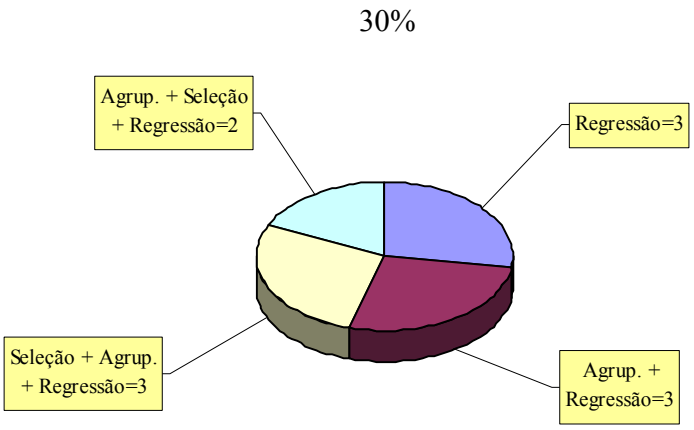

Figura 4.13 - Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation *na ausência percentual de 30% dos valores*

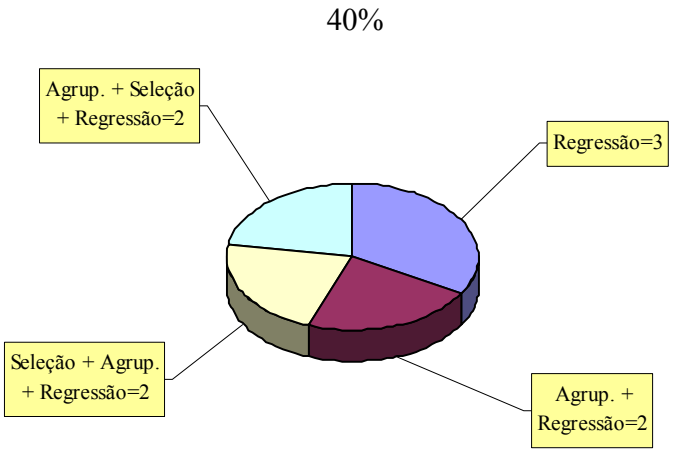

Figura 4.14 - Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation *na ausência percentual de 40% dos valores*

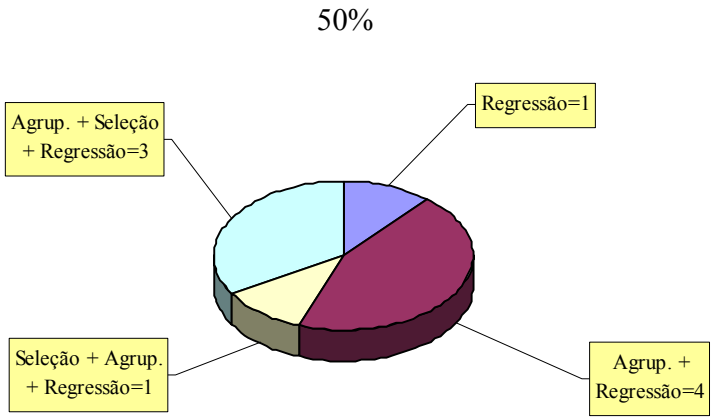

Figura 4.15 - Resultado das estratégias vencedoras dos testes realizados na base Car Evaluation *na ausência percentual de 50% dos valores*

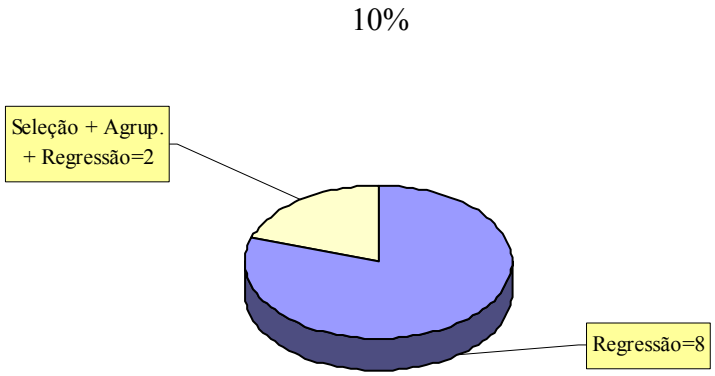

Figura 4.16 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame *na ausência percentual de 10% dos valores*

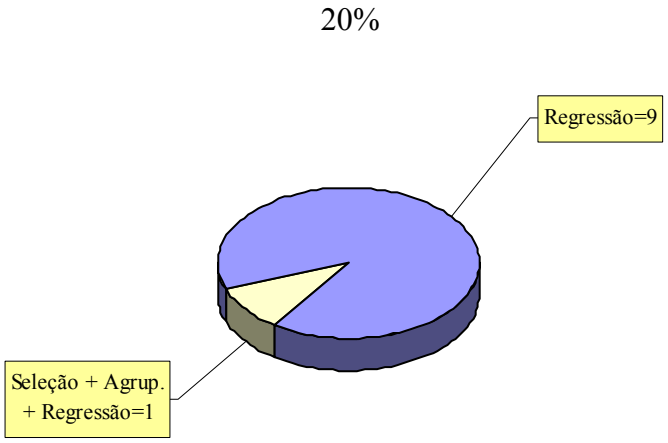

Figura 4.17 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame *na ausência percentual de 20% dos valores*

30%

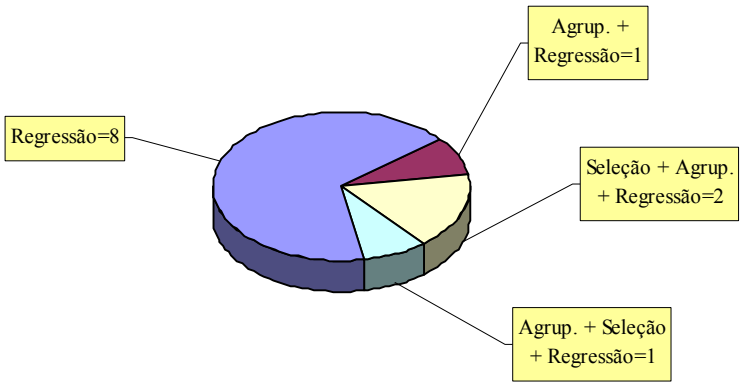

Figura 4.18 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame *na ausência percentual de 30% dos valores*

40%

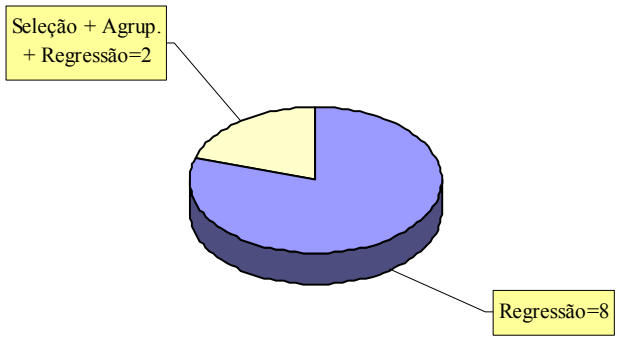

Figura 4.19 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame *na ausência percentual de 40% dos valores*

50%

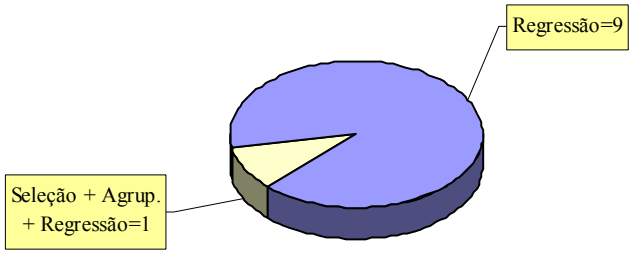

Figura 4.20 - Resultado das estratégias vencedoras dos testes realizados na base Tic-Tac-Toe Endgame *na ausência percentual de 50% dos valores*

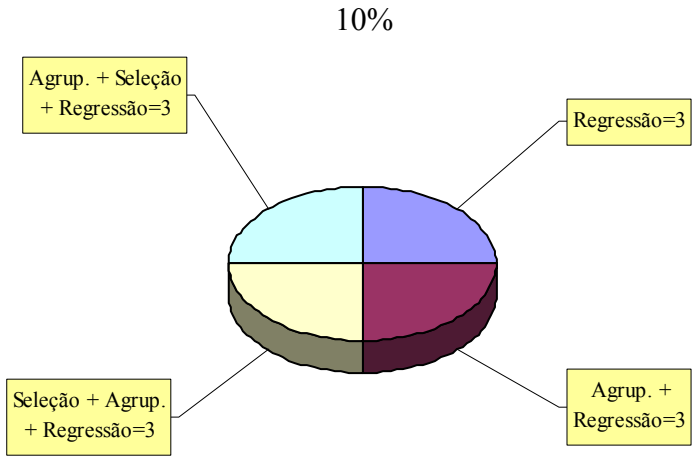

Figura 4.21 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation *na ausência percentual de 10% dos valores*

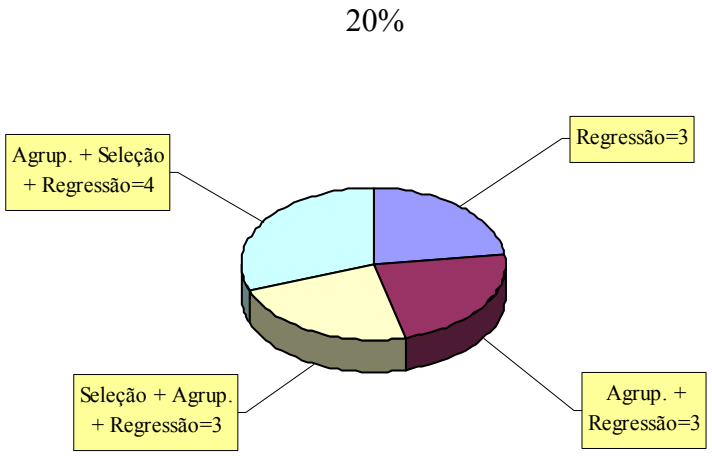

Figura 4.22 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation *na ausência percentual de 20% dos valores*

