Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Experimental Comparison of k-Nearest Neighbour and Mean or Mode Imputation Methods with the Internal Strategies used by C4.5 and CN2 to Treat Missing Data

Gustavo E.A.P.A. Batista
Maria Carolina Monard

Nº 186

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos

Fev./2003

Experimental Comparison of K-Nearest Neighbour and Mean or Mode Imputation Methods with the Internal Strategies used by C4.5 and CN2 to Treat Missing Data*

Gustavo E.A.P.A. Batista and Maria Carolina Monard
University of São Paulo at São Carlos - USP
Institute of Mathematics and Computer Science - ICMC
Department of Computer Science and Statistics - SCE
Laboratory of Computational Intelligence - LABIC
P. O. Box 668, 13560-970 - São Carlos, SP, Brazil
{gbatista, mcmonard}@icmc.sc.usp.br

March 1, 2003

Abstract

Data quality is a major concern in Machine Learning and other correlated areas such as Knowledge Discovery from Databases (KDD). As most Machine Learning algorithms induce knowledge strictly from data, the quality of the knowledge extracted is largely determined by the quality of the underlying data. One relevant problem in data quality is the presence of missing data. Despite the frequent occurrence of missing data, many Machine Learning algorithms handle missing data in a rather naive way. Missing data treatment should be carefully thought, otherwise bias might be introduced into the knowledge induced. In this work, we analyse the use of the k-nearest neighbour algorithm as an imputation method. Imputation is a term that denotes a procedure that replaces the missing values in a data set by some plausible values. Our analysis indicates that missing data imputation based on the k-nearest neighbour algorithm can outperform the internal methods used by C4.5 and CN2 to treat missing data, and can also outperform the mean or mode imputation method, which is a method broadly used to treat missing data.

Keywords: Missing data treatment, imputation, supervised learning, machine learning, data mining

*This research is partially supported by Brazilian Research Councils CAPES and FAPESP.

Contents

List of Figures

List of Tables

1 Introduction

Data quality is a major concern in Machine Learning and other correlated areas such as Knowledge Discovery from Databases (KDD). As most Machine Learning algorithms induce knowledge strictly from data, the quality of the knowledge extracted is largely determined by the quality of the underlying data.

One relevant problem in data quality is the presence of missing data. Missing data may have different sources such as death of patients, equipment malfunctions, refusal of respondents to answer certain questions, and so on. Despite the frequent occurrence of missing data, many Machine Learning algorithms handle missing data in a rather naive way. Missing data treatment should be carefully thought, otherwise bias might be introduced into the knowledge induced.

In most cases, data sets attributes are not independent from each other. Thus, through the identification of relationships among attributes, missing values can be determined. *Imputation* is a term that denotes a procedure that replaces the missing values in a data set by some plausible values. One advantage of this approach is that the missing data treatment is independent of the learning algorithm used. This allows the user to select the most suitable imputation method for each situation.

The objective of this work is to analyse the performance of the K-NEAREST NEIGHBOUR algorithm as an imputation method, comparing its performance with other three missing data treatment methods. The first method is the MEAN OR MODE IMPUTATION. This method is very simple and broadly used. It consists of replacing every missing value of an attribute by the mean (if the attribute is quantitative) or mode (if the attribute is qualitative) of its known values. The other two methods are the internal missing data treatment strategies used by two well known Machine Learning algorithms: CN2 (Clark & Niblett 1989) and C4.5 (Quinlan 1988).

This work is organized as follows: Section 2 describes the taxonomy proposed by (Little & Rubin 1987) to classify the degree of randomness of missing data in a data set; Section 3 surveys the most used methods for missing data treatment; Section 4 is dedicated to a specific class of missing data treatment methods: imputation; Section 5 presents the K-NEAREST NEIGHBOUR algorithm as an imputation method for treating missing values, and the variant of K-NEAREST NEIGHBOUR implemented in this work; Section 6 describes how the Machine Learning algorithms C4.5 and CN2 treat missing data internally; Section 7 performs a comparative study with the K-NEAREST NEIGHBOUR algorithm as an imputation method, the internal methods used by C4.5 and CN2 to treat missing data, as well as the MEAN OR MODE IMPUTATION; finally, Section 8 presents the conclusions of this work.

2 Randomness of Missing Data

Missing data randomness can be divided into three classes, as proposed by (Little & Rubin 1987):

1. *Missing completely at random (MCAR).* This is the highest level of randomness. It occurs when the probability of an instance (case or example) having a missing value for an attribute does not depend on either the known values or the missing data. In this

level of randomness, any missing data treatment method can be applied without risk of introducing bias on the data;

2. *Missing at random (MAR).* When the probability of an instance having a missing value for an attribute may depend on the known values, but not on the value of the missing data itself;

3. *Not missing at random (NMAR).* When the probability of an instance having a missing value for an attribute may depend on the value of that attribute.

3 Methods for Treating Missing Data

There are several methods for treating missing data available in the literature. Many of these methods, such as case substitution, were developed for dealing with missing data in sample surveys, and have some drawbacks when applied to the Data Mining context of the KDD process. Other methods, such as replacement of missing values by the attribute's mean or mode, are very naive and should be carefully used to avoid insertion of bias.

In a general way, missing data treatment methods can be divided into three categories, as proposed in Little & Rubin (1987):

1. *Ignoring and discarding data.* There are two main ways to discard data with missing values. The first one is known as *complete case analysis*, it is available in all statistical programs and is the default method in many programs. This method consists of discarding all instances (cases) with missing data. The second method is known as *discarding instances and/or attributes*. This method consists of determining the extent of missing data on each instance and attribute, and deleting the instances and/or attributes with high levels of missing data. Before deleting any attribute, it is necessary to evaluate its relevance to the analysis. Unfortunately, relevant attributes should be kept even with a high degree of missing values. Both methods, complete case analysis and discarding instances and/or attributes, should be applied only if missing data are MCAR, because missing data that are not MCAR have non-random elements that can bias the results;

2. *Parameter estimation.* Maximum likelihood procedures are used to estimate the parameters of a model defined for the complete data. Maximum likelihood procedures that use variants of the Expectation-Maximization algorithm (Dempster, Laird & Rubin 1977) can handle parameter estimation in the presence of missing data;

3. *Imputation.* Imputation is a class of procedures that aims to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. This papers focus on imputation of missing data. More details about this class of methods are described next.

4 Imputation Methods

Imputation methods involve replacing missing values with estimated ones based on some information available in the data set. There are many options varying from naive methods like MEAN OR MODE IMPUTATION to some more robust methods based on relationships among attributes.

This section surveys some widely used imputation methods, although other forms of imputation are available.

1. CASE SUBSTITUTION. This method is typically used in sample surveys. One instance with missing data (for example, a person that cannot be contacted) is replaced by another non-sampled instance;

2. MEAN OR MODE IMPUTATION. One of the most frequently used methods. This method consists of replacing the missing data for a given attribute by the mean (quantitative attribute) or mode (qualitative attribute) of all known values of that attribute;

3. HOT DECK and COLD DECK. In the hot deck method, a missing attribute value is filled in with a value from an estimated distribution for the missing value from the current data. Hot deck is typically implemented into two stages. In the first stage, the data are partitioned into clusters. And, in the second stage, each instance with missing data is associated with one cluster. The complete cases in a cluster are used to fill in the missing values. This can be done by calculating the mean or mode of the attribute within a cluster. Cold deck imputation is similar to hot deck but the data source must be other than the current data source;

4. PREDICTION MODEL. Prediction models are sophisticated procedures for handling missing data. These methods consist of creating a predictive model to estimate values that will substitute the missing data. The attribute with missing data is used as class-attribute, and the remaining attributes are used as input for the predictive model. An important argument in favor of this approach is that, frequently, attributes have relationships (correlations) among themselves. In this way, those correlations could be used to create a predictive model for classification or regression (depending on the attribute type with missing data, being, respectively, qualitative or quantitative). Some of these relationships among the attributes may be maintained if they were captured by the predictive model. An important drawback of this approach is that the model estimated values are usually more well-behaved than the true values would be, *i.e.*, since the missing values are predicted from a set of attributes, the predicted values are likely to be more consistent with this set of attributes than the true (not known) value would be. A second drawback is the requirement for correlation among the attributes. If there are no relationships among one or more attributes in the data set and the attribute with missing data, then the model will not be precise to estimate the missing values.

5 Imputation with the K-NEAREST NEIGHBOUR Algorithm

In this work we propose the use of K-NEAREST NEIGHBOUR algorithm to estimate and substitute missing data. The main benefits of this approach are:

- The K-NEAREST NEIGHBOUR algorithm can predict both discrete attributes (the most frequent value among the k nearest neighbours) and continuous attributes (the mean among the k nearest neighbours);
- There is no necessity for creating a predictive model for each attribute with missing data. Actually, the K-NEAREST NEIGHBOUR algorithm does not create explicit models (like a decision tree or a set of rules), once the data set is used as a "lazy" model. Thus, the K-NEAREST NEIGHBOUR algorithm can be easily adapted to work with any attribute as class, by just modifying which attributes will be considered in the distance function. Also, this approach can easily treat examples with multiple missing values.

The main drawback of this approach is that whenever the K-NEAREST NEIGHBOUR algorithm looks for the most similar instances, the algorithm searches through all the data set. This limitation can be very critical for KDD applications, since this research area has, as one of its main objectives, the analysis of large databases. Several works that aim to solve this limitation can be found in the literature. One method is the creation of a reduced training set for the K-NEAREST NEIGHBOUR algorithm composed only by prototypical examples (Wilson & Martinez 2000). In this work we use an access method called M-tree (Ciaccia, Patella & Zezula 1997), that we have implemented in our K-NEAREST NEIGHBOUR algorithm. M-trees can organize and search data sets based on a generic metric space. M-trees can drastically reduce the number of distance computations in similarity queries.

Before we discuss the empirical analysis of missing data imputation with K-NEAREST NEIGHBOUR, we present a brief description of the K-NEAREST NEIGHBOUR algorithm and the variant that we have implemented in our work.

5.1 The K-NEAREST NEIGHBOUR Algorithm

The K-NEAREST NEIGHBOUR algorithm is part of a family of learning methods known as *instance-based* (Aha, Kibler & Albert 1991) (Mitchell 1997, Chapter 8). Instance-based learning methods are conceptually straightforward approaches to learn either quantitative or qualitative class attributes. Learning in these algorithms consists of simply storing the training data. When a new instance is presented, a set of similar related instances is retrieved from the training data and used to classify the new instance.

As previously described, one disadvantage of instance-based approaches is that the cost of classifying new instances can be high. This is due to the fact that nearly all computation takes place at classification time rather that when the training examples are first encountered. Therefore, techniques for efficiently indexing training examples are a significant practical issue in reducing the computation required in classification time. A second disadvantage of this approach is that it typically consider all attributes of the instances when attempting to retrieve similar training examples. If the target concept depends on only a few of many available attributes, then the instances that are truly most "similar" may be a large distance apart.

The next sections discuss the K-NEAREST NEIGHBOUR algorithm in more details. Section 5.1.1 describes the most basic and widely used implementation of the K-NEAREST NEIGHBOUR algorithm. Section 5.1.2 describes an improvement over the basic algorithm by weighting the vote of

Training algorithm:

- For each training example $\langle x, f(x)\rangle$, add the example to the list *training_examples*

Classification algorithm:

- Given a query instance x_q to be classified,
  - Let $x_1, \ldots x_k$ denote the k instances from *training_examples* that are nearest to x_q
  - Return

$$\hat{f}(x_q) = \max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$$

where $\delta(a, b) = 1$ if $a = b$ and where $\delta(a, b) = 0$ otherwise.

Table 1: The basic K-NEAREST NEIGHBOUR algorithm for approximating discrete-valued functions.

each instance according to their distance from the query example. Section 5.1.3 describes some distance functions that can outperform the Euclidean distance. Finally, Section 5.1.4 briefly describes the M-tree access method used in this work to make the query computation faster.

5.1.1 The Basic K-NEAREST NEIGHBOUR Algorithm

The basic version of the K-NEAREST NEIGHBOUR algorithm assumes that all instances correspond to points in the n-dimensional space $\Re^n$. The nearest neighbours of an instance are often defined in terms of the standard Euclidean distance. More precisely, let us describe an arbitrary instance by the vector $\langle x, f(x)\rangle$. f is the true concept function that gives the correct class value $f(x)$ for each instance x. The instance x can also be described by the feature vector

$$\langle a_1(x), a_2(x), \ldots a_n(x)\rangle$$

where $a_r(x)$ denotes the value of the rth attribute of instance x.

The distance between two instances x_i and x_j is denoted as $d(x_i, x_j)$. For the standard Euclidean distance, $d(x_i, x_j)$ is defined by Equation 1.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{1}$$

In nearest neighbour learning the concept function can be either discrete-valued (for qualitative class attributes) or real-valued (for quantitative class attributes). Let us first consider learning discrete-valued concept functions of the form $f : \Re^n \to V$, where V is the finite set $\{v_1, \ldots v_s\}$ of class values. The K-NEAREST NEIGHBOUR algorithm is described in Table 1.

The value $\hat{f}(x_q)$ returned by this algorithm as its estimate of $f(x_q)$ is just the mode (most common value) of the true concept function f among k training examples nearest to x_q. If we choose $k = 1$, then the 1-NEAREST NEIGHBOUR algorithm assigns to $\hat{f}(x_q)$ the value $f(x_i)$ where x_i is the training instance nearest to x_q. For larger values of k, the algorithm assigns the mode among the k nearest training exam ples.

The K-NEAREST NEIGHBOUR algorithm is easily adapted to approximating real-valued concept functions. To accomplish this, we have the algorithm calculate the mean value of the k nearest training examples, rather than calculate their most common value. More precisely, to approximate a real-value target function $f : \Re^n \rightarrow \Re$ we replace the final line of the algorithm by the line

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} f(x_i)}{k} \tag{2}$$

5.1.2 Distance-Weighted Nearest Neighbour Algorithm

One refinement to the basic K-NEAREST NEIGHBOUR algorithm is to weight the contribution of each of the k neighbours according to their distance to the query instance x_q, giving greater weight to closer neighbours. For example, in the algorithm of Table 1 on the page before, which approximated discrete-valued concept functions, we might weight the vote of each neighbour according to the inverse square of its distance from x_q. This can be accomplished by replacing the final line of the algorithm by

$$\hat{f}(x_q) = \max_{v \in V} \sum_{i=1}^{k} \omega_i \delta(v, f(x_i)) \tag{3}$$

where

$$\omega_i = \frac{1}{d(x_q, x_i)^2} \tag{4}$$

To accommodate the case where the query point x_q exactly matched one of the training instances x_i and the denominator $d(x_q, x_i)^2$ is therefore zero, we assign $\hat{f}(x_q)$ to be $f(x_i)$ in this case. If there are several such training examples, we assign the majority classification among them.

We can distance-weight the instances for real-valued concept functions in a similar fashion, replacing the final line of the algorithm on Table 1 on the preceding page by

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} \omega_i f(x_i)}{\sum_{i=1}^{k} \omega_i} \tag{5}$$

where ω_i is defined as in Equation 4. Note the denominator in Equation 5 is a constant that normalizes the contributions of the various weights (e.g. it assumes that if $f(x_i) = c$ for all training examples, then $\hat{f}(x_q) = c$ as well).

Note that all of the above variants of the K-NEAREST NEIGHBOUR algorithm consider only k nearest neighbours to classify the query point. Once we add distance weighting, there is really no harm on allowing all training examples to have an influence on the classification of the x_q, because very distant examples will have very little effect on $\hat{f}(x_q)$. The only disadvantage of considering all examples is that our classifier will run more slowly. If all training examples are considered when classifying a new query instance, we call the algorithm a *global* method. If only the nearest training examples are considered, we call it a *local* method. When the rule in Equation 5 on the preceding page is applied as a global method, using all training examples, the method is known as Shepard's method (Shepard 1968).

Our implementation of K-NEAREST NEIGHBOUR algorithm is a local method with weighted-distance classification. In the following sections we describe two improvements we have implemented over this algorithm. First, the substitution of the Euclidean distance by the HVDM distance. Second, the implementation of an index structure known as M-tree to speed up the query computations.

5.1.3 VDM, HEOM and HVDM Distances Functions

The Euclidean distance is widely used and it is well suited for quantitative attributes, but often this distance does not handle qualitative (also called *symbolic*) attributes appropriately.

One way to handle data sets with both qualitative and quantitative attributes is to use a heterogeneous distance function that uses different attribute distance functions on different types of attributes. One approach that has been used is to use the *overlap* metric for qualitative attributes and the normalized Euclidean distance for quantitative attributes. This approach is known as *Heterogeneous Euclidean-Overlap Metric (HEOM)* and defines the distance between two instances x_i and x_j by Equation 6.

$$HEOM(x_i, x_j) = \sqrt{\sum_{r=1}^{n} d_a(a_r(x_i), a_r(x_j))^2} \tag{6}$$

where n is the number of attributes in the data set. $d_a(a_r(x_i), a_r(x_j))$ is the distance between two values $a_r(x_i)$ and $a_r(x_j)$ of a given attribute a_r, and is defined by Equation 7.

$$d_a(a_r(x_i), a_r(x_j)) = \begin{cases} 1, \text{ if } a_r(x_i) \text{ or } a_r(x_j) \text{ is unknown; otherwise} \\ overlap(a_r(x_i), a_r(x_j)), \text{ if } a_r \text{ is qualitative} \\ range\_normalized\_diff(a_r(x_i), a_r(x_j)), \text{ if } a_r \text{ is quantitative} \end{cases} \tag{7}$$

Unknown attribute values are handled by returning an attribute distance value of 1 (*i.e.*, maximal distance) if either of the attribute values is unknown. The function *overlap* and the $range\_normalized\_diff$ are defined by Equation 8 and Equation 9 on the next page, respectively.

$$overlap(a_r(x_i), a_r(x_j)) = \begin{cases} 0, \text{ if } a_r(x_i) = a_r(x_j) \\ 1, \text{ otherwise} \end{cases} \tag{8}$$

$$range\_normalized\_diff(a_r(x_i), a_r(x_j)) = \frac{|a_r(x_i) - a_r(x_j)|}{max_{a_r} - min_{a_r}} \tag{9}$$

where max_{a_r} and min_{a_r} are the maximum and minimum values, respectively, observed in the training set for attribute a_r. This means that it is possible for a new (unseen) input vector to have a value outside this range, producing a difference value greater than one. However, such cases are rare, and when they occur, a larger difference may be acceptable anyway.

The HEOM distance function is overly simplistic approach to handle qualitative attributes and fails to make use of additional information provided by qualitative attributes values that can aid in the learning process.

The *Value Difference Metric (VDM)* introduced by Stanfill & Waltz (1986) provides an appropriated distance function for qualitative attributes. The VDM metric considers the classification similarity for each possible value of an attribute to calculate the distances between these values. As a result, a matrix of distances is created from the training set for each attribute. The distance $VDM_a(a_r(x_i), a_r(x_j))$ between two values $a_r(x_i)$ and $a_r(x_j)$ of a given qualitative attribute a_r, is defined by Equation 10.

$$VDM_a(a_r(x_i), a_r(x_j)) = \sum_{l=1}^{s} \left| \frac{N_{a_r(x_i),v_l}}{N_{a_r(x_i)}} - \frac{N_{a_r(x_j),v_l}}{N_{a_r(x_j)}} \right|^c \tag{10}$$

where

- $N_{a_r(x_i)}$ is the number of instances in the training set that have value $a_r(x_i)$ for attribute a_r;
- $N_{a_r(x_i),v_l}$ is the number of instances in the training set that have value $a_r(x_i)$ for attribute a_r and class value v_l;
- l is the number of class values in the data set;
- c is a constant, usually 1 or 2.

The VDM metric considers two values to be similar if they have more similar classifications (*i.e.*, more similar correlations with the output classes), regardless what order the values may be given in. For example, if an attribute *color* has three values *red*, *green* and *blue*, and the application is to identify whether or not an object is an apple, *red* and *green* would be considered closer than *red* and *blue* because the former two have similar correlations with the output class *apple*.

The original VDM algorithm (Stanfill & Waltz 1986) makes use of feature weights that are not included in the Equation 10, and some variants of VDM, for instance (Cost & Salzberg 1993), have used alternate weighting schemes.

The VDM metric is a distance metric since it has the following properties:

1. $VDM_a(a_r(x_i), a_r(x_i)) = 0$ and $VDM_a(a_r(x_i), a_r(x_j)) > 0, a_r(x_i) \neq a_r(x_j)$ (non negativity);

2. $VDM_a(a_r(x_i), a_r(x_j)) = VDM_a(a_r(x_j), a_r(x_i))$ (symmetry);
3. $VDM_a(a_r(x_i), a_r(x_j)) + VDM_a(a_r(x_j), a_r(x_m)) \geq VDM_a(a_r(x_i), a_r(x_m))$ (triangle inequality).

If VDM is used directly on quantitative attributes, most of the values might be unique, in which case $N_{a_r(x_i)}$ is 1, and $N_{a_r(x_i),v_l}$ is 1 for a value v_l and 0 for all other attribute class values. In addition, query instances are likely to have unique values not present in the training set. In this case $N_{a_r(x_i),v_l}$ will be 0 for all $v_l \in V$ and $N_{a_r(x_i)}$ (which is the sum of $N_{a_r(x_i),v_l}$ for all classes) will also be 0, resulting in a division by zero.

Even if all quantitative attribute values are not unique, there are often too many different values, thus the statistical sample is unreliably small for each value, and the distance measure is still untrustworthy. Because of these problems, it is inappropriate to use the VDM directly on quantitative attributes.

One approach to the problem of using VDM on quantitative attributes is *discretization* (Lebowitz 1985; Kohavi & Sahami 1996). A quantitative attribute can be discretized and treated as a qualitative attribute. However, discretization can lose much of the important information available in quantitative values.

If the Euclidean distance function is inappropriate for qualitative attributes, and VDM is inappropriate for quantitative attributes, so neither of them is sufficient on its own for use on a data set that has both qualitative and quantitative attributes.

The *Heterogeneous Distance Function (HVDM)* (Wilson & Martinez 2000) is a distance function similar to HEOM, except that it uses VDM instead of an overlap metric for qualitative attributes and it also normalizes differently. This distance function is defined by Equation 11.

$$HVDM(x_i, x_j) = \sqrt{\sum_{r=1}^{n} d_a(a_r(x_i), a_r(x_j))^2} \tag{11}$$

where n is the number of attributes in the data set. $d_a(a_r(x_i), a_r(x_j))$ is a distance between two values $a_r(x_i)$ and $a_r(x_j)$ of a given attribute a_r, and is defined as

$$d_a(a_r(x_i), a_r(x_j)) = \begin{cases} 1, \text{ if } a_r(x_i) \text{ or } a_r(x_j) \text{ is unknown; otherwise} \\ normalized\_vdm(a_r(x_i), a_r(x_j)), \text{ if } a_r \text{ is qualitative} \\ normalized\_diff(a_r(x_i), a_r(x_j)), \text{ if } a_r \text{ is quantitative} \end{cases} \tag{12}$$

Since 95% of the values in a normal distribution fall within two standard deviations of the mean, the difference between numeric values is divided by 4 standard deviations to scale each value in a range that is usually of width 1. The function $normalized\_diff$ is therefore defined as

$$normalized\_diff(a_r(x_i), a_r(x_j)) = \frac{(|a_r(x_i) - a_r(x_j)|)}{4\sigma_{a_r}} \tag{13}$$

where σ_{a_r} is the standard deviation of the numeric values of attribute a_r.

The *normalized_vdm* is defined as

$$normalized\_vdm(a_r(x_i), a_r(x_j)) = \sqrt{\sum_{l=1}^{s} \left| \frac{N_{a_r(x_i),v_l}}{N_{a_r(x_i)}} - \frac{N_{a_r(x_j),v_l}}{N_{a_r(x_j)}} \right|^2} \quad (14)$$

In our implementation of the K-NEAREST NEIGHBOUR algorithm, we can select which distance function will be used. In our experiments, for data sets with both qualitative and quantitative attributes, we used the HVDM distance function. For data sets having exclusively quantitative attributes we used the Euclidean distance function.

5.1.4 Speeding Up Queries with M-trees

M-tree (Ciaccia, Patella & Zezula 1997) is an indexing structure able to support execution of *similarity queries*. In principle, there are two basic types of similarity queries:

1. **Range query.** Given a *maximum search distance* and a query example, this similarity query returns all examples in the training set that are in a distance smaller or equal to the maximum search distance from the query example;
2. **k-nearest neighbour query.** Given a query example and a number of neighbours k, this similarity query returns k examples in the training set which have the shortest distance from the query example.

M-tree is a *metric tree* (Uhlmann 1991). Metric trees only consider relative distances of instances (rather than their absolute positions in a multi-dimensional space) to organize and partition the *metric space*. In a metric space, instance proximity is only defined by a distance function that satisfies the positivity, symmetry and triangle inequality postulates.

A M-tree is a bottom-up growing tree with fixed-size nodes, which correspond to constrained regions of the metric space. Leaf nodes store all indexed objects (*i.e.*, instances), and internal nodes store the so-called *routing objects*. A routing object is an instance to which a routing role is assigned.

For each routing object O_r there is an associated pointer, denoted $ptr(T(O_r))$, which references the root of a sub-tree, $T(O_r)$, called the *covering tree* of O_r. All objects in the covering tree of O_r are within the distance $r(O_r)$ from O_r, $r(O_r) > 0$, which is called the *covering radius* of O_r and form a part of the O_r entry in a particular M-tree node.

An entry for an object O_l in a leaf node is quite similar to that of a routing object, but no covering radius is needed. Ciaccia, Patella & Zezula (1997) provides additional information about the structure of a M-tree and detailed algorithms for searching and building a M-tree.

Figure 1 on the next page presents an example of a M-tree with two routing nodes and three leaf nodes, and Figure 2 on page 12 shows a graphical representation of this M-tree.

A M-tree grows in a bottom-up fashion. When a new object is inserted, M-tree locates the most suitable leaf node for accommodating it. If the leaf node is full then a new leaf node is allocated, the entries are partitioned among these two nodes, and two entries are promoted to the parent

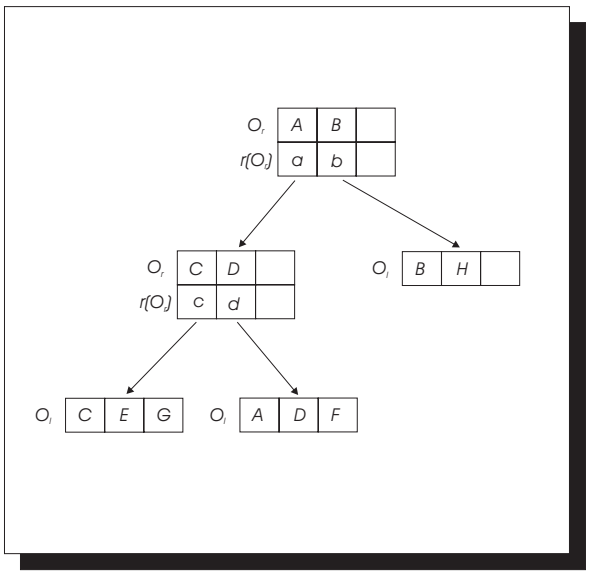

Figure 1: Example of a M-tree structure.

node. If the parent node is full, then a similar algorithm is applied, *i.e.* a new node is allocated at the same level of the full node, the entries are partitioned and two entries are promoted to the parent node. When a root splits, a new root is created and the M-tree grows by one level up.

