Study of missing data imputation techniques for the purpose of classification

Auteur : Baudenne, Céline
Promoteur(s) : Haesbroeck, Gentiane
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil en science des données, à finalité spécialisée
Année académique : 2020-2021
URI/URL : http://hdl.handle.net/2268.2/11228

Master's thesis carried out to obtain the degree of
Master in Data Science and Engineering

Study of missing data imputation techniques for the purpose of classification

Céline Baudenne

Supervised by

Prof. *Gentiane Haesbroeck*

University of Liège - Belgium
Faculty of Applied Sciences
Academic year 2020-2021

Acknowledgements

First of all, I would like to express my grateful thanks to Professor Gentiane Haesbroeck for working with me on this subject. She has supported me throughout this work and her advice has always been very valuable.
I would also like to thank my family for their continuous support and encouragement during my studies and the development of this project.
Finally, I wish to thank the professors of the University of Liège for their teaching during these five years.

Abstract

Data quality is a central concern in data science and missing values are an important part of it. Missing data can have a detrimental effect in particular in the context of classification. This study evaluates the performances of missing data treatments by comparing the classification results obtained after applying classifiers on data sets imputed by means of various imputation methods. In addition, we decided to take a further step in addressing data quality issues by considering classification data containing atypical values in addition to missing values. The objective is thus to investigate whether a combination of imputation and classification methods could mitigate the potential adverse effect of contamination and missingness in order to achieve satisfactory classification results. For this purpose, we performed simulations within a well-defined theoretical framework in order to have full control over the data characteristics to better understand the different possible situations. Our observations enabled us to draw the main conclusion that, in such context, robust methods should be favoured both for imputation and classification methods.

Contents

1 Introduction **6**
- 1.1 The missing data problem . . . 6
  - 1.1.1 Definition . . . 6
  - 1.1.2 Challenges . . . 6
  - 1.1.3 Work content . . . 6
- 1.2 Missing data mechanism . . . 7

2 Missing data treatment methods **9**
- 2.1 Classification of treatment methods . . . 9
- 2.2 Review of traditional imputation methods . . . 10
  - 2.2.1 Single imputation . . . 10
  - 2.2.2 Multiple imputation . . . 12

3 Literature review on the comparison of missing data treatments **13**
- 3.1 Overall review . . . 13
- 3.2 Missing data and classification . . . 14
- 3.3 Missing data and outliers . . . 16
- 3.4 Conclusion . . . 17

4 Theoretical framework **19**
- 4.1 Introduction . . . 19
- 4.2 Definition of the theoretical model . . . 19
- 4.3 Contamination . . . 22
- 4.4 Missingness process . . . 24
- 4.5 Imputation techniques . . . 25
  - 4.5.1 Mean and median imputation . . . 25
  - 4.5.2 K-Nearest Neighbours imputation . . . 26
  - 4.5.3 SEQimpute and ROBimpute . . . 33
- 4.6 Classification techniques . . . 34
- 4.7 Evaluation . . . 35

5 Simulation set-ups **36**
- 5.1 Introduction . . . 36
- 5.2 Procedure . . . 36
  - 5.2.1 Data set . . . 36
  - 5.2.2 Contamination . . . 37
  - 5.2.3 Missingness introduction . . . 39

5.2.4 Imputation 40
5.2.5 Classification and evaluation 42
5.2.6 Summary 42

6 Simulation results **45**
6.1 Observations and analysis 45
6.1.1 Outlyingness 45
6.1.2 Classification results 50
6.2 Conclusion on the simulations 58
6.3 Contaminated cases imputation 58

7 Conclusion **60**

Appendix **64**
A.1 Outlyingness after missingness 64
A.2 Robustness breakdown 65
A.3 Classification results 66
A.4 Classification criterion plots 72
A.4.1 MCAR - Independent setting 72
A.4.2 MCAR - Correlated setting 76
A.4.3 MAR - Independent setting 80
A.4.4 MAR - Correlated setting 84

Chapter 1

Introduction

1.1 The missing data problem

1.1.1 Definition

The data preparation phase usually accounts for 50 to 60% of the Data Mining process [8]. Among other things, the identification and treatment of missing values is an important and challenging step. In fact, missing data is a frequent problem in many real-world data sets. For example, it is usual in medical studies not to obtain a response rate close to 100%, to such an extent that a very high rate should raise researchers' attention concerning the survey design and respondents selection [3].

Data can be missing for several reasons such as response refusal, problem in survey design, malfunction of a measuring device, etc. Missing values seriously impact the quality of the data. It can cause problems for the extraction of valuable information from the data set due to a loss of information and especially affect the efficiency of classifier algorithms.

1.1.2 Challenges

There exist several ways of dealing with missing data going from simple deletion of incomplete cases to much more complicated imputation methods. As the missing rate increases (10-15% and above), more sophisticated techniques are required to handle the problem [1].

Inappropriate treatment of missing values may induce bias and could lead to inaccurate conclusions drawn from the data. Therefore, it is important to ensure that the distribution of the data is preserved as well as the true relationship between the variables [25].

In the particular context of classification, it is necessary to distinguish two different settings, namely missing entries in the learning data and incomplete test cases to be predicted [26]. Also, a distinction should be made when missing values occur in the explanatory features or on the target variable [28].

1.1.3 Work content

The main focus of this work thus relates to the missing data in the particular context of classification. This implies that we will assess the quality of the treatment methods by evaluating the classification

performances of different classifiers trained on data sets in which the missing values have been treated beforehand. Furthermore, we will take a further step in the study of data quality issues by considering classification data containing atypical values, in addition to missing values. Missing data and contamination are two important issues that can adversely affect any analysis done on the data. Our objective is thus to investigate whether a combination of missing values treatment method and classification method can compensate for the effect of the contamination and missingness, in order to obtain satisfactory classification results.

We will carry out our analysis by means of simulations in which several imputation methods and several classification methods will be investigated. Our simulations will take place in a well-defined theoretical framework in order to have the complete control over the data sets and their characteristics. This will enable more accurate comparisons and analyses of our results.

The work is structured as follows: we describe the missingness mechanisms in the next section. In Chapter 2, methods for dealing with missing data are reviewed. The related literature is then analysed in Chapter 3. Chapter 4 defines in detail the theoretical framework within which the simulations will be carried out, together with an in-depth analysis of the different possible situations. The actual simulation procedure is described in Chapter 5, followed by the presentation and analysis of the results in Chapter 6. Finally, Chapter 7 concludes on the findings and ends with a discussion on future work.

1.2 Missing data mechanism

Before applying any treatment method, it is essential to investigate on the possible causes and the type of missing data contained in the data set. For example, it may happen that some survey participants are not concerned by certain questions. This would lead to missing values which should not be treated as such.

The type of missing data can be evaluated along two axes : the missing data pattern and the missing mechanism. The former refers to the configuration of the missing values, that is the disposition of the missing entries among the observed values. Prevalent patterns are the univariate pattern which has all missing values restrained to one variable and the general pattern in which there is, at first sight, no particular structure. Indeed, in the general pattern, all the missing entries appear to be distributed in an arbitrary way across the data set [10, 20]. The type of missing values can then be refined by reviewing the underlying missing data mechanism which aims at describing how the missingness propensity relates to the measured variables.

One can distinguish three different mechanisms : missing completely at random, missing at random and not missing at random [10, 6, 25, 20] :
In the following, let X_{full} represents the entire data set. It is composed of X_{obs} and X_{mis}, the observed part and the missing part of the data, respectively, i.e. $X_{full} = (X_{obs}, X_{mis})$. Let M denote a binary variable indicating whether a value of a particular variable is known or missing.

- **Missing completely at random** (MCAR) : Data is missing completely at random when the probability of missing data items on a variable X is unrelated to other measured variables as well as to the values of the variable X itself. Hence all cases have the same missingness probability and the observed data is a random sample of all the values of the variable. The

distribution of the missingness indicator M can be written as

$$P(M \mid X_{full}) = P(M)$$

Indeed, this probability depends neither on X_{obs} nor on X_{mis}. Although it is often not observed in practice, this setting allows any treatment method to be used without risk.

- **Missing at random** (MAR) : When the probability of data items to be missing depends on the values of other observed variables, the data is said to be missing at random. The missingness is yet not related to the variable with missing data. The observed items represent a random sample of the values when conditioned on the other observed variables. Hence the distribution of M in the MAR case can be written as

  $$P(M \mid X_{full}) = P(M \mid X_{obs})$$

  This actually implies the existence of a relationship between the variable for which data is missing and other features in the data set. The MAR mechanism is more realistic than MCAR.

- **Not missing at random** (NMAR) : The probability of missing data on X is dependent on the value of the variable X itself even after accounting for the other observed variables.

  $$P(M \mid X_{full}) = P(M \mid X_{mis}, X_{obs})$$

  In this particular case, the only possible treatment is by modelling the underlying missing mechanism.

The presence of the MCAR mechanism can be assessed by statistically verifying if there is a difference between the mean of other variables computed on the two groups defined by the binary missingness indicator. No significant difference suggests that the MCAR mechanism likely applies [10]. Indeed, under the MCAR mechanism it is assumed that the distributions of the observed and the missing values are the same [11]. MAR and NMAR mechanisms cannot be verified but are considered as non-ignorable mechanisms that need to be taken into account in the treatment phase [6]. The prior information about the missing mechanism can indeed help to impute the missing items [11].

Chapter 2

Missing data treatment methods

2.1 Classification of treatment methods

Techniques for handling missing data can be classified according to different aspects. First of all on the basis of the timing, one can distinguish the methods used during the data preprocessing phase from the embedded methods which are directly integrated into the data mining step. The former allows for more flexibility whereas the latter is more a cost-effective approach. Next, methods can differ on the basic approach they use : either statistical methods or machine learning methods. Finally, they can be categorized as deterministic methods or multiple methods. Deterministic methods do not take into account the uncertainty of imputed values that multiple methods include by considering multiple values [25].

Missing data treatment methods can further be classified into three main categories :

- **Deletion** : The simplest techniques consist in eliminating cases with missing values. There are two different alternatives : listwise deletion (also called complete-case analysis) and pairwise deletion (also known as available-case analysis). The listwise deletion approach removes all cases with at least one missing value in order to keep only the complete instances for the analysis. While it is the default option in many statistical softwares, it has some disadvantages. Indeed, this technique leads to a loss of information due to the reduction of sample size which can weaken the accuracy of the mining phase [25]. It is thus more appropriate for a relatively small amount of missing data.

  To mitigate the loss of information, the pairwise deletion approach discards cases with missing values only when they occur on the variables used in the computation of components of the statistical analysis [10]. Hence, different subsets of cases may be used for the computation of different elements.

  To be used, those two methods require the data to be missing completely at random. Otherwise, it could bias the data distribution and the outcome of the analysis as the preserved cases may not represent the full population [28].

  It may be also an option to delete an attribute when the level of missing values on this attribute is high. However, even with a high missing rate, relevant features should still be preserved [4].

- **Parameter estimation** : The principle is to derive the parameters of a model for the complete data using maximum likelihood estimation based on all available observed data. This

technique relies on variants of the Expectation-Maximization algorithm [23, 4].

The main limitation is the assumption that needs to be made about the distribution of the variables [1].

- **Imputation** : Each missing value is filled in with an estimate based on the available data. The idea is to use the relationships among the variables to support the estimation. The imputation method should preserve the relationships without strengthening it. A key advantage is that it reduces the loss of power by providing a complete data set for further analysis. Also, the treatment is independent from the mining algorithm allowing to choose the most appropriate method [23, 1]. There is a broad range of imputation techniques from single naive imputation to more robust methods, and multiple imputation.

2.2 Review of traditional imputation methods

In the following, we describe commonly used imputation techniques. This is a non-exhaustive list limited to the most frequently cited methods. Other purpose-oriented techniques will be developed later in the work.

2.2.1 Single imputation

Unconditional imputation

The simplest method only relies on the information provided by the variable of interest containing missing values.

- ***Mean or Mode imputation*** : It consists in replacing the missing data by the mean of the known values for the corresponding attribute. The mode is used in case of nominal variables. In case of skewed distribution with outliers, the median can be used in order to ensure robustness [1].

  Besides being easy to implement, unconditional substitution has several drawbacks. First, the distribution of the initial data set may be distorted since a set of values are now equal to the same constant. Variance and correlation are consequently also impacted. In addition, it does not take the potential relationships in the data set into consideration and hence could bias the following analysis [23, 1]. This method should only be used when, without any other information, the mean (or mode, median) yields the best estimation [6].

Similar cases-based imputation

The following methods exploit similarities among instances to derive the imputed values.

- ***Hot deck imputation*** : The basic idea is to fill in missing values with values from similar cases. Among other variants of hot deck imputation, the random hot deck method replaces the missing value with the observed value of a randomly selected instance from the subsample of similar cases. In the deterministic version of hot deck imputation, the missing value is substituted by the most similar case's value [6, 2]. Similarity between cases is often evaluated by looking at common observed values on other attributes [11]. Those attributes should be related to the missingness of the variable of interest [2].

- ***k-NN imputation*** : Similarly to the hot deck method, the k-nearest neighbour algorithm identifies the k most similar neighbours for each instance with missing values [5]. The imputed value then consists of the mean (or another estimation) of the neighbours' observed values for numeric variables and the most frequent value for nominal variables [22].

  The similarity measure and the choice of the value k also have an impact on the result [23]. In the context of classification, a small neighbourhood size would put too much emphasis on some dominant cases yielding a degradation of the classifier's efficiency. Whereas a large value of k leads to many different instances to be considered and thus alter the quality of the imputation causing the performance of the classifier to decrease [1]. It should be noted that when a single neighbour is considered, the k-NN method corresponds to the deterministic hot deck method.

The main limitation of this type of method is the computing time. In fact, in the search for similar cases, the algorithm reviews the whole data set which may be very large.

Conditional information-based imputation

The techniques reported hereafter use information from complete variables and benefit from their relationships to estimate the missing values.

- ***Regression imputation*** : Missing data are imputed by means of the fitted values of a regression model estimated on all available cases, the prediction being possible as soon as all values of the explanatory variables included in the model have been observed or previously imputed. This method relies on the assumption that the variables tend to be correlated with each other [23]. The base case is to use a linear regression for a linear relation or at least a relationship that can be linearized. Naturally, the type of regression is adapted to the situation and in particular to the type of variables being imputed. For instance, logistic or multinomial logistic regression will be used for qualitative variables.

  While the correlation structure is well taken into account, the relationships between the attributes are strengthened after the imputation [6]. This may not be appropriate if the subsequent mining procedure relies on regression techniques. Also, the variance of the data distribution after imputation may be underestimated. To mitigate this, stochastic regression imputation augments each predicted value with a residual term drawn from a normal distribution [10].

- ***Bayesian iteration imputation*** : Missing values prediction using Naive Bayes classifier occurs in two phases. In the first instance, one need to establish an order of the attributes containing missing values based on an indicator such as information gain, missing rate, etc. Next, the classifier is used to estimate the missing data. It proceeds iteratively starting by the first attribute and then following the order [25, 23].

  The Naive Bayes classifier simplistically assumes the variables to be independent when conditioning on the class. In practice, to carry out the imputation, each variable with missing values is treated as a class attribute and each missing entry is filled in with the class predicted by the classifier based on the conditional probabilities that have been derived during the training phase [11]. Naive Bayes method only passes through the data set once and is thus efficient.

2.2.2 Multiple imputation

As opposed to the single imputation techniques, multiple imputation generates several complete data sets using different missing values estimations [11]. Analyses are performed on each resulting data set which are then combined to produce a final result [28]. This enables to capture the uncertainty related to the imputation decision [11]. There are different ways to perform multiple imputation as long as the method used is non-deterministic. This process is usually computationally consuming. An example of well used multiple imputation method is the multiple imputation by chained equations (MICE). In the first place, all missing values are replaced by random observed values from the data. Then, each incomplete attribute is regressed on the other variables and the replacement values are drawn from the posterior predictive distribution. This process is repeated several times for each variable and until all incomplete features are imputed. The whole procedure is also repeated several times to generate multiple complete data sets which can then be analysed. This method can handle all feature types as each variable is imputed by its own imputation model [34].

Chapter 3

Literature review on the comparison of missing data treatments

This literature review aims to provide an insight on the different types of researches that have been carried out on the subject of missing data treatment and the different aspects that have been covered, while identifying potential shortages. The literary research was conducted first to have a general picture of the topic and then into more details according to the different aspects identified, always with a references back tracking and an evaluation of subsequent researches.

3.1 Overall review

Missing data treatment methods, especially imputation methods, have been subject to many researches in the literature. A comprehensive review of this literature from 2006 to 2017 has been done by Lin and Tsai [19]. They identified the main issues and limitations of the experimental protocol for missing value imputation research. The principal elements to be paid attention to are :

- the choice of data sets,
- the proportion of missing values,
- the missingness mechanisms,
- the imputation techniques,
- the evaluation metrics.

Among all the reviewed articles, they noticed that the data sets used are often rather small data sets from the UCI Machine Learning Repository which is a popular benchmark. The missing rates considered are usually limited and the focus is mostly on MCAR type missing values. The mean/mode imputation method has been identified as the most representative baseline for statistical methods. While the most prevalent and representative baseline machine learning type method is k-NN. Regarding the evaluation metrics, most of the studies decide to artificially introduce missing values in order to directly assess the discrepancy between the original values and the imputed values. Another way of assessing the performance of the imputation methods used in the researches is to use the data in a classification context. In that respect, the imputation efficiency is evaluated using the classification accuracy after training a classifier on the imputed data set. This will be

investigated in more detail in the following. Among the reviewed articles, the authors observed a lack of investigation on the different types of feature that can be found in real data sets and the impact they can have on the imputation performance. They also raised the question of including feature or instance selection within the process of imputation.

3.2 Missing data and classification

Our focus will be on studies about imputation methods in the particular context of classification. Incomplete data sets are problematic in that context as the majority of classification methods are unable to deal efficiently with missing values. Indeed, they usually make use of the most straightforward method which consists in discarding incomplete cases. This has been shown not to be optimal. For that reason, many authors have investigated the influence of imputation methods on classification performances. In this sense, they compare imputation techniques on basis of the accuracy of one or more classification algorithms trained after performing imputation on the incomplete data. Part of the researches reviewed by Lin and Tsai [19] resort to this practice to assess the imputation methods. Particularly, they identified a lack of consideration of missingness in the testing data set when such a subset is used.

While focusing on discrete data, Farhangfar et al. (2008) [11] analysed six missing data imputation methods including single imputation methods (mean, hot deck, Naive-Bayes plus the last two within a boosting framework) and one multiple imputation method (based on polytomous regression) together with five machine learning classifiers (RIPPER, C4.5, k-Nearest Neighbour, SVM and Naive-Bayes). Experiments have been done on fifteen data sets with artificially introduced missing values into all attributes of the training set, following the MCAR mechanism with missing rate ranging from 5 to 50%. They used as baseline the accuracy obtained with the incomplete data without any imputation treatment and with the original complete data. Although the impact of imputation differs depending on the classifiers, their analysis shows that, on average, imputation improves the classification accuracy hence being beneficial for classification tasks. Yet, they concluded that there is no universally best imputation method and they noticed that some classifiers (C4.5 and Naive-Bayes) appear to be rather resistant to missing values. In the same way, other authors made similar analyses [1, 4, 35, 24, 13].

Luengo et al. (2012) [22] presented a more comprehensive study as they investigated fourteen different imputation techniques and about twenty classification methods on twenty-one UCI data sets containing natural missing values. The missing values have not been induced in the different data sets in order to remain closer to real-world data and they assumed them to be of MAR type. They considered three categories of classifiers : rule induction learning, approximate models ("black box" kind of model) and lazy learning (using similarity measures to perform classification). Again, the imputation methods have been compared on the basis of the classifiers' accuracy. They have been able to identify recommended imputation methods depending on the category of classifier. However, they agree with Farhangfar et al. [11] on the fact that there is no best imputation method for all classifiers. The benefit of imputation against non-imputation strategies is also reflected in their results. They came to the final conclusion that the choice of imputation method should depend on the type of classification method. They also have investigated the effect of imputation on the instances and the features, evaluating the possible induced noise and the impact on features relationship with the target class.

The literature of missing values imputation also includes studies on specific domain problems. One that is widely considered is the medical field [19]. Indeed, missing values are commonly present in medical data sets. The treatment of incomplete cases in such data sets is all the more important as the issue can be detrimental in certain circumstances. Focusing on a real breast cancer data set containing missing values, the authors [17] compared statistical imputation methods (mean, hot-deck and multiple imputation based on regression and the EM algorithm) against machine learning imputation methods (multi-layer perceptron, self-organisation maps and k-NN). The evaluation has been made on the capacity of the imputation methods to improve the accuracy of predictions obtained using artificial neural network. Their findings show that all imputation methods, apart from the hot-deck imputation method, gave rise to an improvement of the prediction in terms of area under the ROC curve (AUC) values. Particularly, the machine learning methods show a significant difference from the reference model. They also have explored the impact of the size of the available data set on the performance of the different methods which suggests that imputation techniques are important in case of small data sets. In this study, only a single prediction method and a single data set have been used, hence the results may not be completely generalisable.

Garciarena and Santana (2017) [12] took an extra step by including the characteristics of missing data in the analysis. They assessed the relation between the type of missing data and the imputation method and their impact on the classifier performance. They proposed, for each missing mechanism, a strategy to introduce corresponding missing values into the data set. Those methods have been applied on ten UCI data sets with no missing values and they decided to set the missing rate to a fixed value. They compared eight imputation methods and eleven different classifiers. Their results suggest that the bias produced when using the data for classification would be related to the random character of the missing mechanism. For instance, in case of MAR missing values, as the values are absent due to an underlying reason, one can expect the incomplete instances to be somewhat similar to each other hence labeled with the same class. This could possibly impact the classification. Whereas when the missing values are more evenly distributed, the loss of information is likely to be less significant. They also investigated the effect of the percentage of missing values and noticed that when the missing rate increases, the performance of the classifier declines, yet some more complex imputation methods seem to still achieve acceptable results. Overall, they concluded that, as far as possible, the missing mechanism should be investigated before imputation.

In a more recent work, Choudhury and Kosorok (2020) [7] proposed a new version of the k-NN imputation method. They decided to work in a slightly different way. On the one hand, they have chosen to include information about the class label via the use of mutual information whereas most of prior studies do no account for this information in the imputation. On the other hand, they tested their novel imputation technique on simulated data in addition to real UCI data sets. They also assumed values to be missing at random both in the training and testing data sets. Their k-NN variant relies on, in their opinion, a more appropriate distance measure (Grey distance), which is then weighted by the mutual information of each attribute relative to the class label. The reason for this is to account for features relevance in the selection of the most relevant neighbours for the imputation. The performance has been compared against six classical multiple imputation methods by means of both imputation accuracy, with the root mean square error, and classification accuracy using Naive Bayes classifier. Their results showed that the proposed method outperforms the other methods mainly for higher missing rates and that it better improves the classifier accuracy. The main limitations of their research are first that they have tested the performance only using a single classification algorithm and second they did not experiment missing rates above 20%. Many other authors also proposed their own imputation technique which they compared to more classical ones.

Tran et al. (2018) [30] have examined how clustering combined with feature selection can help improving classification with incomplete data while not affecting the accuracy achieved using imputation methods. They have considered three ways of combining imputation with clustering and feature selection : (1) Integrating clustering into imputation so as to reduce the original training data to a smaller more representative subset to be used for the imputation during the prediction phase. This enables to lower the computing time required to estimate missing data. (2) Integrating feature selection into imputation to remove irrelevant features in the imputed data for purpose of building better classifiers as well as reducing the amount of incomplete new cases. (3) Integrating both clustering and feature selection into imputation with the idea of removing redundant features to improve the clustering performance. They used both a single and a multiple imputation methods: k-NN and MICE and compared the effect on three classification algorithms namely C4.5, k-NN and Naive-Bayes. Tests have been done on ten incomplete UCI data sets of various domain problems and with various missing rates. From their results, it can be seen that the three processes speed up the computing time and that the integration of feature selection also helps to improve the classification accuracy. In particular, Naive-Bayes classifier benefits the most of the integration.

3.3 Missing data and outliers

This literature review aims to report on the different perspectives that have been investigated. One topic which has not been widely considered is the behaviour of missing values imputation techniques and classifiers performance when the data set contains atypical data. Noisy observations or outliers which have a value far away from the center of the data may somewhat distort the values estimated for imputation. In the end, this may deteriorate the quality of the classification. Some authors proposed to include clustering or instance selection procedures to filter the most representative observations.

Tsai and Chang (2016) [31] and Huang, Lin and Tsai (2018) [14] carried out the same kind of analysis whose purpose was to investigate whether performing an additional instance selection step to remove noisy observations could, at the end, improve the classifier performance. The former study investigated several ways of combining instance selection with one imputation method whereas the latter compared the combination of three instance selection algorithms with three imputation methods focusing on medical data sets. Both studies made the comparison based on the final classification results while evaluating separately the results according to the feature types in the data sets. They came to the similar conclusion that performing instance selection before applying the imputation methods can have a positive impact on the classification performances mainly in the case of numerical and mixed-type data sets. As for categorical data sets, the impact is not always definitely positive. Also, instance selection in case of small data sets should be used with great care. Overall, the quality of the data should be attentively evaluated beforehand.

Other authors focused on the direct use of robust imputation methods. Toka and Cetin (2016) [29] compared classic and robust imputation techniques including a robust version of the EM algorithm (ER) and of the sequential imputation method (ROBimpute). They worked with simulated data under different levels of contamination (10% - 20%) and random missingness (5% - 10%) on which they evaluated the mean vector and covariance matrix after performing the different imputations and compared them on the basis of the mean squared error. They also applied the techniques on a well-known contaminated data set with introduced missing values. They compared the imputation results as well as the mean imputation error. At first sight, they logically observed that the mean method is not able to correctly approximate the value of a missing value from the contaminated

part of the data whereas the robust methods provide good results. Overall, they concluded that robust methods can deal relatively well with outlying observations as shown by the smaller MSE they induced. Several researchers also presented their robust version of imputation methods.

3.4 Conclusion

Table 3.1 summarises the main elements of the related researches presented above. The main conclusion that can be drawn from this literature overview is that most imputation methods generally help in coping with the loss of information induced by missing values, yet no universal method could be identified. Focusing on the classification context, it has been observed that the performance of classifiers declines with increasing missing rates, but that complex imputation methods enable to achieve acceptable results. Moreover, we rarely observe missingness in the testing set, although test cases are even likely to contain missing values. Another observation from this literature review is that many researchers seem to use real data in which they usually artificially introduce missing values. Few studies rely on fully simulated data. Furthermore, it appears that barely no comprehensive studies have been done investigating the impact of different imputation methods in the context of classification for data sets containing potential atypical data. It is therefore in this perspective that the study will continue.

Table 3.1: Summary of related researches

Reference	Imputation methods	Classifiers	Datasets	% of missing values	Missing mechanism
Farhangfar et al. [11]	Mean, Hot deck, Naive-Bayes, Imputation framework with Hot deck, Imputation framework with Naive-Bayes, Polynomial multiple regression	Ripper, C4.5, k-NN, SVM, Naive-Bayes	15 UCI data sets	5 - 50%	MCAR
Luengo et al. [22]	(Class) Mean and mode, (Weighted) k-NN, (Fuzzy) K-means clustering, SVM, Event covering, Regularized expectation-maximization, Singular value decomposition, Bayesian principal component analysis, Local least squares	- Rule induction learning (C4.5, Ripper,Part,...) - Approximate models (MLP, SVM, Logistic, Naive Bayes,...) - Lazy learning (k-NN,...)	21 UCI data sets	0.05 - 22%	MAR (Original missing values)
Jerez et al. [17]	Mean, Hot deck, Multiple regression, MLP, Self-organisation maps, k-NN	ANN	Breast cancer data set	5.61%	MAR (Original missing values)
Garciarena and Santana [12]	Mean, Median, Mode, Last Value, Interpolation, Hot deck, Iterative imputation, Multiple Imputation	Logistic regression, Linear and quadratic discriminant analysis, Neural Network, SVM, Radial basis function, Gaussian Naive-Bayes, Gradient boosting, Random forests, DT, k-NN	10 UCI data sets	7 - 42%	MAR, MCAR, MNAR
Tran et al. [30]	K-NN, MICE	C4.5, k-NN, Naive-Bayes	10 UCI data sets	5-100% (% of incomplete instances)	MAR (Original missing values)
Choudhury and Kosorok [7]	Class-weighted grey k-NN, MICE, MissForest, Iterative k-NN, Mutual information based k-NN, Grey k-NN, Feature weighted k-NN.	Naive-Bayes	2 simulated + 3 UCI data sets	5% - 20%	MCAR, MAR

Chapter 4

Theoretical framework

4.1 Introduction

The previous chapter enabled us to identify an interesting direction of research. In fact, we will analyse **the combined effect of missing values and outliers on classification performances**. The study will be done by comparing the classification results obtained after applying classifiers on data sets imputed by means of various imputation methods. In this respect, our primary objective is not to evaluate the ability of the imputation methods to reconstruct the original data sets but rather to investigate whether a combination of imputation and classification methods could mitigate the potential adverse effect of contamination and missingness in order to achieve satisfactory classification results. Indeed, one can expect the observed outlying data to have an influence on the estimation of the missing values which may impact the classification, at least for non-robust methods.

The experimentation will be executed in a well-defined theoretical framework. To this end, the chapter is organized as follows: the first section defines the theoretical model. The configurations of the contamination are then described, followed by the missingness process. Finally, the various investigated imputation methods and classification methods, both classic and robust, are presented together with their evaluation.