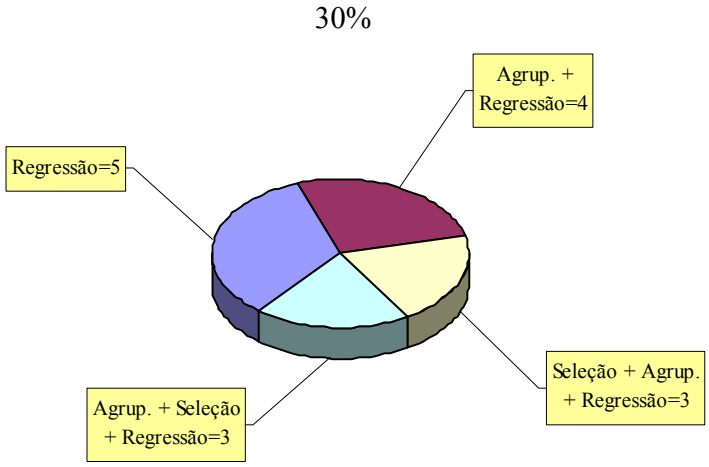

Figura 4.23 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation *na ausência percentual de 30% dos valores*

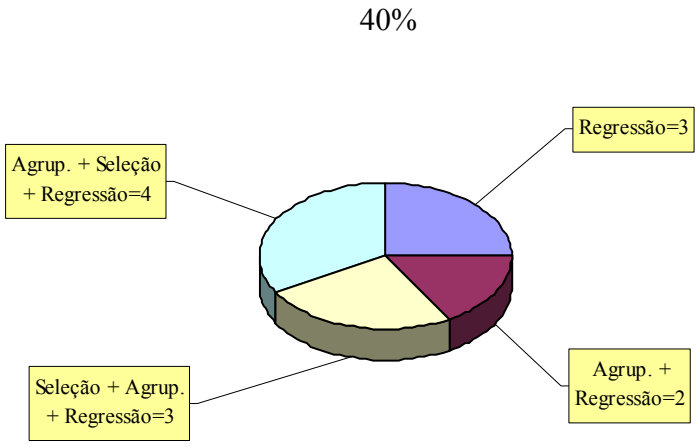

Figura 4.24 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation *na ausência percentual de 40% dos valores*

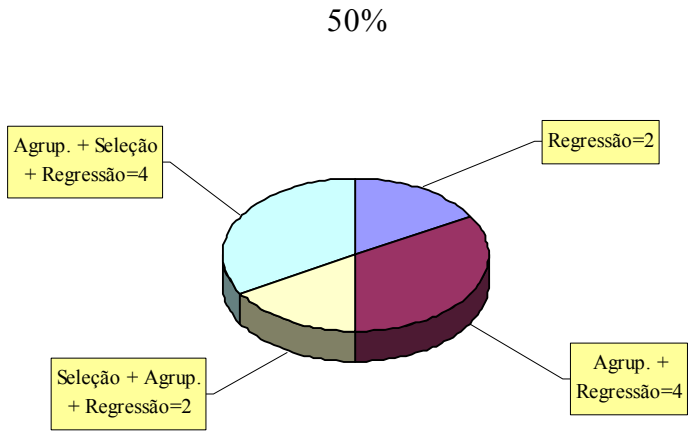

Figura 4.25 - Resultado das estratégias vencedoras dos testes realizados na base Teaching Assistant Evaluation *na ausência percentual de 50% dos valores*

4.2.4 Execução dos Planos de Imputação

4.2.4.1 Análise dos Resultados

A imputação com a moda, como mostram os resultados dos experimentos, é uma opção ruim, já que apresenta uma das piores médias de erro relativo absoluto. Na base *Car Evaluation,* a moda quase sempre ficou na antepenúltima colocação, enquanto nas outras duas bases, esta não passou da penúltima colocação. Quando a imputação por moda 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.

Na estratégia de seleção de atributos seguida do agrupamento e imputação com moda houve uma pequena melhora nos resultados obtidos somente com a imputação com moda precedida do agrupamento apenas na base *Car Evaluation*. Contudo, nas outras duas bases os resultados foram ruins, figurando na antepenúltima posição. Ou seja, não houve melhora significativa em relação aos planos de imputação com moda, com a seleção precedendo essa imputação.

O agrupamento precedendo a seleção de atributos e a imputação por moda apresenta resultados mais regulares, e mais satisfatórios, porém ainda não sendo o ideal. Os resultados demonstram que o agrupamento quando vem antes da seleção faz com que o processo de complementação de dados ausentes leve em conta os aspectos inerentes ao grupo, proporcionando desta forma uma melhor seleção de atributos. Na base *Tic-Tac-Toe Endgame* esta estratégia ficou na maioria das vezes, na terceira e segunda posições, respectivamente.

Os resultados mais animadores de uma maneira geral apareceram com a utilização da imputação com o algoritmo dos *k* vizinhos mais próximos (*ck*-NN). Porém, ela ainda apresenta certa instabilidade para a base *Car Evaluation*, apesar de seus resultados serem muito mais interessantes do que as obtidas com a moda. Seus resultados variam entre a

primeira e oitava colocações. A utilização dos *k* registros mais semelhantes no processo de imputação melhora a qualidade do dado imputado, como verificado nas outras duas bases.

Os resultados alcançados pelo plano de agrupamento precedendo a regressão com *ck*-NN na base *Teaching Assistant Evaluation* foram muito bons. A formação de grupos ajudou o algoritmo dos *k* vizinhos a melhor selecionar as tuplas mais similares. Na base *Tic-Tac-Toe Endgame*, os resultados não são os melhores, porém seu resultado mediano possa ser explicado pela alta correlação entre os seus atributos. Na outra base, *Car Evaluation*, os resultados encontram-se mais distribuídos, porém mesmo assim, satisfatórios.

De uma forma geral obtivemos resultados medianos com a aplicação da seleção precedendo o agrupamento e a imputação com o algoritmo *ck*-NN nas bases *Car Evaluation* e *Tic-Tac-Toe Endgame*. Os resultados foram bons na base *Teaching Assistant Evaluation*, 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.

Na aplicação do plano de imputação envolvendo o agrupamento seguido de seleção de dados, e utilizando como algoritmo regressor o *ck*-NN, obtivemos bons resultados nas bases *Car Evaluation*. Na *Teaching Assistant Evaluation* obtivemos os melhores resultados utilizando esta estratégia. Nessas duas bases de dados utilizadas, o desempenho foi bastante satisfatório e regular. Porém, mais uma vez o desempenho na base *Tic-Tac-Toe Endgame* foi apenas mediano. 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 *ck*-NN, já que os vizinhos mais próximos são os melhores qualificados para participar do processo de imputação.

Analisando os vários experimentos realizados neste projeto monográfico, podemos dizer que a utilização apenas da regressão simples utilizando o algoritmo *ck*-NN mostra-se como a estratégia mais satisfatória de uma maneira geral, considerando as três bases juntas. Ao analisarmos individualmente as bases, podemos destacar na tabela 4.11 as melhores estratégias por base de dados.

Tabela 4.11 - Melhore estratégia de complementação por base de dados

Base	Melhor estratégia
Car Evaluation	agrupamento, seleção e imputação com *ck*-NN
Tic-Tac-Toe Endgame	imputação com *ck*-NN
Teaching Assistant Evaluation	agrupamento, seleção e imputação com c*k*-NN

De uma maneira geral, podemos dizer que os testes realizados nesta monografia indicam que a seleção e o agrupamento de dados são tarefas importantes para a qualidade do processo de imputação quando combinadas, e dependendo da disponibilidade de recursos e de tempo, além da característica das bases, também podem ser utilizadas.

4.2.4.2 Gráficos

Plano 1: Imputação com Moda

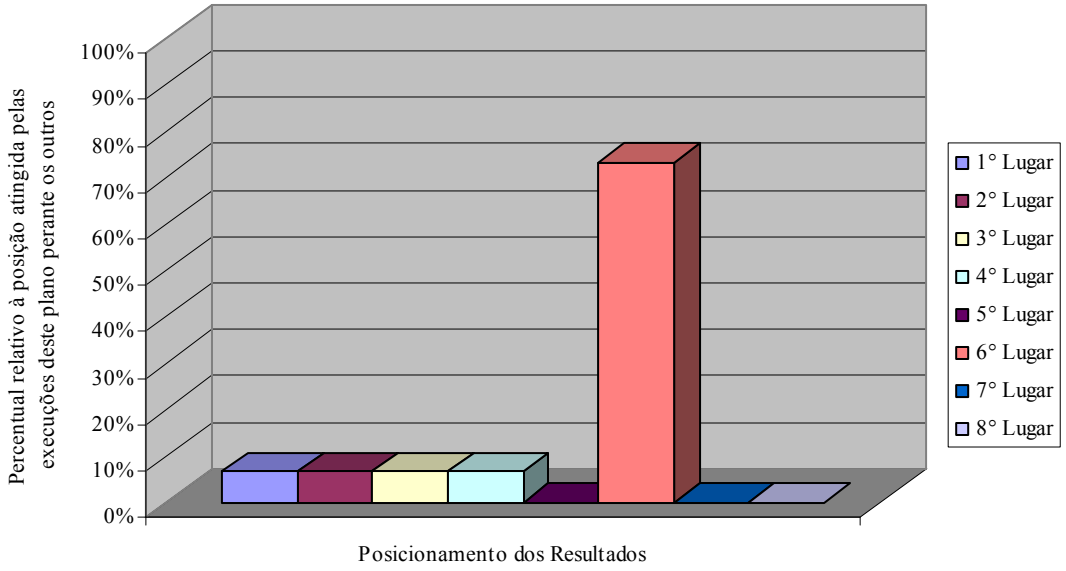

Figura 4.26 - Classificação dos resultados para o plano de Imputação com Moda da base Car Evaluation