A specific implementation of the Promote and Partition methods defines what is called a *split policy*. There are several alternatives for implementing these two methods. In our implementation, we chose to use an algorithm based on *Minimal Spanning Trees (MST)* as proposed by Jr., Traina, Seeger & Faloutsos (2000).

An ideal split policy should promote and partition the entries so that the two obtained regions would have minimum "volume" and minimum "overlap". Both criteria aim to improve the effectiveness of search algorithms, since having small (low volume) regions leads to well-clustered trees and reduces the amount of *dead space* — space where no object is present — and having small (possibly null) overlap between regions reduces the number of paths to be traversed for answering a query.

6 How C4.5 and CN2 Treat Missing Data

C4.5 (Quinlan 1988) and CN2 (Clark & Boswell 1991) are two well known Machine Learning algorithms that induce propositional concepts: decision trees and rules, respectively. These algorithms were selected because they are considered two of the best Machine Learning algorithms. C4.5 seems to have a good internal algorithm to treat missing values, since a recent comparative study, with other simple methods to treat missing values, concluded that it was one of the best methods (Grzymala-Busse & Hu 2000). On the other hand, CN2 seems to use a rather simple method to treat missing data.

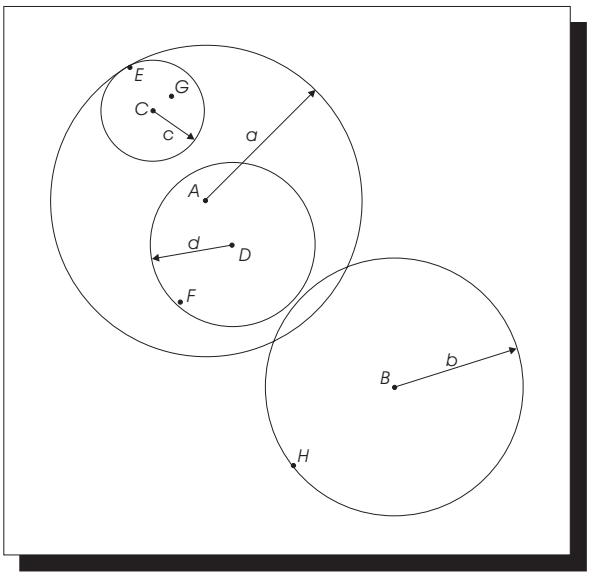

Figure 2: Graphical representation of the M-tree in Figure 1.

C4.5 uses a probabilistic approach to handle missing data. Missing values can be present in any attribute, except the class attribute, in training and test files.

Given a training set, T, C4.5 finds a suitable test, based on a single attribute, that has one or more mutually exclusive outcomes $O_1, O_2, \ldots, O_n$. T is partitioned into subsets T_1, T_2, ..., T_n, where T_i contains all the instances in T that satisfy the test with outcome O_i. The same algorithm is applied to each subset T_i until a stop criteria is applied.

C4.5 uses the *information gain ratio* measure to choose a good test to partition the instances. If there exist missing values in an attribute X, C4.5 uses the subset with all known values of X to calculate the information gain.

Once a test based on an attribute X is chosen, C4.5 uses a probabilistic approach to partition the instances with missing values in X. When an instance in T with known value is assigned to a subset T_i, this indicates that the probability of that instance belonging to subset T_i is 1 and to all other subsets is 0. When the value is not known, only a weaker probabilistic statement can be made. C4.5 associates to each instance in T_i a *weight* representing the probability of that instance belonging to T_i. If the instance has a known value, and satisfies the test with outcome O_i, then this instance is assigned to T_i with weight 1; if the instance has an unknown value, this instance is assigned to all partitions with different weights for each one. The weight for the partition T_i is the probability that instance belongs to T_i. This probability is estimated as the sum of the weights of instances in T known to satisfy the test with outcome O_i, divided by the sum of weights of the cases in T with known values on the attribute X.

The CN2 algorithm uses a rather simple approach to treat missing data. Every missing value is filled in with its attribute most common known value, before calculating the entropy measure (Clark & Niblett 1989).

7 Experimental Analysis

The main objective of the experiments conducted in this work is to evaluate the efficiency of the K-NEAREST NEIGHBOUR algorithm as an imputation method to treat missing data, comparing its performance with the performance obtained by the internal algorithms used by C4.5 and CN2 to learn with missing data, and by the MEAN OR MODE IMPUTATION method.

In these experiments, missing values were artificially implanted, in different rates and attributes, into the data sets. The performance of all four missing data treatments were compared using cross-validation estimated error rates. In particular, we are interested in analysing the behaviour of these treatments when the amount of missing data is high since some researchers have reported finding databases where more than 50% of the data were missing (Lakshminarayan, Harp & Samad 1999).

The experiments were carried out with the following data sets from UCI (Blake & Merz 1998):

Bupa This data set was contributed by R. S. Forsyth to the UCI repository. The problem is to predict whether or not a male patient has liver disorders based on various blood tests and the amount of alcohol consumption.

Pima This data set was donated by V. Sigillito, Applied Physics Laboratory, Johns Hopkins University to the UCI repository. This data set is also a subset of a larger database maintained by the National Institute of Diabetes and Digestive and Kidney Diseases.

All patients are females at least 21 years old of Pima Indian heritage living near Phoenix, Arizona, USA. The problem is to predict whether a patient would test positive for diabetes according to World Health Organization (WHO) criteria — *i.e.*, if the 2-hour post-load plasma glucose is at least 200 mg/dl at any survey examination or if found during routine medical care — given a number of physiological measurements and medical test results.

CMC This data set is composed by a subset of the 1987 National Indonesia Contraceptive Prevalence Survey and was donated by Tjen-Sien Lim. The samples are married women who were either not pregnant or do not know if they were at the time of the interview. The problem is to predict the current contraceptive method choice (no use, long-term methods or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Breast This data set was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg (Mangasarian & Wolbert 1990). The problem is to predict whether a tissue sample taken from a patient's breast is malignant or benign. Tissue samples consist of visually assessed nuclear features of fine needle aspirates taken from patient's breast. Each sample was assigned a 9-dimensional vector, where each attribute is in the range 1 to 10, with 1 referring to a normal state and 10 to a most abnormal one. Malignancy is determined by taking a tissue sample from patient's breast and performing a biopsy on it. A benign diagnosis is confirmed by biopsy or by periodic examination, depending on the patient's choice.

The first three data sets have no missing values. Breast has very few cases with missing values (in total 16 cases or 2.28%) which were removed before starting the experiments. The main

reason for not using data with missing values is the wish to have total control over the missing data in the data set. For instance, we would like the test sets to have no missing data. If some test set has missing data, then the inducer's ability to classify missing data properly may influence the result. This influence is undesirable since the objective of this work is to analyse the viability of the K-NEAREST NEIGHBOUR as an imputation method for missing data and the inducer learning ability when missing values are present.

Table 2 summarizes the data sets employed in this study. It shows, for each data set, the number of instances (#Instances), number and percentage of duplicate (appearing more than once) or conflicting (same attribute-value but different class attribute) instances, number of attributes (#Attributes), number of quantitative and qualitative attributes, class attribute distribution and the majority class error. This information was obtained using the MLC++ *info* utility (Kohavi, Sommerfield & Dougherty 1997).

Data set	# Instances	#Duplicate or conflicting (%)	#Attributes (quanti., quali.)	Class	Class %	Majority Error
Bupa	345	4 (1.16%)	6 (6,0)	1	42.03%	42.03%
				2	57.97%	on value 2
CMC	1473	115 (7.81%)	9 (2,7)	1	42.70%	57.30%
				2	22.61%	on value 1
				3	34.69%	
Pima	769	1 (0.13%)	8 (8,0)	0	65.02%	34.98%
				1	34.98%	on value 0
Breast	699	8 (1.15%)	9 (9,0)	2	65.52%	34.48%
				4	34.48%	on value 2

Table 2: Data sets summary descriptions.

Figure 3 on the facing page illustrates the methodology used in our experiments. Initially, each one of the original data set is partitioned into 10 pairs of training and test sets through the application of the resampling method stratified 10-fold cross validation. For each fold, missing values are implanted into the training set. First, copies of the training set with missing data are given directly to C4.5 and CN2. Two classifiers are induced and the error is measured in the test set. Later, the training set with missing data is treated by the imputation methods K-NEAREST NEIGHBOUR and MEAN OR MODE. Copies of the two treated training are given to C4.5 and CN2 and the respective error rates are measured in the test set. At the end of 10 iterations, the true error rate of each method can be estimated by calculating the mean of the error rates of each iteration. Finally, the performances of C4.5 and CN2 allied to the K-NEAREST NEIGHBOUR imputation method can be analysed and compared to the performances of the methods used internally by C4.5 and CN2 to learn when missing values are present, and to the performances of C4.5 and CN2 allied to the MEAN OR MODE IMPUTATION.

In order to insert missing data into the training sets, some attributes have to be chosen, and some of their values modified to unknown. Which attributes will be chosen and how many of their values will be modified to unknown is an important decision. It is straightforward to see that the most representative attributes of the data set are a sensible choice for the attributes that should have their values modified to unknown. Otherwise, the analysis may be compromised by treating non-representative attributes that will not be incorporated into the classifier by the learning system. Since finding the most representative attributes of a data set is not a trivial task, we used the results of (Lee, Monard & Baranauskas 1999) to select the three most relevant attributes according to several feature subset selection methods such as wrapper and filter.

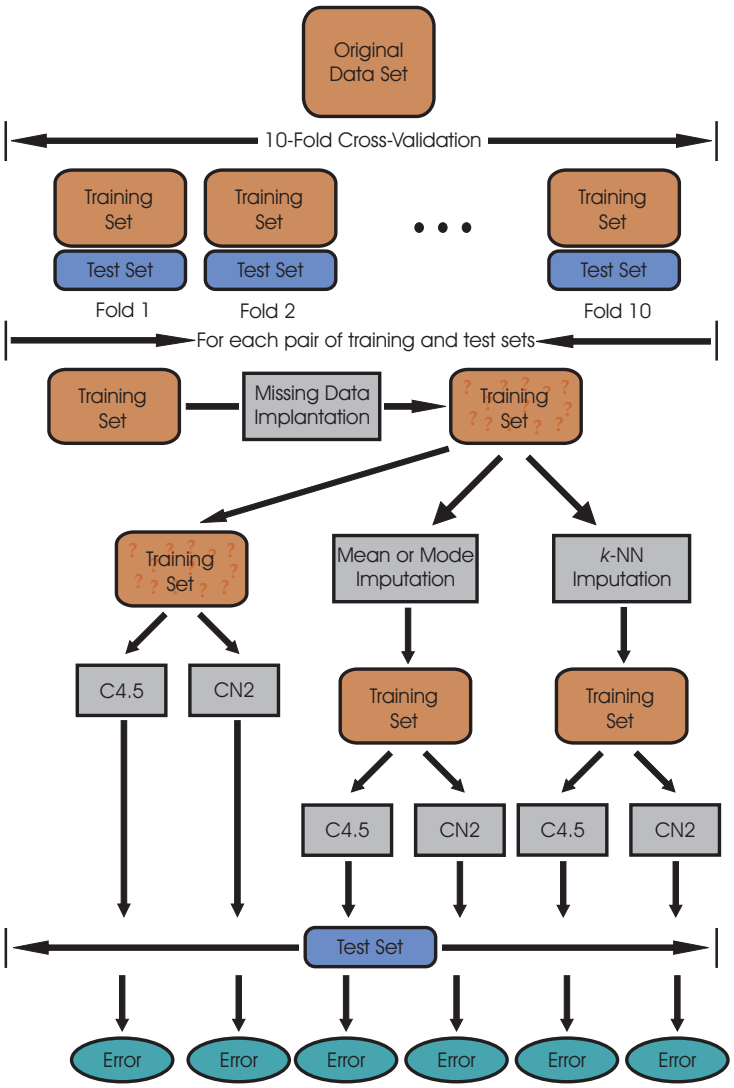

Figure 3: Methodology used in the experimental analysis.

Related to the amount of missing data to be inserted into the training sets, we want to analyse the behaviour of the methods with different amounts of missing data. In this way, missing data was inserted completely at random (MCAR) in the following percentages: 10%, 20%, 30%, 40%, 50% and 60% of the total of instances in the training set. The experiments were performed with missing data inserted into one, two and three of the attributes selected as the most representatives. The missing values were replaced by estimated values using 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours. In order to keep the analysis focus, only results with 10-nearest neighbour, identified as 10-NNI, will be showed in this section. Appendix A on page 42 presents the remaining results in graphical form.

The following sections elucidate some decisions taken in the methodology used in this work. First, Section 7.1 explains how three attributes were selected as the three most important attributes in each data set. Later, Section 7.2 presents part of the experimental results that were carried out in order to identify a value for the k parameter (number of neighbours) which is

appropriate for the majority of the data sets[1]. Finally, Section 7.3 presents and discusses the results of the experiments, for each data set, comparing the K-NEAREST NEIGHBOUR algorithm as an imputation method to the MEAN AND MODE IMPUTATION and the internal strategy to deal with missing data used by C4.5 and CN2.

7.1 Identification of Relevant Attributes

As explained earlier, we would like to select the most representative attributes in each data set in order to artificially insert missing values. The selection of the most representative attributes would increase the likelihood of these attributes to be included into the induced classifier. Otherwise, the analysis may be compromised by treating non-representative attributes that will not be incorporated into the classifier by the learning system.

However, there is no assurance that these attributes will be incorporated by C4.5 and CN2 into the classifiers induced in the experiments. The existence of an attribute with similar information (high correlation) to one of the selected attributes, can make the inducers decide not to use the selected attribute.

Since finding the most representative attributes of a data set is not a trivial task, we used the results of (Lee, Monard & Baranauskas 1999) to select the three most relevant attributes according to several feature subset selection. In (Lee, Monard & Baranauskas 1999) the following approaches were used:

Filter A feature selection method is applied to filter the features before the induction process occurs. The inducers C4.5 and ID3 (Quinlan 1986) were used as filters, as well as the column importance facility provided by MineSet™ from SGI[2].

Wrapper The wrapper algorithm makes a search for a subset of representative features by adding or removing features and measuring the resulting error rate. The wrapper search can be forward, starting with an empty set of features and adding features step by step; or backward, beginning with the full set of features and removing features in each step until a stop criteria is reached. The wrapper approach uses an induction algorithm as a black box. Lee, Monard & Baranauskas (1999) use C4.5, C4.5-RULES and CN2 algorithms as black boxes. Also, both forward and backward searches were applied.

In our work, we identified three attributes that were more frequently selected by the feature subset selection methods. When possible, we tried to identify the most relevant attribute, the second and the third most relevant attributes. Table 3 on the next page summarizes the selected attributes for each data set ordered by its relevance.

7.2 Estimating a Good Value for the Parameter k

The K-NEAREST NEIGHBOUR algorithm, as any other learning algorithm, has parameters that need to be tuned in order to optimize its performance. The main parameter of K-NEAREST

[1] Appendix B on page 55 presents the remaining results in both graphical and numerical forms.
[2] http://www.sgi.com

Data Set	Selected Attributes		
	Attribute Name	Attribute Position	Attribute Type
Bupa	Gammagt	4	real
	Sgpt	2	real
	Drinks	5	real
CMC	Nchi	3	real
	Wage	0	real
	Wedu	1	nominal
Pima	Plasma	1	real
	Body	5	real
	Number	0	real
Breast Cancer	Uniformity of Cell Size	1	integer
	Bare Nuclei	5	integer
	Clump Thickness	0	integer

Table 3: Selected attributes for each data set.

NEIGHBOUR algorithm is the number of nearest neighbours, k, used to make a prediction.

Due to the complexity of the data we deal with, the optimal value of k can vary from data set to data set. However, we can still find a value of k that leads to a good performance of the K-NEAREST NEIGHBOUR algorithm in most of the domains treated. This value will be used as default in our implementation of the K-NEAREST NEIGHBOUR algorithm for imputation of missing data.

As previously described, in our experiments missing values were artificially implanted in the training set into one, two and three selected attributes, in different proportions (from 10% up to 60%) and treated with 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours, as well as MEAN OR MODE IMPUTATION.

As we implanted missing values artificially, the actual value for each missing value is known. Thus, the error among the actual and predicted values can be measured. Means and standard deviations of these errors can be used as a statistic to evaluate the performance of each value of k (number of nearest neighbours). Also, the error obtained by the MEAN OR MODE IMPUTATION can be used as a basis for analysing if the K-NEAREST NEIGHBOUR computation is worth or a simpler method could provide similar results.

There are several measures that can be used to measure the error among actual and predicted values. The main measures are:

- **Classification.** For qualitative attributes the most used measure is the error rate which is defined as:

$$E = \frac{\sum_{i=1}^{m} \varepsilon(\hat{f}(x_i), f(x_i))}{m} \tag{15}$$

  where

  - x_i is the ith instance in the data set;
  - $f(x_i)$ is the true class value, and $\hat{f}(x_i)$ is the class value given by the learning system to the instance x_i;
  - $\varepsilon(a, b) = 1$ if $a \neq b$ and $\varepsilon(a, b) = 0$ otherwise;

– m is the number of instances in the data set.

- **Regression.** For quantitative attributes there are two commonly used measures: the mean absolute difference (*mad*) and the mean square error (*mse*). These measures are defined as:

$$mad = \frac{1}{m}\sum_{i=1}^{m}|\hat{f}(x_i) - f(x_i)| \tag{16}$$

and

$$mse = \frac{1}{m}\sum_{i=1}^{m}(\hat{f}(x_i) - f(x_i))^2 \tag{17}$$

Usually, *mad* error rates are slightly smaller than the square root of *mse*.

In this work we uses the error rate E — Equation 15 — for qualitative attributes and the mean square error *mse* —Equao 17 — for quantitative attributes.

Since three attributes were selected as three of the most representative attributes in each data set, six graphs with the results obtained were generated for each data set. For instance, for the breast data set the attributes 1, 5 and 0[3] were selected. It means that missing values were implanted into this data set in three configurations: missing data inserted only in attribute 1; missing data inserted into attributes 1 and 5; and finally, missing data inserted into attributes 1, 5 and 0. When missing data are inserted in only one attribute (say attribute 1) then a mean error is measured for this attribute. When missing data are inserted into 2 attributes (say attributes 1 and 5) then two mean errors are measured, one for each attribute. And finally, the mean errors are measured when missing data is inserted into all three selected attributes.

This section shows only one graph for each data set. The remaining graphs and tables containing the numerical information of each graph are presented in Appendix B on page 55.

Figures 4 to 7 on pages 19–20 present the *mse* values for several values of k (number of nearest neighbours), for the data sets Bupa, CMC, Pima and Breast, respectively.

It can be observed that the graphics show a general pattern. Usually, 1 and 3 nearest neighbours do not give the best results. Also, 50 and 100 nearest neighbours seem to be a too large number of neighbours which do not improve the final results. The best results are usually obtained by 5, 10, 20 or 30 nearest neighbours. In order to make the searches for the nearest neighbours less computationally intensive, we chose 10 nearest neighbours to be the default setting for our algorithms[4]. In the following section we show only results with 10 nearest neighbours, identified as 10-NNI. Complete results are shown in the Appendixes of this work.

[3]For simplicity, we use the attribute relative position in the data set, instead of its name. Refer to Table 3 on the preceding page in order to identify the attribute's name.

[4]If a traditional implementation of the K-NEAREST NEIGHBOUR algorithm is used then any number of nearest neighbours require a search through all training instances. Our implementation uses a M-tree to speed up the search for nearest neighbours. A lower value for k can make the M-tree prune some branches and make the search considerably faster.

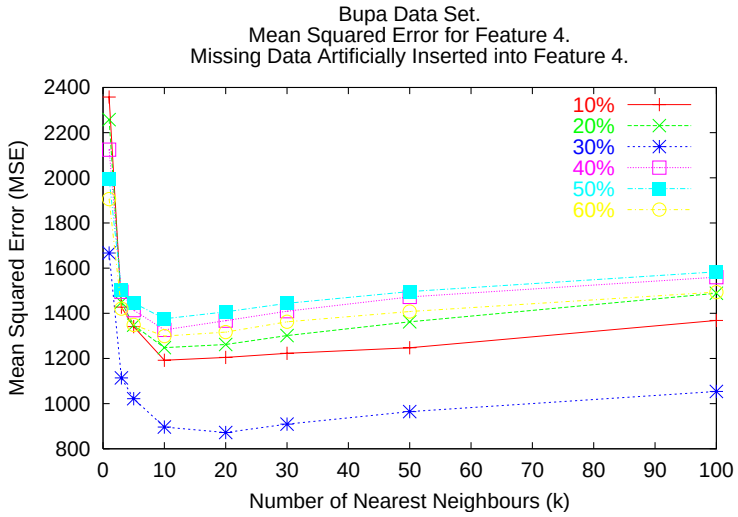

Figure 4: Bupa data set. Mean squared error measured over attribute 4 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attribute 4. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[1616.44 \pm 56.69, 1704.55 \pm 118.03]$. See Table 16 on page 65 for numerical results.

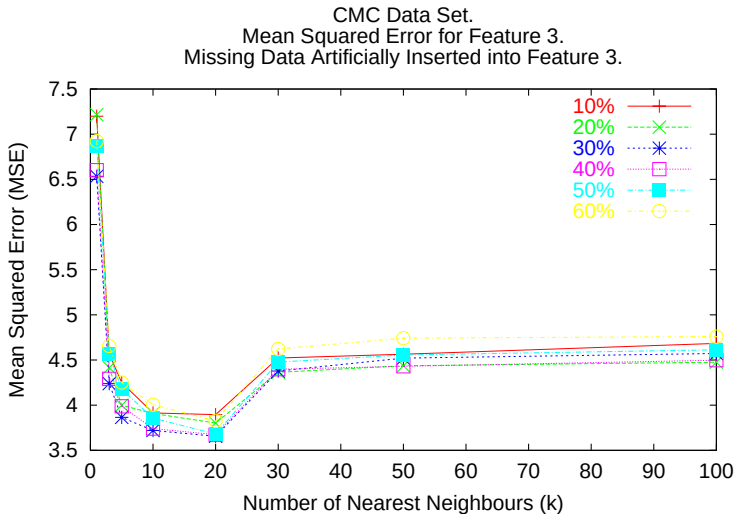

Figure 5: CMC data set. Mean squared error measured over attribute 3 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attribute 3. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[5.54 \pm 0.10, 5.97 \pm 0.30]$. See Table 22 on page 71 for numerical results.

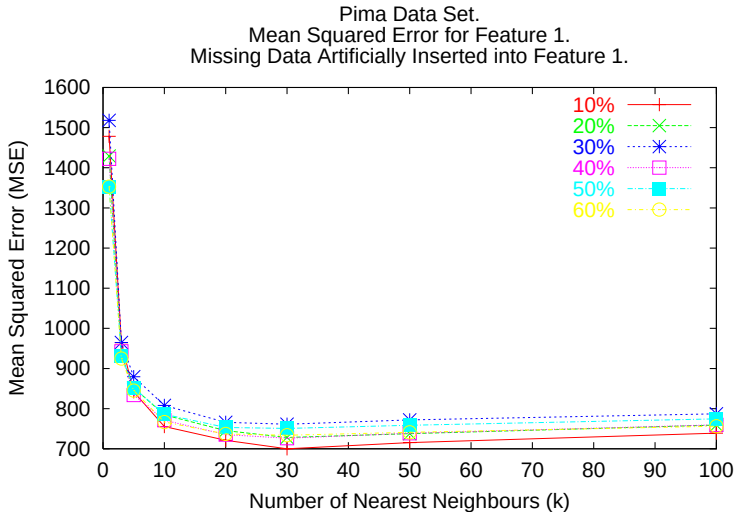

Figure 6: Pima data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attribute 1. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[989.81 \pm 29.45, 1044.24 \pm 50.58]$. See Table 28 on page 77 for numerical results.

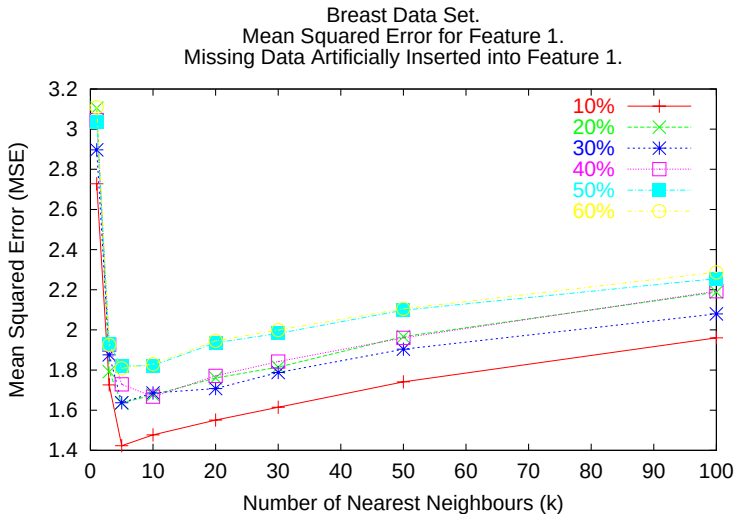

Figure 7: Breast data set. Mean squared error measured over attribute 0 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attribute 1. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[8.98 \pm 0.33, 9.39 \pm 0.12]$. See Table 10 on page 59 for numerical results.

7.3 Experimental Results

This sections shows the error rate of the classifiers induced by C4.5 and CN2 learning algorithms. For each data set, six graphs are shown. Each graph compares the performance of three "classifiers[5]" induced from data with different levels of missing values on a set of attributes. The three classifiers are:

No treatment The classifier is generated by inducing a model from data with untreated missing values;

10-NNI The missing values are substituted for values estimated by the 10-NNI method. After the imputation step, a classifier is induced from the data set with no missing data;

Mean or Mode imputation The MEAN OR MODE IMPUTATION method is applied substitute the missing values, before the induction process. The classifier is induced from complete data.

For each data set, there is one table with the numerical information shown in the graphs. The tables compare the results obtained by the missing data treatment methods using the *resampling 10-fold cross-validation paired t-test* (Dietterich 1997). This test aims to determine when one method outperforms other method with 95% or 99% of confidence.

The following sections show the experimental results for the Bupa, CMC, Pima and Breast data sets, respectively.

7.3.1 Bupa Data Set

Considering the results shown in Figures 8 to 13 on pages 23–25 for the Bupa data set, it can be observed that the performance of 10-NNI is in most cases superior to the performances of C4.5 and CN2 internal algorithms, and the mean imputation method. Furthermore, the C4.5 internal algorithm is competitive to 10-NNI only when missing values are inserted into the attributes 2, 4 and 5. The MEAN OR MODE IMPUTATION obtained good results when missing values are inserted into the attributes 2, 4 and 5, for the CN2 inducer.

[5] As we apply the 10-fold cross-validation method, the performance of each classifier is actually the mean performance of 10 classifiers.