4.2 Definition of the theoretical model

In this section, we introduce the theoretical model that will be used for the experimentation. Suppose we have two p-dimensional populations distributed according to two normal distributions. This constitutes a two-class data set built from a mixture of two multivariate normal distributions, one for each class. The covariates X come from one of the populations with a probability π_0 and π_1 respectively (with $\pi_0 + \pi_1 = 1$). More precisely:

$$X \sim \begin{cases} \mathcal{N}_p(\mu_0, \Sigma_0) \text{ with probability } \pi_0, \\ \mathcal{N}_p(\mu_1, \Sigma_1) \text{ with probability } \pi_1. \end{cases}$$

where the $p \times p$ squared matrices Σ_0 and Σ_1 are assumed to be positive definite and symmetric. The variable Y denotes the class membership:

$$Y = \begin{cases} 1 \text{ with probability } \pi_1, \\ 0 \text{ with probability } \pi_0 = 1 - \pi_1. \end{cases}$$

Besides being under normality, we will also assume homoscedasticity i.e. the covariance matrices are made equal for the two classes $\Sigma_0 = \Sigma_1 = \Sigma$, Σ being assumed to be positive definite. Under those assumptions, it has been proven that logistic discrimination and Fisher discrimination coincide at the population level [9]. An observation with observed covariates x is classified in population P_1 if $\alpha + \beta^\top x > 0$ (where α is the intercept and β is the slope parameter in the logistic model), and in population P_0 otherwise. In our setting, we have

$$\alpha = \ln \frac{\pi_1}{\pi_0} - (\mu_1 - \mu_0)^\top \Sigma^{-1} \left(\frac{\mu_1 + \mu_0}{2} \right) \qquad \text{and} \qquad \beta = \Sigma^{-1}(\mu_1 - \mu_0) \tag{4.1}$$

To simplify the context, we assume $\pi_0 = \pi_1$ (balanced case) and $\mu_1 = -\mu_0$. The parameter α is thus equal to zero so the discrimination rule simplifies to $\beta^\top x > 0$. Under these assumptions, the computation of the error rate is straightforward, as will be explained below. We know an error will be attributed to the population P_0 if $\beta^\top x > 0$, and to the population P_1 if $\beta^\top x < 0$. Therefore, all necessary information is available in order to compute the probability of misclassification.

$$\text{Misclassification rate} = \pi_0 \mathbb{P}\left(\beta^\top X > 0 \mid P_0 \right) + \pi_1 \mathbb{P}\left(\beta^\top X < 0 \mid P_1 \right).$$

If $X \sim \mathcal{N}_p(\mu_0, \Sigma)$ then

$$\beta^\top X = (\mu_1 - \mu_0)^\top \Sigma^{-1} X \sim \mathcal{N}_1 \left((\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_0, D^2_\Sigma(\mu_0, \mu_1) \right)$$

and if $X \sim \mathcal{N}_p(\mu_1, \Sigma)$ then

$$\beta^\top X = (\mu_1 - \mu_0)^\top \Sigma^{-1} X \sim \mathcal{N}_1 \left((\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_1, D^2_\Sigma(\mu_0, \mu_1) \right)$$

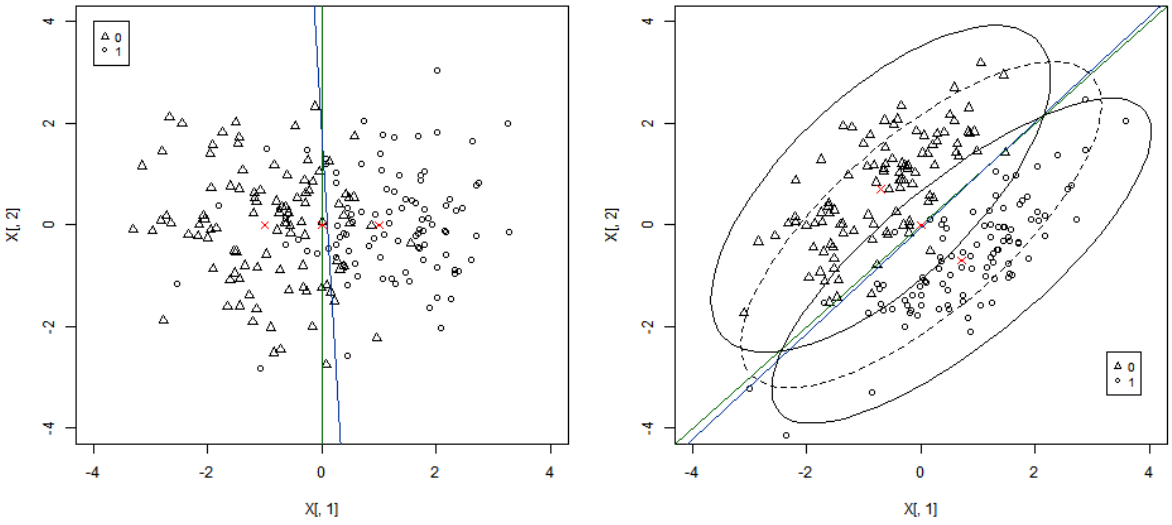
where $D^2_\Sigma(\mu_0, \mu_1) = (\mu_1 - \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0)$ is the squared Mahalanobis distance between the two group means while taking the common covariance structure into account.

Thus, we have

$$\begin{aligned}
\text{Misclassification rate} &= \pi_0 \mathbb{P} \left(\frac{\beta^\top X - (\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} > \frac{-(\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \\
&\quad + \pi_1 \mathbb{P} \left(\frac{\beta^\top X - (\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_1}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} < \frac{-(\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_1}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \\
&= \pi_0 \left(1 - \Phi \left(- \frac{(\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \right) + \pi_1 \Phi \left(- \frac{(\mu_1 - \mu_0)^\top \Sigma^{-1} \mu_1}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \\
&= \pi_0 \left(1 - \Phi \left(\frac{2\mu_0^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \right) + \pi_1 \Phi \left(- \frac{2\mu_0^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) \\
&= \Phi \left(- \frac{2\mu_0^\top \Sigma^{-1} \mu_0}{\sqrt{D^2_\Sigma(\mu_0, \mu_1)}} \right) = \Phi \left(- \sqrt{\mu_0^\top \Sigma^{-1} \mu_0} \right)
\end{aligned} \tag{4.2}$$

The final expression of the misclassification rate tells us that the position of μ_0 will have an impact on the classification and, in particular, it will either facilitate or complicate the separation of the two populations. For instance, if $\|\mu_0\|$ increases, the misclassification error will decrease as the two data groups will be spaced further apart. If μ_0 is taken on the principal direction induced by the covariance matrix Σ then the error will be much greater than if it was taken on an orthogonal direction. In addition, this determines the discriminating power of the variables. For example, if the probability of error is close to 50%, it means that the two populations are highly overlapping. In our configurations, this probability will not exceed 20%. We have decided to simplify the context by taking $\mu_0 = -\mu_1$, however, in the general case, the misclassification rate is equal to $\Phi\left(-\frac{1}{2}\sqrt{(\mu_1-\mu_0)^\top\Sigma^{-1}(\mu_1-\mu_0)}\right)$ which is of the same kind.

Figure 4.1a illustrates, with simulated data (sample size = 200), the $p = 2$ dimensional case. The means and covariance matrix are respectively $\mu_1 = (1,0)^\top$, $\mu_0 = -\mu_1$ and $\Sigma = \mathbf{I}_2$. Due to the absence of a privileged direction, taking here $\mu_0 = -\mu_1$ is representative of all cases. The two different data groups can clearly be observed on the figure with the true group means and the true overall mean represented by red crosses. The green line depicts the true discriminating line while the blue one corresponds to the estimated discriminating line. The estimation has been done by computing α and β (4.1) using the empirical mean vectors computed in each group, $\bar{x}_0$ and $\bar{x}_1$, and the pooled covariance matrix $S = \frac{(n_0-1)S_0+(n_1-1)S_1}{n_0+n_1-2}$ where S_0 and S_1 are the covariance matrices computed on the two separate groups. As expected, in such a simulated case where the distributions correspond to the theoretical ones, both discriminating lines are similar. The probability of misclassification is 16% while the percentage of error computed on the training data is 14%. It should be noted that in the simulation process, the error rate will be estimated on a test data set generated according to the same underlying distributions.



(a) Independent setting.

(b) Correlated setting.

Figure 4.1: Illustration of the logistic/Fisher discrimination under a balanced mixture of 2-dim normal distributions. *Red crosses represent the true group and overall means. The green line is the true discriminating line, the blue line is the estimated discriminating line.*

Together with this independent setting, an equi-correlated structure will be examined still under the homoscedasticity assumption, as represented on figure 4.1b. The covariance matrix Σ is now equi-correlated with $\sigma_{ii}^2 = 1$ and $\sigma_{ij}^2 = \rho$ $(i \neq j)$ with $\rho = 0.75$. The means are taken on the orthogonal direction with respect to the first eigenvector of Σ, with the additional constraint that $\mu_0 = -\mu_1$. Here, $\mu_1 = \frac{\sqrt{2}}{2}(1, -1)^\top$. This set up does not correspond to the hardest nor the easiest discrimination setting. The 99% tolerance ellipses are plotted on top of the data points. The two solid ellipses are estimated using S_0 and S_1 respectively while the dashed ellipse corresponds to the pooled covariance matrix S. They illustrate the populations quasi-separation and thus the discriminating nature of the variables. As explained earlier, this can also be measured by the probability of error. For instance, here the probability of misclassification is 2% while the percentage of error computed on the training data is 3%.

Along with the $p = 2$ dimensional case which enables a representative visualization of what is happening, a $p = 5$ dimensional case will also be investigated using the same type of covariance structures.

4.3 Contamination

There exist several ways of formalizing the presence of contamination in data. The most well-known contaminated model is the ϵ-contamination neighbourhood implying that while the true model is F, the actual data generating function belongs to the set

$$\mathcal{F}_{\epsilon'} = \left\{ F_{\epsilon'} = (1 - \epsilon')F + \epsilon' G; \, G \text{ any distribution}, \, 0 < \epsilon' < 1 \right\}.$$

We expect ϵ' to be reasonably close to zero. Assuming such a model in our 2-group setting would imply the data generating distribution to be

$$(1 - \epsilon')(\pi_0 F_0 + \pi_1 F_1) + \epsilon' G.$$

This means that a fraction $1 - \epsilon'$ of the data points comes from the *good* mixture model and the remaining fraction ϵ' consists of outliers [29, 9, 15].

However, in this classification setting, we need to attribute a class membership to all observations of the training data. This could be done by means of a binary indicator with success probability π_1. Another possibility, allowing to preserve the balance between the two groups, consists of modifying the contamination model as follows:

$$\pi_0 F_0 + \pi_1 F_{1,\epsilon'} \quad \text{with } F_{1,\epsilon'} = (1 - \epsilon')F_1 + \epsilon' G \tag{4.3}$$

This implies that the data coming from F_0 are clean by default as contamination only occurs in the population P_1. In particular, we will have a fraction $\pi_0 + (1 - \epsilon')\pi_1$ of *good* points and $\epsilon = \epsilon'\pi_1$ of *bad* points. Such setting is comparable to a situation in which observations in one population are more prone to contamination. This is the strategy we will adopt here as it is easier to interpret. It should be noted that contamination may happen both in the training and test sample. The $\epsilon' = 0$ case will be used as a baseline.

The distribution G in (4.3) can be any distribution. For example, it might be a uniform distribution as depicted on figure 4.2. In that situation, the outliers are not necessarily extreme and as illustrated here, the impact on the estimation of the discriminating line is expected to be limited. A more

challenging situation is when the outliers (distributed according to G) form a cluster (a concentrated normal distribution say, or even a Dirac distribution). This is the approach followed here : $G = \mathcal{N}_p(\mu_c, \Sigma_c)$ with μ_c and Σ_c to be defined.

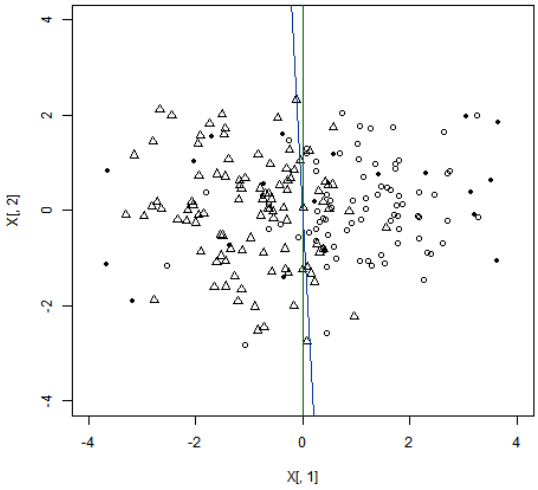
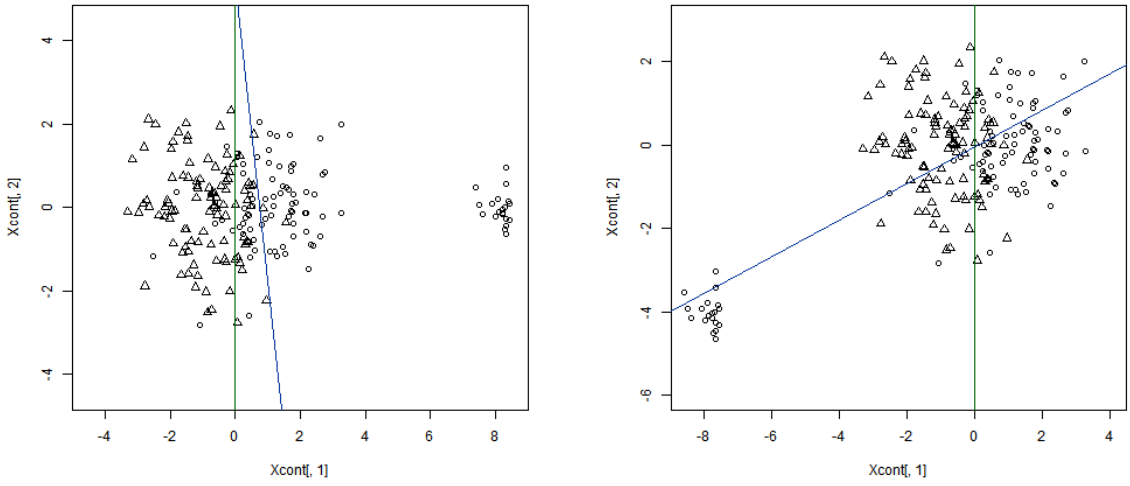


Figure 4.2: Uniform noise contamination.
Contamination in P_1 is highlighted by the bullets.

Depending on μ_c, the contamination cluster can be placed in several locations which may have a different impact on the classification results. In particular, we will investigate the position of this contaminating group with respect to the discriminating line.



(a) Good leverage points.

(b) Bad leverage points.

Figure 4.3: Contamination location impact on the discriminating line - 10% contamination on P_1.

As can be seen on figure 4.3a, the contamination may remain on the same side of the true discriminating line. This has a limited impact on the estimated discriminating line and on the misclassification

rate, as long as the outliers are not too extreme. They are usually called *good* leverage observations. By contrast, *bad* leverage points, which are located on the other side of the discriminating line, affect to a greater extent the discriminating line estimated in the classic way. Indeed, it can be noticed on figure 4.3b that the estimated discriminating line deviates much more from the true line when there is a cluster of such points in the data. In fact, it has been attracted by the outliers. The initial misclassification rate was 16% and it remains close to that value in the left figure (17%) while it increases to 30% in the right figure.

As a summary, contamination in our simulation study will be introduced only in population P_1 according to a distribution G which will take the form of a normal distribution shifted and more concentrated in comparison with the initial population distribution.

4.4 Missingness process

Missing values can be artificially introduced in the data according to the different mechanisms presented in section 1.2. Here, the missingness will be introduced following the two subsequent mechanisms:

- Missing completely at random (MCAR)
- Missing at random (MAR)

As far as the third missingness mechanism (NMAR) is concerned, the introduction of missing values under this mechanism basically follows the same idea as the generation under MAR, with the exception that the variable in which missing values are introduced would come into play in the computation of the missingness probability. This explains our decision to concentrate on two mechanisms.

Missing values are introduced only in the explanatory variables. This is because, in the context of classification, missing target labels y would correspond to new instances to be classified. Following the idea of Choudhury and Kosorok [7], missingness is introduced only in some variables while the remaining ones stay non-missing (not containing missing values). Indeed, it is not uncommon in real life to have data sets containing demographic type variables which are non-missing. It should be noted that although the missingness is restricted to certain covariates, the MCAR mechanism remains true as the process is random on those variables cells.

Another point is whether to make some contaminated data be missing. Keeping the contaminated data as such would enable to evaluate the impact they have on the imputation and classification results. Whereas letting some contaminated data be missing may reveal some improvement on the results as this could limit the impact those atypical values can have. Indeed, it may happen, depending on the imputation method, that some missing but initially contaminated cells are imputed by clean observations. However, preventing the contaminated observations to be missing would imply that the MCAR mechanism does not hold anymore. Given the random nature of the mechanism and the proportions of missing values and contamination that will be investigated, some contamination should remain in the data anyway, even after imputation. What is important then is being able to quantify the contamination at the different stages of the process. Here, this is done using the robust Mahalanobis distance computed based on a robust estimation of the data set mean and covariance matrix using the S-estimator (calculated in R using the `fastSloc` function [27]). The outliers detection threshold is set to the 99% quantile of the χ^2 distribution with $p = 2$ degrees of freedom. This quantile has been chosen in order to identify the most extreme outliers.

At this point, it is important to differentiate between what the percentage of missing values and the percentage of contamination respectively apply to. The contamination proportion $\epsilon = \epsilon'\pi_1$ applies in terms of rows of the data set. This means that although the contamination may occur only on one dimension, the whole instance is considered as contaminated. Thus we have $\epsilon * 100\%$ of the data set size n that will be contaminated. Whereas the proportion of missing values π applies in terms of cells of the data matrix. It can either be applied on the total number of cells $n * p$ (where n is the number of instances and p the number of dimensions of the data set) or on the number of cells of the missing dimensions only (e.g. $n * 2$ if only two dimensions can be missing). Here, the second convention will be used as we restrict the number of dimensions which can be missing.

4.5 Imputation techniques

Missing values need to be treated before classification. To this end, several methods will be used and compared. The most basic technique, case deletion, will be investigated together with other methods that will produce complete data sets for classification, namely:

- Mean imputation
- Median imputation
- K-Nearest Neighbours imputation (using mean & median)
- Sequential imputation (SEQimpute)
- Robust sequential imputation (ROBimpute)

The idea is to compare different types of missing values imputation methods and evaluate their performances in the presence of atypical data. Case deletion is the most basic and straightforward technique as all instances (i.e. rows) with at least one missing value are eliminated from the data. As mentioned in Chapter 3, the mean imputation method has been identified by Lin and Tsai [19] as one representative baseline for statistical imputation methods. Its robust version, the median imputation method will also be evaluated. Lin and Tsai [19] as well as other authors refer to the k-nearest neighbours method as one of the most popular and representative machine learning based imputation methods. Finally, the choice of the two last methods, SEQimpute and ROBimpute, has been inspired by Toka and Cetin [29].

Our final goal is to evaluate the impact that a combination of imputation and classification techniques can have on the classification results in a context of contamination. In that respect, we will investigate to what extent the imputation can benefit the estimation of the classification rule in the different configurations considered.

4.5.1 Mean and median imputation

The advantage of using the median in place of the mean for the imputation takes on more meaning in a contaminated setting. Indeed, in an uncontaminated setting, the difference between the mean and the median imputation is often not significant.

In the classification context, the goal, in the end, is being able to *separate* one population from the other in the best possible way. All observations of the training data having a class membership label, we could use this extra information, as Choudhury and Kosorok did in their study [7], to compute a conditional mean or median which should bring the missing values back to their respective data

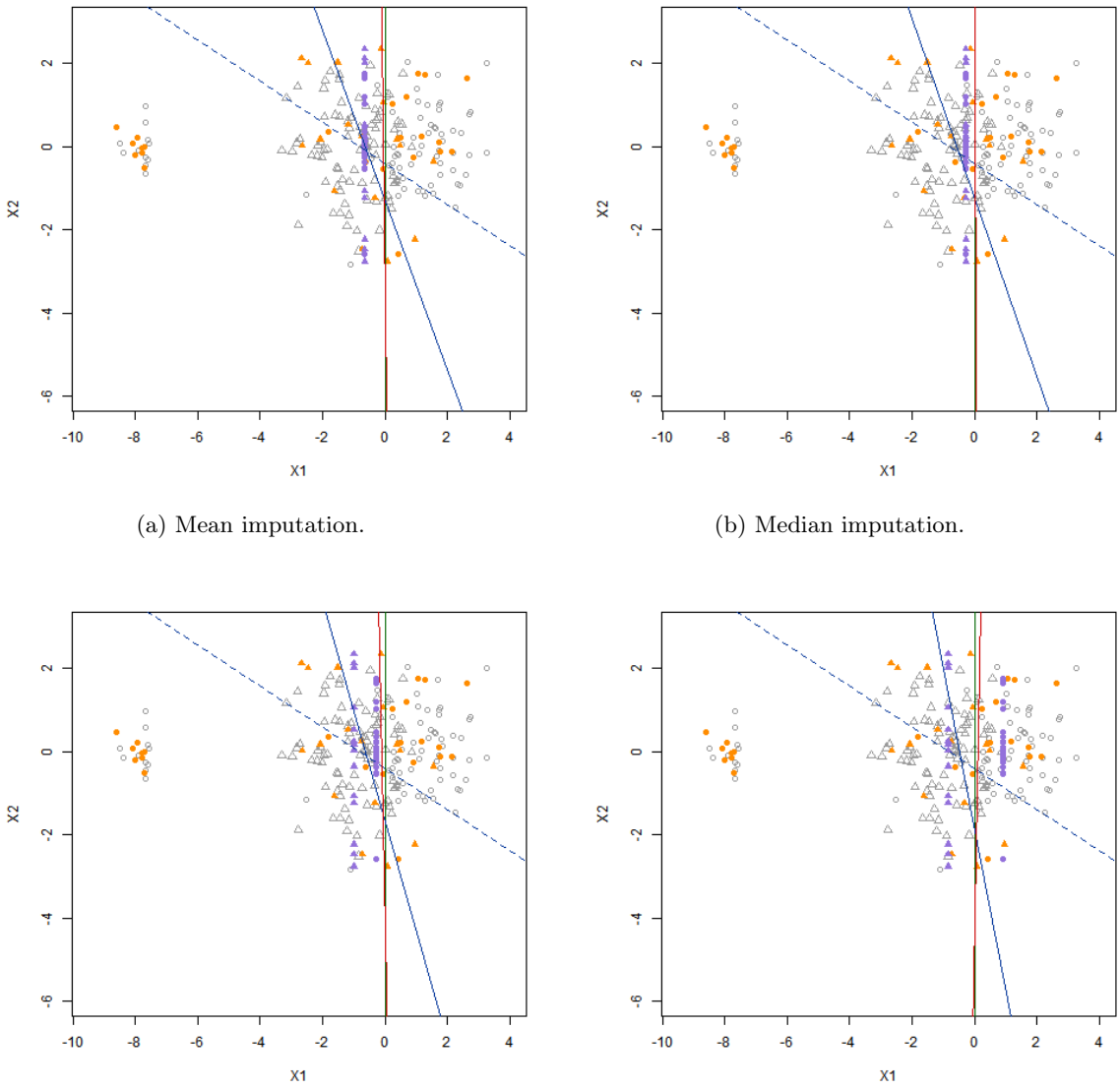
group. Obviously, the conditional mean or median can only be calculated when the class membership is available which is not the case for new instances that need to be classified. Nevertheless, the objective here is to use all information at our disposal that could improve the imputation in order to eventually enhance the estimated classification rule which is used afterwards to classify the new instances.

As previously seen, the extreme outliers disrupt the discriminating line. If some of those outliers are imputed so that they become *clean*, one can expect improvement on the discriminating line estimation. This can be observed on figure 4.4 which compares the mean and median imputations, both unconditional and conditional with respect to the membership. The represented setting is a 10% *bad* contamination on X_1 and 20% of MCAR missing values on X_1. We therefore expect to have 8% of contamination remaining after the missing values are introduced. The blue lines represent the discriminating lines, the dashed line being estimated on the contaminated set before missingness introduction while the solid one is estimated on the imputed data set. The orange symbols depict the missing values and the purple symbols illustrate their imputation. It can clearly be seen that the dashed line badly separates the two populations. In all of the four cases however, the imputation comparatively improves the estimations (solid lines). This can be explained by the fact that the missing but initially contaminated instances are imputed by clean values which reduces the number of observations disturbing the estimation. Indeed, we notice that the orange points in the outlying cluster are shifted to a single value in the main data group after imputation. The resulting estimated discriminating line closest to the true discriminating line is obtained using the conditional median imputation (panel (d)). Unlike the mean, the conditional median is not attracted by the outlying values. To measure the ability of the estimated discriminating lines to separate the two populations, we have evaluated the produced discriminating rules on the initial uncontaminated data set. The initial misclassification rate (according to the dashed line) is 36% and it drops to 21% when relying on the estimation obtained in (d).

However, in certain configurations or when the proportion of contamination remains too high, the imputation may not be sufficient to mitigate the outliers impact. In such cases, the last option is to resort to a robust classification technique. This is illustrated by the red lines on figures 4.4 and 4.5. This robust discriminating line is obtained by plugging robust estimations of the means and covariance matrix in the definition of α and β (4.1) following the idea of Hubert and Van Driessen [16]. Here, the robust estimations have been computed using the S-estimator (calculated in R using the `fastSloc` function [27]). We notice that the robust discriminating lines are really close to the true line whatever the applied imputation and thus better separate the two populations. In particular, on figure 4.5, the 10% contamination is now applied on both dimensions. In this setting, the contamination after missingness introduction still drops to 8% but even after imputation the contaminated cases remain outlying and the proportion of contamination after imputation is again 10%. This is explained by the fact that the contamination can only be cleaned if the values become missing. Indeed, on figure 4.5, the missing but initially contaminated instances are cleaned on the missing dimension X_1 but remain contaminated on dimension X_2 for which the values have not been modified. This is why those instances are still considered as outliers.

4.5.2 K-Nearest Neighbours imputation

In the case of continuous data, the most straightforward way of robustifying the k-nearest neighbours imputation technique is by computing the estimation of the missing values by means of the median instead of the mean. Given this estimation is calculated on a local basis, one could expect the



(a) Mean imputation.

(b) Median imputation.

(c) Conditional mean imputation.

(d) Conditional median imputation.

Figure 4.4: Illustration of the missing values estimation with the mean and median imputation method in a 10% contaminated setting . *The orange symbols represent the missing values (20% missing on X_1) while the purple symbols represent their imputation.*

outlying values to be used for the computation in very few cases. Indeed, the computation only relies on the closest neighbours classically selected based on the Euclidean distance. This may be true for the imputation of clean instances but not for imputation of contaminated instances. In fact, the imputation results will widely depend on the dimension on which the contamination occurs with respect to the missing dimension. In addition, the number of neighbours k and the estimator used for the computation of the final imputed values will also have an impact. It should be noted that other distance measures can be used, as done in the study of Choudhury and Kosorok [7] where they additionally account for features relevance in the neighbours selection. However, here, we decided to evaluate the performance of the classic version.

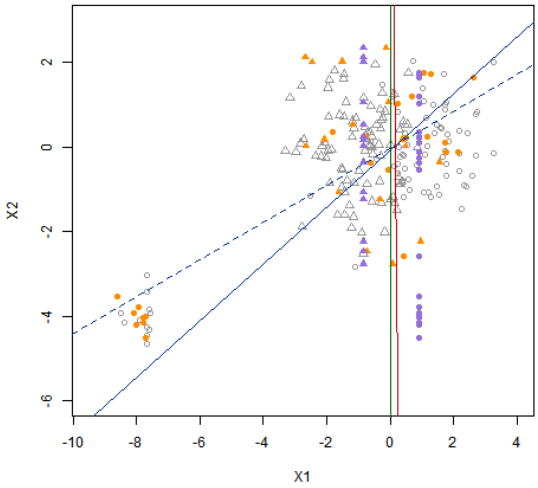


Figure 4.5: Illustration of the missing values estimation with conditional median imputation in a 10% contaminated setting. *The orange symbols represent the missing values (20% missing on X_1) while the purple symbols represent their imputation. The red line represents the robust discriminating line.*

To illustrate the k-NN imputation results, we will use the $p = 3$ dimensional case. Indeed, the $p = 2$ dimensional case is in fact a specific and too limited case in the sense that the neighbours selection is done exclusively according to one dimension (the distance being computed only on the non-missing dimension). This implies that the selected instances are not *real* bivariate neighbours as illustrated on figure 4.6 where the neighbours are selected only according to X_2. Indeed, we can notice there that all neighbours (red triangles or circles depending on the population they come from) are located along a *parallel* to the first axis, illustrating the fact that only distances in the dimension set by X_2 have been used to find the neighbours. Moreover, while the two populations are quite well discriminated on the plane, among the selected neighbours we see observations coming from the other population, which seems odd especially for the orange circle which lies at the outward frontier of population 1. Figure 4.7 represents the $p = 3$ dimensional case following the same idea as our 2-dimensional theoretical model without contamination. We can see that each missing value is imputed by a value *close* to the actual one. Also, the neighbours appear to be *real* neighbours. This better reflects what will happen in the simulations with the $p = 5$ dimensional case. However, it is important to note that as the class membership is not taken into account in the neighbours selection, any close neighbour can be selected, regardless of the population, especially when the missing observation is located in the *overlap area*. This means that we could observe a *group change* in the case where the majority of neighbours belong to the other population. Indeed, the estimated imputation value is expected to be closer to the values of the population to which the majority of neighbours belong. Therefore, the imputed value would be located in the other population group, at least according to the missing dimension, which could have an impact on the classification results.

Let us turn to the performance of k-NN imputation under contamination. Three main contamination configurations will be considered:

- Configuration n°1: contamination on X_1.
- Configuration n°2: contamination on X_1 and X_2.