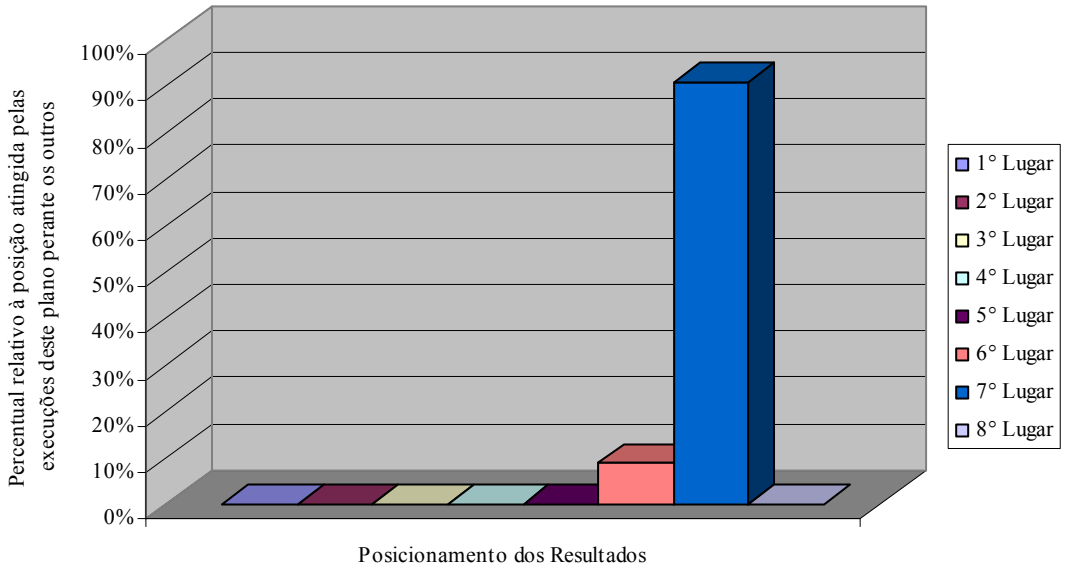

Figura 4.27 - Classificação dos resultados para o plano de Imputação com Moda da base Tic-Tac-Toe Endgame

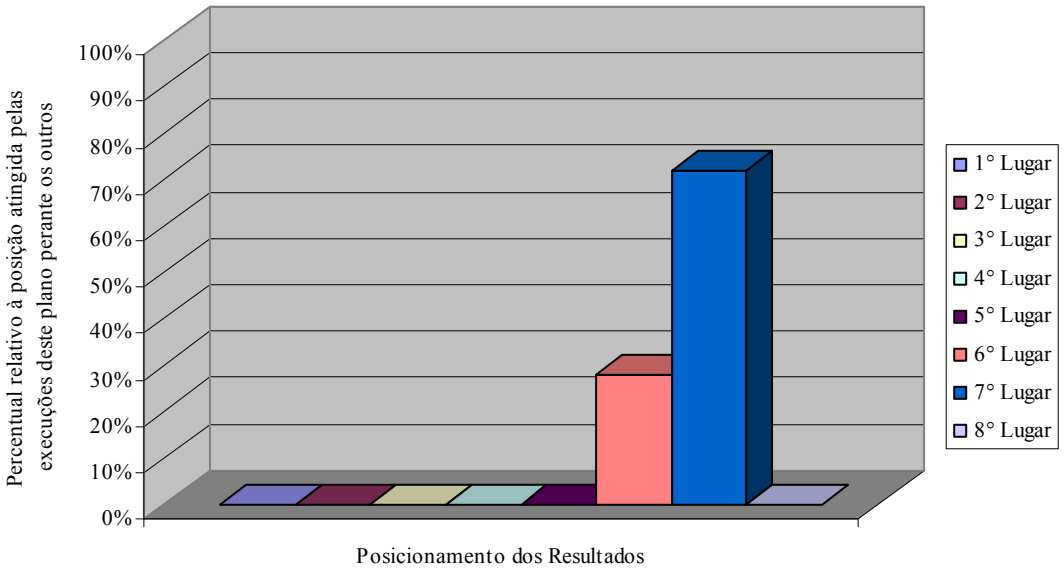

Figura 4.28 - Classificação dos resultados para o plano de Imputação com Moda da base Teaching Assistant Evaluation

Plano 2: Agrupamento com CK-Means *e Imputação com Moda*

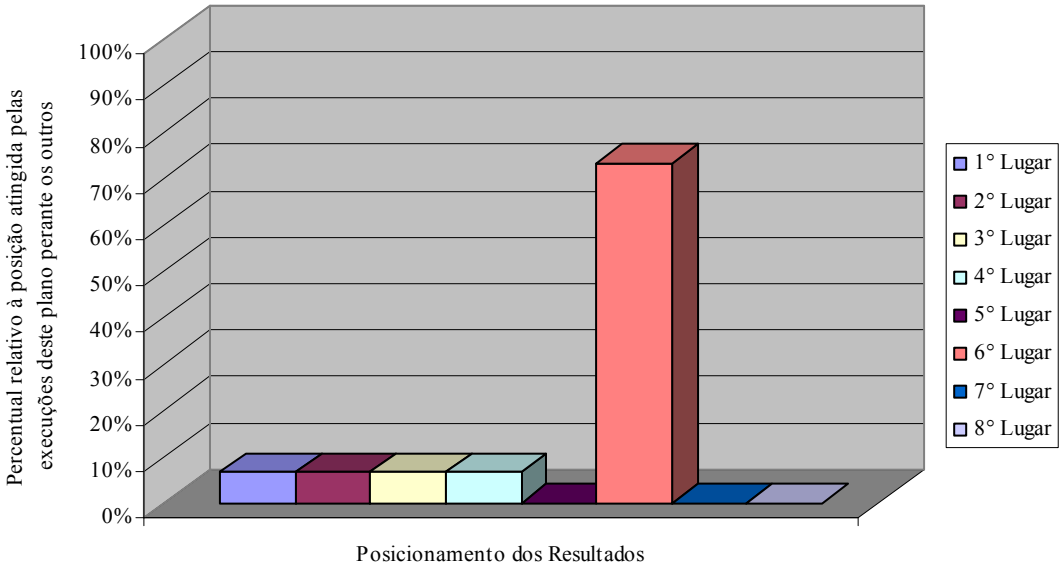

Figura 4.29 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com Moda da base* Car Evalution

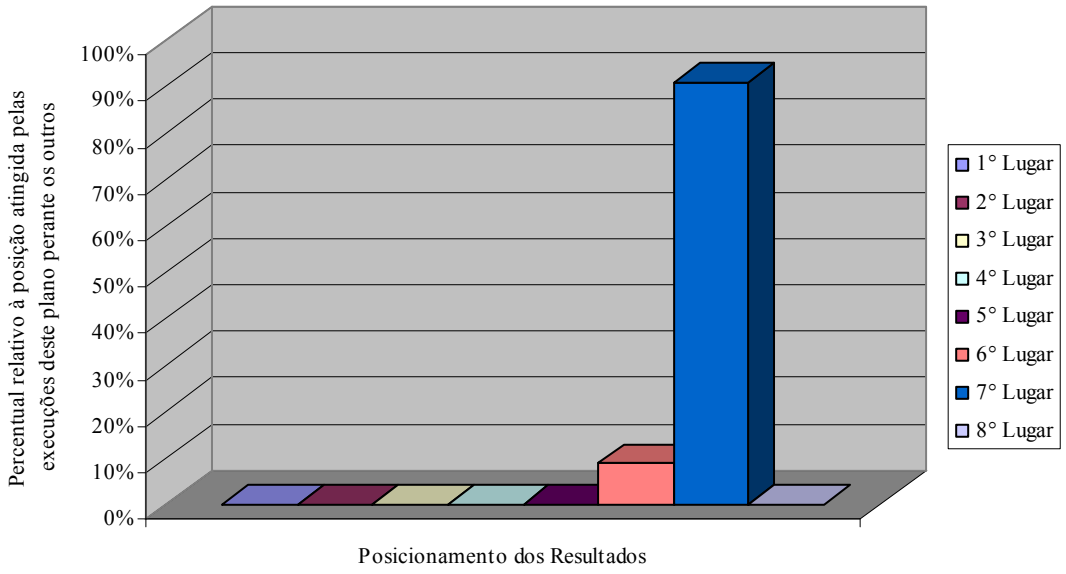

Figura 4.30 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com Moda da base* Tic-Tac-Toe Endgame