Inducer: C4.5						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
4	0	36.82 ± 2.69	-	-	-	-
	10	38.56 ± 1.74	↑ 3.01	29.87 ± 1.76	⇑ 3.45	36.50 ± 1.76
	20	35.95 ± 1.24	0.67	34.78 ± 2.43	0.32	35.66 ± 1.61
	30	37.36 ± 1.89	0.72	35.36 ± 2.71	1.97	39.14 ± 2.41
	40	40.56 ± 2.05	↑ 3.09	31.55 ± 1.86	2.17	36.78 ± 1.72
	50	37.62 ± 2.35	1.16	33.34 ± 2.54	1.17	38.22 ± 3.03
	60	42.31 ± 2.11	↑ 2.62	31.22 ± 3.33	↑ 3.05	43.45 ± 2.08
4 and 2	0	36.82 ± 2.69	-	-	-	-
	10	35.32 ± 2.36	0.52	34.18 ± 1.72	0.01	34.20 ± 2.23
	20	36.22 ± 2.18	0.57	34.51 ± 2.16	1.60	38.21 ± 2.54
	30	37.70 ± 2.40	0.90	35.96 ± 2.05	0.53	37.07 ± 2.44
	40	37.08 ± 1.42	↑ 2.51	32.45 ± 1.09	0.84	34.25 ± 1.76
	50	39.71 ± 2.76	1.85	33.28 ± 3.07	1.84	40.89 ± 2.31
	60	36.21 ± 1.84	1.09	33.57 ± 2.38	2.13	39.36 ± 2.30
4, 2 and 5	0	36.82 ± 2.69	-	-	-	-
	10	35.36 ± 1.76	1.51	31.56 ± 2.44	↑ 3.03	39.71 ± 1.91
	20	33.92 ± 2.07	0.36	33.05 ± 2.09	1.53	35.92 ± 1.17
	30	35.97 ± 2.90	0.08	35.61 ± 3.00	0.41	36.52 ± 1.68
	40	36.19 ± 2.39	0.38	35.11 ± 2.14	1.30	40.29 ± 2.47
	50	34.39 ± 2.84	-0.97	36.75 ± 2.12	-1.16	34.45 ± 1.75
	60	34.48 ± 1.77	0.00	34.47 ± 3.02	0.60	36.46 ± 1.71
Inducer: CN2						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
4	0	35.39 ± 2.47	-	-	-	-
	10	33.58 ± 1.94	-0.22	34.19 ± 1.45	-1.05	31.91 ± 1.88
	20	36.82 ± 0.96	⇑ 3.51	32.45 ± 0.95	0.79	33.95 ± 1.70
	30	38.53 ± 2.16	⇑ 4.02	31.56 ± 2.71	1.82	36.52 ± 1.74
	40	39.13 ± 1.09	⇑ 6.21	28.96 ± 2.24	↑ 2.56	33.91 ± 1.36
	50	37.35 ± 2.74	2.02	31.28 ± 1.91	1.69	35.92 ± 2.09
	60	39.41 ± 1.20	↑ 2.38	33.29 ± 2.64	0.48	34.51 ± 2.78
4 and 2	0	35.39 ± 2.47	-	-	-	-
	10	34.75 ± 2.01	0.57	33.63 ± 1.77	0.73	35.45 ± 2.21
	20	33.81 ± 3.23	0.64	31.81 ± 2.65	1.08	33.89 ± 1.49
	30	37.66 ± 1.48	↑ 2.65	33.34 ± 1.88	0.13	33.61 ± 1.96
	40	39.67 ± 1.98	2.25	33.02 ± 2.44	0.32	33.88 ± 1.27
	50	41.72 ± 1.38	↑ 3.04	34.51 ± 2.40	0.71	36.83 ± 1.88
	60	38.81 ± 1.58	⇑ 3.90	31.01 ± 1.48	⇑ 3.48	36.51 ± 2.33
4, 2 and 5	0	35.39 ± 2.47	-	-	-	-
	10	37.09 ± 2.55	↑ 2.45	30.71 ± 2.47	1.69	34.50 ± 1.81
	20	34.18 ± 2.03	-0.38	34.81 ± 1.49	0.30	35.39 ± 1.75
	30	35.94 ± 2.14	0.24	35.35 ± 1.39	-0.46	34.18 ± 1.92
	40	38.25 ± 1.49	↑ 3.09	32.49 ± 1.20	-0.31	31.59 ± 2.51
	50	41.97 ± 1.58	⇑ 5.34	31.56 ± 1.58	0.21	32.18 ± 2.24
	60	40.56 ± 1.88	2.05	34.82 ± 2.04	1.94	39.72 ± 1.63

Table 4: Comparative results for the Bupa data set.

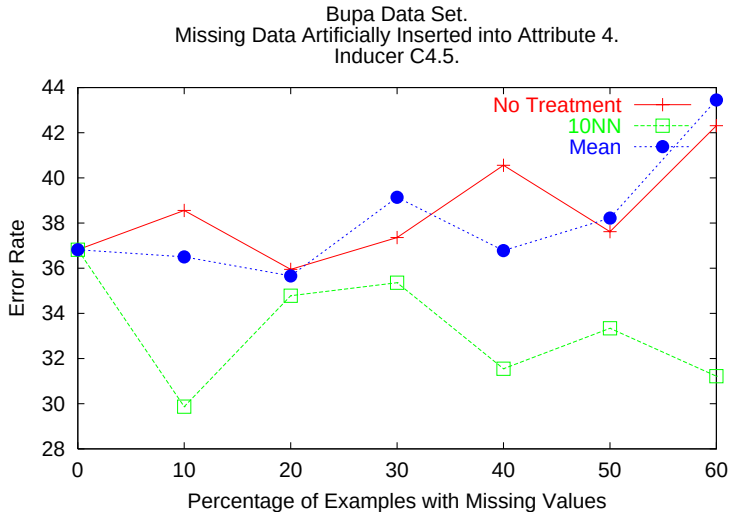

Figure 8: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 4 of Bupa data set. See Table 4 on the preceding page for numerical results.

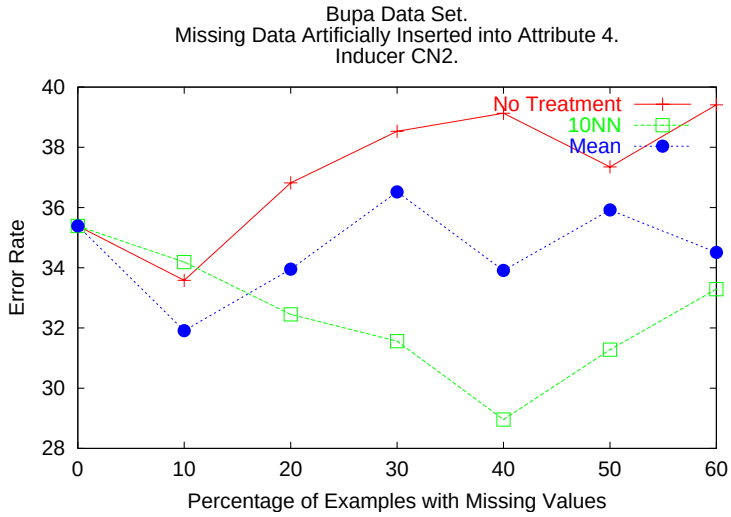

Figure 9: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 4 of Bupa data set. See Table 4 on the preceding page for numerical results.

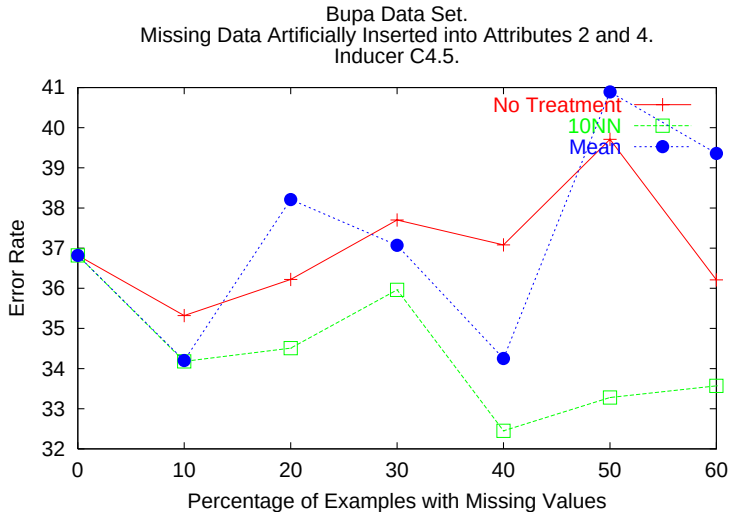

Figure 10: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2 and 4 of Bupa data set. See Table 4 on page 22 for numerical results.

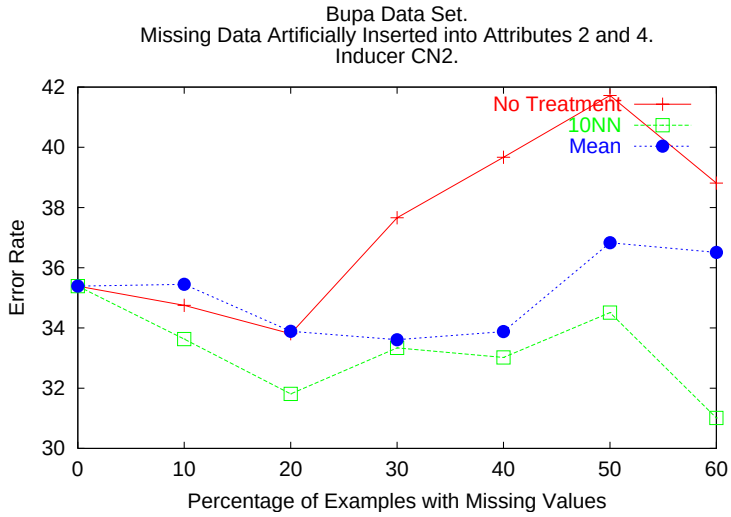

Figure 11: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2 and 4 of Bupa data set. See Table 4 on page 22 for numerical results.

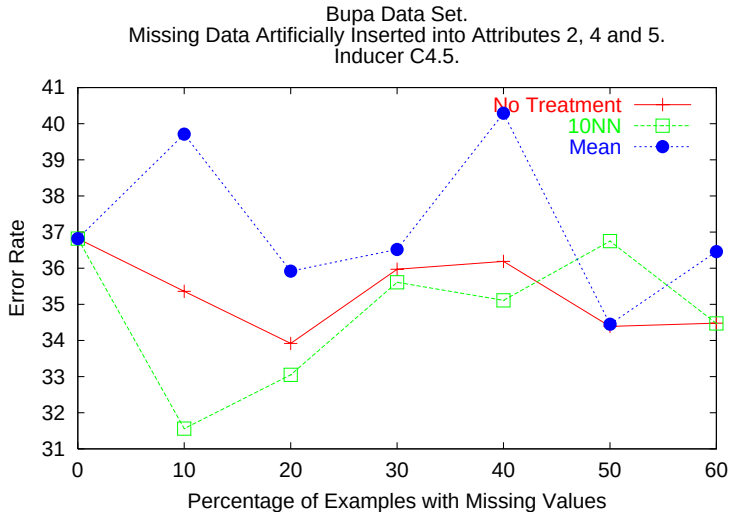

Figure 12: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2, 4 and 5 of Bupa data set. See Table 4 on page 22 for numerical results.

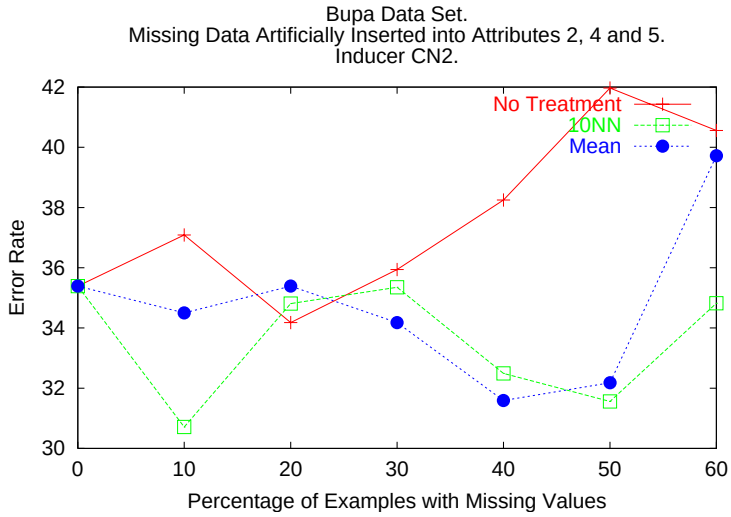

Figure 13: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2, 4 and 5 of Bupa data set. See Table 4 on page 22 for numerical results.

7.3.2 CMC Data Set

Similar results are shown in Figures 14 to 19 on pages 26–30 for the CMC data set. The performance of 10-NNI is in most cases superior to the performance obtained without missing data treatment, for both C4.5 and CN2. The performance of 10-NNI is in most cases superior to the performance of the MEAN OR MODE IMPUTATION method. In fact, the MEAN OR MODE IMPUTATION method is competitive to 10-NNI only when missing values are inserted into the attributes 0 and 3 and 0, 1, and 3, using CN2 as inducer.

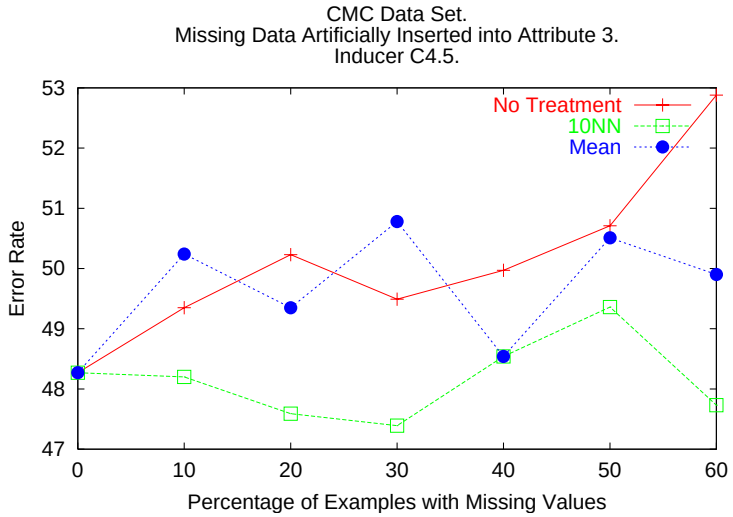

Figure 14: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 3 of CMC data set. See Table 5 on the next page for numerical results.

Inducer: C4.5						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
	0	48.27 ± 0.83	-	-	-	-
	10	49.35 ± 1.14	1.27	48.20 ± 1.16	1.59	50.24 ± 1.15
	20	50.23 ± 1.12	↑ 2.31	47.59 ± 0.98	1.62	49.35 ± 0.85
3	30	49.49 ± 0.95	2.16	47.39 ± 1.48	⇑ 3.66	50.78 ± 1.45
	40	49.97 ± 0.87	1.58	48.54 ± 1.12	0.00	48.54 ± 1.46
	50	50.71 ± 1.11	1.14	49.36 ± 0.91	0.87	50.51 ± 1.15
	60	52.88 ± 1.25	⇑ 7.08	47.73 ± 0.95	↑ 2.99	49.90 ± 1.07
	0	48.27 ± 0.83	-	-	-	-
	10	48.27 ± 0.67	1.10	47.32 ± 1.30	0.78	48.27 ± 1.37
	20	48.27 ± 0.99	-0.30	48.61 ± 1.30	0.56	49.62 ± 1.42
3 and 0	30	48.88 ± 1.40	-0.09	49.02 ± 1.36	1.33	50.58 ± 0.98
	40	48.61 ± 1.20	0.76	47.59 ± 1.53	0.91	49.56 ± 1.33
	50	49.49 ± 0.84	↑ 3.16	46.23 ± 1.06	↑ 2.34	49.15 ± 1.38
	60	50.64 ± 1.16	1.62	47.39 ± 1.87	1.48	50.24 ± 0.91
	0	48.27 ± 0.83	-	-	-	-
	10	46.78 ± 1.46	-0.29	47.18 ± 1.19	0.13	47.32 ± 0.78
	20	49.56 ± 1.34	1.75	48.34 ± 1.29	↑ 2.76	51.40 ± 1.49
3, 0 and 1	30	48.20 ± 1.19	0.05	48.13 ± 1.51	1.87	51.18 ± 0.89
	40	51.26 ± 1.33	↑ 3.18	47.45 ± 1.46	0.81	48.54 ± 1.12
	50	50.31 ± 1.23	↑ 2.52	47.38 ± 1.74	2.06	50.84 ± 1.61
	60	52.75 ± 1.16	⇑ 3.49	48.75 ± 1.86	↑ 2.53	51.46 ± 1.06
Inducer: CN2						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
	0	51.25 ± 0.80	-	-	-	-
	10	51.19 ± 1.51	0.48	50.64 ± 1.22	-0.99	49.69 ± 1.34
	20	51.73 ± 1.17	↑ 2.61	49.08 ± 0.95	0.05	49.15 ± 1.42
3	30	52.27 ± 0.94	1.96	49.70 ± 1.71	1.85	52.21 ± 1.13
	40	53.56 ± 1.47	↑ 2.28	50.51 ± 1.11	1.36	51.60 ± 0.73
	50	54.92 ± 0.95	⇑ 4.18	49.56 ± 1.74	1.46	51.39 ± 1.39
	60	54.24 ± 1.31	↑ 2.63	50.51 ± 1.12	1.13	51.93 ± 1.50
	0	51.25 ± 0.80	-	-	-	-
	10	51.26 ± 0.80	2.16	48.75 ± 1.42	0.76	49.83 ± 0.77
	20	52.48 ± 1.51	↑ 2.84	48.88 ± 1.46	1.82	50.78 ± 1.20
3 and 0	30	52.68 ± 0.91	↑ 2.77	48.54 ± 1.34	2.25	50.92 ± 0.95
	40	52.35 ± 1.10	↑ 2.88	50.44 ± 1.09	-0.28	50.11 ± 1.43
	50	52.68 ± 0.81	1.60	50.03 ± 1.76	-0.99	48.68 ± 1.18
	60	51.12 ± 1.53	0.28	50.85 ± 1.49	-0.69	50.10 ± 1.38
	0	51.25 ± 0.80	-	-	-	-
	10	51.32 ± 1.19	1.13	49.70 ± 1.61	-0.10	49.56 ± 1.46
	20	52.14 ± 1.04	0.39	51.66 ± 1.06	-0.53	51.12 ± 1.07
3, 0 and 1	30	52.95 ± 1.25	1.55	51.46 ± 1.15	0.60	52.34 ± 1.45
	40	53.36 ± 1.23	↑ 2.91	50.10 ± 1.49	-0.34	49.83 ± 0.85
	50	52.68 ± 1.02	0.55	51.73 ± 1.82	0.11	51.94 ± 1.29
	60	52.88 ± 0.76	1.32	50.92 ± 1.35	1.50	52.75 ± 1.05

Table 5: Comparative results for the CMC data set.

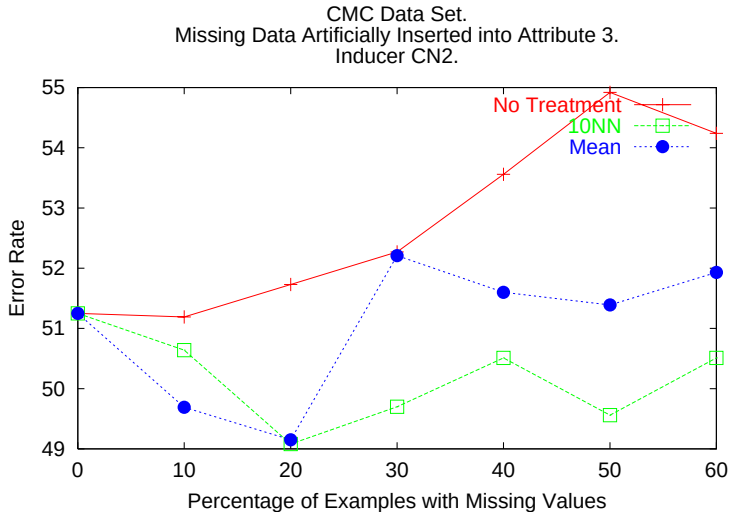

Figure 15: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 3 of CMC data set. See Table 5 on the page before for numerical results.

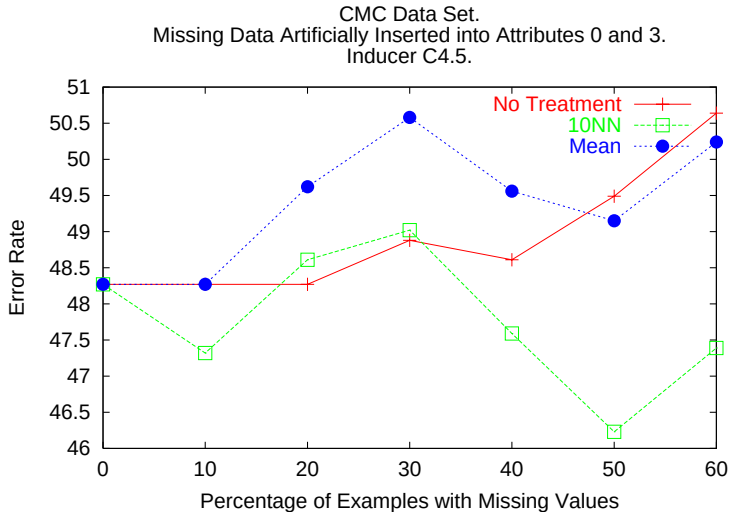

Figure 16: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0 and 3 of CMC data set. See Table 5 on the page before for numerical results.

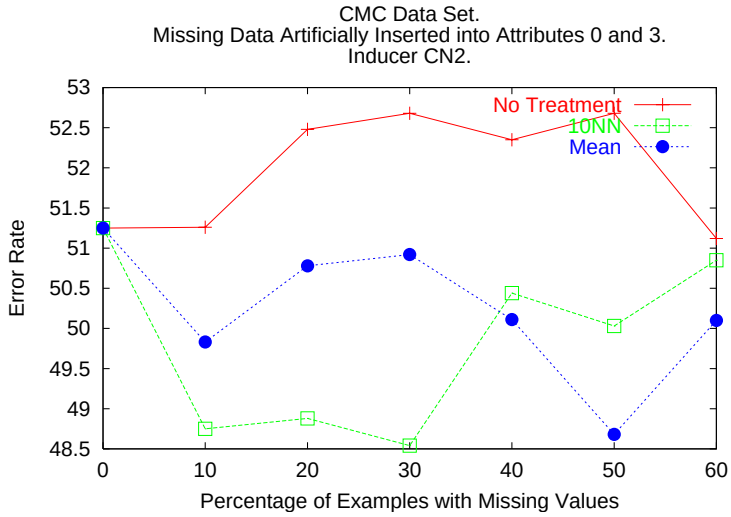

Figure 17: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0 and 3 of CMC data set. See Table 5 on page 27 for numerical results.

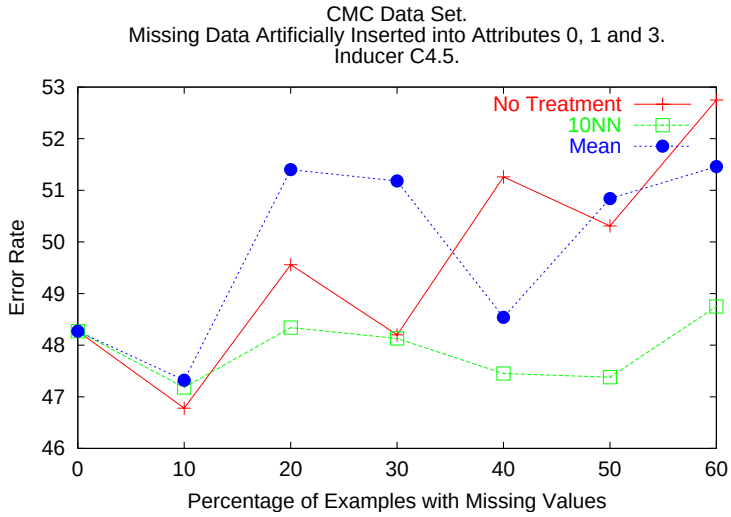

Figure 18: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 3 of CMC data set. See Table 5 on page 27 for numerical results.

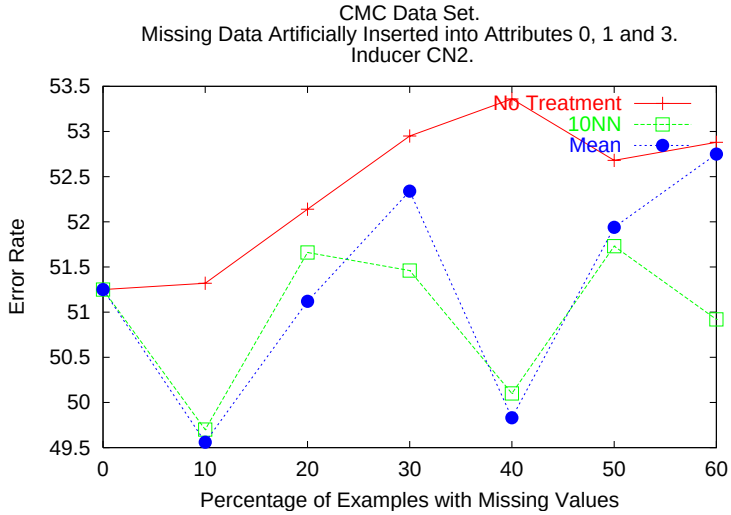

Figure 19: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 3 of CMC data set. See Table 5 on page 27 for numerical results.

7.3.3 Pima Data Set

Figures 20 to 25 on pages 31–35 show the experimental results for Pima data set. The performance of 10-NNI is in most cases superior to the performance obtained without missing data treatment for both C4.5 and CN2. Besides, 10-NNI is always superior to the MEAN OR MODE IMPUTATION as well as no missing data treatment when missing data are inserted into attribute 1 for both inducers.

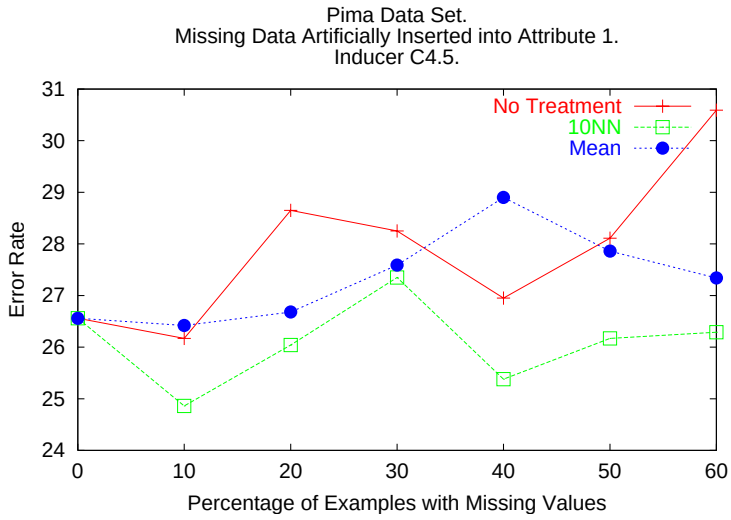

Figure 20: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Pima data set. See Table 6 on the next page for numerical results.