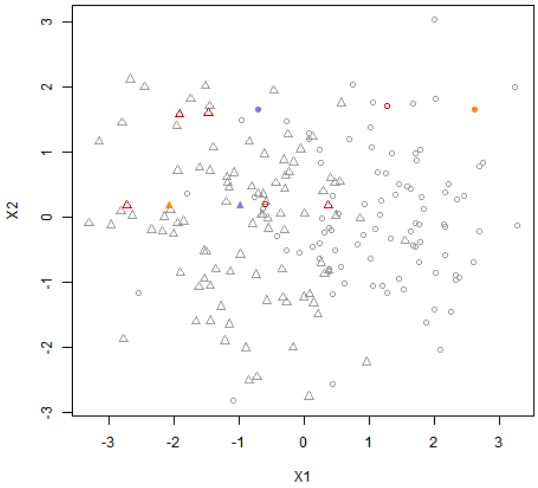
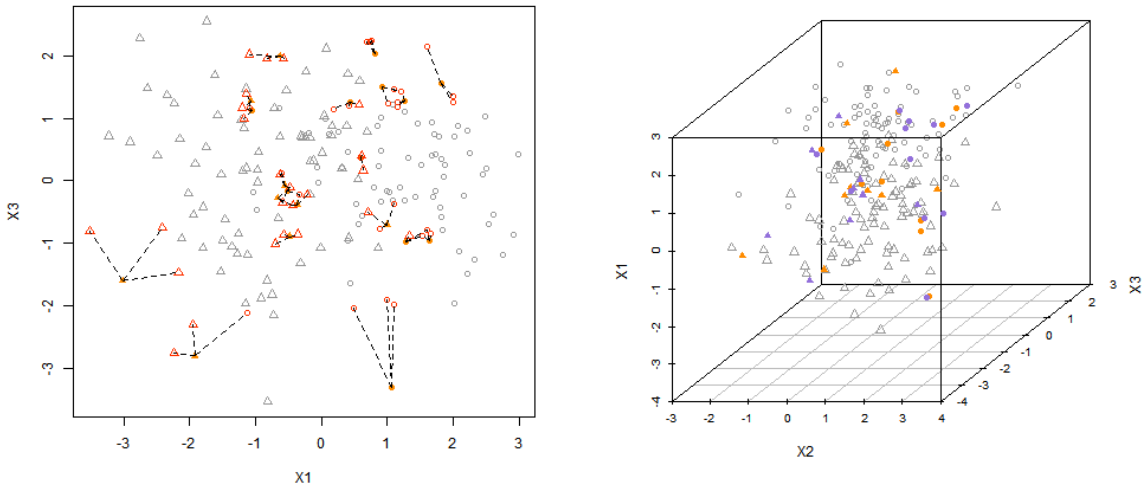


Figure 4.6: Illustration of the 3-NN neighbours selection in the 2-dimensional independent case. *The orange symbols represent the missing values (missing on X_1) and the purple symbols represent their imputation. The red shapes are the selected neighbours.*



(a) 3-NN Neighbours selection. *The red shapes represent the selected neighbours.*

(b) 3-NN imputation. *The purple symbols represent the imputation.*

Figure 4.7: Illustration of the 3-NN imputation under a balanced mixture of 3-dim normal distributions.

– Configuration n°3: contamination on all dimensions X_1, X_2, X_3.

Focus is on contamination on the discriminating dimension X_1 because it is expected to have a greater impact on the classification rule. The missingness can occur on one of the three dimensions. Depending on the contamination configuration and the missing dimensions, the imputation results may be different. By analysing the different situations, we are trying to identify the cases where contaminated instances can be cleaned. The investigation is done by considering extreme *bad* contamination.

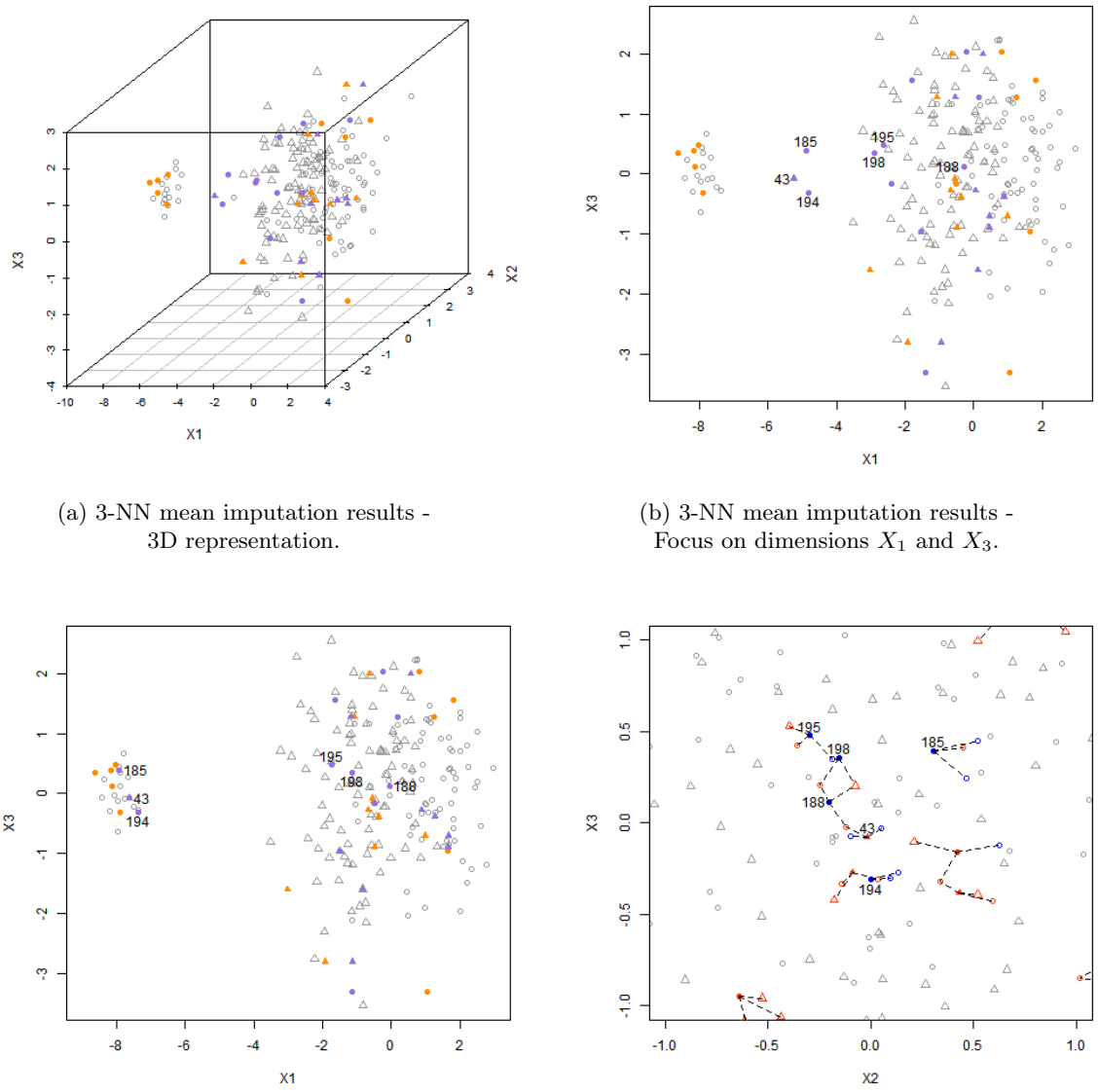
Configuration n°1 In this first configuration, the contamination occurs only on the discriminating dimension X_1. Regarding the missingness, we can make the difference between two cases: whether the missing values are introduced in the contaminated or uncontaminated dimension.

Let's first consider the case where the contamination and missingness occur on the same dimension X_1. In that case, the neighbours are selected according to X_2 and X_3 and the estimation will be computed using the X_1 values of the neighbours. The neighbours being selected on the clean dimensions, any close instance can be selected, regardless of whether the missing value was initially clean or contaminated. What will make the difference then is whether the selected neighbours have a clean or contaminated X_1 value. This way, for any instance, if all neighbours are clean then the imputation will be clean as the mean is computed on clean X_1 values. If the majority of neighbours are clean, then the median estimation should produce a clean instance as well. By contrast, if all or the majority of neighbours have contaminated X_1 values then the imputation will be unclean. The result will also depend on the number of neighbours k in the sense that it will allow more or less clean or unclean instances to be selected depending on the instances. All of this implies that four situations can happen:

- Outlying instance $\xrightarrow{Imputation}$ Clean instance,
- Outlying instance $\xrightarrow{Imputation}$ Outlying instance,
- Clean instance $\xrightarrow{Imputation}$ Outlying instance,
- Clean instance $\xrightarrow{Imputation}$ Clean instance.

Figure 4.8 illustrates an example of this configuration. The contamination only affects the discriminating dimension X_1 and the missing values are introduced in that same dimension. Figure 4.8a is the 3D representation of the 3-NN mean imputation results. Focusing on two dimensions (here X_1 and X_3) enables to easily identify certain instances with particular results. For instance, initially observations $185, 188, 194, 195$ and 198 were contaminated and instance 43 was clean. After 3-NN imputation using the mean as estimator, it can be seen on figure 4.8b that instances $188, 195$ and 198 became clean. Instance 43 is now outstanding the main data group and instances 185 and 194 are no longer as extreme outliers as they were initially. The results are different when the median is used for the estimation (fig. 4.8c). Indeed, while instances $188, 195$ and 198 are again clean, the three other instances are now completely outlying. This can be explained on the basis of figure 4.8d which represents the neighbours selection, focusing on the particular instances of interest. The blue symbols are instances which have a contaminated X_1 value and the red symbols are the clean instances. The shapes connected to the symbols by dashed segments represent the neighbours. It can be seen that instance 43 has, among its three neighbours, two which have unclean X_1 values. Therefore, when the mean is used, it produces a value in between the two data groups while with the median, the instance is attracted by the contaminated values. The imputation thus produces a contaminated instance from an initially clean one. By contrast, all neighbours of the instance 188 are clean which produces a clean imputation value. It should be noted that the results obtained here clearly depend on the number of selected neighbours k. Here, k was taken small for the ease of illustration and this explains why the median is less robust than the mean, what is somewhat counter-intuitive. Figure 4.9 illustrates the same example in the case of 5-NN imputation. In that case, both with the mean and median, the instance 43 is outlying but all the contaminated instances are now clean. However, it is important to make the difference between *clean* with respect to the main data group and *clean* with respect to the population group. Indeed, not all of the initially con-

taminated values that are now located within the main data group, are positioned in the expected area as far as the population is concerned. This is an example of the group change phenomenon described earlier.



(a) 3-NN mean imputation results - 3D representation.

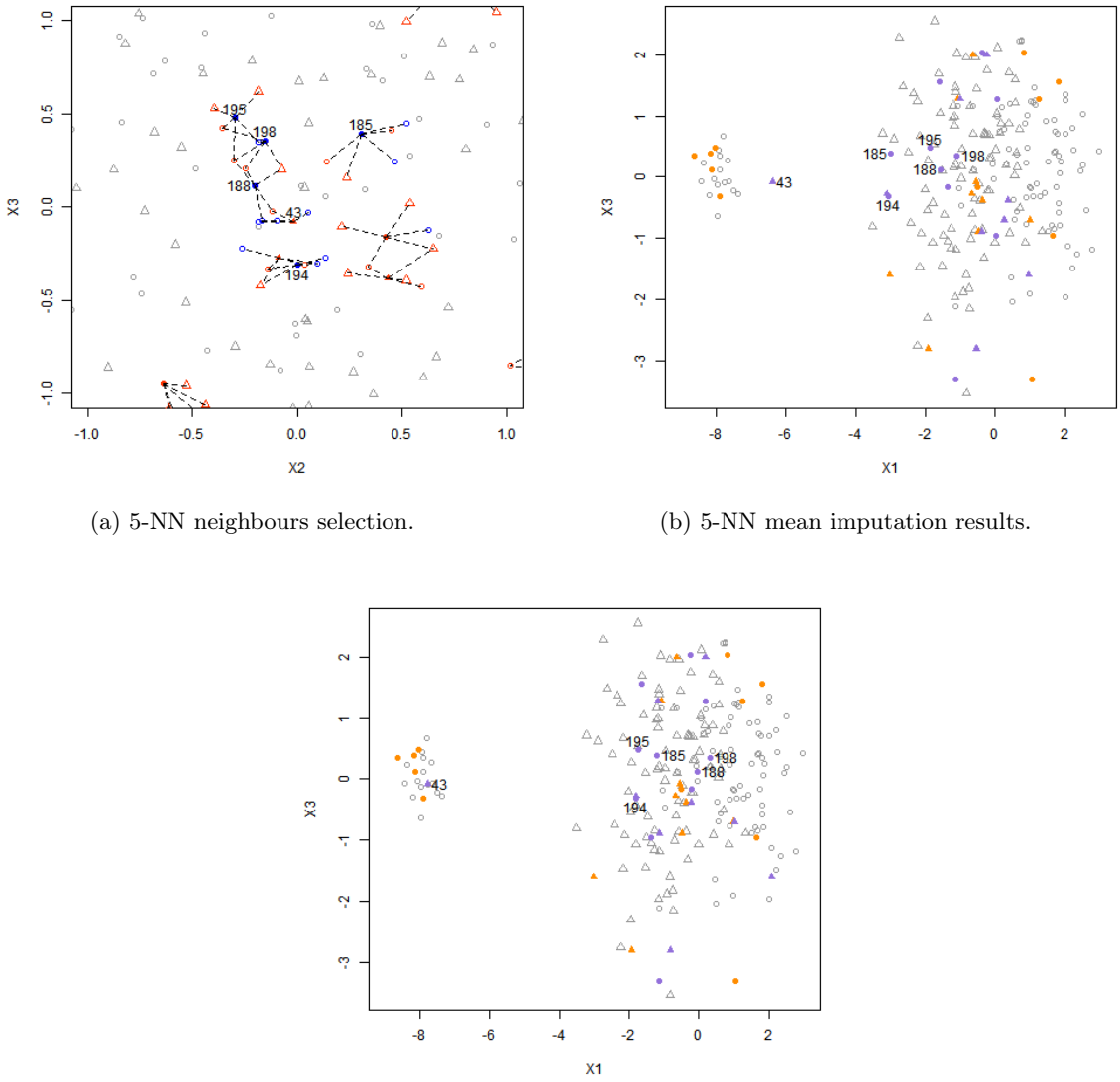
(b) 3-NN mean imputation results - Focus on dimensions X_1 and X_3.

(c) 3-NN median imputation results - Focus on dimensions X_1 and X_3.

(d) 3-NN Neighbours selection - Focus on specific instances.

Figure 4.8: Illustration of the 3-NN imputation in configuration n°1 : 10% contamination on X_1 and 10% missingness on X_1. *The red symbols represent the clean missing instances while the blue symbols represent the contaminated missing instances. The red/blue shapes represent the selected neighbours.*

In case the missing values are introduced in one of the two clean dimensions X_2 or X_3, the analysis is straightforward in the sense that the contaminated instances will simply remain contaminated as their X_1 values won't be modified.



(a) 5-NN neighbours selection.

(b) 5-NN mean imputation results.

(c) 5-NN median imputation results.

Figure 4.9: Illustration of the 5-NN imputation in configuration n°1 : 10% contamination on X_1 and 10% missingness on X_1. *The red symbols represent the clean missing instances while the blue symbols represent the contaminated missing instances. The red/blue shapes represent the selected neighbours.*

Configuration n°2 In configuration n°2, contamination occurs on the two first dimensions X_1 and X_2 (note that it would be the same if X_3 was contaminated instead of X_2). Again, we can distinguish the case where the missing dimension is contaminated from the one where it is uncontaminated.

Let's consider missingness on X_1. In that case, neighbours are selected according to both a contaminated and a uncontaminated dimension. This implies that when k is *small*, a clean missing instance will have clean neighbours as the cluster of outliers has been put *far away* whereas a contaminated missing instance will have at least one contaminated neighbour (assuming that the clusters are not restricted to singleton). The only way of producing a clean imputation value for an initially contaminated instance (only on X_1 which is the dimension of interest), is to have a higher proportion of clean neighbours in the selection. When k is *large*, we can expect the median to produce clean

results. Thus, several elements come into play in the determination of the imputation result : the number of neighbours k, the method used for the estimation (mean or median) but also the number of contaminated instances after missingness introduction and the proportion of clean vs unclean instances in the neighbours selection. The choice of the appropriate value for k will be investigated in Chapter 5. What has already been noticed is the fact that when k is too *large* the quality of the mean imputation can be deteriorated, at least for clean instances. It is important not to forget that our final goal remains the classification performances and so we try to clean as much as possible the contaminated instances but also to impute as well as possible the clean instances while maintaining them in their respective area as far as the membership is concerned.

Similar to configuration n°1, when the missing values are introduced in the clean dimension, here X_3, the contaminated instances will remain contaminated.

Configuration n°3 In the last configuration, all dimensions are contaminated. In that case, whether the missing values are introduced in X_1, X_2 or X_3 will have the same effect. Indeed, the neighbours being selected on contaminated dimensions, this is the same as the first case of configuration n°2.

Through this analysis, we highlighted the different situations that may occur. It has been shown that several parameters come into play, e.g the initial contamination percentage as well as after missingness introduction, the contamination dimensions, the missingness percentage, the number of neighbours, the estimator used,... each of which can influence the imputation result. We will try to monitor the different elements at best in the simulations.

As illustrated in the previous subsection about mean and median imputation, when the imputation method succeeds in converting contaminated instances into clean instances while remaining in the expected area as far as the membership is concerned (at least on the discriminating dimension), this limits the impact of the outlying values on the classification rule. However, here again, when the contamination level remains high, a robust classification method should be considered.

4.5.3 SEQimpute and ROBimpute

The last two imputation methods studied here are the SEQimpute and ROBimpute techniques. The SEQimpute method is a sequential imputation method which exploits the data set covariance matrix and its determinant. This method is called sequential because it estimates the missing values by successively considering one incomplete instance at a time. Taking one incomplete instance $x^* = [x^*_m, x^*_o]$ (where x^*_m and x^*_o respectively represent the missing and observed parts) and the set of complete observations of the data X_{obs}, the missing values of the incomplete instance are imputed by minimizing the determinant of the data covariance matrix $X = [X_{obs}, x^*]$.

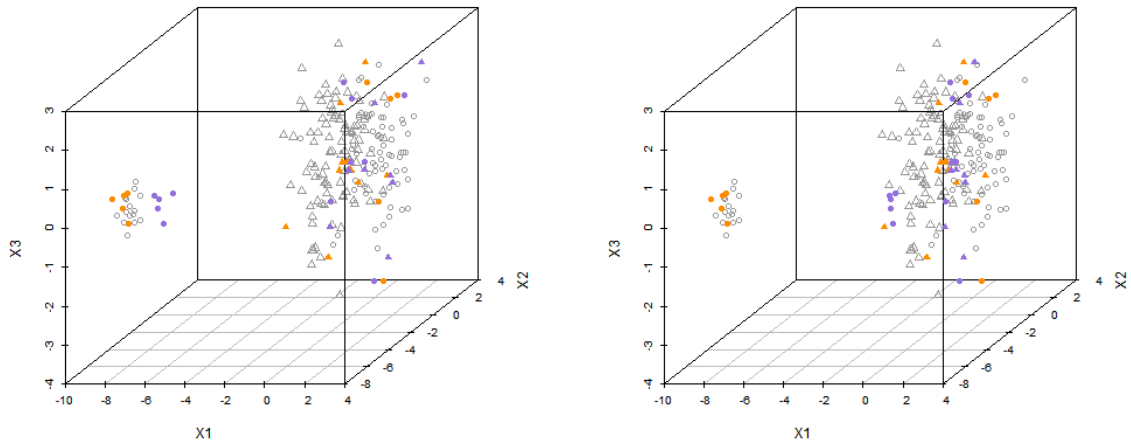
$$\hat{x}^*_m = \text{argmin}_{x^*_m} |cov([X_{obs}; x^*])|$$

Then, this new complete instance is added to the complete set and the process continues until all missing values are replaced. This method is driven by the principle that the determinant of the covariance matrix is proportional to the volume of the tolerance ellipse and thus the goal is to minimize in that way the concentration of the data. The detailed algorithm is presented in [33].

The ROBimpute method is the robust version of the SEQimpute method and is obtained from it by robustifying the computation of the mean and covariance matrix. The robust estimators are computed using $(\alpha * 100)\%$ of the observed data with the smallest outlyingness, where $(1 - \alpha)$ is

the estimated contamination proportion and the outlyingness is measured following the idea used in Hubert et al. [15]. As we are in a multivariate setting, the data points are projected on each considered direction before computing the univariate outlyingness [15]. (Other robustification methods could also be used). The missing values are then imputed in the same way as for SEQimpute. More details can be found in [32].

Figure 4.10 illustrates the imputation results of the SEQimpute and ROBimpute methods in a 3-dimensional independent setting with 10% contamination on X_1 and X_2 and 10% missingness in X_1. The ROBimpute method succeeds in cleaning the missing contaminated instances (at least on the missing dimension X_1) while this is clearly not the case for SEQimpute.



(a) SEQimpute. (b) ROBimpute.

Figure 4.10: Illustration of the classic and robust sequential imputation in the 3-dim case with 10% contamination on X_1 and X_2 and 10% missingness on X_1.

4.6 Classification techniques

As outlined before, we are interested in evaluating the impact that imputation can have on classification performances. This will be done by applying classification methods on the complete data sets obtained after imputation.

Three main classification techniques will be compared:

- Linear discrimination (classic and robust)
- K-Nearest Neighbours classifier
- Gaussian Naive Bayes classifier

This choice has been made so as to compare three types of classification method, namely, a statistical method, a machine learning method and an intermediary method.

The analysis of imputation impact on the classification rule in the previous section has highlighted the need of robust classification methods when the contamination remains significant, either because the imputation does not succeed in cleaning the contamination or because the contamination level

is still high. For that reason, we will investigate the linear discrimination technique by considering different estimations including a robust one like it has been done in subsection 4.5.1.

Regarding the Naive Bayes classifier, working with continuous numerical values requires to make the assumption that the values associated with each class are distributed according to a parametric distribution. Our theoretical model precisely meets this condition as the likelihood of the features is indeed Gaussian.

The literature review has shown that most of the studies investigating missing values in the context of classification consider the testing set to be free of missing values. However, it seems rational to believe that new instances to be classified are even likely to contain missing values if the training set contains any. For that reason, as done by Choudhury and Kosorok [7], we will generate both complete and incomplete testing sets based on the same underlying distributions. Recall that in the case of test instances, the use of class membership information for the imputation will not be possible. We have seen that in some cases, this information can be useful to improve the learning of the classification rule. However, none of the techniques absolutely require the class information to operate. Thus, each technique can be applied directly to impute the incomplete test sets.

4.7 Evaluation

Various model evaluation techniques are available in the context of classification. In some real classification cases, one of the classification outcomes may be much sensitive, like in diagnostic classification. If that is the case, one should prefer evaluation techniques measuring the proportion of false positives or false negatives like precision or recall measures. However, in our simulation case, the interest is more on the global performances of the models as neither class is of particular concern. Indeed, we are more interested on the global impact of the combination of imputation and classification methods on the classification results and thus in comparing the overall classification performances. In that respect and in our particular balanced case, we can rely on the accuracy measure, and equivalently the misclassification error, for the evaluation.

In addition, in our simulated setting, we have all necessary information to follow the suggestion of Liu and Brown [21] which consists in measuring the difference in misclassification error between the original complete data set and the complete data set after imputation.

$$\delta_{err} = \frac{(e_2 - e_1)}{e_1} \tag{4.4}$$

where e_1 and e_2 represent the misclassification error on the original data and the imputed data respectively. The lower the value of δ_{err}, the better the effect of the combined imputation and classification techniques.

Our simulation setting will enable us to put into perspective the results obtained after classification with the evolution of the data sets characteristics throughout the process from the starting situation, in particular the contamination proportion. An additional benchmark will thus be used (see Chapter 5) in order to take into account the contamination process. Indeed, it will be interesting to know if the imputation has succeeded in improving the situation prior to classification. This will help us for the analysis and understanding of the final results.

Chapter 5

Simulation set-ups

5.1 Introduction

Now that our theoretical framework is well-defined, we can move on to the precise design of the simulations. The experimentation will thus be performed in a simulated setting with data generated according to the theoretical framework. Using simulated data allows full control over the data set and its characteristics which enables a more accurate comparison and analysis of the results, especially in the context of contaminated data.

Following the procedure summarised in the diagram of the figure 5.5[1], the first step consists in generating the (un)contaminated data sets. Next, missingness can be artificially introduced into the data. Several imputation methods are then used to fill in the missing values. Finally, classification methods can be applied on the complete imputed data sets in order to evaluate the imputation quality and the impact on the classifiers' performances.

5.2 Procedure

5.2.1 Data set

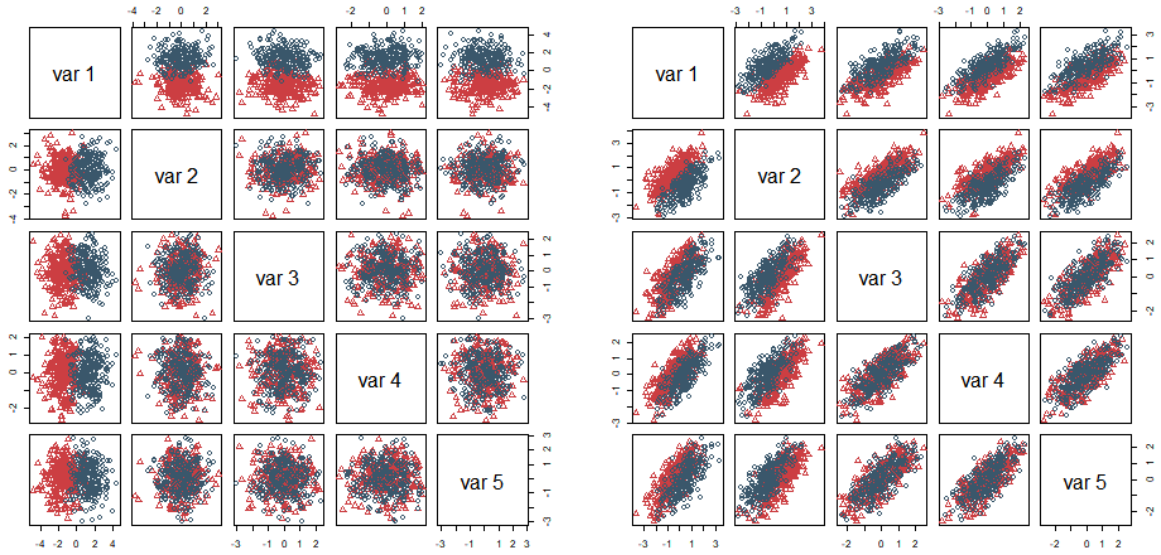
For the simulations, we will work with two-class data sets containing $n = 500$ observations from a balanced mixture of multivariate normal distributions of dimension $p = 5$. The two covariance structures, independent and equi-correlated, will be investigated. Recall that, in both cases, we assume homoscedasticity. More precisely, the two considered settings are described as follows.

Independent setting Following the theoretical model, in the independent configuration, the covariance matrix Σ is set equal to the identity matrix $\mathbf{I}_5$. The means μ_0 and μ_1 are taken in such a way that $\mu_1 = -\mu_0$ with the additional constraint that the last four coordinates are equal to zero. This last constraint forces the first dimension to be the only discriminating one.

Correlated setting In the correlated case, we consider an equi-correlated covariance matrix Σ, where $\sigma_{ii}^2 = 1$ and $\sigma_{ij}^2 = 0.75$ $(i \neq j)$. The means μ_0 and μ_1 are then taken on the orthogonal space with respect to the first eigenvector of Σ, with the additional constraints that $\mu_1 = -\mu_0$ and the last three coordinates are equal to zero. The discriminating dimensions, in this case, are thus X_1 and X_2.

[1] see p.44

Figure 5.1a and 5.1b represent pairs plots of the balanced mixture of 5-dimensional normal distributions under the independent and correlated setting respectively. In particular, $\mu_0 = (-1.5, 0, 0, 0, 0)$ in the independent setting and $(-0.5, 0.5, 0, 0, 0)$ in the correlated case. Those particular settings have been chosen so as to guarantee at least a slight overlap between the two data groups. The means have thus been taken so that the probability of misclassification (4.2) is about 7% which is a quite satisfactory classification error. One can clearly see on the plots of the left panel, that the two colors can only be distinguished on the first row/column while the populations are completely mixed on the other dimensions. On the right panel, the discriminating variables X_1 and X_2 may also be spotted quite easily.



(a) Independent setting.

(b) Correlated setting.