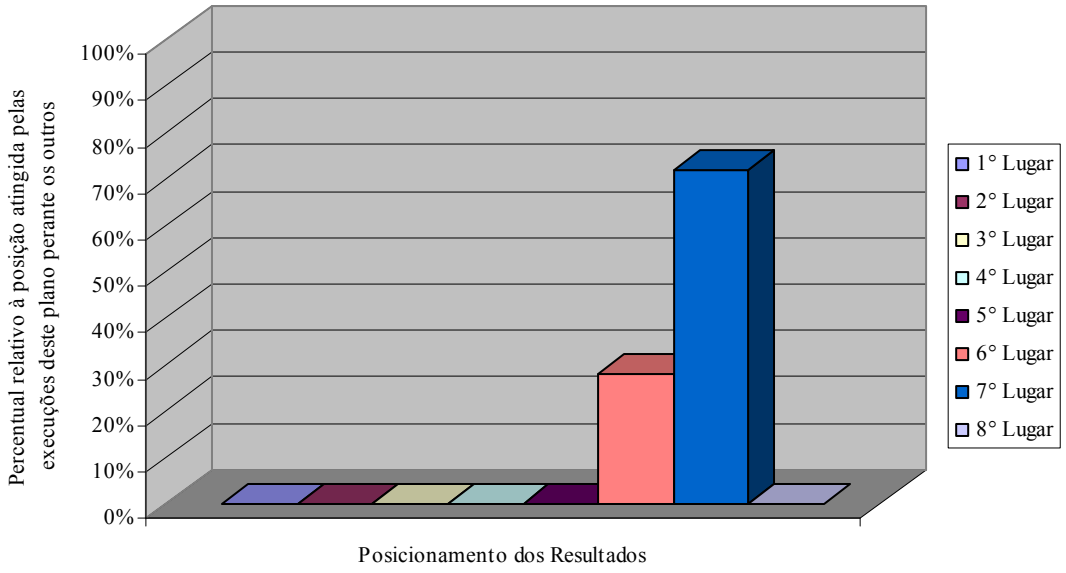

Figura 4.31 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com Moda da base* Teaching Assistant Evaluation

Plano 3: Seleção com Algoritmo Genético, Agrupamento com CK-Means *e Imputação com Moda*

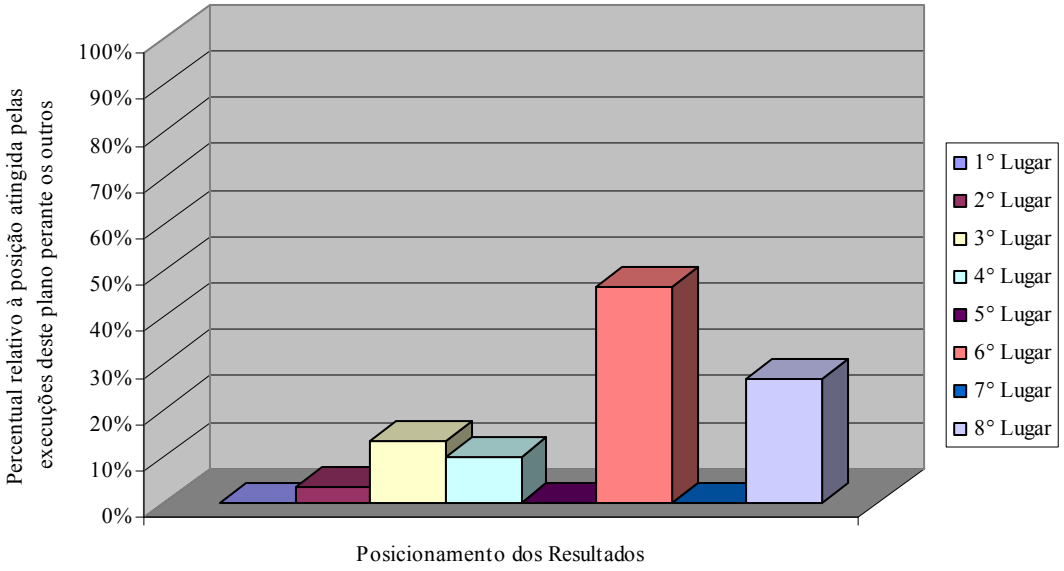

Figura 4.32 - Classificação dos resultados para o plano de Seleção com Algoritmos Genéticos, Agrupamento com CK-Means *e Imputação com Moda da base* Car Evaluation

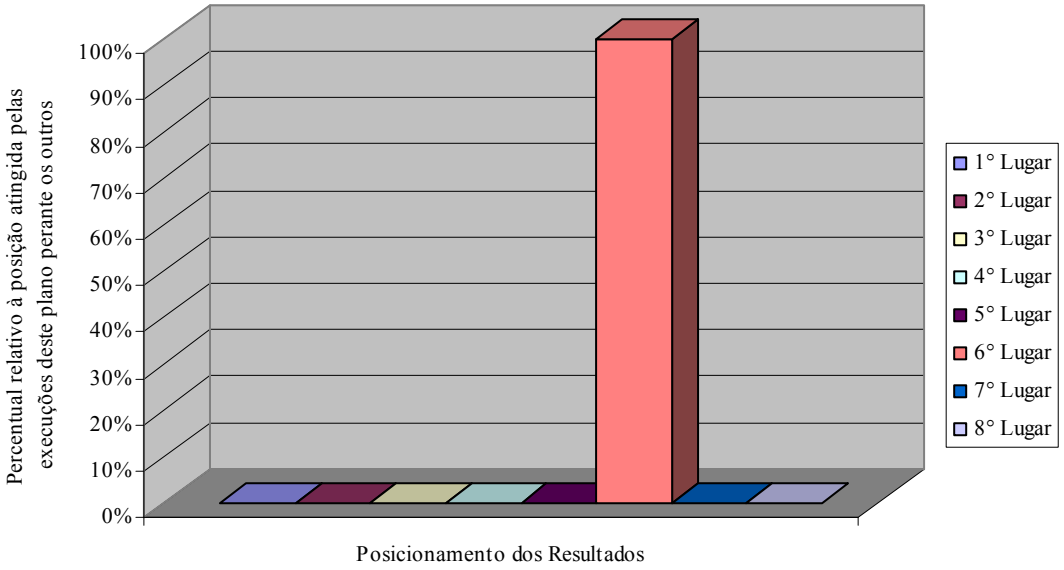

Figura 4.33 - Classificação dos resultados para o plano de Seleção com Algoritmos Genéticos, Agrupamento com CK-Means *e Imputação com Moda da base* Tic-Tac-Toe Endgame

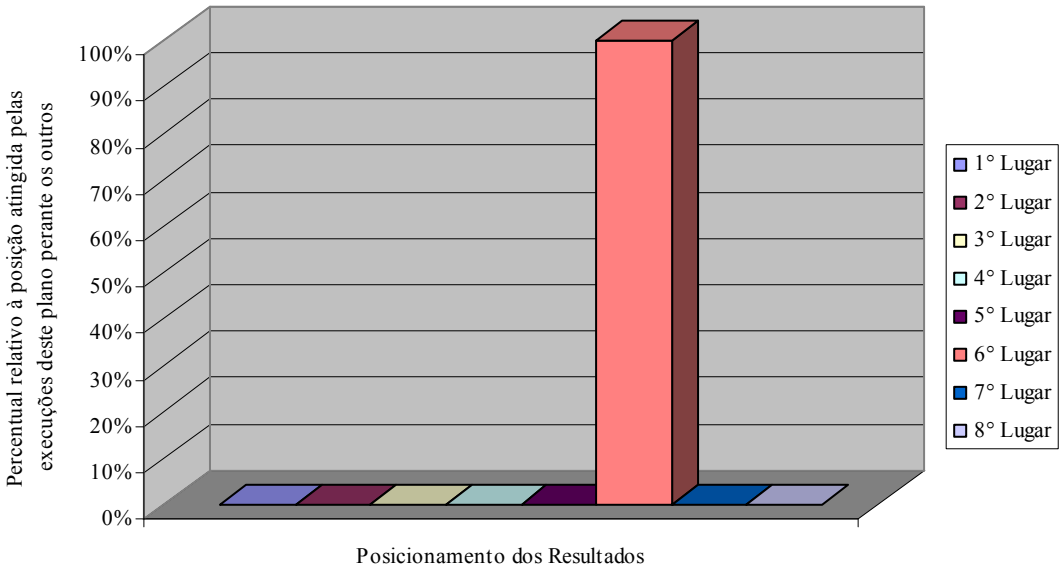

Figura 4.34 - Classificação dos resultados para o plano de Seleção com Algoritmos Genéticos, Agrupamento com CK-Means *e Imputação com Moda da base* Teaching Assisant Evaluation

Plano 4: Agrupamento com CK-Means*, Seleção com Algoritmo Genético e Imputação com Moda*

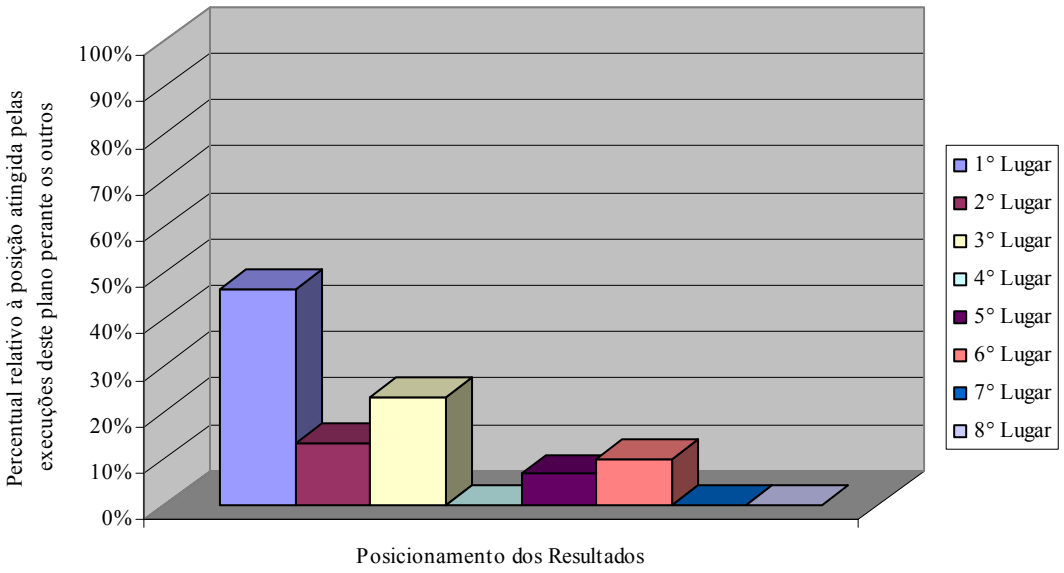

Figura 4.35 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmos Genéticos e Imputação com Moda da base* Car Evaluation