Inducer: C4.5						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
	0	26.56 ± 1.16	-	-	-	-
	10	26.17 ± 1.03	1.06	24.86 ± 0.88	1.01	26.42 ± 1.48
	20	28.65 ± 1.15	1.48	26.04 ± 1.68	0.43	26.68 ± 1.18
1	30	28.25 ± 1.85	0.46	27.35 ± 1.03	0.18	27.59 ± 1.38
	40	26.95 ± 1.67	0.93	25.38 ± 1.15	↑ 2.56	28.90 ± 1.23
	50	28.11 ± 1.14	1.10	26.17 ± 1.11	1.39	27.86 ± 0.84
	60	30.59 ± 1.13	2.21	26.29 ± 1.90	0.56	27.34 ± 1.05
	0	26.56 ± 1.16	-	-	-	-
	10	25.25 ± 1.10	-2.00	27.86 ± 1.15	-0.85	26.56 ± 1.08
	20	26.94 ± 1.22	0.39	26.43 ± 1.08	-0.49	25.91 ± 1.34
1 and 5	30	27.73 ± 1.60	1.26	25.39 ± 0.81	0.62	26.42 ± 1.27
	40	27.21 ± 1.45	0.51	26.29 ± 1.69	0.89	28.12 ± 1.11
	50	25.78 ± 1.13	-1.39	27.46 ± 1.16	0.36	27.99 ± 1.37
	60	29.81 ± 1.43	1.18	27.85 ± 1.51	-0.18	27.46 ± 1.67
	0	26.56 ± 1.16	-	-	-	-
	10	25.11 ± 1.70	-0.01	25.13 ± 0.90	0.18	25.51 ± 1.90
	20	26.30 ± 1.01	0.66	25.65 ± 1.35	0.88	27.33 ± 1.42
1, 5 and 0	30	26.17 ± 1.35	0.38	25.51 ± 1.75	1.17	27.48 ± 1.19
	40	26.82 ± 1.28	0.67	25.91 ± 1.44	-0.21	25.65 ± 0.84
	50	28.11 ± 1.32	⇑ 3.41	24.61 ± 1.16	2.15	28.11 ± 1.65
	60	27.60 ± 1.05	-0.19	27.86 ± 1.55	-0.41	27.34 ± 1.53
Inducer: CN2						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
	0	25.77 ± 1.12	-	-	-	-
	10	27.99 ± 0.98	2.23	25.91 ± 0.86	1.82	28.38 ± 0.87
	20	28.51 ± 1.06	1.80	26.18 ± 0.78	1.66	28.76 ± 1.51
1	30	27.47 ± 1.11	0.41	26.69 ± 1.61	1.14	29.30 ± 1.23
	40	30.21 ± 1.08	2.00	26.82 ± 0.98	↑ 3.15	30.34 ± 1.59
	50	30.34 ± 1.21	1.54	27.35 ± 1.47	1.23	29.68 ± 1.58
	60	30.21 ± 1.28	2.21	25.78 ± 1.33	↑ 2.55	30.72 ± 1.47
	0	25.77 ± 1.12	-	-	-	-
	10	28.38 ± 0.87	1.32	27.08 ± 0.98	-0.30	26.69 ± 1.31
	20	28.76 ± 1.51	0.32	28.25 ± 1.09	⇓ -4.97	23.43 ± 0.68
1 and 5	30	29.30 ± 1.23	↑ 2.45	25.65 ± 1.13	1.81	27.86 ± 1.16
	40	30.34 ± 1.59	↑ 2.56	26.17 ± 1.07	0.27	26.57 ± 1.73
	50	29.68 ± 1.58	2.02	25.91 ± 1.08	0.23	26.17 ± 0.82
	60	30.72 ± 1.47	1.18	27.60 ± 1.47	-0.07	27.47 ± 0.75
	0	25.77 ± 1.12	-	-	-	-
	10	27.48 ± 1.00	-0.21	27.73 ± 0.68	0.79	28.38 ± 0.99
	20	29.82 ± 0.82	1.48	27.87 ± 1.26	-1.31	26.30 ± 1.13
1, 5 and 0	30	31.25 ± 0.89	⇑ 3.55	26.17 ± 1.32	1.97	27.73 ± 0.91
	40	29.03 ± 0.90	⇑ 3.67	25.92 ± 1.32	1.77	27.35 ± 0.92
	50	29.69 ± 0.41	⇑ 7.98	25.26 ± 0.68	1.31	26.83 ± 1.29
	60	31.51 ± 1.17	↑ 3.05	26.05 ± 0.86	-1.83	23.83 ± 0.95

Table 6: Comparative results for the Pima data set.

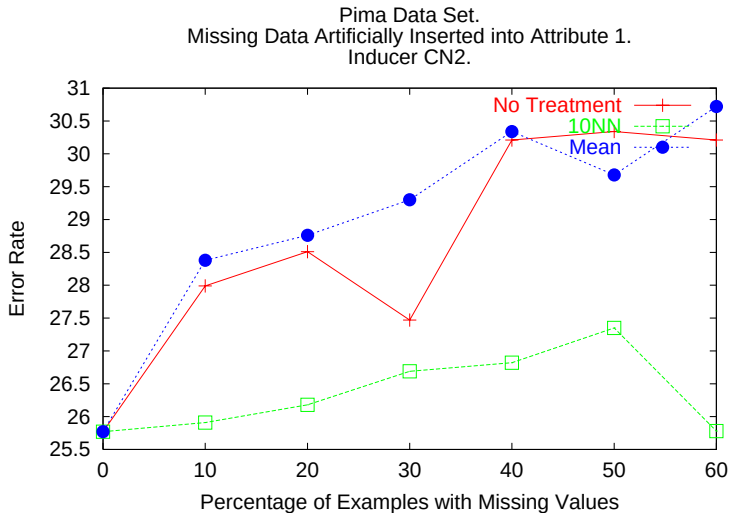

Figure 21: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Pima data set. See Table 6 on the preceding page for numerical results.

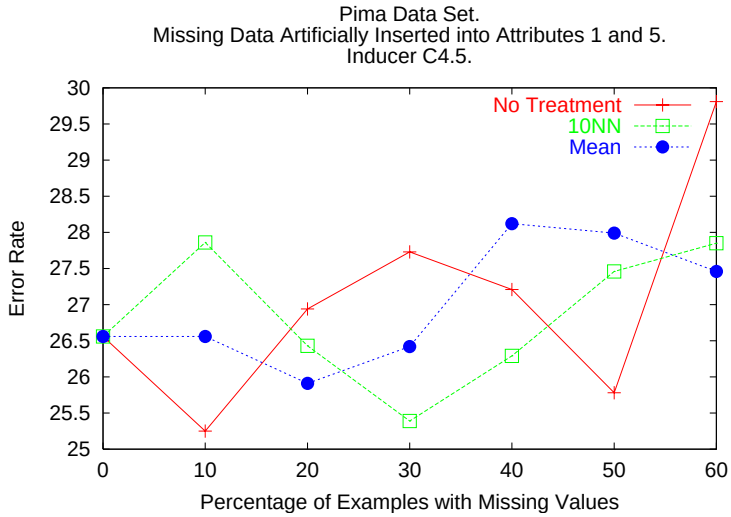

Figure 22: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Pima data set. See Table 6 on the preceding page for numerical results.

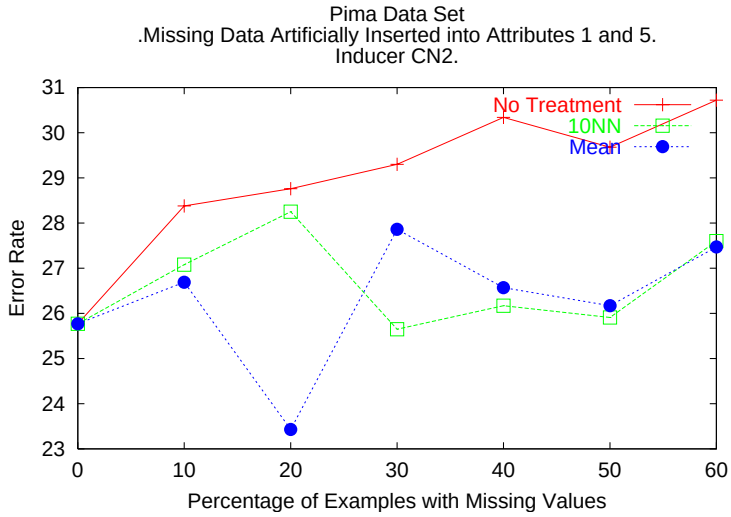

Figure 23: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Pima data set. See Table 6 on page 32 for numerical results.

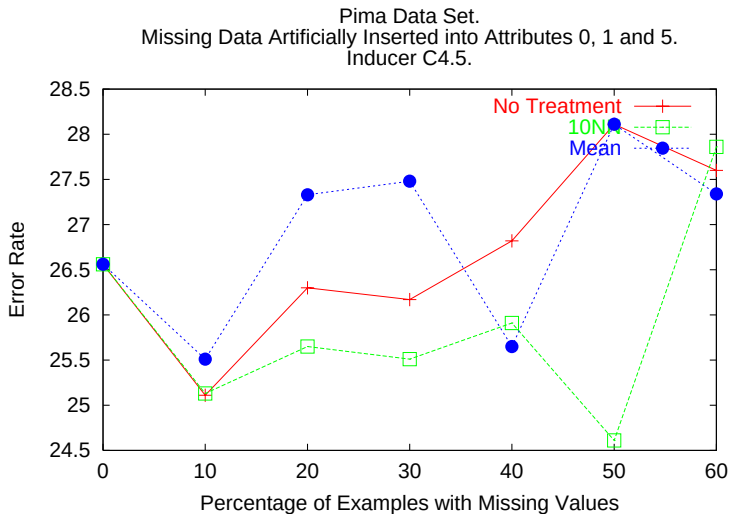

Figure 24: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Pima data set. See Table 6 on page 32 for numerical results.

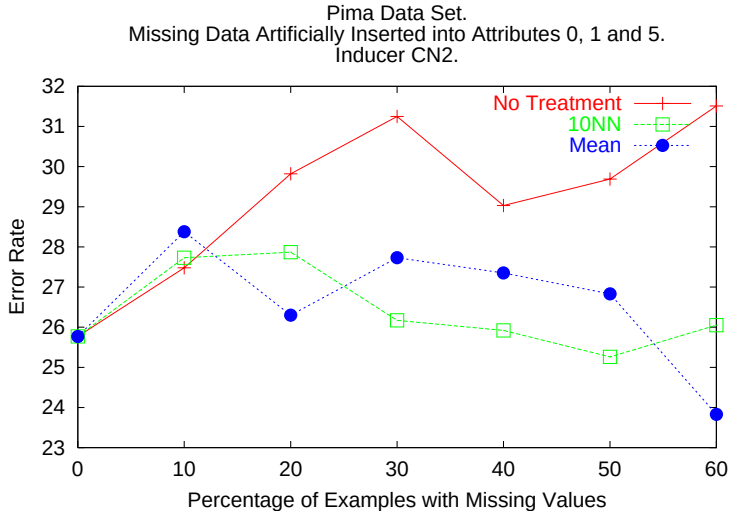

Figure 25: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Pima data set. See Table 6 on page 32 for numerical results.

7.3.4 Breast Data Set

Although missing data imputation with K-NEAREST NEIGHBOUR can provide good results, there are occasions that its use should be avoided. This is illustrated by the Breast data set, which was chosen because its attributes have strong correlations among themselves. These correlations cause an interesting situation: in one hand, the K-NEAREST NEIGHBOUR can predict the missing values with precision; on the other hand, the inducer can decide not to use the treated attribute, replacing it by another attribute with high correlation. The results for Breast data set are shown in Figures 26 to 31 on pages 36–40, where it can be seen that 10-NNI does not outperform the others missing data treatment methods.

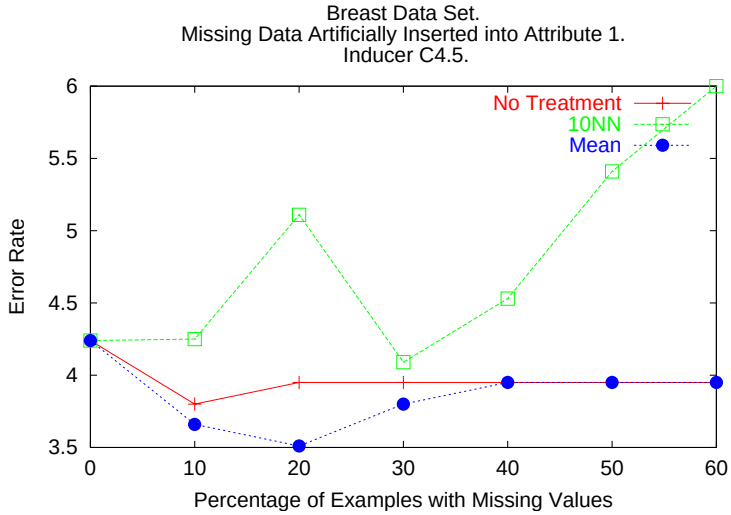

Figure 26: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Breast data set. See Table 8 on the next page for numerical results.

Attribute	MSE 10-NNI	MSE MEAN/MODE
0 (Clump Thickness)	4.02 ± 0.14	7.70 ± 0.28
1 (Uniformity of Cell Size)	1.72 ± 0.11	8.96 ± 0.36
5 (Bare Nuclei)	4.23 ± 0.30	13.29 ± 0.46

Table 7: Mean square error (MSE) between predicted and actual values for 10-NNI and MEAN OR MODE IMPUTATION — Breast data set.

Inducer: C4.5						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
1	0	4.24 ± 0.67	-	-	-	-
	10	3.80 ± 0.93	-0.61	4.25 ± 0.67	-0.94	3.66 ± 0.82
	20	3.95 ± 0.90	-1.56	5.11 ± 0.99	-2.01	3.51 ± 0.88
	30	3.95 ± 0.90	-0.24	4.09 ± 0.91	-0.47	3.80 ± 0.93
	40	3.95 ± 0.90	-0.73	4.53 ± 0.82	-0.73	3.95 ± 0.90
	50	3.95 ± 0.90	-1.79	5.41 ± 1.00	-1.79	3.95 ± 0.90
	60	3.95 ± 0.90	↓ -2.94	6.00 ± 0.88	↓ -2.94	3.95 ± 0.90
1 and 5	0	4.24 ± 0.67	-	-	-	-
	10	4.83 ± 0.61	0.99	4.10 ± 0.61	-0.44	3.80 ± 0.85
	20	4.97 ± 0.65	1.49	3.80 ± 0.88	0.97	4.68 ± 0.64
	30	4.68 ± 0.61	-0.21	4.83 ± 0.69	-1.40	4.39 ± 0.65
	40	4.39 ± 0.65	-0.77	4.98 ± 0.54	-0.01	4.97 ± 0.44
	50	4.98 ± 0.73	1.44	3.81 ± 0.63	1.50	4.69 ± 0.37
	60	4.54 ± 0.71	-1.96	5.85 ± 0.53	-1.92	4.68 ± 0.65
1, 5 and 0	0	4.24 ± 0.67	-	-	-	-
	10	4.68 ± 0.75	-0.19	4.83 ± 0.81	-1.34	4.10 ± 0.61
	20	5.12 ± 0.73	0.40	4.69 ± 0.68	0.14	4.83 ± 0.69
	30	5.42 ± 0.69	0.63	4.69 ± 1.02	0.28	4.98 ± 0.50
	40	4.97 ± 0.62	-0.32	5.27 ± 0.85	-1.40	4.09 ± 0.68
	50	5.41 ± 0.57	1.17	4.10 ± 0.84	0.68	4.83 ± 0.61
	60	4.97 ± 0.73	0.34	4.68 ± 0.80	0.00	4.68 ± 0.78
Inducer: CN2						
Attributes	% Missing	No Imputation	t-test	10-NNI	t-test	Mean or Mode
1	0	4.68 ± 0.60	-	-	-	-
	10	4.39 ± 0.44	-1.10	5.12 ± 0.84	-1.03	4.24 ± 0.46
	20	4.68 ± 0.75	0.32	4.39 ± 0.57	0.67	4.83 ± 0.69
	30	4.97 ± 0.82	0.00	4.97 ± 0.62	-0.39	4.67 ± 1.03
	40	4.53 ± 0.73	0.00	4.53 ± 0.70	1.82	5.12 ± 0.90
	50	4.53 ± 0.91	0.00	4.53 ± 0.63	0.43	4.82 ± 0.87
	60	4.83 ± 0.84	-0.33	5.12 ± 0.69	-0.36	4.83 ± 1.07
1 and 5	0	4.68 ± 0.60	-	-	-	-
	10	4.38 ± 0.65	0.00	4.38 ± 0.75	-0.87	3.80 ± 0.66
	20	3.65 ± 0.84	⇓ -3.55	5.56 ± 0.77	-1.91	4.53 ± 0.67
	30	3.95 ± 0.54	-1.87	4.69 ± 0.57	-0.70	4.09 ± 0.97
	40	3.95 ± 0.87	-1.29	4.96 ± 0.99	-2.12	3.51 ± 0.66
	50	4.53 ± 0.63	-0.56	4.98 ± 0.76	-0.50	4.68 ± 0.75
	60	4.39 ± 0.95	0.16	4.25 ± 0.64	-0.85	3.66 ± 0.66
1, 5 and 0	0	4.68 ± 0.60	-	-	-	-
	10	4.25 ± 0.71	-0.88	4.83 ± 0.76	-1.05	4.10 ± 0.52
	20	4.97 ± 0.79	2.05	3.80 ± 0.62	1.32	4.39 ± 0.66
	30	5.12 ± 0.54	1.41	4.24 ± 0.80	1.44	5.41 ± 0.78
	40	5.13 ± 0.55	0.32	4.83 ± 0.62	-1.45	3.65 ± 0.82
	50	5.85 ± 0.76	1.82	4.24 ± 0.91	-1.57	3.07 ± 0.82
	60	5.85 ± 0.61	0.87	5.11 ± 0.97	-1.78	3.80 ± 0.73

Table 8: Comparative results for the Breast data set.

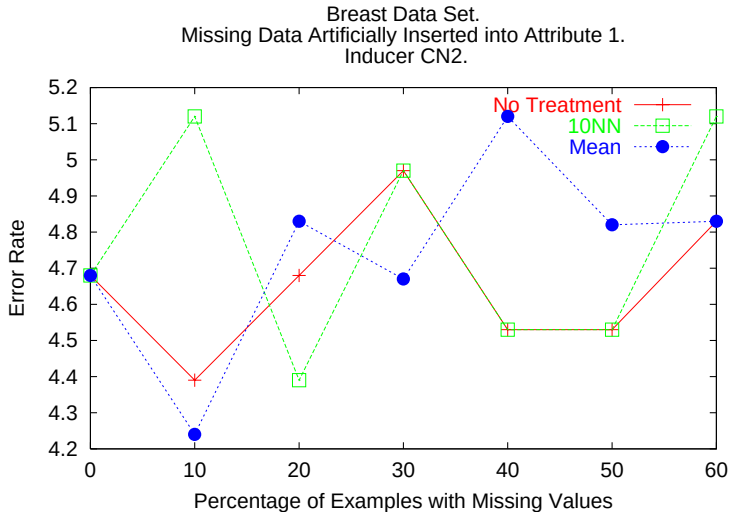

Figure 27: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Breast data set. See Table 8 on the page before for numerical results.

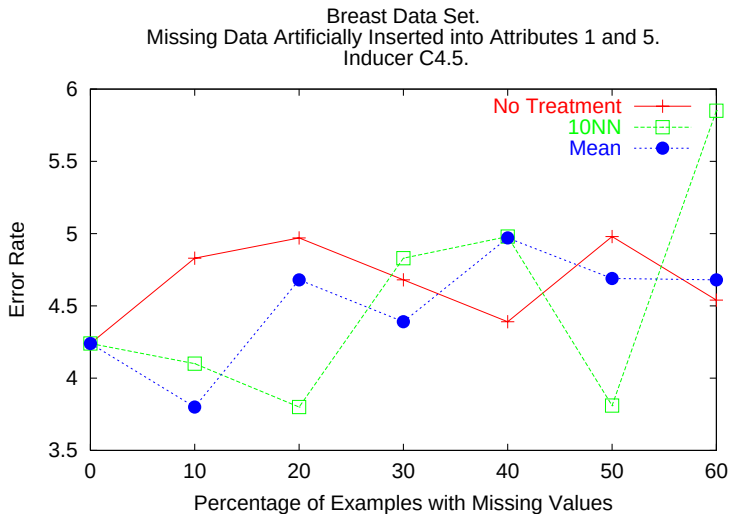

Figure 28: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Breast data set. See Table 8 on the page before for numerical results.

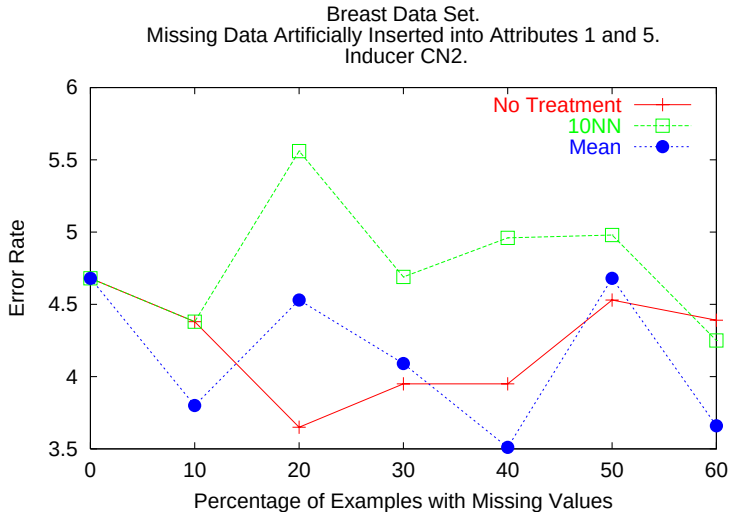

Figure 29: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Breast data set. See Table 8 on page 37 for numerical results.

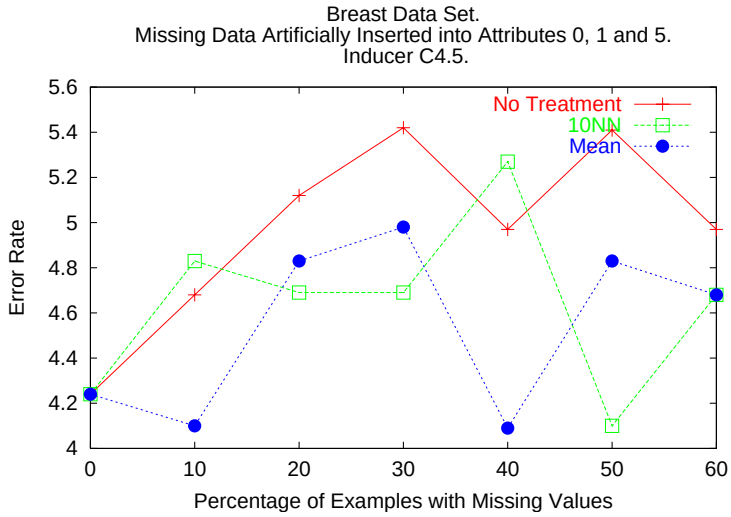

Figure 30: Comparison of C4.5 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Breast data set. See Table 8 on page 37 for numerical results.

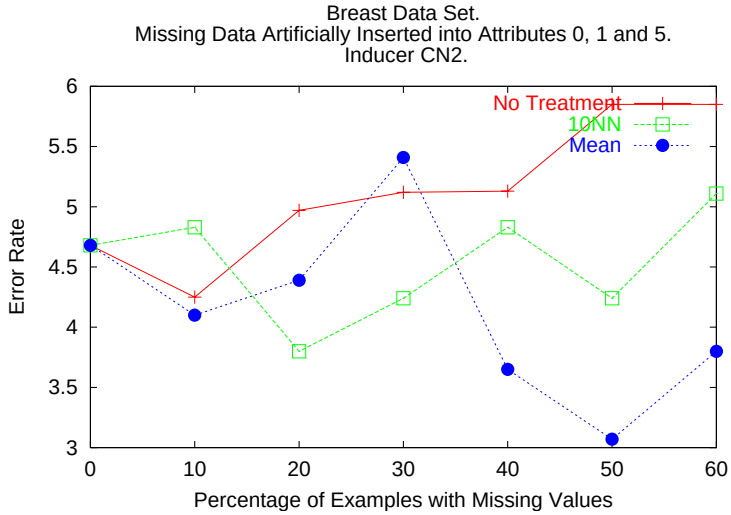

Figure 31: Comparison of CN2 internal strategy, 10-NNI and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Breast data set. See Table 8 on page 37 for numerical results.

This scenario is interesting because 10-NNI was able to predict the missing data with higher precision than the MEAN OR MODE IMPUTATION. As explained before, since missing values were artificially implanted into the data, the mean square error (*mse*) between the predicted values and the actual ones can be measured. These errors are presented in Table 7 on page 36 for the three most relevant attributes of Breast data set.

If the 10-NNI method is more accurate in predicting missing values, why this higher accuracy is not translated into a more precise classifier? The answer may be in the high correlation among the data set attributes and, because (or consequently) Breast data set has several attributes with similar predicting power.

In order to perform a deeper analysis, we need to verify how each attribute is used into the induced classifier. For instance, it is interesting to understand how C4.5 was able to obtain a constant error rate even with high levels of missing values inserted into attribute 1 (Figure 26 on page 36). Analysing the decision trees generated by C4.5, it is possible to verify that C4.5 was able to substitute attribute 1 — Uniformity of Cell Size — by attribute 2 — Uniformity of Cell Shape. This substitution was possible because these two attributes have a high level of correlation (linear correlation coefficient $r = 0.9072$). In a general way, for Breast data set, C4.5 was able to replace every attribute with missing values by other attributes, and still be competitive with 10-NNI. Using the highest level of the decision tree in which the attribute was incorporated as a heuristical measure of attribute importance in the model, Table 9 shows that C4.5 was able to gradually discard the attributes with missing values as the amount of missing data increased. In a similar way, C4.5 shows a tendency to discard the attributes with missing values when those attributes were treated with MEAN OR MODE IMPUTATION. This result is expected since in MEAN OR MODE IMPUTATION all missing values are replaced by the same value (the attribute mean or mode). Consequently, the attribute discriminatory power (measured by the C4.5 decision tree algorithm through entropy) tends to decrease. The same did not occur when the missing data were treated by 10-NNI. In this scenario, C4.5 kept the attributes with missing values as the upmost attributes into the decision tree. This situation would had been an advantage if Breast data set do not have other attributes with similar predicting power.