Figure 5.1: Illustration of the balanced mixture of 5-dim normal distributions.
Red triangles belong to P_0 and grey circles belong to P_1

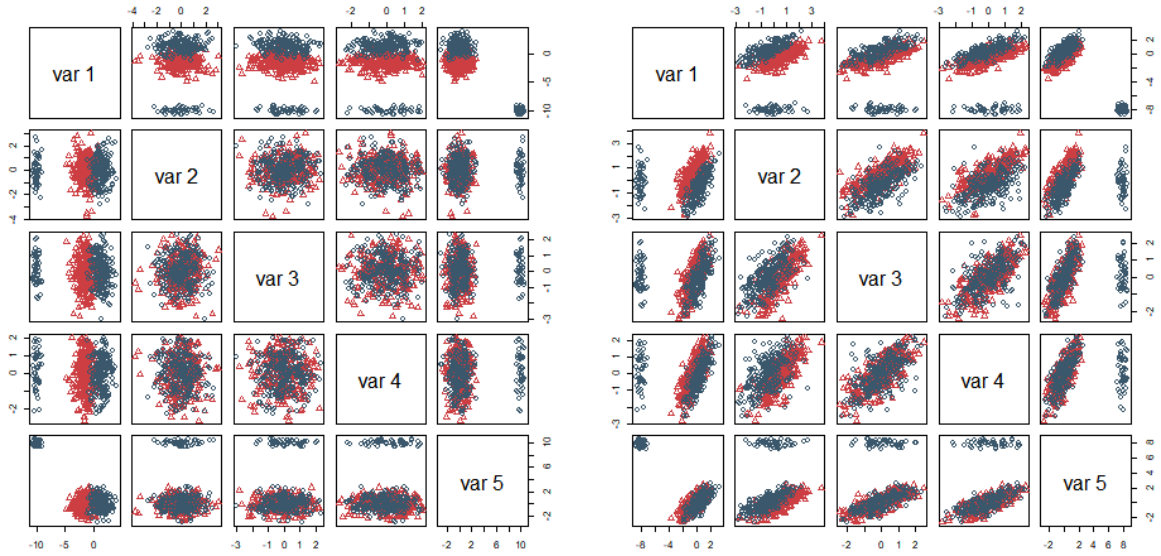
For the classification, we will need both training and testing sets to properly evaluate the performances. Therefore, besides the training data sets, testing sets containing $n = 200$ observations will be generated according to the same underlying distributions. This is equivalent to a 70-30% split approximately.

5.2.2 Contamination

The next step is thus to introduce outlying values in the generated data according to the contamination process described in section 4.3. The data sets will be contaminated by different percentages of 5%, 10% and 20% of the data set size. Recall that the contamination will only be introduced in population P_1 and thus $\epsilon = (\pi_1 * \epsilon') * 100\%$ where π_1 is the prior probability of population P_1. Experiment will also be done on the clean data sets ($\epsilon' = 0$) for comparison purposes. The contamination level will be monitored at the different stages of the process.

It has been shown in the previous chapter that depending on the position of the contaminated cluster, the imputation results can be different. Remember that this position is determined by the distribution mean μ_c. Here, we have decided to focus on extreme *bad* leverage points. Two of the five dimensions will be contaminated, one of which is discriminating.

In practice, the initial data set will be generated so as to contain $(1-\epsilon)*n$ *good* observations to which we add $\epsilon * n$ observations from the contaminated model. The outlying data points will be generated according to the following model : $\mathcal{N}_5((a,0,0,0,b),\Sigma_c)$ where $\Sigma_c = \text{diag}(0.1,1,1,1,0.1)$. Extreme *bad* outliers (with respect to the discriminating dimension) correspond to $a < 0$. For instance, $a = -10$ and $b = 10$ in the independent setting while $a = -8$ and $b = 8$ in the correlated one, as represented on figure 5.2a and 5.2b respectively. Those differences in configuration are induced by the difference in the shape of the distribution in the two cases. In fact, as the contamination only occurs on two dimensions, this has an impact on the outliers detection in particular for high contamination proportion.



(a) Contaminated independent setting.

(b) Contamined correlated setting.

Figure 5.2: Illustration of the balanced mixture of 5-dim normal distributions under contaminated setting.

It should be noted that the testing sets are not generated under the contaminated model. In fact, the idea is to evaluate how the classifiers built on contaminated data perform on *good* data.

The contamination will be monitored at the different steps of the simulation process, namely after missingness introduction and after imputation. Remember that contaminated instances will possibly be cleaned up only if the contaminated value become missing. In our particular setting, the dimension X_5 will not be involved in the missingness process as will be explained in the next section. Therefore, the *global* contamination level (with respect to the five dimensions) is not expected to decrease all at once after imputation. The discriminating dimension X_1, however, will be both under contamination and missingness. It is thus in this dimension that some improvements might be observed. We will thus also measure the contamination proportion in regard to this particular dimension. In practice, the *global* outliers detection is done using the Mahalanobis distance computed on the theoretical parameters of the distributions. Given that contamination is only introduced in the training set, this means we have the class membership information at our disposal to perform

the detection on each population separately. Using as threshold the 99.5% quantile of the χ^2 distribution with $p = 5$ degrees of freedom ensures a global probability of erroneously detecting an observation of 1%. The detection according to X_1, on the other hand, has been done by computing a z-score with respect to the true mean and standard deviation of variable X_1. Again, the detection is done in each population separately.

5.2.3 Missingness introduction

Once the data sets are built, missing values can be artificially introduced both in the training and test data. Different percentages of missing values will be investigated, namely 10%, 20% and 40%. Again, the complete data sets will be used as a baseline.

As explained in section 4.4, missing values are introduced only in certain variables while the other covariates remain non-missing. Dimensions X_1 and X_2 were chosen as the missing dimensions. The reason for this is that we are ultimately interested in the classification performances and therefore we should focus on variables most likely to be incorporated into the classifier, i.e. the discriminating dimensions.

MCAR Under this first missingness mechanism, the missing values are introduced randomly into the data without any dependence on variables of the data set. Each cell has the same probability of having a value missing.

MAR In this case, the missing values will depend on other variables present in the data set. For instance, it has been decided to design the MAR mechanism as follows: let M denote a binary matrix indicator of missingness. The success probability of M will be modeled by means of a function depending on some covariates. The parameters of the model need to be fixed such as to achieve a pre-fixed probability of being missing. To do so, the following strategy (illustrated on the simpler 2D case) has been chosen:

Assuming we have a data set with two variables and only X_2 contains missing values. The probability of an observation to be missing under MAR will depend on X_1 and has been modeled here as follows:

$$P(M = 1 \mid X_1 = x_1) = \frac{1}{1 + \exp(a + bx_1)}$$

for given constants a and b. Other models could have been used, e.g. a logistic model.

The aim is to fix a and b such that the overall missing probability for any observation of X_2 is equal to the respective percentage of interest, i.e,

$$P(M = 1) = \pi$$

where $\pi = 0.10, 0.20$ and 0.40 respectively.

According to the extension of the formula of total probability for the continuous case, we have

$$P(M = 1) = \int_{\mathbb{R}} P(M = 1 \mid X_1 = x_1) f_{X_1}(x_1) dx_1. \tag{5.1}$$

The value of the coefficients can be found by solving equation (5.1) in terms of a and b using numerical integration so as to reach the desired percentage π.

In the present case, the model is much more complex as not just one but three variables come into play in the regression expression. Indeed, we have

$$P(M_{1,2} = 1 \mid X_3 = x_3, X_4 = x_4, X_5 = x_5) = \frac{1}{1 + \exp(a + bx_3 + cx_4 + dx_5)}. \tag{5.2}$$

However, the coefficients determination process remains the same. Once the coefficients are determined, we are able to build the binary matrix indicator of missingness M as we know that the probability for any value of X_1 and X_2 to be missing depends on the values of X_3, X_4 and X_5 as defined in (5.2). Then, the observations of X_1 and X_2 for which the missingness indicator takes the value 1, will be set as missing. Concretely, as far as the coefficients are concerned, we fixed $b = c = d = 1$ and a differs depending on the target proportion π and the covariance setting. Figure 5.3 illustrates the probability function used to insert 20% of missing data in the independent setting.

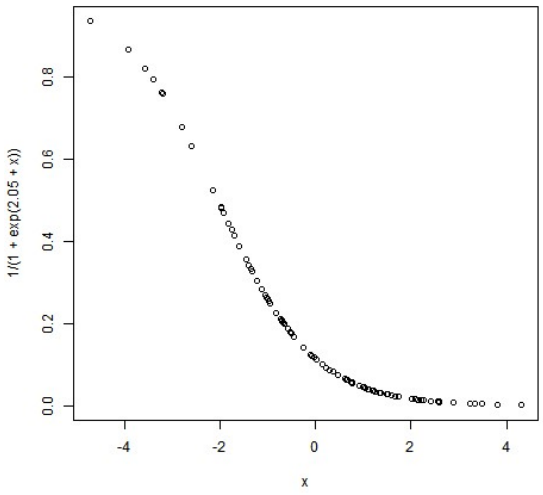


Figure 5.3: Missingness probability function for 20% of missing values in the independent setting.

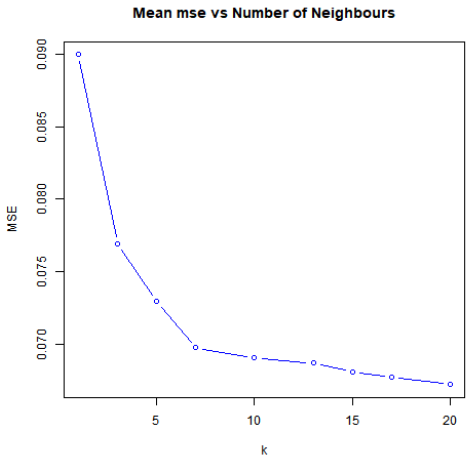
In their articles, Verboven and Vanden Branden [33, 32] put emphasis on the idea of keeping 5% of data free of missing value. Indeed, some imputation methods considered need a subset of clean instances to be able to operate. As regards the definition of our missing values introduction strategy, this condition should be satisfied either way.

The total number of data sets at this stage is $2 \times 3 \times 3 + 2 = 20$. Indeed, the three contamination percentages (5, 10 and 20%) and the three missingness percentages (10, 20 and 40%) will be evaluated for both data sets (independent and correlated). In addition, the clean and complete data sets are used as a baseline.

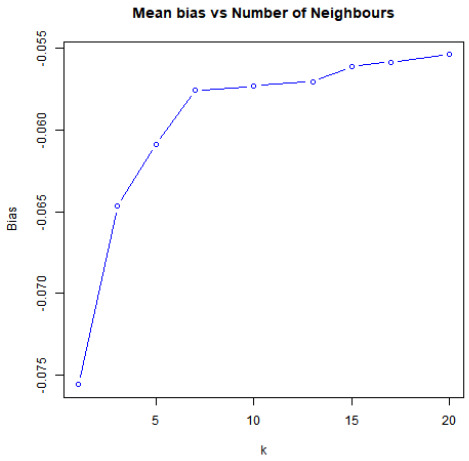
5.2.4 Imputation

The implementation of the deletion technique as well as the mean and median imputation are quite straightforward. We showed in the previous chapter the benefit on the classification model of using the conditional mean/median to impute the training set while this is not practicable for the test set. In the simulations, we will thus use the conditional estimator for the imputation of the training set and the unconditional estimator for the test set imputation.

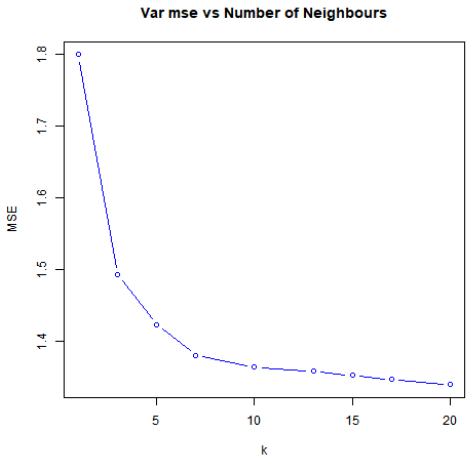
The k-Nearest Neighbours imputation has been implemented from scratch. An important matter in that method is the determination of the parameter k, the number of nearest neighbours selected for the estimation. The literature on this subject in the particular context of imputation is not really developed. The main reflections thereon are the following: (1) the smaller the k, the greater the variability of the estimation, (2) the more k increases, the more neighbours are involved in the calculation which implies the loss of the local character of the estimation. In addition, the estimation may be less precise as the distance to the neighbours gets bigger [18]. We decided to investigate this by means of simulations. Concretely, we inserted missing values in generated data and then imputed them using different values for the parameter k, taking k in the range $1 - 20$. Each time, we computed the bias and mean squared error of the mean and variance of the variables containing missing values. The simulations have been carried out for the uncontaminated data set only.



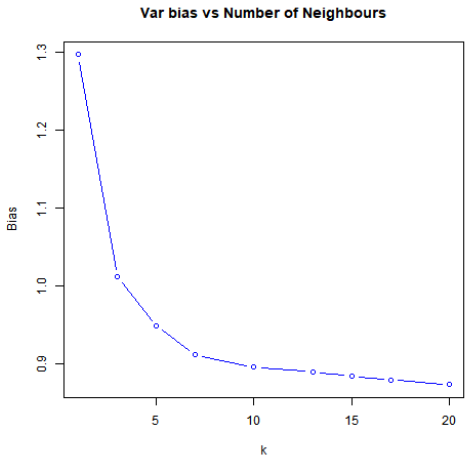
(a) MSE computed on the mean



(b) Bias computed on the mean



(c) MSE computed on the variance



(d) Bias computed on the variance

Figure 5.4: Results of simulations to determine the optimal value of k.

The results are shown in figure 5.4. It can be seen on the two first sub-figures that the MSE and bias computed on the mean respectively decreases and tends towards zero as k increases. Indeed, the mean will get better with k increasing. In fact, with large value of k, the estimation will tend more

and more towards the real mean value. Regarding the variance, its value is expected to decrease as k increases. We can indeed see, on the two last sub-figures, decreasing MSE and bias. However, those results do not highlight any particular value. To preserve the local character of the estimation, we decided to choose $k = 5$ as this value already causes a relative decrease of the MSE and bias.

As far as the SEQimpute and ROBimpute methods are concerned, their R implementation (`impSeq` and `impSeqRob`) in the package *rrcovNA*[2] have been used. As explained in subsection 4.5.3, the ROBimpute method relies on a parameter α determining the proportion of observations with the smallest outlyingness to involve in the estimation. This parameter will depend on the proportion of data contamination. In our simulations, we used the default value $\alpha = 0.9$ when $\epsilon = 0.05$ and $\alpha = 0.75$ for the higher contamination proportions.

5.2.5 Classification and evaluation

Evaluation of the classification performances is done by means of the misclassification error and the classification criterion presented in section 4.7 (eq. 4.4):

$$\delta_{err} = \frac{(e_2 - e_1)}{e_1}$$

where e_1 and e_2 represent the misclassification error on the original data and the imputed data respectively. Note that we are trying to evaluate whether imputation and classification can mitigate missingness and contamination effect. Therefore, e_1 is actually computed on the clean and complete data sets (no contamination and no missing values). In practice, considering our contaminated data generation process, this is done by fixing the same seed for the generation of both data sets, clean and contaminated.

When analysing the results, we should keep in mind that the treatment of missing values with the case deletion technique actually reduces the size of the data set. Especially for the test set, this implies that a number of instances will be discarded and therefore not classified. This technique is thus not the most appropriate for treating missing data in the test set but we will still evaluate its performances.

5.2.6 Summary

Let's recap the main characteristics of our simulated data sets before presenting and analysing the results.

- Training set : $n = 500$, testing set : $n = 200$, $p = 5$
- Two covariance structures : independent and equi-correlated
- Contamination : bad leverage points on dimensions X_1 and X_5 of population P_1
  - $\epsilon = (0.05, 0.1, 0.2)$, only on training
- Missingness : MCAR and MAR on dimensions X_1 and X_2
  - $\pi = (0.1, 0.2, 0.4)$, both on training and testing

Choices had to be made concerning all those different aspects which we tried to justify at best. We believe those settings will enable us to observe on a larger scale the different situations we have encountered during our preliminary analysis in the previous chapter. As mentioned previously, our

[2] https://cran.r-project.org/web/packages/rrcovNA/rrcovNA.pdf

objective is to detect whether a combination of imputation and classification methods can produce satisfactory classification results in a context of contamination and missingness.

Over the global simulation process, we thus have four main settings: (1) MCAR in the independent setting, (2) MCAR in the correlated setting, (3) MAR in the independent setting and (4) MAR in the correlated setting. In each setting, there will be $3 \times 3 \times 7$ imputed data sets to classify using the four different classifiers, in addition to the three different baselines: (1) $\epsilon = 0$ and $\pi = 0$; (2) $\epsilon \neq 0$ and $\pi = 0$; (3) $\epsilon = 0$ and $\pi \neq 0$. Overall, this represents $(1 + 3 + 3 \times 7 + 3 \times 3 \times 7) \times 4$ data sets to be classified by the four classifiers, making a total of 1408 classification results. Furthermore, the process of generating data, inserting missing values, imputing them and performing the classification is repeated 100 times. The final presented results will then be the average of the outlyingness percentages and the average of the misclassification errors.

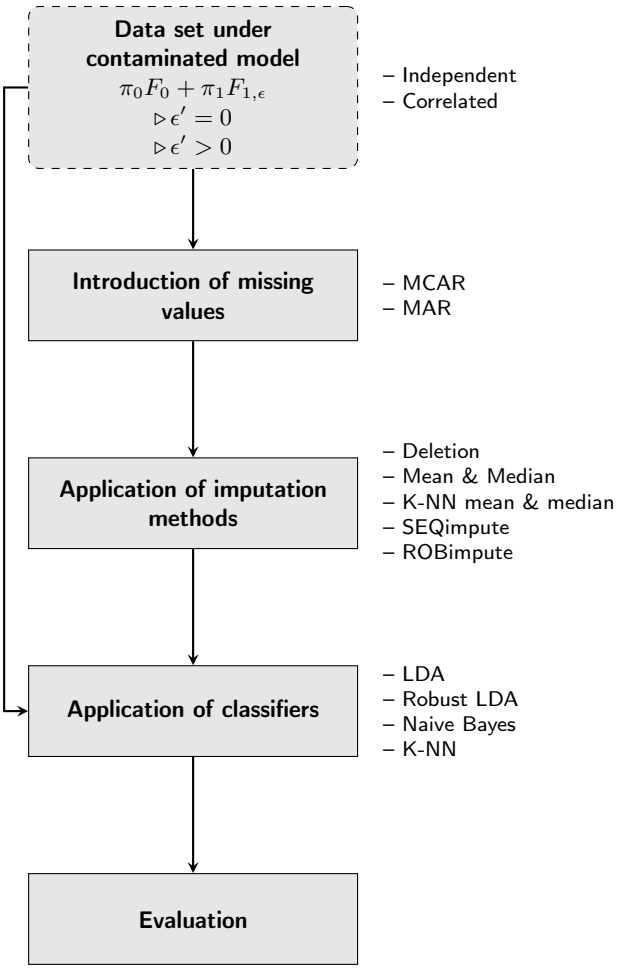


Figure 5.5: Simulation procedure

Chapter 6

Simulation results

6.1 Observations and analysis

6.1.1 Outlyingness

As explained in the previous chapters, by allowing values of contaminated instances to be missing, it was hoped that the data would be cleaned up when imputed using certain methods. To monitor the evolution of the contamination throughout the simulation process, we measured the outlyingness at the different steps, namely at the beginning, after missingness introduction and after imputation. Overall at the beginning, there is a slight increase in the measurement compared to the theoretical value as can be noticed in table 6.1.

ϵ	0.0	0.05	0.1	0.2
Global contamination	0.005	0.056	0.102	0.204
Contamination X_1	0.011	0.058	0.113	0.209

Table 6.1: Contamination measurements

Remember that the outlyingness is measured separately in the two populations. This is the reason why the measure should be put into perspective. In fact, an observation detected as outlying could either be a *real* extreme outlier or a *good* observation which is slightly out of its own population group whereas globally this observation is part of the main data group.

6.1.1.1 Outlyingness after missingness introduction

Note first that, after missingness introduction, as expected, the *global* outlyingness, measured on the five dimensions, remained constant. This is due to the fact that, in our framework, only the contaminated dimension X_1 can be missing, leaving the second contaminated dimension X_5 unmodified. For this reason, we decided to focus on the outlyingness measured with respect to X_1.

In the four simulation settings (MCAR in the independent setting, MCAR in the correlated setting, MAR in the independent setting and MAR in the correlated setting), the contamination levels after the missing values have been introduced are very similar and correspond well to what is expected as can be seen when comparing tables (a) and (b) of table 6.2. The table (b) shows the results for

the first simulation setting, the three remaining ones can be found in the appendix (tables A.1, A.2, A.3).

ϵ \ π	0.1	0.2	0.4
0.05	0.045	0.04	0.03
0.1	0.09	0.08	0.06
0.2	0.18	0.16	0.12

(a) Theoretical values

ϵ \ π	0.1	0.2	0.4
0.05	0.0510	0.0450	0.0325
0.1	0.1016	0.0807	0.0680
0.2	0.1888	0.1691	0.1255

(b) Results for MCAR in independent setting

Table 6.2: Outlyingness in X_1 after missingness introduction

Indeed, with the MCAR mechanism each observation (cell) has equal chance to be missing whether it is contaminated or not. In practice, a proportion π of observations in X_1 and in X_2 are randomly selected and set as missing. With the MAR mechanism, however, the probability of each cell to be missing depends on the values of the other dimensions. Remember that our objective was twofold: (1) to observe the impact of contaminated values on the imputation but also (2) to observe the impact of imputation on contaminated values. Therefore, we had to make sure that certain contaminated values would be missing. This has been done by considering their fifth dimension to be uncontaminated so as to prevent their missingness probability to drop to zero. We would have had a similar result if we had considered only dimensions three and four for the determination of the missing probability. In the four cases, we therefore have a decrease in contamination in line with the proportion of missing values introduced.

6.1.1.2 Outlyingness after imputation

The results after imputation clearly depend on the imputation method. Globally, we see no difference between the MCAR and MAR mechanisms. Depending on the imputation method, the covariance structure can have an impact as will be seen in the following.

Mean imputation The results obtained with mean imputation in the different settings are showed in table 6.3. We notice that the results depend on the contamination proportion. Globally, when the contamination proportion is less or equal to 0.1, the outlyingness remains similar to the one after missingness introduction. However, for higher contamination proportion, there is an increase in the outlyingness values above the initial values. Indeed, as the mean is not robust to atypical values, when the contamination is significant, the estimation is affected and some observations that were initially good become contaminated.

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.051	0.045	0.033	0.053	0.048	0.035	0.052	0.046	0.034	0.052	0.046	0.035
0.1	0.102	0.081	0.072	0.099	0.089	0.068	0.098	0.094	0.101	0.096	0.083	0.065
0.2	0.238	0.269	0.326	0.238	0.265	0.325	0.236	0.268	0.319	0.235	0.264	0.307

Table 6.3: Outlyingness in X_1 after imputation by the mean

Median imputation As can be seen in table 6.4, in all cases, the imputation by the median produces similar results to those obtained after missingness introduction. This shows that the median is able to replace by clean observations the contaminated ones that were missing and is not affected by the remaining outliers.

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.051	0.045	0.032	0.053	0.048	0.035	0.052	0.046	0.034	0.052	0.046	0.035
0.1	0.102	0.081	0.068	0.099	0.089	0.068	0.098	0.088	0.066	0.096	0.083	0.065
0.2	0.189	0.169	0.125	0.186	0.166	0.123	0.187	0.164	0.130	0.184	0.164	0.123

Table 6.4: Outlyingness in X_1 after imputation by the median

K-NN mean and median imputation Here, we have to make a difference based on the covariance structure of the data. In the independent setting, there is an increase in the outlyingness results compared to the initial values, before missingness, for both k-NN methods as shown in table 6.5 and 6.6 respectively. In this case, however, the cause is not related to the contaminated values. Indeed, when analysing what is actually happening during the simulations with those imputation methods, we noticed that none of the *good* incomplete instances has contaminated neighbours selected for the estimation. Also, none of the contaminated values which became missing is cleaned up after imputation, contrary to what might be expected given the observations made in Chapter 4. This is because, as the contaminated dimension X_5 is involved in the distance computation, *good* instances will tend to have *good* neighbours and contaminated instances will tend to have contaminated neighbours.

The increase in outlyingness is then related to the way we are measuring it and to the characteristics of our data. As we now know that contaminated instances and good instances do not *mix* for the estimation, let's concentrate on the initially good instances which became contaminated after imputation. The computation of the distance for the neighbours selection is done based on dimensions X_2, X_3, X_4 and X_5 if the value is missing in X_1; X_1, X_3, X_4 and X_5 if the value is missing in X_2 and X_3, X_4, X_5 if both values X_1 and X_2 are missing. However, as stated in the data set presentation of the previous chapter, in the independent setting, only X_1 is discriminating. This means that on all other dimensions, we cannot distinguish the two populations as they are completely mixed (see figure 5.1a). This way, as the distance is computed on those dimensions, any close neighbour can be selected regardless of its population. The issue is then when a value of X_1 needs to be imputed. If the majority of neighbours belong to population P_0 then the mean of their X_1 values will be closer to the theoretical mean of the distribution of P_0 while if the majority of neighbours belong to P_1, the estimated value will be much closer to the mean of P_1. Similar to the observation made in Chapter 4, there is thus a *change of group* according to X_1 when the instance has a majority of neighbours belonging to the other population. This group change is identified by our outlyingness detection method as it is measured separately on the two populations. The situation remains the same when the median is used for the estimation as soon as there are 3 out of 5 neighbours belonging to the other population. This is not surprising as all observations are good observations. In order to *quantify* this phenomenon, we measured the outlyingness with respect to X_1 after imputation on the uncontaminated data set. As presented in table 6.7, the measures for the k-NN imputations in the independent settings are the highest of all. Those values are actually nearly equal to the increase observed in the contaminated case. Figure 6.1 illustrates the situation in

the clean and contaminated setting respectively. The red symbols are the *good* observations which are detected as outlying after imputation. We can clearly see that while those red observations are in fact closer to the mean of the other population and so at the edge of the tolerance ellipse, they are all in the main data group.

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.067	0.076	0.088	0.058	0.058	0.056	0.066	0.072	0.090	0.057	0.056	0.056
0.1	0.118	0.128	0.136	0.110	0.108	0.107	0.116	0.124	0.137	0.105	0.105	0.104
0.2	0.215	0.223	0.236	0.207	0.207	0.206	0.213	0.216	0.231	0.205	0.205	0.204

Table 6.5: Outlyingness in X_1 after imputation by k-NN mean

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.081	0.097	0.132	0.058	0.059	0.057	0.078	0.096	0.129	0.056	0.056	0.056
0.1	0.129	0.136	0.169	0.110	0.110	0.108	0.126	0.144	0.177	0.106	0.105	0.104
0.2	0.224	0.240	0.270	0.208	0.207	0.207	0.220	0.231	0.261	0.205	0.205	0.204

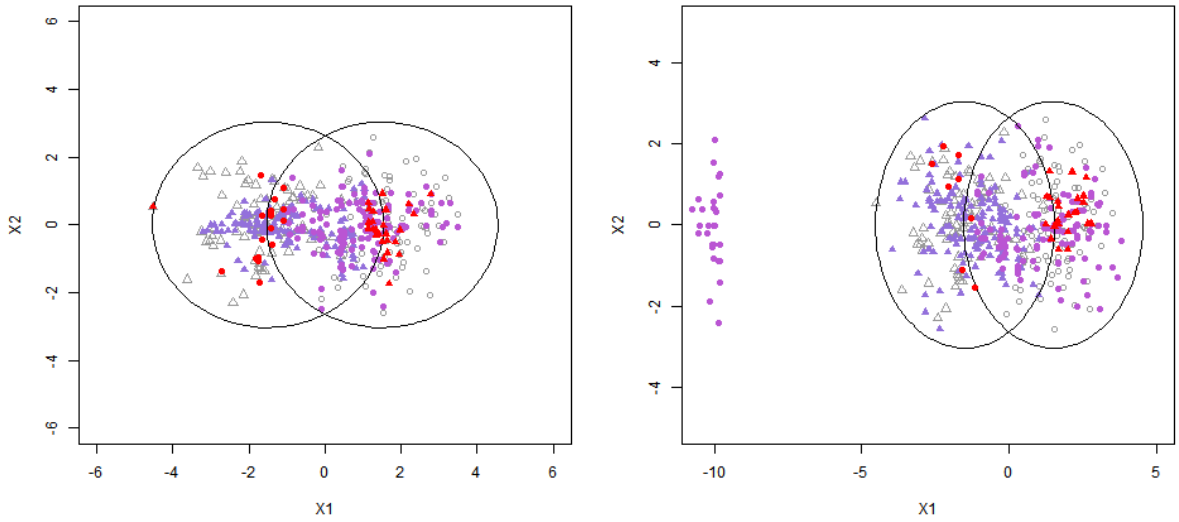
Table 6.6: Outlyingness in X_1 after imputation by k-NN median

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
Imp. methods	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
Mean Imp.	0.011	0.010	0.007	0.010	0.008	0.006	0.008	0.007	0.006	0.006	0.006	0.005
Median Imp.	0.011	0.010	0.007	0.010	0.008	0.006	0.008	0.007	0.006	0.006	0.006	0.005
K-NN mean	0.020	0.026	0.047	0.010	0.009	0.007	0.017	0.027	0.037	0.006	0.006	0.005
K-NN median	0.029	0.045	0.091	0.010	0.010	0.008	0.028	0.048	0.082	0.006	0.006	0.005
SEQimpute	0.011	0.010	0.008	0.010	0.009	0.008	0.008	0.007	0.006	0.010	0.010	0.010
ROBimpute	0.011	0.010	0.010	0.010	0.009	0.008	0.009	0.008	0.009	0.010	0.010	0.010

Table 6.7: Outlyingness in X_1 after imputation on the uncontaminated data

While those observations are not extreme outliers at all, this detection is very interesting in our particular context of classification. Indeed, the group change will have an impact on the classification results. For example, if we consider linear discrimination, those imputed instances are clearly on the wrong side of the discriminating line. A solution would be to take the population membership into account for the computation of the distances but, like with the mean and median imputation, this cannot be used for the test set. This is an indication that while certain imputation methods are good in a general imputation context, it might not be the case in particular context, like classification. However, what is observed here is also specific to our data and the choices that have been made for the theoretical framework.

As a matter of fact, the situation is not the same in the correlated setting. In fact, in this case, both variables X_1 and X_2 are discriminating. This implies that a similar situation could only happen when both variables are missing. This is not the case, however, because the correlation between the variables helps in the selection of the closest neighbours which reduces the risk of selecting neighbours belonging to the other population. Nevertheless, it remains that none of the



(a) Clean setting.