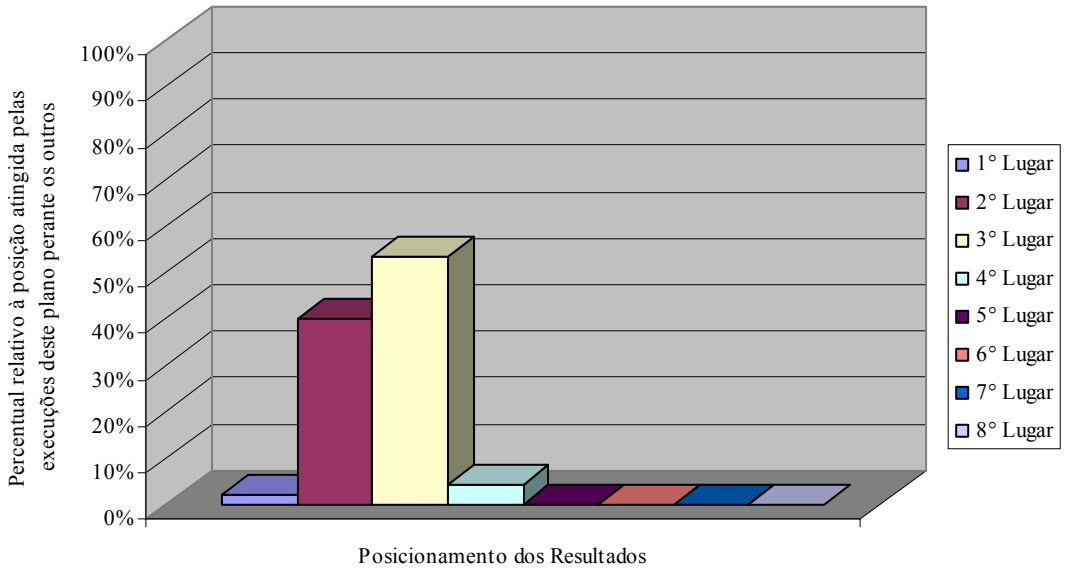

Figura 4.36 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmos Genéticos e Imputação com Moda da base* Tic-Tac-Toe Endgame

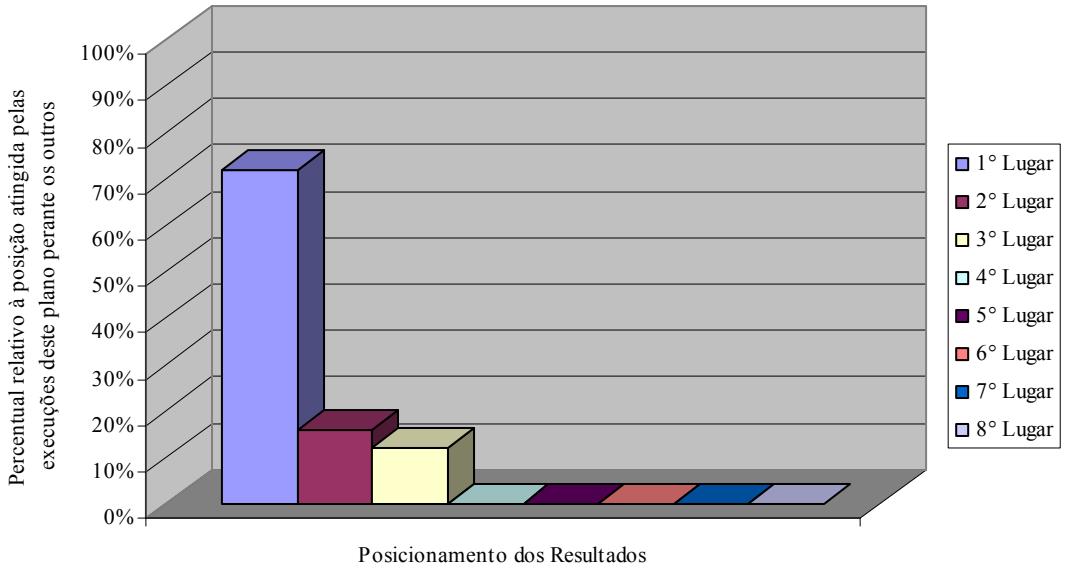

Figura 4.37 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmos Genéticos e Imputação com Moda da base* Teaching Assistant Evaluation

Plano 5: Imputação com ck-*NN*

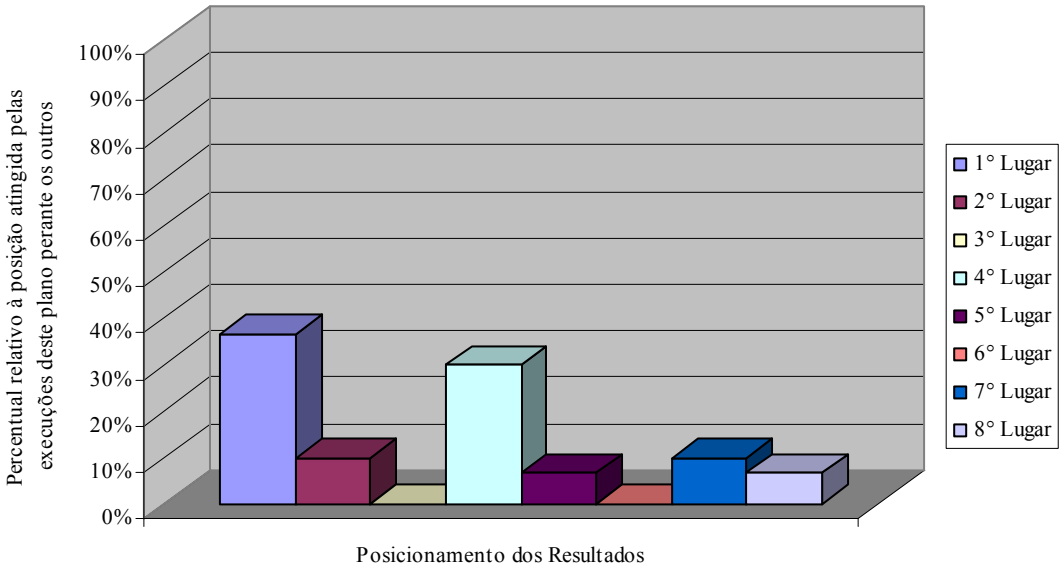

Figura 4.38 - Classificação dos resultados para o plano de Imputação com ck-*NN da base* Car Evaluation

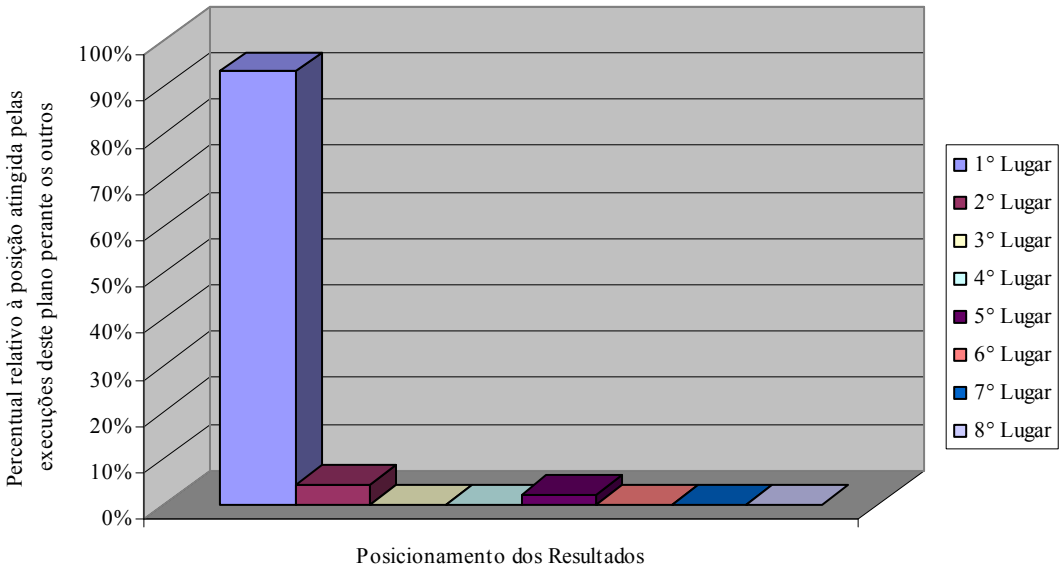

Figura 4.39 - Classificação dos resultados para o plano de Imputação com ck-*NN da base* Tic-Tac-Toe Endgame