% of Missing	No Imputation			Mean/Mode			10-NNI		
	Attr. 1	Attr. 5	Attr. 0	Attr. 1	Attr. 5	Attr. 0	Attr. 1	Attr. 5	Attr. 0
0%	1	2	3	1	2	3	1	2	3
10%	2	2	3	2	2	3	1	2	3
20%	-	2	3	-	3	3	1	2	3
30%	-	5	-	-	3	-	1	2	3
40%	5	4	-	3	-	-	1	2	3
50%	-	-	-	6	7	3	1	3	2
60%	-	5	-	-	3	-	1	2	3

Table 9: Level in which the attributes 1, 5 and 0 of Breast data set were incorporated into the decision tree induced by C4.5. "-" means that the attribute was not selected to be part of the tree. Level 1 is the root of the decision tree.

8 Conclusions and Limitations

This work analyses the behaviour of four methods for missing data treatment: the 10-NNI method using a K-NEAREST NEIGHBOUR algorithm for missing data imputation; the MEAN OR

MODE IMPUTATION; and the internal algorithms used by C4.5 and CN2 to treat missing data. These methods were analysed inserting different percentages of missing data into different attributes of four data sets, showing promising results. For the data sets Bupa, CMC and Pima, the 10-NNI method provides very good results in most cases, even for training sets having a large amount of missing data.

In fact, for Bupa, CMC and Pima data sets, the internal methods used by C4.5 and CN2 to treat missing data show lower error rates compared to 10-NNI in only 11 of 108 measurements (8 for C4.5 and 3 for CN2), Tables 4, 5 and 6. In just one of these 11 measurements the internal methods show a statistically significant difference. On the other hand, 10-NNI shows statistically significant difference in 35 measurements (13 of them are highly significant). Comparing 10-NNI to the MEAN OR MODE IMPUTATION method, the MEAN OR MODE IMPUTATION shows lower error rate in 20 of 108 measurements (5 for C4.5 and 15 for CN2), one of them with highly significant difference. 10-NNI shows statistically significant differences in 11 measurements (3 of them are highly significant).

The Breast data set provided a valuable insight into the limitations of the missing data treatment methods. The first decision to be taken is if the attribute should be treated. The existence of others attributes with similar information (high correlation), or similar predicting power can make the missing data imputation useless, or even harmful. Missing data imputation can be harmful because even the most advanced imputation method is only able to approximate the actual (missing) value. The predicted values are usually more well-behaved, since they conform with other attributes values. In the experiments carried out, as more attributes with missing values were inserted and as the amount of missing data increased, more simple were the induced models. In this way, missing data imputation should be carefully applied, under the risk of oversimplifying the problem under study.

In future works, the missing data treatment methods will be analyzed in other data sets. Furthermore, in this work missing values were inserted completely at random (MCAR). In a future work, we will analyze the behaviour of these methods when missing values are not randomly distributed. In this case, there is a possibility of creating invalid knowledge. For an effective analysis, we will have to inspect not only the error rate, but also the quality of the knowledge induced by the learning system.

A Complementary Results: Runs with 1, 3, 5, 10, 20, 30, 50 and 100 of Nearest Neighbours

This section presents the complete results with several values of nearest neighbours. Although these graphs are rather confusing, they are useful to identify tendencies. For instance, in some data sets, the K-NEAREST NEIGHBOUR imputation was superior to the other missing data treatments for several values of k. This tendency reinforces that the superiority of 10-NNI, in those data sets, was not merely result of chance.

In this appendix we present complementary results that were not shown in Sections 7.3.1 to 7.3.4 on pages 21–36. Sections B.5 to A.3 on pages 80–51 present the results in graphical form for the Breast, Bupa, CMC and Pima data sets.

A.1 Bupa Data Set

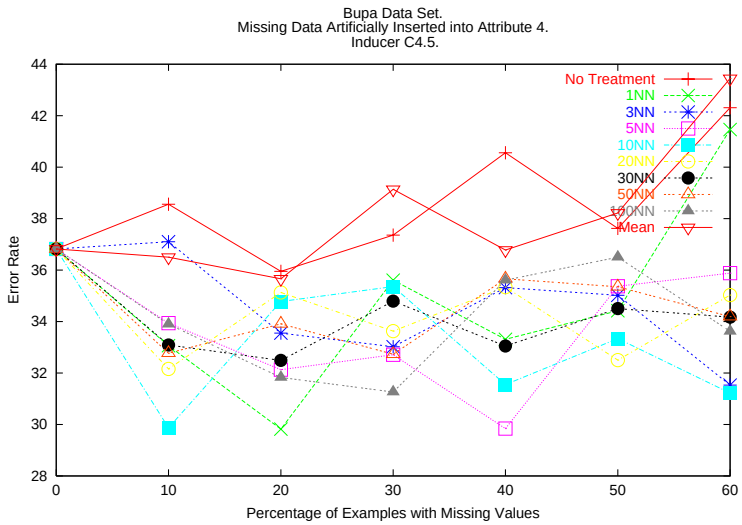

Figure 32: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 4 of Bupa data set.

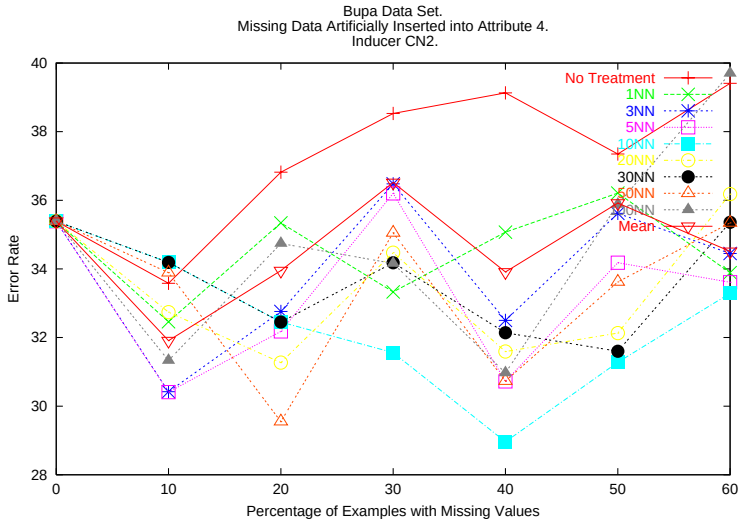

Figure 33: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 4 of Bupa data set.

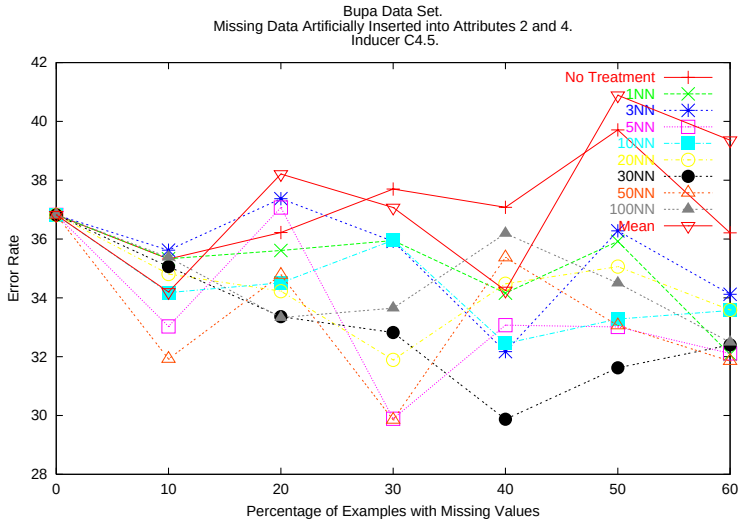

Figure 34: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2 and 4 of Bupa data set.

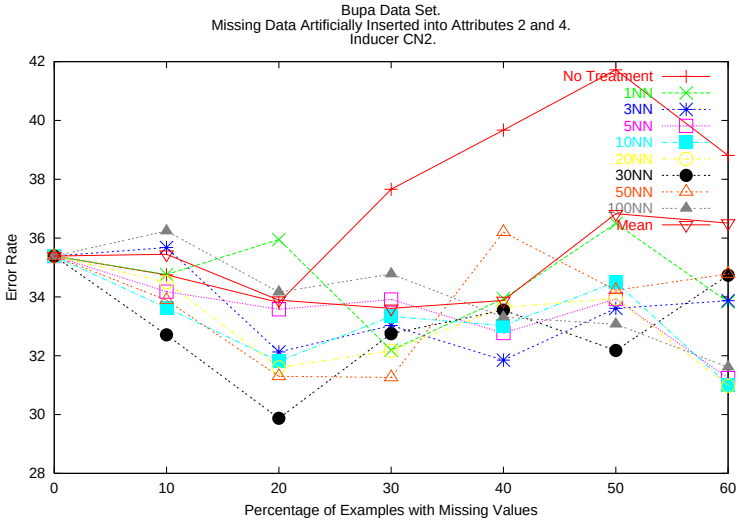

Figure 35: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2 and 4 of Bupa data set.

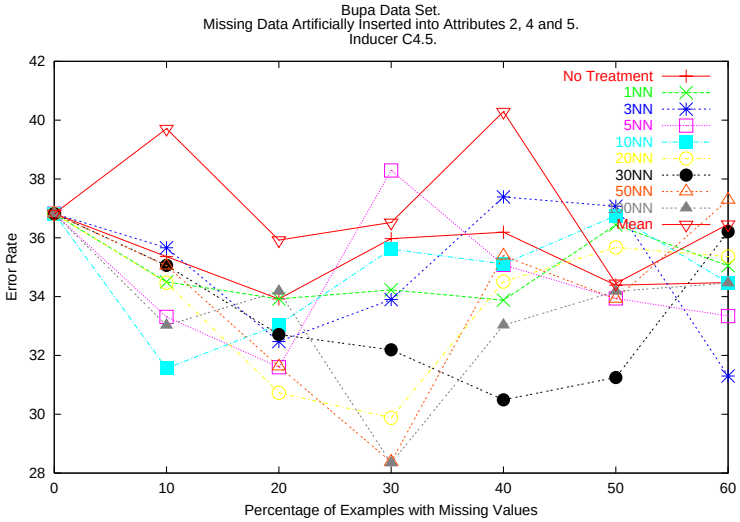

Figure 36: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2, 4 and 5 of Bupa data set.

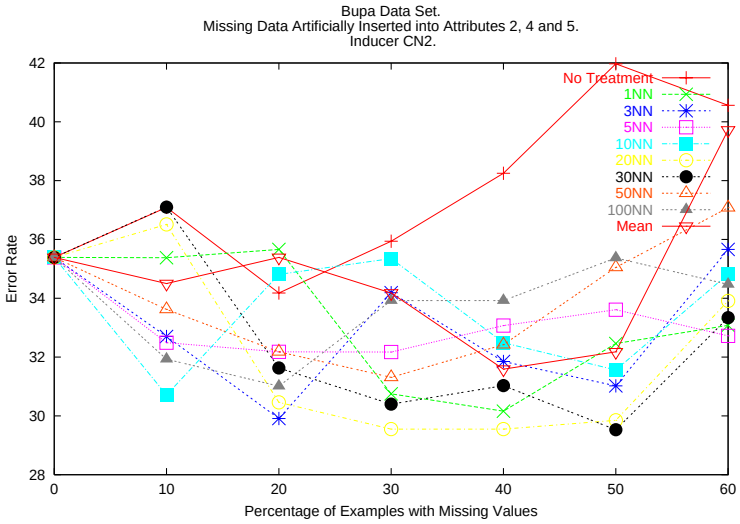

Figure 37: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 2, 4 and 5 of Bupa data set.

A.2 CMC Data Set

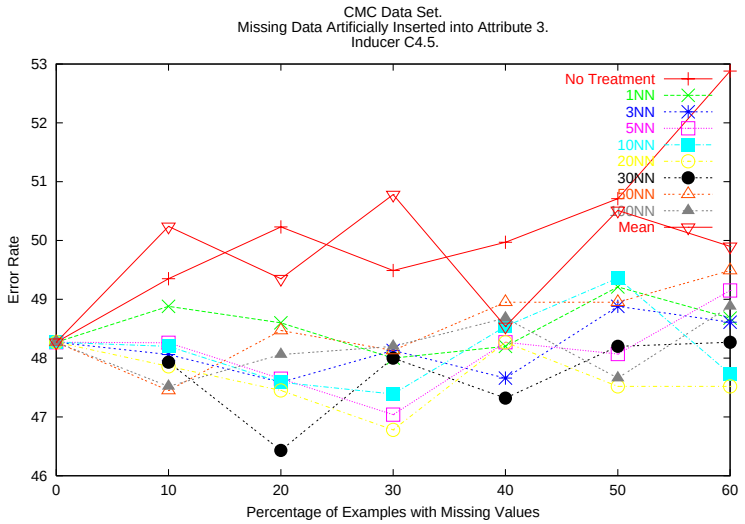

Figure 38: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 3 of CMC data set.

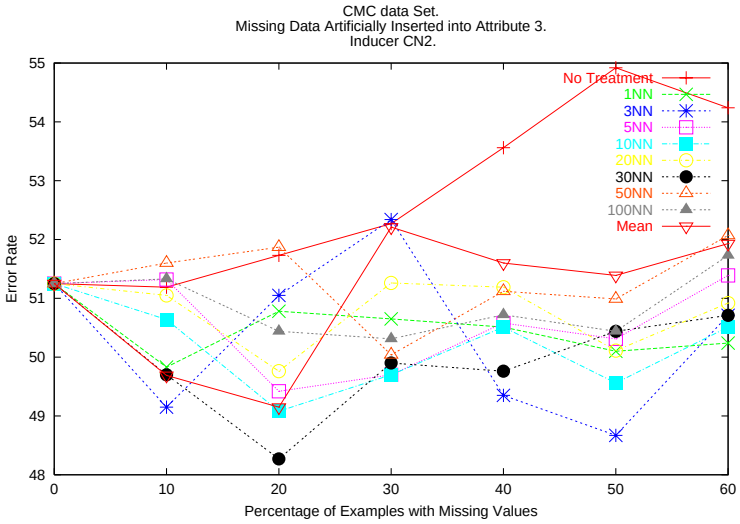

Figure 39: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 3 of CMC data set.

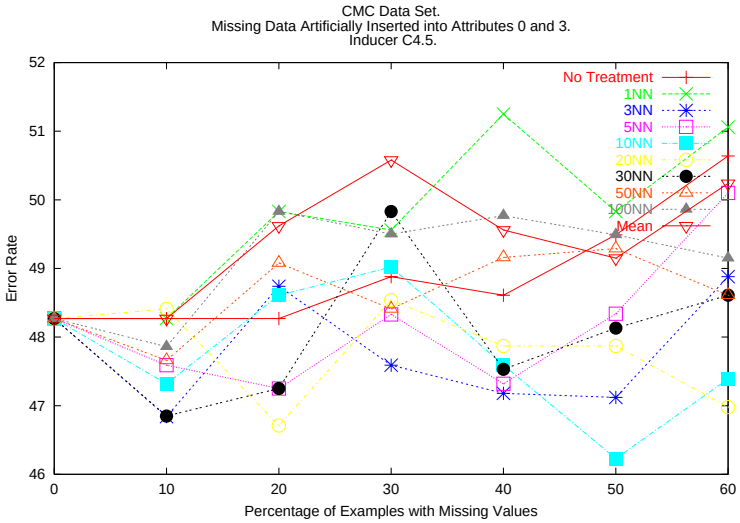

Figure 40: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0 and 3 of CMC data set.

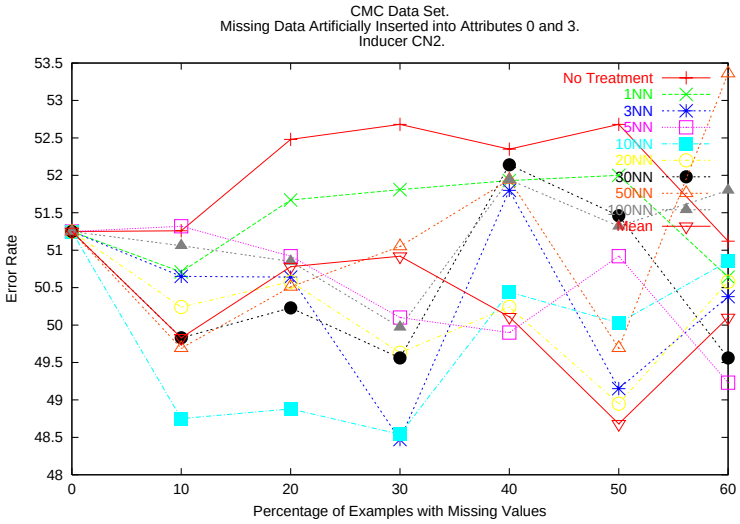

Figure 41: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0 and 3 of CMC data set.

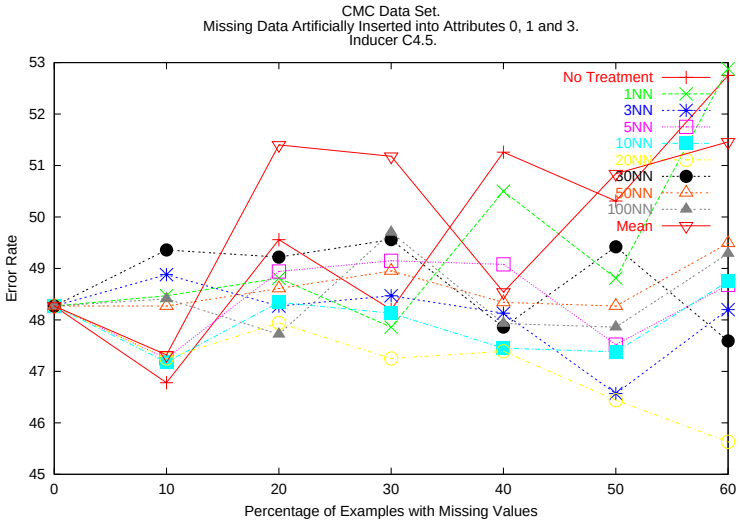

Figure 42: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 3 of CMC data set.

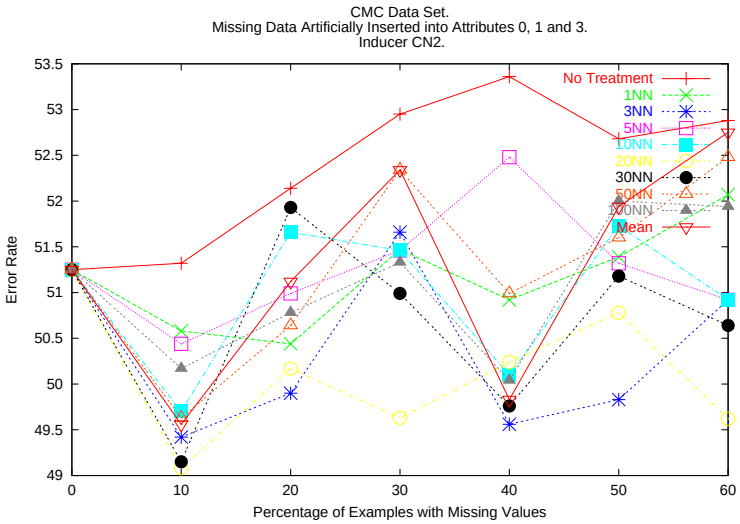

Figure 43: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 3 of CMC data set.

A.3 Pima Data Set

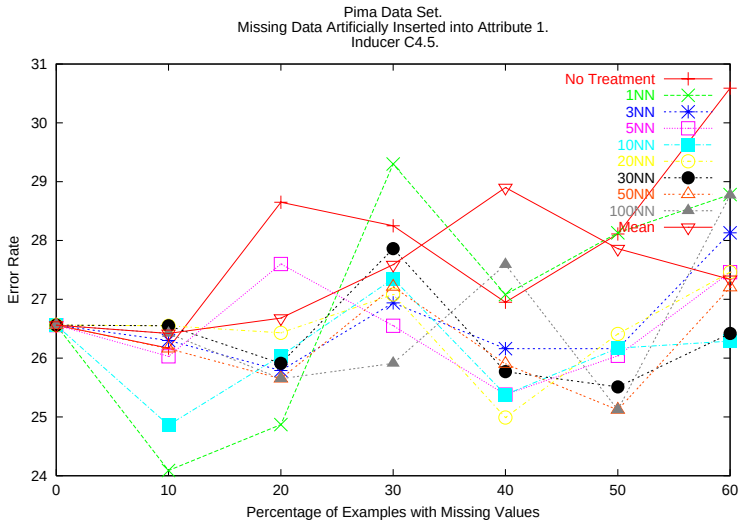

Figure 44: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Pima data set.

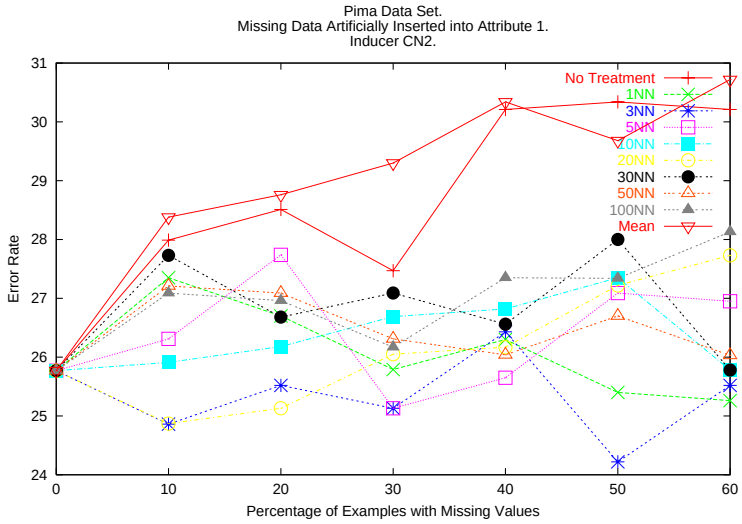

Figure 45: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Pima data set.

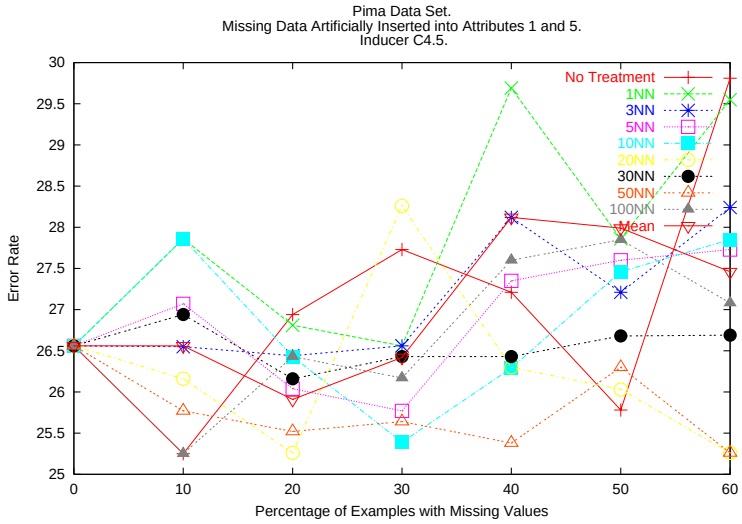

Figure 46: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Pima data set.

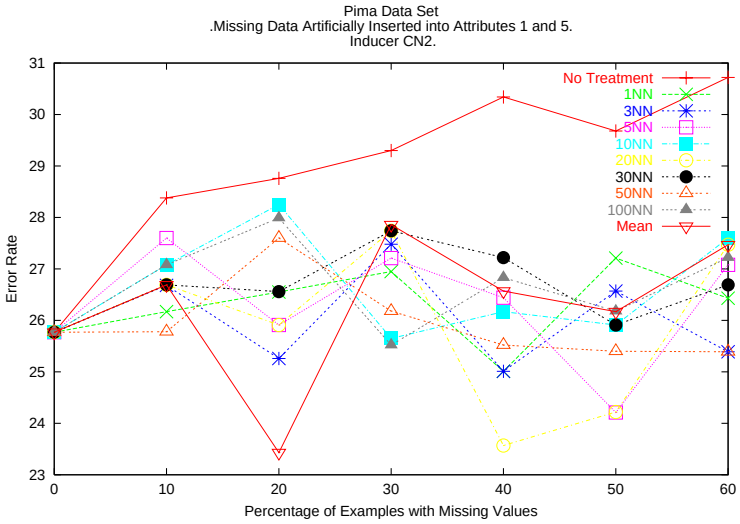

Figure 47: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Pima data set.

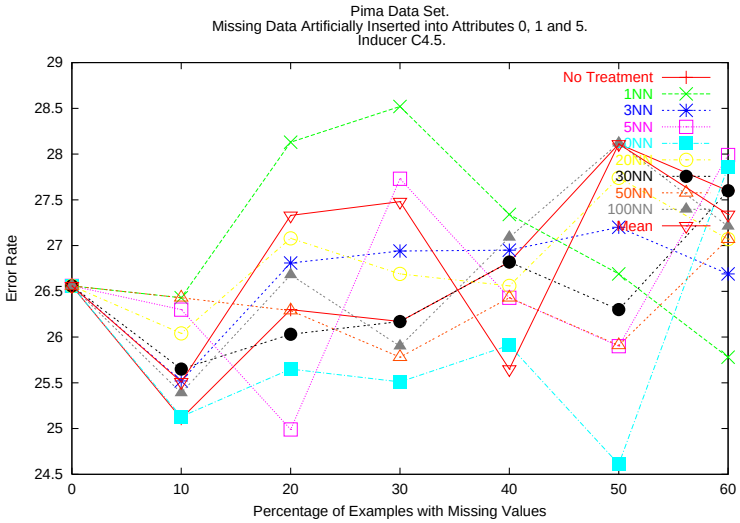

Figure 48: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Pima data set.

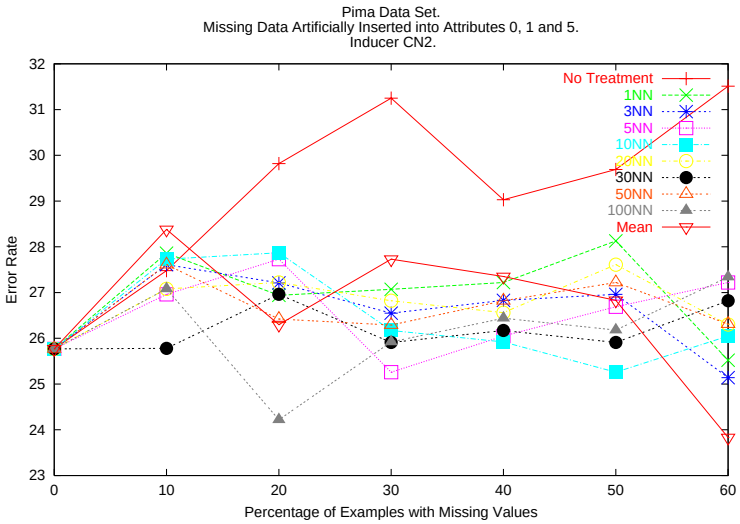

Figure 49: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Pima data set.

B Complementary Results: Mean Squared Errors for Actual and Predicted Missing Values

As stated before, the K-NEAREST NEIGHBOUR algorithm has, as its main parameter, the number of nearest neighbours used to make a prediction, identified by the letter k. Due to the complexity of the data we deal with, the optimal value of the k can vary from data set to data set. However, we can still find a value of k that leads to a good performance of the K-NEAREST NEIGHBOUR in several domains. This value will be used as default in our implementation of the k nearest neighbour algorithm for imputation of missing data.