(b) Contaminated setting.

Figure 6.1: Illustration of group change induced by the k-NN median imputation. *The purple and violet symbols represent the imputation, the red symbols are the good observations detected by the outlyingness detection method. The two ellipses are the 99% tolerance ellipse of each population.*

contaminated instances which are missing is cleaned up so the outlyingness stays equal to the initial contamination proportion.

SEQimpute and ROBimpute As far as the SEQ- and ROBimpute methods are concerned, we observe a difference between the two. As expected, the SEQimpute method is not quite appropriate when the data contains atypical values and in particular in the independent setting. Indeed, we observe an increase of the outlyingness measures compared to the initial values before missingness introduction (table 6.8). Here, the increases are not due to *group changes*. Indeed, as seen in table 6.7, there is no increase in outlyingness in the uncontaminated setting. It therefore concerns both contaminated observations which are contaminated again and good observations which become contaminated. In the correlated setting, the results are close but sometimes slightly higher than the initial values. The difference in covariance structure explains the difference of results. As a matter of fact, the SEQimpute method is precisely a method which relies on the covariance matrix for the imputation as it tries to minimize the concentration of the data. It is thus expected to be *easier* in the case where there is correlation between the data. Also, we should remember that in the correlated case, the extreme outliers are placed slightly less far than in the independent case.

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.065	0.078	0.093	0.059	0.060	0.061	0.072	0.087	0.108	0.061	0.061	0.060
0.1	0.122	0.118	0.158	0.112	0.111	0.110	0.123	0.135	0.162	0.108	0.108	0.108
0.2	0.217	0.230	0.247	0.208	0.207	0.209	0.219	0.235	0.261	0.206	0.206	0.206

Table 6.8: Outlyingness in X_1 after imputation by SEQimpute

The ROBimpute method, on the other hand, is designed to be more robust to outliers. This is reflected in the results (see table 6.9) as there is no increase in outlyingness compared to the initial values (outlyingness before missingness introduction). Moreover, there is even a decrease in outlyingness as the proportion of missingness increases. The decrease being slightly more significant in the correlated case. This means that the ROBimpute method is able, in certain configurations, to clean up some of the initially contaminated missing values.

	Missingness											
	MCAR - Ind.			**MCAR - Corr.**			**MAR - Ind.**			**MAR - Corr.**		
ϵ	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
0.05	0.052	0.051	0.042	0.056	0.054	0.048	0.054	0.047	0.039	0.060	0.056	0.050
0.1	0.102	0.096	0.078	0.105	0.096	0.089	0.103	0.098	0.087	0.100	0.090	0.080
0.2	0.204	0.199	0.183	0.193	0.182	0.158	0.204	0.202	0.161	0.191	0.173	0.140

Table 6.9: Outlyingness in X_1 after imputation by ROBimpute

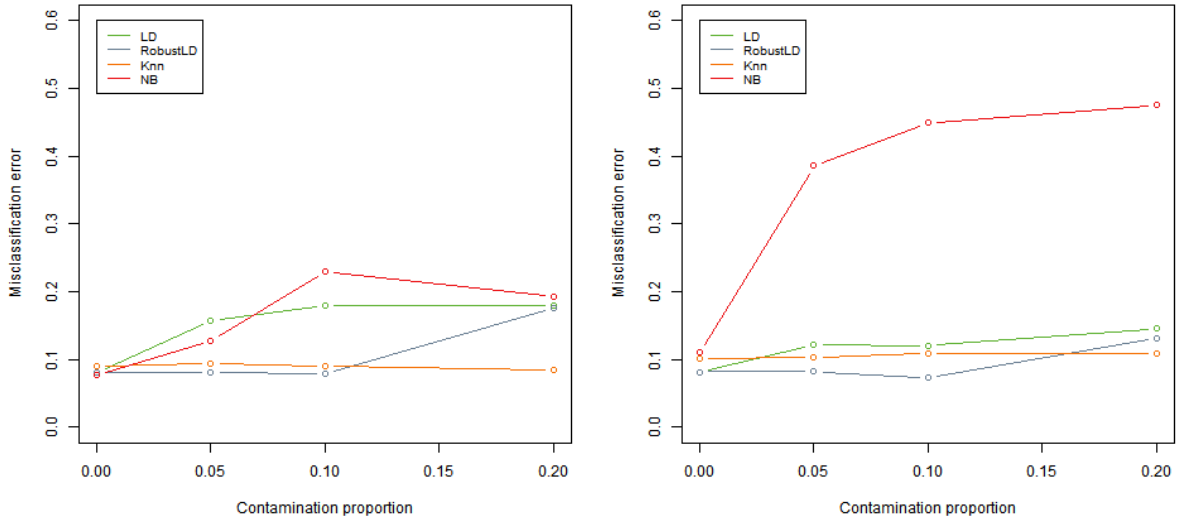
6.1.2 Classification results

Now that we have a better view of the situation after imputation, we can analyse the classification results, presented in tables (for the average misclassification errors) and figures (for the classification criterion). Let's begin with general observations made on all the simulation settings.

Our first baseline is the clean and complete data, in other words, the data without contamination and without missing values ($\epsilon = 0$ and $\pi = 0$). The mean misclassification rates for that data are close to what was expected with the theoretical computation (around 0.07 for all classifiers). The k-NN and Naive Bayes classifiers applied in the correlated setting produce a slightly higher misclassification error (around 0.1) (see columns 1 and 5 of table 6.10).

Our second baseline is the complete contaminated data ($\epsilon \neq 0$ and $\pi = 0$). This baseline enables us to detect the most robust techniques among the considered classifiers. Figure 6.2 and table 6.10 depict the misclassification errors for the different classifiers as a function of the contamination proportion. As long as the contamination proportion is not extreme, the robust linear discrimination handles well the outliers contrary to the classic linear discrimination. There is a jump in error when the contamination is equal to 0.2, especially in the independent case. A contamination of 20% of the data set size means that the population P_1 actually contains 40% of outliers. To determine the robust discriminating line, the means and covariance matrices of each population must be estimated. However, with a contamination of 40%, the estimation is expected to be difficult for the robust estimator which may explain the poorer results (illustration of the robust estimator breakdown can be found in appendix A.2). The k-NN classifier, on the other hand, performs well whatever the contamination level. In principle, this technique could be influenced by outliers. However, it proceeds following a local approach as the class label is determined based on a limited set of nearest instances. In our framework, the contaminated observations form a remote cluster and all belong to the same population. Therefore, the likelihood for an outlying observation to be used in the class determination is limited. Furthermore, one can clearly see that the Naive Bayes classifier really stands out particularly in the correlated setting. As will be seen in the following, globally the results obtained with Naive Bayes are the worst and especially in the correlated setting. This is not in line with one of the conclusions of Farhanghar et al. [11], however, in their study, they focused on clean discrete data. One possible explanation for our results is the fact that this method relies on the assumption that the features are independent which is clearly not the case

in that setting. The reason why this method does not seem to perform well in the presence of atypical values comes from the fact that the decision rule is based on the likelihood, which, in the case of Gaussian Naive Bayes, requires estimating means and standard deviations for the different variables. Those estimations are obviously affected by the outliers.



(a) Independent setting.

(b) Correlated setting.

Figure 6.2: Misclassification errors for the complete contaminated data

	Contamination							
	Independent set.				Correlated set.			
Classification methods	0.0	0.05	0.1	0.2	0.0	0.05	0.1	0.2
LD	0.078	0.156	0.179	0.180	0.077	0.122	0.120	0.145
Robust LD	0.077	0.080	0.079	0.176	0.078	0.082	0.072	0.130
K-NN	0.088	0.094	0.090	0.085	0.102	0.103	0.108	0.109
Naive Bayes	0.075	0.127	0.230	0.193	0.106	0.386	0.449	0.475

Table 6.10: Misclassification errors for the complete contaminated data

The third baseline is the clean incomplete data set ($\epsilon = 0$ and $\pi \neq 0$). Its results depend on the method used for imputation and are therefore presented in the following along with the results of the incomplete and contaminated data sets ($\epsilon \neq 0$ and $\pi \neq 0$) (see tables 6.11 and 6.12). It is important to remember that the classification results for all incomplete data sets presented here have been obtained using clean test sets containing missing values, as stated in the simulation set-ups. The test sets are imputed using the equivalent method as for the training sets. This has an impact on the classification results as the source of disturbance is twofold. First, the classification rule is estimated on imputed data sets, some of which being contaminated, and this rule is then applied on new imputed data sets. Simulations have been done on the same basis but with clean and complete test sets. Similar conclusions as those that will be presented in the following could be drawn with the main difference that the results were globally closer to the classification results of the first baseline. In particular, results for the third baseline were nearly equal to the results obtained for the first baseline which shows that the imputation methods succeed in overcoming the

loss of information induced by missing values. However, in the following, we decided to focus on the results of the *worst* case, i.e. incomplete testing sets.

To further analyse the classification results, we will differentiate the settings with respect to their covariance structure. Together with the classification error, we evaluate the classification criterion (4.4). We should remember that a small classification criterion δ_{err} does not mean that the classification result is the best but rather that the classification result after imputation is relatively close to the one of the clean and complete data set. This measure thus enables to assess in what extent the combination of imputation and classification methods succeeds in handling the perturbations induced by the missingness and contamination. The main features of the results are highlighted in the following.

6.1.2.1 Independent setting

The focus here is on the classification results obtained under MCAR in the independent setting. The misclassification errors for the different combinations of imputation method and classification method as a function of the contamination and missingness proportions are presented in tables 6.11 and 6.12. Similar tables for the MAR setting can be found in the appendix A.3 (tables A.6 and A.7).

At first sight, one can see that globally the results deteriorate as the contamination proportion and the missingness proportion increase. In addition, none of the combinations of techniques can do as well as the results of the first baseline. This clearly shows the adverse effect that the contamination and missingness can have on the data analysis and especially on classification.

As for the second baseline, the advantage of the robust linear discrimination over the classic technique remains true when missing values are involved, whatever imputation method is used. For those classification techniques, none of the imputation really stands out except for the median imputation which produces the best results for the highest contamination proportion when combined with the robust classification. The mean imputation, on the other hand, produces the worst results in that case with a difference of 0.1 with respect to the median imputation. Figure 6.3 represents the classification criterion as a function of the missingness proportion for the different imputation methods for the robust linear discrimination technique. One can clearly see, when the contamination is of 20% (fig.6.3d), the big difference between the mean and median imputation especially for high missingness proportion.

The k-NN classifier produces better results than the robust linear discrimination when contamination is at its highest. As far as the different imputation methods are concerned, the best results are obtained with the mean and median imputation. Again, the median performs at best for the highest contamination.

The results obtained with the Naive Bayes classifier are globally less good than the rest. For that classifier, the best imputation methods are the mean and median imputation. We observe particularly high values for the classification criterion when using k-NN mean imputation, k-NN median imputation and the SEQimpute method as represented on figure 6.4. A value of 5 for the classification criterion actually corresponds to the initial error rate multiplied by 6. This coincides with observations made for the outyingness. In fact, we observed an increase in outlyingness after imputation for those imputation methods. As we know, the Naive Bayes classifier is not really robust to outliers so that might be an explanation. The same is true when the increase in outlyingness is due to a *group change* as this can also disrupt the estimation of the distribution parameters.

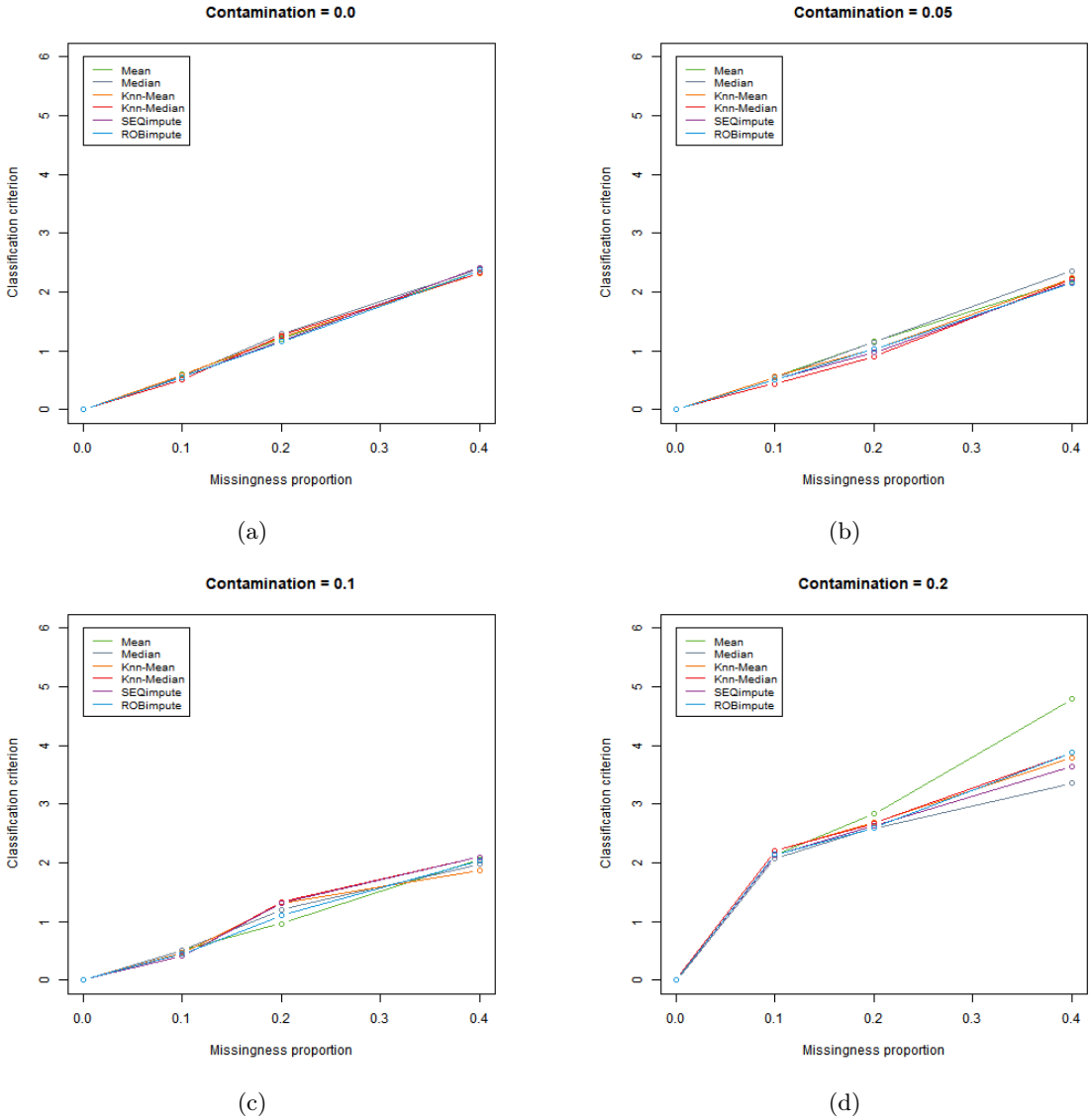
Overall, there is no fundamental difference between the two missingness mechanisms considered as can be noticed by looking at tables A.6 and A.7 in the appendix. All figures representing the classification criterion for the different classification methods in all settings can be found in appendix A.4.

		Missingness					
		LDA			**Robust LDA**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.073	0.074	0.078	0.072	0.070	0.076
	0.05	0.159	0.162	0.172	0.077	0.077	0.094
	0.1	0.195	0.164	0.248	0.068	0.070	0.098
	0.2	0.184	0.184	0.180	0.179	0.179	0.174
Mean Imputation	0.0	0.119	0.162	0.244	0.120	0.163	0.245
	0.05	0.184	0.209	0.263	0.119	0.168	0.247
	0.1	0.232	0.208	0.277	0.120	0.156	0.242
	0.2	0.228	0.279	0.398	0.208	0.253	0.381
Median Imputation	0.0	0.120	0.172	0.250	0.118	0.169	0.249
	0.05	0.180	0.213	0.263	0.119	0.167	0.259
	0.1	0.218	0.213	0.268	0.120	0.174	0.236
	0.2	0.242	0.284	0.351	0.204	0.236	0.284
K-NN Algorithm - Mean	0.0	0.116	0.167	0.238	0.119	0.166	0.243
	0.05	0.190	0.226	0.284	0.122	0.158	0.247
	0.1	0.224	0.219	0.294	0.117	0.182	0.227
	0.2	0.216	0.247	0.321	0.212	0.243	0.315
K-NN Algorithm - Median	0.0	0.112	0.166	0.242	0.112	0.169	0.243
	0.05	0.186	0.225	0.288	0.112	0.149	0.242
	0.1	0.224	0.232	0.294	0.112	0.184	0.246
	0.2	0.216	0.248	0.329	0.213	0.243	0.319
SEQimpute	0.0	0.116	0.163	0.246	0.116	0.160	0.251
	0.05	0.192	0.228	0.278	0.117	0.152	0.242
	0.1	0.232	0.227	0.287	0.112	0.184	0.246
	0.2	0.211	0.245	0.309	0.208	0.239	0.305
ROBimpute	0.0	0.116	0.159	0.243	0.116	0.160	0.248
	0.05	0.185	0.218	0.275	0.117	0.155	0.241
	0.1	0.214	0.210	0.287	0.115	0.166	0.241
	0.2	0.233	0.281	0.362	0.208	0.237	0.319

Table 6.11: Misclassification errors - MCAR in the independent setting (part I)

		Missingness					
		k-NN			NB		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.087	0.080	0.092	0.071	0.072	0.082
	0.05	0.089	0.093	0.110	0.122	0.126	0.132
	0.1	0.086	0.078	0.123	0.214	0.182	0.242
	0.2	0.087	0.092	0.092	0.191	0.204	0.183
Mean Imputation	0.0	0.132	0.169	0.247	0.113	0.165	0.241
	0.05	0.130	0.167	0.244	0.151	0.195	0.265
	0.1	0.135	0.168	0.263	0.237	0.227	0.287
	0.2	0.134	0.189	0.280	0.205	0.222	0.263
Median Imputation	0.0	0.131	0.171	0.247	0.115	0.165	0.239
	0.05	0.136	0.173	0.250	0.151	0.197	0.265
	0.1	0.125	0.175	0.264	0.232	0.227	0.293
	0.2	0.122	0.169	0.255	0.203	0.217	0.265
K-NN Algorithm - Mean	0.0	0.138	0.195	0.288	0.117	0.165	0.245
	0.05	0.135	0.191	0.269	0.181	0.251	0.354
	0.1	0.145	0.211	0.308	0.274	0.294	0.437
	0.2	0.135	0.189	0.281	0.263	0.312	0.384
K-NN Algorithm - Median	0.0	0.138	0.195	0.298	0.113	0.169	0.246
	0.05	0.128	0.180	0.275	0.188	0.265	0.365
	0.1	0.144	0.193	0.267	0.288	0.313	0.447
	0.2	0.133	0.191	0.291	0.268	0.318	0.393
SEQimpute	0.0	0.138	0.188	0.267	0.114	0.166	0.248
	0.05	0.141	0.186	0.272	0.184	0.238	0.342
	0.1	0.140	0.190	0.292	0.278	0.302	0.436
	0.2	0.133	0.190	0.276	0.262	0.311	0.378
ROBimpute	0.0	0.140	0.188	0.271	0.110	0.160	0.242
	0.05	0.140	0.195	0.259	0.175	0.228	0.331
	0.1	0.136	0.194	0.282	0.281	0.287	0.392
	0.2	0.134	0.191	0.269	0.244	0.281	0.331

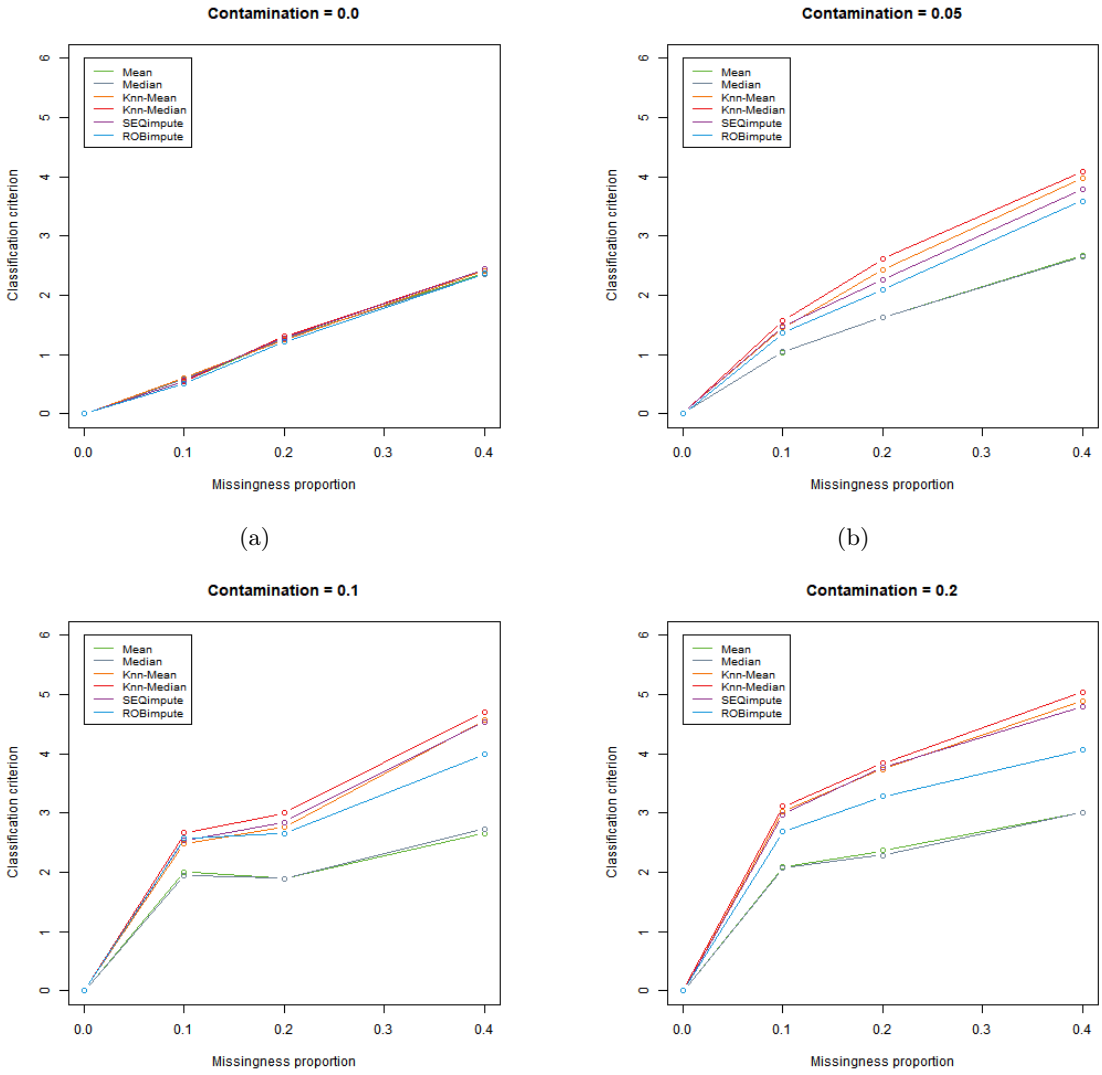
Table 6.12: Misclassification errors - MCAR in the independent setting (part II)



(a) (b)

(c) (d)

Figure 6.3: Classification criterion for the Robust Linear Discrimination in the independent setting, under MCAR.



(a) (b)

(c) (d)

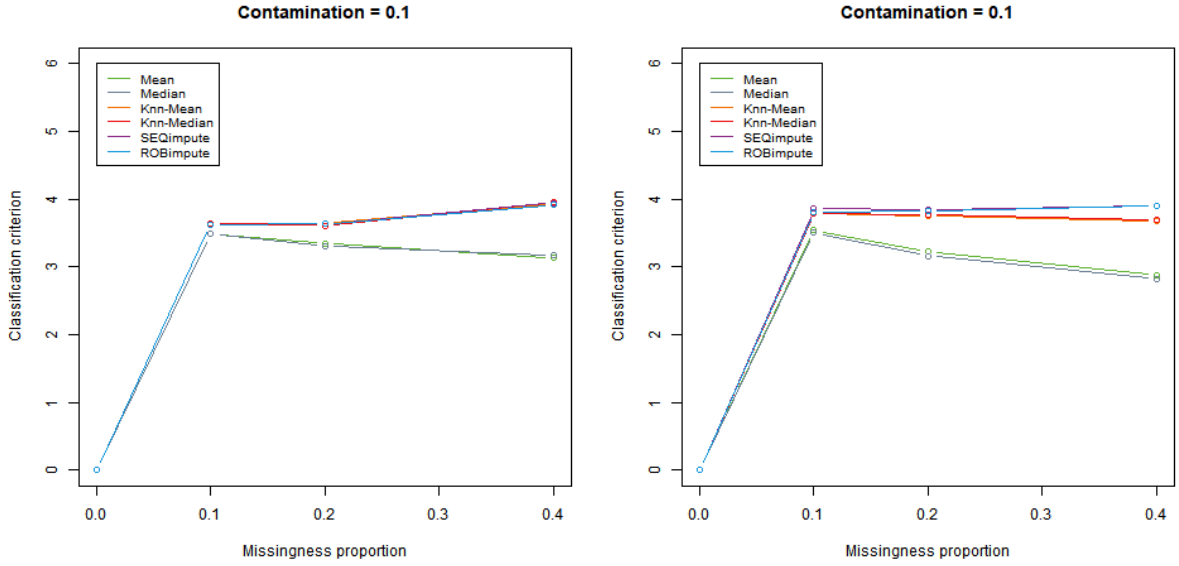
Figure 6.4: Classification criterion for the Naive Bayes classifier in the independent setting, under MCAR.

6.1.2.2 Correlated setting

In the correlated setting (tables A.4, A.5 and A.8, A.9), one can make the same observation, as in the previous setting, about the increase of the error as a function of the contamination and missingness except for one classifier which will be detailed in the following. In addition, results are better with the robust linear discrimination except for high contamination where k-NN classifier performs better.

Regarding the imputation methods, there is a dominance of the SEQimpute and ROBimpute methods. The ROBimpute method is particularly successful for the highest contamination when combined with the most robust classification techniques. As already mentioned, those imputation methods seem to be more efficient when variables are correlated. The mean imputation, on the other hand, globally produces the worst results.

As far as the Naive Bayes classifier is concerned, the results are globally the worst of the whole simulations. The misclassification error is close to 0.5, especially for high contamination, which is equivalent to throwing a coin. However, what is interesting in this case is that with the mean and median imputation, the error decreases with the missingness proportion increasing. This means that having missing values actually improves the results compared to the contaminated data without missing values. This is not the case with the other imputation methods but, here, we observe a difference between the two missingness mechanisms. As can be seen on figure 6.5a and 6.5b, the classification criterion increase for the remaining imputation methods in the MCAR case while,in the MAR case, it remains similar for the SEQ- and ROBimpute methods and it slightly decreases with the k-NN methods. However, as there is no real difference of that type with the other classification methods, it is difficult to draw particular conclusions on that level.



(a) MCAR mechanism.

(b) MAR mechanism.

Figure 6.5: Classification criterion for the Naive Bayes classifier in the correlated setting.

6.2 Conclusion on the simulations

Overall, the whole set of simulation results demonstrated the adverse impact of missingness and contamination on classification results. The most detrimental to the results is obviously the combination of missingness and contamination in the training sets and incomplete test sets. Indeed, in that case, the disturbance for the results is twofold. First, the classification rule is estimated on an imputed data set under influence of contamination and then this rule is applied on new imputed data. The baselines have shown that when the contamination and missingness occur separately, certain methods considered succeed in handling the perturbation. When both contamination and missingness occur simultaneously, combinations of imputation and classification methods succeed in mitigating the effect on the classification results although none of them is able to completely resolve the problems.

More precisely, the main differences are noticeable between the covariance structure of the data rather than the missingness mechanism considered. Regarding the imputation methods especially, we can summarise the observations by saying that in the independent setting, the mean and median imputations are giving the best results. The latter performing best when the contamination is at its highest. In the correlated setting, on the other hand, the best results are obtained with the SEQimpute and ROBimpute methods, the ROBimpute method performing best when the contamination is high. From this, one can already draw a conclusion as to the fact that robust imputation methods seem to better manage the effect that the contaminated values have on the imputation. In that respect, the observations made on the outlyingness measures show that those robust imputation methods are also the one able to clean up some of the missing values originally contaminated. In addition, the characteristics of the data seem to influence the choice of the most appropriate imputation methods. Indeed, in a setting where there is a particular link between the data, imputation methods which use that link for the imputation are favoured.

As far as the classification technique is concerned, one can make a similar conclusion about the robustness. In fact, results show that more robust techniques perform better. However, the choice of the ultimate technique will depend on the level of contamination.

Similarly to other studies which have been presented in Chapter 3, we have not been able to identify one specific combination of methods working best in all cases. Nevertheless, except from certain combinations which have been identified in the observations, all results are globally similar. One should always take into consideration the characteristics of the data and the context in order to choose which methods should be favoured.

6.3 Contaminated cases imputation

As presented in our theoretical framework, being in a normal setting, the contamination that is introduced in the data is really a disturbance to the main data group. The effect of the contamination was clearly visible in the simulations. This has led us to reflect on the fact that we could treat these atypical values the same way we treat missing values. It means that we could replace the identified contaminated values by NA's and then impute them using the different imputation methods. This is a way to overcome the adverse effect that contamination can have on the classification results. In a real-life case, it is obviously necessary to ensure that the contaminated instances are not of real interest and may be considered truly as gross errors.