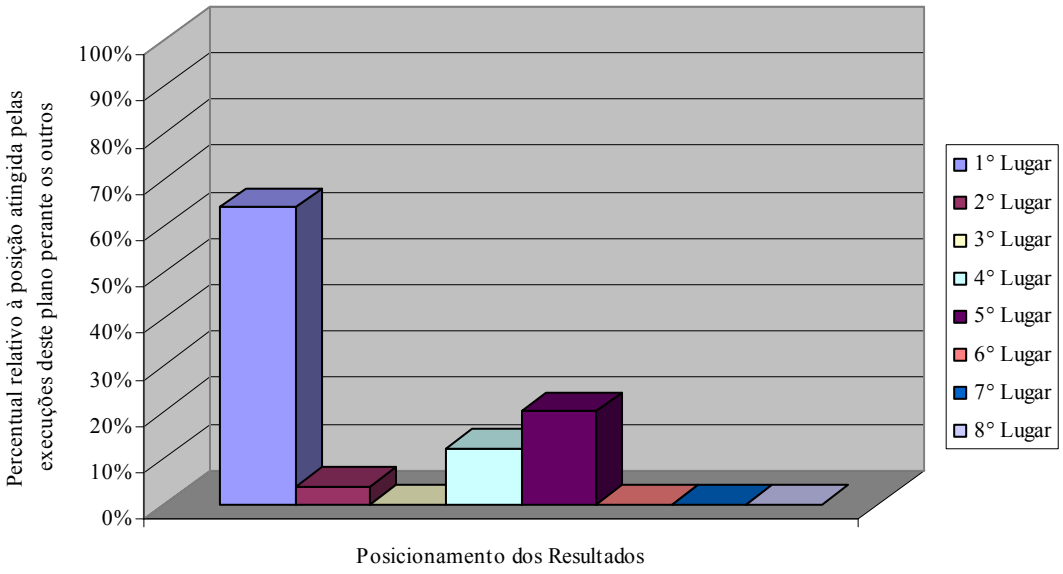

Figura 4.40 - Classificação dos resultados para o plano de Imputação com ck-*NN da base* Teaching Assistant Evaluation

Plano 6: Agrupamento com CK-Means *e Imputação com* ck-*NN*

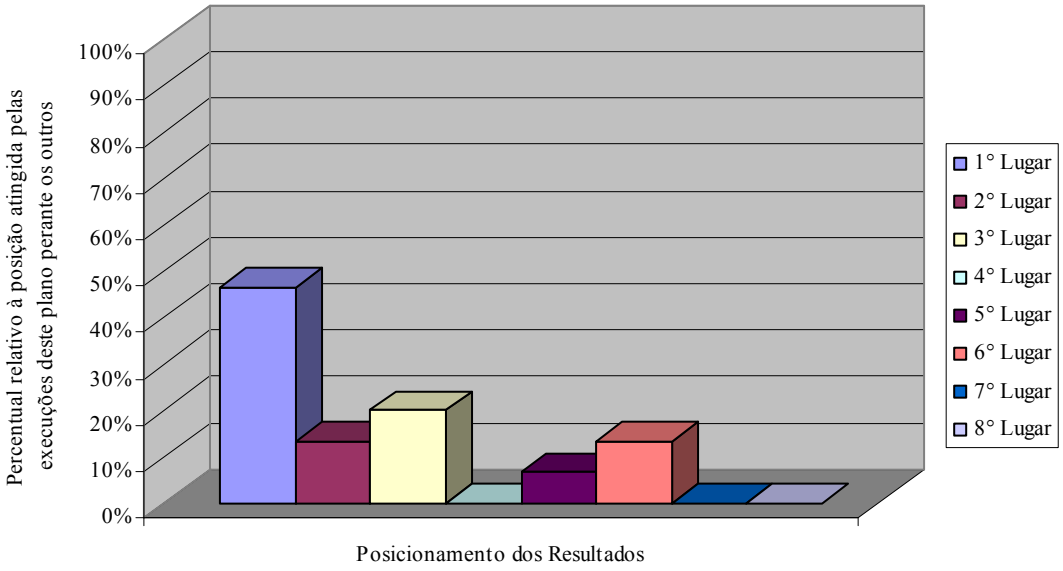

Figura 4.41 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Car Evaluation

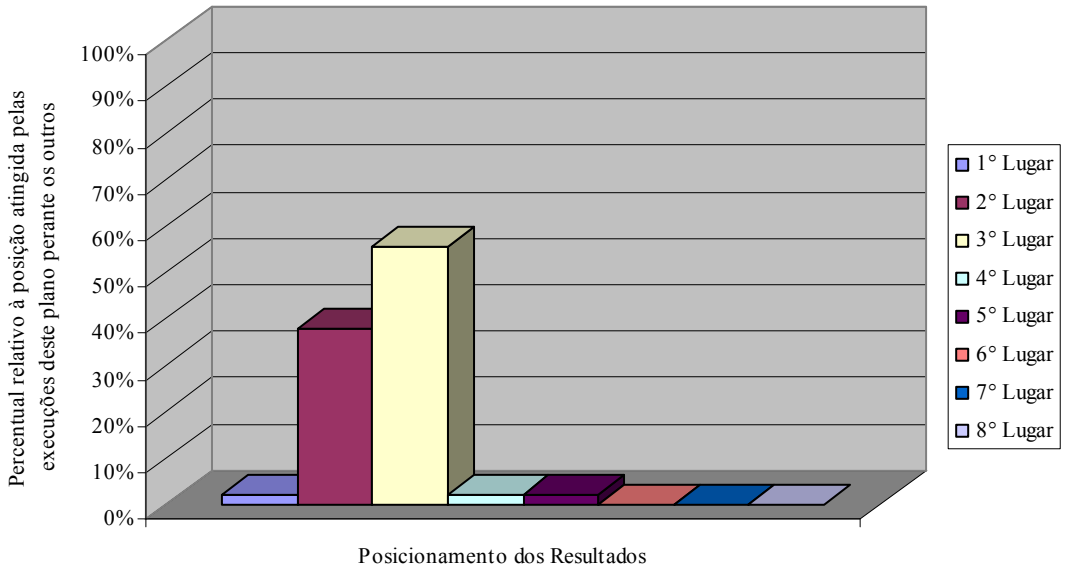

Figura 4.42 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Tic-Tac-Toe-Endgame

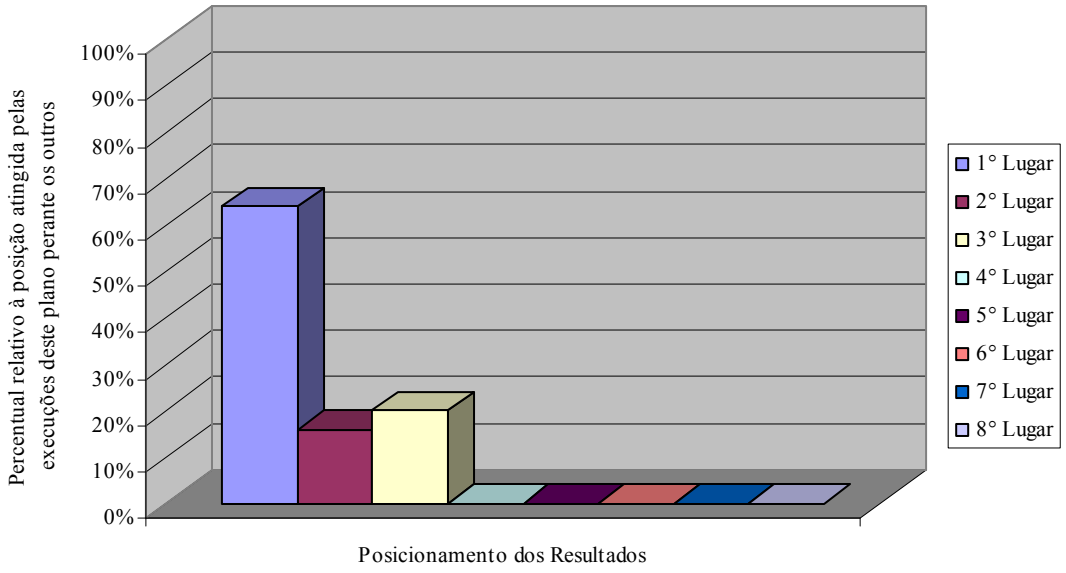

Figura 4.43 - Classificação dos resultados para o plano de Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Teaching Assistant Evaluation

Plano 7: Seleção com Algoritmo Genético, Agrupamento com CK-Means *e Imputação com* ck-*NN*

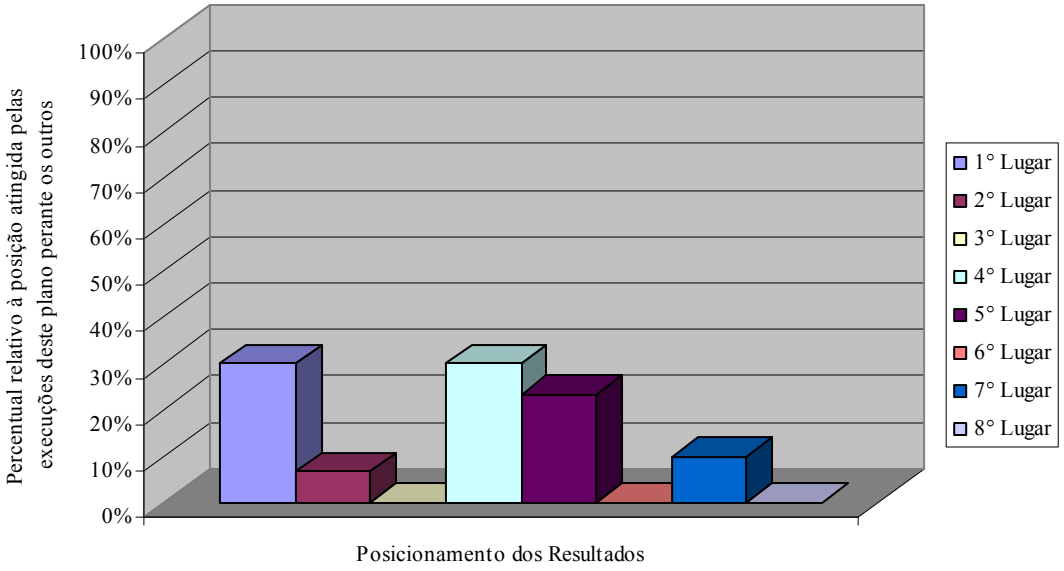

Figura 4.44 - Classificação dos resultados para o plano de Seleção com Algoritmo Genético, Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Car Evaluation