As previously described, in our experiments missing values were artificially implanted into one, two and three selected attributes, in different proportions (from 10% up to 60%) and treated with 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours, as well as MEAN OR MODE IMPUTATION.

As we implanted missing values artificially, the actual value for each missing value is known. Thus, the error among actual and predicted values can be measured. Means and standard deviations of these errors can be used as a statistic to evaluate the performance of each parameter value (k). Also, the error obtained by the MEAN OR MODE IMPUTATION can be used as a basis for analysing if the K-NEAREST NEIGHBOUR computation is worth or a simpler method could provide similar results.

In this appendix we present complementary results that were not shown in Section 7.2 on page 16. Sections B.1 to B.4 on pages 55–74 present the results in both graphical and numerical forms. Rows marked with the labels *Mean* or *Mode* are results of MEAN OR MODE IMPUTATION. These results are included as a basis for comparing the results obtained with different values of k.

B.1 Breast Data Set

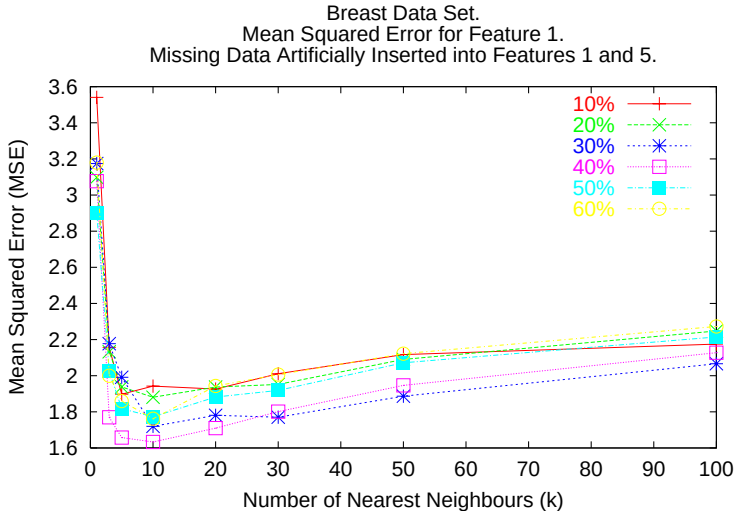

Figure 50: Breast data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1 and 5. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[9.34 \pm 0.31, 10.31 \pm 0.39]$. See Table 11 on page 59 for numerical results.

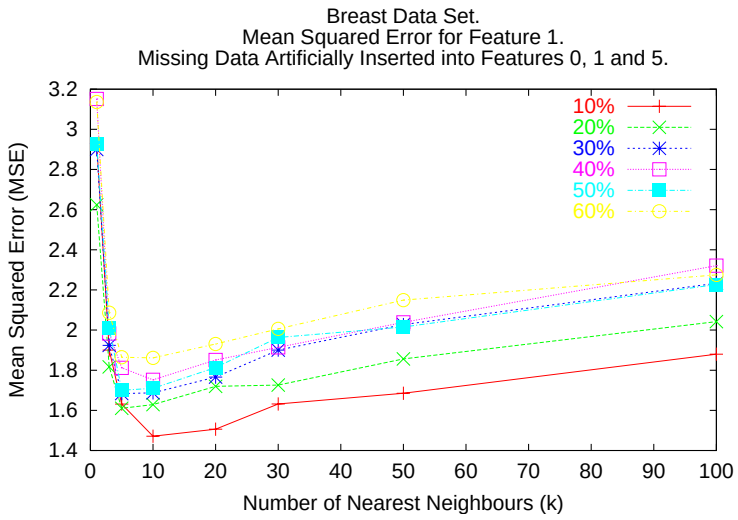

Figure 51: Breast data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[9.15 \pm 0.39, 10.07 \pm 0.45]$. See Table 12 on page 60 for numerical results.

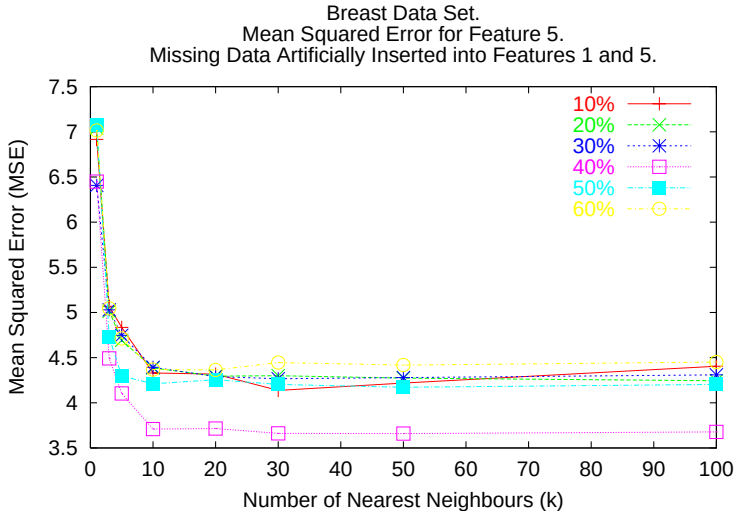

Figure 52: Breast data set. Mean squared error measured over attribute 5 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[12.88 \pm 0.40, 13.58 \pm 0.74]$. See Table 13 on page 60 for numerical results.

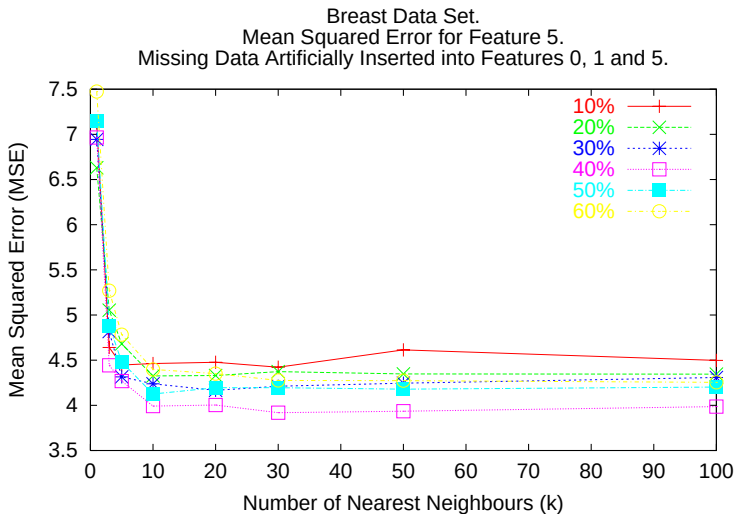

Figure 53: Breast data set. Mean squared error measured over attribute 5 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[12.42 \pm 1.01, 14.45 \pm 0.69]$. See Table 14 on page 61 for numerical results.

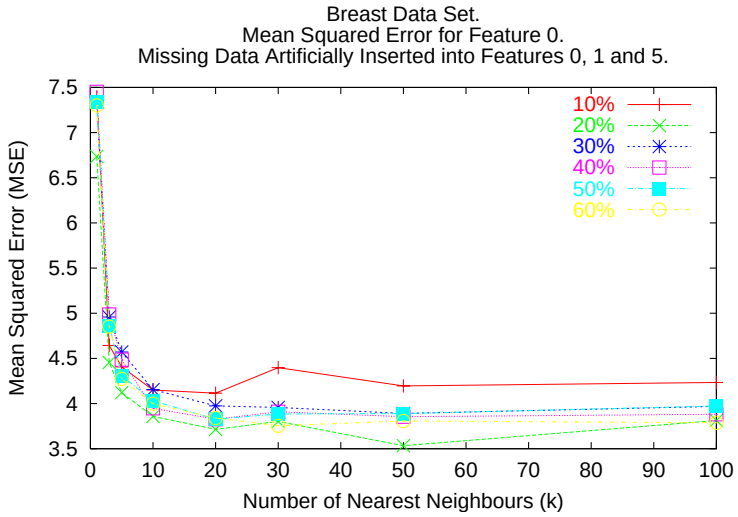

Figure 54: Breast data set. Mean squared error measured over attribute 0 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[7.36 \pm 0.26, 7.98 \pm 0.17]$. See Table 15 on page 61 for numerical results.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	2.73 ± 0.22	3.10 ± 0.18	2.90 ± 0.15
03	1.73 ± 0.15	1.79 ± 0.10	1.88 ± 0.12
05	1.42 ± 0.14	1.64 ± 0.11	1.64 ± 0.09
10	1.48 ± 0.17	1.68 ± 0.09	1.69 ± 0.10
20	1.55 ± 0.15	1.76 ± 0.09	1.71 ± 0.10
30	1.61 ± 0.14	1.81 ± 0.10	1.79 ± 0.11
50	1.74 ± 0.16	1.97 ± 0.10	1.90 ± 0.13
100	1.96 ± 0.16	2.19 ± 0.10	2.08 ± 0.14
Mean	8.98 ± 0.33	9.27 ± 0.36	9.31 ± 0.28
	40 %	50 %	60 %
01	3.04 ± 0.10	3.04 ± 0.09	3.11 ± 0.14
03	1.93 ± 0.09	1.93 ± 0.06	1.93 ± 0.07
05	1.73 ± 0.09	1.82 ± 0.07	1.81 ± 0.08
10	1.67 ± 0.06	1.82 ± 0.05	1.83 ± 0.08
20	1.77 ± 0.06	1.94 ± 0.04	1.95 ± 0.08
30	1.84 ± 0.06	1.98 ± 0.05	2.00 ± 0.08
50	1.96 ± 0.05	2.10 ± 0.05	2.11 ± 0.07
100	2.19 ± 0.07	2.26 ± 0.07	2.29 ± 0.07
Mean	9.36 ± 0.22	9.19 ± 0.19	9.39 ± 0.12

Table 10: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 1, with missing data inserted into attribute 1.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	3.54 ± 0.39	3.10 ± 0.27	3.17 ± 0.20
03	2.16 ± 0.26	2.13 ± 0.18	2.18 ± 0.11
05	1.90 ± 0.22	1.94 ± 0.13	1.99 ± 0.09
10	1.94 ± 0.19	1.88 ± 0.14	1.72 ± 0.07
20	1.93 ± 0.17	1.94 ± 0.12	1.78 ± 0.07
30	2.01 ± 0.15	1.95 ± 0.10	1.77 ± 0.09
50	2.12 ± 0.17	2.09 ± 0.11	1.89 ± 0.11
100	2.18 ± 0.11	2.25 ± 0.09	2.07 ± 0.09
Mean	10.31 ± 0.39	9.87 ± 0.53	9.54 ± 0.32
	40 %	50 %	60 %
01	3.08 ± 0.19	2.90 ± 0.16	3.18 ± 0.10
03	1.77 ± 0.11	2.02 ± 0.10	2.00 ± 0.10
05	1.66 ± 0.11	1.82 ± 0.08	1.86 ± 0.09
10	1.63 ± 0.11	1.77 ± 0.07	1.76 ± 0.08
20	1.71 ± 0.11	1.88 ± 0.06	1.94 ± 0.10
30	1.80 ± 0.12	1.92 ± 0.08	2.01 ± 0.08
50	1.95 ± 0.12	2.07 ± 0.08	2.12 ± 0.08
100	2.13 ± 0.14	2.21 ± 0.09	2.27 ± 0.08
Mean	9.55 ± 0.41	9.34 ± 0.31	9.66 ± 0.20

Table 11: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 1, with missing data inserted into attributes 1 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	2.93 ± 0.41	2.62 ± 0.26	2.90 ± 0.13
03	1.90 ± 0.28	1.82 ± 0.15	1.92 ± 0.08
05	1.63 ± 0.22	1.61 ± 0.15	1.68 ± 0.09
10	1.47 ± 0.21	1.63 ± 0.16	1.69 ± 0.11
20	1.51 ± 0.16	1.72 ± 0.16	1.77 ± 0.13
30	1.63 ± 0.20	1.73 ± 0.15	1.90 ± 0.12
50	1.68 ± 0.18	1.86 ± 0.17	2.03 ± 0.14
100	1.88 ± 0.17	2.04 ± 0.19	2.23 ± 0.14
Mean	9.56 ± 0.78	9.22 ± 0.76	10.07 ± 0.45
	40 %	50 %	60 %
01	3.15 ± 0.13	2.93 ± 0.13	3.14 ± 0.17
03	1.98 ± 0.11	2.01 ± 0.11	2.09 ± 0.09
05	1.81 ± 0.09	1.70 ± 0.09	1.86 ± 0.09
10	1.75 ± 0.09	1.71 ± 0.09	1.86 ± 0.13
20	1.85 ± 0.10	1.81 ± 0.11	1.93 ± 0.14
30	1.91 ± 0.11	1.96 ± 0.11	2.01 ± 0.13
50	2.04 ± 0.11	2.02 ± 0.13	2.15 ± 0.14
100	2.32 ± 0.12	2.23 ± 0.14	2.27 ± 0.13
Mean	9.15 ± 0.39	9.30 ± 0.27	9.16 ± 0.31

Table 12: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 1, with missing data inserted into attributes 1, 5 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	6.92 ± 0.78	7.04 ± 0.62	6.41 ± 0.44
03	5.07 ± 0.62	5.01 ± 0.32	5.03 ± 0.33
05	4.83 ± 0.70	4.69 ± 0.18	4.75 ± 0.34
10	4.33 ± 0.57	4.39 ± 0.23	4.39 ± 0.29
20	4.32 ± 0.62	4.30 ± 0.25	4.29 ± 0.26
30	4.14 ± 0.55	4.30 ± 0.24	4.27 ± 0.24
50	4.22 ± 0.56	4.27 ± 0.25	4.28 ± 0.22
100	4.40 ± 0.61	4.24 ± 0.22	4.31 ± 0.22
Mean	13.58 ± 0.74	13.35 ± 0.35	12.88 ± 0.40
	40 %	50 %	60 %
01	6.45 ± 0.43	7.07 ± 0.41	7.01 ± 0.37
03	4.49 ± 0.21	4.73 ± 0.24	5.04 ± 0.26
05	4.10 ± 0.21	4.30 ± 0.26	4.71 ± 0.27
10	3.71 ± 0.24	4.21 ± 0.25	4.37 ± 0.24
20	3.72 ± 0.24	4.26 ± 0.21	4.36 ± 0.25
30	3.66 ± 0.25	4.20 ± 0.21	4.45 ± 0.26
50	3.66 ± 0.26	4.17 ± 0.19	4.42 ± 0.27
100	3.68 ± 0.25	4.20 ± 0.19	4.45 ± 0.27
Mean	12.90 ± 0.24	13.04 ± 0.28	13.54 ± 0.30

Table 13: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 5, with missing data inserted into attributes 1 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	7.13 ± 0.84	6.63 ± 0.32	6.94 ± 0.59
03	4.64 ± 0.34	5.05 ± 0.33	4.82 ± 0.42
05	4.45 ± 0.46	4.68 ± 0.34	4.31 ± 0.35
10	4.46 ± 0.62	4.33 ± 0.26	4.24 ± 0.36
20	4.48 ± 0.71	4.33 ± 0.27	4.17 ± 0.37
30	4.42 ± 0.75	4.37 ± 0.26	4.21 ± 0.38
50	4.61 ± 0.76	4.35 ± 0.28	4.25 ± 0.39
100	4.50 ± 0.76	4.35 ± 0.31	4.31 ± 0.39
Mean	12.42 ± 1.01	14.45 ± 0.69	13.30 ± 0.61
	40 %	50 %	60 %
01	6.96 ± 0.49	7.15 ± 0.27	7.47 ± 0.38
03	4.44 ± 0.23	4.88 ± 0.16	5.27 ± 0.23
05	4.27 ± 0.19	4.48 ± 0.11	4.78 ± 0.22
10	3.99 ± 0.22	4.13 ± 0.12	4.40 ± 0.21
20	4.00 ± 0.24	4.19 ± 0.14	4.35 ± 0.19
30	3.92 ± 0.24	4.20 ± 0.14	4.28 ± 0.19
50	3.94 ± 0.22	4.18 ± 0.15	4.27 ± 0.17
100	3.99 ± 0.23	4.20 ± 0.11	4.25 ± 0.14
Mean	13.42 ± 0.42	13.54 ± 0.29	13.16 ± 0.24

Table 14: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 5, with missing data inserted into attributes 1, 5 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
01	7.39 ± 0.67	6.73 ± 0.30	7.33 ± 0.25
03	4.64 ± 0.34	4.45 ± 0.14	4.96 ± 0.15
05	4.41 ± 0.41	4.12 ± 0.16	4.57 ± 0.09
10	4.15 ± 0.27	3.86 ± 0.13	4.15 ± 0.12
20	4.12 ± 0.21	3.71 ± 0.13	3.97 ± 0.10
30	4.40 ± 0.20	3.80 ± 0.15	3.96 ± 0.08
50	4.20 ± 0.19	3.53 ± 0.10	3.89 ± 0.07
100	4.23 ± 0.26	3.81 ± 0.12	3.97 ± 0.08
Mean	7.54 ± 0.44	7.55 ± 0.32	7.36 ± 0.26
	40 %	50 %	60 %
01	7.45 ± 0.29	7.34 ± 0.25	7.30 ± 0.19
03	4.99 ± 0.23	4.86 ± 0.21	4.85 ± 0.14
05	4.49 ± 0.17	4.31 ± 0.17	4.27 ± 0.10
10	3.95 ± 0.11	4.03 ± 0.11	4.00 ± 0.11
20	3.83 ± 0.07	3.83 ± 0.10	3.84 ± 0.09
30	3.91 ± 0.06	3.89 ± 0.10	3.75 ± 0.12
50	3.85 ± 0.11	3.89 ± 0.10	3.81 ± 0.10
100	3.88 ± 0.11	3.97 ± 0.12	3.79 ± 0.09
Mean	7.85 ± 0.33	7.92 ± 0.17	7.98 ± 0.17

Table 15: Mean squared errors (mse) between actual and predicted values for Breast data set. mse measured over attribute 0, with missing data inserted into attributes 1, 5 and 0.

B.2 Bupa Data Set

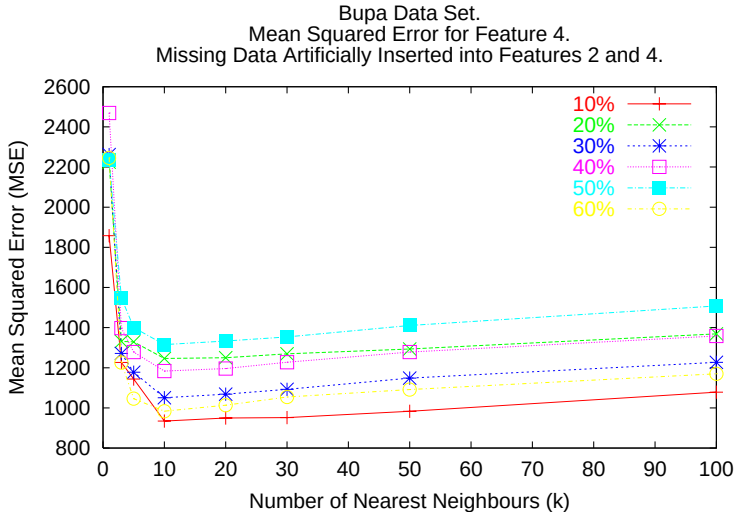

Figure 55: Bupa data set. Mean squared error measured over attribute 4 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 4 and 2. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[1317.46 \pm 114.82, 1653.72 \pm 143.55]$. See Table 17 on page 65 for numerical results.

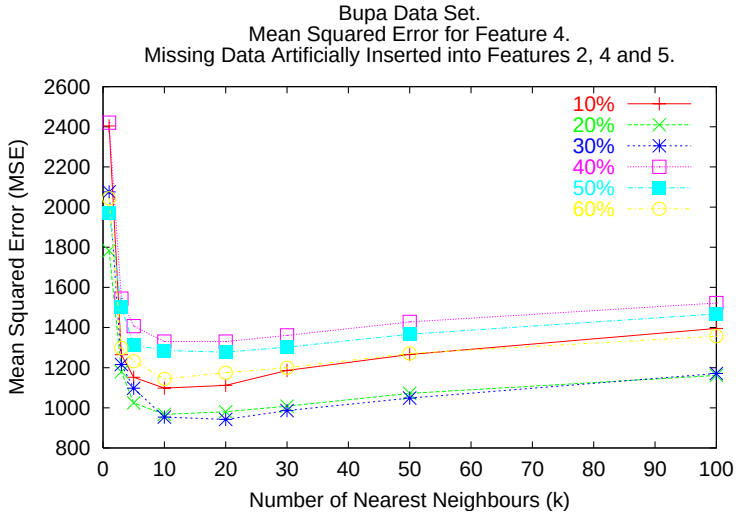

Figure 56: Bupa data set. Mean squared error measured over attribute 4 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 4, 2 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[1345.47 \pm 232.06, 1653.57 \pm 219.26]$. See Table 18 on page 66 for numerical results.

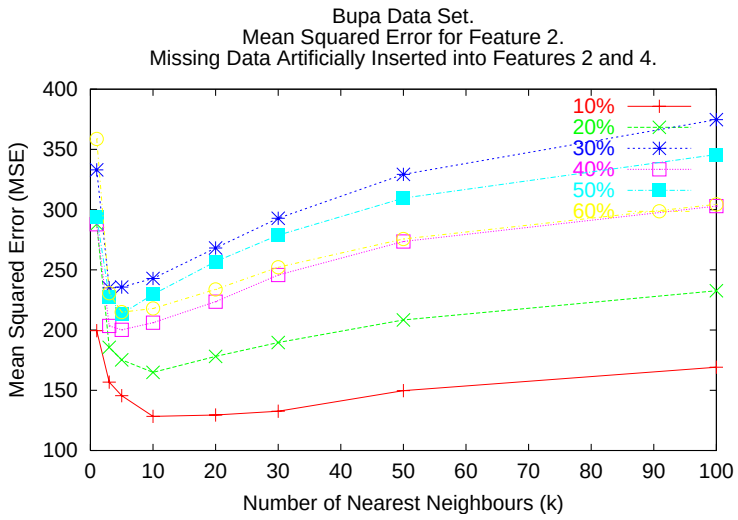

Figure 57: Bupa data set. Mean squared error measured over attribute 2 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 4 and 2. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[204.59 \pm 32.60, 430.56 \pm 70.66]$. See Table 19 on page 66 for numerical results.

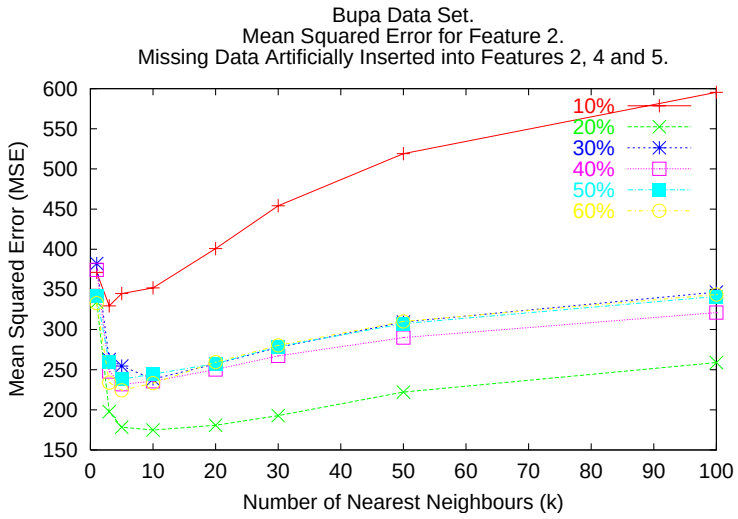

Figure 58: Bupa data set. Mean squared error measured over attribute 2 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 4, 2 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[304.99 \pm 48.07, 680.38 \pm 214.23]$. See Table 20 on page 67 for numerical results.