In practice, setting all contaminated values as missing is a kind of not missing at random mecha-

nism (NMAR). Indeed, the missing values will depend on the values of the variables in which the missingness is introduced. In our case, the smallest values of X_1 and the biggest values of X_5 are replaced by missing values.

Results of simulations in that setting, restricted to misclassification errors, are presented in table 6.13, the first rows (no treatment) depicting the results on the contaminated data. As expected, for non-robust classification methods, imputing the contaminated values clearly improves the classification results compared to the results obtained on the contaminated data. The results are close to the classification results of the clean and complete data set (columns 1 and 5 of table 6.10). For the robust linear discrimination, results after imputation are as good as the results on the contaminated data and even better for the highest contamination. We observe, however, an increase of the error in certain configurations for the k-NN classifier. Overall, the conclusion can be drawn that, in our particular context, imputing the contaminated values improves the classification results compared to the results on contaminated data (first rows of table 6.13). In addition, this shows that the imputation methods globally succeed in coping with the loss of information induced by missing values.

		Classifier							
		Independent set.				**Correlated set.**			
Imputation meth.	ϵ	LDA	RLDA	k-NN	NB	LDA	RLDA	k-NN	NB
No treatment	0.05	0.155	0.068	0.080	0.125	0.121	0.074	0.119	0.383
	0.1	0.167	0.074	0.084	0.148	0.122	0.079	0.103	0.440
	0.2	0.174	0.173	0.083	0.169	0.129	0.111	0.105	0.476
Mean Imputation	0.05	0.068	0.068	0.079	0.068	0.082	0.081	0.119	0.117
	0.1	0.073	0.074	0.084	0.075	0.084	0.084	0.105	0.126
	0.2	0.068	0.069	0.082	0.068	0.085	0.086	0.103	0.140
Median Imputation	0.05	0.068	0.068	0.079	0.068	0.082	0.080	0.119	0.119
	0.1	0.073	0.074	0.085	0.075	0.084	0.083	0.105	0.125
	0.2	0.069	0.070	0.081	0.066	0.087	0.087	0.104	0.136
K-NN - Mean	0.05	0.068	0.069	0.086	0.067	0.080	0.077	0.124	0.112
	0.1	0.074	0.076	0.102	0.074	0.086	0.081	0.105	0.122
	0.2	0.092	0.096	0.126	0.088	0.086	0.082	0.126	0.098
K-NN - Median	0.05	0.068	0.069	0.085	0.068	0.079	0.077	0.125	0.110
	0.1	0.074	0.075	0.100	0.074	0.085	0.081	0.108	0.121
	0.2	0.096	0.098	0.131	0.088	0.086	0.081	0.123	0.098
SEQimpute	0.05	0.068	0.069	0.086	0.067	0.081	0.075	0.121	0.113
	0.1	0.075	0.076	0.101	0.073	0.087	0.081	0.107	0.121
	0.2	0.092	0.102	0.136	0.088	0.087	0.091	0.129	0.107
ROBimpute	0.05	0.068	0.069	0.086	0.068	0.080	0.076	0.121	0.113
	0.1	0.074	0.076	0.106	0.073	0.087	0.082	0.107	0.121
	0.2	0.092	0.106	0.139	0.089	0.085	0.091	0.126	0.108

Table 6.13: Misclassification errors - Imputation of contaminated observations

Chapter 7

Conclusion

The data quality is a central concern in data science and missing values are an important part of it. Missing data may be present in a data set according to different patterns and mechanisms. It is thus important to analyse their characteristics and try to understand the underlying reasons behind the missingness. The issue then is to determine the best way of handling the missing values. Different treatment approaches can be considered. The simplest way consists in deleting the cases containing missing values, the main drawback being the loss of information and the loss of statistical efficiency. The second type of treatment methods is imputation which consists in filling in the missing values with estimations. One can follow two different approaches for the estimation: either statistical or machine learning procedures. Another type of treatment relies on the estimation of model parameters for the complete data. We focused on the two first types of methods.

The literature about missing values is very broad. The main conclusions we were able to draw from our literature overview are as follows. Most imputation methods generally help in coping with the loss of information induced by missing values yet no universal method could be identified. Although comparing the techniques on real data sets serves the reality purpose, evaluation in simulated settings enables to have a complete control over the characteristics of the data sets and in particular on the scale and type of missing data which can enhance the comparison. Moreover, when the classification result is regarded as the comparison metrics, missingness in the testing subset should definitely be considered as new instances to be classified are even likely to contain missing values. However, this has not been taken into account in most studies. Those findings guided us in the continuation of our work and particularly for the simulations.

The treatment of missing values will greatly depend on the context. In some real-life scenarios where interpretation is key, imputation or other estimation procedures may not be appropriate. In the context of classification, missing values can lead to serious problems and in particular when occurring in the testing set. The case deletion technique is not recommended as it results in unclassified instances. In addition, doing nothing would imply omitting missing information that could play a role in the prediction and thus produce biased prediction. The loss of information from an incomplete training set is also harmful. This is the reason why missing data should definitely be handled with the main objective of improving the classification performances. One idea proposed by some software (e.g. SAS) is to add, to the training data, a binary indicator reflecting the missingness imputation so that the model can adjust the predictions in the event the missingness is related to the target itself. In real-life scenarios, an important aspect may be the time taken for imputation. Indeed, in some classification cases, the timing for obtaining the classification results must be short.

Another point which matters is whether the classification happens offline or online, in other words if one has all the data at one time or one observation at a time. This will influence the choice and execution of the treatment procedure.

We decided to take a further step in addressing data quality issues by considering classification data containing atypical values in addition to missing values. The objective was to see whether the combined effect of missingness and contamination on the classification performances could be reduced thanks to a combination of imputation and classification methods. For this purpose, we performed simulations within a well-defined theoretical framework. The definition of a precise generation model enabled us to have full control over the data characteristics and thus to better understand the different possible situations. Data sets have been built according to two different covariance structures and missing values have been introduced following two different mechanisms. Both training and testing sets were affected by missing values. Several imputation methods have been investigated including statistical and machine learning methods. Also, several classification methods have been considered. As we were working in a contaminated context, we examined, for both the imputation and classification steps, both classic and robust techniques. The definition of such a precise simulation framework allowed us to analyse in detail the different procedures in order to better understand how they operate. Moreover, the prior analysis of the different possible situations enabled us to better explain and make sense of the results.

It was expected to find imputation methods capable of replacing by clean observations certain contaminated ones that were missing. For this reason, we monitored the contamination level by means of an outlyingness measure at the different stages of the simulation process. This has indeed enabled us to identify such methods. There is no particular imputation method standing out from the other, nor is there a specific combination of imputation and classification methods that works best in all cases. However, the main conclusion we can draw is that, in context of contamination, both for imputation and classification, robust methods should be favoured. Indeed, combinations of robust methods seem to better manage the adverse effect of contamination and missingness. The characteristics of the data also influence the performances of the different methods. For that reason, one should always study the structure of the data, its context, etc in order to choose the most appropriate methods.

Future work

In our simulations, we have tried to integrate different characteristics of a possible data set. We had to limit the choice of methods to investigate, the proportion ranges, the number of different settings,... To broaden the scope of the simulations, it could be interesting to investigate unbalanced data sets, other correlation structures (e.g. $\sigma_{ij}^2 = \rho^{|i-j|}$), other contamination positions or types, different data configurations, etc. Also, tests on real classification data sets could be carried out in order to evaluate the performance of the different methods in real-world situations. While it was not the primary goal of this work, it might be of interest to evaluate the ability of the imputation methods to reconstruct the original data sets. The advantage of working with simulated data is that we have all necessary information to compare the different results with the original values. Furthermore, it could be interesting to conduct a similar study in the context of regression where y is no longer a membership label but rather a continuous value.

Bibliography

[1] E. Acuna and C. Rodriguez. "The treatment of missing values and its effect in the classifier accuracy, classification, clustering, and data mining applications". In: *Proceedings of the Meeting of the International Federation of Classification Societies (IFCS)*. 2004, pp. 639–647.

[2] R.R. Andridge and R.JA. Little. "A review of hot deck imputation for survey non-response". In: *International statistical review* 78.1 (2010), pp. 40–64.

[3] J. Barnard and XL. Meng. "Applications of multiple imputation in medical studies: from AIDS to NHANES". In: *Statistical methods in medical research* 8.1 (1999), pp. 17–36.

[4] GE. Batista and MC. Monard. "An analysis of four missing data treatment methods for supervised learning". In: *Applied artificial intelligence* 17.5-6 (2003), pp. 519–533.

[5] L. Beretta and A. Santaniello. "Nearest neighbor imputation algorithms: a critical evaluation". In: *BMC medical informatics and decision making* 16.3 (2016), p. 74.

[6] M.L. Brown and J.F. Kros. "Data mining and the impact of missing data". In: *Industrial Management & Data Systems* (2003).

[7] A. Choudhury and M.R. Kosorok. "Missing Data Imputation for Classification Problems". In: *arXiv preprint arXiv:2002.10709* (2020).

[8] K.J. Cios and L.A. Kurgan. "Trends in data mining and knowledge discovery". In: *Advanced techniques in knowledge discovery and data mining*. Springer, 2005, pp. 1–26.

[9] C. Croux, G. Haesbroeck, and K. Joossens. "Logistic discrimination using robust estimators: an influence function approach". In: *Canadian Journal of Statistics* 36.1 (2008), pp. 157–174.

[10] C.K. Enders. *Applied missing data analysis*. Guilford press, 2010.

[11] A. Farhangfar, L. Kurgan, and J. Dy. "Impact of imputation of missing values on classification error for discrete data". In: *Pattern Recognition* 41.12 (2008), pp. 3692–3705.

[12] U. Garciarena and R. Santana. "An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers". In: *Expert Systems with Applications* 89 (2017), pp. 52–65.

[13] P.J. Garcıa-Laencina, JL. Sancho-Gómez, and A.R. Figueiras-Vidal. "Pattern classification with missing data: a review". In: *Neural Computing and Applications* 19.2 (2010), pp. 263–282.

[14] MW. Huang, WC. Lin, and CF. Tsai. "Outlier removal in model-based missing value imputation for medical datasets". In: *Journal of healthcare engineering* 2018 (2018).

[15] M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. "ROBPCA: a new approach to robust principal component analysis". In: *Technometrics* 47.1 (2005), pp. 64–79.

[16] Mia Hubert and Katrien Van Driessen. "Fast and robust discriminant analysis". In: *Computational Statistics & Data Analysis* 45.2 (2004), pp. 301–320.

[17] J.M. Jerez, I. Molina, P.J. Garcıa-Laencina, E. Alba, N. Ribelles, M. Martın, and L. Franco. "Missing data imputation using statistical and machine learning methods in a real breast cancer problem". In: *Artificial intelligence in medicine* 50.2 (2010), pp. 105–115.

[18] P. Jonsson and C. Wohlin. "An evaluation of k-nearest neighbour imputation using likert data". In: *10th International Symposium on Software Metrics, 2004. Proceedings.* IEEE. 2004, pp. 108–118.

[19] WC. Lin and CF. Tsai. "Missing value imputation: a review and analysis of the literature (2006–2017)". In: *Artificial Intelligence Review* 53.2 (2020), pp. 1487–1509.

[20] R.JA. Little. "Regression with missing X's: a review". In: *Journal of the American Statistical Association* 87.420 (1992), pp. 1227–1237.

[21] Y. Liu and S. D. Brown. "Comparison of five iterative imputation methods for multivariate classification". In: *Chemometrics and Intelligent Laboratory Systems* 120 (2013), pp. 106–115.

[22] J. Luengo, S. Garcıa, and F. Herrera. "On the choice of the best imputation methods for missing values considering three groups of classification methods". In: *Knowledge and information systems* 32.1 (2012), pp. 77–108.

[23] L. Peng and L. Lei. "A review of missing data treatment methods". In: *Intelligent Information Management Systems and Technologies* 1.3 (2005), pp. 412–419.

[24] MG. Rahman and MZ. Islam. "Data quality improvement by imputation of missing values". In: *International Conference on Computer Science and Information Technology (CSIT-2013), Yogyakarta, Indonesia.* 2013, pp. 82–88.

[25] S. Rawal, SC. Gupta, and S. Singh. "Predicting missing values in a dataset: challenges and approaches". In: *International Journal of Recent Research Aspects* 4.3 (2017), pp. 34–38.

[26] M. Saar-Tsechansky and F. Provost. "Handling missing values when applying classification models". In: *Journal of machine learning research* 8.Jul (2007), pp. 1623–1657.

[27] M. Salibian-Barrera and V. J Yohai. "A fast algorithm for S-regression estimates". In: *Journal of computational and Graphical Statistics* 15.2 (2006), pp. 414–427.

[28] JL. Schafer and JW. Graham. "Missing data: our view of the state of the art." In: *Psychological methods* 7.2 (2002), pp. 147–177.

[29] O. Toka and M. Çetin. "Imputation and Deletion Methods Under The Presence of Missing Values and Outliers: A Comparative Study." In: *Gazi University Journal of Science* 29.4 (2016).

[30] C.T. Tran, M. Zhang, P. Andreae, B. Xue, and L.T. Bui. "Improving performance of classification on incomplete data using feature selection and clustering". In: *Applied Soft Computing* 73 (2018), pp. 848–861.

[31] CF. Tsai and FY. Chang. "Combining instance selection for better missing value imputation". In: *Journal of Systems and Software* 122 (2016), pp. 63–71.

[32] K. Vanden Branden and S. Verboven. "Robust data imputation". In: *Computational Biology and Chemistry* 33.1 (2009), pp. 7–13.

[33] S. Verboven, K. Vanden Branden, and P. Goos. "Sequential imputation for missing values". In: *Computational Biology and Chemistry* 31.5-6 (2007), pp. 320–327.

[34] I.R. White, P. Royston, and A.M. Wood. "Multiple imputation using chained equations: issues and guidance for practice". In: *Statistics in medicine* 30.4 (2011), pp. 377–399.

[35] Y. Zhang, C. Kambhampati, D.N. Davis, K. Goode, and J. GF. Cleland. "A comparative study of missing value imputation with multiclass classification for clinical heart failure data". In: *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery.* IEEE. 2012, pp. 2840–2844.

Appendix

A.1 Outlyingness after missingness

ϵ \ π	0.1	0.2	0.4
0.05	0.0526	0.0482	0.0350
0.1	0.0991	0.0894	0.0681
0.2	0.1865	0.1661	0.1233

Table A.1: Outlyingness in X_1 after missingness introduction - MCAR in correlated setting

ϵ \ π	0.1	0.2	0.4
0.05	0.0524	0.0456	0.0345
0.1	0.0981	0.0881	0.0664
0.2	0.1873	0.1644	0.1299

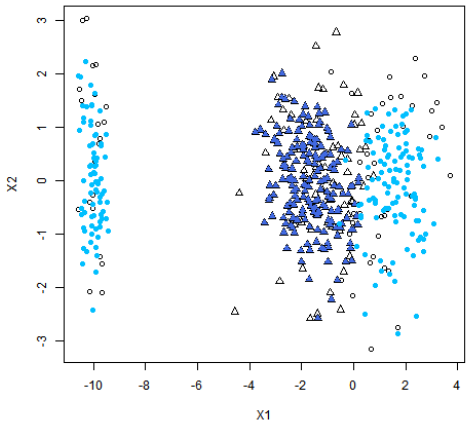
Table A.2: Outlyingness in X_1 after missingness introduction - MAR in independent setting

ϵ \ π	0.1	0.2	0.4
0.05	0.0522	0.0463	0.0351
0.1	0.0956	0.0828	0.0650
0.2	0.1838	0.1642	0.1232

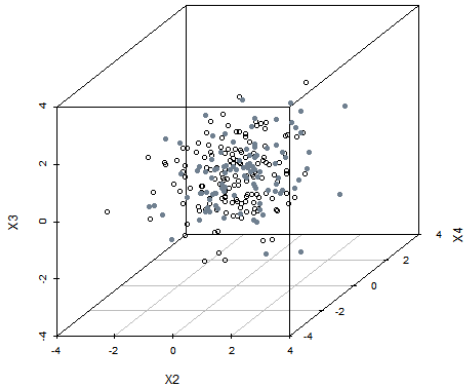
Table A.3: Outlyingness in X_1 after missingness introduction - MAR in correlated setting

A.2 Robustness breakdown

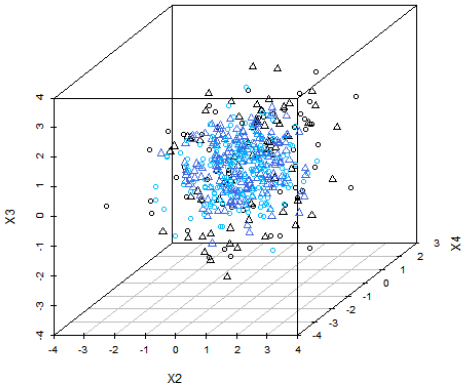
Figure A.1 illustrates what is happening with the robust estimator when the contamination proportion is 0.2. Remember that a contamination of 20% of the data set size means that the population P_1 actually contains 40% of outliers. As represented on figures A.1a and A.1d, part of the outlying values are included in the optimal subset used for the estimation of the parameters of population P_1. Figure A.1b shows the population P_1 with the outliers highlighted in grey. One can notice that on the three dimensions (X_2, X_3,X_4) which are uncontaminated, the outliers are indistinguishable from the good observations (fig. A.1b and A.1c).



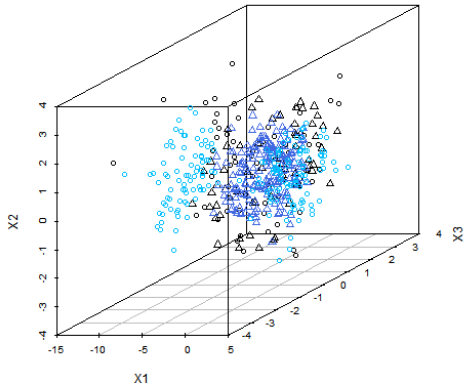
(a) 2D representation of the two populations with the optimal subsets respectively used for the estimations by the MCD estimator.



(b) 3D representation of the population P_1. The grey points depict the outliers.



(c) 3D representation of the two populations with the optimal subsets respectively used for the estimations by the MCD estimator (according to X_2, X_3,X_4).



(d) 3D representation of the two populations with the optimal subsets respectively used for the estimations by the MCD estimator (according to X_1, X_2,X_3).

Figure A.1: Illustration of the robust estimator failure in case of high contamination.

A.3 Classification results

		Missingness					
		LDA			Robust LDA		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.0784	0.0770	0.0786	0.0801	0.0785	0.0829
	0.05	0.1216	0.1236	0.1246	0.0813	0.0815	0.0804
	0.1	0.1288	0.1167	0.1421	0.0775	0.0752	0.0815
	0.2	0.1459	0.1505	0.1489	0.1279	0.1334	0.1296
Mean Imputation	0.0	0.1277	0.1672	0.2610	0.1297	0.1671	0.2637
	0.05	0.1657	0.2087	0.3019	0.1301	0.1796	0.2775
	0.1	0.1646	0.2162	0.3199	0.1391	0.1948	0.2999
	0.2	0.1966	0.2610	0.3598	0.1691	0.2379	0.3510
Median Imputation	0.0	0.1277	0.1675	0.2643	0.1307	0.1699	0.2675
	0.05	0.1649	0.2054	0.2915	0.1292	0.1744	0.2595
	0.1	0.1681	0.2068	0.3126	0.1311	0.1653	0.2833
	0.2	0.1948	0.2504	0.3288	0.1665	0.2108	0.2833
K-NN Algorithm - Mean	0.0	0.1100	0.1398	0.2154	0.1112	0.1421	0.2172
	0.05	0.1466	0.1780	0.2429	0.1113	0.1454	0.2149
	0.1	0.1553	0.1765	0.2657	0.1130	0.1226	0.2106
	0.2	0.1708	0.2026	0.2656	0.1540	0.1882	0.2540
K-NN Algorithm - Median	0.0	0.1095	0.1405	0.2203	0.1107	0.1429	0.2215
	0.05	0.1481	0.1790	0.2474	0.1125	0.1476	0.2226
	0.1	0.1461	0.1715	0.2570	0.1062	0.1293	0.2239
	0.2	0.1714	0.2024	0.2661	0.1544	0.1865	0.2524
SEQimpute	0.0	0.1027	0.1282	0.2027	0.1026	0.1294	0.2034
	0.05	0.1419	0.1674	0.2292	0.1081	0.1401	0.2077
	0.1	0.1466	0.1629	0.2288	0.0971	0.1134	0.2142
	0.2	0.1656	0.1938	0.2450	0.1505	0.1812	0.2335
ROBimpute	0.0	0.1037	0.1295	0.2046	0.1036	0.1293	0.2078
	0.05	0.1542	0.1863	0.2618	0.1077	0.1396	0.2029
	0.1	0.1638	0.1870	0.2990	0.0993	0.1187	0.2132
	0.2	0.2717	0.3714	0.4570	0.1480	0.1721	0.2160

Table A.4: Misclassification errors - MCAR in the correlated setting (part I)

		Missingness					
		k-NN			**NB**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.1021	0.1010	0.1176	0.1124	0.1264	0.1467
	0.05	0.1023	0.1056	0.1115	0.3796	0.3913	0.3808
	0.1	0.1101	0.0963	0.1341	0.4380	0.4395	0.4583
	0.2	0.1106	0.1182	0.1341	0.4721	0.4755	0.4715
Mean Imputation	0.0	0.1449	0.1872	0.2786	0.1459	0.1817	0.2662
	0.05	0.1510	0.1975	0.2819	0.3759	0.3740	0.3714
	0.1	0.1686	0.1901	0.3026	0.4441	0.4297	0.4126
	0.2	0.1569	0.2044	0.2979	0.4692	0.4644	0.4589
Median Imputation	0.0	0.1469	0.1893	0.2773	0.1467	0.1843	0.2718
	0.05	0.1494	0.1933	0.2767	0.3750	0.3724	0.3699
	0.1	0.1567	0.1857	0.2822	0.4440	0.4254	0.4153
	0.2	0.1562	0.1994	0.2912	0.4666	0.4612	0.4475
K-NN Algorithm - Mean	0.0	0.1359	0.1730	0.2651	0.1370	0.1663	0.2464
	0.05	0.1368	0.1770	0.2611	0.4067	0.4297	0.4612
	0.1	0.1433	0.1653	0.2693	0.4591	0.4597	0.4881
	0.2	0.1424	0.1832	0.2680	0.4779	0.4836	0.4918
K-NN Algorithm - Median	0.0	0.1352	0.1712	0.2691	0.1376	0.1675	0.2491
	0.05	0.1383	0.1786	0.2681	0.4061	0.4293	0.4620
	0.1	0.1380	0.1663	0.2765	0.4591	0.4560	0.4898
	0.2	0.1444	0.1835	0.2690	0.4778	0.4833	0.4889
SEQimpute	0.0	0.1291	0.1615	0.2488	0.1323	0.1603	0.2313
	0.05	0.1340	0.1694	0.2522	0.4084	0.4296	0.4556
	0.1	0.1293	0.1704	0.2411	0.4582	0.4587	0.4886
	0.2	0.1429	0.1767	0.2581	0.4773	0.4819	0.4870
ROBimpute	0.0	0.1298	0.1637	0.2508	0.1328	0.1623	0.2282
	0.05	0.1337	0.1684	0.2411	0.4076	0.4257	0.4523
	0.1	0.1308	0.1589	0.2544	0.4578	0.4575	0.4878
	0.2	0.1389	0.1750	0.2485	0.4734	0.4782	0.4843

Table A.5: Misclassification errors - MCAR in the correlated setting (part II)

		Missingness					
		LDA			**Robust LDA**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.0662	0.0639	0.0685	0.0657	0.0635	0.0708
	0.05	0.1544	0.1514	0.1519	0.0661	0.0632	0.0665
	0.1	0.1710	0.1657	0.1616	0.0720	0.0738	0.0725
	0.2	0.1811	0.1820	0.2023	0.1811	0.1836	0.1927
Mean Imputation	0.0	0.1092	0.1615	0.2570	0.1097	0.1605	0.2549
	0.05	0.1830	0.2145	0.2804	0.1149	0.1570	0.2447
	0.1	0.2066	0.2365	0.3157	0.1187	0.1610	0.2550
	0.2	0.2344	0.3111	0.4116	0.2064	0.2733	0.3816
Median Imputation	0.0	0.1111	0.1634	0.2565	0.1109	0.1625	0.2555
	0.05	0.1830	0.2110	0.2698	0.1149	0.1590	0.2473
	0.1	0.2050	0.2296	0.2978	0.1178	0.1625	0.2581
	0.2	0.2401	0.3034	0.3604	0.2101	0.2541	0.2893
K-NN Algorithm - Mean	0.0	0.1075	0.1562	0.2505	0.1073	0.1567	0.2512
	0.05	0.1927	0.2298	0.2986	0.1133	0.1596	0.2396
	0.1	0.2088	0.2440	0.3200	0.1202	0.1553	0.2541
	0.2	0.2093	0.2569	0.3382	0.2143	0.2584	0.3386
K-NN Algorithm - Median	0.0	0.1076	0.1524	0.2465	0.1071	0.1540	0.2503
	0.05	0.1921	0.2272	0.3024	0.1144	0.1573	0.2419
	0.1	0.2078	0.2437	0.3199	0.1225	0.1568	0.2545
	0.2	0.2098	0.2509	0.3460	0.2142	0.2497	0.3353
SEQimpute	0.0	0.1111	0.1614	0.2407	0.1103	0.1582	0.2424
	0.05	0.1920	0.2258	0.2921	0.1125	0.1625	0.2561
	0.1	0.2087	0.2395	0.3125	0.1197	0.1642	0.2621
	0.2	0.2136	0.2555	0.3230	0.2140	0.2538	0.3235
ROBimpute	0.0	0.1117	0.1574	0.2496	0.1106	0.1594	0.2502
	0.05	0.1890	0.2173	0.2798	0.1157	0.1555	0.2437
	0.1	0.2073	0.2342	0.3129	0.1179	0.1590	0.2494
	0.2	0.2296	0.2898	0.3856	0.2099	0.2544	0.3075

Table A.6: Misclassification errors - MAR in the independent setting (part I)

		Missingness					
		k-NN			**NB**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.0845	0.0835	0.0820	0.0634	0.0604	0.0650
	0.05	0.0768	0.0738	0.0769	0.1147	0.1149	0.1224
	0.1	0.0840	0.0850	0.0850	0.1465	0.1429	0.1542
	0.2	0.0841	0.0768	0.0860	0.2179	0.2077	0.1977
Mean Imputation	0.0	0.1266	0.1745	0.2623	0.1105	0.1616	0.2556
	0.05	0.1256	0.1652	0.2516	0.1477	0.1825	0.2685
	0.1	0.1374	0.1840	0.2733	0.1724	0.2002	0.2737
	0.2	0.1335	0.1963	0.3013	0.2212	0.2539	0.3049
Median Imputation	0.0	0.1269	0.1754	0.2624	0.1103	0.1623	0.2567
	0.05	0.1250	0.1652	0.2505	0.1482	0.1835	0.2677
	0.1	0.1302	0.1697	0.2635	0.1721	0.1976	0.2759
	0.2	0.1276	0.1838	0.2727	0.2207	0.2509	0.3035
K-NN Algorithm - Mean	0.0	0.1343	0.1860	0.2780	0.1047	0.1553	0.2472
	0.05	0.1320	0.1775	0.2655	0.1872	0.2565	0.3629
	0.1	0.1329	0.1759	0.2803	0.2236	0.2880	0.3821
	0.2	0.1252	0.1900	0.2977	0.2720	0.3678	0.4214
K-NN Algorithm - Median	0.0	0.1307	0.1819	0.2808	0.1055	0.1493	0.2492
	0.05	0.1295	0.1783	0.2695	0.1953	0.2665	0.3762
	0.1	0.1346	0.1794	0.2806	0.2303	0.2974	0.3930
	0.2	0.1205	0.1732	0.3089	0.2753	0.3620	0.4252
SEQimpute	0.0	0.1354	0.1801	0.2763	0.1103	0.1581	0.2530
	0.05	0.1326	0.1832	0.2715	0.2143	0.3059	0.4152
	0.1	0.1378	0.1883	0.2765	0.2617	0.3427	0.4490
	0.2	0.1365	0.1886	0.2922	0.3268	0.4209	0.4704
ROBimpute	0.0	0.1332	0.1824	0.2749	0.1114	0.1576	0.2529
	0.05	0.1313	0.1722	0.2668	0.1773	0.2335	0.3281
	0.1	0.1388	0.1762	0.2676	0.2111	0.2647	0.3449
	0.2	0.1236	0.1802	0.2777	0.2646	0.3225	0.3701

Table A.7: Misclassification errors - MAR in the independent setting (part II)