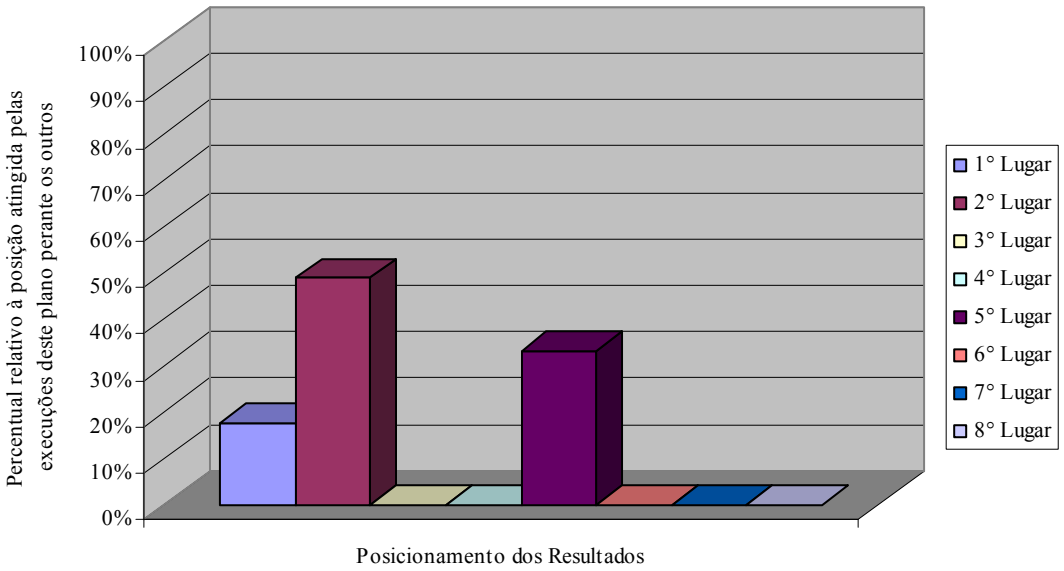

Figura 4.45 - Classificação dos resultados para o plano de Seleção com Algoritmo Genético, Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Tic-Tac-Toe Endgame

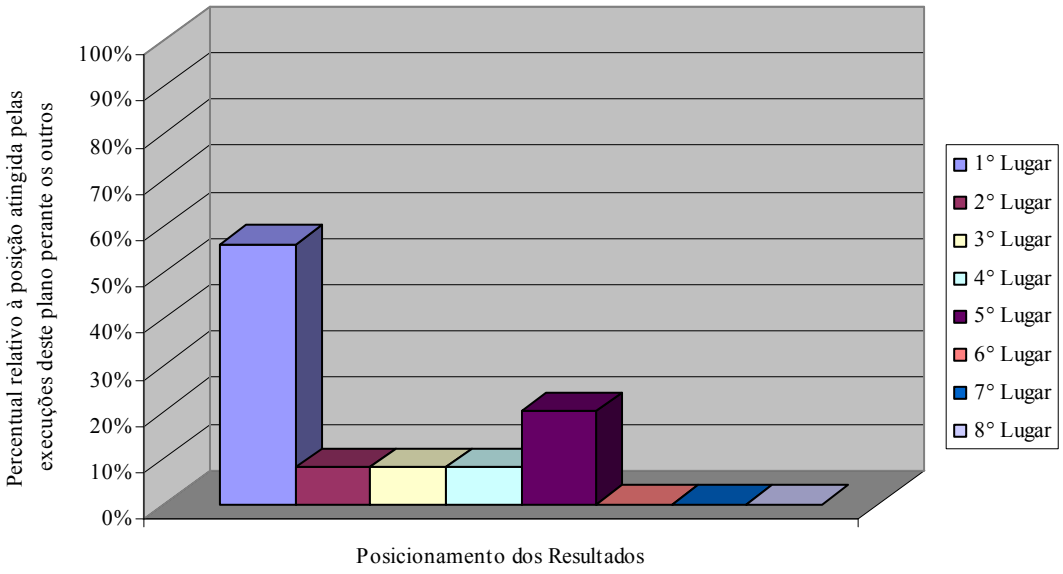

Figura 4.46 - Classificação dos resultados para o plano de Seleção com Algoritmo Genético, Agrupamento com CK-Means *e Imputação com* ck-*NN da base* Teaching Assistant Evaluation

Plano 8: Agrupamento com CK-Means*, Seleção com Algoritmo Genético e Imputação com* ck-*NN*

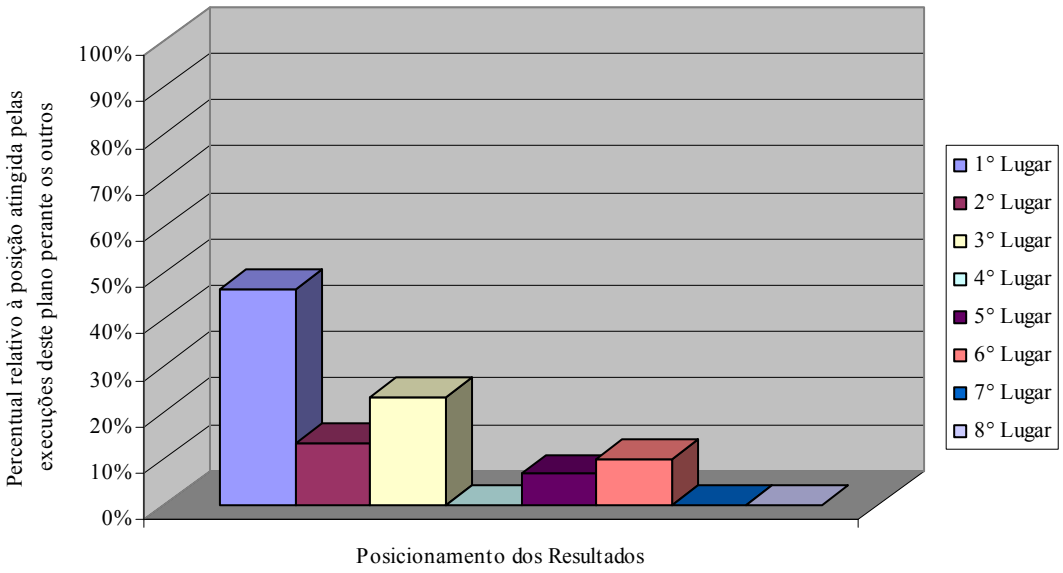

Figura 4.47 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmo Genético e Imputação com* ck-*NN da base* Car Evaluation

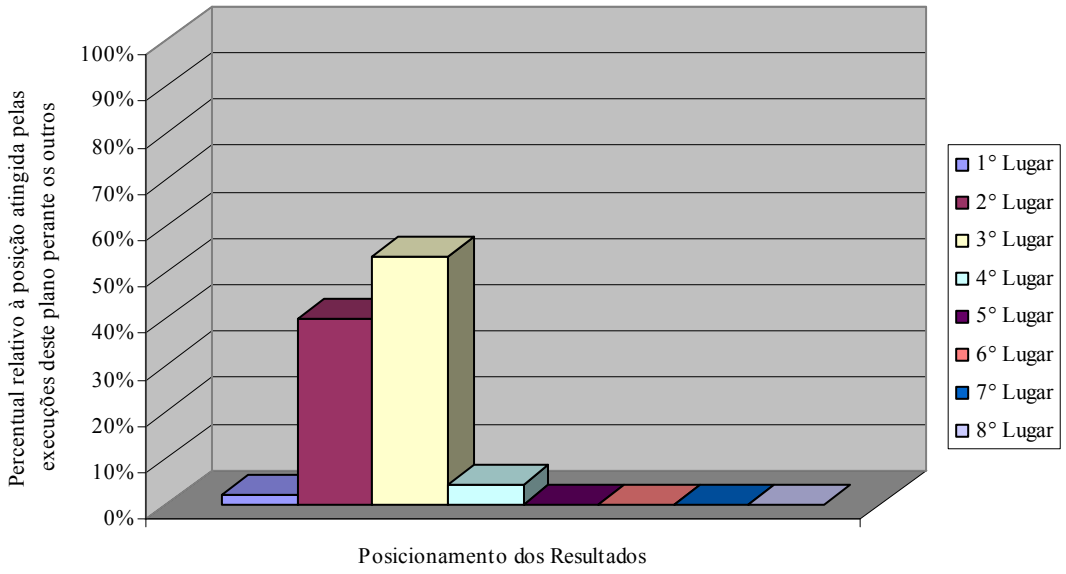

Figura 4.48 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmo Genético e Imputação com* ck-*NN da base* Tic-Tac-Toe Endgame

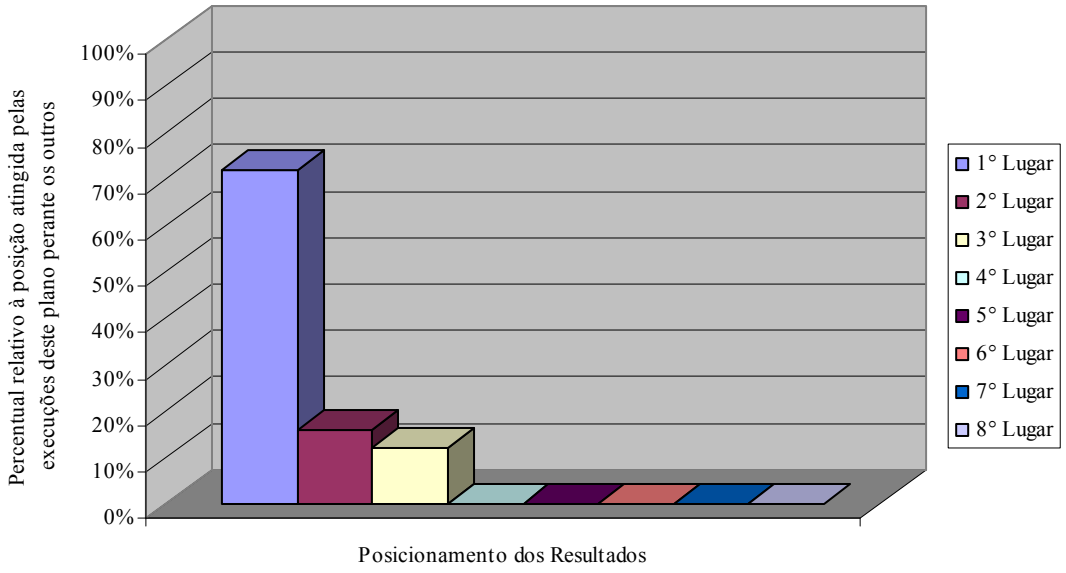

Figura 4.49 - Classificação dos resultados para o plano de Agrupamento com CK-Means*, Seleção com Algoritmo Genético e Imputação com* ck-*NN da base* Teaching Assistant Evaluation