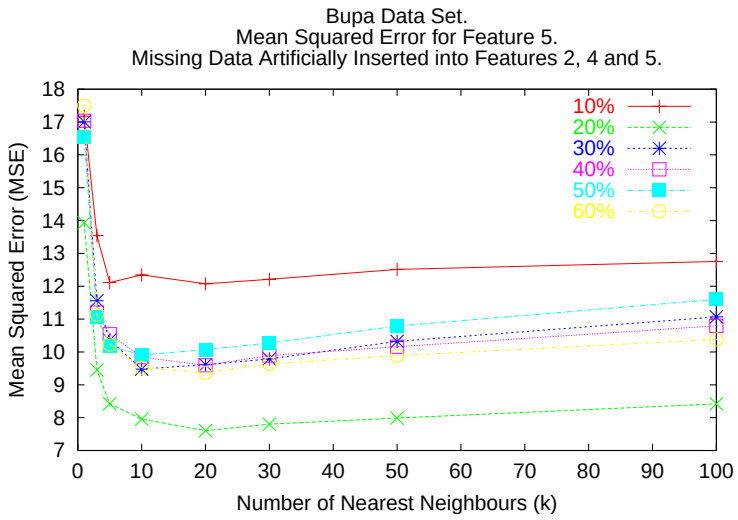

Figure 59: Bupa data set. Mean squared error measured over attribute 5 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 4, 2 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[9.04 \pm 0.90, 13.05 \pm 2.63.]$. See Table 21 on page 67 for numerical results.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	2357.51 ± 384.63	2257.49 ± 282.43	1666.94 ± 100.26
003	1427.45 ± 344.43	1448.79 ± 186.85	1113.78 ± 84.95
005	1340.91 ± 377.19	1347.44 ± 208.20	1022.04 ± 101.05
010	1192.20 ± 390.11	1247.98 ± 217.89	896.15 ± 109.37
020	1204.52 ± 406.43	1262.04 ± 227.76	872.28 ± 120.28
030	1222.75 ± 408.93	1301.28 ± 231.87	909.52 ± 125.18
050	1247.37 ± 416.53	1362.21 ± 245.34	964.80 ± 128.21
100	1368.45 ± 436.86	1488.90 ± 255.00	1054.16 ± 130.64
Mean	1562.19 ± 438.78	1693.74 ± 250.33	1226.04 ± 130.79
	40 %	50 %	60 %
001	2124.82 ± 130.59	1995.38 ± 125.22	1905.49 ± 129.64
003	1495.17 ± 104.63	1504.58 ± 88.87	1421.06 ± 71.97
005	1413.88 ± 105.91	1445.83 ± 88.14	1348.83 ± 64.64
010	1326.50 ± 103.19	1376.03 ± 98.89	1299.20 ± 61.89
020	1367.29 ± 109.51	1406.37 ± 102.14	1316.66 ± 66.45
030	1409.43 ± 107.17	1444.15 ± 102.89	1362.23 ± 66.96
050	1472.53 ± 113.03	1496.37 ± 104.84	1407.72 ± 71.25
100	1560.15 ± 117.13	1583.75 ± 109.85	1493.80 ± 64.84
Mean	1704.55 ± 118.03	1700.95 ± 100.88	1616.44 ± 56.69

Table 16: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 4, with missing data inserted into attribute 4.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	1857.98 ± 418.80	2224.12 ± 409.27	2262.10 ± 302.16
003	1226.44 ± 300.46	1331.21 ± 335.25	1271.48 ± 147.43
005	1143.76 ± 250.90	1329.03 ± 360.10	1177.71 ± 139.01
010	934.68 ± 216.65	1246.23 ± 336.10	1050.58 ± 122.66
020	949.48 ± 288.32	1250.22 ± 352.70	1068.44 ± 136.32
030	951.90 ± 322.93	1269.51 ± 360.23	1092.72 ± 138.45
050	983.08 ± 337.46	1293.68 ± 366.69	1148.71 ± 150.65
100	1078.46 ± 382.73	1368.99 ± 372.96	1227.20 ± 164.19
Mean	1321.47 ± 459.16	1532.41 ± 378.43	1384.51 ± 165.42
	40 %	50 %	60 %
001	2469.27 ± 258.34	2234.11 ± 163.41	2245.70 ± 215.57
003	1397.30 ± 98.99	1548.30 ± 138.27	1226.09 ± 87.44
005	1278.14 ± 91.14	1400.32 ± 142.62	1045.79 ± 79.59
010	1183.32 ± 99.76	1315.34 ± 138.63	983.70 ± 84.81
020	1195.92 ± 94.77	1333.35 ± 139.29	1013.58 ± 94.32
030	1227.38 ± 98.23	1353.37 ± 141.41	1054.52 ± 99.84
050	1278.31 ± 101.89	1410.63 ± 147.85	1091.22 ± 109.39
100	1358.39 ± 101.35	1508.12 ± 149.05	1170.16 ± 114.90
Mean	1513.01 ± 97.17	1653.72 ± 143.55	1317.46 ± 114.82

Table 17: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 4, with missing data inserted into attributes 4 and 2.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	2404.31 ± 738.95	1782.91 ± 382.28	2075.53 ± 264.69
003	1264.74 ± 318.60	1179.92 ± 226.35	1216.18 ± 114.25
005	1151.99 ± 303.02	1024.39 ± 225.22	1097.60 ± 113.37
010	1098.29 ± 309.77	967.60 ± 250.57	953.82 ± 102.68
020	1111.75 ± 327.78	980.63 ± 251.62	942.27 ± 121.90
030	1186.02 ± 365.17	1008.44 ± 247.77	986.08 ± 139.19
050	1265.54 ± 384.82	1072.35 ± 248.83	1048.74 ± 149.61
100	1395.49 ± 432.20	1161.65 ± 252.10	1170.96 ± 170.86
Mean	1598.74 ± 432.57	1345.47 ± 232.06	1371.33 ± 181.65
	40 %	50 %	60 %
001	2420.43 ± 291.04	1969.52 ± 140.96	2043.85 ± 253.03
003	1544.22 ± 174.17	1502.10 ± 118.36	1298.37 ± 104.90
005	1407.02 ± 172.69	1313.29 ± 120.16	1231.77 ± 97.41
010	1329.80 ± 174.20	1285.79 ± 120.98	1144.44 ± 85.65
020	1329.81 ± 189.77	1276.95 ± 120.26	1174.80 ± 94.17
030	1360.70 ± 205.97	1302.55 ± 115.96	1199.93 ± 94.81
050	1428.05 ± 214.06	1366.51 ± 118.38	1270.84 ± 101.98
100	1521.85 ± 218.77	1467.49 ± 118.76	1356.93 ± 107.42
Mean	1653.57 ± 219.26	1601.17 ± 112.51	1478.30 ± 113.32

Table 18: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 4, with missing data inserted into attributes 4, 2 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	199.47 ± 19.05	287.98 ± 53.09	333.02 ± 59.78
003	156.79 ± 30.99	185.64 ± 24.42	235.08 ± 38.17
005	145.56 ± 32.69	175.26 ± 23.21	235.69 ± 37.52
010	128.33 ± 26.85	164.90 ± 22.48	242.83 ± 38.49
020	129.44 ± 25.26	178.16 ± 27.98	268.13 ± 45.85
030	132.59 ± 26.00	189.62 ± 32.91	292.85 ± 51.66
050	149.70 ± 30.28	208.38 ± 39.32	329.23 ± 57.86
100	169.19 ± 33.40	232.65 ± 45.67	374.71 ± 65.99
Mean	204.59 ± 32.60	282.65 ± 50.52	430.56 ± 70.66
	40 %	50 %	60 %
001	287.86 ± 37.89	294.01 ± 23.80	358.75 ± 40.54
003	203.46 ± 16.12	227.21 ± 14.93	230.65 ± 28.19
005	200.17 ± 19.20	213.58 ± 12.85	214.82 ± 23.68
010	206.14 ± 19.84	229.60 ± 14.94	217.94 ± 30.51
020	223.59 ± 24.07	256.85 ± 18.39	233.81 ± 33.11
030	245.72 ± 29.53	278.80 ± 20.06	252.22 ± 36.56
050	273.57 ± 35.09	309.52 ± 22.77	275.67 ± 39.92
100	302.98 ± 40.77	345.74 ± 24.97	304.38 ± 42.53
Mean	348.26 ± 47.19	393.30 ± 26.59	344.94 ± 43.25

Table 19: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 2, with missing data inserted into attributes 4 and 2.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	371.16 ± 71.66	334.14 ± 72.26	382.19 ± 42.94
003	329.51 ± 103.15	197.75 ± 26.40	262.59 ± 22.69
005	344.76 ± 112.40	178.44 ± 22.97	254.49 ± 21.37
010	351.83 ± 112.73	174.84 ± 24.74	238.30 ± 20.32
020	400.76 ± 129.03	180.72 ± 24.04	257.20 ± 25.91
030	454.19 ± 145.98	192.76 ± 27.19	277.86 ± 31.35
050	519.02 ± 166.79	222.09 ± 34.34	309.31 ± 38.19
100	595.44 ± 192.08	258.87 ± 41.78	346.62 ± 45.13
Mean	680.38 ± 214.23	304.99 ± 48.07	401.99 ± 51.61
	40 %	50 %	60 %
001	374.34 ± 56.01	341.55 ± 38.61	332.84 ± 26.88
003	247.76 ± 39.08	259.26 ± 27.14	233.83 ± 19.70
005	231.85 ± 41.05	238.18 ± 26.18	224.31 ± 24.45
010	235.57 ± 46.35	244.41 ± 30.01	233.23 ± 29.72
020	250.21 ± 54.54	257.53 ± 30.76	259.33 ± 35.54
030	267.25 ± 59.58	278.70 ± 36.07	280.27 ± 39.03
050	289.91 ± 67.60	307.38 ± 42.12	310.13 ± 43.93
100	321.09 ± 73.76	341.06 ± 45.00	343.59 ± 49.53
Mean	368.15 ± 77.42	377.74 ± 48.25	373.45 ± 51.06

Table 20: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 2, with missing data inserted into attributes 4, 2 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	17.18 ± 2.99	13.93 ± 1.73	17.00 ± 0.96
003	13.54 ± 2.54	9.45 ± 1.17	11.56 ± 0.83
005	12.11 ± 2.37	8.42 ± 0.75	10.36 ± 0.93
010	12.35 ± 2.27	7.96 ± 0.71	9.48 ± 0.88
020	12.07 ± 2.24	7.60 ± 0.70	9.61 ± 1.10
030	12.21 ± 2.46	7.80 ± 0.75	9.79 ± 1.20
050	12.51 ± 2.65	7.99 ± 0.80	10.32 ± 1.35
100	12.76 ± 2.79	8.42 ± 0.88	11.07 ± 1.47
Mean	13.05 ± 2.63	9.04 ± 0.90	11.53 ± 1.44
	40 %	50 %	60 %
001	17.04 ± 0.88	16.54 ± 0.69	17.50 ± 0.97
003	11.20 ± 0.52	11.06 ± 0.40	11.14 ± 0.51
005	10.54 ± 0.42	10.18 ± 0.35	10.23 ± 0.53
010	9.84 ± 0.47	9.91 ± 0.37	9.52 ± 0.60
020	9.59 ± 0.44	10.07 ± 0.46	9.37 ± 0.72
030	9.89 ± 0.46	10.27 ± 0.51	9.63 ± 0.78
050	10.16 ± 0.50	10.79 ± 0.59	9.89 ± 0.85
100	10.80 ± 0.47	11.60 ± 0.60	10.38 ± 0.89
Mean	11.15 ± 0.54	12.08 ± 0.63	10.81 ± 0.90

Table 21: Mean squared errors (mse) between actual and predicted values for Bupa data set. mse measured over attribute 5, with missing data inserted into attributes 4, 2 and 5.

B.3 CMC Data Set

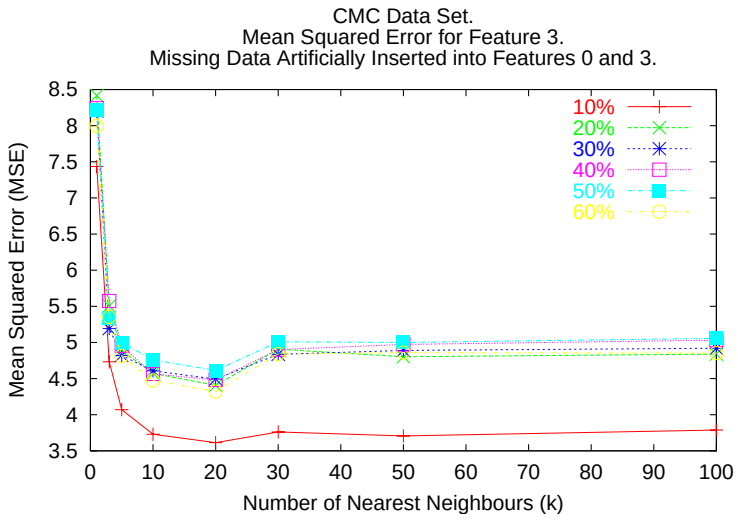

Figure 60: CMC data set. Mean squared error measured over attribute 3 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 3 and 0. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[4.48 \pm 0.25, 5.72 \pm 0.21]$. See Table 23 on page 71 for numerical results.

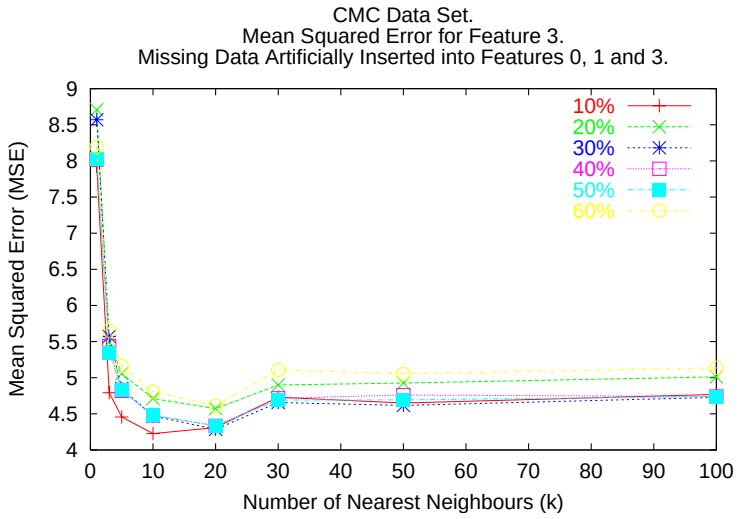

Figure 61: CMC data set. Mean squared error measured over attribute 3 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 3, 0 and 1. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[5.37 \pm 0.10, 5.86 \pm 0.34]$. See Table 24 on page 72 for numerical results.

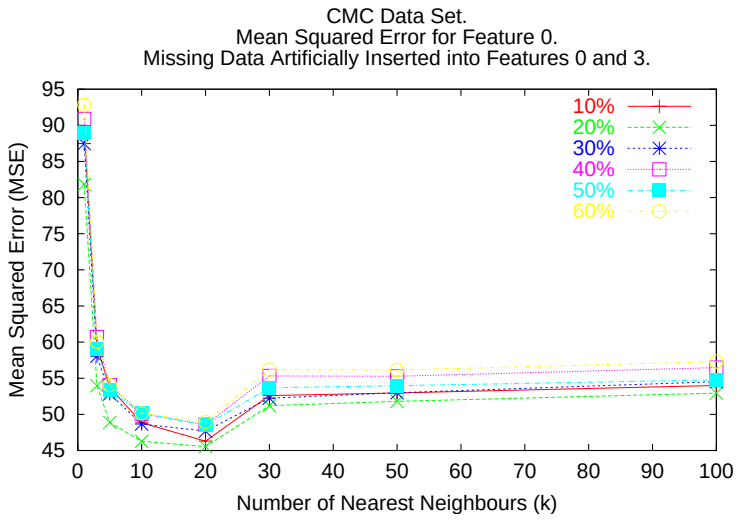

Figure 62: CMC data set. Mean squared error measured over attribute 0 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 3 and 0. MEAN OR MODE IMPUTATION obtained mse errors in the interval $[66.34 \pm 2.13, 68.30 \pm 1.17]$. See Table 25 on page 72 for numerical results.

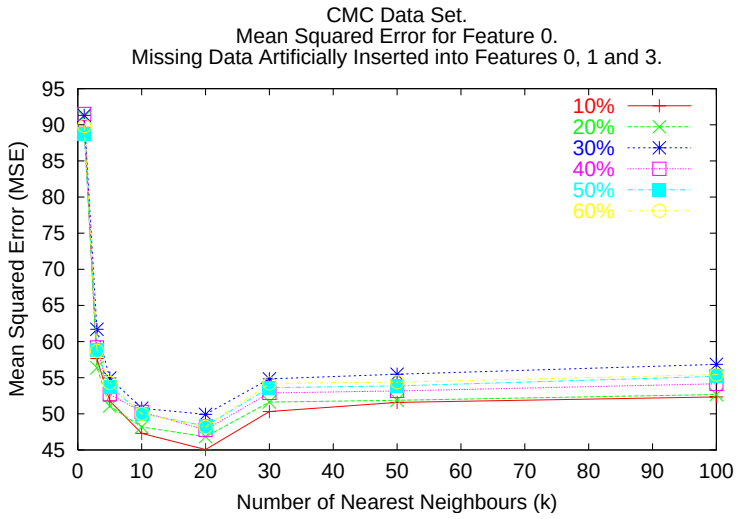

Figure 63: CMC data set. Mean squared error measured over attribute 0 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 3, 0 and 1. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[65.81 \pm 1.74, 70.14 \pm 1.11]$. See Table 26 on page 73 for numerical results.

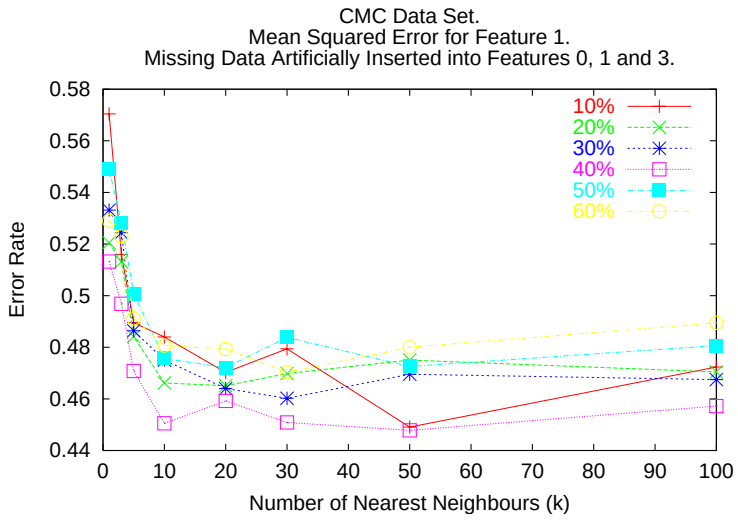

Figure 64: CMC data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 3, 0 and 1. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[0.60 \pm 0.01, 0.62 \pm 0.01]$. See Table 27 on page 73 for numerical results.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	7.20 ± 0.46	7.22 ± 0.29	6.53 ± 0.19
003	4.57 ± 0.22	4.41 ± 0.16	4.24 ± 0.14
005	4.25 ± 0.20	4.00 ± 0.12	3.87 ± 0.13
010	3.91 ± 0.17	3.91 ± 0.13	3.72 ± 0.11
020	3.90 ± 0.20	3.80 ± 0.11	3.66 ± 0.12
030	4.52 ± 0.27	4.36 ± 0.15	4.38 ± 0.16
050	4.56 ± 0.21	4.44 ± 0.16	4.52 ± 0.15
100	4.68 ± 0.24	4.48 ± 0.16	4.57 ± 0.15
Mean	5.97 ± 0.30	5.61 ± 0.16	5.68 ± 0.18
	40 %	50 %	60 %
001	6.60 ± 0.12	6.87 ± 0.17	6.92 ± 0.16
003	4.29 ± 0.07	4.56 ± 0.12	4.65 ± 0.09
005	3.99 ± 0.08	4.17 ± 0.08	4.25 ± 0.09
010	3.74 ± 0.09	3.86 ± 0.05	4.00 ± 0.09
020	3.67 ± 0.08	3.68 ± 0.05	3.83 ± 0.08
030	4.40 ± 0.11	4.48 ± 0.09	4.62 ± 0.10
050	4.43 ± 0.10	4.56 ± 0.09	4.74 ± 0.11
100	4.50 ± 0.09	4.61 ± 0.09	4.76 ± 0.11
Mean	5.54 ± 0.10	5.59 ± 0.10	5.64 ± 0.10

Table 22: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 3, with missing data inserted into attribute 3.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	7.43 ± 0.61	8.42 ± 0.44	8.20 ± 0.25
003	4.73 ± 0.39	5.51 ± 0.39	5.19 ± 0.19
005	4.07 ± 0.25	4.89 ± 0.34	4.82 ± 0.21
010	3.73 ± 0.22	4.57 ± 0.23	4.61 ± 0.19
020	3.61 ± 0.19	4.41 ± 0.23	4.49 ± 0.17
030	3.76 ± 0.21	4.91 ± 0.29	4.84 ± 0.19
050	3.71 ± 0.20	4.80 ± 0.24	4.89 ± 0.20
100	3.79 ± 0.24	4.84 ± 0.26	4.92 ± 0.16
Mean	4.48 ± 0.25	5.60 ± 0.23	5.57 ± 0.17
	40 %	50 %	60 %
001	8.24 ± 0.33	8.21 ± 0.23	7.99 ± 0.25
003	5.57 ± 0.20	5.34 ± 0.14	5.35 ± 0.16
005	4.95 ± 0.19	5.00 ± 0.11	4.81 ± 0.14
010	4.57 ± 0.17	4.76 ± 0.11	4.47 ± 0.10
020	4.48 ± 0.19	4.61 ± 0.10	4.32 ± 0.10
030	4.90 ± 0.19	5.01 ± 0.12	4.84 ± 0.13
050	4.97 ± 0.21	5.00 ± 0.11	4.86 ± 0.11
100	5.03 ± 0.22	5.06 ± 0.12	4.85 ± 0.10
Mean	5.72 ± 0.21	5.70 ± 0.11	5.46 ± 0.12

Table 23: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 3, with missing data inserted into attributes 3 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	7.93 ± 0.45	8.71 ± 0.48	8.57 ± 0.28
003	4.79 ± 0.21	5.49 ± 0.23	5.57 ± 0.19
005	4.46 ± 0.21	5.06 ± 0.25	4.85 ± 0.16
010	4.23 ± 0.22	4.71 ± 0.23	4.47 ± 0.15
020	4.31 ± 0.21	4.57 ± 0.23	4.29 ± 0.15
030	4.73 ± 0.30	4.90 ± 0.29	4.66 ± 0.18
050	4.65 ± 0.25	4.93 ± 0.28	4.62 ± 0.17
100	4.77 ± 0.29	5.01 ± 0.30	4.73 ± 0.20
Mean	5.75 ± 0.39	5.86 ± 0.34	5.47 ± 0.21
	40 %	50 %	60 %
001	8.03 ± 0.24	8.03 ± 0.30	8.20 ± 0.23
003	5.44 ± 0.13	5.35 ± 0.21	5.65 ± 0.13
005	4.82 ± 0.12	4.83 ± 0.17	5.16 ± 0.11
010	4.48 ± 0.10	4.48 ± 0.11	4.81 ± 0.12
020	4.33 ± 0.09	4.34 ± 0.11	4.61 ± 0.11
030	4.72 ± 0.10	4.69 ± 0.13	5.11 ± 0.13
050	4.76 ± 0.12	4.69 ± 0.11	5.05 ± 0.12
100	4.74 ± 0.14	4.74 ± 0.09	5.13 ± 0.11
Mean	5.46 ± 0.15	5.37 ± 0.10	5.70 ± 0.10

Table 24: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 3, with missing data inserted into attributes 3, 0 and 1.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	89.10 ± 6.93	81.83 ± 4.11	87.48 ± 2.54
003	59.47 ± 3.00	53.96 ± 2.20	58.06 ± 1.46
005	53.73 ± 2.56	48.88 ± 1.96	52.83 ± 1.21
010	48.84 ± 2.47	46.27 ± 1.59	48.66 ± 1.07
020	46.30 ± 2.43	45.53 ± 2.02	47.69 ± 0.84
030	52.61 ± 1.53	51.21 ± 1.84	52.24 ± 1.01
050	52.97 ± 1.88	51.81 ± 2.01	52.99 ± 0.95
100	54.00 ± 1.81	52.93 ± 1.96	54.52 ± 0.90
Mean	67.68 ± 1.86	66.34 ± 2.13	68.30 ± 1.17
	40 %	50 %	60 %
001	90.88 ± 2.38	88.99 ± 2.77	92.84 ± 1.85
003	60.69 ± 1.27	59.06 ± 1.71	60.04 ± 1.47
005	54.04 ± 0.83	53.33 ± 1.27	54.08 ± 1.06
010	50.10 ± 0.84	50.16 ± 1.27	50.14 ± 0.79
020	48.58 ± 0.66	48.51 ± 1.19	48.95 ± 0.66
030	55.32 ± 0.57	53.68 ± 1.28	56.19 ± 0.70
050	55.28 ± 0.54	53.97 ± 1.27	56.14 ± 0.55
100	56.46 ± 0.63	54.77 ± 1.27	57.30 ± 0.46
Mean	67.29 ± 0.71	66.66 ± 1.28	67.89 ± 0.63

Table 25: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 0, with missing data inserted into attributes 3 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	89.65 ± 5.01	91.12 ± 4.49	91.30 ± 2.06
003	57.66 ± 3.89	56.37 ± 3.03	61.70 ± 1.36
005	51.79 ± 2.70	51.08 ± 2.60	54.95 ± 1.39
010	47.30 ± 2.06	48.18 ± 2.23	50.77 ± 1.38
020	45.06 ± 1.82	46.83 ± 2.12	49.93 ± 1.33
030	50.32 ± 2.62	51.63 ± 1.43	54.84 ± 1.05
050	51.59 ± 2.99	51.87 ± 1.39	55.48 ± 0.90
100	52.35 ± 3.04	52.68 ± 1.56	56.85 ± 0.83
Mean	66.97 ± 3.35	65.81 ± 1.74	70.14 ± 1.11
	40 %	50 %	60 %
001	91.50 ± 3.19	88.66 ± 2.22	89.83 ± 2.24
003	59.18 ± 2.27	58.80 ± 0.84	58.70 ± 0.71
005	52.70 ± 1.95	53.75 ± 1.11	54.26 ± 0.81
010	50.25 ± 1.78	50.05 ± 1.08	50.01 ± 0.86
020	47.79 ± 1.66	48.31 ± 1.21	48.19 ± 0.89
030	52.89 ± 1.62	53.62 ± 1.33	54.22 ± 1.12
050	53.17 ± 1.53	53.86 ± 1.49	54.37 ± 1.29
100	54.16 ± 1.55	55.19 ± 1.44	55.44 ± 1.34
Mean	65.88 ± 1.56	67.85 ± 1.65	66.68 ± 1.47

Table 26: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 0, with missing data inserted into attributes 3, 0 and 1.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	0.57 ± 0.01	0.52 ± 0.01	0.53 ± 0.01
003	0.52 ± 0.02	0.51 ± 0.01	0.52 ± 0.01
005	0.49 ± 0.02	0.48 ± 0.01	0.49 ± 0.01
010	0.48 ± 0.01	0.47 ± 0.01	0.48 ± 0.01
020	0.47 ± 0.01	0.47 ± 0.01	0.46 ± 0.01
030	0.48 ± 0.02	0.47 ± 0.01	0.46 ± 0.01
050	0.45 ± 0.02	0.48 ± 0.01	0.47 ± 0.01
100	0.47 ± 0.01	0.47 ± 0.01	0.47 ± 0.01
Mean	0.60 ± 0.01	0.60 ± 0.01	0.60 ± 0.01
	40 %	50 %	60 %
001	0.51 ± 0.01	0.55 ± 0.01	0.53 ± 0.01
003	0.50 ± 0.01	0.53 ± 0.01	0.52 ± 0.01
005	0.47 ± 0.01	0.50 ± 0.01	0.49 ± 0.01
010	0.45 ± 0.01	0.48 ± 0.01	0.48 ± 0.01
020	0.46 ± 0.01	0.47 ± 0.01	0.48 ± 0.01
030	0.45 ± 0.01	0.48 ± 0.00	0.47 ± 0.01
050	0.45 ± 0.01	0.47 ± 0.01	0.48 ± 0.01
100	0.46 ± 0.01	0.48 ± 0.01	0.49 ± 0.01
Mean	0.60 ± 0.01	0.61 ± 0.01	0.62 ± 0.01

Table 27: Mean squared errors (mse) between actual and predicted values for CMC data set. mse measured over attribute 1, with missing data inserted into attributes 3, 0 and 1.

B.4 Pima Data Set

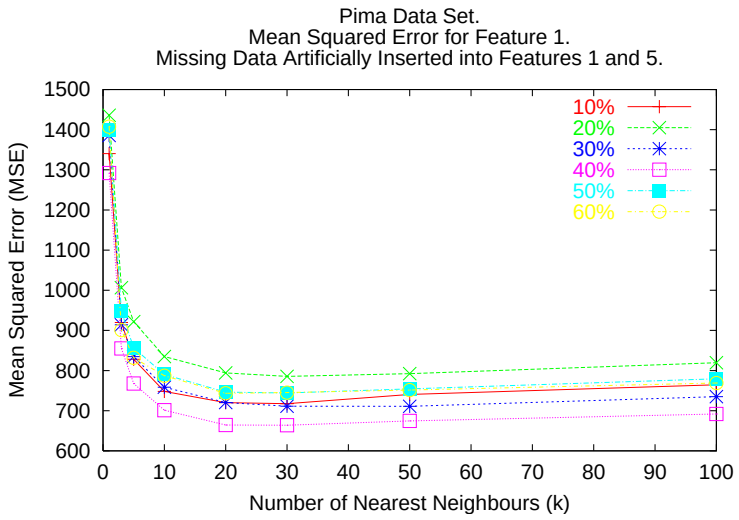

Figure 65: Pima data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[962.82 \pm 26.52, 1101.01 \pm 99.64]$. See Table 29 on page 77 for numerical results.