		Missingness					
		LDA			**Robust LDA**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.0805	0.0788	0.0814	0.0812	0.0796	0.0793
	0.05	0.1164	0.1158	0.1033	0.0832	0.0839	0.0757
	0.1	0.1187	0.1201	0.1241	0.0759	0.0776	0.0802
	0.2	0.1459	0.1440	0.1510	0.1314	0.1292	0.1364
Mean Imputation	0.0	0.1369	0.1780	0.2641	0.1368	0.1794	0.2646
	0.05	0.1707	0.2076	0.2947	0.1431	0.1866	0.2758
	0.1	0.1829	0.2381	0.3241	0.1348	0.1835	0.2982
	0.2	0.2165	0.2777	0.3534	0.1801	0.2588	0.3574
Median Imputation	0.0	0.1349	0.1796	0.2648	0.1346	0.1801	0.2666
	0.05	0.1689	0.2016	0.2838	0.1416	0.1804	0.2756
	0.1	0.1795	0.2286	0.3088	0.1306	0.1724	0.2519
	0.2	0.2125	0.2601	0.3346	0.1800	0.2237	0.2773
K-NN Algorithm - Mean	0.0	0.1196	0.1574	0.2411	0.1203	0.1564	0.2445
	0.05	0.1540	0.1873	0.2503	0.1295	0.1674	0.2326
	0.1	0.1587	0.1893	0.2700	0.1130	0.1550	0.2349
	0.2	0.1785	0.2138	0.2812	0.1644	0.2013	0.2722
K-NN Algorithm - Median	0.0	0.1198	0.1590	0.2389	0.1217	0.1595	0.2405
	0.05	0.1540	0.1868	0.2495	0.1308	0.1635	0.2286
	0.1	0.1601	0.1905	0.2686	0.1166	0.1571	0.2401
	0.2	0.1778	0.2147	0.2852	0.1651	0.2016	0.2794
SEQimpute	0.0	0.1145	0.1500	0.2308	0.1157	0.1513	0.2335
	0.05	0.1453	0.1775	0.2483	0.1200	0.1615	0.2384
	0.1	0.1564	0.1808	0.2539	0.1103	0.1491	0.2263
	0.2	0.1745	0.2079	0.2766	0.1612	0.1925	0.2640
ROBimpute	0.0	0.1141	0.1539	0.2314	0.1143	0.1551	0.2350
	0.05	0.1526	0.1949	0.2753	0.1212	0.1552	0.2291
	0.1	0.1673	0.2022	0.2820	0.1114	0.1499	0.2295
	0.2	0.2447	0.3058	0.3962	0.1585	0.1915	0.2455

Table A.8: Misclassification errors - MAR in the correlated setting (part I)

		Missingness					
		k-NN			**NB**		
Imputation meth.	ϵ	0.1	0.2	0.4	0.1	0.2	0.4
Case Deletion	0.0	0.1066	0.1045	0.1040	0.1036	0.1009	0.1029
	0.05	0.1111	0.1172	0.1203	0.3705	0.3620	0.3039
	0.1	0.0997	0.1034	0.1016	0.4201	0.4163	0.4073
	0.2	0.1112	0.1124	0.1212	0.4685	0.4575	0.4525
Mean Imputation	0.0	0.1607	0.1999	0.2828	0.1502	0.1896	0.2683
	0.05	0.1681	0.2175	0.2977	0.3591	0.3291	0.3289
	0.1	0.1606	0.2124	0.2872	0.4143	0.3867	0.3551
	0.2	0.1694	0.2144	0.2929	0.4648	0.4468	0.4113
Median Imputation	0.0	0.1593	0.1986	0.2838	0.1487	0.1901	0.2676
	0.05	0.1658	0.2168	0.2984	0.3580	0.3286	0.3287
	0.1	0.1543	0.1987	0.2735	0.4108	0.3815	0.3507
	0.2	0.1606	0.2066	0.2821	0.4623	0.4396	0.3962
K-NN Algorithm - Mean	0.0	0.1453	0.1788	0.2691	0.1403	0.1776	0.2553
	0.05	0.1585	0.2016	0.2823	0.4071	0.4110	0.4128
	0.1	0.1409	0.1837	0.2607	0.4383	0.4345	0.4277
	0.2	0.1515	0.1914	0.2739	0.4696	0.4639	0.4551
K-NN Algorithm - Median	0.0	0.1458	0.1839	0.2628	0.1404	0.1782	0.2585
	0.05	0.1598	0.1964	0.2722	0.4064	0.4085	0.4149
	0.1	0.1424	0.1820	0.2638	0.4379	0.4361	0.4288
	0.2	0.1504	0.1878	0.2787	0.4694	0.4641	0.4545
SEQimpute	0.0	0.1419	0.1797	0.2632	0.1380	0.1752	0.2517
	0.05	0.1482	0.1943	0.2767	0.4160	0.4246	0.4320
	0.1	0.1424	0.1775	0.2572	0.4430	0.4424	0.4465
	0.2	0.1499	0.1871	0.2730	0.4728	0.4686	0.4714
ROBimpute	0.0	0.1414	0.1798	0.2607	0.1383	0.1752	0.2511
	0.05	0.1472	0.1906	0.2704	0.4109	0.4236	0.4290
	0.1	0.1429	0.1789	0.2634	0.4399	0.4420	0.4466
	0.2	0.1497	0.1863	0.2660	0.4733	0.4660	0.4691

Table A.9: Misclassification errors - MAR in the correlated setting (part II)

A.4 Classification criterion plots

A.4.1 MCAR - Independent setting

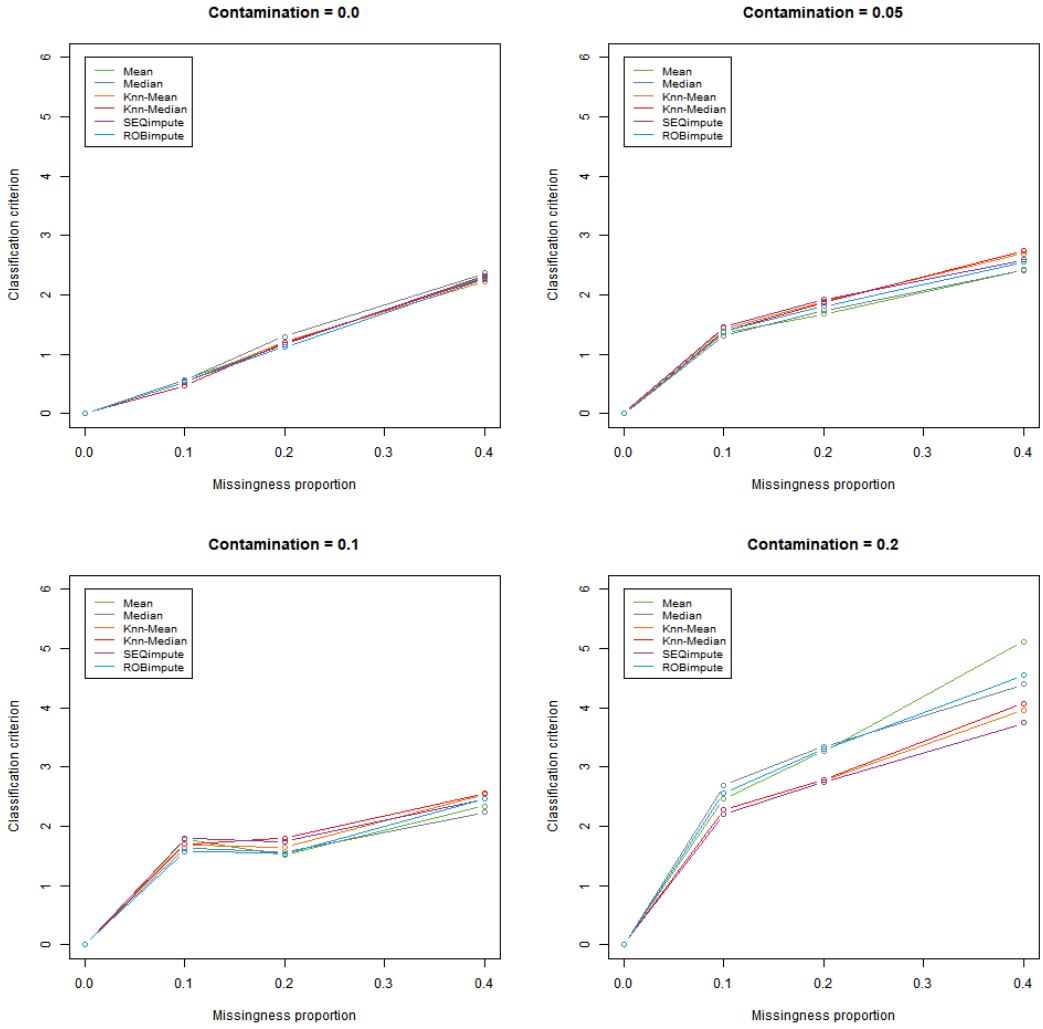


Figure A.2: Linear discrimination.

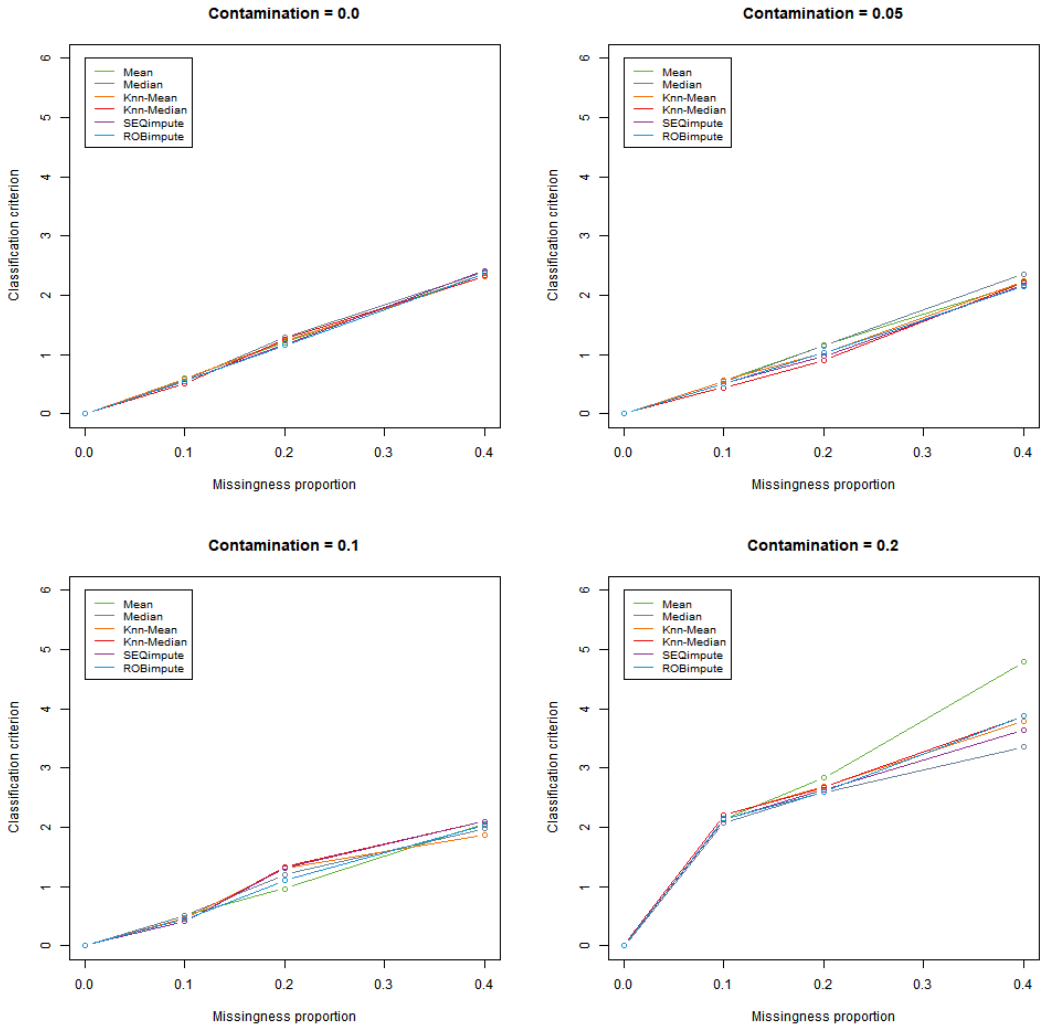


Figure A.3: Robust linear discrimination.

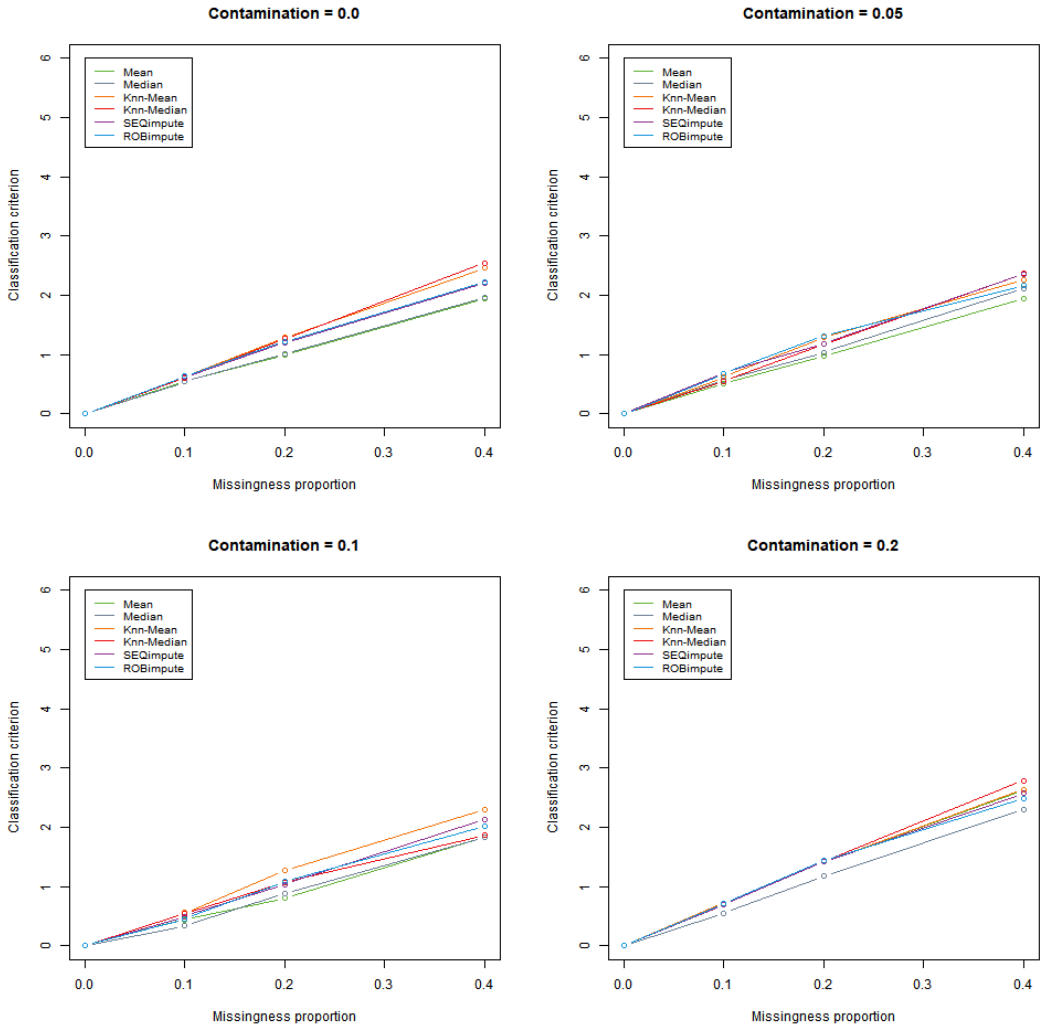


Figure A.4: K-Nearest Neighbours.

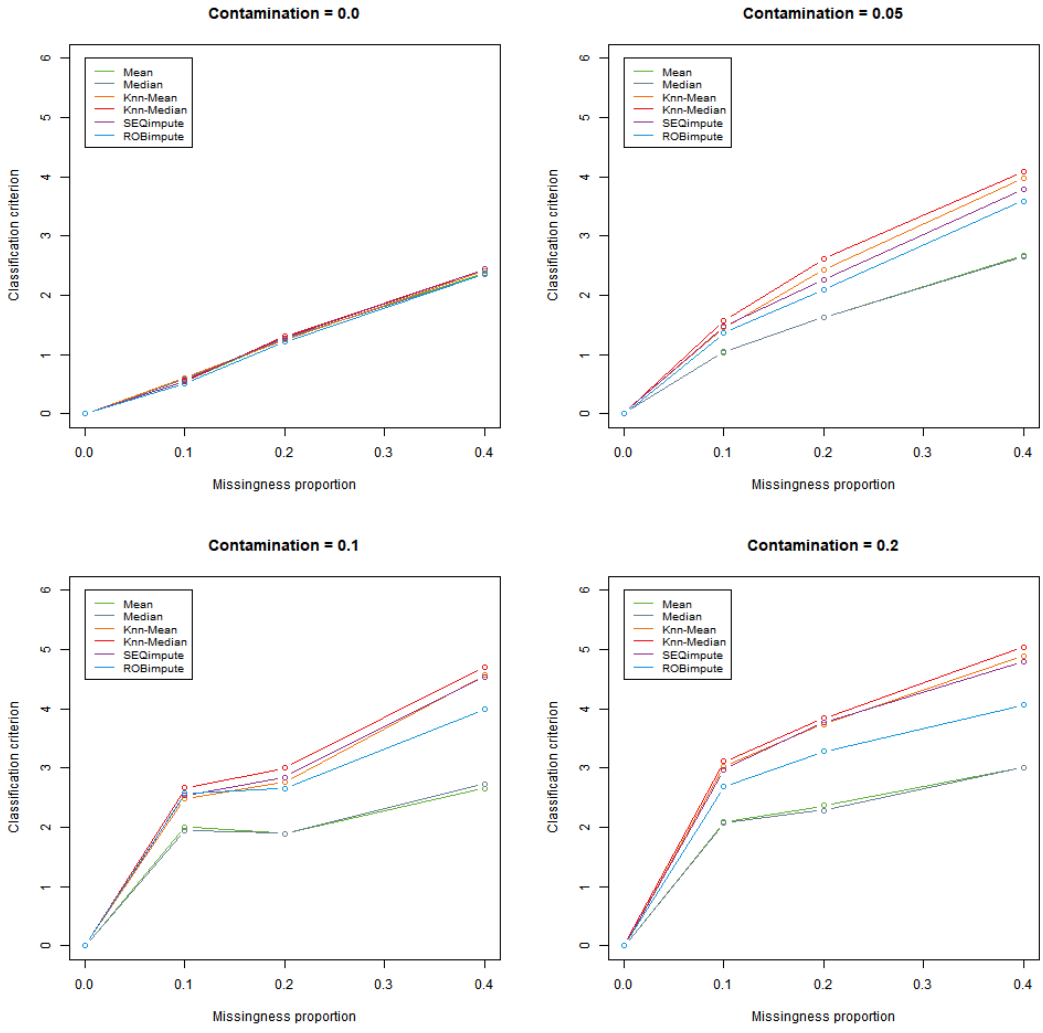


Figure A.5: Naive Bayes.

A.4.2 MCAR - Correlated setting

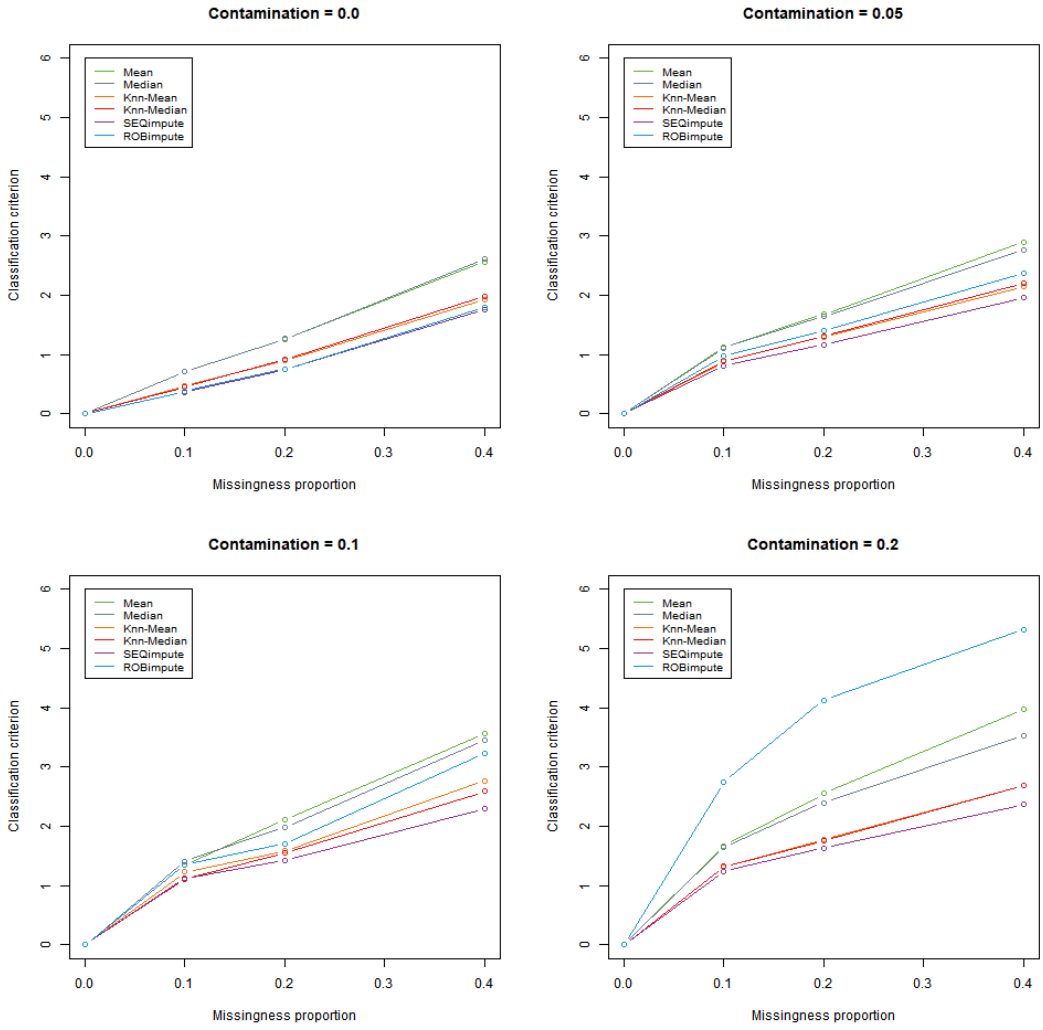


Figure A.6: Linear discrimination.

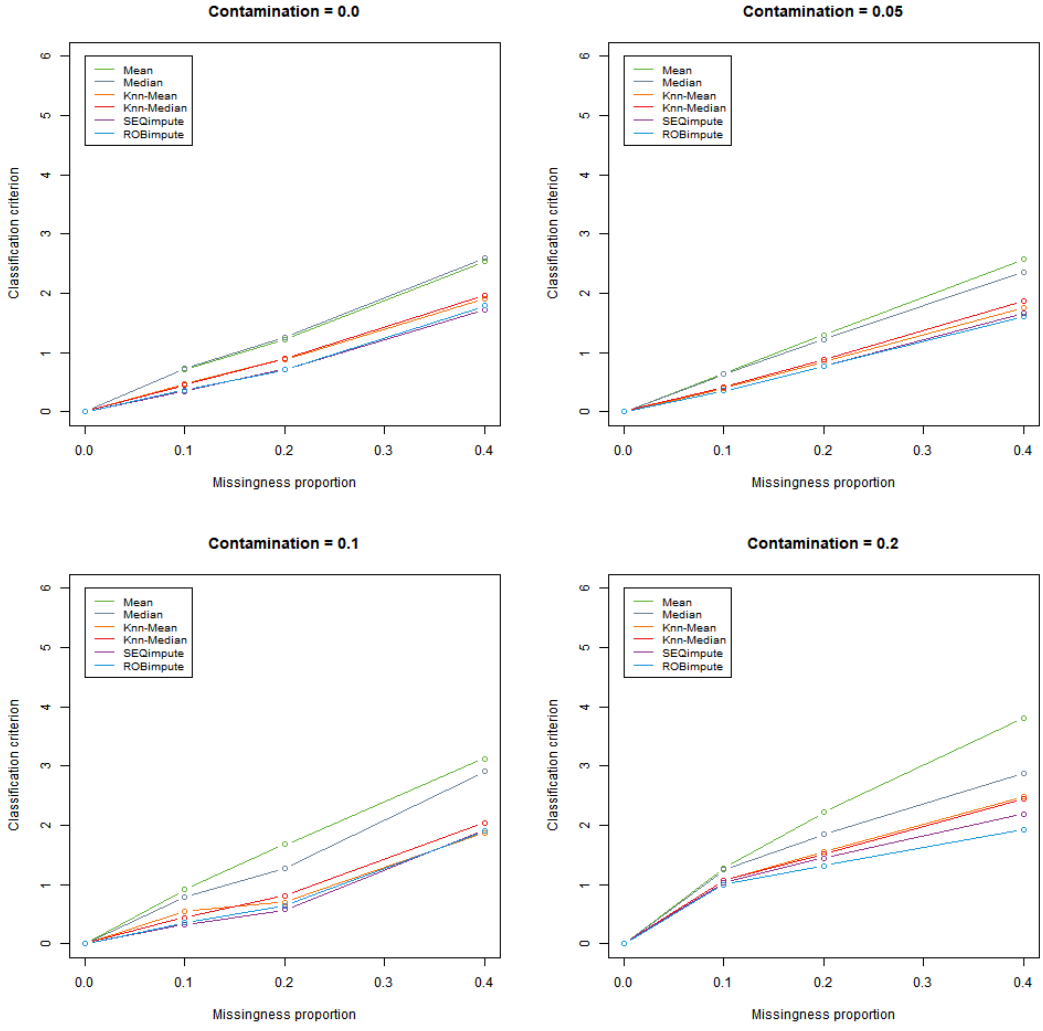


Figure A.7: Robust linear discrimination.

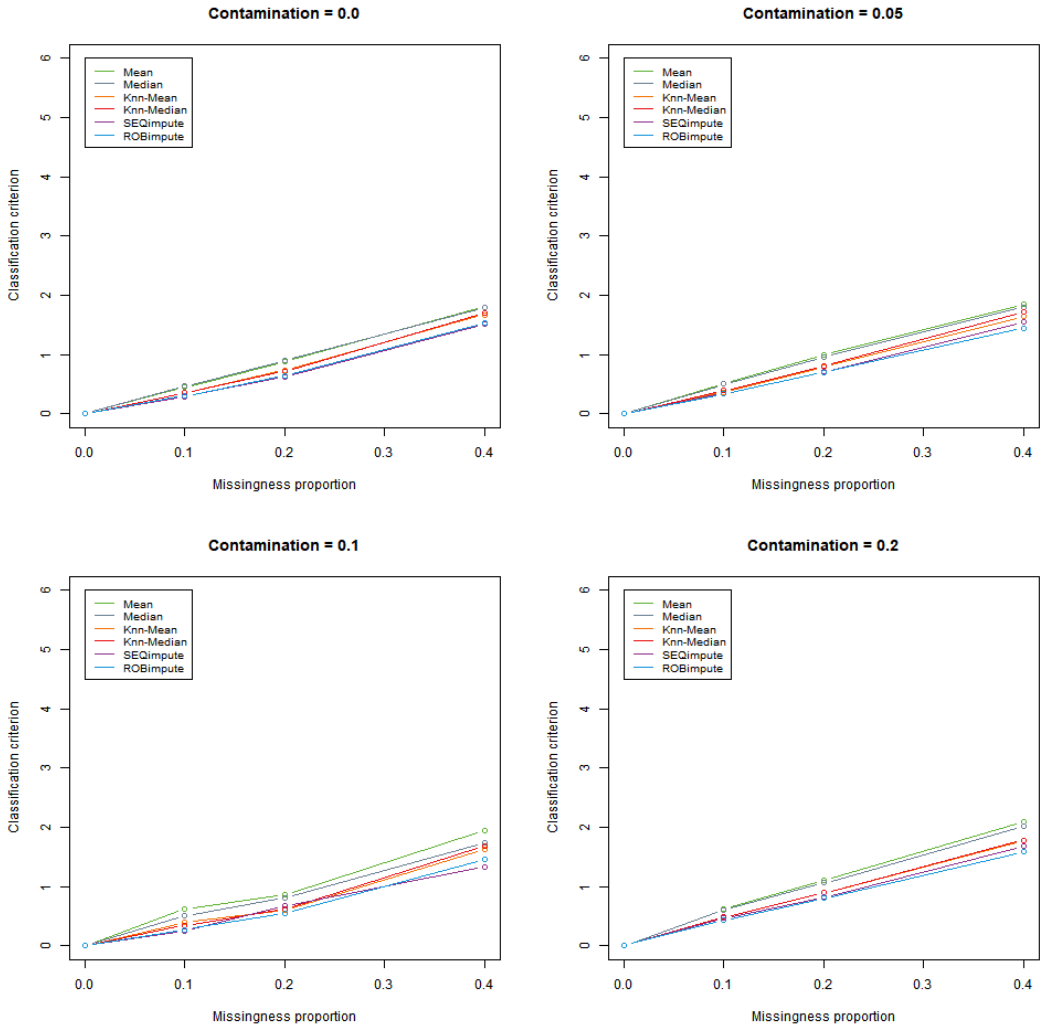


Figure A.8: K-Nearest Neighbours.