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

5.1 Resumo do Trabalho

É importante perceber que a qualidade dos dados possui grande influência na qualidade dos modelos de conhecimento a serem abstraídos a partir desses dados. Quanto pior for a qualidade dos dados informados ao processo de Descoberta de Conhecimento em Bases de Dados, pior será a qualidade dos modelos de conhecimento gerados (GIGO – *Garbage in, Garbage out*) (GOLDSCHMIDT, PASSOS, 2005).

Neste contexto, apresentamos neste projeto, o problema de imputação categórica em bases de dados, e destacamos a sua importância no processo de descoberta de conhecimento, bem como algumas áreas de sua aplicação. Dados ausentes podem comprometer decisões tomadas ou levar-nos a criação de padrões errados, ou seja, podem ser extremamente prejudiciais ao processo.

Analisando as soluções propostas na literatura pesquisada, vemos uma grande dificuldade em se imputar dados de natureza categórica, já que a grande maioria dos trabalhos encontrados explora apenas os dados de natureza numérica.

Propusemos-nos nesse trabalho a avaliar e testar mecanismos voltados para a imputação de dados de natureza categórica e para isso utilizamos a técnica de imputação composta categórica. Essa técnica foi baseada em prévios estudos feitos e consiste no conceito de estratégias de complementação de dados ausentes, onde cada estratégia reflete a aplicação seqüenciada das tarefas de seleção de atributos e agrupamento de dados, precedendo o processo de complementação de dados.

Para materializar as idéias da imputação composta categórica, utilizamos um sistema chamado *Appraisal* como base para nossa implementação. O sistema que desenvolvemos, assim como o original, implementa a imputação com estratégias e planos de imputação, mas também avalia a qualidade do dado imputado, do ponto de vista do valor de erro gerado (o quão distante ele está do valor original).

Nessa abordagem, realizamos vários testes sobre dados com características diferentes, tanto na quantidade de registros quando na quantidade de colunas. As bases utilizadas foram: *Car Evaluation*, *Tic-Tac-Toe Endgame* e *Teaching Assistant Evaluation.* 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, na base *Car Evaluation*, a estratégia composta de agrupamento de dados seguida da imputação é a que mostrou melhores resultados. Na base *Teaching Assistant Evaluation*, a estratégia vencedora foi a do agrupamento precedendo a seleção e a imputação. Já na base *Tic-Tac-Toe Endgame*, a estratégia de imputação simples mostrou-se a melhor solução. Nesta última, os resultados devem-se provavelmente ao fato da baixa correlação entre os atributos.

A imputação com o algoritmo dos *k* vizinhos mais próximos, em relação à moda, 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.

5.2 Contribuições do Trabalho

Apontamos nesta seção, algumas contribuições dos estudos realizados neste projeto:

1) Compilação de algumas soluções propostas na literatura para a resolução do problema de ausência de dados de natureza categórica em base de dados, a imputação categórica;
2) Proposta e desenvolvimento de um componente de imputação de dados categóricos, que foi incorporado ao *Appraisal* (SOARES, 2007). Esse componente implementa a técnica de imputação composta categórica, proposta inicialmente para tratar da imputação composta de dados numéricos;
3) A criação de novos métodos de seleção de atributos categóricos e de agrupamento de dados categóricos, que são utilizados durante o processo de Descoberta de Conhecimento em Bases de Dados com a missão de melhorar a qualidade dos dados que serão imputados;
4) A verificação, também baseada no escopo dos testes feitos, de que o percentual de valores ausentes em um conjunto de dados de natureza categórica não afeta a qualidade da classificação dos dados, assim como verificado por SOARES (2007) em relação aos dados de natureza numérica;
5) A observação, através dos experimentos realizados neste projeto, de que as estratégias compostas produzem dados a serem imputados de melhor qualidade (com um menor índice de erros), exceto em bases onde os atributos possuam uma baixa correlação.

5.3 Trabalhos Futuros

Neste projeto, podemos apresentar, após os estudos apresentados, uma série de questões que podem ser aprimoradas.

Um passo que se vê necessário é o estudo de outras técnicas de seleção, agrupamento e imputação para dados de natureza categórica. Seria interessante vermos testes com a utilização, por exemplo, de técnicas de *Redes Neuronais* no tratamento de dados categóricos.

A avaliação de novos mecanismos de cálculo de distância e novas formas de medição da taxa de erro dos planos de imputação é vista com extrema importância como tentativa de melhor expressar suas similaridades.

Uma alternativa interessante seria a utilização de técnicas de codificação *categórica → numérica*, como por exemplo, a representação binária por temperatura, que consiste em representar os *N* valores de um domínio em *N* bits. Com os dados representados em bits, podemos calcular a distância entre os atributos categóricos através da *Distância de Hamming*, que consiste em verificar para cada posição *i* da cadeia de bits, se eles são iguais ou diferentes. Se forem diferentes, soma-se uma unidade a um contador (GOLDSCHMIDT, PASSOS, 2005).

Na tabela 5.1 podemos ver como funciona essa representação.

Tabela 5.1 – Codificação de valores categóricos com a representação binária por temperatura

Excelente	00001
Bom	00011
Regular	00111
Ruim	01111
Péssimo	11111

Neste exemplo, o valor calculado para a *Distância de Hamming* entre os valores "*Bom*" e "*Ruim*" é igual a 2, já que o segundo e o terceiro bit destes valores são diferentes.

Poderíamos sugerir ainda o desenvolvimento do módulo *Reviewer* para ser utilizado com dados categóricos e medir a qualidade dos dados imputados pelo módulo *Crowner*, através de métodos de reclassificação das tuplas imputadas e de métricas para medição do erro de imputação.

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

AHA, D. W., KIBLER, D., ALBERT, M., 1991, "*Instance-based Learning Algorithms*", *Machine Learning*, v. 6, pp. 37-66.

ALLISON, P. D., 2005, *"Imputation of Categorical Variables with PROC MI",* SUGI 30 Proceedings

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., 2003, "*An Analysis of Four Missing Data Treatment Methods for Supervised Learning*", *Applied Artificial Intelligence*, v. 17, n. 5 (May-Jun), pp. 519-533.

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.

CONDE, B., 2005, "*Seleção de Variáveis na Descoberta de Conhecimento em Bases de Dados*", NUPAC/UNIVERCIDADE, Rio de Janeiro, RJ, Brasil.

DASARATHY, B., 1990, "*Nearest Neighbor (NN) norms: NN pattern classification techniques"*, 1 ed., *IEEE Computer Society Press*, Los Alamitos.

ELMASRI, R. NAVATHE, S. B., 2002, "*Sistemas de Banco de Dados Fundamentos e Aplicações*". LTC, São Paulo.

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*.

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., PASSOS, E., 2005, "*Data Mining: Um Guia Prático"*. 1 ed., Editora Campus

HE, Z., 2006, "*Approximation Algorithms for K-Modes Clustering*", *Harbin Institute of Technology,* Harbin, China

HUISMAN, M., 2000, "*Imputation of Missing Item Responses: Some Simple Techniques*", *Quality and Quantity*, v. 34, n. 4, pp. 331-351, Nov.

INMON, W. H., 1993, "*Building the Data Warehousing*". John Wiley & Sons.

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.

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.

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.

LARSEN , M., 2006, "*Fractional Imputation for Categorical Data*", Department of Statistics, Iowa State University

LEI, M., HE, P., LI, Z., 2006, "*An Improved K-means Algorithm for Clustering Categorical Data*", Journal of Communication and Computer, v. 3, n. 8.

LITTLE, R. J. A., RUBIN, D. B., 1987, "*Statistical Analysis with Missing Data"*. John Wiley and Sons, New York.

LIU, H., SETIONO, R. 1996, "*A Probabilistic Approach to Feature Selection - A Filter Solution"*. In: *Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann.

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.

MACQUEEN, 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.

MITCHELL, T. M., 1997, *Machine Learning*. Ed. McGraw-Hill.

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*.

POE, V., 1996, "*Building a Data Warehousing for Decision Support"*. Prentice-Hall

RUBIN, D. B., 1988, "*An Overview of Multiple Imputation*", In: *Proceedings of the Section on Survey Research Methods*, pp. 79-84, American Statistical Association.

SENTAS, P., ANGELIS, L., 2006, *"Categorical missing data imputation for software cost estimation by multinomial logistic regression", The Journal of Systems and Software*

SILVA, E. B., 2006*, "Agrupamento Semi-Supervisionado de Documentos XML".* Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

SOARES, J., 2007, "*Pré-processamento em Mineração de Dados: Um Estudo Comparativo em Complementação".* Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

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. 6, pp. 520-525.

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.

YANG, J., HONAVAR, V., 1998, "*Feature Subset Selection Using a Genetic Algorithm"*. *IEEE Expert*