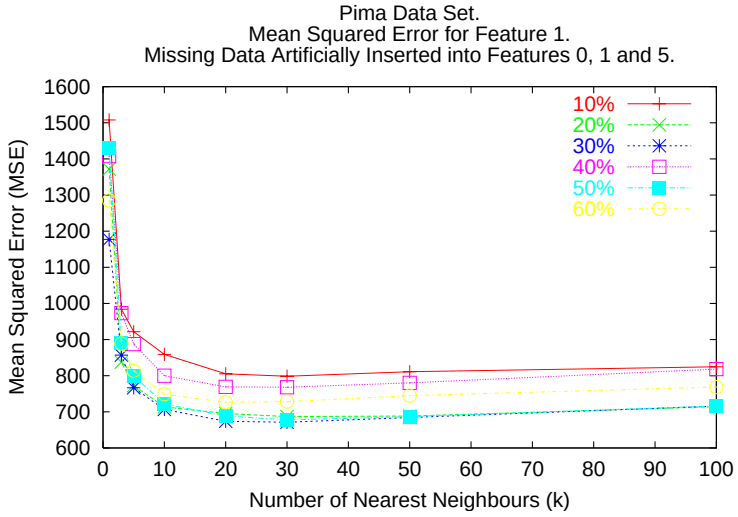

Figure 66: Pima data set. Mean squared error measured over attribute 1 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[993.62 \pm 68.20, 1087.74 \pm 34.74]$. See Table 30 on page 78 for numerical results.

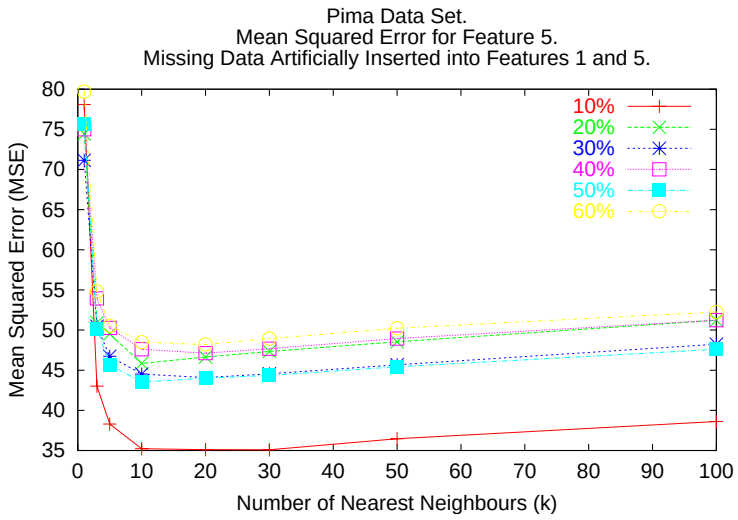

Figure 67: Pima data set. Mean squared error measured over attribute 5 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1 and 5. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[53.91 \pm 8.59, 67.34 \pm 4.59]$. See Table 29 on page 77 for numerical results.

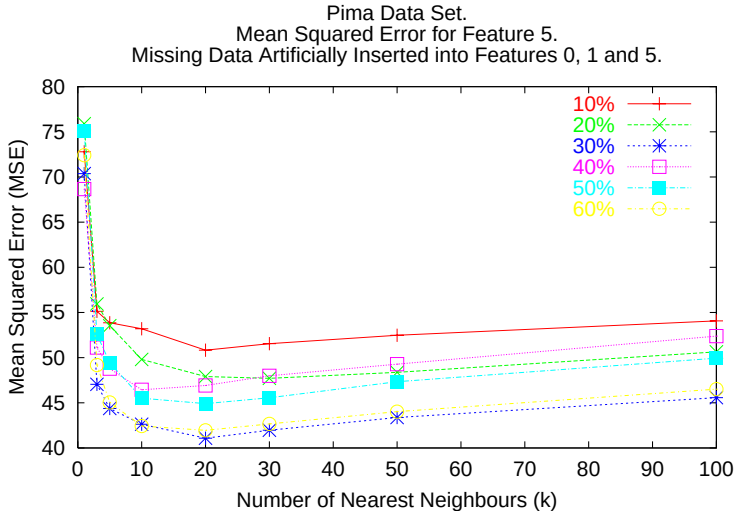

Figure 68: Pima data set. Mean squared error measured over attribute 5 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[57.99 \pm 2.80, 66.91 \pm 8.00]$. See Table 32 on page 79 for numerical results.

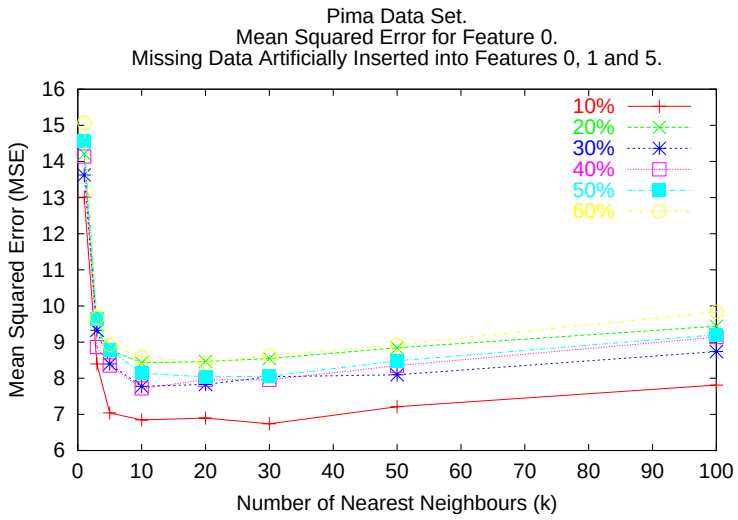

Figure 69: Pima data set. Mean squared error measured over attribute 0 obtained by K-NEAREST NEIGHBOUR imputation. Missing values inserted in attributes 1, 5 and 0. MEAN OR MODE IMPUTATION obtained *mse* errors in the interval $[10.45 \pm 0.58, 12.34 \pm 0.50]$. See Table 33 on page 79 for numerical results.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	1478.34 ± 90.57	1429.07 ± 87.20	1518.19 ± 68.22
003	964.34 ± 48.89	932.95 ± 43.80	964.71 ± 31.08
005	842.08 ± 36.48	852.03 ± 39.39	879.44 ± 29.74
010	755.47 ± 37.94	784.41 ± 31.54	807.95 ± 25.82
020	721.23 ± 35.77	745.19 ± 33.60	765.89 ± 23.30
030	700.10 ± 34.37	728.38 ± 36.12	761.24 ± 24.42
050	715.33 ± 36.97	737.57 ± 36.53	771.85 ± 27.13
100	739.05 ± 39.58	759.04 ± 39.45	787.05 ± 27.96
Mean	1014.94 ± 36.38	1044.24 ± 50.58	1024.33 ± 32.61
	40 %	50 %	60 %
001	1422.23 ± 36.08	1352.07 ± 40.86	1352.92 ± 54.93
003	942.88 ± 29.56	931.82 ± 23.77	924.62 ± 26.38
005	834.05 ± 30.96	851.90 ± 21.05	845.33 ± 18.34
010	771.20 ± 26.67	785.95 ± 18.91	766.66 ± 13.31
020	735.25 ± 27.11	753.00 ± 19.33	736.76 ± 13.62
030	726.52 ± 27.60	750.64 ± 21.15	733.80 ± 13.26
050	738.68 ± 28.50	758.48 ± 21.25	740.99 ± 13.62
100	759.76 ± 28.14	774.56 ± 20.79	756.17 ± 14.19
Mean	989.81 ± 29.45	1038.67 ± 33.05	1004.65 ± 16.11

Table 28: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 1, with missing data inserted into attribute 1.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	1340.39 ± 68.28	1435.96 ± 85.92	1385.55 ± 69.73
003	919.67 ± 102.82	1006.24 ± 78.60	914.02 ± 43.00
005	828.07 ± 104.67	921.74 ± 68.80	834.66 ± 29.19
010	748.18 ± 88.43	834.97 ± 66.91	758.54 ± 32.62
020	720.19 ± 92.88	794.08 ± 59.82	720.22 ± 37.41
030	717.52 ± 94.90	785.56 ± 59.58	711.14 ± 37.39
050	740.61 ± 96.16	792.27 ± 59.48	710.92 ± 38.01
100	764.53 ± 96.59	819.45 ± 64.05	735.05 ± 39.92
Mean	1101.01 ± 99.64	1045.23 ± 61.22	1044.53 ± 35.25
	40 %	50 %	60 %
001	1292.06 ± 44.14	1399.40 ± 61.62	1407.65 ± 69.93
003	855.14 ± 34.90	947.37 ± 46.00	900.99 ± 34.61
005	767.60 ± 36.66	856.27 ± 46.65	831.93 ± 27.43
010	701.35 ± 26.53	790.20 ± 41.39	787.10 ± 22.02
020	664.45 ± 26.97	746.16 ± 39.89	743.37 ± 19.81
030	663.96 ± 27.06	744.21 ± 40.54	745.22 ± 20.17
050	674.55 ± 24.70	754.40 ± 39.05	751.58 ± 20.20
100	691.83 ± 26.95	779.57 ± 40.09	769.00 ± 20.70
Mean	962.82 ± 26.52	1027.65 ± 37.76	1043.68 ± 31.87

Table 29: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 1, with missing data inserted into attributes 1 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	1507.86 ± 175.20	1371.82 ± 75.91	1177.36 ± 58.79
003	983.99 ± 75.87	837.57 ± 41.47	857.05 ± 35.90
005	921.88 ± 55.45	772.81 ± 35.75	766.47 ± 31.85
010	859.13 ± 58.26	713.65 ± 35.89	706.54 ± 34.30
020	805.34 ± 52.96	695.32 ± 41.78	673.70 ± 32.72
030	798.48 ± 51.07	686.91 ± 42.31	671.09 ± 29.82
050	811.04 ± 60.30	688.34 ± 44.67	684.27 ± 31.34
100	824.79 ± 66.27	714.39 ± 48.42	716.13 ± 31.09
Mean	1068.51 ± 63.03	993.62 ± 68.20	1003.20 ± 33.39
	40 %	50 %	60 %
001	1407.73 ± 74.60	1430.09 ± 80.45	1285.06 ± 56.40
003	973.37 ± 32.42	889.88 ± 35.07	888.97 ± 31.50
005	887.45 ± 31.83	799.59 ± 37.23	813.64 ± 31.96
010	800.14 ± 28.10	722.19 ± 36.65	748.53 ± 33.30
020	769.08 ± 27.99	688.75 ± 31.94	726.74 ± 31.04
030	767.95 ± 28.36	677.49 ± 30.30	728.12 ± 29.29
050	780.29 ± 33.16	686.60 ± 29.67	744.81 ± 30.02
100	818.28 ± 36.57	716.23 ± 28.71	768.54 ± 30.68
Mean	1087.74 ± 34.74	998.49 ± 18.92	1014.62 ± 29.23

Table 30: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 1, with missing data inserted into attributes 1, 5 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	78.09 ± 7.67	74.43 ± 4.68	71.14 ± 2.79
003	43.02 ± 3.81	50.93 ± 3.18	50.48 ± 3.08
005	38.30 ± 3.50	49.52 ± 3.60	46.73 ± 2.92
010	35.22 ± 3.26	45.84 ± 3.70	44.53 ± 3.23
020	35.10 ± 3.93	46.64 ± 3.95	44.07 ± 3.93
030	35.10 ± 4.66	47.34 ± 4.04	44.55 ± 4.18
050	36.46 ± 5.20	48.54 ± 4.20	45.67 ± 4.28
100	38.60 ± 5.65	51.22 ± 4.33	48.24 ± 4.42
Mean	53.91 ± 8.59	67.34 ± 4.59	61.72 ± 5.25
	40 %	50 %	60 %
001	75.00 ± 4.10	75.65 ± 2.39	79.71 ± 2.98
003	53.95 ± 2.64	50.18 ± 1.43	54.80 ± 2.01
005	50.26 ± 2.27	45.66 ± 1.30	50.54 ± 1.95
010	47.61 ± 2.37	43.53 ± 1.25	48.50 ± 2.03
020	47.12 ± 2.50	44.00 ± 1.30	48.20 ± 2.36
030	47.68 ± 2.46	44.35 ± 1.35	48.93 ± 2.51
050	48.94 ± 2.59	45.43 ± 1.50	50.22 ± 2.60
100	51.23 ± 2.80	47.60 ± 1.52	52.25 ± 2.72
Mean	63.44 ± 3.37	59.90 ± 1.79	64.26 ± 3.52

Table 31: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 5, with missing data inserted into attributes 1 and 5.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	72.79 ± 6.52	75.89 ± 3.18	70.37 ± 3.99
003	55.13 ± 7.27	55.94 ± 4.55	47.05 ± 4.01
005	53.85 ± 7.55	53.53 ± 4.74	44.37 ± 4.43
010	53.20 ± 7.60	49.80 ± 4.59	42.63 ± 4.11
020	50.82 ± 7.37	47.91 ± 4.15	41.09 ± 3.91
030	51.55 ± 7.13	47.71 ± 4.15	41.97 ± 3.72
050	52.47 ± 7.61	48.37 ± 3.91	43.37 ± 3.68
100	54.07 ± 7.64	50.64 ± 4.09	45.57 ± 3.61
Mean	66.91 ± 8.00	63.25 ± 4.26	58.58 ± 3.88
	40 %	50 %	60 %
001	68.66 ± 3.62	75.09 ± 2.62	72.42 ± 3.52
003	51.12 ± 2.89	52.57 ± 2.08	49.18 ± 2.21
005	48.77 ± 2.50	49.44 ± 2.55	45.03 ± 2.20
010	46.45 ± 2.10	45.51 ± 2.20	42.43 ± 2.20
020	46.91 ± 2.50	44.91 ± 2.51	41.95 ± 2.04
030	47.99 ± 2.47	45.55 ± 2.50	42.67 ± 2.11
050	49.28 ± 2.38	47.34 ± 2.53	44.02 ± 2.20
100	52.39 ± 2.47	49.94 ± 2.80	46.52 ± 2.31
Mean	66.10 ± 2.64	62.99 ± 3.19	57.99 ± 2.80

Table 32: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 5, with missing data inserted into attributes 1, 5 and 0.

k	Mean Squared Error (mse)		
	10 %	20 %	30 %
001	13.01 ± 1.40	14.21 ± 0.74	13.62 ± 0.81
003	8.39 ± 0.49	9.33 ± 0.46	9.33 ± 0.48
005	7.04 ± 0.46	8.79 ± 0.43	8.40 ± 0.44
010	6.85 ± 0.41	8.42 ± 0.42	7.78 ± 0.37
020	6.90 ± 0.41	8.46 ± 0.44	7.83 ± 0.31
030	6.74 ± 0.37	8.55 ± 0.39	8.04 ± 0.37
050	7.21 ± 0.46	8.84 ± 0.41	8.10 ± 0.32
100	7.81 ± 0.53	9.44 ± 0.46	8.74 ± 0.38
Mean	10.45 ± 0.58	12.34 ± 0.50	10.91 ± 0.58
	40 %	50 %	60 %
001	14.14 ± 0.42	14.57 ± 0.58	15.07 ± 0.52
003	8.86 ± 0.30	9.65 ± 0.35	9.67 ± 0.28
005	8.35 ± 0.28	8.79 ± 0.33	8.92 ± 0.22
010	7.72 ± 0.28	8.14 ± 0.24	8.58 ± 0.15
020	7.96 ± 0.39	8.04 ± 0.25	8.43 ± 0.13
030	7.96 ± 0.41	8.05 ± 0.20	8.63 ± 0.14
050	8.35 ± 0.42	8.48 ± 0.23	8.93 ± 0.10
100	9.14 ± 0.42	9.20 ± 0.29	9.84 ± 0.13
Mean	11.31 ± 0.45	11.86 ± 0.34	11.79 ± 0.26

Table 33: Mean squared errors (mse) between actual and predicted values for Pima data set. mse measured over attribute 0, with missing data inserted into attributes 1, 5 and 0.

B.5 Breast Data Set

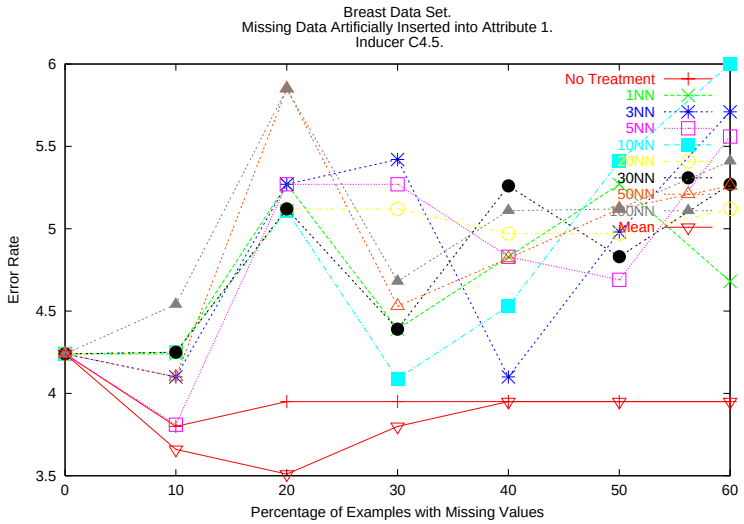

Figure 70: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Breast data set.

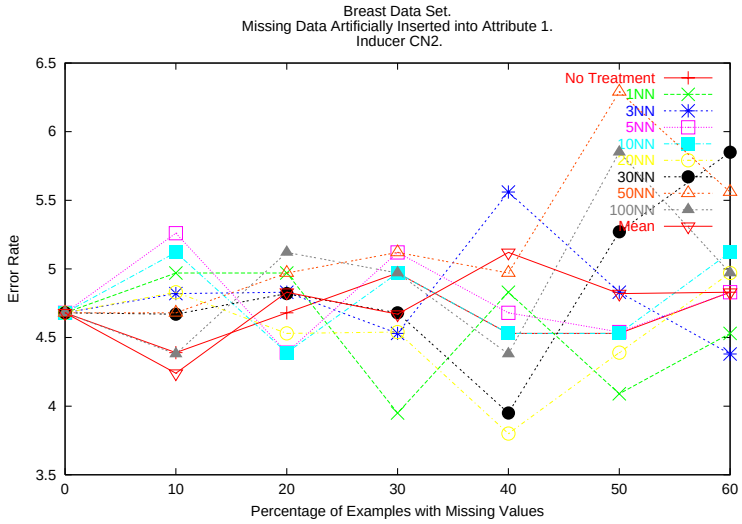

Figure 71: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attribute 1 of Breast data set.

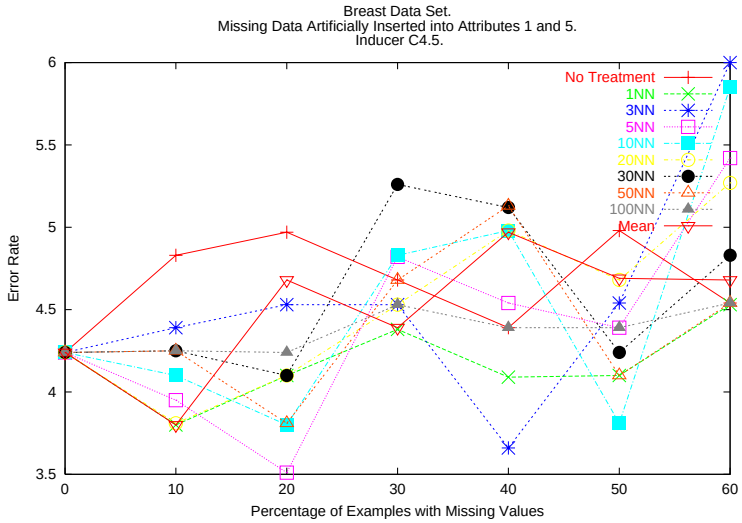

Figure 72: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Breast data set.

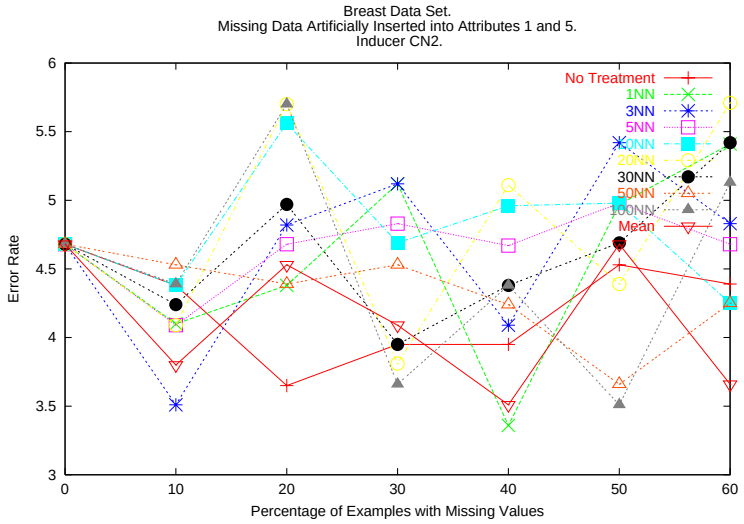

Figure 73: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 1 and 5 of Breast data set.

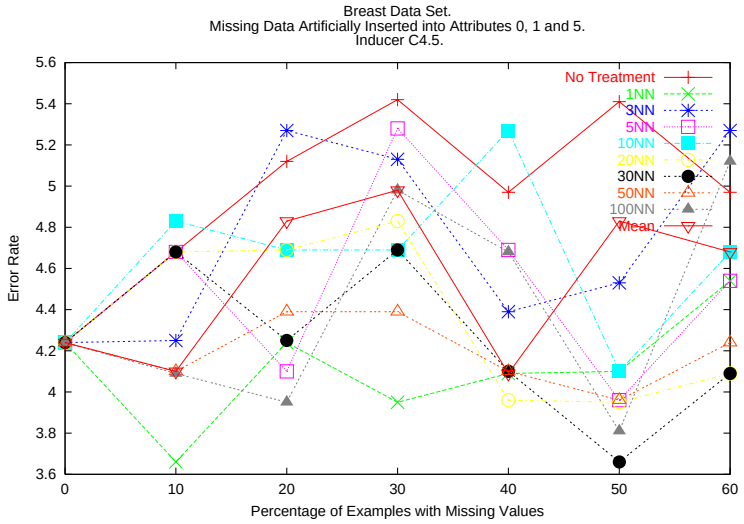

Figure 74: Comparison of C4.5 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Breast data set.

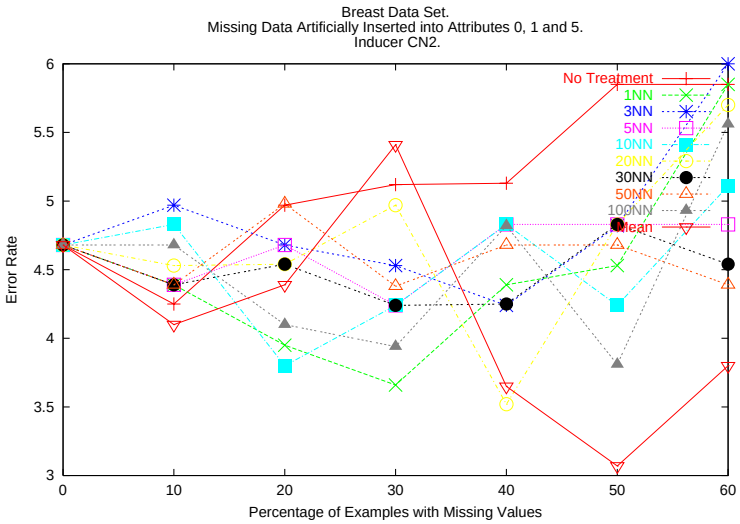

Figure 75: Comparison of CN2 internal strategy, 1, 3, 5, 10, 20, 30, 50 and 100 nearest neighbours and MEAN OR MODE IMPUTATION. Missing data inserted into attributes 0, 1 and 5 of Breast data set.

References

Aha, D. W., Kibler, D. & Albert, M. (1991). Instance-based Learning Algorithms. *Machine Learning 6*, 37–66. 4

Blake, C. & Merz, C. (1998). UCI Repository of machine learning databases. 13

Ciaccia, P., Patella, M. & Zezula, P. (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Internation Conference on Very Large Data Bases*, pp. 426–435. 4, 10

Clark, P. & Boswell, R. (1991). Rule Induction with CN2: Some Recent Improvements. In Y. Kodratoff (Ed.), *Fifth European Conference (EWSL 91)*, pp. 151–163. Springer-Verlag. 11

Clark, P. & Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning 3*(4), 261–283. 1, 12

Cost, S. & Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning 10*(1), 57–78. 8

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm (with Discussion). *Journal of Royal Statistical Society B39*, 1–38. 2

Dietterich, T. G. (1997). Statistical Tests for Comparing Supervised Classification Learning Algorithms. ftp://ftp.cs.orst.edu/pub/tgd/papers/stats.ps.gz. 21

Grzymala-Busse, J. W. & Hu, M. (2000). A Comparison of Several Approaches to Missing Attribute Values in Data Mining. In *RSCTC'2000*, pp. 340–347. 11

Jr., C. T., Traina, A., Seeger, B. & Faloutsos, C. (2000). Slim-trees: High Performance Metric Trees Minimizing Overlap Between Nodes. In *EDBT'2000*, pp. 51–65. 11

Kohavi, R. & Sahami, M. (1996). Error-Based and Entropy-Based Discretization of Continuous Features. In *Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, pp. 114–119. American Association for Artificial Intelligence. 9

Kohavi, R., Sommerfield, D. & Dougherty, J. (1997). Data Mining Using MLC++: A Machine Learning Library in C++. *International Journal on Artificial Intelligence Tools 6*(4), 537–566. 14

Lakshminarayan, K., Harp, S. A. & Samad, T. (1999). Imputation of Missing Data in Industrial Databases. *Applied Intelligence 11*, 259–275. 13

Lebowitz, M. (1985). Categorizing Numeric Information for Generalization. *Cognitive Science 9*(3), 285–308. 9

Lee, H. D., Monard, M. C. & Baranauskas, J. A. (1999). Empirical Comparison of Wrapper and Filter Approaches for Feature Subset Selection. Technical Report 94, ICMC-USP. 14, 16

Little, R. J. & Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. New York: John Wiley and Sons. 1, 2

Mangasarian, O. L. & Wolbert, W. H. (1990). Cancer Diagnosis Via Linear Programming. *SIAM News 23*(5), 1–18. 13

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. 4

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning 1*, 81–106. Reprinted in Shavlik and Dieterich (eds.) Readings in Machine Learning. 16

Quinlan, J. R. (1988). *C4.5 Programs for Machine Learning*. CA: Morgan Kaufmann. 1, 11

Shepard, D. (1968). A Two-Dimensional Interpolation Function for Irregularly Spaced Data. In *Proccedings of the 23rd National Conference of the ACM*, pp. 517–523. 7

Stanfill, C. & Waltz, D. (1986). Instance-based Learning Algorithms. *Communications of the ACM 12*, 1213–1228. 8

Uhlmann, J. K. (1991). Satisfying General Proximity/Similarity Queries with Metric Trees. *Information Processing Letters 40*, 175–179. 10

Wilson, D. R. & Martinez, T. R. (2000). Reduction Techniques for Exemplar-Based Learning Algorithms. *Machine Learning 38*(3), 257–286. 4, 9