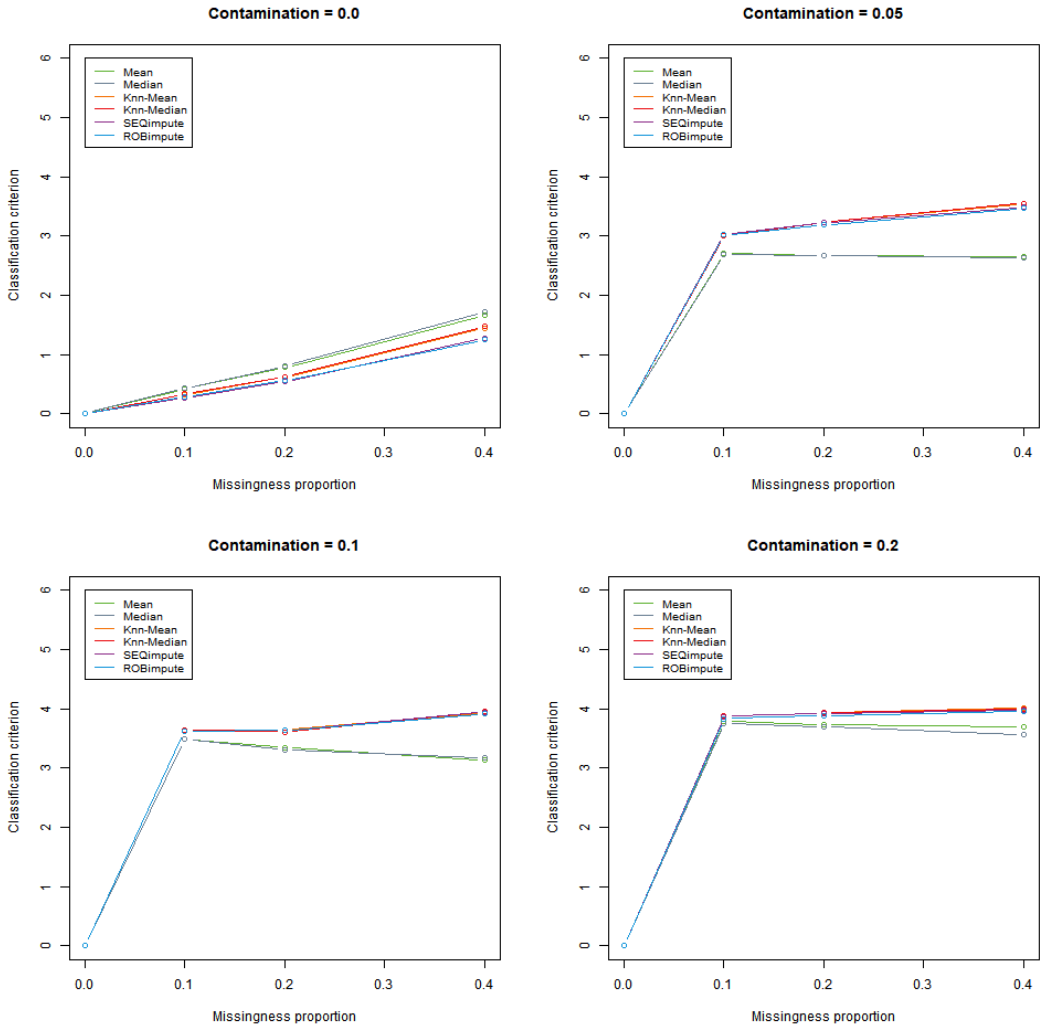


Figure A.9: Naive Bayes.

A.4.3 MAR - Independent setting

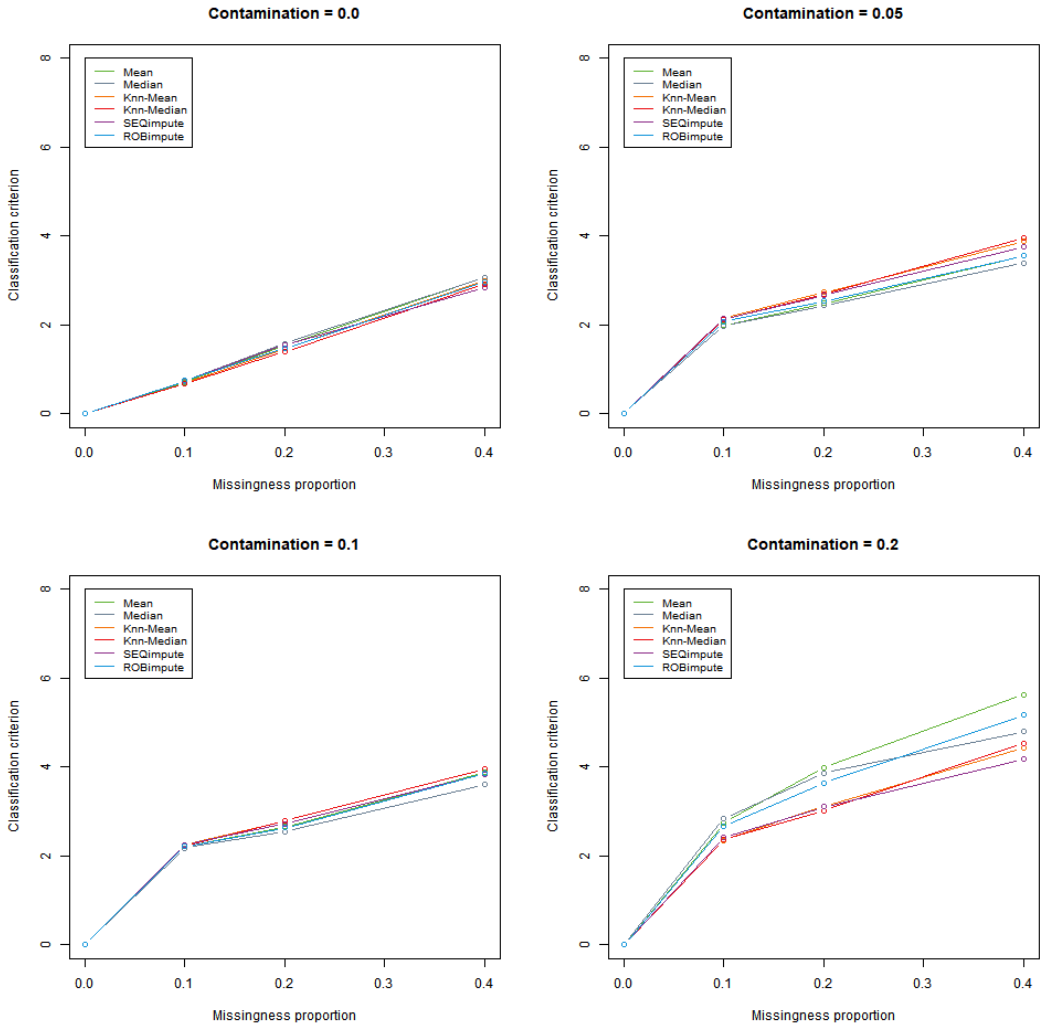


Figure A.10: Linear discrimination.

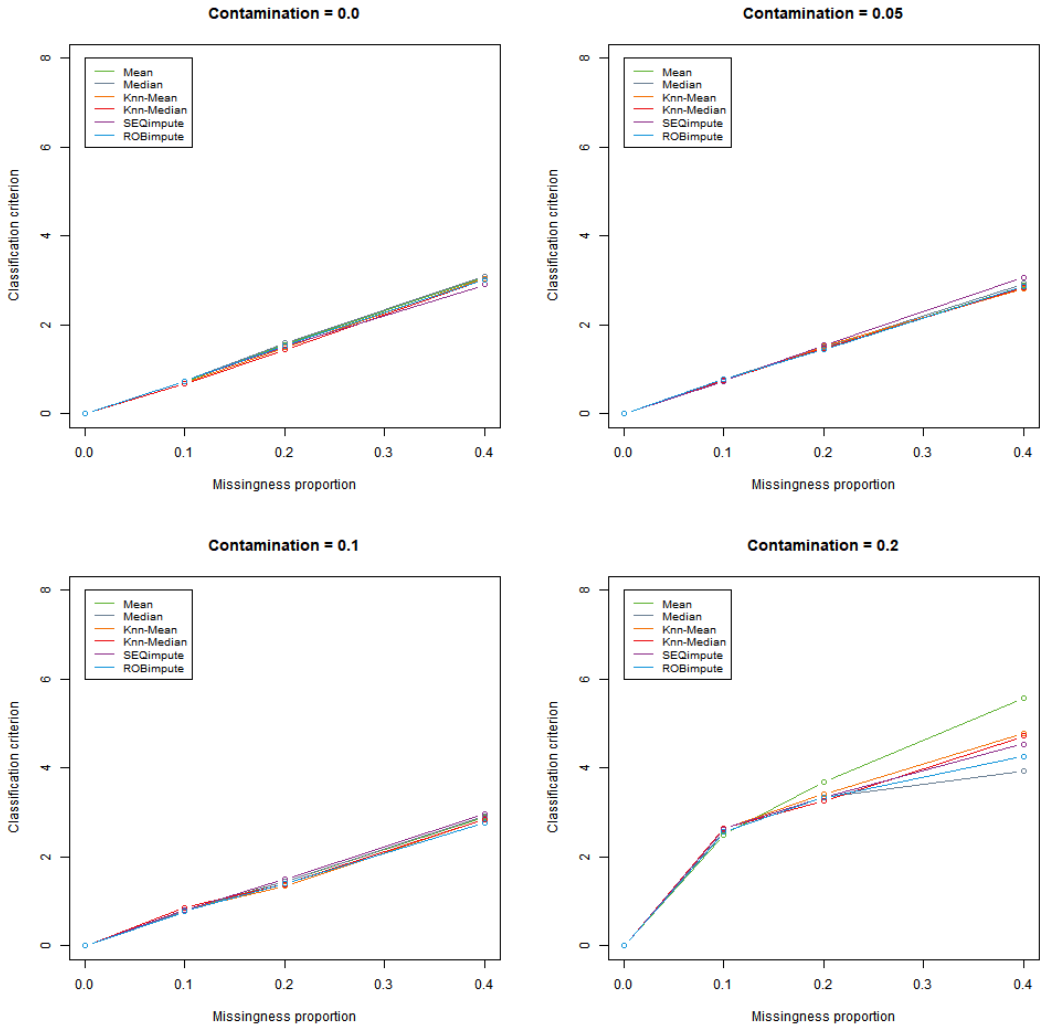


Figure A.11: Robust linear discrimination.

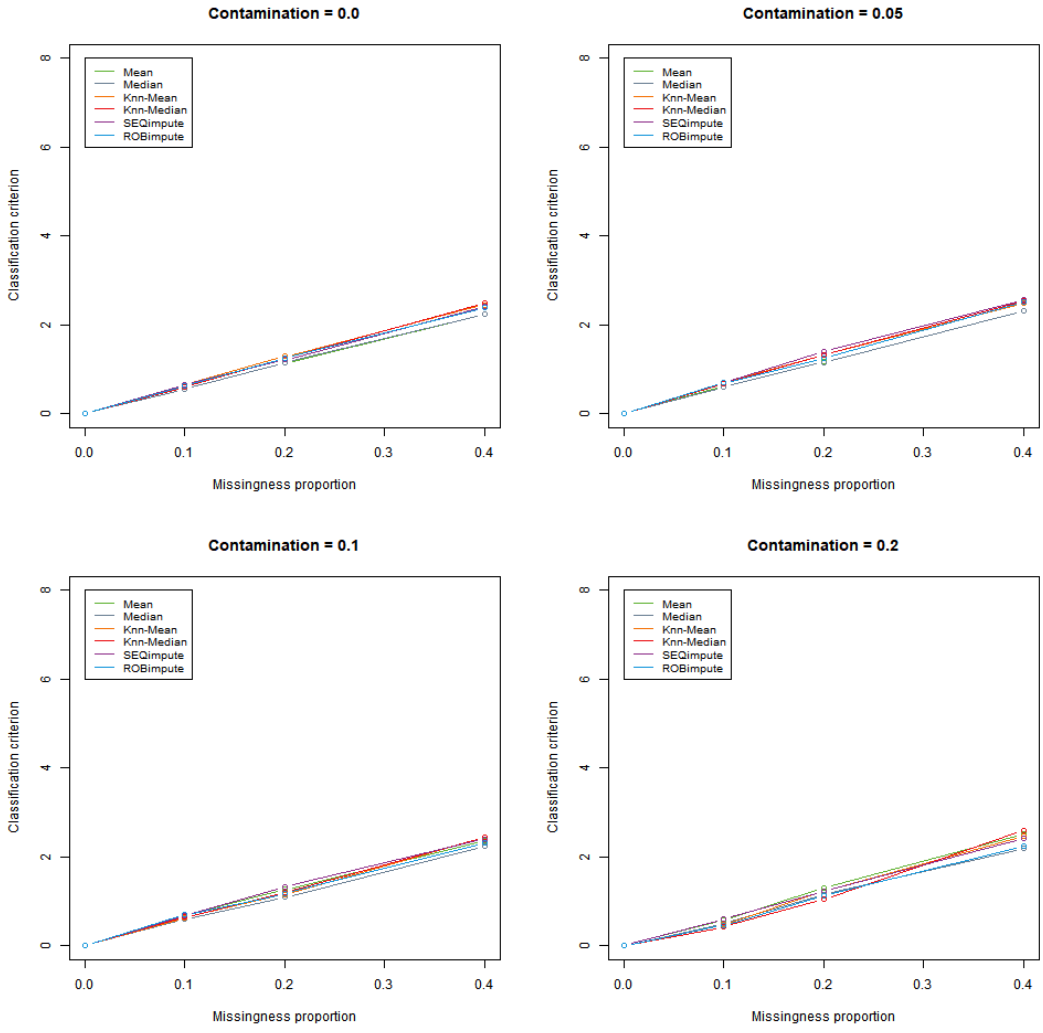


Figure A.12: K-Nearest Neighbours.

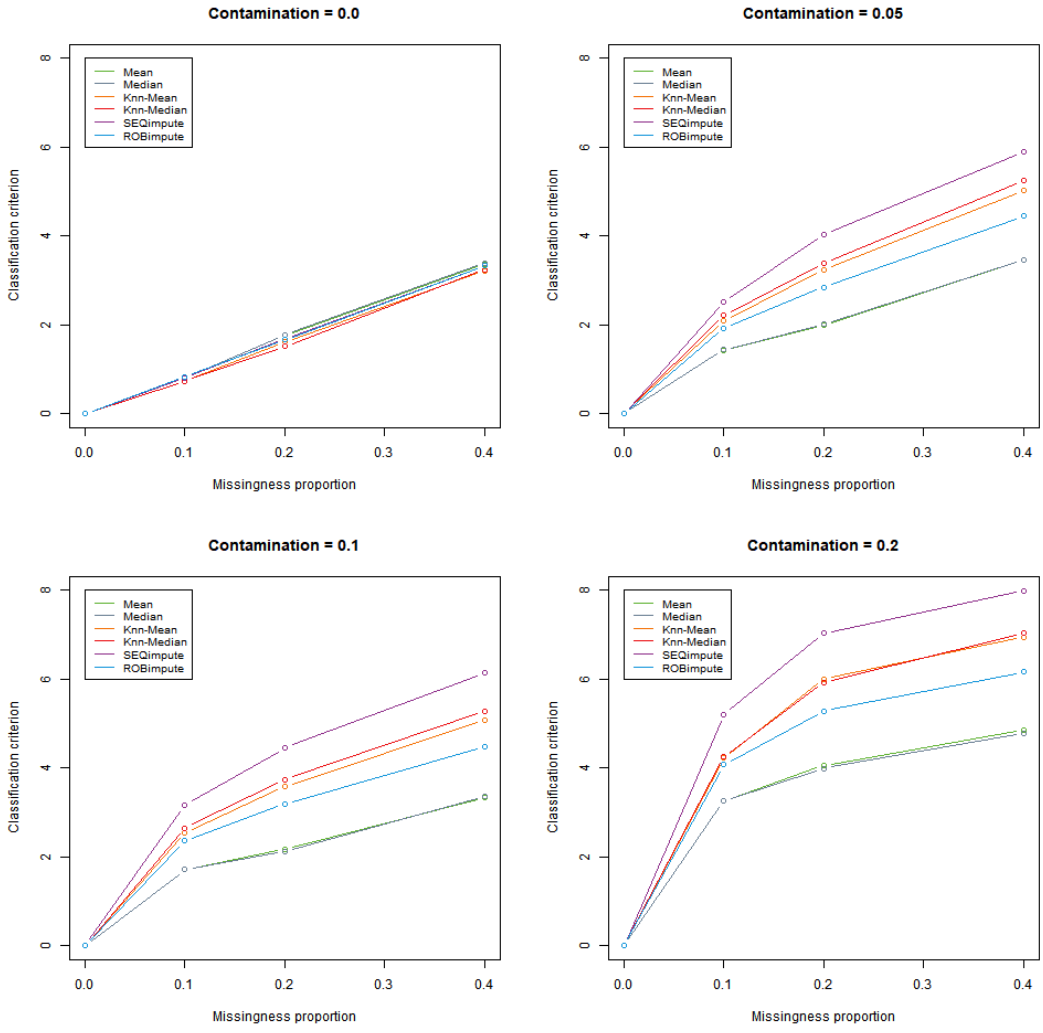


Figure A.13: Naive Bayes.

A.4.4 MAR - Correlated setting

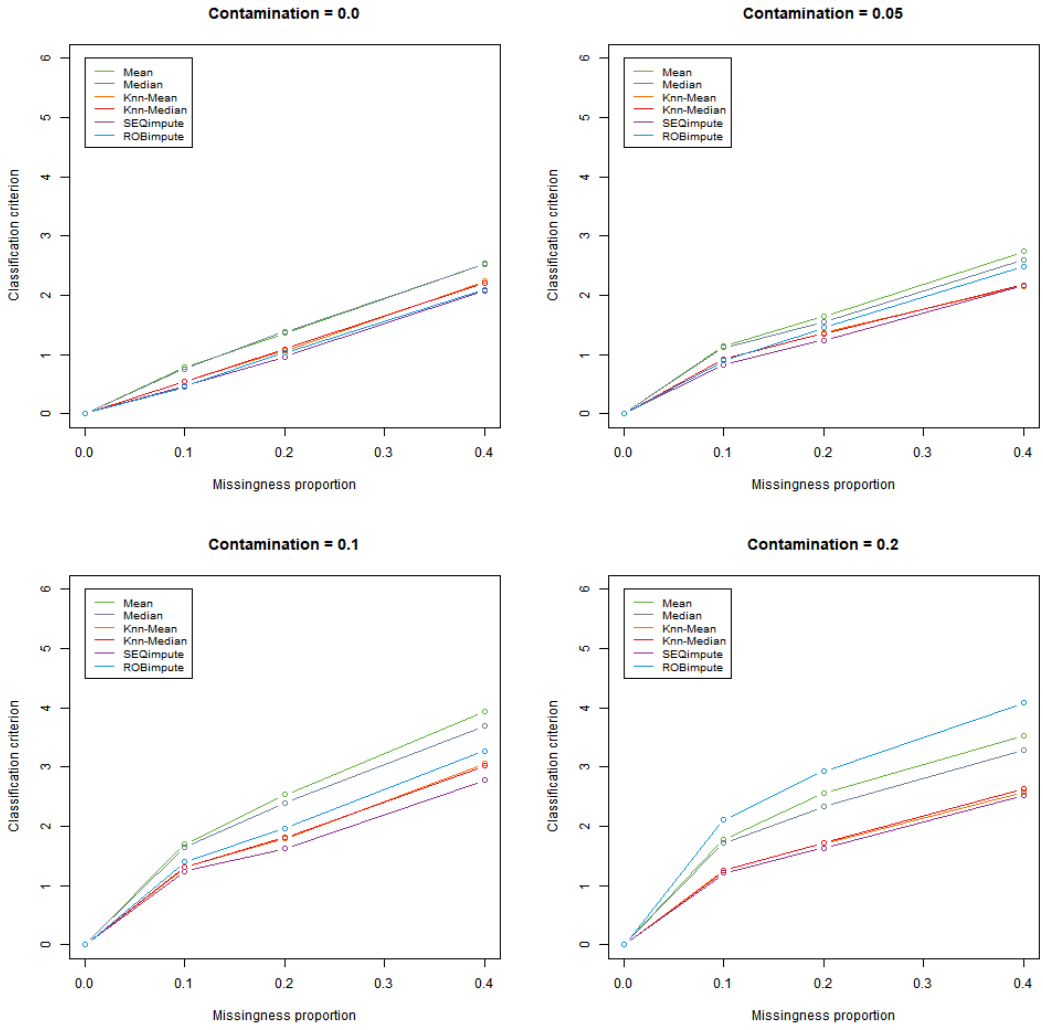


Figure A.14: Linear discrimination.

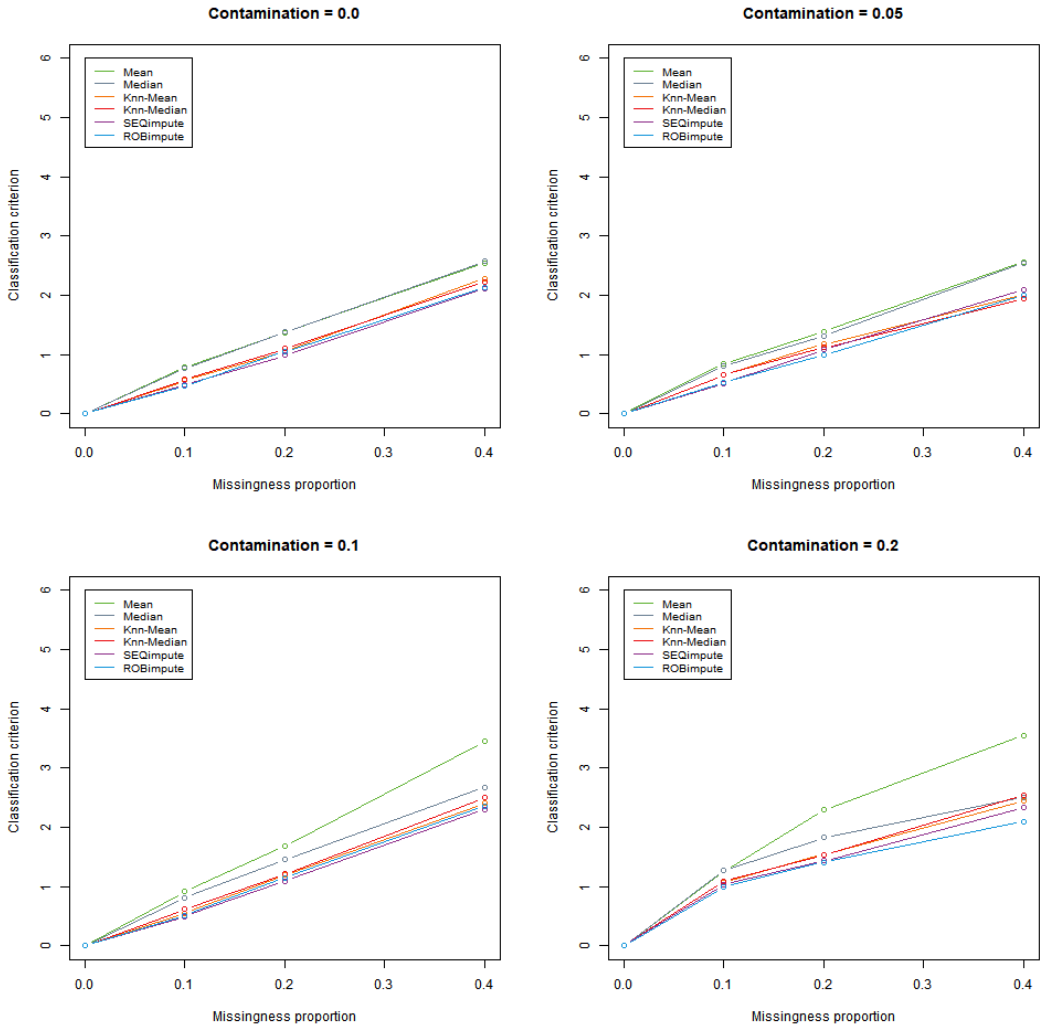


Figure A.15: Robust linear discrimination.

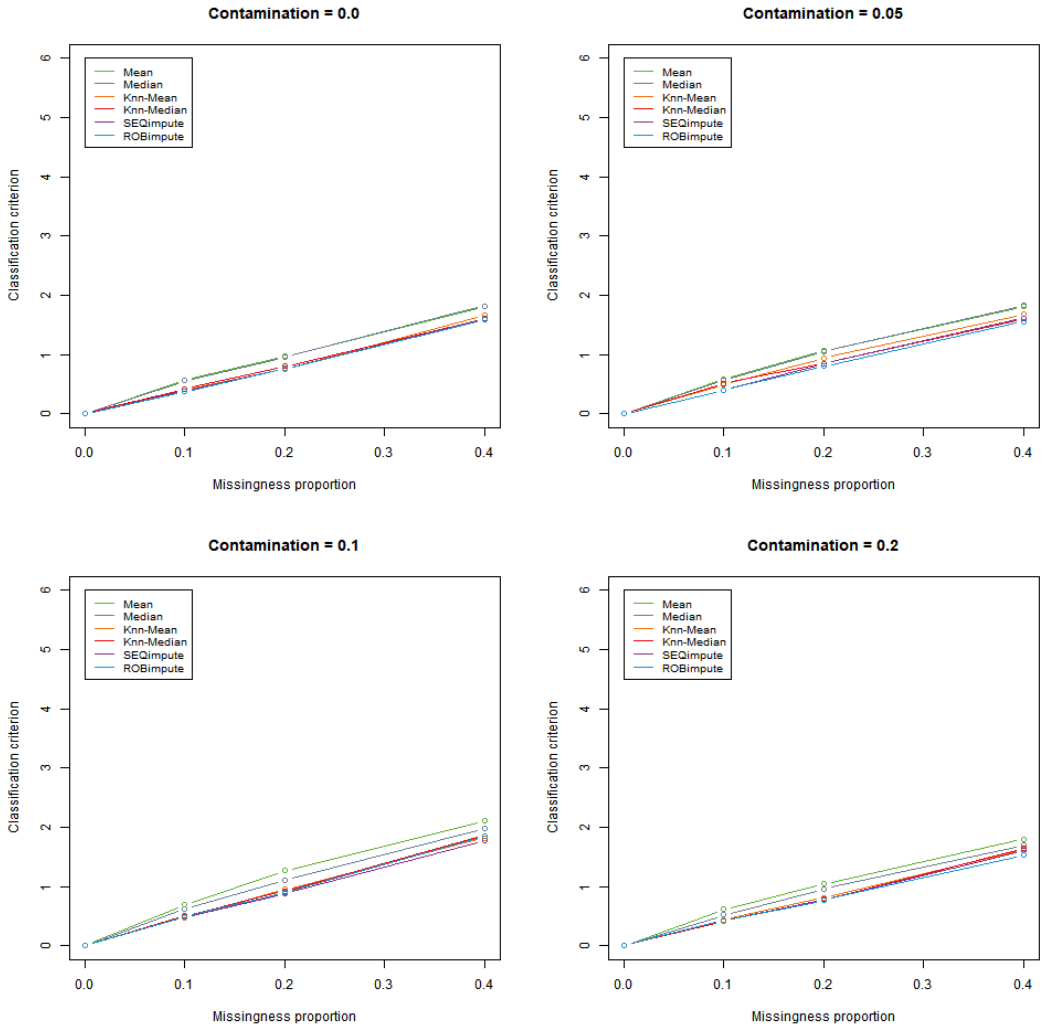


Figure A.16: K-Nearest Neighbours.

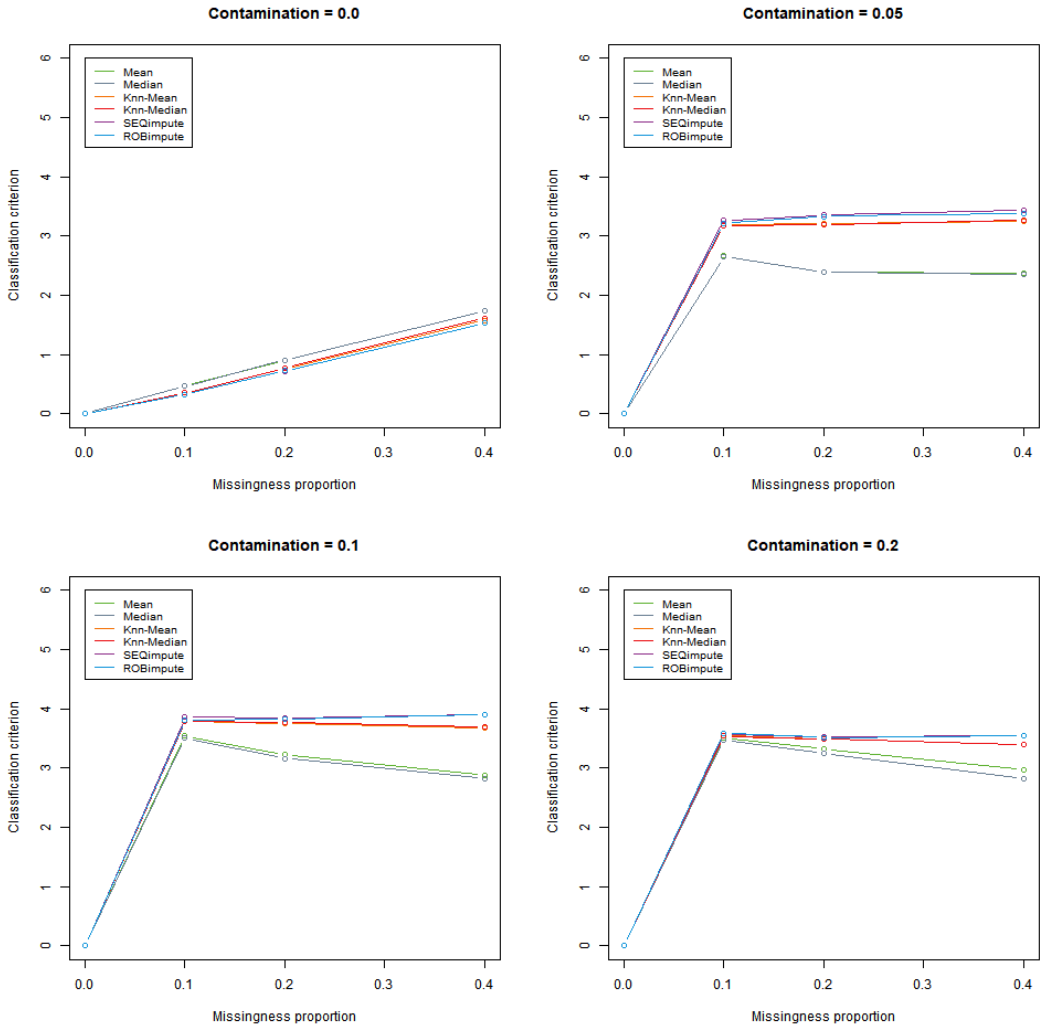


Figure A.17: Naive Bayes.